Welcome

Ken Leonard, Director
ITS Joint Program Office
Ken.Leonard@dot.gov

www.pcb.its.dot.gov
A315b:
Understanding Requirements
for Actuated Traffic Signal Controllers (ASC)
Based on NTCIP 1202 Standard
Part 1 of 2
Instructor

Kenneth L. Vaughn, P.E.
President
Trevilon LLC
Magnolia, TX, USA
Target Audience

- Traffic management and engineering staff
- TMC/operations staff
- Traffic signal maintenance staff
- System developers
- Private and public sector users including manufacturers
Recommended Prerequisite(s)

- I101: Using ITS Standards: An Overview
- A101: Introduction to Acquiring Standards-based ITS Systems
- A102: Introduction to User Needs Identification
- A201: Details On Acquiring Standards-based ITS Systems
- A202: Identifying and Writing User Needs When ITS Standards Do Not Have SEP Content
Recommended Prerequisites Continued

- A103: Introduction to ITS Standards Requirements Development
- A203: Writing Requirements When ITS Standards Do Not Have SEP Content
- C101: Introduction to the Communications Protocols and Their Uses in ITS Applications
- A315a: Understanding User Needs for Actuated Traffic Signal Controllers (ASC) Based on NTCIP 1202 Standard
Curriculum Path (Non-SEP)

I101 Using ITS Standards: An Overview

A101 Introduction to Acquiring Standards-based ITS Systems

A102 Introduction to User Needs Identification

A201 Details on Acquiring Standards-based ITS Systems

A202 Identifying and Writing User Needs When ITS Standards Do Not Have SEP Content

A103 Introduction to ITS Standards Requirements Development

A203 Writing Requirements When ITS Standards Do Not Have SEP Content

C101 Intro. to Comm. Protocols and Their Uses in ITS Applications

A315a Understanding User Needs for ASC Based on NTCIP 1202 Standard

A315b Understanding Requirements for ASC Based on NTCIP 1202 Standard (Part 1)

A315b Understanding Requirements for ASC Based on NTCIP 1202 Standard (Part 2)

T315 Applying Your Test Plan to NTCIP 1202 ASC Standard
Learning Objectives

1. Learn how to develop requirements using the NTCIP 1202 v02 standard
2. Achieve interoperability and interchangeability
3. Understand traceability
4. Develop the specification
5. Manage special issues for ASC
6. Incorporate requirements not supported by standardized objects
Learning Objective #1 – Learn How to Develop Requirements

- Review the structure of the NTCIP 1202 v02 standard
- Learn how to identify requirements from
  - User needs
  - Content from SEP-based standards
  - Conformance groups (CGs)
  - Configuration, control, and monitoring perspectives
- Review criteria for well-written requirements
- Develop sample requirements based on A315a example
Learn How to Develop Requirements

Review the Structure of NTCIP 1202 v02

Section 1: General
Section 2: Object Definitions
Section 3: Block Object Definitions
Annex A: Information Profile
Annex B: Consistency Checks
Annex C: Concept of Operations
Annex D: Deprecated Objects
What is Missing from NTCIP 1202 v02

SEP Information

- User needs (addressed in Module A315a)
- Requirements
- Message dialogs
- Traceability among these concepts and objects
How Do We Identify Requirements?

Discover Requirements from Various Sources

- PCB Module A315a
- ASC
- User Needs

- SEP-Based Standards

- NTCIP 1202 v02 Annex A

- User Needs
- Available Content
- Conformance Groups and Objects
- Config., Control, & Monitoring Analysis
Learn How to Develop Requirements

Identify from User Needs

- Development from user needs is the top-down approach
- Requirements development is:
  - Recursive
  - Iterative
  - Process of discovery
- User needs should be updated to reflect justifications
Learn How to Develop Requirements

Identify from User Needs

<table>
<thead>
<tr>
<th>User Need ID</th>
<th>User Need</th>
<th>RID</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.1</td>
<td>Architectural User Needs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.1.1</td>
<td>Live Data Exchange</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.1.2</td>
<td>Logged Data Exchange</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.1.3</td>
<td>Support Legacy Communication Networks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.2</td>
<td>Configure User Needs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.2.1</td>
<td>Safely Configure Signal for Intersection Layout</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.3</td>
<td>Control User Needs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.3.1</td>
<td>Control Selection of Timing Pattern</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.4</td>
<td>Monitor User Needs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.4.1</td>
<td>Monitor Signal Configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.4.2</td>
<td>Monitor Signal Timing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.4.3</td>
<td>Monitor Timing Pattern Selection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.4.4</td>
<td>Monitor Signal Diagnostics</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#.# Subject of this module
Learn How to Develop Requirements

Control Selection of Timing Pattern

- 2.1.3.1: The agency needs to be able to control intersection timing to accommodate the demands on the signal while also providing “green waves” to allow smooth and efficient flow of traffic through the signal system.

Requirements

- Configure timing pattern
  - Do we need multiple? How many?
- Select timing pattern
  - How should this be done? Just a manual command?
- Monitor current timing pattern
Learn How to Develop Requirements

Revise User Need

- 2.1.3.1: The agency needs to be able to control adjust intersection timing to accommodate the dynamic demands on the signal while also providing “green waves” for smooth and efficient traffic flow.

The pattern will typically be selected from a predefined list by the central system based on network conditions, but the local controller needs to be able to default to a specified schedule if any problems occur with receiving these commands, and field personnel should be able to override these commands. The controller should allow for storage of sufficient patterns for all of these purposes.
Learn How to Develop Requirements

Identify Requirements from User Needs

Control selection of timing pattern

- Configure a timing pattern
  - Support at least 32 timing patterns
- Configure timing pattern selection logic
- Activate timing pattern remotely
- Activate timing pattern per a schedule
  - Support Schedule
- Override timing pattern
Learn How to Develop Requirements

Identify from SEP-based Standards

Try not to reinvent the wheel

- Previous module demonstrated that some user needs are common among different devices
  - Example: Live data exchange
  - Can typically re-use associated requirements
- Some requirements can be copied for different uses
  - Example: Support schedule
    - Signals need schedule for timing pattern
    - Signs need schedule for message display
- Relevant text in Student Supplement
Learn How to Develop Requirements

Identify from SEP-based Standards

- Identify requirement for your signal controller
- Consider if similar requirements might exist elsewhere
- Obtain the associated standard
- Look over Protocol Requirements List (PRL) to see if it has already been defined
- Review to see if it is applicable to your need
- Reference if appropriate
  - Reference, do not “copy”
Learn How to Develop Requirements

Common User Need – Logged Data Exchange

Traced Requirements per NTCIP 1203v03 PRL

- Set Time
- Set Time Zone
- Set Daylight Savings Mode
- Verify Current Time
- Determine Current Configuration of Logging Service
- Configure Logging Service
- Retrieve Logged Data
- Clear Log
- Determine Capabilities of Event Logging Service
- ...
Learn How to Develop Requirements

Common Requirement Set – Support Schedule

Requirements for Schedule Messages for Display in NTCIP 1203v03 PRL (DMS)

- Retrieve a Schedule
- Define a Schedule
- Set Time
- Set Time Zone
- Set Daylight Savings Mode
- Verify Current Time
- Determine Maximum Number of Schedules
- Monitor Current Schedule
- Supplemental Requirements for Scheduling
Learn How to Develop Requirements

Identify from Conformance Groups

- Development from conformance groups (CGs) is the bottom-up approach
- Identify relevant CG(s)
- Look at objects in those CG(s)
- Identify those objects that:
  - You did not think of yet
  - You do not intuitively understand
- Investigate those objects further
- Determine if the functionality is needed for your project
- Revise user need documentation, as needed
Learn How to Develop Requirements

Identify from Conformance Groups

<table>
<thead>
<tr>
<th>Clause</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>Coordination Conformance Group</td>
</tr>
<tr>
<td>2.5.2</td>
<td>coordCorrectionMode</td>
</tr>
<tr>
<td></td>
<td>Other (1)</td>
</tr>
<tr>
<td></td>
<td>Dwell (2)</td>
</tr>
<tr>
<td></td>
<td>Shortway (3)</td>
</tr>
<tr>
<td></td>
<td>AddOnly (4)</td>
</tr>
</tbody>
</table>

- How should the controller transition to the new timing pattern?
  - If you don’t care, no need to specify
  - If you do, you need to specify minimal options
Learn How to Develop Requirements

Identify from Conformance Groups

Control selection of timing pattern

- Configure timing pattern transition mechanism
Learn How to Develop Requirements

Identify from 3 Perspectives

- As a final check, consider the three perspectives again:
  - Configure
  - Control
  - Monitor

- May overlap with other user needs
  - Goal is to ensure all angles are covered
  - Duplicate requirements can be identified and re-organized; missing requirements are the real problem
Learn How to Develop Requirements

Identify from 3 Perspectives

Control Selection of Timing Pattern

- Monitor Perspective
  - Monitor timing pattern configuration
  - Monitor current timing pattern
  - Monitor last timing pattern requested
  - Monitor source of last timing pattern
  - Monitor transition mechanism selected
  - Monitor timing pattern schedule

- Likely overlaps with 2.1.4.3 “Monitor Timing Pattern Selection”
Learn How to Develop Requirements

Only Define Real Requirements

- 24 potential requirements for the “Control Selection of Timing Pattern” user need
  - They could easily be restated to be more or less
- Every requirement should trace to a user need
- Only use requirements that apply to your project
  - Minimize costs to develop specification
  - Minimize costs for product
  - Minimize costs for testing
  - Minimize effort for future upgrades
Importance of Considering All Sources

An Incomplete Specification

- Agency may get a “standard-conformant” system that does not meet project needs
- Agency may get a “specification-compliant” system that does not meet project needs
- Agency may get a “non-conformant” system that uses proprietary mechanisms
- Agency may get an “incomplete” system that only partially implements functionality
Learn How to Develop Requirements

Criteria for Well-Written Requirements

[Actor] [Action] [Target] [Constraint] [Localization]

- **Actor**: Who or what does the action
- **Action**: Identifies what is to happen
- **Target**: Identifies who or what receives the action
- **Optional**:
  - Constraint: Identifies how to measure success or failure
  - Localization: Identifies circumstances under which requirement applies
Criteria for Well-Written Requirements

Context Diagram for Actor

Central System → Request → ASC → Response
Learn How to Develop Requirements

Criteria for Well-Written Requirements

- Necessary
- Concise
- Attainable
- Standalone
- Consistent
- Unambiguous
- Verifiable
Learn How to Develop Requirements

Requirement from NTCIP 1203

Set Time: The central system shall configure the ASC with the current coordinated universal time (UTC) to the nearest second.

- Actor: “the central system”
- Action: “shall configure”
- Target: “the ASC”
- Constraint: “with the current UTC to the nearest second”
- Localization: none
Learn How to Develop Requirements

Requirement from User Need

Monitor Current Timing Pattern: The central system shall monitor the ASC to determine which timing pattern is currently active.

- Actor: “the central system”
- Action: “shall monitor”
- Target: “the ASC”
- Constraint: “to determine which timing pattern is currently active”
- Localization: none
Learn How to Develop Requirements

Requirement from User Need

Configure a Timing Pattern: When requested by the operator, the central system shall configure the ASC with timing pattern information subject to ASC-imposed validation rules.

- Actor: “the central system”
- Action: “shall configure”
- Target: “the ASC”
- Constraint: “with timing pattern information subject to ASC-imposed validation rules”
- Localization: “when requested by the operator”

“Pattern” is a defined term in NTCIP 1202
Which of the following statements is NOT true?

Answer Choices

a) User needs are used in a top-down approach to identify requirements

b) You should read every SEP-based standard in order to get ideas for requirements

c) Conformance groups (CGs) are used in a bottom-up approach to identify requirements

d) You may discover overlaps in requirements from different user needs
Review of Answers

a) User needs are used in a top-down approach
   True! This is the traditional systems engineering approach to defining requirements.

b) Read every SEP-based standard to get ideas
   False. You should reference SEP-based standards when you think they may be of benefit.

c) CGs are used in a bottom-up approach
   True! Reverse engineering allows you to benefit from a known solution.

d) There may be overlaps in requirements
   True! Harmonization of this issue is a later topic.
Summary of Learning Objective # 1

Learn How to Develop Requirements

- Reviewed structure of standard
- Discussed rules for identifying requirements
  - Identified 28 sample requirements
- Discussed rules for writing requirements
  - Provided a sample referenced requirement
  - Provided 2 sample new requirements
Learning Objective # 2 – Achieve Interoperability and Interchangeability

- Review terminology definitions
- Understand SNMP interface and dialogs
- Understand NTCIP objects (including reference to NTCIP 1201)
- Develop sample dialogs
- Prepare sample specification text
Achieve Interoperability/Interchangeability

Interoperability
The ability of two or more systems or components to exchange information and to use the information that has been exchanged

ISO/IEC 24765:2010

Systems and Software Engineering – Vocabulary
Achieve Interoperability/Interchangeability

Interchangeability

A condition which exists when two or more items possess such functional and physical characteristics as to be equivalent in performance and durability, and are capable of being exchanged one for the other without alteration of the items themselves, or adjoining items, except for adjustment, and without selection for fit and performance.

_Nat’l Telecommunications & Information Admin._

_US Department of Commerce_
Achieve Interoperability/Interchangeability

Rules for exchanging information:

- Dialogs
  - Simple Network Management Protocol (SNMP)
  - Simple Transportation Management Protocol (STMP)
- Messages

Rules for using the information:

- Objects
- Object Ranges
Achieve Interoperability/Interchangeability

Basic SNMP Dialogs and Messages

GET

GET-NEXT

SET
Achieve Interoperability/Interchangeability

Monitor Current Timing Pattern

Find Objects

- Annex A.2 of NTCIP 1202 v02 lists conformance groups
- Each conformance group identifies a list of objects
  - A name that indicates function
  - A clause number to learn more
- Object defined in computer-readable format
Achieve Interoperability/Interchangeability

Monitor Current Timing Pattern

Find Objects

<table>
<thead>
<tr>
<th>NTCIP 1202 Clause</th>
<th>Object Name</th>
<th>Object Type</th>
<th>Object Status</th>
<th>Object Support</th>
<th>Allowed Values</th>
<th>Supported Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.9.5</td>
<td>splitCoordPhase</td>
<td>P</td>
<td>2.5 : M</td>
<td>Yes</td>
<td>Yes / No</td>
<td>0-1</td>
</tr>
<tr>
<td>2.5.10</td>
<td>coordPatternStatus</td>
<td>S</td>
<td>2.5 : M</td>
<td>Yes</td>
<td>Yes</td>
<td>0-255</td>
</tr>
<tr>
<td>2.5.11</td>
<td>localFreeStatus</td>
<td>S</td>
<td>2.5 : M</td>
<td>Yes</td>
<td>Yes</td>
<td>1-11</td>
</tr>
<tr>
<td>2.5.12</td>
<td>coordCycleStatus</td>
<td>S</td>
<td>2.5 : M</td>
<td>Yes</td>
<td>Yes</td>
<td>0-510</td>
</tr>
<tr>
<td>2.5.13</td>
<td>coordSyncStatus</td>
<td>S</td>
<td>2.5 : M</td>
<td>Yes</td>
<td>Yes</td>
<td>0-510</td>
</tr>
<tr>
<td>2.5.14</td>
<td>systemPatternControl</td>
<td>C</td>
<td>2.5 : M</td>
<td>Yes</td>
<td>Yes</td>
<td>0-255</td>
</tr>
<tr>
<td>2.5.15</td>
<td>systemSyncControl</td>
<td>C</td>
<td>2.5 : M</td>
<td>Yes</td>
<td>Yes</td>
<td>0-255</td>
</tr>
</tbody>
</table>

NTCIP 1202v02 Page 140
Achieve Interoperability/Interchangeability

Monitor Current Timing Pattern

Find Objects

coordPatternStatus OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS optional
DESCRIPTION
"<Definition> This object defines the running coordination pattern/mode in the device. The possible values are:

Value Description
0 Not used
1-253 Pattern - indicates the currently running pattern
254 Free - indicates Free operation without coordination.
255 Flash - indicates Automatic Flash without coordination.

<DescriptiveName> NTCIP-1202::ASC.coordPatternStatus
<DataConceptType> Data Element"

 ::= { coord 10 }
Achieve Interoperability/Interchangeability

Monitor Current Timing Pattern

Dialog Definition
“The management station shall GET coordPatternStatus.0”
Achieve Interoperability/Interchangeability

Dialog for a Referenced Requirement

Set Time

- Don’t reinvent the wheel
- NTCIP 1203 (DMS) dialog for Set Time works for ASC as well
  - Just reference
Achieve Interoperability/Interchangeability

Configure a Timing Pattern

Coordination Conformance Group

<table>
<thead>
<tr>
<th>Clause</th>
<th>Object Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>Coordination Conformance Group</td>
</tr>
<tr>
<td>2.5.7</td>
<td>patternTable</td>
</tr>
<tr>
<td>2.5.7.1</td>
<td>patternNumber</td>
</tr>
<tr>
<td>2.5.7.2</td>
<td>patternCycleTime</td>
</tr>
<tr>
<td>2.5.7.3</td>
<td>patternOffsetTime</td>
</tr>
<tr>
<td>2.5.7.4</td>
<td>patternSplitNumber</td>
</tr>
<tr>
<td>2.5.7.5</td>
<td>patternSequenceNumber</td>
</tr>
<tr>
<td>2.5.9</td>
<td>splitTable</td>
</tr>
<tr>
<td>2.5.9.1</td>
<td>splitNumber</td>
</tr>
<tr>
<td>2.5.9.2</td>
<td>splitPhase</td>
</tr>
<tr>
<td>2.5.9.3</td>
<td>splitTime</td>
</tr>
<tr>
<td>2.5.9.4</td>
<td>splitMode</td>
</tr>
<tr>
<td>2.5.9.5</td>
<td>splitCoordPhase</td>
</tr>
</tbody>
</table>
Achieve Interoperability/Interchangeability

Configure a Timing Pattern

(Precondition) Verify that the ASC supports the desired splits, phases, and patterns.

Repeat for each phase

Set(splitTime.x,y, splitMode.x,y, splitCoordPhase.x,y)

Set(patternCycleTime.z, patternOffsetTime.z, patternSplitNumber.z, patternSequenceNumber.z)

x = splitNumber
y = splitPhase
z = patternNumber
Achieve Interoperability/Interchangeability

Configure a Timing Pattern

1. (Precondition) The management station shall confirm that the signal controller supports the desired splits, phases, and patterns.

2. For each enabled phase, repeat step 3.

3. The management station shall SET the following objects to the desired values:
   a) splitTime.x.y
   b) splitMode.x.y
   c) splitCoordPhase.x.y

Where x = splitNumber and y = phaseNumber.
Achieve Interoperability/Interchangeability

Configure a Timing Pattern

4. The management station shall SET the following objects to the desired values:

   a) patternCycleTime.z
   b) patternOffsetTime.z
   c) patternSplitNumber.z
   d) patternSequenceNumber.z

Where \( z = \text{patternNumber} \)
Achieve Interoperability/Interchangeability

What are the Implications?

- Procurement specification must specify all dialogs
  - Otherwise there is a high probability that incompatibilities will arise
- Dialog must be supported by “management station” and device
- Dialog definition is a low-level design issue
  - Requires detailed expertise to define
Achieve Interoperability/Interchangeability

What are the Implications?

- Existing products already implement dialogs
  - Don’t break what already works
- Developer of “management station” is best positioned to define the actual dialog
  - Developer needs to be selected prior to finalizing procurement specification
  - Care must be taken when upgrading “management station” to not break existing deployment
Why should dialogs be defined in a procurement specification?

a) Devices are more likely to conform with the standard
b) Devices are more likely to interoperate with the central system
c) Devices are more likely to be interchangeable with other devices using the same procurement
d) Devices are likely to be less expensive
Review of Answers

a) Likely to conform to the standard
   *False. The dialog will be separate from the standard and will not affect conformance.*

b) Likely to interoperate with the central system
   *True! Dialogs promote a common expectation on how objects are to be exchanged.*

c) Likely to be interchangeable with other devices using the same procurement
   *True! Dialogs promote a common expectation on how objects are to be exchanged.*

d) Likely to be less expensive
   *False. Additional specifications will likely mean added costs for the controller, but integration costs should significantly decrease.*
Summary of Learning Objective # 2

Achieve Interoperability and Interchangeability

- Reviewed terminology
- Discussed dialogs and objects
- Prepared sample dialogs
- Prepared sample text for dialogs
Learning Objective # 3 – Understand Traceability

- User need to requirements traceability
- Requirements to design traceability
- Benefits of documenting traceability
# Understand Traceability

## Needs to Requirements

<table>
<thead>
<tr>
<th>UN ID</th>
<th>User Need</th>
<th>RID</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.1</td>
<td>Architectural User Needs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.1.1</td>
<td>Live Data Exchange</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.1.2</td>
<td>Logged Data Exchange</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.1.3</td>
<td>Support Legacy Communication Networks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.2</td>
<td>Configure User Needs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.2.1</td>
<td>Safely Configure Signal for Intersection Layout</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.3</td>
<td>Control User Needs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.3.1</td>
<td>Control Selection of Timing Pattern</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#.#</td>
<td>See module A315b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.4</td>
<td>Monitor User Needs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.4.1</td>
<td>Monitor Signal Configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.4.2</td>
<td>Monitor Signal Timing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.4.3</td>
<td>Monitor Timing Pattern Selection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.4.4</td>
<td>Monitor Signal Diagnostics</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Understand Traceability

### Needs to Requirements

<table>
<thead>
<tr>
<th>UN ID</th>
<th>User Need</th>
<th>RID</th>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.3.1</td>
<td>Control Selection of Timing Pattern</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.2.1.1</td>
<td></td>
<td>Configure a Timing Pattern</td>
</tr>
<tr>
<td></td>
<td>2.2.4.1</td>
<td></td>
<td>Support at Least 32 Timing Patterns</td>
</tr>
<tr>
<td></td>
<td>2.2.1.2</td>
<td></td>
<td>Configure Timing Pattern Selection Logic</td>
</tr>
<tr>
<td></td>
<td>2.2.2.1</td>
<td></td>
<td>Activate Timing Pattern Remotely</td>
</tr>
<tr>
<td></td>
<td>2.2.2.2</td>
<td></td>
<td>Activate a Timing Pattern per a Schedule</td>
</tr>
<tr>
<td></td>
<td>2.2.2.3</td>
<td></td>
<td>Override Timing Pattern</td>
</tr>
<tr>
<td></td>
<td>2.2.3.6</td>
<td></td>
<td>Retrieve a Schedule</td>
</tr>
<tr>
<td></td>
<td>2.2.2.8</td>
<td></td>
<td>Define a Schedule</td>
</tr>
<tr>
<td></td>
<td>NTCIP 1203v03 H.2.2.1</td>
<td></td>
<td>Set Time</td>
</tr>
<tr>
<td></td>
<td>NTCIP 1203v03 H.2.2.2</td>
<td></td>
<td>Set Time Zone</td>
</tr>
<tr>
<td></td>
<td>NTCIP 1203v03 H.2.2.3</td>
<td></td>
<td>Set Daylight Savings Mode</td>
</tr>
<tr>
<td></td>
<td>NTCIP 1203v03 H.2.2.4</td>
<td></td>
<td>Verify Current Time</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
Understand Traceability

Contrast Needs-to-Requirements and PRL

- Several columns are not needed
  - Conformance
  - Support
  - Additional project requirements
Understand Traceability

Requirements to Design

- Identical to Requirements Traceability Matrix (RTM)
  - Trace requirements to dialog
  - Trace requirements to objects
Understand Traceability

Requirements to Design

<table>
<thead>
<tr>
<th>RID</th>
<th>Requirement</th>
<th>Dialog</th>
<th>Object ID</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.3.10</td>
<td>Monitor Current Timing Pattern</td>
<td>NTCIP 1203 G.1</td>
<td>NTCIP 1202 2.5.10</td>
<td>coordPatternStatus</td>
</tr>
</tbody>
</table>
Understand Traceability

Complex Example – Configure a Timing Pattern

(Precondition) Verify that the ASC supports the desired splits, phases, and patterns.

Repeat for each phase

Set(splitTime.x.y
  splitMode.x.y
  splitCoordPhase.x.y)

Set(patternCycleTime.z
  patternOffsetTime.z
  patternSplitNumber.z
  patternSequenceNumber.z)

x = splitNumber
y = splitPhase
z = patternNumber
**Understand Traceability**

### Requirements to Design

<table>
<thead>
<tr>
<th>RID</th>
<th>Requirement</th>
<th>Dialog</th>
<th>Object ID</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Configure a Timing Pattern</td>
<td>2.3.1</td>
<td>NTCIP 1202 2.5.7.1</td>
<td>patternNumber</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NTCIP 1202 2.5.7.2</td>
<td>patternCycleTime</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NTCIP 1202 2.5.7.3</td>
<td>patternOffsetTime</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NTCIP 1202 2.5.7.4</td>
<td>patternSplitNumber</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NTCIP 1202 2.5.7.5</td>
<td>patternSequenceNumber</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NTCIP 1202 2.5.9.1</td>
<td>splitNumber</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NTCIP 1202 2.5.9.2</td>
<td>splitPhase</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NTCIP 1202 2.5.9.3</td>
<td>splitTime</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NTCIP 1202 2.5.9.4</td>
<td>splitMode</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NTCIP 1202 2.5.9.5</td>
<td>splitCoordPhase</td>
</tr>
</tbody>
</table>
Understand Traceability

Benefits of Traceability

- Quickly identify:
  - The purpose of a given design element
  - Objects that are required
  - The standardized dialogs
  - The clauses of the standard where details are defined
  - Features impacted if an object is not supported
  - Whether an object used in a test case is required
Understand Traceability

Scope of the Effort Requires Special Tools

- An ASC specification will likely have hundreds of
  - User needs
  - Requirements
  - Dialogs
  - Objects

- Keeping traceability tables in sync during editing demands special requirements management tools
Which is NOT a benefit of traceability tables?

a) They clearly identify the objects associated with a requirement
b) They reference the relevant clauses where items are defined
c) They explain the steps required in a test procedure
d) They provide traceability back to the user need
Review of Answers

a) Clearly identify the objects
True! The traceability tables list each object required for each requirement.

b) Provide references to relevant clauses
True! Each item is associated with a reference to the clause where the item is defined.

c) Explain steps required in a test procedure
False. Traceability tables do not address steps within a test procedure.

d) Provide traceability back to the user need
True! A user can trace from object to user need in either direction.
Summary of Learning Objective # 3

Understand Traceability

- User needs to requirements traceability is a simplified version of the PRL
- Requirements traceability matrix is identical to that in the standards
- Traceability takes effort, but it is well worth it
Learning Objective # 4 – Develop the Specification

- Checklist of key elements that must be present
- How the NTCIP specification fits into the specification package
Develop the Specification

Consistency

- NTCIP is one part of interface specification
- Interface specification is one part of larger package
- Features often require components in each part of specification
Develop the Specification

Multiple Interface Specifications

- A component may need to support multiple interfaces
  - May desire support of proprietary and NTCIP protocol
  - Central systems support multiple device types/versions
- Each interface needs to be defined separately
Develop the Specification

Complete Interface Specification for ASC

Source: NTCIP 9001v04, Page 12, Figure 4
Develop the Specification

Interface Specification

- Example Text in Student Supplement
  - Boilerplate
  - User needs
  - Requirements
  - Dialogs
  - Any custom objects
  - User needs to requirements traceability table
  - Requirements traceability matrix (RTM)
  - Communication stack specifications
Which of the following statements are true?

a) The interface specification is the most important part of a procurement
b) An interface specification should contain or reference dialogs
c) An interface specification must define the communications stack
d) A central system should only support one interface
Review of Answers

a) Most important part of a procurement
   False. The schedule, budget, hardware, and software specifications are all important.

b) Should contain or reference dialogs
   True! In order to achieve interoperability, the dialogs should be explicit.

c) Should define the communications stack
   True! The communications stack defines over what medium components will communicate.

d) Central system should only support one interface
   False. Most central systems will need to support at least one interface per device type.
Summary of Learning Objective # 4

Develop the Specification

- The interface specification is one part of a larger package
- A component may need to support multiple interfaces
- There are several parts to an interface specification
What We Have Learned

1) Requirements can be identified from **user needs**, **SEP-based standards**, **conformance groups**, and the three (3) perspectives.

2) Well-written requirements clearly identify the actor, action and **target**.

3) **Dialogs** define rules on how to exchange information while **objects** define the meaning of the information; both must be defined to achieve **interoperability**.

4) **Traceability** tables allow a user to quickly determine why a design feature is included.

5) An interface specification should **reference** standards whenever possible rather than copying their content.
Resources

- NTCIP 1202:2005 v02.19
  - Object Definitions for Actuated Traffic Signal Controller (ASC) Units v02
    - www.ntcip.org

- NTCIP 9001:2009 v04
  - The NTCIP Guide
    - www.ntcip.org

    - www.iso.org
QUESTIONS?