Applying Systems Engineering Principles to the Development of Transportation Communication Standards

Paul Gonzalez
Sr. Principal Engineer
Noblis, Inc.

Jeris White
Sr. Principal Engineer
Noblis, Inc.

Blake Christie
Principal Engineer
Noblis, Inc.

Abstract

Using traffic control devices and communications equipment to monitor and control arterials and freeways has brought with it new challenges and a demand for new skills to the transportation industry. Some of the challenges include integrating information from disparate and proprietary systems that were not designed to interoperate, designing systems that demonstrate they satisfy public transportation agency needs, and overcoming not-invented-here (NIH) attitudes. To address these issues the US Department of Transportation (USDOT) sponsored the development of interface standards for Intelligent Transportation Systems (ITS) technology. This paper describes how systems engineering principles were applied and the life-cycle model used to develop ITS communications standards to overcome the challenges introduced with the application of new technology[1].

Background

Anyone who drives in or around a major US city knows that its transportation system is faltering under the volume of privately owned vehicles that take to the roads each day. The situation has been worsening over the last twenty-five years and there are limited options available for mitigating it. States and local transportation system operators increasingly find themselves with limited cash resources, much of which has to be devoted to maintaining the existing transportation infrastructure. Even if the money were available, building new roads or expanding old ones would be difficult, given that land – the scarcest resource – has multiple uses other than roads to which it must be put. Mass transit, a much more efficient means of moving people than privately owned vehicles, is frequently opposed by local groups that don't want their personal interests affected by light rail construction projects.

Faced with limited resources and public resistance to better transportation planning, transportation system managers have turned to technology to provide a means of using existing resources more efficiently. The goal of technology use, in the form of Intelligent Transportation Systems (ITS), is to provide transportation system managers with better information about conditions on transportation networks. The better information should allow transportation system managers to react more quickly to problems that occur, e.g., traffic accidents, roadway spills, buses caught in traffic that fall behind schedule, and make better decisions about actions to address those problems.

[1] For the purposes of ITS Standards development, the life-cycle model for software development is consistent with the INCOSE life-cycle described and referenced later in this paper.

The introduction of new technology generally brings new challenges and the introduction of ITS to transportation was no different. The types of common challenges that transportation system managers faced when beginning to use ITS included:

- Proprietary technology – when first introduced, new technology is generally proprietary and expensive.
- Incompatible systems – new computing technologies don't start out interoperating[2]. This makes it complicated to use similar items in different locations and creates a vendor dependency. To ensure that items work together, it is generally necessary to acquire them from the same vendor.
- Lack of an integrating vision – since the technology is new and the users generally inexperienced in its use, there is no vision for how to integrate different systems into a cohesive whole.
- 'NIH' (not invented here) syndrome – people are most comfortable going with solutions that they know. Therefore, once they've applied one solution, they're generally reluctant to try a newer approach.

When the US Department of Transportation (USDOT) perceived the problems that transportation organizations were having in adopting ITS, they decided to sponsor the development of interface standards for ITS technology. In the mid-1990s, the USDOT kicked off a program that used standards development organizations (SDOs) to develop interface (communications) standards for ITS. The USDOT provided the funding and the SDOs undertook the development of these standards through their normal standards development processes.

Typically, the early standards developed through this process weren't as successful as desired. Common problems were:

- Standards failed to capture the full scope of the areas they were intended to address
- Technical and domain representatives on the standards development working groups lacked the systems experience to recognize all of the factors affecting the required interfaces
- Standards were hard to read and ambiguous

As a result, the very organizations that one would expect to adopt the standards ignored them. The agencies fielding ITS capabilities didn't see any value in standards that were incomplete, ambiguous, and hard to understand, and failed to guarantee the very benefits they were developed to deliver. As a result of the inability of the initial standards to gather much support within the transportation community, the SDOs were asked to revise them. However, the very problems that led to poor initial standards hampered the effort to revise them. These problems included:

---

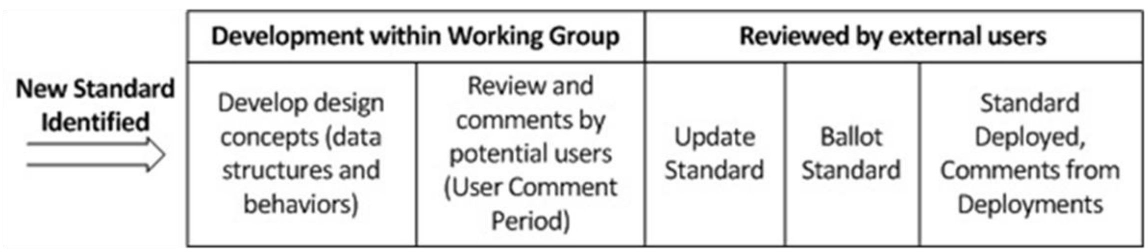
[2] Interoperability: "The ability of two or more systems or components to exchange information and use the information that has been exchanged." IEEE Std. 610.12-1990 – IEEE Standard Glossary of Software Engineering Terminology

- Lack of systems expertise among working group members
- Inadequate involvement of users from the transportation agencies, who recognized their lack of technical expertise and deferred unnecessarily to the “systems experts”

In particular, the inadequate involvement of users from transportation agencies was troubling. Since they, in essence, are the ones expected to adopt these standards when implementing ITS projects, their marginalized involvement meant that the perspective of user needs was being downplayed. The “technical experts”, individuals representing system developers, were the more influential members of the committees, even though their perspective was from the vendor side, not the user side of the equation.

The initial process used by the SDOs is shown in Figure 1 below. When the need for a new standard was identified, it was assigned to one of the SDOs under contract with the USDOT. That SDO assigned the standard to the working group that it believed had the relevant knowledge to develop the standard. Most of the working groups for ITS standards were created specifically to develop those standards and had representatives from both industry and the public sector agencies that would use the standards. In addition, each of the working groups used consultants to assist in technical details.

Figure 1- Initial ITS Standards Development Process



The process shown in Figure 1 is solution-centric. By that, we mean that it focuses too quickly on developing answers to problems before being sure what the problems really are. Validation of the solution is highly dependent on:

1. The consultants clearly understanding user needs, and
2. Feedback from deployments to correct any deficiencies in satisfying user needs.

If, as it turned out, user needs were not well understood or addressed, feedback from deployments tends to be fairly critical (in all sense of that word). The next cycle of standard revision, however, carries no guarantee that the changes to the standard will address all unmet needs. At best, the revisions can only address those unmet needs that were identified through deployments. And, if the revisions to the standard are broad, older deployments may require extensive revisions of software implementations, with costs that strain transportation operators’ budgets.

The process resembles, in many ways, the process of finding an error in a complex system where one cannot isolate the component that is failing. In such a case, one has to make changes to different components, frequently one at a time, to determine which component is causing the problem. If no single component is at fault, then one has to test combinations of components to see if their combined use is causing the fault. This is a very laborious, time-consuming, and expensive process and is generally the last resort one wants to use in problem resolution.

Systems Engineering to the Rescue

As an alternative to what was perceived as a flawed process, a proposal was made to the SDOs and the USDOT to use a systems engineering process for developing ITS standards. The argument used in the proposal was that using a systems engineering process would yield the following benefits:

- Provide a context for the standard, a concept of how the standard would be used in actual operation of a system, expressing the user needs that the standard would address
- Develop clear-cut requirements, based on user needs, for the interfaces and devices requiring standardization
- Trace the requirements back to user needs, to show users how the standard evolved and how these requirements met their needs
- Design standard solutions that addressed those requirements, to support consistent solutions and interoperability
- Trace standard solutions back to the requirements that they addressed
- Create a mechanism for testing products that claimed conformance to the standard

Since any good systems engineering process is tailored to the specific area in which it is expected to function, an engineering profile was created to define the purposes, qualities, and development life-cycle needed for ITS standards. This profile identified the following:

- ***Purpose***: Provide transportation related interface standards that support the integration of interoperable systems.
- ***Qualities***: These are the set of qualities needed by the documents that define ITS standards. The specific qualities to be sought were identified as:
  - Usability
  - Readability
  - Maintainability
  - Interoperability
  - Flexibility

  The above qualities focus on how ITS standards are to be used by deployers to satisfy their needs.
- ***Development Life-cycle***: The development life-cycle is similar to the previous one in that it addresses both the development of the standard within the working group and the external reviews that provide feedback to the working group. However, in detail, there are some key differences.

  Figure 2 illustrates this revised standards development life-cycle.

Figure 2-ITS Standards Development Life-Cycle Using Systems Engineering Principles

Existing Standard →

Concept Definition	Requirements Definition	Mapping Design Solutions and Develop Dialogs	Review and comments by potential users (User Comment Period)	Update Standard	Ballot Standard	Standard Deployed, Comments from Deployments
Development within Working Group			Reviewed by external users			

There are three stages for which ITS Standards Working Groups are solely responsible. These are:

- Concept definition – this corresponds to what would be the system conception stage in the product development life-cycle depicted in the INCOSE Systems Engineering Handbook[3], ending in the preparation of a Concept of Operations document. In this systems engineering process for ITS standards, an abbreviated Concept of Operations is embedded in the standard.
- Requirements definition – this corresponds to the requirements definition activities within the development stage in the product development life-cycle depicted in the INCOSE Systems Engineering Handbook[4] that usually ends with the preparation of a System Requirements Specification (SyRS), if one is using IEEE standards. For ITS standards, requirements are developed and a Needs-to-Requirements traceability matrix is created.
- Design – in the ITS standards process, this step leads to the design of interface dialogs and messages. It also involves the creation of a Requirements Traceability Matrix (RTM) that traces requirements to specific design elements that fulfill said requirements. The design stage relates to interface design activities within the development stage in the product development life-cycle depicted in the INCOSE Systems Engineering Handbook.

At each stage in the process, the Working Group performs Verification and Validation (V&V) to ensure that they are both building the right thing and building it right.

The first three stages in the process, shown in Figure 2, correspond to ones found in the product development life-cycle depicted in the INCOSE Systems Engineering Handbook. The second three stages (User Comment Period, Update Standard, and Ballot Standard) are part of the standards approval process required by the SDOs. The second three stages relate to the acceptance activities that are part of the production stage of the INCOSE Systems Engineering Handbook and the last ITS Standards development stage (Standard deployed, comments from deployments) relates to the INCOSE Systems Engineering Handbook's utilization[5] and support stages[6].

Several ITS standards profiles were developed as a result of following these systems engineering concepts. These include:

[3] INCOSE SYSTEMS ENGINEERING HANDBOOK, version 3.1, August 2007.section 3.3.2
[4] INCOSE SYSTEMS ENGINEERING HANDBOOK, version 3.1, August 2007.Appendix I
[5] INCOSE SYSTEMS ENGINEERING HANDBOOK, version 3.1, August 2007.section 3.3.5
[6] INCOSE SYSTEMS ENGINEERING HANDBOOK, version 3.1, August 2007.section 3.3.6

- A concept of operations (ConOps) section
- A requirements section
- A design solutions section

The ConOps profile (adapted from IEEE 1362-1998) focused on identifying user operational needs as they relate to the interface. The ConOps accomplishes the following:

- Defines what the user wants to do in terms of operational needs (highest level requirements)
- Defines operational policies and constraints (e.g. what policies govern the operation of the system, and what constraints does the system have to accommodate)
- Delineates modes of operation (e.g. normal mode and exception modes)
- Provides operational Scenarios (optional) – used to give examples of how the user (or system) may operate with the capability desired
- Provides one or more common architecture descriptions wherein the interface can be employed
- Tells a story and is easy to read

The Requirements profile is organized into functionally logical sections and introduces the following characteristics of well formed requirements:

- Necessary
- Concise (*minimal, understandable)*
- Attainable (*achievable or feasible*)
- Complete (*standalone)*
- Consistent
- Unambiguous
- Verifiable

In addition, requirements were developed using the form of localization, actor, action, target, and constraint, as recommended by the National Aeronautics and Space Administration (NASA) Software Assurance Technology Center (SATC).

The design solutions profile consists of the existing message definitions and added dialogs. Each solution was based upon one or more requirements and was verified for consistency with said requirements. To support the principle of interoperability, each design solution followed the rule of one and only one solution for a given requirement.

A key aspect of the application of systems engineering to ITS standards development is the creation of a traceability matrix, called a Needs to Requirements Traceability Matrix (NRTM) for center-to-center interfaces and a Protocol Requirements List (PRL) for center-to-field device interfaces. Many of the ITS standards have been updated to provide either a NRTM or PRL. This matrix helps satisfy the qualities of usability, readability, and flexibility. The user can see the interface features summarized, how they relate to each other, and any constraints; all in one location.

Typical architecture diagrams are also provided to help the user visualize the interface and the associated subsystems.

As with any complex project, quality concerns exist for the development of ITS Standards. Those quality concerns show up within each phase and at the transitions between phases. For example; within a phase, solutions must validly reflect the intent and content of the product(s) of the previous phase, and at the transitions, the stakeholders must verify the products that result from a stage. Therefore, at each phase of the ITS Standards development process verification and validation (V&V) must occur. Public agency representatives are responsible for making sure the product is complete in terms of satisfying the agencies needs. Vendors and integrators are responsible for making sure the product is correct in terms of fulfilling requirements. In this way, doing V&V at each stage of development improves the quality of the standard.

A set of rules were also developed to help the working groups conduct V&V on the phases of the standards development. These rules are summarized below:

- For a ConOps: Is this a complete set of needs and is each need correctly described?
- For requirements: Do the requirements address all the needs (completeness) and do respective requirements fulfill the need (correctness)?
- For the design: Are the requirement(s) all addressed (completeness) and does the design fulfill each requirement (correctness)?

How System Engineered ITS Standards Contribute To System Development

During the initial, system conception stages of system development, users may identify system level needs that relate to a potential interface. If the relevant ITS standard has been developed using the systems engineering process described above, one can compare the interface level user needs identified in the standard that describe the operational features and select the needs that match features users want to use in their operation. Once these set of interface needs are selected, the user can also identify the associated requirements (in the standard) traced to those selected needs and define any project-related constraints to each requirement as part of the requirements development step. In short, the user needs and associated requirements are already developed, within the ITS standards, to aid in problem assessment and functional decomposition.

At this point in time, the user has a set of requirements for the software aspects of the interface. Once this is coupled into the other system level and subsystem level requirements, they are ready to begin an acquisition for the system or subsystems as appropriate. Therefore, the ITS standards containing user needs and requirements help make the acquisition process easier.

Further functional decomposition may be necessary for the subsystems, but the ITS standards provide a complete set of interface requirements.

The ITS standards provide specific solutions to fulfill the interface requirements for use in the design and implementation steps. Use of these specific solutions support interoperability for those functions selected. Another traceability matrix was developed to provide a mapping of specific solutions to requirements. The requirements traceability matrix (RTM) maps a specific dialog and associated messages and data objects to each requirement. In this way, the ITS standards support readability, usability, and interoperability during the design steps. Once the interface requirements were determined in the requirements steps, the ITS standards provide the specific solutions in terms of dialog behavior and message content.

Project Benefits in Developing ITS Standards Using Systems Engineering Principles

Developing ITS standards using these systems engineering concepts provides the following specific project benefits:

- Creates a quality product (ensures that the interface meets user needs)
- Supports verification that the interface is complete
- Supports interoperable interfaces
- Reduces cost of interface development

Conclusion

By applying systems engineering concepts to the development of ITS Standards, those same standards make it easier to acquire, implement, and verify interfaces that support interoperability throughout the phases of the deployment lifecycle.