T1M1 CONTRIBUTION

Meeting Date: June 10, 2004

Source: **Mike Fargano, Qwest**
Michael.Fargano@Qwest.com, 303-896-3618

Title: **Generic Standards Development and Coordination Framework**

Abstract

This document provides a generic framework for standards development with well defined cross committee/forum interactions and handoff points to facilitate structured cross committee/forum cooperation. The intent of this contribution is to apply this framework to various standards collaborations as needed. This framework is based on sound software/systems engineering principles and has been applied (for several years) to standards development, e.g., by ATIS Committees/Forums (OBF, TCIF, and T1M1) and ITU-T (SG4) for Operations Support Systems (OSS) Interconnection Interfaces (between service providers).

Standards Development and Coordination Framework

Introduction

The key to successful end-to-end standards development in any domain is well executed cross committee/forum collaborations. The key to successful cross committee/forum collaborations is to have the collaborations built on a sound standards development and coordination framework that provides for well defined cross committee/forum interactions and handoff points to facilitate structured cross committee/forum cooperation. The Unified Modeling Approach (UMA) that has been used by T1M1, TCIF, and OBF for several years is a specific example of the successful use of this generic framework for Operations Support System (OSS) Application standards development – specifically using the Unified Modeling Language (UML).

At the high level, the Standards Development and Coordination Framework is a simple four step process:

1. **Requirements** – for systems, networks, and interoperability points (e.g., network inoperability points) to drive *Analysis* and *Design.* The output contains the following for the application/release (or problem or domain) at issue:

   - Business and technical requirements

   A structure requirements approach is preferred where requirements are identified (e.g., numbered) in some logical manner (e.g., hierarchy). The technical requirements can be relatively high level and the *Analysis* step (i.e., next step) could define the detailed technical requirements.

2. **Analysis** – of systems, networks, and interoperability points based on *Requirements* to drive *Design.* The output contains the technical analysis of the application/release (or problem or domain) at issue. The analysis can include details such as:

   - Detailed requirements
   - Architecture (e.g., network architecture)
   - System interaction/message diagrams
   - Procedure outline/steps
   - Flow charts
   - Data definitions and/or data models
   - Process models
   - State diagrams
   - Object models

   The Unified Modeling Language (UML) can be used as the overall methodology for information technology related issues and work items. All analysis descriptions and modeling should strive for commonality, reuse, and extensibility – within reason. Specifically the analysis descriptions and modeling should strive to be technology/protocol *independent* (within reason) to drive the extensible technology/protocol *dependent* design and to allow for analysis output reuse with multiple technologies/protocols and to allow for technology/protocol evolution. A structured linkage to the requirements that drive the parts of the analysis should be implicitly or explicitly provided.

3. **Design** – of systems, networks, and interoperability points based on *Requirements* and *Analysis* where the design specification is appropriate to be published as an American National Standard (ANS). The output contains the technical design for the

application/release (or problem or domain) at issue. The design can include details such as:

- Detailed architecture (e.g., detailed network architecture)
- System or subsystem architecture
- System or subsystem interaction/message diagrams
- Detailed procedure steps
- Schema (e.g., XML schema)
- Protocol
- Transport technology

The design specification is technology/protocol *dependent* and should be positioned for extensibility. A structured linkage to the parts of the analysis that drive the parts of the design should be implicitly or explicitly provided.

4. **Implementation (Guidelines)** – in support of system implementation, interoperability, and interconnection. The output contains implementation guidelines in support interoperability and interconnection for the application/release at issue. The implementation guidelines include technology/protocol dependent details to facilitate efficient and effective implementations.

Project Execution

Higher success likelihood will be seen in executing a project (for at leased the first three steps of an application/release) by using the *spiral model* of delivery (not the *waterfall model*). The *waterfall model* is characterized by a single large delivery of output (at each step) – even thought there are parts of the output that are stable (or partially stable) and should be released early to get some downstream work started. The *spiral model* is characterized by several short cycles of output (delivery) and downstream work initiation (with execution) – even if just in stable draft form. This allows for early necessary work to be initiated, resource loads to be spread out and more easily managed, and critical adjustments to be made. Revisions or updates are executed as needed and can start at the *Requirements* step and flow through the entire process.

Cross Committee/Forum Interactions and Handoff Points

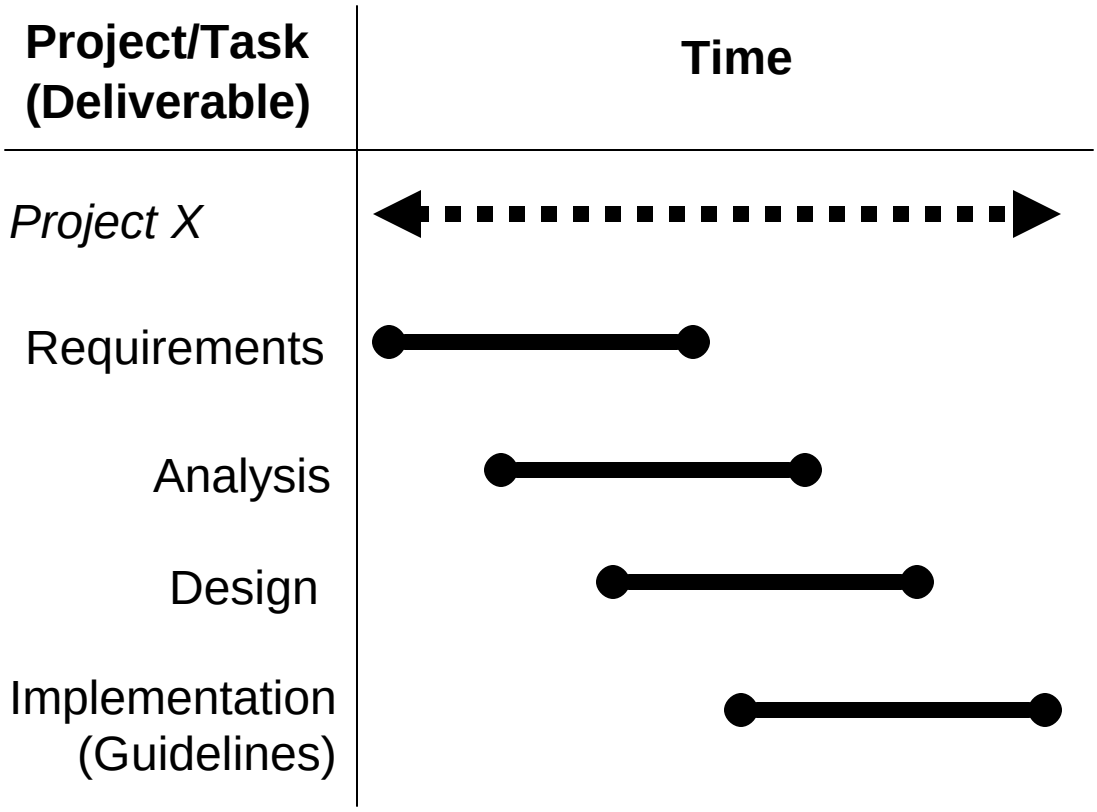
With the Standards Development and Coordination Framework in place, multiple committees/forums can provide the integrated outputs in a multi-volume (and cross-referenced) set. As an example, with just two committees/forums collaborating, a reasonable handoff point is from the *Requirements* to *Analysis* steps or from the *Analysis* to *Design* steps. Several factors should drive the committee/forum handoff point decision, i.e., charters, missions/scopes, expertise, ANSI accreditation, etc. Having a cross committee/forum team (or sub-team) that can move among the various committees with the work will facilitate smooth handoffs and work transitions. See *Committee/Forum Coordination and Responsibility Matrix for* a simple table to assign and track responsibility of collaborating committees/forums working on several projects.

Committee/Forum Coordination and Responsibility Matrix

Project ////////////// ////////////// ***Step***	**Project 1**	**Project 2**	**Project 3**	**Project N**
Requirements				
Analysis				
Design				
Implementation (Guidelines)				

The responsible committees/forums are identified in the appropriate cells of the matrix for the projects and steps in the standards development process. The lead committee/forum for the project (e.g., project coordinator/manager) can be identified as well (e.g., with a footnote). In the case of multiple committee/forum responsibility for a given project and given step, the shared responsibility is simply noted in the cell as appropriate.

Generic GANTT Chart for Project/Program Management



Notes on Generic GANTT Chart:

- **All Projects and Tasks should be *Deliverable or Achievement focused*, e.g., "Delivery of Application X *Requirements*." These Deliverables or Achievements should be end products of the Committee/Forum (e.g., Published Standards Documents).**
- **Sub tasks (i.e., specific intermediate tasks done by a Committee/Forum to work toward a Project or Task) should *not* be managed at the overall Project or Program Management level. Sub tasks and their management should be defined and managed by the responsible Committee/Forum(s).**
- **The overlap of Tasks is intended to signify the execution of the *spiral model* project method.**
- **The duration of the overlap of Tasks and duration of the Tasks themselves will vary depending on the Project (i.e., base work/contributions available, resources available, etc.). For example the Analysis and Design Tasks could start together soon after the Requirements Task starts.**