Requirements Development & Management Using Models

Loyd Baker, Jr.
Vitech Corporation
555 Sparkman Dr., Suite 1213
Huntsville, Alabama 35816

Ann Christian
Standard Missile Company
1505 Farm Credit Drive
McLean, Virginia 22102

Abstract. Good program managers understand that a high pay-off in cost savings and schedule reduction can be attained by early analysis of their system requirements. Programs employing a traditional text-oriented requirements development approach experience significant numbers of misinterpretations and late-detected design errors resulting in increased costs, schedule slips, reduced capability, and canceled programs. This paper discusses how a well defined model-based approach to the development of requirements can eliminate or greatly reduce the incorporation of incorrect assumptions into system requirements.

PROBLEMS WITH CURRENT SPECIFICATION APPROACH

The conventional way to derive and present requirements is in the form of a hierarchy in which each textual requirement ("what", "how well", or "when" statement) is divided into more detailed, specialized, textual statements. As shown in Figure 1, this approach does not provide the team members an explicit view of the required flow (i.e., sequencing, performance) of the selected requirements set. In order to understand and analyze the performance of the specified requirements, the sequencing paths through the functional requirements must be manually derived from the individual text statements, a very costly, error- prone process.

Even though many of our system/process development projects are based on hierarchical textual requirement specifications, the experienced system engineer typically constructs high-level functional *models* to develop and evaluate a feasible set of functional and performance requirements that will satisfy the customer's needs. The problem has been that these knowledge-rich engineering models developed on an adhoc, informal basis were not formally retained as part of the performance specification. Therefore the models were not available to support down-stream development/test activities or future operation and maintenance changes.

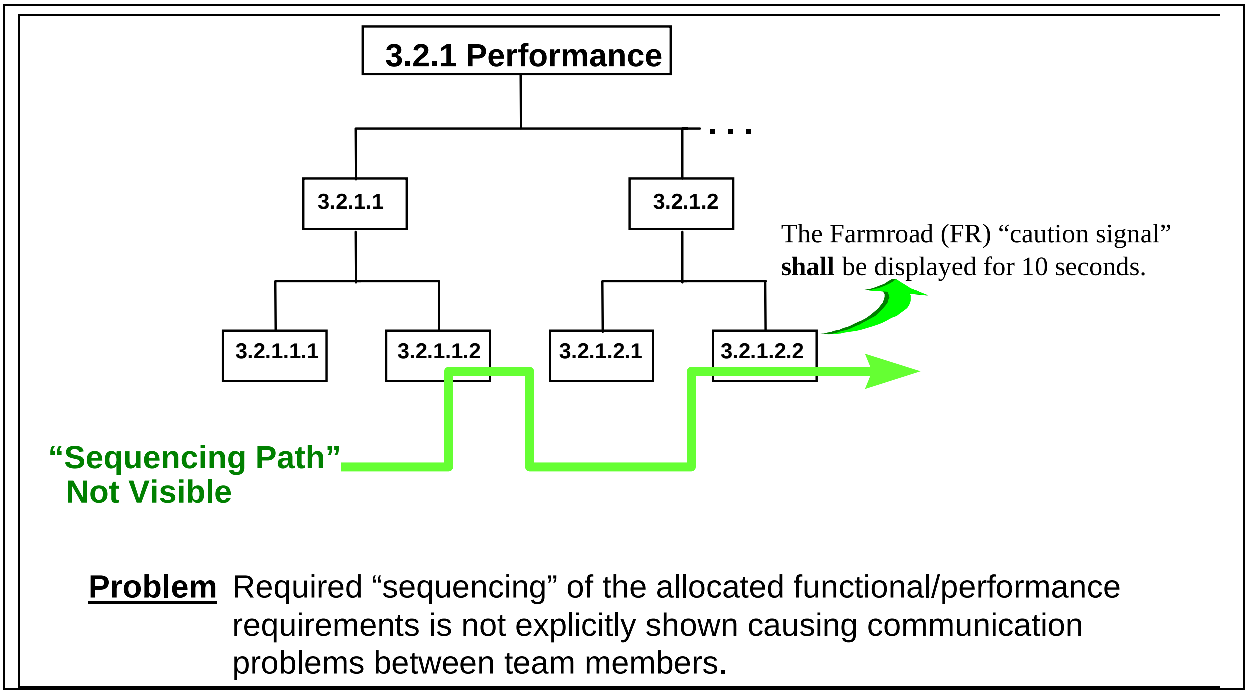

Figure 1 - Conventional Hierarchical Requirement Specification

The purpose of this paper is to present a formal, disciplined, model-based approach for specifying *flow-oriented performance specifications*. A flow-oriented specification eliminates the need to reverse engineer the textual statements in order to identify and understand the required system "performance flows" (i.e., required system behavior). Perhaps of most importance, the models act as a definitive communication medium between team members and between the customer and project liaison.

SPECIFICATION IMPROVEMENTS VIA REQUIREMENTS MODELS

A "*performance-flow*" is a functional model that defines conditional sequencing of actions, and the flow of interfacing data passing into and/or out of the required actions being specified. The modeling technique presented in this paper uses time precedence action-nodes (square-cornered rectangles), data-nodes (round-cornered rectangles), control-nodes (circles), and validation-nodes (solid black circles) to represent sequential, parallel, conditional, and iterative operations, with time flowing from left to right. A complete description of this modeling technique can be found in [1].

These models are used at all levels of performance requirements specification (i.e., concept of operations, system-level, subsystem-level, CI-level, etc) to represent what actions must be performed, when they should be performed, and how well they should be performed. The amount of detail determines whether a model is a requirements model or a design model. For a discussion of the different kinds of models used in the system engineering process, see the 1996 concept paper prepared by the Model Driven System Design (MDSD) INCOSE Working Group [2].

To illustrate the simplicity and usability of these models, an example is presented in Figure 2, the Traffic Control Requirements Model. The initial requirements (i.e., customer requirements) are as follows:

1) The contractor **shall** design a traffic-control system for the intersection of state highway 127 and farm road 2, that will allow farm vehicles to cross the busy highway.

2) The system **shall** operate 24 hours per day in all weather conditions.

3) The system **shall** be constructed from existing technology.

4) The highway traffic **shall** not be required to wait longer than 50 seconds at any one time.

Based on these customer requirements, the interactions between the Traffic Controller System and Vehicles on the Farmroad (FR) are modeled in Figure 2. The first AND control-node (circle containing the word AND) identifies two parallel paths of actions to be performed. The top path, labeled "Farmroad" contains the flow-oriented actions associated with the farmroad/farm vehicles. The bottom path, labeled "Controller" contains the flow-oriented actions (functional/ performance requirements) associated with the Traffic Controller System. The first action-node (square-cornered rectangle) on the bottom parallel path is

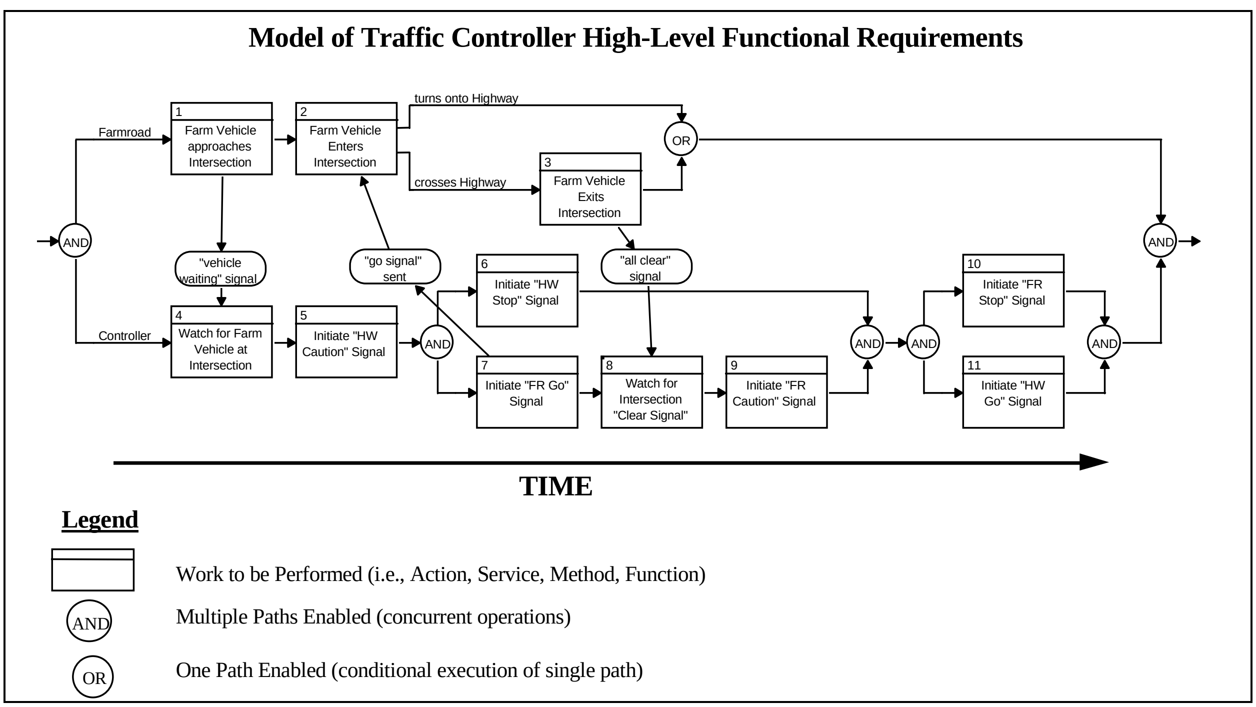

Figure 2 - Traffic-Controller Requirements Model

watching for farm vehicles to approach the intersection. When the “vehicle waiting” signal is received (round-cornered rectangle), the “Watch for Farm Vehicles at Intersection” function terminates, initiating the highway (HW) caution signal. The required duration of the caution signal will be discussed later in this section. After the HW caution signal function completes, the HW “Stop Signal” is initiated in parallel with the FR “Go Signal”. The AND node to the right of the Initiate “HW Caution” Signal function explicitly identifies this parallel condition. The second action-node on the bottom parallel path is watching for an “all clear” signal indicating that the farm vehicle has exited the intersection. Notice the black square in the upper left corner of the action-node. This indicates that a more detailed sub-model exists for the action. The sub-model shown in Figure 3 uses the OR node to wait for one of the two conditions (see labeled branches) to occur. The FR “Go Signal” shall stay on until the farm vehicle clears the intersection or a specified time elapses, whichever comes first. If time out occurs, a warning horn is sounded, followed by initiation of the FR caution signal. After the specified caution period, the following parallel actions occur: Initiate FR stop signal; and Initiate HW go signal.

This *requirements-model* diagram provides both the development team and customers a good view of the proposed solution, allowing everyone to make informed specification suggestions, and accomplish early verification and validation plans.

In addition to the information on the diagram, the model-based approach links other kinds of information to the action nodes. For example,

- a rationale for the specification decisions,
- traceability of derived requirements back to the customer’s requirements,
- any issues, risks, and assumptions made.

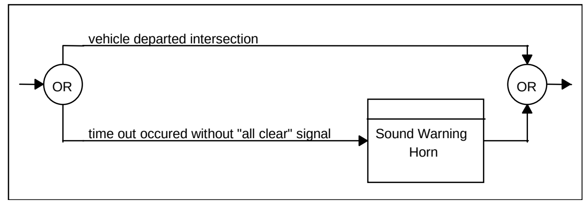

Figure 3 - Sub-Model of “Watch for Intersection “Clear Signal”

Establishing these traceability linkages supports the subsequent verification and compliance assessment activity.

ASSIGNMENT OF PERFORMANCE MEASURES VIA REQUIREMENTS MODELS

The following example (see Figure 4) illustrates the instrumentation of a requirements model to specify performance measures in order to better track performance limits. The validation points (darkened circles) are used to specify the performance requirements for a specific path of the model.

The availability of an instrumented requirements model speeds development of the test plans (see Figure 5), and improves the coverage due to the expanded visibility of the planner (i.e., the amount of reverse-engineering is minimized).

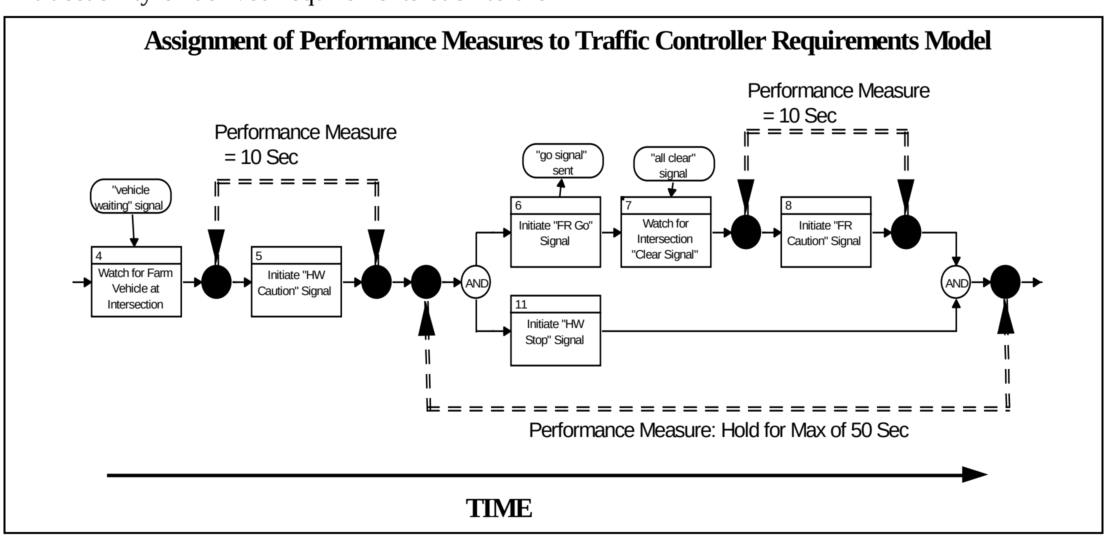

Figure 4 - Assignment of Performance Measures

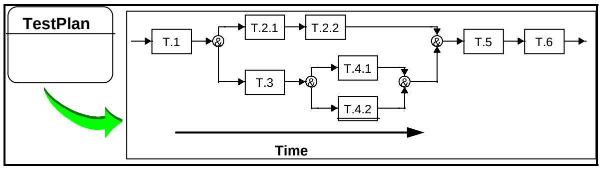

Figure 5 - Test Planning Model

PERFORMANCE SPECIFICATION PREPARATION

The requirements model structure and contents are the basis for the preparation of the performance characteristics section of the Performance Specification, (see Figure 6).

Using an automatic documentation generation capability, such as that provided in a systems engineering support tool, eliminates, or at least greatly reduces, the need to manually build project review material and formal performance specifications. This capability produces significant savings in both time and resources.

CONCLUSION

The Requirements Modeling approach summarized in this paper is based on a proven model-based system engineering process and a commercially available support tool set. The process was designed to provide a focused approach for identifying feasible functional/ performance requirements, including consideration of alternatives and capturing the rationale for key decisions that are made, and a method for automatically producing a flow-oriented performance specification.

In summary:

- Models provide an improved view of traceability and its evolution.
- Graphical presentation of performance flows enable group participation in concurrent engineering.
- Automated documentation from models produces up-to-date documentation on demand at a low cost.

The level of understanding and the sharing of ideas increases dramatically over the traditional textual hierarchical requirements approach. A well defined model-based approach to the development of requirements can eliminate or greatly reduce the incorporation of incorrect assumptions into system requirements.

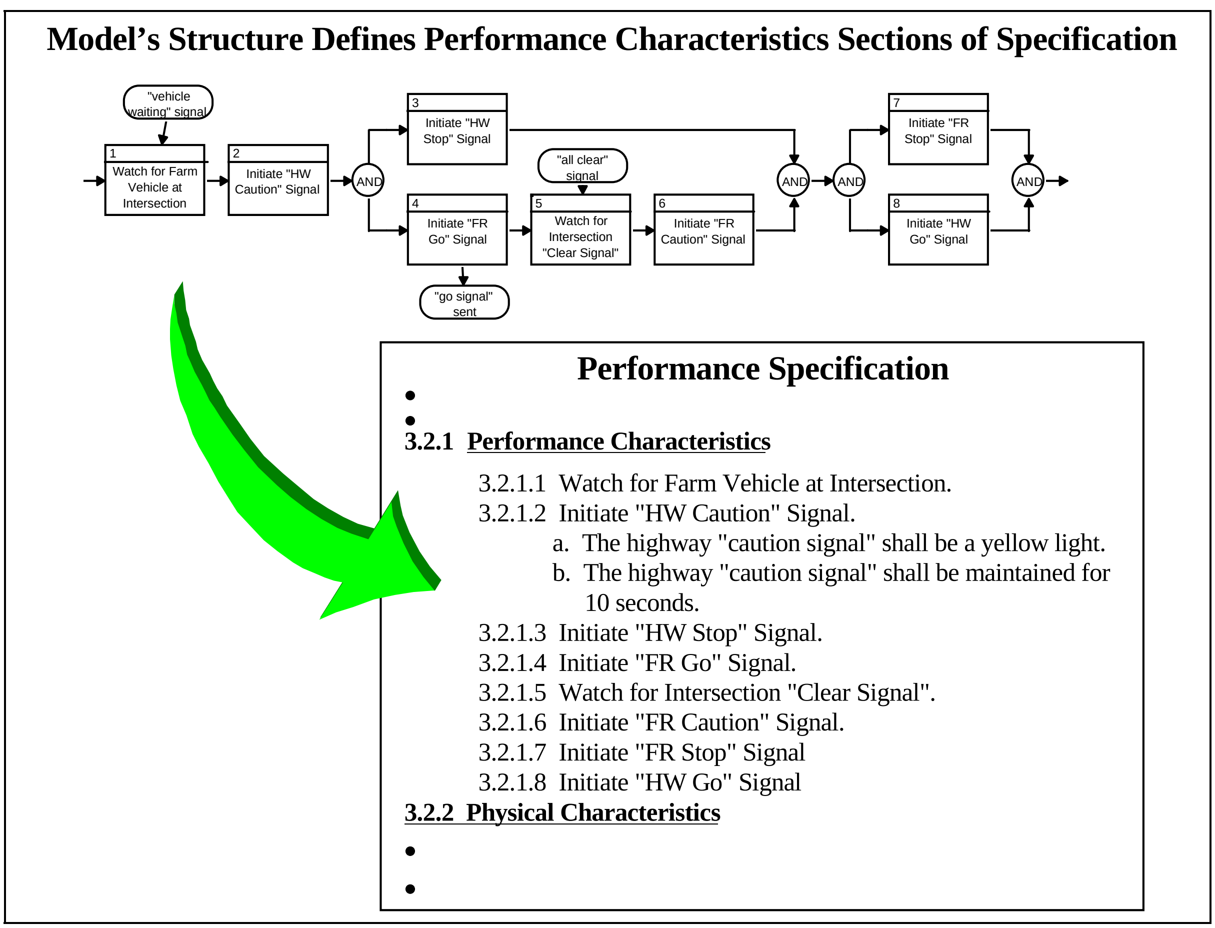

Figure 6 - Requirements Model Structure Is Basis for Performance Specification

REFERENCES

Baker, Loyd Jr., "Distributed Computing Design System: A Technical Overview, CDRL B003, TRW System Development Division, Huntsville, Alabama, July 1987.

Baker, Loyd, Clemente, Paul, Cohen, Bob, Permenter, Larry, Purves, Byron, and Salmon, Pete; “Foundational Concepts For Model Driven System Design”; INCOSE Interest Group White Paper for 1996 International Symposium.

BIOGRAPHY

Loyd Baker, Jr. is a Principal Engineer for Vitech Corporation, and serves as the co-chair of the INCOSE, Model Driven System Design, Interest Group.

Ann Christian is a System Engineer for Standard Missile Company.