Applying Structured Requirements to Infrastructure Process Development Using TeamCenter SE

Sharon Trauth
Engineering Requirements, Department 2993
Sandia National Laboratories
sltraut@sandia.gov
505-844-1957

Abstract

Delivering process-based infrastructure systems that meet user needs is seen to translate into a structured mapping of the user needs to collections of individual capabilities that, when structured together, form an overall process flow. This presentation will highlight the application of the Teamcenter Systems Engineering (TcSE) tool to a process development effort. The use of TcSE in this effort is shown to aid in synthesizing fundamental process capabilities necessary to create the infrastructure and in generating a capability deployment schedule and corresponding work breakdown structure.

Background

During a Value Stream effort conducted on the approaches applied to the management and control of Pro/Engineer solid model files across the 2990 Design group, the absence of a consistently applied Configuration Management (CM) process was identified. A team of representatives from across the Design Group was formed to develop the CM process and to plan the implementation of the associated infrastructure.

Initial work focused on identifying what the management and design staff wanted from such a process and to identify the characteristics of the incoming design jobs that the staff routinely faced as they did their work. The incoming job types, with their associated problems and complexities had to be addressed by the process. The process was engineered to guide staff through process steps and decisions that would lead to reduction in redundant and incorrect data and to generation of information and metadata that could be more easily referenced and maintained over time.

In addition to the initial set of needs, the pilot process concepts were reviewed and modifications were requested by pilot users. Associated infrastructure and training needs were also identified. The detailed development of the CM process and its infrastructure quickly led to a collection of user and process needs that were difficult at best to use and were virtually impossible to assure were being addressed. The process infrastructure itself began to grow in complexity, and it became unclear to the development team whether the proper infrastructure elements were being considered and what their relative priorities were for implementation.

Structured Requirements

Needs were identified from various sources, such as:

- Management
- Product definition staff
- Pilot users
- Trainers
- The CM process itself
- Individual work scenarios
- Technical Business Practice requirements
- Information systems

These needs were entered into the commercial tool, Teamcenter System Engineering. At the start of this work, team members were not fully experienced in using the tool to engineer solutions. Several attempts were made at capturing information such as use case scenarios, needs, diagrams, and supplemental requirements. Figure 1 shows an image of the structure of the TcSE attempts to manage the information associated with this effort.

One systems engineering capability offered by the TcSE tool provided means to determine the infrastructure elements that would need to be

addressed. Once the major process functions were identified, prioritization became easier, and determining feasible solutions to problems became more manageable.

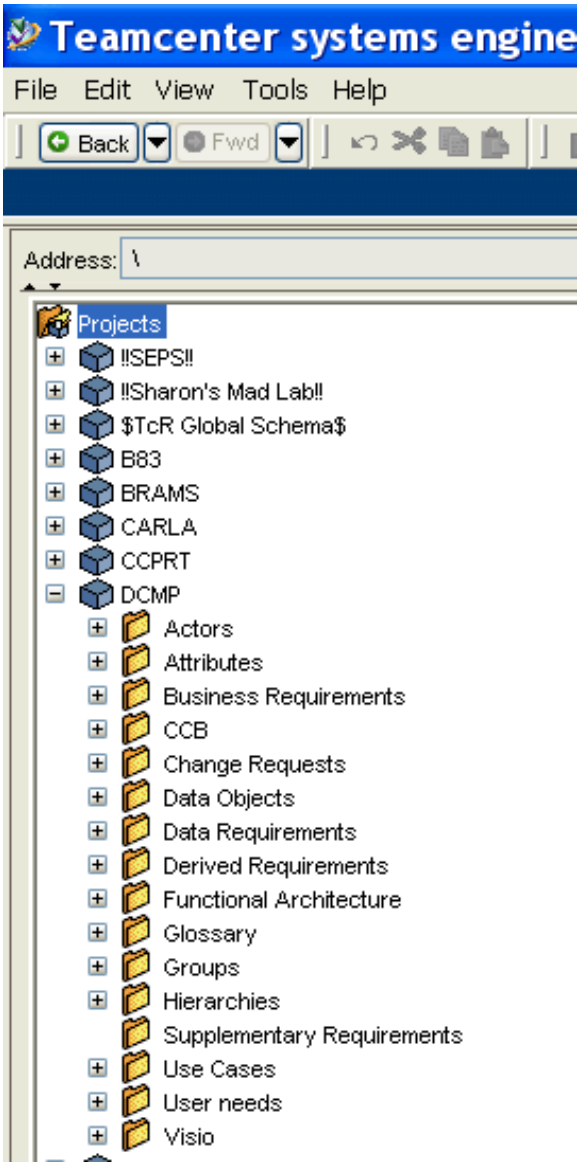

Figure 1: Use of Teamcenter for Analysis of the DCMP Project

Synthesizing Functional Process Capabilities

After several attempts to analyze the data in TcSE, developing a functional architecture became the focus. In developing a functional architecture, the Systems Engineer develops the collection of functions that have to be implemented in a solution to meet the needs of a particular set of requirements, i.e., to develop a solution to the particular problem that will meet the requirements.

Figure 2 illustrates the fundamental process capabilities identified in TcSE as functional architecture elements. The diagram also shows the first and second level decomposition, or breakdown, of the fundamental functions into sub-functions. The lower portions of Figure 2 illustrate the use of the traceability capabilities of the TcSE tool. The window in the lower left indicates that the Associate Multiple Model [files] functional capability is a sub-function (or sub-process) of the Create Product Baseline functional capability. It also illustrates that the Associate Multiple Model [files] function is driven by 3 requirements. Turning to the lower right hand window of Figure 2, it is seen that the Associate Multiple Model [files] function has 5 sub-processes or sub-functions that comply with {or exist because of, or trace to}. Functional decomposition of the system can occur to whatever level of detail or granularity is necessary to develop a viable functional solution to the problem.

Depicting Functional Architecture

Figure 3 illustrates the Visio Diagram exported TcSE and color coded to show the prioritization assigned (1,2,and 3) for a 3-phased implementation of the capabilities. Development of a specialized script function in TcSE permitted the export of the complete functional architecture into an Excel spreadsheet that readily was imported into Microsoft Project. Figure 4 illustrates the resultant project Work Breakdown structure that is fully consistent with the functional architecture engineered within the TcSE tool.

Summary

The application of the Teamcenter Systems Engineering (TcSE) software tool provided a mechanism to capture a number of different types of information associated with the development of a process for configuration management of design definition, in addition to requirements, or needs.

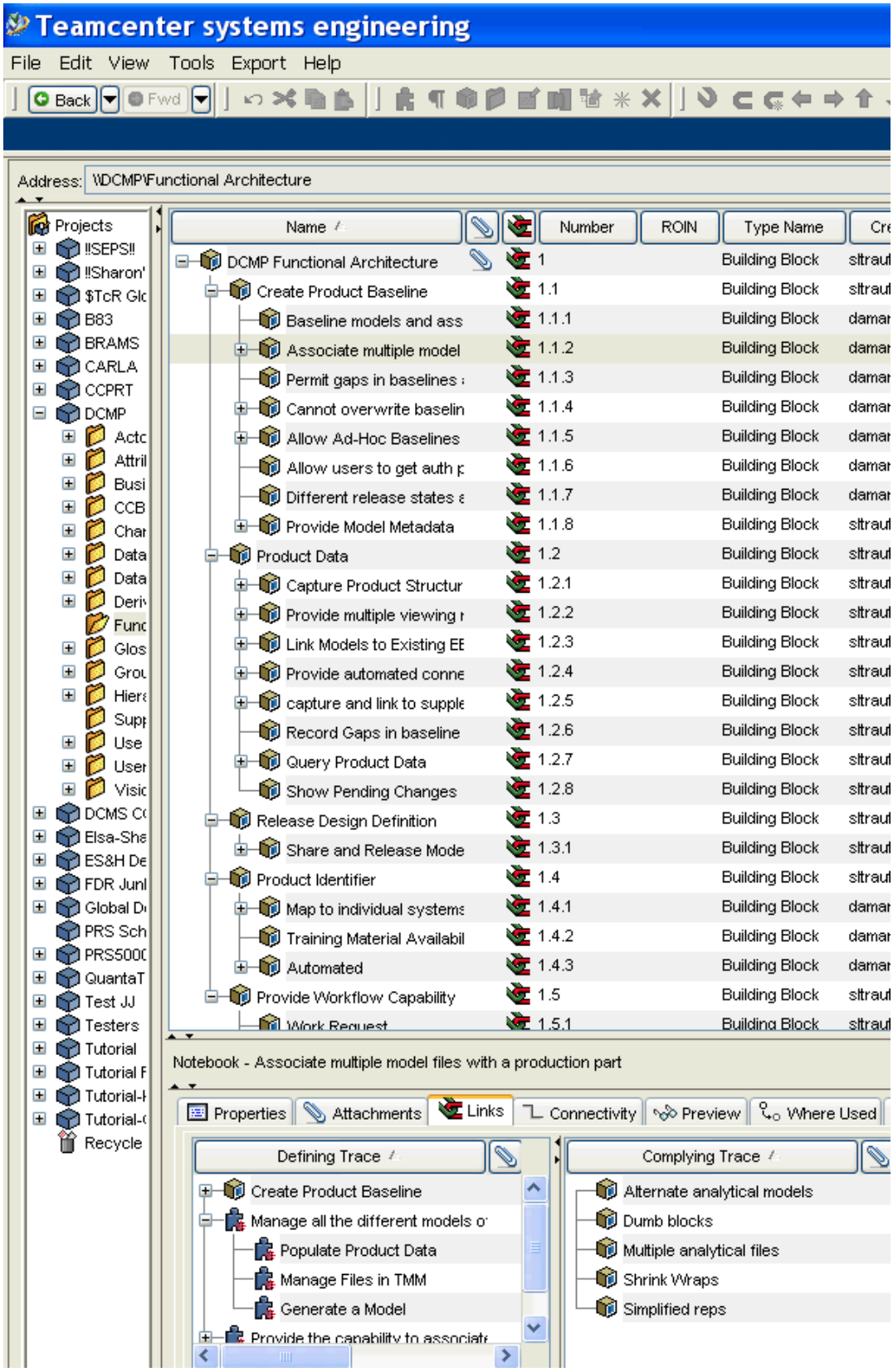

Figure 2: DCMP Project Functional Architecture and Traceability in TcSE

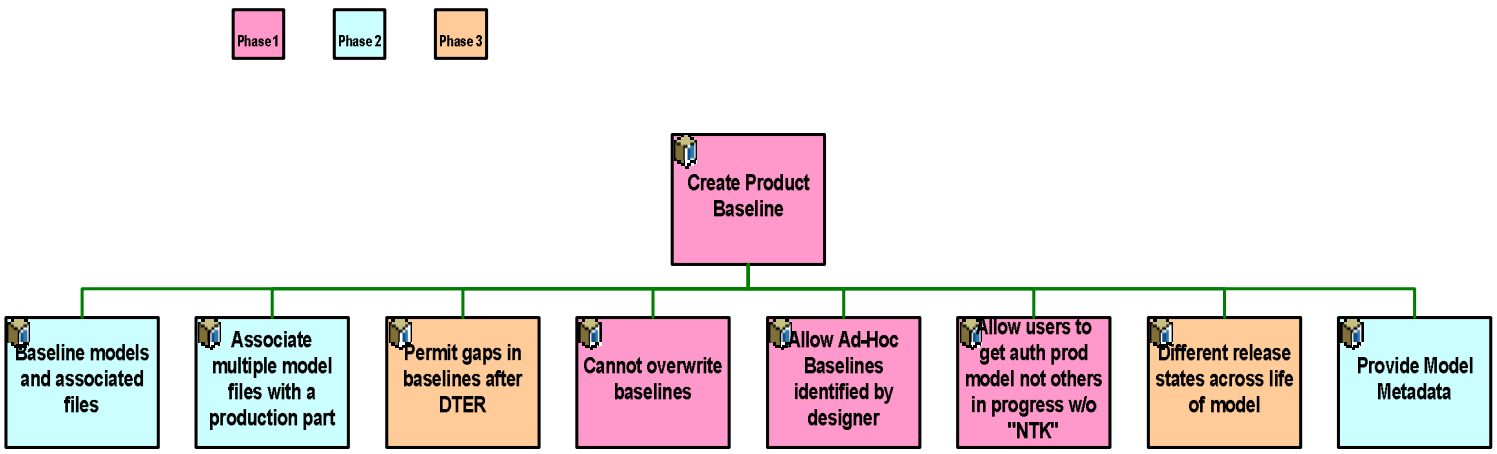

Figure 3: Visio Functional Diagram (Pink, Blue, & orange represent Phases 1, 2, 3 respectively)

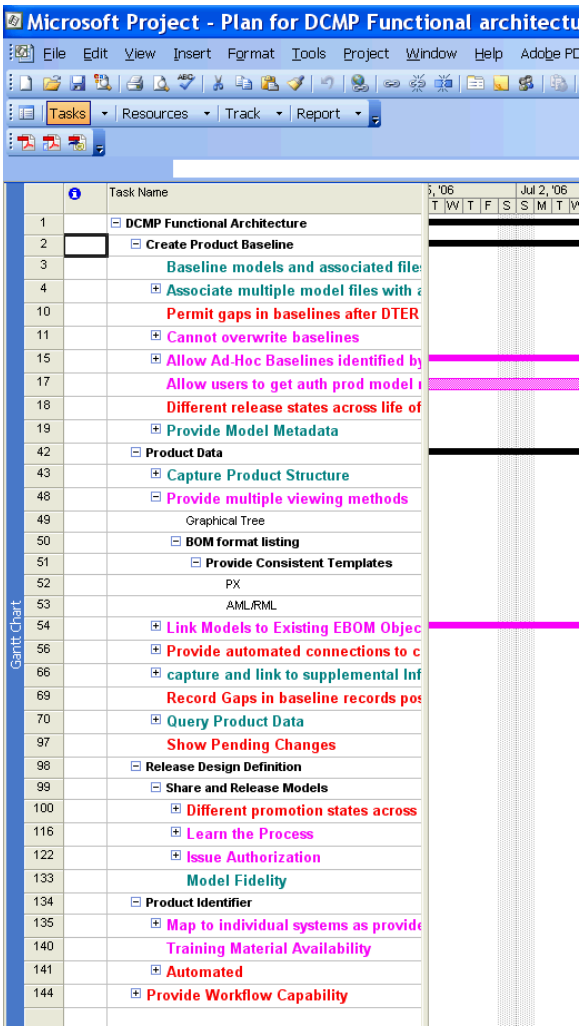

Figure 4: Resultant Microsoft Project Work Breakdown Structure Exported from TcSE

Engineering a solution to a collection of requirements, or needs (*i.e.,* developing the solution) is not accomplished solely through the management and flow-down of the project requirements. As solution concepts are explored, both functional and physical aspects of the solution evolve. TcSE provided a mechanism to capture the evolving architectural concepts and to tie those concepts to the requirements and needs for the problem solution.

By associating the needs to the architecture, the system engineer can analyze whether the solution concepts are viable. In other words, an architecture that does not address (or trace) to all the requirements, has to be an incomplete solution, by definition. Similarly, a solution that has structure elements that do not map to any requirements is a solution that either has an incomplete set of requirements or solves a somewhat different problem.

This project demonstrated that not only can a tool such as Teamcenter Systems Engineering be applied to the development of physical hardware systems, but also such tool capabilities can assist in the development of functional capabilities associated with process implementations.

The results of this initial process development effort have led to concept explorations for a number of infrastructure process projects in areas of ES&H, test equipment design processes, and in development of standard engineering processes.

To follow-on to the positive results experienced with the work described here, the author recommends that process developers go beyond the typical mapping of external contractual or customer requirements to process steps in a particular topic (domain) area. By synthesizing needs and requirements from a number of various sources, systems engineers will enable verification and validation of process functions, development of streamlined and efficient processes, and modification of structured processes in response to changing needs and requirements. Rather than starting from scratch whenever a new collection of needs is encountered, the engineer can focus only on those portions of the process flow and architecture that *really have to change*.

References

1. *Operational Concepts and Implementation Strategies for the Design Configuration Management Process*, SAND2007-3192, Sandia National Laboratories, Albuquerque, NM, May 2007. [Unclassified]

2. Design Configuration Management Process development website: http://www-irn.sandia.gov/organization/div2000/ctr2900/dpt2993/2990%20BOM%20Design%20Process/BDPIndex.htm