A CONCURRENT METHODOLOGY FOR THE SYSTEM ENGINEERING DESIGN PROCESS

Susan Rose Childers
Computer Sciences Corporation
7471 Candlewood Road
Hanover, Maryland 21076

James E. Long
Vitech Corporation
2422 Rocky Branch Road
Vienna, Virginia 22181

Abstract. The system engineering design process as detailed in MIL-STD 499 does not adequately address the incorporation of design issues relevant to specialty engineering disciplines. This paper details a process by which the concerns of specialty engineers are addressed and incorporated into the system design in a consistent and efficient manner. The paper also introduces an incremental system engineering process (The Onion Model) which allows complete interim solutions at increasing levels of detail during the system specification process.

Along with the three standard concurrent engineering focuses of requirements analysis, functional analysis, and architecture definition, the process includes concurrencies for specialty engineering. Each specialty engineer assesses the implications of the system engineering design process in relation to the specific specialty requirements and evaluates the implications of the system functional and physical architectures on the realization of an adequate specialty engineering solution. Once an integrated solution achieves the approval of the acquisition agency, it is refined to the next level of detail. When the system level design is sufficiently refined to generate the System/Segment Specification, the segment development process may begin.

BACKGROUND

The system engineering process as described in MIL-STD 499 does not address the issue of how to incorporate the needs of specialty engineers (MIL-STD-499B, 1992). While engineering disciplines such as security, power, reliability, nuclear, or thermal have real requirements and generate functional and physical constraints on the system, the inputs of these specialty engineers generally are sought after the system design is in place and are often ignored, discarded, or seriously compromised due to a cost or performance impact. Such system solutions are not system solutions at all as they ignore or respond poorly to some of the requirements of the acquisition agency.

By using an integrated system/specialty approach, the design team can address all system needs in parallel. Concurrent engineering allows the issues and impacts of each facet of the design to influence the overall system design solution. An iterative approach allows the design to evolve to the proper level of detail. Such a methodology is described in this paper. The methodology is independent of toolset; however, it can be implemented using tools such as CORE™ or RDD-100®. The figures in this paper are derived from the CORE product.

CONCURRENT SPECIALTY ENGINEERING

System engineers have long realized that system design must address three primary activities: requirements analysis, functional analysis and architecture analysis (Long et al.,1968) (DSMC, 1989). In order to achieve a leveled design for all aspects of the system, a concurrent process is included for each applicable specialty engineering discipline. Within each of these processes, three concurrencies parallel the traditional system design activities, as shown in Figure 1. These concurrencies provide for review by the specialty engineering disciplines of the system-level requirements, behavior, and architecture designs and for the design of high level specialty engineering implementations.

Interfaces between the activities are triggered by requirements review cycles or by issues pertaining to implications of handling the specialty within the system requirements, behavior, or architecture. Issue resolutions are input directly into the appropriate system design branches.

THE ONION MODEL

An incremental system engineering process allows the system engineers to maintain a consistent level of analysis and provides complete interim solutions to the system development process. This approach, called the "Onion Model," provides a lower risk design approach since complete solutions at increasing levels of detail are available for early review and validation. The Onion Model process is shown in Figure 2. As the system engineering team successfully completes one level of system design, they peel off a layer of the onion and start to explore the next layer. When the

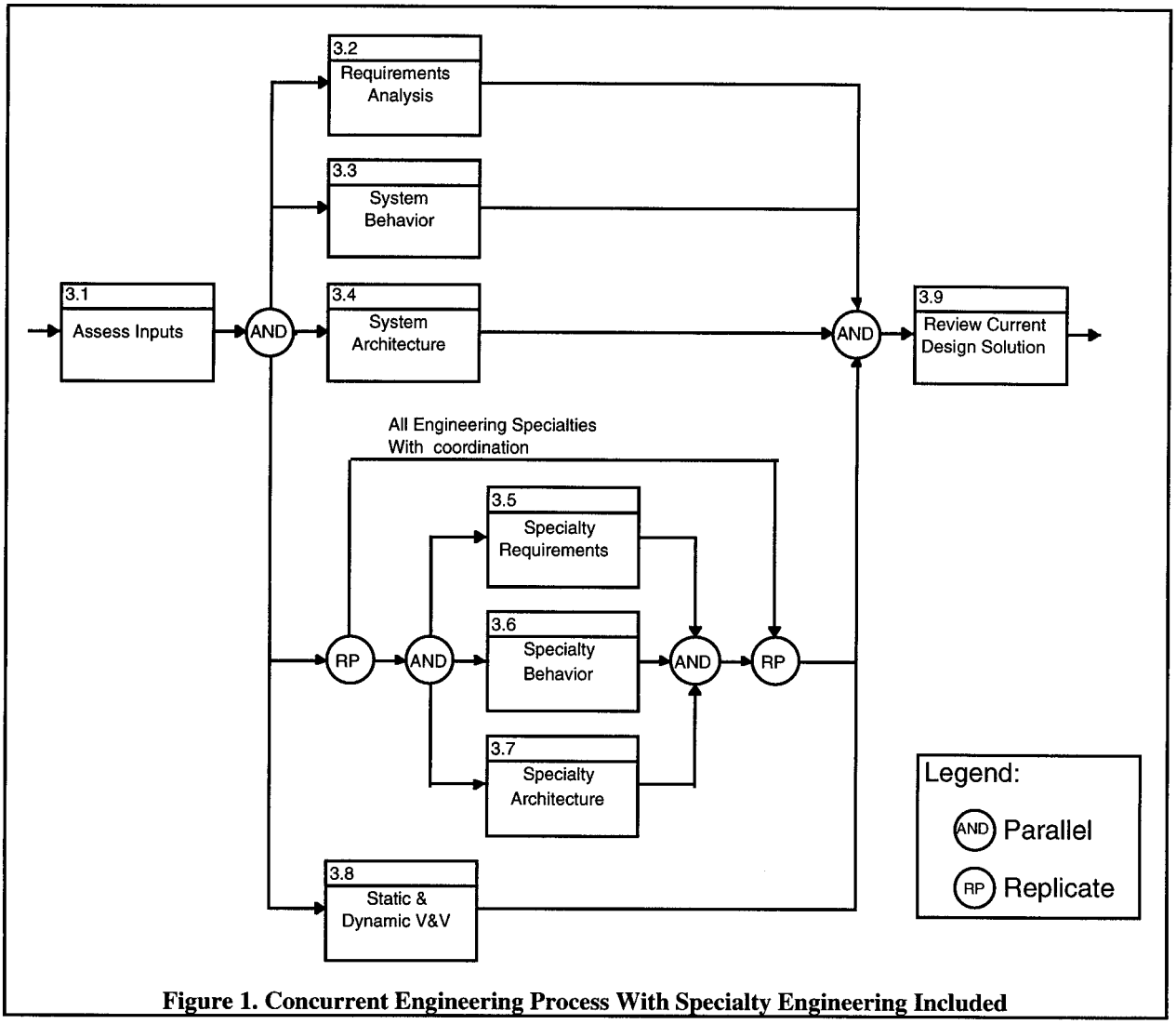

Figure 1. Concurrent Engineering Process With Specialty Engineering Included

team reaches the desired level of detail (the center), their design is complete.

If no valid, consistent solution can be found at any layer, the system engineer must check if the system statement is overly constrained and may need to negotiate modifications. More likely, modifications to the design implementation at the previous layer are required. Such instances are commonplace. However, problems necessitating changes to the design two or more levels above the current layer constitute system design breakage and can have serious cost and schedule impacts. If the acquisition agency redirects the program, the design process must start again at the context definition stage.

CONTEXT DEFINITION

The first step of any successful system design process is understanding the needs and desires of the acquisition agency and understanding the context in which the system will operate. Prior to deriving a system solution, it is necessary to determine which customer inputs are requirements and which are merely implementation directions for the designer. This distinction allows the engineer to evaluate customer expectations and to define the system boundary based upon requirements rather than a directed implementation. If any of the source requirements are implementation-specific, they should be abstracted back up to the system level in order to achieve a consistent set of originating requirements from which to begin the design. Preferred customer implementations are captured as system constraints. Upon completion of this initial process, the system engineer has a set of system originating requirements, the system boundary definition, and an initial set of critical issues. Validation of these items by the acquisition agency provides a baseline for the design process.

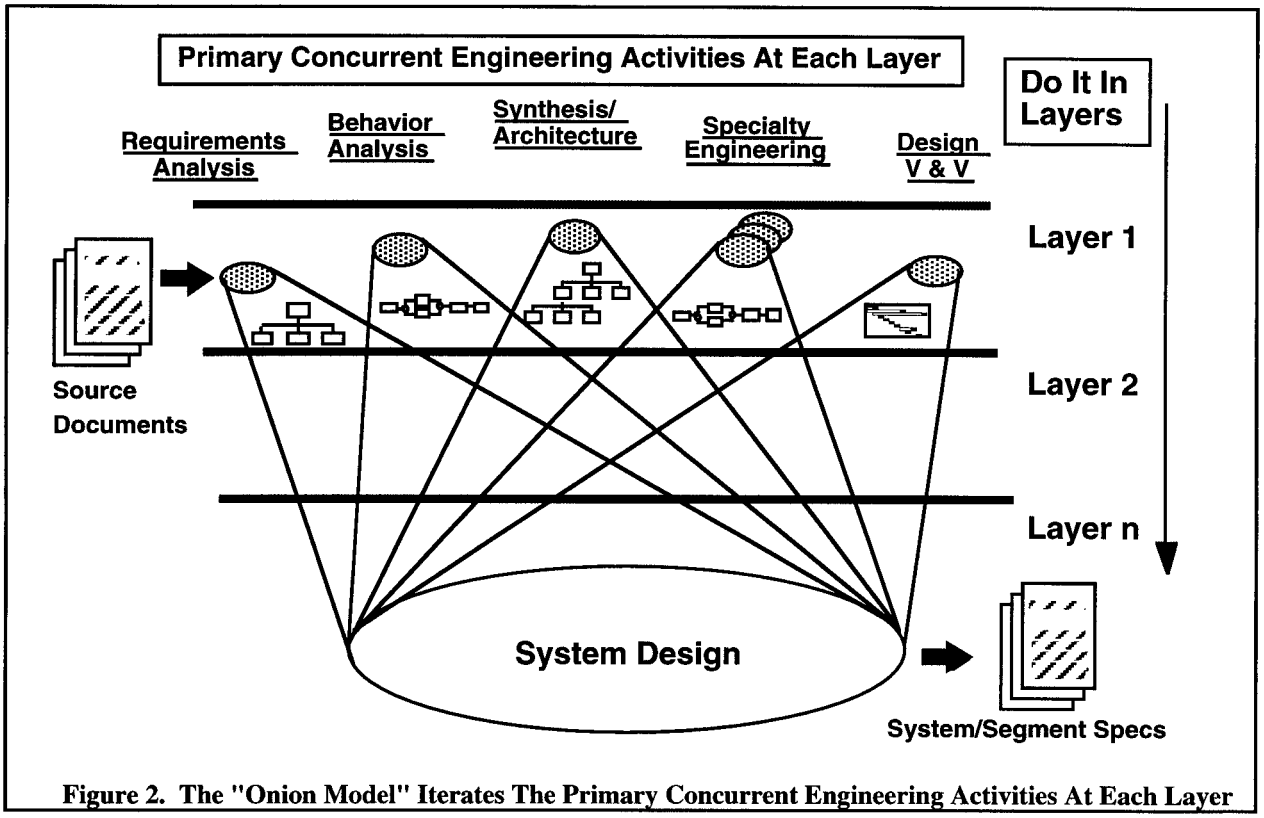

Figure 2. The "Onion Model" Iterates The Primary Concurrent Engineering Activities At Each Layer

SECURITY ENGINEERING EXAMPLE

Security engineering is a specialty which is becoming more important as time goes on. Commercial and Government customers alike are insisting that their systems and data be adequately protected from attack. In order to design a secure system, the concurrent engineering process may be used to assure a complete and integrated solution. The following discussion walks through the methodology using security engineering as the specialty engineering discipline.

CONTEXT. The first step of the system design example is to define the context of the system. The system engineer must assess the customer requirements and concept of operations for completeness and consistency. After negotiating necessary changes, the system engineer defines the boundary of the system. If customer desires do not permit a top down design, the originating requirements must be generalized to the top level and the specifics must be captured as design constraints. Figure 3 depicts the context definition process.

In our example, the system's Security Policy is among the set of customer requirements. Therefore, an INFOSEC (Information Security) engineer is among the members of the system engineering team. The system design process includes concurrent engineering of requirements, system behavior, physical architecture, and the security specialty.

REQUIREMENTS ANALYSIS. Requirements analysis includes capturing the system originating requirements, as shown in Figure 4. The initial requirements set is reviewed by the INFOSEC engineer for completeness and consistency with respect to the system's security environment and the Security Policy. Additionally, any requirements issues dealing with system security are coordinated by the INFOSEC engineer. The system engineer integrates the security engineering requirements and formalizes the requirements traceability.

The initial requirements set and any updated requirements sets are distributed to the INFOSEC engineer and are also used as inputs to the functional and physical architecture branches. The system engineer receives feedback from these processes in the form of requirements issues. The issues are addressed and a new requirements baseline is achieved. Each updated requirements set is reviewed by the INFOSEC engineer and becomes a trigger for the functional and physical architecture processes. Figure 4 depicts the requirements analysis process.

FUNCTIONAL ANALYSIS. To start the functional analysis activity, as shown in Figure 5, system scenarios are generated to characterize the functional behavior of the current set of requirement refinements. Since these scenarios are generated based upon a specific set of requirements, a valid functional

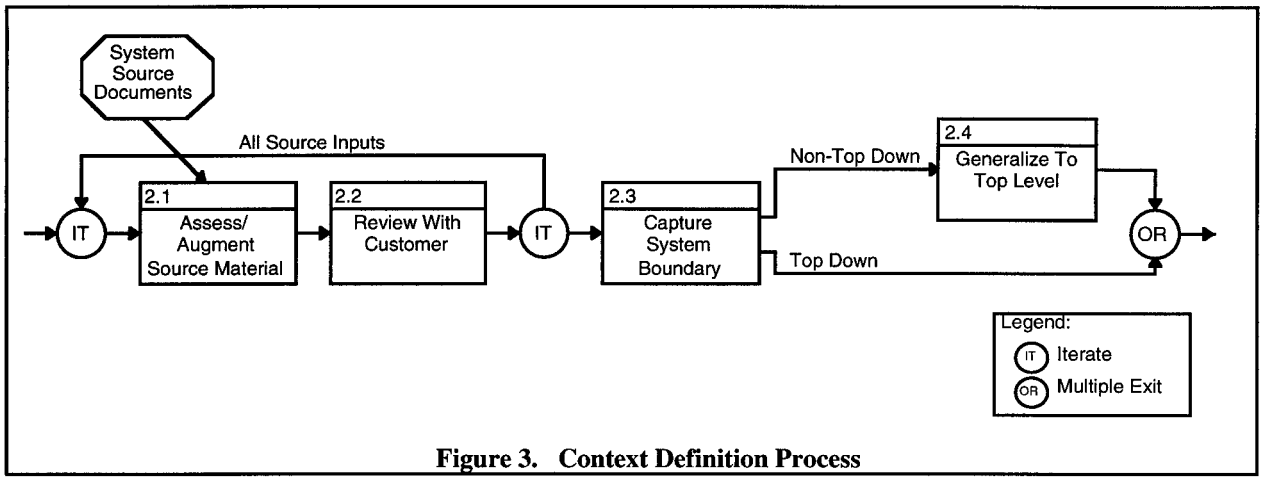

Figure 3. Context Definition Process

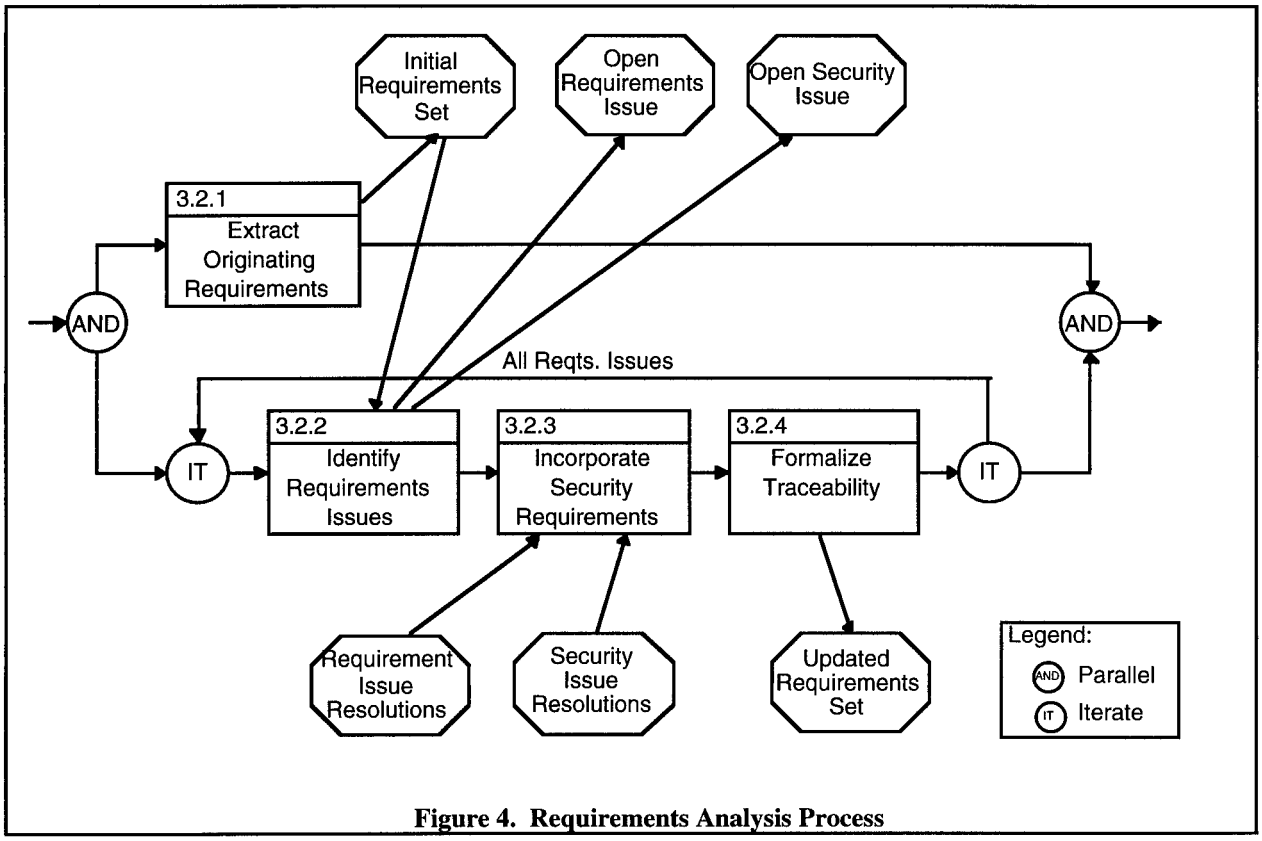

Figure 4. Requirements Analysis Process

solution which satisfies the input requirements achieves consistency between the two branches.

If the scenario entails security, the INFOSEC engineer is requested to recommend features which will satisfy the security requirements. These recommendations are integrated into the scenario. Once the scenarios are complete, the system engineer generates the integrated system behavior. The system behavior and functional performance are analyzed to determine the adequacy of the functional solution. The process continues until an adequate solution is found, providing consistency between the system and security functional branches, or until the system engineer determines that the system is over-constrained and no functional solution can be found.

PHYSICAL ARCHITECTURE. Within the process of deriving the physical solution, as shown in Figure 6, candidate architectures are also generated for each set of requirements refinements. Physical implementations may be suggested by the INFOSEC engineer based upon her knowledge of the system

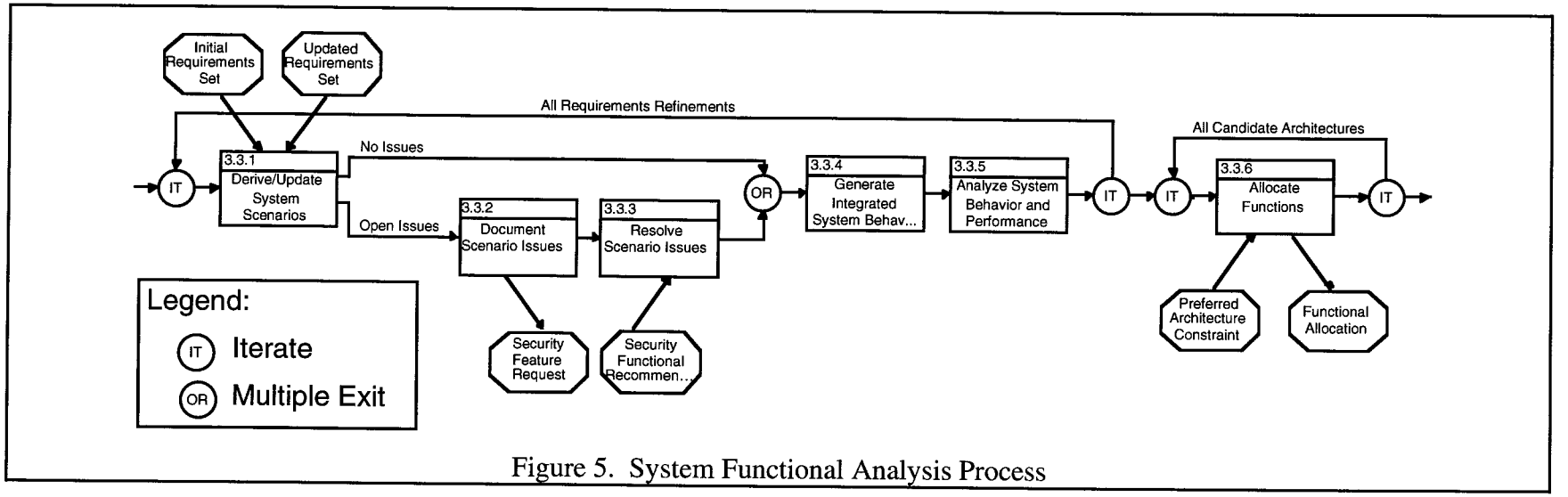

Figure 5. System Functional Analysis Process

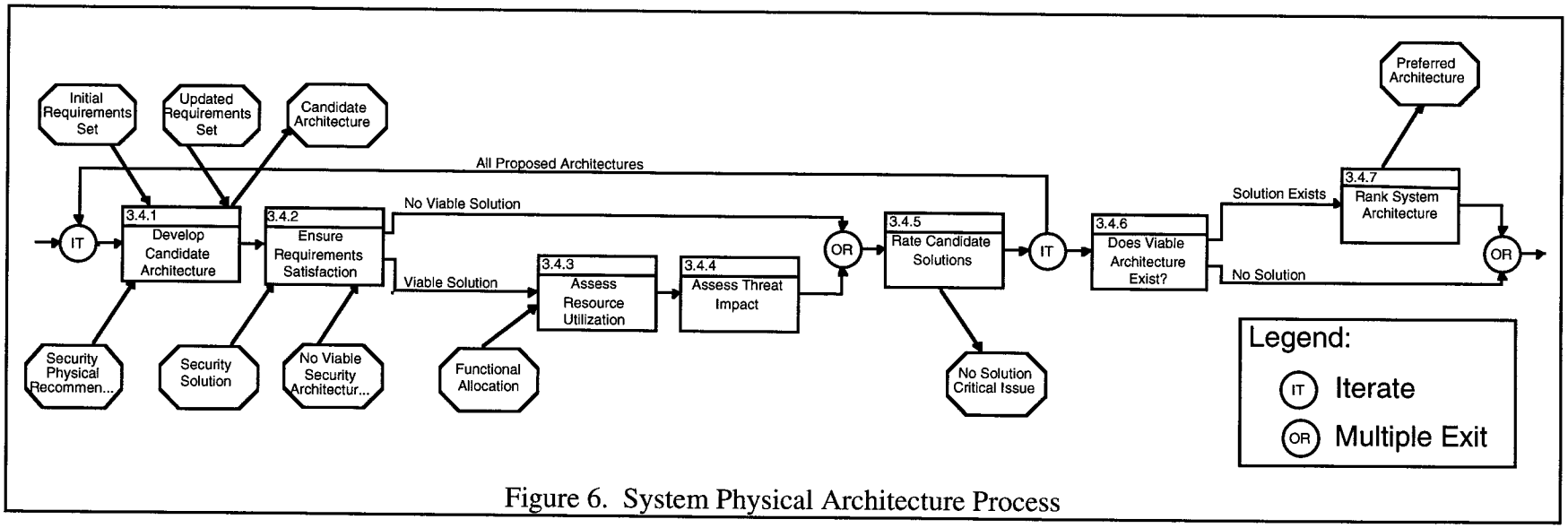

Figure 6. System Physical Architecture Process

security requirements and physical implementations which could satisfy those needs. The candidate architecture is proposed to the INFOSEC engineer who assesses whether that physical implementation satisfies security needs. This allows the INFOSEC engineer to veto an unsuitable candidate architecture and to propose recommendations to the system engineer concerning ways to implement a secure solution. The system engineer assesses the candidate architecture to ensure requirements satisfaction and to achieve concurrency with the requirements branch. The process continues until an adequate solution is found or until the system engineer determines that the system is over-constrained and no physical solution can be found.

The integrated solution is evaluated, validated, and verified at every level/layer of the system engineering analysis. Specialized tools including simulations are commonly used for this activity. Viable alternatives undergo resource utilization and threat impact analyses, and a functional allocation is performed to map the functional solution to the candidate architecture providing concurrency between the functional and physical architecture branches. Results of modeling which profiles the security implementation are used to ensure the architecture satisfies the security requirements.

The successful design is documented at each level of decomposition and the acquisition agency is apprised of the design progress. Upon customer approval, the system design moves to the next level of detail or culminates in the production of the System/Segment Specification. If the customer requests modification to the current layer of design, the concurrent processing of requirements, functional behavior, architecture, and security engineering resumes to provide an adequate solution at that layer. If the customer redirects the program, the system design starts over with the context definition process.

CONCLUSION

Concurrent methodology can aid in the problem of integrating specialty engineering into the mainstream design process. It provides the specialty engineers with well-defined interfaces into the system design during the design process, thus eliminating the need to retrofit specialty solutions into an otherwise completed design. Using the incremental system engineering "Onion Model" provides a low risk method for achieving the appropriate level of design via interim solutions at increasing levels of detail during the specification process.

REFERENCES

Defense Systems Management College, *Systems Engineering Management Guide,* U.S. government Printing Office, Washington D.C., 1989.

Long, Dinsmore, Spadaro, Alford, et al., "The Engagement Logic and Control Methodology as Derived, Defined, and Applied at TRW", unpublished notes, 1968 - 1972.

MIL-STD-499B (DRAFT), *Systems Engineering, AFSC/ASD/EN*, Wright Patterson AFB, OH, May 1992.

ACKNOWLEDGMENT

The authors would like to thank Mr. James Fink for his sponsorship and support.

ABOUT THE AUTHORS

Susan Rose Childers is a Consulting Engineer at the INFOSEC Center for Excellence of the Computer Sciences Corporation. During the course of her career, Mrs. Childers has performed software and systems engineering tasks which cover every phase of the system life-cycle from cradle to grave. In her role as an INFOSEC engineer, Mrs. Childers has developed methodologies which incorporate security into the mainstream of system development.

James E. Long is the President of Vitech Corporation, the developer of the system engineering support tool, CORE™. He has been a performing system engineer and innovator since creating the first behavior diagrams (then called function sequence diagrams) at TRW in late 1967. He played a key technical and management role in the maturing of that technology at TRW. Both CORE™ and RDD-100® are based on the TRW developments.