



**Realizing System Success
Using Model-Based Design (MBD)
in an Integrated Development Environment (IDE)**

Prepared by:

John F. Shaker
(j.f.shaker@novatekinc.com)

Robert C. Wagner
(r.c.wagner@novatekinc.com)

NovaTek Engineering, Inc.
Cleveland, Ohio

November 30, 2004

Introduction

This paper is intended to provide an elaboration of the systems engineering challenges confronting modern technical programs followed by a solution framework that holds the key to successful complex system development. The principal context being addressed in this case deals mostly with aerospace and defense programs, although the concept and rationale advanced applies equally to any complex program undertaking.

The Challenge

The problems related to ineffective system design, development and operations have received wide attention and publication. It is becoming abundantly clear from ongoing studies that the ***cause of system under-performance is poor system engineering execution across the System Development Life Cycle (SDLC)***. Given the dynamics of frequent change and increasing complexity in modern programs, understanding user needs, development of valid requirements, transformation of requirements into design features, verification and validation activities, as well as infusion of risk-based perspectives tend to be deficient in these cases. Exacerbating these deficiencies at the systems level are emergent properties that result from the synthesis of interactions between the components at each level of integration. The net effect of these factors, coupled with constantly shrinking cycle times, is ***poor requirements engineering compounded by inadequate risk identification/mitigation often translates into poor design quality and failed systems***. This eventually leads to unanticipated programmatic and technical failure.

Meeting the Challenges of Complex System Development

Successful creation of effective system designs can be achieved if the development environment properly accounts for all competing extremes across the SDLC and adapts readily to dynamic events and uncertainties. It is a judicious integration of people, process, and enabling technology that is guided and governed by enterprise inputs that direct the transformation of raw materials into value-adding products and services. Paramount to the success of this enterprise “symphony” is the linkage of downstream and upstream management and engineering activities so that holistic designs can be achieved within the constraints of the environment. The basis of the proposed systems development framework, therefore, is ***the fusion of skilled management and engineering practitioners using information and engineering technology in an integrated systems development environment whose foundation is systems engineering principles and processes***. The integration of people, process, and enabling technology in this manner supports disciplined, fact-based decision making across the SLDC resulting in “real” accountability.

Realizing an effective system-level design while addressing current development issues requires an approach that focuses on areas for improving the design process. Based on recent failure reviews, and our engineering experience, it is believed that areas of systems development can be exploited to achieve immediate improvements in design quality, reliability, risk aversion, and overall value. The principal benefit of this approach is that it provides focus on areas that can be leveraged to increase the probability of program success. These areas include: (1) early design definition and analysis upstream, (2) verification and validation downstream, and (3) design insight/oversight at all points in-between. Embodied within this framework at each level and linked directly to the elements are risk identification and mitigation mechanisms. It is important to note that this framework is independent of, and adaptable to any SDLC methodology. The implementation of this systems development framework requires three principal components: (1) a disciplined systems engineering process, (2) a well structured “system model” of the target system, and (3) an integrated development environment (IDE). The implementation and justification of a formal systems engineering process has received extensive treatment in the literature. However, elements two and three of the proposed systems development framework require further elucidation.

Realizing State-of-the Art System Design

A significant critical success factor for exploiting value-adding areas in the SDLC is the utilization of ***model-based design (MBD)*** that supports effective decision making at all levels of development. The principles of MBD include (1) the capture and unambiguous traceability of requirements, from mission objectives through design and test, (2) the embodiment of system functionality & design in a modeling environment, (3) the utilization of “executable” models to simulate system behavior, and (4) model-based risk assessment of various mission scenarios and usage profiles. A model-based context for system design transforms the traditional paper-based systems engineering approach to one that supports extensive evaluation of design trade spaces early in the SDLC. The model-based simulation of operational scenarios allows characterization of system performance with respect to user requirements. Executing simulations in an IDE using a rules-based application of design requirements and performance “states” enables (1) rapid design tradeoffs, (2) a quick evaluation of operations scenarios, and (3) a thorough

examination of design trade spaces and associated risks. In the event of design changes an MBD system can be updated quickly and reliably with no sacrifice to development continuity. The MBD systems models can also be integrated with design artifacts of detailed design and with testing phases of the SDLC for more comprehensive assessments. System design implemented in this way supports balance across the extremes of cost, schedule, performance and risk.

The envisioned IDE will be a high-performing, collaborative system design environment made up of advanced computing, networking, and information management systems. The central purpose of the IDE will be the cross-functional development of a comprehensive “system model” to capture the functional behavior, performance, cost, and risk factors in the emerging design and link it to user need (represented by scenarios).

Beginning with user needs and constraints, system architecture and operational scenarios are developed and used to create high-level system requirements and design. System design and requirements information is input to the system simulation environment, which is comprised of modeling and analysis tools. Once the system models are created and linked with operational parameters, scenarios can be executed and the resulting system performance compared to user requirements. The system parameters can be systematically varied until the desired performance is achieved within operational constraints.

Fundamental to the proposed approach is that risk management becomes an integral part of overall program management. In order to achieve this end, program management and systems engineering risk management processes are integrated to form a complete and coherent characterization of project risk. This will require an integrated risk management process based on advanced information science and technology methods to quantitatively assess risk and to enhance existing processes to mitigate risk continuously throughout the life cycle. A model-based approach provides the best assurance of successfully transitioning to an integrated project methodology.

Another important feature of this development framework is that it can be instituted in a piecewise-continuous fashion starting with improvements in high-level system design and management processes. Once the initial process modifications in these areas have been stabilized, the transformation of other portions of the SDLC such as detailed design, testing, and system qualification can be updated as well. In this way, overall process transformation risks can be better managed because improvements are evolutionary rather than revolutionary.

The specific work products that can be systematically generated using the combined power of skilled professionals, an MBD system engineering process, and an IDE include: (1) early phase program development and risk management plans, (2) validated and documented needs statement traceable to user requirements, (3) multiple views of dependent requirements and their traceability to identified risks, (4) integrated system concepts, architectures and design documents, (5) documented and traceable trade studies and scenario analyses reports, (6) Management Plans, and (7) Web-based design configurations and document management. All of these work products are based upon integrated, accurate and valid system development data.

Tangible benefits can accrue for both management and engineering teams such as: (1) mission assurance due to requirements-driven, systematic evolution and verification, (2) improved visibility and understanding due to common lexicons and representations, (3) reduced programmatic risk, (4) increased accountability, (5) concrete design and organization performance metrics, (6) rapid trade study and scenario analyses, (7) unobtrusive collaboration across the SDLC and the implementing organization, (8) knowledge capture and reuse, and (9) a platform capable of supporting integrated hardware/software development. Each of these benefits, and their combinations, has great potential for increasing the overall value of the systems development process.

Solution Applications

Application of the proposed systems development framework will benefit complex development projects such as NASA’s Exploration Initiative programs (Constellation, Prometheus, Mars/Lunar) and other highly distributed systems with many physical and organizational interfaces. It will benefit smaller programs as well by streamlining the design and development process and relying on engineering artifact “reuse” and development “templates.” The proposed solution emphasizes performance during up-front system definition and downstream validation, which has proven to be the “Achilles-heel” of many programs. In addition to large aerospace programs, the proposed solution schema harbors great potential for application in other industrial sector and defense applications.

Summary

The problem of developing complex systems is one that requires a complete understanding of the technical and management components across the SDLC. Achieving this end necessitates the modeling, analyzing, and verifying of system behavior in the context of its mission environment to ensure stakeholder needs are met. The proposed system development framework can achieve this end by tightly coupling MBD techniques and procedures with an IDE that support rapid and repeatable systems simulation. This approach provides valuable technical and management decision-making support early in the development process when the cost of correcting design deficiencies is a minimum. By addressing issues early in the SDLC it is also possible to build robust systems that are more likely to evolve within program constraints. A salient feature of the proposed systems development framework is that it provides a vehicle for risk management that is integral to the process itself and yields quantitative assessments of risk levels as the program evolves. Finally, the proposed systems development framework represents a basis for transforming inadequate text-based system design processes into something that can support holistic design inquiry.

References

- [1] Columbia Accident Investigation Report, Volume 1, August 2003.
- [2] Defense Science Board (DSB) and Air Force Scientific Advisory Board (AFSAB), "Report of the Defense Science Board/Air Force Scientific Advisory Board Joint Task Force on Acquisition of National Security Space Programs," May 2003.
- [3] "Top Execs Say IT Can Inhibit Growth," Information Week, Eric Chabrow, August 26, 2004.
- [4] B. Blanchard, *Systems Engineering Management*, Second Edition, John Wiley & Sons, 1998.
- [5] B. Blanchard and W. Fabrycky, *Systems Engineering and Analysis*, Third Edition, Prentice Hall, 1998.
- [6] J. Martin, *Systems Engineering Guidebook*, CRC Press, 2000.
- [7] A. Sage, W. Rouse (Editors), *Handbook of Systems Engineering and Management*, John Wiley and Sons, 1999.
- [8] J. Juhasz, *Enterprise Engineering in the Systems Age*, Handbook of Technology Management, Gerald H. Gaynor (Editor), McGraw Hill, 1996.
- [9] L. Baker, P. Clemente, R. Cohen, L. Permenter, B. Purves, and P. Salmon, "Foundational Concepts for Model-Driven Systems Design," *Proceedings of the INCOSE Model Driven Systems Design Working Group*, 1997.
- [10] R. Wagner, "Applications of Modern Structured Analysis Paradigms to the Development of Complex Systems," *Second Annual Technical Symposium, National Council of Systems Engineers*, 1992.
- [11] I. Ogren, On Principles for Model-Based Systems Engineering, *Systems Engineering* 2 (1999).
- [12] S. Wall, "Model-Based Engineering Design for Space Missions," *IEEE Aerospace Conference*, 2003.
- [13] J. Rose, "Space Systems Engineering and Risk Management – Joined at the Hip,"
- [14] S. Guarro, "Challenges of Risk Management as a Systems Engineering Tool," The Aerospace Corporation, 2004.
- [15] M. Guance, D. Bergner, and A. Wong, Design for Safety: NASA is Serious about Risk Assessment,"