REQUIREMENTS MODELING TECHNOLOGY
A VISION FOR BETTER, FASTER, and CHEAPER SYSTEMS

Darrell Barker
Information Directorate (AFRL/IFTA)
Air Force Research Laboratory
Wright-Patterson Air Force Base, Ohio
darrell.barker@wpafb.af.mil

Abstract

This paper discusses a vision, motivation, and objectives for a new engineering technology based on the Rosetta System Level Design Language (SLDL) that we call "requirements modeling". Requirements modeling is what systems engineers and systems analysts do with Rosetta SLDL. Rosetta SLDL is information technology that supports the specification of multi-technology, multi-faceted, multi-discipline systems. Until the Rosetta SLDL emerged, no good way existed to help the design team "get its arms around" all of the requirements for a complex system. Requirements related issues are the most common source of systems engineering problems and they impact all aspects of product development. Requirements models help design teams stay "on the same page" with each other. Requirements modeling reduces cycle time, reduces costs, reduces the probability of overruns, and helps with support and sustainment of the product. Requirements models will be the input to many new and enhanced design automation tools and processes. The benefits of requirements modeling technology are wide reaching, because system requirements touch everything and everyone associated with a product.

1. Introduction

Despite the industry's successes in product and system development, there are some very common and serious problems with systems engineering today. These problems include longer than desired development cycles, frequent and significant cost and schedule inflation, failure to meet user requirements, errors ("bugs") not detected until system integration or after the system is in use, difficulty with identification of suitable off-the-shelf components, parts obsolescence, and (especially for military systems) high upgrade and modernization costs. Even though these problems frequently manifest themselves in hardware or software products, they are really systems engineering problems and can happen in any type of system and with any engineering discipline. This paper discusses a vision, motivation, and objectives for a new systems engineering technology based on the Rosetta System Level Design Language (SLDL) that we are calling "requirements modeling". Requirements modeling is what systems engineers and systems analysts do with Rosetta. Requirements modeling addresses many fundamental systems engineering problems for systems of all kinds. Requirements models will be the input to many new and enhanced design automation tools and processes. Rosetta SLDL is an information technology that supports the specification of multi-technology, multi-faceted, multi-discipline systems. Until the Rosetta SLDL emerged, no good way existed to help the design team "get its arms around" all of the requirements for a complex system. Requirements related issues are the most common source of systems engineering problems and they impact all aspects of product development.

2. Requirements Modeling With Rosetta Addresses Universal Problems

Cost, schedule, and quality problems are certainly not new to experienced program managers and project engineers. Defense programs are notorious for large overruns. But, relatively recent studies show that overruns are by no means unique to defense systems. They are, in fact, more extensive than most people realize. According to findings by the Standish Group International, Inc., on a survey of 8,000 information technology companies, 46% percent of projects were "challenged" – that means that they were completed over budget, past the original deadline, or failed to meet some requirements. Also 28% of projects "failed" – meaning that they were either canceled or did not meet enough requirements to be usable. Therefore, only 26% of information technology programs were completely successful! This survey also showed that the average project exceeded its planned budget by 90% and its schedule by 120%. Therefore, neither the defense-sector nor the private-sector can claim a very impressive track record. Another "reality" is that with "System-on-a-Chip" and "System of Systems" technologies, things are becoming even more complex – thus the system engineering problems will be compounded. This is not good news for the industry. The DoD spends billions of dollars developing electronic and software products every year. The private-sector has a huge demand for electronic products and information technologies, with millions of dollars at stake if new products are late to market. Semiconductor companies need products developed faster and cheaper to keep their fabrication lines full or they also stand to lose billions in the next decade. Large program overruns are eating away both DoD and private-sector budgets and delaying availability of much needed products and systems. Numerous tools are available to aid developers of individual circuits or software programs, but they are insufficient for large and complex systems. Thus, solutions must be found for making large and complex systems better, faster, and cheaper!

3. Requirements Problems -- The Primary Source Of Cost, Schedule, and Quality Problems

What are the sources of cost, schedule, and quality problems? There are several things that lead to programs becoming "challenged". For example, a program may be mismanaged, budgets may be unstable, developers may not have the necessary skills or experience, key people may leave the project, or there may even be a motivation problem with respect to getting the job done or related to the end user's acceptance of a new product and a new way of doing things. But problems like these are most often caused by some combination of incomplete, inconsistent, misunderstood, overlooked, ignored, unverified, unvalidated, changing, or simply an overwhelming number of systems engineering requirements. The larger and more complex the system is and the more people working on the project there is, the more likely it is that a problem will occur. Engineering requirements are the requirements used by the development team to create the system. They are different than, but derived from, the user requirements – such as the military's Operational Requirements Documents and Mission Needs Statements and the ad-hoc documents (if anything is actually documented) used to capture commercial systems' requirements. As an example, the user requirements for a car could be that is fast and sporty. The engineering requirements for the car would specify things such as engine horsepower and body aerodynamic drag that will satisfy the user requirements. Systems engineers and systems analysts create engineering requirements and their Rosetta SLDL specifications.

4. An Information Technology For Specifying Requirements

Requirements modeling is an information technology for making it easier to capture, communicate, track, analyze, verify, validate, view, and manage the hundreds of hierarchical and interrelated engineering requirements necessary for large and/or complex systems. It will significantly reduce the cost of systems development and will reduce the probability and severity of cost and schedule growth (overruns) by enabling the description of human readable and "computer friendly", specifications of a system's engineering requirements. Note that the term "model", with respect to requirements modeling, does not imply a simulation model. A large part of a requirements model is not simulatable. A more general meaning of the term "model" is used here -- an accurate representation of a system. Requirements modeling is not a completely new or foreign concept. Designers have used various forms of requirements modeling for years and these have improved the design process. Some examples of "partial" requirements models are behavioral hardware description language models, algorithm simulations (e.g. Mathworks Matlab) of signal processing systems, block diagrams, state diagrams, and data-flow diagrams. In fact, conversions between other requirements representations (including UML, SDL, and IDEFx) that capture some of a system's requirements (but not all) and SLDL will be possible. "Complete" requirements models, like we are striving for with SLDL, are intended to represent everything that the design team needs to know about the system to develop it. We want a complete representation of the specification. No other modeling technology allows the requirements for mechanical components (e.g. circuit packages and enclosures), analog hardware in the continuous time and frequency domains, digital hardware, software, architecture, interfaces, networks, synchronous events, asynchronous events, information flows, process flows, constraints, verifications, usage assumptions, etc. to be computer modeled together (when consistent with reality). No other modeling technology helps engineers from different disciplines, such as design, manufacturing, test, 'ilities, and human-factors to capture their requirements and inter-relationships. Most modeling technologies allow modeling of one or two of these aspects and for some of these aspects (like constraints) there is currently no wide-use modeling technology available. No other modeling technology allows modelers to create various perspectives and views (called facets in requirements modeling) of the system's requirements. For example, the requirements model of a circuit card assembly might have a functional facet, an architectural facet, a power facet, a size facet, an electrical facet, a connector facet, etc. Requirements modeling pulls together the requirements for all of the other models that are required to develop the system. Other types of models are referenced by the requirements model to show how they fit into the system's development and verification. For example, a requirements model of a digital circuit might reference behavioral, RTL, macrocell, and standard cell hardware description language models and SPICE circuit models. A requirements model is the only place where all the models necessary for a system's development can be represented together.

5. Many Benefits From Requirements Modeling

Requirements modeling will reduce the cycle time needed to develop large and complex systems by making the system's specification easier to work with than the "mountains" of English text documents that many system developers use today. Requirements modeling will reduce overruns by reducing the rework caused by requirements related errors. Requirements modeling will improve the quality of systems because it improves the quality of the specification of a system's requirements and because quality is determined by a system's conformance to its requirements. Requirements modeling will be a key element in achieving an improved collaborative engineering process as it enables concurrent engineering processes to work more efficiently through models that more effectively communicate the engineering requirements and interactions of multi-faceted, multi-domain, multi-technology, and multi-disciplined systems. Requirements models will be used to control specialists such as integrated circuit designers and software programmers. Requirements models will inform them about their portion of the development and how it fits within the overall system. Requirements modeling supports upgrading and re-engineering of systems by offering the sustainers a clear and concise picture of everything that the system does, how it is verified, how it interfaces within itself and with other systems, and the constraints and boundaries that it is expected to operate within. Requirements models will be used by component vendors in lieu of or in addition to data sheets to describe their products. Because system requirements touch everything and everyone associated with a product's development, the benefits of requirements modeling technology are wide reaching.

6. Requirements Models Enable New and Improved Design Automation Tools

Requirements models are human and computer readable engineering specifications. Having system and component specifications that can be interpreted by a computer will open many doors for complexity mitigation and automated analysis and engineering. Requirements modeling technology will bring about many new engineering tools and enhancements to several others. Some completely new tool types that are envisioned include: a) user friendly tools for capturing and viewing requirements models, b) simulation and animation tools for behavioral/operational requirements, c) requirements analysis tools for checking completeness, consistency, redundancy, etc. of requirements models, d) constraint checkers that check the design against the constraint, e) test plan, test pattern, test program and test model generation tools, f) "system" synthesis tools, g) component and intellectual property search and retrieval tools, and h) hardware/software and analog/digital partitioning tools that help determine the best implementation technology to meet the system's requirements. Tool categories that will be enhanced to support requirements modeling are a) requirements management tools, b) formal model checkers, c) formal functionality comparison tools, d) application specific integrated circuit, field programmable gate array, and programmable logic device synthesis tools, e) software autocoding tools, f) cost modeling tools, and g) other engineering analysis and modeling and simulation tools.

7. Dealing With Requirements In Large/Complex System Design

Requirements modeling helps the users and developers learn about the system being designed. The big challenge with requirements of large systems is that they are literally a large complex hierarchical and interrelated mess! It is very difficult to understand and keep track of all of this information, especially if the development team is large and information disbursed. For a large system there may be thousands of engineering requirements. Requirements are hierarchical in that they are abstract and relatively independent of technology at the beginning of the design. Then more detailed, technology- specific requirements are added during the design process as the architecture and implementation technologies are selected. Requirements are often derived from other requirements. Many requirements are mutually exclusive and therefore must be balanced or optimized by the development team as a part of the systems engineering process. There are many stipulations on these requirements. For example, these thousands of requirements can have no redundancies or overlaps; else they will confuse the designers. They must be complete before the design can be complete – a guess about an incomplete requirement can lead to costly re-work. They must satisfy the customer's intent or the system will be worthless. They must be verifiable and be traceable from requirement, to design, to test, to test result in order to determine if the requirement has been met.

They must be realistic or the design can go nowhere. Each requirement will have many relationships and these must be consistent. For example, two electronic components may have an interface relationship. It would be inconsistent for both components to produce output signals that are tied directly together because the components could produce no useful information (not to mention that the output drivers of the components would burn out). It would be inconsistent for a system to have a power constraint of 10 watts and two components of the system to have a power constraint of 8 watts each. Another example is that a requirement should not have multiple sources (i.e. parent requirements); this hinders tracking and tracing. There are many relationships associated with requirements. There are relationships between child and parent requirements, relationships between requirements and design objects, relationships between requirements and test functions, relationship between test functions and test results, and relationships between requirements and configuration item structure, work breakdown structure, and schedule. There can also be relationships between individual requirements. For example, there is a relationship between the power required by electronic components and the amount of heat dissipated. It is small wonder that requirements related items “fall through the cracks” and lead to schedule slips and cost overruns. Requirements modeling aids understanding by assisting in capturing, viewing, checking, and tracking all of these aspects, relationships, and requirements interactions.

8. Managing Requirements

Requirements models can be managed with minor extensions to requirements management tools such as RDD-100, DOORS, and SLATE. All of the above have been demonstrated to reduce development cycle time and cost. This lends additional confidence that requirements modeling technology will provide the advertised benefits.

9. Conclusions

Just like many other industries have turned to information technologies to solve their high volume and complexity problems, systems engineers and analysts must do the same. The solution and the entry portal to better, faster, and cheaper systems is through Rosetta. Because one needs to know the system’s engineering requirements before any other work on system development can begin and because requirements effect everything and everyone associated with the product development, requirements modeling will be the keystone technology for engineering productivity and collaborative engineering. The process of requirements modeling and the greater understanding of the system’s requirements that it enables will improve productivity and resolve error-producing situations much earlier. As a bonus, requirements modeling will usher in a new era of design automation products that will further enhance the system developer’s productivity. Requirements modeling technology is now in exploratory research and development, starting with the AFRL Information Directorate’s sponsorship of the development of the Rosetta SLDL. Several DoD organizations have found requirements modeling technology to be an innovative solution with high payoff potential to some very serious and all-to-common problems. They are each investing a little in developing or applying Rosetta-based requirements modeling technologies; primarily through Small Business Innovative Research and Manufacturing Technology funding. If we are to achieve the vision summarized in this paper, there must continue to be this partnership between government, academia, the electronic design automation industry, the original equipment manufacturing industry, the semiconductor industry, and industry standards organizations that we have started to build. Contrary to what many people believe, challenged and failed projects do not have to be the norm for high-tech systems. Requirements modeling with Rosetta is the key to better, faster, and cheaper systems.

10. Additional Information

Additional information on Rosetta SLDL can be found at http://www.intermetrics.com/sldl and http://www.ittc.ukans.edu/Projects/SLDG/rosetta. Information on the Standish Group Survey can be found at http://standishgroup.com/visitor/chaos.htm.