

Visualizing Verification

Adrian A. Marsh

April 2004

Abstract

This paper proposes extending UML modeling to system verification. Changing the utilization of the UML diagrams will increase the quality of systems requirements, ease the development of verification testing, and result in an accurate graphical depiction of the system throughout the design process.

Keywords

Systems Engineering, Requirements Engineering, UML and System Verification

Table of Contents

1. Problem Statement	3
2. Scope and Objectives	4
3. Current Systems Engineering Design Process	5
4. Proposed Systems Engineering Design Process	9
5. Implementation	12
A. Parallel Verification Diagram	12
B. UML/Verification Extension.....	13
6. Conclusions	15

Table of Figures

Figure 1. Systems Development pathway.....	5
Figure 2. UML system design pathway.....	6
Figure 3. Sample Requirement Matrix.....	6
Figure 4. Sample Verification Matrix.....	7
Figure 5. Flowdown from UML diagrams to Requirements matrices.....	7
Figure 6. System verification tied to UML modeling.....	9
Figure 7. O'Grady's expanding verification pattern.	10
Figure 8. Parallel UML modeling of system and verification requirements	11
Figure 9. Parallel System and Verification Modeling.	12
Figure 10. UML/Verification Extension.....	13

"In the middle of difficulty lies opportunity"
Albert Einstein

1. Problem Statement

Bill is the leader of an engineering team responsible for planning systems verification testing. The division supervisor called Bill into his office and handed over a 700-page document containing systems requirements and verification matrices. He tells Bill the company is crashing on delivering this system and his team has two weeks to compile a test plan, complete with resource and time requirements. Bill, feeling the weight of both the manuscript and the task, wondered if there was ever a crueller fate than interpreting unending pages of verification matrices. As he navigated the cubical filled office, florescent lights blinking overhead, to inform his team members of the good news he thought.... there must be a better way visualize verification.

A comprehensive and effective verification plan plays a key role in the development of any system. In order to be effective, the verification must clearly convey the intended performance of the system and the means for checking this performance. Currently, systems verification management is conducted through the use of numbers and text descriptions, tied to requirements via a verification traceability matrix.¹ The document defines a verification method for each system requirement, and depicts the compilation in a matrix format. This method of describing verification requirements relies on words and numbers to convey the intended performance of the system. As systems become more and more complex, easing the visualization of verification requirements will shorten the development cycle and reduce interpretation errors. One way of simplifying verification management is to utilize graphical representations. Applying the logic expressed by Fred R. Barnard, "One picture is worth a thousand words," to systems verification may be the key to simplifying the current process.

The Institute for Systems Research (ISR), at the University of Maryland, College Park, promotes the use of the Unified Modeling Language (UML) to derive the systems engineering design. ISR recognizes the potential for UML to "visually depict a system before it is realized, in a manner similar to a 'blueprint'" and uses UML modeling to derive system requirements.² This methodology could easily be extended beyond the design of the system to include the development verification of requirements.

There are three potential benefits to extending the UML diagrams to the verification process:

- 1) Allows for the visualization of verification requirements;
- 2) Improves the quality of requirements by ensuring they are verifiable; and
- 3) Improves documentation by maintaining "up to date" UML diagrams throughout the design process.

2. Scope and Objectives

The scope of this research is limited to design concepts and analysis. This paper will present concepts for linking UML diagrams to the verification process. This report proposes the extension of the systems engineering process taught at the University of Maryland, College Park, Institute for Systems Research (ISR), to include the use of UML diagrams to conduct system verification.

The objective of this research is to examine the use of UML diagrams in aiding systems verification. This paper will evaluate the following: 1) examine the current ISR systems engineering design process; 2) propose an alteration to the current process; 3) provide examples of UML visualization of systems verification.

"Everything is in a state of flux, including the status quo."
Robert Byrne

3. Current Systems Engineering Design Process

The ISR model for systems engineering uses semi-formal and formal UML modeling to develop a design pathway (Figure 1)³. The design process begins with the construction of goals and scenarios to create project requirements, which are further analyzed through modeling of system structure and behavior. Traceability of the requirements is mapped throughout the process and results in a system design and final system specifications.

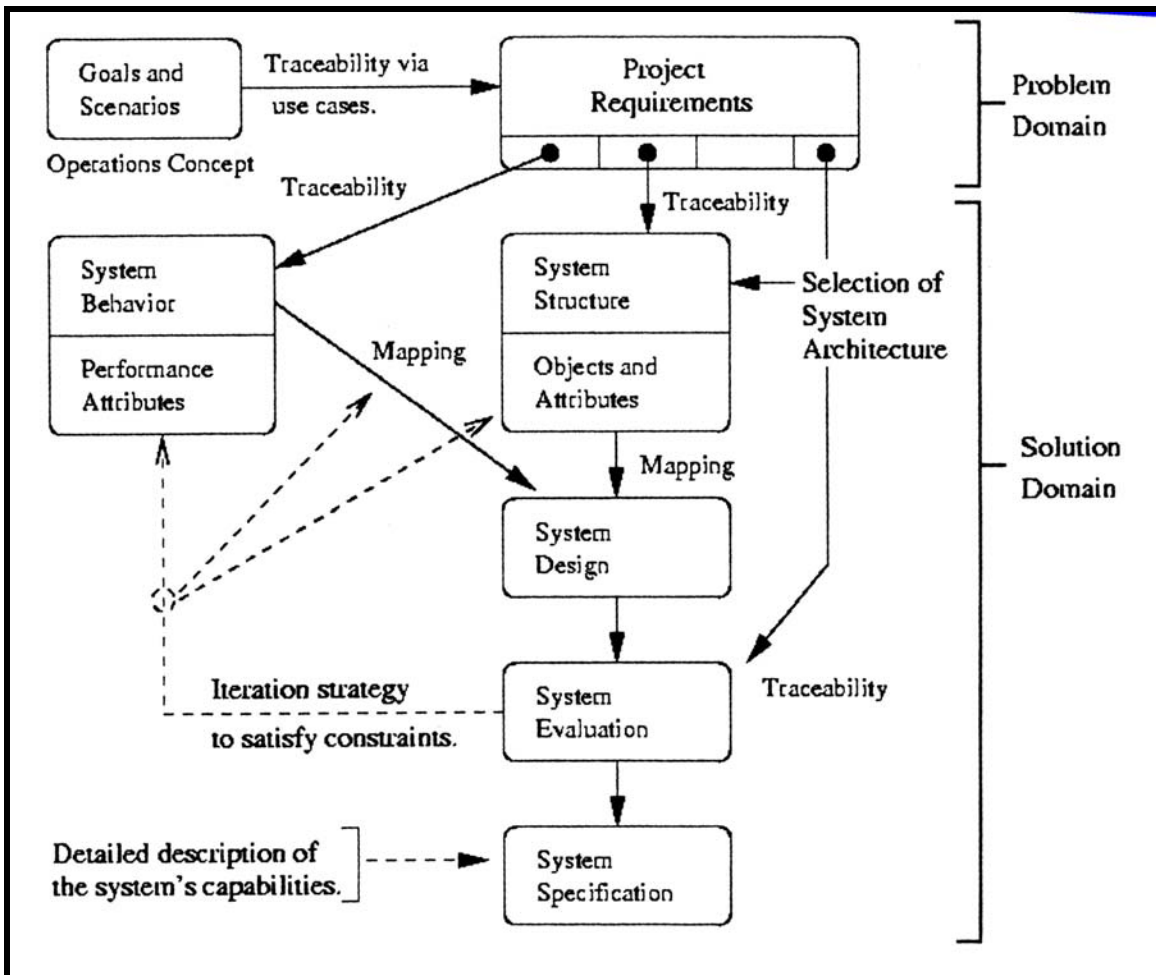


Figure 1. Systems Development pathway. Austin, ENSE 622 Lecture notes, Feb 04, p46.

UML modeling plays a key role in the system development by providing a mechanism to develop and visualize the system behavior and structure. Iterative UML modeling is done in the above process, to analyze the goals and scenarios, through Use Case design, to develop system requirements. Once the initial requirements are

developed, UML modeling is again used to decompose the requirements into system, subsystem and component behavior and structure diagrams (Figure 2).

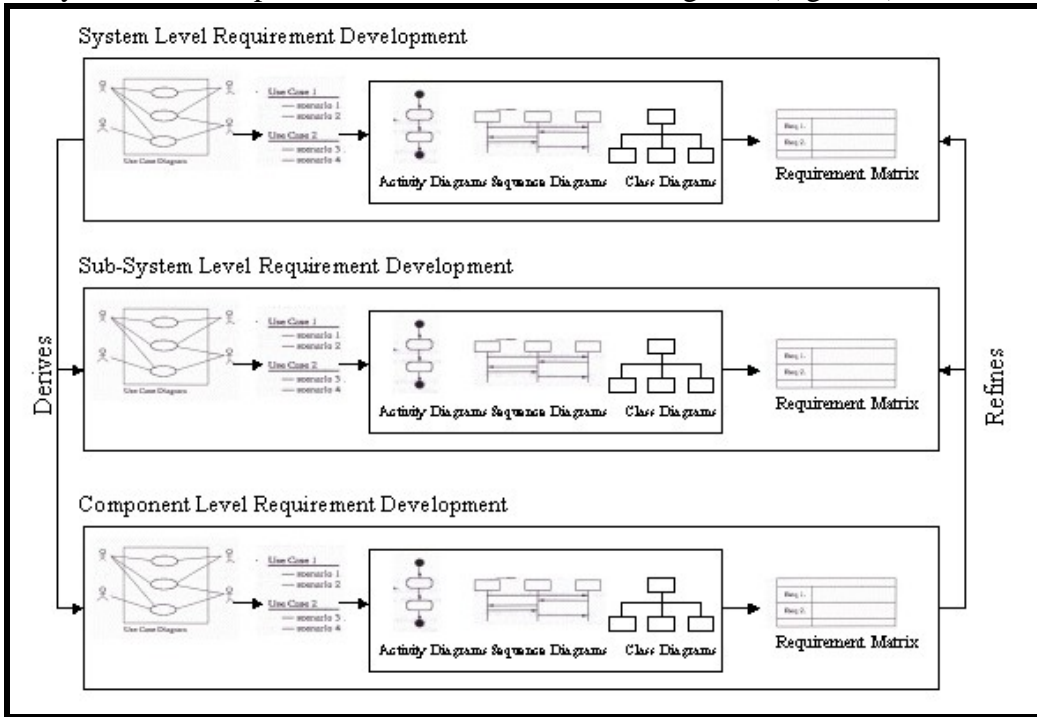


Figure 2. UML system design pathway.

Utilizing UML modeling for system design results in a large quantity of diagrams depicting system structure and behavior at every level of the design. The resultant wealth of graphical models is used to construct a system requirements matrix exemplified in Figure 3. In order to ensure valid requirements, the verification requirements are developed simultaneously with the system requirements.⁴ The verification requirements are captured in a verification diagram, which extends the requirement matrix, and lays the foundation for system testing (Figure 4).

Requirement Num				Requirement Text	Object	SPECIFICATIONS	
Number	Sub Number	Requirement number	Specification Number			MIN	MAX
2	1	3		The system shall allow the crew to conduct a system function check, fire the markers, and apply the safety from inside the marker vehicle	Hand Control Unit		
			3	Controls shall be usable while user is wearing NBC gear		loggies 5', rotary .75'	loggies 5', rotary .75'
			4	Fire mechanism shall be easily manipulated		L=15" to 20"	L=15" to 20"

Figure 3. Sample Requirement Matrix

Requirement Number				Verification Method				VSN	Verification Requirement	VTN
Number	Sub Number	Requirement number	Specification Number	Test	Analysis	Demo	Exam			
2	1	a	0			X		0.002	HCU connected to a test bench dispenser will successfully conduct a function check, fire application and receive safety. Done with over 99% success rate through 20 laboratory test trials.	H002
2	1	a	1				X	0.003	Subjected to a light spectrometer component when powered by system power source will immit a minimum of 50% ambient light	C100
2	1	a	2			X		0.004	Human Factors Requirement- must follow logical progression -- provide a tests election of 10 soldiers, unfamiliar with the system. All soldiers can successfully use the equipment after receiving a 10 minute block of instruction.	H002

Figure 4. Sample Verification Matrix

As shown the current process relies on UML diagrams heavily to generate system requirements, however, they are seldom referenced again once the requirements are captured. The development flow from modeling to requirements documentation is a one-way interaction with models resulting in requirement text (Figure 5). Typically, the UML diagrams are not updated once a change to the requirements text occurs. Failure to utilize these diagrams throughout the design process, specifically tying them to system verification, results in lost opportunities and redundant work though design implementation.

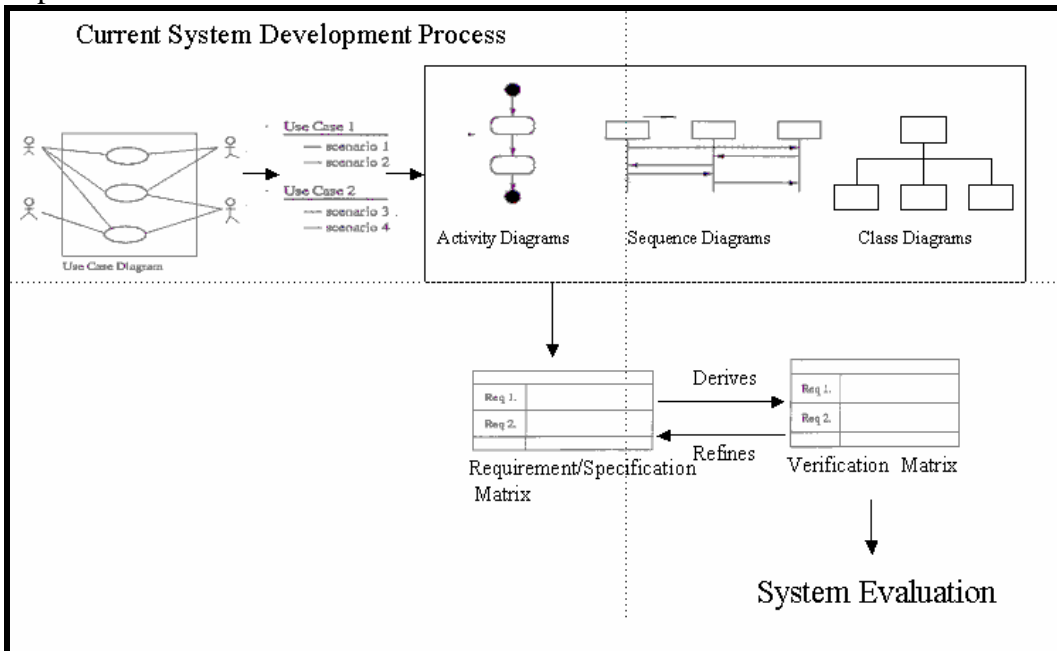


Figure 5. Flowdown from UML diagrams to Requirements matrices.

Firstly, the design team will spend hours creating numerous graphic diagrams depicting the system behavior and structure, only to discard them once the requirements

are generated. As inevitable changes or refinements to the design occur, they are captured on the requirements matrix; however, they are seldom reflected on UML diagrams used to initially create the requirements. There is little incentive to modify diagrams as refinements are made, because they have no long-term use in the current design process. The end result is UML diagrams that do not accurately reflect the requirements and verification matrices at the end of the design process.

Secondly, the current design method ultimately relies on requirements/verification matrices to convey the designer's intent for the system. The tedious format of this material, primarily consisting of words and numbers, presents an interpretation challenge for subsequent design teams or testers to fully understand the intended workings of the system.

Finally, the current process does not force the engineer to construct verification methods, while developing requirements. Although suggested, in order to ensure quality requirements, the parallel development of requirements and verification methods is up to the individual designer. The practice of writing system requirements, then at a later date returning to write the verification requirements, generally leads to invalid or unverifiable requirements in the first place, resulting in lost time and resources.

This report will now examine how the current systems engineering process could be altered to improve the usability of UML diagrams created in the initial design phase of development.

"Efficiency is intelligent laziness"
David Dunham

4. Proposed Systems Engineering Design Process

The proposed change to the systems engineering design process extends the usability of UML diagrams beyond the creation of requirements. The new design method follows the development flow depicted in Figure 2, and adds a link between verification and modeling (Figure 6). The linking of UML diagrams and verification creates a UML diagram "cradle to grave" development process by extending the life of the models through the development process. Aside from maintaining an accurate model of the system, this approach to verification will likely increase the initial quality of system requirements and aid in the visualization of testing requirements.

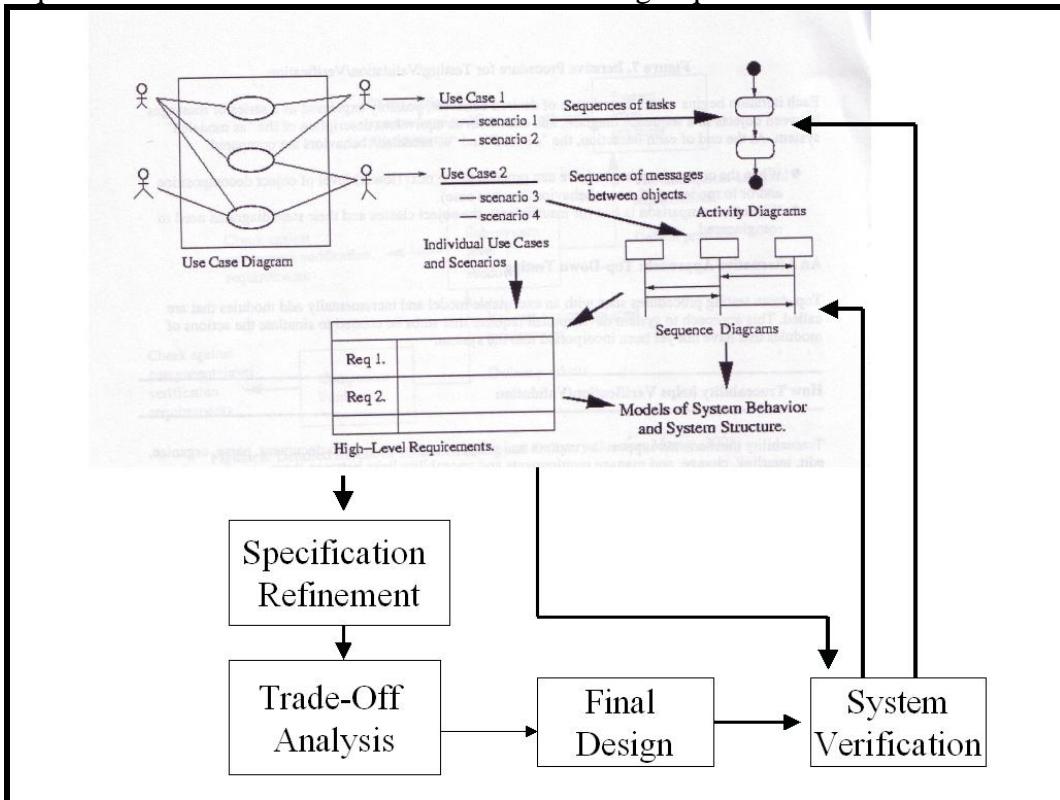


Figure 6. System verification tied to UML modeling

The first step in demonstrating the use of UML diagrams for supporting systems verification is to review the verification process. O'Grady stated, "one of the most effective ways to insure that good requirements are written is to require they be verifiable through some practical process."⁵ He also advocates the simultaneous creation of system requirements and verification methods through all levels of the systems design (Figure 7).

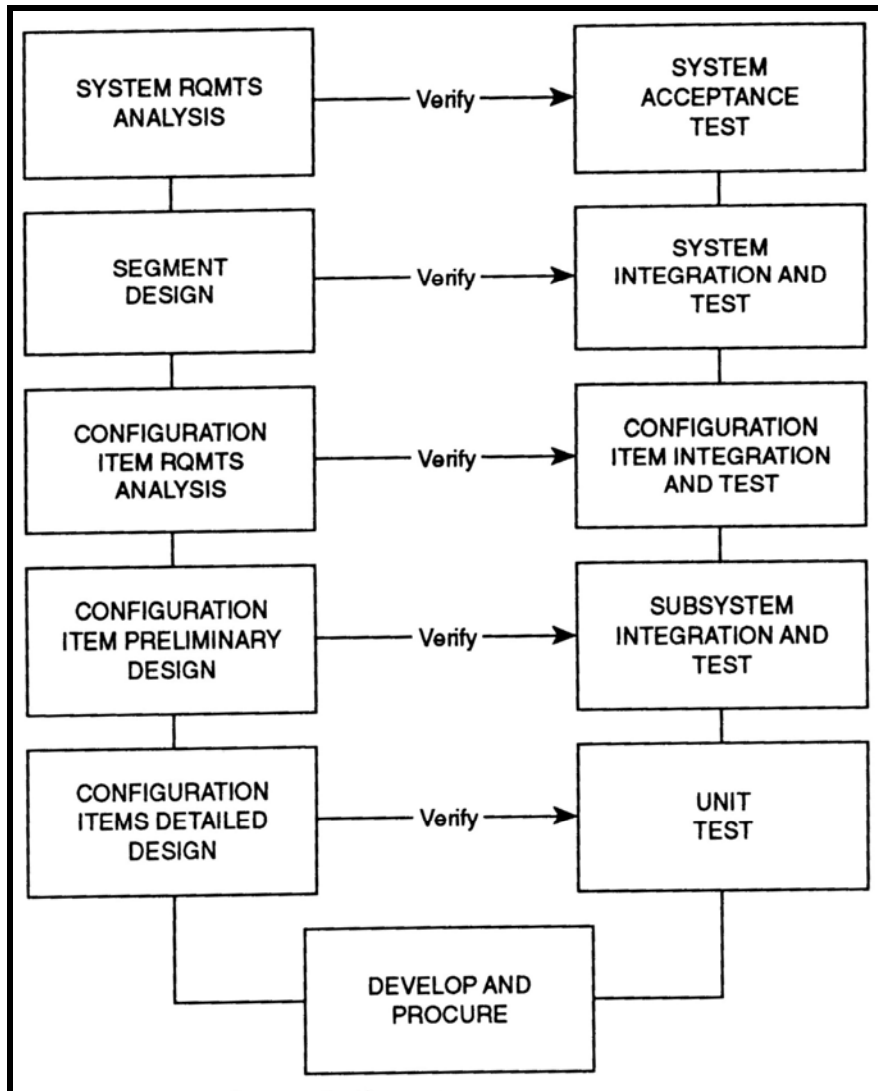


Figure 7. Expanding verification pattern. O'Grady, *System Requirements Analysis*, p.423.

The proposed design methodology will promote the simultaneous construction of system and verification requirements by utilizing UML diagrams. The linking of UML diagrams to verification will extend their life beyond the conceptual and into the practical. The revised process calls for the simultaneous construction of UML diagrams along with a verification diagram depicting the testing method (Figure 8).

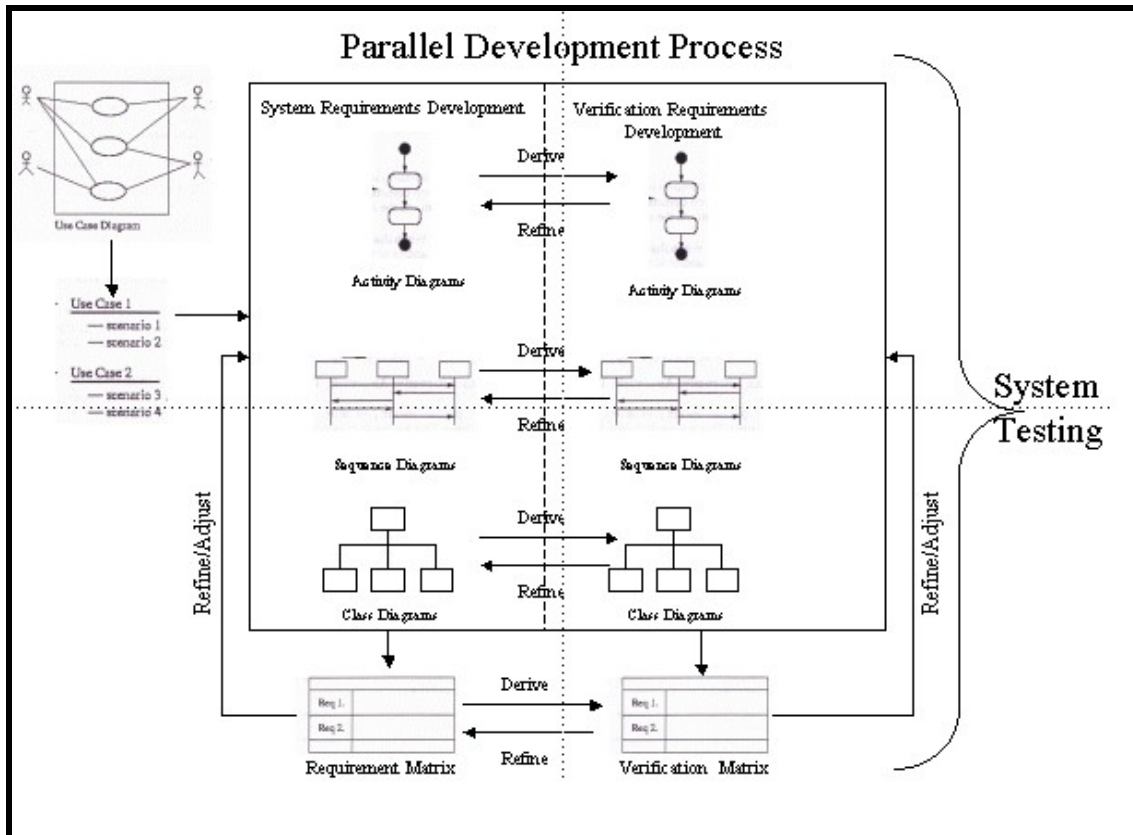


Figure 8. Parallel UML modeling of system and verification requirements

As indicated in the above diagram, UML models would be constructed for the systems behavior and structure along with a parallel diagram or description outlining the verification methods for the system. There are several benefits for this approach to the system design process.

Firstly, the marriage of UML modeling for both the system and the verification of the system will ensure the models stay current through the design process. As indicated in Figure 2, the design of a system is an iterative process that is in constant refinement. The proposed method will ensure that system models are refined as the verification model is constructed, and vice versa. The process of simultaneous modeling of the system and its verification method will keep the models current and a valuable resource for further development.

Secondly, the marriage of the two diagrams will increase the quality of the modeled requirements. By taking the time to model the verification method, one can easily ascertain if the system model is indeed a verifiable, quality requirement. This additional step will allow the designer to graphically step through the verification process, to determine the effectiveness of the modeled requirement.

Additionally, the modeling of the systems verification method will reduce confusion when the system goes to testing. Stephens stated: "words alone offer a poor choice for portraying the structure and relationships" of a system.⁶ By providing the testers of the system both a verification matrix with accompanying UML verification models, the testers of the system will easily grasp the designer's intended system performance.

"I must create a system, or be enslaved by another man's"
William Blake

5. Implementation

Having reviewed the benefits of concurrent modeling of system and verification requirements, the question remains how to implement the ideas into practical design. The problem with utilizing UML modeling for systems verification is that UML was not designed for testing analysis, rather for system design.⁷ However, since the output of the UML process is verifiable system requirements, it is logical to extend UML models to represent the ultimate verification of the requirements.

We will now examine methods for visually representing verification in conjunction with UML modeling. There are two methods for UML visualization of verification information: the parallel verification diagram and the UML/Verification extension process. Both methods are intended to augment, not replace, the verification matrix.

A. Parallel Verification Diagram

The parallel verification diagram creates two graphical depictions of the system requirements and verification measures. The first UML diagram is created through the design process and demonstrates the functionality of the system. The second diagram uses a mirror format to present the requirement verification steps (Figure 9).

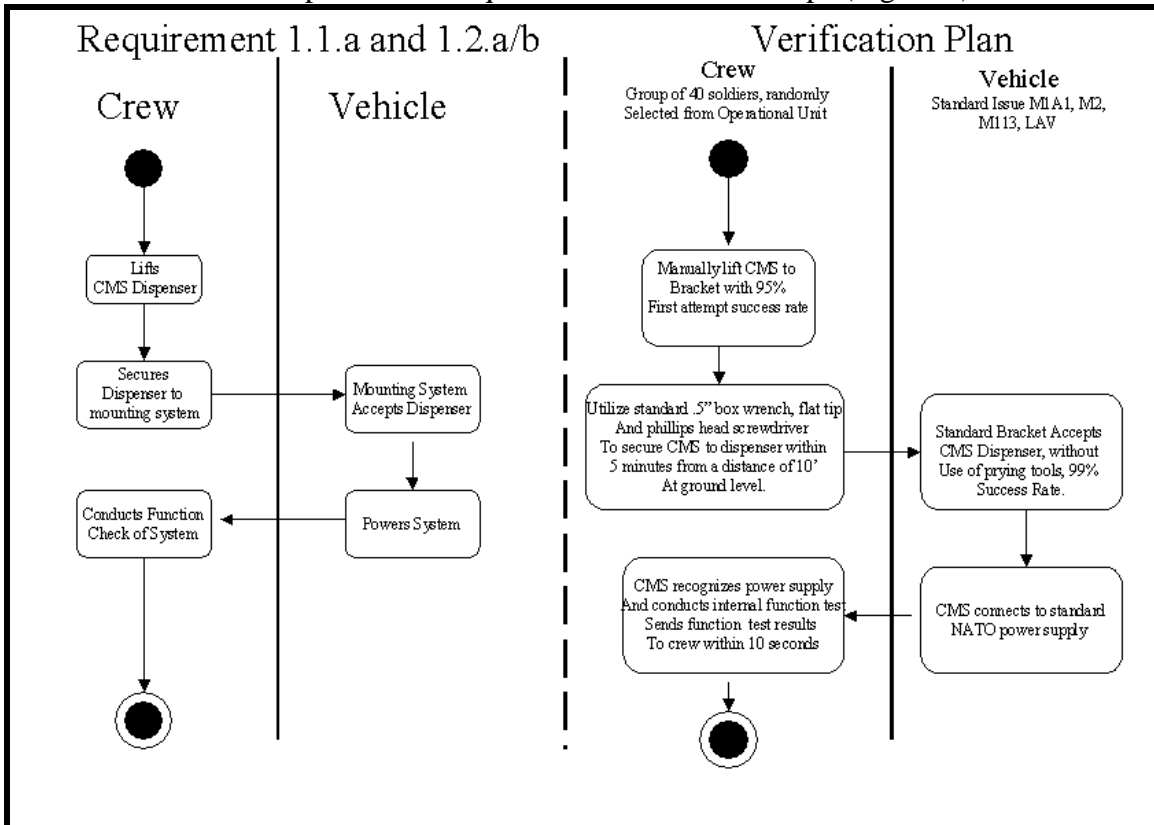


Figure 9. Parallel System and Verification Modeling.

The intent of the parallel verification diagram is to prompt the systems engineer to design the verification steps concurrent with system modeling. The designer will use the same design functionality flow to depict the verification steps. This will ensure the requirements derived from the process are verifiable, as both are done concurrently.

Parallel formatting of system verification steps will benefit the testers of the system by simplifying design interpretation. Graphically depicting system functions alongside representations of system verification reduces confusion inherent in interpreting a verification matrix. Additionally, this method of formatting will ensure the diagrams used to create the system requirements remain current. As changes to the requirements occur, the designer will adjust both the UML model and subsequent verification test plan.

B. UML/Verification Extension

The extended version of the UML diagram consists of the standard UML diagram depiction, mirrored with the verification data. This approach maintains the UML depiction of the system and augments it with a textual based verification plan (Figure 10).

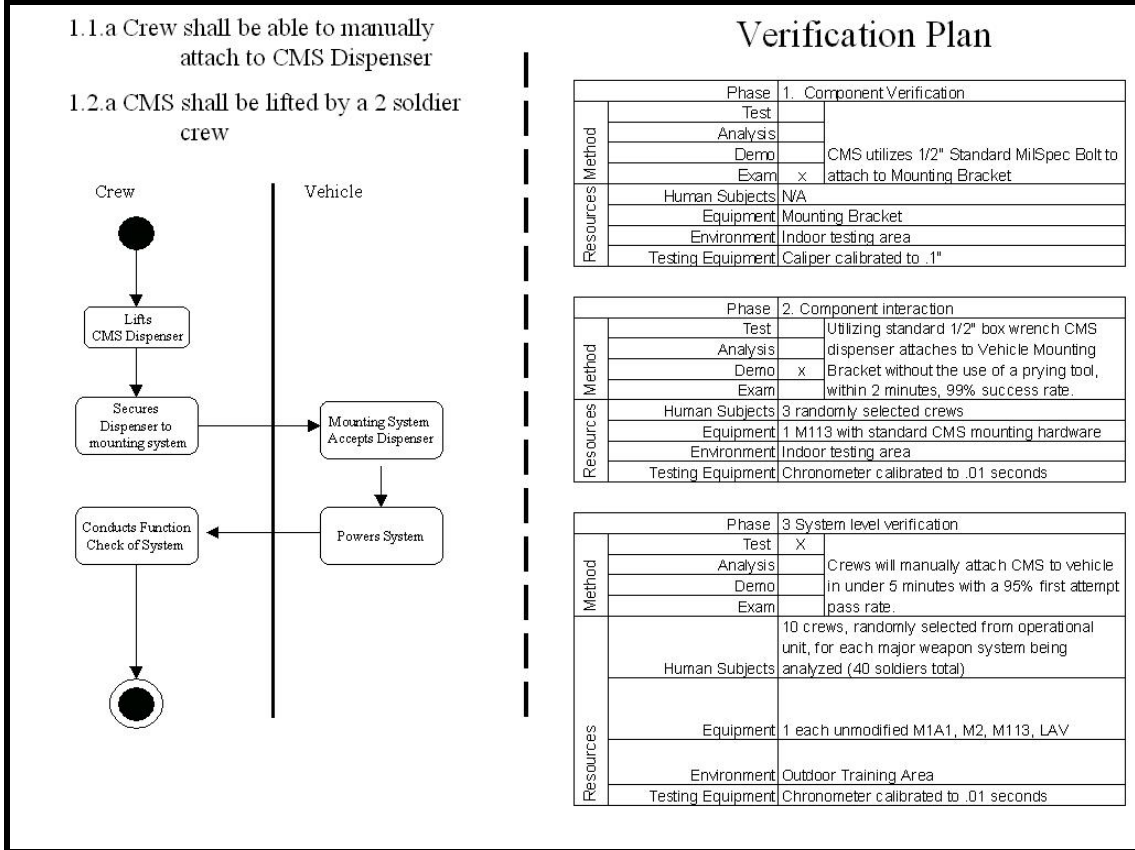


Figure 10. UML/Verification Extension

The UML/Verification extension model retains the benefit of parallel development, causing the designer to input a verification plan concurrently with the system model. It replaces the UML model of verification with a textual description of the verification process. This method allows for more information to be captured in the verification design, including resource and time estimates along with the verification

plan. Instead of graphically depicting the verification process, it allows the UML model to convey the desired functionality, and augments the graphic with textual verification measures.

The UML/Verification extension is intended to benefit testers of the system by allowing them to visualize the intended functionality of the system via the UML model. The verification steps outlined by the designers can easily be converted into a comprehensive system test plan shortening the development cycle of the system.

"It must be remembered that there is nothing more difficult to plan, more doubtful of success, not more dangerous to manage, than the creation of a new system. For the initiator has the enmity of all who would profit by the preservation of the old institutions and merely the lukewarm defenders in those who would gain by the new ones."

Niccolo Machiavelli

6. Conclusions

Integrating UML modeling and system verification will enhance the current ISR systems engineering design process. The marriage of UML models and verification will increase the quality of the system requirements, improve the usability of the verification plan, and keep diagrams current throughout the design process.

Currently, ISR students are working to automate UML systems engineering management through software development. The ideas presented in this paper can easily be integrated into this UML system design software initiative. By linking the requirement design to verification via UML modeling the software has the potential to capitalize on the information value of graphical models.

The end result of this integration will be UML models that are usable beyond the solution domain and into the systems testing and production domain. Capitalizing on the graphical models and linking them to the textual documents in a program format will greatly ease the conveyance of the project information.

The intended automated program will greatly simplify the task assigned to Bill, our systems analyst. He will be able to click on a requirement and simultaneously view the verification measure. The program will also allow Bill to view the requirement functionality via the UML model and access a graphical representation of the verification steps. Linking the design process to verification planning will ease the systems process, reduce overall development time, and produce a better end result.

¹ O'Grady, Jeffrey, "System Validation and Verification" p45.

² Austin, Mark, "ENSE 622, Requirements, Design, Trade-Off Analysis, Lecture Notes", Feb 2004, p108.

³ Austin, Mark, "ENSE 622, Requirements, Design, Trade-Off Analysis, Lecture Notes", Feb 2004, p46.

⁴ O'Grady, "System Requirements Analysis", p416.

⁵ O'Grady, "System Requirements Analysis", p416.

⁶ Stephens, Sharon A. "Requirements Expressions and Verification Aid" p101.

⁷ Darryl K. Taft "Developers expect nod for UML 2.0 standard" eweek, May 26, 2003, p10.