Conceptual communications: The key to engineering valid systems and models

Randall N. Brooks, *Graduate Student, Old Dominion University*

***Abstract*—Verification and validation provide benefits both to system engineers and simulation engineers. Common verification and validation (V&V) challenges of both professions are explored and the differences examined. The need to gather requirements, perform systems design, and implement the design before validation can be attempted is a problem that creates risks to the overall project success. It is suggested that conceptual barriers exist that prevent unambiguous communication during early project phases. These barriers to understanding are discovered later in V&V activity. They add cost to a project at best or completely derail the project at worst. Ways to recognize conceptual barriers that lead to misunderstanding are examined and ways to limit the impact on an engineering project are explored.**

***Index Terms*—Verification, Validation, Systems Engineering, Modeling and Simulation**

I. Introduction

SYSTEMS engineers place early emphasis on verification and validation (V&V) as part of the systems engineering methodology. Introducing V&V planning activities early with stakeholder involvement from the beginning of the simulation study will produce the best simulation model outcome. M&S engineers need to recognize that verification and validation are an integral part of the whole process and has the highest significance when measuring a simulation project's success or failure.

Systems Engineers are concerned with system verification and validation and Modeling and Simulation (M&S) Engineers focus on simulation model verification and validation. They both have common interest in the V&V problem and seek to determine whether they are creating an accurate representation of a system. Validation is said to be one of the most challenging problems in building simulation models [1]. Some systems engineering professionals ask if V&V activities are a help or a hindrance since these activities could be considered as overhead or non value added steps in the process to produce systems [2].

The process of determining model reality to the model's useful representation is referred to as the validation process. Law warns that validation is often attempted after model development when little time and money remain [1]. Other authors of simulation products have sited that V&V is often neglected for reasons, such as, time and budget pressures, laziness, overconfidence, and ignorance [4]. Both SE and M&S professionals will agree that finding and fixing errors early leads to less schedule delays, cost overruns, and a more satisfied customer. M&S engineers have the addition burden since model is an approximation to an actual system; therefore absolute validity is not possible [1].

Verification involves building the system correctly [2]. It is a less subjective iterative process because formal testing can find error or inconsistency in data when compared to the validation process. There are two types of verification errors when working with simulation models [4]. The first error type is syntactical, indicating that data format or logical rule errors are embedded in the model. The second verification error type is more elusive. These are semantic errors that are associated with meaning or intent of the group building a system or model. An example of a semantic error is referring to a weapons system being in an idle state. The intended meaning of an idle state may mean that the component is simply doing nothing or waiting. Idle could justifiable mean waiting, such as, waiting for arming or waiting for maintenance or other unknown states that needs clarification because it will affect the behavior of the system. The problem is reminiscent of asking an out-of-view child what he or she is doing and receiving a verbal "nothing" response. I immediately question the meaning of doing "nothing." Unclear semantics leads to unclear model behavior. This is a particular difficult problem with distributed simulations that are created without the knowledge of another group's semantic reference.

II. Comparing System Engineering to Modeling & Simulation Engineering

A. *Common Activities*

Simulation studies focus importance on requirements gathering and data correctness. Additionally, architecture and design play importance roles to extend the finished product in terms of interoperability and composability. However, an invalid model is useless, regardless of decorations used to sell the model to stakeholders and sponsors.

System engineering activities are tightly aligned to corresponding modeling and simulation activities. The goal differs since system engineers create a real system as the customer deliverable. Modeling and simulation is interested in providing answers to a question. The deliverable is often an analysis of the simulation output with recommendations that

Manuscript not submitted. November 30, 2009.

R. N. Brooks is a graduate student at Old Dominion University, Norfolk, VA. Phone: 757-513-9638 E-mail: randynbrooks@gmail.com

have been summarized or condensed. Common activities to both engineering sciences are the following:

--Define problem / objective
--Develop concept of operations
--Design system / model
--Define architecture / framework
--Implement design
--Execute system / model
--Verify build correctness
--Validate usefulness for purpose
--Deliver system / results

B. Common Challenges

In addition to common activities there are common implementation challenges. Both SE and M&S deal with common challenges listed below:

--Ambiguous requirements
--Schedule constraints
--Cost constraints
--High system complexity
--Changing requirements
--Multi-discipline workforce
--High cost of failure (loss of life)
--Creditability of deliverable

Systems verification and validation activities produce feedback into the systems engineering project design. This constant verification and validation feedback helps to ensure success in systems engineering. M&S professionals can learn from this observation. Models should be designed with V&V as a primary concern and feedback channels explicitly provided. One behavior of engineers is to seek perfection in their work. Mapping the *Concept of Operations* to model behavior that will certainly contain known imperfections may be difficult for M&S professionals. The problem lies in the difficulty of overcoming conceptual barriers [8] in both professions.

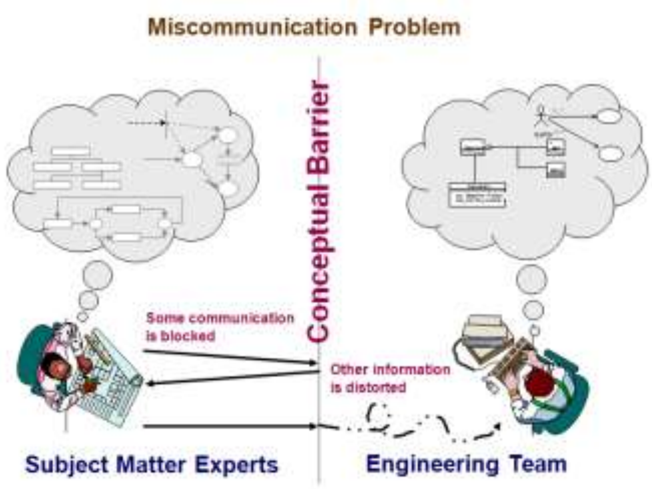

Fig. 1. Conceptual barrier impede flow of ideas between team members [8].

A conceptual barrier is defined as a communication obstacle of a mental representation between groups or individuals. The transfer of a concept is impeded. Figure 1 shows a graphic representation of the situation. An IBM study of 1,500 change management executives found that the biggest obstacle to a project's success was changing mindsets and attitudes [7]. Even a fully dedicated and experienced labor force may have problems overcoming the conceptual barrier of a project as it may require changing one's viewpoint and perspective.

C. Recognizing Imperfection

System engineering involves developing concrete systems. M&S engineers work to develop live, virtual, and constructive models that are abstractions of reality. Even live model are most likely a mock reality of some sort. M&S engineers may be required to design imaginary systems or operate using abnormal parameters to experiment with unusual system behaviors. M&S engineers recognize the model of a system and the output data is imperfect. Unique challenges for M&S engineers that differ from System Engineers are the following:

--Models are imperfect abstractions
--Models may be imaginary or future system
--Reliance of stochastic data, processes, output
--Final product is intangible

How can system engineering verification and validation principles apply to modeling and simulation work when such differences exist? To answer this question we look at the Systems Engineering Vee chart [6] shown in figure 2. You can see that verification and validation feedback from the right to the left hand side of the chart. *The problem is we are attempting to create a verifiable and valid system that has yet to be built.* Constant feedback from expected future activity is needed during *Project Definition* activities to break down conceptual barriers before implementation is started.

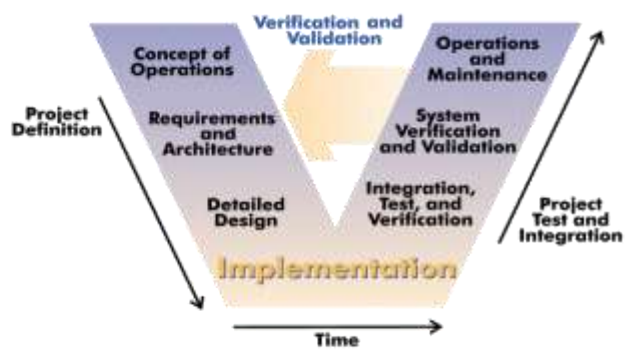

Fig. 2. The SE Vee chart shows the process from the original concept to the operational system. V&V link the two halves of the pre-implementation activities to post-implementations activities.

The problem is also found in software engineering by Alistair Cockburn [5]. He states, "An inescapable and unpleasant fact of system development is that the earlier and more crucial design decisions can be validated only after some later and less important decisions have been addressed and validated." He discusses the waterfall design process as a *fact of life*. The phases of the development processes are defined as: requirements, design and code. Each phase flows down into the next process. Figure 3 show a diagram of Cockburn's process flow and mapping.

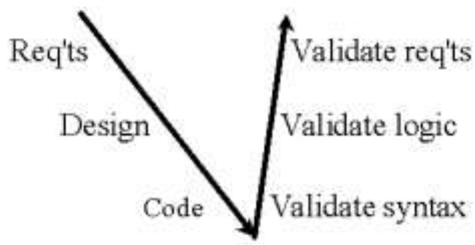

Fig. 3. The waterfall "fact of life" according to Alistair Cockburn.

He calls this the *validation v* as each process maps to a corresponding validation activity on the right side of the vee. This blurs our definitions of verification and validation as both system and simulation engineers view these as primarily verification activities. He suggests that the vee model overlaps as originated from Forsberg and Mooz [9] thus creating a repeatable series these activities. The following observations are made by Cockburn:

--The validation v is unavoidable.
--Developers get better at the end of each cycle.
--The project needs a steady stream of results.
--The process can improve during the project

III. Barrier Reduction techniques

Figure 4 shows the goal of achieving unimpeded communication. The most important finding is the observation that developers improve with each cycle. I believe the reasons developers get better at the end of each cycle is because the conceptual barrier begins to breakdown.

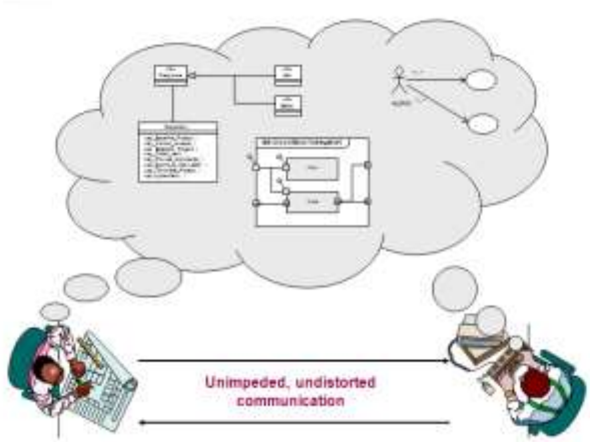

Fig. 4. Removal of conceptual barriers permits mental communication between team members to flow [8].

Cockburn's theory should apply to system and simulation engineers during the critical conceptual model development phase of the project. Recall the IBM study that listed mindsets and attitudes as the biggest challenge to projects involving change. This challenge is also found in systems and simulation engineering. Some sort of change will be implemented by creating a new system. The impact of simulation results may lead to concrete changes in business process.

Ideas to prevent conceptual barriers from forming early in the project are the following:

--Use common modeling language
--Co-locate team members
--Promote ambidextrous mental abilities
--Down play rank and hierarchy

A. Common modeling language

Use of a modeling language known to both engineers and subject matter experts (SME) can provide universal translation of each other domain knowledge. Good examples are IDEF0, UML activity, UML state diagrams, and UML use case diagrams. A secondary benefit of a common modeling language is that all team members begin to use common terms and expressions. These can be refined through paraphrasing to gain agreement of meaning and understanding.

Experience has shown this can be difficult as SMEs view the need to learn new subjects somehow detracts from their expertise. They may define themselves as military veterans that have fixed ideas and not be flexible when unfamiliar material is presented. Likewise, engineers may not be familiar with the semantics or context of the problem that the team is attempting to solve.

B. Co-locate team members

The team should be closely located or virtually located via video or other communications. Many project fail due to lack of SME involvement. They should be fully dedicated to the project. This is difficult because domain experts can be highly compensated individuals that have limited availability (business or military obligations), or be geographically dispersed. I recall an example of this problem while working on a flight operations model for aircraft carriers. A requirement existed for airplanes to taxi on the flight deck prior to catapult launch. The requirement seems simple enough at face value. During the inspection phase of validation it was discovered that certain types of aircraft cannot be waiting in front of others aircraft (e.g. E2C turbo prop turbulence will extinguish an F-18's aircraft's engine). This revelation was widely known by SMEs but never understood until very late in the validation phase by the model developers. As a side note, the scenario picked aircraft according to a rule-based although random process; hence the problem only appeared in certain situations.

C. Promote ambidextrous mental abilities

The most productive team members are able to overcome conceptual barrier with ease. They tend to be equally comfortable speaking about domain knowledge or technical system development. Management should recognize these individuals with ambidextrous mental abilities and use then to assist less flexible team members. They tend to act as a bridge to cross the conceptual barrier. Many times they can synthesize a difficult concept through the use of pictures. A pattern has been suggested that really smart people in a specific domain do not have the ability to communicate through all levels of understanding [10]. This problem can make the situation worst, as the inability to convey knowledge to others can make the information seem more complex [10].

D. Down play rank and hierarchy

Subject Matter Experts may be retired or active military or senior staff in leadership positions that are accustomed to making command decisions. More damaging is that their opinion may hold higher favor with stakeholders when compared to engineering professionals. They cannot command or decry a particular viewpoint to a problem. Making a model with higher fidelity does not make it more or less valid. The underlying concept of the project must be agreed upon by general consensus and mutual understanding.

Often team members are a mix of degreed professional and non-degreed individuals. There can be generation gaps and other hierarchies that effect working relationships. In my experience, the mindset has been that newly minted graduates must pass inspection from seasoned but non-degreed subject experts.

SMEs have important *tribal knowledge* about the way systems really work. Vietnam era flight operations experts related to me how helicopters in need of repair and located in critical deck locations were routinely discarded overboard [11]. This made room for approaching aircraft that were low on fuel to land safety. In tribal language a helicopter in this state was called, *going palm tree*, since it has the appearance and usefulness of a palm tree on the flight deck with its rotor blades extended (i.e. fully extended it takes up a large amount of critical space on the flight deck). Engineering professional have a tendency to reject field tested *tribal knowledge* as they are more comfortable with procedure manuals depicting defined processes that have limited exceptions. The team must recognize that no one person has either the depth or width of knowledge to develop the system alone. It takes a team.

IV. Conclusion

Systems engineers benefit by seeing the creation of their work being built and used in production. The validity of these systems becomes obvious through deployment and use. M&S engineers rely on execution of the model to prove it is a valid representation of a system. Validation of models is extremely importance for M&S engineers due to the abstract nature of their work. Both must deal with conceptual barriers to understand the mental picture of what is being created and how it will function in the intended context.

Live, virtual and constructive models may have stochastic behavior making outcomes naturally random. The problem of validation is furthermore amplified for M&S engineers since newly conceived systems exist only in the collective imagination of the team. By using conceptual barrier reduction techniques some of the difficulties faced by development teams during early project phases can be minimized.

Acknowledgment

R. N. Brooks thanks Thomas Fargnoli, a colleague and author whom unknowingly inspired me to research the subject of this paper.

References

[1] A. M. Law, W. D. Kelton, *Simulation Modeling and Analysis.* Tucson, AZ: McGraw-Hill, 2000 pp. 264-265.
[2] D. M. Buede, *The Engineering Design of System*. Fairfax, VA: John Wiley & Sons, 2000, pp. 13, 123–135.
[3] C. S. Wasson, *System Analysis, Design, and Development*. Hoboken, NJ: John Wiley & Sons, 2006, p 707.
[4] C. Harrell, B. K. Ghosh, and R. Bowden, *Simulation Using ProModel*. McGraw-Hill, 2000, pp. 175-177, 275.
[5] A. Cockburn, "Managing Increments and Iterations with 'V-W' Staging." MS PowerPoint, Available: http://Alistair.Cockburn.us
[6] Image extracted from *Clarus Concept of Operations.* Publication No. FHWA-JPO-05-072, Federal Highway Administration (FHWA), 2005
[7] IBM Global Services Survey: *Making Change Work*. Oct 2008, Available: http://www-935.ibm.com/services/us/gbs/bus/pdf/gbe03100-usen-03-making-change-work.pdf
[8] M. Chonoles, "Introduction to the Model-Based System Engineering Method," Microsoft PowerPoint, Center for Performance Excellence, Lockheed Martin, 2000, unpublished.
[9] K. Forsberg and H. Mooz, "Application of the 'Vee' to Incremental and Evolutionary Development," Proceedings of the National Concil for System Engineering Symposium, St. Louis, MO: July 1995
[10] T. Fargnoli, *A bridge to simplicity through diagrams*. Bloomington, IN: Authorhouse, Jan. 2008, p-2
[11] R. "Duke" Ellington, discussions on flight deck operations. Newport News, VA, 2005

Randall N. Brooks holds a Bachelors Degree in computer science from University of California, Chico, California in 1981. He is currently pursuing a Masters of Engineering degree in modeling and simulation at Old Dominion University, Norfolk, Virginia.

He has worked in the financial services industry as a Senior Planner performing business process improvement and implementing workflow solutions for over eight years. He was later employed as a Project Engineer for Northrop Grumman to model flight operations on Ford class aircraft carriers. Currently, he is with Lockheed Martin in Mooretown, New Jersey working on Aegis Ballistic Missile Defense modeling, simulation, and analysis. Research interests are training, simulation, and serious gaming. Other interests include brain-computer interface devices and research into brain-in-the-loop applications.