



International Council on Systems Engineering Colorado Front Range Chapter

System Engineering and the “Two Cultures” of Engineering

Michael D. Griffin, Administrator

National Aeronautics and Space Administration

Boeing Lecture, given at Purdue University on March 28, 2007

Most of you will have heard of Baron Charles Percy (C. P.) Snow, and will know of his observations on the breakdown in communication between the humanities and the sciences. Trained as a scientist, Snow served as Minister for Technology under Prime Minister Harold Wilson, yet was more famous as an author, with sixteen novels and eight works of non-fiction to his credit. He would be near the top of nearly any list of scientifically literate authors, or of literarily-talented scientists. Snow developed his theme in *The Two Cultures and the Scientific Revolution*, in 1959, and explored it further in *The Two Cultures and a Second Look*, in 1963. He decried the decline in standards of higher education, and in particular what he viewed as the almost willful ignorance by the modern cultural elite of scientific fundamentals. In a summary of his theme, Snow noted,

(Continued on page 2 - Griffin, “Two Cultures”)

Inside This Issue

Cover Story

System Engineering Speech 1
by NASA Administrator
Michael Griffin

President’s Corner - Jim 1
Russell

From The Editor 6

New Air Force Academy 7
Student Chapter

Joint Meeting with ASQ 9

2007 Chapter Activities 11
cast

“State Analysis” Web-cast 12

President’s Corner

On behalf of your Colorado Front Range Executive Committee (EXCOM), I’d like to welcome you to the first edition of your newsletter. It has been some time since we had this vehicle to communicate news and information to you, and it has been sorely missed. I’d like to thank Dr. Kip Tarpley for assuming the role of newsletter editor.

As you may remember, the Colorado Front Range Chapter received a Silver Award last June from INCOSE

(Continued on page 2 - President’s Corner)

(Continued from page 1 - President's Corner)

Central. This award recognizes our chapter for performing "over and above" the expected level of all INCOSE chapters. I am proud to say that we have worked hard over the past year to improve our Chapter, and we expect to receive a Gold Award at the next International Symposium (IS08) in Utrecht, the Netherlands in June. Much of this success is due to your suggestions for improvement that you provided the EXCOM both at meetings and through our annual survey. This has resulted in our being able to provide many opportunities to you; and has, in turn, enabled us to be recognized as one of the best chapters in INCOSE. In a future President's Corner, we will explain the Award purpose and criteria in more detail. For now, I'd like to thank the entire member-

ship of the Colorado Front Range Chapter, and especially those members who worked so hard to give us the opportunities we need to improve in our profession.

Finally, I'd like to urge anyone who wants to be more involved in your Chapter to volunteer. The EXCOM elections are fast approaching, and we need people to volunteer to serve. We also need volunteers to host meetings, act as (or provide) speakers, and also to provide tutorials.

Once again, thank you all for all your involvement, efforts, and support!

Jim Russell, President

Colorado Front Range Chapter of INCOSE ■

(Continued from page 1 - Griffin, "Two Cultures")

"A good many times I have been present at gatherings of people who, by the standards of the traditional culture, are thought highly educated and who have with considerable gusto been expressing their incredulity at the illiteracy of scientists. Once or twice I have been provoked and have asked the company how many of them could describe the, Second Law of Thermodynamics, the law of entropy. The response was cold: it was also negative. Yet I was asking something which is about the scientific equivalent of: 'Have you read a work of Shakespeare's?' I now believe that if I had asked an even simpler question — such as, What do you mean by mass, or acceleration, which is the scientific equivalent of saying, 'Can you read?' — not more than one in ten of the highly educated would have felt that I was speaking the same language. So the great edifice of modern physics goes up, and the majority of the cleverest people in the western world have about as much insight into it as their Neolithic ancestors would have had."

While Snow's criticisms did not go unanswered — most famously by literary critic F.R. Leavis — the essential truth of his observations was, and is, widely acknowledged. His elucidation of the "two cultures" has become a societal paradigm, a bumper-sticker phrase to describe the basic cultural separation between the arts and the sciences that is clearly visible to most of us. Even those who know nothing

else of Snow's work are probably familiar with this one phrase.

Today, I want to discuss the two cultures that, if we think about it, we find embedded in the profession we call 'engineering', and how we are linking them, and must link them, through the discipline known as 'system engineering', a product of the American aerospace sector.

Let us first explore the nature of the "two cultures" in engineering. I have always loved the view of the engineering profession captured by the great Theodore von Karman when he said, "Scientists study the world as it is; engineers create the world that has never been." Less eloquently, engineers are designers; they synthesize knowledge to produce new artifacts. Von Karman speaks to what most of us, and certainly most laymen, would consider the essence of engineering: engineers create things to solve problems.

But all of us who are engineers know that the engineering profession also has a rich scientific side, the analysis of these artifacts and the prediction of their behavior under various environmental and operational conditions. Adapting von Karman's observations, it may be said that *engineering* science is the study of that part of the world which has been created by man.

Sadly, many students have been led to believe that engineering science *is* engineering! In a curriculum of 120 or more credits leading to a bache-

(Continued on page 3 - Griffin, "Two Cultures")

(Continued from page 2 - Griffin, "Two Cultures")

lor's degree in a branch of engineering, the typical student is required to take one, or maybe two, courses in design. Everything else, aside from general-education requirements, focuses on the analysis, rather than the creation, of engineered objects. Graduate education often has no design orientation at all. So, engineering as taught really deals with only a part of engineering as it is practiced.

This trait is so pronounced that engineers who have spent their careers – even widely-recognized careers – in design and development, focusing on the creation of objects rather than the creation of papers for publication in refereed journals, are essentially unemployable, hence unemployed, in academia. No matter how well credentialed a practicing engineer may be, when the inevitable search committee meets to rank the applicants for a department chair, or a tenured position, it is a rare designer who can offer even the minimum of "academic" qualifications expected of an applicant for the position of assistant professor.

Some universities have recognized this inherent bias and its consequences for the training of their students, and have sought to remedy it by creating titles such as "Professor of Practice", or similar appellations. But it is a truism that the longer the title, the less important the job. So this term serves only to emphasize the point that these particular faculty members are not "real" professors, hired and promoted on their merits in a straight-up competition among all candidates. One wonders if this is the message we really want to send to those who will design – or not – the world of the next generation.

But if the present excessive focus on engineering science in the engineering curriculum is of concern, it is nonetheless true that the fundamental difference between modern engineering and that practiced prior to the Enlightenment is the development of formal analytical methods and their application to man-made objects. This has allowed the prediction of performance, and the limits of that performance, in the environment in which a given device must function. It has allowed the refinement of designs through methods more sophisticated than the trial-and-error techniques to which our ancestors were limited. It has enormously shortened the time required for a design cycle for the objects we create. A control system engineer might say that the formal methods of engineering science have produced an

enormously improved feedback path for the engineering design loop. More simply, engineering science has taken engineering beyond artisanship.

But, interestingly, the development of formal methods has not altered in any way the fundamental nature of design, which still depends, as it did in antiquity, upon the generation of a concept for a process, technique, or device by which a given problem might be solved. The engineering sciences have provided better, and certainly quicker, insight for the designer into the suitability of the concept than can be provided solely by building it and examining its performance in its intended application. But a human being must still intuit the concept. We have no idea how we do that. And until we do, we have little hope of developing a formal method by which it can be accomplished.

It must be said that some progress in this area has been through research into "genetic algorithms", which use the tools of engineering science and mathematical simulation to explore the consequences of iterative random changes to a given design. The performance of the design is evaluated against objective criteria. If a change results in a net improvement it is retained; otherwise, it is discarded. In this manner, the design "evolves" to a higher state of suitability to its intended "environment" through the pressures of artificial, rather than natural, "selection". Modern engineering analysis tools offer the ability to conduct what is essentially a very large number of randomized design cycles in an acceptable period of time.

But this process does not seem, at least to me, to be much akin to the intuitive synthesis of a human brain when it leaps almost instantly from a perception of a problem to an idea for its solution. "Creativity", used in this sense, remains thus far the sole province of biological computers.

However, my colleague, NASA Associate Administrator Lisa Porter, has pointed out to me that, precisely because genetic algorithms work differently and produce different results than would a human designer, they can offer new, unusual, and potentially useful solutions for consideration by humans. So as the field of genetic algorithms matures, it may well be that the methods of engineering science will yield solid contributions to the synthetic aspect of engineering.

But at least for now, there remains an artistic

(Continued on page 4 - Griffin, "Two Cultures")

(Continued from page 3 - Griffin, "Two Cultures")

side of engineering, and it is fully as much an art for its practitioners as any painting, sculpture, poem, song, dance, movie, play, culinary masterpiece, or literary work. The difference between the cultural and engineering arts lies not so much in the manner of creation of a given work, but in the standards by which that work is judged. In the humanistic disciplines, human aesthetics sets the standard by which merit is assigned to a finished product. In the end, aesthetic sensibilities vary with place and time, and are ultimately matters of opinion. The role of opinion in evaluating a work of engineering is, by comparison, much restricted. In engineering, more objective methods are employed to judge the degree to which the completed work meets the standards established for it, or fails to do so.

This brings us to the role of failure in engineering design. Regardless of the sophistication of the analytical methods brought to bear, they are applied to a theoretical model of a device operating in a theoretical model of the real world. The model is not reality, and the differences produce opportunities for the real device to fail to operate as intended in the real environment. An evolutionary biologist might say that the gap between model and reality is an environmental niche in which failure, like a new species, can thrive.

Civil engineer and author Henry Petroski has, in a series of essays and books, explicitly noted the crucial role of failure in producing ultimately successful designs. In *Success Through Failure: The Paradox of Design*, and other works, Petroski establishes the point that new designs, or successive iterations and refinements of a basic design, have as their essential purpose the elimination of failure modes known to be inherent in earlier designs. He further argues, by means of many examples, that designers must go beyond merely ensuring success; they must strive to anticipate the ways in which a design might fail. Great designers and successful designs incorporate, in advance, methods to mitigate such anticipated failures.

But in recent decades human artifacts have become increasingly complex, building upon and extending former art and, especially, combining disparate elements of established art in new ways. This has been accomplished at an astonishing pace, a cause and a result of Moore's Law, the approximate two-year doubling time of computational throughput,

which has held sway for several decades. While a large bridge cannot properly be considered a "simple" structure, involving as it does the interaction of thousands of component parts, it clearly pales in complexity relative to, say, a space shuttle, which relies for its success upon the interaction of millions of parts derived from a dozen technical disciplines.

Failure in complex systems can arise in so many more ways than in simpler systems that the quantitative difference ultimately produces qualitatively different behavior. It becomes unreasonable to expect, other than through the harshest of hindsight, that a particular failure mode might have been or ought to have been anticipated. Indeed, results from the modern study of complexity theory indicate that complex systems can experience highly non-linear departures from normal state-space trajectories – i.e., "failure" – without anything being "wrong".

Among the first to study complex engineering systems was Charles Perrow, in the landmark work *Normal Accidents*. Perrow argued that adding additional processes, safety measures, and alerts to complex systems – the traditional design approach to improving system safety – was inherently flawed, because for complex, tightly-coupled systems and organizations, failure is inevitable.

Perrow is a sociologist, not an engineer, but his points are well taken. Those of us who are aviators, or who are familiar with the history of aviation, can point to numerous high-profile accidents where the crew became occupied with minor anomalies and their warning systems, only to fly a perfectly good airplane into the ground. Most of us can also cite analogous incidents from other fields.

Yet, we have evolved complex systems for good reasons, and we will clearly continue to do so. The modern air transport aircraft is an incredibly complex device, and the system within which such aircraft operate is far more so. But in the last five decades this system has revolutionized world society, culture, and economics. It will not be shut down merely because it cannot be made perfectly reliable. Nor will we do so with any of the other complex appurtenances of modern society which did not exist a century ago, but which are now deemed essential. So, if we are not to eschew the use of complex systems, how do we make them as reliable as possible?

I believe that the answer to the above question is "system engineering". This is an entirely ap-

(Continued on page 5 - Griffin, "Two Cultures")

(Continued from page 4 - Griffin, "Two Cultures")

appropriate answer for the Boeing Lecture here at Purdue University, for system engineering has evolved as a discipline of modern engineering from its roots in the American aerospace system development culture.

System engineering and its allied discipline of systems management are treated from a historical perspective in the excellent text by Stephen Johnson, *The Secret of Apollo*. Johnson retraces Petroski's path, showing the development of system-oriented disciplines to be the natural reaction to the failure of early, complex aerospace systems, including large aircraft, ballistic missiles, and spacecraft.

From its first introduction into the engineering lexicon, "system engineering" has been a question-begging term. In earlier times, it was considered by many in the traditional engineering disciplines to be a category without a subject matter. Even today I find the term to be, in my opinion, misused and misunderstood by many who claim to be practitioners of the art. So, having spent what I believe to be the most productive part of my career as a system engineer, let me say a few words about what I believe system engineering is, and what it is not.

System engineering is the art and science of developing an operable system capable of meeting requirements within imposed constraints. The definition is somewhat independent of scale, and so these words are useful only if one understands that it is the big-picture view which is taken here. We are talking here about developing an airplane, a spacecraft, a power plant, a computer network. We are not talking about designing a beam to carry a particular load across a known span.

System engineering is a holistic, integrative discipline, wherein the contributions of structural engineers, electrical engineers, mechanism designers, power engineers, and many, many more disciplines are weighted and considered and balanced, one against another, to produce a coherent whole that is not dominated by the view from the perspective of a single discipline. System engineering is about trade-offs and compromises, about generalists rather than specialists.

System engineering is not about the details of requirements and interfaces between and among subsystems. Such details are important, of course, in the same way that accurate accounting is important to the Chief Financial Officer of an organization. But

accurate accounting will not distinguish between a good financial plan and a bad one, nor help to make a bad one better. Accurate control of interfaces and requirements is necessary to good system engineering, but no amount of care in such matters can make a poor design concept better. System engineering is about getting the right design.

Complex systems usually come to grief, when they do, not because they fail to accomplish their nominal purpose. While exceptions certainly exist, it remains true that almost all systems which proceed past the preliminary design phase will, in fact, accomplish the tasks for which they were explicitly designed. Complex systems typically fail because of the *unintended* consequences of their design, the things they do that were not intended to be done. The Second Law of Thermodynamics is sufficient to guarantee that most of these things will be harmful! I like to think of system engineering as being fundamentally concerned with minimizing, in a complex artifact, unintended interactions between elements desired to be separate. Essentially, this addresses Perrow's concerns about tightly coupled systems. System engineering seeks to assure that elements of a complex artifact are coupled only as intended.

C.P. Snow believed that mutual comprehension and appreciation between the arts and the sciences, which had existed in earlier times, had been erased by his time. He did not find a means to restore it. I sometimes think that the gap between synthesis and analysis in engineering is as wide as that between the arts and the sciences of Snow's "two cultures". But the fact remains that designers simply do not think or work in the same way as analysts, and this does on occasion produce a certain cognitive dissonance. When it occurs in the context of a complex system development, catastrophe is a likely result.

System engineering is the link which has evolved between the art and science of engineering. The system engineer designs little or nothing of the finished product; rather, he seeks a balanced design in the face of opposing interests and interlocking constraints. The system engineer is not an analyst; rather, he focuses analytical resources upon those assessments deemed to be particularly important, from among the universe of possible analyses which might be performed, but whose completion would not necessarily best inform the final design. There is an art to knowing where to probe and what to pass by,

(Continued on page 6 - Griffin, "Two Cultures")

(Continued from page 5 - Griffin, "Two Cultures")

and every system engineer knows it.

Like other branches of engineering, system engineering has evolved out of the need to obviate dramatic failures in complex systems. Such failures are not new. One of my favorite books is a fascinating text entitled "*Structures: or, Why Things Don't Fall Down*", by Prof. J.E. Gordon of the University of Reading, England, written in 1978, at the end of Prof. Gordon's long career as a structural analyst. It is aimed at a level appropriate to an intelligent technical professional in any field. I recommend it highly. Regarding the matter of spectacular engineering failures, I quote Professor Gordon (pps. 352-353):

"... there are, of course, a certain number of great dramatic accidents which, for a while, monopolize the headlines. Of such a kind were ... [numerous disasters follow] ... These are very often intensely human and intensely political affairs, caused basically by ambition and pride. ... One can at once recognize a certain inevitability about the whole procedure. Under the pressure of pride and jealousy and ambition and political rivalry, attention is concentrated on the day-to-day details. The broad judgments, *the generalship of engineering*, [my emphasis] end by being impossible. The whole thing becomes unstoppable and slides to disaster before one's eyes. ..."

In thirty-six years of engineering practice, of many kinds and in many situations, I have not seen a more appropriate assessment of what is truly important in engineering. We must of course get the details right. However, to be a complete engineer, one must also master what Professor Gordon calls "the generalship of engineering".

I will be frank. Educators, and I include myself, for I have spent many years as an adjunct professor at various institutions, are far less certain how to teach "generalship" than we are of how to teach the laws of thermodynamics. And yet it is clear that an understanding of the broad issues, the big picture, is so much more influential in determining the ultimate success or failure of an enterprise than is the mastery of any given technical detail. The understanding of the organizational and technical interactions in our systems, emphatically including the human beings who are a part of them, is the present-

day frontier of both engineering education and practice. ■

From the Editor

I am honored to be chosen as your newsletter editor and I look forward to serving the Executive Committee and the Membership in this role. My name is Kip Tarpley. I am an Aerospace Engineer, working from my home for Astrox Corporation headquartered in College Park, Maryland. We design and analyze next generation launch vehicles (rocket and air-breathing) for the Air Force, NASA, and Navy. I have worked for Astrox since 1995, traveling 2-3 times a year to industry conferences or meetings at client sites. I worked for Morrison-Knudsen Company for 12 years in Idaho, Saudi Arabia, and Colombia in various IT capacities. In 1985 I returned to school at the University of Maryland in College Park, MD and, in 1995, received a PhD in Aerospace Engineering.

My interest in Systems Engineering stems from a speech (see our cover story) given by Dr. Griffin at Purdue in March of 2007. During grad school I was unknowingly doing a system engineer's job when I optimized the hypersonic vehicle design that was the topic of my dissertation. Balancing the best parts of each discipline included in the vehicle analysis against the requirements and constraints of the mission is what Dr. Griffin talks about in the first part of his speech. He has captured the essence of what I was doing in his excellent introduction to the art and science of system engineering.

I invite all of you to submit articles, suggestions for articles, comments/letters, or any other material (including photos of chapter events) for the newsletter to [me](#) for consideration. ■

April Meeting

Systems Engineering at the US Air Force Academy

Our April Meeting was hosted by the newly formed Student Division of the INCOSE Front Range Chapter at the Air Force Academy. Captain Cooper (Faculty Advisor to the Student Division) briefed on the Systems Engineering program at the Academy and Cadet 1st Class Mark Wojtowicz (President of the Student Division) briefed on the FalconSAT-5 program. Their briefings gave our group a much better appreciation for the outstanding systems engineering programs implemented to train our future Air Force engineers and leaders. Norman Jarvis is the INCOSE Front Range Chapter Representative Liaison to the USAFA Student Division. Below is described the Systems Engineering program at the Air Force Academy.



Cadet 1st Class Wojtowicz briefs attendees on the FalconSAT-5 Program.

Systems Engineering at the Air Force Academy by Captain Cooper

The systems engineering program started at the academy in late 2003-early 2004. The first graduating class of its majors was in June 2006. The program is comprised of the Systems Engineering (SE) major (in 9 different engineering concentrations) and the Systems Engineering Management (SEM) major. On average, we have 40-50 SEs and 70-90 SEMs in each class year. Currently, we have 396 cadets declared (145 SEs and 251 SEMs).

The program is administered by a working group of faculty from the 7 primary supporting departments (Departments of Aeronautical Engineering, Astronautical Engineering, Behavioral Sciences and Leadership, Computer Science, Electrical and Computer Engineering, Engineering Mechanics and the Department of Management). Overhead functions of the SE program (budget, personnel, vertical reporting, a few dedicated billets, etc) are administered under the Department of Aeronautical Engineering. For the SEM program, these overhead functions are handled by the Department of Management.

The nature of our highly-matrixed program is a little

(Continued on page 8)



Captain Cooper (left) and Cadet 1st Class Wojtowicz (right) accept award from Norman Jarvis.

(Continued from page 7)

“s” and large “E.” We built the program largely on existing engineering, computer science, operations research, and human systems courses. Every cadet takes ~5-6 “core SE” courses dovetailed with ~7-8 courses in the concentration that they choose (see curriculum summary below). This construct was built partly on the constraints of the program formulation (not able to add many new courses). We’ve found, though, that building SEs at the undergraduate level is more effective when a strong foundation of a specific domain accompanies the SE concepts. This allows the cadets the chance to apply their systems-thinking to an application area and eventually a capstone design project in that domain. While we are cognizant of the ongoing discussion of how best to develop an SE (undergrad vs. experience + training) we feel that we have reached a working solution to the AF’s urgent need for educated Systems Engineers. We are currently preparing for ABET accreditation this coming fall. It will be the first time that ABET has had a chance to visit our new program and we are optimistic that we will be accredited based on the strong support we have from the existing accredited departments.

Roughly 50% of our graduates go into rated (flying) career fields in the AF. The remaining cadets go to a variety of support career fields, primarily developmental engineering and acquisitions for our SE/SEMs. On inception of the program, AF Secretary Roche wanted to increase the systems-thinking of all graduates of USAFA and AFIT and noted that he would be happy with ~10 straight SE majors a year. The academy has taken steps with core courses to enable systems thinking for all graduates. In SE, we’ve far exceeded his expectations of direct SE/SEM majors. The SE program (in its combined look of SE/SEMs) has grown to be the most populated major program at the Academy (~9% of the cadet wing).

Our Student Division of INCOSE has just started. This year we have identified student officers in the first-class cadet (senior) class to set a good foundation (developing charter documents and laying a plan for the future of the division). There are 6 officers; Cadet First Class Mark Wojtowicz is the current president. Hopefully in the near future, we will be recruiting underclassmen to develop the division and continue a strong relationship with the Front Range

Chapter.

From our curriculum handbook summary of the SE major:

The Systems Engineering Major at a Glance:

Systems Engineering is an interdisciplinary major administered by the Department of Aeronautical Engineering and supported by the Departments of Aeronautical Engineering, Astronautical Engineering, Behavioral Sciences and Leadership, Computer Science, Electrical and Computer Engineering, and Engineering Mechanics with participation by the Department of Management. Systems Engineering is a broad discipline that addresses the engineering of large, complex systems and the integration of the many subsystems that comprise the larger system. All of these various components must function together in an effective and efficient manner in order to carry out the mission. The systems engineer designs, integrates, and helps to ensure smooth functioning of complex systems typical in today’s high-tech Air Force. The systems engineer is a “big picture” engineer, architecting the engineering process for the design of the overall system to ensure that it will meet the needs of all the system’s stakeholders including operators, maintainers and commanders, and even our ultimate customer the American public!

The systems engineer must consider elements of system development, verification, manufacturing, deployment, training, operations, support, and disposal. The entire life cycle of the system is considered in a holistic fashion early in the system’s development cycle. To accomplish this difficult job, the SE must have broad interdisciplinary knowledge across many areas of study. USAFA’s system engineering program emphasizes a systems-of-systems approach that integrates a rigorous engineering curriculum augmented with studies in human systems, operations research analysis, program management, and the core curriculum. Cadets will learn that the systems engineering process is an interdisciplinary engineering process that evolves, verifies, and documents an integrated, life-cycle-balanced set of system solutions that satisfy customer needs. Cadets

(Continued on page 9)

(Continued from page 8)

will specialize in one of nine defined option areas which include: Aeronautical Systems, Communication Systems, Computer Systems, Control Systems,

Electronic Systems Design, Human Systems, Information Systems, Mechanical Systems, and Space Systems.

Cadets who successfully complete the Systems Engineering major are awarded a Bachelor of Science in Systems Engineering degree.

Systems Engineering Program Operational Goals:

Two to three years after graduation, our graduates are expected to be officers who:

- Possess breadth of integrated, fundamental knowledge in the basic sciences, engineering, humanities, and social sciences; and depth of knowledge in the selected option sequence,
- Can communicate effectively,
- Can work effectively on teams and grow into team leaders,
- Are independent thinkers and learners,
- Can apply their knowledge and skills to solve Air Force engineering problems, both well- and ill-defined, and
- Know and practice their ethical and professional responsibilities as embodied in the United States Air Force core values.

For more information:

Cory Cooper, Capt, USAF
 Chief of Ops, Systems Engineering
 Advisor in Charge, Systems Engineering
 Department of Aeronautics, US Air Force Academy
 (719) 333-3545, DSN 333-3545

Email Address: cory.cooper@usafa.edu ■

January Meeting

Joint Meeting with ASQ

Our January 2008 meeting was a joint meeting with the American Society for Quality. The meeting featured three breakout sessions as described below. Jerry Huller, the INCOSE Colorado Front Range Chapter Treasurer presented one of the sessions on Requirements Decomposition and Completeness. Jerry is seen below at the start of his excellent presentation.

TOPIC: Joint Meeting with American Society for Quality (ASQ) featuring three breakout session topics.

1. Killer Team Building Exercises (Esther Evans)

Teambuilding Activities provide the opportunity for adults to challenge their ability to problem solve as a team. They provide a common experience to bring work-centers together on a different operational level. The right activity can get amazing results for work

(Continued on page 10 - Joint Meeting)



Jerry Huller briefs the Joint ASQ Meeting on Requirements Completeness

(Continued from page 9)

center cohesiveness.

This presentation will encompass proven teambuilding activities for the attendees.

2. Am I the only one who wonders what I am... (Mike Boyd)

Nobody will stop you in the hallway at work to ask if your career provides meaning and personal fulfillment. Recognizing that something's missing in your career and taking the initiative to make a career change, or reinvigorate the career you have now, must come from within. Let's talk about ways you can take charge of re-igniting your career search, or career, and make the most of it.

3. Requirements Decomposition and Completeness (Jerry Huller)

Learn about how to make your requirements set more complete and issues that may occur when you don't. Learn about pitfalls into decomposing requirements to lower levels. Defect prevention begins with writing good requirements.

SPEAKERS:

ESTHER EVANS

Esther Evans is the Deputy Director of the Mission Support Squadron, Peterson AFB, CO, and is an Adjunct faculty member at Pikes Peak Community College. Her prior experience includes: Senior Consultant for Human Resources, Edwards AFB, CA 2002-2004; the Performance Management and Manpower Director at 21st Space Wing, CO, 1999 – 2000; the Evaluator for the Presidents Quality Award 1997-1999; an Air Force Master Facilitator; and a BBB Excellence in Customer Service evaluator/trainer. Her education includes: B.A Auburn University; M.Ed. The Citadel; Master of Military Arts and Science, Air University 2005.

MIKE BOYD

Mike Boyd has extensive experience in business development, general management and human resources which he gained as an executive in International Fortune 500 companies, hi tech startups, and as the President/CEO of a successful start-up corpo-

ration.

He is a member of the American Society for Training and Development and the Society for Human Resource Management. He has been appointed to oversight positions by mayors and Governors and sits on the boards of several non-profit organizations

Mike is certified by the HR Certification Institute as a Senior Professional in Human Resources (SPHR) and is currently a candidate for the status of Fellow as granted by the Association of Career Consulting Firms International. His undergraduate degree is in Philosophy and he has an MBA with an emphasis in Organizational Management from the University of St. Thomas in St. Paul Minnesota.

JERRY HULLER

Jerry Huller is currently the Raytheon Requirements Lead on the Geostationary Operational Environmental Satellite System - R Series (GOES-R) study project. Prior to that, he was the Raytheon System Test Lead and Requirements Management lead on the National Polar-orbiting Operational Environmental Satellite System (NPOESS) program. Jerry has also served as an Integration and Test Lead on international projects. Jerry has done systems engineering and test for the U.S. Army, E-Systems, TRW, and SkyTel.

Jerry is a Certified Systems Engineering Professional (CSEP) and an active member and officer of the Colorado chapter of the International Council on Systems Engineering (INCOSE). Within INCOSE, he is a member of the Verification and Validation Working Group and the Requirements Working Group. ■

Chapter Activities in 2007

2007 was a very active year for the Colorado Front Range Chapter of INCOSE. Our membership at the end of 2007 stood at 165. Monthly meetings were held throughout the year except for the summer and December. Monthly Executive Committee meetings were held as well except for the summer months. Outreach to the community and other organizations was done by chapter members. Our activities should lead to a Gold Award for the chapter at the Annual INCOSE meeting, Utrecht, Netherlands, June 2008.

Highlights of the year include:

- Membership Survey completed in December.
- Jerry Huller and Leslie Koshigoe gave an SE presentation at a Career Fair at Cherry Creek High School, Englewood, CO, on Tuesday, 27 November.
- Spring Tutorial on VV&T and Fall Tutorial on *Complex Systems for the Systems Engineer* given by Sarah Sheard
- Student Chapter at AFA started.
- Jeff Hayden presented the paper "Architecting the Communication and Navigation Networks for NASA's Space Exploration Systems " at the 2007 IEEE International Conference on System of Systems Engineering (SoSE) on April 16-18, 2007 in San Antonio, TX.
- Natalie Salazar reviewed technical papers for the 2007 INCOSE International Symposium (IS).
- Kip Tarpley served as judge at a High School Robotics contest at the University of Colorado at Colorado Springs on 10 Nov. The contest was part of the "Power Puzzle 2007 First Lego League (FLL) Challenge".
- Trace Baker and Jerry Huller served on the Sponsorship Committee for the 2007 INCOSE International Symposium (IS).
- Jerry Huller performed as Certification Application Reviewer (CAR) for numerous CSEP applications.

- Jerry Huller helped with review and editing of the appendices for the INCOSE Systems Engineering Handbook version 3.1.
- John Parr and Scott Frisch serve as members of the SE Advisory Board for the Colorado Technical University in Colorado Springs.
- Mark Mihelic served as judge/consultant at the Fall 2007 Design Expo at the University of Colorado at Boulder on Saturday, December 8, 2007. ■



A student is briefed on System Engineering by Leslie Koshigoe.



Jerry Huller describes the role of INCOSE to a prospective Systems Engineer.

March Meeting

“State Analysis” Webcast

Our March Meeting was a joint meeting with the LA area INCOSE chapter. We attended via webcast from Raytheon in Boulder. The webcast and facility were setup and hosted by Trace Baker as seen in the photo below. The abstract and speaker bios follow. The spring tutorial is tentatively scheduled to be a full day on the topic of State Analysis.

Abstract:

Spacecraft system complexity is reaching a threshold where customary methods of control are not affordable or sufficiently reliable. The conventional approaches to systems and software engineering, based on functional decomposition, fail to scale in the tangled web of interactions encountered in complex spacecraft designs. Furthermore, there is a gap between the requirements on software specified by systems engineers and the implementation of these requirements by software engineers. Software engineers must perform the translation of requirements into software, hoping to accurately capture the systems engineer’s understanding of the system behavior. This gap opens the possibility for misinterpretation of the systems engineer’s intent, leading to software errors. This problem is addressed by a systems engineering methodology called State Analysis, which provides a process for capturing requirements on system and software design in the form of explicit models of system behavior, and defines a state-based architecture for the control system. It provides a common language for systems and software engineers to communicate, and thus bridges the traditional gap between software requirements and software implementation.

Speaker Bios:

Mitch Ingham is a software systems engineer and software architect at the Jet Propulsion Laboratory. He received his Sc.D. and S.M. degrees from MIT’s Department of Aeronautics and Astronautics, and his B.Eng. in Honours Mechanical Engineering from

McGill University. Mitch has played a central role in the development and formalization of the model based systems engineering methodology called State Analysis, and its application to the design and implementation of state-based software architectures. As lead software systems engineer for the Mars Science Laboratory (MSL) Entry, Descent, and Landing (EDL) Prototype, he coordinated activity across the MSL Project to perform the end-to-end systems engineering, software development, and validation for an integrated software demonstration of EDL in 2004. He is currently working on the Constellation Lunar Lander Vehicle (Altair) project as a fault protection engineer for the guidance, navigation, and control subsystem.

Bob Rasmussen is Chief Engineer for the Systems and Software Division of the Jet Propulsion Laboratory. He has been a systems engineer at JPL since 1975. He received his Ph.D. in Electrical Engineering from Iowa State University. He supported the development of Voyager and has subsequently supported many flight projects since then, in both line and project leadership roles. His experience covers the areas of spacecraft guidance and control, avionics, test and flight operations, and automation and autonomy - particularly for fault protection.

See this [link](#) for more information. ■



Trace Baker introduces the State Analysis Webcast held jointly with the INCOSE LA Chapter.

Colorado Front Range Chapter Officers

President	Jim Russell
Past President	
Vice President	Leslie Koshigoe
Treasurer	Jerry Huller
Secretary	
Membership Director	Scott Frisch
Technical Program Director	John Parr
Denver Area Director	Rob Vincent
Colorado Springs Area Director	Norm Jarvis
Boulder Area Director	
Communications Director	Kirk Moen
Newsletter Editor	Kip Tarpley
Webmaster	Tom Schei/Kris Howard

Up Coming Activities

Monthly Meetings

May - tentative for May 27, 2008

Topic: DoDAF / Service Oriented Architecture / Model Based Architecture-SE at Booz Allen Hamilton in Colorado Springs.

June thru August - No Meetings

September - TBD

Tutorials

The Spring 2008 Tutorial is tentatively set for the first week of June. It will be a full day workshop on State Analysis methodology that was the topic of our March meeting at Raytheon. Information will be emailed to you as details become available.

Elections

Elections for the 2008-2009 Program Year will be held in May 2008. If you are interested in serving, please email [Jim Russell](#) for more information.

Call For Officers

There are open positions on the Chapter Executive Committee for the Secretary and the Boulder Area Director. If you are interested in filling either of these positions, please contact Chapter President [Jim Russell](#).

Call for Articles

The next newsletter, the Fall 2008 issue, will be published in October. If you have any material for that issue, please email to [me](#) by the end of September 2008.

For More Information

Email: [Jim Russell](#)

Snail Mail: INCOSE Colorado Front Range Chapter
P.O. Box 631201
Highlands Ranch, CO 80163-1201

Links:

International Council on Systems Engineering - www.incose.org

Colorado Front Range Chapter - www.incoseco.org

Change of Address

If your email address changes, please send the change to our Communications Director, [Kirk Moen](#).