2006-2085: A CASE-BASED APPROACH TO SYSTEMS ARCHITECTURE AND ENGINEERING EDUCATION

Jonathan Weaver, University of Detroit Mercy

JONATHAN M. WEAVER, PH.D. is an Associate professor of Mechanical Engineering at the University of Detroit Mercy (UDM). He received his BSME from Virginia Tech in 1986, his MSME and PhD in ME from RPI in 1990 and 1993, respectively. He has several years of industry experience and regularly consults with an automaker on projects related to CAD, DOE, and product development. He can be reached at weaverjm@udmercy.edu.

Michael Vinarcik, University of Detroit Mercy

MICHAEL J. VINARCIK, P.E. is an Interior Trim Engineer with Ford Motor Company and an adjunct faculty member at the University of Detroit Mercy. He received a B.S.(Metallurgical Engineering) from The Ohio State University in 1990, an MBA from the University of Michigan in 1997, and an MS Product Development from the University of Detroit Mercy in 2004. He has fifteen years of automotive experience and is active in numerous technical and professional societies.

A Case-Based Approach to Systems Architecture and Engineering Education

Abstract

Good systems architecture and systems engineering processes are key enablers for the development of innovative, robust engineering systems. Many product failures can be traced directly to breakdowns in the architectural or systems engineering practices of the design team.

Despite the increased emphasis on systems engineering, most systems engineering textbooks tend to focus on specific tools (such as requirements or interface management systems) or describe the systems engineering and systems architecting process in a rather generic discussion. Case studies are typically brief and relatively sparse.

A typical teaching approach is to introduce a tool, illustrate how the tool can be applied, introduce another tool, etc. However, cultivating expertise in specific tools that may not be in use by a student's employer adds little value – particularly if the student misses the holistic understanding of the topic because he is focusing on details of the tool. The authors believe that it is more useful to focus on teaching students to intuitively understand architectural and systems engineering issues. For that reason, they have adopted a case-based approach to teaching these topics.

Using topics drawn from history (ancient tombs and medieval cathedrals) and current events (the Airbus A380/Boeing 787 and the Ansari X Prize Competition), the authors present a broad spectrum of cases to their students. This engages the students, sparks classroom discussion, and enhances learning and retention of key topics.

The cases are presented using a variety of media (including PowerPoint slides, audio-visual presentations, or show-and-tell artifacts). The cases are typically used as lead-ins to the lecture, allowing the instructor to draw upon the outcomes (both positive and negative) of the case to illustrate key learning principles in the main lecture. Relevant and useful tools are still taught (such as QFD, Design Structure Matrices, functional decomposition, etc.) but the case studies provide interesting, motivational examples illustrating the need for such tools and the authors find it useful to ask the students to discuss how the tools of today might be (or have been) utilized in the design of the subjects of the case studies.

Case studies are also assigned as homework, allowing the students to research a topic and draw their own conclusions from their research and the course material. These assignments are sufficiently structured to foster students' development but allow them some latitude to explore the topic. The purpose is to develop their analytical skills and encourage holistic viewpoints rather than requiring simple rote learning.

This paper will summarize several of the specific case studies which the authors use and discuss how each one is tied to specific topics and learning objectives of the courses. This case-based approach has been applied to separate, semester long courses in Systems Architecture and

Systems Engineering as well as a condensed version of those two courses (a single semester course entitled Systems Architecture and Systems Engineering).

Introduction

The authors have been involved with the teaching of the Systems Architecture and Systems Engineering courses which are required content in the Master's of Product Development (MPD) Program at The University of Detroit Mercy (UDM). They have also developed a condensed version of the courses, a single Systems Architecting and Systems Engineering elective, taught as part of UDM's Master's in Engineering Management program. In each course, they review how various authors and organizations define systems architecting and systems engineering, ultimately deducing that the following definitions succinctly and effectively define these terms in the context of our courses:

Systems Architecting: The mapping of function to form via concept.[1]

Systems Engineering: An interdisciplinary approach and means to enable the realization of successful systems.[2]

The architecting activity is critically important; that is the time to fully understand the problem and to search for creative alternative architectures that optimally map the system's required functions to form (hardware and software). Well-executed architectural explorations will result in the most competitive and successful systems because no amount of systems engineering or detailed design can overcome a fundamentally flawed architecture. It is most often poor judgment during the architecting phase of system development, not an erroneous calculation in the detailed design phase, that leads to failed systems. However, in many cases involving the development of highly complex systems, poor systems engineering practices can result in difficulties or failures executing sound architectures.

Systems architecting can be a difficult topic to teach since it typically involves an eclectic blend of art, science, judgment, and the application of heuristics; it also requires a holistic understanding of technologies, politics, and society. Despite recent increased emphasis on systems engineering, most systems engineering textbooks focus either on exclusively software topics or on specific tools (such as Quality Function Deployment (QFD), Design Structure Matrices (DSM), or tools for managing requirements (DOORS, et al.). Case studies are typically brief, relatively sparse, and often not well integrated into the pedagogical flow of material.

While cultivating expertise in a specific tool that is in use by a student's organization may be of direct immediate value to the student and his/her employer, doing so can be difficult in the systems architecture and systems engineering domain due to the lack of commonality and relatively limited deployment of such tools. The authors feel that in-depth instruction on a tool that the student is not likely to use in the field adds little value; they believe that it is more useful to focus on teaching students to intuitively understand architectural and systems engineering issues. For that reason, they have adopted a case-based approach to introducing these topics to the students. Nonetheless, a few tools are in sufficiently widespread use that the authors do choose to specifically cover them; these include QFD, DSM, and functional decomposition). On other topics – requirements management and traceability for example – we choose not to cover any specific tool, but instead focus on understanding what needs to be done with regard to that aspect of systems engineering.

The next section of this paper summarizes cases that have been developed and presented. A complete discussion of all of the cases would be too lengthy for this forum, so only a very brief description is given for each. In addition, the authors welcome inquiries from interested readers who wish to adopt or comment on the cases. The authors welcome any suggestions for additional cases or suggested enhancements to the cases already in use.

Lead-In Cases

Using topics drawn from history (early battleships, ancient tombs, medieval cathedrals, early automobiles, etc.) and current events (Airbus A380/Boeing 787, Ansari X Prize Competition, recent NASA missions to Mars, etc.), the authors present a broad spectrum of cases to their students. This engages the students, sparks classroom discussion, and enhances learning and retention of key topics. The majority of these cases are used in what the authors refer to as "lead-in" cases (simply referred to as cases hereafter) since a class session is typically begun with a case which illustrates some of the major points of the lecture. This allows the instructor to draw upon the outcomes (both positive and negative) of the case to illustrate key learning principles. When specific tools are discussed as part of the main lecture, a class discussion about how that tool could be (or could have been) applied to the subject of the case. The cases are presented using a variety of media (including PowerPoint slides, audio-visual presentations, or show-and-tell artifacts). The authors typically choose not to reveal the subject of the case at the outset, but rather to show a sequence of increasingly revealing slides until one of the students can deduce the subject of the case (for some of the more obscure cases, that point does not come and the subject is eventually revealed, but for most of the cases, the students eventually narrow in on it).

Case studies are also assigned as homework, allowing the students to research a topic and draw their own conclusions in light of the course material. These assignments are sufficiently structured to foster a student's development while allowing him/her some latitude to explore a self-selected topic. The purpose is to develop the students' analytical skills and encourage holistic viewpoints rather than requiring simple rote learning. Allowing students to choose their own topics for cases also provides an opportunity for students to apply the course content in domains of particular interest.

The instructors have prepared more than ten cases for systems architecture and more than ten cases for systems engineering. To provide an overview of the cases, five specific cases the authors have used in each course are summarized below. If the authors gage that there is interest in the remaining cases, a future follow-on paper will summarize them.

Systems Architecture Cases

Case 1. The *Vasa*

During the 1620's, Sweden was at war with Poland. In 1625, the Swedish King Gustavus Adolphus ordered new warships, among them the *Vasa*. The King had been shown a copperplate of the *Saint Louis* (a Dutch-built French warship of 1,000 tons displacement built in 1626) and wanted a similar ship to be built—only he wanted the ship to be larger and to have an additional gun deck. The *Vasa* was to be mightiest warship ever with sixty-four guns on two gun decks. It was built in Stockholm by Henrik Hybertsson and, after Henrik's death, his brother. On August 10, 1628, the *Vasa* made her maiden voyage. She set sail and fired a salute, then sank within one nautical mile of shore, killing thirty to fifty of the 150 aboard. The bronze cannons and other items of value were quickly salvaged, after which the *Vasa* was forgotten until it was rediscovered in 1956.

The class discussion around the *Vasa* involves several aspects relating to systems architecting. First, the topic of form versus function is explored as we discuss some of the elaborate wood carvings on the *Vasa*'s superstructure. In addition, we discuss how the apparent naval architecting technique, which was apparently based on reckonings intended for smaller ships, was insufficient when applied to a vessel the size of the *Vasa*. From a system certification point of view, it is also apparent that a stability test conducted on the *Vasa* was cut short when it was clear she would have capsized, yet the development continued without corrective action. Most ironically, it appears that the celebratory simultaneous firing of all the guns on one side of the *Vasa* which led to the sinking may have been a more extreme situation than would have been likely to occur in a real battle.

While the students' background rarely includes any naval architecting, the students are asked to discuss how they think modern tools are applied to avoid similar architectural flaws in modern ship design.

Figure 1. The *Vasa*[3]

Case 2. Theodoric's Tomb

Theodoric the Great (493-526 AD) was King of the Ostrogoths (474-526 AD), King of Italy (493-526 AD), and King of the Goths (511-526 AD). Theodoric was presumably the architect of his own tomb, shown in Figure 2. As an introduction to the process of mapping function to form, the authors ask the students to list the functional requirements of the tomb. After some class discussion, including some input from the instructor, this typically results in functional requirements which include: provide burial place, withstand eternity, prevent orthodox Italians from destroying it (his Arian beliefs placed him in the minority) in the chaos expected after his death, symbolize his Arian beliefs, and able to be built in time and with the resources available. The class then proceeds to discuss how these functional requirements dictated the form. For example, the large dome is cut from a single piece of stone and its most likely function was to discourage vandalism. Similarly, as shown in Figure 3, a system of interlocking masonry joints made it extremely difficult to tamper with the structure without risk of collapsing the roof.

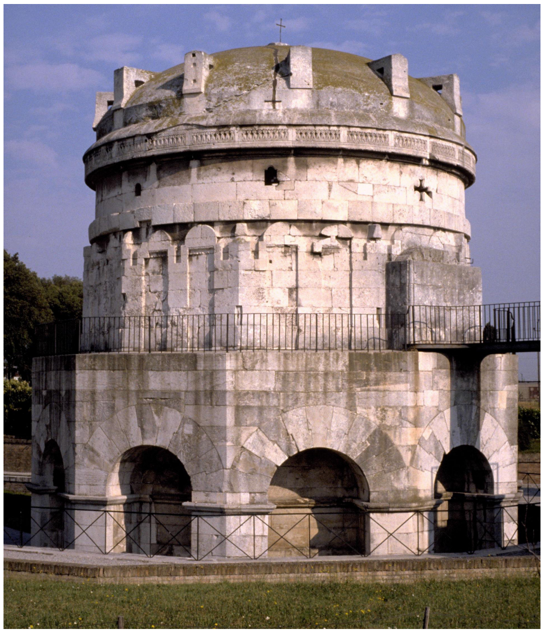

Figure 2. Theodoric's Tomb, Built c. 570 AD, Ravenna, Italy[4]

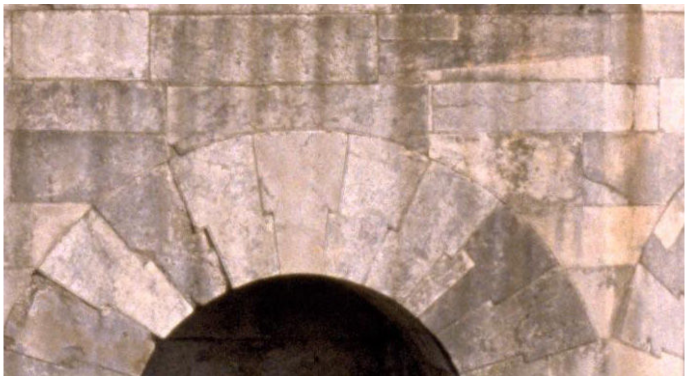

Figure 3. Interlocking Masonry Joints[4]

Case 3. The Roman Pantheon

The Roman Pantheon, shown in Figure 4, which was rebuilt in its present, domed form in 126 AD by Emperor Hadrian, provides for several nice topics of discussion. These include several of the architectural aspects, including the thick walls that support the thrust load from the dome, the open oculus in the dome, the selection and grading of the aggregate material, and the fact that this appears to be one of the first structures designed with an emphasis on its interior, see Figure 5, rather than its exterior. The authors extend this case with an assignment asking the students to, in one page, play the role of the architect (most likely Hadrian, although no one can be certain as of today) and make sketches and provide instructions to the workforce as to how the dome was to be built. A follow-on clip from the History Channel's series entitled *Engineering an Empire* is subsequently shown in class.

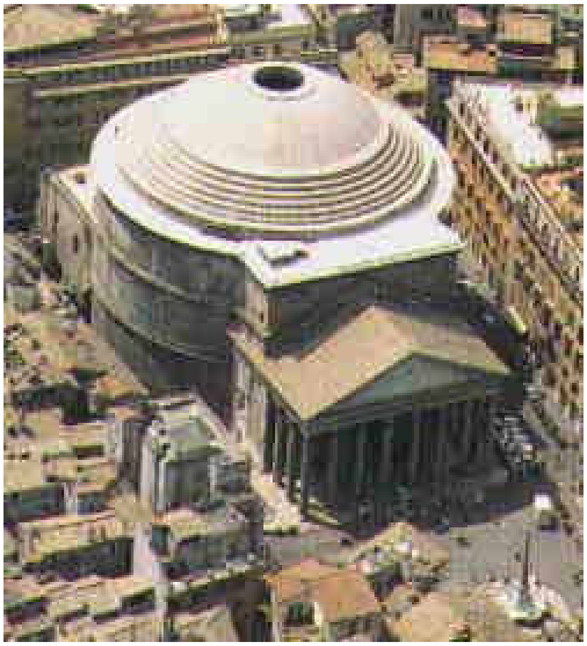

Figure 4. The Pantheon

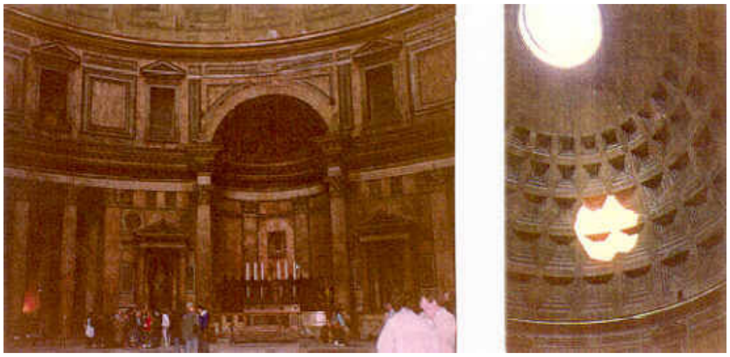

Figure 5. Interior of the Pantheon (Note the rectangular depressions in the ceiling - presumably an aesthetically nice feature which also happens to significantly reduce the weight of the dome)

Case 4. The H.M.S. *Dreadnought*

The H.M.S. *Dreadnought*, Figure 6, was a sufficiently revolutionary naval architecture that many ships have since been classified as belonging to either the pre-*Dreadnought* era or post-*Dreadnought* era. The *Dreadnought* makes an excellent example of how great architectures are often the result of a single great mind; in this case Admiral Sir John "Jackie" Fisher, the lead architect. While Fisher had other notable traits (such as a keen recognition of the importance of the human capital involved in fighting a war), the biggest breakthrough on the *Dreadnought* was the use of the Parsons turbine. This propulsion system, which provided a minimum three knot advantage over reciprocating engine designs of the time, coupled with the "all big gun" armament of the vessel, rendered other battleships obsolete overnight. Constructed in less than a year, the *Dreadnought* was also a triumph of project management.

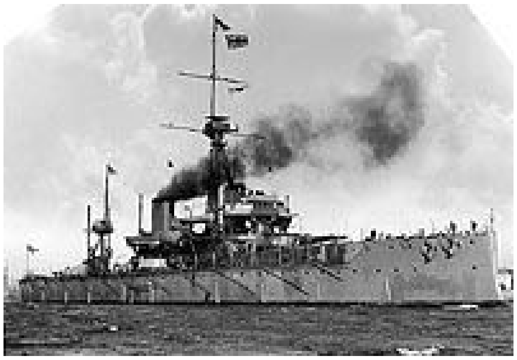

Figure 6. H.M. S. Dreadnought[5]

Case 5. Isambard Kingdom Brunel

Isambard Kingdom Brunel (IKB) went to work in his father's (Sir Marc Isambard Brunel, 1769-1849 AD, himself a famous engineer best known for developing assembly lines and for his tunnel boring machines) small engineering office in England when he was seventeen or eighteen years old. While recovering from a tunnel construction accident, IKB became involved in his first major project—a Suspension Bridge on the Avon Gorge. Although that bridge was not completed during IKB's lifetime (see Figure 7), it illustrated and further honed one trait necessary of an excellent systems architect: the ability to achieve political consensus while effectively blending technical functionality with taste.

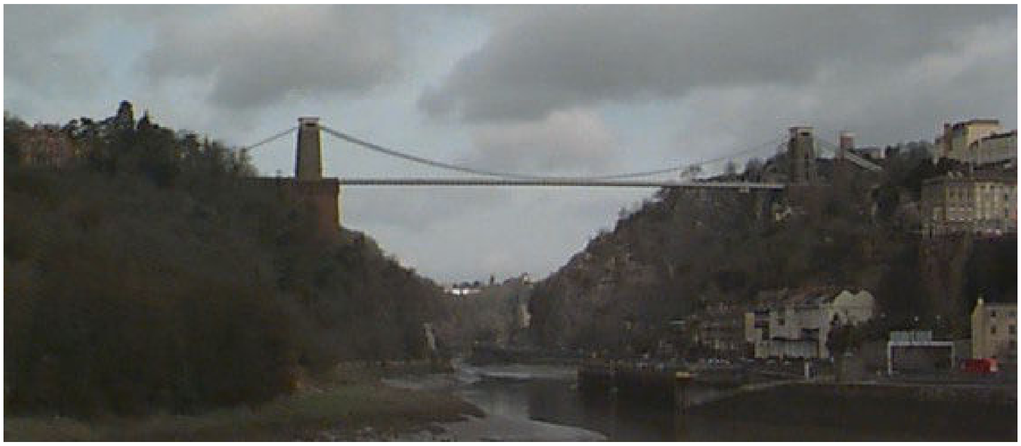

Figure 7. The Clifton Suspension Bridge completed in 1864[6]

Brunel was awarded a survey contract in railroad development, and went on to be the architect of the Great Western Railway (GWR). Overall, the GWR was a great success, although one innovation which did not work out was the gauge. IKB proposed a 7 ft. gauge for better ride and stability, bucking the 4 ft. 8.5 inch standard then in place. There is no doubt that the broad gauge gave superior ride and stability, but it was fighting a *de facto* standard. Standards often win, even when they are not, technically speaking, the best alternative (another example is Betamax vs. VHS). The GWR was built at the wider gauge but was later retrofitted with the standard gage in 1892. IKB's prior knowledge of bridges proved critical as he went on to design and oversee construction of many bridges necessary to complete the GWR. Later, he went on to become a revolutionary naval architect as he developed the *Great Western* (a wooden paddle vessel that was the first steamship to provide regular transatlantic service), the *Great Britain* (an iron-hulled steamship which was the first large vessel driven by a screw propeller), and the *Great Eastern*, Figure 8 (which was not successful as a passenger ship, but achieved fame by laying the first successful transatlantic cable). IKB's experiences provide for a nice discussion of how even great system architects often produce some less than excellent results along the way; IKB promoted and built a pedestrian toll bridge over the Thames which was a commercial flop, as well as an atmospheric railway system which had several issues rendering it a failure.[7,8,9]

Figure 8. The Great Eastern

The authors are also considering developing new systems architecture cases related to the Curta calculator and the Unimog. It is a continuing challenge to determine which cases should be presented in class and which topics should be left for students to explore. Self-directed out-of-class review of some of the case studies followed by related assignments is one potential way to address this dilemma.

Systems Engineering Cases

Case 6. NASA's Deep Impact Mission[10,11,12,13,14]

NASA's Deep Impact Mission consisted of the 2005 impact of a payload with a comet while a flyby craft captured images and data for relay back to Earth. Since UDM students take the systems architecture course in the semester prior to the systems engineering course, the authors begin this by reviewing some of the architectural aspects of the flyby craft and the impactor craft, Figure 9. Next, many of the publicly known details of how the architecture was executed, i.e., the systems engineering aspects of the project were summarized. This provided a nice introductory case illustrating system partitioning, the development and flowdown of technical specifications, the development and execution of testing at various system levels, and the validation of the completed system as it executed its objective. There are a few system engineering issues to discuss in more detail, such as mass allocation and management throughout the development. For example, coming in underweight is not a bonus when the team had designed the gyroscopes and control systems assuming target mass. The development team also struggled with a relatively late discovery of a cracked mirror mount issue and a post-launch mirror focus/resolution problem. Overall, the mission appears to have achieved all of its objectives with the exception of obtaining clear images of the impact crater (the severity of the dust cloud from the impact was apparently underestimated).

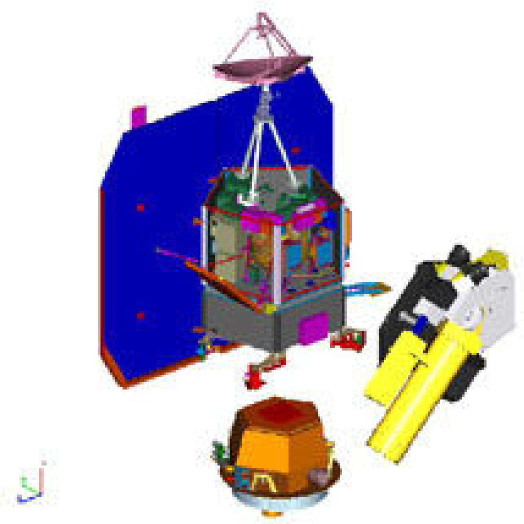

Figure 9. Deep Impact Architecture

Case 7. Ariane 5

The Ariane 5 is a rocket developed by the European Space Agency (ESA). It was developed by approximately 500 scientists in over 10 years for approximately $500 million. It launched June 16, 1996 carrying four satellites. Then, 37 seconds after launch, at 3,700 m altitude the backup Inertial Reference System failed, followed immediately by failure of the active Inertial Reference System. The launcher veered abruptly and exploded, Figure 10.

Figure 10. Explosion of Ariane 5

The failure was determined to have resulted from an Internal Inertial Reference System software exception caused during data conversion from 64-bit floating point to 16-bit signed integer. The floating point number had a value too large to represent as a 16-bit signed integer. This resulted in an operand error (in the main and redundant computers). It turns out that the Ariane 5 software performed exactly to specification but that the specification was wrong. It was originally written for the Ariane 4; the Ariane 5 team made assumptions that were invalid for the Ariane 5 (the horizontal velocities were much higher for Ariane 5). Due to budget constraints, these assumptions were never appropriately (re)validated. Ironically, these pieces of code served no function after launch, and were only left running for 40 seconds into launch in case of a brief delay before the scheduled launch.

The Ariane 5 case provides for a nice discussion on the importance of many systems engineering activities, including specifications, completeness of failure mode and effects analyses, common mode failures for redundant systems, and the importance of fully understanding legacy systems which are reused.

Case 8. Mars Pathfinder[15, 16, 17]

The Mars Pathfinder was a major NASA initiative to develop systems "Faster, Better, and Cheaper (FBC)." The launch, landing, and deployment of the Sojourner rover were flawless. Then, a few days into the mission, occasional software resets were occurring, rendering the system useless until the next day's commands were uploaded. The Pathfinder contained an interface (a bus or shared memory area) which linked many systems of varying priority. One low priority task, meteorological data transmission, involved occasional transmission of large amounts of data. Therefore the transmission was broken into smaller, but substantial, packets of data between which higher priority tasks could obtain the bus and execute their task. Occasionally some longer running medium priority tasks would get into the queue ahead of a high priority task (a priority inheritance feature of the VxWorks bus was not enabled). After a while, watchdog software on the spacecraft noticed that the high priority task had not completed within its designated time and initiated an automatic reset. The system sat idle until it got the next day's communication from Earth. Provisions had been made to enable uploading of software changes, so this was fixed and the problem never reoccurred.

Resets had occurred in the Jet Propulsion Laboratory's (JPL) months of pre-flight testing. They had never been reproducible or explainable, and were dismissed as probable "hardware glitches." Interestingly enough, the JPL engineers actually created a priority inversion situation during testing, but did not manage to analyze their recorded data well enough to conclude that priority inversion was indeed a bug in their system. The tests run were sufficient, but analysis of the test results was insufficient.

From a systems engineering perspective, the Mars Pathfinder nicely illustrates the importance of having complete and appropriate validation tests and the importance of understanding/acting upon all test anomalies.

Case 9. Therac-25[18, 19, 20, 21]

The Therac-6 was an X-ray-only predecessor to the Therac-25. The Therac-20 was an X-ray or high energy electron treatment machine similar to what would later become the Therac-25. However, the Therac-20 featured independent hardware safeguards and interlocks designed to prevent radiation overdoses and injury (potentially fatal) to patients.

The Therac-25 was designed with more attention on software interactions with the operator; software, not hardware, was used to provide crucial safety precautions. A hard wired version was in operation in 1976, with the software-intensive versions first out in 1982. From 1985 until 1987, six patients received massive radiation overdoes, several of them dying from the overexposure.

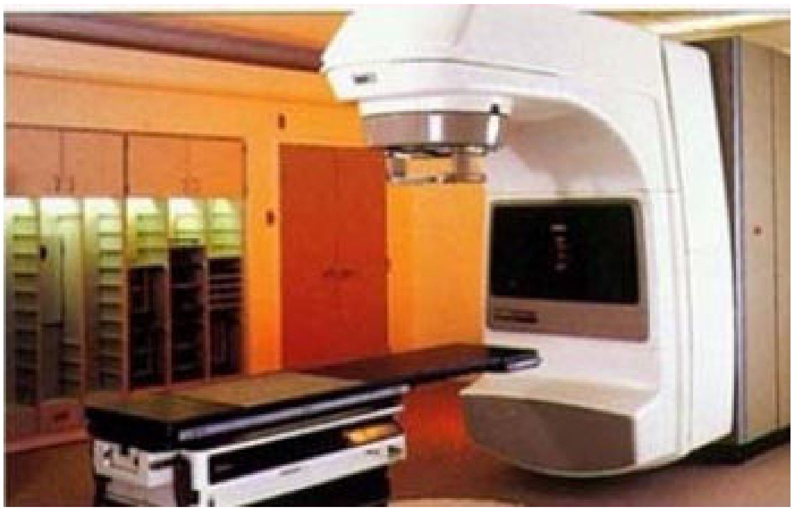

Figure 11. Therac-25

After much denial of a problem, the development team investigated the situation. On the Therac-25, the part of the computer program that is often referred to as the “house-keeper task” continuously checked to see whether the turntable was correctly positioned. A zero on the counter indicated to the technician that the turntable was in the correct position. But the highest value the counter could register was 255. If the program reached 256 checks, the counter automatically clicked back to zero. For that split second, the Therac-25 believed it was safe to proceed when, in fact, it was not. If the technician hit the "set" button to begin treatment at that instant, the turntable would be in the wrong position and the patient would be struck by a full-power, unshielded X-ray beam.

The turntable problem was fixed but accidents continued. After more attempts to deny/isolate/correct the continuing problems, it was found that operators who had become very adept with the software could make edits to the input parameters so quickly that the software could not keep up, generally resulting in a massive overdose coupled with an error message and an indication that the “patient received no dose” (compounding failure modes that lead to several more massive overdoes).

The Therac-25 case provides a convenient topic around which to discuss several important aspects related to systems engineering, including the role of software and its validation in complex systems, the importance of fully understanding the man/machine interface, designing for safety, and the importance of a thorough failure modes and effects analysis.

Case 10. R-16

In 1960, Russia and the United States were in an arms and space race. This created tremendous pressure for quick successes. The R-16, Figure 12, was the brainchild of Mikhail Yangel and was designed to deliver a ten megaton warhead a distance of over 10,000 miles.

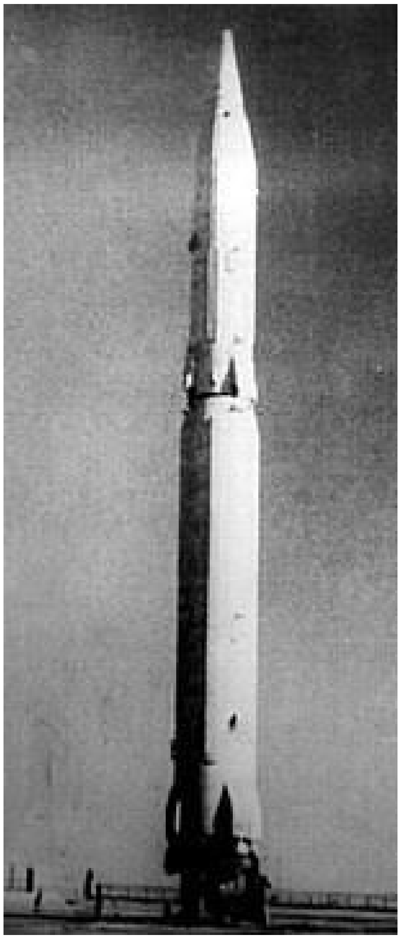

Figure 12. Russian R-16

The first R-16 missile was fueled on October 23, 1960 with a nitric acid-based fuel. Systems checks revealed various problems in the electrical connections and with the targeting system. Marshal Nedelin, commander of the Strategic Rocket Forces, made a decision to resolve all the issues without defueling the launch vehicle. He had a deck chair brought out to the launch pad so he could watch and supervise the work first-hand. A device called the PTR (or Programming Current Distributor in English), which activates the systems onboard the rocket in a certain sequence, was left in a post-launch position after a series of tests. Not knowing the errant status of the PTR and fearing that the cold might affect the batteries, the team energized the batteries earlier than specified in normal launch procedures. The membranes on the fuel and oxidizer lines of the second stage had been activated earlier due to an electrical wiring error. Because of these actions and faults, only one valve kept the components of the self-igniting propellant out of the second stage's combustion chamber.

A witness in the command bunker reportedly overheard someone ask "So should I move PTR to zero?" and someone else reply, "Go ahead." On its way to a "zero" position, the PTR switch activated an electrically-driven pneumatic valve controlling the ignition of the engine on the second stage of the rocket. This command was intended as a back up to the primary system, which normally would ignite the engine of the second stage in flight. The second stage engine ignited, the flame of the engine burst through the fuel tank of the first stage directly below, and the resultant enormous explosion killed 92, including Nedelin himself, Figure 13. Ironically, Yangel was unharmed because he had left the area to smoke a cigarette.

Figure 13. R-16 Explodes

The commission investigating the incident concluded that the management of the testing was overly confident in the safe performance of the complex vehicle, which resulted in the making of decisions without thorough analysis. The direct cause of the accident was the shortcomings in the design of the control system; these allowed unscheduled operation of the valve that triggered the incident. This problem was not discovered during all previous tests. The disaster could have been avoided if the reconfiguration of the current distributor was conducted before the activation of the onboard power supply. [22,23,24,25]

As a class, we continue to discuss similarities from a systems engineering perspective between the R-16 and the Hubble Telescope, Three Mile Island nuclear plant, Chernobyl, and Apollo 13.

Additional Cases

As mentioned earlier, the above ten cases are a subset of the cases the authors use in the Systems Architecture course, the Systems Engineering course, and/or the combined Systems Architecture and Engineering course. Some of the additional lead-in cases relating to systems architecture which the authors use include:

- Robert Goddard’s Rockets
- Project Orion
- Boeing 787 and Airbus A380
- Invention of the Pipe Organ
- Mars PathFinder
- NASA’s Great Observatories
- The work of James Eads
- Siegfried Marcus and his development of gasoline powered motorcars

Additional cases relating to systems engineering which the authors use include:

- Hubble Telescope
- NASA's Cassini/Huygens mission
- Automotive Engine Cooling Systems
- Engine Powertrain Development using Design Structure Matrices
- Role of systems engineering in the development of the Boeing 777

In addition to the relatively short lead-in case studies used mainly in an introductory and motivational fashion, the authors employ several other substantially longer systems architecture case studies (the details of which would be too lengthy for this forum). These longer cases include Intelligent Transportation Systems (ITS), high speed trains, Iowa Class battleships, the B-52 bomber, and the F/A-22 Raptor. Some of these cases are the work of prior students who were asked to develop a new case study as a course assignment. For example, the F/A-22 case study is a result of work done by the second author (in a small team) while he was a student in the MPD Program.

Summary and Conclusions

The authors believe that a case based approach to teaching systems architecture and systems engineering is a powerful way to motivate and engage the students while broadening their perspectives on technology and complex system development. This paper has summarized ten of the specific case studies which the authors use with a brief discussion of how each one relates to specific topics and learning objectives of the courses. This case-based approach has been applied to separate, semester long courses in Systems Architecture and Systems Engineering as well as a condensed version of those two courses into a single semester course entitled Systems Architecture and Systems Engineering.

Future Work

The authors are continually looking for additional topics around which to develop case studies and plan to develop new case studies and updating existing ones on a regular basis. In addition, the authors plan to further strengthen the connections between the case studies and the content and tools introduced in the course.

If there is sufficient interest amongst the academic community in further publication, the authors plan to publish a follow-on paper summarizing some of the cases omitted herein. Additional publications on some of the more substantial case studies may also be developed.

References

[1] Edward Crawley, Ph.D.'s definition of Systems Architecture, presented during curriculum development meetings between the University of Detroit Mercy, Massachusetts Institute of Technology, and Rochester Institute of Technology, 1999.

[2] SE Handbook Working Group, *Systems Engineering Handbook*: International Council on Systems Engineering, 1999.

[3] The Royal Ship *Vasa*, available at http://home.swipnet.se/~w-70853/WASA2e.htm, accessed January 17, 2006.

[4] “The Mystery of Theoderic's Tomb Solved!,” *Archaeology Odyssey*, Nov/Dec 2003.

[5] British Navy Ships—HMS *Dreadnought* (1906-1922), available at http://www.history.navy.mil/photos/sh-fornv/uk/uksh-d/drednt9.htm, accessed January 17, 2006.

[6] Little Engineering Giant, available at http://www.engr.mun.ca/~gpeters/5101brun.html, accessed January 17, 2006.

[7] Historical Metallurgy Society, Spring 2004 Newsletter

[8] Isambard Kingdom Brunel, available at http://web.ukonline.co.uk/b.gardner/brunel/kingbrun.html, accessed January 17, 2006.

[9] www.theengineerguy.com **Tunnels** (broadcast May 25, 2004)

[10] Discovery Channel: Comet Collision! TV Special, last aired on August 27th, 2005.

[11] Official Deep Impact Website of Jet Propulsion Labs and University of Maryland, available at: http://deepimpact.jpl.nasa.gov/home/index.html, accessed January 17, 2006.

[12] Space.com Special Report on Deep Impact, available at: http://www.space.com/deepimpact/, accessed January 17, 2006.

[13] Space camera flaw traced to earthly mirror, available at: http://msnbc.msn.com/id/8247741/, accessed January 17, 2006.

[14] Ball Aerospace (Primary Contractor) website on project, available at: http://www.ballaerospace.com/deepimpact.html, accessed January 17, 2006.

[15] Mars Pathfinder Home, available at http://mars.jpl.nasa.gov/MPF, accessed January 17, 2006.

[16] Spirit Rover Rebooted On Mars, available at http://www.msnbc.msn.com/id/5000299, accessed January 17, 2006.

[17] Files Overloaded Mars Probe, available at http://news.bbc.co.uk/1/hi/sci/tech/3431617.stm, accessed January 17, 2006.

[18] D.M. Buede, *The Engineering Design of Systems: Models and Methods*, Wiley Interscience, 2000.

[19] The Therac-25 Incicdent, available at http://courses.cs.vt.edu/~cs3604/lib/Therac_25/TheracClass.html, accessed January 17, 2006.

[20] Therac-25 accidents, available at http://www.mrs.umn.edu/~mcphee/Courses/Readings/Therac_25_accidents.html, accessed January 17, 2006.

[21] FATAL DOSE: Radiation Deaths linked to AECL Computer Errors, available at http://www.ccnr.org/fatal_dose.html, accessed January 17, 2006.

[22] The Nedelin catastrophe, available at http://www.astronautix.com/articles/therophe.htm, accessed January 17, 2006.

[23] The R-16 ICBM, available at http://www.videocosmos.com/r16.shtm, accessed January 17, 2006.

[24] R-16, available at http://www.astronautix.com/lvs/r16.htm, accessed January 17, 2006.

[25] Nedelin disaster, available at http://www.russianspaceweb.com/r16_disaster.html, accessed January 17, 2006.