

# Can Models Capture the Complexity of the Systems Engineering Process?

**Krishna Boppana, Sam Chow, Olivier L. de Weck, Christian LaFon,  
Spyridon D. Lekkakos, James Lyneis, Matthew Rinaldi, Zhiyong  
Wang, Paul Wheeler, Marat Zborovskiy**  
Engineering Systems Division (ESD)  
Massachusetts Institute of Technology (MIT)

**Leonard A. Wojcik**  
Center for Advanced Aviation System Development (CAASD)  
The MITRE Corporation

## 1.1. Introduction

Many large-scale, complex systems engineering (SE) programs have been problematic; a few examples are listed below (Bar-Yam, 2003 and Cullen, 2004), and many others have been late, well over budget, or have failed:

- Hilton/Marriott/American Airlines system for hotel reservations and flights; 1988-1992; \$125 million; “scrapped”
- Federal Aviation Administration Advanced Automation System; 1982-1994; \$3+ billion; “scrapped”
- Internal Revenue Service tax system modernization; 1989-1987; \$4 billion; “scrapped”
- Boston “Big Dig” highway infrastructure project; \$14 billion; \$10 billion over budget and late.

Traditional systems engineering (TSE) defines a step-by-step planned approach to systems engineering, which has proven effective across many systems engineering efforts. However, some systems engineering efforts defy the TSE process, due to various complexity-related factors along a set of characteristics summarized in Figure 1 (White, 2005). In this paper, we compare three modeling approaches in terms of their ability to represent the complexity of the SE process. We apply two of these

2 Can Models Capture the Complexity of the Systems Engineering Process?

approaches to the specific case of the Federal Aviation Administration (FAA) Advanced Automation System program.

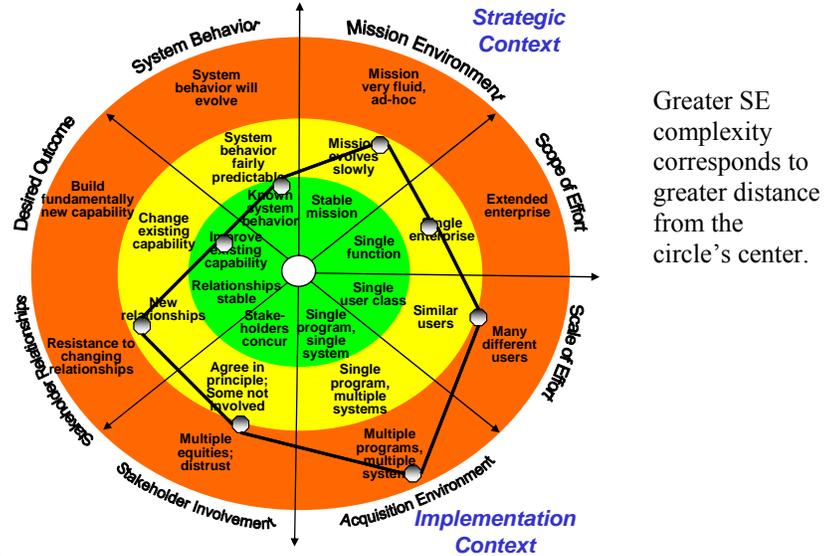


Figure 1. Enterprise systems engineering environment characterization template.

1.2. The Advanced Automation System (AAS) Program as a Case Study to Model the SE Process

The AAS program (1982-1994) was chosen as a case study to compare modeling approaches to the SE process because of AAS program complexity and difficulty, and because extensive information is available about the AAS program in the open literature – we base the model comparison entirely on open literature information.

The AAS schedule, at the time of the contract award to IBM (1988), was broken up into five major portions. These were:

- PAMRI- The Peripheral Adapter Module Replacement Item – This was the initial project of AAS and improved the radar and communication systems of the air traffic control system. This was completed on time.
- ISSS- The Initial Sector Suite System- This is the project which was intended as the initial replacement of the controller workstations.
- TAAS- The Terminal Advanced Automation system- The updating of the terminal area departure and approach control.
- TCCC- Tower Control Computer Complex- This was intended as the automation system for the control towers.
- ACCC- Area Control Computer Complex- This was intended as the automation system for enroute aircraft.

PAMRI, consisting of the peripheral hardware updates was completed on time. The schedule and budget problems were associated with the software development, notably the ISSS development effort.

The AAS event and budget timeline, based on open literature sources (Krebs and Snyder, 1988; Scott, 1988, Levin, et al., 1992; Del Balzo, 1993; Ebker, 1993; Lewyn, 1993; Barlas, 1996; Beitel, et al., 1998), is briefly summarized below:

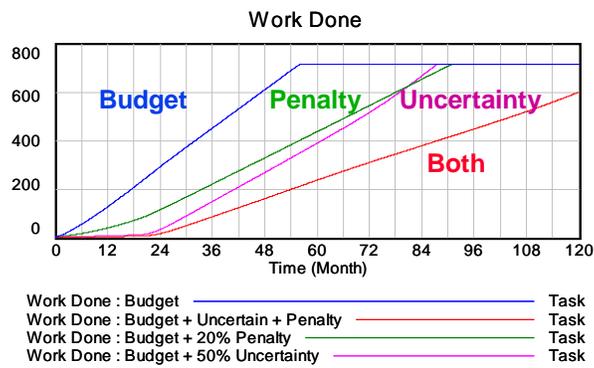
- 1982: The FAA sets the initial requirements for AAS and seeks contractors.
- 1984: IBM and Hughes named the finalists to build the prototype. At this point \$500 million has been spent developing the bid.
- 1988: The FAA awards the prime contract to IBM worth \$3.6 Billion. Hughes protests the award causing an initial project delay.
- 1989: IBM begins work on the AAS. The software component of the project is estimated to be 2 million lines of code.
- 1990 Requirements are still unclear for ISSS as indicated by the 500-700 requirements change requests for the year. To help finalize requirements, IBM builds a prototype center in Gaithersburg, Maryland so that controllers can try out the software under development. Despite the fact that requirements were not clear, approximately 1 million lines of code have already been written. Estimates indicate that 150,000 lines of code will need to be rewritten due to the requirements changes and the resulting bugs. To date the cost overrun is \$242 million.
- 1992: The FAA announces a 14-month delay to the project completion. FAA and IBM shake up their management.
- 1993 April: IBM and the FAA freeze the requirements for ISSS.
- 1993: IBM announces that the project will not be ready until after year 2000. IBM starts working on more methodical, communication-oriented project management philosophy with new managers.
- 1994: The AAS program ceases to exist as originally conceived, leaving its various elements terminated, restructured, or as parts of smaller programs.

### **1.3. Application of System Dynamics to the AAS Program**

System Dynamics (SD) is a methodology for understanding the structures which create dynamic behavior. On a project, those structures consist of (1) work backlogs and the cycling of work among tasks to be done, tasks completed but needing rework (often with significant delays in discovering the rework), and tasks really completed; (2) the feedback control mechanisms by which managers attempt to accomplish the project (adding resources, exerting schedule pressure, and working overtime); and (3) the secondary and tertiary impacts of project control (experience dilution, overcrowding, “haste makes waste”, fatigue, etc.) that undermine the recovery strategy. The interplay of these structures, in the face of project problems (e.g., changing requirements, changing scope, resource bottlenecks, etc.) determines how disruptive these problems become, and ultimately project cost and schedule overrun (Lyneis, 2001).

An important part of the AAS program was the ISSS, and we focused on modeling that aspect of AAS with SD. A simple SD project model developed for the MIT course ESD.36 (System and Project Management) was adapted to represent ISSS. We did this by altering budgets, schedule, normal productivity, and normal work quality to achieve

the budgeted ISSS program, and then exogenously specifying problems that the project experienced. It is very clear from the background research on AAS that uncertain and unrealistic requirements profoundly affected the program. In fact, GAO reports of the mid to late eighties indicate that requirements knowledge was very poor to the point where the GAO had little confidence the AAS would be delivered on time. Requirements were not frozen until 1993, which was approximately 48 months after the project started (Del Balzo, 1993). We represented these requirements problems in two ways: (1) a time-varying impact of *uncertain* requirements on work *quality* (a 50% reduction for the first 18 months of ISSS) which was significantly reduced when the Gaithersburg facility came on line<sup>3</sup>; and (2) a constant “penalty” from *unrealistic* requirements to project work *quality* and *productivity* which persisted throughout the project (a 20% penalty).



**Figure 2.** Impact of uncertainty and penalty on ISSS performance.

Figure 2 shows four simulations from the SD model: (1) Budget; (2) 20% Penalty; (3) 50% Uncertainty; and (4) a combination of penalty and uncertainty. The budget finishes in month 58, approximately on schedule. However, with the requirements problems actually experienced by the project (represented by the combination simulation), the project is still not completed by month 120. Because we assume that staffing levels are limited to those actually experienced on the project, any unbudgeted rework generated by the requirements problems translates directly into delayed completion of the project. The other two simulations shown in the figure represent the impacts of uncertain requirements and unrealistic requirements operating alone on the project. Note that with the severe impact of uncertainty early in the project, very little work actually gets completed. However, once Gaithersburg comes online and uncertainty is eliminated, progress accelerates. In contrast, with a requirements penalty progress occurs sooner but the pace never really accelerates as the penalty persists

<sup>3</sup> Gaithersburg coming on line also significantly reduced the time to discover any rework.

throughout the project. With either the penalty or uncertain requirements alone, the project would have finished around month 88.<sup>4</sup>

We then conducted a series of sensitivity tests to see which aspect of the project problems had the greatest impact on project performance. We varied: sensitivity to penalty (to 10% and 30% from the base case of 20%); sensitivity to uncertainty (to 25% and 75% from the base case of 50%); sensitivity to Gaithersburg online (to month 9 and month 27 from the base case of month 18). The results are shown in Figure 3. Note that the severity of any penalty from unrealistic requirements has the greatest impact on performance. This is largely because the assumed impact persists throughout the project. Even strong uncertainty has little incremental effect as long as it is eliminated early. When Gaithersburg comes online has little impact within the range tested. This suggests (and we emphasize suggest) that the unrealistic requirements had a greater effect on the failure of ISSS than the skipped testing and the late opening of the Gaithersburg testing center.

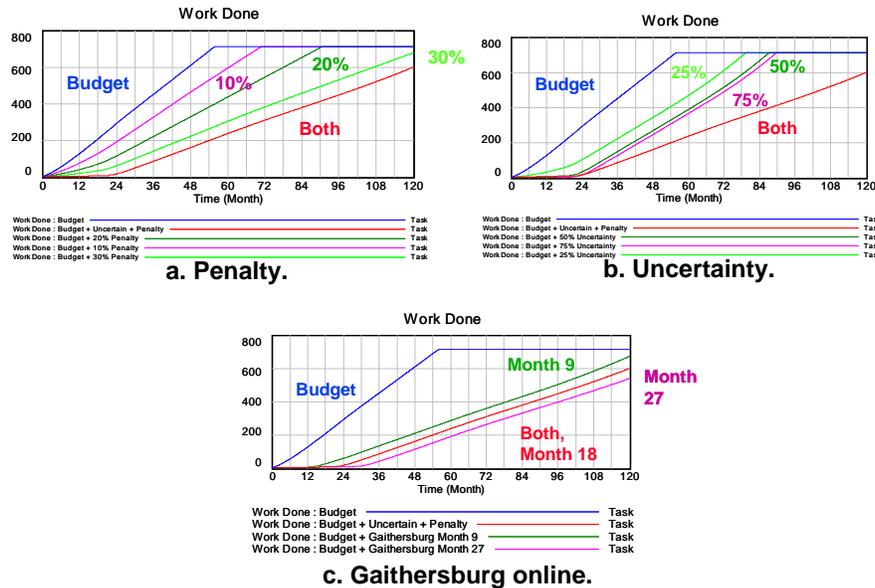


Figure 3. Sensitivity to penalty, uncertainty and Gaithersburg online.

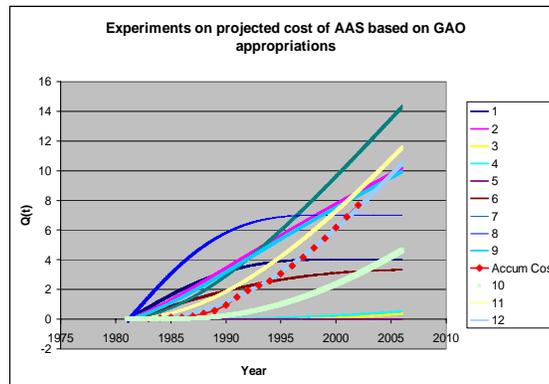
### 1.4. Application of HOT to the AAS Program

Highly optimized tolerance (HOT) is a framework for understanding certain aspects of complexity in designed or engineered systems. Carlson and Doyle (1999) originally developed the HOT concept and applied it to forest fire management. They showed how power laws in event size emerge from minimization of expected cost in the face of design tradeoffs. Since then, HOT has been associated with tradeoff analyses in such

<sup>4</sup> The magnitude of the uncertainty and the penalty are both estimates. We choose these values for the base case simulation as they produced approximately equal impact on project progress.

systems as internet architecture (Alderson, et al., 2004). The HOT approach has been adapted to the SE process (Wojcik, 2004), resulting in an open-loop control model.

In the HOT model of the SE process, the function  $s(t)$  represents the progress of the SE program and may reach a maximum value of 1 if the program is completed. An incomplete SE program will have a maximum value of  $s(t)$  less than 1. The HOT model is based on minimizing total cost across program lifetime, where cost per unit time has three terms. The first term in the cost density function is the pressure to finish the project. Coefficients  $A$  and  $\tau$  generate cost pressure to complete the program: the larger the value of  $A$ , the more cost pressure to complete the program. The parameter  $\tau$  is a constant representing how pressure builds up over time; the larger the value of  $\tau$ , the more time it takes for pressure to build up. The second term in the cost density function is the impact from random events, rolled up into a stochastic function of time to simulate stakeholder interactions, external factors and other unanticipated events. Two parameters,  $B$  (magnitude of random event cost impact) and  $p$  (probability per unit time of random events) combine to form a product coefficient  $Bp$ . The third term in the cost density function models the inherent technical difficulty of the SE program, parameterized by a coefficient  $D$ .



Each curve corresponds to a different set of HOT model parameters. Details available upon request.

**Figure 4.** Experiments on projected cost of AAS based on program appropriations.

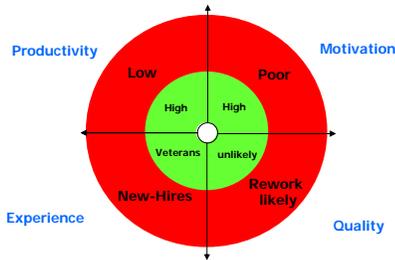
To apply the HOT model to the whole AAS program, we used the project plan and original budget to calibrate the values of  $A$ ,  $Bp$ ,  $D$  and  $\tau$ , as they appeared at the beginning of the AAS program. In particular,  $Bp$  is set equal to 0. Then, a revised set of parameters  $A^*$ ,  $Bp^*$ ,  $D^*$  and  $\tau^*$  was estimated from the actual trajectory of the AAS program, using a design of experiment (DOE) based on an orthogonal array for four factors at three levels and three confirmation runs. The results are summarized in Figure 4, which shows realized cumulative cost of the program for various parameter value combinations and a “best fit” to the actual cumulative cost. We found that the best fit  $B^*p^*$  has non-zero value,  $A^* > A$ , and  $D^* > D$ . From these parameter comparisons, combined with post-mortem assessments of the AAS program, our interpretation is that the original AAS plan underestimated both the level of uncertainty of the effort ( $B^*p^* > Bp$ ) and the inherent technical difficulty of the AAS program ( $D^* > D$ ). We also suggest that  $A^* > A$  may indicate a problem with managerial effectiveness in the AAS program; more pressure to complete was applied from outside the program than originally anticipated.

### 1.5. Extensions to the Original HOT Model Based on COSYSMO

COSYSMO is a parametric cost model for systems engineering projects derived from a combination of data from past systems engineering efforts and subjective inputs from systems engineering experts (Valerdi, Miller and Thomas, 2004). Data used to generate COSYSMO parameters comes from systems engineering efforts across multiple systems and companies (Center for Software Engineering, 2005).

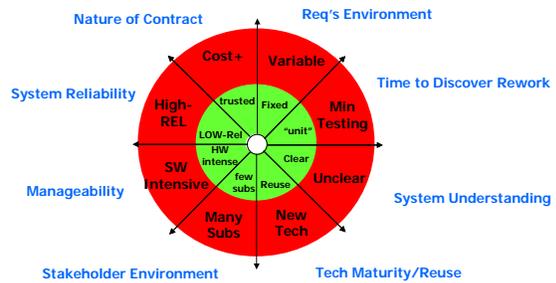
We did not attempt to apply COSYSMO to the AAS SE process; instead we extend the ESE Environment Characterization Template (Figure 1) to include factors in the COSYSMO model. These could potentially be scored subjectively to permit estimation of HOT parameters A, Bp and D, as suggested in Figure 5.

#### Drivers for Pressure to Complete



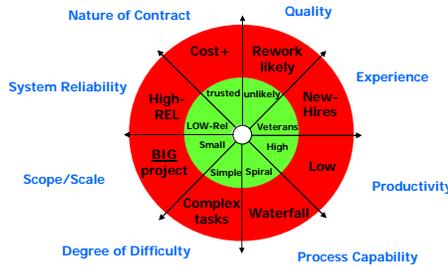
$$A = (0,1)F_1 + (0,1)F_2 + \dots + (0,1)F_4$$

#### Drivers of Uncertainty



$$Bp = (0,1)F_1 + (0,1)F_2 + \dots + (0,1)F_8$$

#### Drivers of Cost due to Build Speed



$$D = (0,1)F_1 + (0,1)F_2 + \dots + (0,1)F_8$$

Figure 5. COSYSMO-based drivers for HOT SE model parameters, A, Bp and D.

### 1.6. Conclusions

SD emphasizes schedule and is validated against significant past experience. SD potentially can show emergent behaviors through interactions and feedback loops. HOT models the whole SE “iron triangle” (cost, schedule, performance) with a relatively simple model; but it is not well-validated against experience. HOT has promise as a higher-level model showing emergent behaviors from planning and re-planning cycles. COSYSMO is calibrated with expert inputs and past experience and covers the whole triangle, but has less potential for displaying truly emergent behaviors. Retrospective application of SD and HOT to the single example of the AAS program is

inconclusive on whether modeling can provide useful insight into complexity and emergence in the SE process, but we see *integration* of the three approaches is a promising approach for future research into complexity in SE, to build on the strengths of all three approaches: SD for relatively detailed process modeling, HOT for coarse, higher-level modeling, and COSYSMO for calibration reference and tie-in of modeling to key factors in previous SE experience. Modeling of both successes and failures in past large-scale engineering programs will provide further insights.

## References

- Bar-Yam, Y., 2003, "When Systems Engineering Fails-- Toward Complex Systems Engineering," *International Conference on Systems, Man & Cybernetics*, Vol. 2: 2021- 2028, IEEE Press, Piscataway, NJ.
- Barlas, S., 1996, "Anatomy of a Runaway: What Grounded the AAS." *IEEE Software* 13(1): 105.
- Carlson, J. M. and Doyle, J., 1999, "Highly Optimized Tolerance: A Mechanism for Power Laws in Designed Systems," *Physical Review E* 60(2): 1412-1427.
- Center for Software Engineering, 2005, "COSYSMO" web page, University of Southern California, <http://www.valerdi.com/cosysmo/>
- Cullen, A., 2004, "The Big Money for Big Projects," *Harvard Business School Working Knowledge*, June 14, 2004, U.S.
- Del Balzo, 1993, Statement Before the House Committee on Appropriations, Subcommittee on Transportation, Concerning the Advanced Automation System, April 19.
- Erbker, G. W., 1993, Statement of Gerald W. Ebker, IBM Vice President Chairman and CEO of IBM's Federal Systems Company Before the Subcommittee on Transportation and Related Agencies, House Appropriations Committee.
- Krebs, M. and J. Snyder, 1988, "Hughes Quits Fight with IBM for Air-Traffic-Control Pact." *Automotive News*: 8.
- Levin, R. E., R. D. Wurster, et al., 1992, GAO/RCED-92-264 Air Traffic Control: Advanced Automation System Still Vulnerable to Cost and Schedule Problems, GAO: 5.
- Lewyn, M., 1993, "Flying In Place: The FAA's Air-Control Fiasco." *Business Week*: 90.
- Li, A., 1994, Advanced Automation System: Implications of Problems and Recent Changes: Statement of Allen Li, GAO: 10.
- Lyneis, J., Cooper, and Els, 2001, "Strategic Management of Complex Projects: A Case Study Using System Dynamics," *System Dynamics Review* 17(3).
- Richard C. Beitel, J., M. P. Dunn, et al., 1998, AV-1998-113 Office of Inspector General Audit Report Advance(sic) Automation System, DOT: 8.
- Scott, W. B., 1988, "Protest Prompts Suspension of \$3.6-Billion FAA Contract to IBM." *Aviation Week and Space Technology* 129(7): 27.
- Valerdi, R., Boehm, B. W. & Reifer, D. J., 2003, "COSYSMO: A Constructive Systems Engineering Cost Model Coming of Age," submitted for the *INCOSE 2003 Symposium*, June 29 – July 3, Washington, DC.
- White, B., 2005, "A Complementary Approach to Enterprise Systems Engineering," The MITRE Corporation, presented at the *National Defense Industrial Association 8th Annual Systems Engineering Conference*, San Diego, CA.
- Wojcik, L., 2004, "A Highly-Optimized Tolerance (HOT)-Inspired Model of the Large Scale Systems Engineering Process," The MITRE Corporation, in *Student Papers: Complex Systems Summer School*, Santa Fe Institute, Santa Fe, New Mexico, U.S., June 6-July 2.

The contents of this document reflect the views of the authors and The MITRE Corporation and do not necessarily reflect the views of the FAA or the DOT. Neither the Federal Aviation Administration nor the Department of Transportation makes any warranty or guarantee, expressed or implied, concerning the content or accuracy of these views.

Copyright © 2006, The MITRE Corporation and Massachusetts Institute of Technology. All rights reserved.