GLOBAL PRODUCT DATA INTEROPERABILITY SUMMIT 2022

Slow Down to Go Fast; Effect of Early Modeling on Life Cycle Costs

Paul A. Lucas
Northrop Grumman Mission Systems
Technical Fellow
09/27/22

Paul A. Lucas Bio

- **MS Physics-1995 SUNY Stony Brook**
- **MSEE-2008 UMBC Maryland**
- **NY State Tetris Champion, 1989**

- **Worked primarily in modeling and simulation since 1996. Started at Northop Grumman on the SPQ-9B shipboard radar and have largely worked on radar systems. THAAD and SPY-3 for Raytheon. APG-81/83 and SAP systems for Northrop Grumman Mission Systems. Systems lead for the AN/AAQ-37, APG-83, SMADL systems.**

- **Free time spent raising my two sons, scuba diving, and doing most things aquatic.**

BLUF

- **Slowing down and modeling systems early in the lifecycle saves schedule, reduces cost, and drastically increases system capability**

- **We will compare and contrast two similar modeling efforts to create digital twins for the F-35 program**
  - **APG-81 RADAR**
    - **Airborne fire control radar**
      - **Air to Air, Air to Ground, Target ID, Mapping, Electronic Warfare, Navigation Support**

  - **AN/AAQ-37 Electro Optical Infrared Distributed Aperture System (DAS)**
    - **Passive EO/IR 4Pi Steradian sensor**
      - **Missile Warning, Aircraft Situation Awareness, Imaging, Launch Point Detection**

Taking the time early to model early saves budget, improves later program schedule, and increases capability

Customer Base for Both Twins

- **Contracted by Lockheed Martin Aero for the purpose of their integration lab facilities for the F-35 program**
  - **Integration and Test System Integration Lab (SIL)**
    - **Purpose to integrate and sell off the F-35 fighter in a simulation environment**
    - **Migrated from the test SIL in Fort Worth to a government directed effort at various SILS (Pt. Mugu, Pax River, etc.)**
  - **Fusion SIL**
    - **Labs focused on early integration of systems negotiating through the LM Aero Fusion effort**
    - **Heavy focus on early look at data flows and cueing**
  - **Trainer SIL**
    - **Separate division that Aero focusing on Pilot trainers for the U.S. and Foreign customers**
    - **Focus on Pilot level observable effect that could be exported to foreign powers**

- **Internal Customer: Northrop Grumman produced content can be used for internal integration, system development, verification**

Purpose and users of both twins identical

Systems Magnitude Comparison

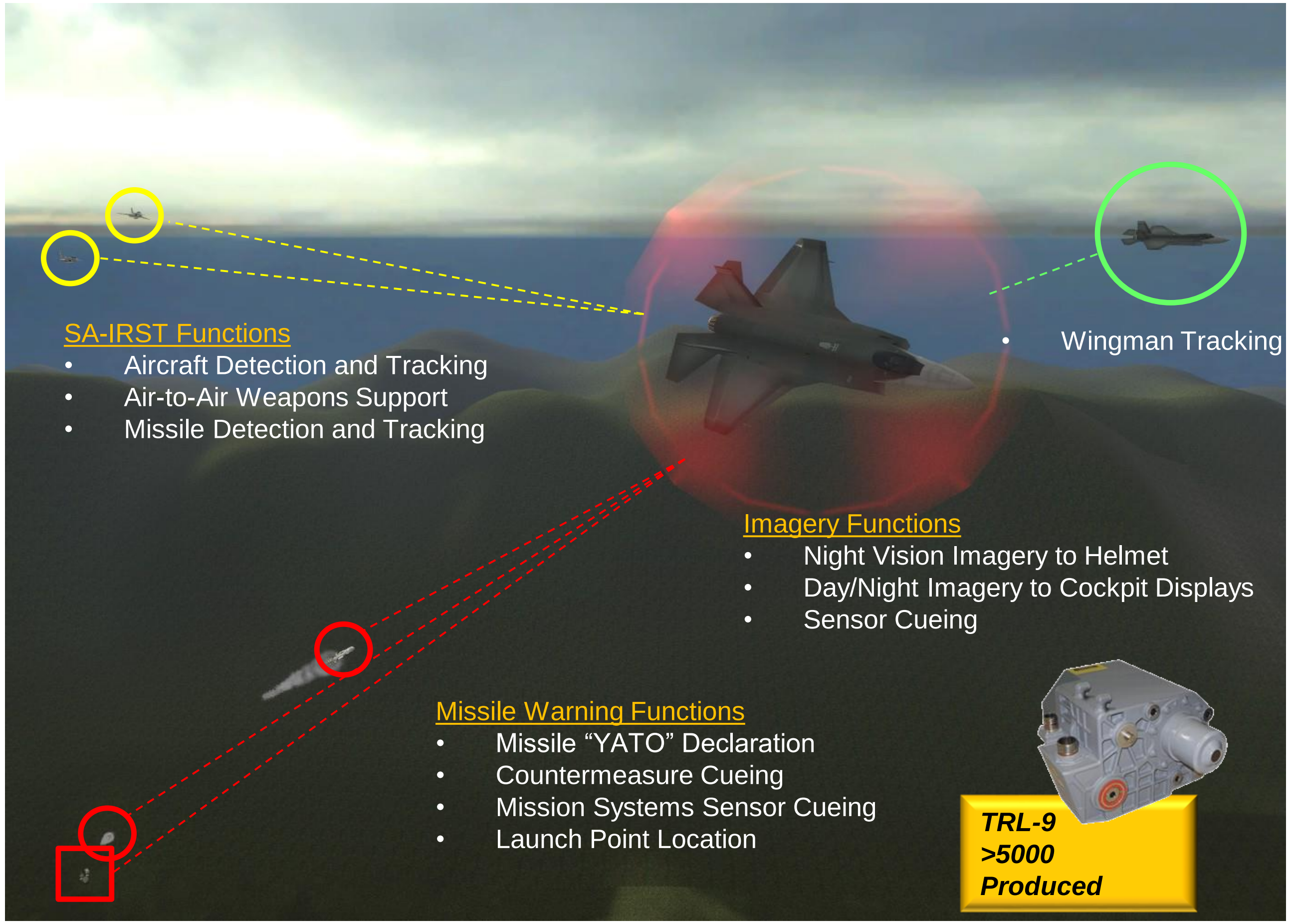
- **Relative magnitude to normalize for differences in systems**
  - **Systems normalized to DAS to preserve company design information**

	Number of Modes	Relative Budget	Time to Complete initial Development	System Problem Report (SPR) count
APG-81 (Radar)	22 total, but large numbers of modes are similar (Air to Air, GMTI, ID)	3.2 X	1.3 X	19.5 X
AN/AAQ-37 (DAS)	Five distinct modes	1 X	1 X	1 X

Total benefit extends well past SPR count

F-35 EO/DAS Capabilities

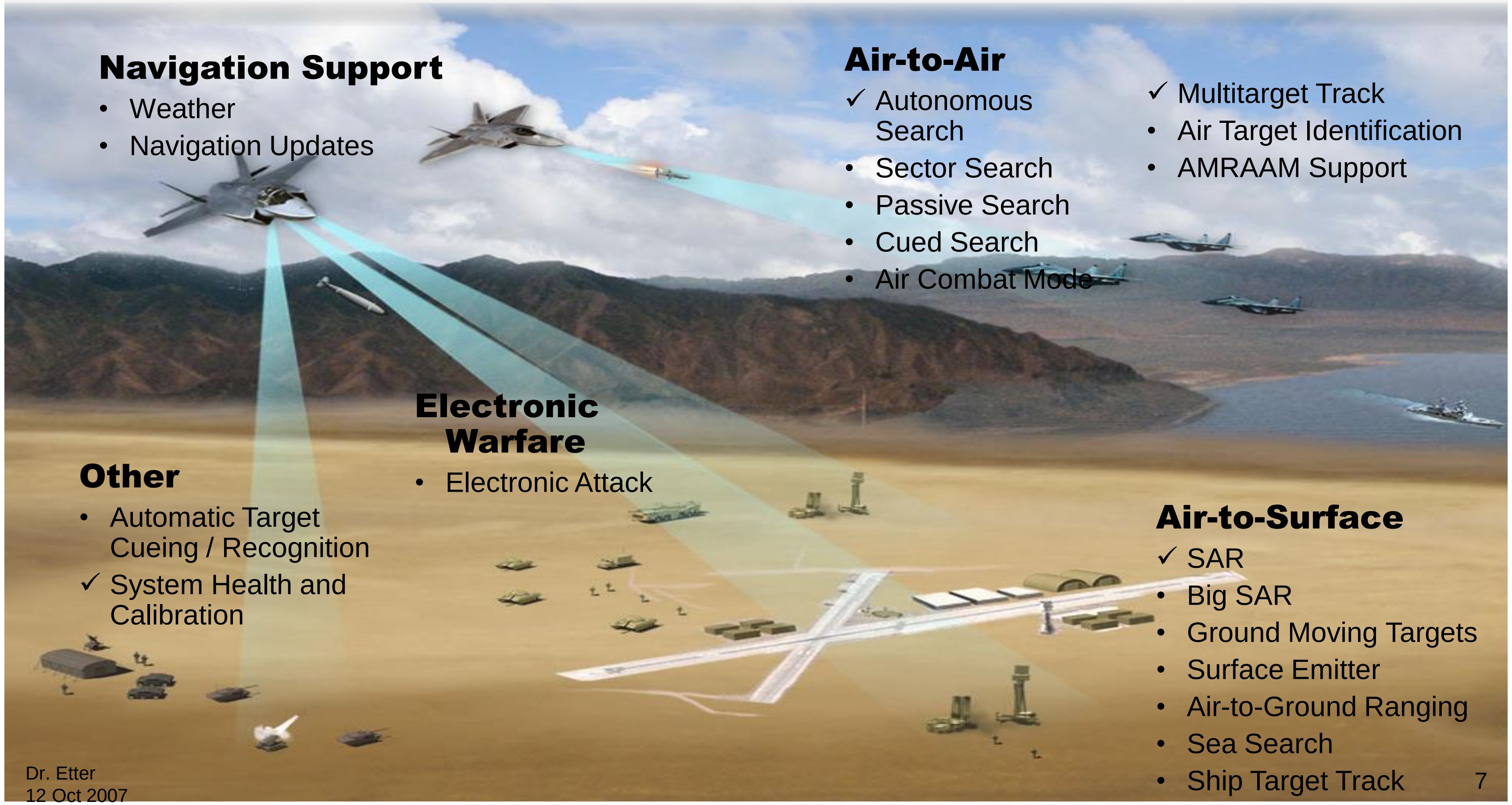
Global Product Data Interoperability Summit | **2022**



Passive Defensive Focused System

F-35 Radar Capabilities

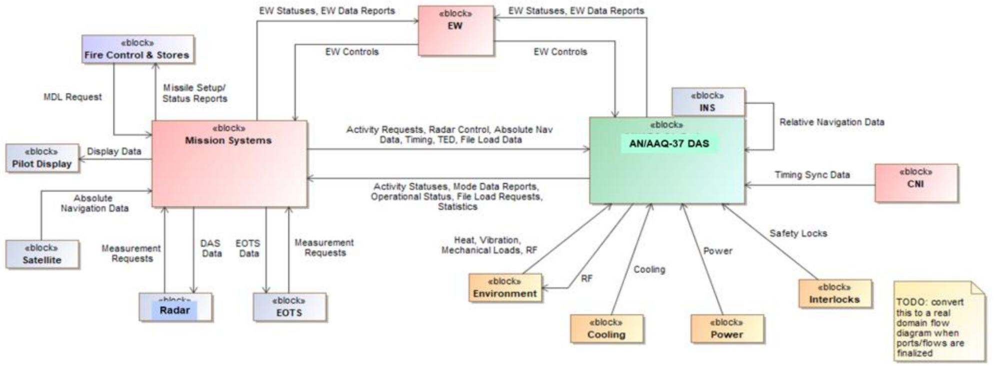
Global Product Data Interoperability Summit | **2022**



Active and Passive Attack and Defense System

Differences in Development of the Twins

- **The F-35 DAS Model effort implemented a SysML of the system before formal development**



- **Additional time spent early in DAS design to create commonality and reuse of Simulation code in the Development code**

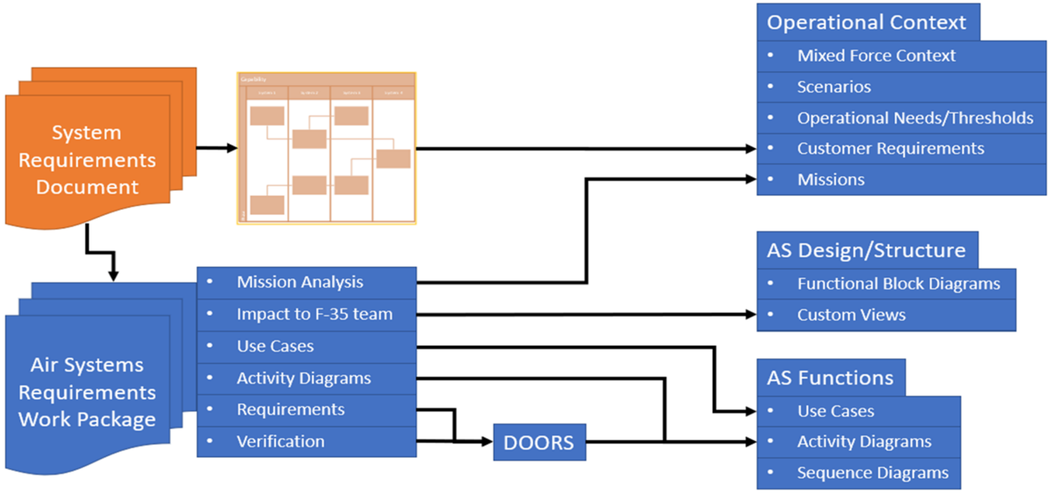
- **F-35 DAS simulation use cases form basis of sell off and verification of baseline system**

DAS did not have two separate development threads between Simulation and Baseline

DAS SysML model

- **SysML is a modeling language which is similar to UML with a systems engineering focus**

- **Northrop Grumman built a SysML Model of the DAS system before code creation to plan and code design**



- **Requirements allocated to the SysML model and flowed through the Twin model into the code and the Use Cases form the basis of the verification scenarios**

Flowing requirements through your model allows for traceability and ease of verifications scenarios

Development of Code for Twins

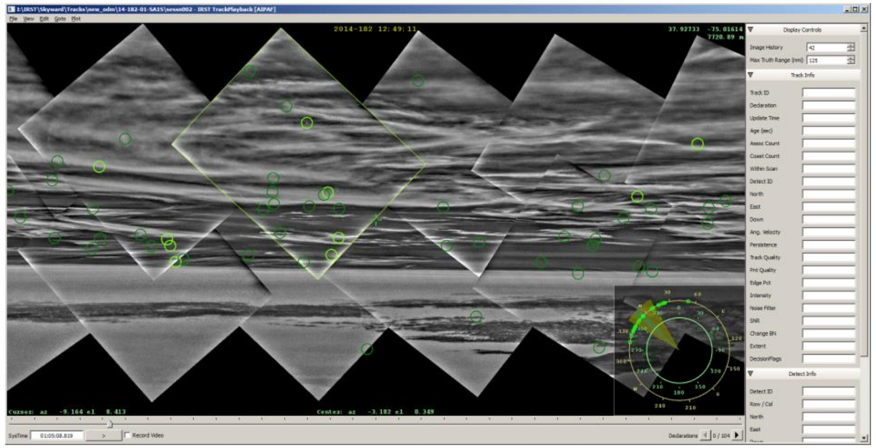
- **F-35 APG-81 Model**
  - **Early prototyping/similar code from another program used as basis of model**
  - **Quick turn around relative to development time of F-35 DAS**
    - **Allowed for early integration to check interfaces**
    - **Allowed for early fusion algorithm development**
  - **Final radar system performance deviated greatly from model performance.**
    - **Significant rework and System Problem Report count resulted**

- **F-35 AN/AAQ-37**
  - **Code was designed with reuse intended**
  - **Slower initial code development**
    - **Simulation code and Development code are essentially identical**
    - **No duplicated effort**
  - **System Problem Reports with Simulation in most cases were actual features of the system**

Early prototyping provides quick results but low fidelity capability that suffer lack of reuse

Verification of Systems Using Twins

- **F-35 APG-81 Model**
  - **Model validated by flight test**
  - **Integrated into customer large-scale airframe simulation to form basis of system and test for government program office customer**
  - **Used by internal Tracker team to verify results and integrate**

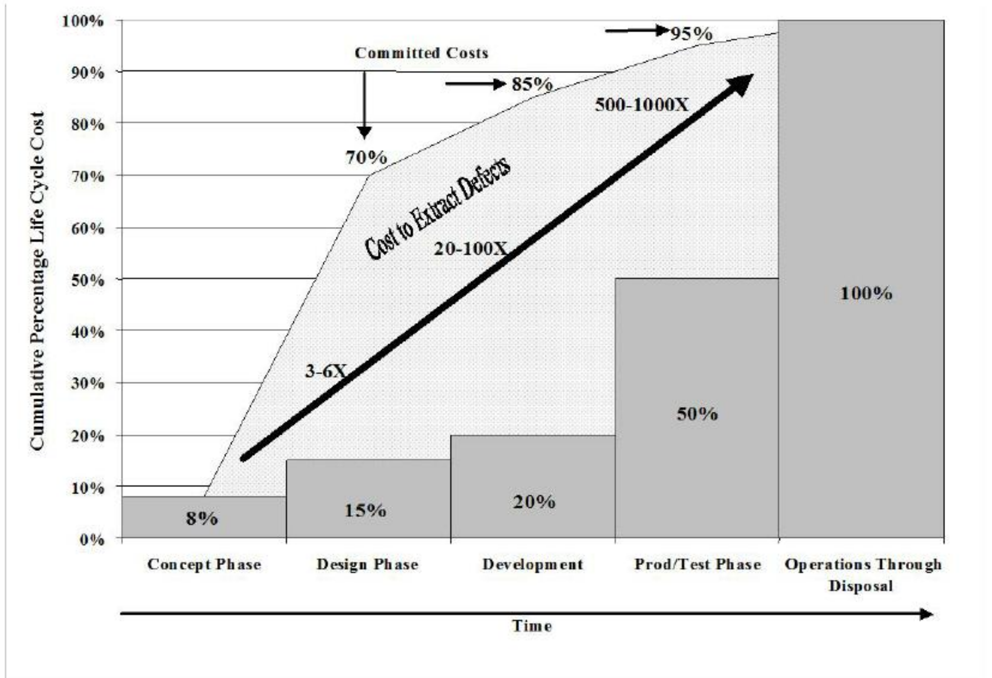


- **F-35 AN/AAQ-37**
  - **Model also validated by flight and integrated into government SIL**
  - **Model forms basis of Verification and Validation of the DAS system**
    - **Simulated results accepted as verification evidence**
    - **Functional Configuration Audit used traceability embedded in the model for proof of compliance**

Program utility of twin is unlimited if proper development and investment is made

Conclusions and Costs By Customer

- **Integration and test SIL**
  - **Significant rework was needed for Radar because of deviation from notional existing code used as basis. High SPR count continues to current day due to approximation of functions. Basic system engineering dictates as life cycle progresses costs to remove defects, increase geometrically. For DAS the code bases are relatively identical, the SPR reports against the twin tend to be features of the system, which is highly desirable in a twin.**



- **Fusion SIL**
  - **Initially Fusion group did not use DAS Twin because of time taken to create initial working code base. Radar was successful in populating interfaces needed for fusion development. As Fusion development lagged, DAS was eventually integrated into lab. DAS worked adequately for Fusion's needs.**

High defect count will drastically increase lifetime costs, but early integration has value

Conclusions and Costs By Customer Continued

- **Trainer SIL**
  - **Both DAS and Radar Twins required exportability reworks to remove any system CPI. Same SPR problems that plagued Integration SIL plagued Radar trainer to a lesser extent. DAS rework was slightly more intensive because of commonality with DAS flight program code base.**

- **Internal Customer**
  - **Internally the Radar model was only used by the Tracker team to integrate and test. For DAS the common code base became the basis for how verification and validation was performed. Model code base was verified against significant data collects during local flight test. Model was then used to demonstrate performance to the Lockheed integration team and government customer. The Functional Configuration Audit of the AN/AAQ-37 was performed using synthetic imagery and simulated results. Flight test data was also used to demonstrate elements of the systems performance. The cost savings and utility of this approach are difficult to characterize but is on order of magnitude with the cost of the entire development effort.**

Despite overhead of system CPI a high-fidelity Twin is useful for system demonstration and sell off

Summary

- **The Northrop Grumman simulation team had the rare chance in its role developing multiple digital twin efforts for the F-35 program with identical requirements and usages to see the benefits of different design choices**
  - **Early prototyping is a focus of the Industry and has its benefits**
  - **Lifecyle costs due to deviation from model and final design can be prohibitive**
  - **Common code base suffers from poor exportability**
  - **Common code base can allow effort to leverage fidelity to provide the simulation as the ultimate source of truth**

Slowing down during initial development allows immense late life cycle benefits including lower costs, schedule improvement, and quality of product

Questions