NORTHROP GRUMMAN

DEFINING THE FUTURE

Automatic Code Generation at Northrop Grumman

June 6, 2007

Robert H. Miller, Ph.D.
Director, Future Unmanned Systems
Northrop Grumman Corporation

History of Automatic Code Generation at Northrop Grumman

- **1997**
  - Initial internal project to look at efficiencies of automatic code generation
  - Prototype implementation of Global Hawk guidance and control
- **2000**
  - Fire Scout prototype flight test demonstration utilized automatic code generation for 6DOF, guidance and control
- **2003**
  - First flight of X-47A using automatic code generation
- **2004**
  - Decision made to consolidate on MathWorks Toolset
  - Prototype Demonstration of UCAR guidance on RMAX helicopter
  - DARPA software enabled control flight test utilizes Stateflow for mode control in addition to guidance

Cleared for Public Release
Control Number 07-057 dated 6/04/07
AFRL-WS 07-1235 dated 5/28/07

NORTHROP GRUMMAN

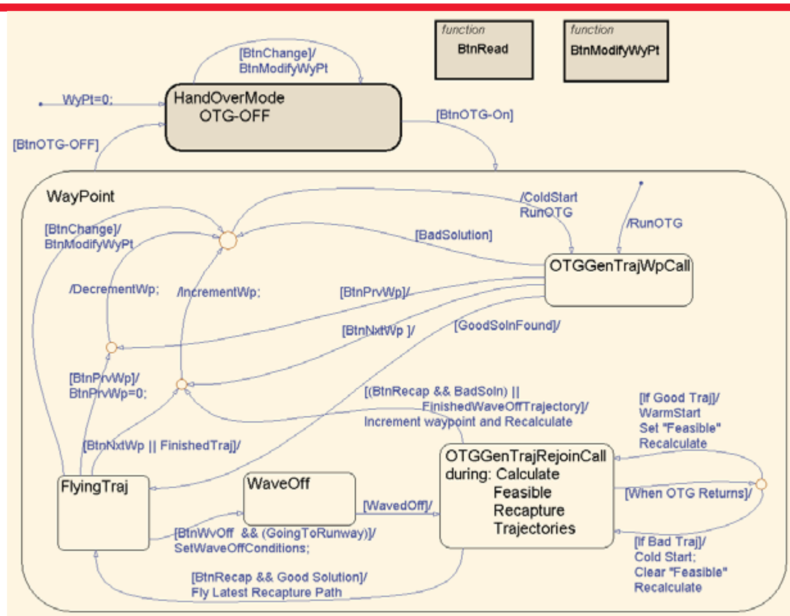
History of Automatic Code Generation at Northrop Grumman

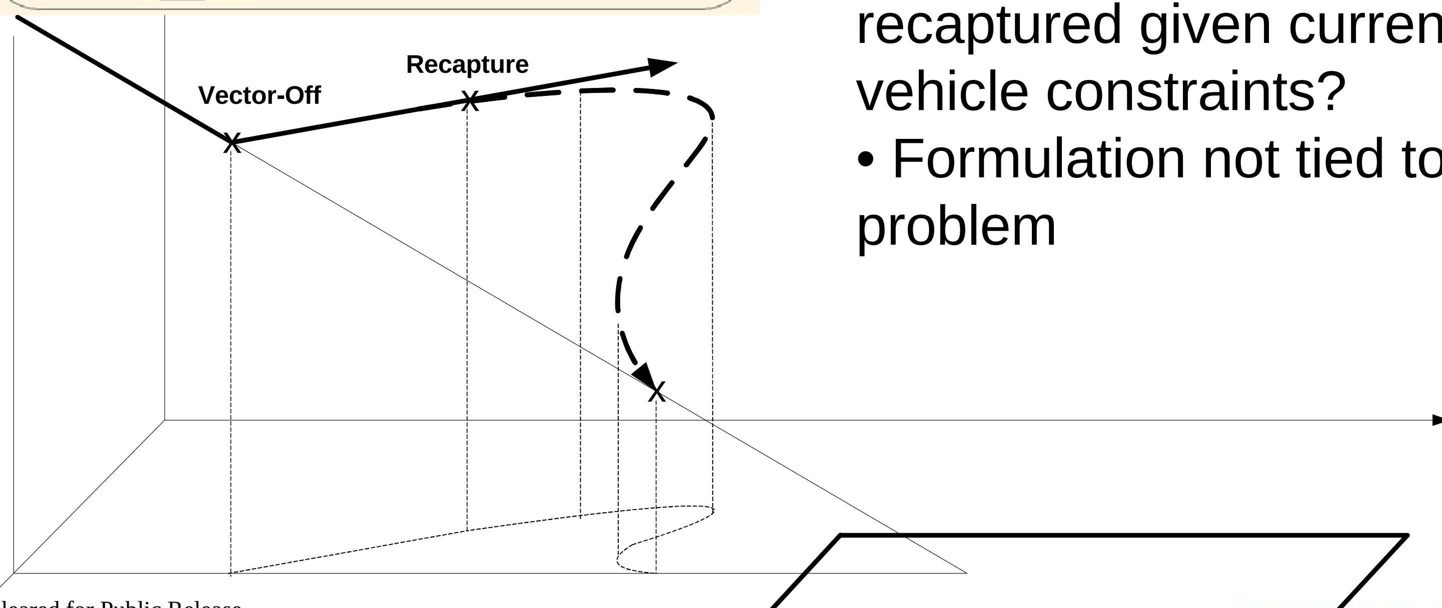
- **2005**
  - SeFAR
  - AEI
- **2006**
  - Expanded Use of Stateflow (not done by GNC but SW people)
    - UUV Logic
    - UAV Logic managing Mode control via Air Vehicle Manager
- **2007**
  - Using Stateflow for CCDL (cross channel data link) voting
  - Demonstrations of UAV and UUV using automatically generated code from Real-Time Workshop Embedded Coder

Cleared for Public Release
Control Number 07-057 dated 6/04/07
AFRL-WS 07-1235 dated 5/28/07

NORTHROP GRUMMAN

SEC NGC Exp #2 Flight Test
Basic Maneuver





- All guidance performed by OTG at fly time
- Vector-off occurs due to runway-not-ready condition (user set)
- Shortly thereafter runway-clear signal given. (User set)
- Can runway interface point be recaptured given current state and vehicle constraints?
- Formulation not tied to specific problem

Cleared for Public Release
Control Number 07-057 dated 6/04/07
AFRL-WS 07-1235 dated 5/28/07

NORTHROP GRUMMAN

Embedded Coder structure for SEC

- **Effort utilized first flight of R14sp3**
  - Driven by need/desire for EML
- **The SEC code structure contained two subsystems which were each coded "build Subsystem" and compiled into the existing (Boeing) environment for the flight test.**
- **One of the subsystems ran in a real time section of the operating system.**
- **The second subsystem was implemented as an 'anytime' task**
  - Contained optimization codes and interfaces written in both C and FORTRAN
  - Some of the called functions were precompiled and linked in

NORTHROP GRUMMAN

Embedded Coder structure for SEC

- **The software solution used to complete this task required an array of tools**
  - Precompiled FORTRAN, C code, precompiled C libraries, Simulink (Block build up, S-Functions, Stateflow)
  - Environments, were varied, Windows, Windows soft real-time, Real-Time Linux.
- **The Test Process for SEC**
  - Almost all design, debugging, validation and testing was done in Simulink with the simulation running much faster than real-time on Windows
    - Coverage analysis utilized to catch Stateflow design bug
  - Code was generated and a brief check was made in a soft real-time windows environment
  - It was then compiled into the Linux real-time HWIL environment used in the flight test
- **The process from code generation to Linux real-time HWIL test was about 5 min. of user time (total time was larger due to environment size)**
- **No hand modifications to code or interface made once ICD established**

Cleared for Public Release
Control Number 07-057 dated 6/04/07
AFRL-WS 07-1235 dated 5/28/07

NORTHROP GRUMMAN

Embedded Coder structure for AEI

- **AEI**
  - Active control of a wind tunnel model
  - Two DSpace control boxes
    - One running at 1000Hz, the other at 200Hz
  - Entire system, two flight computers, and model were built up in Simulink for faster than real time design and analysis
  - The subsystems representing the flight code were libraries
    - The master hardware interface diagram was kept by NASA and would call the same libraries as the simulation
    - The entire model built using Real-Time Workshop Embedded Coder
  - The subsystems contained many Simulink architectures
    - Stateflow was used to manage the many different modes and operation of the code
    - Heavy use of enabled subsystems to reduce computational requirements
  - Iteration time from Simulink modification to the new system loaded in the dSpace controller and running on the hardware was < 10min
  - Allowed for rapid solution iteration and testing
  - Based on the success of this process, the next entry will follow a similar process

Evolution of SAA Simulation Development

PRISM Simulation (Starting Point)

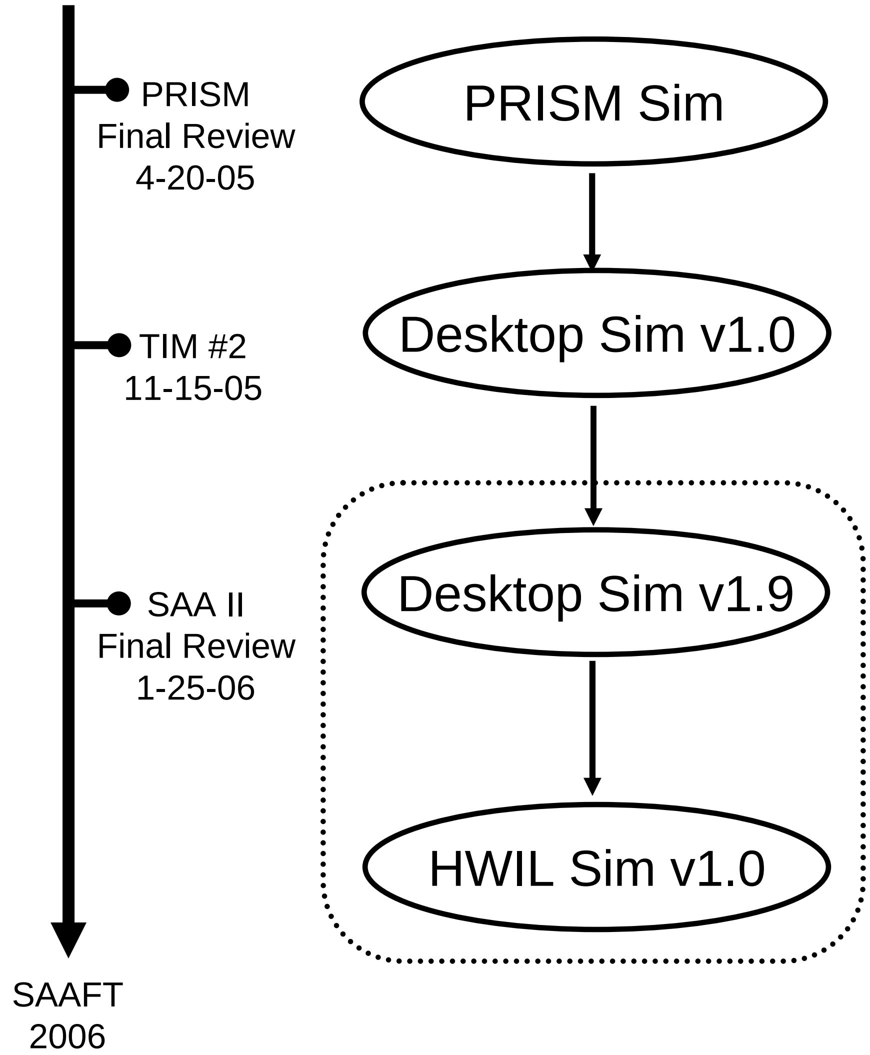
- Passive Ranging EKF
- Auto-ACAS Collision Avoidance

SAA II Desktop Simulation

- Fully Integrated Closed-loop Simulation
- Hosted In Simulink Environment With HWIL Support

SAA II HWIL Simulation (Current)

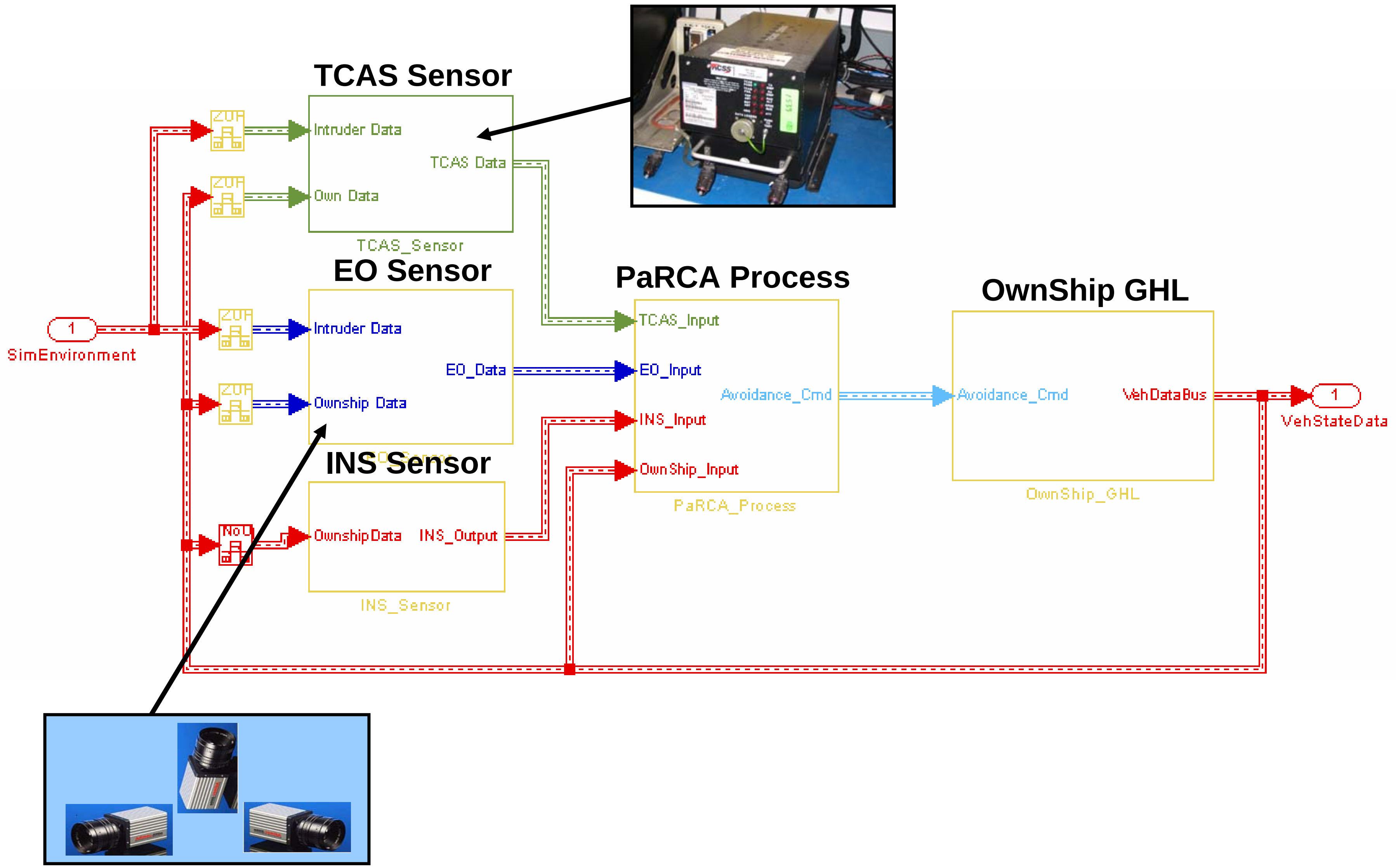
- Real-time Closed-loop Simulation
- X-Plane Display Environment
- Multiple Processors
- IEEE 1394 Data Communication



Cleared for Public Release
Control Number 07-057 dated 6/04/07
AFRL-WS 07-1235 dated 5/28/07

NORTHROP GRUMMAN

Simulation Architecture Facilitates Transition from Desktop to HWIL



Cleared for Public Release
Control Number 07-057 dated 6/04/07
AFRL-WS 07-1235 dated 5/28/07

Desktop Simulation Bus Objects

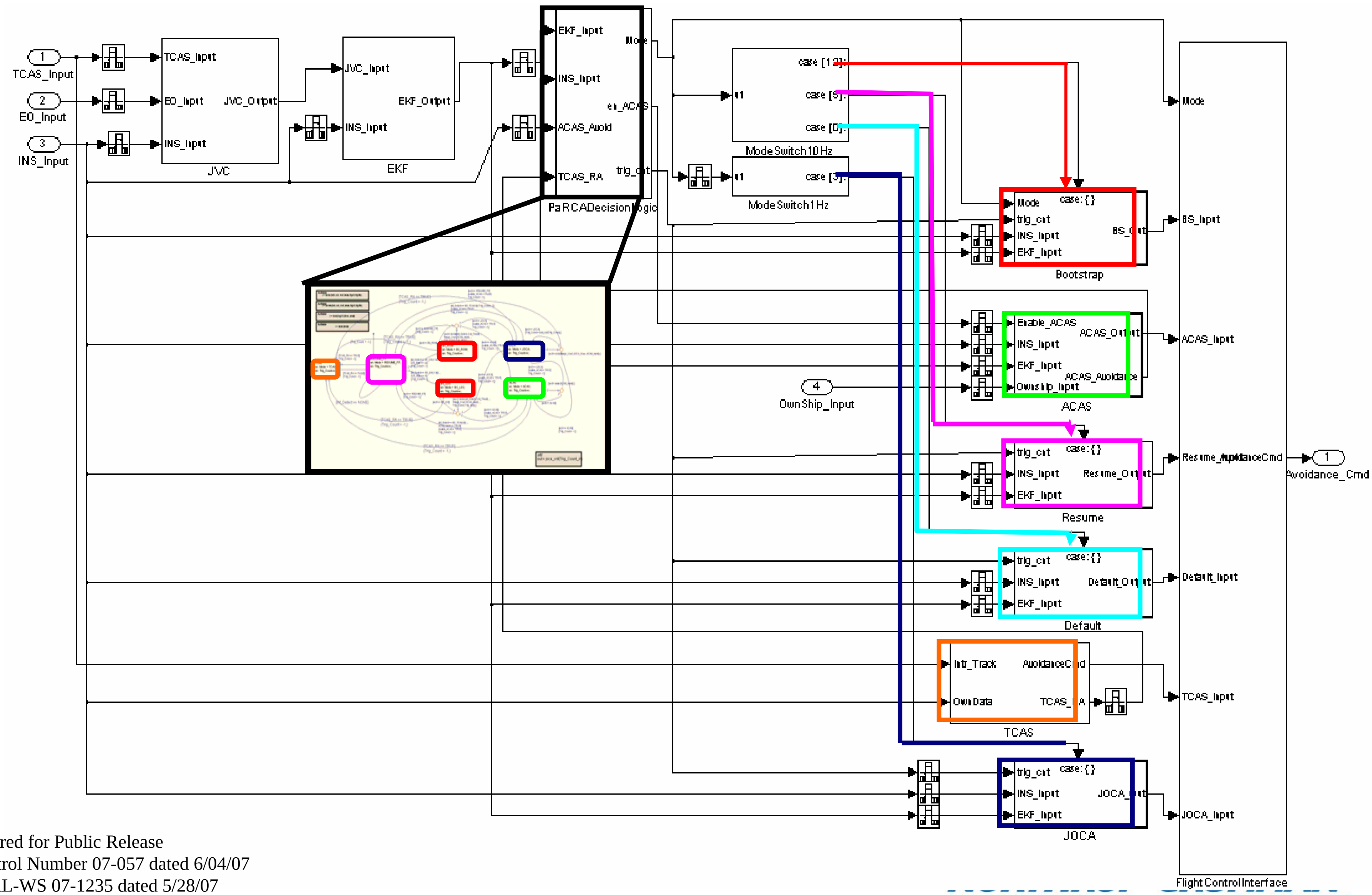
Data Structure that Specifies Key Attributes of Block Outputs (data type, tunability, value range, etc…)

- Benefits
  - Helps Encapsulate Inputs and Outputs of Each Module
  - Platform Independent Reusable Data Structure
  - Seamless Conversion From Simulation Data Structure to IEEE 1394 Data Packet
  - Implements ICD between Modules
- Overhead
  - Require Detailed Definition and Attentive Maintenance in Desktop Simulation

Cleared for Public Release
Control Number 07-057 dated 6/04/07
AFRL-WS 07-1235 dated 5/28/07

NORTHROP GRUMMAN

PaRCA Decision Logic



Cleared for Public Release
Control Number 07-057 dated 6/04/07
AFRL-WS 07-1235 dated 5/28/07

SAA Desktop Simulation Module Updates

- **Sensor**
  - TCAS **1 Hz** **SFun**
  - EO **25 Hz** **EML**
  - INS **100 Hz** **Sys**
- **Vehicle**
  - Ownship **100 Hz** **Sys**
  - Intruder **100 Hz** **Sys**
- **PaRCA Decision Process**
  - EKF **25 Hz** **EML**
  - PaR **10 Hz** **EML**
  - JOCA **1 Hz** **SFun/EML**
  - ACAS **10 Hz** **SFun**
  - Data Association **25 Hz** **SFun**
  - Decision Logic **10 Hz** **EML**
  - Track Manager **25 Hz** **EML**
  - Flight Control IO **10 Hz** **Sys**

Sampling Rate
Module Type

Sys	Simulink
EML	Embedded MATLAB
SFun	S-Function
SFlow	Stateflow

Cleared for Public Release
Control Number 07-057 dated 6/04/07
AFRL-WS 07-1235 dated 5/28/07

NORTHROP GRUMMAN

Desktop To HWIL Build Up Process

- Each module in desktop simulation is partitioned into individual simulink models

- Real-time Software was generated for each module
  - C/C++ code was generated using MATLAB/Simulink Real-Time Workshop embedded coder for specific target:
    - RTOS VxWorks-based VMS
    - Linux-based Ownship model

- Custom wrapper function developed for target code
  - Scheduling, data communication

NORTHROP GRUMMAN

Conclusions

- **Lessons learned**
  - MATLAB/Simulink/Stateflow design process is a powerful toolchain which facilitates real-time HWIL and flight code generation
    - Desktop directly to flight code on flight hardware
  - Bus objects drastically reduce HWIL transition and data communication
  - Coverage analysis facilitates debugging and verification
  - Stateflow extensively used for mode control

Cleared for Public Release
Control Number 07-057 dated 6/04/07
AFRL-WS 07-1235 dated 5/28/07

NORTHROP GRUMMAN

Future Issues

- **Large scale modeling**
  - More than 30,000 blocks
  - Aerospace model more complicated than models typically used in other industries
  - More extensive use of trim and linearization features
  - Model reference
  - Version control

Cleared for Public Release
Control Number 07-057 dated 6/04/07
AFRL-WS 07-1235 dated 5/28/07

NORTHROP GRUMMAN