

2003 FAA National Software Conference SPIDER Research and DO-254 Experiences



Langley Research Center



The SPIDER Project and DO-254 Experiences

Paul S. Miner

FAA National Software Conference
September 17, 2003



Langley Research Center



Project Goals

- FAA goals:
 - Develop case study application of DO-254
 - Provide feedback on problem areas
 - Provide material for DO-254 training
- NASA goals:
 - Demonstrate application of formal methods in certification context
 - Develop research platform for exploring recovery from correlated transient faults

September 17, 2003 SPIDER Lessons Learned 2

2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences



Langley Research Center



Team Members and Responsibilities

- NASA
 - Paul Miner, project lead, formal models
 - Mahyar Malekpour, design engineer
 - Wilfredo Torres, design engineer
 - Jeff Maddalon, formal models
- NIA
 - Alfons Geser, formal models

September 17, 2003 SPIDER Lessons Learned 3



Langley Research Center



Project Overview

- Design part of a new fault-tolerant Integrated Modular Avionics (IMA) architecture
 - Fault-tolerance is inherently complex
 - System description is compact
- DO-254 case study applied to the Reliable Optical Bus (ROBUS) of the Scalable Processor-Independent Design for EME Resilience (SPIDER).

September 17, 2003 SPIDER Lessons Learned 4

2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences



Langley Research Center



What is SPIDER?

- A family of fault-tolerant IMA architectures
- Inspired by several earlier designs
 - Main concept inspired by Palumbo's Fault-tolerant processing system (U.S. Patent 5,533,188)
 - Developed as part of Fly-By-Light/Power-By-Wire project
 - Other ideas from Draper's FTTP, FTP, and FTMP; Allied-Signal's MAFT; SRI's SIFT; Kopetz's TTA; Honeywell's SAFEbus; . . .

September 17, 2003 SPIDER Lessons Learned 5



Langley Research Center



SPIDER Architecture

- N general purpose Processing Elements (PEs) logically connected via a Reliable Optical BUS (ROBUS)
- The ROBUS is an ultra-reliable unit providing basic fault-tolerant communication services
- ROBUS contains no software

September 17, 2003 SPIDER Lessons Learned 6

2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences



Langley Research Center



DO-254 Issues for Level A

- Specific guidance in Appendix B
 - Functional Failure Path Assessment (FFPA)
 - Design Assurance Methods
 - Architectural Mitigation
 - Service History
 - Advanced Analysis Techniques
 - Elemental Analysis
 - Safety-Specific Analysis
 - Formal Methods

September 17, 2003 SPIDER Lessons Learned 7



Langley Research Center



ROBUS FFPA

- Goal of ROBUS is to provide reliable communication between various devices attached to the bus
- These devices may have differing design assurance levels
- Must ensure proper communication even when some devices are behaving arbitrarily
 - That is, any function computed on SPIDER could be compromised, if some attached node could disrupt ROBUS communication

September 17, 2003 SPIDER Lessons Learned 8

2003 FAA National Software Conference SPIDER Research and DO-254 Experiences

NASA Langley Research Center **SPIDER**

Failures contained by ROBUS

- Arbitrary failure in any attached Processing Element
 - Hardware or Software
 - Converts asymmetric error manifestations to symmetric
- Must also operate correctly if a bounded number of internal hardware devices fail
- Cannot tolerate design error within ROBUS

September 17, 2003 SPIDER Lessons Learned 9

NASA Langley Research Center **SPIDER**

Logical view of ROBUS (Sample Configuration)

September 17, 2003 SPIDER Lessons Learned 10

2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences



Langley Research Center



Logical View of ROBUS

- ROBUS operates as a time-division multiple access broadcast bus
- ROBUS strictly enforces write access
 - no babbling idiots (prevented by ROBUS topology)
- Processing nodes may be grouped to provide differing degrees of fault-tolerance
 - PEs cannot fail asymmetrically (prevented by ROBUS topology)

September 17, 2003 SPIDER Lessons Learned 11



Langley Research Center



ROBUS Characteristics

- All good nodes agree on communication schedule
 - Currently bus access schedule statically determined
 - similar to SAFEbus, TTA
 - Architecture can support on-the-fly schedule updates
 - similar to FTTP
 - Preliminary capability will be in our next prototype
- Some fault-tolerance functions must be provided by processing elements
 - Similar to FT-Layer in TTA
- Processing Elements need not be uniform
 - support for dissimilar architectures

September 17, 2003 SPIDER Lessons Learned 12

2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences



Langley Research Center



ROBUS Requirements

- All fault-free nodes observe the *exact* same sequence of messages
- ROBUS provides a reliable time source (RTS)
 - The PEs are synchronized relative to this RTS
- ROBUS provides correct and consistent ROBUS diagnostic information to all fault-free nodes
- For 10 hour mission, $P(\text{ROBUS Failure}) < 10^{-10}$

September 17, 2003 SPIDER Lessons Learned 13



Langley Research Center



Appendix B Design Assurance

- Architectural Mitigation
- **Service History**
- Advanced Analysis Techniques
 - Elemental Analysis
 - **Safety-Specific Analysis**
 - Formal Methods

September 17, 2003 SPIDER Lessons Learned 14

2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences



Langley Research Center



Not relevant to this design

- Service History - New design, so N/A
- Safety-specific analysis - This design is independent of aircraft function, so N/A

September 17, 2003 SPIDER Lessons Learned 15



Langley Research Center



Architectural Mitigation

- The ROBUS is designed to mitigate effects of various faults
 - The topology and protocols also mitigate random hardware failures within the ROBUS
- This case study illustrates some steps that may be used to justify an architectural mitigation strategy

September 17, 2003 SPIDER Lessons Learned 16

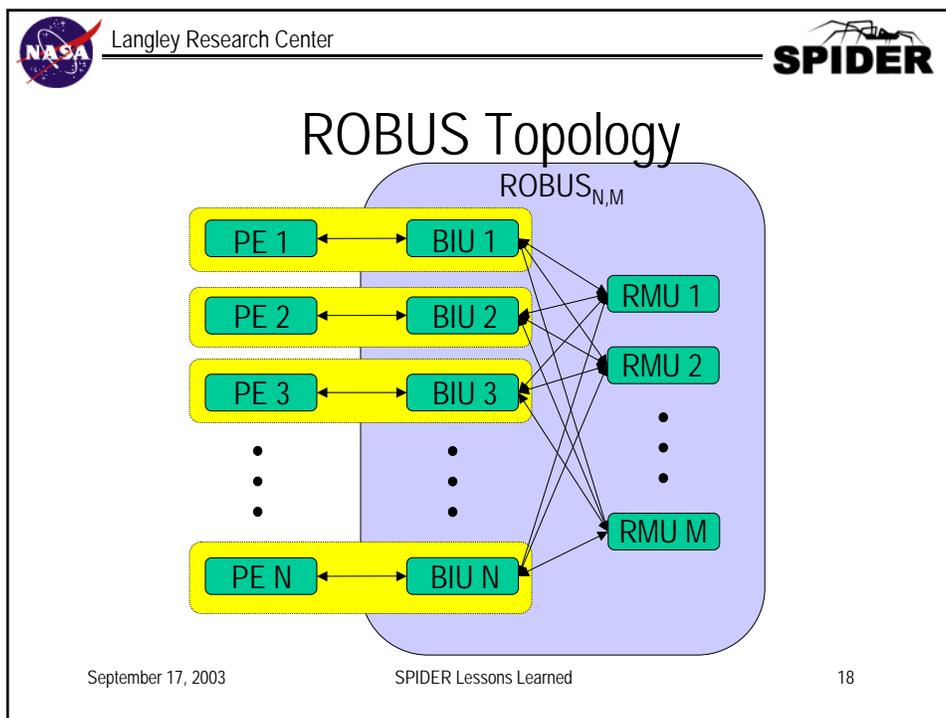
2003 FAA National Software Conference SPIDER Research and DO-254 Experiences

NASA Langley Research Center **SPIDER**

Aspects of Architectural Mitigation

- Fault model
 - What kinds of failures is the architectural mitigation system designed to withstand?
 - Design Flaws?
 - Random HW failures?
 - Is there a reasonable classification of fault effects?
 - Does the fault model include a catch-all failure mode?
 - I.e. something we haven't thought of?

September 17, 2003 SPIDER Lessons Learned 17



2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences



Langley Research Center



Physical Segregation

- ROBUS decomposed into physically isolated Fault Containment Regions (FCR)
 - Two main design elements
 - Bus Interface Unit (BIU)
 - Redundancy Management Unit (RMU)
 - Processing elements may form separate FCRs
- FCRs fail independently

September 17, 2003 SPIDER Lessons Learned 19



Langley Research Center



Design Assurance Strategy

- Fault-tolerance protocols and reliability models use the same fault classifications
- Reliability analysis using SURE (Butler)
 - Calculates $P(\text{enough good hardware})$
- Formal proof of fault-tolerance protocols using PVS (SRI)
 - enough good hardware \Rightarrow correct operation

September 17, 2003 SPIDER Lessons Learned 20

2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences



Langley Research Center



Fault Assumptions

- The ROBUS is designed to operate correctly, even if some RMUs/BIUs have suffered a *physical* fault
- The failure status of a BIU or RMU is subdivided into four cases
 - Good (or fault-free)
 - Benign faulty (Obviously bad to all good)
 - Symmetric Faulty (Same manifestation to all good)
 - Asymmetric Faulty (Byzantine)
- Models use these classifications
- This is a global classification

September 17, 2003 SPIDER Lessons Learned 21



Langley Research Center



Evolving Fault Assumptions

- For our first prototype, we used a simplifying assumption that all faults were permanent
 - However, we tried to leave placeholders for transient fault recovery
- For our current design, we added a requirement that the ROBUS can recover from a bounded number of transient faults
 - This had a much larger impact on our design assurance than we had anticipated
 - Simple modifications to fault assumptions and subsequent revision to protocols introduced subtle bugs

September 17, 2003 SPIDER Lessons Learned 22

2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences

Langley Research Center

Eligible Voters Property (EVP)

- Hybrid fault model implies ability to locally detect and diagnose all benign faulty nodes
- Each node maintains a local determination of which nodes are *trusted* (E_i , Eligible)
 - All good nodes are *trusted* by all good observers
 - No benign faulty node is *trusted* by any good observer
 - If a symmetric faulty node is *trusted* by any good observer, then it is *trusted* by all good observers
 - Asymmetric faulty nodes may be *trusted* by some, but not necessarily all, good observers

September 17, 2003SPIDER Lessons Learned23

Langley Research Center

Dynamic Maximum Fault Assumption

1. $|GB \cap E_i| > |AB \cap E_i| + |SB \cap E_i|$, for all RMU i
2. $|GR \cap E_i| > |AR \cap E_i| + |SR \cap E_i|$, for all BIU i
3. $|AR \cap E_i| = 0$, for all BIU i or $|AB \cap E_i| = 0$, for all RMU i

September 17, 2003SPIDER Lessons Learned24

2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences



Langley Research Center



Elemental Analysis

- DO-254 analog of structural coverage
- Selected TransEDA's VN-cover tool for coverage analysis
 - Supports several different types of coverage
 - Control logic tests
 - statement, branch, condition, path
 - Data tests
 - trigger, signal trace, toggle

September 17, 2003 SPIDER Lessons Learned 25



Langley Research Center



Focused Expression Coverage

- VN-cover's default condition coverage for VHDL code is Focused Expression Coverage (FEC)
- We have determined that FEC is the same as Masking MC/DC
 - By examining TransEDA documentation
 - By analyzing results for simple designs

September 17, 2003 SPIDER Lessons Learned 26

2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences



Langley Research Center



Assessment of VN-cover

- DO-254 does not require detailed assessment of tools supporting elemental analysis
 - *“If the tool is ... used to assess the completion of verification testing, such as in elemental analysis, no further assessment is necessary”* p. 76, item 4.

September 17, 2003 SPIDER Lessons Learned 27



Langley Research Center



Elemental Analysis Results

- Preliminary investigations on portions of our initial design did not produce any surprises
- No results yet for current design
 - Significant redesign
 - Still incomplete

September 17, 2003 SPIDER Lessons Learned 28

2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences



Langley Research Center



Formal Methods

- This is dominant design assurance strategy for this project
- Emphasis on early life-cycle verification
- Formal proof of key fault-tolerance protocols
 - Interactive Consistency
 - Distributed Diagnosis
 - Clock Synchronization

September 17, 2003 SPIDER Lessons Learned 29



Langley Research Center



Strength of Formal Verification

- Proofs equivalent to testing the protocols
 - for all possible ROBUS configurations (i.e. for all N, M)
 - for all possible combinations of faults that satisfy the maximum fault assumption for each possible ROBUS configuration
 - for all possible message values
- The PVS proofs provides verification coverage equivalent to an infinite number of test cases.
 - Provided that the PVS model of the protocols is faithful to the VHDL model

September 17, 2003 SPIDER Lessons Learned 30

2003 FAA National Software Conference SPIDER Research and DO-254 Experiences

Langley Research Center

Modeling Issues

- Are the models meaningful?
 - Are abstractions valid?
 - e.g. synchronous composition, functional abstraction
 - Are assumptions satisfiable?
 - Is there a typical case?
 - Are assumptions true for initial conditions?
 - Are assumptions preserved through execution of protocol?

September 17, 2003SPIDER Lessons Learned31

Langley Research Center

More Modeling Issues

- How is the formal model related to the modeled artifact?
 - Compilation of VHDL to model?
 - Compilation of model to VHDL?
 - Manual comparison?

September 17, 2003SPIDER Lessons Learned32

2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences



Langley Research Center



Formal Proof Issues

- Have you proven the claim you intended to prove?
 - Sanity checks:
 - For each hypothesis, demonstrate why proof fails when hypothesis removed (may be an informal argument)
 - Confirm that you haven't assumed the conclusion
 - Confirm that models of system components only have access to same set of data as the modeled component

September 17, 2003 SPIDER Lessons Learned 33



Langley Research Center



Added Benefits of Formal Methods

- Formal Models provide detailed understanding of why protocols work
- This sometimes results in ability to recognize improvements to protocols
 - Verification of original diagnosis protocol provided insights that allowed us to provide same guarantees with much simpler protocol
 - Simpler design and simpler proofs

September 17, 2003 SPIDER Lessons Learned 34

2003 FAA National Software Conference

SPIDER Research and DO-254 Experiences



Langley Research Center



Summary

- ROBUS development exploring Appendix B of DO-254
- Some insight on architectural mitigation and formal methods
- Elemental analysis still pending

September 17, 2003 SPIDER Lessons Learned 35