

# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications



*Study of COTS RTOSs in Aviation Applications*

Jim Krodel  
United Technologies Research Center  
East Hartford, CT, USA  
16-May-02



1



Gee...

I wonder how that COTS RTOS I coded is doing?



# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications



### *Outline*

- Background – Regulatory & other issues
- Partitioning
- RTOS failures wrt potential safety impact
- Robustness testing benchmarking techniques
- Fault Containment Techniques
- Wrappers
- RTOS vulnerability analysis & Robustness test plan



3



### *Background*

- COTS – The Hope for Reduced Development Costs
- Applicant's are thus applying pressures on JAA/FAA to approve systems with COTS Software (SW)
- All Airborne Software (COTS or not) Must Still Follow DO-178B



4

# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications



### *Regulatory Issues*



- COTS SW Vendor Market
  - Typically not aerospace – lacks Do-178B Rigor
- Regulatory Assessment of COTS SW
  - Pedigree difficult to assess
- Alternate methods
  - Reverse Engineering, Wrappers, Service History, etc. being offered for DO-178B compliance
- Competitive and Management Concerns
  - Access to records, etc.

5



### *Other Issues Concerning Usage of COTS Software*



- Vendor & Applicant Business Relationship
- Problem Reports
- Unused / Unintended Functions
- Previous COTS SW Operational Environment
- Version Control
- New Releases

6

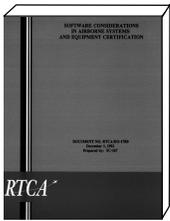
# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications



### *COTS RTOS*

- Growing in Airborne Applications
  - Cost, Schedule Reductions, etc.
  - RTOS Services in the Aircraft Domain are Increasing
    - COTS RTOS expertise may be a better suited developer
    - Risk is lack of Vendor DO-178B knowledge
  - Several RTOS Vendors are making "DO-178B Ready RTOS' Available"



7



### *RTOS Partitioning Considerations*

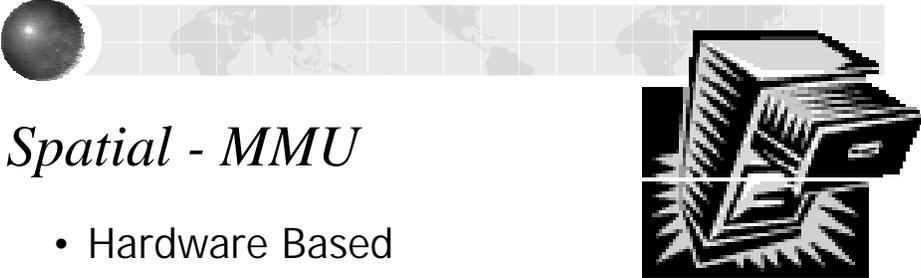
- Spatial
  - Prevent a function in one partition from corrupting the data space (i.e., memory) of a function in another partition
    - Memory Management Unit
    - Software Partition Fault Isolation
- Temporal
  - Ensures that each function has sufficient processing time to complete its operation
    - Static Scheduling
    - Dynamic Scheduling



8

# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications



*Spatial - MMU*

- Hardware Based
- MMU's can be contained in uP
- Manages Memory Partitions
- Performs HW Address checking
- Very complex and have raised certification concerns

9



*Spatial - Software Fault Isolation*

- Logically checks memory access
  - Direct addressing can be static checked
  - Indirect addressing is dynamically checked
    - Code added to check address register at runtime
  - Imposes some code overhead penalty
  - Additional analysis and certification effort is needed for this technique

10

# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications



### *Temporal – Static Scheduling*

- Tasks are executed under a fixed cyclic schedule
- Not flexible, but it is deterministic
- *Sequenced decided at design time*
- *Rate Monotonic and Dead-line Monotonic Scheduling – proven determinism*
- *Time sliced*



11



### *Temporal – Dynamic Scheduling*

- No predefined schedule
  - Task priorities assigned at design time
  - Task scheduled at runtime
- Task Monitors can be used to accommodate task overruns
- *Round robin*
- *Earliest deadline first*
  - *Whoever is active, one with earliest dead line runs*
- *Highest priority run*
  - *If not period they will all by dynamic.*



12

# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications



### *RTOS Specific Failures with Potential Safety Impact*

- Data consistency
- Inclusion of deactivated code or dead code
- Tasking
- Scheduling
- Memory and I/O device access
- Queuing
- Interrupts and Exceptions



### *Data consistency*

- Data corruption or loss within the RTOS by the RTOS itself
- Input data corruption or loss by the RTOS
- Erroneous data or results caused by incorrect calculations or operations by the RTOS
- Calculations performed by the math library functions may return unpredictable small numbers if the values passed as parameters are abnormal.



# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications



### *Inclusion of deactivated code or dead code*

- Deactivated code
  - Unused functions loaded by RTOS by design
- Generation of dead code
- Linker/Loader can introduce deactivated code
- Unintended activation may have unknown effects



15



### *Tasking*

- Major task terminates or is deleted
- Kernel's storage area overflow
- Task stack size is exceeded



16

# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications



### *Scheduling*



- Corrupted task control blocks (TCB)
- Excessive task blocking through priority inversion
- Deadlock
- Tasks spawns additional tasks that starve CPU resources
- Corruption in task priority assignment
- Service calls with unbounded execution times

17



### *Memory and I/O device access*



- Fragmentation of heap memory space
- Incorrect pointer referencing/de-referencing
- Data overwrite
- Compromised cache coherency
- Memory may be locked or unavailable
- Unauthorized access to critical system devices

18

# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications



### *Queuing*

- Task queue overflow
- Message queue overflow
- Kernel work queue overflow



19



### *Interrupts and Exceptions*

- Interrupts-atomic ops
- No interrupt handler
- No exception handler
- Signal is raised without a corresponding handler
- Improper protection of supervisor task



20

# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications



### *Robustness testing benchmarking techniques*

- Crashme
  - Random data in memory
  - Tasks spawned to operate on that data
- Fuzz
  - Random data injection
  - Open source O/S more robust than COTS
- Ballista
  - Exception handling via API

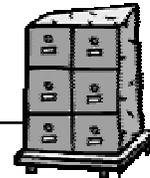


21



### *Fault-containment*

- When fault cannot be removed
  - Identify
    - Isolate / Contain
  - Techniques
    - Partitioning
    - Timing monitors
- Techniques
    - Data Validation Checks
    - Redundant Functions
    - BIT
    - Data queue monitors
    - Range Checking
    - Wrappers



22

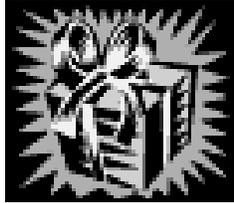
# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications



### *Wrappers*

- Intercept problematic API calls
- Intercept for proper API calls
  - Perform logical checks on proper call and parameters
- Implement additional features
- Data wrapper - CRC
- I/O Wrapper
- Substitute Safe State or defaults
- Kernel based wrappers
  - May already exist in COTS RTOS



23



### *Wrapper Woes*

- Global Data
- COTS RTOS visibility minimal
- RTOS – highly integrated in system
  - I/O, data communications, tasking, partitioning, etc.
- Some problems not 'wrapable'
  - Data inconsistencies
- Access to RTOS source code may be required
- Can be significantly complex software



24

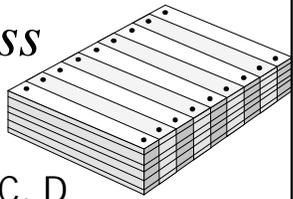
# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications



### *A plan for RTOS Robustness*

- Robustness testing required
  - DO-178B, 6.4.2.2, levels A, B, C, D
- Select, Acquire, Integrate the COTS RTOS
- COTS RTOS Software Vulnerability Analysis for safety
- COTS RTOS Stress Test Plan



25



### *Summary*

- RTOS service – heart of the system
- COTS RTOS may not have DO-178B rigor
- DO-178B requires robustness testing
- Software vulnerability analysis for safety
- RTOS robustness test plan

26

# FAA National Software Conference, May 2002

## COTS RTOSs in Aviation Applications

