Mixed Criticality Architecture

Mixed Criticality: In order to optimize weight/volume for UAV systems, it is tempting to mix mission and flight critical functionality within the same computation platform. For manned systems, this would warrant that all of the functionality be tested at the flight critical-level of certification. The trend for UAVs is to rely on time-space partitioning (e.g. ARINC 653) to effectively separate the two functionalities and assume non-interference. While this may work from a performance viewpoint, the failure mechanisms of the partitioning must be considered and understood when determining the probability of loss of control (PLOC) and fault tolerance of the system. As with other flight critical functionality, the failure mechanisms have to be validated then by testing. As the two criticalities are sharing resources, the impact of hardware failure must be predictable and testable. This means that the hardware architecture of the computation platform is a factor, which will drive up testing cost due to the countless variations.

MCAR Program Objective and Approach: is to create a compose-able architecture where safety and security are assumed system characteristics. Most safety-critical systems have the same attributes: most are time-critical, most are fault tolerant/redundant, and most provide date/system integrity. Some of these attributes are shared with security. These attributes can be built into an “open” architecture that can be utilized across various applications. This architecture could also include a “cyber-segregation”, where non-critical and safety-critical systems could safety and efficiently utilize the same computational resources. A layered software approach is being considered, consisting of a high confidence Real-Time Operating System (HC-RTOS) and Flight Management System specific Middleware. The HC-RTOS provides the foundation for the architecture, providing the low-level fault tolerance and separation support for safety and security applications. The HC-RTOS would also be required to perform the priority-based scheduling that would assure the execution of safety-critical functionality and would perform exception handling should the non-critical functions over-utilize the resources. The domain-specific Middleware Layer would provide the higher-level execution scheduling and services for distributed processing and communication with other entities. The application software would focus on the specifics of the design and interface to the middleware.

The AFRL-MCAR Meeting is open to the CPS workshop audience on November 18th 2008, morning. If you are interested in attending, please fill the mandatory registration form.

For further information on MCAR, contact:
Dr. Russ Urzi
Program Manager,
AFRL/RBCC, Control Systems Dev'mt & Appl's Branch,
Control Sciences Division,
Air Vehicles Directorate.
Phone: (937) 255-8294; Fax: (937) 255-8297; E-mail: Russell.Urzi@wpafb.af.mil