AIR - Overview
AIR - ORIGINS

 Origins

 AIR was developed during two ESA studies:
 - AIR and AIR II
 - By GMV, University of Lisbon, THALES-Alenia Space

 Components

 - Partition Management Kernel (PMK)
 - Support libraries, drivers, etc.
 - RTEMS Personality + IMA-SP Abstraction Layer
 - Configuration and Compilation tool chain
 - Analysis tools

 Targets

 - Available: SPARC-LEON
 - In progress: IA-32, AMD-64, PPC, ARM
AIR ARCHITECTURE - PERSONALITIES

- Different Personalities:
  - RTEMS
  - RTEMS + APEX
  - APEX
  - OS-less (static scheduling)
- In future:
  - Linux
“System” Partitions:
- Higher Privileges
- Implement I/O
AIR ARCHITECTURE - MICROKERNEL

- Microkernel:
  - No “kernel threads”
  - Instead “Co-Partitions” that are scheduled with “client partitions”
  - Hardware Abstraction Layer is RTEMS
Co-Partitions implement “kernel-tasks”, e.g.
- Drivers
- Message Transportation
- Debugging

Co-Partitions share execution windows with their “clients” up to a pre-defined percentage (“sharing quota”), iff:
- Critical tasks have terminated (“sharing barrier”)

Objective:
- Flexible time allocation
- “Fair” time allocation
- Predictable time allocation
AIR ARCHITECTURE – MEMORY LAYOUT

- **Partition Layout**
  - Each partition has its own page directory
  - Changing the pointer (= changing address space) is part of the partition switch
  - Flexible Sharing of memory pages between partitions

- Use of “Shared Libraries” to optimise memory use:
  - POS and APEX code shared
  - Use of data backups to optimise partition reset
Inter-Partition Communication

- Queuing and Sampling Ports
- Standard communication

Channels:
- Messages are usually copied (by a co-partition) from one application to another
- Fast channels: Based on Shared Memory

Shared Memory

- Memory Pages shared between partitions
- Used as a means (e.g. fast channels)
- Memory Block API available in ARINC 653 APEX
HM

- Handle Exceptions (Hardware or Software Traps)
- Handle on one of three levels:
  - Process (user defined)
  - Partition (pre-defined actions)
  - Module (pre-defined actions)

Actions:
- Shutdown
- Restart
- WARM_START
- COLD_START
- IGNORE
AIR ARCHITECTURE – TOOL CHAIN

- PMK is “stupid”
- Code base is small and simple
- The intelligence is in the configuration tool
- Tool chain is hence complex

Platform Independent Model Mapping A653 Config

Pocket Model
POS

Manual Configuration Tool

DIANA Experiment
Standard + AIR Extensions
Includes Validation

DIANA Experiment

User application Python GNU Standard Definition

Tool
usr_partitions.c
usr_schedule.c
linkcmds
hm_tables.bin
make
ld
gcc
*.c
*.c
*.c
apex.a
airimage.exe

AIR User application Python GNU Standard...
VALIDATION APPROACH

- Validation
  - Based on formal specification
    - Verify consistency and completeness of
      - Scheduler
      - Configuration
  - But also traditional testing
    - Requirements coverage
    - Unit testing for kernel
    - Robustness testing
    - Benchmarking
- Demonstration
  - Real data profiles shall be used to define test-bench
  - Demonstration will focus on I/O
Thank You

Tobias Schoofs
tobias.schoofs@gmv.com
Av. D. João II, Lote 1.17.02
Torre Fernão Magalhães, 7º
1998-025 Lisboa
Portugal
Tel. +351 21 382 93 66
Fax +351 21 386 64 93

www.gmv.com