

A Survey of Time and Space Partitioning for Space Avionics

Presentation at DASIA 2018

Jan Brederke




HSB

Hochschule Bremen
City University of Applied Sciences

31st May 2018

Legend for the Slide Handout

This handout comprises all slides shown. Additionally, it comprises notes for oral explanations. The notes are marked with “” in the head line (like this page).

The Basic Idea of TSP

Motivation

Time and Space Partitioning (TSP) – Why?

several computing tasks
with mixed dependability requirements
on a single computer
in order to save weight

Overview

A Survey of Time and Space Partitioning for Space Avionics

- 1 Systems with Mixed Dependability
- 2 Integrated Modular Avionics (IMA) for Aircraft
- 3 Adaption of IMA for Space Avionics
- 4 Some Research Challenges

The Notion of Dependability

Systems with Mixed Dependability

Dependability (Avižienis et al. 2004)

“the ability of a system to avoid service failures that are more frequent and more severe than is acceptable”

dependability: **must be validated**

The Problem with Mixed Dependability

Systems with Mixed Dependability

several computing tasks on a single computer

- with mixed dependability requirements
- most critical task:
determines criticality of *all* software on this computer
example: danger of writing into memory of another task

consequence

- for all tasks: degree of effort for validation of dependability
= degree of the most critical task
- high costs for development and maintenance,
if many tasks on a computer
which all might impair each other

Solutions

Systems with Mixed Dependability

	separation kernel	virtualization
idea	a kind of operating system + hardware support	hypervisor + hardware support
effect on task	appears to be alone on computer + operating system	appears to be alone on bare computer (except for “holes in CPU time”)
validation effort for task	as required for this task	
validation effort for kernel/hypervisor	like for the most critical task, but only once	
amount of latter validation effort	medium	small
operating system support	yes	no

separation kernel:

- originally proposed for security (Rushby 1981)

hardware support:

- need of memory protection

- (not always available on space computers)

alone on computer:

- can communicate via wires only (maybe virtual wires)

- (e.g., no program calls)

hypervisor:

- simple, small monitor program

- allocates computing time, memory, and peripheral devices
to “partitions”

- no further functionality

- (is “simple”, in contrast to an operating system)

- usually static, cyclic scheduling of the partitions

Overview

A Survey of Time and Space Partitioning for Space Avionics

- 1 Systems with Mixed Dependability
- 2 Integrated Modular Avionics (IMA) for Aircraft
- 3 Adaption of IMA for Space Avionics
- 4 Some Research Challenges

Motivation: Evolution of the Avionics Architecture

Integrated Modular Avionics (IMA) for Aircraft

- trend to sharing computer hardware:

feasible because of ever faster computers

(often: 1 computer much faster than needs of 1 application)

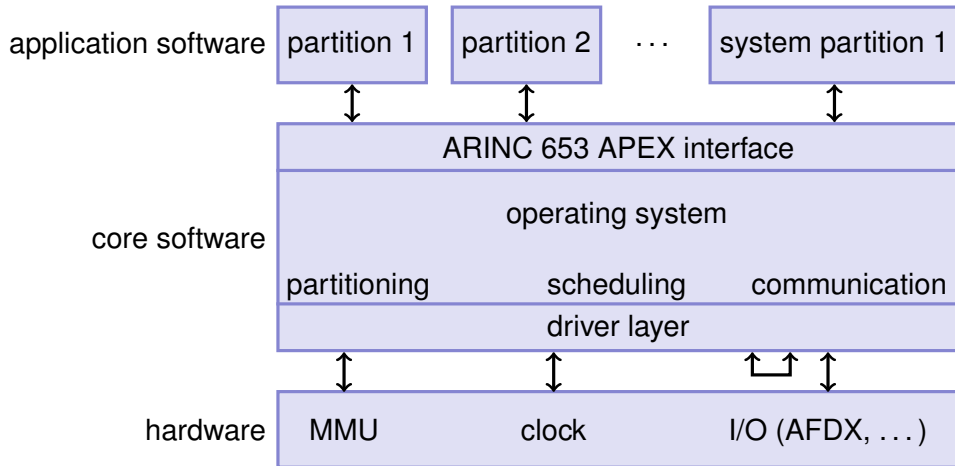
saves weight on aircraft
and thus saves cost

- trend to general-purpose computing modules:

saves on development and on worldwide stock of replacement units
and thus saves cost

System Architecture of an IMA module

Integrated Modular Avionics (IMA) for Aircraft



Legend: System Architecture of an IMA module

Integrated Modular Avionics (IMA) for Aircraft

■ IMA: separation kernel approach

partition 1, 2, ...: one encapsulated application each

system partition: administrates the other partitions (start/stop...)

APEX: = API (here: for access to operating system)

MMU: hardware support for memory access protection
(for enforcing the separation of the partitions)

AFDX: "**A**vionics **F**ull **D**uple**X** Switched Ethernet"
100 MBit/s real-time Ethernet

communication: with other computers or
loopback to other partitions on same computer

Summary of Overview

Integrated Modular Avionics (IMA) for Aircraft

Integrated Modular Avionics

- few, **standardized computing modules**
- **1 standardized type of bus** (fast, real-time)
- **1 standardized IMA operating system interface** (with partitioning)
(separation kernel approach)

Used in Practice

Integrated Modular Avionics (IMA) for Aircraft

- Airbus A380
- Airbus A400M
- Airbus A350XWB
- Boeing 787 Dreamliner
- ...

Extension/Research: Distributed Modul Avionics (DME)

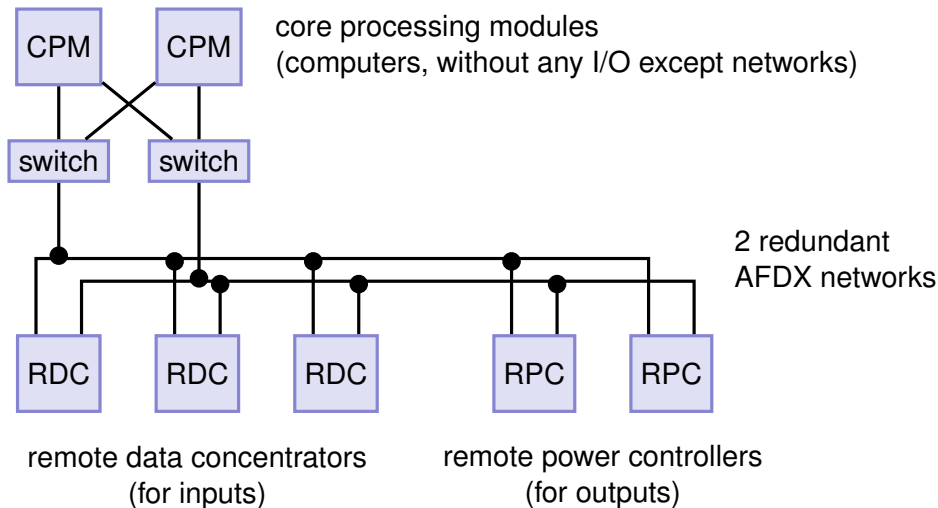
Integrated Modular Avionics (IMA) for Aircraft

idea

- IMA:
each sensor/actuator hard-wired to 1 IMA module
- DME:
separate processing power from sensor/actuator interfaces
(thus reducing the number of component types to a minimum)

System Architecture of Distributed Modular Electronics (DME)

Integrated Modular Avionics (IMA) for Aircraft



Overview

A Survey of Time and Space Partitioning for Space Avionics

- 1 Systems with Mixed Dependability
- 2 Integrated Modular Avionics (IMA) for Aircraft
- 3 Adaption of IMA for Space Avionics
- 4 Some Research Challenges

Differences Between the Aeronautical and the Space Domain

Adaption of IMA for Space Avionics

- the speed of growth of (software) complexity
- scale of communication demands (among computers)
- online/offline maintenance
- pronounced mission phases
- radiation
- availability of a hardware-based memory protection unit

more details: [see my full paper](#)

Differences Between the Aeronautical and the Space Domain

(1)

Adaption of IMA for Space Avionics

speed of growth of complexity:

complexity problems hit only later (=now)

scale of communication demands:

since less complex

maybe even 1 hardware node only, without inter-node communication

(except redundant node)

online/offline maintenance:

offline reconfiguration like with IMA/AFDX not possible

pronounced mission phases:

e.g., ascent, orbit insertion, orbital payload operation, deorbiting with a lot of time inbetween

allows for reconfiguration of computing resources and even software updates

all of this applies to satellites only, not to launchers

radiation:

- modern off-the-shelf processors malfunction or fail permanently
- larger chip structures necessary:
 - custom processors, much slower

availability of a hardware-based memory protection unit:

- because of radiation:
 - available in quite recent processors (starting from Leon3)

The Original IMA-SP Project

Adaption of IMA for Space Avionics

- IMA-SP: "Integrated Modular Avionics for Space" research project of the European Space Agency (ESA)
- motivation similar to IMA
- but tailored for space domain:
 - slower processors because of radiation
 - less complex systems (compare above)

- original project ended 2012
- several follow-up projects
(more on them: [see my full paper](#))

The IMA-SP Platform

Adaption of IMA for Space Avionics

- adoption of the basic IMA concept,
addition of space-specific requirements,
removal of the standardized communication via AFDX
- **result: a rather specific platform**
(not even suitable for launchers, suitable for satellites only)

No Standardized Communication (AFDX) in IMA-SP

Adaption of IMA for Space Avionics

- in space: less complex systems
→ more often: all functionality on 1 node only,
then communication among nodes not relevant
- in space: often other, slower buses than Ethernet/AFDX,
e.g., SpaceWire

- therefore in IMA-SP: AFDX not mandatory
(also not mandatory: other communication)

My Opinion

Adaption of IMA for Space Avionics

- the sum of “user requirements” results in an **architecture for a rather narrow application area**

example:

additional services for communication via shared memory are mandatory in IMA-SP, instead of optional

- apparently no generalization step by an up-front investigation of the common requirements of the aeronautical and the space domain
- emphasis: preserving long-proven ideas, approaches, and even hardware from the (satellite) space domain

Extensions for Multi-Core Processors:

The MultiPARTES Project

Adaption of IMA for Space Avionics

- **"Multi-cores Partitioning for Trusted Embedded Systems"**
- adapts the XtratuM hypervisor for **multi-core processors**
- reason: nearly all modern processors are multi-core
- **more** details: **see my full paper**

- **problem:**
verification of real-time properties very hard with multi-core,
because of common resources (e.g., cache)
- **solution** brings **limited progress**, only:
simply several independent Leon3 CPUs on a single FPGA chip,
under a single hypervisor, at least

The MultiPARTES Project

Adaption of IMA for Space Avionics

verification . . . very hard:

even Intel and AMD failed, see current "Meltdown" security flaw
in most modern processor architectures

Overview

A Survey of Time and Space Partitioning for Space Avionics

- 1 Systems with Mixed Dependability
- 2 Integrated Modular Avionics (IMA) for Aircraft
- 3 Adaption of IMA for Space Avionics
- 4 Some Research Challenges

Research Challenges

Some Research Challenges

Research Challenges for Time Partitioning

- multi-core CPUs
- direct memory access (DMA)

Research Challenges for Real-Time Property Proofs

- worst-case performance and processor architecture
- timing anomalies and processor architecture

refs to some work on this: see my full paper

Research Challenges for Time Partitioning

Some Research Challenges

multi-core CPUs:

current CPUs: several cores, but (some) shared resources (e.g., caches)

consequence: dependency of the execution time between cores

turn off caches? – maybe slower than 1 core only

direct memory access (DMA):

DMA controller contends with CPU for memory bus

consequence: real-time partition slowed down by DMA of a non-real-time partition

even CPUs can contend for memory bus in a similar way

Research Challenges for Real-Time Property Proofs

Some Research Challenges

worst-case performance and processor architecture:

conventional: architecture optimizes average-case performance

necessary: architecture optimizes worst-case performance

timing anomalies and processor architecture:



complex processor architectures: timing anomalies

consequence: worst-case execution time hard to determine

consequence: often rough estimates only

advantage in space domain: simpler processors because of radiation

References

-  **Avižienis, Algirdas et al. (2004).** “Basic Concepts and Taxonomy of Dependable and Secure Computing”. In: *IEEE Trans. on Dependable and Secure Computing* 1.1.
-  **Rushby, John (1981).** “The Design and Verification of Secure Systems”. Reprint of a paper presented at the 8th ACM Symposium on Operating System Principles, Pacific Grove, CA, USA, 14–16 Dec. 1981. In: *ACM Operating Systems Review* 15.5, pp. 12–21.