Some issues about IMA in safety critical applications

Paul Caspi

Former researcher at Laboratoire **Verimag** *(CNRS-UJF-INPG)*

IMA Artist 2 Workshop 12 – 13 November 2007

- Where does this viewpoint comes from?
- Why simple federated architectures were successful?
- Two accidents avoided by simple federated architectures
- What kind of guarantees are required from IMA?

Where does this viewpoint comes from?

Besides being a researcher in safety critical embedded control systems, I have been involved in consulting and certification activities, mainly in the railway field:

- The Hermes space-shuttle software consulting group (1987)
- The RER A emergency braking system certification committee (1986-1987)
- The driver-less Lyon subway scientific advisory group (1994-1997)
- Expert in computerised control at the Certifer certification agency (1998-)

In many of these activities, I was faced with the IMA question
This is why I am interested in this question

Safety-critical computerised control

Computer technology is known as being poorly reliable:

- thousands of car “recalled” for computing bugs
- big electricity and telephone crashes
- Ariane V accident
- your personal computer . . .

Safety-critical computerised control

Computer technology is known as being poorly reliable:

- thousands of car “recalled” for computing bugs
- big electricity and telephone crashes
- Ariane V accident
- your personal computer . . .

Two questions:

1. *Is it wise to use this poor technology in safety critical systems?*

Safety-critical computerised control

Computer technology is known as being poorly reliable:

- thousands of car “recalled” for computing bugs
- big electricity and telephone crashes
- Ariane V accident
- your personal computer . . .

Two questions:

1. *Is it wise to use this poor technology in safety critical systems?*

2. *Why, nevertheless, things are not as bad as it could be expected?*

Some partial answers

The safety-critical control industry has designed a very strong model-based development method

This is very important but is not today's question

Some partial answers

The safety-critical control industry has designed a very robust computer (hardware/software) architecture:

- Each computer is periodic and triggers a single loop software

```
read inputs
compute output and next state
wait for the next clock tick
```

(synchronous technology)

  - no dynamic memory, no unbounded loops

$\Rightarrow$ Schedulability is easy: $WCET < Period$
$\Rightarrow$ Jitter is minimised
$\Rightarrow$ The WCET evaluation is made easier
$\Rightarrow$ The need for operating system services is minimised

  - (a single interrupt which takes place when the computer is idle)

Some partial answers

The safety-critical control industry has designed a very robust computer (hardware/software) architecture:

- Computers periodically sample the physical world but also the other computers
  non blocking *communication by sampling*
  later modified by so-called *time-triggered architecture*

Some partial answers

The safety-critical control industry has designed a very robust computer (hardware/software) architecture:

- Segregation between critical and less critical tasks
  allows extending FMEA/FTA methods from hardware to computers and software
  criticality can be inherited backward from outputs to tasks

Some problems found when departing from this model

- The Ariane V accident
- Priority inversion in the MarsPathfinder

The Ariane V accident

An uncaught exception (overflow) causes the failure

A conjunction of several developments flaws

The Ariane V accident

An uncaught exception (overflow) causes the failure

A conjunction of several developments flaws

One of them is non segregation:

- The fault appeared in a non-critical function packed in the same computers as critical ones

The Ariane V accident

An uncaught exception (overflow) causes the failure

A conjunction of several developments flaws

One of them is non segregation:

- The fault appeared in a non-critical function packed in the same computers as critical ones

Another is a wrong use of FMEA/FTA techniques

- Since the function was not critical, exceptions needed not be caught

Priority inversion

- Takes place when multitasking (multi-threading) is used in conjunction with synchronisation)
- multitasking is mandatory in several cases:
  - multi-periodic systems
    (very frequent in control)
  - mixed event and time-triggered systems
- synchronisation is needed for communication between tasks ??
  other non blocking solutions exist!

Multitasking

- Raises scheduling problems

Multitasking

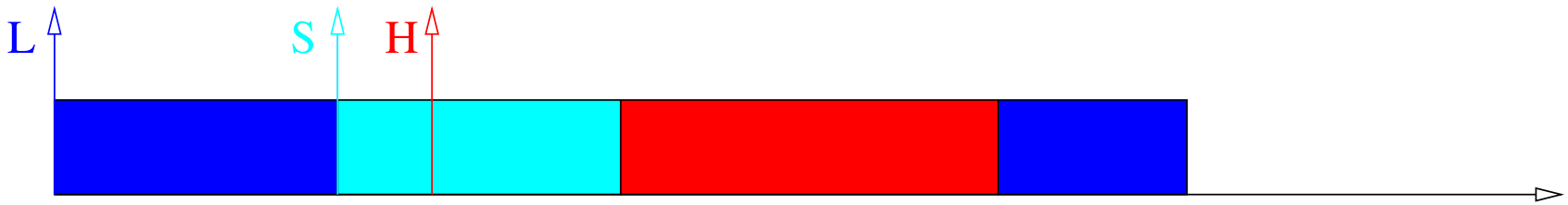
- Raises scheduling problems
  - efficient scheduling policies and associated scheduling tests exist for multi-periodic and event-triggered systems (with minimum inter-arrival time)
  - for instance dead-line monotonic fixed priority

Multitasking

- Raises scheduling problems
- Raises communications problems
  - preemption can corrupt data
  - critical sections can be a solution
  - scheduling tests can take it into account

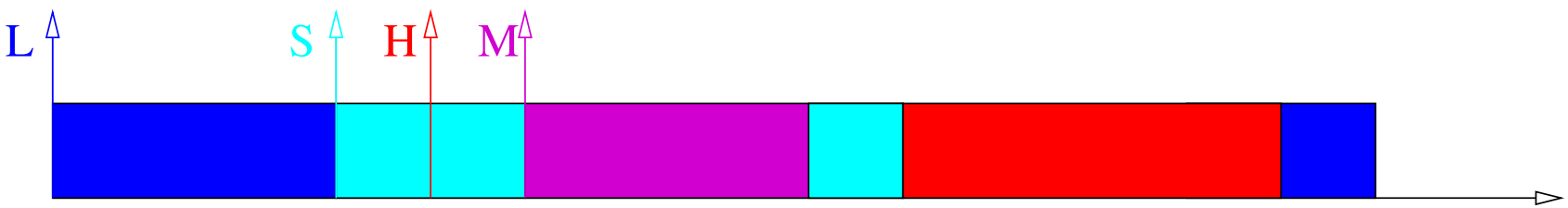
Synchronisation

High and Low share a critical section
High wants to execute when Low is in critical section
High is stalled until Low gets out of the critical section
No Problem: the schedulability test can account for that



Synchronisation

High and Low share a critical section
High wants to execute when Low is in critical section
Medium doesn't share this critical section
Medium occurs when Low is in critical section
Medium preempts Low
High is stalled
Priority Inversion



Hopefully it wasn't a safety-critical system

What kind of guarantees are required from IMA?

- An important certification principle is:

Non regression

What kind of guarantees are required from IMA?

- An important certification principle is:

  Non regression

- IMA has thus to prove at least that:
  - no side effect can result from violating the segregation principle
  - no side effect can result from violating the single thread principle
  - and possibly many other things...

What kind of guarantees are required from IMA?

- An important certification principle is:

  Non regression

- IMA has thus to prove at least that:
  - no side effect can result from violating the segregation principle
  - no side effect can result from violating the single thread principle
  - and possibly many other things...

Thus the use of a new technology like IMA should be thoroughly justified and its introduction should be progressive and careful.