Reconfiguration Strategies for Critical Adaptive Distributed Embedded Systems

Alberto Ballesteros
Julián Proenza
Manuel Barranco
Luís Almeida
Introduction

Adaptive Distributed Embedded Systems (ADES) can change autonomously and dynamically in response to unexpected operational requirements or conditions.
Adaptive Distributed Embedded Systems (ADES) can change autonomously and dynamically in response to unexpected operational requirements or conditions.

Adaptivity is an interesting feature in terms of:

- **Functionality** → Change the behaviour
- **Efficiency** → Load the necessary functionalities
- **Dependability** → Adaptive fault tolerance
The DFT4FTT project

To **properly implement an ADES** it must be provided with the appropriate **architecture** and **mechanisms**, that make it possible to fulfil its **real-time, dependability** and **adaptivity** requirements.
To properly implement an ADES it must be provided with the appropriate architecture and mechanisms, that make it possible to fulfil its real-time, dependability and adaptivity requirements.
The DFT4FTT project

To **properly implement an ADES** it must be provided with the appropriate **architecture** and **mechanisms**, that make it possible to fulfil its **real-time**, **dependability** and **adaptivity** requirements.
At the **node level**, the DFT4FTT architecture is composed of **various components**
At the **node level**, the DFT4FTT architecture is composed of **various components**
At the node level, the DFT4FTT architecture is composed of various components.
At the **node level**, the DFT4FTT architecture is composed of **various components**
At the **node level**, the DFT4FTT architecture is composed of **various components**

- Monitor
- Detect
- Configuration change

The System Architecture
At the node level, the DFT4FTT architecture is composed of various components.
At the **node level**, the DFT4FTT architecture is composed of **various components**.
At the **node level**, the DFT4FTT architecture is composed of **various components**
The task allocation scheme

At the **node level**, the DFT4FTT architecture is composed of **various components**.
At the **node level**, the DFT4FTT architecture is composed of **various components**.
Reconfiguration Strategies

Constantly verify that the system reqs are fulfilled

- Faulty CNs
- Tasks executed
- Failure rates
- ...

System state → Fulfilment? \[\downarrow\] System reqs

Change configuration

List of tasks

RT requirements

R(t) requirements
Reliability perspective

The reconfiguration capabilities of the NM allows us to reallocate the tasks being executed in one CN to another, when the first one suffers a permanent failure.

Non-critical tasks

- The service is restores after some downtime.

Critical (replicated) tasks

- We have redundancy preservation.
- Equivalent to N-Modular Redund. scheme with spares.
Reliability

The reconfiguration capability reallocates the tasks being executed when the first one suffers a permanent failure.

Non-critical tasks
- The service is restored after some downtime.

Critical (replicated) tasks
- We have redundancy preservation.
- Equivalent to N-Modular Redundancy scheme with spares.

Thank you for your attention!

See you at the poster session!