FDIR

Spacecraft fault protection system

Project 1
Part 1

Euro Team

Alauzet Pierre, Ahvenniemi Mikko,

Colin Julien, Starck Benoit

KAIST

CS554 - Design for Software & Systems

September 24th, 2009

TABLE OF CONTENTS

BACKGROUND

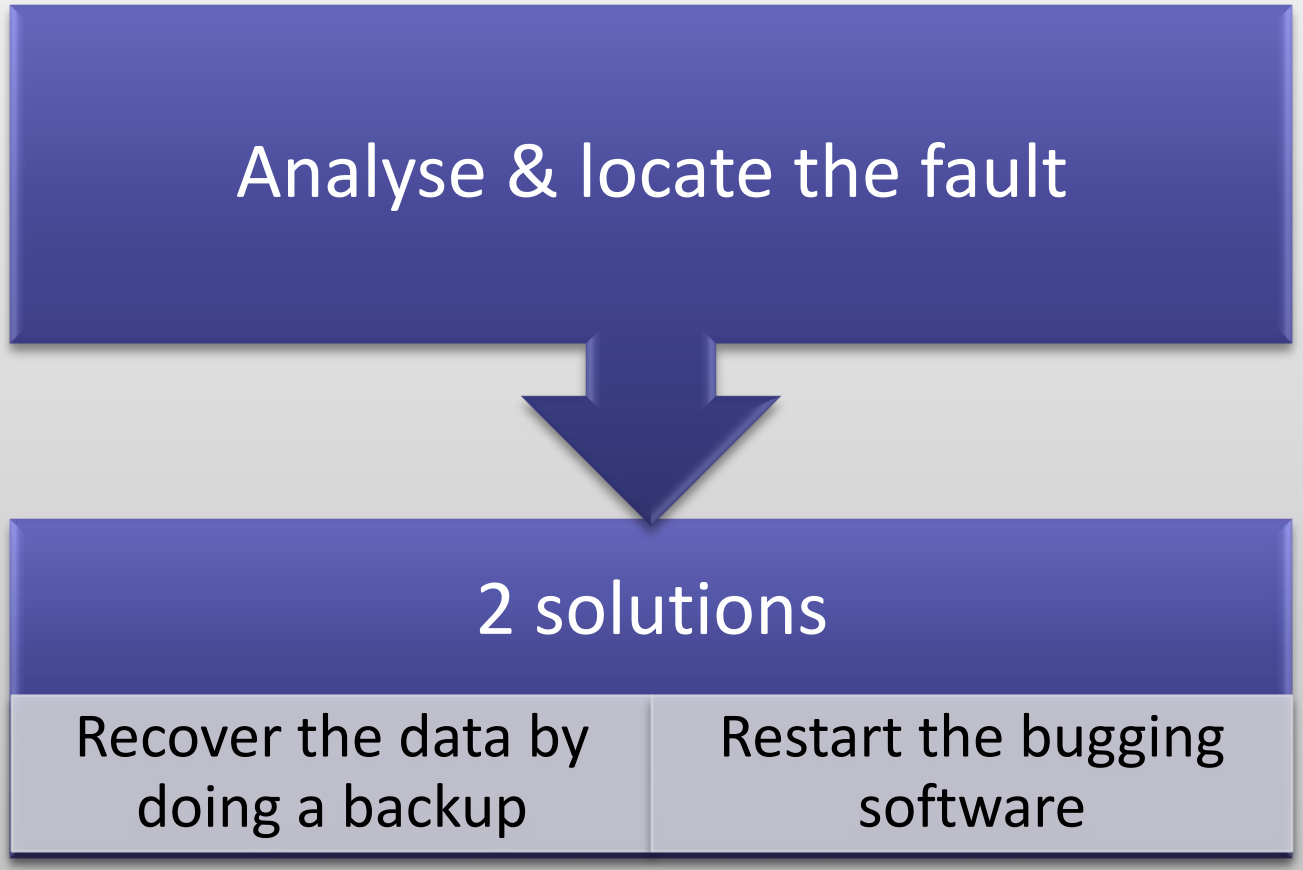
- Fault
  - Detected when monitored values are out-of-tolerance
- Fault protection system (FDIR)
  - Act when the spacecraft is going through an error or a fault
  - FDIR is a layered system. If a lower layer cannot resolve an issue it's forwarded to an upper layer. If the Issue cannot be resolved by the system. It's escalated to manual control.
  - Automatic system
- Spaceship crew and flight control can manually control the system

REQUIREMENTS

- *Global Requirements :*
  - Guarantee the completion of any time critical activities of the spaceship
  - Keep the control of the spacecraft with safety, observability & commandability

REQUIREMENTS (CONT.)

- ***Fault Recovery :***



REQUIREMENTS (CONT.)

- *Safing response in the case of hazardous conditions :*

Isolate the problem	
Unmanned spacecraft : Shutdown all non-critical functions	Manned spacecraft: Crew Intervention

NON-FUNCTIONAL REQUIREMENTS

- Testability
  - The system and its parts have to be able to be tested through inspections, simulations and analyses before on-board installation
- Reliability
  - The system must be reliable in all operating conditions. System failure could lead to loss of human life.
- Availability
  - The system must not lock or stall when processing data. It must work asynchronously.

NON-FUNCTIONAL REQUIREMENTS

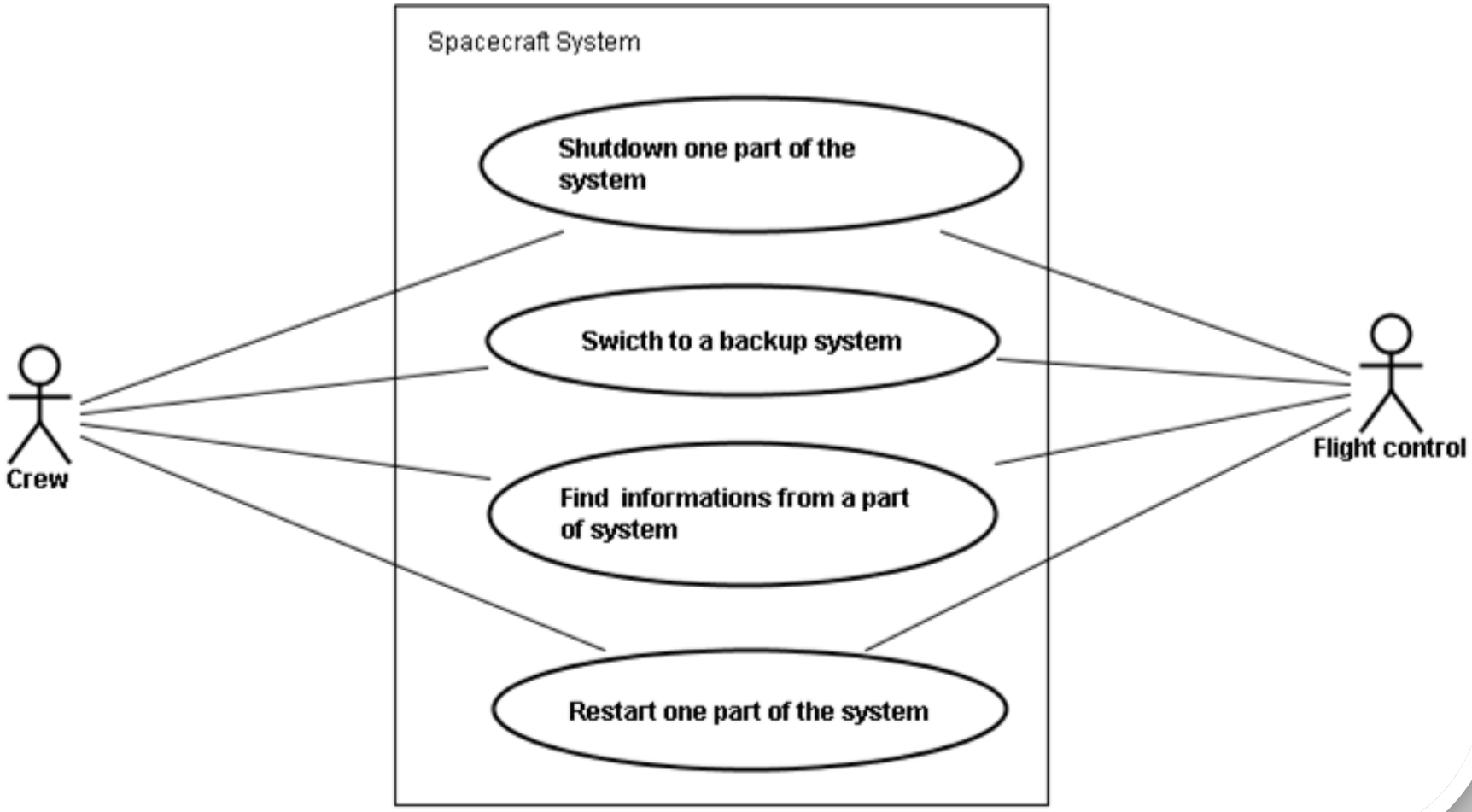
- Resilience
  - The system must be able to maintain an acceptable level of service in spite failures in parts of the FDRI system.
- Response time
  - The system must respond in timely manner so that problematic systems can be shut down before any damage is done.
- Documentation
  - Technical and software documentation has to be accurate so that the spacecraft crew and flight control know how to administer the system and perform actions through it.

USE CASE DIAGRAM EXPLANATIONS

- Most of the time FDRI works automatically. However, spaceship crew and flight control can manually control the system.

- These interactions are represented in the following use case diagram.

USE CASE DIAGRAM



USE CASES DESCRIPTIONS

- Shutdown one part of the system:
  - Context of use : a part of the system is failed, we want to shutdown it
  - Actors : Crew / Flight control
  - Pre-condition: none
  - Post-condition: the part of the system is shutdown
  - Guaranty in case of success : any actions could not be done to this part
  - Guaranty in case of failure : the system turns on

USE CASES DESCRIPTIONS (CONT.)

- Restart one part of the system:
  - Context of use : a part of the system is shutdown, we want to restart it
  - Actors : Crew / Flight control
  - Pre-condition: the part is shutdown
  - Post-condition: the part is switched on
  - Guaranty in case of success : Any action could be done after reboot
  - Guaranty in case of failure : the system is off yet

USE CASES DESCRIPTIONS (CONT.)

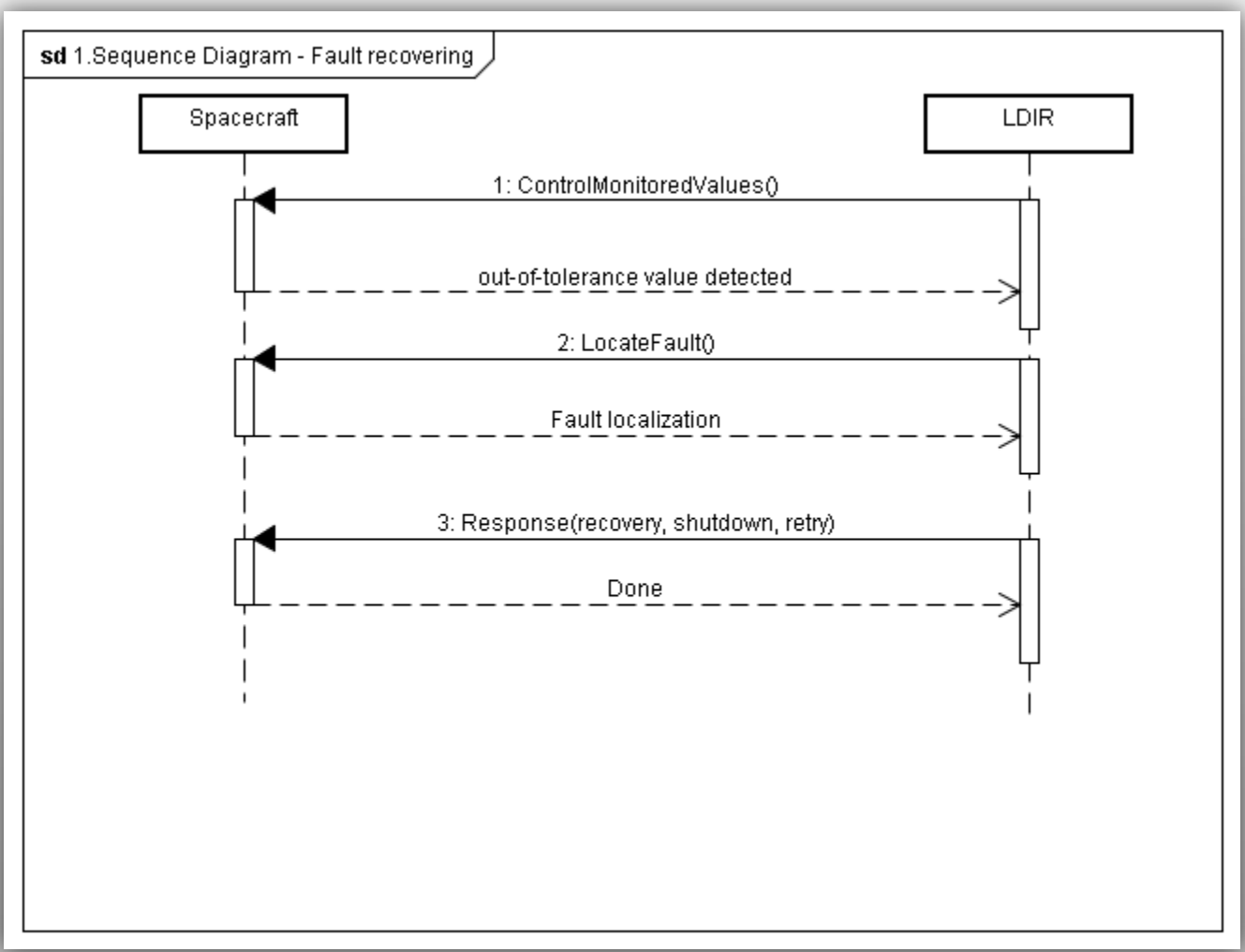
- Switch to backup system:
  - Context of use : the current system is corrupted
  - Actors : Crew / Flight control
  - Pre-condition: none
  - Post-condition: the faulty system is switched to a spare system
  - Guaranty in case of success : the system is not corrupted yet
  - Guaranty in case of failure : none

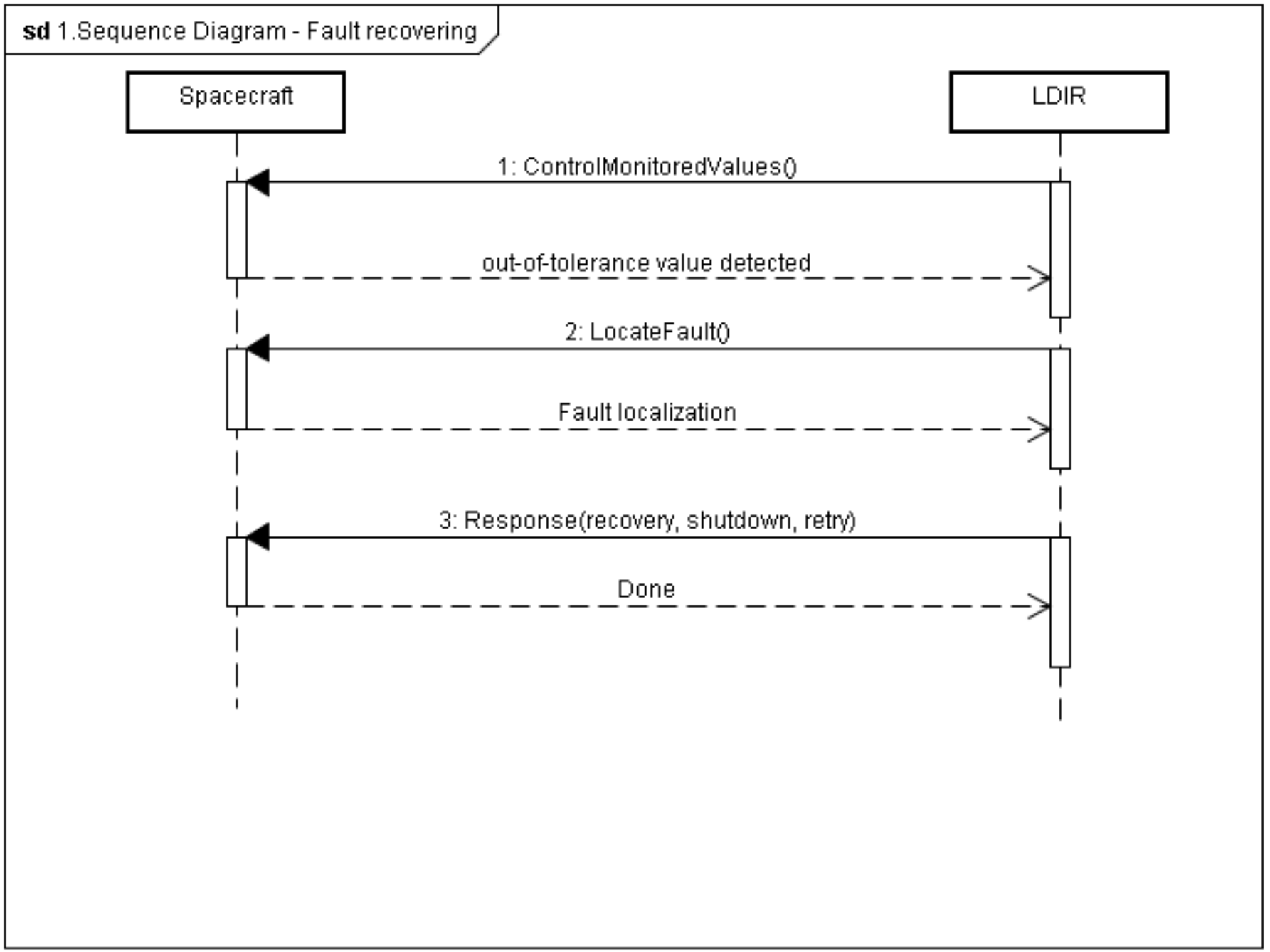
USE CASES DESCRIPTIONS (CONT.)

- Find information about a part of the system:
  - Context of use : the crew wants to know some information about a part of the system
  - Actors : Crew / Flight control
  - Pre-condition: none
  - Post-condition: the crew receives the information
  - Guaranty in case of success : we have the information
  - Guaranty in case of failure : we don't have the information

SEQUENCE DIAGRAM

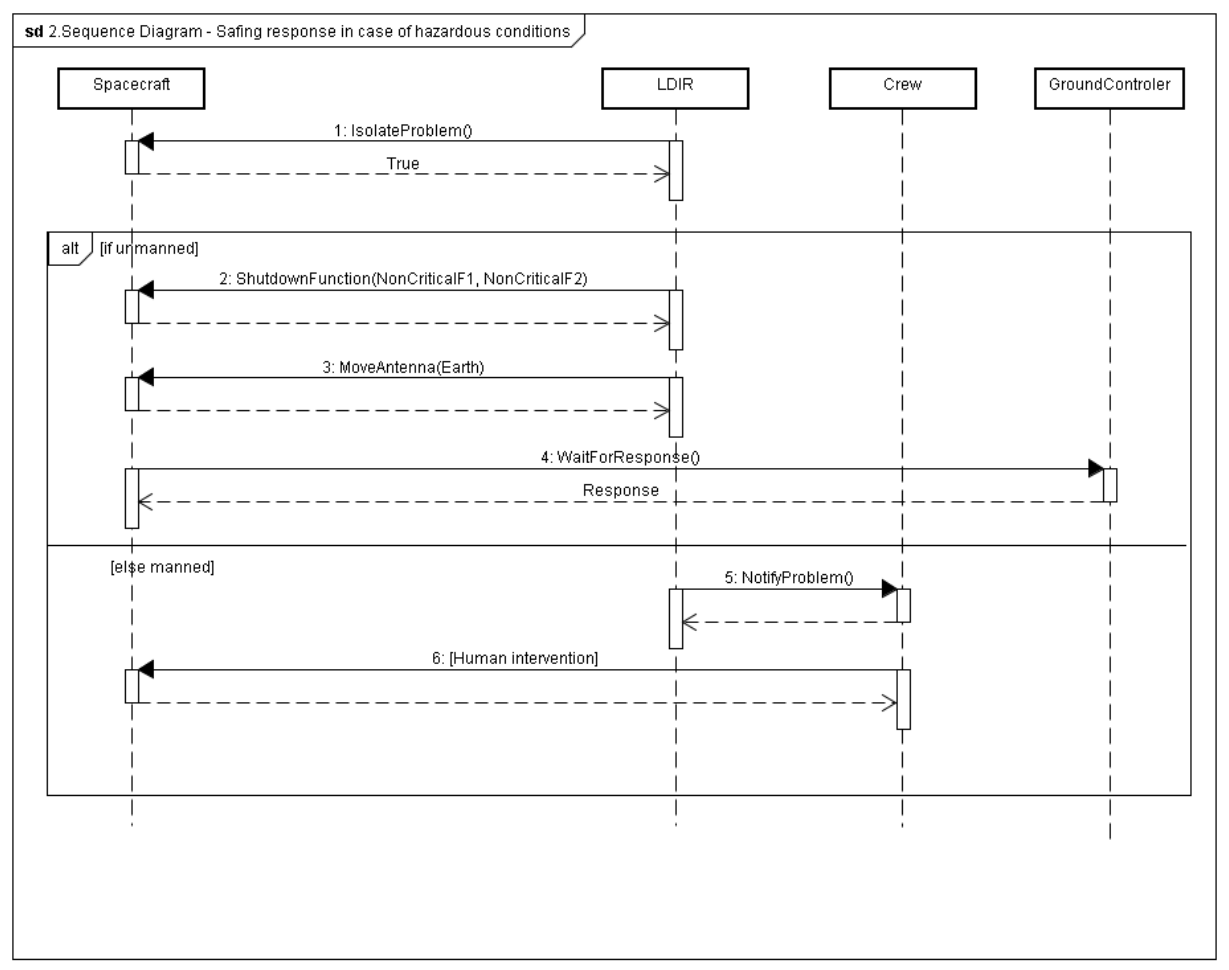
- Fault recovering

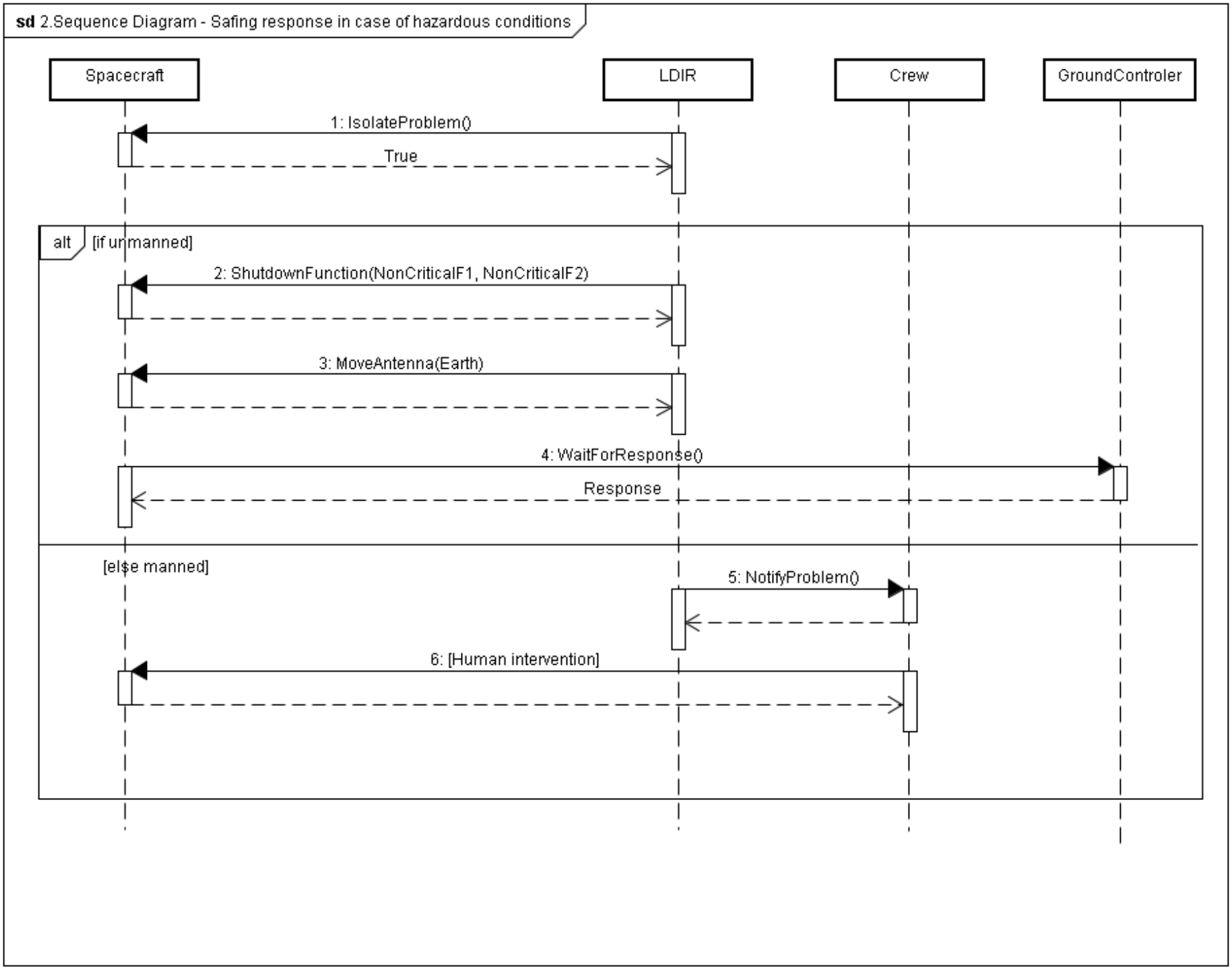



sd 1.Sequence Diagram - Fault recovering
Spacecraft
LDIR
1: ControlMonitoredValues()
out-of-tolerance value detected
2: LocateFault()
Fault localization
3: Response(recovery, shutdown, retry)
Done

SEQUENCE DIAGRAM (CONT.)

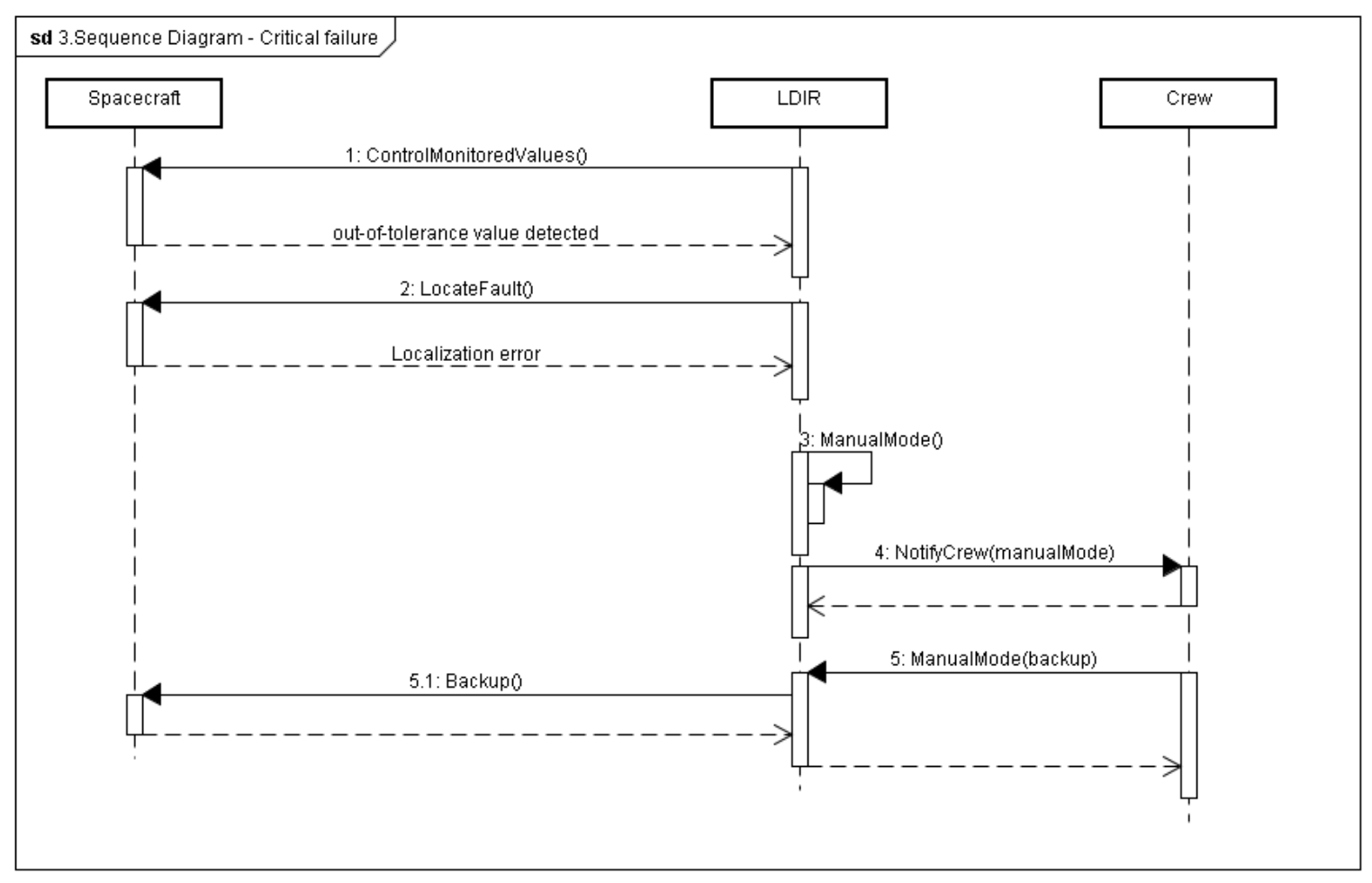
- Safing response in case of hazardous condition

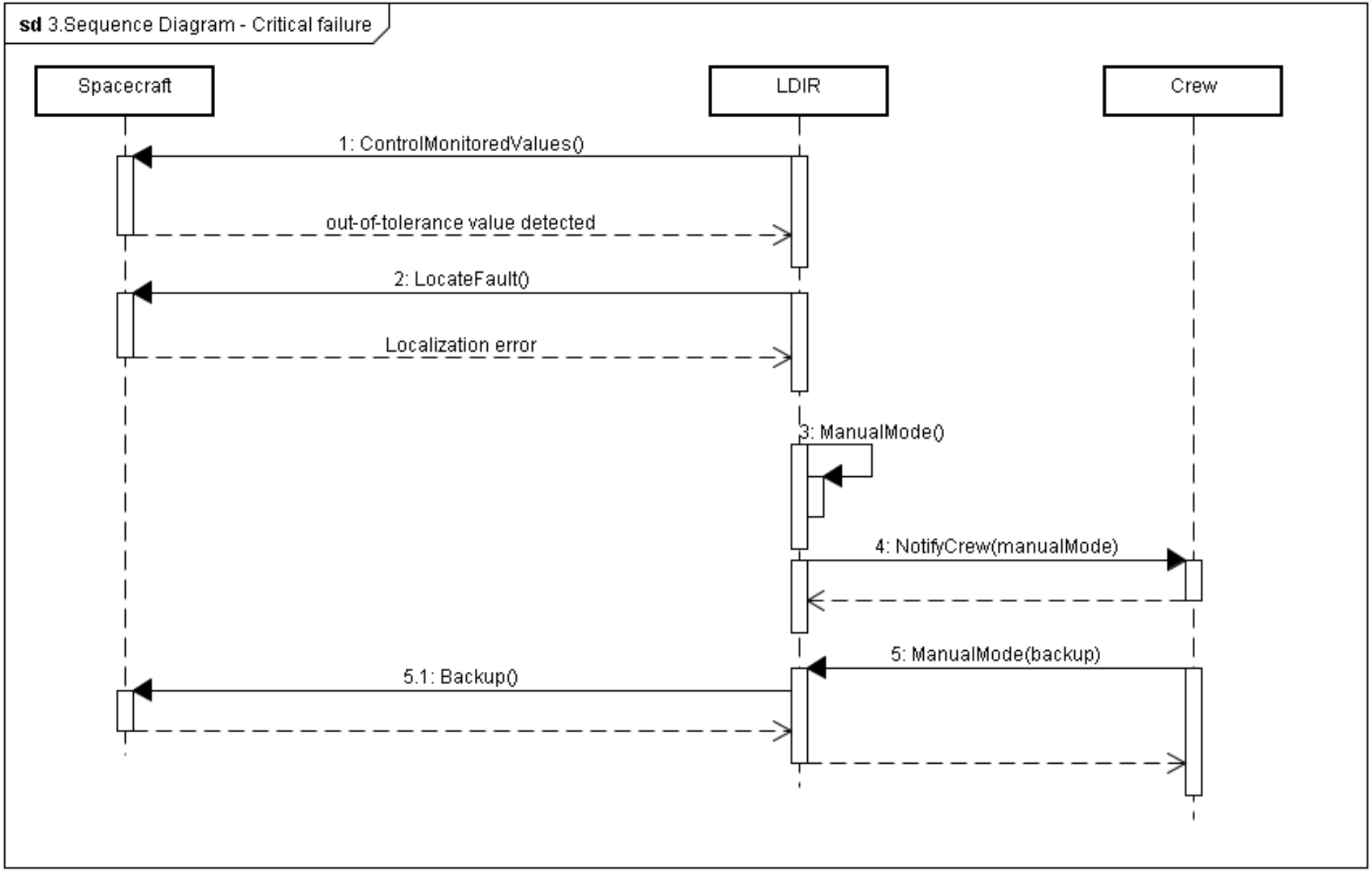



sd 2.Sequence Diagram - Safing response in case of hazardous conditions
Spacecraft
LDIR
Crew
GroundControler
1: IsolateProblem()
True
alt
[if unmanned]
2: ShutdownFunction(NonCriticalF1, NonCriticalF2)
3: MoveAntenna(Earth)
4: WaitForResponse()
Response
[el$e manned]
5: NotifyProblem()
6: [Human intervention]

SEQUENCE DIAGRAM (CONT.)

- Critical failure




sd 3.Sequence Diagram - Critical failure
Spacecraft
LDIR
Crew
1: ControlMonitoredValues()
out-of-tolerance value detected
2: LocateFault()
Localization error
3: ManualMode()
4: NotifyCrew(manualMode)
5: ManualMode(backup)
5.1: Backup()

FDIR

Spacecraft fault protection system

Project 1
Part 1

Euro Team

Alauzet Pierre, Ahvenniemi Mikko,

Colin Julien, Starck Benoit

KAIST

CS554 - Design for Software & Systems

September 24th, 2009

REFERENCES

1. [Eas98] Steve Easterbrook, and et al., “Experiences Using Lightweight Formal Methods for Requirements Modeling,” IEEE Transactions on Software Engineering, Vol. 24, No. 1, January 1998.