**Project 1**





**FDIR**

*Spacecraft fault protection system*

**Euro Team**

|  |  |
| --- | --- |
| **Mikko AHVENNIEMI** | 20096680 |
| **Pierre ALAUZET** | 20096699 |
| **Julien COLIN** | 20096706 |
| **Benoît STARCK** | 20096705 |

**Table of contents**

October 15th, 2009

**CS554 - Design for Software & Systems**

[**1.** **Problem understanding** 6](#_Toc242772199)

[a. Business case & system context 6](#_Toc242772200)

[**i.** **FDIR utility** 6](#_Toc242772201)

[**ii.** **System requirements** 6](#_Toc242772202)

[b. Problem frames 6](#_Toc242772203)

[**i.** **Domains identification** 6](#_Toc242772204)

[**ii.** **Context Diagram** 6](#_Toc242772205)

[**iii.** **Problem frames** 6](#_Toc242772206)

[**2.** **Functional requirements (use-case model)** 7](#_Toc242772207)

[a. Use-case diagram 7](#_Toc242772208)

[**i.** **Actors description** 7](#_Toc242772209)

[**ii.** **Use-case diagram** 7](#_Toc242772210)

[**iii.** **Use-case specifications** 7](#_Toc242772211)

[b. Sequence diagrams 10](#_Toc242772212)

[**i.** **Fault recovering** 10](#_Toc242772213)

[**ii.** **Safing response in case of hazardous conditions** 10](#_Toc242772214)

[**iii.** **Critical failure** 10](#_Toc242772215)

[**3.** **Non-functional requirements** 11](#_Toc242772216)

[a. Identified quality attributes 11](#_Toc242772217)

[**i.** **Testability** 11](#_Toc242772218)

[**ii.** **Availability** 11](#_Toc242772219)

[b. Improvised quality attributes 11](#_Toc242772220)

[**i.** **Reliability** 11](#_Toc242772221)

[**ii.** **Resilience** 11](#_Toc242772222)

[**iii.** **Response time** 11](#_Toc242772223)

[**4.** **Usability analysis & design** 12](#_Toc242772224)

[a. Preliminary user interface design 12](#_Toc242772225)

[b. Discussion on usability scenarios 12](#_Toc242772226)

Illustration table

[Figure 1: Use case diagram 6](file:///D:\Documents\UTBM\MON%20CURSUS\GI05\KAIST\Projects\euroteam\Project%201\Part%20Final\CS554_EuroTeam_Project1.docx#_Toc242773046)

**Introduction**

1. **Problem understanding**
   1. Business case & system context
      1. FDIR utility
      2. System requirements
   2. Problem frames
      1. Domains identification

We identified the FDIR system domains as depicted on the context diagram (Figure 1: Context diagram). Crew is identified as a biddable domain. The crew consists of the people on board of the spacecraft if any. In case of a satellite there might not be any crew. Ground Control is a biddable domain as well. It consists of the people on earth monitoring the progress of space missions. They have to possibility to issue commands to the system in addition of the crew.

Information display is a causal domain and its essentially displays installed in the spacecraft, but it can also be a display in the ground control premises on earth. Other kinds of displays could be envisioned as well, for example PDA’s. The FDIR storage system is a lexical domain and contains the historical data from different systems plugged in to the FDIR as well as the data from FDIR operation.

Report is a causal domain. It contains information as specified by a search usually performed on the FDIR Storage System. Systems is a causal domain. It consist of the different devices and systems plugged in to the control of the FDIR system.

* + 1. Context Diagram

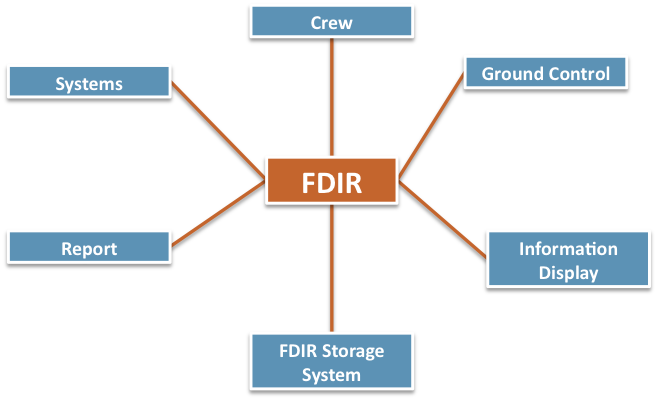


Figure : Context diagram

* + 1. Problem frames

We identified distinct problem frames depicting certain scenarios in the FDIR systems. These frames are in order of appearance: Automatic recovery from failure, Manual control of FDIR, Displaying information continuously, Collecting system data to data storage, Information retrieval, Providing failure localization, and Response in case of irresolvable failure.

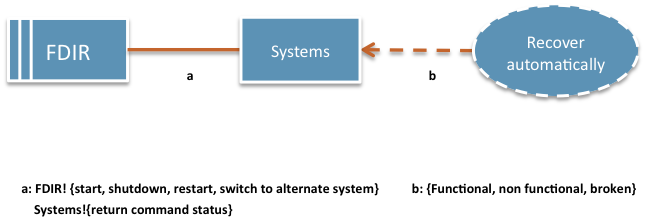


Figure : Automatic recovery from failure

Automatic recovery from failure () uses the required behavior problem frame. If FDIR deems that a failure has occurred it will try to remedy the situation by a series of actions. It can try to start, shutdown, restart the system, or try to switch to an alternate system. The systems can be, at any point in time, be functional, non-functional or broken. After issuing commands it will receive the command status from the systems.

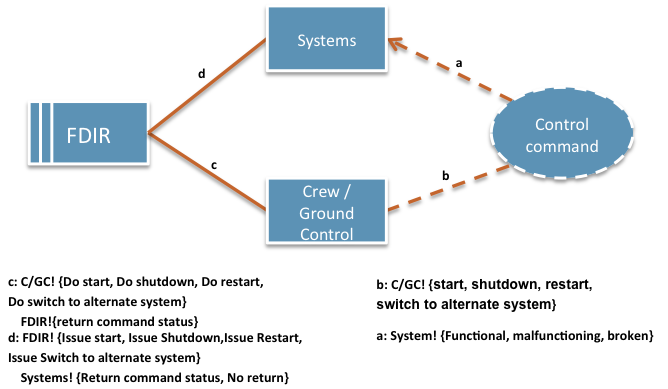
Manual control of FDIR uses the commanded behavior problem frame. The FDIR provides an interface for issuing manual commands for the crew and ground control. They might want to e.g. restart a system that is not operating in tolerance, or switch that function to use an alternate, or backup system.

Figure : Manual control of FDIR

Commands issued by the crew or ground control are relayed to the systems by FDIR and the command status is returned to FDIR. In case the system is broken the system might not respond. The system can be Functional, malfunctioning or broken.

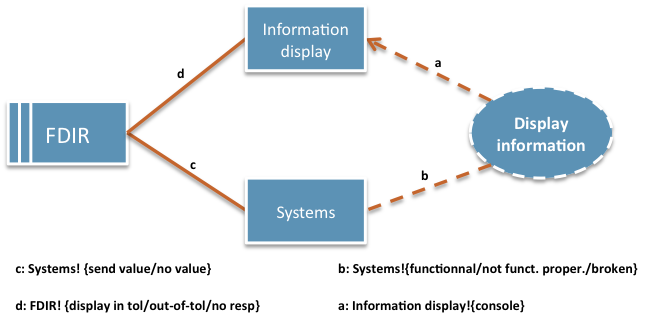


Figure : Displaying information continuously

Display information continuously (Figure 4) uses the display problem frame. The FDIR constantly relays information from the systems to be displayed on screen. The crew and flight control can make deductions about the systems based on this data.

Systems send the values to FDIR and no value if they are broken. FDIR on the other hand displays the value and interprets whether it’s in tolerance. The information is displayed on the console depending on what kind of view is selected in the user interface.

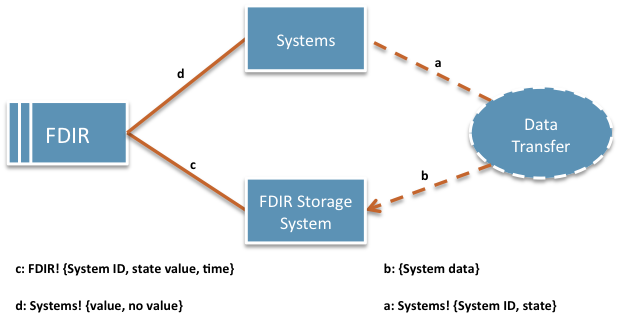


Figure : Collect systems data to data storage

Collecting systems data to data storage (Figure 6) uses the transformation problem frame. The systems send measurement values to FDIR at regular intervals. This data is stored with a timestamp to the FDIR Storage System. This approach has a link to non-functional requirements as well. If the data is stored in a centralized place with proper backup in place it doesn’t matter if some parts of the system go down and the data can still be accessed. This contributes to better availability of data used for critical decision making.

All systems send some kind of data. The FDIR knows the system ID based on where the system is located. The state value is then stored with a timestamp.

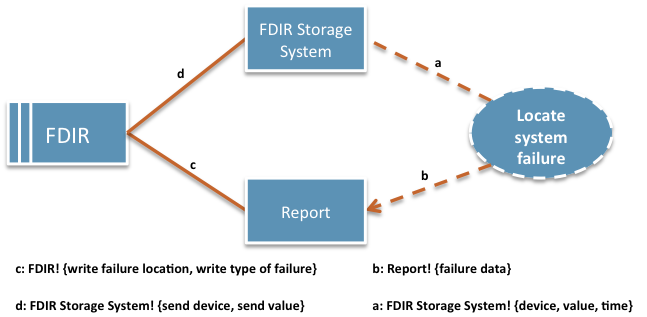


Figure : Providing failure localization

Providing failure localization (Figure 7) uses the transformation problem frame. Data is retrieved from the FDIR Storage System and then FDIR determines based on this data where the failure originates. This information is then written on a report that is later displayed based on which user interface view is open. The type of the failure is recorded in the report in addition to the localization data.

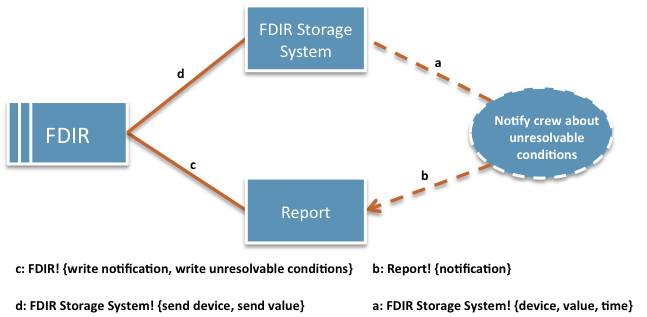


Figure : Response in case of unresolvable conditions

Response in case of irresolvable conditions (Figure 8) uses the transformation problem frame. When automatic recovery fails and the issue cannot be resolved by the FDIR by itself a report is written that is then actionable by the crew or ground control. This report is based on the data collected to the FDIR Storage System. Information retrieval (Figure 9) uses the commanded behavior problem frame. The crew can search information from the system logs. Each log item is stored separately in the FDIR Storage System. By specifying search criteria specific information can be retrieved. This data can then be used as basis for important decision regarding the use or repair of the systems. This data is can be more specific then general information displayed on the screen via the user interface.

The query goes through the FDIR to the storage system. Data is then sent back to FDIR for processing. Data is then sent to the crew and ground control for viewing.

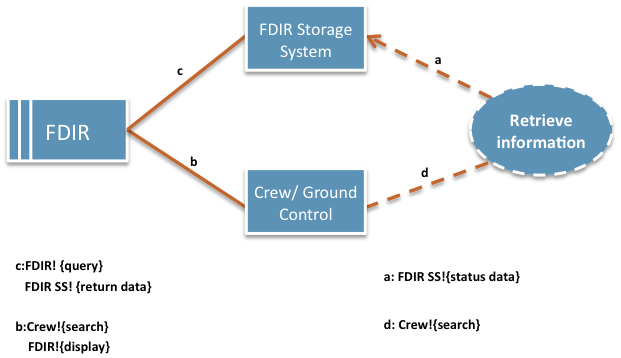


Figure : Information retrieval

* **Functional requirements (use-case model)**
  1. Use-case diagram
     1. Actors description
     2. Use-case diagram

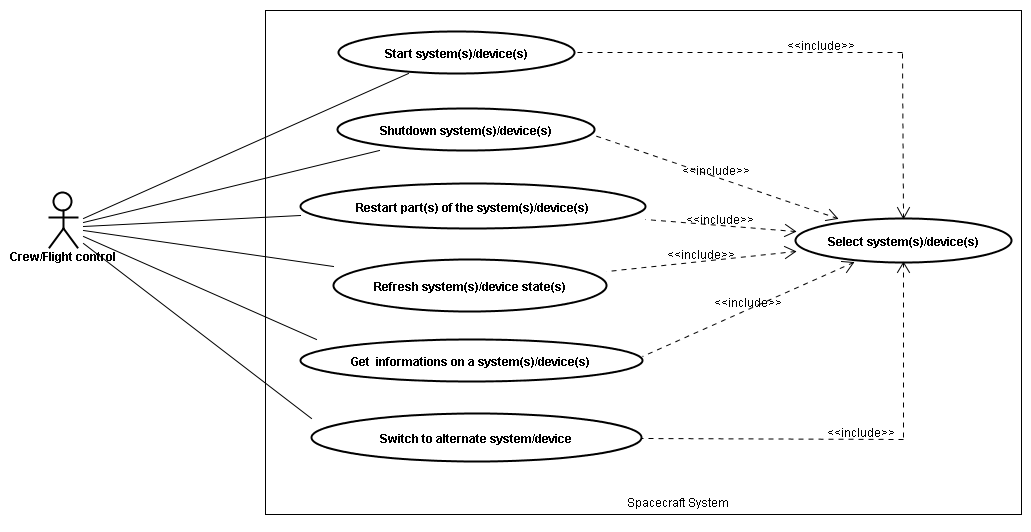


Figure 10: Use case diagram

* + 1. Use-case specifications

|  |  |
| --- | --- |
| Name | Start system(s)/device(s) |
| Actors | Crew / Flight control |
| Description | User can start device(s) or system(s) of the spacecraft system whenever he wants or because it was shutdown |
| Precondition | - System(s) are off  - One or several device(s) or system(s) have been selected  - Selected device(s) or system(s) must not be dependant on other device(s) or system(s) that are offline. |
| Events flow | 1. Click on the “start” button  2. Wait for the system to start |
| Post-condition | System(s) has been started |
| Exception | * System states could be already turned on while it is still displayed as “off” on the FDIR system * System cannot be started because it’s broken. Display on error message. * If selected device(s) or system(s) are dependant on other device(s) or system(s), display a warning and information about the dependencies, and offer to override. |

|  |  |
| --- | --- |
| Name | Shutdown system(s)/device(s) |
| Actors | Crew / Flight control |
| Description | User can shutdown device(s) or a part(s) of the spacecraft system whenever he wants or in case of failure |
| Precondition | - Devices or parts of the system have to be running  - One or several device(s) or system(s) have been selected  - Other systems must not have dependencies to the selected device(s) or system(s) |
| Events flow | 1. Click on the “shutdown” button |
| Post-condition | System(s) has been shutdown |
| Exception | * System states could be already turned off while it is still displayed as “running” on the FDIR system * If other systems have dependencies to the selected devices(s) or system(s) display a warning and information about the dependencies and offer to override. |

|  |  |
| --- | --- |
| Name | Restart part(s) of the system(s)/device(s) |
| Actors | Crew / Flight control |
| Description | User can restart a device or a part of the spacecraft system whenever he wants or in case of failure |
| Precondition | - Device or part of the system has to be running  - Requested system(s)/device(s) have been selected  - Other systems must not have dependencies to the selected device(s) or system(s) |
| Events flow | 1. Click on the “restart” button |
| Post-condition | System is restarting |
| Exception | * System states could be already turned off while it is still displayed as “running” on the FDIR system * If other systems have dependencies to the selected devices(s) or system(s) display a warning and information about the dependencies and offer to override. |

|  |  |
| --- | --- |
| Name | Refresh system(s)/device(s) states |
| Actors | Crew / Flight control |
| Description | User can refresh the states of any device or system to see if this one is still working correctly or not |
| Precondition | - Device or part of the system has to be running  - Requested system(s)/device(s) have been selected |
| Events flow | 1. Click on the “refresh” button |
| Post-condition | System is refreshing |
| Exception | - If the system(s) or device(s) are not responding change status not responding. |

|  |  |
| --- | --- |
| Name | Switch to alternate system/device |
| Actors | Crew / Flight control |
| Description | If the device is not responding or if there is a failure, user may switch to another system/device |
| Precondition | - Select **one and only one** system or device  - Requested part has been selected  - Other system(s) or device(s) must not have dependencies to the selected system, or the switch has to be able to be done seamlessly. |
| Events flow | 1. Click on the “switch” button |
| Post-condition | Alternate system takes the control. |
| Exception | * If the alternate system is broken as well, it may generate a fatal error of the system * If other system(s) or device(s) have dependencies to the selected systems and the switch cannot be made seamlessly, display a warning and information about the dependencies, and offer to override. |

|  |  |
| --- | --- |
| Name | Get information on a system(s)/device(s) |
| Actors | Crew / Flight control |
| Description | User may seek information about any device or system on the spacecraft |
| Precondition | - Requested system(s)/device(s) have been selected |
| Events flow | 1. Specify query  2. Click on the “GetInfo” button |
| Post-condition | Information about the selected system appears on the screen. |
| Exception |  |

|  |  |
| --- | --- |
| Name | Select system(s)/device(s) |
| Actors | Crew / Flight control |
| Description | User can select any system or device in order to issue commands |
| Precondition |  |
| Events flow | 1. Select the requested part(s) |
| Post-condition | The chosen part is selected. |
| Exception | The chosen part is still not selected. |

* 1. Sequence diagrams
     1. Fault recovering
     2. Safe response in case of hazardous conditions
     3. Critical failure
* **Non-functional requirements**
  1. Identified quality attributes
     1. Testability
     2. Availability
  2. Improvised quality attributes
     1. Avaibility
     2. Reliability
     3. Resilience
     4. Response time
* **Usability analysis & design**

1. Preliminary user interface design

Displaying information continuously 🡪 multiple screens but no tabs

Overview of the spacecraft always available 🡪 list of systems & subsystems with information (monitoring values, temperature, pressure)

Professional interface 🡪 non useful features like displaying spacecraft screens. Crew would be formed to be on the spacecraft and do not need to have a visualization of the spacecraft. Title of device or system is enough

Locate the fault 🡪 displaying left panel with spacecraft scheme. List of the systems appearing on the tree should appears on the scheme too. If an alert appears on one system, we should be able to localize it geographically on the scheme.

Keep the control of the spacecraft with safety, observability & commandability 🡪 bottom panel with buttons (shutdown, restart, backup and recovery data)

1. Discussion on usability scenarios

**Conclusion**

References

**Web Sites**

**Articles**

**Books**