FDIR

Spacecraft fault protection system

Project 1
Part 2

Euro Team

Alauzet Pierre, Ahvenniemi Mikko,
Colin Julien, Starck Benoit

KAIST

CS554 - Design for Software & Systems

October 1st, 2009

TABLE OF CONTENTS

1. Domains identification
2. Requirements
3. Context diagram
4. Problem diagram

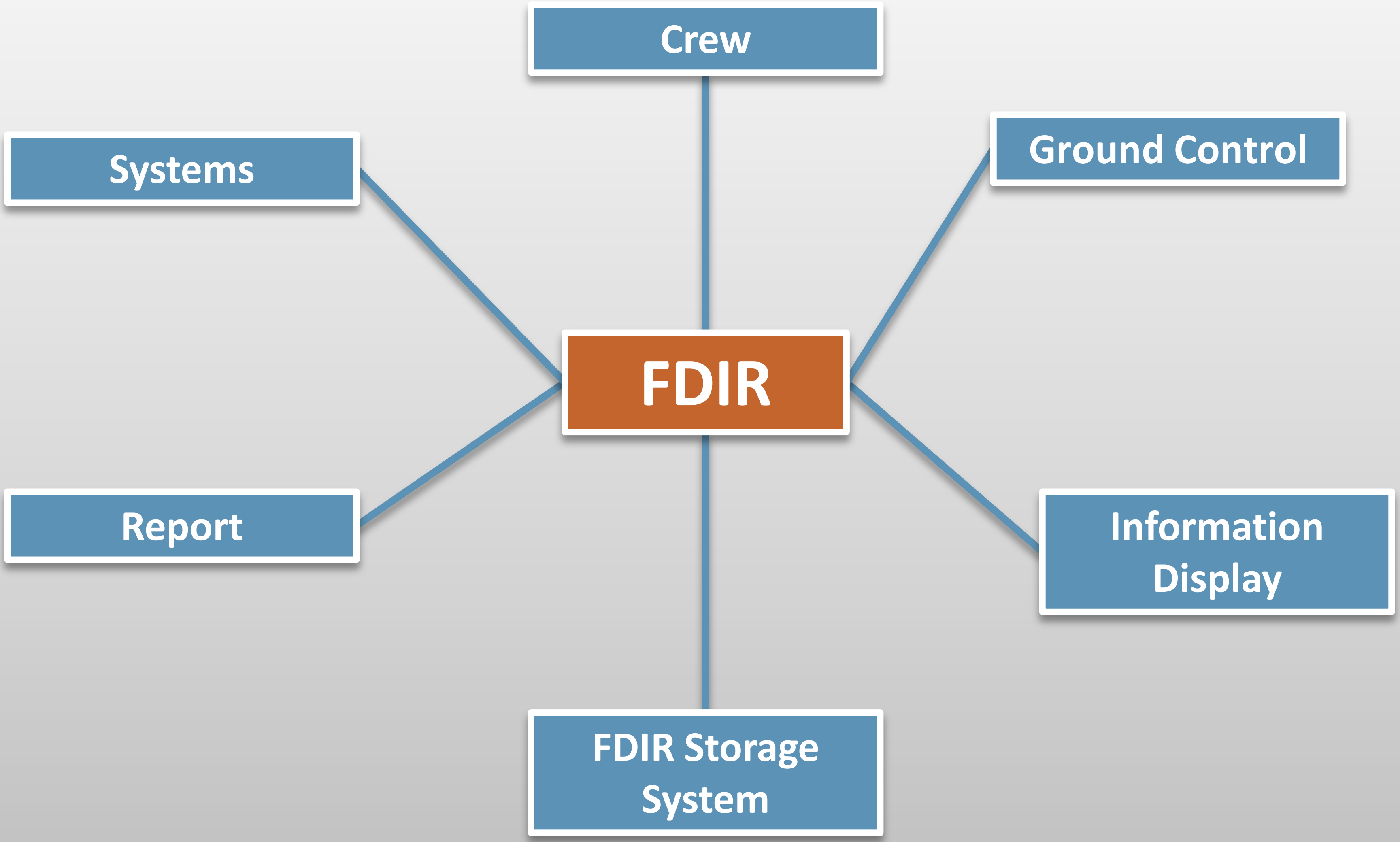
DOMAINS IDENTIFICATION

- FDIR storage system
- Crew
- Information display
- Ground control
- Systems
- Report

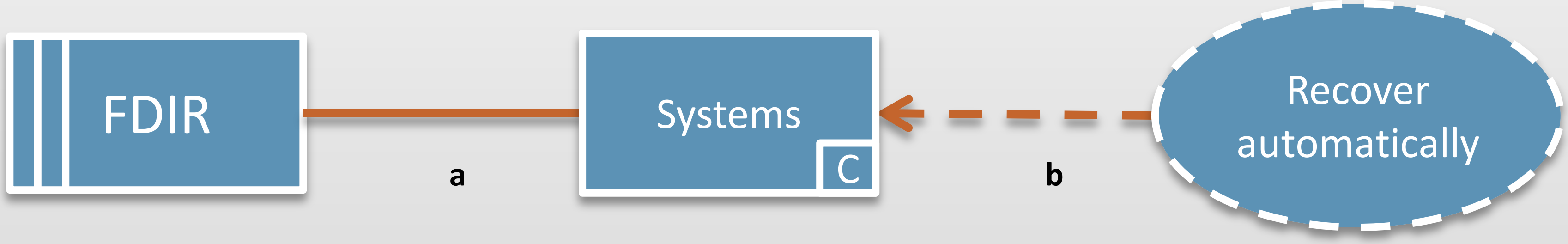
REQUIREMENTS

1. Automatic recovery to failure
2. Manual control of FDIR
   - ❑ Crew is able to shutdown part of the system
   - ❑ Crew is able to restart part of the system
   - ❑ Crew is able to switch to a spare system
3. Displaying information continuously
4. Collect system data to data storage
5. Information retrieval
6. Providing failure localization
7. Response in case of unresolvable failure

CONTEXT DIAGRAM



AUTOMATIC RECOVERY TO FAILURE



a: FDIR! {backup, restart,shutdown}
Systems!{return command status}

b: {Functional, non functional, broken}

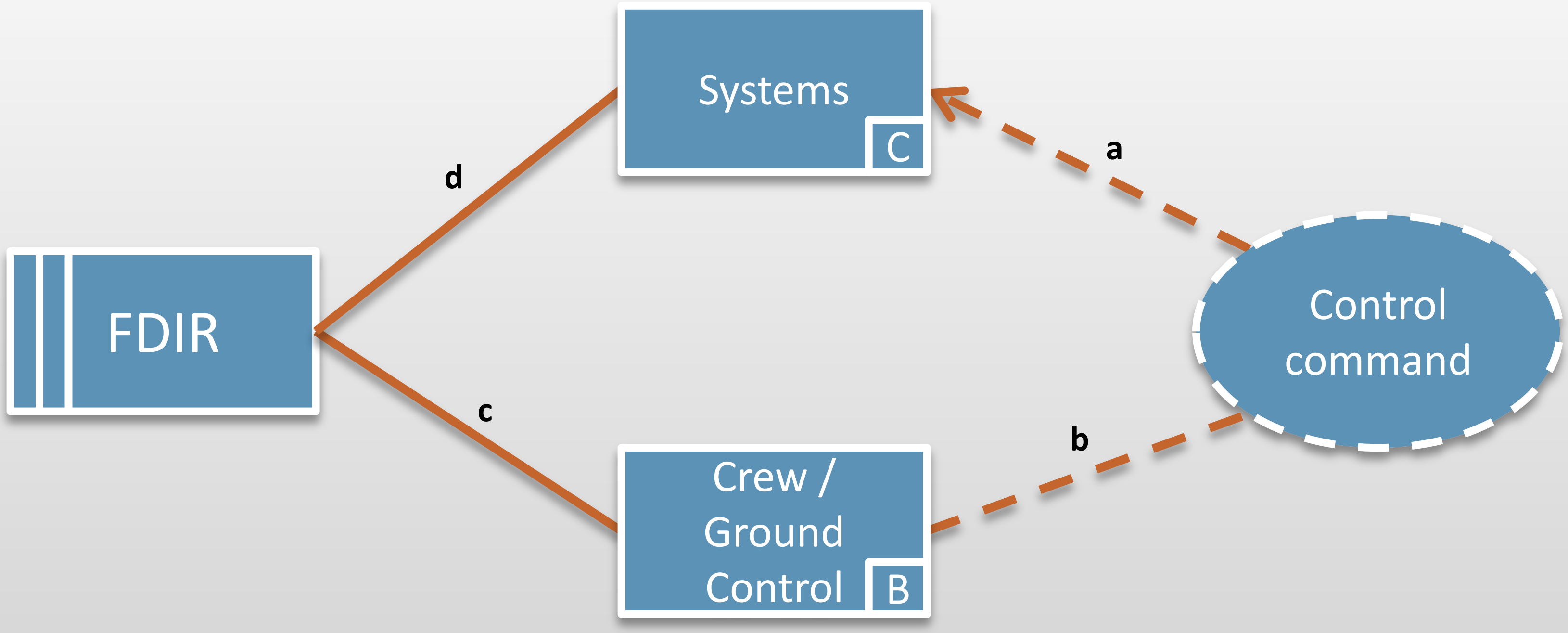
Required behaviour problem frame

AUTOMATIC RECOVERY TO FAILURE (CONT.)

- The FDIR can launch a restart of the system automatically, in the goal to recover in case of a failure.

- The systems, during these operations, return their status to the FDIR.

MANUAL CONTROL OF FDIR



c: C/GC! {Do shutdown,Do restart,Do switch to backup}
FDIR!{return command status}

b: C/GC! {Shutdown, Restart, Switch to backup}

d: FDIR! {Issue Shutdown,Issue Restart, Issue Switch}
Systems! {Return command status, No return}

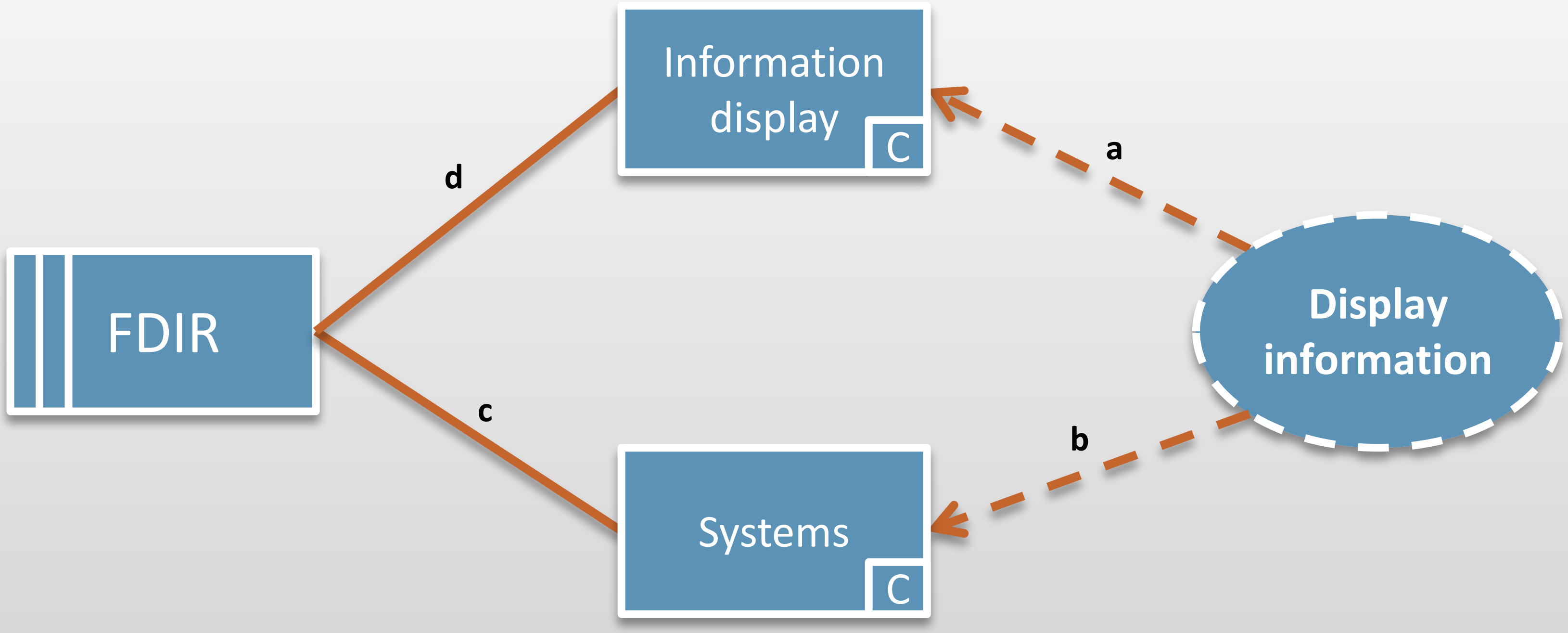
a: System! {Functional, malfunctioning, broken}

Commanded behaviour problem frame

MANUAL CONTROL OF FDIR (CONT.)

- FDIR has to provide interface for issuing manual commands from the crew or ground control at anytime
- FDIR is able to send commands (shutdown,restart,switch to a different backup) to the spacecraft’s several systems
- The systems has to remain available and responding while processing commands
- FDIR must be able to multitask commands

DISPLAYING INFORMATION CONTINUOUSLY



c: Systems! {send value/no value}

b: Systems!{functionnal/not funct. proper./broken}

d: FDIR! {display in tol/out-of-tol/no resp}

a: Information display!{console}

Display problem frame

DISPLAYING INFORMATION CONTINUOUSLY (CONT.)

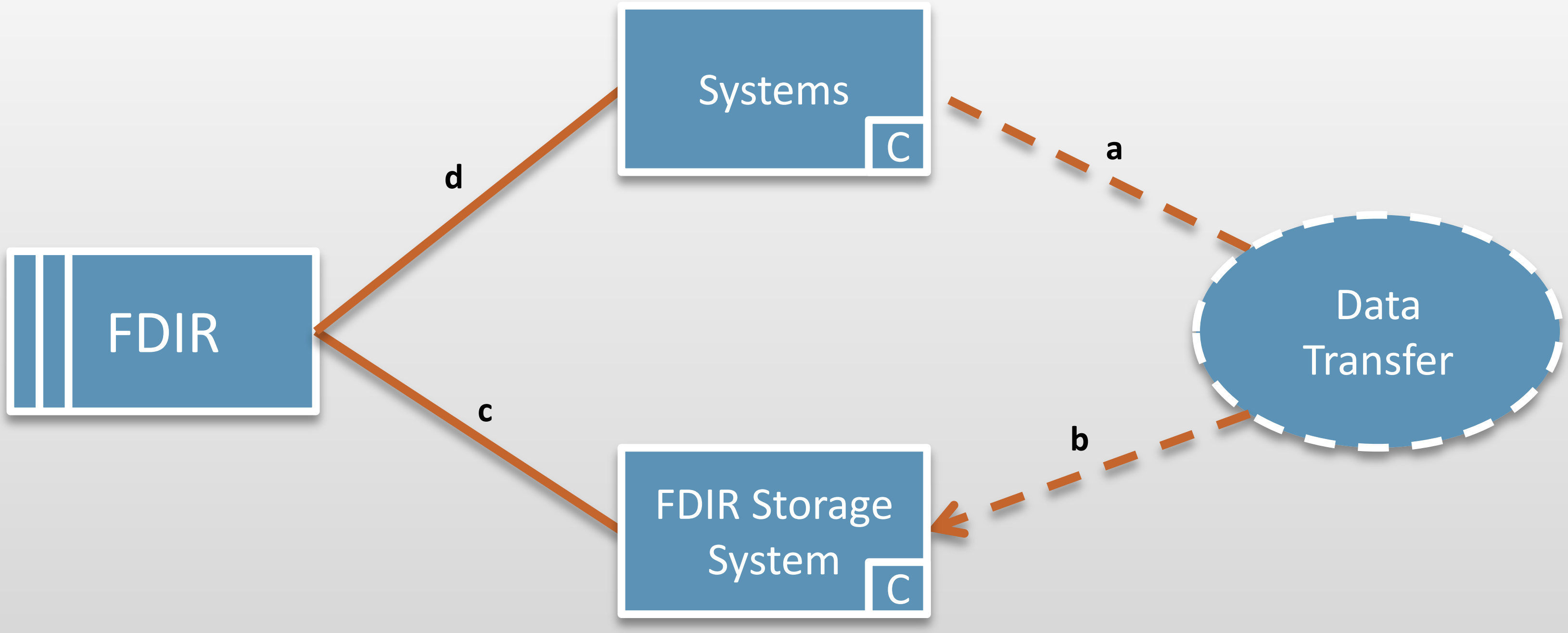
❑ FDIR should display continuous information about state of the systems

❑ FDIR has to interprate monitored values from each space craft system

❑ Return it into a standard message displayed on the FDIR console

❑ Considering received message, the FDIR or the crew should be able to understand what was the current state of the systems

COLLECT SYSTEMS DATA TO DATA STORAGE



c: FDIR! {System ID, state value, time}

b: {System data}

d: Systems! {value, no value}

a: Systems! {System ID, state}

Display problem frame

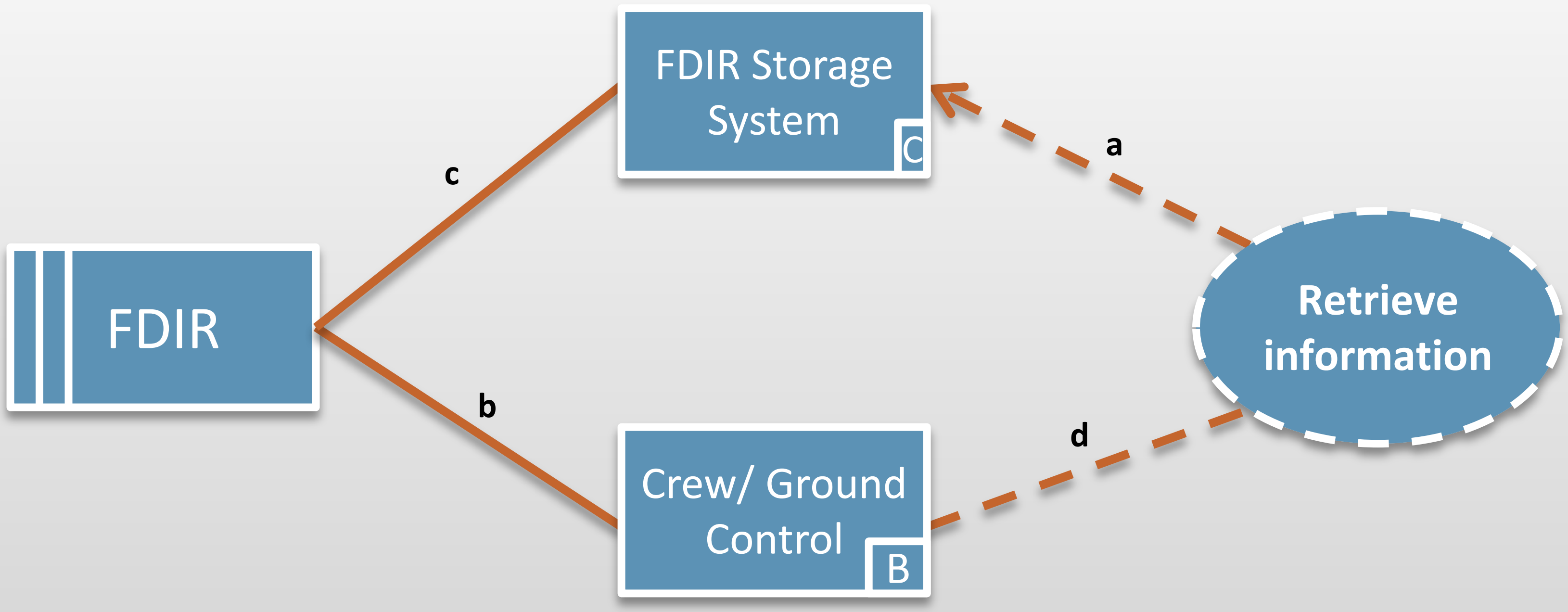
COLLECT SYSTEMS DATA TO DATA STORAGE(CONT.)

- State values are collected from the systems at regular intervals

- The FDRI receives the data and stores it with a timestamp to the FDRI Storage System for further use

- When data storage is centralized it doesn't matter if some systems go down, because data analysis can still be done on the stored data.

INFORMATION RETRIEVAL



c:FDIR! {query}
FDIR SS! {return data}

b:Crew!{search}
FDIR!{display}

a: FDIR SS!{status data}

d: Crew!{search}

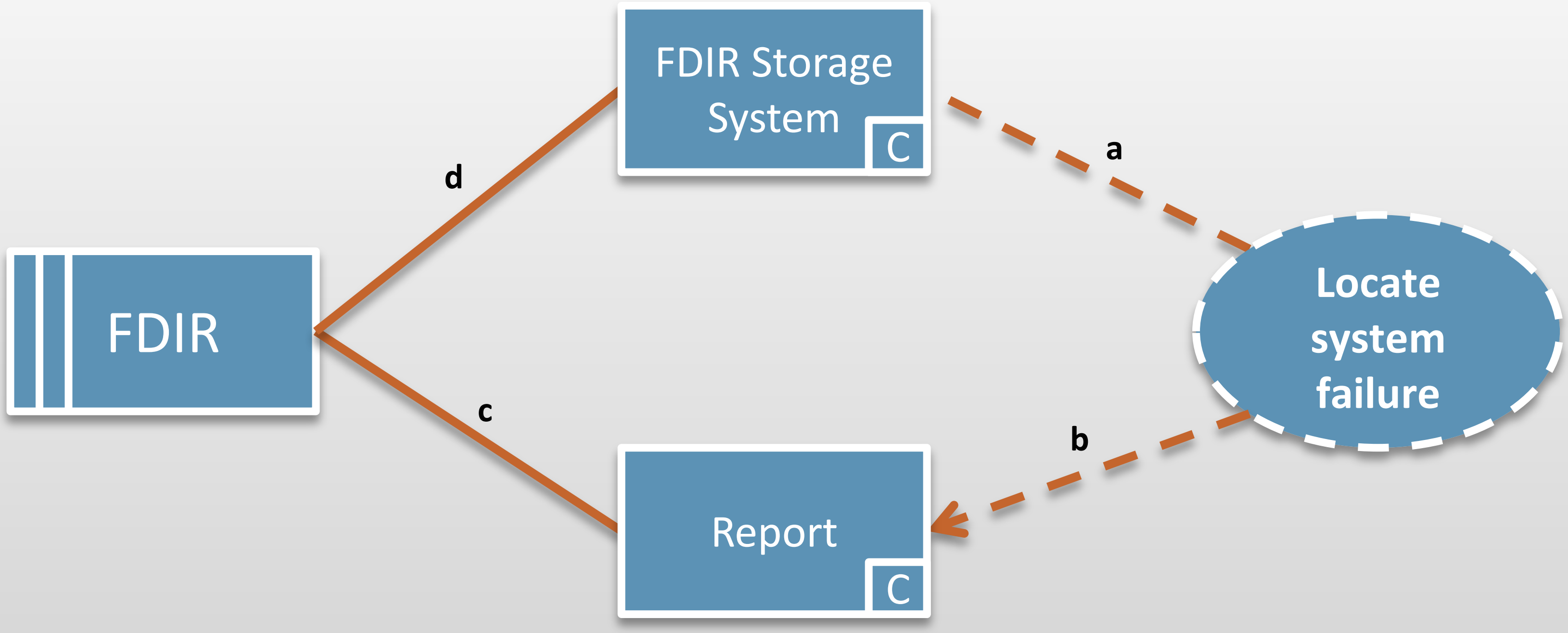
Commanded behaviour problem frame

INFORMATION RETRIEVAL (CONT.)

- FDIR executes query, and the FDIR Subsystems reply

- The crew or ground control can search data, and the FDIR displays its

PROVIDING FAILURE LOCALIZATION



c: FDIR! {write failure location, write type of failure}

b: Report! {failure data}

d: FDIR Storage System! {send device, send value}

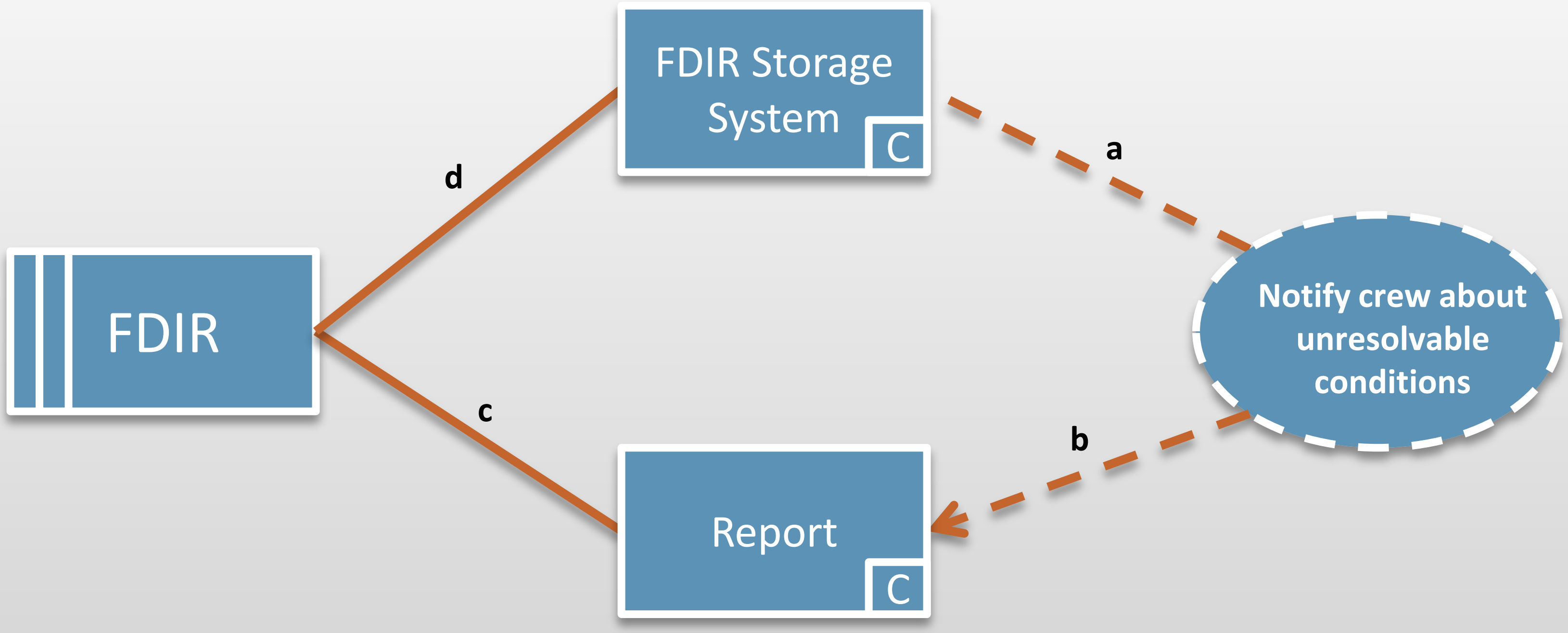
a: FDIR Storage System! {device, value, time}

Transformation problem frame

PROVIDING FAILURE LOCALIZATION(CONT.)

- The FDIR Storage System contains the collected values or data from devices
- FDIR checks the inputs from the storage system, and analyses these inputs to determine failure location.
- Failure location is written into a report.

RESPONSE IN CASE OF UNRESOLVABLE CONDITIONS



c: FDIR! {write notification, write unresolvable conditions}

b: Report! {notification}

d: FDIR Storage System! {send device, send value}

a: FDIR Storage System! {device, value, time}

Transformation problem frame

RESPONSE IN CASE OF UNRESOLVABLE CONDITIONS (CONT.)

- This case is achieved when automatical recovering failed

- The FDIR Storage System contains the collected values or data from devices

- FDIR checks the inputs from the storage system, and analyses these inputs to determine if unresolvable condition has been reached.

- Informations about unresolvable condition is written into a report sent as a notification to the crew members

FDIR

Spacecraft fault protection system

Project 1
Part 2

Euro Team

Alauzet Pierre, Ahvenniemi Mikko,

Colin Julien, Starck Benoit

KAIST

CS554 - Design for Software & Systems

October 1st, 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.

2. [Jac05] **Michael Jackson**, *Problem frames and software engineering*, Information and Software Technology, Special Issue: 1st Int Workshop on Advances and Applications of Problem Frames, K. Cox, et al. eds, Vol. 47 No. 14, pp. 903-912, Nov. 2005.