FAA Requirements Engineering Management Handbook

6. Revise the Architecture to Meet Implementation Constraints

Kansas State University

Steps in the REMH

1. Develop the System Overview
2. Identify the System Boundary
3. Develop the Operational Concepts
4. Identify the Environmental Assumptions
5. Develop the Functional Architecture
6. **Revise the Architecture to Meet Implementation Constraints**
7. Identify System Modes
8. Develop the Detailed Behavior and Performance Requirements
9. Define the Software Requirements
10. Allocate System Requirements to Subsystems
11. Provide Rationale

Architecture Revision: Goals

What are we trying to achieve with this step in the requirements engineering process?

- Iteratively update the functional architecture from Section 5 to arrive at the final architecture
- Take into account additional constraints that were not directly implied/uncovered by use-case and functional architecture process
  - Implementation constraints (e.g., available hardware components, the need to deploy on a particular platform)
  - Safety constraints
  - The need to integrate with legacy systems
- Note: if this step is not taken, we might be in an unfortunate position where we have two architectures (the "abstract functional" one, and the real final one) and we need to continuously map from functional architecture to final architecture

Architecture Revision: Artifacts

What artifacts should we produce as a result of this step?

- Revised functional architecture that takes into account:
  - Component failure / safety
  - Legacy systems
  - Implementation constraints
- Revised architecture should give us a framework for organizing detailed requirements

6 Revise the Architecture to Meet Implementation Constraints

6 Revise the Architecture to Meet Implementation Constraints: The organization produced through functional analysis is a logical architecture that may not take into account additional constraints, such as the need to satisfy system safety requirements, integrate with legacy systems, or to meet implementation constraints imposed by a particular platform. This practice describes an iterative process that starts from the previously developed functional architecture and leads to an architecture that addresses these concerns. This architecture is then used as the framework for organizing the detailed requirements.

6.1 If implementation constraints cannot be satisfied with the ideal functional architecture that is developed during functional analysis, **modify the functional architecture as necessary**, and use the final system architecture as the framework for organizing the detailed requirements

6.2 When modifying the functional architecture to accommodate implementation constraints, **keep the final system architecture as close to the ideal functional architecture as possible.**

6.3 Revise the system overview to reflect any changes in how the system interacts with its environment, any new functionality added to the system to satisfy the implementation constraints, or any changes in system goals.

6.4 Revise the operational concepts to reflect any changes in how operators or other systems interact with the revised system architecture.

6.5 Review the use cases to identify steps where exceptions to the nominal behavior could occur. Develop exception cases to identify how each exception will be handled.

6.6 If an exception can only occur at a few points, **link** those **steps to the exception case**. If the exception can occur at almost any point, use the exception case precondition to identify when the exception case occurs.

6.7 Revise the system boundary to reflect any changes in the monitored and controlled variables.

6.8 Identify and **document** any **new or changed environmental assumptions** for the revised functional architecture.

6.9 Revise the dependency diagrams to show the revised functional architecture.

6.10 Revise any high-level requirements affected by the changes in the revised functional architecture.

6.1 Modify the Architecture to Meet Implementation Constraints

If implementation constraints cannot be satisfied with the ideal functional architecture that is developed during functional analysis, modify the functional architecture as necessary…

- Final architecture allows organization of detailed requirements
- Diagrams generated in this phase are a "graphical table of contents" for detailed requirements
- Consider that the functional architecture may or may not need significant changing

6.2 Keep Final System Architecture Close to Ideal Functional Architecture

But on the other hand… don't stray too far from the ideal functional architecture.

- Original functional architecture was developed through analyzing the problem domain without being encumbered by the solution domain.
- Problem domain is less likely to change than implementation constraints
  - Thus, the closer the final architecture is to the ideal functional architecture, the more stable it will be
  - It's desirable to minimize differences between ideal functional architecture and the final functional architecture.

The Impact of Designing for Safety

Let's focus on how the architecture and requirements document might change as we incorporate notions of safety/reliability into our work

- Safety-critical systems need very high levels of reliability
  - Even though this often conflicts with keeping costs reasonable
- In avionics, the ARP 4761 process involves
  - Performing Functional Hazard Assessment (FHA) that identifies high-level system hazards.
  - FHA is then used during Preliminary System Safety Assessment (PSSA) to determine if could contribute to the realization of these hazards.
  - If so, we will have relevant safety requirements on the determined by the PSSA.

Functional Hazard Analysis

Isolette Example Hazard

H1. Prolonged exposure of Infant to unsafe heat or cold

Classification: catastrophic
Probability: $<10^{-9}$ per hour of operation

What are the ways that this hazard could be realized?

PSSA of Isolate System

The Isolette system PSSA (not the Thermostate itself) identifies several ways this hazard could be realized.

- The Thermostat could fail and turn the Heat Source on or off for too long.
- The Temperature Sensor could provide an incorrect temperature to the Thermostat.
- The Operator Interface could provide the wrong Desired Temperature Range to the Thermostat.
- The Heat Source could fail, either by remaining on or off for too long or by failing to provide sufficient heat to maintain the Desired Temperature Range.

What constraints do the initial reliability requirement of the Isolette impose on the reliability of the components mentioned above? We can reason about such things using a ***fault tree.***

Example Isolette Fault Tree

Fault Tree Analysis (FTA) is a top down, deductive failure analysis in which an undesired state of a system is analyzed using boolean logic to combine a series of lower-level events

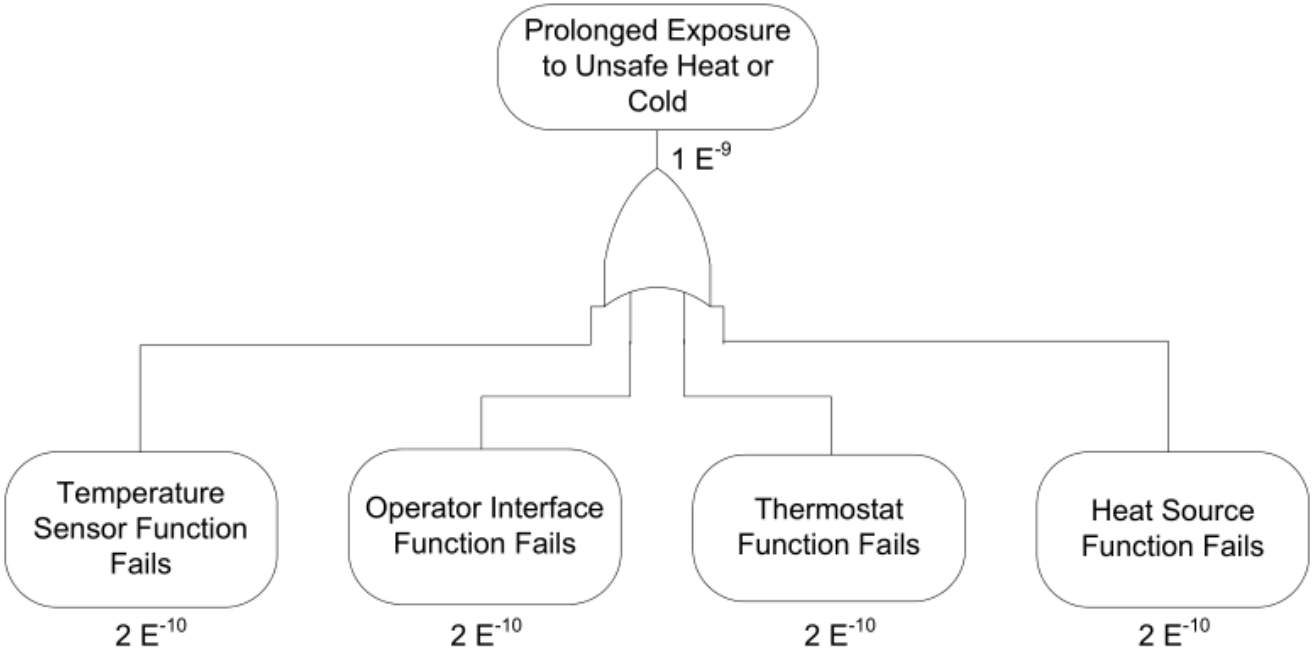


Figure 5. Initial Isolette Fault Tree

The fault tree derived during the PSSA of the Isolette system is shown above. Since each System Function could cause hazard H1, each function is assigned a probability of failure of less than 2 x 10-10 per hour of operation.

Assessment

- Developing individual components that achieve this level of reliability would be very expensive.
- Even designing a Thermostat that provides this level of reliability would contradict goal G2 to produce the Thermostat at minimal manufacturing cost.
- A less costly solution is to add a monitor that activates an alarm if the Current Temperature in the Isolette falls below or rises above a safe level.
  - Note: we often refer to this architecture strategy as a *safety system.*

Which failures would this protect against?

Ans: against a failed Thermostat Function and a failed Manage Heat Source Function (but not against a misleading Temperature Sensor Function or a misleading Operator Interface Function)

Revised Fault Tree

Using a safety system…

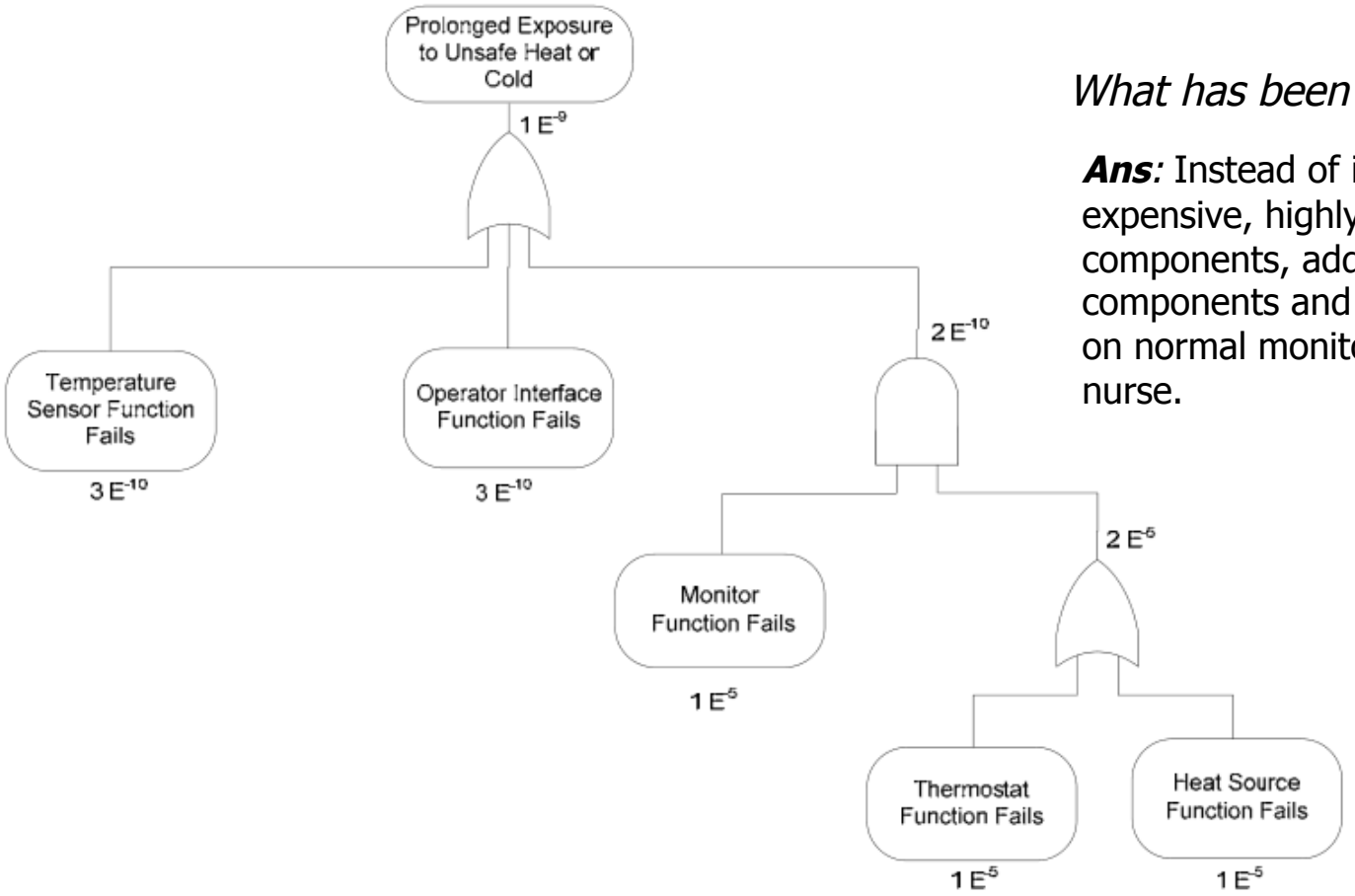


Figure 6. Revised Isolette Fault Tree

What has been achieved?

***Ans**:* Instead of incredibly expensive, highly-reliable components, add cheaper components and an alarm plus rely on normal monitoring procedures of nurse.

PSSA Yields Derived Safety Reqs

This change in architecture strategy gives rise to new high-level requirements

- The Isolette shall include an independent Thermostat Function that maintains the Current Temperature within the Desired Temperature Range inside the Isolette.

  Rationale: The Desired Temperature Range will be set to the ideal range by the Nurse based on the Infant's weight and health. The Thermostat should maintain the Current Temperature within this range under normal operation.

  Allowed probability of failure: $<10^{-5}$ per hour.

PSSA Yields Derived Safety Reqs

This change in architecture strategy gives rise to new high-level requirements

- The Isolette shall include an independent Monitor Function that activates an Alarm within 5 seconds whenever

  - the Current Temperature inside the Isolette falls below or rises above the Alarm Temperature Range.

  - the Current Temperature is flagged as invalid.

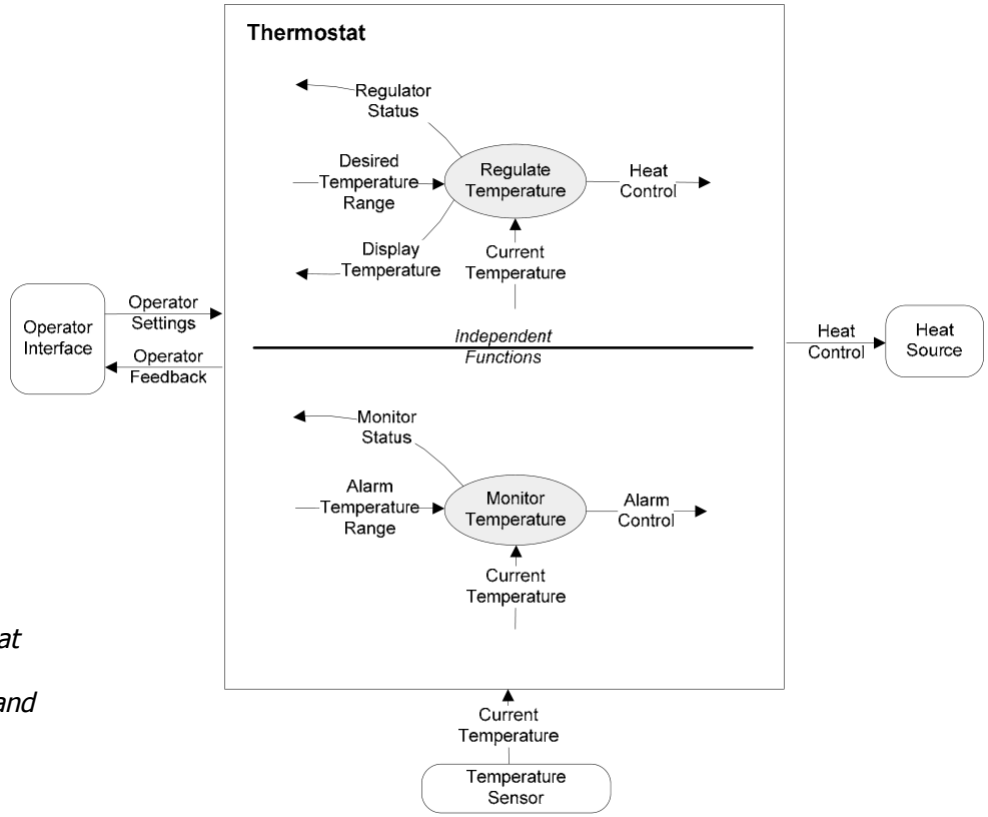
  Rationale: The Alarm Temperature Range will be set by the Nurse based on the Infant's weight and health. The Infant should be removed from the Isolette within 15 seconds after the Current Temperature falls below or rises above the Alarm Temperature Range. With the normal monitoring provided by the Nurse, this can be accomplished within 10 seconds, leaving 5 seconds for the system to activate the Alarm. Activating the Alarm in less time is desirable.

  Allowed probability of failure: $<10^{-5}$ per hour.

To meet these requirements while minimizing manufacturing costs, the Isolette designers proposed a design in which the monitor function is implemented within the thermostat itself. After extending the PSSA to this design, this was acceptable, providing the independence of the monitor is maintained.

Changes Ripple Throughout

- Note the "ripple" of changes this causes through the entire requirements document:



To avoid confusion, the Thermostat Function was renamed as the Regulator Function, where the Thermostat is now considered the combination of the Regulator and Monitor Functions.

Figure 7. Revised Thermostat Dependency Diagram

6.3 Revise the System Overview

Revise the system overview to reflect any changes in how the system interacts with its environment, any new functionality added to the system to satisfy the implementation constraints, or any changes in system goals.

- The system boundary has been modified
  - New monitored variable: "Alarm Temperature Range"
  - New controlled variable: "Monitor Status"
  - Thermostat status renamed to Regulator Status
- Revisions are needed to:
  - System overview
    - Setting alarm temperature range and activating the alarm
  - System boundary
  - Operational concepts
  - Environmental assumptions
  - System goals
    - Warn the clinician if the infant becomes too hot or cold

6.4 Revise the Operational Concepts

If the interaction with other systems or system operators was changed to meet implementation constraints, the operational concepts should also be updated.

Isolette Example -- what changes are needed to the operational concepts?

- Entering bounds for alarm?
- Raising and responding to alarm?

6.5 Develop Exception Cases

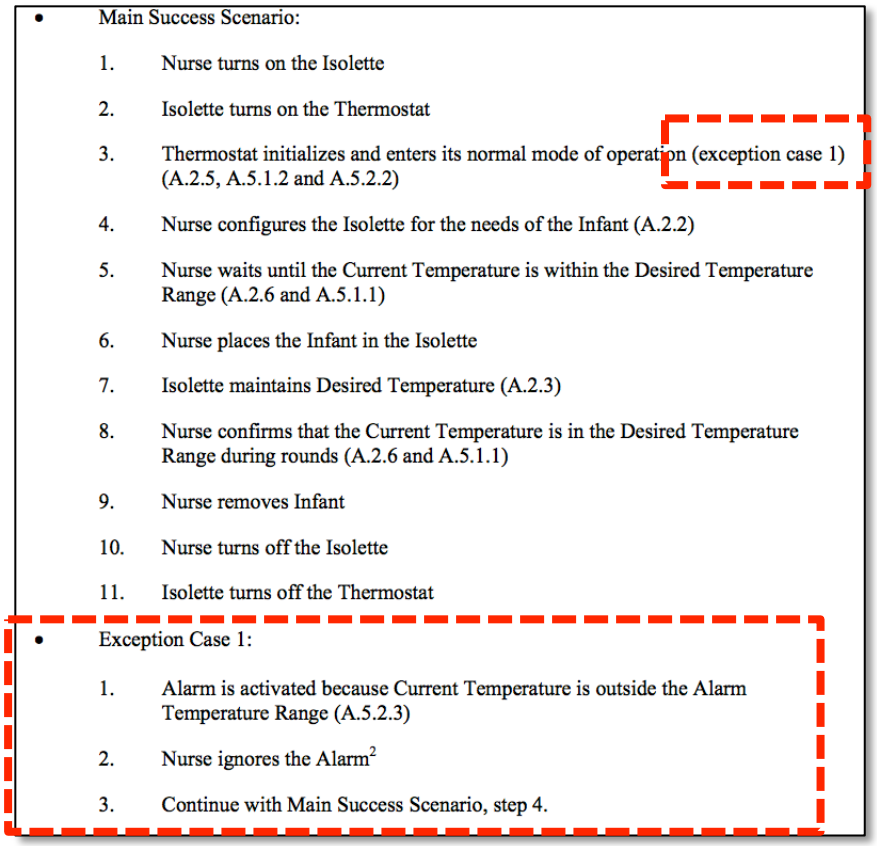
Since the PSSA initiated consideration of how failures should be handled, this is also an appropriate time to go back and extend the use cases with exception cases. As the use cases are reviewed and new functionality is added, steps at which exceptions to the nominal (sunny day) behavior might occur should be identified. Exception cases should be defined, describing how each exception will be handled.

Isolette Example -- what are examples of exception use cases?

- Failure to maintain desired temperature
- Failure to maintain safe temperature
- ...*others will be revealed in subsequent lectures*

6.6 Link Exception Cases to Use Cases

- If an exception can only occur at a few steps in a use case, those points should be linked to the exception cases



- Main Success Scenario:
  1. Nurse turns on the Isolette
  2. Isolette turns on the Thermostat
  3. Thermostat initializes and enters its normal mode of operation (exception case 1) (A.2.5, A.5.1.2 and A.5.2.2)
  4. Nurse configures the Isolette for the needs of the Infant (A.2.2)
  5. Nurse waits until the Current Temperature is within the Desired Temperature Range (A.2.6 and A.5.1.1)
  6. Nurse places the Infant in the Isolette
  7. Isolette maintains Desired Temperature (A.2.3)
  8. Nurse confirms that the Current Temperature is in the Desired Temperature Range during rounds (A.2.6 and A.5.1.1)
  9. Nurse removes Infant
  10. Nurse turns off the Isolette
  11. Isolette turns off the Thermostat
- Exception Case 1:
  1. Alarm is activated because Current Temperature is outside the Alarm Temperature Range (A.5.2.3)
  2. Nurse ignores the Alarm[2]
  3. Continue with Main Success Scenario, step 4.

Here, we deal with the special case where the alarm may come on because the Isolette is not yet "warmed up".

6.6 Link Exception Cases to Use Cases

- If an exception can occur almost anywhere, specify when it can occur in a precondition

A.2.4 EXCEPTION CASE: FAILURE TO MAINTAIN SAFE TEMPERATURE.

This exception case describes how the Thermostat and Nurse respond when the Isolette is unable to maintain Current Temperature within the Alarm Temperature Range.

- Related System Goals: G2
- Primary Actor: Thermostat
- Precondition:
  - The Isolette and Thermostat are turned on
  - The Current Temperature is within the Alarm Temperature Range
  - The Alarm is off
- Postcondition:
  - The Isolette and Thermostat are turned on
  - The Current Temperature is within the Desired Temperature Range
  - The Alarm is off

Continued on next slide…

6.6 Link Exception Cases to Use Cases

- If an exception can occur almost anywhere, specify when it can occur in a precondition

Continued from previous slide…

- Main Success Scenario:

  1. Current Temperature falls below or rises above the Alarm Temperature Range
  2. Thermostat activates the Alarm (A.5.2.3)
  3. Nurse responds to the Alarm and sees that the Display Temperature is in the Alarm Temperature Range (A.5.1.1)
  4. Nurse removes Infant from the Isolette
  5. Nurse corrects the problem, e.g., closing an open door (alternate course 1)
  6. Nurse waits until the Display Temperature is within the Desired Temperature Range (A.2.6 and A.5.1.1)
  7. Nurse places Infant back in the Isolette

- Alternate Course 1:

  1. Nurse is unable to correct the problem
  2. Nurse obtains another Isolette
  3. Nurse starts normal operation of the new Isolette (A.2.1)

6.6 Link Exception Cases to Use Cases

- Consider whether or not the given exception can contribute to a system hazard identified by the FHA.

6.7 Revise the System Boundary

- If the revised functional architecture created new monitored or controlled variables, the system boundary should be updated

For the Isolette Thermostat, the Alarm Temperature Range monitored variable and the Alarm Control controlled variable were added, and the Thermostat Status controlled variable was replaced by the Regulator Status and the Monitor Status controlled variables.

6.8 Document changes to the Environmental Assumptions

- New environmental assumptions need to be identified and documented
- With each additional variable, environmental assumptions should be re-examined.

- Example: With the new alarm, the temperature range should be documented, along with supporting rationale.

6.9 Revise Dependency Diagrams

- Dependency diagrams should be updated as well.

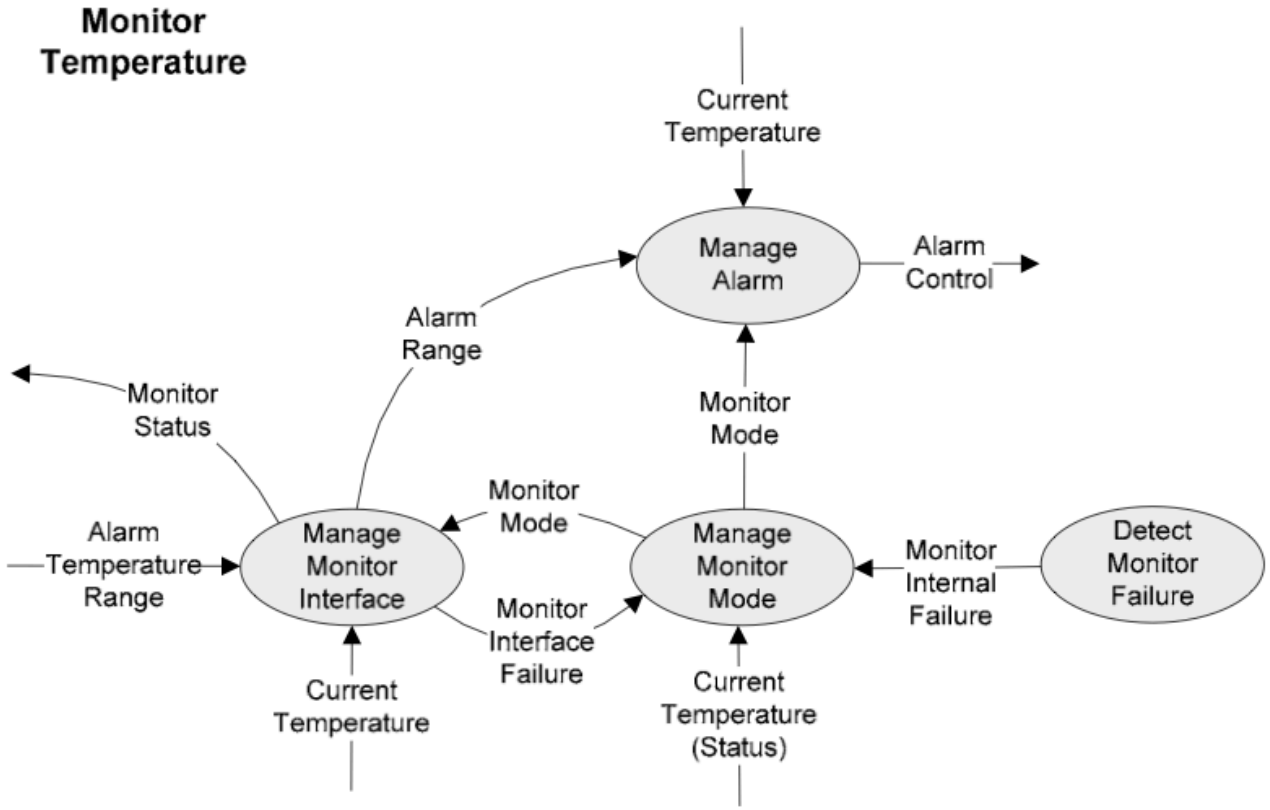


Figure 9. Monitor Temperature Dependency Diagram

6.10 Revise High-Level Requirements

- Any high-level requirements should be updated if the change in the system functional architecture affects them

Summary

- Update the functional architecture to reflect implementation constraints
- Update supporting documentation accordingly

For You To Do

Acknowledgements

- The material in this lecture is based almost entirely on
  - *FAA* DOT/FAA/AR-08/32*, Requirements Engineering Management Handbook.* David L. Lempia & Steven P. Miller.