

# FAA Requirements Engineering Management Handbook

## 2. Identify the System Boundary

---

*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

# System Boundary: Goals

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

- Provide a sound understanding of what is inside and what is outside the system
- Avoid duplication of / conflicting with higher-level requirements
  - Particularly important when working with multiple development entities
  - If the boundary is imagined to be larger than it actually is...
    - ...you may be writing requirements for entities that are someone else's control
  - If the boundary is imagined to be smaller than it actually is...
    - ...you will omit writing requirements for some entities for which you are responsible

# System Boundary: Artifacts

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

- Set of monitored variables
  - “Inputs” from external environment
- Set of controlled variables
  - “Outputs” to external environment
- Set of environmental variables
  - External variables that are neither directly sensed nor controlled

# 2 Identify the System Boundary

**2 Identify the System Boundary:** Develop a clear definition of the boundary between the system and its environment. This provides a solid understanding of what lies within the system to be built and what lies within a larger environment. This is done by identifying a set of variables in the environment the system will monitor and control.

**2.1 Define the system boundary early** in the requirements engineering process by identifying a preliminary set of monitored and controlled variables.

**2.2 Choose environmental variables** that exist in the environment independently of the system to be developed.

**2.3 Choose controlled variables** that are under the direct control of the system being specified.

**2.4 Choose monitored variables** that are being directly sensed by the system being specified.

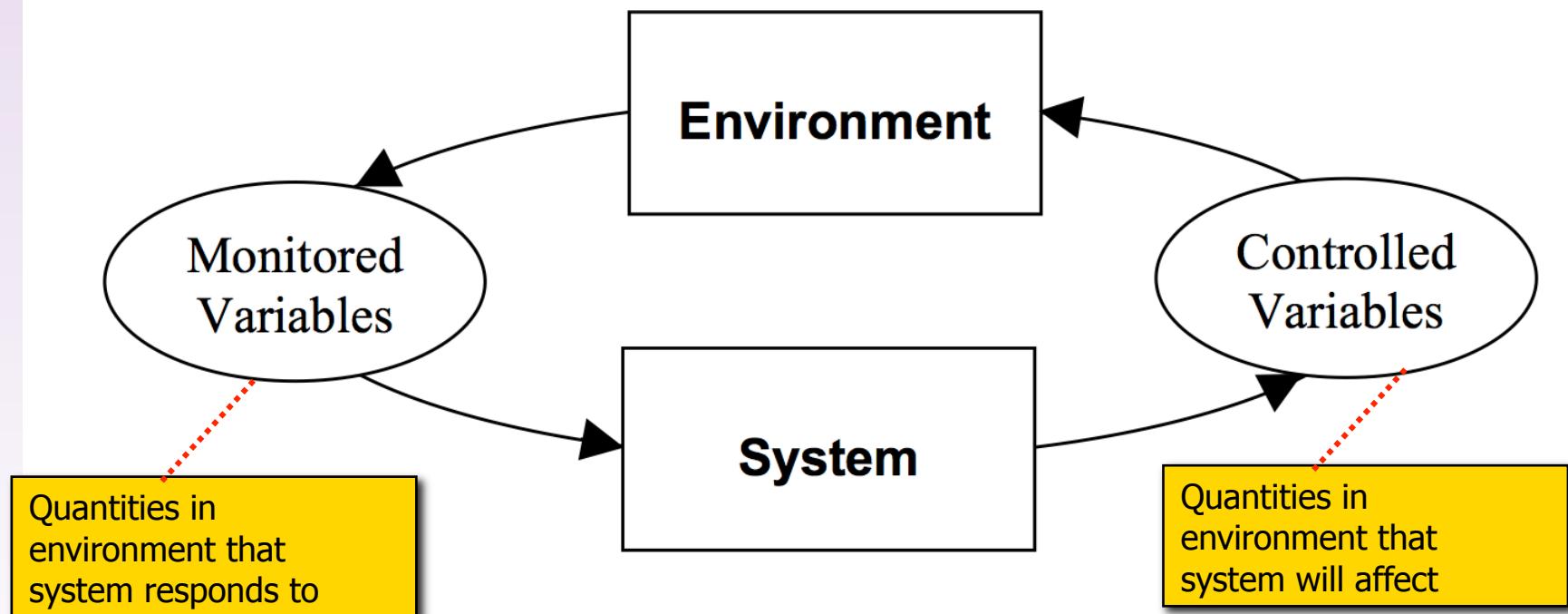
**2.5 Ensure the monitored and controlled variables are as abstract as possible** and do not include implementation details.

**2.6 Avoid incorporating details of the operator interface in the monitored and controlled variables.** Instead, define monitored or controlled variables that describe the information to be conveyed independent of its presentation format.

**2.7 Completely define all physical interfaces to the system,** including definitions for all discrete inputs, all messages, all fields in a message, and all protocols followed.

# The System and its Environment

How should the system boundary be described? One method is to view the system as a component that interacts with its environment through monitored and controlled variables.



*The purpose of the system is to maintain a relationship between the monitored and controlled variables that achieves the system goals.*

## 2.1 Identify the System Boundary Early

### Advice

- Variable identification should be started early
  - Even if everything isn't 100% clear
- A notion of the system boundary simplifies following tasks
  - Identification of the boundary may raise additional, important questions

# Isolette Example Variables

Name	Type	Physical Interpretation
Current Temperature	Monitored	Current air temperature inside Isolette
Operator Settings		Thermostat settings provided by the operator
Desired Temperature Range		Desired range of Isolette temperature
Lower Desired Temperature	Monitored	Lower value of the Desired Temperature Range
Upper Desired Temperature	Monitored	Upper value of the Desired Temperature Range
Operator Feedback		Information provided back to the operator
Display Temperature	Controlled	Displayed temperature of the air in Isolette
Heat Control	Controlled	Command to turn the heat source on or off

Early Monitored and Controlled Variables from the Isolette Example

An early listing of environment variables will tend to raise questions...

- What are the limits of the Desired Temperature Ranges?
- Is the temperature stated in degrees Centigrade or Fahrenheit?

## 2.2 Choose Environmental Variables

### Advice

- Do not expect to perfectly identify all variables immediately
  - Unless you're replacing an existing system
- Environment variables should:
  - **Exist outside the system**
  - **Exist independent of the system**
- Ask yourself the question: would the variable exist even if the system were eliminated? (*the answer should be "yes"*).

## 2.3 Choose Controlled Variables

Controlled variables...

- Should correspond exclusively to things that the system can directly control
- In the isolette example, air temperature *is not* a controlled variable
  - The air temperature cannot be directly controlled
  - Instead the Heat Control is the controlled variable

# Example Controlled Variables

Refining the description of *controlled variables* for the Isolette Thermostat

Table A-4. Thermostat Controlled Variables for Heat Source

Name	Type	Range	Units	Physical Interpretation
Heat Control	Enumerated	Off, On		Command to turn Heat Source on and off

Table A-6. Thermostat Controlled Variables for Operator Interface

Name	Type	Range	Units	Physical Interpretation
Operator Feedback				Information provided back to the operator
Regulator Status	Enumerated	Init, On, Failed		Status of the Thermostat Regulator Function
Monitor Status	Enumerated	Init, On, Failed		Status of the Thermostat Monitor Function
Display Temperature	Integer	[68..105]	°F	Displayed temperature of Isolette
Alarm	Enumerated	Off, On		Command to turn Alarm on or off

## 2.4 Choose Monitored Variables

Monitored variables...

- Should correspond to physical quantities the system can sense

Advice

- Choosing the correct level of abstraction is key
- Consider a system to determine airplane altitude:
  - One might consider defining “Actual Altitude” as a monitored variable, but different parts of the plane may be at different altitudes!
  - The altitude monitoring system might have individual altitude sensors as monitored variables, and the system combines this to compute a projected altitude

# Example Monitored Variable

Refining the description of *monitored variables* for the Isolette Thermostat

Table A-3. Thermostat Monitored Variables for Temperature Sensor

Name	Type	Range	Units	Physical Interpretation
Current Temperature	Real	[68.0..105.0]	°F	Current air temperature inside Isolette
	Status	•Invalid, Valid		

Table A-5. Thermostat Monitored Variables for Operator Interface

Name	Type	Range	Units	Physical Interpretation
Operator Settings				Thermostat settings provided by operator
Desired Temperature Range				Desired range of Isolette temperature
Lower Desired Temperature	Integer	[97..99]	°F	Lower value of Desired Temperature Range
	Status	•Invalid, Valid		
Upper Desired Temperature	Integer	[98..100]	°F	Upper value of Desired Temperature Range
	Status	•Invalid, Valid		
Alarm Temperature Range				Activate Alarm when outside of this range
Lower Alarm Temperature	Integer	[93..98]	°F	Lower value of Alarm Temperature Range
	Status	•Invalid, Valid		
Upper Alarm Temperature	Integer	[99..103]	°F	Upper value of Alarm Temperature Range
	Status	•Invalid, Valid		

## 2.5 Ensure Environmental Variables are Abstract

### Advice

- Variables should not have implementation details
- In the isolette example, the temperature might range from 68.0 to 105.0°F
  - Specifying storage as a “IEEE 8-Bit Floating Point” is inappropriate

## 2.6 Avoid Presentation Details

### Advice

- Avoid presentation details in operator-facing variables
  - This is part of the Human Machine Interface process
- Variable attributes (type, range, precision, etc.) are “environmental assumptions” (Section 4)

*Main point: in this stage, we are interested in the abstract input/output behavior of the system – not in the details of how values are presented to users.*

## 2.7 Define Physical Interfaces

- The system boundary (eventually) is extended into the set of physical interfaces
  - E.g., identify all discrete inputs and outputs, all messages, all fields in a message, and the protocols used to receive and send messages
  - This extension should not be done until monitored and controlled variables are identified.
  - The physical interface is at a lower level of abstraction than monitored / controlled variables, and thus is more likely to change

# Summary

To define the scope of the system (and our requirements writing effort), we specify the system boundary early in the requirements engineering process

- It is useful to state the system boundary in terms of *controlled variables* and *monitored variables*.
- Controlled variables specifies the quantities in the environment that the system affects
- Monitored variables specifies the quantities in the environment to which the system responds

# For You To Do

- Define *controlled variable*
- Define *monitored variable*
- Describe the role of controlled and monitored variables in the definition of the system boundary

# 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.