High-Level Structuring of Requirements Models

Topics for Today

- Structuring based on fundamental concepts
  - How do we structure a model?
- A little bit about what we are doing in my group
- You will learn
  - How these fundamental concepts can be used
  - How the process of developing the specification and the structure of the specification is related to these concepts

Overview

- Where to start?
- 4-variable overview
- How do I get the monitored and controlled variables?
- Specifying REQ
- Now I want to build it, how do I get the monitored/controlled variable for ____?
- Refining REQ to SOFT
- ASW Examples
Where to Start?

- We would like a well organized requirements document, that captures fundamental facts about the system and is easy to use and maintain
  - “Coffee stain” test
  - Expected/likely changes should not cause “ripple effect”
- Requirements development must be iterative
  - Things change, it is impossible to know all details up front
  - See Parnas, “Rational Design Process: How and why to fake it”
- Requirements process and requirements structuring are interrelated
  - Requirements structuring = putting first things first

Process Control Model

The Altitude Switch

- The purpose of the ASW is to turn a device of interest (DOI) on when the plane descends below a threshold altitude
- The DOI could be, for example, a ground proximity radar
- Inputs:
  - Altitude
  - Reset and Inhibit Signals
  - DOI Status
- Outputs:
  - Command to turn the DOI on
  - Failure Indication
Classes of MON and CON

- Quantities to/from the Environment
  - Altitude
  - DOI Command
  - DOI Status
- Quantities to/from the Operator
  - Inhibit
  - Reset
  - Failure
- Quality indications
  - Altitude Bad
- Quantities to/from another subsystem or abstractions
  - Any of these in the ASW?

How to Select MON and CON?

- Basic Process:
  - What do you want the system to do?
  - What controlled variables do you need to accomplish it?
  - What monitored variables are needed to determine the values of the controlled variables?
- In practice:
  - Knowledge of sensors can affect MON
  - Knowledge of actuators can affect CON
  - Delicate balance between introducing (too much) implementation, and being realistic

What to Define About Each Variable?

- Name and purpose (intent)
- Type (numeric, Boolean, enumerated)
- Properties:
  - Expected range
  - Precision
  - Physical meaning
- How each variable takes on its values (for controlled variables)
  - End result of specifying this for each Controlled variable is a specification of REQ
Sensors and Actuators

- You must be able to deal with sensor failures
  - To deal with sensor failures at the REQ level, you introduce quality indications like Altitude_Bad
- Sensors and actuators are noisy
  - Noise in sensors must allow you to construct an estimate of the monitored quantity within the designated precision
- You often must fuse the information from several sensors to get the information you need
  - Example: 3 Altimeters of the ASW
  - Example: Nuclear Power
- This seems complicated ?!
  - I thought REQ was the important/hard part?
Process Control Model

Possible Changes to the System

- I’ve got a better sensor to measure altitude and I’d like to use it in the system because it will mean I will get into the failure modes of REQ less often
- We need the system to perform a new task with the existing sensors and actuators
- There is a new lifting machine on the assembly line that has a different control language than the old one, but it does the same task
- We have a new sensor that will allow us to get additional information about the environment, and we want to use that information when it’s available

Structuring SOFT
Modeling Process

Figure conceived by Dr. Steve Miller

Input: MON → CON
Output: MON' → CON'

Specification Based Prototyping

- **Goal**: Combine the advantages of formal specifications with the advantages of prototyping while eliminating some of the drawbacks of both approaches.
  - Advantages of Formal Specifications
    - Clear, well understood semantics
    - Readable
    - Analyzable
  - Advantages of Prototyping
    - Risk management
    - Early customer involvement

ASW Details

Altitude, Inhibit, Reset, DOI Status

DOI Command, Fault Indicator

Altitude

Altimeter
Nimbus Goals

- Support formal definition of *inter-component communication*
- Support execution of the formal specification while interacting with *accurate* models of the controller’s environment
- As the specification is refined during the requirements process, there should not be any large conceptual leaps in the way the model communicates with the environment

Nimbus Environment

ASW High-Level Prototyping

Altitude
- Reset
- Inhibit
- DOI Status
- Watchdog
- DOI Command
Refining the ASW

- The ASW actually has **three** altimeters. Thus, the REQ relation altitude should be split into three separate inputs in the IN relation:

```
MON  Attitude   DigitalAlt_1  DigitalAlt_2  AnalogAlt  INPUT
```

Furthermore, the digital altimeters give the altitude as a fraction of 8,192 whereas the analog altimeters give only above or below (threshold altitude is hard-wired).

```
DigitalAlt = \frac{\text{Altitude}}{8,192}
AnalogAlt = \begin{cases} 
\text{Above} & \text{if Altitude} > \text{Threshold} \\
\text{Below} & \text{if Altitude} \leq \text{Threshold}
\end{cases}
```

Refine to Include 3 Altimeters

Model Extension
Refined Model of the ASW

Macro from the Refined ASW

Specification-Based Prototyping Specification Refinement
Conclusion

- We have defined how to extend system requirements to software requirements
- **Specification-based prototyping** is a powerful combination of formal modeling with a prototyping methodology
- **Nimbus** allows for flexible execution and easy integration in system simulations so that the right tool can be used to accomplish the evaluation and modeling task

What Have we Learned?

- What effects your choices of monitored and controlled variables
  - What quantities are candidates
  - What process you use in selecting them
  - How sensors and actuators can affect those decisions
- How you can structure the requirements document, and the requirements effort, to isolate common changes
  - What changes are likely; which relations will be affected
  - How to refine REQ to SOFT
  - How to structure SOFT to isolate anticipated changes
- Next time
  - RSML* and Nimbus

http://www.umsec.umn.edu