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 to 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

http://www.umsec.umn.edu
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

MON CON

IN OUT

NAT

REQ

Process

Sensors

Actuators

Controller

INPUT SOFT

OUTPUT

http://www.umsec.umn.edu

http://www.umsec.umn.edu
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

MON \rightarrow \text{INPUT} \rightarrow \text{SOFT} \rightarrow \text{OUTPUT} \rightarrow \text{CON}

\text{IN} \rightarrow \text{MON'} \rightarrow \text{CON'} \rightarrow \text{OUT'}

Modeling Process

MON \rightarrow \text{REQ} \rightarrow \text{CON}

\text{IN} \rightarrow \text{MON'} \rightarrow \text{REQ'} \rightarrow \text{CON'} \rightarrow \text{OUT'}

INPUT \rightarrow \text{SOFT} \rightarrow \text{OUTPUT}

Figure conceived by Dr. Steve Miller
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

ASW

DOI Command, Fault Indicator

ASW

Altitude

Inhibit

Reset

Fault Indicator

Altimeter

DOI Command

DOI Status
High-Level RSML-e Model of the ASW

A Transition from an RSML-e Model

**Transition**: Above ➔ Below

**Location**: AltitudeStatus

**Condition**: 
- BelowThreshold() ➔ T
- AltitudeQualityOK() ➔ T

**Macro**: BelowThreshold()

Altitude < AltitudeThreshold ➔ T
Visualization & Execution

Formal Model

Model of Environment

Input

Output

Statecharts
SCR*
Stateflow
Etc.

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

- **Sensors**
  - Text Files
  - RSML-e Models
  - Software Simulations
  - Hardware

- **Process Model**

- **Actuators**
  - Text Files
  - RSML-e Models
  - Software Simulations
  - Hardware

- **Control (RSML-e Model)**

---

### ASW High-Level Prototyping

- **RSML-e Simulator**
  - Altitude
  - Reset
  - Inhibit
  - DOI Status
  - Watchdog
  - DOI Command

- **Microsoft Excel**
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:

\[
\begin{align*}
\text{MON} & \quad \text{Altitude} \\
\text{DigitalAlt}_1 & \quad \text{DigitalAlt}_2 \\
\text{AnalogAlt} & \quad \text{INPUT}
\end{align*}
\]

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).

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

Refine to Include 3 Altimeters

- Microsoft Excel
- RSML-e Simulator
- DOI Status
- Inhibit
- Reset
- Digital Alt. 1
- Digital Alt. 2
- Analog Alt
Model Extension

Altitude
+ more

REQ

CON

+more

BelowThreshold()

IN

IN'

DigitalAlt_1
DigitalAlt_2
AnalogAlt

Refined Model of the ASW

FullyOperational

AltitudeStatus

Above

AltBad

Below

ASWStatus

Ok

Failure

DigitalAltimeter-1

OK

Failed

DigitalAltimeter-2

OK

Failed

AnalogAltimeter

PowerOn

Attempting

PowerOff

DOI

Failed

Failed

Failed
Macro from the Refined ASW

Macro: BelowThreshold()

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>MeasuredDigitalAlt(DigitalAlt_1) &lt; AltitudeThreshold</td>
<td>T</td>
<td>.</td>
</tr>
<tr>
<td>DigitalAltimeter_1_OK()</td>
<td>T</td>
<td>.</td>
</tr>
<tr>
<td>MeasuredDigitalAlt(DigitalAlt_2) &lt; AltitudeThreshold</td>
<td>.</td>
<td>T</td>
</tr>
<tr>
<td>DigitalAltimeter_2_OK()</td>
<td>.</td>
<td>T</td>
</tr>
<tr>
<td>AnalogAltitudeMeasure() = Below</td>
<td>.</td>
<td>T</td>
</tr>
<tr>
<td>AnalogAltimeter_OK()</td>
<td>.</td>
<td>T</td>
</tr>
</tbody>
</table>

Specification-Based Prototyping
Specification Refinement

Execute & Evaluate
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-e and Nimbus