AADL Requirements Annex Review

Dominique Blouin
Lab-STICC
Université de Bretagne-Occidentale
Université de Bretagne-Sud
Bretagne, France
Agenda

- Motivations for a Requirements Annex
- Language Overview
- Simple Pacemaker Example
- REMH Isolette Example
- Conclusion and Perspectives
Motivations for a Requirements Annex

Benefits of early fault discovering (Feiler 2009)

AADL Requirements Annex Overview, September 25th, 2013
Why a new RE Language?

Existing languages:

URN (ITU Z.151)

GRL

URM (ITU Z.151)

UCM
Problems with Existing RE Languages

- Include constructs for both the **problem** (requirements) and **solution** (design) domains.

- Implies duplication of information.
  - Model transformation used to derive an initial AADL design.
  - Need for model synchronization.

- Difficult to trace requirements to AADL design elements.

- Not always adapted to support RE bests practices of embedded systems development.

  *Solution: create a new RE language that is meant to be composed with other existing languages.*
Objectives for Requirements Annex

- **Separate the concerns:**
  - Problem (requirements) / solution (design)
  - Easy to use with AADL (and also other ADLs).

- **Emphasis on semantics:**
  - Analysis
    - Partially automated **validation** of requirements specifications.
    - Automated **verification** of formally expressed requirements.
  - Documentation generation.
    - Requirements spec. generated from the model (not the reverse).

- **Fitted for the embedded systems domain.**
Objectives for Requirements Annex

- Extensible with respect to:
  - Expression languages (natural, OCL, Lute, PSL, BLESS, etc.).
  - Traceability (traceability links to models from other languages for different concerns can be defined).

- Incorporate good ideas from other languages and methods:
  - SysML
  - KAOS
  - URN
  - FAA Requirements Engineering Management Handbook

- Do not restrict the use of the requirements annex to a specific process.
Agenda

- Motivations for a Requirements Annex
- Language Overview
- Simple Pacemaker Example
- REMH Isolette Example
- Conclusion and Perspectives
Class Diagram Conventions

Language Color Legend
- RDAL (Requirements Definition and Analysis Language)
- CLML (Constraints Language Modeling Language)
- Settings
- Ecore

Reference (blue)
Inheritance (black)

What
When
Core RDAL Language
Contractual Element Realizations
Motivations for a Requirements Annex

Language Overview

Simple Pacemaker Example

REMH Isolette Example

Conclusion and Perspectives
Pulse Generator Requirements

Performance

PERF-REQ-1: Power Consumption less Max Allowed

PERF-REQ-2: Availability greater than Min Allowed

PERF-REQ-3: Latency less than Max Allowed

VER-PERF-REQ-1

VER-PERF-REQ-2

VER-PERF-REQ3

AADL Requirements Annex Overview, September 25th, 2013
Hard / Soft Bridges to Design Elements

- A soft bridge provides a **view** of the design for assigning requirements and goals to design elements.

- The bridged elements must be part of the scope of the specification.
Requirements Expressions

The pulse generator power consumption shall be less than or equal to the maximum allowed power consumption.

AADL Requirements Annex Overview, September 25th, 2013
Requirements Expressions

Choice of constraint languages

Expression

Extensible Choice of Constraint Languages

Libraries
Automated *verification* of formally expressed requirements.
Goals Modeling

- Requirements are *verifiable*:
  - Boolean world: *power consumption less than value*.

- Goals are *satisfiable*:
  - Quantitative world: *power consumption should be minimal*.

- RDAL specifications incorporate both to ensure requirements are always met as the design is optimized.
Agenda

- Motivations for a Requirements Annex
- Language Overview
- Simple Pacemaker Example
- REMH Isolette Example
- Conclusion and Perspectives
Set of 11 best practices to enable successful management of requirements.
Based on a study of the literature and industry practices.

Methods and tools from research not adopted because of the difficulty to change existing practices and tools.
REMH’s practices that can be adopted incrementally.
Isolette Thermostat Example

- Two functions introduced for safety reasons:
  - Maintain constant current temperature.
  - Monitor current temperature.

A.1.2 SYSTEM GOALS.
The high-level goals (G) of the system are:

- G1—The Infant should be kept at a safe and comfortable temperature.
- G2—The Nurse should be warned if the Infant becomes too hot or too cold.
- G3—The cost of manufacturing the Thermostat should be as low as possible.
Modeling of the Example

APPENDIX A—ISOLETTE THERMOSTAT EXAMPLE

This appendix contains an example requirements specification for the Isolette Thermostat discussed in section 3. The presented format is one example of how the best practices of section 2 could be realized. There are many other formats that would be equally effective.

A.1. SYSTEM OVERVIEW

The system being specified is the Thermostat of an Isolette. An Isolette is an incubator for an Infant that provides controlled temperature, humidity, and oxygen (if necessary). Isolettes are used extensively in Neonatal Intensive Care Units for the care of premature infants.

The purpose of the Isolette Thermostat is to maintain the air temperature of an Isolette within a desired range. It senses the Current Temperature of the Isolette and turns the Heat Source on and off to warm the air as needed. If the temperature falls too far below or rises too far above the Desired Temperature Range, it activates an alarm to alert the Nurse. The system allows the Nurse to set the Desired Temperature Range and to set the Alarm Temperature Range outside the Desired Temperature Range of which the alarm should be activated.

A.1.1. SYSTEM CONTEXT

- Nurse
  - Operator Interface
    - Operator Settings
      - Operator Feedback
    - Current Temperature
      - Temperature Sensor
        - Thermostat
          - Heat Control
            - Heat Source

## Formalizing REMH Best Practices

<table>
<thead>
<tr>
<th>Practice #</th>
<th>Description</th>
<th>Modeling Language(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Develop the System Overview</td>
<td>RDAL, AADL</td>
</tr>
<tr>
<td>2</td>
<td>Identify the System Boundary</td>
<td>RDAL, AADL</td>
</tr>
<tr>
<td>3</td>
<td>Develop the Operational Concepts</td>
<td>RDAL, URN</td>
</tr>
<tr>
<td>4</td>
<td>Identify the Environmental Assumptions</td>
<td>RDAL, AADL</td>
</tr>
<tr>
<td>5</td>
<td>Develop the Functional Architecture</td>
<td>RDAL, AADL</td>
</tr>
<tr>
<td>6</td>
<td>Revise the Architecture to Meet Implementation Constraints</td>
<td>RDAL</td>
</tr>
<tr>
<td>7</td>
<td>Identify System Modes</td>
<td>AADL</td>
</tr>
<tr>
<td>8</td>
<td>Develop the Detailed Behavior and Performance Requirements</td>
<td>RDAL, AADL</td>
</tr>
<tr>
<td>9</td>
<td>Define the Software Requirements</td>
<td>RDAL, AADL</td>
</tr>
<tr>
<td>10</td>
<td>Allocate System Requirements to Subsystems</td>
<td>RDAL, AADL</td>
</tr>
<tr>
<td>11</td>
<td>Provide Rationale</td>
<td>RDAL</td>
</tr>
</tbody>
</table>
The system interacts with its environment through variables (monitored / controlled).

System requirements define a precise relationship between monitored and controlled variables for all contexts of operation of the system.

System Boundary == Environment Variables.

“Getting the system boundary correct is 90% of the problem!”

- Stuart Faulk to Steven Miller... 😊
Define the System Overview
System Overview Diagram

- Combined RDAL and AADL to formalize the system overview.
  - Profiled AADL for defining the system overview.
  - RDAL to represent system overview concepts and trace AADL elements.

![Diagram of system overview with labels and connections between components like Thermostat, Temperature Sensor, Heat Source, and Operator Interface. The diagram shows system boundary markers and flow arrows between elements.](image-url)
System Context Definition

- Normal context of operation.
package isclette

public

with Isclette_Properties, Data_Model;

--@description An Isolette is an incubator for an Infant that provides controlled temperature,
-- humidity, and oxygen (if necessary). Isolettes are used extensively in Neonatal Intensive Care
-- Units for the care of premature infants.

system Isolette

properties

   Isolette_Properties::Temperature_Increase_Rate => 0.5 Fdeg_Minute;
   Isolette_Properties::Temperature_Decrease_Rate => 0.5 Fdeg_Minute;

end Isolette;

system Thermostat

features

   current_temperature : in feature group Current_Temperature;
   operator_settings : in feature group Operator_Settings;
   heat_control : out data port Heat_Control;
   operator_feedback : out feature group Operator_Feedback;

flows

   current_temperature_heat_control : flow path current_temperature -> heat_control;
   desired_temp_regulator_status : flow path operator_settings.desired_temperature_range ->
                                  operator_feedback.regulator_status;
   alarm_temp_monitor_status : flow path operator_settings.alarm_temperature_range ->
                               operator_feedback.monitor_status;

end Thermostat;

--@description The Temperature Sensor provides the Current Temperature of the Air in the Isolette to
the Thermostat.

system Temperature_Sensor

features

   heat : in data port Heat;
   current_temperature : feature group inverse of Current_Temperature;

flows

   temp_sensor_source : flow source current_temperature;

properties

   Isclette_Properties::Failure_Probability => 0.000000005 abs;

end Temperature_Sensor;

system Operator_Interface

features

   operator_interaction : in out data port Interface_interaction;
   operator_feedback : feature group inverse of Operator_Feedback;
   operator_settings : feature group inverse of Operator_Settings;

flows

   desired_temp_range_source : flow source operator_settings.desired_temperature_range;

end Operator_Interface;
Develop the Operational Concepts

Simulating Use Case Maps

- Revealed several inconsistencies and incompletenesses...

- 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)
Develop the Operational Concepts

A.2.2 USE CASE: CONFIGURE THE ISOLETTE.

This use case describes how the Nurse configures the Isolette and Thermostat for the Infant.

- Related System Goals: G1 and G2
- Primary Actor: Nurse
- Precondition: The Isolette and Thermostat are turned on
- Postcondition:
  - The Desired Temperature Range is set for the needs of the Infant
  - The Alarm Temperature Range is set for the needs of the Infant
  - The Current Temperature in the Isolette is in the Desired Temperature Range

- Main Success Scenario:
  1. Nurse sets the Alarm Temperature Range for the Infant (A.5.2.1)
  2. Nurse sets the Desired Temperature Range for the Infant (A.5.1.1)
  3. Thermostat maintains Desired Temperature Range (A.2.3)
# Issues in Use Cases

<table>
<thead>
<tr>
<th>#</th>
<th>Use Case</th>
<th>Step</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>UC A.2.1 : Normal Operation of Isolette</strong></td>
<td>4 ➔ 5. Nurse waits until the Current Temperature is within the Desired Temperature Range (A.2.6 and A.5.1.1)</td>
<td>The post condition of A.2.2 is that the Current Temperature is within the Desired Temperature Range. Why the wait?</td>
</tr>
<tr>
<td>2</td>
<td><strong>UC A.2.2: Configure the Isolette</strong></td>
<td>1. Nurse sets the Alarm Temperature Range for the Infant (A.5.2.1)</td>
<td>Activate alarm step missing after setting the alarm temperature.</td>
</tr>
<tr>
<td>3</td>
<td><strong>UC A.2.1 : Normal Operation of Isolette</strong></td>
<td>4 ➔ 5. Nurse waits until the Current Temperature is within the Desired Temperature Range (A.2.6 and A.5.1.1)</td>
<td>Pre-condition of exception case A.2.6 too restrictive. The Isolette may never reach the alarm temperature range after configuration.</td>
</tr>
<tr>
<td>4</td>
<td><strong>UC A.2.1 : Normal Operation of Isolette</strong></td>
<td>4 ➔ 5. Nurse waits until the Current Temperature is within the Desired Temperature Range (A.2.6 and A.5.1.1)</td>
<td>There is a step to remove the infant in an alternate course of exception case A.2.6. However, when called from here, the infant is not in the Isolette yet.</td>
</tr>
<tr>
<td>5</td>
<td><strong>UC A.2.1 : Normal Operation of Isolette</strong></td>
<td>EC A.2.4, step 3</td>
<td>Difference in terminology between UC A.2.1 (normal operation of isolette) and EC A.2.4 (failure to maintain safe temperature): Current Temperature used instead of Display Temperature.</td>
</tr>
<tr>
<td>#</td>
<td>Use Case</td>
<td>Step</td>
<td>Issue</td>
</tr>
<tr>
<td>---</td>
<td>----------------------------------------------</td>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>6</td>
<td>UC A.2.1: Normal Operation of Isolette</td>
<td>Step 1 on sub exception case 1</td>
<td>No step to turn the alarm off in case it is turned on in step 1 on sub exception case 1.</td>
</tr>
<tr>
<td>7</td>
<td>EC A.2.4: Failure to Maintain Safe Temperature</td>
<td>2</td>
<td>No step to turn the alarm off while alarm is turned on and post condition is that alarm is off.</td>
</tr>
<tr>
<td>8</td>
<td>EC A.2.4: Failure to Maintain Safe Temperature</td>
<td>3</td>
<td>Error in the label: Nurse sees that the Display Temperature is in <strong>not</strong> the Alarm Temperature Range.</td>
</tr>
<tr>
<td>9</td>
<td>EC A.2.4: Failure to Maintain Safe Temperature</td>
<td>-</td>
<td>No indication where the exception case is called.</td>
</tr>
<tr>
<td>10</td>
<td>UC A.2.3: Maintain Desired Temperature</td>
<td></td>
<td>When should it really start? Since the thermostat can turn the alarm on before the Isolette is configured, should it also try to maintain the desired temperature then too? This raises the question of the status variable for the operator settings not being set to <em>unspecified</em> at startup.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td>Need a use case for monitoring the temperature like there is one for maintaining the current temperature within the desired temperature.</td>
</tr>
</tbody>
</table>
## Issues in Use Cases (cont’d)

<table>
<thead>
<tr>
<th>#</th>
<th>Use Case</th>
<th>Step</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>UC A.2.1: Normal Operation of Isolette</td>
<td>Step 1 on sub exception case 1</td>
<td>No step to turn the alarm off in case it is turned on in step 1 on sub exception case 1.</td>
</tr>
<tr>
<td>7</td>
<td>EC A.2.4: Failure to Maintain Safe Temperature</td>
<td>2</td>
<td>No step to turn the alarm off while alarm is turned on and post condition is that alarm is off.</td>
</tr>
<tr>
<td>8</td>
<td>EC A.2.4: Failure to Maintain Safe Temperature</td>
<td>3</td>
<td>Error in the label: Nurse sees that the Display Temperature is in <strong>not</strong> the Alarm Temperature Range.</td>
</tr>
<tr>
<td>9</td>
<td>EC A.2.4: Failure to Maintain Safe Temperature</td>
<td>-</td>
<td>No indication where the exception case is called.</td>
</tr>
<tr>
<td>10</td>
<td>UC A.2.3: Maintain Desired Temperature</td>
<td></td>
<td>When should it really start? Since the thermostat can turn the alarm on before the Isolette is configured, should it also try to maintain the desired temperature then too? This raises the question of the status variable for the operator settings not being set to <em>unspecified</em> at startup.</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td><strong>Need a use case for monitoring the temperature like there is one for maintaining the current temperature within the desired temperature.</strong></td>
</tr>
</tbody>
</table>
Identify the Environmental Assumptions

- Environmental assumptions constrain the entities of the environment (not the system we are building).

- Having a formalized system boundary allows to automatically *distinguish* between assumptions and requirements.

- Allow checking that an assumption on an entity has a corresponding requirement in the requirements specification of the entity.
Identify the Environmental Assumptions

- EA-IS-1: Current Temp Increase Rate
- EA-IS-2: Current Temp Decrease Rate

Description:
When the Heat Source is turned on and the Isolette is properly shut, the Current Temperature will increase at a rate of no more than 1°F per minute.
Develop the Functional Architecture

- Use AADL for data flow...

- Mapping:
  - Functions ➞ Subprograms
  - Variables ➞ Data Ports /
  - Feature Groups
  - Flows ➞ Port Connections
  - Independent Functions ➞ Separate Processes.
Adele Diagram
Develop the Detailed Behavior and Performance Requirements

- Behavior Requirements:
  - Specify the *relationships* between controlled and monitored variables to be imposed by the *system*.

The requirements for the Heat Control controlled variable are as follows:

- **REQ-MHS-1**: If the Regulator Mode is INIT, the Heat Control shall be set to Off.
  
  **Rationale**: A regulator that is initializing cannot regulate the Current Temperature of the Isolette and the Heat Control should be turned off.

- **REQ-MHS-2**: If the Regulator Mode is NORMAL and the Current Temperature is less than the Lower Desired Temperature, the Heat Control shall be set to On.

- **REQ-MHS-3**: If the Regulator Mode is NORMAL and the Current Temperature is greater than the Upper Desired Temperature, the Heat Control shall be set to Off.

...
Detailed Behavior Requirement

NuSMV CTL Formal Requirement (not yet implemented)

AG(REG_NORMAL_MODE & Current_Temp > Upper_Desired_Temp -> Heat_Control = Off)
Allocate System Requirements to Subsystems

Design

- isolette.aadl
- heat_source.aadl
- thermostat.aadl
- temperature_sensor.aadl
- operator_interface.aadl
- display.aadl
- keypad.aadl

Integration

- isolette_integration1.aadl
- heat_source.aadl
- thermostat.aadl
- temperature_sensor.aadl
- operator_interface_integration.aadl
- display.aadl
- keypad.aadl
Allocate System Requirements to Subsystems

Requirements
- isolette.rdal
- heat_source.rdal
- thermostat.rdal
- temperature_sensor.rdal
- operator_interface.rdal
- display.rdal
- keypad.rdal

Design
- isolette.aadl
- heat_source.aadl
- thermostat.aadl
- temperature_sensor.aadl
- operator_interface.aadl
- display.aadl
- keypad.aadl
Provide Rationale

- Providing rationale is the single most effective way of reducing the cost and improving the quality of requirements*.
  - Thinking of the rationale at elicitation provides better requirements.
  - Reduce the amount of time to understand requirements.
  - Providing insight into the impact of changing the requirements.

Results of Analyzing the Isolette Thermostat Example Models

- 11 inconsistencies and shortcomings in natural language use cases.
- 2 missing environmental assumptions.
- One vacuous requirement.
- One *irrational* rationale.
- Support for automated verification of requirements by design.
Agenda

- Motivations for a Requirements Annex
- Language Overview
- Simple Pacemaker Example
- REMH Isolette Example
- Conclusion and Perspectives
Conclusion

- The requirements annex is an important complement of the AADL providing model based RE.
  - Provides support for quality assurance of the requirements specification.
    - Consistency
    - Completeness
    - Unambiguity
    - Etc...
  - Support for automated requirements verification.
  - Support for traceability to other languages (AADL, UCM).
  - Formal identification of system overview and environmental assumptions.
  - Rationale capture and analysis with link to stakeholders.
Perspectives

- First prototype tool integrated with OSATE 2.
  - RDALTE
  - GMF based diagram editor.
  - \textit{Does not yet cover all features of the language...}
  - \url{https://wiki.sei.cmu.edu/aadl/index.php/RDALTE}
  - Includes a complete model of the Isolette Thermostat example.

- A version 2 of the tool is under development.

- Users required to validate the language.