UML Extensions: the SysML profile

Excerpts from Bran Selic (IBM) tutorial on UML & SysML (2007)
Rationale

Systems engineering typically involves complex combinations of diverse disciplines and technologies

- Difficult to understand
- Many integration problems

Modeling can alleviate many of these problems

- Raising the level of abstraction hides technological detail that can be confusing

Why a UML profile?

- Reuse of widely-available UML expertise
- Reuse of UML tooling
UML 2 and SysML

Uses a subset of UML concepts

- Simplified language
- Provides SE-specific customization of certain UML concepts
- However, it is possible to combine the excluded concepts if desired

Charles ANDRE - UNSA
Some UML diagrams were modified, others omitted, and new SysML-specific diagrams added.
SysML Diagram format

Simpler and more systematic approach than UML
– All diagrams have a common format
Defining & Specifying Physical Quantities

- Using *value types*
  - e.g. a delay expressed in seconds: `timeDelay : s`
- *ValueType* is a specialization of the UML *DataType* concept and has
  - a *dimension* and
  - a *unit*:

```
< dimension >
  Time

< unit >
  Second
  dimension = Time
  unit = Second

< valueType >
  s
```

Pre-defined units
- Time
- Length
- Mass
- Power
- ...
**SysML Blocks**

- **Block** = a unifying SysML concept that unifies the UML Class and Collaboration concepts into a concept more familiar to systems engineers

- **Models:**
  - Hardware
  - Software
  - Data
  - Facilities
  - Physical entities
  - etc.

A non-encapsulated block is logically equivalent to a collaboration.

Parts are elements of the internal structure of a block.

Parts that are not owned by the block.

Defines useful values related to the block.

Charles ANDRE - UNSA
Block Definition Diagrams (bdd)

- Play the same role as UML Class diagrams
Internal Block diagram (ibd)

Captures the internal structure of a block
Nested Connectors

- Connectors that reach inside a non-encapsulated block instance

![Diagram of Wheel Assembly with nested connectors]

Charles ANDRE - UNSA
SysML Ports and Flows

Two kinds:
- Standard ports = UML ports
- Flow ports = support the transfer of flows

A flow models a *streaming* phenomena (energy, liquids, electrical currents, data streams, etc.)
- Flows have a *direction* relative to a block
**SysML Parametrics**

Specify relationships (equations) between value properties
- Used for engineering analysis
- Have a block-like syntax

*Constraint blocks* define a constraint and identify its parameters

```
« constraint »
NewtonLaw

constraints
{ f = m*a }

parameters
m: Mass
a: Acceleration
f: Force
```

An occurrence of the constraint
Parametrics Diagram

Used for engineering analysis

\[ d(n+1) = d(n) + v^* dt \]

\[ v(n+1) = v(n) + a^* dt \]

\[ f = m^* a \]
**SysML Allocation**

- Mapping of a set of (client) elements in a model to another (target) element

- An abstract concept with many potential interpretations
  - The target element is an implementation of the client elements
  - The client element is an abstract representation of the target
  - The target is the hardware on which the client software is deployed
  - The target is responsible for the behavior represented by the client
  - etc.
SysML Requirements Modeling

• Requirements represent an important and dynamic element of system engineering
  – SysML provides a set of modeling concepts and relationships for capturing requirements and their relationships to other system engineering artifacts
  – Complement to use case modeling

• Basic concepts:

```
<table>
<thead>
<tr>
<th>« requirement »</th>
<th>« testCase »</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemRecovery</td>
<td>TestRecovery</td>
</tr>
<tr>
<td>id = &quot;SR100/07&quot;</td>
<td>result : VerdictKind</td>
</tr>
<tr>
<td>text = &quot;The system shall ...&quot;</td>
<td>parameter [0..*]</td>
</tr>
</tbody>
</table>
```

```
« enumeration » VerdictKind
pass
fail
inconclusive
error
```

Requirement       TestCase
Hierarchical Requirements

For decomposing complex requirements into sub-requirements
References

OMG SysML website

http://www.omgsysml.org

OMG Systems Modeling Language, v1.0

SysML tutorial
Sanford Friedenthal, Alan Moore, Rick Steiner
OMG Systems Modeling Language Tutorial
Incose 2007, San Diego, June 2007
Distiller Sample Problem
Distiller Problem Statement

- The following problem was posed to the SysML team in Dec ’05 by D. Oliver:
  - Describe a system for purifying dirty water.
    - Heat dirty water and condense steam are performed by a Counter Flow Heat Exchanger
    - Boil dirty water is performed by a Boiler
    - Drain residue is performed by a Drain
    - The water has properties: vol = 1 liter, density 1 gm/cm3, temp 20 deg C, specific heat 1cal/gm deg C, heat of vaporization 540 cal/gm.

- A crude behavior diagram is shown.
Distiller Types

Batch Distiller

Continuous Distiller

Note: Not all aspects of the distiller are modeled in the example
Distiller Problem – Process Used

- Organize the model, identify libraries needed
- List requirements and assumptions
- Model behavior
  - In similar form to problem statement
  - Elaborate as necessary
- Model structure
  - Capture implied inputs and outputs
    - segregate I/O from behavioral flows
  - Allocate behavior onto structure, flow onto I/O
- Capture and evaluate parametric constraints
  - Heat balance equation
- Modify design as required to meet constraints
- Model the user interaction
- Modify design to reflect user interaction
Distiller Problem – Package Diagram: Model Structure and Libraries
Distiller Example
Requirements Diagram

req [package] DistillerRequirements

Source

Id = S0.0
text = Describe a system for purifying dirty water.
- Heat dirty water and condense steam are performed by a Counter Flow Heat Exchanger
- Boil dirty water is performed by a Boiler. Drain residue is performed by a Drain.
The water has properties: vol = 1 liter, density 1 gm/cm3, temp 20 deg C, specific heat 1cal/gm deg C, heat of vaporization 540 cal/gm.

«requirement»
Id = S1.0
text = The system shall purify dirty water.

«requirement» PurifyWater
Id = S1.0
text = The system shall purify dirty water.

«rationale»
The requirement for a boiling function and a boiler implies that the water must be purified by distillation

«requirement»
Id = S2.0
text = Heat dirty water and condense steam are performed by a Counter Flow Heat Exchanger

«requirement» HeatExchanger
Id = S2.0
text = Heat dirty water and condense steam are performed by a Counter Flow Heat Exchanger

«requirement»
Id = S3.0
text = Boil dirty water is performed by a Boiler.

«requirement» Boiler
Id = S3.0
text = Boil dirty water is performed by a Boiler.

«requirement»
Id = S4.0
text = Drain residue is performed by a Drain.

«requirement» Drain
Id = S4.0
text = Drain residue is performed by a Drain.

«requirement»
Id = S5.0
text = water has properties: density 1 gm/cm3, temp 20 deg C, specific heat 1cal/gm deg C, heat of vaporization 540 cal/gm.

«requirement» WaterProperties
Id = S5.0
text = water has properties: density 1 gm/cm3, temp 20 deg C, specific heat 1cal/gm deg C, heat of vaporization 540 cal/gm.

«requirement»
Id = S5.1
text = water has an initial temp 20 deg C

«requirement» WaterInitialTemp
Id = S5.1
text = water has an initial temp 20 deg C

«deriveReqt»

«requirement»
Id = S0.0
text = Describe a system for purifying dirty water.
- Heat dirty water and condense steam are performed by a Counter Flow Heat Exchanger
- Boil dirty water is performed by a Boiler. Drain residue is performed by a Drain.
The water has properties: vol = 1 liter, density 1 gm/cm3, temp 20 deg C, specific heat 1cal/gm deg C, heat of vaporization 540 cal/gm.

«requirement» OriginalStatement
Id = S0.0
text = Describe a system for purifying dirty water.
- Heat dirty water and condense steam are performed by a Counter Flow Heat Exchanger
- Boil dirty water is performed by a Boiler. Drain residue is performed by a Drain.
The water has properties: vol = 1 liter, density 1 gm/cm3, temp 20 deg C, specific heat 1cal/gm deg C, heat of vaporization 540 cal/gm.

«requirement»
Id = S1.0
text = The system shall purify dirty water.

«requirement» PurifyWater
Id = S1.0
text = The system shall purify dirty water.

«rationale»
The requirement for a boiling function and a boiler implies that the water must be purified by distillation

«requirement»
Id = S2.0
text = Heat dirty water and condense steam are performed by a Counter Flow Heat Exchanger

«requirement» HeatExchanger
Id = S2.0
text = Heat dirty water and condense steam are performed by a Counter Flow Heat Exchanger

«requirement»
Id = S3.0
text = Boil dirty water is performed by a Boiler.

«requirement» Boiler
Id = S3.0
text = Boil dirty water is performed by a Boiler.

«requirement»
Id = S4.0
text = Drain residue is performed by a Drain.

«requirement» Drain
Id = S4.0
text = Drain residue is performed by a Drain.

«requirement»
Id = S5.0
text = water has properties: density 1 gm/cm3, temp 20 deg C, specific heat 1cal/gm deg C, heat of vaporization 540 cal/gm.

«requirement» WaterProperties
Id = S5.0
text = water has properties: density 1 gm/cm3, temp 20 deg C, specific heat 1cal/gm deg C, heat of vaporization 540 cal/gm.

«requirement»
Id = S5.1
text = water has an initial temp 20 deg C

«requirement» WaterInitialTemp
Id = S5.1
text = water has an initial temp 20 deg C

«requirement»
Id = S0.0
text = Describe a system for purifying dirty water.
- Heat dirty water and condense steam are performed by a Counter Flow Heat Exchanger
- Boil dirty water is performed by a Boiler. Drain residue is performed by a Drain.
The water has properties: vol = 1 liter, density 1 gm/cm3, temp 20 deg C, specific heat 1cal/gm deg C, heat of vaporization 540 cal/gm.

«requirement» OriginalStatement
Id = S0.0
text = Describe a system for purifying dirty water.
- Heat dirty water and condense steam are performed by a Counter Flow Heat Exchanger
- Boil dirty water is performed by a Boiler. Drain residue is performed by a Drain.
The water has properties: vol = 1 liter, density 1 gm/cm3, temp 20 deg C, specific heat 1cal/gm deg C, heat of vaporization 540 cal/gm.
Distiller Example: Requirements Tables

**Table [requirement] OriginalStatement [Decomposition of OriginalStatement]**

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0.0</td>
<td>OriginalStatement</td>
<td>Describe a system for purifying dirty water. ...</td>
</tr>
<tr>
<td>S1.0</td>
<td>PurifyWater</td>
<td>The system shall purify dirty water.</td>
</tr>
<tr>
<td>S2.0</td>
<td>HeatExchanger</td>
<td>Heat dirty water and condense steam are performed by a ...</td>
</tr>
<tr>
<td>S3.0</td>
<td>Boiler</td>
<td>Boil dirty water is performed by a Boiler.</td>
</tr>
<tr>
<td>S4.0</td>
<td>Drain</td>
<td>Drain residue is performed by a Drain.</td>
</tr>
<tr>
<td>S5.0</td>
<td>WaterProperties</td>
<td>water has properties: density 1 gm/cm3, temp 20 deg C, ...</td>
</tr>
<tr>
<td>S5.1</td>
<td>WaterInitialTemp</td>
<td>water has an initial temp 20 deg C</td>
</tr>
</tbody>
</table>

**Table [requirement] PurifyWater [Requirements Tree]**

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>relation</th>
<th>id</th>
<th>name</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1.0</td>
<td>PurifyWater</td>
<td>deriveReqt</td>
<td>D1.0</td>
<td>DistillWater</td>
<td>The requirement for a boiling function and a boiler implies that the water must be purified by distillation</td>
</tr>
</tbody>
</table>
Distiller Example – Activity Diagram: Initial Diagram for DistillWater

- This activity diagram applies the SysML EFFBD profile, and formalizes the diagram in the problem statement.
Distiller Example – Activity Diagram: Control-Driven: Serial Behavior

Batch Distiller

Continuous Distiller Here
Distiller Example – Block Definition

Diagram: DistillerBehavior

Activities (Functions)
- DistillWater
- HeatWater
- BoilWater
- CondenseSteam
- DrainResidue

Need to consider phases of H₂O

Control (not shown on BDD)

Things that flow (ObjectNodes)

7/26/2007

Distiller Example – State Machine Diagram: States of H2O

States:
- Gas: [water_temp==100 & latent_heat_of_vaporization_added]
- Liquid: [water_temp==100 & latent_heat_of_vaporization_removed]
- Solid: [water_temp==0 & latent_heat_of_liquification_added]
- Solid: [water_temp==0 & latent_heat_of_liquification_removed]

Transitions:
- From Gas to Liquid: [water_temp==100 & latent_heat_of_vaporization_removed]
- From Liquid to Solid: [water_temp==0 & latent_heat_of_liquification_added]
- From Solid to Gas: [water_temp==0 & latent_heat_of_liquification_removed]
Distiller Example – Activity Diagram: I/O Driven: Continuous Parallel Behavior
Distiller Example – Activity Diagram:
No Control Flow, ActionPin Notation,
Simultaneous Behavior
Distiller Example – Activity Diagram (with Swimlanes): DistillWater
Distiller Example – Block Definition Diagram: Heat Exchanger Flow Ports

Flow Ports (typed by things that flow)

Generic Things That Flow (Blocks)

Generic Subsystems (Blocks)
Distiller Example – Internal Block Diagram: Distiller Initial Design

Parts
(Blocks used in context)

Flow Ports

Connectors

Things That Flow In Context
(ItemFlows)
Distiller Example – Internal Block Diagram: Distiller with Allocation

Exercise for student:
Is allocation complete?
Where is “<objectFlow>of8”??
Distiller Example – Parametric Diagram: Heat Balance Equations

Par {block} Distiller {Simplified Isobaric Heat Balance Analysis}

- **Value Properties**
  - Water in: H2O
  - Water temp: °C
  - Water flow: gm/sec
  - Hx water out: H2O
  - Water temp: °C
  - Water flow: gm/sec
  - Bx steam out: H2O
  - Steam flow: gm/sec
  - Water out: H2O
  - Water flow: gm/sec
  - Heat in: Heat
dQ/dt: cal/sec

- **Value Bindings**
  - r1: equivalent {r1=r2}
  - e1: equivalent {r1=r2}

- **Constraints**
  - Qrate = (th-tc)*mRate/sh
  - Qrate = mRate*lh

- **Note:** Underline these are invariant properties of all uses of H2O

**Distiller Example – Heat Balance Results**

<table>
<thead>
<tr>
<th>Table: IsobaricHeatBalance1 [Results of Isobaric Heat Balance]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>specific heat cal/gm-°C</strong></td>
</tr>
<tr>
<td><strong>latent heat cal/cm</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>water_in</th>
<th>bx_water_in</th>
<th>bx_water_in</th>
<th>bx_steam_out</th>
<th>water_out</th>
</tr>
</thead>
<tbody>
<tr>
<td>mass flow rate gm/sec</td>
<td>6.75</td>
<td>6.75</td>
<td>6.75</td>
<td>6.75</td>
<td>6.75</td>
</tr>
<tr>
<td>temp °C</td>
<td>20</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>dQ/dt cooling water cal/sec</td>
<td>540</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dQ/dt steam-condensate cal/sec</td>
<td>540</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>condenser efficency</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>heat deficit</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dQ/dt condensate-steam cal/sec</td>
<td>540</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>boiler efficiency</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dQ/dt in boiler cal/sec</td>
<td>540</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Satisfies «requirement» WaterSpecificHeat
- Satisfies «requirement» WaterHeatOfVaporization
- Satisfies «requirement» WaterInitialTemp

Note: Cooling water needs to have 6x flow of steam! Need bypass between hx_water_out and bx_water_in!
Distiller Example – Activity Diagram: Updated DistillWater
Distiller Example – Internal Block Diagram: Updated Distiller
Distiller Example – Use Case and Sequence Diagrams

[Diagram of Use Case and Sequence Diagrams showing the interactions between User and Distiller.]
Distiller Example – State Machine
Diagram: Distiller Controller

stm [state machine] Controller_State_Machine [Distiller States]

- **Off**
  - pwrLightOFF
  - bxLevelLow

- **Filling**
  - open feed:Valve
  - NOT bxLevelLow
  - bxHtrON

- **WarmingUp**
  - bxHtrON

- **Distilling**
  - ControllingBoilerLevel
    - NOT bxLevelLow
    - bxLevelHigh
    - LevelLow
      - open feed:Valve
    - LevelOK
      - close drain&fill
    - LevelHigh
      - open drain:Valve
    - bxLevelLow
    - NOT bxLevelHigh

- **ControllingBoilerResidue**
  - residueTimer
  - PurgingResidue
    - open drain:Valve

- **CoolingOff**
  - bxHtrOff
  - open drain:Valve
  - open fill:Valve
  - bxTemp<100degrees

- **Draining**
  - open drain:Valve
  - bxLevelLow

- **BuildingUpResidue**
  - close drain:Valve
  - bxHtrON

- **shutDown**
  - bxHtrON
  - bxTemp=100degrees

- **NOT bxLevelLow**
  - bxLevelHigh
  - NOT bxLevelLow

7/26/2007