Using SysML and Assurance Case in a Robotic Application: Case Study & Tool Demonstration

Kenji Hiranabe
CEO, Change Vision, Inc
Agenda

• Introduction
• Background and Goals
• SysML + RTC (Robotics Technology Component)
• Demo
• SysML + Assurance Case
• Conclusion
Robotics in today’s Component Workshop

1400-1425
Robotics Technology Component Specification

Noriaki Ando    Geoffrey Biggs

Honda R&D Team

1425-1425
Finite State Machine for Robotics Technology Component (FSM4RTC) RFP

Makoto Sekiya    Toyotaka Torii

1620-1645
Using SysML and Assurance Case in a Robotic Application: Case Study & Tool Demonstration

Kenji Hiranabe    Toshiki Iwanaga
<table>
<thead>
<tr>
<th>Name</th>
<th>Vendor</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenRTM-aist</td>
<td>AIST</td>
<td>C++, Python, Java</td>
</tr>
<tr>
<td>OpenRTM.NET</td>
<td>SEC</td>
<td>.NET(C#, VB, C++/CLI, F#, etc..)</td>
</tr>
<tr>
<td>miniRTC, microRTC</td>
<td>SEC</td>
<td>RTC implementation for CAN•ZigBee based systems</td>
</tr>
<tr>
<td>Dependable RTM</td>
<td>SEC/AIST</td>
<td>Functional safety standard (IEC61508) capable RTM implementation</td>
</tr>
<tr>
<td>RTC CANOpen</td>
<td>SIT, CiA</td>
<td>Standard for RTC mapping to CANOpen by CiA (Can in automation) and implementation by SIT</td>
</tr>
<tr>
<td>PALRO</td>
<td>Fuji Soft</td>
<td>C++ PSM implementation for small humanoid robot</td>
</tr>
<tr>
<td>OPRoS</td>
<td>ETRI</td>
<td>Developed by Korean national project</td>
</tr>
<tr>
<td>GostaiRTC</td>
<td>GOSTAI, THALES</td>
<td>C++ PSM implementation on URBI</td>
</tr>
<tr>
<td>Honda R&amp;D RTM</td>
<td>Honda R&amp;D</td>
<td>C++, Python. FSM Component.</td>
</tr>
</tbody>
</table>
Who am I?

- Kenji Hiranabe, Change Vision, Inc. (maker of Astah)
- Astah is a UML editor popular in Japan
  - http://astah.net/
- Astah/SysML
  - Newly developed
  - Focused on “Usability”
- RTC (Robotic Technology Component) plug-in
  - Plug-in for Astah/SysML to generate RTC.xml to OpenRTM
• **SysML + RTC**
  Evaluate how SysML can help design a component-based robotic application using a simple sample problem.

• **SysML + Assurance Case**
  Try an effective usage of Assurance Case with the above SysML-based component model.
Demonstrate the movements (Spiral and Back-and-Forth) by controlling multiple autonomous robots from externally. Operator can switch between the autonomous mode and demonstration mode. Hardware architecture is already known, we use Roomba with PC that can control it using Wi-Fi and use Kinect to switch the mode.
SysML +
RTC(Robotic Technology Component)
Process overview

Analysis

- astah SysML
  - SysML Requirements
  - SysML Use cases
  - SysML Context (Block)
- RTC Plugin
  - Component spec.
    - RTC.xml
    - RTS.xml

Design

- OpenRTM-aist
  - RTCBuilder
    - RTCs
  - RTSSystemEditor
    - Executable RTC
      - RTC source codes (Skeleton)
      - Restore connectors
  - another RTM
    - astah RTM
      - RTC FSMs
      - RTCs
      - Executable RTC
      - RTC source codes (Skeleton)

Implementation
Analysis and Design Diagrams in Astah / SysML(Demo)
**Demonstration by robots**

- **Id**: DR001
- **text**: Demonstrate the movements (figure-eight and back-and-forth movements) by controlling multiple autonomous robots from externally. Operator can switch between the autonomous mode and demonstration mode.
- Hardware architecture is already known, we use Roomba with PC that can control it using WiFi and use Kinect to switch the mode.

**Provide demonstration synchronously**

- **Id**: DR002
- **text**: Perform a synchronized demonstration of the movements (figure-eight and back-and-forth movements) by controlling multiple robots.

**Control of Roomba**

- **Id**: DR004
- **text**: Use Roomba as a demonstration robot which can be controlled from externally and put the WiFi-enabled PC on it to control it.

**Operator control**

- **Id**: DR003
- **text**: Operator can switch between the autonomous mode and demonstration mode.

**Autonomous drive**

- **Id**: DR006
- **text**: Robot can run on the autonomous mode.

**Operated by Kinect**

- **Id**: DR005
- **text**: Use Kinect to switch between autonomous mode and the demonstration mode.
req [Robot requirements]

- **Control of Roomba**
  - Id: CR004
  - Text: Use Roomba as a demonstration robot which can be controlled from externally and put the unit's enabled PC on it to control it.

- **Obstacle detection**
  - Id: PR030
  - Text: Controlling all obstacles is available only on autonomous mode.

- **Autonomous drive**
  - Id: PR031
  - Text: Robot can run on the autonomous mode.

- **Forward/reverse motion**
  - Id: PR032
  - Text: Robot can move forward and reverse.

- **Sharp turning**
  - Id: PR033
  - Text: Robot shall be capable of turning in a sufficient range of the conference hall's demonstration space.

- **Provide demonstration synchronously**
  - Id: CR002
  - Text: Perform synchronized demonstration of the movements (figure-eight and back-and-forth movements) by controlling multiple robots.

- **Speed control**
  - Id: PR034
  - Text: Speed of movement of robot should be controllable.

- **Stop immediately**
  - Id: PR032
  - Text: It shall stop the motion without an excess of the inertia when its mode is switched.
Operator control

Id = DR003

text = Operator can switch between the autonomous mode and demonstration mode.

Provide demonstration synchronously

Id = DR002

text = Perform a synchronized demonstration of the movements (figure-eight and back-and-forth movements) by controlling multiple robots.

Operated by Kinect

Id = DR005

text = Use Kinect to switch between autonomous mode and the demonstration mode.

Controller

Kinect

req [Controller requirements]
Robot Demo System

Operator

Autonomous drive

Move spiral

Move back and forth

<<block>>
<<external>>
Obstacle

<<block>>
<<system_of_interest>>
Controller

<<block>>
<<system_of_interest>>
Demonstration Robot
ibd [block] Controller [Physical structure]

kinect : Kinect

EvLetsDancing : TimedDouble
EvNext : TimedDouble
EvOff : TimedDouble
EvOn : TimedDouble
EvWaiting : TimedDouble

coordinator : Coordinator

drive : -DriveVelocity
out : -DriveVelocity

servicePort

RobotService
stm [State machine of controller]
• SysML “Block”s map to “RTC”s nicely.
• <<Satisfy>> relationships between “Requirements” and “Components” can be visualized to show the intentions of components reasonably.
• An Easy-to-use tool(Astah/SysML) boosted effectiveness of modeling.
• Communication between teams worked well using web-based model sharing feature of the tool.
SysML + Assurance Case
Assurance Case + SysML

- Try to find an effective combination of SysML and Assurance Case

**Assurance Case**
Goal, Sub-Goal, Context, Strategy, Evidence

**SysML**
Requirement, block, behavior, usecase
Stakeholders expressed as actors in use case model.

Stakeholders expressed as Context to the top goal.
Functional and non-functional requirements Expressed as context to a goal.
Goal: G_21
A robot can run back and forth; the demonstration of moving a figure-of-eight shall be done.

Context: C_15
SysML/<<requirement>>
CR002: Spiral drive

<<requirement>>
Spiral drive
Id = CR002
text = Robots run a figure-of-eight.

<<block>>
Roomba

Traceability from/to Goal-Context-Requirement-Block
Domain Model(SysML) and Context(Assurance Case)

System Context expressed as Domain model(block diagram)

Context to the top level goal
Conclusion

• SysML expresses a system’s design view whereas Assurance Case expresses a system’s discussion/explanation view to make a case.

• They can be used together to;
  – Show a AC-context as a SysML-model
  – Make traceability between cases and designs

• A tool which can draw both and set relationship between them will be useful.
• Visit us at the Exhibition during this OMG Meeting
• Visit us at [http://astah.net](http://astah.net) for free evaluation of the tool.