Introduction to Finite State Machine for Robotic Technology Components (FSM4RTC)

Makoto Sekiya, Honda R&D Co., Ltd.
Toyotaka Torii, Honda R&D Co., Ltd.
Noriaki Ando, AIST
Motivation

• We developed a GUI tool to generate code of a FSM from UML Statechart and incorporated it with RTC by an extension (demonstrated in the last Burlingame Robotics Information Day)

• We want to define methods/operations to communicate with FSM components and data model to reuse/exchange structures of FSMs.
Our FSM Component

- **FSM is modeled with UML Statechart**
  - Source code is automatically generated
- **Embedded FSM in data flow Component**
  - Executable under a limited resource platform
  - Events are input from ports
- **Operations to communicate with the GUI tool are added**
  - Notification methods, the interface to get the current state
Why Statechart?

• Lots of robotic software includes FSMs in its code
• Coding a FSM by hand makes things bad
  – It’s impossible to maintain a complex FSM
  – Only you can understand it
    • For a few months...
• UML Statechart can reduce development cost and improve quality of robotic software

Our efforts in the reusable way!
Example of Statechart

Graphical notation with hierarchy reduce complexity of a FSM!
From graphical notation to implementations

Automatic generation of implementations is important for Model Based Development
Steps to develop FSM Component

1. Model a FSM as a UML Statechart diagram
2. Generate source code of the FSM
3. Implement actions invoked with events/transitions
4. Test the FSM
5. Reuse it
Steps to develop FSM Component

1. ✔ Model a FSM as a UML Statechart diagram
2. ✔ Generate source code of the FSM
3. □ Implement actions invoked with events/transitions
4. □ Test the FSM
5. □ Reuse it

UML design tool supports step 1 and 2
Steps to develop FSM Component

1. Model a FSM as a UML Statechart diagram
2. Generate source code of the FSM
3. Implement actions invoked with events/transitions
4. Test the FSM
5. Reuse it

Users implement actions as step 3
Steps to develop FSM Component

1. Model a FSM as a UML Statechart diagram
2. Generate source code of the FSM
3. Implement actions invoked with events/transitions
4. Test the FSM
5. Reuse it

Extension to RTC is required
How to test a FSM

1. Prepare the diagram or the structure definition format of a FSM
2. Check if the FSM is in its initial state
3. Send a test event to the FSM
4. Get notifications of state transitions and actions
5. Check if the notifications are correct compared to the diagram or the structure definition format

The diagram or the structure definition format and interfaces to get the current state/notifications are required to test a FSM
How to reuse a FSM

1. Extract the structure definition format of a FSM
2. Convert the structure to the diagram
   OR
1. Just reuse the diagram used to generate the source code

The diagram or the structure definition format is required to reuse a FSM
Current RTC specification requires rich resources to execute a FSM
Current RTC specification required extension

Methods to send an event with parameters

Methods to know the current state of a FSM

A data model describes the structure of a FSM
Steps to develop FSM Component

1. Model a FSM as a UML Statechart diagram
2. Generate source code of the FSM
3. Implement actions invoked with events/transitions
4. Test the FSM
5. Reuse it

All steps will be realized with FSM4RTC
Concept

FsmService

get_fsm_model(fsm_model,..)
get_fsm_state() :

Data Model for definition of a FSM structure (ex. SCXML)

Implementation of FSM is hidden from outside of the component
Thank you!