

Navigating the rover with xtUML

MDETools'18 Challenge Problem

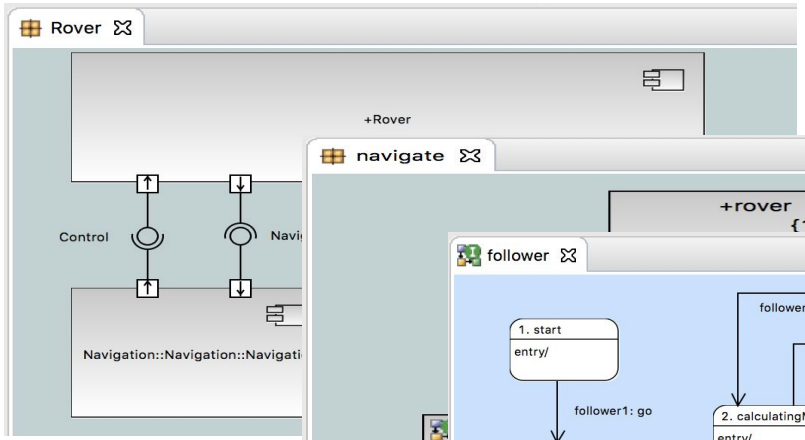


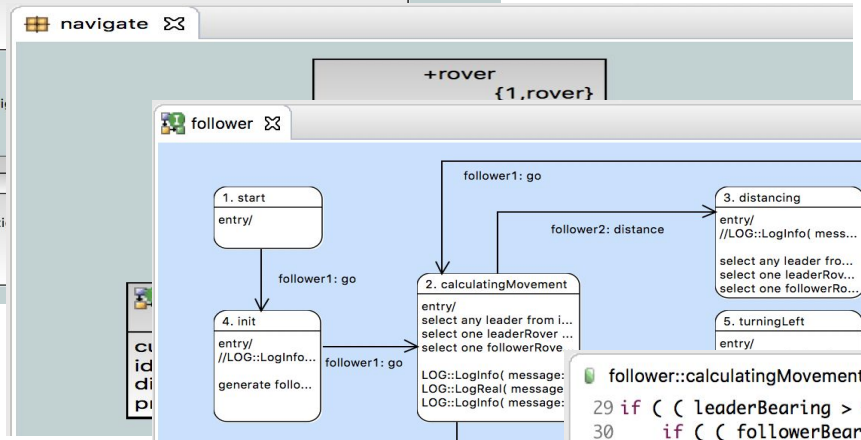
BridgePoint xtUML

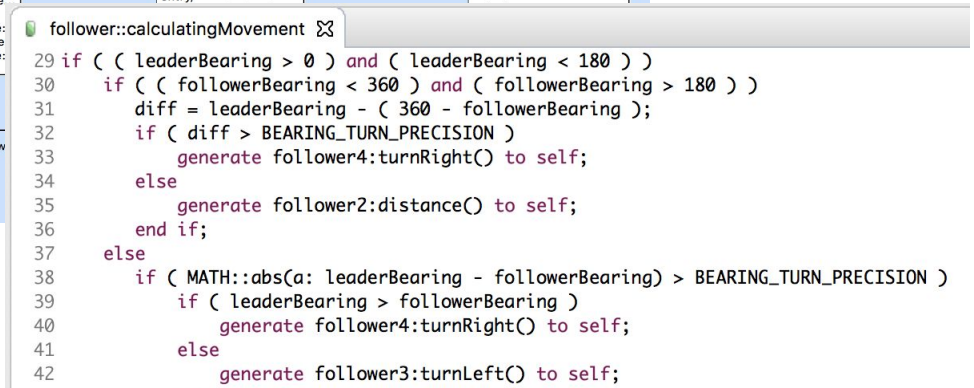


- Eclipse-based UML2.x diagram editor
- xtUML model execution (interpreter)
- xtUML model translators (model compilers)
- Significant body of existing, running, translating models in production
- Various model compilers (C, C++, Ada, SystemC, Java)
- Tool is self-hosted (BridgePoint tool is xtUML model)
- Open source: Eclipse plugins, pyrsl, MCs

xtUML Flow - Data, Control, Processing







Component Diagram

- Decompose the application
- Define interfaces

State Diagram

- Functional lifecycle
- Event handling

Class Diagram

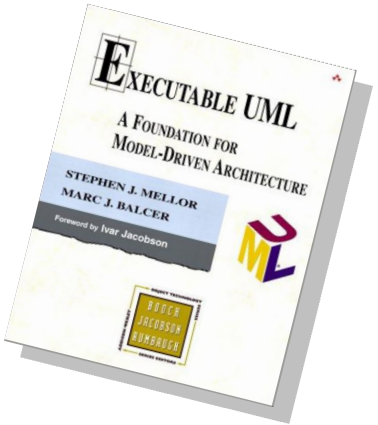
- Abstractions
- Associations
- Operations

Action Specification

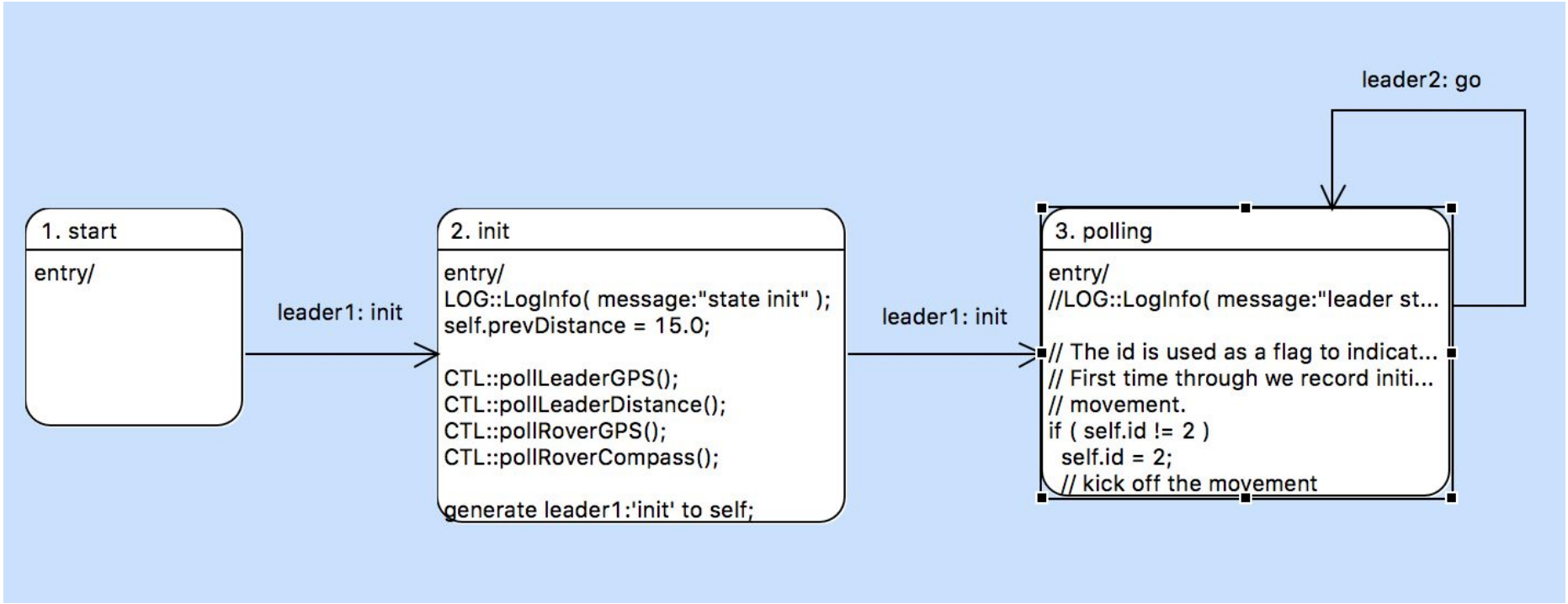
- Processing

Platform Independent Model (PIM)

- Retarget application models across platforms
- Model compilers and architecture are application agnostic and reusable
- Standards compliance
- Action languages are key to PIMs
  - Raise the level of abstraction
  - Structure and/or semantics
  - OAL vs MASL

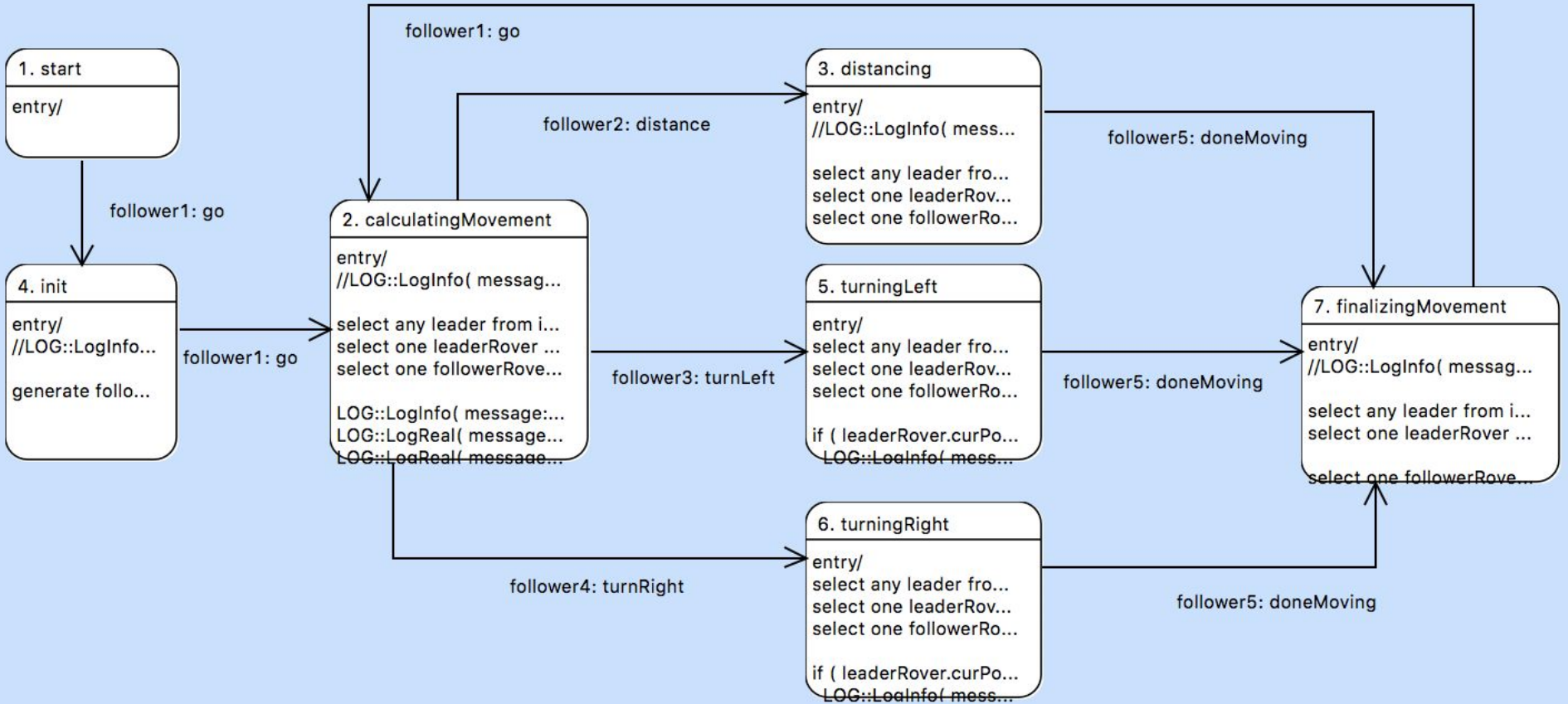


Data Update


leader2: go
1. start
entry/
leader1: init
2. init
entry/
LOG::LogInfo(message:"state init");
self.prevDistance = 15.0;
CTL::pollLeaderGPS();
CTL::pollLeaderDistance();
CTL::pollRoverGPS();
CTL::pollRoverCompass();
generate leader1:'init' to self;
leader1: init
3. polling
entry/
//LOG::LogInfo(message:"leader st...
// The id is used as a flag to indicat...
// First time through we record initi...
// movement.
if (self.id != 2)
self.id = 2;
// kick off the movement

Control Model





Processing



- Data comparison and manipulation
- Send events to transition states

*follower::calculatingMovement

```
leaderBearing = MATH::bearing(x1:leaderRover.oldPosition.x, x2:leaderRover.curPosition.x, y1:leaderRov
followerBearing = self.bearing;

if ( ( leaderBearing > 0 ) and ( leaderBearing < 180 ) )
    if ( ( followerBearing < 360 ) and ( followerBearing > 180 ) )
       diff = leaderBearing - ( 360 - followerBearing );
       if ( diff > BEARING_TURN_PRECISION )
           generate follower4:turnRight() to self;
       else
           generate follower2:distance() to self;
       end if;
    else
       if ( MATH::abs(a: leaderBearing - followerBearing) > BEARING_TURN_PRECISION )
           if ( leaderBearing > followerBearing )
               generate follower4:turnRight() to self;
           else
               generate follower3:turnLeft() to self;
           end if;
       else
           generate follower2:distance() to self;
       end if;
    end if;
```

Interfacing with Handcraft Code



- External Entities (EEs)
  - Best for simple function calls
  - Time
  - Logging
  - Math
- Realized Components
  - Best for modeled interfacing with bigger chunks of functionality
- Both are supported by Model Verifier

RComm Realized Component



- Java implementation for Verifier
- C version for host machine
- C version for Lego EV3

RComm.java

```
81      @Override
82      public void pollLeaderDistance( ComponentIn
83          String reply = ref.send( "Leader,Distan
84          float dist = 0;
85          try {
86              Matcher matcher = replyPattern.matcl
87              if ( matcher.matches() ) {
88                  dist = Float.parseFloat( matche
89              }
90              else CorePlugin.err.println( "Match
91          }
92          catch ( NullPointerException e ) {}
93          catch ( NumberFormatException e ) {}
94          location_data_to_provider.leaderDistanc
95      }
96
97      @Override
98      public void pollRoverGPS( ComponentInstance
99          String reply = rover.send( "Rover,GPS()
100         float x = 0;
101         float z = 0;
```

RComm.c

```
195 void
196 RComm_RComm_setForwardPower( const i
197 {
198   RComm_send_bt_message("Rover,setFo
199   EV3M_set_power( DEV_MOTOR_LEFT, p_
200   EV3M_set_power( DEV_MOTOR_RIGHT, p
201 }
202
203 /*
204  * Interface:  RComm
205  * Provided Port:  RComm
206  * To Provider Message:  setLRPower
207  */
208 void
209 RComm_RComm_setLRPower( const i_t p_lpower, const i_t p_rpower )
210 {
211   RComm_send_bt_message("Rover,setLRPower(%d,%d)", p_lpower, p_rpower);
212   EV3M_set_power( DEV_MOTOR_LEFT, p_lpower );
213   EV3M_set_power( DEV_MOTOR_RIGHT, p_rpower );
214 }
215
```

RComm.c

```
93  void
94  RComm_RComm_pollLeaderDistance()
95  {
96      if (-1 == leaderSock) { leaderSock = RComm_connect(OBSRV_PORT); }
97
98      int err;
99      char buf[50];
100     sprintf(buf, "Leader,Distance()\n");
101     if ((err = send(leaderSock, buf, strlen(buf), 0)) < 0)
102         handle_error_en(err, "Send error ");
103
104     char socketBuf[CBSIZE];          // file buffer
105     read_line(leaderSock, socketBuf, sizeof(socketBuf));
106
107     r_t ret = 0.0;
108     char *numStart;
109
110     numStart = strstr(socketBuf, "Leader,");
111     if (numStart != NULL) {
112         char *ptr;
113         numStart = socketBuf + strlen("Leader,");
```

Tooling Strengths and Weaknesses



- Strengths
  - xtUML is well suited to state-based operational behavior
  - Interpretive execution
  - Translation
    - Built-in execution architecture
  - IDE features such as search and content-aware assistance
- Weaknesses
  - Barrier to entry for programmers. Have to learn the tool *and* the methodology
  - Lack of built-in libraries
  - Tools have idiosyncrasies and bugs