Getting Started With The M.A.S.S

Ben Gotthold

May 12, 2015

1 Contents

- Running The Simulator
  - Configuration File
  - Ctaems Input
  - Connecting Agents
- Building Agents
  - Basic functionality
  - Message Passing
- Collecting Results
  - Results
  - Ordered Log
  - Sequential Log

2 Running The Simulator

To run the simulator you will need a ctaems simulation file and an appropriate agent. From the command line run: Java -jar simulation.jar <cteams file> <configuration file>.

- Ctaems Input

  A ctaems input file is the first input to the simulator and is required to create a simulation. For an example please look at trunk/runFiles. Currently we support the ctaems features of:

  - earliest_start_time
  - deadline
  - qaf q_or
  - qaf q_and

  - qaf q_sum
  - quality_distribution
  - duration_distribution
  - facilitates
  - hinders
  - enables
  - disables
  - duration_power
  - quality_power

- Configuration File

  The configuration file is an optional plaintext file named config.ini. It is the second input to the simulator. All items in the file exist on their own line. If a configuration file is not provided, the simulator will create one using default values. For an example, please refer to Trunk/runFiles/config.ini. In the configuration file one can specify one or more of the following items:

  - seed
  - outputDestination
  - tickLength
  - maxTicks
  - port

- Connecting Agents

  When the simulator starts it outputs the address and port it is running on as well as a list of the agents it expects to connect. For example: [Agent_Green]

  Server running on 10.0.0.11:9876

  When an agent joins the simulation make sure it has the approprate name in its registration message.

3 Building Agents

- Basic Functionality

  An agent must be able to read and create json strings to be successful in the simulation. The simulator and agents communicate by passing json messages back and forth. Basic agent functionality requires agents to send a registration message when they connect to the simulation. This looks like:

  {"MessageType":"AgentRegistrationMessage","Message":

{"MsgSender":"Agent1","MsgDest":"SIMULATOR"}}

For more information on the different message types, check out the examples in trunk/testdata/aisim/simulation/communication

- Message passing

  Agents have the ability to pass json messages to each other with any content they desire. For example:

  { "MessageType":"AgentToAgentMessage", "Message": { "MsgSender":"Agent_Green", "MsgDest":"Agent_Red", "Content": { "Content1":1, "Content2":2 } } }

4 Collecting Results

- Results

  At the end of a simulation some basic results are returned to the user. For example:

  ```
  {
  End of simulation.
  Results:
  Final Duration: 10
  Final Quality: 500.0
  Log files can be located at:
  /Users/logs/Log_Ordered_2015_May_12_10_22_23.txt
  /Users/logs/Log_Sequential_2015_May_12_10_22_23.txt
  }
  ```

  These results can be used to determine the quality and duration of the simulation as well as the location of the log files.

- Ordered Log

  The Ordered Log produces a human-readable summary of the simulation which includes the methods that were completed, the relationships that were activated, as well as a breakdown of what each agent did.

- Sequential Log

  The Sequential Log produces a detailed transcript of all the messages that were passed as well as the time methods and tasks were completed and the time relationships were activated.