Jason

a Java-based interpreter for an extended version of AgentSpeak

EMATM0042 – Intelligent Information Systems
Monday 18 March – Part 2

Content:

1. Extensions to AgentSpeak language

- Beliefs
  - Annotation: to provide details that are strongly associated with one particular belief
  - Strong Negation: explicitly believes somethings to be false
  - Belief-base Rules: as in Prolog
  - Expressions: as in Prolog

- Actions
  - Internal actions: run internally within an agent rather than change the environment

2. How to use Jason

- Jason IDE or Jason Eclipse Plugin
- Hello World Agent
- Multi-agent systems
- Virtual Environment and Environment Actions using Jason+Java

Belief Annotation

Note: Not strictly necessary for implementing meaningful Jason program

- Annotated predicate

  $predicate(t_1, \dots, t_n)[a_1, \dots, a_m]$

  where a_i are first order terms

- Example

  $busy(john)[expires(autumn)]$

  The agent believes that John is busy, but as soon as the autumn starts, that should be no longer believed to hold.

- Special notations of Jason

$source(percept)$ → meaning → perceived beliefs → example → $location(waste, MVB)[source(percept)]$

$source(name\_other\_agent)$ → meaning → communicated beliefs → example → $likes(john, music)[source(john)]$

$source(self)$ → meaning → mental notes → example → $cleaned(MVB, date(2019, 3, 18)[source(self)]$

Belief Negation

Closed World Assumption

Anything not believed to be true is assumed to be false

- AgentSpeak: the belief base is a set of atoms
  - Known positives
    - "b": b is known to be true
  - Known negatives
    - "not b": b is assumed to be false

Negation

Open World Assumption

Anything not believed to be true or false is simply unknown

- Jason: the belief base is a set of literals
  - Known positives
    - "b": b is known to be true
  - Known negatives
    - "~ b": b is known to be false
  - Unknown positives
    - "not b": b is not known to be true
  - Unknown negatives
    - "not ~ b": b is not known to be false

Strong Negation: ~

Weak Negationx: not

Belief Negation

Jason

Strong Negation: ~

Weak Negation: *not*

Syntax	Meaning
l	The agent believes l is true
$\sim l$	The agent believes l is false
`not` l	The agent does not believe l is true
`not` $\sim l$	The agent does not believe l is false

Belief-base Rules

Support the prolog-like rules in the belief base

Jason

```
/* Facts */

female(alice).
male(bob).
female(carol).
male(dave).
female(eve).

parent(alice, carol).
parent(bob, carol).
parent(carol, eve).
parent(dave, eve).

/* Rules */

mother(X, Y) :- parent(X, Y) & female(X).
father(X, Y) :- parent(X, Y) & male(X).

grandparent(X, Z) :- parent(X, Y) & parent(Y, Z).

/* Inferences */

// |= mother(alice, carol)
// |= mother(carol, eve)
// |= father(bob, carol)
// |= father(dave, eve)

// |= grandparent(alice, eve)
// |= grandparent(bob, eve)
```

Relational Expressions in the Context

- Example:
  - $>=$: greater equal than
  - $<=$: less equal than
  - $>$: greater than
  - $<$: less than
  - $=$: equal to
  - $\==$: different from
  - $=$: unified

- Plan Example:

```
+!buy(something) : not ~legal(something) & price(something, P) & bank_balance(B) & B > P <- buy(something).
```

the agent does not believe that the legalness of something is false.

the agent has enough money to buy such something.

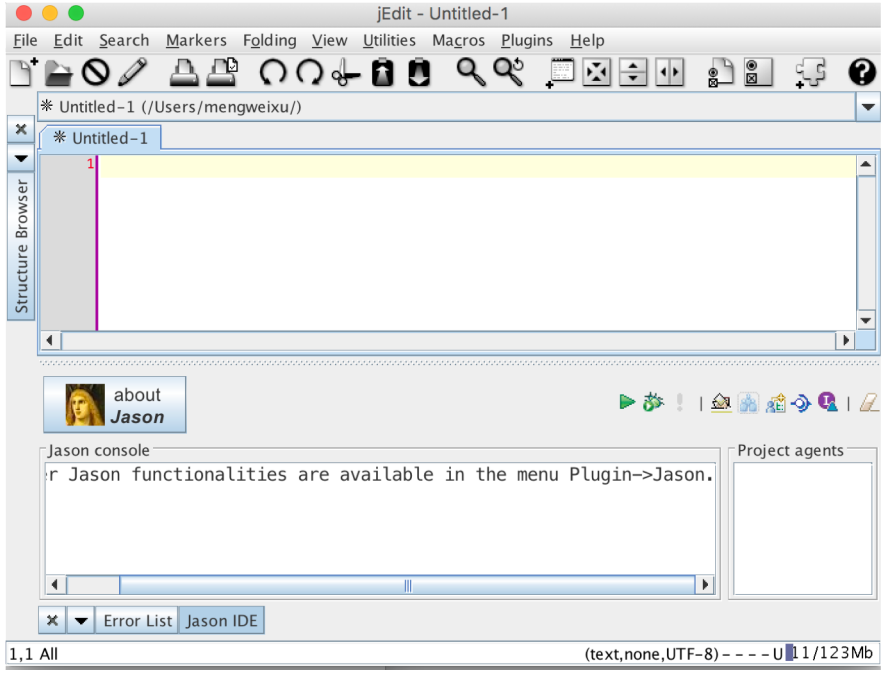
Internal Actions

- Definition: Internal actions, in contrast to (normal) actions, are run internally within an agent (i.e. the agent's mind) rather than change the environment.
- Quick Example: $.print(\text{“}hello\ world\text{”})$ to display text on a programmer's console.
- Syntax: having a ‘ . ’ character with the action name
- Types:
  - Libraries of user-defined internal actions: $lib\_name.action\_name(\dots)$
  - Pre-defined internal action have an empty library name: $.print(hello\ world)$
  - Internal actions for inter-agent communication: .send(..)
- More Examples of BDI-related internal actions:
  - $.desire(literal)$
  - $.intend(literal)$
  - $.drop\_desires(literal)$
  - $.drop\_intentions(literal)$
- link of standard internal action list:

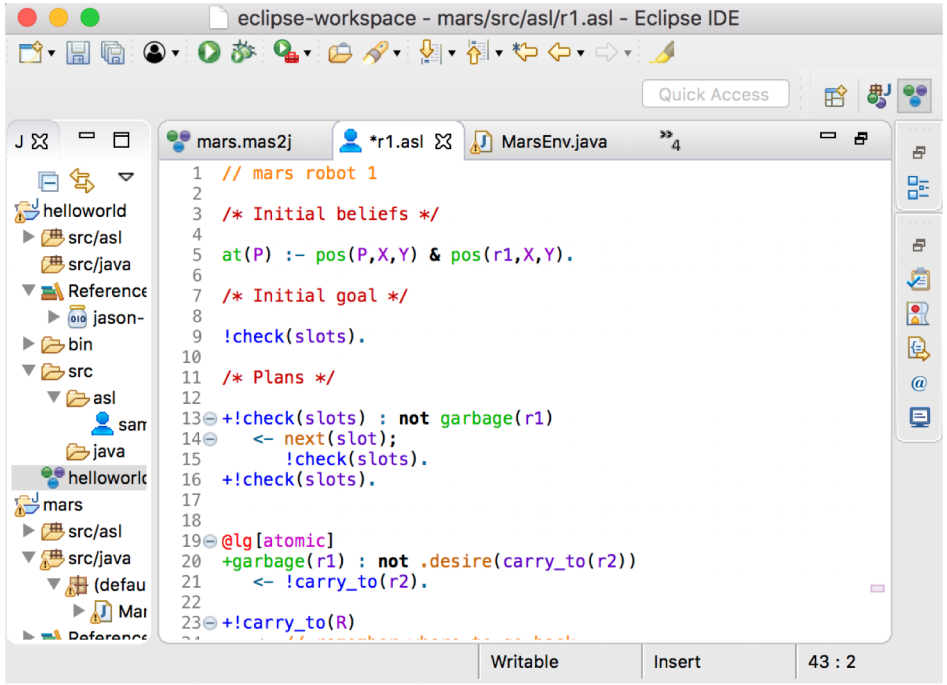
  http://jason.sourceforge.net/api/jason/stdlib/package-summary.html#package.description

Running Jason

- Jason JDE:



- Jason Eclipse Plugin



Instruction link for using Eclipse Plugin: http://jason.sourceforge.net/mini-tutorial/eclipse-plugin/

Hello World Example

- Jason Project: MAS

```
1  MAS helloworld {
2
3      infrastructure: Centralised
4
5      agents:
6          sample_agent;
7
8      aslSourcePath:
9          "src/asl";
10 }
```

The system is called “helloworld”

How the agent system is organised

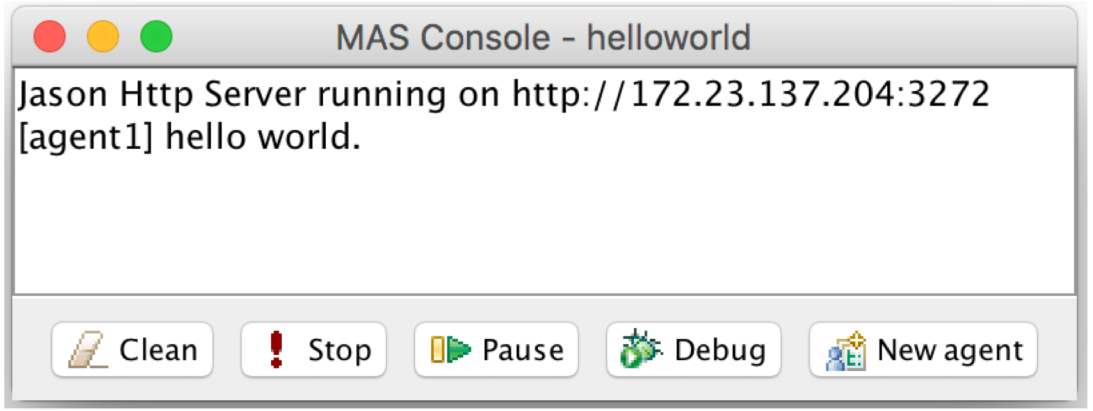
It has one agent called “sample_agent”

Path from the MAS file to the agent descriptions

- Hello agent in Jason project helloworld.mas2j

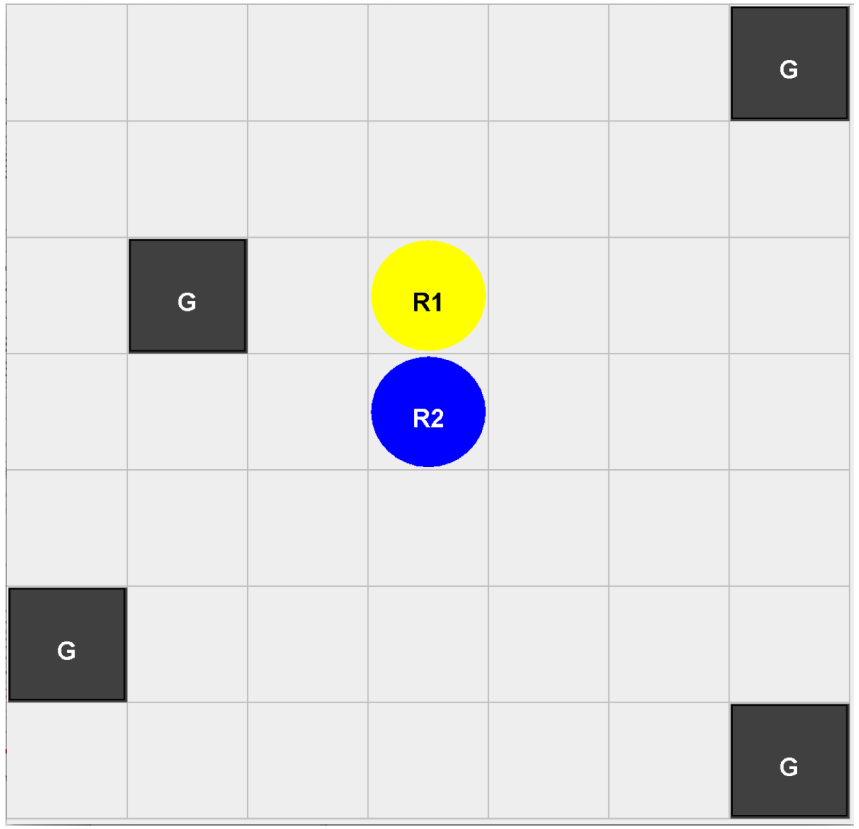
```
1  // Agent sample_agent in project helloworld
2
3  /* Initial beliefs and rules */
4
5  /* Initial goals */
6
7  !start.
8
9  /* Plans */
10
11 +!start : true <- .print("hello world.").
12
```

- “hello world” input



Cleaning Robots Example

Grid world:



Cleaning Task

The robot R1 searches the whole grid for pieces of garbage, and when one is found, it takes it to another robot R2, located in the center of the grid, where there is an incinerator; the moving robot R1 then goes back to the place where the last piece of garbage was found and continues the search from there.

- G --- Garbage
- R1 --- Moving robot
- R2 --- Incinerator robot

Cleaning Robots Example

- Jason Project: MAS

```
1  MAS mars {
2
3      infrastructure: Centralised
4
5      environment: MarsEnv
6
7      agents:
8          r2;
9          r1;
10     aslSourcePath:
11         "src/asl";
12 }
```

The system is called “mars”

How the agent system is organised

It has a virtual environment (using Jason+ Java)

It has two agent called “r2” and “r1”.

Path from the MAS file to the agent descriptions

Cleaning Robots Example

- Agent r2 in Jason project mars.mas2j

```
1  // mars robot 2
2
3  +garbage(r2) : true <- burn(garb).
```

If the agent r2 finds it has the garbage, then it will burn the garbage using environment action burn(garb) (using Jason+Java) in virtual environment MarsEnv.

Cleaning Robots Example

- Agent r1 in Jason project mars.mas2j

```
1  // mars robot 1
2
3  /* Initial beliefs */
4
5  at(P) :- pos(P,X,Y) & pos(r1,X,Y).
6
7  /* Initial goal */
8
9  !check(slots).
10
11 /* Plans */
12
13 +!check(slots) : not garbage(r1)
14    <- next(slot);
15       !check(slots).
16 +!check(slots).
17
18
19 @lg[atomic]
20 +garbage(r1) : not .desire(carry_to(r2))
21    <- !carry_to(r2).
```

if both r1 and P are at (X,Y), we say that r1 at P place

initial goal to search all slots

if no garbage in the current slot, the agent moving to next slot to keep searching.

If there is a garbage, goal !check(slots) does nothing

atomic label plan annotation: this plan is executed without any inference

Once found the garbage, carry to incinerator r2

Cleaning Robots Example

- Agent r1 in Jason project mars.mas2j

```
23 +!carry_to(R)
24     <- // remember where to go back
25        ?pos(r1,X,Y);
26        -+pos(last,X,Y);
27
28        // carry garbage to r2
29        !take(garb,R);
30
31        // goes back and continue to check
32        !at(last);
33        !check(slots).
34
35 +!take(S,L) : true
36     <- !ensure_pick(S);
37        !at(L);
38        drop(S).
39
40 +!ensure_pick(S) : garbage(r1)
41     <- pick(garb);
42        !ensure_pick(S).
43 +!ensure_pick(_).
44
45 +!at(L) : at(L).
46 +!at(L) <- ?pos(L,X,Y);
47            move_towards(X,Y);
48            !at(L).
```

query where the agent is so that it can go back.

deleting the form instance of last location (if any exist) while adding the new instance of last location.

to achieve an goal of picking up S

to achieve an goal of being at L

to perform an action dropping S

ensure that the agent indeed picks up the garbage.

do nothing if already at L

otherwise query where L is, move to the location of L , and double check

Cleaning Robots Example

- The full Java code of MarsEnv.java can be found in the link below

  https://github.com/jason-lang/jason/blob/master/examples/cleaning-robots/MarsEnv.java

- A quick guide what Jason Java codes look like

- A running Jason demon of this cleaning robot example

Cleaning Robots Example

- Environment MarsEnv.java
  - Claim the virtual environment

```
1  import jason.asSyntax.*;
2  import jason.environment.Environment;
3  import jason.environment.grid.GridWorldModel;
4  import jason.environment.grid.GridWorldView;
5  import jason.environment.grid.Location;
6
7  import java.awt.Color;
8  import java.awt.Font;
9  import java.awt.Graphics;
10 import java.util.Random;
11 import java.util.logging.Logger;
12
13 public class MarsEnv extends Environment {
14
15     public static final int GSize = 7; // grid size
16     public static final int GARB  = 16; // garbage code in grid model
17
18     public static final Term    ns = Literal.parseLiteral("next(slot)");
19     public static final Term    pg = Literal.parseLiteral("pick(garb)");
20     public static final Term    dg = Literal.parseLiteral("drop(garb)");
21     public static final Term    bg = Literal.parseLiteral("burn(garb)");
22     public static final Literal g1 = Literal.parseLiteral("garbage(r1)");
23     public static final Literal g2 = Literal.parseLiteral("garbage(r2)");
24
25     static Logger logger = Logger.getLogger(MarsEnv.class.getName());
26
27     private MarsModel model;
28     private MarsView  view;
29
```

Cleaning Robots Example

- Environment MarsEnv.java
  - Define the percept update

```
69
70      /** creates the agents perception based on the MarsModel */
71      void updatePercepts() {
72          clearPercepts();
73
74          Location r1Loc = model.getAgPos(0);
75          Location r2Loc = model.getAgPos(1);
76
77          Literal pos1 = Literal.parseLiteral("pos(r1," + r1Loc.x + "," + r1Loc.y + ")");
78          Literal pos2 = Literal.parseLiteral("pos(r2," + r2Loc.x + "," + r2Loc.y + ")");
79
80          addPercept(pos1);
81          addPercept(pos2);
82
83          if (model.hasObject(GARB, r1Loc)) {
84              addPercept(g1);
85          }
86          if (model.hasObject(GARB, r2Loc)) {
87              addPercept(g2);
88          }
89      }
90
```

Cleaning Robots Example

- Environment MarsEnv.java
  - Define the initial location of agents and garbages

```
private MarsModel() {
    super(GSize, GSize, 2);

    // initial location of agents
    try {
        setAgPos(0, 0, 0);

        Location r2Loc = new Location(GSize/2, GSize/2);
        setAgPos(1, r2Loc);
    } catch (Exception e) {
        e.printStackTrace();
    }

    // initial location of garbage
    add(GARB, 3, 0);
    add(GARB, GSize-1, 0);
    add(GARB, 1, 2);
    add(GARB, 0, GSize-2);
    add(GARB, GSize-1, GSize-1);
}
```

Cleaning Robots Example

- Environment MarsEnv.java
  - Define the environment actions

```
120         void nextSlot() throws Exception {
121             Location r1 = getAgPos(0);
122             r1.x++;
123             if (r1.x == getWidth()) {
124                 r1.x = 0;
125                 r1.y++;
126             }
127             // finished searching the whole grid
128             if (r1.y == getHeight()) {
129                 return;
130             }
131             setAgPos(0, r1);
132             setAgPos(1, getAgPos(1)); // just to draw it in the view
133         }
134 
135         void moveTowards(int x, int y) throws Exception {
136             Location r1 = getAgPos(0);
137             if (r1.x < x)
138                 r1.x++;
139             else if (r1.x > x)
140                 r1.x--;
141             if (r1.y < y)
142                 r1.y++;
143             else if (r1.y > y)
144                 r1.y--;
145             setAgPos(0, r1);
146             setAgPos(1, getAgPos(1)); // just to draw it in the view
147         }
148 
```

DO NOT FEEL OVERWHELMED BY THE JAVA CODES

It is difficult for me too. ☹

Practice is the only way to learn!

Bibliography and Resources

- Rafael H. Bordini, Jomi Fred Hübner, & Michael Wooldridge. *Programming Multi-Agent Systems in AgentSpeak Using Jason*. John Wiley & Sons, 2007.
- Jason Official Website: http://jason.sourceforge.net/wp/
- Jason github link: https://github.com/jason-lang/jason