

AgentSpeak & Jason

Alexandre Zamberlam e Rodrigo Goulart

Baseado no trabalho de Hübner, J.F.; Bordini, R.H. & Vieira, R. intitulado
"Introdução ao desenvolvimento de sistemas multiagentes com Jason."
UNICENTRO, 2004 , 2 , 51-89

Linguagens de programação

- Shoham 1993
 - Programação orientada a agentes
 - Visão social de computação.
 - Paradigma BDI (Belief, Desire and Intention)
 - Agir e pensar de forma racional.
- Mora 2000
 - X-BDI
- Rao 1996
 - AgentSpeak(L)

BDI

- Crenças

- *"... representam aquilo que o agente sabe sobre o estado do ambiente e dos agentes naquele ambiente (inclusive sobre si mesmo)."*

- Desejos

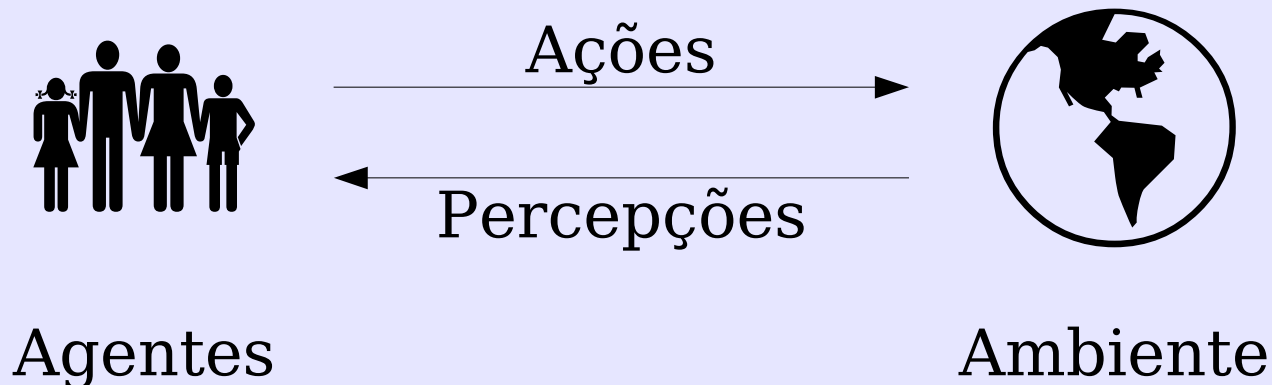
- *"... representam estados do mundo que o agente quer atingir."*

- Intenções

- *"... representam seqüências de ações específicas que um agente se compromete a executar para alcançar determinados objetivos."*

Visão geral deste material

- Teoria → Atitudes Mentais
 - Arquitetura → BDI
 - Linguagem → X-BDI ou AgentSpeak
 - Ferramenta → Interpretador X-BDI ou Jason



AgentSpeak

```
+show(A,L) : gosta(A)
    <- !reserva_ingressos(A,L);

+!reserva_ingressos(A,L) :  $\neg$ ocupado(número)
    <- chamar(número);
...;
!escolher_poltrona(A,L);
```

AgentSpeak

Planos e sub-planos

eventos ativadores externos e internos

Contexto

```
+show(A,L) : gosta(A)
  <- !reserva_ingressos(A,L);

+!reserva_ingressos(A,L) : ¬ocupado(número)
  <- chamar(número);
...;
!escolher_poltrona(A,V);
```

Ações no ambiente

Tipos de negação:

AgentSpeak

- Crenças: predicado(termo)
- Planos: +evento : contexto <- corpo
- Corpo:
 - Ações: ação(termos);
 - Sub-planos: +!novo_objetivo(termos);
 - Falha de planos: -!evento: contexto
 - Testar objetivos: ?crença(termos);
 - Adição de crenças: +crença(termos);
 - Remoção de crenças: -crença(termos);

Jason

File Edit Search Markers Folding View Utilities Macros Plugins Help

Quarto.mas2j (/home/alexandre/bin/jason-1.0/examples/room/)

Quarto.mas2j claustrrofobico.asl paranoico.asl porteiro.asl AmbienteQuarto.java

```

1  /*
2  Neste sistema, dois agentes estao em um quarto. Um deles (claustrrofobico)
3  quer manter a porta sempre destrancada, enquanto o outro (paranoico), quer
4  mante-la trancada.
5
6  O porteiro eh um agente que eh capaz de (des)trancar a porta (ou seja, age no
7  ambiente). Assim, os outros dois agentes solicitam seus servicos.
8  */
9
10 MAS quarto {
11     infrastructure: Centralised
12
13     environment: AmbienteQuarto
14
15     agents: porteiro; claustrrofobico; paranoico;
16 }
17

```

about jason

Jason console

```

compile:

run:
Running project quarto

BUILD SUCCESSFUL

```

Project agents

```

porteiro;
claustrrofobico;
paranoico;

```

Error List Jason IDE

13,29-32 All (mas2j,none,UTF-8) - - - - U 8/10Mb

File Edit Search Markers Folding View Utilities Macros Plugins Help

claustrorobico.asl (/home/alexandre/bin/jason-1.0/examples/room/)

Quarto.mas2j claustrorobico.asl paranoico.asl porteiro.asl AmbienteQuarto.java

```
1 /* Eventos ativadores que vem do ambiente (eventos externos).
2    Planos que ao final de suas execucoes, confirmam as
3    crenças do agente CLAUSTROFOBICO - adicao ou remocao */
4
5 // quando o agente percebe (evento ativador) porta trancada,
6 // solicita a acao de destrancar a porta ao agente porteiro, mas, ao final,
7 // ADICIONA a crença de que a porta estah trancada no ambiente
8 +trancada(porta) : true
9     // pede ao porteiro para destrancar a porta
10    <- .send(porteiro,achieve,destrancada(porta)).
11
12 // quando o agente percebe ausencia da crença de porta trancada,
13 // imprime mensagem no Console e
14 // REMOVE a crença de que a porta estah trancada no ambiente
15 -trancada(porta) : true
16    <- .print("Obrigado por destrancar a porta!").
17
```

about jason

Jason console

```
compile:
run:
Running project quarto

BUILD SUCCESSFUL
```

Project agents

```
porteiro;
claustrorobico;
paranoico;
```

Error List Jason IDE

12,64 All (asl,none,UTF-8) - - - U 8/10MB

File Edit Search Markers Folding View Utilities Macros Plugins Help

◇ paranoico.asl (/home/alexandre/bin/jason-1.0/examples/room/)

◇ Quarto.mas2j ◇ claustrofobico.asl ◇ paranoico.asl ◇ porteiro.asl ◇ AmbienteQuarto.java

```

1  /* Eventos ativadores que vem do ambiente (eventos externos).
2     Planos que ao final de suas execucoes, confirmam as
3     crenças do agente PARANOICO - adicao ou remocao */
4
5  // quando o agente percebe (evento ativador) porta NAO trancada,
6  // solicita a acao de trancar a porta ao agente porteiro, mas, ao final,
7  // ADICIONA a crença de que a porta NAO estah trancada no ambiente
8  +~trancada(porta) : true
9      // pede ao porteiro para trancar
10     <- .send(porteiro,achieve,trancada(porta)).
11
12 // quando o agente percebe porta trancada,
13 // imprime mensagem no Console e
14 // ADICIONA a crença de que a porta estah trancada no ambiente
15 +trancada(porta) : true
16     <- .print("Obrigado por trancar a porta").
17

```

about jason

Jason console

```

compile:

run:
Running project quarto

BUILD SUCCESSFUL

```

Project agents

```

porteiro;
claustrofobico;
paranoico;

```

Error List Jason IDE

13,31 All (asl,none,UTF-8) - - - U 8/10Mb

File Edit Search Markers Folding View Utilities Macros Plugins Help

porteiro.asl (/home/alexandre/bin/Jason-1.0/examples/room/)

Quarto.mas2j claustrofobico.asl paranoico.asl **porteiro.asl** AmbienteQuarto.java

```

1 /* Eventos ativadores que vem dos agentes Paranoico e/ou Claustrofobico.
2    Planos do agente Porteiro que, ao final de suas execucoes, confirmam as
3    crencas porta NAO trancada ou porta trancada. Alem das acoes para
4    trancar ou destrancar a porta no ambiente */
5
6 // quando o agente percebe o subplano !trancada, enviado pelo agente
7 // paranoico, em que acredita que a porta NAO trancada (contexto/condicao),
8 // dispara no ambiente a acao de trancar a porta
9 +!trancada(porta)[source(paranoico)]
10 : ~trancada(porta) // contexto ou condicao
11 <- trancar. //acao para o ambiente
12
13 // quando o agente percebe o subplano !destrancada, enviado pelo agente
14 // claustrofobico, em que acredita que a porta trancada (contexto/condicao),
15 // dispara no ambiente a acao de destrancar a porta
16 +!destrancada(porta)[source(claustrofobico)]
17 : trancada(porta) //contexto ou condicao
18 <- destrancar. //acao para o ambiente
19

```

about Jason

Jason console

```

compile:
run:
Running project quarto

BUILD SUCCESSFUL

```

Project agents

```

porteiro;
claustrofobico;
paranoico;

```

Error List Jason IDE

16,3 All (asl,none,UTF-8) - - - U 8/10MB

