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

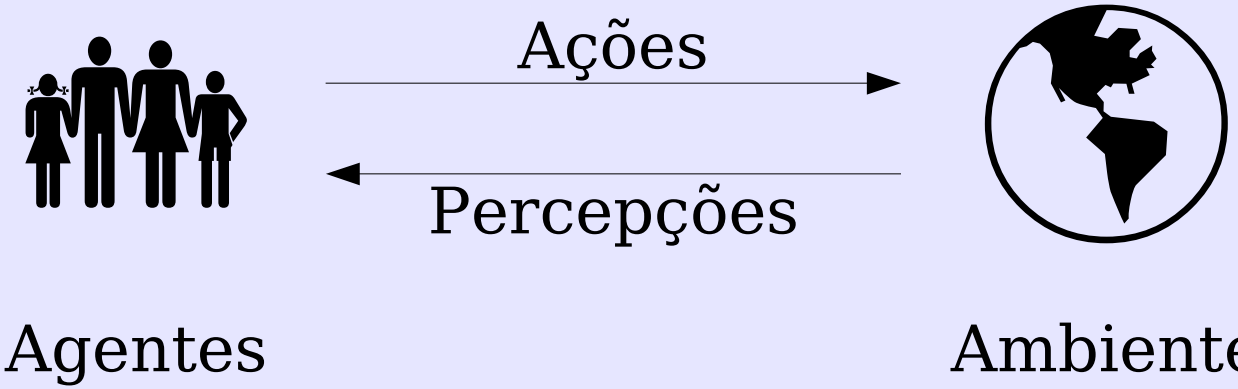
- Shoham 1993
  - Programação orientada a agentes
    - Visão social de computação.
  - Paradigma BDI (Belief, Desire and Intention)
    - Agir e pensar de forma racional.
- Mora 2000
  - X-BDI
- Rao 1996
  - AgentSpeak(L)

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

gosta(“Mano Lima”).
gosta(“Lulu Santos”).
gosta(“Pity”).

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);
```

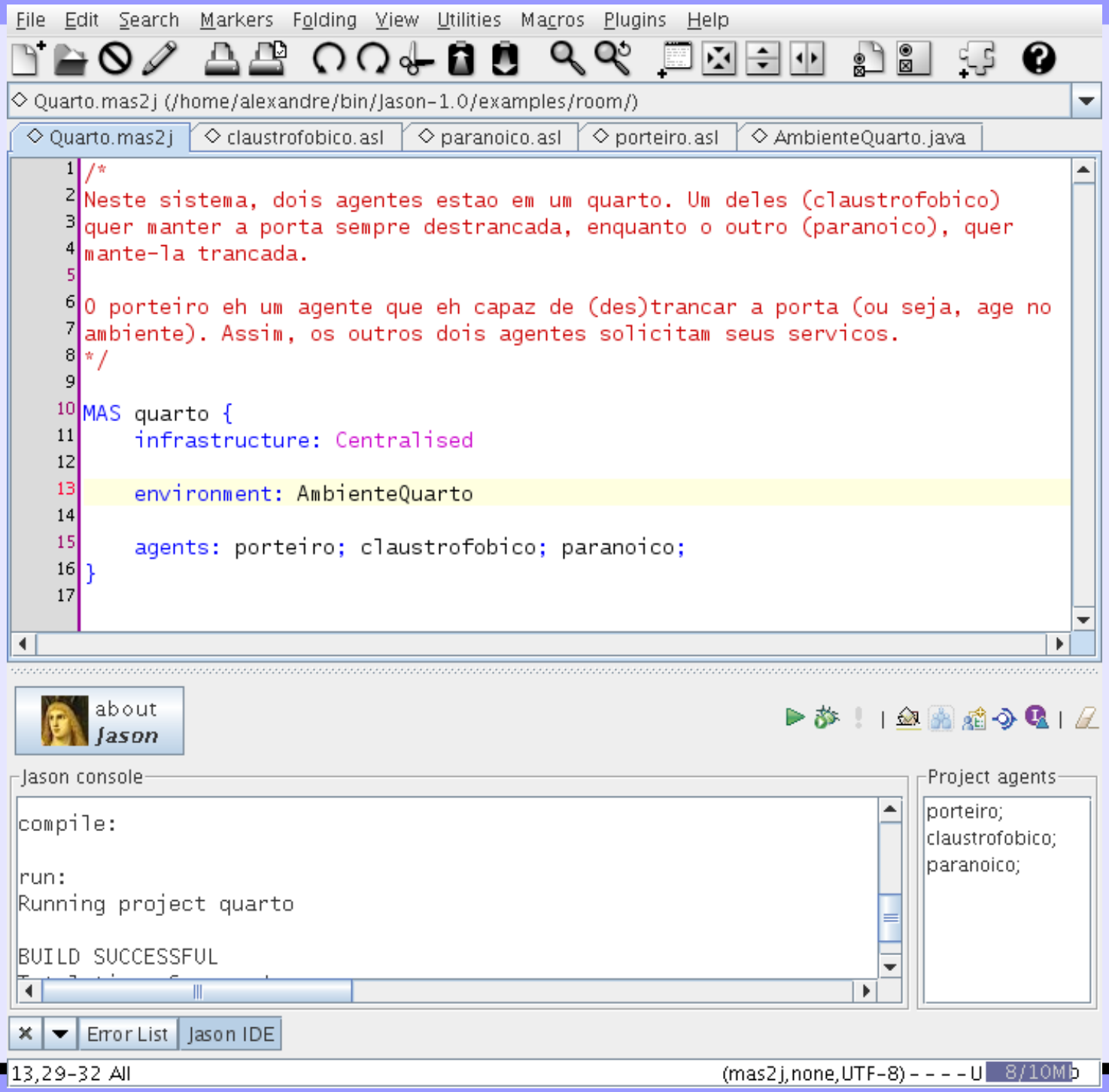
Ações no ambiente

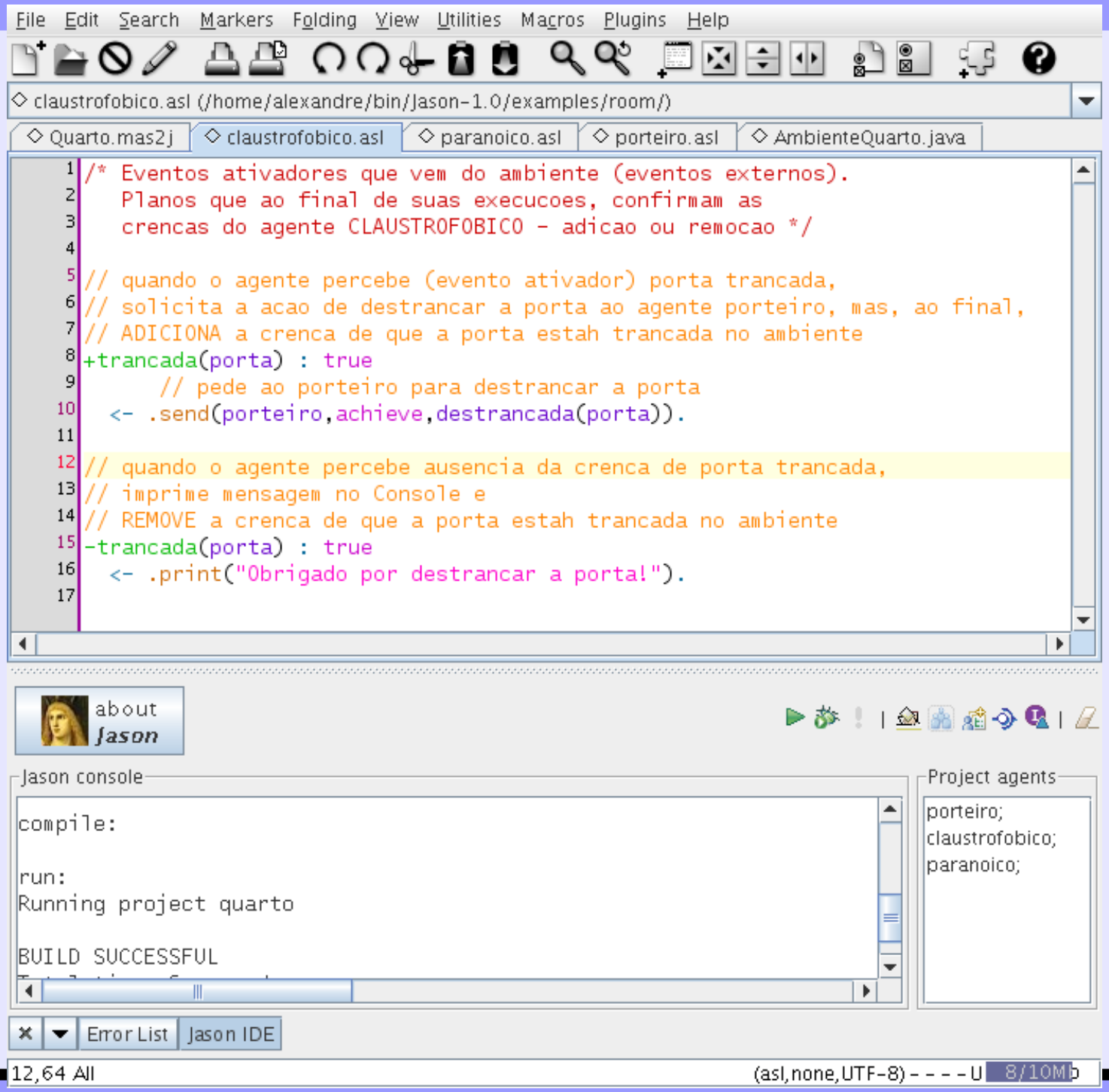
Tipos de negação:

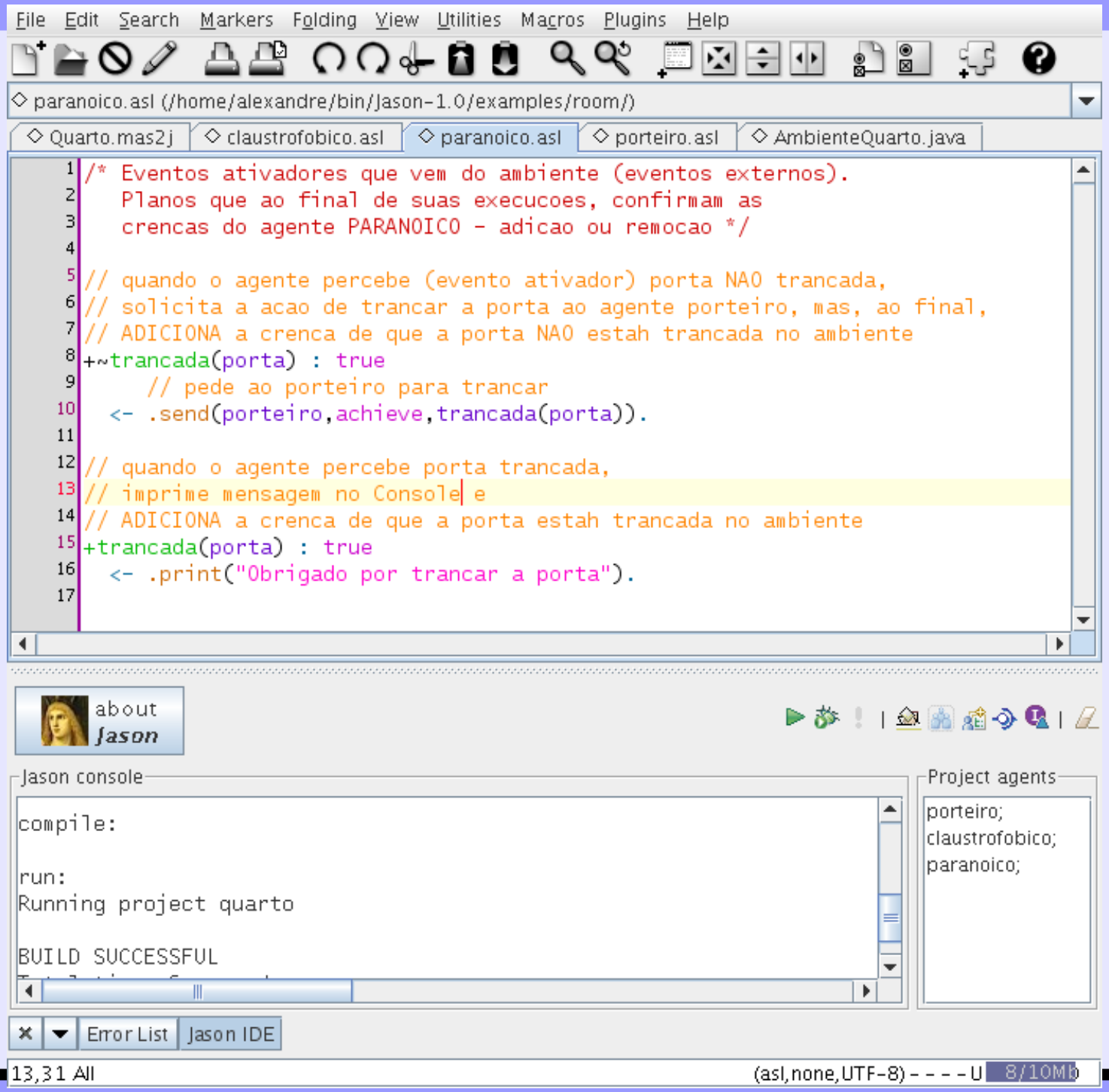
AgentSpeak

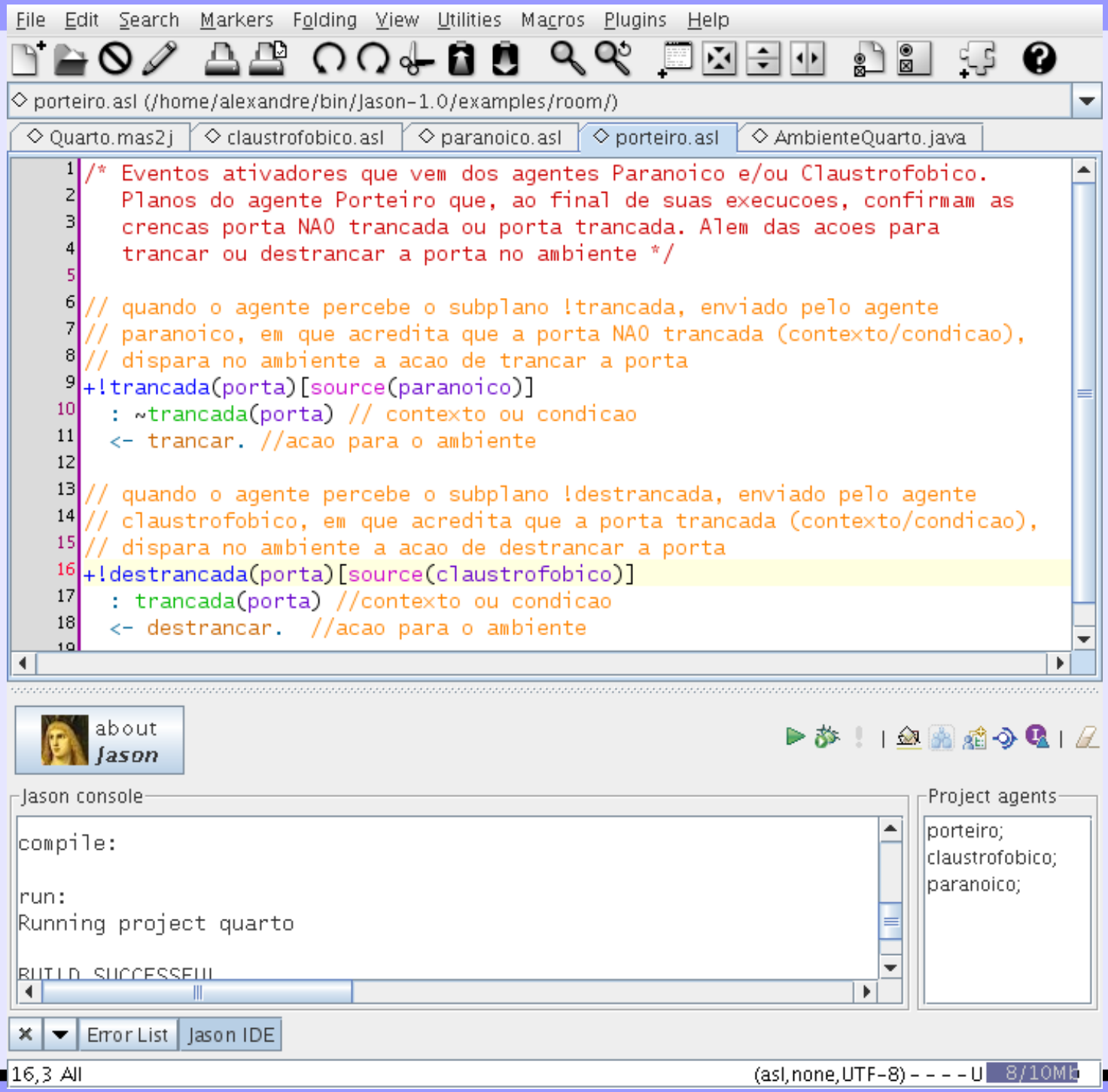
- Crenças: predicado(termo)
- Planos: +evento : contexto <- corpo
- Corpo:
  - Ações: ação(termos);
  - Sub-planos: +!novo_objetivo(termos);
  - Falha de planos: -!evento: contexto
  - Testar objetivos: ?crença(termos);
  - Adição de crenças: +crença(termos);
  - Remoção de crenças: -crença(termos);

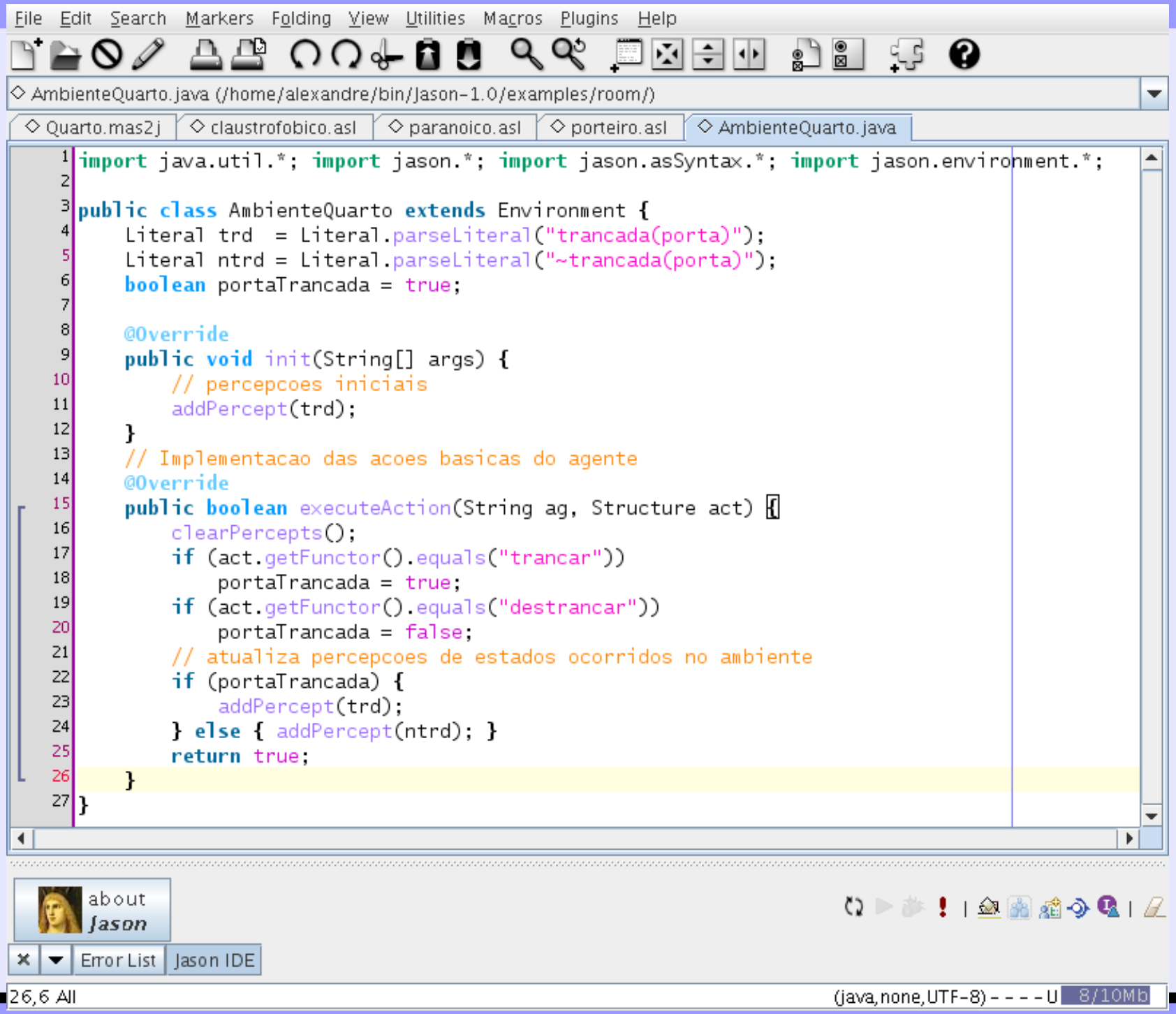
Jason

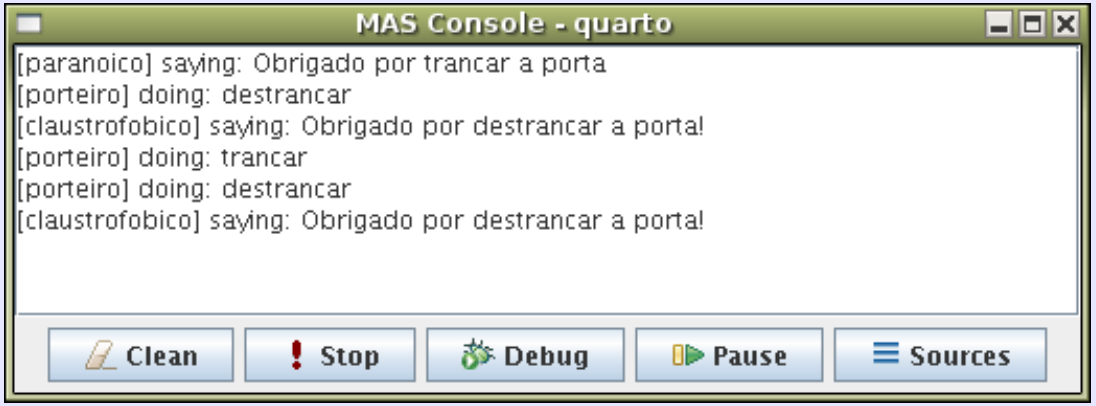

File Edit Search Markers Folding View Utilities Macros Plugins Help
Quarto.mas2j (/home/alexandre/bin/Jason-1.0/examples/room/)
Quarto.mas2j
claustrofobico.asl
paranoico.asl
porteiro.asl
AmbienteQuarto.java
1 /*
2 Neste sistema, dois agentes estao em um quarto. Um deles (claustrofobico)
3 quer manter 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; claustrofobico; paranoico;
16 }
17
about Jason
Jason console
compile:
run:
Running project quarto
BUILD SUCCESSFUL
Project agents
porteiro;
claustrofobico;
paranoico;
Error List
Jason IDE
13,29-32 All
(mas2j,none,UTF-8) - - - - U 8/10M


File Edit Search Markers Folding View Utilities Macros Plugins Help
claustrofobico.asl (/home/alexandre/bin/Jason-1.0/examples/room/)
Quarto.mas2j
claustrofobico.asl
paranoico.asl
porteiro.asl
AmbienteQuarto.java
/* Eventos ativadores que vem do ambiente (eventos externos).
Planos que ao final de suas execucoes, confirmam as
crencas do agente CLAUSTROFOBICO - adicao ou remocao */
// quando o agente percebe (evento ativador) porta trancada,
// solicita a acao de destrancar a porta ao agente porteiro, mas, ao final,
// ADICIONA a crenca de que a porta estah trancada no ambiente
+trancada(porta) : true
// pede ao porteiro para destrancar a porta
<- .send(porteiro,achieve,destrancada(porta)).
// quando o agente percebe ausencia da crenca de porta trancada,
// imprime mensagem no Console e
// REMOVE a crenca de que a porta estah trancada no ambiente
-trancada(porta) : true
<- .print("Obrigado por destrancar a porta!").
about Jason
Jason console
compile:
run:
Running project quarto
BUILD SUCCESSFUL
Project agents
porteiro;
claustrofobico;
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
/* Eventos ativadores que vem do ambiente (eventos externos).
Planos que ao final de suas execucoes, confirmam as
crencas do agente PARANOICO - adicao ou remocao */
// quando o agente percebe (evento ativador) porta NAO trancada,
// solicita a acao de trancar a porta ao agente porteiro, mas, ao final,
// ADICIONA a crenca de que a porta NAO estah trancada no ambiente
+~trancada(porta) : true
// pede ao porteiro para trancar
<- .send(porteiro,achieve,trancada(porta)).
// quando o agente percebe porta trancada,
// imprime mensagem no Console e
// ADICIONA a crenca de que a porta estah trancada no ambiente
+trancada(porta) : true
<- .print("Obrigado por trancar a porta").
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
/* Eventos ativadores que vem dos agentes Paranoico e/ou Claustrofobico.
Planos do agente Porteiro que, ao final de suas execucoes, confirmam as
crencas porta NAO trancada ou porta trancada. Alem das acoes para
trancar ou destrancar a porta no ambiente */
// quando o agente percebe o subplano !trancada, enviado pelo agente
// paranoico, em que acredita que a porta NAO trancada (contexto/condicao),
// dispara no ambiente a acao de trancar a porta
+!trancada(porta)[source(paranoico)]
: ~trancada(porta) // contexto ou condicao
<- trancar. //acao para o ambiente
// quando o agente percebe o subplano !destrancada, enviado pelo agente
// claustrofobico, em que acredita que a porta trancada (contexto/condicao),
// dispara no ambiente a acao de destrancar a porta
+!destrancada(porta)[source(claustrofobico)]
: trancada(porta) //contexto ou condicao
<- destrancar. //acao para o ambiente
about Jason
Jason console
compile:
run:
Running project quarto
Project agents
porteiro;
claustrofobico;
paranoico;
Error List
Jason IDE
16,3 All
(asl,none,UTF-8) - - - - U 8/10Mb


File Edit Search Markers Folding View Utilities Macros Plugins Help
AmbienteQuarto.java (/home/alexandre/bin/Jason-1.0/examples/room/)
Quarto.mas2j
claustrofobico.asl
paranoico.asl
porteiro.asl
AmbienteQuarto.java
import java.util.*; import jason.*; import jason.asSyntax.*; import jason.environment.*;
public class AmbienteQuarto extends Environment {
Literal trd = Literal.parseLiteral("trancada(porta)");
Literal ntrd = Literal.parseLiteral("~trancada(porta)");
boolean portaTrancada = true;
@Override
public void init(String[] args) {
// percepcoes iniciais
addPercept(trd);
}
// Implementacao das acoes basicas do agente
@Override
public boolean executeAction(String ag, Structure act) {
clearPercepts();
if (act.getFunctor().equals("trancar"))
portaTrancada = true;
if (act.getFunctor().equals("destrancar"))
portaTrancada = false;
// atualiza percepcoes de estados ocorridos no ambiente
if (portaTrancada) {
addPercept(trd);
} else { addPercept(ntrd); }
return true;
}
}
about Jason
Error List
Jason IDE
26,6 All
(java,none,UTF-8) - - - - U
8/10Mb


MAS Console - quarto
[paranoico] saying: Obrigado por trancar a porta
[porteiro] doing: destrancar
[claustrofobico] saying: Obrigado por destrancar a porta!
[porteiro] doing: trancar
[porteiro] doing: destrancar
[claustrofobico] saying: Obrigado por destrancar a porta!
Clean
Stop
Debug
Pause
Sources