

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

CAIO ARCE NISHIBE
LUÍS GUILHERME BERGAMINI MENDES

**MODELAGEM DE CRENÇAS E AMBIENTE NO
FUTEBOL DE ROBÔS**

TAREFA I

CURITIBA

2011

CAIO ARCE NISHIBE
LUÍS GUILHERME BERGAMINI MENDES

MODELAGEM DE CRENÇAS E AMBIENTE NO FUTEBOL DE ROBÔS

Relatório apresentado à Disciplina de Sistemas Inteligentes 2, do Departamento Acadêmico de Informática - DAINF - do Curso Superior de Engenharia de Computação da Universidade Tecnológica Federal do Paraná - UTFPR, como requisito parcial para finalização da disciplina.

Prof. Dr. Cesar Augusto Tacla
Prof. Dr. Gustavo Giménez Lugo

CURITIBA

2011

SUMÁRIO

1	Introdução	p. 3
1.1	Objetivos	p. 3
2	Ambiente	p. 4
2.1	Descrição do Ambiente	p. 4
3	Agentes	p. 5
3.1	Movimentos Possíveis	p. 5
3.2	Crenças Iniciais	p. 6
3.3	Ações Possíveis	p. 7
3.3.1	Atacante	p. 7
3.3.2	Goleiro	p. 8
4	Tewnta e Jason	p. 9
4.1	Código Jason	p. 9
4.2	Código Java/Tewnta	p. 9
5	Conclusão	p. 11
	Referências	p. 12

1 INTRODUÇÃO

Neste documento, será apresentada uma solução de modelagem de um grupo de agentes inseridos no futebol de robôs. O cenário considerado contém três agentes atacantes e um goleiro. Deve-se elaborar uma sequência de ações para que os atacantes consigam marcar um gol. Nesse processo estão envolvidas técnicas de modelagem e escolhas (COELHO, 2008).

1.1 Objetivos

Inicialmente será feita a modelagem do ambiente no qual os agentes estão inseridos, assim como suas crenças iniciais e possíveis ações, tanto dos atacantes quanto do goleiro. A ferramenta de modelagem utilizada será o Jason (HÜBNER; BORDINI, 2010), integrado com o simulador Tewnta para a liga F180 (DETONI, 2008).

O ambiente Jason irá se comunicar com o simulador Tewnta, que fará a tradução para as ações dos jogadores na interface gráfica. O contrário também é válido, já que percepções no ambiente do Tewnta deverão ser enviadas ao Jason.

2 AMBIENTE

2.1 Descrição do Ambiente

No simulador Tewnta versão 1.3, o campo possui a dimensão de 490 x 338 pixels. Cada robô possui, aproximadamente, um diâmetro de 20 pixels. Com isso, foi criado um grid 25 x 17 em que somente um robô pode ocupar uma das posições no grid, dado um instante de tempo (Figura 1). A bola não é considerada um agente, sendo apenas um objeto do mundo, portanto poderá estar em uma mesma célula que um jogador.

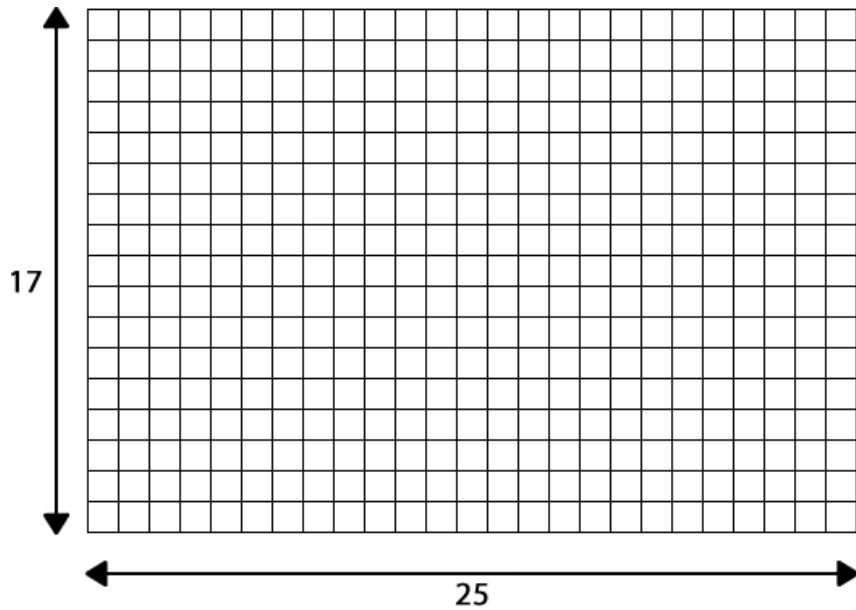


Figura 1: Grid modelado

3 AGENTES

3.1 Movimentos Possíveis

Considerando um agente que ocupa uma posição dentro do grid definido no ambiente, existem oito possíveis direções de movimentação (Figura 2).

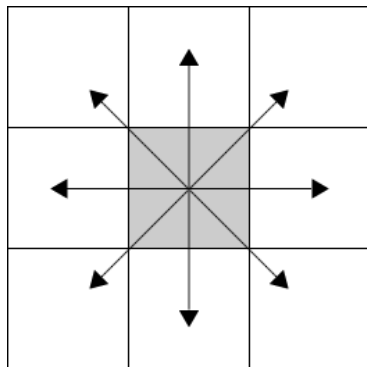


Figura 2: Movimentos possíveis para um agente que ocupa uma célula do grid

Os oito movimentos foram definidos da seguinte forma:

- **UP_CENTER**: verticalmente para cima.
- **UP_RIGHT**: diagonal para cima e para direita.
- **UP_LEFT**: diagonal para cima e para esquerda.
- **DOWN_CENTER**: verticalmente para baixo.
- **DOWN_RIGHT**: diagonalmente para baixo e para direita.
- **DOWN_LEFT**: diagonalmente para baixo e para esquerda.
- **RIGHT**: horizontalmente para direita.
- **LEFT**: horizontalmente para esquerda.

3.2 Crenças Iniciais

Quando o sistema é iniciado, o agente necessita de crenças iniciais para que possa começar seus procedimentos de inferência que levem a próximas ações. Nessa modelagem, foram definidas as seguintes crenças iniciais:

- **Posição inicial:** em qual célula do grid o agente está localizado. Para o Tewnta, representa uma coordenada (x,y) em pixels.
- **Time:** em qual time o agente está inserido, de forma a poder definir alvos de ataque e defesa.
- **Entrar em campo:** os agentes são posicionados no campo. A partir daí, no caso específico do goleiro, será iniciado o plano de defesa. No caso do atacante, será iniciado o plano de marcar um gol.

As linhas de código apresentadas no Quadro 1 mostram as implementações de crenças iniciais para o atacante e goleiro, em Jason, assim como seus objetivos iniciais.

```
// Agente: Atacante (atacante.asl)

/* Initial beliefs and rules */
posicao(5, 8).
time(team_a).

/* Initial goals */
// Objetivo inicial: entrar em campo
!entrarEmCampo.

// Agente: Goleiro (goleiro.asl)

/* Initial beliefs and rules */
posicaoInicial(24, 8).
time(team_b).

/* Initial goal */
// Objetivo inicial: entrar em campo
!entrarEmCampo.
```

Quadro 1: Definição de crenças iniciais, em Jason

3.3 Ações Possíveis

Ambos os agentes, atacante e goleiro, possuem um conjunto básico de ações possíveis. A ação de rotacionar consiste em definir um vetor de direção e apontar para ele, sem sair de sua posição atual. Com a direção definida, é possível praticar a ação de andar em linha reta no intuito de se aproximar do ponto desejado. Em resumo:

- Rotacionar;
- Andar em linha reta;

3.3.1 Atacante

No caso específico do atacante, há uma série de possíveis ações. O chute ao gol ocorre após a deliberação do agente ao perceber que está numa situação favorável para chutar ao gol. O passe é o toque da bola a um agente próximo, desde que seja do mesmo time. Há também a marcação, que consiste em acompanhar um jogador adversário no intuito de interceptar a bola. A ação de posicionar-se no campo é necessária, pois um agente pode escolher esperar a bola em um determinado local antes de executar a próxima ação. Há também ações de enviar e receber estratégia, que faz uso das funções de *broadcast* para informar os outros agentes das intenções atuais. A ação de pedir bola é feita quando o agente infere que está numa posição favorável para chute. Por fim, um agente pode buscar a bola em alguma posição do campo, caso nenhum outro agente esteja com posse. Em resumo:

- Chute ao gol;
- Passe;
- Marcação;
- Posicionar-se;
- Enviar estratégia;
- Receber estratégia;
- Pedir bola;
- Buscar a bola;

3.3.2 Goleiro

O goleiro não poderá sair de sua área em frente ao gol, com exceção de um pequeno avanço ou recuo, caso este posicionamento seja vantajoso para sua estratégia. Sua principal ação é a de defesa, que ocorre quanto a bola possui rota encaminhada ao gol. Após uma defesa, o agente poderá executar a ação de chute, enviando a bola o mais longe possível de sua região, ou de passe, que corresponde a tocar a bola para um agente próximo, que seja do mesmo time. Em resumo:

- Avanço;
- Recuo;
- Defesa;
- Chute;
- Passe;

4 TEWNTA E JASON

Para a modelagem das crenças iniciais assim como das possíveis ações de cada agente, foi adotada a linguagem Jason, baseada em *AgentSpeak*. Para colocar em prática a modelagem em questão, foi usado o simulador Tewnta.

4.1 Código Jason

Inicialmente, deve-se criar um arquivo em Jason (extensão `.mas2j`) que define o ambiente, agentes e classpaths em Java para executar (BORDINI; HÜBNER; WOOLDRIDGE, 2007). Os arquivos de classpath correspondem aos executáveis do Tewnta. O ambiente corresponde ao nome da classe em Java que implementa as ações definidas para o agente (Quadro 2).

```

MAS jasonSoccer {
  infrastructure: Centralised
  environment: SoccerEnv
  agents:

    g1 /src/agents/goleiro.asl;
    at1 /src/agents/atacante.asl;

  classpath: "lib/tewntaCommons.jar"; "lib/simulator.jar";
}

```

Quadro 2: Código de inicialização em Jason

Os arquivos `goleiro.asl` e `atacante.asl` contêm as definições de crenças e ações.

4.2 Código Java/Tewnta

Na sequência, a classe `SoccerEnv`, em Java, implementa a comunicação com o Tewnta através da conexão com o *webservice* criado pela aplicação. O ambiente modelo é em

seguida instanciado, através da classe `FieldModel`, que contém a definição do grid e métodos de conversão deste grid para medidas do campo. Ações definidas em Jason devem ser acessadas em Java através de uma sintaxe apropriada, que envolve a classe `Literal` ou diretamente por Strings. O Quadro 3 demonstra esse acesso:

```
/* Ações */
private static final Term TERM_GIRE = Literal.parseLiteral("gire");
private static final String FUNCTOR_CREATE_PLAYER = "createPlayer";
private static final String FUNCTOR_ROTACIONE_PARA_BOLA =
    "rotacioneParaBola";
private static final String FUNCTOR_IR_LINHA_RETA = "irLinhaReta";
```

Quadro 3: Conversão de termo Jason em termo Java

Os demais métodos implementados na classe `SoccerEnv` correspondem ao controle do agente dentro do campo, fazendo uso de técnicas como por exemplo PID, para estabilizar os movimentos.

5 CONCLUSÃO

Através da realização desta tarefa, foi possível testar de maneira prática a implementação de um agente que segue o modelo BDI para atingir um objetivo. Durante esta primeira fase, foram modeladas as crenças iniciais dos agentes, assim como suas possíveis ações dentro do ambiente definido. A comunicação do ambiente Jason, que permite escrever sentenças com a linguagem *AgentSpeak*, foi posta em prática com o simulador Tewnnta. Com esta base definida, pode-se partir para fases futuras que incluem a implementação dos comportamentos definidos para os agentes modelados.

REFERÊNCIAS

BORDINI, R. H.; HÜBNER, J. F.; WOOLDRIDGE, M. *Programming Multi-Agent Systems in AgentSpeak using Jason*. [S.l.]: Wiley, 2007.

COELHO, H. *Teoria da Agência: Arquitetura e Cenografia*. [S.l.]: LabMAg e ICC, FCUL, 2008.

DETONI, G. G. *tewnta: Robocup Small Size League (SSL) F180 Simulator*. 2008. Disponível em: <<http://code.google.com/p/tewnta/>>.

HÜBNER, J. F.; BORDINI, R. H. *Jason: a Java-based interpreter for an extended version of AgentSpeak*. 2010. Disponível em: <<http://jason.sourceforge.net/Jason/Jason.html>>.