CORDENONSI, A. Z. ; HAMMEL, G. *Desenvolvimento de um Agente Zagueiro para a Robocup.* In: III Simpósio de Informática do Planalto Médio, 2002, Passo Fundo. Anais do III Simpósio de Informática do Planalto Médio, 2002.

DESENVOLVIMENTO DE UM AGENTE ZAGUEIRO PARA A ROBOCUP

Andre Zaki Cordenonsi, Gilberto Hammel
andrezc@unifra.br, hammel@bol.com.br
Centro Universitário Franciscano – UNIFRA
Rua dos Andradas 1614 – CEP 97010 032 – Santa Maria, RS

Resumo

Sistemas multiagentes reativos são formados por entidades autônomas que possuem as seguintes características: (1)atuam direcionados a um objetivo, (2)estão inseridas em um ambiente e (3) reagem as mudanças ocorridas no ambiente. Os agentes reativos são agentes que trabalham no modelo ação-reação e possuem, usualmente, comportamentos pré-definidos. No entanto, estes agentes não possuem nenhum tipo de memória interna sobre suas ações ou sobre as ações dos demais agentes. De acordo com estas características, percebe-se que este tipo de agente é bastante indicado para aplicações em tempo-real, devido à sua simplicidade e rapidez na tomada de decisões. Baseado neste modelo de agentes, foi desenvolvido um agente com características reativas para ser utilizado na Robocup, uma competição entre robôs ou simuladores cujo objetivo é jogar futebol. Para o desenvolvimento e avaliação do agente, foi utilizado o servidor da Robocup (*Soccer Server*) que resolve todas as questões ligadas à implementação das regras, aspectos gráficos, gestão da comunicação e definição dos protocolos de comunicação entre os jogadores.

Abstract

Reactive multi-agent systems are composed by autonomous entities with some characteristics: (1) they act focused on a objective, (2) they are in an environment and (3) react by any changes in the environment. The reactive agents use the action-reaction model with pre-defined behavior. However, these agents have no internal memory of their actions or some else actions. Based in these assertions, the reactive agents can be used in real-time application, because of their simplicity and velocity. This paper presents a reactive agent for the Robocup Initiative, a soccer robot championship. It was used the Robocup Server (Soccer Server) which simulates the game, the rules, the graphical interface, defines the communication protocol and the communication among the agents and the agents with the environment.

1. Introdução

Os Sistemas Multiagentes(SMA) concentram-se no estudo de agentes *autônomos* em um universo multiagente. Para os SMA, o termo autônomo designa o fato de que os agentes têm uma existência própria, independente da existência de outros agentes. Usualmente, cada agente possui um conjunto de capacidades comportamentais que definem sua competência, um conjunto de objetivos e a autonomia necessária para utilizar suas capacidades comportamentais a fim de alcançar seus objetivos.

Segundo (Demazeau, 1995) pode-se definir como um agente todas as entidades ditas *ativas* de um sistema. Este conjunto de agentes forma uma *sociedade*. As entidades passivas são designadas pelo termo *ambiente*. Um agente recebe informações e raciocina sobre o ambiente e sobre outros agentes, decidindo quais ações deve realizar e

quais objetivos deve seguir. Um agente é uma entidade ativa, ou seja, capaz de controlar suas ações, diferentemente das noções estáticas tais como módulos, conjuntos de regras e bases de conhecimento.

Não existe uma definição para agente que seja aceita por toda a comunidade científica. Uma possível definição é proposta por (Ferrer, 1991): chama-se *agente* uma entidade real ou abstrata que é capaz de agir sobre ela mesma e sobre seu ambiente, que dispõe de uma representação parcial deste ambiente que, em um universo multiagente, pode comunicar-se com outros agentes, e cujo comportamento é consequência de suas observações, de seu conhecimento e das interações com outros agentes.

A definição proposta acima preocupa-se com os mecanismos internos para o tratamento do agente, não estabelecendo o tipo de comunicação possível nem a granularidade dos agentes. Em (Gasser, 1992) temos uma definição que ressalta o aspecto da *identidade* de cada agente: um agente é uma entidade à qual pode-se associar uma identidade única, e que é capaz de realizar tarefas formalizadas. Um agente pode ser considerado como um meio que produz um certo número de ações a partir dos conhecimentos e mecanismos internos que lhe são próprios.

Os Sistemas Multiagentes Reativos (SMAR) são baseados em modelos de organização biológica ou etológica. O modelo de ação de uma agente reativo é formado pelo par *estímulo-resposta* (ação-reação). As principais características dos sistemas reativos (Brooks, 1986) estão citadas a seguir:

- não há representação explícita do conhecimento;
- não há representação interna do ambiente;
- não há memória das ações;
- organização usualmente etológica;
- grande número de membros.

Resumidamente, os agentes reativos são muito simples, possuem comportamento pré-definidos e não possuem representação do ambiente. Suas ações são tomadas de acordo com a percepção do ambiente. Uma forma usual de representar o comportamento dos agentes é através de um conjunto de regras.

No entanto, apesar de sua aparente simplicidade, os SMAR podem ser utilizados na resolução de diversos problemas, principalmente quando a velocidade de processamento é crucial. Nestas circustâncias, a simplicidade do agente é um elemento favorável, pois os mesmos podem atuar de forma mais rápida. Diversos modelos tem sido apresentados para a solução de problemas utilizando SMAR, tais como o problema dos robôs mineradores (Drogoul, 1993), o modelo Paco (Demazeau, 1993), *fungus eaters* (Pfeifer, 1996) e Simula (Frozza, 1998).

2. *The Robot Word Cup Initiative*

A *RoboCup (The Robot World Cup Initiative)* é uma competição internacional entre robôs ou simuladores cujo objetivo é incentivar, comparar e divulgar métodos e trabalhos realizados em inteligência artificial, provendo um ambiente único e imutável entre as comparações. Para tanto, a *Robocup* escolheu jogos de futebol como forma de atrair a comunidade científica.

Atualmente, a *Robocup* possui três categorias principais. As duas primeiras envolvem disputas entre times de robôs propriamente ditos, sendo uma com robôs pequenos (*small size league*) e outra com robôs médios (*middle size league*). A terceira categoria envolve partidas disputadas em um simulador (*Soccer Server*), onde os jogadores são programas-clientes que conectam-se a um servidor central através do protocolo UDP/IP. Esta última categoria permite que grupos de pesquisa que estudam a área de Sistemas Multiagentes ou Inteligência Artificial desenvolvam times através da implementação de agentes lógicos capazes de cooperar para disputar uma partida de futebol. Essa arquitetura permite que os clientes sejam escritos em qualquer linguagem que suporte esse tipo de comunicação.

Nas duas primeiras categorias, os robôs podem ser implementados de maneira livre. Já na liga de simulação a comunicação entre jogadores é controlada pelo servidor. A primeira *Robocup* ("*Robot World Cup*") aconteceu em agosto de 1997, em Nagoya, Japão, durante uma conferencia de Inteligência Artificial e contou com a participação de quarenta equipes.

O simulador da *Robocup* (*Soccer Server*) é um ambiente dinâmico, executado em tempo real e que possibilita que os agentes interajam com o meio. A simulação do jogo de futebol tem regras bastante semelhantes aos jogos reais. Cada time consiste de onze agentes, ou seja, onze programas que executam funções diferentes para alcançar o objetivo final. Um cliente (agente) conecta-se com o servidor através de *sockets*. Cada um dos agentes inclusos no meio tem um estado corrente, o qual varia de acordo com a mudança do meio. Baseado neste estado, o agente escolhe a ação que deve realizar para chegar a sua satisfação. O objetivo de um agente nem sempre é chegar ao objetivo final. Como o sistema é um Sistema Multiagente, o objetivo do conjunto de agentes é alcançar o resultado final, ou seja a vitória, enquanto que cada agente específico pode ter objetivos diferenciados.

O simulador da *Robocup* aceita comandos de ação de clientes (agentes) durante um ciclo. A cada ciclo, o simulador envia novas mensagens as quais devem ser interpretadas pelos clientes para tomar as decisões devidas para o momento. Estas mensagens representam a *visão do agente* e contém dados acerca do que o agente consegue perceber do ambiente, tais como a bola, os demais jogadores, as linhas de campo, as goleiras, etc. Para cada objeto que o agente percebe no ambiente, é retornado a distância e o ângulo que o mesmo faz com a frente do agente, tal como percebe-se na figura 1. A rotação do agente em relação ao campo não é informada. O agente só consegue perceber objetos que estejam dentro de seu campo de visão, representado pelo cone pontilhado da figura 1. No entanto, a acuidade da percepção depende da distância do objeto ao agente. Se um agente estiver muito longe, as informações podem retornar incompletas. Um exemplo bastante comum é um agente perceber que outro agente está no seu campo de visão, mas ele não conseguir distinguir se o mesmo representa um jogador do seu time ou do time adversário.

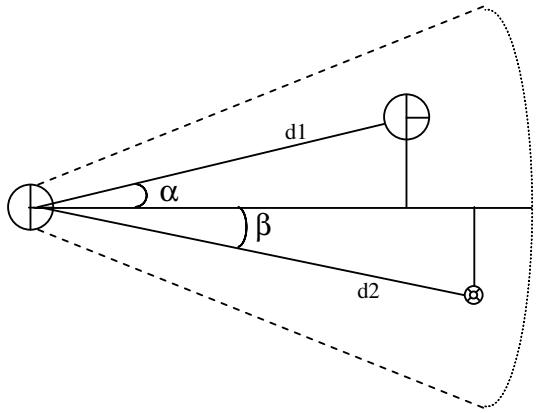

Figura 1 – Representação da visão de um agente na *Robocup*. No exemplo acima, o agente está visualizando um agente jogador, que está a uma distância *d1* e um ângulo α, e o agente bola, que está a uma distância *d2* e um ângulo β.

Dentro desse contexto, propõe-se aqui uma implementação de um agente reativo para o simulador *Soccer Server*, que atue na mesma função de um zagueiro de um time real. A escolha de uma arquitetura reativa deve-se a exigência de velocidade que o simulador necessita, pois o mesmo não espera pelos comandos do agente. Caso um agente demore para processar sua visualização e tomar uma decisão sobre que ação deve realizar, o simulador irá ignorá-lo, atendendo os demais agentes.

3. Definição da Arquitetura

Para a implementação do agente zagueiro, foram estudadas diversas arquiteturas reativas, destacando-as a arquitetura de subordinação, proposta por (Brooks, 1986) e o modelo SIEME, proposto por (Magnin, 1998). A arquitetura de subordinação foi escolhida devido a possibilidade de incrementar, gradativamente, o grau de complexidade do agente desenvolvido, o que possibilitaria a inserção de novas funções. Estas funções, de mais alto nível, poderão controlar um futuro protocolo de comunicação entre os agentes do mesmo time, o que aumentaria o grau de controle e de integração entre os diversos agentes.

A arquitetura de subordinação baseia-se na superposição de níveis, onde os níveis mais altos (de maior complexidade) estão subordinados aos níveis mais baixos, de menor complexidade mas de maior grau de reatividade. Um exemplo clássico de um agente que utiliza a arquitetura de subordinação pode ser visualizado na figura 2. Nesta figura, o agente tem a função de explorar o ambiente sem esbarrar nos objetos que encontrar pelo caminho.

Para a implementação desta arquitetura, foi utilizado um esquema baseado em regras *if-then* com prioridade. Este esquema adapta-se bem à arquitetura de subordinação, pois cada nível da arquitetura de Brooks pode ser comparado a um nível de prioridade no conjunto de regras, formando uma hierarquia facilmente ajustável.

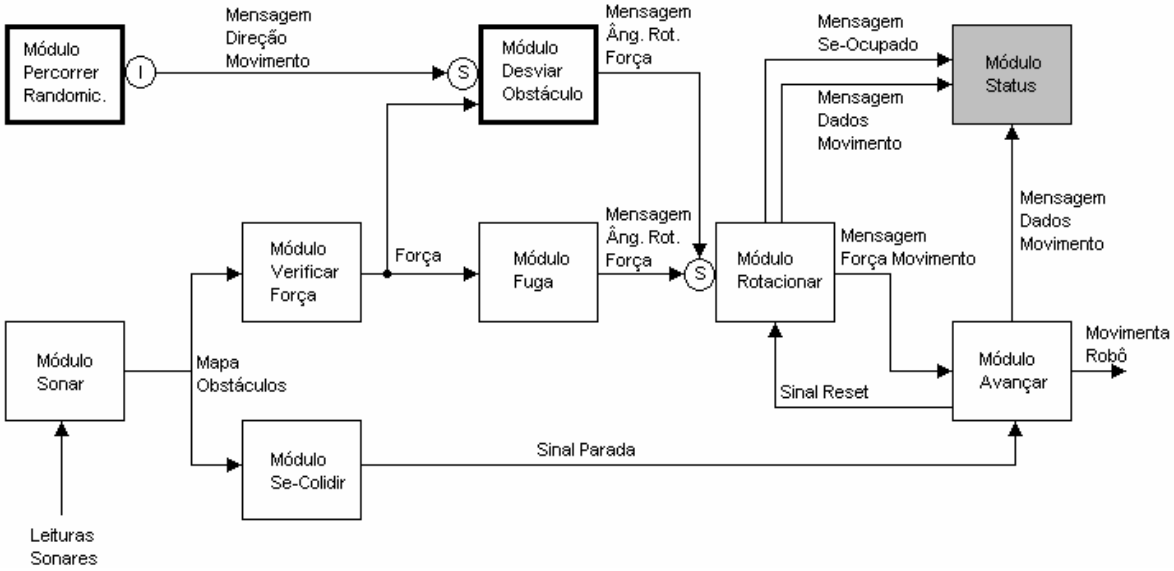

Figura 2 – Agente robô baseado na arquitetura de subordinação (adaptado de Brooks, 1986)

4. Definição do Agente Zagueiro

Para a implementação do agente zagueiro, algumas considerações foram tomadas em relação as habilidades disponíveis para cada jogador e as funções que os mesmos deveriam exercer. O simulador *Soccer Server* disponibiliza as seguintes ações aos jogadores: chutar a bola, girar em torno de seu eixo, correr em uma determinada direção, e segurar a bola. A última ação só pode ser exercida pelo agente goleiro. A ação de chutar a bola só é permitida se o jogador estiver a uma distância de 0.5 *metros* da bola (todas as distância são relativas e proporcionais ao tamanho do campo). Para esta ação, o jogador deve fornecer a força e o ângulo do chute.

A ação de girar em torno do seu eixo é realizada informando apenas o ângulo de giro. A ação de correr em uma determina direção é realizada através de um ângulo e de uma força que deverão ser fornecidos. Todos os parâmetros devem ser fornecidos de acordo com a *frente* do jogador, que representa o ângulo zero.

O agente desenvolvido possui um conjunto de regras. Cada regra possui uma pré-condição e uma ação associada. Para que uma regra seja executada, sua pré-condição deve ser verdadeira. Caso mais de uma regra tenha sua pré-condição validada, a que tiver prioridade mais alta será executada.

4.1 Posição de Referência

Um dos principais problemas ao desenvolver agentes para o simulador *Soccer Server* é que o mesmo não informa a posição absoluta dos mesmos. Para que um agente consiga perceber sua posição em relação ao campo, o simulador fornece uma série de *flags* que estão espalhados pelas linhas do campo simulado. Estes *flags* são considerados objetos do campo, logo o simulador retorna a distância e o ângulo em relação aos mesmos, como se fosse um outro agente qualquer. A posição dos principais

flags pode ser observada na figura 3. Baseado na visão dos *flags*, o agente pode inferir sua posição relativa no campo.

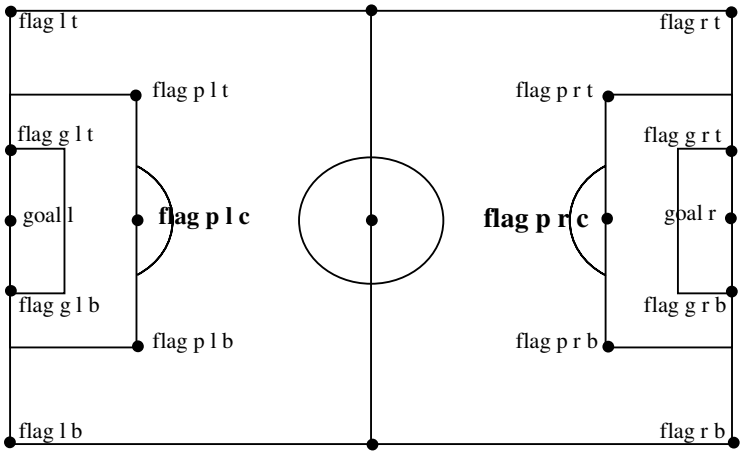

Figura 3 – Posicionamento dos *flags* do Simulador *Soccer Server.* Em destaque, os *flags* utilizados para a referência de posição do agente zagueiro

A concepção do agente zagueiro baseou-se nas características observadas em jogadores reais. Como os zagueiros tem posições defensivas, optou-se por conceber um agente que protegesse o seu próprio gol, dentro de limites pré-definidos, não tendo nenhuma participação em jogadas ofensivas. Tal característica possui duas vantagens:

- cada agente, possuindo uma área pré-estabelecida para atuar, não interfere diretamente nas ações dos demais agentes. Este fato impede a criação de aglomerados de agentes do mesmo time em volta da bola;
- simplificação nas regras de decisão, o que influencia diretamente na velocidade de definição de ação do agente.

Para o agente zagueiro, foi definido uma área circular de raio 35 *metros* em torno do (*flag p l c*) ou (*flag p r c*), que fica exatamente na frente da goleira do seu time, em cima da linha da grande área. A escolha de qual *flag* basear seu comportamento é feita antes do início da partida, quando o simulador passa esta informação para os agentes. Todas as ações do agente zagueiro se darão dentro desta área circular. Se, por qualquer motivo, o agente afastar-se desta área, ele deverá retornar à mesma, de acordo com as regras de comportamento que estão definidas na próxima seção.

O comportamento básico para o agente, em relação à sua posição, é explicado à seguir. A cada ciclo, o agente verifica a distância entre ele e o *flag* de posicionamento. Se a distância for superior que o raio de 35 *metros*, o agente deve retornar em direção ao *flag* até que a distância for menor que um metro. Ou seja, o agente não fica exatamente em cima do ponto, mas bem próximo dele. Esta perca de precisão é tolerável pois o posicionamento do jogador durante o jogo é bastante impreciso, pois o mesmo só é obtido através de sucessivos comandos de corrida. Por outro lado, se a distância for a menor que a definida como limite, o agente pode andar dentro da área, se estiver vendo a bola.

4.2 Regras de Comportamento

As regras de comportamento do agente zagueiro são descritas na tabela 1, em ordem decrescente do grau de prioridade.

Tabela 1 – Regras de Comportamento para o Agente Zagueiro

1	SE a distância do agente até a bola for menor que 0,5 *metros*	ENTÃO o agente deve chutá-la em direção ao gol adversário
2	SE a distância do agente até o *flag* de posicionamento for superior a trinta e cinco *metros*	ENTÃO o agente deve virar-se para o *flag* e correr em direção ao mesmo
3	SE o agente percebe a bola e a mesma estiver dentro de sua área	ENTÃO o agente deve correr até a bola
4	SE o agente percebe a bola, mas esta não está dentro de sua área	ENTÃO o agente deve girar em direção a bola
5	SE o agente não percebe a bola, percebe o *flag* de posicionamenteo e a distância deste até o agente for maior que 0,1 *metro*	ENTÃO o agente deve correr até o *flag* de posicionamento
6	SE o agente não percebe a bola, mas a distância do *flag* de posicionamento for menor que 0,1 *metro*	ENTÃO o agente deve girar em torno do seu eixo
7	SE o agente não percebe a bola nem o *flag* de posicionamento	ENTÃO o agente deve girar em torno do seu eixo

O agente, quando está em condições de chutar a bola, sempre deverá chutá-la em direção ao gol adversário. Neste sentido, outro(s) agente(s) do mesmo time deverão ser implementados para que atuem na área de meio-campo, para que recebam a bola chutada pelo zagueiro. De acordo com a segunda regra, percebe-se que o agente só poderá ir de encontro a bola se a mesma estiver dentro de sua área de atuação. Caso contrário, outros agentes deverão interceptar a mesma.

Quando o agente percebe a bola em sua área de percepção, ele irá diretamente de encontro a ela. Ao diminuir a distância para menos de meio metro, ele poderá chutar a bola em direção ao gol adversário. No entanto, se o agente percebe a bola, mas esta não está dentro de sua área de atuação, o mesmo deverá girar em torno do seu eixo em direção da bola. Desta maneira, mesmo não indo de encontro a bola, ele se posicionará de frente para mesma. Esta regra diminui o tempo de resposta do agente quando este tiver que correr até a bola, pois evita-se que o mesmo deva girar primeiro e depois correr (dois ciclos de simulação). Logo, pela regra quatro, o agente deverá sempre ficar de frente para a bola enquanto a tiver no seu campo de visão.

Em relação as duas últimas regras, se o agente não perceber a bola, ele deve retornar à sua posição original, perto do *flag* de posicionamento. Caso ele já esteja perto do *flag,* ele deverá girar em torno do seu eixo para tentar encontrar a bola. Finalmente, caso o agente não esteja percebendo a bola nem o *flag*, ele deverá girar em torno do seu eixo até encontrá-los.

5. Resultados e Discussões

O agente zagueiro foi testado individualmente, jogando com outros agentes que atuavam como atacantes do time adversário. Foram realizados testes com um, dois e três atacantes adversários. Nos testes realizados, o zagueiro sempre atuou sozinho, ou seja, o time era composto somente por ele.

Inicialmente, o agente era posicionado sob o *flag p l c*, antes do início da partida. O simulador *Soccer Server* permite que todos os jogadores sejam posicionados antes do início de cada tempo. Quando do início da partida, percebe-se que o agente rapidamente observa a bola, pois está de frente para a mesma, e acompanha seus movimentos, girando em torno do seu eixo. Quando a bola entra dentro da área de atuação do agente, ele dirige-se imediatamente para a mesma. Algumas obeservações foram realizadas para esta situação:

- em algumas situações, devido ao grau de instabilidade inserido pelo simulador, o agente acaba por sair de sua área de atuação. Tal fato ocorre pela forma como o agente se desloca, através de um vetor força que é fornecido ao simulador. No entanto, outras variáveis influenciam no deslocamento do jogador, tais como o vento e o grau de condicionamento *físico* do jogador, que desgasta-se com o decorrer do tempo. Logo, não existe uma relação direta entre a força fornecida ao jogador e a distância percorrida pelo mesmo;
- considerando a forma de deslocamento descrita acima, o jogador, às vezes, passa da bola que vem de encontro ao mesmo. Neste caso, ele acaba por descrever uma curva de 180° até encontrar a bola novamente, como visualizado na figura 4;
- a direção para onde o jogador chuta a bola é de extrema importância para o bom desenvolvimento do jogo. Em algumas situações, a regra 01 se mostrou inadequada, pois o zagueiro acabava por chutar a bola de encontro ao adversário. Além disso, em certas ocasiões, o jogador está de costas para o gol adversário. Para resolver este problema, a regra foi modificada para que, caso a diferença entre o ângulo do gol adversário e algum agente adversário for menor que cinco graus, o jogador deve escolher uma outra direção onde não existam agentes adversários. Esta outra direção não pode coincidir com o próprio gol. Caso o agente não observe o gol adversário, ele deve chutar a bola para uma das laterais.

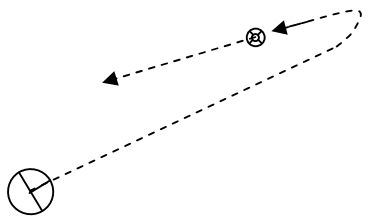

Figura 4 – A volta realizada pelo jogador, quando este passa da linha da bola

6. Conclusões

A utilização de modelos multiagentes apresentam bons resultados em diversas áreas de aplicação. As arquiteturas desenvolvidas para a resolução de um problema podem ser facilmente convertidas e testadas em problemas semelhantes, aumentando a reutilização dos códigos e incrementando a robustez do sistema, por testá-los nos mais diversos tipos de ambientes e situações diversas.

O agente zagueiro apresentou resultados satisfatórios com os jogos realizados. Usualmente, a grande maioria dos ataques foram rechaçados pelo agente. Somente nos testes realizados com três atacantes houve um maior número de ataques bem sucedidos do que defesas. Tal fato pode ser explicado pela insistência dos atacantes, pois quando os mesmos se concentravam na bola, o zagueiro não tinha muitas opções de ângulo disponíveis para tirar a bola da situação de gol.

A modificação da regra 1 trouxe um grande ganho no desempenho do agente, pois ele conseguir chutar a bola para longe de seu gol de forma mais rápida e eficiente. Isto é interessante pois a performance do agente depende fortemente de seu posicionamento. Normalmente, o agente conseguia retornar a sua posição inicial e observar a formação dos atacantes com uma certa antecedência. No entanto, tal característica deve ser melhor observada quando da utilização de times completos, pois desta forma a bola tende a retornar ao ataque mais rapidamente.

O principal objetivo deste trabalho era desenvolver um agente zagueiro que pudesse ser, futuramente, utilizado em um time completo para a *Robocup*. Como as áreas de atuação dos diversos jogadores tendem a não se sobrepor em demasia, para evitar um acúmulo de jogadores em torno da bola, as regras definidas para os agentes sozinhos podem ser aplicadas facilmente em um sistema multiagente.

Como trabalhos futuros, pode-se citar:

- um estudo mais aprofundado para a solução do problema do deslocamento do jogador, para previnir que o mesmo ultrapasse a bola;
- a utilização de dois e três zagueiros e decorrente observação de sua atuação, para a configuração de esquemas diferentes de jogos;
- e a inclusão da observação de jogadores do próprio time para implementar a troca de passes entre os jogadores.

Bibliografia

BROOKS, R. A. A Robust Layered Control for a Mobile Robot. *IEEE Journal of Robotics and Automation*, New York, v. 2, n. 1, Mar. 1986.

DEMAZEAU, Y. From interactions to collective behaviour agent-based system. In: *European Conference On Cognitive Science*, 1., 1995, St. Malo. Proceedings... France: [s.n], 1995.

DROGOUL, A. *De la Simulation Multi-Agents à la Résolution Collective de Problèmes.* Paris, France: Université Paris VI, 1993. Thèse de Doctorat.

FERBER, J.; GASSER, L. Intelligence artificielle distribuée. In: *International Workshop On Expert Systems & Their Applications*, 10., 1991, Avignon. Cours n. 9. France: [s.n], 1991.

FROZZA, R. *SIMULA – Ambiente para Desenvolvimento de Sistemas Multiagentes Reativos.* Porto Alegre: CPGCC da UFRGS, 1998. Dissertação de mestrado.

GASSER, L. Boundaries, identity and aggregation: plurality issues in multiagent systems. In: WERNER, E.; DEMAZEAU, Y. (Eds.). *Decentralized AI 3.* Amsterdam: North-Holland, 1992.

MAGNIN, L. SIEME: an Interactions Based Simulation Model. In: *XII European Simulation Multiconference (ESM 98)*, Manchester, United Kingdon, 1998.

PFEIFER, R. Building "Fungus Eaters": Design Principles of Autonomous Agentes, In: *International Conference On Simulation Of Adaptive Behavior*, 4., 1986. Proceedings... Cambridge: MIT, 1996. p. 3-12.