PROPOSTA DE UM SISTEMA MULTIAGENTE REATIVO PARA A *ROBOCUP*

*Alisson Moscato Loy, Bruno Cechella, Leone Cesca, **Simone Regina Ceolin ***Andre Zanki Cordenonsi, Guilherme Dhein

RESUMO

Atualmente, o conceito de agentes domina a construção de sistemas de inteligência artificial, tanto para estudos de inteligência como para a solução de problemas complexos. Agente é um conceito que define entidades inteligentes que apresentam, entre outras, as seguintes características: estar inserido em um ambiente e ter capacidade para perceber e atuar sobre este meio; possuir autonomia; se adaptar a novas situações; ser dotado de reatividade, ou seja, ser capaz de perceber alterações no ambiente e atuar em função destas alterações; exibir comportamento direcionado a solução dos seus objetivos; e exibir habilidade social. Portanto, o estudo de sistemas multiagentes contribuiu para a pesquisa na solução computacional de problemas complexos, que exigem a decomposição como forma de resolução. Este trabalho modelou uma arquitetura de agentes que atuem cooperativamente para a resolução de um problema bem definido. Para a validação da arquitetura, foi escolhido o simulador de futebol de robôs *SoccerServer*. Este simulador é um ambiente dinâmico, que exige respostas em tempo-real, aproximando-se muito das necessidades verificadas em aplicações práticas de sistemas multiagentes.

INTRODUÇÃO

Segundo (RUSSELL, 95), a Inteligência Artificial pode ser definida como o campo de estudos da Ciência da Computação que busca a compreensão de entidades inteligentes e a reprodução em sistemas artificiais (de *software* ou *hardware*) dos mecanismos responsáveis pelo comportamento inteligente. Sistemas de Inteligência Artificial podem ser muito úteis na solução computacional de problemas complexos. Trabalhando em conjunto com seres humanos, ou até substituindo os

---

* Orientados
** Bolsista FAPERGS
*** Orientadores

humanos em algumas atividades, sistemas inteligentes têm apresentado resultados expressivos em certas aplicações práticas.

O presente trabalho tem por objetivo montar um time de futebol de robôs que joguem uma partida de futebol de forma totalmente automatizada, sem qualquer intervenção humana em tempo real. O campeonato de Futebol de Robôs é realizado no Simulador *SoccerServer*, onde o principal objetivo é construir agentes inteligentes que possam realizar suas tarefas de forma autônoma e eficiente. Para a realização da percepção, construção de estratégias e tomada de decisão de cada agente, é necessário um estudo aprofundado de cada tarefa.

O simulador *SoccerServer* proporciona um ambiente em tempo-real que possibilita analisar o desempenho de diferentes Sistemas Multiagentes. Cada agente pode efetuar diferentes ações, tais como: *kick*, que é o comando onde o agente chuta a bola, passando a força e o ângulo como informações para o servidor; o *see*, que é uma *string* passada do servidor para o agente pelo *socket*, que informa ângulos e distâncias da bola e dos demais agentes, em relação aos *flags* e outros agentes.

O time consiste de onze agentes, cada um com sua visão (limitada por um ângulo de 90 graus), movimentação e uma certa quota de energia. Cada jogador possui sua função. Como um time na vida real, existe o goleiro, o atacante, o defensor, cada um programado para sua função.

O Futebol de Robôs é uma atividade científica reconhecida pelo IEEE-*Institute of Electrical and Electronics Engineers*, que tem apoiado a realização das "Copas Mundiais" na forma de congressos científicos. Assim, os participantes exibem o desempenho de suas equipes e trocam experiências**.**

ROBOCUP

The Robot World Cup Initiative (*RoboCup*) é uma motivação na área de inteligência artificial e de sistemas multiagentes (SMA), pois estimula a resolução de um problema (jogar futebol) onde uma ampla variedade de tecnologias pode ser integrada, examinada e comparada.

O projeto *The Robot World Cup Initiative (RoboCup)* tem como objetivos o incentivo e a popularização dos Sistemas Multiagentes como uma metodologia de resolução computacional de problemas. Por apresentar um problema padrão (times de futebol compostos unicamente por agentes autônomos) e por não restringir formas de solução, esta iniciativa permite que uma vasta gama de tecnologias sejam integradas, examinadas e comparadas. Existe um órgão internacional (*The RoboCup Federation*),

formado por um grupo de pesquisadores que implementa e distribuem as atualizações do simulador, que organiza as competições e normatiza as regras para as disputas. (KITANO, 95)

A *RoboCup* possui diversas categorias de disputa: Robôs de Pequeno Porte, Robôs de Médio Porte, Robôs de Quatro Pernas e a Liga de Simulação. Para esta última, utiliza-se um sistema que simula um campo de futebol (*SoccerServer*), onde os times são compostos unicamente por agentes implementados via *software*, não envolvendo a construção física de um robô. O *SoccerServer* (Figura 1) é um ambiente de simulação dinâmico e em tempo-real, onde os agentes são clientes do servidor e comunicam-se com este através de *sockets* UDP.

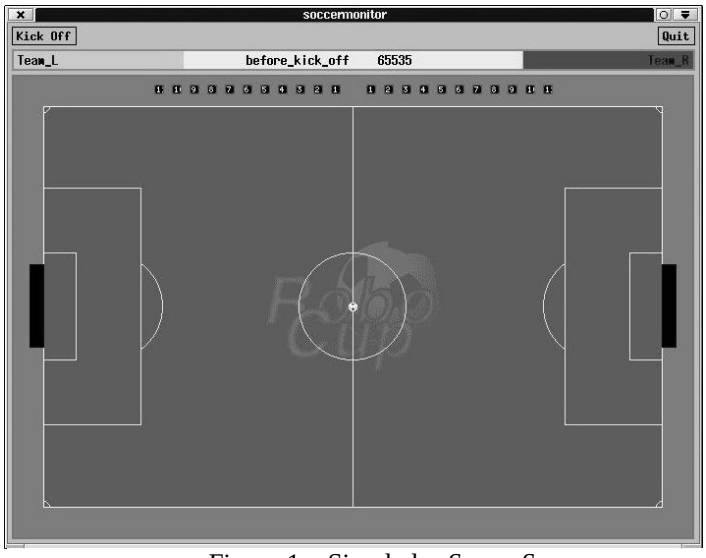

Figura 1 – Simulador *SoccerServer*

AGENTES

Define entidades inteligentes que apresentam, entre outras, as seguintes características: estar inserido em um ambiente e ter capacidade para perceber e atuar sobre este meio; possuir autonomia; se adaptar a novas situações; ser capaz de perceber alterações no ambiente e atuar em função destas alterações; exibir comportamento direcionado a solução dos seus

objetivos; e exibir habilidade social. Além destas, as implementações de agentes em Inteligência Artificial podem incorporar outras características, como aprendizagem.

Os agentes inteligentes podem ser classificados em várias categorias, de acordo com suas características. A seguinte classificação é apresentada por (WOOLDRIDGE, 1994):

- **Conselheiro**: oferece ajuda e treinamento. Ensina os passos iniciais para usar um determinado sistema. Fornece suporte contínuo, observando todas as ações do usuário, as quais ele pode interceptar e pedir confirmação. Pode ser consultado para mostrar como executar uma atividade particular, ou então, sugerir métodos alternativos e mais rápidos para executá-la.
- **Guia**: ajuda a navegação em bancos de dados e hipermídia. Classifica, recupera e filtra grandes quantidades de informações, apresentando somente os dados relevantes e importantes aos usuários, no formato personalizado. Fornece caminhos apropriados para o usuário navegar pelo banco de dados, e auxilia-o caso se sinta "perdido".
- **Empregado**: executa as atividades tediosas ou repetitivas. Atividades são executadas imediatamente, e algum tipo de *feedback* pode ser fornecido tanto pelo usuário como pelo próprio agente.
- **Representante**: trabalha na ausência do usuário. De certa forma, seria parecido ao agente Empregado citado anteriormente, exceto pelo fato de que as atividades não precisam ser imediatamente executadas ou então, são executadas somente após eventos específicos. Por exemplo, pode fazer *backups* de arquivos de madrugada ou fazer pedidos de compras, caso algum produto atinja o limite mínimo no estoque.
- **Comunicador**: trabalha com outros usuários e seus agentes, para assim, conseguir executar as atividades às quais foi designado. Pode, por exemplo, organizar reuniões de recursos e pessoas. Ou então, pode reunir um grupo de agentes para que assim possam executar uma atividade mais complexa.

SISTEMA MULTIAGENTES

Os agentes devem raciocinar a respeito das ações e sobre o processo de coordenação em si. As suas arquiteturas são mais flexíveis e a organização do sistema está sujeita a mudanças, visando adaptar-se às variações no ambiente e/ou no problema a ser resolvido.

Para Pattie Maes[1], as arquiteturas são especificações de como podemos construir agentes a partir de módulos e como estes módulos interagem. O conjunto total de módulos e suas interações devem especificar como, a partir dos dados dos sensores e do estado interno atual do agente, são obtidos ações e o novo estado interno. Assim, cabe ao construtor da arquitetura definir a forma de dividir o problema e distribuir capacidades aos módulos do agente, além da melhor forma de construir estes módulos e de como será a sua interação.

Sistemas baseados nestas arquiteturas trabalham, geralmente, com diversos agentes que atuam concomitantemente em busca de uma solução para o problema proposto. A implementação de uma coletividade de agentes acarreta num aumento da complexidade do sistema como um todo, pois torna-se necessário a elaboração de diversos mecanismos, tais como: coordenação, controle, organização e planejamento conjunto. Um conjunto de agentes trabalhando em busca de um mesmo objetivo comum é denominado Sistema Multiagente. A metáfora de inteligência utilizada pelos Sistemas Multiagentes é a *comunidade inteligente*, ou seja, o comportamento social que é base para a inteligência do sistema.

A construção de um Sistema Multiagente envolve a definição da arquitetura do agente como indivíduo e da arquitetura do sistema multiagente (ambiente e comportamento social), bem como a sua implementação. Diversos sistemas multiagentes têm sido propostos para a resolução de problemas: mineração de dados, controle de sistemas industriais, de tráfego aéreo, de redes de distribuição de energia elétrica, entre outros.

IMPLEMENTAÇÃO DE UM AGENTE JOGADOR

Uma forma de se definir agentes é através de ações que o agente deve tomar quando verificar um determinado estado no ambiente em que estiver inserido. Assim, num primeiro momento foram estudadas formas de se trabalhar a percepção que o agente tem do seu ambiente. O agente recebe

---

[1] Citada em (WOOLDRIDGE, 1995).

informações do *SoccerServer* através de uma *string*. As informações contidas nesta *string* simulam a percepção visual do agente, ou seja, contém dados a respeito do estado daquela porção do ambiente que se encontra no campo de visão do agente. Foram trabalhadas formas de tratar a *string* para extrair dela as informações necessárias para a tomada de decisão do agente, como informações sobre a localização do jogador, dos jogadores adversários e da mesma equipe, localização da bola, localização do gol, entre outros.

Por outro lado, o agente deve ser capaz de atuar sobre o ambiente, transformando-o de acordo com seus objetivos. Foram implementadas diversas ações para que os agentes atuem no simulador *SoccerServer*. Estas ações podem ser separadas em dois grupos: o primeiro é composto por comportamentos individuais como correr ao encontro da bola, chutá-la, detectar o gol e correr em sua direção, chutar em gol, entre outros; o segundo é composto por capacidades de atuação em equipe, como a capacidade de realizar passes de bola, não disputar a bola com um companheiro de equipe, detectar se há companheiro mais próximo da bola, entre outros.

Definidas percepções e ações básicas, foi trabalhada a idéia de integrar os aspectos de percepção com os de atuação através de uma árvore binária que defina os comportamentos dos agentes em determinadas situações do ambiente. Os nós da árvore que não são folhas representam testes sobre as condições do ambiente, os arcos representam o resultado positivo ou negativo do teste, e as folhas são ações que devem ser executadas se há um caminho entre a raiz e aquela folha. Isto significa que são feitos diversos testes sobre as informações visuais, de forma que apenas a ação correspondente àquela situação do ambiente é executada.

A idéia da árvore de decisão levou ao desenvolvimento de um protótipo de uma ferramenta visual para a geração de comportamentos de agentes. Esta ferramenta implementa a árvore de decisão descrita anteriormente. Tanto os testes quanto as ações são componentes visuais, o que permite a rápida geração de diferentes combinações de comportamentos. Por ser uma ferramenta visual, esta tarefa de geração é bastante simplificada, o que permite que sejam testada uma grande variedade de combinações de comportamentos.

Por fim, foram implementados agentes dotados dos comportamentos testados na ferramenta usando a Linguagem de Programação C++. Assim, temos atualmente uma equipe de agentes implementados, dotados de comportamentos básicos, capazes de compor um time e jogarem uma partida no *SoccerServer*.

Basicamente, o trabalho desenvolveu as seguintes etapas:

- Primeiramente, foi realizado o *download* do *Soccerserver* Manual Versão 5 para estudos e auxílio do desenvolvimento do time com melhor precisão. Também foi feito do*wnload* e estudo do Simulador *Soccerserver*, que é o ambiente onde possibilita em tempo real, visualizar e analisar um sistema multiaagente;
- Foi realizado o estudo do funcionamento dos *Sockets UDP*, sendo um sistema simples de comunicação que permite aos programas trocarem *strings* durante o jogo. Estas *strings* contém o que o jogador enxerga (bola, outros jogadores e *flags*), suas distâncias e ângulos;
- Na seqüência, foi realizado o estudo de diversas Arquiteturas de Agentes. Baseados nestes estudos, pretende-se propor uma arquitetura para um Sistema Multiagente que atue no Simulador Soccerserver;
- Foi criada uma árvore binária para controlar os comportamentos dos agentes, onde os nós da árvore são os testes e as folhas são as ações.
- Foi implementado um agente jogador básico, usando a Linguagem de Programação C++ e utilizando o Sistema Operacional Linux;
- Foi criado de uma função que permite aos jogadores interceptar a trajetória da bola, podendo assim impedir um passe entre adversários e receber com maior performance a bola de um companheiro.

DISCUSSÃO DAS DIFICULDADES ENCONTRADAS

Durante o desenvolvimento do agente jogador, algumas dificuldades foram encontradas e estão citadas à seguir:

1. O jogador passa da bola quando vai de encontro com a mesma. Esta situação ocorre quando o jogador vai ao encontro da bola, mas devido à aceleração do jogador e da bola, ele acaba passando do ponto de encontro da mesma. Esta situação pode ser observada na Figura 2. Usualmente, a trajetória do jogador acaba descrevendo uma parábola, enquanto que a trajetória otimizada seria uma reta até a bola.

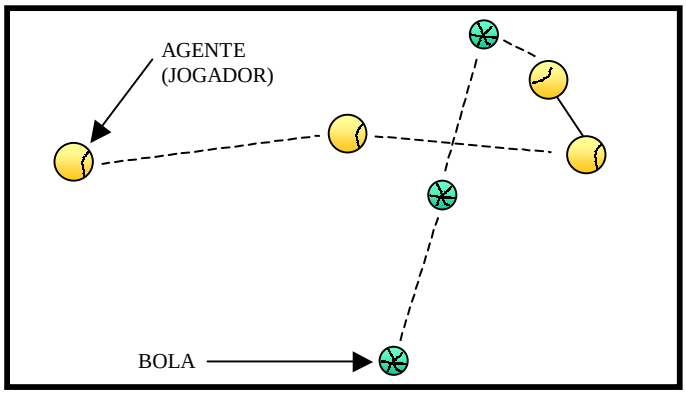

Figura 2 – Jogador passa da bola

2. O jogador, no momento de interceptar a bola, percorre um caminho desnecessário para alcançá-la (figura 3). O jogador perde tempo, pois a bola não se encontra mais na posição a qual ele visualizou no primeiro momento.

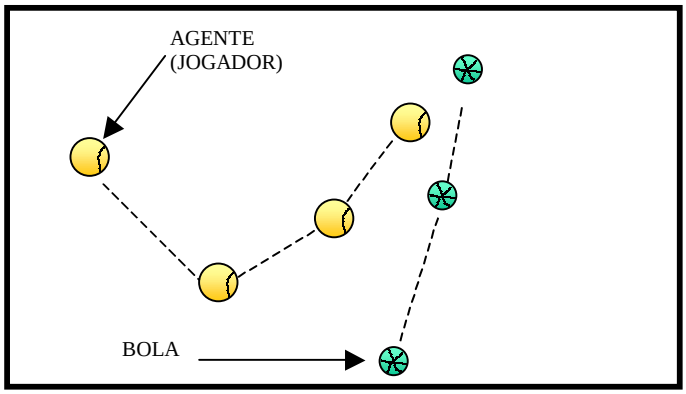

Figura 3 – Jogador não intercepta a bola

3. Não há regiões delimitadas para o funcionamento dos agentes, pois a visão dele é baseada em *flags*. Logo, ocorre a superpopulação de agentes no local onde se encontra a bola.

Para resolver os problemas um e dois, buscaremos uma abordagem aproximativa, pois o nível de incerteza das variáveis não permite prever exatamente a posição da bola num dado momento (variáveis como vento, velocidade, aceleração, variação no ângulo, variação no agente jogador, etc).

Em relação ao terceiro problema, nossa proposta baseia-se na construção de um sistema de posicionamento global (Figura 4), para que os jogadores possam se posicionar em regiões do campo, como jogadores normais.

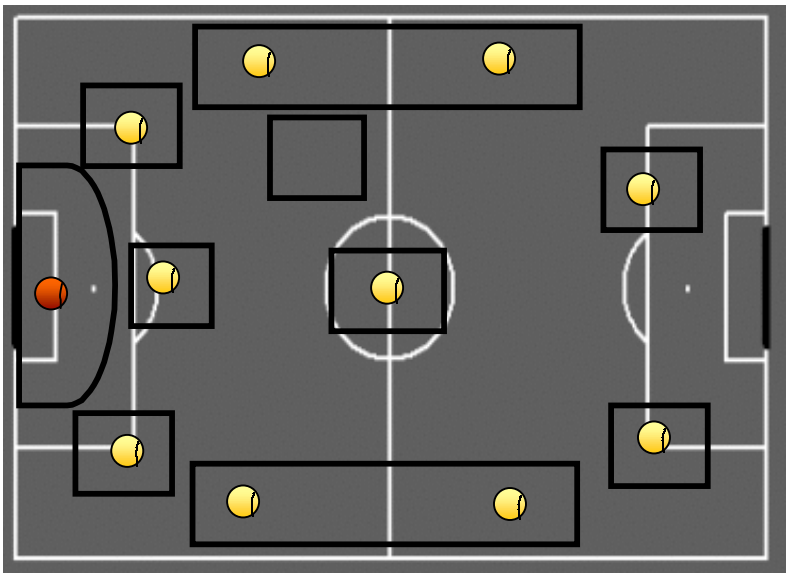

Figura 4 – Posicionamento dos Jogadores

CONCLUSÕES E TRABALHOS FUTUROS

No atual estágio do projeto, os agentes ainda não possuem configurações específicas. Isto significa que os agentes que formam uma equipe são todos iguais, não apresentando estruturas internas diferentes de acordo com a sua função dentro do time. A incorporação de características específicas passa pela definição de novas formas de percepção e por novas formas de atuação.

Uma vez definidas estas formas, sua incorporação aos agentes será bastante simples, graças à ferramenta visual desenvolvida. Basta que as novas capacidades sejam associadas a elementos visuais da ferramenta, para que possam ser utilizados na definição de novas árvores binárias. Testes com estas árvores serão realizados, indicando e validando a melhor configuração.

Vale salientar que já existe um conjunto de agentes capazes de atuar no simulador como uma equipe, capazes de competir com outros times. No entanto, a performance ainda é pobre no que corresponde ao aspecto tático do jogo.

AGRADECIMENTOS

Agradecemos a FAPERGS pelo auxílio, sem o qual o presente trabalho não poderia ter sido realizado.

REFERÊNCIAS BIBLIOGRÁFICAS

BRATMAN, Michael E. What is Intention? In: COHEN, P. R.; MORGAN, J.; POLLAC, M. E. (Eds.). **Intentions in Communication**. p. 15-32. MIT Press, 1990.

BROOKS, Rodney A. A Robust Layered Control System for a Mobile Robot. **IEEE Journal of Robotics and Automation**, 2(1), p. 14 - 23. 1986.

CHAIB-DRAA B.; MOULIN, B.; MANDIAU, R.; MILLOT, P. Trends in Distributed Artificial Intelligence. **Artificial Itelligence Rewiew**, v. 6, p. 35- 66, 1992.

COCKBURN, David; JENNINGS, Nick R. ARCHON: A Distributed Artificial Intelligence System for Industrial Applications. In : O'HARE, G. M. P.; JENNINGS, N. R. (Eds.) **Foundations of Distributed Artificial Intelligence**. p. 319-344. Wiley Inter-Science, 1995.

CORDENONSI, A.Z., ALVARES, L.O. The Simulation of the Behaviour Evolution for a Rective Multi-agent System. In: **Argentine Symposium in Artificial Intelligence**, 1999, Buenos Aires. Proceedings... Argentina, 1999.

CORDENONSI, A.Z. **Um Ambiente de Evolução de Comportamentos para Sistemas Multiagentes Reativos**, Porto Alegre, 2000. Dissertação de Mestrado. Instituto de Informática, Universidade Federal do Rio Grande do Sul, 2000.

DHEIN, G. **Integrando Deliberação e Reatividade em uma Arquitetura de Agentes Híbrida Homogênea**, Porto Alegre, 2000. Dissertação de Mestrado. Instituto de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, 1998.

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

FIKES, Richard E.; NILSSON, Nils J. STRIPS: a new approach to the application of theorem proving to problem solving. **Artificial Intelligence**, v. 2, p. 189-208, 1971.

HADDADI, Afsaneh. Comunications and Cooperation in Agent Systems: A Pragmatic Theory**. Lecture Notes in Artificial Intelligence**, Heidelberg, v. 1056, 1996.

KITANO, H. RoboCup: The Robot World Cup Initiative. In: **International Joint Conference on Artificial Intelligence** (IJCAI'95), 14., 1995, Montreal. Proceedings... Canada, 1995.

MINSKY, Marvin. **The Society of Mind.** New York: A Touchstone book, 1988.

MOULIN, Bernard; CHAIB-DRAA, Brahim. An Overview of Distributed Artificial Intelligence. In : O'HARE, G. M. P.; JENNINGS, N. R. (Eds.) **Foundations of Distributed Artificial Intelligence**. p. 3-55. Wiley Inter-Science, 1995.

RUSSELL, Stuart, NORVIG, Peter. **Artificial Inteligence: A Modern Approach**. Prentice Hall, USA, 1995.

SICHMAN, Jaime S., DEMAZEAU, Yves, CONTE, Rosaria, CASTELFRANCHI, Cristiano. A Social Reasoning Mechanism Based on Dependence Networks. In : COHN, A. **Proceedings of the 11th**

European Conference on Artificial Intelligence. Holanda : John Wiley & Sons, Ltda, 1994. p. 188-192.

WOOLDRIDGE, Michael, JENNINGS, Nicholas R. Intelligent Agents: Theory and Practice. **Knowledge Engineering Review**. 1994.

WOOLDRIDGE, Michael, JENNINGS, Nicholas R. Intelligent Agents: Theory and Practice. **Knowledge Engineering Review**, v. 10(2), p. 115-152. 1995.