

Modelagem Comportamental de Sistemas Multiagentes

Marcelo Santos Linder¹, Augusto Loureiro da Costa²

¹Colegiado de Engenharia da Computação – Universidade Federal do Vale do São Francisco (UNIVASF)
CEP 48909-810 – Juazeiro – BA – Brasil

²Programa de Pós-Graduação em Mecatrônica, Universidade Federal da Bahia (UFBA)
CEP: 40210-630 – Salvador – BA – Brasil

marcelo.linder@univasf.edu.br, augusto.loureiro@ufba.br

Abstract. *Multiagents systems with hundreds or even a dozen of agents have a enormous number of interactions, highly dynamics and complex, making it difficult the behavior analysis of these systems by humans. This article describes a method for obtaining a behavior model of multiagents systems. The model is obtained from the analysis, in real-time, of a simulated soccer match of robots. Heuristic rules are applied to obtain the Petri Nets, which represent the behavior model of the analyzed team.*

Resumo. *Sistemas multiagentes com centenas ou até mesmo uma dezena de agentes, têm um grande número de interações, altamente dinâmicas e complexas, tornando difícil a análise comportamental desses sistemas por humanos. Este artigo descreve um método para obtenção do modelo comportamental de sistemas multiagentes. O modelo é obtido através da análise, em tempo real, de uma partida de futebol simulado de robôs, onde são aplicadas regras heurísticas para a geração de Redes de Petri, que representam o modelo comportamental da equipe analisada.*

1. Introdução

A Inteligência Artificial Distribuída (IAD) é uma das áreas da Inteligência Artificial (IA) que mais se desenvolveu nos últimos anos e apresenta um enorme potencial de aplicação [Cheny et al, 2002]. Segundo [Weiss, 1999], esta área envolve o estudo, construção e aplicação de sistemas multiagentes, onde agentes inteligentes (entidade lógica ou física), interagem para realizar um conjunto de objetivos ou tarefas, como em missões espaciais com sistemas multirobôs autônomos [Dorais et al, 1999], em simulações de confrontos militares e tráfego de pedestres [Silva, 2003], etc.

Uma forma comum de incentivo à pesquisa em IA, mais especificamente em IAD é a proposta de problemas padrão. Um problema bastante interessante, que vem atraindo interesse de diversos grupos de pesquisa em robótica e IAD é a realização de partidas de futebol entre robôs autônomos. Esta proposta surgiu da iniciativa de um grupo de pesquisadores da área, que em 1997 resultou na Primeira Copa Mundial de Futebol de Robôs a RoboCup [RoboCup_a, 2007] em Nagoya, Japão.

Reconhecendo a potencialidade deste problema, a Universidade Federal da Bahia (UFBA), implementou dois projetos de pesquisa para se dedicar a este tema, o

AxeBot [Axebot, 2007] e o MecaTeam [Mecateam, 2007], que se utilizam do domínio do futebol de robôs como laboratório para o ensino e pesquisa da mecatrônica e automação industrial. Estes projetos promovem a integração de várias tecnologias e áreas de pesquisa distintas, como: agentes inteligentes, sistemas multiagentes, estratégias de aquisição de conhecimento, sistemas de tempo real, sistemas distribuídos, reconhecimento de padrões, integração de sensores, controle de processos, sistemas embarcados, entre outras.

Pelas características do problema e dos desafios envolvidos numa partida de futebol, o futebol de robôs tornou-se um importante domínio principalmente no que diz respeito à pesquisa na área de sistemas multiagentes. No entanto, a maioria dos trabalhos que envolvem times de agentes autônomos, dedicam-se a obter sinergia na realização das tarefas ou dos objetivos.

No âmbito de uma competição entre equipes de agentes, modelar o comportamento global do oponente se torna uma vantagem significativa, pois desta forma pode-se antecipar as ações do adversário. Logo, surge a necessidade da existência de ferramentas que auxiliem ou até mesmo façam a análise do comportamento desses sistemas. Segundo [Nair; Tambe, 2002], sistemas multiagentes com centenas ou até mesmo uma dezena de agentes, têm um grande número de interações, altamente dinâmicas e complexas, tornando difícil a análise comportamental desses sistemas por humanos.

Buscando solucionar o problema exposto, este artigo descreve um método para obtenção de um modelo comportamental de uma equipe de agentes.

Este artigo está estruturado da seguinte forma: a Seção 2 apresenta uma visão geral do ambiente em que o método proposto é aplicado; na Seção 3 serão discutidos trabalhos correlatos a este tema; já a Seção 4, abordará a metodologia aplicada para solucionar o problema da modelagem comportamental de sistemas multiagentes autônomos; finalizando, a Seção 5 apresenta conclusões e discorre sobre perspectivas para trabalhos futuros.

2. Ambiente de Simulação

Para validação do método proposto, foi utilizado como domínio o futebol de robôs. Sendo alvo da modelagem a equipe oponente. Visando não atrelar o desenvolvimento do trabalho à aquisição de simuladores proprietários ou até mesmo robôs reais, foi utilizado para a análise, a Liga de Futebol Simulado da RoboCup, em razão desta disponibilizar, gratuitamente, todos os recursos necessários para os testes e validações.

Em virtude da utilização da RoboCup como domínio para a aplicação, torna-se interessante uma pequena familiarização com a mesma. Atualmente, a RoboCup é composta de diferentes provas e modalidades em que são implementados e testados diversos problemas relacionados a sistemas multiagentes. Nem todas as provas que integram este campeonato faziam parte do seu formato original, que era baseado principalmente em um problema padrão: jogar futebol. As principais atividades englobadas pela RoboCup no atual formato são descritas a seguir:

- Competição de Futebol de Robôs (RoboCup Soccer) - é considerada a atividade mais importante do campeonato. Inclui cinco competições: simulação, pequeno

porte (18 cm), médio porte (50 cm), robôs de quatro patas e humanóides. Na Liga de Futebol Simulado, que será utilizada como domínio para validação deste trabalho, dois times de robôs virtuais, com onze jogadores cada, onde cada robô é controlado independentemente por um agente autônomo que possui uma percepção parcial do ambiente, jogam futebol num campo virtual simulado por computador, como pode ser observado na figura 1. Os jogos são divididos em dois tempos com duração de 5 minutos cada [RoboCup_a, 2006].

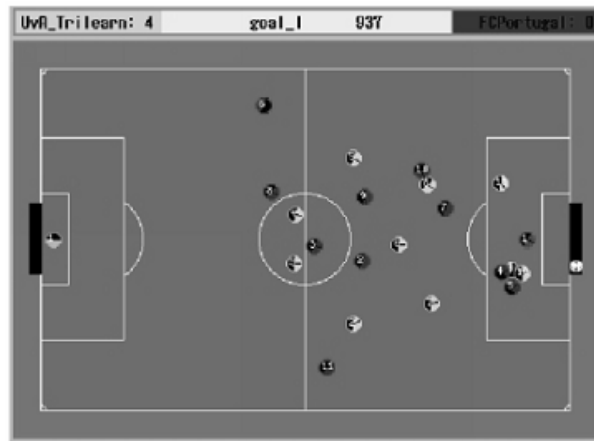


Figura 1 – Ambiente Simulado - Time UvA Trilearn X Time FCPortugal

- Salvamento (RoboCup Rescue) - nesta modalidade as competições desenvolvem-se em torno de robôs capazes de efetuar salvamentos em situações de desastre. Inclui uma competição de simulação e uma competição com robôs reais [RoboCup_b, 2006].
- RoboCup Junior - é fundamentalmente uma iniciativa educativa e de sensibilização, que combina cooperação com competição. Crianças e jovens utilizando uma infra-estrutura simples, criam equipes de robôs para jogar futebol (2 contra 2), dançar ou resgatar vítimas [RoboCup_c, 2006].

3. Trabalhos Correlatos

Como foi dito anteriormente, a maioria dos trabalhos com sistemas multiagentes dedicam-se a obter sinergia na realização das tarefas ou dos objetivos. Porém, observando a necessidade de desenvolver técnicas automatizadas para analisar equipes de agentes, um grupo da University of Southern California, desenvolveu um assistente automatizado que realiza análises de equipes de agentes. Para validar estas análises, foi utilizado como domínio, o campeonato simulado da RoboCup. O assistente desenvolvido é denominado ISAAC [Raines; Tambe; Marsella, 2000], suas análises são feitas sob jogos realizados. ISAAC requer pouco conhecimento do domínio, ele extrai dados da própria partida, a qual está analisando, e utiliza ferramentas de aprendizagem indutiva para fazer suas análises.

Considerando o problema de modelar globalmente o comportamento da equipe oponente no decorrer do confronto, ISAAC apresenta duas deficiências em suas

análises, estas, não ocorrerem simultaneamente com o decorrer do jogo e não geram um modelo comportamental global, mas sim classificam conjuntos de eventos executados por um agente, em bem e mal sucedidos, extraíndo fatores que possam ter influenciado nestes desfechos.

Com a continuação das pesquisas, o ISAAC foi aperfeiçoado [Nair; Tambe, 2002], passando a efetuar análises em três níveis distintos, destinados a analisar:

1. Comportamentos individuais de agentes;
2. Interações de um pequeno grupo do sistema multiagente, considerado como fundamental para a obtenção de sucesso ou fracasso da equipe;
3. Tendências de comportamentos globais de equipes.

O primeiro nível de análise já foi discutido. Quanto ao segundo, este ocorre da seguinte forma: a análise é feita através de um jogo de padrões, cada um com uma probabilidade de ocorrência, os padrões são caracterizados por um autômato finito obtido com base nas interações entre os agentes. Este processo pode ser mais facilmente entendido quando aplicado no domínio da RoboCup.

Fazendo esta análise no domínio do futebol, sabe-se que, por exemplo, um gol acontece como resultado de uma sequência de chutes dados por um ou mais jogadores. Não é apenas o último chute que é responsável pelo gol, mas, sim o conjunto de ações de diferentes agentes que interagiram antes do gol. Logo, gera-se um autômato finito que representa este conjunto de interações. Contudo, ao se analisar uma sequência muito ampla de interações, verifica-se que, as interações iniciais têm um impacto muito pouco significativo sobre o desfecho final.

Por fim a análise das tendências de comportamentos globais de equipes, assim como na análise dos comportamentos individuais de agentes, utiliza uma aprendizagem por árvore de decisão, porém ao invés de se analisar pontos (casos) da história como sucesso intermediário ou fracasso, são analisadas características globais que conduzem ao resultado final sobre um plano completo. Estes resultados podem ser classificados como sucesso, fracasso, empate, etc. Neste caso, se faz necessário um especialista no domínio da aplicação para escolher estas características e prover as classes dos resultados finais. O problema é que esta análise foge do escopo do trabalho apresentado aqui, uma vez que, além de ser feita off-line (após o termino da partida) ainda não gera uma análise comportamental de uma equipe em específico, e sim de porque times têm sucesso ou falham em um determinado domínio.

Um trabalho nesta área, porém cujo foco não se sobrepõe ao do trabalho apresentado aqui, é o desenvolvido em [Stone; Veloso, 1998], no qual é incorporada uma árvore de decisão previamente treinada com um comportamento completo de uma equipe de agentes para controlar os agentes ao longo de um confronto inteiro, também se utilizou a RoboCup como domínio. Neste trabalho foram criadas situações artificiais para treinar a árvore de decisão, e posteriormente os comportamentos instruídos da árvore eram utilizados em situações de jogo.

Em [Sukthankar; Sycara, 2005] é proposta uma metodologia para reconhecer, representar e registrar comportamentos executados por uma equipe de jogadores humanos em um Torneio Fictício de Operações Militares em Terreno Urbano (OMTU).

Para monitorar diretamente o desempenho de jogadores humanos, desenvolveu-se uma versão de Torneio Fictício que registra a posição e orientação no tempo de todos os membros da equipe, como eles participam em um contexto simulado de uma equipe de ataque que se move por uma área urbana em OMTU. O reconhecimento de comportamento é executado off-line usando um jogo de modelos de Markov escolhidos em sucessões de movimentos curtos que são traduzidos em uma armação de referência canônica; o modelo de comportamento com a probabilidade mais alta para uma determinada sucessão é identificado como correto. O trabalho pressupõe que um reconhecimento off-line preciso de comportamentos de time é uma condição prévia importante para construir ambientes de treinamento virtuais para tarefas de trabalho de equipe.

Baseado na premissa de que em domínios multiagentes com adversários e agentes cooperativos, os agentes do time deveriam ser adaptáveis ao ambiente e oponente atual. [Riley; Veloso, 2001] introduz um método on-line que proporciona planos para o time de agentes, os quais são gerados por um agente específico denominado “treinador”, obtidos com base em oponentes específicos. O treinador é equipado com vários modelos de oponentes pré-definidos, sendo este capaz de selecionar rapidamente, no decorrer da partida, um dentre os diferentes modelos usando um algoritmo Bayes, gerando um plano, para os agentes do time, adaptado ao adversário atual. O treinador usa uma Rede Temporal Simples para representar planos de time, como movimentos múltiplos coordenados entre os agentes, e procura um plano dependente-do-oponente para seus companheiros de time. Este plano é comunicado então aos agentes que o executam de forma distribuída. O sistema é implementado completamente em um domínio de futebol de robôs simulado.

4. Aquisição do Modelo Comportamental

Como foi exposto, neste trabalho, a equipe de agentes alvo da modelagem comportamental é o time oponente em uma partida da Liga de Futebol Simulado da RoboCup. Uma vez que esta, além das vantagens expostas, dá suporte à existência de um agente especial, em cada uma das equipes, denominado treinador, o qual possui uma visão global da partida.

O modelo comportamental do time adversário é construído pelo treinador com base na análise dos dados extraídos da partida pelo mesmo, através da aplicação de regras heurísticas capazes de com base nas informações que descrevem frames da partida, identificar interações entre os agentes, presentes no sistema, e ações desses sobre o ambiente. O modelo é gerado no decorrer da partida analisada, ou seja, no início do jogo o modelo do oponente é nulo. Com o decorrer da partida, modelos parciais são gerados, esses são construídos através da caracterização das jogadas do oponente. Definindo-se como jogada o conjunto de interações como passes de bola e ações como conduções de bola que ocorrem desde a cobrança de lateral, tiro de meta, roubada de bola, etc. até o chute a gol, o sofrimento de uma falta ou a perda da posse de bola pelo oponente.

São utilizadas Redes de Petri (RdP) para representar as jogadas identificadas do oponente, devido ao potencial de representatividade das mesmas, que têm sido amplamente utilizadas para modelagem de sistemas concorrentes, temporizados e/ou

estocásticos, como demonstram: [Girault, 2002], [Desel, 2000], [Maciel, 1996], [Murata, 1989], [Peterson, 1981] e [Peterson, 1977].

O método de aquisição do modelo comportamental, ou seja, o processo de determinação das jogadas do oponente, pode ser caracterizado pelos seguintes passos:

1. A cada ciclo de atualização do ambiente, é identificado, através da aplicação de regras heurísticas, se foi dado início a uma jogada do oponente (se houve uma roubada de bola, cobrança de lateral, tiro de meta, escanteio, falta ou uma saída de bola), caso tenha ocorrido, uma nova RdP é gerada para representar a jogada em questão. Um lugar inicial é inserido na RdP e o estado do jogo no momento em que foi dado início à jogada é registrado para em uma análise posterior vir a constituir a ficha responsável pela transição que terá este estado como origem, em seguida passa-se ao passo 2; Caso contrário continua-se no passo 1;
2. É analisado, a cada ciclo de atualização do ambiente, através da aplicação de regras heurísticas, se houve uma interação entre agentes do time, em específico, um passe ou uma ação de um agente do time sobre o ambiente, em específico, uma condução de bola. Caso tenha existido, passa-se ao passo 3; Caso contrário, verifica-se a ocorrência de finalização da jogada (efetuação de um chute, sofrimento de uma falta ou ocorrência de um desarme), em caso positivo, segue-se para o passo 4; Caso contrário continua-se no passo 2;
3. A interação ou a ação é registrada na RdP (que representa a jogada analisada no momento) como uma transição que tem como origem o último lugar inserido na rede e destino um novo lugar inserido. Para obtenção da ficha, que viabiliza a transição, se faz a análise do estado do jogo armazenado no momento da inserção do lugar de origem da transição em questão, visando a extração das características deste estado que são condições necessárias para transição. O estado do jogo logo após a interação em questão é armazenado para em uma análise posterior vir a constituir a ficha responsável pela transição que terá o último estado inserido como origem; e retorna-se para o passo 2;
4. Identifica-se, através da aplicação de regras heurísticas, a forma como foi finalizada a jogada. Se esta foi finalizada através de um chute a gol este é registrado na RdP (que representa a jogada analisada no momento) como uma transição que tem como origem o último lugar inserido na rede e destino um novo lugar inserido. Para obtenção da ficha, que viabiliza a transição, se faz a análise do estado do jogo armazenado no momento da inserção do lugar de origem da transição em questão, visando a extração das características deste estado que são condições necessárias para transição. A RdP é finalizada e volta-se ao passo 1; Caso a finalização da jogada tenha sido fruto de uma perda da posse de bola ou sofrimento de uma falta, é inserida uma transição na RdP (que representa a jogada analisada no momento) que tem como origem o último lugar inserido na rede e destino um novo lugar inserido. Para obtenção da ficha, que viabiliza a transição, se faz a análise do estado do jogo armazenado no momento da inserção do lugar de origem da transição em questão, visando a extração das características deste estado que são condições responsáveis pela finalização incompleta da jogada. A rede é finalizada e volta-se ao passo 1.

As RdP's geradas constituem o modelo comportamental do oponente, uma vez que estas representam o conjunto de jogadas efetuadas pelo oponente no decorrer da partida. As características analisadas em um determinado estado do jogo para constituição das fichas de transição são: posicionamento dos jogadores do time envolvidos na interação ou posicionamento do jogador do time que efetuou a ação sobre o ambiente e posicionamento dos oponentes que podem inviabilizar a transição.

4.1. Estudo de Caso

Para uma melhor compreensão deste processo analisaremos a obtenção, através da aplicação dos passos descritos anteriormente, da RdP da figura 2, a qual descreve uma jogada iniciada a partir de uma cobrança de falta e concluída com a obtenção do gol pela equipe modelada.

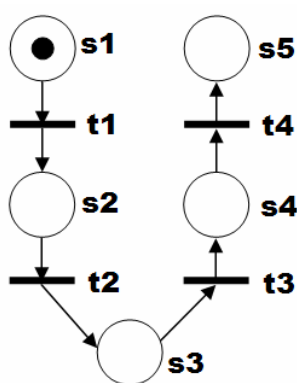


Figura 2 – Rede de Petri representando uma jogada

Inicialmente o passo 1 é executado até ser observado o início da jogada (cobrança de falta) gerando a criação de uma nova RdP com estado inicial s1 e o estado do jogo no momento em que foi dado início à jogada (figura 3) é registrado. Passa-se então ao passo 2, onde se observa uma interação entre agentes (passe de bola), conduzindo ao passo 3, onde, esta interação é registrada como a transição t1 entre o lugar s1 e s2, inserido na rede como local de destino da transição, a ficha para transição t1 é obtida através da análise do estado registrado no momento da inserção de s1, extraíndo-se as características expostas pelo estado representado pela figura 7 (a ficha para transição) e o estado do jogo após a interação (figura 4) é armazenado. Retorna-se ao passo 2, onde, se observa uma ação (condução), levando a execução do passo 3, onde, esta ação é registrada como a transição t2 entre o lugar s2 e s3, inserido na rede como local de destino da transição, a ficha para transição t2 é obtida através da análise do estado registrado no momento da inserção de s2, extraíndo-se as características expostas pelo estado representado pela figura 8 (a ficha para transição) e o estado do jogo após a ação (figura 5) é armazenado. Retorna-se ao passo 2, onde, se observa uma nova interação (passe), conduzindo ao passo 3 onde esta interação é registrada como a transição t3 entre o lugar s3 e s4, inserido na rede como local de destino da transição, a ficha para transição t3 é obtida através da análise do estado registrado no momento da inserção de s3, extraíndo-se as características expostas pelo estado representado pela figura 9 (a ficha para transição) e o estado do jogo após a interação (figura 6) é armazenado. Retorna-se ao passo 2, onde, se observa a finalização da jogada, o que leva à execução do passo 4, esta é registrada na rede como a transição t4 entre o lugar s4 e

s5, inserido na rede como local de destino da transição, a ficha para transição t4 é obtida através da análise do estado registrado no momento da inserção de s4, extraindo-se as características expostas pelo estado representado pela figura 10 (a ficha para transição) e a RdP que representa a jogada em questão é finalizada.

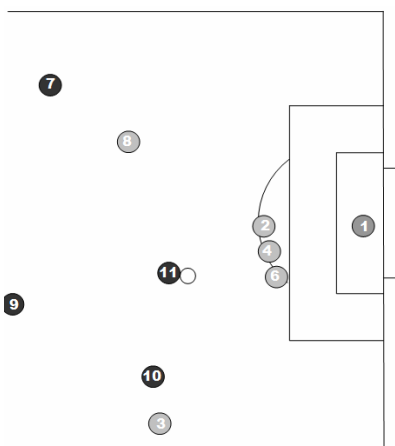


Figura 3 – Instante da cobrança da falta

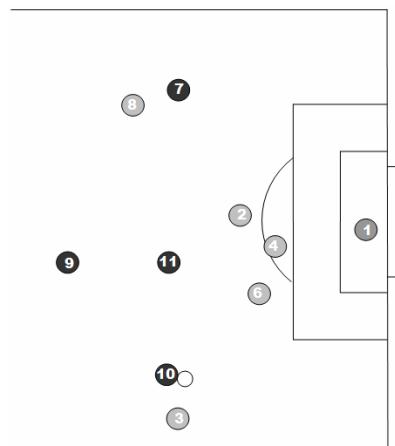


Figura 4 – Instante da recepção do passe de bola

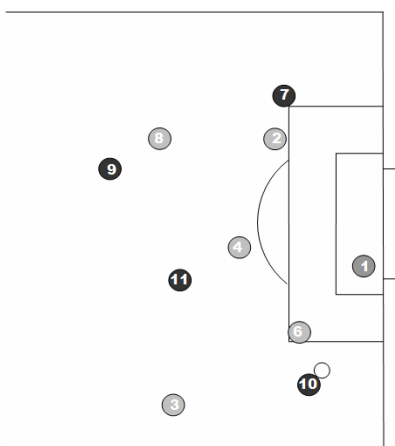


Figura 5 – Instante da efetuação do passe de bola

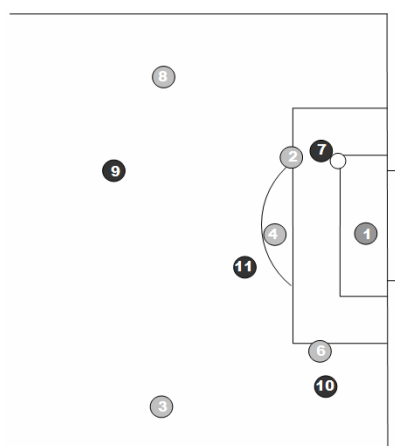


Figura 6 – Instante da recepção do passe de bola e chute a gol

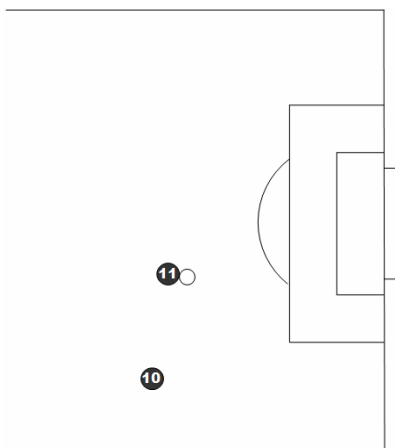


Figura 7 – Ficha da transição t1

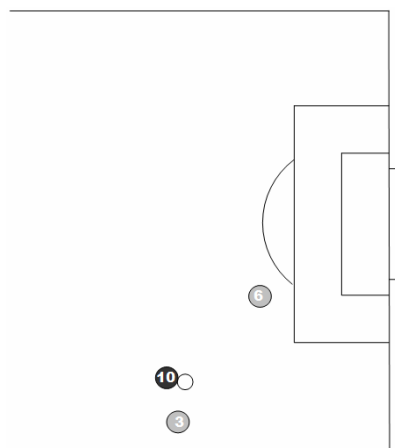


Figura 8 – Ficha da transição t2

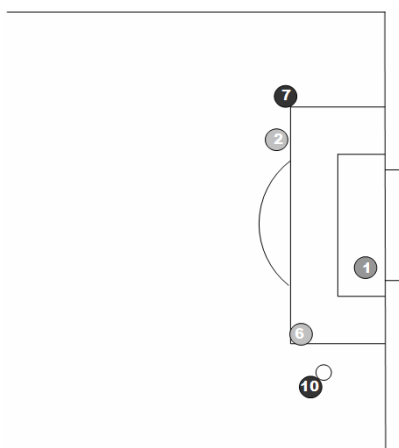


Figura 9 – Ficha da transição t3

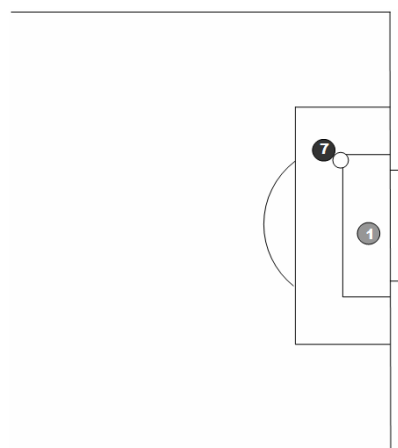


Figura 10 – Ficha da transição t4

5. Conclusões e Perspectivas para Trabalhos Futuros

Através da comparação entre o modelo comportamental do time, obtido com o emprego do método proposto, e o comportamento apresentado pelo mesmo no decorrer da partida, através da análise dos *logfiles*. Observa-se que o modelo reflete o comportamento do time em questão, no que se refere às jogadas efetuadas pelo mesmo. Contudo ao final da partida obtém-se um número muito grande de Redes de Petri e observa-se que algumas são muito similares.

Com base nessa observação, uma etapa futura deste trabalho é a implementação de um módulo, cujo objetivo é fazer uma análise das redes geradas, separando-as por grupos de equivalências e buscando obter uma RdP que representa cada grupo identificado.

Referências

- AxeBot (2007) “Projeto AxeBot: Agentes Autônomos”, Disponível em: <<http://www.axebot.ufba.br>>. Acessado em janeiro.
- Caruana, R.; Freitag, D. (1994) “Greedy Attribute Selection”, In 11th Proceedings of the 11th International Conference on Machine Learning (ICML).
- Chen, M. et al. (2002) “RoboCup Soccer Server”, for Soccer Server Version 7.07 and later.
- Desel, J. (2000), “Place/Transitions Nets I”, Introductory Tutorial Petri Nets, Petri Nets 2000, 21st International Conference on Application And Theory of Petri Nets, Denmark, pp 111-160.
- Dorais, G. et al. (1999) “Adjustable Autonomy for Human-Centered Autonomous Systems”, Working notes of the Sixteenth International Joint Conference on Artificial Intelligence Workshop on Adjustable Autonomy Systems.
- Girault, C., Valk, R. (2002) “Petri Nets for Systems Engineering: A Guide for Modeling, Verification and Application”, Springer Verlag.
- Maciel, P. R., et al. (1996), “Introdução às Redes de Petri e Aplicações”, X Escola de Computação, Campinas SP.

- Mecateam (2007) “Projeto Mecateam: Futebol entre Robôs Autônomos”, Disponível em: <<http://twiki.im.ufba.br/bin/view/Mecateam>>. Acessado em janeiro.
- Murata, T. (1989) "Petri Nets: Properties, Analysis and Applications", Proceedings of the IEEE, Vol. 77, pp. 541-580.
- Nair, R.; Tambe, M. (2002) “Automated Assistants for Analyzing Team Behaviors”, Kluwer Academic Publishers.
- Peterson, J. L. (1977), “Petri Nets”, ACM Computing Surveys, vol. 9, No. 3, pp. 223-252
- Peterson, J. L. (1981), “Petri Net Theory and the Modeling of Systems”, Englewood Cliffs, Prentice Hall
- Raines, T.; Tambe, M.; Marsella, M. (2000), “Agent Assistants for Team Analysis”, AI Magazine, Vol 21, Num 3, pp 27-31, Fall.
- Riley, P.; Veloso, M. (2001) “Coaching a Simulated Soccer Team by Opponent Model Recognition”, In Proceedings of the Fifth International Conference on Autonomous Agents.
- RoboCup_a (2007) “The RoboCup Federation”, Disponível em: <<http://www.robocup.org>>. Acessado em janeiro.
- RoboCup _b (2007) “RoboCupRescue: The Grand Challenge”, Disponível em: <<http://www.robocup.org/games/36.html>>. Acessado em janeiro.
- RoboCup _c (2007) “RoboCupJunior”, Disponível em: <<http://www.robocup.org/junior/index.html>>. Acessado em janeiro.
- Russell, S.; Norvig, P. (1995) “Artificial Intelligence: A Modern Approach”, Prentice-Hall.
- Silva, C. (2003) “Modelagem Comportamental para Agentes Autônomos em Ambientes Reais”, 82 p. – Tese – M.Sc. em Engenharia da Computação – Área de Concentração Geomática – Universidade do Estado do Rio de Janeiro – UERJ, Rio de Janeiro – RJ.
- Stone, P.; Veloso, M. (1998) “Using Decision Tree Confidence Factors for Multiagent Control”, In Proceedings of International Conference on Autonomous Agents.
- Sukthankar, G.; Sycara, K. (2005) “Automatic Recognition of Human Team Behaviors” Robotics Institute, Carnegie Mellon University, in Modeling Others from Observations (MOO 2005), Workshop at the International Joint Conference on Artificial Intelligence (IJCAI), Edinburgh, Scotland, July 2005.
- Weiss, G. (1999) “Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence”.