

## **SISTEMAS MULTIAGENTES BASEADOS EM MODELAGEM POR REDES DE PETRI: UM ESTUDO DE CASO**

*André Mendes da Rosa<sup>1</sup>, Fabio Aiub Sperotto<sup>2</sup>, Diana Francisca Adamatti<sup>1,2</sup>*

<sup>1</sup>*Programa de Pós-graduação em Engenharia de Computação, Universidade Federal do Rio Grande (FURG), Rio Grande, RS – Brasil*

<sup>2</sup>*Programa de Pós-graduação em Modelagem Computacional, Universidade Federal do Rio Grande (FURG), Rio Grande, RS – Brasil*

*E-mail: [seteeng, fabio.aiub, dianaada}@gmail.com](mailto:{seteeng, fabio.aiub, dianaada}@gmail.com)*

---

### **ÁREA**

Conhecimento como aliado às novas Tecnologias para otimização de processos.

---

---

### **RESUMO**

No artigo desenvolve-se uma rede de Petri como ferramenta de apoio na modelagem e simulação de Sistemas Multiagentes (SMA). O estudo utiliza esta ferramenta para modelar as estruturas do SMA e visualizar com mais clareza as primitivas e instruções que deverão ser usadas para implementação do sistema com objetivo de conceber graficamente uma visão sobre regras e cooperação dos agentes envolvidos na resolução de um problema. São revistos alguns casos, onde redes de Petri foram utilizadas para modelagem, controle e simulação, depois é abordada a metodologia usada e parte-se para o estudo propriamente dito. O estudo de caso utilizado é o combate a incêndio com dois bombeiros que tem que agir de forma coordenada para executar suas ações. Na codificação do SMA foi utilizada a plataforma Jason com a linguagem orientada a agentes AgentSpeak(L).

**Palavras-chave:** Redes da Petri, SMA, Jason, Modelagem, AgentSpeak(L)

---

## 1 INTRODUÇÃO

No desenvolvimento de modelos computacionais, ferramentas têm sido utilizadas para auxiliar aos programadores e demais especialistas na construção de sistemas. Desde a fase de concepção de um sistema até sua fase de implementação, existe uma preocupação da organização dos desenvolvedores na utilização de ferramentas que possam auxiliar ou cooperar umas com as outras na modelagem, análise e construção.

Segundo APPIO e HÜBNER<sup>1</sup>, em Sistemas Multiagentes, são construídos sistemas, onde entidades computacionais cooperam entre si para resolver um determinado problema. Estes sistemas podem possuir atividades dos agentes, aspectos de controle, fluxo de informações ou mensagens.

Conforme ZHOU e DICESARE<sup>2</sup>, rede de Petri é uma ferramenta que mesmo possuindo um formalismo matemático não possui uma rigidez física nos termos utilizados em sua estrutura. O desenvolvedor, para qualquer domínio, pode fornecer a sua própria interpretação sobre as atividades e fluxo de informações no domínio, em uma modelagem gráfica.

Este trabalho apresenta a utilização e cooperação da rede de Petri na modelagem de coordenação em um Sistema Multiagente, verificando o fluxo de informações, obstáculos e atividades na modelagem antes da implementação do sistema em si.

## 2 REVISÃO DE LITERATURA

Devido à natureza dos sistemas multiagentes, estes podem ser representados satisfatoriamente por redes de Petri, ou seja, um SMA congrega uma sociedade de agentes que buscam a cooperação para atender as necessidades de um ambiente.

A sincronização de atividades, sendo elas sequenciais ou paralelas estabelece o processo de cooperação entre os agentes constituintes da sociedade. Já internamente ao agente, as relações entre os níveis de decisão (de cooperação, paralelas, ou competição entre recursos) também encontram nas Redes de Petri uma boa ferramenta para a sua especificação.

Em sistemas multiagentes podem ser encontrados vários trabalhos que utilizam redes de Petri. Como o estudado em LINDER<sup>3</sup>, que trata da modelagem comportamental de agentes na competição de futebol de robôs, utilizando a rede de Petri na modelagem das jogadas. As redes também podem ser utilizadas em protocolos de interação entre agentes como mostrado em CABAC<sup>4</sup> e BAI<sup>5</sup>.

CELAYA<sup>6</sup> mostra outra aplicação encontrada na modelagem de SMA, utilizando redes de Petri, no sentido de representar as propriedades fundamentais do sistema, antes do desenvolvimento das simulações. No estudo, os autores descrevem as características do SMA que se assemelham a sistemas modelados pelas redes de Petri. Através disto, formalizam o modelo e “como” representar metodicamente o SMA em Petri. O modelo torna-se útil como método de análise preliminar estrutural do SMA.

Outro estudo mostrado por FERNANDES<sup>7</sup>, referente a um sistema industrial que utiliza agentes para o controle e gerenciamento da produção de metais. É considerado o uso de redes Petri Coloridas para modelar o comportamento dos agentes internos e externos da produção, devido à necessidade de criar um sistema mais robusto, resistente a retenções ou conflitos. Incluindo nesta necessidade, o uso da rede de Petri, como modelagem visual dos eventos.

Em COST<sup>8</sup>, são desenvolvidos estudos na modelagem de conversas entre agentes utilizando redes de Petri Coloridas. Dois modelos são formalizados: registro de conversas e negociação entre agentes. Cada transição da rede é responsável em disparar para um ou mais estados/lugares uma

determinada condição. Essas condições, dependendo do que receberem como entrada pode ser um erro de comunicação onde o protocolo tem como responsabilidade retornar uma mensagem ao agente de origem relatando o problema. Os estados também podem representar recursos, como arquivamento de identificadores para que no disparo de transições de respostas, o recurso identificador dos agentes origem e destinatário possa ser reutilizado.

PADILHA<sup>9</sup> investiga a modelagem de agentes em sistemas educacionais e usa redes de Petri devido a duas de suas principais propriedades, a vivacidade, onde qualquer estado do sistema pode ser acessível devido ao sistema ser livre de deadlocks (impasses) e reiniciabilidade, sendo que o sistema pode retornar ao seu estado inicial mesmo após a execução de uma ou mais tarefas. As redes de Petri são usadas, principalmente, para representação de mecanismos paralelos.

O estudo de SILVA<sup>10</sup> mostra a possibilidade de uma sistematização na análise de rotinas organizacionais, com a possibilidade da detecção de desvios nessas rotinas podendo auxiliar a antecipação da ocorrência de ações, fatos ou eventos inesperados no futuro, onde as redes de Petri são utilizadas para desenvolver um método para especificação de rotinas. O trabalho estuda as rotinas de um ambiente familiar, visando analisar o consumo de energia.

Vários trabalhos relacionam redes de Petri com sistemas multiagentes. Entretanto nem todos formalizam e integram os componentes característicos de ambas as áreas. Alguns estudos utilizam a modelagem conceitual da estrutura do SMA, enquanto outros aprofundam essas estruturas, também conceitualmente, adicionando a representação dos aspectos de interação e eventos entre agentes em modelo de Petri. O foco deste trabalho é na utilização das redes de Petri como ferramenta auxiliar no desenvolvimento de sistemas multiagentes, focando na parte de modelagem do SMA.

### **3 METODOLOGIA**

#### **3.1 Sistemas Multiagentes**

WOOLDRIDGE<sup>11</sup> e BORDINI<sup>12</sup> definem que agente é uma entidade de software que possui ação autônoma e que está inserido em um determinado ambiente, da qual percebe características deste meio, cooperando com outros agentes para atingir seu objetivo. Estes agentes possuem implementações de noções mentais ou de percepções, normalmente humanas, tais como crenças e intenções segundo WOOLDRIDGE<sup>13</sup>.

Sobre os objetivos, os agentes podem cooperar ou competir dentro de um ambiente e, propositalmente, em algum momento vão interagir conversando uns com os outros conforme HUHNS<sup>14</sup>. Para o conjunto destes agentes e suas interações é dado o nome de Sistema Multiagente (SMA) em ZAMBONELLI<sup>15</sup>. É um campo de bastante interesse, tanto para a representação social quanto para a resolução distribuída de problemas. Através destas interações, os agentes formulam o que compreendem sobre seu ambiente e buscam, continuamente, a conquista (resolução) de seu objetivo (problema).

#### **3.2 Jason**

Para a implementação de uma SMA as propriedades referentes a objetivos e interações devem ser consideradas no desenvolvimento computacional. A linguagem de programação abstrata AgentSpeak(L) descreve como implementar eventos e ações sobre o comportamento do agente com seu ambiente como explicado em RAO<sup>16</sup>, ou seja, na implementação de agentes BDI (Belief-Desire-Intention ou Crença-Desejo-Intenção).

Jason é um interpretador baseado em Java e é uma extensão da linguagem AgentSpeak(L). Segundo BORDINI<sup>17</sup> possui os seguintes requisitos:

- A linguagem deve apoiar a delegação em nível de objetivos;
- A linguagem deve fornecer mecanismos para resolver problemas orientados a estes objetivos (ação dos agentes no cumprimento do dever);
- A linguagem deve servir na produção de sistemas sensíveis ao ambiente;
- A linguagem deve integrar os aspectos de orientação a objetivo com comportamento de respostas dos agentes;
- A linguagem deve apoiar o conhecimento em nível de comunicação e cooperação.

Desta forma Jason é uma plataforma que cumpre com as necessidades de desenvolvimento de SMA, considerando todas as características dos agentes, como crenças, comunicação e a possibilidade de desenvolvimento do ambiente, onde os agentes possam interagir.

### 3.3 Redes de Petri

Segundo MURATA<sup>18</sup>, a rede de Petri pode ser considerada como uma ferramenta matemática e gráfica, útil no estudo de processamento de informações em sistemas que possuem características como assíncrono, concorrente, distribuído, não determinístico, e/ou estocástico.

Em uma abordagem de redes de Petri, sistemas são construídos a partir de seus eventos, atividades e processos. Sendo um conjunto de processos caracterizado pelo modo que evoluem, o qual pode ser simultâneo ou não. Segundo GONÇALVES<sup>19</sup>, em se tratando de evoluções simultâneas pode-se medir o grau de independência entre processos pela necessidade de pontos de sincronização, o que diferencia as redes de Petri de outros modelos, pois permitem a representação de atividades paralelas cooperativas, competitivas ou concorrentes. Uma Rede de Petri tem três elementos fundamentais de livre interpretação, que são:

- Lugar: pode ser um estado parcial, uma condição, um conjunto de recursos, um procedimento, etc;
- Ficha: é representada por um ponto em um lugar. Pode representar que a condição que remete ao lugar é verdadeira, representar objetos em um algum estado, representar quantidade e etc;
- Transição: é associada a um evento que ocorre no sistema e pode ser representada por uma barra ou retângulo.

CASSANDRAS<sup>20</sup> define rede de Petri como um grafo bipartido, o qual pode ser definido da seguinte forma  $(P, T, A, w)$ , onde:

- $P$  é o conjunto de lugares, os nós do grafo;
- $T$  é o conjunto de transições (outro tipo de nó do grafo);
- $A \subseteq (P \times T) \cup (T \times P)$ , é o conjunto dos arcos que vão de transição para lugar e lugar para transição (bipartição refere-se a esta característica do grafo);
- $w$  é a função peso dos arcos.

Nestes lugares podem constar fichas, que descrevem informações ou atividades, representando o fluxo da rede de Petri. Aliado a este conceito também existem as redes de Petri coloridas. São redes que possuem maior expressividade dos processos que ocorrem pelo fluxo da rede em nível de dados, tal e qual uma linguagem de programação pode representar, sendo que a cor é um tipo de dado específico conforme BORGES<sup>21</sup>. As fichas nas redes Coloridas são associadas com dados específicos.

Desta forma, essa técnica tem sido utilizada para modelar computacionalmente sistemas, representando seus processos, transições, utilização de recursos, delimitadores e outras características importantes para o funcionamento do objeto estudado.

## 4 RESULTADOS E DISCUSSÕES

### 4.1 Modelagem em rede de Petri

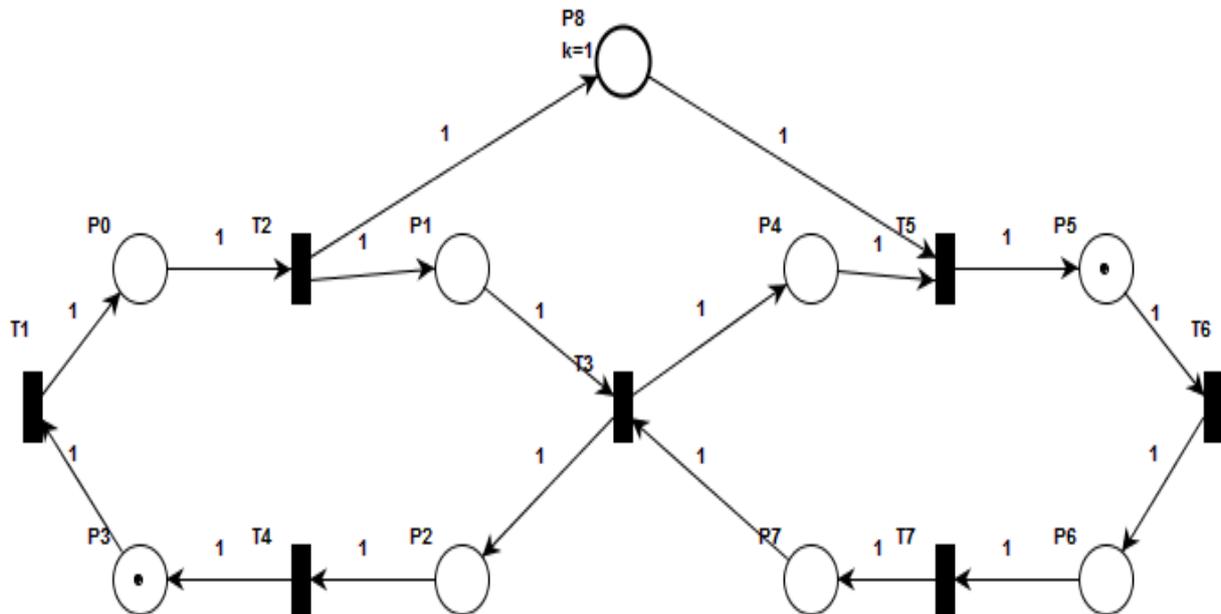


Figura 1. Modelagem do SMA em rede de Petri

O estudo apresentado aqui tem como objetivo o alinhamento da rede de Petri para auxiliar o desenvolvimento de sistemas multiagentes. A representação entre SMA e rede Petri se baseia no modelo proposto por CELAYA<sup>6</sup>.

O lugar modela uma atividade em particular, a ficha especifica a realização de uma atividade, a transição modela o fim de uma atividade de um agente. O disparo da transição indica que o agente terminou a atividade do lugar anterior à transição e iniciou a posterior da transição. O estudo de caso a ser utilizado é um sistema simples onde dois bombeiros interagem em um ambiente para controlar um incêndio.

No sistema multiagente da Figura 1, o bombeiro2 é responsável em acessar o lugar para reabastecer os equipamentos de água (neste exemplo considera-se somente o uso de baldes) e leva até o ponto de troca onde o bombeiro1 tem como missão apagar o incêndio. As transições T1 e T6 relacionam o disparo de momentos envolvendo as atividades: encher balde de água no hidrante e apagar incêndio, respectivamente. Na transição T3 têm-se as atividades para a troca de baldes onde bombeiro2 à esquerda entrega um equipamento cheio e na esquerda o bombeiro1 entrega equipamento vazio.

As crenças e intenções que permeiam o agente bombeiro1 são relacionadas em: na existência de incêndio, ir até o ponto de troca, conseguir um balde cheio de água e só então se direcionar ao local do incêndio. Para isso em P8 é modelado um recurso de rede Petri onde o máximo de unidades é  $k = 1$ , ou seja, o bombeiro que irá apagar incêndio vai aguardar até que o outro agente forneça um balde de água cheio.

## 4.2 Aplicação

Através da modelagem do sistema em rede Petri é possível visualizar algumas estruturas chaves e controles necessários para implementar o SMA. Existem atividades a serem cumpridas para cada um dos dois agentes envolvidos no sistema, assim como uma restrição para o equipamento com água.

Através da modelagem, conferida com algumas simulações da rede Petri, verifica-se que além de estruturada as propriedades, os agentes também, em alguns estados, podem trabalhar simultaneamente, em outros não. Como é o caso do bombeiro encarregado de manter o equipamento com água, que vai esperar pela execução do bombeiro no combate ao incêndio e também para repor mais água, aguardando ao lugar próximo ao hidrante.

Analisando as simulações modeladas através da rede de Petri podemos ter uma visão das primitivas e instruções da linguagem AgentSpeak(L) que deverão ser usadas na plataforma Jason para que haja coordenação no sistema, então são implementadas em Jason algumas funções, crenças e comunicações, baseado no modelo em rede de Petri.

Ordenando conforme os fatos ocorrem no modelo, temos que o sistema inicia em bombeiro1 com:

```
+!start : true <-  
  
+contFogo(5);  
  
.send(bombeiro2,tell,incendio).
```

Onde a simulação é iniciada (*!start*), o bombeiro1 avista os focos de incêndio e acredita que são cinco (isto é específico neste exemplo, pois poderia ser utilizado o método *.random()* para que o agente descreva aleatoriamente quantos focos visualiza). Em seguida se comunica com bombeiro2 avisando do incêndio. Em bombeiro2 temos que:

```
+incendio[source(A)] <-  
  
.print("Mensagem de incêndio recebida de ",A);  
  
!encherAgua.  
  
+!encherAgua : true <-  
  
.print("Pegando Agua...");  
  
.wait(1000);  
  
.print("Enchendo ...");  
  
.wait(2000);  
  
.send(bombeiro1,achieve,baldeCheio).
```

O agente bombeiro2 recebe o aviso sobre o incêndio, devido a primitiva *tell* no método *.send()*, a crença sobre incêndio é adicionada direto em sua base de crenças. Em seguida, aciona seu plano *!encherAgua* para abastecer o equipamento no hidrante.

Os tempos de 1000ms e 2000ms inseridos nos métodos `.wait()` são de acordo com o tempo levado pelo bombeiro2 em chegar até o hidrante, encher o balde de água e retornar ao ponto de troca. Quando realiza a sua tarefa, avisa ao agente bombeiro1 sobre a entrega do balde cheio de água, através também de um método `.send()` entretanto usa a primitiva `achieve`, para que o destinatário da mensagem inclua a atividade com `baldeCheio` em sua lista de eventos. Bombeiro1 aciona, portanto:

```
+!baldeCheio: not baldeCheio <-

    +baldeCheio;

    .print("Indo até incêndio.");

    .wait(2000);

    !carry_to(bombeiro2).
```

Sendo que o bombeiro1 se possuir a crença no balde cheio, este leva 2000ms para chegar ao incêndio e utilizar a água. Em seguida retorna até o agente bombeiro2 que já está no ponto de troca, conforme movimento de estados simulados na rede de Petri.

Os agentes ficam interagindo com estes planos, bombeiro1 fica percorrendo todo o ambiente até que todos os focos de incêndios sejam controlados, avisando para bombeiro2 se precisa de água ou se já conseguiu controlar o incêndio conforme plano abaixo:

```
+!caminhada: baldeCheio & incendio(bombeiro1) <-

    apaga(fogo);

    -incendio(bombeiro1);

    -baldeCheio;

    .print("Sem água, retornando ao ponto de troca");

        ?contFogo(X);

        ++contFogo(X - 1);

        .print(X," focos de incêndio!");

    if(X > 0.0){

        .send(bombeiro2,achieve,encherAgua);

    }else{

        .print("Incêndio controlado!");

        .send(bombeiro2,tell,incendioControlado);

    }.
```

O ambiente foi desenvolvido na plataforma Jason, em Java, através de uma matriz onde os agentes percorrem as posições até apagar todos os focos de incêndio, utilizando as bibliotecas de *Grid* de *Jason.environment*. A única crença inicial para todos os agentes é a posição da matriz para a troca de baldes que é programada antes do início da simulação, que é nas coordenadas do bombeiro2. Toda a matriz é visualizada na Figura 2.

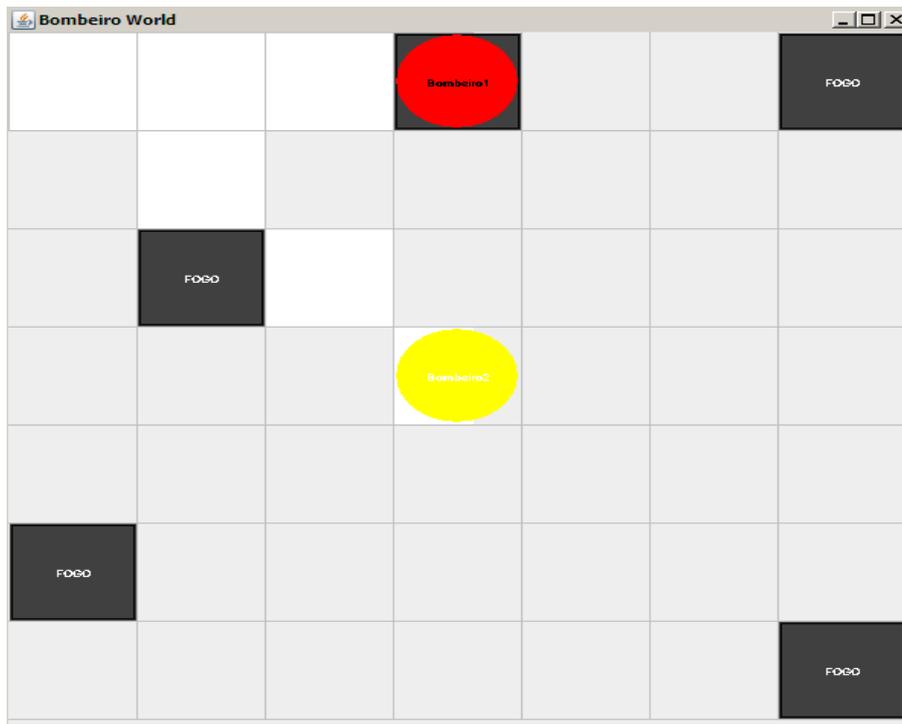


Figura 2. Visualização gráfica do sistema

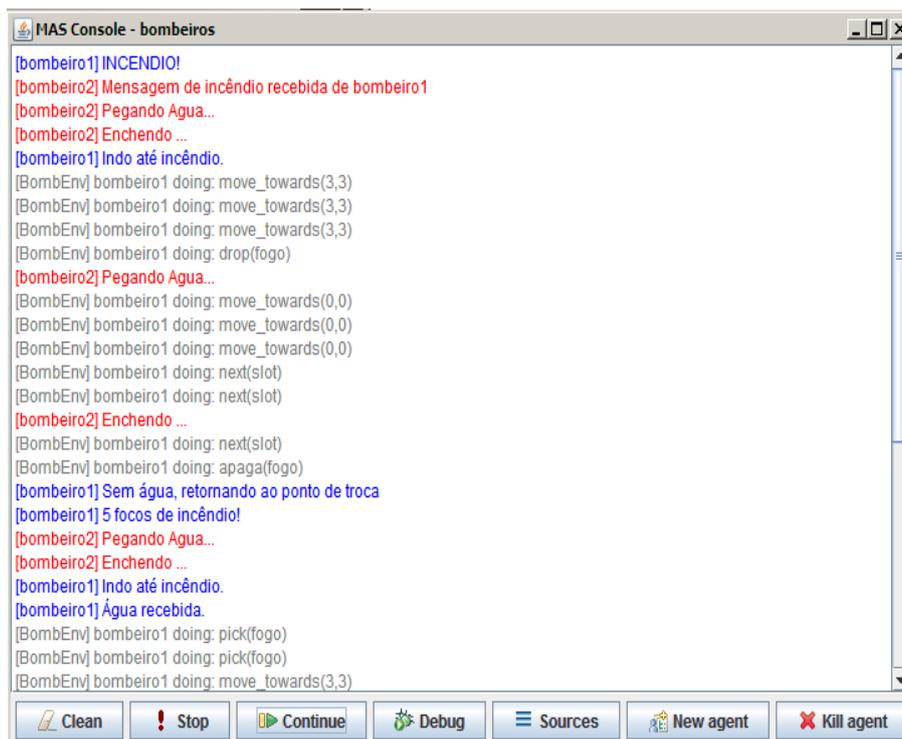


Figura 3. Visualização das mensagens dos agentes no MAS Console do Jason

Conforme visualizado na modelagem em rede de Petri das atividades dos bombeiros, algumas ações dos agentes dependem das trocas de mensagens uns com os outros. Na Figura 3 acima pode ser observado que, além da utilização dos métodos `.send()`, também são exibidas, na tela de console do ambiente Jason, outras mensagens para que usuários do SMA possam acompanhar o trabalho dos agentes.

#### 4 CONSIDERAÇÕES FINAIS

Através do desenvolvimento do projeto observa-se um alinhamento entre a rede Petri e o desenvolvimento de Sistemas Multiagentes. Redes de Petri tornaram-se um bom método para modelar e observar situações ou eventos discretos sobre o sistema de agentes. Favorecendo um ganho de tempo na modelagem e avaliação das estruturas do ambiente do SMA antes das decisões de codificação.

Como a técnica proporciona um desenho sobre as estruturas do sistema proposto, facilita aos desenvolvedores tomar decisões em quais situações seriam necessárias troca de mensagens ou quais pontos de bloqueio deveriam ser cuidados no momento da implementação. Assim, através destas observações, propriedades de comunicação e interações entre os agentes podem ser visualizadas mais facilmente para a obtenção do sistema completo.

Fica claro que o sistema utilizado é simples, porém o mesmo serve para mostrar a possibilidade de expandir esse sistema para maiores proporções, claro, acarretando maior complexidade, porém não há dúvidas que um sistema maior com seus problemas de sincronismo e coordenação bem resolvidos, poderia ser usado em uma empresa para controlar diversos robôs autônomos para controlar um incêndio de grandes proporções.

No entanto, esta ideia precisa ser amadurecida e muito bem pesquisada, passando-se a utilizar na parte de modelagem, redes de Petri de alto nível para proporcionar a construção de um modelo de grande porte.

#### 5 REFERÊNCIAS

1. APPIO, A. R.; HÜBNER, J. F. (2005) Sistema Multiagentes Utilizando a Linguagem AgentSpeak(L) para Criar Estratégias de Armadilha e Cooperação em um Jogo Tipo PacMan. XIV Seminário de Computação, Universidade Regional de Blumenau, 127-138pp.
2. ZHOU, M.; DICESARE, F. Petri Net Synthesis for Discrete Event Control of Manufacturing Systems. Ed. Kluwer Academic Publishers, 1993, 1-229p.
3. LINDER, M. S.; COSTA, A. L. (2007) Modelagem Comportamental de Sistemas Multiagentes. Anais do XXVII Congresso da SBC. VI ENIA - Encontro Nacional de Inteligência Artificial, 1172-1181pp.
4. CABAC, L.; MOLDT, D. (2004) Formal semantics for AUML agent interaction protocol diagrams. Proceedings of AOSE, 47-61pp.
5. BAI, Q.; ZHANG, M.; WIN, K. T. (2004) A Colored Petri Net Based Approach for Multi-agent Interactions. 2nd International Conference on Autonomous Robots and Agents, 152-157pp.
6. CELAYA, J. R.; DESROCHERS, A. A.; GRAVES, R. J. (2009) Modeling and Analysis of Multi-agent Systems using Petri Nets. Journal of Computers, Vol. 4, N. 10, 981-996pp.
7. FERNANDES, J. M.; BELO, O. (1998) Modeling Multi-Agent Systems Activities Through Colored Petri Nets: An Industrial Production System Case Study. 16th IASTED International Conference on Applied Informatics, 1-4pp.

8. COST, R. S.; CHEN, Y.; FININ, T.; LABROU, Y.; PENG, Y. (1999) Modeling Agent Conversations with Colored Petri Nets. Working Notes of the Workshop on Specifying and Implementing Conversation Policies, Autonomous Agents, 1-8pp.
9. PADILHA, T. P. P.; JACOMÉ, T. F. (2002) O Uso de Técnicas de Modelagem de Agentes em Ambientes Educacionais. VI Congresso Iberoamericano, IV Simpósio Internacional de Informática Educativa, VII Taller Internacional de Informática Educativa, 226-229pp.
10. SILVA, C. E. O.; DIMURO, G. P.; COSTA, A. C. R. (2010) Especificando Rotinas Organizacionais de Sistemas Multiagentes: Detecção de Desvios com Base no Conceito de Expectativa. IV Workshop Escola de Sistemas de Agentes, seus Ambientes e aplicações – IV WESAAC. 139-142pp.
11. WOOLDRIDGE, M.; JENNINGS, N. R. (1994) Intelligent Agents: Theory and Practice. Knowledge Engineering Review, Vol. 10, N. 2. 115-152pp.
12. BORDINI, R. H.; VIEIRA, R.; MOREIRA, Á. F. (2001) Fundamentos de Sistemas Multiagentes. In Ferreira, C. E., ed., Anais do XXI Congresso da Sociedade Brasileira de Computação (SBC2001), Volume 2, XX Jornada de Atualização em Informática (JAI), 30 de julho - 3 de agosto, Fortaleza-CE, Brasil, 1-39pp.
13. WOOLDRIDGE, M. J. (2002) An Introduction to Multiagent Systems. Ed. John Wiley & Sons Ltd. Inglaterra, 1-362pp.
14. HUHNS, M. N.; SINGH, M. P. (1997) Conversational Agents. IEEE Internet Computing, vol. 1, N. 2, 73-75pp.
15. ZAMBONELLI, F.; JENNINGS, N. R.; WOOLDRIDGE, M. (2003) Developing Multiagent Systems: The Gaia Methodology. ACM Transactions on Software Engineering and Methodology, Vol. 12, No. 3, 317-370pp.
16. RAO, A. S. (1996) AgentSpeak (L): BDI Agents Speak Out in a Logical Computable Language. 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, 42-55pp.
17. BORDINI, R. H.; HUBNER, J. F.; WOOLDRIDGE, M. Programming multi-agent systems in AgentSpeak using Jason. Ed. John Wiley & Sons Ltd., 2007, 1-261pp.
18. MURATA, T. (1989) Petri Nets: Properties, Analysis and Applications. Proceedings of IEEE, Vol. 77, N. 4, 541-580pp.
19. GONÇALVES, E. M. N. Uma abordagem para especificação de conhecimento para sistemas multiagentes cognitivos. 2012.
20. CASSANDRAS, C. G.; LAFORTUNE S. Introduction to Discrete Event Systems. Ed. Springer, 2008, Cap. 4, 223-266pp.
21. BORGES, G. A. P. Fluxo de dados em redes de Petri coloridas em grafos orientados a atores. Dissertação de Mestrado em Ciência da Computação. Instituto de Matemática e Estatística da Universidade de São Paulo, 2008, 1-74pp.