

IDENTIFICAÇÃO DE PAPÉIS POR OBSERVAÇÃO DE COMPORTAMENTO DE AGENTES

Jomi Fred Hübner
FURB - Universidade Regional de Blumenau
Antônio da Veiga, 140. Blumenau, SC, Brasil
e-mail: jomi@furb.rct-sc.br

Antônio Carlos da Rocha Costa
Instituto de Informática - PUCRS
Av. Ipiranga, 6681. Porto Alegre - RS - BRASIL
e-mail: rocha@inf.pucrs.br

Abstract

This paper concerns the study a frequent problem in Multi-Agent Systems, how will the agents know themselves in order to get cooperation in global problem solution.

It's proposed an approach for agents' role identification based on behavior observation that will be compared with a pre-defined set of roles' description.

1 Introdução

Este trabalho aborda um problema que se insere no âmbito da Inteligência Artificial Distribuída (IAD), mais especificamente, Sistemas Multi-Agentes (SMA) [6, 2, 10]. Os SMA se opõe às abordagens da IA clássica, que tem como metáfora o comportamento humano individual, colocando como metáfora o comportamento social, com os sistemas computacionais sendo vistos como sociedades de agentes inteligentes e autônomos.

Para que os agentes possam cooperar e coordenar suas ações com os outros membros de sociedade, é importante que tenham conhecimento das capacidades, habilidades, desejos e planos dos outros agentes. A obtenção destas informações por comunicação entre os agentes nem sempre é possível em sociedades abertas, seja pela inexistência de tal capacidade nos agentes, por problemas de diferenças semânticas na linguagem de comunicação ou mesmo pelo custo de comunicação envolvido [5]. Entretanto, grande parte do conhecimento a respeito dos outros pode ser extraído dos papéis que estes podem assumir na sociedade. Neste contexto, o problema é como os agentes da sociedade conhecerão ou aprenderão os papéis uns dos outros para cooperarem na solução de problemas. Berthet, Demazeau e Boissier [1] colocam que a identificação do papel de um outro agente pode ser feita de duas formas: (i) *diretamente*, através de uma requisição explícita aos agentes (uma forma de utilização pode ser verificada em [9]); ou (ii) *indiretamente*,

inferindo o papel a partir dos atos de comunicação dos agentes (duas abordagens são apresentadas em [8]).

Este trabalho apresenta um mecanismo indireto de identificação de papéis, onde é proposta uma linguagem para descrição de papéis (seção 2) e um mecanismo de obtenção do papel a partir de observação (seção 3). A fim de validar o mecanismo proposto, são feitos testes numa sociedade hipotética, a sociedade de produtores e consumidores (seção 4).

Para os exemplos, é utilizada a sociedade de Produtores e Consumidores. Esta sociedade é composta principalmente por agentes que assumem justamente estes dois papéis. Resumidamente, Produtores produzem peças e Consumidores as consomem (cf. [3] para uma descrição funcional desta sociedade). Para que esta sociedade funcione de maneira eficiente, introduz-se o papel de Intermediário que recebe as peças dos Produtores, guarda-as, e quando uma delas for requisitada, envia-a para um Consumidor. É notória a necessidade dos agentes se conhecerem nesta sociedade, como fica exemplificado no caso de Produtores que precisam conhecer os agentes que assumem papel de Intermediários para poder enviar-lhes suas peças.

2 Descrição de Papéis de Agentes

Conforme Rocha Costa [3], existem três formas de descrever um agente: pela sua estrutura, pelo seu funcionamento e/ou pelo seu comportamento. Vê-se a seguinte relação entre estes três aspectos: a estrutura possibilita certos funcionamentos, dos quais o agente possui um; um funcionamento permite que o agente assuma um determinado conjunto de comportamentos, sendo o comportamento a parte externa (no sentido de observável no tempo) do funcionamento (cf. [4, p. 27ss] para uma análise destes três aspectos). O papel de um agente é visto como o comportamento que ele apresenta na sociedade, sendo que este comportamento é determinado pelo seu funcionamento.

Descrever os comportamentos possíveis para modelar outro agente e não seus estados mentais, como é comum nos trabalhos onde existem modelos dos outros agentes com vista à cooperação, apresenta a vantagem de que não é necessário conhecer ou descobrir qual seu estado mental, uma vez que esta informação nem sempre é acessível ou mesmo confiável.

2.1 Processos

Para descrever o processo que rege o comportamento do agente, é utilizado o formalismo de especificação de processos definido por Hoare para descrever comportamentos de objetos [7]. A base para a especificação de processo está nos eventos em que determinado objeto participa. Na verdade, um *evento*, como utilizado aqui, denota uma classe de eventos, ou seja, podem existir

várias ocorrências de um evento no decorrer do tempo. O conjunto de eventos considerados relevantes para descrever o comportamento de um objeto, neste caso, de um agente, é chamado *alfabeto*. Um *processo* é descrito como um conjunto de eventos pertencentes ao alfabeto do objeto. A seguir, serão apresentadas algumas notações iniciais para especificação de processos.

Prefix

Tomando x como um evento e P como um processo. Então ¹

$$Q = (x \rightarrow P)$$

descreve um processo Q que primeiro participa no evento x e então se comporta conforme o processo P . O alfabeto de Q , denotado $\alpha(Q)$, é igual ao de P , ou seja, x está no alfabeto de P .

$$\alpha(x \rightarrow P) = \alpha P$$

Existe um tipo especial de processo que marca o fim de uma execução, denotado por **STOP**.

Exemplo 1: sendo um agente do tipo Consumidor, com o alfabeto $\{pegar_peça, consumir\}$ e que consome uma única peça, então tem-se o seguinte processo:

$$P = (pegar_peça \rightarrow consumir \rightarrow \mathbf{STOP}).$$

$$\alpha P = \{pegar_peça, consumir\}$$

Recursão

Prefix somente pode descrever processos que param. Processos que realizam coisas repetidamente podem ser descritos de maneira mais elegante utilizando recursão.

Exemplo 2: tomando o processo

$$C = (pedir_peça \rightarrow receber_peça \rightarrow consumir \rightarrow C).$$

$$\alpha C = \{pedir_peça, receber_peça, consumir\}$$

tem-se a descrição de um objeto que infinitamente pede peças, recebe as peças e as consome, ou seja, o processo realizado por um consumidor real.

Escolha

Um objeto pode se comportar de formas diferentes, dependendo das circunstâncias em que se encontra. Esse comportamento alternativo é descrito pela notação

¹ Serão utilizadas as seguinte convenções: letras minúsculas em *itálico* para eventos; letras maiúsculas em **NEGRITO** para processos.

$(x \rightarrow \mathbf{P} \mid y \rightarrow \mathbf{Q})$, sendo $x \neq y$

onde o objeto pode participar de um dos dois eventos (x ou y) e, depois disso, se comportar conforme \mathbf{P} , se o primeiro evento foi x , ou conforme \mathbf{Q} se foi y .

Exemplo 3: no caso do exemplo 2, o consumidor poderia não aceitar a peça e devolvê-la.

$\mathbf{C} = (\text{pedir_peça} \rightarrow \text{receber_peça} \rightarrow (\text{devolver} \rightarrow \mathbf{C} \mid \text{consumir} \rightarrow \mathbf{C}))$.
 $\alpha\mathbf{C} = \{\text{pedir_peça}, \text{receber_peça}, \text{devolver}, \text{consumir}\}$

Seqüência

Dois processos podem ser unidos, de modo que, quando um acaba, o outro inicia. Para isso, é utilizada a notação

$\mathbf{P} ; \mathbf{Q}$.

Normalmente um processo termina executando **STOP**, na caso da seqüência o primeiro processo não pode parar o funcionamento, mas deve passa-lo para o segundo processo. Assim, uma condição para utilização deste operador é que o último evento do processo \mathbf{P} seja **SKIP**.

Exemplo 4: um consumidor que consome duas peças poderia ser descrito pelo processo $\mathbf{P2}$

$\mathbf{P} = (\text{pegar_peça} \rightarrow \text{consumir} \rightarrow \mathbf{SKIP})$
 $\mathbf{P2} = \mathbf{P} ; \mathbf{P} ; \mathbf{STOP}$.

2.2 Papéis de Agentes

O papel de um agente é definido em função de suas relações com outros papéis, sendo descrito por um conjunto de *processos de interação* (PI). Um PI é um processo do agente que inclui eventos globais, um evento é caracterizado como global se envolve mais de um agente, correspondendo a uma interação. Do ponto de vista da sociedade como um todo, somente existem *processos globais* que envolvem vários agentes por meio de eventos globais. Os eventos globais de um processo de interação de um agente está incluído em um processo global, representando o comportamento do agente perante a sociedade.

Propõe-se que um modelo de papel seja descrito pelo predicado

$\text{papel}(\langle \text{nome_do_papel} \rangle, \langle \text{lista_de_PI} \rangle)$

onde

$\langle \text{nome_do_papel} \rangle$ é um identificador do papel sendo descrito e

<lista_de_PI> é um conjunto de processos de interação, cada PI, por sua vez, é descrito pelo predicado

pi(<PRPI>, <lista_de_processos>).

O PRPI (papel relativo ao processo de interação) é o papel relacionado ao PI sendo definido. Por exemplo, um agente Produtor tem um processo de interação onde o PRPI é Intermediário; um agente Intermediário tem, no mínimo, dois processos de interação, um com PRPI Produtor e outro com PRPI Consumidor. Cada processo de interação, além do PRPI, possui uma lista de processos que descreve como se dá esta interação. Por sua vez, o processo é descrito pelo predicado

p(<nome_do_processo>),

onde o <nome_do_processo> pode ser:

- o processo **SKIP**;
- o processo **STOP**;
- a chamada de outro processo, na forma “p(<nome_do_processo>)”.

O processo também pode ser descrito por um predicado binário:

p(<nome_do_processo>, <corpo>),

onde <corpo> pode ser:

- para o operador prefix “→”, “<evento> -> <evento>”;
- para o operador alternativa “|”, “<processo> ou <processo>”;
- para o operador seqüência “;”, “<processo> seq <processo>”.

Exemplo 5: considerando as definições feitas acima, pode-se descrever o papel do consumidor do exemplo 2 da seguinte forma:

```
papel('Consumidor', [
pi( 'Intermediário', [ p('MAIN', p('CICLO')),
p('CICLO', p('CICLO'(1)) seq p('CICLO')),
p('CICLO'(1), p(prefix, envia_mensagem(request(peca))->
p(prefix, recebe_mensagem(reply(peca(X)))->
p(skip))))
]) ]). % Fim do papel
```

onde o Consumidor tem um PI com um Intermediário ao qual ele pede peças e recebe peças.

3 Identificação de Papéis por Observação e Classificação

Este mecanismo de identificação de papéis funciona, resumidamente, da seguinte maneira: um agente observador observa uma seqüência de eventos

globais em que o agente observado está participando, montando um *trace* que é uma representação do comportamento do agente observado em relação a um outro agente. A partir do *trace*, é construído um processo que descreve o comportamento observado, da comparação deste processo obtido com outros processos previamente definidos (por exemplo, o processo que define Consumidor no exemplo 5) tem-se o papel do agente observado.

3.1 Traces de Processos

“Um *trace* do comportamento de um processo é uma seqüência finita de símbolos representando os eventos em que o processo tem participado até certo momento.” [7, p. 41] Um *trace* é denotado por uma seqüência de símbolos, separados por vírgulas, colocados entre os sinais de maior e menor.

Exemplo 6: considerando o processo definido no exemplo 1, os seguintes *traces* seriam possíveis:

<>	quando ainda não fez nada
<pegar_peça>	quando pegou a peça
<pegar_peça, consumir>	quando está consumindo a peça

Exemplo 7: considerando o processo definido no exemplo 3 (que possui recursão e escolha), poderia-se ter o seguinte *trace*:

<pedir_peça, receber_peça, devolver, pedir_peça, receber_peça, devolver, pedir_peça, receber_peça, consumir, pedir_peça, receber_peça, devolver, pedir_peça, receber_peça, consumir>

3.2 Construção de Processos a Partir de *traces*

Tendo-se um *trace* é possível inferir o processo que o originou. O problema é encontrar o melhor processo que descreve o *trace*, uma vez que um mesmo *trace* é permitido por vários processos. Por exemplo, o *trace*:

<x, y, x, y, x, y>

pode ter sido gerado por qualquer um três processos seguintes:

P1 = (x → y → x → y → x → y → **STOP**).

P2 = (x → y → x → **P2**).

P3 = (x → y → **P3**). { provavelmente a melhor opção }

Identificação de Prefix em Processos

É o caso mais simples, o processo é a própria seqüência de eventos do *trace*. Por exemplo, para o último *trace* do exemplo 6, o processo seria

P = (pegar_peça → consumir → **STOP**).

Identificação de Recursão

Processos muito simples (com pequeno número de eventos) podem ter *traces* muito extensos, é o caso de processos recursivos. Nestes casos, a técnica acima (identificação de prefix) não é eficiente. Considerando que os agentes sempre têm um comportamento cíclico, e portando a descrição de seus papéis incluem recursão, a identificação destes ciclos é essencial para a comparação do processo observado com o previamente descrito.

O método de identificação de ciclos consiste basicamente em verificar se duas sub-sequências quaisquer no *trace* se repetem, bem com identificar aquelas sequências que não se repetem.

Exemplo 8: o trace $s = \langle \text{apresenta}, \text{pegar_peça}, \text{consumir}, \text{pegar_peça}, \text{consumir}, \text{pegar_peça}, \text{consumir}, \text{pegar_peça} \rangle$, tem um ciclo, $\langle \text{pegar_peça}, \text{consumir} \rangle$ e o início $\langle \text{apresenta} \rangle$. Tem-se assim, o seguinte processo:

$(\text{apresenta} \rightarrow \mathbf{SKIP}) ; C.$

$C = (\text{pegar_peça} \rightarrow \text{consumir} \rightarrow \mathbf{SKIP}) ; C.$

Identificação de Escolha

A identificação de escolha em um processo somente é perceptível nos casos em que a escolha está dentro de um ciclo, caso contrário, o comportamento alternativo nunca será observado. A princípio, cada ciclo identificado é uma alternativa de comportamento. Assim, para o *trace* do exemplo 7 seriam identificados dois ciclos:

$\langle \text{pedir_peça}, \text{receber_peça}, \text{devolver} \rangle$ e

$\langle \text{pedir_peça}, \text{receber_peça}, \text{consumir} \rangle$

Dando origem aos processos

$C = P_1 \mid P_2 ; C.$

$P_1 = \text{pedir_peça} \rightarrow \text{receber_peça} \rightarrow \text{devolver} \rightarrow \mathbf{SKIP}.$

$P_2 = \text{pedir_peça} \rightarrow \text{receber_peça} \rightarrow \text{consumir} \rightarrow \mathbf{SKIP}.$

que possui indeterminismo entre os processos P_1 e P_2 . Portanto, o processo original do *trace* deveria ser definido como

$C = P_1 ; C.$

$P_1 = \text{pedir_peça} \rightarrow \text{receber_peça} \rightarrow ($
 $\text{devolver} \rightarrow \mathbf{SKIP} \mid \text{consumir} \rightarrow \mathbf{SKIP})$

Um descrição mais formal dos algoritmos de transformação de *trace* em processo, incluindo os algoritmos, pode ser obtido em [8].

3.3 Reconhecimento de Papéis a Partir do Comportamento

Como processos não consideram relações entre agentes, o que é essencial conforme a definição de papéis, torna-se necessário acrescentar ao procedimento de reconhecimento de processos o aspecto relacional, ou seja, as interações do agente. Nos processos, as interações são consideradas como eventos globais, embora estes eventos não considerem a identificação de quem é o outro agente na interação. Este problema é solucionado alterando-se a forma de obtenção do *trace*, considera-se o *trace* da observação de um agente como sendo relativo às suas interações dele com **um** outro agente. No caso do agente comunicar-se com mais de um agente no intervalo de observação, têm-se vários *traces* (uma para cada agente com quem mantém comunicação) e os seus vários processos reconhecidos. Estes processos reconhecidos são os processos de interação (PI).²

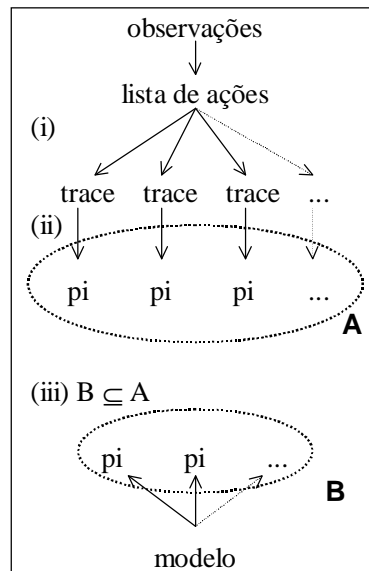


Figura 1 - Esquema da identificação de papéis por observação e classificação.

Para proceder com a identificação do papel a partir do *trace* o agente observador deve possuir um conjunto pré-estabelecido de papéis e seus processos associados, onde cada elemento deste conjunto é chamado de *modelo* (como o apresentado no exemplo 5). Este conjunto de modelos deve

² Da mesma forma que o agente tem um comportamento cíclico, também os atos de comunicação (parte dos processos do agente que é formada pelos eventos globais) apresentam-se ciclicamente formando ciclos de comunicação.

descrever os papéis em função de processos de interação. Caso o papel tenha mais de um PI, considera-se que estes PIs executam concorrentemente formando um processo único e maior que caracteriza o papel.

Para ir da lista de ações observadas (na forma de *traces*) de um agente a uma descrição de papel, são realizadas as seguintes etapas (como mostra a figura 1):

- i) Para cada agente com quem o agente alvo teve interações é criado um *trace*.
- ii) Para cada *trace* observado é construído um PI associado (cf. seção 3.2). De todos os PIs, tem-se uma lista de PIs.
- iii) A lista de PIs observada é comparada com a lista de PIs dos modelos. Esta comparação consiste em verificar se os processos de interação do modelo são subconjunto dos processos de interação observados.

Esta abordagem apresenta como vantagem o fato de se ter uma descrição (o processo) do agente observado mesmo que um processo pré-definido correspondente ao observado não tenha sido encontrado. O processo reconhecido pode ser utilizado de outras formas que não seja a de associar um papel ao agente, como por exemplo, para prever seu comportamento.

4 Um Estudo de Caso: a Sociedade Produtor-Consumidor

Para fins de exemplificação, duas descrições para o papel de Intermediário são apresentadas na figura 2, uma em que o Intermediário responde para o Produtor somente quando tem espaço disponível e outra em que responde negativamente ao pedido de espaço.

4.1 Identificação dos Papéis na Sociedade PIC

Os agentes da sociedade PIC se comunicam segundo regras definidas em um protocolo de comunicação [9]. Da observação destes agentes é obtido um *trace* formado pelas ações de comunicação realizadas por eles no decorrer do funcionamento da sociedade.

Exemplo 9: considerando uma sociedade com os agentes Jomi, Rocha, e Renata, poderia-se observar o seguinte *trace* para as ações do agente Renata:

```
recebe_mensagem(request(peça),Jomi),
recebe_mensagem(request(espço),Rocha),
envia_mensagem(reply(espço_ok),Rocha),
recebe_mensagem(inform(peça(_)),Rocha),
envia_mensagem(reply(peça(_)),Jomi),
recebe_mensagem(request(espço),Rocha),
```

```

envia_mensagem(reply(espaco_ok),Rocha),
recebe_mensagem(inform(peca(_)),Rocha),
recebe_mensagem(request(peca),Jomi),
envia_mensagem(reply(peca(_)),Jomi).

```

```

papel( 'Intermediário', [
pi( 'Consumidor', [ p('MAIN', p('CICLO')),
p('CICLO', p('CICLO'(1)) seq p('CICLO')),
p('CICLO'(1), p(prefix, recebe_mensagem(request(peca))->
p(prefix, envia_mensagem(reply(peca(X)))->
p(skip)))) ]), % Fim do pi
pi( 'Produtor', [ p('MAIN', p('CICLO')),
p('CICLO', p('CICLO'(1)) seq p('CICLO')),
p('CICLO'(1), p(prefix, recebe_mensagem(request(espaco)) ->
p(prefix, envia_mensagem(reply(espaco_ok)) ->
p(prefix, recebe_mensagem(inform(peca(X)) ->
p(skip))))))
] ) ]. % Fim do papel

papel( 'Intermediário', [
pi( 'Consumidor', [ p('MAIN', p('CICLO')),
p('CICLO', p('CICLO'(1)) seq p('CICLO')),
p('CICLO'(1), p(prefix, recebe_mensagem(request(peca))->
p(prefix, envia_mensagem(reply(peca(X)))->
p(skip)))) ]), % Fim do pi
pi( 'Produtor', [ p('MAIN', p('CICLO')),
p('CICLO', p('CICLO'(1)) seq p('CICLO')),
p('CICLO'(1), p(prefix, recebe_mensagem(request(espaco)) ->
(p(prefix, envia_mensagem(reply(espaco_ok)) ->
p(prefix, recebe_mensagem(inform(peca(X)) ->
p(skip))))
ou
p(prefix, envia_mensagem(reply(espaco_nok))->
p(skip))
)))
] ) ]. % Fim do papel

```

Figura 2: duas descrições para o papel de Intermediário

Exemplo 10: dando seqüência ao exemplo anterior, as etapas de construção do modelo do papel (cf. seção 3.3) realizam as seguintes operações:

pelo passo (i), a agente Renata tem os seguintes *traces*³:

```

t(Rocha, [recebe_mensagem(request(espaco)),envia_mensagem(reply(espaco_ok)),
recebe_mensagem(inform(peca(_))),recebe_mensagem(request(espaco)),
envia_mensagem(reply(espaco_ok)),recebe_mensagem(inform(peca(_)))])
t(Jomi, [recebe_mensagem(request(peca)),
envia_mensagem(reply(peca(_))),recebe_mensagem(request(peca)),
envia_mensagem(reply(peca(_)))])
t(Renata, []).

```

³ Os *traces* apresentam o formato: t(<nome do agente>, <eventos observados>).

o passo (ii) cria os processos:

Processo de Renata relacionado ao agente Rocha

```
MAIN      = CICLO.  
CICLO     = (CICLO(1)) ; (CICLO).  
CICLO(1) = recebe_mensagem(request(espaco)) ->  
           envia_mensagem(reply(espaco_ok)) ->  
           recebe_mensagem(inform(peca(_))) ->  
           SKIP.
```

Processo de Renata relacionado ao agente Jomi

```
MAIN = CICLO.  
CICLO = (CICLO(1)) ; (CICLO).  
CICLO(1) = recebe_mensagem(request(peca)) ->  
           envia_mensagem(reply(peca(_))) ->  
           SKIP.
```

O passo (iii) junta estes dois processos de interação formando um conjunto do qual a descrição de Intermediário apresentada na figura 2 é subconjunto, logo, o agente observado é Intermediário. Como a agente Renata poderia ter interações com mais de um Produtor e/ou Consumidor, poderiam existir vários processos de interação (mais PIs do que no modelo), mas para caracterizar a agente como Intermediária seriam necessários pelo menos um PI com um Consumidor e um PI com um Produtor.

O passo três coloca um problema interessante. É condição que se conheça o papel dos agentes relacionados com o agente alvo, ou melhor, os processos de interação tem um PRPI que, até o momento, não são conhecidos. No exemplo 10, deve-se conhecer os papéis de Rocha e Jomi (Produtor e Consumidor, respectivamente) para que o papel de Renata possa ser classificado como Intermediário. Sendo assim, a primeira vez que se classifica o papel de um agente não se pode considerar o PRPI. É necessário uma revisão futura dos papéis já identificados, quando um maior número de papéis for conhecido e existirem mais eventos a serem observados.

Os experimentos foram realizados a partir da implementação da sociedade PIC no ambiente de desenvolvimento de agentes desenvolvido por Luiz Muniz [11].

5 Conclusões

Conforme resultados obtidos pelos experimentos realizados, o mecanismo de identificação de papéis apresentado mostrou-se bastante adequado, principalmente no caso de sociedades abertas, onde novos agentes podem dinamicamente entrar na sociedade já em funcionamento. A principal vantagem deste mecanismo é que o agente que deseja identificar o papel do

outro não precisa interagir com ele, basta observá-lo, não sendo necessário conhecer a linguagem e protocolo do outro.

Poderia-se argumentar que este mecanismo é muito rígido, já que o agente deve se comportar exatamente como um dos processos pré-definidos. Esta rigidez pode ser atenuada de duas formas: primeiro, descrevendo para cada papel vários processos que o caracterizam e, segundo, definindo os processos de maneira suficientemente genérica, dando mais “liberdade” ao comportamento dos agentes.

6 Referências Bibliográficas

- [1] BERTHET, Sabine, DEMAZEAU, Yves, BOISSIER, Oliver. **Knowing Each Other Better**. In: *11st International Workshop on Distributed Artificial Intelligence*. Glen Arbor, 1992. p. 1-20.
- [2] BOND, Alan H.; Gasser, Les (Eds.). **Readings in Distributed Artificial Intelligence**. San Mateo, CA: Morgan Kaufmann, 1988.
- [3] COSTA, Antônio Carlos da Rocha; CASTILHO, José Mauro Volkmer de; CLÁUDIO, Dalcídio Moraes. Functional Processes and Functional Roles in Societies of Computing Agents. In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 10., 1993, Porto Alegre. **Anais...** Porto Alegre: SBC, 1993.
- [5] COSTA, Antônio Carlos da Rocha; HÜBNER, Jomi Fred; BORDINI, Rafael Heitor. On Entering an Open Society. In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 11., 1994, Fortaleza. **Anais...** Fortaleza: SBC, 1994. p. 535-546.
- [6] DEMAZEAU, Yves, MÜLLER, Jean Pierre. **Decentralized Artificial Intelligence**. In: *Decentralized Artificial Intelligence - I*. Organização: DEMAZEAU, Yves, MÜLLER Jean-Pierre. North-Holland, Elsevier Science Publishers, 1990. p. 3ss.
- [7] HOARE, Charles Antony Richard. **Communicating Sequential Processes**. New Jersey: Prentice-Hall, 1985. 256p.
- [8] HÜBNER, Jomi Fred. **Migração de Agentes em Sistemas Multi-Agentes Abertos**. Porto Alegre, UFRGS, 1995. 124p. Dissertação de Mestrado.
- [9] HÜBNER, Jomi Fred; COSTA, Antônio C. R.; BORDINI, Rafael Heitor. Identificação de Papéis por Apresentação na Entrada de Agentes em Sociedades Abertas. **Dynamis**, Blumenau, vl. 3, nr. 13, p.41-53, out/dez 1995.
- [10] WOOLDRIDGE, Michael J.; JENNINGS, Nicholas R (Eds.). **Intelligent Agents**. Berlin, Germany: Springer-Verlag, 1994.
- [11] MONIZ, Luis Manuel Ferreira Fernandes. **SSAA: Sistema para Simulação de Agentes e Ambientes**. Lisboa: Universidade Técnica de Lisboa, jun. 1993. Dissertação de Mestrado.