IDENTIFICAÇÃO DE PAPÉIS POR OBSERVAÇÃO DE COMPORTAMENTO EM SOCIEDADES DE AGENTES

Jomi Fred Hübner
e-mail: jomi@furb.rct-sc.br

Antônio Carlos da Rocha Costa
e-mail: rocha@inf.ufrgs.br

Resumo

Este trabalho apresenta um estudo sobre um problema comum em Sistemas Multi-Agentes, como os agente irão se conhecer para cooperarem na solução de problemas globais. Para resolve-lo, é proposto uma abordagem de identificação dos papéis dos agentes a partir da observação de seus comportamento confrontados com descrições pré-definidas de papéis.

Abstract

This paper concerns the study a frequent problem in Multi-Agent Systems, how will the agents know themselves for them 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.

Palavras-chave

Inteligência Artificial Distribuída, Sistemas Multi-Agentes, descrição de papéis, processos, identificação de papéis, aprendizado em Sistemas Multi-Agentes.

1. Introdução

Este trabalho aborda um problema que se insere no âmbito da Inteligência Artificial Distribuída (IAD), mais especificamente, Sistemas Multi-Agentes (SMA) [DEM90, BON88, WOO94]. 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 da sociedade, é necessário que tenham conhecimento das capacidades, habilidades, desejos e planos dos outros agentes. Grande parte do conhecimento a respeito dos outros pode ser extraído dos papéis que estes podem assumir na sociedade. Assim sendo, o problema colocado é como os agentes da sociedade conhecerão ou aprenderão os papéis uns dos outros para cooperarem na solução de problemas. Berthet, Demazeau e Boissier [BER92] 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 [HÜB95b]); ou (ii) **indiretamente**, inferindo o papel a partir dos atos de comunicação dos agentes (duas abordagens são apresentadas em [HÜB95a]).

Neste trabalho será descrito um mecanismo indireto de identificação de papéis, onde é proposta um linguagem para descrição de papéis e um mecanismo de obtenção do papel a partir da observação. A fim de validar os mecanismos propostos, foram feitos testes numa sociedade hipotética, a sociedade de produtores e consumidores.

2. Descrição de papéis de agentes

Conforme [COS93a], 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) do funcionamento (cf. [COS93b, p. 27ss] para uma análise destes três aspectos).

Para descrever o funcionamento do agente, pode-se utilizar a noção de processos, uma vez que estes também descrevem um conjunto de comportamentos. Descrever os comportamentos possíveis do 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 [HOA85]. 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, ou mais especificamente, de um agente, é chamado *alfabeto*. Um *processo* pode ser descrito em termos de um conjunto de eventos pertencentes ao alfabeto do objeto. A seguir, serão apresentadas algumas notações iniciais para especificação de processos.

2.1.1 Prefix

Tomando x como um evento e **P** como um processo. Então [1]

$$\mathbf{Q} = (x \rightarrow \mathbf{P})$$

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

$$\alpha(x \rightarrow \mathbf{P}) = \alpha\mathbf{P}$$

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:

$$\mathbf{P} = (\mathit{pegar\_peça} \rightarrow \mathit{consumir} \rightarrow \mathbf{STOP}).$$
$$\alpha\mathbf{P} = \{\mathit{pegar\_peça}, \mathit{consumir}\}$$

2.1.2 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

$$\mathbf{C} = (\mathit{pedir\_peça} \rightarrow \mathit{receber\_peça} \rightarrow \mathit{consumir} \rightarrow \mathbf{C}).$$
$$\alpha\mathbf{C} = \{\mathit{pedir\_peça}, \mathit{receber\_peça}, \mathit{consumir}\}$$

---

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

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

2.1.3 Escolha

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

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

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

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

$\mathbf{C} = (\textit{pedir\_peça} \rightarrow \textit{receber\_peça} \rightarrow (\textit{devolver} \rightarrow \mathbf{C} \mid \textit{consumir} \rightarrow \mathbf{C}))$.
$\alpha\mathbf{C} = \{\textit{pedir\_peça}, \textit{receber\_peça}, \textit{devolver}, \textit{consumir}\}$

2.1.4 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 **P** seja **SKIP**.

Exemplo 4: um consumidor que consome duas peças poderia ser descrito pelo processo **P2**

$\mathbf{P} = (\textit{pegar\_peça} \rightarrow \textit{consumir} \rightarrow \mathbf{SKIP})$
$\mathbf{P2} = \mathbf{P} \,;\, \mathbf{P} \,;\, \mathbf{STOP}$.

2.2 Papéis de agentes

Um agente tem processos internos e participa de processos globais. *Processos internos* são aqueles onde não existe interação com outros processos de outros agentes. Os *processos globais* são da sociedade como um todo, não "pertencem" a nenhum agente específico. Processos globais são formados por eventos globais, um evento é caracterizado como global se envolve mais de um agente, logo, corresponde a uma interação. O conjunto dos eventos globais onde o agente participa define o seu comportamento. A participação do agente num processo global é feita por um processo que mistura eventos internos e eventos globais, este tipo de processo é chamado *processo de interação* (PI). Um PI define um comportamento parcial do agente, ou seja, o comportamento do agente em relação a um processo global. Um conjunto de PIs forma um modelo de *papel*.

Os modelos de papéis são descritos, em Prolog, pelo predicado

papel(Nome_do_Papel, Lista_de_PI).

onde, os elementos de Lista_de_PI são

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, os processos são descritos pelo predicado p(Nome_do_Processo, Corpo),

onde o Corpo pode ser:

- o processo **SKIP**;
- o processo **STOP**;
- a chamada de outro processo, na forma "p(Nome_Processo)";
- o operador "→", na forma "p(prefix, Evento → Processo)";
- o operador "|", na forma "Processo ou Processo";
- o operador ";", na forma "Processo seq Processo".

Exemplo 5: considerando as definições feitas acima, pode-se descrever o papel de consumidor 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 pi
]). % 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 se dá 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 que caracterizam papéis (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." [HOA85, 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, 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 }

3.2.1 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**).

3.2.2 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-seqüências quaisquer no *trace* se repetem no *trace*, bem com identificar aquelas seqüências que não se repetem.

Exemplo 8: o trace *s* = <*apresenta, pegar_peça, consumir, pegar_peça, consumir, pegar_peça, consumir, pegar_peça*>, tem um ciclo, <*pegar_peça*, *consumir*> e o início <*apresenta*>. Tem-se assim, o seguinte processo:

(*apresenta* → **SKIP**) ; **C**.
C = (*pegar_peça* → *consumir* → **SKIP**); **C**.

3.2.3 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:

< *pedir_peça*, *receber_peça*, *devolver*> e
< *pedir_peça*, *receber_peça*, *consumir*>

Dando origem aos processos

C = $\mathbf{P_1}$ | $\mathbf{P_2}$; **C**.
$\mathbf{P_1}$ = *pedir_peça* → *receber_peça* → *devolver* → **SKIP**.
$\mathbf{P_2}$ = *pedir_peça* → *receber_peça* → *consumir* → **SKIP**.

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

C = $\mathbf{P_1}$; **C**.
$\mathbf{P_1}$ = *pedir_peça* → *receber_peça* → (*devolver* → **SKIP** | *consumir* → **SKIP**)

Um descrição mais formaldos processos de transformação de *trace* em processo, incluindo os algoritmos, pode ser obtido em [HÜB95a].

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 quem é o outro agente na interação. Este problema é solucionado com a alteração na forma de obtenção do *trace*, considera-se o *trace* da observação de um agente como sendo relativo às suas interações com **um** outro. 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]

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 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.

Nesta forma de identificação de papel, procura-se nos modelos o processo correspondente ao processo observado (construído conforme visto acima). Quando encontrado, pode-se dizer que o papel do agente observado é o papel associado ao modelo encontrado. 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 sejam a de associar um papel ao agente, como por exemplo, para prever seu comportamento.

4. Um Estudo de Caso: a Sociedade Produtor-Consumidor

4.1 Descrição da sociedade

A sociedade de Produtores e Consumidores é composta principalmente por agentes que assumem justamente estes dois papéis. Resumidamente, Produtores produzem peças e Consumidores as consomem (cf. [COS93a] 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.

4.2 Papéis na sociedade PIC

O papel de Produtor tem duas descrições, uma que considera a resposta negativa de espaço e outra que não a considera. Isto se faz necessário porque um Produtor pode nunca receber essa resposta negativa (pelo fato do Intermediário sempre ter espaço). Se isso acontecer, o papel deste agente nunca será conhecido sem o primeiro modelo de Produtor.

```
papel('Produtor', [
pi( 'Intermediário', [
   p('MAIN', p('CICLO')),
   p('CICLO', p('CICLO'(1)) seq p('CICLO')),
```

---

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.

```
   p('CICLO'(1), p(prefix, envia_mensagem(request(espaco)) ->
                 p(prefix, recebe_mensagem(reply(espaco_ok)) ->
                 p(prefix, envia_mensagem(inform(peca(X))) ->
                 p(skip)))))
]) ]). % Fim do papel

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

O papel Intermediário tem a seguinte descrição:

```
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
```

4.3 *Traces* na sociedade PIC

Os agentes da sociedade PIC se comunicam segundo regras definidas em um protocolo de comunicação [HÜB95b], desta comunicação é gerado um *trace* formado pelas ações de comunicação realizadas pelos agentes no decorrer do funcionamento da sociedade.

Exemplo 9: considerando os agentes Jomi, Rocha, e Renara, poderia-se ter o seguinte *trace* para as ações do agente Renata:

recebe_mensagem(request(peça),Jomi),
recebe_mensagem(request(espaço),Rocha),
envia_mensagem(reply(espaço_ok),Rocha),
recebe_mensagem(inform(peça(_)),Rocha),
envia_mensagem(reply(peça(_)),Jomi),
recebe_mensagem(request(espaço),Rocha),
envia_mensagem(reply(espaço_ok),Rocha),

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

4.4 Identificação dos papéis

Para ir da lista de ações de um agente em uma descrição de papel, são realizadas as seguintes transformações à lista de ações resultante da observação (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. 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.

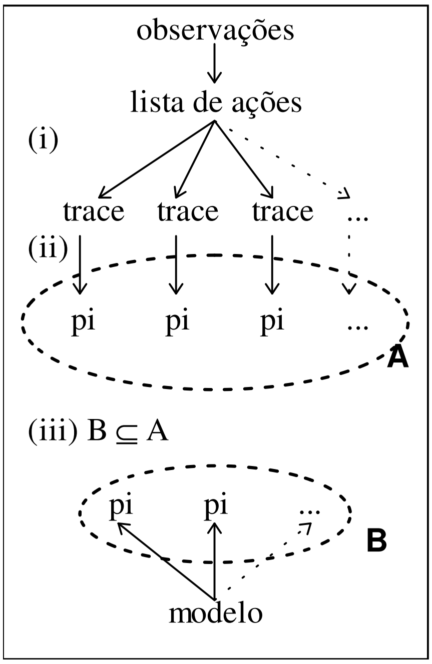

Figura 1 - Esquema da identificação de papéis por observação e classificação.

Exemplo 10: dando seqüência ao exemplo 9, os passos acima realizam as seguintes transformações:

Pelo passo (i), a agente Renata tem os seguintes traces[3]:

t(Rocha, [recebe_mensagem(request(espaço)),envia_mensagem(reply(espaço_ok)),
recebe_mensagem(inform(peca(_))),recebe_mensagem(request(espaço)),
envia_mensagem(reply(espaço_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, []).

O passo (ii) cria os processos:

---

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

Processo de Renata relacionado ao agente Rocha

```
MAIN     = CICLO.
CICLO    = (CICLO(1)) ; (CICLO).
CICLO(1) =
         recebe_mensagem(request(espaço)) ->
         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 feita na seção 4.2 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, a princípio, 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.

Para o exemplo 10, no passo (ii), a revisão do papel de Renata passaria a ser:

Processo de Renata relacionado ao **papel Produtor**

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

Processo de Renata relacionado ao **papel Consumidor**

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

Estes dois processos de interação realmente caracterizam um Intermediário, por considerarem o papel dos agentes relacionados.

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. Porém, devido a complexidade dos algoritmos, a identificação do papel de um agente pode ser demorada.

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.

Novos trabalhos poderiam ser realizados considerando que este mecanismo não garante que o papel identificado é o real, pelos seguintes motivos:

- o agente observador pode não ter observado suficientemente o agente alvo, e assim, o agente observado ainda pode apresentar um comportamento não observado anteriormente: o *problema da parada*;
- o agente observador não tem o papel observado no conjunto de papéis pré-definidos;
- o agente observador não tem capacidade de descrever o papel do agente observado.

6. Referências Bibliográficas

[BER92] BERTHET, Sabine, DEMAZEAU, Yves, BOISSIER, Oliver. **Knowing Each Other Better.** In: *11st International Workshop on Distributed Artificial Intelligence*. Glen Arbor, 1992. p. 1-20.

[BON88] BOND, Alan H.; Gasser, Les (Eds.). **Readings in Distributed Artificial Intelligence**. San Matew, CA: Morgan Kaufmann, 1988.

[COS93a] 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.

[COS93b] COSTA, Antônio Carlos da Rocha. **Inteligência de Máquina**: Esboço de uma Abordagem Construtivista. Porto Alegre: CPGCC da UFRGS, 1993. 168p. Tese de Doutorado.

[COS94] 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.

[DEM90] DEMAZEAU, Yves, MÜLLER, Jean Pierre. **Decentralized Artificial Intelligence.** In: *Decentralized Artificial Intelligence - 1*. Organização: DEMAZEAU, Yves, MÜLLER Jean-Pierre. North-Holland, Elsevier Science Publishers, 1990. p. 3ss.

[HOA85] HOARE, Charles Antony Richard. **Communicating Sequential Processes.** New Jersey: Prentice-Hall, 1985. 256p.

[HUB95a] HÜBNER, Jomi Fred. **Migração de Agentes em Sistemas Multi-Agentes Abertos**. Porto Alegre, UFRGS, 1995. 124p. Dissertação de Mestrado.

[HÜB95b] HÜBNER, Jomi Fred; COSTA, Antônio C. R.; BORDINI; Rafael Heitor. Identificação de Papéis por Apresentação na Entrada de Agentes em Sociedades Abertas. **Dymamis**, Blumenau, vl. 3, nr. 13, p.41-53, out/dez 1995.

[WOO94] WOOLDRIDGE, Michael J.; JENNINGS, Nicholas R (Eds.). **Intelligent Agentes**. Berlin, Germany: Springer-Verlag, 1994.