

# PartNet+: Simulando Parcerias entre Múltiplos Agentes

Júlio de L. do R. Monteiro, Jaime S. Sichman

Laboratório de Técnicas Inteligentes (LTI)  
Escola Politécnica (EP)  
Universidade de São Paulo (USP)  
Av. Prof. Luciano Gualberto, 158 travessa 3  
05508-900 – São Paulo – SP – Brasil

{julio.monteiro, jaime.sichman}@poli.usp.br

***Abstract.** The PART-NET system was developed to study partnerships amongst agents with a simple architecture, striving to observe macroscopic relations in the application of a social theory, using social simulation as a theory proving tool. PartNET+, written in Java, tries to achieve operational improvements to the predecessor, also expanding its functionality and creating a framework to facilitate the development of other partnership simulation tools.*

***Resumo.** O sistema PART-NET foi desenvolvido para estudar parcerias entre agentes com uma arquitetura bastante simples, buscando observar as relações macroscópicas obtidas através da aplicação de uma teoria social, utilizando a simulação como ferramenta de prova. O PartNET+, escrito em Java, procura melhorar o primeiro trazendo benefícios operacionais, e até mesmo expandir sua funcionalidade, criando uma estrutura para facilitar o desenvolvimento de novas ferramentas de simulação de parcerias.*

## 1. Introdução

Este trabalho situa-se no contexto de Simulação Baseada em Multiagentes (MABS, do inglês, Multi-Agent Based Simulation), A proposta é estudar um sistema de simulação MABS em detalhes para poder entender os conceitos de sistemas multiagentes (SMA) envolvidos, através da simulação computacional, além de desenvolver um software de simulação versátil e eficiente e fácil de utilizar para realizar outros tipos de simulações.

O sistema escolhido foi o PART-NET [PEDONE'98]. Utilizando uma arquitetura de agentes e um algoritmo de formação de parcerias bastante simplificados, esse sistema mostrou-se um sistema suficientemente adequado para estudar MABS.

O PART-NET busca comprovar, através de uma técnica de simulação, uma teoria social sobre a importância da existência de agentes com diferentes perfis numa sociedade que contém agentes dotados da capacidade de realizar ações, e que têm uma série de objetivos individuais, e que podem eventualmente estabelecer parcerias para atingir tais objetivos. A intenção é estudar como estas parcerias se estabelecem de acordo com as diferentes distribuições de agentes de três diferentes tipos: substancialistas, instrumentalistas e avaros.

Estes 3 tipos de agentes possuem objetivos e ações e cada um deles irá procurar parceiros na sociedade para ajudá-los a cumprir todos os seus objetivos. Cada ação tem

um custo associado e cada objetivo tem uma importância, sendo que dependendo da estratégia de cada agente, serão decididos quais os parceiros mais adequados.

A simulação se passa em ciclos, onde cada agente tem a possibilidade de escolher parceiros ordenadamente, sendo que se um agente for escolhido como parceiro por mais de um, terá o direito de escolher com qual deles irá realizar a parceria naquele ciclo.

O software original foi escrito em 1997, utilizando a linguagem C, sem utilizar técnicas de orientação a objetos. Apesar de não ser um programa extremamente complexo (aprox. 3000 linhas de código), e não necessitar de uma programação mais estruturada, a sua capacidade de desenvolvimento posterior ficou muito prejudicada. Além do mais, sua utilização era extremamente dificultada para usuários sem algum conhecimento prévio de programação.

Assim, surgiu a necessidade de se criar uma nova versão da ferramenta, mais fácil de se utilizar e mais flexível. Surgiu então o PartNET+, desenvolvido neste trabalho para estender o primeiro e permitir que diversos tipos de usuários utilizassem a ferramenta para fazer experiências sociais em simulação social baseada em agentes.

As extensões propostas podem ser divididas em duas classes:

1) Extensões **estruturais**, relativas ao projeto e implementação do sistema e à sua interface com os usuários, tais como:

- adoção de uma filosofia de projeto de sistema orientado a objetos, utilizando-se a linguagem UML para descrever suas classes;
- nova implementação na linguagem JAVA, permitindo a execução do sistema também em navegadores, através de applets;
- desenvolvimento de uma interface homem máquina mais amigável, com um sistema de janelas e elementos gráficos;
- gestão da persistência e suporte à interoperabilidade, através da utilização da linguagem XML para armazenamento e recuperação dos resultados de simulação.

2) Extensões **funcionais**, relativas às funções disponíveis no sistema, tais como:

- extensão da definição de parcerias para mais de 2 (dois) agentes, com a utilização da noção de planos;
- possibilidade de detecção de parcerias caracterizadas por relações de dependência circulares;
- adoção de um nível de visibilidade para os agentes da sociedade, possibilitando simular a presença de agentes intermediários (brokers).

As extensões estruturais já se encontram finalizadas, sendo que as figuras 2a a 2e apresentam detalhes da interface principal do simulador, bem como uma janela com os resultados, em forma gráfica, de uma simulação.

Na próxima seção iremos explicar um pouco do histórico e motivação deste projeto, falando do sistema que deu origem ao atual, de seu propósito e operação. Na seção 3, vamos explicar detalhadamente o problema abordado, esclarecendo os algoritmos e o funcionamento do PartNET+, bem como sua interface com o usuário. Então, na seção 4, vamos ver um exemplo de execução do PartNET+, e interpretaremos os resultados. Por final, iremos sumarizar o estado atual do projeto citando os possíveis desenvolvimentos futuros e resultados esperados.

## 2. O PART-NET

O software original – o PART-NET – foi desenvolvido utilizando MABS para mostrar como são formadas parcerias entre tipos heterogêneos de agentes.

Este software buscava comprovar, através de uma técnica de simulação, uma teoria social sobre a importância da existência de agentes com diferentes perfis numa sociedade que contém agentes dotados da capacidade de realizar ações, e que têm uma série de objetivos individuais, e que podem eventualmente estabelecer parcerias para atingir tais objetivos. A intenção era estudar como estas parcerias se estabelecem de acordo com as diferentes distribuições de agentes de três diferentes tipos: substancialistas, instrumentalistas e avaros.

O PART-NET tem como objetivo reforçar a distinção entre duas visões a respeito da racionalidade, a substancialista, onde os objetivos finais tem relevância, e a instrumentalista, muito em voga hoje em dia, onde os objetivos finais não tem importância. Esta segunda se baseia na teoria dos jogos [MYERSON'91] para atingir o ponto de equilíbrio entre ganho e custo. O problema é que esta teoria gera uma incerteza quando há múltiplos pontos de equilíbrio.

O PART-NET lida com parcerias entre agentes heterogêneos em um ambiente comum. Ele lida exclusivamente com dependência recíproca, portanto com relações de troca e não cooperação [SICHTMAN'94].

A arquitetura dos agentes no sistema PART-NET segue os seguintes princípios:

- um conjunto de **objetivos**, definidos pelo seu estado (0 para ativo, 1 para concluído), o seu tipo, e uma prioridade (um valor subjetivo, quanto menor melhor)

- um conjunto de **ações**, constando de seu estado (0 para disponível, e 1 para já utilizada), o tipo do objetivo que ela satisfaz, e o custo (subjetivo) – cada ação atinge um objetivo numa relação um para um.

- o **conhecimento** sobre as relações de dependência existentes, em particular a respeito das dependências bilaterais.

- uma **estratégia** de tomada de decisões, de acordo com o tipo do agente.

Existem 3 tipos básicos de agentes no PART-NET:

- utilitaristas** (ou instrumentalistas), que tentam maximizar a diferença entre o benefício (objetivos atingidos) e os custos;

-**substancialistas**, que não se importam com os custos, mas sim com a ordem de prioridade dos seus objetivos;

-**avaros** que ordenam as alternativas de parceria de forma a minimizar os seus custos.

O conceito de dependência social utilizado considera que um agente A é socialmente dependente de um agente B, quando A não é capaz de realizar ao menos uma ação a ser utilizada para realizar um dos seus objetivos, sendo que B é capaz de tal ação [CASTELFRANCHI'94].

A dependência social pode ser unilateral, quando um agente depende de outros, mas estes outros não dependem dele; ou bilateral, onde a dependência acontece dos dois lados.

No caso de uma dependência bilateral, pode haver uma interação mútua, quando os dois agentes têm o mesmo objetivo, ou recíproca, quando os agentes têm objetivos diferentes. Uma variedade de ações sociais podem ser previstas: exploração, influência, troca e cooperação.

De todos os relacionamentos sociais possíveis, no PART-NET somente são consideradas as relações recíprocas, ou seja, o melhor parceiro para o substancialista irá cumprir o seu objetivo mais valioso a qualquer custo; para o utilitarista, o melhor parceiro irá satisfazer o seu objetivo mais valioso que tenha o menor custo; para os avaros, basta promover o menor custo, qualquer que seja o benefício.

No início da simulação, uma lista de possíveis parceiros (PML) é gerada, e a simulação se dá por sucessivas passadas por esta lista, removendo parcerias bem sucedidas e inválidas.

A cada ciclo da simulação, a PML é percorrida, e os itens onde dois agentes formam uma parceria são removidos e colocados numa outra lista, a EML (lista de parcerias efetivas). A seguir, são removidos da PML todas as ocorrências dos agentes que já realizaram ações, pois no modelo do PART-NET somente é possível realizar uma ação por ciclo. As incompatibilidades restantes (um agente é relacionado com mais de um parceiro) são então removidas (o agente comum irá escolher qual o parceiro), enquanto as outras parcerias serão canceladas.

As parcerias são executadas simultaneamente ciclo a ciclo da simulação, a estrutura dos agentes, caso necessário, é modificada, e os custos e benefícios são computados. A simulação termina quando não houver mais parcerias possíveis.

Em princípio, o esperado é que os substancialistas tenham o maior ganho bruto, enquanto os utilitaristas tenham o maior ganho líquido, e os avaros tenham os menores custos. Em segunda instância, esperamos que os substancialistas se comportem de maneira menos flexível dos que os utilitaristas. Estudos anteriores mostraram que os substancialistas obtêm resultados comparáveis aos dos utilitaristas somente em grandes populações, justamente porque os substancialistas não são indiferentes ao contexto, quanto mais agentes maior a chance de eles encontrarem um parceiro ideal. Entretanto, em estudos anteriores [CONTE'97], as duas estratégias se mostraram igualmente benéficas em ambientes com grandes populações.

Confirmando os estudos anteriores, ao analisar alguns resultados, pode-se observar que os substancialistas se beneficiam mais quando há mais objetivos na

sociedade. Em uma primeira análise, percebe-se que os substancialistas são mais rígidos do que os utilitaristas, no entanto, uma vez que os substancialistas se preocupam com aquilo que recebem sem se preocupar com o que terão de gastar, eles utilizam uma estratégia social mais neutra, menos anti-social. Por outro lado, os utilitaristas se preocupam com o que irão receber e com o que irão gastar, e os avaros se preocupam com o que irão gastar. Dessa forma, os substancialistas serão mais requisitados como parceiros.

O simulador utiliza uma interface gráfica bastante simples, recebendo a entrada de parâmetros da simulação através de arquivos texto. A saída também é feita através de um arquivo texto, como se pode observar nas figuras 2a, 2b.

Concluindo, o simulador PART-NET, mesmo com uma interface simples e de difícil acesso, pôde permitir que as hipóteses apresentadas fossem testadas de uma maneira procedimental, de forma que se tornou possível analisar as modificações na estrutura de cada agente a cada ciclo de simulação. Esse estudo permitiu avaliar quais das estratégias abordadas se mostram mais adequadas em cada situação.

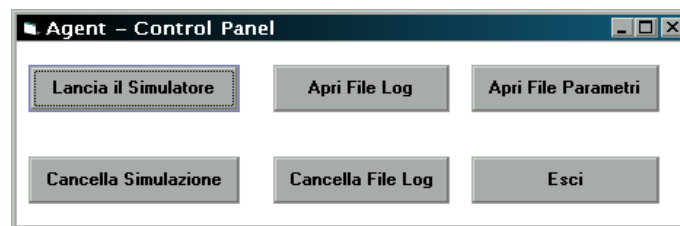


Figura 2a. Exemplo de interface do PART-NET

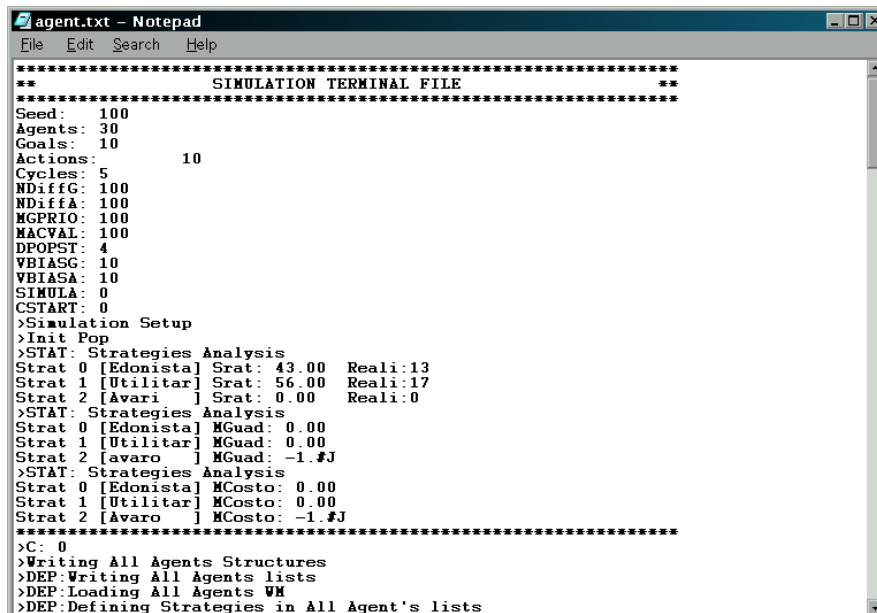


Figura 1b. Saída de dados da simulação

### 3. O PartNET+

Para aprimorar o PART-NET seria então necessário remodelá-lo, utilizando técnicas de programação mais recentes, adequadas ao paradigma de programação atual.

Analisando o código fonte original, foi possível entender e visualizar as etapas da simulação, identificando uma hierarquia de objetos que representariam as diversas entidades presentes na simulação, e elaborar um diagrama de classes no formato UML, possibilitando desenvolver o novo software, utilizando a linguagem Java.

Comparando com seu antecessor, o PartNET+ possui, uma interface gráfica mais requintada e útil e foi desenvolvido utilizando técnicas de projeto e modelagem de software mais atualizadas, de forma a facilitar o desenvolvimento de novas versões.

O algoritmo central consiste em processar uma lista de parcerias possíveis (PML), e selecionar as parcerias mais adequadas para cada agente, de acordo com sua estratégia (utilitarista, substancialista ou avaro). As parcerias selecionadas ficam em uma lista intermediária (SML). Uma vez selecionadas as parcerias desejadas por cada agente, poderá haver conflitos de interesse, onde se formam parcerias recíprocas com mais de um agente. Neste caso, o agente que tem a ação solicitada pelos outros (agindo como detentor de recurso) escolhe a parceria mais conveniente e cancela as outras. Um agente somente pode realizar uma ação por ciclo. A cada ciclo as parcerias realizadas são excluídas da lista principal e colocadas em uma lista de parcerias realizadas (EML). A simulação termina quando não houver mais parcerias a serem realizadas.

No início temos 3 listas PML, EML e SML, sendo que a primeira contém a lista das parcerias possíveis e as outras duas estão vazias. Assim, passo a passo temos:

```
CICLO: (se houver parcerias na PML)
  para cada agente randomicamente:
    lPref = PML.ordenarPorPreferencia(agente)
    para cada parceria em lPref:
      se parceria tem recíproca:
        moverPML_SML(parceria)
        moverPML_SML(recíproca)
  para cada agente randomicamente:
    lOpções = SML.obterParceriasDoAgente(agente)
    lOpções = lOpções.ordenarPorPreferencia(agente)
    lOpções.removePrimeira()
    para cada parceria em lOpções:
      moverSML_PML(parceria)
      moverSML_PML(recíproca)
  para cada parceria em SML:
    realizar(parceria)
    realizar(recíproca)
  computarGanhos()
```

Figura 3a – Algoritmo central do PartNET+

As parcerias têm a forma (Mestre, Ação) -> (Dependente, Objetivo).

Não havendo mais parcerias disponíveis, a simulação é encerrada, e o gráfico correspondente, contendo os resultados é mostrado na tela.

Para avaliar quais parcerias recíprocas são mais adequadas, os agentes utilizam suas estratégias, sendo que para um substancialista, quanto maior o valor subjetivo de um objetivo, mas valiosa é a parceria; para um utilitarista, quanto maior a diferença entre o valor do objetivo e o custo da ação a ser realizada; para um avaro, quanto menor o custo das ações a serem realizadas melhor a parceria, não importando o valor subjetivo do objetivo.

A princípio, o simulador permite a geração de 3 gráficos de saída:

- o gráfico de benefício líquido acumulado por ciclos de simulação;
- o gráfico da evolução da média do benefício líquido entre a população;
- o gráfico do desvio padrão do benefício líquido.

Porém utilizando a mesma biblioteca de geração de gráficos, é possível gerar uma infinidade de gráficos com todos os tipos de estatísticas que possam ser necessárias, bastando uma simples derivação da classe ResultGraph.

No PartNET+, a intenção foi criar uma interface gráfica que pudesse ser utilizada pelo cientista social que não é um programador especializado, ou seja, deve ser uma interface intuitiva e prática, de fácil acessibilidade e manuseio.

Além disto, com uma saída gráfica e visual, é mais fácil interpretar o resultado de uma simulação e tomar decisões a respeito das hipóteses experimentais. Para ver um exemplo da interface, veja a figura 3b.

Name	Goals	Actions	Strategy	Benefit	Cost	Net	Goals ok
a0U	(1, (3), (4))	[2], [3], [4], [0]	Utilitarian	0/219	0	0	0/3
a1U	(4), (3), (2)	[1], [3], [0], [2]	Utilitarian	0/172	0	0	0/3
a2U	(2), (3), (4)	[3], [4], [2], [1]	Utilitarian	0/97	0	0	0/3
a3U	(2), (4), (3)	[4], [1], [0], [2]	Utilitarian	0/166	0	0	0/3
a4U	(2), (3), (1)	[3], [2], [4], [1]	Utilitarian	0/83	0	0	0/3
a5U	(0), (1), (4)	[0], [3], [1], [4]	Utilitarian	0/62	0	0	0/3
a6U	(3), (1), (4)	[2], [1], [0], [4]	Utilitarian	0/205	0	0	0/3
a7U	(3), (0), (4)	[1], [3], [4], [2]	Utilitarian	0/181	0	0	0/3
a8U	(0), (1), (4)	[4], [1], [3], [0]	Utilitarian	0/74	0	0	0/3
a9S	(1), (0), (2)	[3], [4], [1], [0]	Substantialist	0/139	0	0	0/3
a10S	(1), (0), (2)	[4], [0], [3], [2]	Substantialist	0/180	0	0	0/3
a11S	(3), (2), (1)	[0], [4], [3], [2]	Substantialist	0/209	0	0	0/3
a12S	(1), (0), (2)	[3], [0], [4], [2]	Substantialist	0/166	0	0	0/3
a13S	(0), (3), (1)	[3], [1], [0], [2]	Substantialist	0/159	0	0	0/3
a14S	(0), (1), (4)	[1], [3], [2], [4]	Substantialist	0/166	0	0	0/3
a15S	(0), (2), (4)	[2], [1], [3], [4]	Substantialist	0/146	0	0	0/3
a16S	(1), (0), (2)	[1], [3], [4], [2]	Substantialist	0/140	0	0	0/3
a17M	(2), (3), (0)	[0], [2], [4], [3]	Miser	0/70	0	0	0/3
a18M	(1), (0), (3)	[2], [4], [3], [1]	Miser	0/90	0	0	0/3
a19M	(0), (1), (4)	[1], [2], [4], [3]	Miser	0/107	0	0	0/3
a20M	(1), (2), (0)	[4], [2], [3], [1]	Miser	0/171	0	0	0/3
a21M	(2), (0), (4)	[2], [3], [1], [4]	Miser	0/117	0	0	0/3
a22M	(1), (3), (4)	[3], [2], [1], [0]	Miser	0/136	0	0	0/3
a23M	(3), (2), (4)	[2], [0], [1], [3]	Miser	0/175	0	0	0/3
a24M	(2), (0), (3)	[2], [3], [4], [0]	Miser	0/156	0	0	0/3

Figura 3b. Tela principal do PartNET+

A lista de agentes fica exposta o tempo todo, sendo extremamente fácil a alteração dos parâmetros de um agente, bastando clicar duas vezes sobre o seu nome. Vide figura 3c.

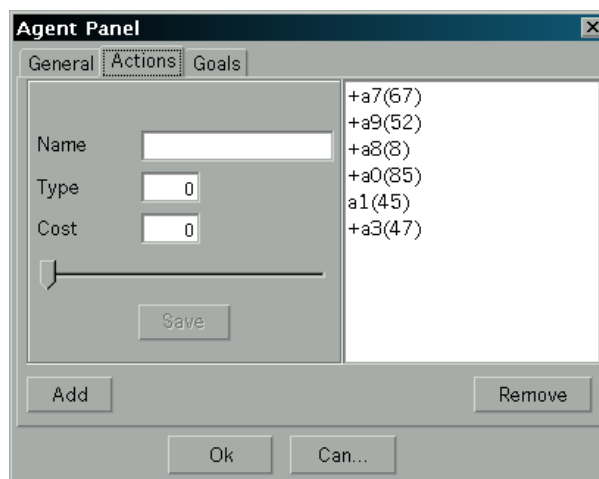


Figura 3c. Interface de cadastramento de agentes

A saída gráfica mostra resultados tanto tabulares quanto gráficos, permitindo obter uma visão detalhada da conclusão da simulação. Vide figura 3d.

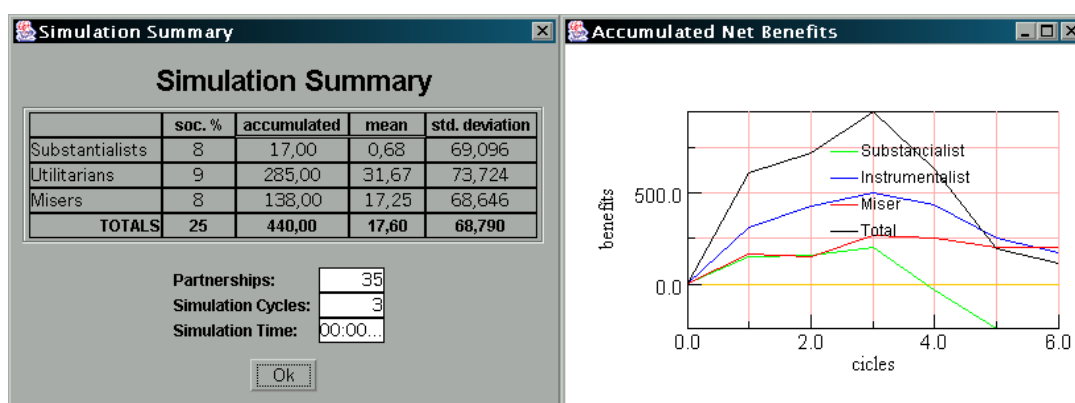


Figura 3d. Resultados da Simulação

Comparando o software original de 1997 e o novo, nota-se que a usabilidade aumentou significativamente. O sistema desenvolvido permite fácil extensão e modificação para acomodar os mais diferentes tipos de simulação.

Apesar de não ser um sistema genérico de simulação, e sim uma versão mais estruturada de um outro sistema de simulação de parcerias, o PartNET+ dá início a uma nova linha de ferramentas de simulação baseadas em sua interface e algoritmo de simulação interno.

## 4. Resultados

Veremos a seguir um exemplo de execução do PartNET+ para um exemplo bastante simples somente para mostrar como o algoritmo se comporta em uma situação fictícia.

Suponhamos que existam definidos em uma determinada sociedade, cinco agentes:

-Jaime, que deseja aprender Java(30), Perl(40) e italiano(20), e sabe ensinar matemática(10) e Prolog(20).

-Júlio, que quer aprender Prolog(30), e sabe ensinar Java(20).

-Rosaria, que deseja aprender matemática(30), e sabe ensinar italiano(2).

-Roberto, que deseja aprender Prolog(7), sabendo ensinar Java(10).

-Fabrício, que ensina Perl(25), e quer aprender matemática(5).

Entre parênteses estão os números que significam a importância dos objetivos ou o custo das ações para cada agente. Vamos supor também que todos estes agentes sejam do tipo Substancialista.

Se a simulação começar pelo Jaime, como sua estratégia é Substancialista, ele irá buscar realizar os seus objetivos mais importantes, sem se preocupar com o custo a pagar para atingí-los.

Assim, sendo substancialista, ele deverá escolher realizar uma parceria com o Fabrício, que ensina Perl cobrando 25, sendo que em troca, ele irá ensinar matemática, que também é um dos objetivos do Fabrício, lembrando que as parcerias precisam ser recíprocas sempre.

Se Jaime fosse Utilitarista, escolheria realizar uma parceria com Roberto, que ensina Java por apenas 10, sendo que a importância deste objetivo de Jaime é 30, portanto há um benefício líquido de 20.

Se Jaime utilizasse a estratégia dos avaros, teria escolhido pagar o mínimo possível, escolhendo aprender italiano com a Rosaria, recebendo um benefício baixo (o valor que ele dá para este objetivo é somente 20), porém pagando somente 2 para Rosaria.

Este exemplo ilustra o funcionamento do algoritmo do PartNET+, e estes resultados podem ser observados durante a execução da simulação. No entanto para obter valores estatísticos interessantes, é necessário utilizar sociedades maiores, onde os diferentes papéis ficam mais claros.

## 5. Conclusões

A arquitetura simplificada de agentes utilizada no PART-NET é bastante interessante para prova de conceito, e mesmo assim, permite chegar a resultados interessantes, porém a ausência de planos no modelo dificulta a expansão da ferramenta e limita o seu uso na grande maioria das simulações interessantes.

Para tornar o modelo mais interessante, o PartNET+ busca adicionar o conceito de planos, ou seja, permitir que para realizar um objetivo, sejam necessárias uma ou mais ações encadeadas, que podem ser fornecidas por vários parceiros simultâneos.

Outra extensão prevista para o simulador envolve a implementação de algoritmos que lidam com redes de dependências como se elas fossem grafos direcionados. Esse tipo de paradigma [SICHMAN'02] permite detectar relações de dependência circulares com um custo computacional muito menos elevado, possibilitando simulações mais interessantes.

Dessa forma, o estudo do PART-NET, como caso particular, nos possibilitou obter experiência específica e detalhada a respeito de um sistema em MABS. Juntamente com o estudo geral e abrangente realizado com diversos sistemas em MABS, foi possível conseguir uma visão bastante privilegiada da área.

## **6. Referências Bibliográficas**

CASTELFRANCHI, Cristiano; SICHMAN, Jaime S.; CONTE, Rosaria and DEMAZEAU, Yves. "A social reasoning mechanism based on dependence networks", in ECAI 94. 11th European Conference on Artificial Intelligence, p. 188-192. John Wiley and Sons, 1994.

PEDONE, Roberto; CONTE, Rosaria. "Finding the Best Partner: The PART-NET System", in: Proc. First International Workshop (MABS'98) Paris, França, Julho 1998, Lecture Notes in Artificial Intelligence, Vol. 1534, p.156-168, Springer-Verlag, 1998.

CONTE, Rosaria. "Diversity in rationality. A multi-agent perspective", in: Dagstuhl Seminar on Social Science Microsimulation, Maio 5-9, 1997

MYERSON, Roger B. "Game Theory: Analysis of Conflict", Harvard University Press, 1991

SICHMAN, Jaime S.; CONTE, Rosaria; DEMAZEAU, Yves and CASTELFRANCHI, Cristiano. "A social reasoning mechanism based on dependence networks" In: Proc. 12th European Conference on Artificial Intelligence (ECAI'94) Amsterdam, The Netherlands, August 1994.

CONTE, Rosaria; SICHMAN, Jaime S. "Dependence Graphs: Dependence Within and Between Groups" In: Computational and Mathematical Organization Theory, 8(2), 87-112, 2002