

Uma Arquitetura de Sistema Multiagente para o TAC-SCM

Robison Cris Brito¹, Cesar Augusto Tacla¹

¹Curso de Pós Graduação em Engenharia Elétrica e Informática Industrial –
Universidade Tecnológica Federal do Paraná (UTFPR)

Av. 7 de Setembro, 3165 – 80230-901 – Curitiba – PR – Brasil

robison@utfpr.edu.br, tacla@cpgei.cefetpr.br

Abstract. *Supplies chains presents interesting problems, due to the complexity that has occurred by elements distribution, absence of global information about all supply chain elements and dynamic feature intrinsic to the chain. In this context, this paper as goals propose and evaluate the performance of the simulation multi agent architecture to be used on TAC-SCM. In the proposed architecture, the agents are organized by groups and contain roles and mission. To perform their missions, the agents compete and/or collaborate respecting the directions of the leader agent group. The evaluation of the architecture related to the amount of the agents of system demonstrates a linear performance in relation to the allocation of memory and the amount of messages exchanged.*

Resumo. *As cadeias de suprimentos apresentam problemas interessantes, devido à complexidade advinda da distribuição dos elementos, à ausência de informações globais em todos os elementos da cadeia e ao caráter dinâmico inerente à rede. Nesse contexto, o objetivo do artigo é propor e avaliar o desempenho de uma arquitetura multiagente para ser utilizada na TAC-SCM. Nessa arquitetura, os agentes são organizados em grupos, possuem papéis e missões. Para executar as missões, competem e/ou colaboram sob a liderança de um dos agentes do grupo. A avaliação da arquitetura em função do número de agentes do sistema mostra um desempenho linear em relação ao consumo de memória e ao número de mensagens trocadas entre os agentes.*

1. Introdução

No mercado globalizado existente hoje, para que uma empresa se torne competitiva, ela deve trabalhar em conjunto com outras empresas do segmento, formando uma cadeia de suprimento que deve ser gerenciada. Em Simchi-Levi, Kaminsky *et al.* (2000), o Gerenciamento de Cadeia de Suprimento é definido como um conjunto de abordagens para integrar eficientemente fornecedores, fábricas, depósitos e armazéns, sendo que as mercadorias são distribuídas na quantidade certa, no local certo e na hora certa, minimizando os custos globais do sistema e satisfazendo os requisitos impostos.

Segundo Kwon, Paul Im *et al.* (2007), existe uma tendência da modelagem de *software* para gerenciamento de cadeias de suprimentos utilizando sistemas baseados em agentes, isso porque existe uma correspondência natural entre os participantes de uma cadeia de suprimentos (fábricas, consumidores, fornecedores, etc.) e agentes em um modelo de simulação. Além disso, cadeias de suprimentos tendem a ser descentralizadas, com seus participantes atuando de maneira independente, agindo de acordo com seus próprios

interesses e políticas, mas compartilhando informações e recursos para alcançar os interesses globais do grupo onde está inserido.

Para incentivar pesquisas na área de agentes autônomos, principalmente no que diz respeito ao gerenciamento de cadeias de suprimentos, foi criada a TAC-SCM (*Trade Agent Competition – Supply Chain Management*), que proporciona um ambiente de competição entre agentes autônomos que representam fabricantes de computadores pessoais. Estes agentes disputam os pedidos dos consumidores e os serviços dos fornecedores levando em conta a capacidade produtiva de suas respectivas fábricas.

Cada competidor desenvolve um agente que deve receber as requisições de compras de computadores vindas dos consumidores, negociar com os fornecedores os componentes para fabricação dos mesmos, planejar e coordenar a produção e estocagem tanto dos computadores prontos quanto dos componentes para a sua fabricação e ser responsável pela entrega dos pedidos feitos pelos consumidores no prazo estipulado.

Embora o objetivo da competição seja estimular a utilização de agentes para gerenciar cadeias de suprimentos, poucos competidores utilizam mais de um agente nessa competição [Keller, Duguay *et al.* 2004; He, Rogers *et al.* 2006], embora o problema seja complexo o suficiente para se empregar um sistema multiagente.

Este artigo apresenta uma arquitetura de SMA colaborativo, onde os agentes distribuídos são organizados em grupos, possuem papéis e missões. Para executar as missões que lhe são designadas, agentes de um grupo competem e/ou colaboram sob a liderança de um dos agentes. O resultado do processamento de um grupo é enviado ao próximo grupo, formando um *workflow* para cada tarefa da cadeia de suprimentos.

O artigo está dividido da seguinte maneira: Na seção 2, é apresentada uma visão geral sobre a TAC-SCM. A seção 3 apresenta alguns dos agentes que se destacaram nas competições realizadas entre 2003 e 2007. A seção 4 traz a arquitetura multiagente proposta para o gerenciamento de cadeias de suprimento. A seção 5 apresenta um estudo de caso, utilizando a TAC-SCM. A seção 6 finaliza o trabalho com as conclusões e sugestões de trabalhos futuros.

2. TAC-SCM

A competição consiste na disputa entre seis agentes. Cada agente gerencia uma fábrica de computadores tendo por objetivo obter mais lucro que seus concorrentes. Na TAC-SCM, o agente interage com o simulador e, para bater seus concorrentes, procura vender computadores com menores preços e dentro do prazo de entrega, buscando fazer sua fábrica trabalhar no limite da produção máxima diária [Collins, 2006]. Nesse ambiente, o simulador desempenha os seguintes papéis:

Consumidor: faz pedidos de cotação (RFQ – *Request for Quotation*) para compra de computadores aos diferentes agentes e seleciona aquele que oferecer o melhor preço e atender seu pedido dentro do prazo.

Fornecedor: para cada pedido de cotação de componentes recebida de um agente, retorna uma oferta. O valor do componente não é constante, depende da demanda, tempo decorrido de competição e da fidelidade do agente ao fornecedor.

Fábrica: recebe um agendamento para produção de computadores do agente, se existir os componentes do computador em estoque, o computador é fabricado e armazenado.

Estoque: tem capacidade limitada de armazenamento e possui a função de armazenar componentes e computadores prontos. Para que um componente ou um computador fique no estoque, é cobrada uma taxa.

Banco: cada agente possui uma conta bancária, inicialmente zerada e que permite valores negativos. O pagamento das vendas realizadas é adicionado à conta bancária. Se o saldo bancário for negativo, são cobrados juros diários sobre o valor negativo, caso contrário, funciona como uma poupança, acrescentando juros diariamente.

3. Trabalhos Relacionados

Com o objetivo de estudar a estrutura interna dos agentes que participaram da competição e, mais precisamente, de levantar as tarefas e subtarefas realizadas pelos agentes, foi realizado um estudo sobre os agentes que mais se destacaram nas últimas competições. Dois dos competidores apresentam um sistema multiagente, são eles o *RedAgent* [Keller, Duguay *et al.* 2004], campeão da primeira competição da TAC-SCM em 2003, e o *Southampton* [He, Rogers *et al.* 2006], segundo lugar na competição de 2005. Outros três agentes que utilizam módulos com funções bem definidas e também se destacaram na competição são: *TacTex* [Pardoe e Stone 2006], campeão das competições de 2005 e 2006 e segundo lugar na competição de 2007; *CMieux* [Benisch, Sardinha *et al.*, 2006] e *Deep Maize* [Kiekintveld, Miller *et al.*, 2006], que estiveram entre os cinco melhores agentes na competição de 2005.

Com base nas informações coletadas dos agentes, foram identificadas as tarefas e subtarefas mais comuns detalhados na seqüência.

Oferta/Venda de Computadores: calcula um preço ideal de venda, que pode variar de acordo com a simulação. Esse preço permite a maximização da produção de computadores. Também identifica quantos dos pedidos de cotação irão se transformar em ordem de compra, evitando vender mais computadores que a fábrica possa fabricar.

Aquisição de Componentes: calcula o preço ideal para compra de componentes. Quando o preço oferecido pelo fornecedor é significativamente inferior ao preço ideal, compra componentes para estocagem. Também procura manter uma boa reputação junto aos fornecedores e atribui uma reputação a eles.

Planejamento de Produção: planifica a fabricação de computadores eliminando períodos de ociosidades para adiantar a fabricação de computadores que poderão ser vendidos a curto prazo. Também calcula quantos computadores de cada tipo podem ser fabricados a cada dia.

Estratégia: verifica qual o segmento de mercado mais lucrativo, focando assim as vendas. Essa informação é importante para as demais funcionalidades. A identificação de um nível ideal para o estoque também é o objetivo dessa tarefa.

Previsão: fornece informações para praticamente todas as outras funcionalidades realizando previsões de dados futuros, a curto e a longo prazo, mesmo com a variação estocásticas de algumas variáveis do ambiente. Para realizar previsões, alguns agentes utilizam a média dos dados dos dias anteriores, já outros utilizam algoritmos de aprendizagem de máquina, utilizando como fonte de dados os dados da própria partida ou históricos de partidas anteriores. Alguns agentes, entretanto, utilizam uma abordagem híbrida.

Todos os agentes estudados realizam as tarefas citadas, os agentes que não realizam explicitamente estas tarefas em um agente ou módulo, as possuem fragmentadas em sua arquitetura. A tabela 1 apresenta as tarefas/subtarefas realizadas pelos agentes.

Tabela 1. Tabela de recursos utilizados pelos agentes na competição TAC.

Características	RedAgent	Southampton	CMieux	Deep Maize	Tac Tex
Oferta de Computadores	X	X	X	X	X
Identificação do melhor preço	X	X	X	X	X
Estima quantas ofertas serão aceitas pelo cliente		X		X	X
Aquisição de components	X	X	X	X	X
Identificação do melhor preço	X	X	X	X	X
Planejamento de produção	X	X	X	X	X
Agenda a produção para vários dias futuros		X	X	X	X
Identifica quantos computadores de cada tipo podem ser confeccionados	X		X	X	X
Estratégia	X	X	X	X	X
Identifica qual o melhor segmento de mercado	X	X	X	X	X
Identifica qual o nível ideal de componentes em estoque	X	X	X	X	X
Previsão	X	X	X	X	X
Utiliza Algoritmo de Aprendizagem de máquina			X	X	X
Utiliza média dos últimos dias	X	X		X	X
Faz previsão de demanda		X	X	X	X
Faz previsão de preço do componente		X	X	X	X

4. Arquitetura proposta

Para a arquitetura proposta, foram levadas em consideração as cinco tarefas identificadas na seção anterior. A única mudança foi a fusão das tarefas Estratégia e Previsão, já que estão intimamente ligadas e compartilham das mesmas informações. As tarefas são realizadas pelos grupos de agentes: Oferta de Computadores; Aquisição de Componentes; Planejamento de Produção e; Estratégia/Previsão. Cada grupo conhece e troca mensagens com os outros grupos.

Em Moyaux, Chaib-draa *et al.* (2006) é dito que a distribuição de uma atividade em vários agentes (na TAC-SCM pode-se citar como exemplo de atividade a venda de computadores ou a compra de componentes) pode forçar os programadores a proporem novos algoritmos para resolver o problema. Em particular, a concorrência pode ser usada para acelerar a resolução de problemas complexos. Moyaux, Chaib-draa *et al.* (2006) conclui seu raciocínio dizendo que soluções centralizadas podem ser impossíveis na realidade, por que sistemas e dados podem se encontrar em organizações independentes.

Dessa maneira, para a arquitetura proposta, foi projetado um ambiente onde vários agentes que possuem as mesmas habilidades são agrupados, colaborando ou competindo na realização das tarefas. Como cada agente trata o ambiente de uma maneira, por

utilizar lógicas e fontes de informações distintas, é possível um compartilhamento de diferentes pontos de vistas sobre o mesmo problema.

No caso de competição para realizar uma tarefa, todos os agentes com habilidades semelhantes recebem as mesmas informações, processam, enviando os dados processados para um agente, denominado líder, que por sua vez possui uma função de utilidade para escolher a melhor resposta. Em caso de colaboração, um agente líder pode desmembrar uma tarefa complexa em subtarefas e enviar para os agentes que possuam as habilidades mínimas para realizá-las.

A Figura 1 exibe o fluxo de mensagens entre os participantes de um grupo utilizando competição – processamento concorrente. Têm-se duas Trocas de Mensagens Entre Grupos (TMEG): Msg0Multicast e Msg4Multicast, e $N - 1$ mensagens resposta enviadas ao líder, sendo uma mensagem para cada agente do grupo com exceção do líder que não precisa enviar mensagem para si mesmo.

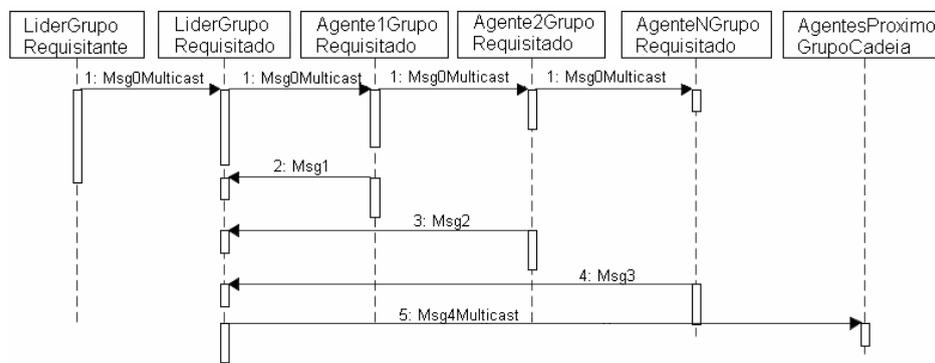


Figura 1: Diagrama de Seqüência 1 – Competição.

O número de mensagens trafegadas para o processamento concorrente, é dado pela equação 1.

$$\text{numeroMensagens} = (\text{TMEG}-1) * (1+N) - (\text{TMEG}-2) \quad (1)$$

Para troca de mensagens com colaboração – Figura 2, tem-se duas Trocas de Mensagens Entre Grupos (Msg0 e Msg7) e duas situações onde são enviadas $N - 1$ mensagens.

Assim, o número de mensagens trafegadas para o processamento colaborativo é dado pela equação 2.

$$\text{numeroMensagens} = (\text{TMEG}-1) * (2N) - (\text{TMEG}-2) \quad (2)$$

Dentro de um grupo, todos os agentes possuem uma função de utilidade que os tornam aptos a serem líderes do grupo. A função de utilidade, herdadas por todos os agentes, possibilita ao líder escolher uma proposta para representar o grupo no caso em que os agentes competem para realizar a mesma tarefa. O programador pode sobrescrever essa função de utilidade, implementando técnicas para decomposição de tarefas ou seleção de respostas.

Cada grupo permite apenas um líder. A troca de líder pode ser realizada via interface gráfica do usuário, ou ainda pode acontecer automaticamente, pelo sistema. A eleição do líder de grupo pode ser feita por meio de algoritmos específicos [Fischer e Jiang 2006].

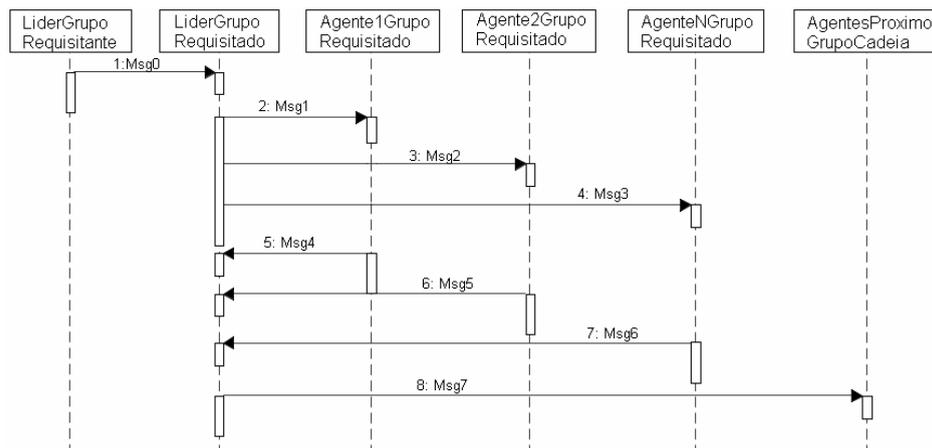


Figura 2: Diagrama de Seqüência 2 – Colaboração.

Para a arquitetura proposta, foi utilizada uma abordagem centrada na organização de agentes, sendo definidos declarativamente os aspectos estruturais (grupos, subgrupos, papéis, relações entre agentes e cardinalidades), funcionais (definição dos esquemas, metas e missões) e deônticos da organização (atribui papéis responsáveis para executar os planos).

Resumidamente, cada esquema, também conhecido como meta global, é formado por um conjunto de metas, que por sua vez são agrupadas em missões. A responsabilidade de executar uma missão é atribuída a um papel ou vários papéis (agentes). A definição declarativa foi elaborada utilizando os elementos do modelo organizacional MOISE+ [Hübner 2003].

O modelo organizacional MOISE+ foi escolhido por estar integrado à ferramenta SACI - *Simple Agent Communication Infrastructure* [Hübner e Sichman 2000], que permite aos agentes se comunicarem por troca de mensagens utilizando ACL (*Agent Communication Language*). O MOISE+ possibilita também o desenvolvimento de agentes cognitivos, utilizando BDI - *Believe, Desire, Intention* - [Rao e Georgeff, 1995].

5. Estudo de caso

Para teste do SMA proposto, foi utilizado a TAC-SCM. Para evitar ambigüidade na nomenclatura, desse ponto em diante serão chamados de competidores os agentes que competem na TAC-SCM, e serão chamados de agentes as entidades de *software* desenvolvidas para a arquitetura multiagente proposta (ex. Agente Vendedor, Agente VendedorLider, Agente Comprador).

Como existe a limitação técnica do simulador aceitar apenas uma conexão por competidor durante todo o período de simulação, foi necessário o desenvolvimento de uma camada intermediária, denominada *proxy*, que recebe as informações vindas do servidor e encaminha para o agente interessado. Com a utilização do *proxy* foi possível a vários agentes localizados em computadores distintos acessarem simultaneamente as informações do servidor TAC-SCM. Os *proxies* não fazem parte da arquitetura, apenas surgiram para resolver uma limitação técnica do simulador TAC-SCM.

Logo, foi desenvolvido um agente chamado AgentProxyTacScm, que troca informações com o simulador TAC-SCM. Para cada tarefa solicitada ou requerida pelo servidor

TAC-SCM, foi desenvolvido um agente *proxy*, que assume a responsabilidade da tarefa específica. A Figura 3 apresenta os recursos utilizados para testar a arquitetura multiagente com a TAC-SCM. A arquitetura proposta é delimitada pelo retângulo pontilhado, o qual é composto pelos grupos (retângulos brancos), um agente líder em cada grupo (retângulo escuro) e zero ou mais agentes jogadores (retângulos claros).

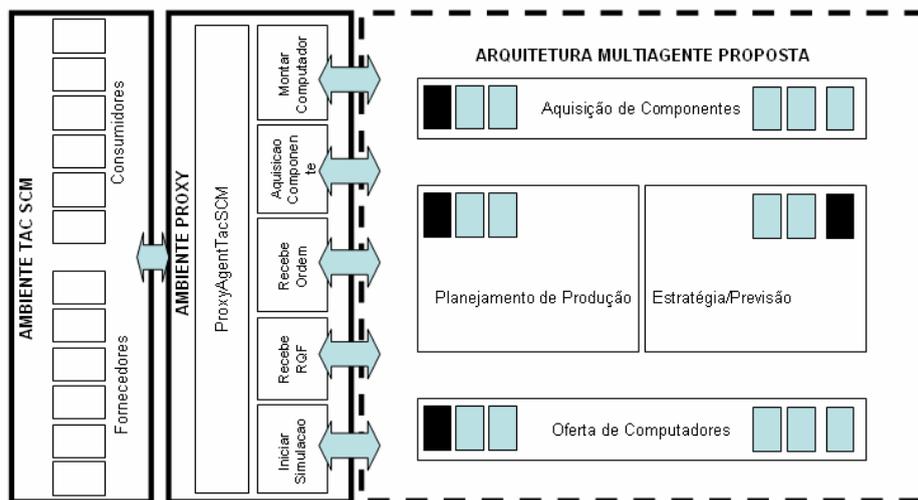


Figura 3: Arquitetura proposta com os *proxies* que fazem a integração com a TAC-SCM.

Como o simulador TAC-SCM é estático, já que não são incluídos novos recursos durante a simulação, foi possível desenvolver uma lista de esquemas diários que os agentes devem executar, bem como um fluxo das informações. Cada esquema é iniciado pelo agente *proxy* correspondente. Os esquemas são:

esquemaIniciarSimulacao: Executado uma vez no início do programa pelo proxyIniciarSimulacao, tem a função de enviar as informações iniciais do simulador a todos os agentes.

esquemaOrcarComputadores: Executado diariamente. O proxyRecebeRQF recebe os pedidos de cotação de computadores, envia-os para os agentes do grupo Estratégia/Previsão que com base nos pedidos realizam uma estratégia de venda e enviam essa informação para o grupo Oferta de Computadores que por sua vez é responsável pela realização dos orçamentos.

esquemaReceberOrdemCotarComponente: Diariamente, o proxyRecebeOrdem recebe as ordens de compras e envia-as ao grupo Planejamento de Produção que define uma agenda de produção. Na sequência, essas informações são enviadas para o grupo Estratégia/Previsão, sendo utilizadas para definir previsões e estratégias futuras, após as informações são enviadas ao grupo Aquisição de Componentes, que realiza o pedido de orçamento de novos componentes.

esquemaCompraComponentes: Ao final de cada dia, o proxyAquisicaoComponente reúne as ofertas de componentes e envia ao grupo Aquisição de Computadores, que seleciona as melhores ofertas e os melhores fornecedores para realizar da compra. Essa informação também é enviada ao grupo Estratégia/Previsão, e serão úteis para decisões futuras.

esquemaMontarComputadores: No final de cada dia, o proxyMontarComputadores solicita ao grupo Planejamento de Produção as agendas de fabricação, de entrega e reserva de computadores para o dia seguinte, sendo enviando-as também ao grupo Estratégia/Previsão.

Os esquemas apresentados representam apenas o envio de informações entre os grupos, entretanto, entende-se que entre a chegada e a saída da mensagem no grupo, acontece o processo de competição, apresentado na Figura 1.

Foram realizados vários testes para testar a escalabilidade do sistema, sendo testada a arquitetura multiagente com diferentes números de indivíduos em cada grupo. Para os testes, foi utilizado um computador *centrino* com um processador de 1.73 GHz, com 1 GB de memória RAM e Sistema Operacional Windows XP. Foram realizados testes com 1, 2, 5 e 10 agentes em cada grupo. Ao tentar executar o sistema com 20 agentes em cada grupo, não foi possível executar o teste devido ao excesso de processos (80 novos processos) evidenciando a necessidade de distribuí-los em diversas máquinas.

Para os testes de memória foram realizados quatro experimentos, uma para cada número de agentes nos grupos. A cada experimento, foi recuperada a memória utilizada pelos agentes em cada dia de simulação na TAC-SCM. O resultado do consumo de memória médio dos agentes pode ser visto no gráfico da Figura 4. É possível concluir que mesmo com a mudança do número de agentes no grupo, o consumo de memória médio dos agentes não altera muito.

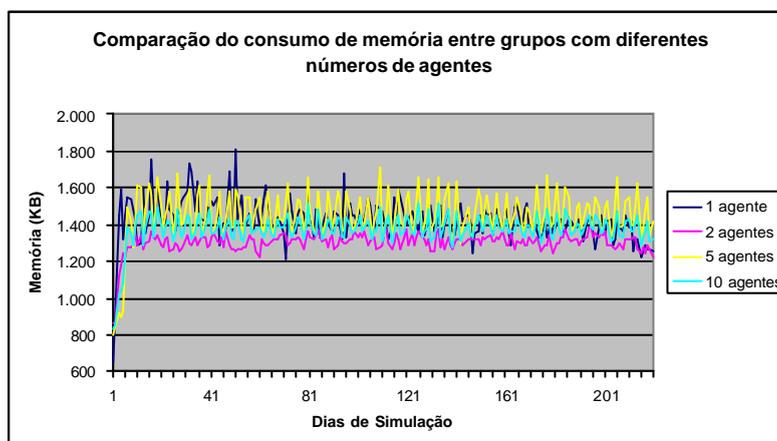


Figura 4: Gráfico comparativo do consumo de memória por agente em grupos com diferentes números de agentes.

Para testar o desempenho quanto à troca de mensagens, foi utilizado como exemplo o esquemaOrcarComputares, onde são feitas três trocas de mensagens entre grupos (proxyRecebeRQF – Estratégia/Previsão – Aquisição de Componentes), em uma situação onde existem 5 agentes em cada grupo que competem pelo melhor resultado, tem-se um total de 18 mensagens trocadas (parâmetros aplicados na equação 1). Com 10 agentes em cada grupo, o número de mensagens é 33. Já quando se tem colaboração, com 5 agentes em cada grupo tem-se 30 mensagens (parâmetros aplicados na equação 2), com 10 agentes, 60 mensagens.

É possível observar uma relação de linearidade entre o número de mensagens e o número de agentes, não havendo um crescimento exponencial, o que resultaria em problemas de escalabilidade do sistema. O gráfico do Número de Agentes x Número de

Mensagens, tanto no processo de competição quando cooperação pode ser visto na Figura 5.

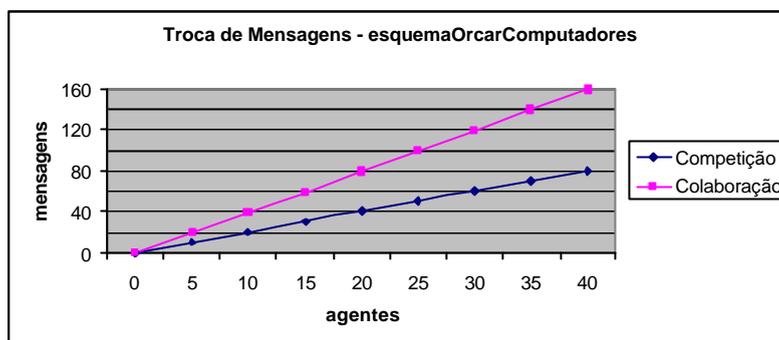


Figura 5: Relação entre o número de agentes e as mensagens trocadas no esquemaOrcarComputadores.

6. Conclusões

Foi considerada satisfatória a utilização de vários agentes dentro de um grupo, já que o problema da TAC-SCM é complexo o suficiente para suportar uma solução multiagente. Das arquiteturas apresentadas no capítulo 3, apenas o *RedAgent* e o *Southampton* utilizavam multiagente, e mesmo assim, a responsabilidade da execução de uma atividade era centralizada em um único elemento.

Na arquitetura proposta, o sistema conta com um ou mais agentes com habilidades específicas, que colaboraram ou competem para a resolução de um problema. Como no estudo de caso foi utilizada a competição entre agentes de um grupo para solucionar um problema, os mesmos trabalharam de forma concorrente, processando as mesmas informações e enviando seus resultados ao agente líder, que pôde contar com uma quantidade maior de opções para escolher a melhor resposta de acordo com uma função de utilidade. Outra vantagem da arquitetura proposta é a flexibilidade, podendo haver em tempo de execução a eliminação dos piores agentes e a troca do agente líder.

O modelo MOISE+ e a ferramenta SACI suprimiram as necessidades da arquitetura multiagente no que diz respeito ao modelo organizacional e a troca de mensagens na rede. Para o gerenciamento dinâmico dos agentes da arquitetura multiagente proposta, foi desenvolvida uma interface gráfica que possibilita as ações de inclusão e exclusão de agentes e a alteração do agente líder em tempo de execução.

A utilização da memória na arquitetura proposta foi estável durante todo o período de simulação, inclusive com o aumento do número de agentes. Quanto à troca de mensagens, as mesmas dependem exclusivamente do número de agentes na arquitetura e não aumentam de forma exponencial o que não impede a escalabilidade da arquitetura.

Como trabalhos futuros, sugere-se desenvolver mecanismos de replicação de agentes, dessa maneira, quando se verifica que um agente saiu da sociedade de forma inesperada, sua réplica que possui as informações copiadas assume seu lugar. Outra atividade é a descentralização do modelo MOISE+, uma vez que a arquitetura multiagente proposta trabalha de forma distribuída e o único recurso centralizado é o MOISE+, que possui a função de gerenciar os recursos estruturais, funcionais e deontológicos da organização. Também é interessante estudar algoritmos para eleição e avaliação dos líderes, assim

como os demais membros de um grupo, eliminando aqueles que produzam resultados ruins.

Referências

- Benisch, M. Sardinha, A. Andrews, J. Sadeh, N. (2006) "CMieux: Adaptive Strategies for Competitive Supply Chain Trading", In: Eighth International Conference on Electronic Commerce, USA, p. 1-10.
- Collins, J. Arunachalam, R. Sadeh, N. Eriksson, J. Finne, N. Janson, S. (2006) "The supply chain management game for the 2007 trading agent competition". Pittsburgh: School of Computer Science Carnegie Mellon University.
- Fischer, M. Jiang, H. (2006) "Self-stabilizing Leader Election in Networks of Finite-state Anonymous Agents", In: Tenth International Conference on Principles of Distributed Systems, Lecture Notes in Computer Science, Springer-Verlag, p. 395-409.
- He, M. Rogers, A. Luo, X. Jennings, N. (2006) "Designing a Successful Trading Agent for supply chain Management", In: Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, Japan, p. 1159-1166.
- Hübner, J. S. (2003) "Um Modelo de Reorganização de Sistemas Multiagente". Tese de Doutorado. Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3141/tde-17052004-151854>. Acessado em: 09/04/2008.
- Hübner, J. F. Sichman, J. S. (2000) "SACI: Uma Ferramenta para Implementação e Monitoração da Comunicação entre Agentes", In: IBERAMIA-SBIA 2000, Brazil, p. 47-56.
- Keller, P. Duguay, F. Procup, D. (2004) "RedAgent 2003: An Autonomous, Market-Based supply chain Management Agent", In: Third International Joint Conference on Autonomous Agents and Multi-Agent Systems, USA, p. 1182-1189.
- Kiekintveld, C. Miller, J. Patrick, J. Wellman, M. (2006) "Controlling a supply chain Agent Using Value-Based Decomposition". Seventh ACM Conference on Electronic Commerce, USA, p. 208-217.
- Kwon, O. Paul Im, G. Lee, K. C. (2007) "MACE-SCM: A multi-agent and case-based reasoning collaboration mechanism for supply chain management under supply and demand uncertainties", In: Expert Systems with Applications: An International Journal, V. 33, Número 3, p. 690-705.
- Moyaux, T. Chaib-draa, B. D'Amours, S. (2006) "Supply Chain Management and Multiagent Systems: An Overview". Book: Multiagent based Supply Chain Management, Springer Berlin Heidelberg, p. 1-27.
- Pardoe, D. Stone, P. (2006) "TacTex 05: A Champion supply chain Management Agent", In: Fifteenth Twenty-First National Conference on Artificial Intelligence, USA.
- RAO, A. GEORGEFF, M. (1995) "BDI agents: from theory to practice", In: First International Conference on MultiAgent Systems (ICMAS'95), p. 312-319.
- Simchi-Levi, D. Kaminsky, P. Simchi-Levi, E. (2000) "Designing and managing the supply chain". McGraw-Hill Higher Education, pages 321.