

Universidade Federal do Espírito Santo - UFES
Programa de Pós Graduação em Engenharia Elétrica – PPGEE
Doutorado em Engenharia Elétrica

**UMA ABORDAGEM BASEADA EM NEGOCIAÇÃO DE AGENTES
PARA A RESOLUÇÃO DO PROBLEMA DE ALOCAÇÃO DINÂMICA
DE NAVIOS EM BERÇOS DE TERMINAIS PORTUÁRIOS**

Rodrigo de Alvarenga Rosa

Vitória
2006

Rodrigo de Alvarenga Rosa

**UMA ABORDAGEM BASEADA EM NEGOCIAÇÃO DE AGENTES
PARA A RESOLUÇÃO DO PROBLEMA DE ALOCAÇÃO DINÂMICA
DE NAVIOS EM BERÇOS DE TERMINAIS PORTUÁRIOS**

Tese apresentada ao Programa de Pós-Graduação em Engenharia
Elétrica do Centro Tecnológico da Universidade
Federal do Espírito Santo, como requisito parcial
para obtenção do Grau de Doutor em Engenharia Elétrica,
na área de concentração em Inteligência Artificial.
Orientador: Hans Jorg Andreas Schneebeli, Dr.rer.nat.

Vitória
2006

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

R788a Rosa, Rodrigo de Alvarenga, 1966-
Uma abordagem baseada em negociação de agentes para a resolução do problema de alocação dinâmica de navios em berços de terminais portuários / Rodrigo de Alvarenga Rosa. – 2006.
104 f. : il.

Orientador: Hans Jorg Andreas Schneebeli.
Tese (doutorado) – Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Portos - Planejamento. 2. Transportes. 3. Agentes inteligentes (Software). 4. Inteligência artificial distribuída. 5. Logística. 6. Otimização estrutural. I. Schneebeli, Hans Jorg Andreas. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Título.

CDU: 621.3

Rodrigo de Alvarenga Rosa

**UMA ABORDAGEM BASEADA EM NEGOCIAÇÃO DE AGENTES
PARA A RESOLUÇÃO DO PROBLEMA DE ALOCAÇÃO DINÂMICA
DE NAVIOS EM BERÇOS DE TERMINAIS PORTUÁRIOS**

Tese submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisição parcial para a obtenção do Grau de Doutor em Engenharia Elétrica – Automação.

Aprovada em 01 de junho de 2006

COMISSÃO EXAMINADORA

Prof. Hans Jörg Andreas Schneebeli, Dr.rer.nat.
Universidade Federal do Espírito Santo (UFES)
Orientador

Prof. Jaime Simão Sichman, Doctorat en Génie Informatique
Escola Politécnica da Universidade de São Paulo (EPUSP)

Prof. Arlindo Gomes de Alvarenga, Doutor
Universidade Federal do Espírito Santo (UFES)

Prof. Flávio Miguel Varejão, Doutor
Universidade Federal do Espírito Santo (UFES)

Prof. Saulo Bortolon, Doutor
Universidade Federal do Espírito Santo (UFES)

Ao meu filho Rafael

Aos meus pais Gastão e Léa

Agradecimentos

A Deus, porque sem Ele nada existe e nada se realiza. Tenho enorme gratidão a Ele por esta oportunidade de estar mais uma vez vivendo aqui e podendo evoluir para esta e outras vidas melhores.

Um agradecimento especial aos meus pais, Gastão e Léa, que sempre me incentivaram para continuar a estudar e crescer pessoalmente.

O meu enorme agradecimento ao Prof. Hans, meu orientador, que com competência, compreensão, apoio e amizade tornou possível fazer este projeto de doutorado se tornar realidade. Hoje mais que orientador tenho um amigo que foi de suma importância para que eu tivesse força e ânimo para chegar ao fim deste projeto. Não é possível deixar de agradecer, também, ao Prof. Hans pelas suas idéias e pelo conhecimento técnico, que ele sempre fez questão de repassar e difundir.

Quero agradecer aos amigos do LAI que tornaram o árduo trabalho de desenvolver esta tese um pouco mais divertido com as brincadeiras e conversas e que também se dispuseram a ajudar a resolver os problemas técnicos, formando realmente uma ótima equipe de trabalho.

Agradeço ao Prof. Dr. Mario Sarcinelli pelo seu empenho em tornar o PPGEE uma realidade e sempre lutar por melhorar as condições de estudo e pesquisa de todos os alunos e professores do programa, bem como disponibilizar recursos para que todos pudessem participar dos congressos e viagens de estudo.

Por fim, quero agradecer a todos os amigos que, direta ou indiretamente, colaboraram para que esta tese chegasse ao fim.

Resumo

Esta tese propõe uma nova abordagem para solução de problemas logísticos dinâmicos e complexos. Esta nova abordagem é baseada na constatação do fato de que na sociedade humana muitos dos problemas são resolvidos por meio da interação e negociação entre diversos seres humanos, os quais sozinhos não poderiam resolver o problema e, no entanto, no momento que se unem e trabalham em conjunto, cada um com seus recursos e capacidades, eles conseguem resolvê-los. É proposta e definida uma abordagem inspirada na sociedade humana com diversos padrões de negociação. Fez-se, também, a opção por propor um padrão de negociação inspirado na emoção humana gratidão que tem como sua principal função ser um elemento para resolver eventuais casos de impasse/conflito. Com base nesta abordagem proposta, foi criado um sistema computacional que cria uma sociedade virtual de indivíduos que possuem recursos não compartilhados e, por meio de mensagens trocadas entre si, negociam até convergir para agendar as solicitações de operação recebidas (navio para carregar e descarregar contêiner). Este sistema foi desenvolvido com base na tecnologia de agentes e sistema multiagentes. Como exemplo prático, a abordagem proposta e o sistema elaborado foram usados para a solução do problema de gerenciamento da fila de navios de um Terminal Portuário de Contêiner. Vários testes foram realizados e os resultados foram apresentados e analisados para cada um dos padrões de negociação.

Abstract

This work shows a new approach to solve dynamic and complex logistics problems based upon the fact that in human society several problems are solved by interaction and negotiation between humans, that alone can't solve the problem, but if they join themselves toward the solution of the problem, all of them with their own resources and capabilities, they can reach one solution. It is proposed and defined one methodology inspired by the human society way of solving problems using several negotiation protocols. It was also proposed a negotiation protocol inspired in human emotion gratitude that has the main goal to solve the conflicts that might occur. Based on this proposed methodology, it was developed a computational system to create a virtual society of agents that have their own resources, non shareable, and by the exchange of messages between the agents, they converge to schedule all the solicitations of operations received (ship to load or unload container). This system was developed based on the agent technology and multiagent system. The proposed methodology and the implemented system were applied to a practical and real application of a Maritime Container Terminal. Several tests were done and the results was shown and analyzed for each negotiation protocol.

Sumário

CAPÍTULO 1 - INTRODUÇÃO.....	1
1.1) MOTIVAÇÃO.....	1
1.2) DESCRIÇÃO DA ÁREA.....	2
1.3) O PROBLEMA DE ELABORAÇÃO DA AGENDA.....	4
1.4) HIPÓTESE.....	5
1.5) METODOLOGIA.....	6
1.6) CONTRIBUIÇÃO CIENTÍFICA.....	7
1.7) ESTRUTURA DO TRABALHO.....	8
CAPÍTULO 2 - REVISÃO DO ESTADO DA ARTE DE NEGOCIAÇÃO DE AGENTES E EMOÇÕES EM AGENTES.....	9
2.1) SOCIOLOGIA E SOCIEDADE HUMANA.....	10
2.2) NEGOCIAÇÃO.....	12
2.3) EMOÇÃO.....	14
2.4) AGENTES E SISTEMAS MULTIAGENTES.....	15
2.5) PROBLEMA DE ESCALONAMENTO DINÂMICO COM RESTRIÇÃO DE RECURSOS.....	22
CAPÍTULO 3 - APRESENTAÇÃO DO PROBLEMA DE ESCALONAMENTO E DAS SOLUÇÕES JÁ EXISTENTES.....	27
CAPÍTULO 4 - UMA ABORDAGEM BASEADA EM NEGOCIAÇÃO DE AGENTES.....	32
4.1) DEFINIÇÃO DA ABORDAGEM PROPOSTA.....	32
4.2) FUNCIONAMENTO DA ABORDAGEM PROPOSTA.....	34
4.3) MODELAGEM MATEMÁTICA DA ABORDAGEM PROPOSTA.....	41
4.4) MODELAGEM MATEMÁTICA DO PADRÃO DE NEGOCIAÇÃO DE ATENDIMENTO POR ORDEM DA DATA DA SOLICITAÇÃO.....	47
4.5) MODELAGEM DO PADRÃO DE NEGOCIAÇÃO INSPIRADO NA EMOÇÃO GRATIDÃO.....	50
4.6) SISTEMA COMPUTACIONAL PARA A ABORDAGEM PROPOSTA.....	52
4.7) ESTUDO DE CASO: TERMINAL MARÍTIMO DE CONTÊINER.....	56
CAPÍTULO 5 - OS EXPERIMENTOS REALIZADOS.....	58
5.1) VALIDAÇÃO DO SISTEMA COMPUTACIONAL.....	58
5.2) VALIDAÇÃO DO FUNCIONAMENTO DO SISTEMA MULTIAGENTES ATRAVÉS DO ESTUDO DE CASO DO TERMINAL MARÍTIMO DE CONTÊINER.....	67
CAPÍTULO 6 - CONCLUSÕES.....	85
6.1) TRABALHOS FUTUROS.....	86
REFERÊNCIAS.....	87
ANEXO A - SISTEMA COMPUTACIONAL PARA A ABORDAGEM PROPOSTA.....	95

Lista de Figuras

Figura 1 - Ser Humano.....	10
Figura 2 - Modelo de Inquietações Duais.....	13
Figura 3 - Taxonomia de Coordenação.....	20
Figura 4 - Cenário 1 - Situação 1: Confirmação de todos os solicitados.....	36
Figura 5 - Cenário 2 - Situação 2: Confirmação de todos os solicitados.....	37
Figura 6 - Cenário 3 - Situação 1: Retorno negação de pelo menos um solicitado.....	38
Figura 7 - Cenário 4 - Situação 2: Retorno negação pelo solicitador.....	39
Figura 8 - Cenário 5 - Situação 2: Retorno negação de pelo menos um solicitado.....	40
Figura 9 - Funcionamento esquemático do Grau de Gratidão.....	41
Figura 10 - Situação 1	43
Figura 11 - Situação 2	44
Figura 12 - Situação 3	44
Figura 13 - Situação 4	45
Figura 14 - Regra de Atendimento 1	47
Figura 15 - Regra de Atendimento 2	47
Figura 16 - Regra de Atendimento 3	48
Figura 17- Regra de Atendimento 4	49
Figura 18 - Arquitetura Proposta do Sistema Multiagentes.....	53
Figura 19 - Plano DECAF parcial para tratar o recebimento da mensagem de Solicitação	55
Figura 20 - Exemplo da primeira situação de teste.....	59
Figura 21 - Exemplo da segunda situação de teste.....	59
Figura 22- Exemplo da terceira situação de teste.....	60
Figura 23 - Exemplo da quarta situação de teste.....	60
Figura 24 - Exemplo da quinta situação de teste	61

Figura 25 - Exemplo da sexta situação de teste.....	61
Figura 26 - Exemplo da sétima situação de teste.....	62
Figura 27 - Exemplo da oitava situação de teste.....	62
Figura 28 - Exemplo da nona situação de teste.....	63
Figura 29 - Exemplo da décima situação de teste.....	63
Figura 30 - Exemplo da décima primeira situação de teste.....	64
Figura 31 - Exemplo da décima segunda situação de teste.....	64
Figura 32 - Exemplo da décima terceira situação de teste.....	65
Figura 33 - Tela final de execução do programa de teste com a biblioteca JUnit.....	66
Figura 34 - Principais características do Modelo de Referência.....	68
Figura 35 - Tela do programa PlanEditor do DECAF para elaboração dos agentes.....	96
Figura 36 - Plano parcial DECAF da mensagem solicitação	98
Figura 37 - Plano parcial DECAF da mensagem resposta.....	99
Figura 38 - Esquemático das classes em Java para o Agente Padrão.....	100
Figura 39 - Camadas do ambiente de produção do sistema multiagentes.....	101
Figura 40 - Tela de inicialização do Agente Padrão.....	102
Figura 41 - Tela de execução da classe Agent rodando o agente Agenc01.....	103

Lista de Tabelas

Tabela 1 - Equivalência do problema de escalonamento com o sistema multiagentes.....	42
Tabela 2 - Cenários de Teste para a Abordagem Proposta.....	69
Tabela 3 - Resultados do Cenário de Teste 1.....	70
Tabela 4 - Resultados do Cenário de Teste 2.....	71
Tabela 5 - Resultados do Cenário de Teste 3.....	71
Tabela 6 - Resultados do Cenário de Teste 4.....	72
Tabela 7 - Resultados do Cenário de Teste 5.....	72
Tabela 8 - Resultados do Cenário de Teste 6.....	73
Tabela 9 - Resultados do Cenário de Teste 7.....	73
Tabela 10 - Resultados do Cenário de Teste 8.....	74
Tabela 11 - Resultados do Cenário de Teste 9.....	74
Tabela 12 - Resultados do Cenário de Teste 10.....	75
Tabela 13 - Resultados do Cenário de Teste 11.....	75
Tabela 14 - Resultados do Cenário de Teste 12.....	76
Tabela 15 - Resultados do Cenário de Teste 13.....	76
Tabela 16 - Resultados do Cenário de Teste 14.....	77
Tabela 17 - Resultados do Cenário de Teste 15.....	77
Tabela 18 - Resultados do Cenário de Teste 16.....	78
Tabela 19 - Resultados do Cenário de Teste 17.....	78
Tabela 20 - Resultados do Cenário de Teste 18.....	79
Tabela 21 - Resultados do Cenário de Teste 19.....	79
Tabela 22 - Resultados do Cenário de Teste 20.....	80
Tabela 23 - Resultados do Cenário de Teste 21.....	80

Tabela 24 - Resultados do Cenário de Teste 22.....	81
Tabela 25 - Resultados do Cenário de Teste 23.....	81
Tabela 26 - Resultados do Cenário de Teste 24.....	82
Tabela 27 - Resultados do Cenário de Teste 25.....	82

Capítulo 1 - Introdução

1.1) Motivação

A elaboração de agendas é um problema freqüente. Sempre que houver escassez de recursos materiais e humanos para executar alguma atividade, uma possibilidade de solução de problema quando pode haver espera pela liberação de recursos é a elaboração de uma agenda de atendimento. Isso implica na elaboração da agenda de cada um desses recursos, ou seja, designá-los a uma atividade num certo período de tempo. Este problema de elaboração de agendas é usualmente tratado como o Problema de Escalonamento com Restrição de Recursos (BLAZEWICZ et. al., 1996).

Em particular, nos portos, uma atividade crítica é a elaboração da agenda de atendimento aos navios que solicitam operação nos mesmos. Um porto pode ser definido como uma área de terra limítrofe com o mar ou rio que possui equipamentos para carregar e descarregar navios ou barças com segurança para a embarcação e para a carga (UNCTAD, 1996) (AGERS-CHOU, 1983) (ROSA, 2006). Em virtude das inúmeras situações que podem ocorrer na navegação e na operação dos navios, ocorrem, eventualmente, atrasos e, também, antecipações da data de chegada dos navios. Essas mudanças de data levam a uma constante alteração da agenda de atendimentos dos navios. Os administradores do porto preparam a agenda de atendimento dos navios no terminal portuário com base nos dados do navio e da carga. Esta agenda do porto é denominada fila de navios. Deve ser destacado, que para a elaboração da agenda do porto, deve-se considerar, não só, a área de atracação dos navios (berços de atracação), mas, também, os equipamentos de carga e descarga dos navios, bem como, a movimentação no pátio e a área de estocagem dos pátios dentre outros fatores.

Para a elaboração e a modificação da agenda, ocorre um longo processo de negociação entre os diversos departamentos operacionais e administrativos do porto, tais como a gerência de cais, a gerência de pátio de contêiner, a gerência de transporte, dentre outros, com as empresas usuárias do porto, com os diversos proprietários de navios que solicitam operação no porto e com os proprietários de navio que já se encontram na fila para serem atendidos e que, eventualmente, podem ter sua agenda de operação alterada.

Neste processo de negociação, nem todas as informações são conhecidas pelos negociadores, tendo em vista que muitas destas informações são confidenciais, internas às empresas e/ou departamentos, e, assim sendo, as decisões são tomadas com as informações disponíveis no momento. Na grande maioria dos casos, não é uma informação completa de todo o problema.

Portanto, decisões têm que ser tomadas sem que se tenham todas as informações sobre o problema.

Assim, mecanismos de otimização baseados em modelagem matemática não se apresentam como solução mais adequada para o tipo de problema anteriormente apresentado, pois geralmente, pressupõem informações de custo disponíveis.

Esta tese propõe uma abordagem inspirada na maneira como as pessoas numa sociedade resolvem os problemas. Usualmente, as pessoas possuem informações e responsabilidades que não são suficientes para resolverem sozinhas um dado problema. Assim, elas solicitam recursos e/ou serviços a outras pessoas/setores que possuem outras competências e habilidades e, através da negociação, trabalham em conjunto para atingir um objetivo comum que é a resolução do problema.

1.2) Descrição da área

No caso específico do problema de geração da agenda diversas abordagens podem ser usadas. Dentre elas, citam-se: 1) Custos totais; 2) Leilão; 3) Métodos clássicos.

1. Na abordagem por custos totais, busca-se apurar o custo de todas as soluções e, então, optar pela solução de menor custo. No entanto, esta abordagem não é muito viável para o problema que está sendo estudado, pois não se sabem os custos envolvidos em todos os departamentos e empresas envolvidas no processo de negociação da elaboração da agenda, tendo em vista serem empresas e departamentos independentes e autônomos e, muitas vezes, de natureza confidencial (BLAZEWICZ et. al., 1996).
2. A abordagem por leilão, também conhecida como rede de contrato, é baseada na idéia que um elemento ou empresa detenha a informação e, então, oferte aos outros participantes serviços que eles possam desempenhar (SMITH, 1980) (SMITH, DAVIS, 1981). Cada um destes indivíduos, após a análise de oferta de serviço recebida, decide se deve propor um preço pelo serviço ou simplesmente desistir da oferta. Com base em todas as respostas recebidas com os preços dos diversos participantes do leilão, o leiloeiro decide quem é o vencedor e o informa que pode fazer o serviço. Este método pressupõe que, pelo menos, um elemento do grupo tenha informações totais e globais, o que na prática dos ambientes logísticos, normalmente, não ocorre. Portanto, este método se mostra pouco adequado para ser aplicado no ambiente do terminal portuário.
3. A abordagem por métodos clássicos, apesar de conseguir bons resultados, esbarra em dois problemas: 1) Necessita, na maior parte dos casos, de informações completas sobre todas as nuances do problema e 2) Necessitam, em sua grande maioria, de refazer toda a agenda, ou escalonamento, quando houver alteração em um item. Estes dois motivos dificultam muita sua utilização, pois dificilmente se têm informações completas para se tomar uma decisão e, também, para se refazer a agenda a todo instante pode-se gastar muito tempo e dinheiro (BLAZEWICZ et. al., 1996).

Observa-se, no entanto, que muitos dos problemas na sociedade humana são resolvidos através da interação entre diversos humanos que sozinhos não poderiam resolver o problema como um todo. No momento que se unem e trabalham em conjunto, cada um com seus recursos e capacidades, conseguem resolver problemas complexos e difíceis (LAKATOS, 1991).

Isto é possível porque os seres humanos, através de diferentes meios de negociação, conseguem estabelecer parcerias para resolver problemas complexos. Estas parcerias na sociedade humana conseguem fazer com que os recursos de cada indivíduo sejam bem utilizados. Apesar dos indivíduos não conhecerem o objetivo global da sociedade, a busca individual de cada um por seus objetivos pessoais pode levar a uma boa utilização dos recursos da sociedade (LEWICKI et. al., 2002).

Uma outra questão observada na sociedade é que, mesmo quando a relação entre os indivíduos não leva inicialmente a uma boa solução do uso dos recursos, a sociedade acaba caminhando para obtenção de bons resultados, os quais são alcançados à medida que vão ocorrendo interações/negociações.

Com base nesta constatação, busca-se analisar a hipótese de se simular computacionalmente um modelo que inspirado na sociedade humana consiga resolver problemas de elaboração de geração da agenda de um terminal portuário.

O problema do terminal portuário tem como primeira característica ser distribuído entre as diversas áreas do porto e os proprietários dos navios. Além de ser distribuído, o problema tem como segunda característica ser resolvido através da negociação da resolução do problema entre os diversos atores. Vale ressaltar que uma terceira característica do problema é que estes diversos atores são autônomos e independentes. Desta forma, buscam-se tecnologias para implementação da abordagem proposta que possam se adaptar a esta característica do problema.

Dentro deste contexto, a primeira tecnologia a ser analisada é a de Inteligência Artificial Distribuída, que corresponde ao caráter distribuído do problema e, também, às características de negociação. A Inteligência Artificial Distribuída tem por função a investigação de modelos de conhecimento, de técnicas de comunicação e de técnicas de argumentação entre um grupo de elementos ativos de uma sociedade. Estes participantes são denominados agentes e o ambiente onde eles estão inseridos é denominado ambiente multiagentes ou sistemas multiagentes (O'HARE, JENNINGS, 1996).

Na área de Inteligência Artificial Distribuída, destaca-se a tecnologia de Agentes (WOOLDRIDGE, 2000) que provê além das questões de distribuição e negociação o quesito de autonomia dos atores envolvidos na negociação. Wooldridge (2000) faz uma definição geral de agente como sendo um programa de computador que está situado em um ambiente e é capaz de tomar ações autônomas neste ambiente no intuito de alcançar seus objetivos.

Por fim, destaca-se a área de conhecimento da classe de problemas denominada Problema de Escalonamento Dinâmico com Restrição de Recursos. No contexto desta tese, o escalonamento é um processo que visa designar máquinas e recursos ao longo do tempo para atender diversas solicitações que são compostas de diversas atividades denominadas tarefas (ZWEBEN, FOX, 1994) (BLAZEWICZ et. al., 1996). Estes recursos, usualmente, são limitados, não sendo suficientes para atender todas as solicitações simultaneamente. Esta designação é feita de tal maneira que sejam respeitadas integralmente as restrições de tempo de execução das atividades e o limite de capacidade do conjunto de máquinas e recursos usados por todas

as atividades. O resultado desta designação afeta diretamente a qualidade do escalonamento no que diz respeito a critérios como: custo, atraso, capacidade de produção, dentre outros (ZWEBEN, FOX, 1994).

Normalmente, problemas de escalonamento são problemas dinâmicos, pois sempre ocorrem imprevistos que obrigam a realização de novos planejamentos das atividades sempre que houver uma alteração no ambiente (BLAZEWICZ et. al., 1996). Para resolver este aspecto do problema, usualmente, adotam-se duas técnicas a fim de acomodar a modificação ou inserção de uma solicitação:

1. Refazer todo o planejamento;
2. Adaptar o planejamento corrente com alterações pontuais do plano.

No primeiro caso, pode ocorrer uma perturbação da agenda de muitas solicitações já agendadas gerando um impacto muito grande para todos os envolvidos no processo de elaboração da agenda. No segundo caso, como as alterações são pontuais no entorno da alteração ou da inserção da solicitação, o impacto gerado é bem menor, além de ser, também, menor o tempo de processamento e, conseqüentemente, melhorando o tempo de resposta do sistema para adaptação da agenda.

1.3) O Problema de elaboração da agenda

Neste trabalho são tratados os problemas de elaboração da agenda de ocupação de recursos para atendimento a solicitação de serviços. Esta agenda a ser elaborada possui algumas características que a diferencia de outros problemas. Dentre estas, destacam-se:

1. Lista de recursos. Estes recursos estão distribuídos entre os diversos atores participantes da negociação e podem ser agendados para as solicitações que são feitas a eles;
2. Lista de solicitações que demandam recursos.

Os diversos participantes devem negociar entre si para que seja estabelecido um horário de consenso para atendimento a uma solicitação de serviço feita pelo cliente. Assim, os recursos de cada um dos diversos participantes ficam reservados para atender a solicitação. Na maioria dos casos, o participante não possui todos os recursos necessários para atender a requisição de serviço recebida e, neste caso, ele deve solicitar estes recursos a outros participantes e estabelecer com eles um horário de consenso entre todos. Caso este horário seja igual ao da requisição ele pode confirmar com o solicitante. Caso contrário, ele deve propor um novo horário que interessa ao solicitante do serviço.

Assim sendo, busca-se desenvolver um sistema para que, uma vez recebida uma solicitação, elabore uma agenda de utilização dos recursos levando-se em conta os seguintes itens:

1. Deve-se, sempre, encontrar uma solução;

2. Deve-se reduzir o impacto nas solicitações previamente agendadas. Caso ocorra alguma mudança de horário de alguma solicitação já agendada na agenda de algum participante, haverá, por conseguinte, a alteração das agendas de todos os outros participantes num processo em cascata que acarretará novas negociações para acomodar esta mudança;
3. Não deve ser exigida a divulgação das informações das partes envolvidas;
4. Deve-se considerar o fato dos diversos atores da negociação estarem distribuídos geograficamente. Esta situação torna mais complexa a negociação em virtude dos participantes estarem dispersos geograficamente, dificultando o contato entre eles e atrasando o processo de troca de sugestões de novos horários e, conseqüentemente, o fechamento da negociação;
5. Deve-se respeitar a autonomia dos diversos envolvidos no processo de negociação. Cada um dos envolvidos no processo possui interesses próprios, atende simultaneamente a diversos outros participantes e não compartilha de um objetivo único na cadeia logística. Com isto, torna-se fundamental a questão da negociação entre eles.
6. A inserção e a alteração das solicitações ocorrem de maneira dinâmica. Esta situação dificulta a resolução dos problemas, pois as requisições feitas podem ter seus horários alterados a qualquer momento ocasionando assim uma constante reprogramação das requisições e com isto a agenda passa a ser dinâmica.

Esta situação é típica das empresas participantes do processo de elaboração de agendas de serviços. Elas estão constantemente elaborando a agenda de ocupação de seus recursos e fornecendo um horário possível de atendimento para a solicitação feita.

1.4) Hipótese

Tem-se como hipótese que, caso seja possível elaborar uma abordagem que represente os padrões de negociação usados pelos seres humanos, então será possível construir um sistema computacional que leve à solução do problema de elaboração da agenda de utilização de recursos. Os procedimentos humanos que mais se destacam são: autonomia e capacidade de negociação.

A abordagem é baseada em uma sociedade virtual composta de indivíduos que tenham as seguintes características: 1) Possuam recursos não compartilhados; 2) Possuam a capacidade de negociar entre si para a elaboração da agenda de seus recursos; 3) Possuam a capacidade de reconhecer os favores recebidos e prestados e 4) Possuam uma representação similar aos padrões de negociação da sociedade humana.

Para a solução do problema deve haver um protocolo de negociação inspirado nos procedimentos utilizados pelos humanos para elaboração da agenda de um porto. A estratégia básica utilizada é a de que o primeiro a chegar ao porto é o primeiro a ser atendido.

Outro protocolo proposto, similar ao anteriormente citado, porém com inspiração na troca de favores adotada pelos humanos, onde, no intuito de resolver conflito, quando duas solicitações requisitam atendimento na mesma data e a decisão é tomada em função do histórico passado, compensação de favores, que os envolvidos tem uns pelos outros.

Em princípio, esta abordagem proposta pode ser estendida para outros problemas similares que se enquadrem na classe maior de problemas denominada Problema de Escalonamento Dinâmico com Restrição de Recursos.

1.5) Metodologia

No intuito de validar a abordagem proposta na tese, deve-se desenvolver um sistema computacional que a implemente e gere resultados a fim de serem analisados. Este sistema deve implementar agentes autônomos que negociem entre si, segundo protocolos de negociação definidos, visando obter uma solução global do problema. O sistema deverá ter as seguintes características:

1. Facilidade de programação;
2. Mecanismos de depuração;
3. Deve ter a possibilidade de ser distribuído entre computadores ligados em rede;
4. Deve possuir simplicidade de implementação de protocolos de negociação.

Para tanto, foi escolhido o *framework* para desenvolvimento de agentes denominado DECAF (*Distributed Environment Centered Agent Framework*) (GRAHAM, 2001, 2000). Este ambiente desenvolvido na linguagem de programação Java possui um ambiente gráfico para implementação da lógica de recebimento, envio e tratamento de mensagem, bem como, um gerador automático de código a partir deste ambiente, o que facilita, em muito, o desenvolvimento do sistema computacional. Tendo em vista que o *framework* DECAF gera o código na linguagem Java, os protocolos de negociação, também, serão escritos na linguagem de programação Java (KEN et. al., 2006) (ECKEL, 2002) (LEA, 1997). Aliado a isso, esta linguagem se adapta muito bem a ambientes multi-plataformas e distribuídos e suporta todos os quesitos de orientação a objetos.

Também deve-se implementar um Modelo de Referência que servirá como base de comparação dos resultados alcançados pelos dois protocolos de negociação propostos na abordagem desta tese. O Modelo de Referência será construído e executado fazendo uso do software de simulação Arena (KELTON et. al., 2004) (AMORIM, 2005) que simula de forma estática o recebimento de solicitações para operar no porto. A escolha pelo modelo de simulação se deu em função da larga utilização desta abordagem nos terminais portuários atualmente.

No intuito de se avaliar os resultados alcançados pela abordagem proposta, serão feitos cenários de testes que visam testar diferentes situações que podem ocorrer na prática. As características dos cenários de testes são as seguintes: tamanho do porto, frequência de chegada de navios ao porto, tempo de operação de cada navio. Como características do tamanho do

porto, serão usadas configurações de berço de atracação, buscando configurações bem restritivas, com um berço somente e outros que representem cenários mais usuais dos portos. Para as características do tamanho do porto, será gerada uma seqüência de chegada de navios com base em uma distribuição exponencial. Para cada um dos navios das seqüências de chegada geradas, serão gerados os respectivos tempos de operação baseados numa distribuição normal.

Nestes cenários, é estabelecido um limite máximo de tempo, a ser definido, entre a data de solicitação de atracação e a data possível de atendimento. Caso não seja possível atender dentro deste limite, o navio não será agendado para operar no porto. Estes cenários serão resolvidos pelo sistema que implementa os protocolos propostos e, também, pelo Modelo de Referência. Serão realizadas várias simulações para cada um dos cenários, pois os resultados dos testes podem não ser determinísticos.

Após isto, serão comparados os resultados alcançados pelo Modelo de Referência e pelos protocolos propostos na abordagem da tese. Para comparação serão usadas três métricas:

1. *Flow time*, tem por objetivo avaliar o tempo entre a data de solicitação de atracação e a data efetiva de atracação;
2. Número de solicitações não atendidas, tem por função medir a capacidade do sistema de agendar o máximo possível de navios no porto respeitando o limite máximo de vinte e quatro horas entre a data de solicitação de atracação e a data possível de atendimento;
3. Número de mensagens realizadas, este valor representa o número total de mensagens e indica o esforço do sistema proposto para chegar à solução final.

Ao final, será possível comparar os resultados alcançados pela abordagem proposta com o Modelo de Referência, a fim de validar a sua eficácia.

1.6) Contribuição Científica

A primeira contribuição científica desta tese é a elaboração de uma abordagem inspirada no funcionamento da sociedade humana com seus padrões de negociação aplicada à solução dos problemas de elaboração de agendas onde poucos recursos, materiais e humanos, estão disponíveis e, também, que nem todas as informações estejam disponíveis.

Como segunda contribuição científica, destaca-se a proposta de um novo mecanismo de negociação que se vale da inspiração na emoção humana, mais especificamente a emoção gratidão, que tem por objetivo a resolução de impasse na negociação.

Como terceira contribuição científica tem-se a implementação da metodologia em ambiente computacional utilizando a técnica de agentes e sistemas multiagentes aplicada ao problema prático do Terminal Portuário (ROSA, SCHNEEBELI, 2005, 2003 (a) (b)).

1.7) Estrutura do trabalho

Esta tese está dividida em seis capítulos que são estruturados como se segue.

No Capítulo 1, é apresentada a introdução ao trabalho que contém a definição do problema a ser tratado, a elaboração da hipótese para solução do problema. Também, fazem parte deste capítulo a metodologia proposta, os objetivos e a contribuição científica desta tese.

O Capítulo 2 elabora uma revisão do estado da arte de negociação de agentes e de emoções em agentes que serve de base para a estruturação da metodologia proposta.

No Capítulo 3, são apresentados o problema de escalonamento e as soluções já existentes para o mesmo.

No Capítulo 4, é descrita a abordagem proposta nesta tese, baseada em negociação de agentes.

O Capítulo 5 apresenta os experimentos realizados e os resultados alcançados.

No Capítulo 6, é elaborada a conclusão do trabalho e são apresentadas propostas de trabalhos futuros que podem ser elaborados a partir da metodologia proposta nesta tese.

Capítulo 2 - Revisão do Estado da Arte de Negociação de Agentes e Emoções em Agentes

Conforme visto no capítulo 1, a abordagem proposta é dividida em duas partes:

1. Mecanismos de negociação inspirados na sociedade humana;
2. Abordagem baseada em agentes e ambientes multiagentes.

A primeira parte da abordagem é o mecanismo de negociação que busca representar em computador como a sociedade humana elabora e conduz a negociação. Dentro deste capítulo, no contexto da sociedade humana são tratados os temas negociação e emoções.

A segunda parte da abordagem faz uso da tecnologia de agentes para a implementação computacional. Dois temas são analisados neste capítulo para a fundamentação teórica desta tese: 1) Inteligência Artificial Distribuída e 2) Agentes e sistemas multiagentes e as formas de coordenação em sistemas multiagentes.

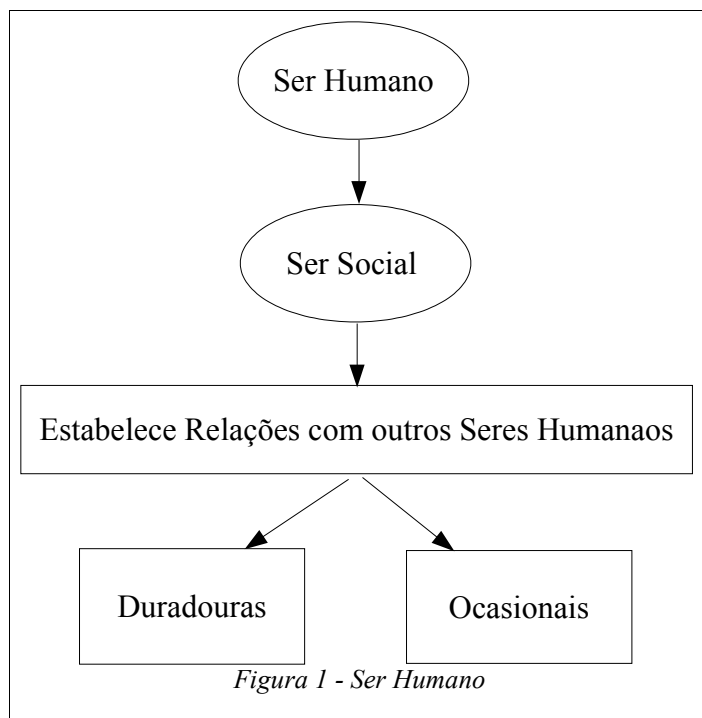
Acrescenta-se aos dois tópicos citados anteriormente a questão da elaboração de agendas de maneira dinâmica. Esta elaboração de agenda é tratada matematicamente como o Problema de Escalonamento Dinâmico com Restrição de Recursos que é apresentado, também, neste capítulo.

Estruturou-se esta revisão nos seguintes tópicos:

1. Sociologia e Sociedade Humana;
2. Negociação;
3. Emoção;
4. Agentes e Sistemas Multiagentes;
5. Problema de Escalonamento Dinâmico com Restrição de Recursos.

2.1) Sociologia e sociedade humana

O homem pode ser visto como um ser social que estabelece relações com outros seres humanos (Figura 1). Estas relações, na maioria das vezes, ocorrem pela impossibilidade de um indivíduo realizar suas tarefas sozinho, isolado de todos, necessitando assim, da ajuda de outros seres humanos. Esta dependência pode ter caráter de longa duração, definida como duradoura, ou em um situação específica, que são as ocasionais (Figura 1).



No intuito de melhor esclarecer estas relações, deve-se analisar o problema pela ótica da sociologia. A sociologia é uma ciência que tem por objeto o estudo dos fatos sociais. Para a sociologia, a sociedade é um grande complexo de relações humanas ou um grande sistema de interação, (DURKHEIN, 1966). Durkheim define o fato social como uma “*coisa*” que, para seu estudo devem ser aplicados métodos e processos, similarmente aos que são empregados nas ciências exatas. Assim, para explicar o fato social deve existir uma investigação das causas sociais e não somente das causas históricas, psicológicas e biológicas (LAKATOS, 1991).

Max Weber conceitua a sociologia a partir da ação social. A ação social segundo Weber é a conduta humana, pública ou não, a que um agente atribui significado subjetivo. Assim, para Weber a sociologia é uma ciência que tem por objetivo compreender claramente a conduta humana e fornecer explicação das causas de sua origem e resultados. Neste contexto, a conduta social seria o elo para a compreensão da situação social e o entendimento das intenções (LAKATOS, 1991).

Talcott Parsons propõe um esquema de referência para o estudo da ação social proposta por Weber com base em três elementos principais (LAKATOS, 1991):

1. O agente, ator da ação social;
2. A situação em que o agente está inserido. A situação é composta de três elementos: 1) Objetos físicos, que são os meios e condições da ação; 2) Objetos sociais, que são as outras pessoas, *alter*; 3) Objetos culturais, que são os elementos simbólicos das tradições culturais;
3. Orientação do agente em relação à situação. Existem dois tipos de orientação: 1) Orientação motivacional, que se refere aos aspectos relacionados com a gratificação ou com a privação atual ou potencial das necessidades do ator. Esta orientação é importante na resolução de conflitos entre diversos interesses; 2) Orientação de valor, que representa o cumprimento das normas e padrões sociais e a submissão do ator a estas normas e padrões. Sua importância é no sentido de orientar o ator a escolher as diversas opções existentes, levando-se em conta as conseqüências relacionadas com a integração dos sistemas sociais em que ele está inserido.

Com base nos conceitos anteriormente enunciados, percebe-se que para uma sociedade existir, é necessário a existência de atores e que estes se relacionem entre si. Desta forma, para que estas relações aconteçam, devem existir processos sociais para se desenvolverem os relacionamentos entre os diversos agentes na sociedade. Os processos sociais possuem três funções importantes (LAKATOS, 1991):

1. O reconhecimento pelo indivíduo da necessidade de sua sociabilidade e participação nos grupos;
2. A observação das eventuais conseqüências psicológicas dos diferentes tipos de isolamento, quer seja nos indivíduos, quer seja na sociedade;
3. A identificação dos diversos aspectos dinâmicos das relações sociais, através de diferentes formas de processo social.

O processo social pode apresentar as seguintes formas descritas a seguir (LAKATOS, 1991):

1. Contato social. É a base dos processos sociais, sem esta forma de processo social, não há possibilidade de existir as outras formas de processo social;
2. Interação social. É vista como sendo a reciprocidade de ações sociais;
3. Comunicação. Tem função chave e é importantíssima para que ocorra a interação social;
4. Cooperação. É o elemento básico para que se tenha continuidade e manutenção dos grupos e sociedades;
5. Competição e conflito. São os fatores que desagregam a sociedade, alterando as relações entre os indivíduos e gerando, conseqüentemente, o fim da sociedade;
6. Adaptação, acomodação e assimilação. São fatores que agregam a sociedade e que leva o indivíduo a aceitar, aderir e se conformar com as normas estabelecidas. Com isto, aumenta-se o nível de coesão da sociedade.

Após estas considerações sobre a sociologia e sobre a sociedade humana, são analisadas as negociações na sociedade humana.

2.2) *Negociação*

A negociação é um processo que ocorre entre todos os indivíduos, ricos e pobres, inteligentes ou ignorantes, todos os dias em toda a sociedade humana. As negociações ocorrem tanto em nível familiar entre pais e filhos, como no âmbito das empresas e chegando ao nível dos países e suas relações diplomáticas. É interessante notar que a estrutura e os processos de negociação são fundamentalmente os mesmos desde o nível pessoal, alcançando o nível diplomático e passando pelo nível empresarial (LEWICKI et. al., 2002).

Na negociação deve haver uma necessidade mútua entre as partes, sendo esta relação de dependência mútua denominada interdependência. As negociações podem ocorrer devido a umas das duas razões: 1) Para realizar ou criar algo novo que nenhuma das partes poderia realizar sozinha e 2) Para resolver um problema ou um conflito entre as partes (LEWICKI et. al., 2002).

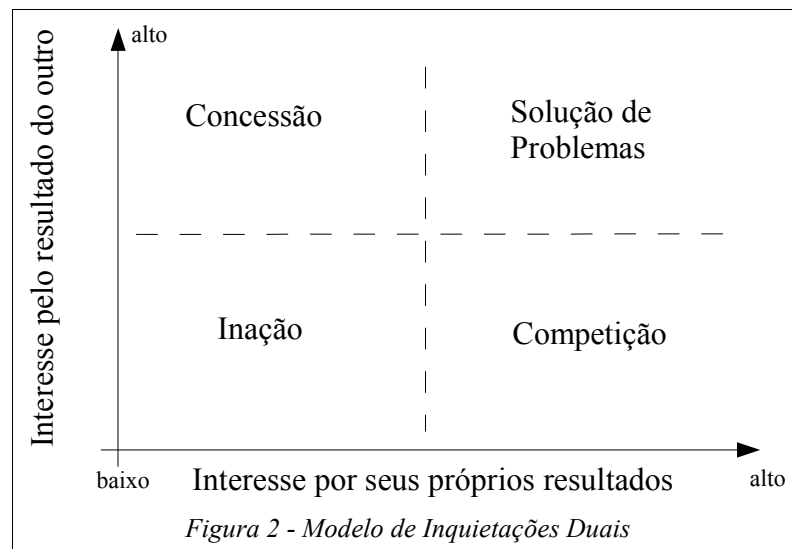
Usualmente as negociações ocorrem em função de alguns fatores: 1) Existem duas ou mais partes envolvidas; 2) Existem conflitos de interesse entre as partes. Isto ocorre porque o que um quer não é necessariamente o que o outro quer e, as partes devem resolver este conflito; 3) As partes preferem tentar chegar a um acordo em vez de iniciar uma luta direta entre elas.

Uma das áreas de maior interesse em negociação é o gerenciamento do conflito, pois este fator pode impedir uma negociação de avançar e/ou trazer grandes prejuízos para as partes. Uma maneira de resolver conflito seria a criação de um elemento na sociedade que seria o juiz do conflito, entretanto, para esta tese, esta hipótese não é considerada, devendo as partes envolvidas na negociação resolver seus próprios conflitos.

A taxonomia das diferentes abordagens para o conflito é feita com base na estrutura bidimensional de Pruitt e Rubin (1986) denominada *Modelo de Inquietações Duais*, Figura 2. Esta estrutura define dois tipos de interesse para cada indivíduo:

1. Interesse por seus próprios resultados;
2. Interesse pelo resultado do outro.

Estes dois tipos de interesse formam um gráfico bidimensional onde o eixo horizontal representa o interesse pelos seus resultados e, o vertical, o interesse pelo resultado do outro, Figura 2. A dimensão vertical é conhecida como a dimensão da cooperação, e a dimensão horizontal é tida como a dimensão da assertividade (LEWICKI et. al., 2002).



Assim sendo, conforme pode ser visto na Figura 2, existem quatro tipos de abordagem para o conflito. São apresentados a seguir os quatro tipos de abordagens para o conflito:

1. Inação. Nesta abordagem, o indivíduo tem pouco, ou nenhum, interesse de alcançar seus próprios resultados e nem tão pouco demonstra interesse pelo fato da obtenção dos resultados da outra parte. Esta abordagem é tida como passividade ou retração, ou falta de vontade de negociar com a outra parte;
2. Concessão. Neste caso, o indivíduo não busca com tanto interesse seus próprios resultados e, ao contrário, demonstra muito interesse em ajudar a outra parte a obter os resultados que ela deseja. O indivíduo, nesta abordagem, passa por sufocar as suas próprias aspirações para deixar o outro vencer;
3. Competição. Na competição, o indivíduo almeja alcançar seus próprios resultados a qualquer custo, demonstrando pouco, ou nenhum, interesse pelos interesses da outra parte. Neste tipo de abordagem, o indivíduo tenta dissuadir a outra parte através de todos os meios, levando-a a conceder o que ele deseja a qualquer custo;
4. Solução de problemas. Na abordagem solução de problemas, o indivíduo procura alcançar seus próprios resultados com muito interesse e, também, via o processo de negociação, deseja que a outra parte alcance seus resultados. Na solução de problemas, ambas as partes buscam abordagens para maximizar o objetivo das duas partes em conjunto. Este é o caso de maior interesse para esta tese.

Dois tipos de fatores devem sempre serem levados em conta numa negociação, são eles: 1) Fatores tangíveis e 2) Fatores intangíveis. Os fatores tangíveis dizem respeito a preços, quantidades, coisas mensuráveis de uma maneira geral. Já os fatores intangíveis são constituídos pelas motivações psicológicas fundamentais que podem, direta ou indiretamente, influenciar os indivíduos durante a negociação (LEWICKI et. al., 2002).

Os fatores intangíveis são fortemente influenciados, em maior ou menor grau, pelas emoções despertadas pelos indivíduos no processo de negociação. Assim sendo, pode-se perceber que

as emoções humanas constituem um fator muito importante nas negociações na sociedade humana.

Na seção seguinte é feita uma análise das emoções humanas e, mais especificamente, das emoções morais.

2.3) Emoção

O artigo de Kleinginna e Kleinginna (1981) descreve mais de cem definições de emoções propostas por renomados cientistas. Isto mostra que a definição de emoção é algo ainda não fechado e, também, que o ser humano conhece pouco sobre sua própria natureza.

Um ponto importante a respeito das emoções é a afirmação de Damasio (1994) que o cérebro humano não consegue tomar decisões quando ocorre perda parcial ou um mal funcionamento do sistema neural responsável por tratar emoções e, portanto, o ser humano sem emoções não consegue deliberar sobre as decisões.

Com base no trabalho de Damásio (1994), a emoção pode ser classificada em dois tipos:

1. Primárias ou básicas. As emoções primárias são inatas aos animais e, portanto, são adaptadas às necessidades das espécies. Elas aparecem em todos os seres humanos e mamíferos. Elas estão relacionadas com manifestações corporais e ligadas às necessidades evolucionárias das espécies. Como exemplo cita-se o medo, que faz com que o animal decida se deve lutar ou fugir. Clore et. al. (1990) propõem um conjunto que representa uma maior convergência entre os cientistas de quais seriam as emoções primárias. Este conjunto é formado pelas seguintes emoções: raiva, medo, aversão, prazer, tristeza;
2. Secundárias ou derivadas. Elas existem somente nos seres humanos e alguns mamíferos e não podem ser consideradas inatas, sendo adquiridas ao longo do desenvolvimento do indivíduo (DAMASIO, 1994) (MURRAY, ARNOTT, 1993) (STIBBARD, 2001). Elas são originadas de uma ou mais emoções primárias.

As emoções podem, ainda, ser classificadas como emoções morais, sendo que existem poucos estudos sobre as emoções morais (HAIDT, 2003).

As emoções morais (HAIDT, 2003) são aquelas que estão relacionadas com o interesse ou o bem estar de uma sociedade, como um todo ou pelo menos de outras pessoas que não sejam os juízes ou a pessoa que fez a ação.

Tendo em vista que a proposta de um dos padrões de negociação é inspirada na emoção gratidão, deve-se então definir o que vem a ser a emoção gratidão. Ferreira (1999) define gratidão como: “1) Qualidade de quem é grato. 2) Reconhecimento por um benefício recebido; agradecimento, reconhecimento”.

A definição anterior reforça a tese de Haidt (2003) que a gratidão é uma emoção importante para unificação e equilíbrio da sociedade. Isto ocorre porque quando o indivíduo busca ajudar

o outro e este busca recompensá-lo, gera uma situação de equilíbrio na sociedade pois todos podem ajudar a todos e, também, podem ser ajudados por todos.

A próxima seção trata sobre agentes e sistemas multiagentes.

2.4) Agentes e Sistemas multiagentes

Agentes e sistemas multiagentes fazem parte de um conceito maior denominado Inteligência Artificial Distribuída (IAD) (*DAI-Distributed Artificial Intelligence*). Assim, é feito nesta seção uma definição de Inteligência Artificial Distribuída e depois são feitas as definições de agentes e de sistemas multiagentes.

Inteligência Artificial Distribuída é uma área de estudo dentro da grande área de estudo denominada Inteligência Artificial. A Inteligência Artificial Distribuída tem por função a investigação de modelos de conhecimento, de técnicas de comunicação e de técnicas de argumentação entre um grupo de elementos ativos de uma sociedade. Estes participantes são denominados agentes e o ambiente onde eles estão inseridos é denominado ambiente multiagentes ou sistemas multiagentes (O'HARE, JENNINGS, 1996).

A Inteligência Artificial Distribuída é fundamentada no conceito no qual a inteligência envolve o comportamento social. Assim, deve-se buscar a elaboração de modelos, arquiteturas e implementações para que os agentes da sociedade possam agir de modo coordenado para que ao final seja obtido um comportamento global coerente, isto é, seja alcançado um bom resultado global para os objetivos da sociedade (GARCIA, SICHMAN, 2003).

A Inteligência Artificial Distribuída é composta de duas grandes áreas (BOND, GASSER, 1988):

1. Solução distribuída de problemas (*DPS-Distributed problem solving*). Neste caso, o problema a ser resolvido é dividido em partes e, para resolver cada uma destas partes é designado um agente que tratará somente da parte específica que lhe cabe. Estes agentes cooperam entre si compartilhando objetivos e informações no intuito de resolver o problema (O'HARE, JENNINGS, 1996);
2. Sistemas multiagentes (*MAS-Multiagent system*). Neste caso, existem diversos agentes que são independentes, autônomos, possuem habilidades. Cada agente deve negociar com os outros agentes a ajuda necessária, recursos ou serviços, para resolver um problema que ele, individualmente, não consegue resolver. Um ponto importante é que os agentes possuem seus próprios objetivos e não compartilham informações sobre os objetivos individuais dos outros agentes nem sobre o objetivo global do sistema (O'HARE, JENNINGS, 1996) (DURFEE et. al., 1989).

Os sistemas multiagentes podem apresentar algumas vantagens sobre os sistemas centralizados e monolíticos (O'HARE, JENNINGS, 1996). Dentre estas vantagens, destacam-se:

1. Maior rapidez na solução de problemas. Como o sistema multiagentes faz uso do paralelismo inerente a este tipo de solução, ele pode chegar à solução de maneira mais rápida;
2. Diminuição da comunicação. Só são transmitidos dados parciais de alto nível das soluções para os outros agentes, em vez de enviar todos os dados brutos para o ambiente centralizado;
3. Mais flexibilidade. Permite de forma rápida e dinâmica agrupar agentes com as habilidades necessárias para resolver um certo problema;
4. Maior grau de confiabilidade. Os agentes podem assumir o lugar de um outro agente que não tenha conseguido realizar suas tarefas e, assim, o sistema tende a ser mais tolerante a falhas, permitindo um agente cobrir a falta gerada por outro agente.

Apesar das vantagens apresentadas anteriormente, algumas dificuldades ocorrem para a execução da modelagem e da implementação dos sistemas multiagentes (GASSER, 1991). Dentre estas dificuldades, citam-se:

1. Como formular, descrever, decompor e designar problemas aos agentes e conseguir ao final um resultado único;
2. Como capacitar os agentes para se comunicar e interagir. Quais as linguagens e protocolos devem ser usados;
3. Como cada agente deve reconhecer e superar pontos de vistas distintos em relação aos outros agentes.

Wooldridge (2000) faz uma definição geral de agente como sendo um programa de computador que está situado em um ambiente e é capaz de tomar ações autônomas neste ambiente no intuito de alcançar seus objetivos. Outros autores como Jennings e outros (1998), Huhns e Singh (1997) apresentam esta mesma definição.

Feber (1999) faz uma definição mais específica e detalhada do que é um agente a partir de algumas características que ele deve possuir. A seguir são citadas algumas destas características:

1. Possui um estado. O estado pode ser físico ou virtual. O estado físico ocorre quando os agentes atuam no mundo real como robôs, aviões, carros, etc. O estado é virtual quando os agentes não existem no mundo real, só em programas de computador;
2. Capaz de executar ações. Esta característica é fundamental para os sistemas multiagentes, pois através das ações executadas, haverá a modificação do ambiente no qual o agente está inserido;
3. Capaz de se comunicar. A comunicação é a capacidade do agente se comunicar com outros agentes. Para os agentes virtuais este é o principal meio de interação entre eles;

4. Possui autonomia. Quando o agente possui autonomia ele decide por suas ações baseado numa série de tendências individuais que o fazem caminhar para a satisfação de seus objetivos individuais;
5. Possuem recursos. Esta característica não é determinante para se definir um agente, no entanto, quando ele possui recursos, ele pode ter mais autonomia;
6. Possui uma representação do ambiente. Esta representação pode ser total ou parcial. A representação total ocorre quando o agente possui completo conhecimento de todo o ambiente que ele está inserido. A representação é parcial quando o agente tem conhecimento das informações que dizem respeito diretamente a ele, não tendo uma visão global do ambiente que ele está inserido. Esta última representação é que é usada nesta tese.

Garcia e Sichman (2003) acrescentam a característica do agente possuir, também, personalidade. A personalidade é a capacidade do agente possuir comportamentos que lembrem as características humanas, como por exemplo a emoção ou o mau humor.

Além das características específicas do agente, Wooldridge (2000) faz uma definição de agentes baseada no comportamento que eles devem possuir no sistema multiagentes no qual está inserido. Têm-se as seguintes características adicionais:

1. Devem ter capacidade de reação. O agente deve, de maneira autônoma, responder às mudanças na área do ambiente que está inserido e que tem conhecimento e, via de regra, não tem conhecimento global do ambiente;
2. Devem ser pró-ativos. Os agentes devem ter o comportamento de tomar ações independentemente de estímulos externos a fim de alcançarem a satisfação de seus objetivos individuais;
3. Devem ter habilidades sociais para que possam interagir com outros agentes e\ou seres humanos.

Subrahmanian e outros (2000) elaboram uma definição de agentes mais focada no desenvolvimento de programas de computador. Assim, um programa de computador agente possui as seguintes características:

1. É capaz de prover outros agentes de pelo menos um serviço útil para o outro agente usar;
2. Possui uma descrição dos serviços que ele provê. Esta descrição deve estar disponível para ser acessada e entendida por outros agentes;
3. Possui a habilidade de agir autonomamente sem necessitar de um direcionamento de um ser humano ou outro agente;

4. Possui uma declaração sucinta e explícita de como ele reage a determinados eventos. Esta declaração pode ou não ser pública para os outros agentes.

O'Hare e Jennings (1996) classificam os agentes com base em sua heterogeneidade. Desta maneira, eles elaboram as seguintes classificações para os agentes:

1. Idênticos, quando os agentes são iguais;
2. Baixa heterogeneidade, quando diferem somente nos recursos que estão disponíveis para eles;
3. Média heterogeneidade, quando os agentes diferem no método de solução do problema e na capacidade de resolver problemas que eles possuem;
4. Alta heterogeneidade, quando os agentes só possuem em comum uma linguagem de interação.

O'Hare e Jennings (1996) classificam, também, os agentes pelo meio ambiente no qual eles estão inseridos, podendo, então, ter os seguintes tipos:

1. Artificiais, quando só existem em computadores, como programas;
2. Humanos, referem-se ao próprio ser humano;
3. Reais, são os agentes físicos, especificamente os robôs.

Os agentes podem, ainda, ser classificados conforme o seu grau de cooperação com os outros agentes. Desta forma, têm-se dois tipos:

1. Cooperativo. O agente é cooperativo quando compartilha de um objetivo global que é igual para todos os agentes do ambiente e trabalha em prol deste objetivo;
2. Competitivo. O agente é competitivo quando possui objetivos próprios e não compartilha um único objetivo global. Neste tipo de agente uma solução global é conseguida através da negociação entre os agentes, sendo que esta solução pode vir a ser diferente da solução individual desejada.

Para entender como os agentes são agrupados em sistemas multiagentes deve-se definir o que vem a ser uma agência. Uma agência é definida como sendo um conjunto de agentes com compromissos mútuos, compromissos globais e crenças mútuas (O'HARE, JENNINGS, 1996). A intensidade dos compromissos e das crenças vai depender do grau de cooperação entre os agentes e da capacidade de troca das informações sobre os compromissos e crenças.

Assim, pode-se dizer que um grupo de agentes forma uma agência quando estão conectados de alguma maneira e a união de suas atividades gera um resultado melhor e mais harmonioso do que quando eles estavam isolados (MALONE, 1990). Uma organização é composta de: 1) Um grupo de agentes; 2) Um conjunto de atividades que são realizadas pelos agentes; 3) Um conjunto de ligações entre os agentes; 4) Um conjunto de objetivos que serve como critério para avaliação do resultado da organização (MALONE, 1990).

As organizações humanas são a principal fonte de inspiração para as organizações citadas e, conseqüentemente, para o estudo da Inteligência Artificial Distribuída (MALLONE, 1990) (O'HARE, JENNINGS, 1996) (WERNER, 1989).

Com base na definição de organização, pode-se definir que sistemas multiagentes são ambientes virtuais ou reais onde existam pelo menos dois agentes e que estes agentes interajam entre si de maneira produtiva utilizando a infra-estrutura que o meio proporciona. Esta infraestrutura é formada por dois tipos de protocolos:

1. Protocolos de comunicação. São os protocolos que habilitam os agentes a trocar e entender mensagens;
2. Protocolos de interação. São os protocolos que habilitam os agentes a desenvolver conversações suportadas pelos protocolos de comunicação.

Os sistemas multiagentes podem ser estruturados com base em quatro arquiteturas descritas a seguir (BAKER, 1998):

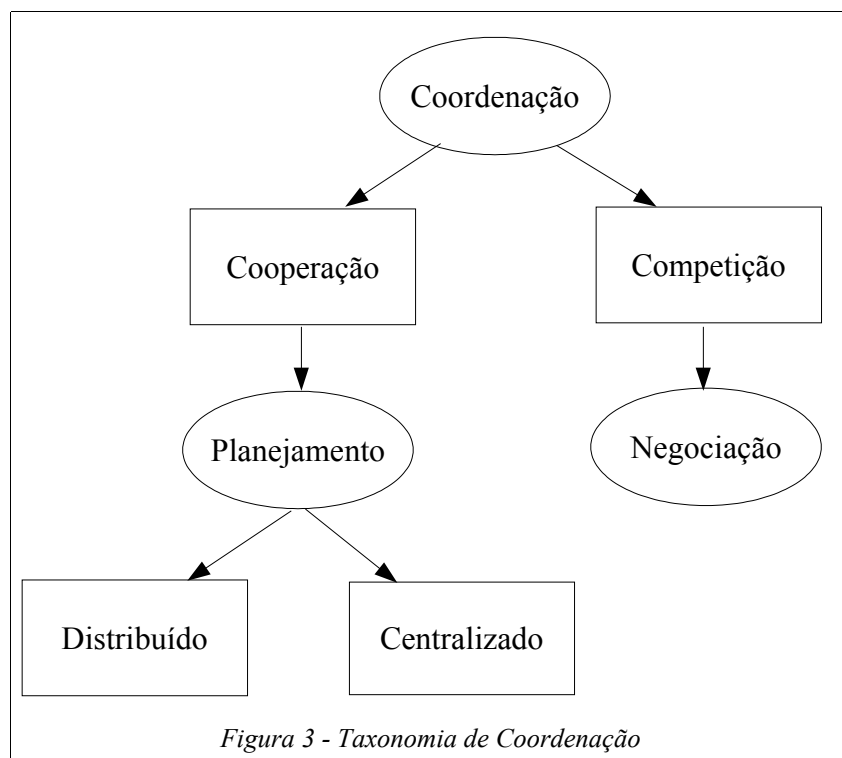
1. Funcional. Nesta arquitetura, cada agente representa uma única função que se comunica com as outras através do uso de um sistema de memória central e a comunicação entre elas é via uma relação de mestre e escravo, *master/slave*, onde uma função solicita a outra função e espera alguns resultados específicos;
2. Quadro de Aviso (*Blackboard*). Nesta arquitetura, existe uma memória central compartilhada por todos os agentes, o quadro de aviso, onde uma solução parcial é registrada. Esta solução pode ser melhorada e/ou completada por outros agentes. O maior problema desta arquitetura é a dificuldade de se manter o sincronismo da atualização do quadro de avisos para todos os agentes, podendo acontecer situações em que os agentes possuam cenários do mundo diferentes dos cenários que já foram atualizados. Além disso, o volume de informações globais pode ser muito grande, sendo inviável o tratamento das informações por um único agente;
3. Hierarquia. Na hierarquia existem diversos níveis de relações mestre/escravo (*master/slave*), onde o agente de um nível da hierarquia inferior são escravos dos agentes de um nível imediatamente superior;
4. Heterarquia. Na arquitetura heterarquia, os agentes se comunicam como amigos, sem nenhuma relação de mestre/escravo. Também não há nenhuma área central para se manter informações globais. As informações são as do próprio agente, portanto, locais.

Um fator relevante em sistemas multiagentes é a questão da coordenação dos agentes. A coordenação em sistemas multiagentes é definida como o processo no qual um agente é capaz de analisar as suas ações e as ações dos outros agentes a fim de tentar garantir um comportamento social coerente (JENNINGS, 1995).

A falta de algum tipo de coordenação acaba levando a perda total dos benefícios de um sistema descentralizado para resolução de um problema. Invariavelmente, a falta de coordenação acarreta a degeneração da sociedade, transformando-a em um conjunto caótico de agentes sem qualquer tipo de agrupamento (DURFEE, LESSER, CORKILL, 1987, 1989).

Segundo Huhns e Stephens (2000) existem dois tipos de coordenação, Figura 3:

1. **Cooperação.** A Cooperação é a coordenação entre dois ou mais agentes que possuem planos que convergem para um mesmo objetivo global, portanto, os agentes não possuem objetivos diferentes;
2. **Competição.** Na coordenação do tipo Competição, os agentes possuem seus próprios objetivos que nem sempre levam a um mesmo objetivo global da sociedade. Portanto, eles negociam entre si, para que se chegue num acordo.



Na coordenação do tipo Cooperação existe pelo menos um agente que, a partir de um plano global, designa outros planos menores e específicos para cada agente que é especialista na solução deste plano menor, (ver Figura 3). Citam-se a seguir dois dos mais importantes mecanismos usados na coordenação do tipo Cooperação:

1. **Rede de Contrato (*Contract Net*).** Um agente faz uma solicitação de um serviço ou recurso que necessita a diversos outros agentes que ele acredita poderem suprir sua necessidade e aguarda as ofertas de outros agentes, decidindo pela melhor oferta. Este processo ocorre em ciclos durante o período de negociação (SMITH, 1980) (SMITH, DAVIS, 1981);
2. **Quadro de avisos.** Neste mecanismo todos os agentes possuem informação completa sobre o ambiente que ficam registradas em um único quadro de avisos. Este mesmo quadro de avisos serve como canal de troca de informações.

No caso de coordenação tipo Competição o que é mais relevante é a negociação, (ver Figura 3). Os principais objetivos da negociação são a modificação dos planos do agente quando houver uma interação prejudicial a ele e a identificação de situações nas quais podem haver interações benéficas (MULLER, 1996).

A negociação numa ótica psicológica é definida como sendo o processo no qual uma decisão conjunta é realizada por pelo menos dois agentes. Inicialmente os agentes fazem propostas contraditórias e através de um processo de concessão e busca de alternativas eles vão fazendo novas propostas até se chegar num acordo (PRUITT, 1981).

Uma outra visão baseada no princípio da organização define a negociação como sendo a troca de informações por agentes envolvidos no processo que visa alcançar um acordo. Nesta definição são destacados três componentes principais: 1) Existe um canal de intercâmbio de informação de via dupla; 2) Cada agente avalia as informações sob sua própria ótica ; 3) O acordo final é alcançado através de uma seleção comum dos elementos do grupo (SMITH, DAVIS, 1981).

Ambas definições convergem para três pontos comuns e fundamentais para a negociação que são apresentados a seguir (O'HARE, JENNINGS, 1996):

1. Linguagem de Comunicação. Através da comunicação os agentes trocam as propostas de negociação visando o acordo final. Não é possível ocorrer negociação caso não seja possível a comunicação entre os agentes;
2. Modelo de decisão. Este modelo tem a função de modelar os objetivos, as preferências e as estratégias que o agente utilizará para a tomada de decisão;
3. Modelo de comportamento. Neste modelo são definidos os procedimentos que o agente pode executar e o comportamento que o agente possui.

Uma área de estudos recente que vem crescendo na área de agentes diz respeito a confiança (*trust*) e reputação (*reputation*). O conceito de confiança pode ser definido como a probabilidade subjetiva que um certo agente irá realizar uma certa ação requerida. Já o conceito de reputação diz respeito à expectativa que um agente tem em relação ao outro em função das informações passadas do comportamento deste outro agente (LIU, 2006) (MUI, 2002).

Estes dois conceitos são importantes, pois a partir deles, pode-se criar mecanismos nas sociedades de agentes com o intuito de se gerar ajuda mútua a partir da reputação que um certo grupo de agentes tenha. Isto é importante em questões comerciais quando o agente que vai fechar um negócio com outro decide se deve ou não confiar na oferta feita por este.

Na próxima seção é feita uma revisão do Problema de Escalonamento Dinâmico com Restrição de Recursos.

2.5) Problema de Escalonamento Dinâmico com Restrição de Recursos

O escalonamento é um processo de otimização no qual máquinas e recursos limitados são designados ao longo do tempo para atender diversas solicitações que são compostas de diversas atividades, denominadas tarefas. Esta designação é feita de tal maneira que sejam respeitadas integralmente as restrições de tempo de execução das atividades e o limite de capacidade do conjunto de máquinas e recursos usados por todas as atividades. O resultado desta designação afeta diretamente a qualidade do escalonamento no que diz respeito a critérios como: custo, atraso, capacidade de produção, dentre outros (ZWEBEN, FOX, 1994).

Normalmente, problemas de escalonamento quando tratados na prática, são problemas dinâmicos, pois sempre ocorrem imprevistos que obrigam a realização de novos planejamentos das atividades sempre que houver uma alteração no ambiente (BLAZEWICZ et. al., 1996).

Esta alteração do ambiente pode ocorrer por vários motivos, dentre eles citam-se: 1) Novas solicitações não previstas são introduzidas no processo; 2) Pode haver o cancelamento de solicitações que já fazem parte do planejamento; 3) Pode ocorrer atraso no início e/ou no término de algumas solicitações ou de suas tarefas; 3) Recursos e/ou máquinas já designados às tarefas das solicitações podem ficar indisponíveis; e 4) Novos recursos podem ser disponibilizados no sistema (SHEN, NORRIE, 1999).

O problema de escalonamento pode ser caracterizado como um problema no qual são feitas solicitações, *jobs*, que possuem diversas tarefas, *tasks*, que para serem realizadas devem fazer uso de uma ou mais máquinas ou processadores, *processors*, e que necessitam ou não de recursos, *resources* (BLAZEWICZ et. al., 1996). Assim sendo, o problema de escalonamento pode ser definido matematicamente a partir de quatro conjuntos:

1. Um conjunto $J=(j_1, j_2, \dots, j_q)$ de q solicitações. Cada solicitação j_i possui x_i tarefas de T sendo que $(\sum_{i=1}^q x_i) \leq n$ tarefas.
2. Um conjunto $T=(t_1, t_2, \dots, t_n)$ de n tarefas;
3. Um conjunto $P=(p_1, p_2, \dots, p_m)$ de m máquinas;
4. Um conjunto $R=(r_1, r_2, \dots, r_s)$ de s recursos.

Criar o escalonamento, significa alocar máquinas do conjunto P e recursos do conjunto R às tarefas do conjunto T a fim de completar as solicitações do conjunto J com todas as suas tarefas do conjunto T e suas respectivas restrições.

No entanto, como está sendo tratado o problema com restrição de recursos, faz-se a seguinte consideração: Existem s recursos r_1, r_2, \dots, r_s que estão disponíveis em quantidades m_1, m_2, \dots, m_s unidades respectivamente. Cada tarefa t_j necessita para seu processamento uma certa quantidade fixa de recursos que é registrada no vetor de recursos requeridos $R(t_j)=[r_1(t_j), r_2(t_j), \dots, r_s(t_j)]$, onde $r_i(t_j)$ determina o número de unidades de recursos

r_l que é necessário para executar a tarefa t_j e $r_l(t_j)$ deve obedecer a seguinte restrição: $(0 \leq r_l(t_j) \leq m_l)$.

Com base na formulação anterior, podem ocorrer dois tipos de restrições referentes aos recursos no problema de escalonamento: 1) Inexistência do tipo de recurso necessário e 2) Quantidade insuficiente de recursos de um tipo específico pelo tempo de execução da tarefa.

Para completar a definição do problema dinâmico tratado deve-se considerar que o conjunto T é dependente do tempo, $T(t)$.

No intuito de detalhar melhor o problema tratado, são definidos a seguir os processadores, os recursos e as tarefas.

Os processadores são classificados conforme o seu tipo de utilização. Assim, têm-se os seguintes tipos:

1. Paralelo (*Parallel*). Os processadores do tipo paralelo são aqueles que executam sempre as mesmas funções, independentemente das tarefas;
2. Dedicado (*Dedicated*). Os processadores do tipo dedicado ocorrem quando um processador é especializado em executar uma tarefa específica.

Os recursos podem ser classificados em função da quantidade de vezes que eles podem ser usados. Assim têm-se três categorias:

1. Recurso renovável (*renewable*). Ocorre quando uma tarefa usa o recurso por um período de tempo e depois outra tarefa pode usá-lo. Portanto, possui somente restrição de tempo de uso. Também, pode-se falar que uma tarefa usa um recurso e ao liberá-lo, ele contém a mesma quantidade que tinha quando a tarefa iniciou o seu uso. Como exemplo, têm-se: um guindaste, um carro, uma ferramenta, etc.
2. Recurso não renovável (*non-renewable*). Ocorre quando uma tarefa usa o recurso e o consome totalmente ou parcialmente e depois não é possível o uso parcial ou integral por outra tarefa do recurso que foi usado. Portanto, possui somente uma restrição de consumo. Também, pode-se falar que uma tarefa usa o recurso e, no seu término, ela terá consumido o recurso, não havendo a possibilidade de uso por outra tarefa. Como exemplo, têm-se: combustível, dinheiro, reagente químico, etc.
3. Recurso duplamente restrito (*doubly constraint*). Ocorre quando um recurso possui restrição de tempo e restrição de consumo.

Uma segunda classificação dos recursos pode ser feita com base na divisibilidade dos recursos. Por esta classificação podem ser identificadas duas categorias:

1. Recurso discreto (*discrete*). É aquele que só pode ser alocado de maneira discreta. Como exemplo, têm-se: um guindaste; um carro, etc. Não pode ser alocado parcialmente.

2. Recurso contínuo (*continuously*). É aquele que é alocado de maneira contínua, divisível. Como exemplo, têm-se: um quilo e trezentos gramas de cimento, dez litros e meio de combustível, etc.

Como última característica dos recursos, diz-se que os recursos são do mesmo tipo quando eles exercem a mesma função.

No problema tratado, cada tarefa t_i de T é caracterizada pelas seguintes informações:

1. Tempo de processamento. É o tempo p_{ij} que o processador p_i leva para processar a tarefa t_j ;
2. Tempo de chegada, *ready time* ou *arrival time*, a_j . Representa o tempo (t) no qual a tarefa t_j está pronta para iniciar o processamento;
3. Tempo limite de término, *due date*, d_j . É o tempo limite no qual cada tarefa t_j deve terminar seu processamento. Caso ele não seja cumprido, pode-se atribuir penalidades em função do atraso do término;
4. *Deadline* \tilde{d}_j . É equivalente ao tempo limite de término. No entanto não pode haver atraso no término sob nenhuma hipótese;
5. Prioridade, w_j . Expressa a prioridade de cada tarefa t_j tem em relação as outras tarefas;
6. Lista de recursos necessários.

As tarefas podem esperar entre o seu término em um processador e o seu início em outro processador, pois existe um *buffer* entre eles. Caso este *buffer* entre os processadores tenha capacidade igual a zero, as solicitações não podem esperar entre dois processadores consecutivos e, então, o problema é denominado como um problema com propriedade de não espera (*non-wait property*).

As tarefas podem ser, também, classificadas quanto a possibilidade de execução simultânea dela com outras tarefas e, assim têm-se:

1. Paralelas, quando são processadas simultaneamente, ao mesmo tempo, em processadores diferentes;
2. seqüências, quando as atividades só iniciam quando a anterior termina, independentemente do processador.

Caso as tarefas possam perder o direito de processamento (i.e. possam ser interrompidas) para que outra tarefa possa ser executada, diz-se que o problema de escalonamento é classificado como preemptivo, *preemptive*, e, caso não seja possível haver esta situação em todas as tarefas, então o problema é classificado como não preemptivo (*non-preemptive*).

Outra classificação que pode ser feita diz respeito à seqüência das tarefas. O símbolo \leftarrow representa sinal de precedência e, se no conjunto T ocorre $t_i \leftarrow t_j$, então, significa que t_j deve iniciar somente após t_i terminar. Esta situação é chamada de restrição de precedência e, caso T tenha pelo menos duas tarefas que sigam esta restrição, as tarefas são denominadas de dependente, *dependent*. Caso não haja esta restrição as tarefas são denominadas independente, *independent*.

A tarefa também pode ser classificada em relação a sua disponibilidade (*availability*). Uma tarefa t_j é dita disponível se num tempo (t) têm-se $a_j \leq (t)$ e todas suas antecessoras, caso haja restrição de precedência, tenham terminado de executar até o tempo (t) .

As tarefas do conjunto T num determinado tempo (t) devem respeitar as condições a seguir:

1. A cada instante, cada processador é designado para no máximo uma única tarefa e cada tarefa é processada por no máximo um processador. Esta restrição eventualmente pode ser relaxada;
2. A tarefa t_{ij} é processada numa janela de tempo $[a_j, \infty)$;
3. Todas tarefas devem ser executadas;
4. Se tarefas t_i e t_j estão na condição $t_i \leftarrow t_j$, o processamento de t_j não pode começar antes de t_i estar concluída;
5. Caso ocorra alguma restrição de recurso, elas devem ser satisfeitas.

Após as definições dos processadores, dos recursos e das tarefas, são definidas a forma de representação do problema de escalonamento e as possíveis medidas de desempenho.

Para cada tarefa t_j do conjunto T pode-se calcular diversos parâmetros de controle do escalonamento. A seguir são citados alguns dos parâmetros mais usados:

1. Tempo de término (*Completion Time*), C_j . Representa o instante que a tarefa terminou;
2. Tempo de processamento (*Flowtime*), $F_j = C_j - a_j$. É o resultado da subtração do tempo de término menos o tempo de chegada. Este tempo representa o tempo total que a tarefa gastou para processar;
3. Tempo de atraso (*Lateness*), $L_j = C_j - d_j$. É a subtração do tempo de término menos o tempo limite de término. Se L_j for negativo, ocorreu um processamento “bom” pois ele terminou antes do limite. Se L_j for positivo, ocorreu um processamento “ruim” pois ele ultrapassou o limite estabelecido. Esta última situação não poderia ocorrer no caso de *Deadline*;

4. Tempo mais tarde que o limite ou demora (*Tardiness*), $D_j = \max(C_j - d_j, 0)$. É a subtração do tempo de término menos o tempo limite de término. Os valores negativos não devem ser considerados pois não são um tempo mais tarde que o limite estabelecido;
5. Tempo mais cedo que o limite, ou adiantamento (*Earliness*), $E_j = \max(d_j - C_j, 0)$. É calculado como sendo a subtração do tempo limite de término menos o tempo de término. Os valores negativos não devem ser considerados pois não são um tempo mais cedo que o limite estabelecido.

Com base nos parâmetros citados anteriormente, pode-se avaliar a qualidade dos escalonamentos elaborados a partir de algumas medidas de *performance* citadas a seguir:

1. Tempo máximo de execução do escalonamento, *makespan*:
 - a) Para tarefas que podem ser executadas em paralelo calcula-se o *makespan* como:

$$C_{max} = \max(C_j) ;$$
 - b) Para tarefas que só podem ser executadas de forma seqüenciais calcula-se o *makespan* como:
$$C_{max} = \sum_{j=1}^n C_j ;$$
2. Média do tempo de processamento:
$$\bar{F} = 1/n \sum_{j=1}^n F_j ;$$
3. Média ponderada pela prioridade de cada tarefa
$$\bar{F}_w = \sum_{j=1}^n w_j F_j / \sum_{j=1}^n w_j ;$$
4. Máximo atraso:
$$L_{max} = \max(L_j) ;$$

Um escalonamento que tiver sua medida de performance no seu mínimo, pode ser dito que é um escalonamento ótimo.

Baker (1974) classifica o problema de escalonamento dinâmico com restrição de recursos descrito anteriormente como sendo *NP-Difícil*. As soluções para os problemas reais não são fáceis de serem alcançadas e mesmo soluções perto do ótimo são difíceis de serem alcançadas.

Após este capítulo, são apresentadas, no próximo capítulo, as soluções publicadas para a resolução do Problema de Escalonamento Dinâmico com Restrição de Recursos.

Capítulo 3 - Apresentação do Problema de Escalonamento e das Soluções já Existentes

O presente levantamento faz uma análise das contribuições científicas na área de conhecimento da abordagem proposta nesta tese. Para tanto, são pesquisados e analisados os trabalhos publicados em livros, teses e artigos científicos que tratam direta e/ou indiretamente do assunto. Assim sendo, três áreas de estudo são abordadas:

1. Aplicação de agentes para o problema de escalonamento/agendamento;
2. Aplicação de agentes direcionado a problemas logísticos e ao estudo de caso do Terminal Marítimo de Contêiner;
3. Uso de emoção para elaborar protocolos de negociação entre agentes.

Inicialmente, quando se fala sobre o uso de agentes para o problema de escalonamento, são citados os trabalhos de Shen, Norrie (1999) e Baker (1998), nos quais ambos fazem uma revisão do uso da abordagem de agentes para resolver o problema de escalonamento.

Shen, Norrie (1999) tratam dos diversos problemas de manufatura/produção, incluindo neste contexto os trabalhos de escalonamento de uma maneira geral. Eles divulgam o resultado de uma pesquisa sobre as técnicas que tem sido usadas na área de agentes aplicadas aos problemas nas áreas de produção, *Supply Chain Management*, planejamento da produção, escalonamento, dentre outras. Shen, Norrie (1999) apresentam uma tabela onde são listados trinta trabalhos relacionados com agentes e escalonamento.

Baker (1998) faz uma boa revisão sobre agentes e ambientes multiagentes e trata mais especificamente da arquitetura de ambientes multiagentes denominada heterarquia, a mesma adotada nesta tese.

Pode-se dizer que um dos primeiros trabalhos na área de escalonamento e chão de fábrica foi o elaborado por Shaw e Whinston (1983) que divide a fábrica em células e estas células podem subcontratar outras para fazer o serviço. Uma continuação deste trabalho foi elaborada por Shaw (1988).

Um dos trabalhos pioneiros na área é o elaborado por Parunak (1987) que propõe o sistema YAMS, baseado em agentes e se utiliza do protocolo de negociação CNP (*Contract Net Protocol*) (SMITH, 1980) (SMITH, DAVIS, 1981). Apesar de ser um trabalho pioneiro, ele é mais voltado para o problema de *shop floor*, mostrando que a cada serviço apresentado para a fábrica, o sistema tem que decidir, no mesmo momento, quais equipamentos devem ser alocados para atender à solicitação. Parunak (2000) faz um resumo de todas as aplicações baseadas em agentes para área industrial desde seu primeiro trabalho em 1987.

Burke e Prosser (1991) propõem um sistema com arquitetura DAS (*Distributed Asynchronous Scheduling*) baseado em agente para criar o escalonamento onde o problema é decomposto por função e por área através de agentes hierarquicamente classificados para se comunicarem. Cada agente deste sistema apresenta características de oportunismo, reação e crenças. Cada agente corresponde a um processo do programa, que permite até todos rodarem simultaneamente. Para o sistema caminhar em direção à solução global, cada agente reage a mudanças impostas a ele por outros agentes e então ele negocia com estes agentes. O mundo externo é visto como um agente, só que não é habilitado a negociar no sistema.

Sycara e outros (SYCARA , 1991) divulgam um método pelo qual o escalonamento é feito por agentes que não possuem uma visão global do sistema, têm conhecimento limitado do meio no qual estão inseridos e têm um número limitado de recursos, sendo que alguns destes recursos podem ser compartilhados. A maior contribuição do trabalho é a introdução da troca de uma estrutura, denominada *texture* no artigo em inglês, que tem a função de gerar a coordenação dos agentes que estão fazendo o escalonamento. Nestas estruturas estão contidas informações sobre como os agentes podem aceitar e aderir às políticas globais para solução do problema.

Neiman e outros (NEIMAN , 1994) descrevem um sistema para escalonamento dos recursos de um aeroporto, visando alocar os recursos para atender os aviões que chegam. O trabalho é baseado no trabalho de Sycara e outros (SYCARA, , 1991) e são feitas algumas modificações. O artigo baseia-se em uma técnica pouco usada atualmente, chamada quadro de aviso. Nesta técnica o escalonamento parcial, não definitivo, é mantido e atualizado por todos os outros agentes em uma área comum, o quadro de avisos, até se chegar ao escalonamento final do recurso.

Liu e Sycara (1994) propõe uma técnica denominada CP&CR (*Constraint Partition and Coordinated Reaction*). O método funciona dividindo os agentes em duas classes: Agentes de Recurso e Agentes de Solicitação (*Job*). Enquanto o primeiro tem por função garantir as restrições de cada recurso, o segundo tem por função manter as restrições das solicitações, precedência e data de término. Cada tarefa do problema de escalonamento é sujeita a ambos os agentes que podem alterar a tarefa a qualquer momento.

Liu e Sycara (1995) apresentam uma técnica denominada *Anchor & Ascend* que é uma extensão do método CP&CR (LIU, SYCARA, 1994). A diferença entre o CP&CR e o *Anchor & Ascend* é que no caso do segundo, ele fixa um recurso denominado crítico e a partir deste ponto ele começa a buscar uma solução.

A idéia central do trabalho de Decker e Li (2000) é um sistema para agendamento de paciente em um hospital através do método GPGP (*Generalized Partial Global Planning*) usando agentes cooperativos, o que não é foco desta tese.

Thurston e Hu (2002) usam agentes para resolver o problema de automação portuária. No entanto, eles definem cada agente como sendo um equipamento (no caso os agentes são de um dos três tipos: a) *Quay Crane*; b) *Straddle Carrier* e c) *Traffic*. Além destes existe um outro tipo chamado *Area Manager* que controla as áreas de estocagem, a área é subdividida em sub áreas. O sistema funciona usando o protocolo *Contract Net* (SMITH, 1980) e o agente do tipo *Quay Crane* solicita ofertas aos agentes *Area Manager* para receber os contêineres desejados.

Rebollo, Carrascosa e Botti (2001, 2000(a)(b), 1999) tratam de um sistema para automação de um terminal de contêiner baseado em agentes. Eles dividem o sistema em agentes: *Ship Agent*, *Stevedore Agent*, *Service Agent*, *Transtainer Agent*, *Gate Agent*. Os agentes, através do protocolo *Contract Net*, negociam para estabelecer o agendamento do terminal.

Henesey, Wernstedt e Davidsson (2003, 2002) apresentam um sistema que também divide o terminal portuário em agentes: *Ship Agent*, *Berth Agent*, *Yard Agent*, *Gate Agent*, e eles negociam através do *Contract Net* (SMITH, 1980) para gerar o melhor planejamento do terminal.

Os três trabalhos citados diferem substancialmente da abordagem desta tese, sobretudo pelo uso do protocolo de negociação *Contract Net*.

Diversos trabalhos foram propostos para solucionar o Problema do Terminal Marítimo de Contêiner. No entanto, estes trabalhos se valem de outras abordagens, tais como: Simulação, Métodos matemáticos, Teoria de Sistemas entre outras. Dentre eles, citam-se: Cruz (1993, 1997), Crainic (1993), Castilho, Daganzo (1993), Hee, Wijbrands (1988), Hee, Huitinick, Legwater (1988), Lorenzoni (2003), Perret (1991), Rudnick (1991), Talleb-Ibrahimi (1989), UNCTAD (1994 (a)(b)).

Os trabalhos em logística utilizando agentes são mais direcionados para a integração dos elementos da cadeia de suprimentos do que propriamente dito para a elaboração de agendas comuns. Nesta linha destaca-se o trabalho de Singh e outros (SINGH, SALAN, IYER, 2005) que propõe uma rede de agentes que atuam em diversos segmentos da logística buscando suprir as demandas das empresas por produto e prospectar novos mercados.

Na área do estudo de caso do Terminal Marítimo de Contêiner destacam-se os trabalhos a seguir:

Reis e Mamede (2001 (a)(b)) trabalham com agentes cooperativos para resolver o problema de escalonamento dinâmico. Eles propõem uma rede de agentes “*agent network*”, onde em uma ponta da rede estão os fornecedores e na outra os clientes e, o que ocorre no meio, os nós da rede, são os recursos. A proposta é dividida em três etapas: 1) Se o problema tem restrição de horário, ele rejeita, senão estabelece uma solução inicial; 2) Se tinha restrição na etapa 1, ele faz o reescalamento, e então define uma solução inicial sem restrição de tempo; 3) Executa o sistema para restrição de recurso, se houver restrição, ele reescala até achar uma solução viável.

Shen, Zhang e Lesser (2004) apresentam um trabalho baseado em agentes cooperativos, em um ambiente evolucionário, onde as condições de negociação e objetivos mudam constantemente, e portanto, podem dificultar o agente ter informações não locais sobre o ambiente, o que o faz ter dificuldade de agir cooperativamente. É, então, proposto um modelo estatístico para analisar de maneira formal a relação entre cooperação local, as características do ambi-

ente e a função de utilidade global. Com isto, pode-se medir o impacto do nível de cooperação local em relação a cooperação global.

A maior parte destes trabalhos caminham para duas vertentes:

1. Cooperação, problemas DSP, cuja característica é o desmembramento de uma solicitação em diversas atividades e cada agente executa a sua parte, e através de mecanismos de coordenação, chega-se ao resultado final o mais próximo do ótimo.
2. Sistemas cooperativos cuja característica principal é o uso do protocolo *Contract Net* (SMITH, 1980) para gerar ofertas aos agentes que coordenam os recursos e selecionar aqueles que dão a melhor proposta.

Kraus (2001) propõe usar agentes autônomos para resolver problemas de alocação de recursos. No entanto, o problema resolvido trata de agentes físicos, robôs, que devem buscar recursos que estão no meio ambiente, não são próprios dos agentes, para exercer alguma atividade e, caso dois deles desejem o mesmo recurso, eles devem negociar para saber quem vai usar o recurso. No caso desta tese, os recursos são de propriedade dos agentes e não são cedidos a outros agentes, sendo simplesmente agendados para exercer alguma função do próprio agente. Apesar de não ser escalonamento com função de planejamento, o conceito de se usar agentes está mais próximo da idéia proposta nesta tese. Trabalho similar ao de Kraus (2001) foi apresentado também por Sherory e Kraus em 1998.

Poucos trabalhos foram publicados sobre a proposta de se elaborar um protocolo de negociação inspirado nas emoções humanas que seja reproduzido no computador. No entanto, diversos artigos versando sobre emoção humana e sua aplicabilidade à computação foram escritos, principalmente com foco na área de robótica.

A área de interface agente versus homem (homem-máquina) foi estudada por Moldt e Scheve (2001).

Reilly e Bates (1995) propõe uma simulação em mundo virtual como uma forma de negociação denominada natural, da qual os seres virtuais poderiam se valer para viverem neste mundo virtual. Na verdade, é mais utilizado como interação social, portanto, mais do que um protocolo de negociação, como citado no próprio texto.

Alguns trabalhos foram escritos no sentido de se mensurar emoções e tratá-las computacionalmente. Nesta linha, estão os trabalhos de Velásquez (1998), Gmytrasiewicz e Doshi (2005) Gmytrasiewicz e Lisetti (2000), Nemani e Allan (2003).

O trabalho de Adamati e Bazzan (2002) trata de emoções em agentes simulando um mundo virtual, o que é destacado no trabalho é a proposta de se modelar emoções e tratá-las. Outro aspecto é que neste artigo é feita menção à emoção gratidão, o que é muito pouco tratado nas publicações que lidam com as emoções em agentes.

Doshi e Gmytrasiewicz (2004) tratam de mundos virtuais e como as emoções afetam o comportamento dos agentes neste mundo. Apesar da pesquisa não ser diretamente relacionada com o tema desta tese, ela apresenta em seu contexto a possibilidade de se expandir, saindo do mundo virtual, e ser aplicada em mundos reais.

O trabalho de Gratch e Marsella (2004) não se refere diretamente ao uso de emoções em protocolos de negociação, mas apresenta uma proposta de modelagem de emoções e seu impacto no comportamento.

Pode-se perceber, também, que a utilização das idéias inspiradas em emoções como mecanismo de negociação e de resolução de conflito não foram tratadas até o momento em publicações científicas. A utilização das idéias de emoção sendo usadas para problemas de escalonamento também não foram divulgadas na área científica.

Portanto, conforme mostrado na revisão do estado da arte, os dois objetivos desta tese são inovadores, não se conhecendo até o presente momento artigos versando sobre o uso de agentes em ambientes multiagentes para a solução do problema de escalonamento dinâmico com restrição de recursos, bem como, a utilização de protocolos de negociação e de resolução de conflitos inspirados em emoções humanas.

Por fim, na área relacionada à Confiança e Reputação em sociedade de agentes, alguns trabalhos foram publicados, sobretudo, dando ênfase à área de negócios eletrônicos e redes P2P (DAMIANI et. al., 2002) (MIR, 2002) (MUI, 2002) (WANG, 2004) o que não tem diretamente ligação com os objetivos desta tese. Outros trabalhos propõe critérios para medição e formalização computacional destes conceitos (CASARE, 2005) (LIU, 2006).

Apesar destes conceitos não serem usados diretamente na abordagem proposta nesta tese, os mesmos são implícitos na mesma quando se admite que os agentes são sempre confiáveis e não blefam, enviando mensagens para enganar os outros agentes.

No próximo capítulo, é apresentada a abordagem proposta nesta tese baseada na negociação de agentes.

Capítulo 4 - Uma Abordagem Baseada em Negociação de Agentes

Este capítulo apresenta a abordagem proposta nesta tese baseada em negociação de agentes e o estudo de caso do Terminal Marítimo de Contêiner.

4.1) Definição da Abordagem Proposta

A abordagem proposta para resolver a elaboração da agenda de atendimento as solicitações de serviço é fundamentada na tecnologia de agentes situados em sistemas multiagentes virtuais.

Os agentes são concebidos com a característica de terem seus próprios objetivos e de não conhecerem os objetivos globais do sistema. Apesar disto, os agentes são confiáveis e não tentam enviar mensagens no intuito de enganar o estado do outro agente.

O sistema multiagentes é concebido como tendo uma arquitetura do tipo heterárquica na qual cada agente se comunica com os outros agentes numa relação de amizade e companheirismo, não havendo entre os agentes nenhuma relação hierárquica.

A coordenação no sistema multiagentes é estruturada na forma de competição e, desta forma, o problema é definido como Sistemas Multiagentes, *MAS (Multiagent System)*.

A abordagem proposta não considera a hipótese de resolução de conflito entre os agentes através de terceiros. Portanto, os agentes devem resolver seus conflitos através da negociação. Para que haja negociação, os agentes devem se comunicar e interagir trocando propostas e contra propostas.

Não existe no sistema multiagentes uma área central para repositório de informações e/ou mensagens e, portanto, os agentes devem trocar mensagens a fim de informar as propostas e contra propostas. O agente possui somente como informação do outro agente as propostas recebidas e seu estado emocional em relação a ele mesmo, este assunto é discutido mais a frente nesta mesma seção.

Como um dos objetivos da abordagem é estudar a viabilidade da negociação como o fator que leve o sistema multiagentes à solução do problema, foram elaborados dois padrões de negociação que serão explicados em mais detalhes neste capítulo:

1. A solicitação que tiver a menor data de solicitação será a primeira a ser atendida;
2. Caso haja duas solicitações com a mesma data de chegada, o critério de escolha é inspirado na emoção humana.

O padrão de negociação inspirado nas emoções humanas tem o propósito de flexibilizar a característica competitiva de cada agente, fazendo-o se tornar mais cooperativo e servindo como fator de resolução de conflitos.

O ser humano aceita fazer algumas concessões ao outro ser humano em função da gratidão que ele sente por ter sido um dia beneficiado por alguma ação do outro ser. Assim, por ter gratidão por este benefício, o indivíduo tenta, sempre que possível, retribuir esta ajuda, flexibilizando suas posições a fim de ajudar aos que mais o ajudaram. Esta prática social é aplicada ao mecanismo de negociação proposto. O uso da emoção no mecanismo de negociação não visa replicar a emoção humana, mas sim, usá-la como base de uma idéia a ser aplicada no sistema multiagentes.

Esta tese explora a idéia do sentimento de amizade, de participar de uma comunidade e, apesar de cada agente ser competitivo e individualista, como o ser humano, eles tentam de alguma forma ajudar os outros agentes do sistema multiagentes no intuito de se tornarem elementos ativos na sociedade e serem merecedores da ajuda dos outros agentes.

Para efeito da abordagem proposta são descartadas as emoções de caridade e benevolência nas quais cada elemento da sociedade busca ajudar o próximo sem interesse de retorno da ajuda concedida. No caso específico desta tese, todo agente quando presta uma ajuda aguarda uma contra ajuda como retorno de seu esforço. Esta contra ajuda não precisa ocorrer imediatamente após a ajuda recebida, podendo ficar registrada e ser retribuída posteriormente.

Este mecanismo de flexibilização e de resolução de conflito é inspirado na emoção gratidão e denominado na abordagem como *Grau de Gratidão*. No mecanismo proposto, cada agente informa ao outro agente o esforço que ele realizou para atender a solicitação feita, esta informação é enviada através de uma mensagem.

O esforço é medido pelo número de solicitações já agendadas que terão suas datas de solicitação reprogramadas em função do atendimento desta nova solicitação do outro agente com o qual ele está negociando. O crédito e débito destas reprogramações é que vai levar o agente a decidir se fará mais ou menos esforço para atender a solicitação. Este mecanismo é explicado em mais detalhes em seções subsequentes deste capítulo.

O *Grau de Gratidão* proposto induz o agente a mudar sua abordagem de Competição para a abordagem de Solução de Problema (ver Figura 2). Apesar desta mudança, ele continua olhando muito para seus interesses, mas através da inserção do mecanismo *Grau de Gratidão* ele olha, também, para o interesse dos outros agentes.

4.2) *Funcionamento da Abordagem Proposta*

Nesta seção é apresentada uma visão geral e esquemática de como a abordagem proposta funciona e nas seções subseqüentes é feita a modelagem matemática do seu funcionamento.

O agente ao *nascer*, isto é, ao ser inicializado, no sistema multiagentes virtual contém as seguintes informações:

1. Um conjunto de regras que estabelece o que é o seu objetivo e como ele deve negociar para alcançar estes objetivos. Fazendo uma analogia com o ser humano pode-se dizer que estas regras formam a *inteligência* do agente;
2. Uma tabela de relacionamentos (*acquaintances*). Esta tabela contém os outros agentes do sistema multiagentes com os quais o agente se relaciona. Assim sendo, o agente sabe de quais agentes ele pode receber solicitações e para quais agentes ele necessita solicitar serviços e recursos a fim de realizar suas tarefas. Esta tabela pode ser alterada dinamicamente;
3. Recursos próprios e não compartilhados. Cada agente pode ou não possuir recursos próprios, mas uma vez que ele seja o proprietário deste recurso, não haverá outro agente proprietário do mesmo recurso.

Para efeito de definição desta abordagem, considera-se o mundo como o ambiente virtual onde estão situados os agentes, ou seja o próprio sistema multiagentes e o mundo externo como sendo o mundo real onde vivem os seres humanos e as empresas que podem fazer solicitações ao sistema multiagentes.

Todas as solicitações que vêm do mundo externo necessitam de diversos recursos e serviços. Não existe, *a priori*, um agente que possua todos os recursos e serviços necessários para resolver sozinho uma solicitação e, assim sendo, para que a solicitação seja atendida é necessário que diversos agentes atuem em conjunto fornecendo seus produtos e serviços a fim de realizar a solicitação.

Para que seja possível encontrar uma data única e comum a todos os agentes, deverá ocorrer entre os agentes um processo de negociação entre eles.

Em relação ao sistema multiagentes proposto define-se que ele pode estar em dois estados:

1. Equilíbrio. Este estado ocorre quando todas as solicitações ao sistema multiagentes estão agendadas/reservadas para uma data única e comum a todos os agentes que necessitam fornecer recursos e serviços para a solicitação;
2. Desequilíbrio. Este estado ocorre quando pelo menos uma solicitação não tenha sido agendada numa data única e comum a todos os agentes necessários para sua realização. Isto implica dizer que os agentes ainda não concluíram o processo de negociação.

O sistema multiagentes terá como base de seu funcionamento a mudança constante de estados. Isto é, sempre que o sistema multiagentes estiver no estado de Equilíbrio e receber uma

nova solicitação do mundo externo, ele muda para o estado de Desequilíbrio. O sistema multi-agentes inicialmente encontra-se no estado de Equilíbrio pois não há nenhuma solicitação do mundo externo.

Ao mudar para o estado de Desequilíbrio, os agentes começam a negociar entre si e à medida que a negociação avança, o sistema vai voltando para o estado de Equilíbrio, sendo este alcançado quando é encontrada uma data única e comum a todos os agentes envolvidos em resolver a solicitação. Quando esta data é encontrada, o sistema retorna para o estado de Equilíbrio.

Este processo é dinâmico pois sempre estão chegando novas solicitações, ou as já agendadas são alteradas, e, portanto, a abordagem funciona sempre mudando para o estado de Desequilíbrio e, através da negociação, retorna ao estado de Equilíbrio.

Ao receber uma solicitação de serviço, o agente verifica quais recursos pode dispor para atender a solicitação na data solicitada. Para verificar se possui os recursos, ele analisa, para cada recurso, a possibilidade de atender na data da solicitação. Caso exista algum recurso que possa atender na data solicitada, este recurso será reservado para atender a solicitação.

Caso todos os recursos não estejam disponíveis na data solicitada, então é feita a análise em qual data existem todos os recursos necessários para operação. É feita, também, a análise do impacto em termos do número de solicitações que terão suas datas previamente agendadas alteradas e a soma dos tempos de cada alteração desta agenda. O recurso que sofrer o menor impacto, terá sua agenda alterada para acomodar a nova solicitação.

Atualmente, não está implementado na abordagem, mas é factível acrescentar outros critérios para desempate entre os recursos que terão sua agenda alterada. Dentre vários, citam-se: importância de cada cliente para a corporação, receita de cada solicitação, tempo de relacionamento do cliente com a empresa, dentre outros.

Caso ele possua recursos suficientes, podem ocorrer duas situações: 1) Ele precisa, além de seus recursos, outros recursos que somente outro agente possui e, portanto, ele deve solicitar ao outro agente tais recursos; 2) Ele não precisa solicitar nenhum recurso a outro agente para realizar a solicitação.

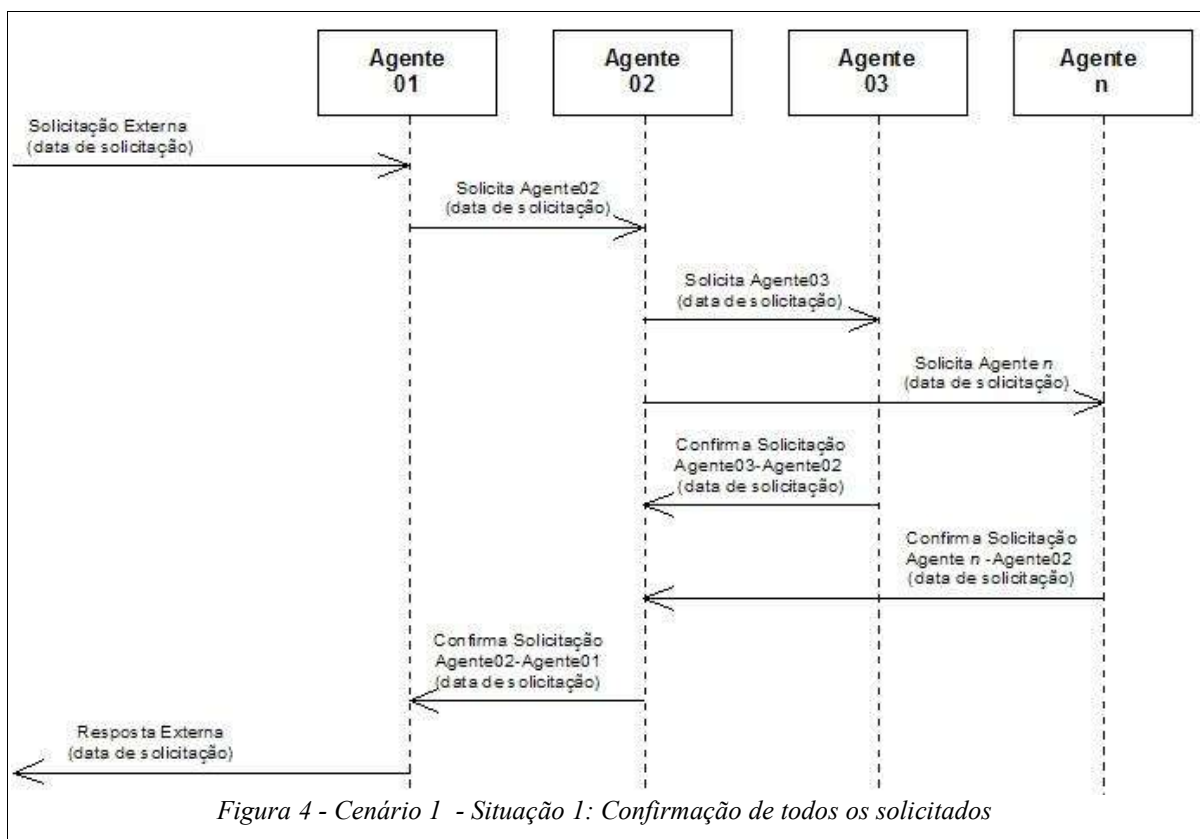
Para entender este funcionamento deve-se entender o funcionamento de cada agente. O agente ao receber a solicitação analisa se é possível atender na data solicitada. Desta análise, duas situações podem ocorrer:

1. O agente pode atender na data solicitada. Caso ele possa atender parcialmente através de seus recursos a solicitação, ele reserva seus recursos para a data solicitada e solicita aos outros agentes que façam a reserva dos recursos necessários para a execução da solicitação. Estes agentes aos quais ele necessita solicitar recursos estão descritos na sua tabela de relacionamentos;
2. O agente não pode atender na data solicitada. Nesse caso ele verifica uma data que seja possível atender com seus recursos e com esta nova data sugerida ele reserva seus recursos para esta data e solicita aos outros agentes que façam a reserva dos recursos necessários para a execução da solicitação na nova data e envia, também, ao solicita-

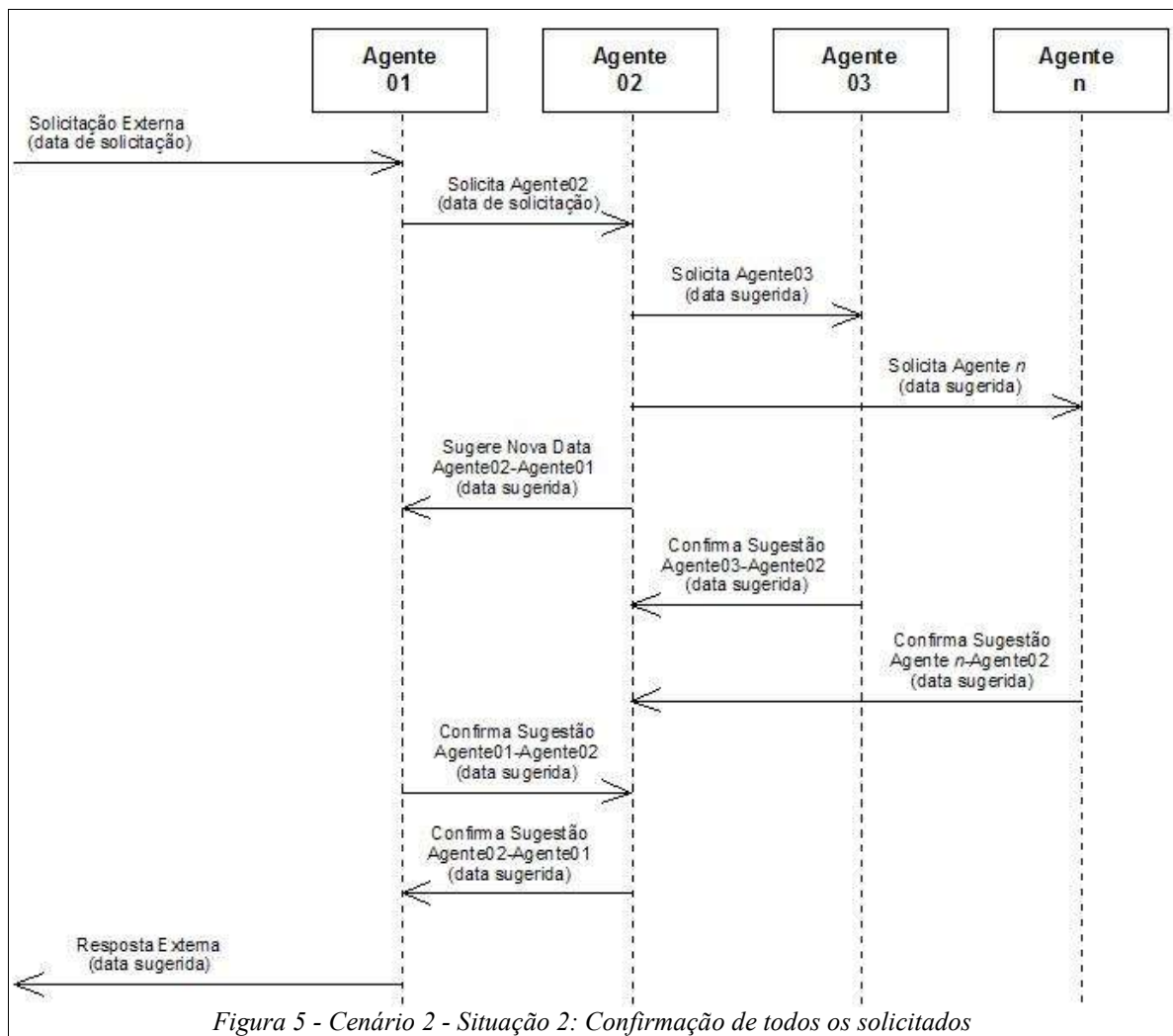
dor uma mensagem de sugestão da nova data. Estes agentes aos quais ele necessita solicitar recursos estão descritos na sua tabela de relacionamentos.

Após o agente ter enviado as solicitações e, quando for o caso, a mensagem de sugestão para o solicitador com a nova data, ele fica aguardando a resposta dos agentes aos quais ele enviou mensagem. Em função das respostas recebidas e das duas situações descritas anteriormente, podem ocorrer cinco cenários possíveis. As figuras a seguir consideram quatro agentes, sendo o último denominado n , pois as situações ocorrerão do mesmo modo independente do número de agentes.

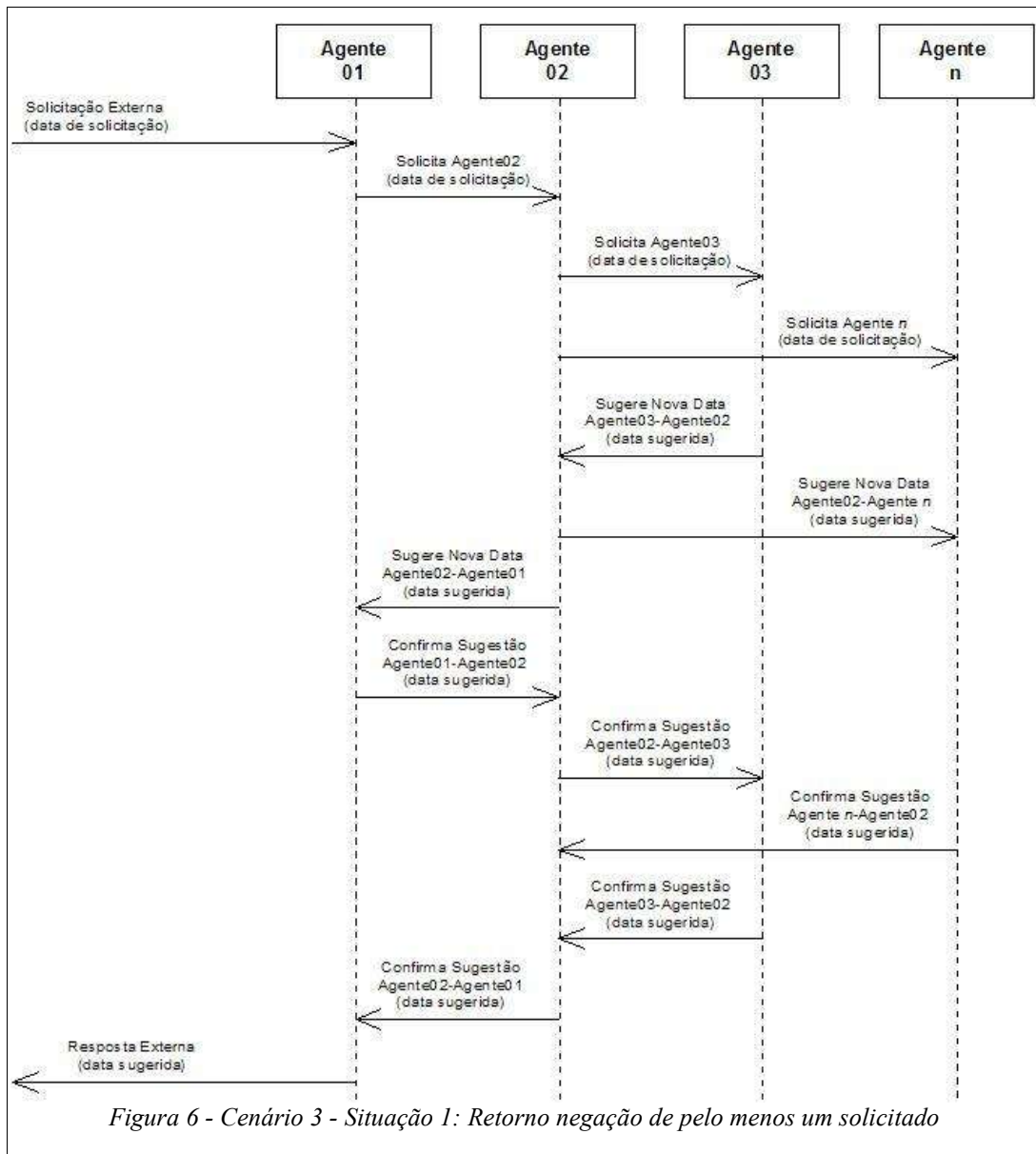
1. Quando ocorrer a situação 1 e todos os agentes solicitados responderem confirmando a solicitação, o agente simplesmente retorna ao solicitador, agente ou mundo externo, e aos agentes solicitados confirmando a data anteriormente solicitada (Figura 4);



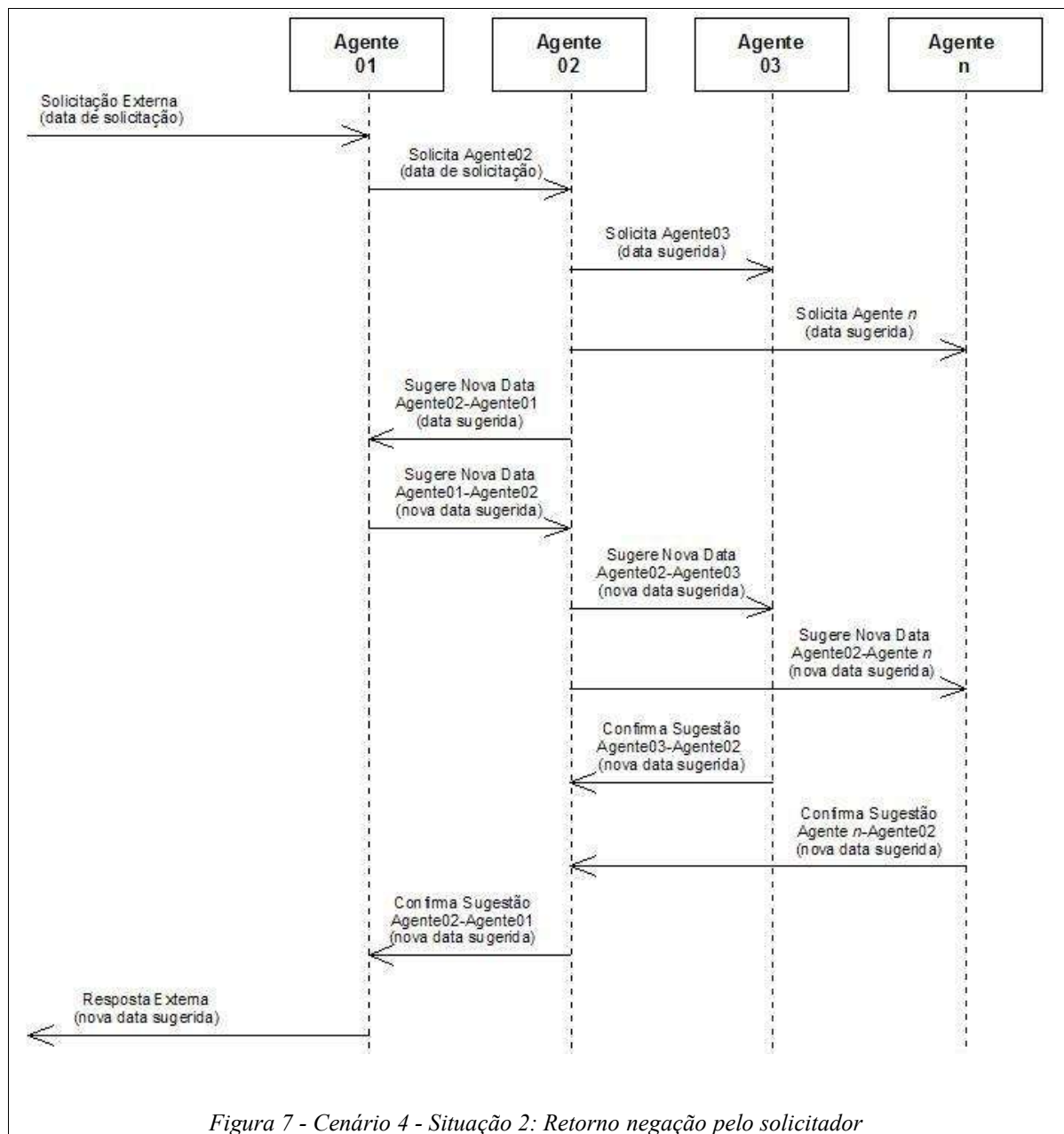
2. Quando ocorrer a situação 2 e todos os agentes solicitados e o solicitador responderem confirmando a solicitação, o agente simplesmente retorna ao solicitador, agente ou mundo externo, e aos agentes solicitados confirmando a data anteriormente solicitada, (Figura 5);



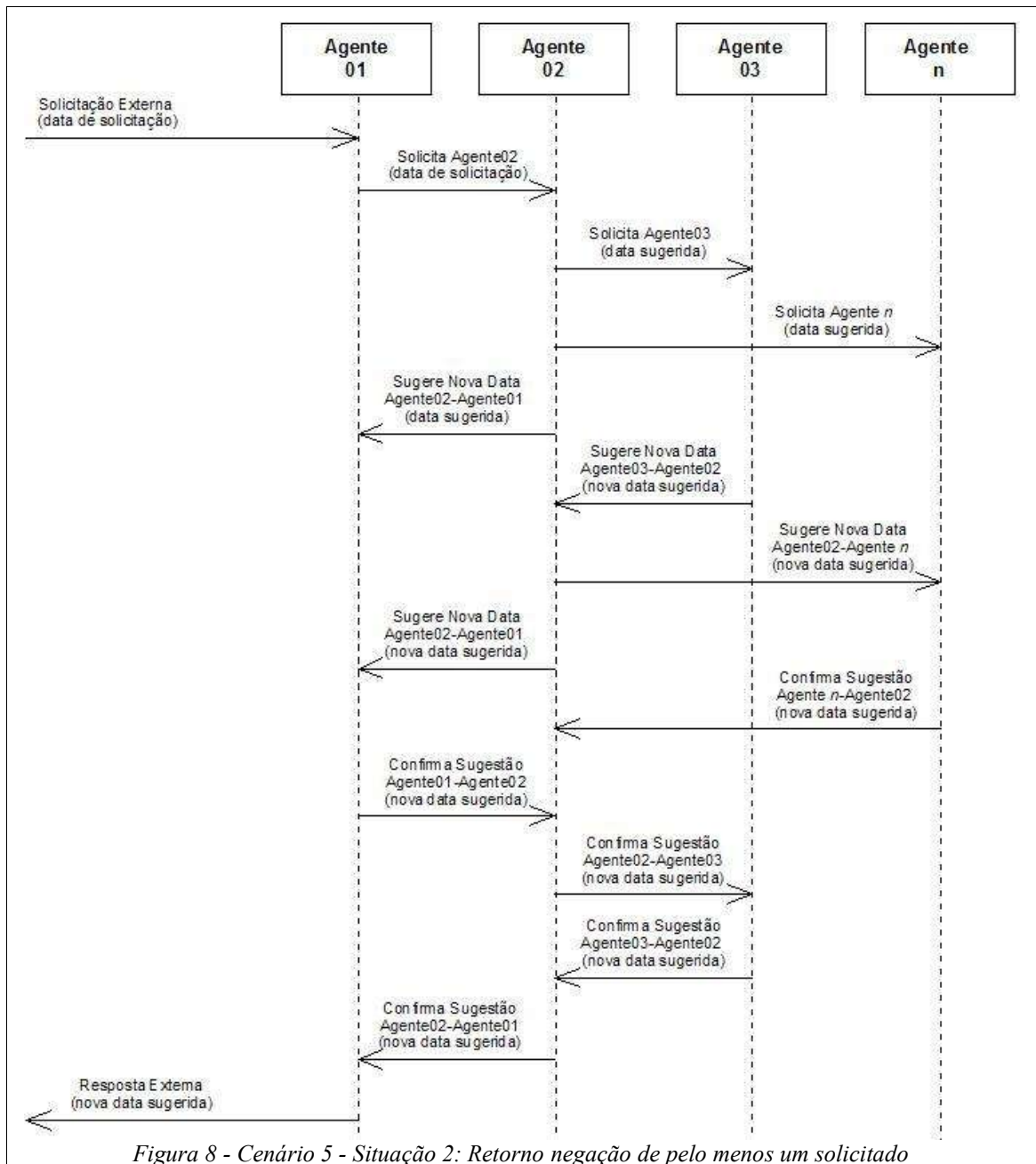
- Na situação 1 pode ocorrer o retorno de pelo menos um agente solicitado informando que não pode atender na data solicitada e enviando uma sugestão de uma nova data posterior à data solicitada. Neste caso, o agente aceita esta data posterior como a data possível e retorna a todos os solicitados e para o solicitador uma sugestão para esta nova data e fica no aguardo da confirmação por todos da possibilidade de atendimento nesta nova data (Figura 6);



- Na situação 2 pode ocorrer o retorno pelo solicitador informando que não pode aceitar a data solicitada e enviando uma sugestão de uma nova data posterior à data solicitada. Neste caso, o agente aceita esta data posterior como a data possível e retorna a todos os solicitados uma sugestão para esta nova data e fica no aguardo da confirmação por todos da possibilidade de atendimento nesta nova data (Figura 7);



5. Na situação 2 pode ocorrer o retorno de pelo menos um agente solicitado informando que não pode atender na data solicitada e enviando uma sugestão de uma nova data posterior à data solicitada. Neste caso, o agente aceita esta data posterior como a data possível e retorna a todos os solicitados e para o solicitador uma sugestão para esta nova data e fica no aguardo da confirmação por todos da possibilidade de atendimento nesta nova data (Figura 8);

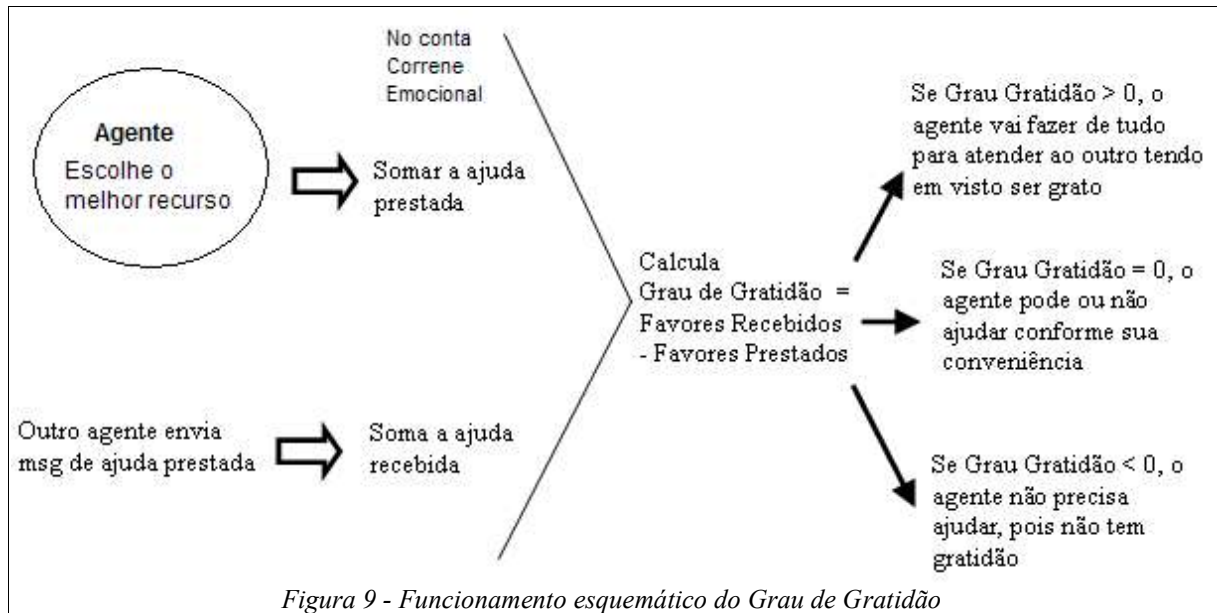


Como pode ser observado nos cenários 3, 4 e 5, o retorno nestas condições gera um novo estado de Desequilíbrio e força a negociação entre os agentes a prosseguir até que se consiga chegar ao estado de Equilíbrio.

A parte mais crítica que cada agente tem de resolver é quando ele já possui diversas solicitações agendadas e ao chegar uma nova solicitação ele deve decidir se altera a data da solicitação previamente agendada a fim de acomodar a nova solicitação ou não. Se ele não conseguir atender a solicitação na data solicitada, ele deve propor uma nova data posterior à data solicitada para esta nova solicitação.

O protocolo de negociação com *Grau de Gratidão*, proposto nesta tese, funciona como uma conta corrente dentro do agente. Nesta conta corrente são registrados os esforços que ele rea-

lizou para atender determinado agente e, também, são registrados os esforços que o outro agente fez para atendê-lo. A subtração do valor da ajuda recebida do outro agente menos o valor da ajuda prestada ao outro agente será o valor do *Grau de Gratidão* (Figura 9).



A situação na qual o *Grau de Gratidão* é muito próximo ou igual a zero, representa o equilíbrio emocional do ser humano transposto para o sistema artificial e é o ponto ideal, onde todos ajudam e são ajudados, trazendo, assim, para o sistema multiagentes o conceito de cooperação.

Na seção seguinte, é apresentada a modelagem matemática da abordagem proposta.

4.3) Modelagem Matemática da Abordagem Proposta

Tendo em vista o problema de elaboração de agenda para atendimento às solicitações em ambientes dinâmicos e complexos ser modelado como o Problema de Escalonamento Dinâmico com Restrição de Recursos e a abordagem proposta ter sido elaborada usando a tecnologia de agentes e sistemas multiagentes, deve-se propor uma correlação entre o problema de escalonamento e o sistema multiagentes para o bom funcionamento da abordagem. A equivalência proposta é apresentada na Tabela 1.

Problema de escalonamento	Sistema multiagentes
Solicitação	Solicitação de serviço externa ao agente
Tarefa	Mensagem de solicitação enviada aos outros agentes
Processador	O agente
Recursos	Os recursos que cada agente possui

Tabela 1 - Equivalência do problema de escalonamento com o sistema multiagentes

Nesta equivalência, cada solicitação de serviço feita pelo mundo externo é equivalente à solicitação do problema de escalonamento, as tarefas do problema de escalonamento são vistas no sistema multiagentes como as mensagens de solicitação de um agente para o outro agente do sistema e o processador é modelado como o próprio agente. Esta correlação foi feita pois o agente é que será responsável por cada uma das tarefas. Por fim, os recursos do problema de escalonamento são os próprios recursos de cada um dos agentes.

Para início da modelagem matemática, o agente é definido por um grupo de conjuntos:

1. Um conjunto $J=(j_1, j_2, \dots, j_q)$ de q solicitações, sendo que cada solicitação j_u é composta de x_u tarefas t_j ;
2. Um conjunto $T=(t_1, t_2, \dots, t_n)$ de n tarefas, sendo n calculado como $n=\sum_{u=1}^q x_u$. Cada tarefa t_j refere-se a uma e somente uma solicitação j_u e é executada por um e somente um agente especializado em atender um tipo de tarefa;
3. Um conjunto $R=(r_1, r_2, \dots, r_s)$ de s recursos próprios e não compartilhados.

Cada agente deve processar de maneira autônoma a melhor agenda que ele possa conseguir para todas as tarefas t_j em função dos recursos r_i que ele possui. Portanto, cada agente deve agendar as tarefas t_j nos recursos r_i .

Cada tarefa t_j têm quatro parâmetros importantes:

1. Data de atendimento solicitada, a_j . Esta é a data que o agente solicitador quer que sua solicitação inicie;
2. Data possível de atendimento, e_{ji} . Esta é a data em que o agente pode efetivamente atender a tarefa t_j no recurso r_i ;
3. Tempo de término da tarefa, C_j ;

4. Tempo de processamento, p_{ji} . É o tempo de processamento da tarefa t_j no recurso r_i de um certo agente. O recurso r_i considerado é o recurso escolhido para atender a tarefa.

Cada agente deve procurar qual o recurso r_i que pode atender a tarefa t_j numa data possível de atendimento e_{ji} o mais próximo possível da data de solicitação a_j , considerando que o tempo de processamento de cada recurso r_i seja igual.

A escolha deve respeitar a condição de que a data possível de atendimento e_{ji} seja maior ou igual a data de solicitação a_j , $e_{ji} \geq a_j$.

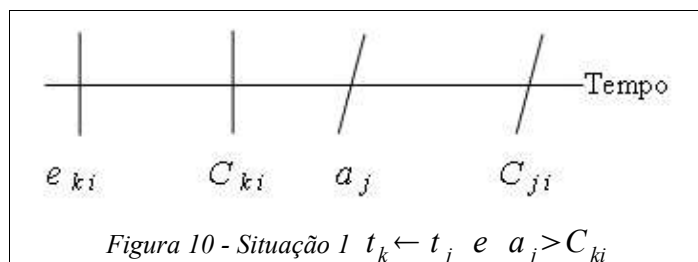
A diferença entre a data possível de atendimento e_{ji} e a data de solicitação a_j é denominada Aa_{ji} . Assim, tem-se: $Aa_{ji} = e_{ji} - a_j$.

Um aumento no valor de Aa_{ji} implica num aumento de C_j e, portanto, visando minimizar C_j , deve-se achar o $\min\{Aa_{ji}\}$, ou seja, o e_{ji} mais perto do valor de a_j . Vale ressaltar que Aa_{ji} nunca é menor que zero, pois como visto anteriormente $e_{ji} \geq a_j$.

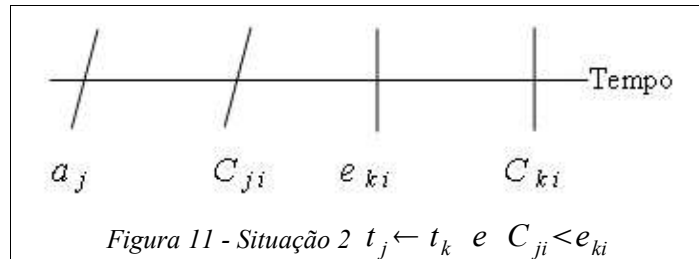
Na equivalência proposta, as tarefas são executadas em paralelo porque cada uma delas roda em um agente específico. Assim, a data de término da solicitação j_u é calculado como o maior tempo de término dentre as tarefas t_j que a compõem. O cálculo da data de término da solicitação é feito através da fórmula $C_u = \max\{C_j\}$.

Quando o agente vai decidir sobre qual recurso deve escolher para atender a tarefa t_j pode acontecer a situação na qual já exista uma tarefa agendada t_k e que ocupe integralmente ou parcialmente a agenda do recurso no qual se deseja agendar a nova tarefa solicitada t_j . Em função das restrições de precedência, \leftarrow , podem ocorrer as seguintes situações para se definir a data possível de atendimento da tarefa solicitada e_{ji} :

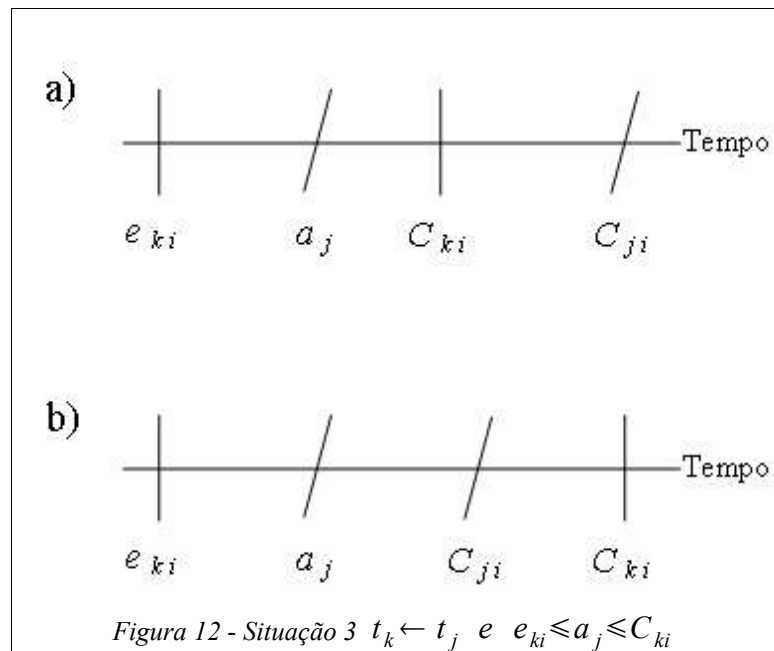
1. O início da tarefa agendada t_k é anterior ao início da tarefa solicitada t_j e o início da tarefa solicitada t_j é posterior ao término da tarefa t_k , ($t_k \leftarrow t_j$ e $a_j > C_{ki}$) (Figura 10);



2. O início da tarefa solicitada t_j é anterior ao início da tarefa agendada t_k e o início da tarefa agendada t_k é posterior ao término da tarefa solicitada t_j , ($t_j \leftarrow t_k$ e $C_{ji} < e_{ki}$) (Figura 11);



3. O início da tarefa agendada t_k é anterior ao início da tarefa solicitada t_j e o início da tarefa solicitada t_j é anterior ao término da tarefa agendada t_k , ($t_k \leftarrow t_j$ e $e_{ki} \leq a_j \leq C_{ki}$) (Figura 12). Nesta situação podem ocorrer duas combinações: a) O término da tarefa solicitada t_j é posterior ao término da tarefa agendada t_k e b) O término da tarefa agendada t_k é posterior ao término da tarefa solicitada t_j ;



4. O início da tarefa solicitada t_j é anterior ao início da tarefa agendada t_k e o início da tarefa agendada t_k é anterior ao término da tarefa solicitada t_j , ($t_j \leftarrow t_k$ e $a_j \leq e_{ki} \leq C_{ji}$) (Figura 13). Nesta situação podem ocorrer duas combina-

- ções: a) O término da tarefa solicitada t_j é anterior ao término da tarefa agendada t_k
 e b) O término da tarefa agendada t_k é anterior ao término da tarefa solicitada t_j ;

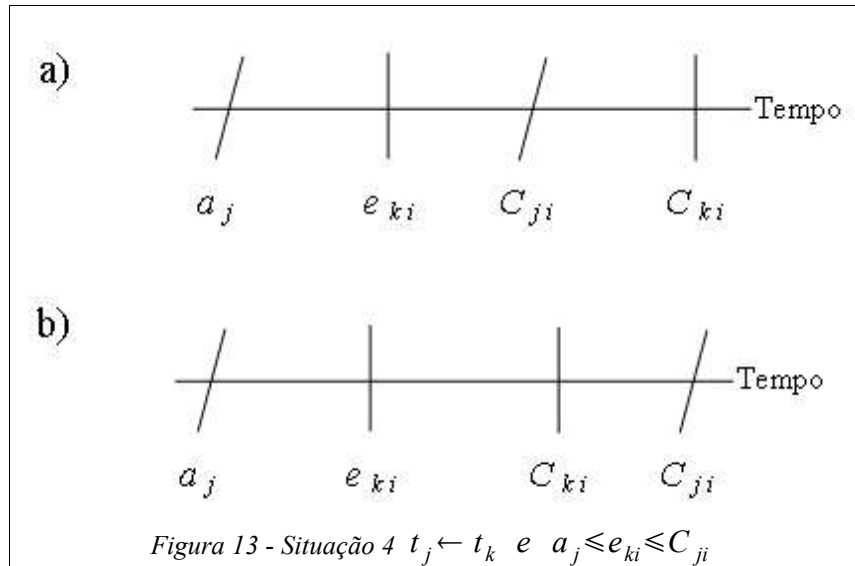


Figura 13 - Situação 4 $t_j \leftarrow t_k$ e $a_j \leq e_{ki} \leq C_{ji}$

A situação 1 e a situação 2 não apresentam problemas de negociação pois a nova tarefa t_j não interfere com a solicitação que já estava agendada t_k . No entanto, a situação 3 e a situação 4 demandam que o agente decida se ele deve manter o início da tarefa t_k sem alteração ou deve alterá-lo a fim de acomodar a nova tarefa t_j .

Para tomar esta decisão o agente pode implementar diversas estratégias de resolução. Conforme visto na definição da abordagem proposta estão sendo implementados dois padrões de decisão em relação à alteração das solicitações previamente agendadas para a acomodação da solicitação que está sendo tratada, que são:

1. A solicitação que tiver a menor data de solicitação será a primeira a ser atendida;
2. Caso haja duas solicitações com a mesma data de chegada, o critério de escolha é inspirado na emoção humana.

Nos dois padrões de decisão, o agente deve escolher qual o melhor recurso para atender a solicitação. O recurso escolhido pelo agente para executar a nova tarefa solicitada t_j é aquele que tenha o menor valor de Aa_{ji} e que respeite a restrição que $Aa_{ji} \geq 0$.

Caso dois recursos tenham o mesmo valor de Aa_{ji} , então faz-se a opção pelo recurso r_i que sofrer o menor impacto por conta da inserção de uma nova tarefa t_j na sua agenda. Este impacto é representado pela variável I_{ji} . A hipótese de se usar o recurso que tenha o menor valor de I_{ji} como critério de desempate é devido ao fato que quanto menor a perturbação no sistema, menos desequilíbrio é gerado no sistema e assim ele tende a voltar mais rapidamente para o estado de equilíbrio.

A variável I_{ji} é calculada como $I_{ji} = \sum_{k=1}^n Detar_{ki} + \alpha \cdot \sum_{k=1}^n Tetar_{ki}$ onde j representa a nova tarefa que precisa ser inserida na agenda, i é o recurso que está sendo usado, k é o número da tarefa agendada e n é o número de solicitações já agendadas.

A variável $Detar_{ki}$ representa se uma tarefa já agendada k vai ter sua data de início alterada ou não. Assim, $Detar_{ki}$ é uma variável discreta que pode ter dois valores: a) Zero, se a solicitação agendada k não teve seu e_{ki} alterado e b) Um, se a solicitação agendada k teve seu e_{ki} alterado.

A variável $Tetar_{ki}$ representa a diferença entre a nova data agendada da tarefa k em função da inserção da nova tarefa t_j na agenda do recurso r_i menos a data agendada e_{ki} . Calcula-se $Tetar_{ki}$ pela fórmula $Tetar_{ki} = (ea_{ki} - e_{ki})$, onde ea_{ki} é a nova data alterada da tarefa k e e_{ki} é a data que estava agendada a tarefa k .

O parâmetro α é um valor empírico que representa um fator para unificar valores de grandezas diferentes, tarefas deslocadas, unidades, e tempo deslocado, horas. Além disso, esta variável pode ter a função de fazer o impacto da inserção ser mais sensível ao deslocamento de tarefas, diminuindo assim o esforço de negociação, ou ser mais sensível ao tempo total deslocado das tarefas afetadas, colocando mais ênfase na diminuição do impacto para os clientes, afetando-os menos. No caso desta tese adotou-se-se o valor do parâmetro α como sendo $\alpha = 0,10$.

No entanto, esta decisão de mudar a data da solicitação agendada t_k fica condicionada à vontade do agente em colaborar com o agente que solicitou a nova tarefa solicitada t_j ou colaborar com a tarefa que já estava agendada t_k e não alterá-la. Para esta decisão é que existem os padrões de negociação.

No padrão de negociação 1 utiliza-se o critério de sempre deslocar a solicitação já agendada fazendo uso direto das regras citadas anteriormente. Já no padrão de negociação 2 faz-se uso do protocolo proposto nesta tese que se inspira na emoção humana e é denominado Grau de Gratidão. A modelagem matemática do mecanismo de decisão adotado no Grau de Gratidão será apresentado nas seções subseqüentes.

Como os agentes trabalham em paralelo, então o tempo para terminar o processamento da solicitação será o tempo do agente que demorar mais para terminar a tarefa sob sua responsabilidade. Assim, o tempo de término da solicitação $C_{u(sistema)}$ será calculado como $C_{u(sistema)} = \max(C_{u(agente)})$. Aliado a isto, todos os agentes iniciam suas atividades ao mesmo tempo.

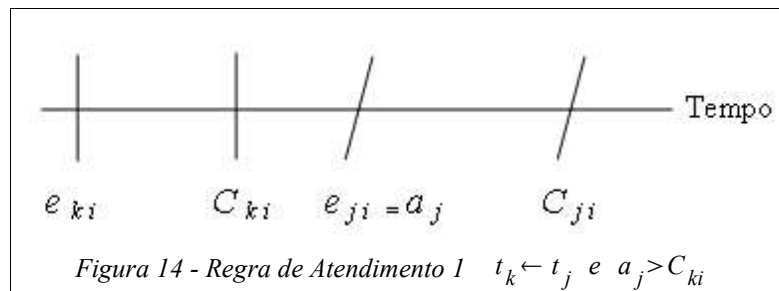
Por fim, pode-se inferir que o tempo total do escalonamento, *makespan*, de uma solicitação vinda do mundo externo para o sistema $C_{u(sistema)}$ é uma boa solução, *quasi-ótima*, caso todos os m agentes consigam para esta solicitação o seu melhor $C_{u(agente)}$.

Nas próximas seções será apresentada a modelagem matemática dos dois padrões de negociação citados.

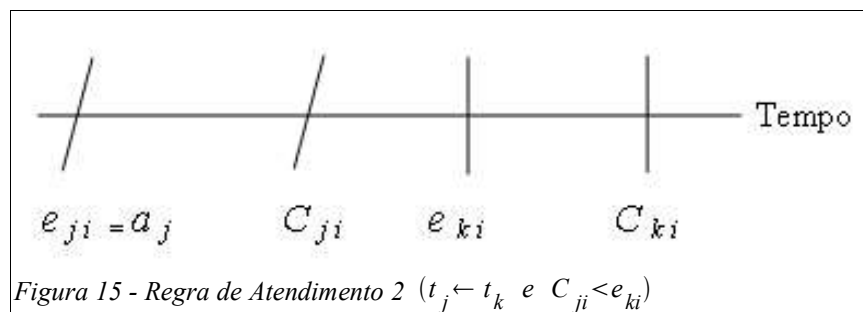
4.4) Modelagem Matemática do Padrão de Negociação de Atendimento por Ordem da Data da Solicitação

Neste padrão de negociação, a solicitação que tiver a menor data de solicitação a_j será sempre atendida primeiro, mesmo que a solicitação tenha sido feita ao sistema posteriormente a outra já alocada. Assim, considerando as restrições de precedência, \leftarrow , e com base nas situações citadas anteriormente, têm-se as seguintes regras para definir a data possível de atendimento e_{ji} :

1. Quando o início da tarefa t_k é anterior ao início da tarefa t_j e o início da tarefa t_j é posterior ao término da tarefa t_k , ($t_k \leftarrow t_j$ e $a_j > C_{ki}$), então a data de atendimento de t_j é igual a data solicitada $e_{ji} = a_j$ e pode-se concluir que $Aa_{ji} = e_{ji} - a_j = 0$ e como não vai haver deslocamento da tarefa agendada t_k , então $Detar_{ki} = 0$ e $Tetar_{ki} = 0$ (Figura 14);

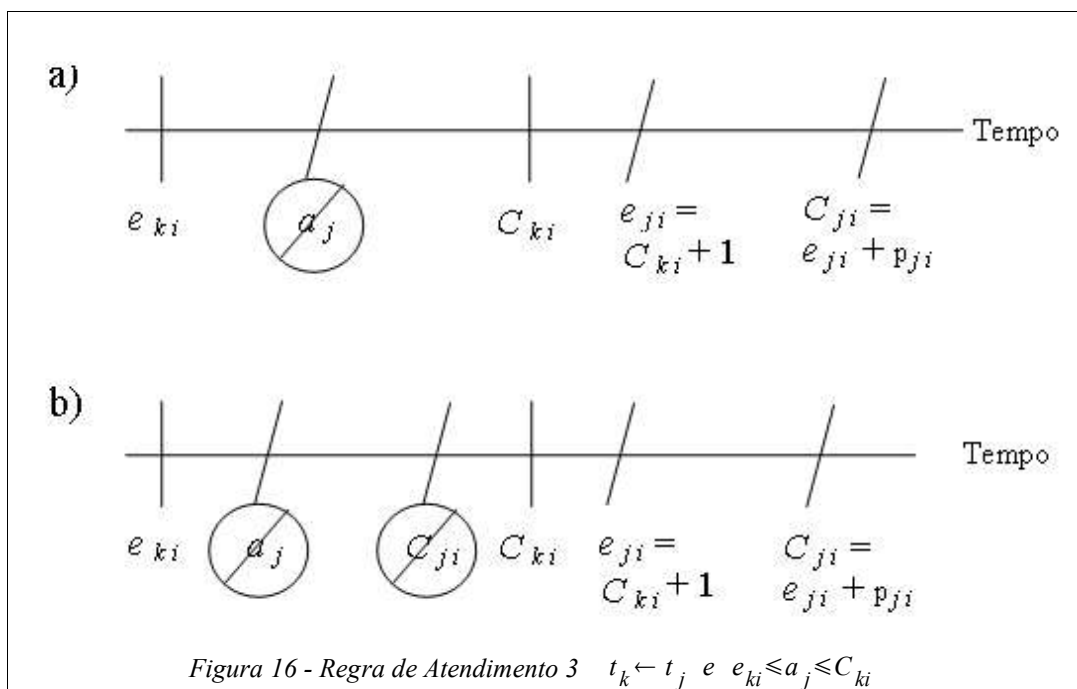


2. Quando o início da tarefa t_j é anterior ao início da tarefa t_k e o início da tarefa t_k é posterior ao término da tarefa t_j , ($t_j \leftarrow t_k$ e $C_{ji} < e_{ki}$), então a data de atendimento de t_j é igual a data solicitada $e_{ji} = a_j$ e pode-se concluir que $Aa_{ji} = e_{ji} - a_j = 0$ e como não vai haver deslocamento da tarefa agendada t_k , então $Detar_{ki} = 0$ e $Tetar_{ki} = 0$ (Figura 15);



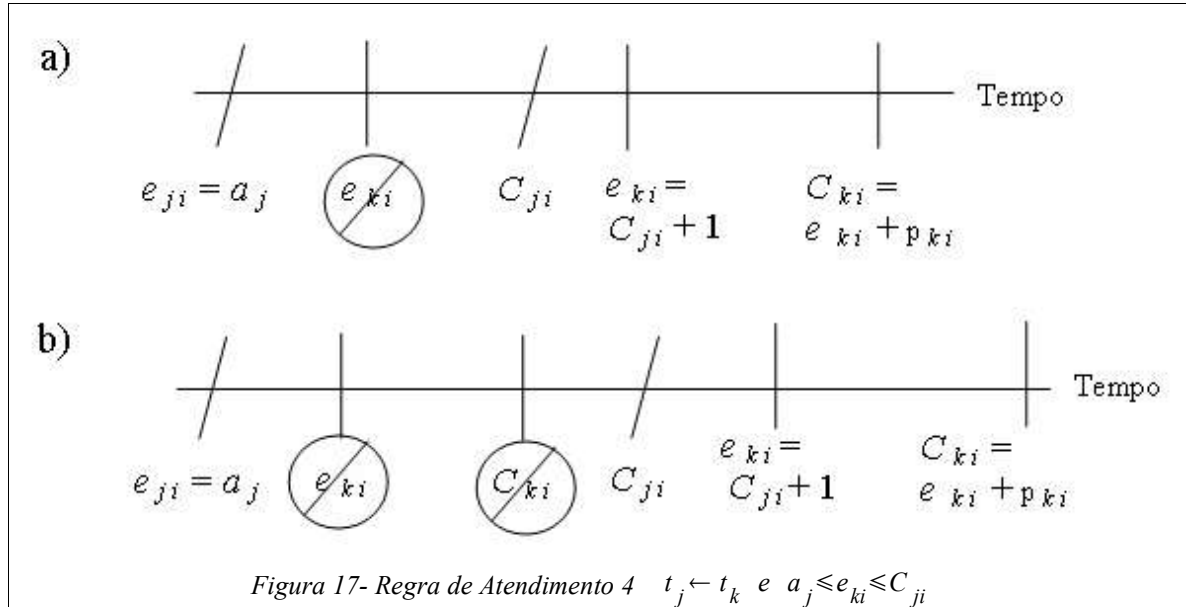
3. Quando o início da tarefa t_k é anterior ao início da tarefa t_j e o início da tarefa t_j é anterior ao término da tarefa t_k , ($t_k \leftarrow t_j$ e $e_{ki} \leq a_j \leq C_{ki}$) podem ocorrer duas combinações: a) O término da tarefa t_j é posterior ao término da tarefa t_k e b) O

término da tarefa t_k é posterior ao término da tarefa t_j . Para ambos os casos tem-se que a data possível de atendimento da tarefa t_j é maior que a data possível de atendimento da tarefa t_k , ou seja $a_j > e_{ki}$. Como a tarefa t_j tem a data solicitada posterior a data possível de atendimento da tarefa t_k , a tarefa t_j só irá começar depois da tarefa t_k terminar, resultando assim numa data possível de atendimento da tarefa t_j posterior a data solicitada da tarefa t_j , ou seja $e_{ji} > a_j$. Pode-se concluir para ambos os casos que $e_{ji} = C_{ki} + 1$ e $Aa_{ji} = e_{ji} - a_j > 0$. Como não vai haver deslocamento da tarefa agendada t_k , então então $Detar_{ki} = 0$ e $Tetar_{ki} = 0$ (Figura 16);



4. Quando o início da tarefa t_j é anterior ao início da tarefa t_k e o início da tarefa t_k é anterior ao término da tarefa t_j , ($t_j \leftarrow t_k$ e $a_j \leq e_{ki} \leq C_{ji}$) podem ocorrer duas combinações: a) O término da tarefa t_j é anterior ao término da tarefa t_k e b) O término da tarefa t_k é anterior ao término da tarefa t_j . Para ambos os casos tem-se que a data solicitada da tarefa t_j é menor que a data possível de atendimento da tarefa t_k , ou seja $a_j < e_{ki}$. Como a tarefa t_j tem a data solicitada anterior à data possível de atendimento da tarefa t_k , a tarefa t_k só irá começar depois da tarefa t_j terminar, resultando assim num deslocamento da data possível de atendimento da tarefa t_k para uma data posterior à data possível de atendimento da tarefa t_j , ou seja $e_{ki(novo)} > e_{ki(anterior)}$ e mantendo a data de atendimento da tarefa t_j igual a data solicitada, ou seja $e_{ji} = a_j$. Pode-se concluir para ambos os casos que $e_{ki} = C_{ji} + 1$ e

$Aa_{ji} = e_{ji} - a_j = 0$. Como vai haver deslocamento da data possível de atendimento da tarefa agendada t_k , então então $Detar_{ki} = 1$ e $Tetar_{ki} > 0$ (Figura 17);



Nas regras 1 e 2 pode-se atender a nova tarefa solicitada t_j na data solicitada sem gerar um deslocamento da data possível de atendimento da tarefa agendada t_k . Estas são as situações ideais pois não geram desequilíbrio no sistema e ainda por cima atendem a nova solicitação na data solicitada. Estas situações mantêm o estado de Equilíbrio do sistema.

No caso da regra 3 não ocorre a necessidade de se deslocar a data possível de início da tarefa t_k mas não se consegue atender a nova tarefa t_j na data solicitada, necessitando assim, sugerir uma nova data possível de atendimento que irá gerar um estado de Desequilíbrio no sistema em função da sugestão de uma nova data possível de atendimento para a tarefa t_j .

A regra 4 é muito similar em impacto no sistema à regra 3, pois ela gera um estado de Desequilíbrio no sistema em função de alterar a data possível de atendimento da tarefa já agendada t_k pois necessita sugerir uma nova data possível de atendimento para a tarefa t_k .

Quando o início da tarefa t_k for igual ao início da tarefa t_j , então a tarefa t_j só irá começar depois da tarefa t_k terminar, resultando assim numa data possível de atendimento da tarefa t_j posterior a data solicitada da tarefa t_j , ou seja $e_{ji} > a_j$. Pode-se concluir que $e_{ji} = C_{ki} + 1$ e $Aa_{ji} = e_{ji} - a_j > 0$. Como não vai haver deslocamento da tarefa agendada t_k , então então $Detar_{ki} = 0$ e $Tetar_{ki} = 0$;

Na próxima seção é apresentado o outro padrão de negociação inspirado na emoção humana.

4.5) Modelagem do Padrão de Negociação Inspirado na Emoção Gratidão

Neste padrão, utiliza-se o mesmo critério adotado no protocolo anterior, no entanto, é acrescentado um outro critério inspirado na emoção humana gratidão. Este padrão de negociação é utilizado na situação na qual uma solicitação é feita para uma data igual ao de uma solicitação já agendada. Neste caso, o agente precisa optar se deve ou não deslocar a solicitação já agendada. Esta decisão é, então, tomada em função do padrão de negociação inspirado na emoção humana gratidão.

Neste padrão de negociação, cada agente p deve possuir uma conta corrente emocional, Cc_{pr} , para todos os outros agentes r com os quais ele se relaciona, constituído de dois valores:

1. Fr_{pr} representa os favores recebidos. É a soma dos favores recebidos de outros agentes e é composto de dois valores: a) Número de solicitações que o outro agente r teve de modificar a agenda e agendar de novo para atender as solicitações feitas por ele, Nar_{pr} ; b) O somatório do tempo total de todas as solicitações que foram deslocadas pelo outro agente r para atender as solicitações feitas por ele, Tdr_{pr} ;
2. Fp_{pr} representa os favores prestados. É a soma dos favores prestados para os outros agentes com os quais ele se relaciona. Assim como os favores recebidos, ele é composto de dois valores: a) Número de solicitações que o agente p teve de modificar a agenda e agendar de novo para atender as solicitações feitas pelo agente r , Nap_{pr} ; b) O somatório do tempo total de todas as solicitações que foram deslocadas pelo agente p para atender as solicitações feitas pelo agente r , Tdp_{pr} ;

Os valores de Nar_{pr} e Tdr_{pr} referentes aos favores recebidos não são calculados e são informados por meio de uma mensagem enviada pelo outro agente r que está negociando com ele. Por este motivo, é fundamental a figura do agente confiável para não mandar mensagem com intenção de enganar o estado do outro agente.

O índice Nap_{pr} é calculado pela fórmula: $Nap_{pr} = \sum_{k=1}^q AltE_k$, onde k representa a solicitação agendada t_k , q é número de solicitações do sistema e $AltE_k$ representa se uma solicitação agendada t_k teve sua data possível de atendimento alterada ou não. Assim, $AltE_k$ é uma variável discreta que pode assumir dois valores: a) Zero, se a solicitação agendada t_k não teve sua data possível de atendimento e_k alterada e b) Um, se a solicitação agendada t_k teve sua data possível de atendimento e_k alterada.

O índice Tdp_{pr} é calculado pela fórmula: $Tdp_{pr} = \sum_{k=1}^q (ne_k - e_k)$, onde k representa a solicitação agendada t_k , q é número de solicitações do sistema, e_k é data possível de atendimento à solicitação t_k e ne_k representa a nova data possível de atendimento para a solicitação agendada t_k .

No tempo $t = 0$ quando os agentes *nascem* para o sistema multiagentes, todos os agentes do sistema estão com os valores dos favores recebidos e dos favores prestados igual a zero para cada um dos agentes r , com os quais eles se relacionam, ou seja $Fr_{pr} = (0, 0)$ e $Fp_{pr} = (0, 0)$.

O agente p terá para com o agente r com o qual está se relacionando um *Grau de Gratidão* Gg_{pr} , definido como $Gg_{pr} = (NDeslo_{pr}, TDeslo_{pr})$, onde:

1. $NDeslo_{pr}$ é o somatório do número de solicitações que o outro agente r teve de modificar a agenda e agendar de novo para atender as solicitações feita pelo agente p menos o somatório do número de solicitações que o agente p teve de modificar a agenda e agendar de novo para atender as solicitações feita pelo agente p . A variável $NDeslo_{pr}$ é, então, calculada como: $NDeslo = (Nar_{pr} - Nap_{pr})$;
2. $TDeslo_{pr}$ é o somatório do tempo total de todas as solicitações que foram deslocadas pelo outro agente r para atender as solicitação feitas pelo agente p menos o somatório do tempo total de todas as solicitações que foram deslocadas pelo agente p para atender as solicitação feitas pelo outro agente r . A variável $TDeslo_{pr}$ é, então, calculada como $TDeslo = (Tdr_{pr} - Tdp_{pr})$.

O Grau de Gratidão pode ter três estados:

1. Positivo. O Grau de Gratidão será positivo se ocorrer uma das duas condições: a) $NDeslo_{pr} > 0$ ou b) $NDeslo_{pr} = 0$ e $TDeslo_{pr} > 0$. Caso Gg_{pr} tenha estado positivo, é razoável supor que o agente p deve favores ao agente r e tentará ajudar ao agente r visando retribuir a ajuda anteriormente recebida;
2. Nulo. O Grau de Gratidão será nulo se ocorrer a seguinte condição: $NDeslo_{pr} = 0$ e $TDeslo_{pr} = 0$. Caso Gg_{pr} tenha estado nulo, ele não deve favores e nem tem crédito de ajuda prestada, portanto, ele decide buscando sua otimalidade;
3. Negativo. O Grau de Gratidão será negativo se ocorrer uma das duas condições: a) $NDeslo_{pr} < 0$ ou b) $NDeslo_{pr} = 0$ e $TDeslo_{pr} < 0$. No caso do Gg_{pr} ser negativo, o agente p prestou mais ajuda do que recebeu do agente r . Nesta situação o agente r deve favores ao agente p e, então o agente p opta por simplesmente não se esforçar para atender à nova solicitação e não altera nenhuma das tarefas agendadas t_k .

No estado 2 e no estado 3, apesar do agente não precisar ajudar, ele pode optar por ajudar como uma estratégia de melhorar sua posição em relação ao outro agente visando futuras negociações.

No intuito de evitar que o sistema possa cair no plano de caridade ou benevolência, onde somente um agente estaria ajudando o outro, e acabaria gerando um desequilíbrio no sistema é estabelecido para todos os agentes do sistema multiagentes uma faixa de valores que Gg_{pr} pode variar. Esta faixa é denominada MGg e composta de duas variáveis:

1. $LimNDeslo$, que é o maior valor possível que $NDeslo_{pr}$ pode assumir;
2. $LimTDeslo$, que é o maior valor possível que $TDeslo_{pr}$ pode assumir.

Os valores de $LimNDeslo$ e $LimTDeslo$ são estabelecidos como parâmetros iniciais do sistema multiagentes. Assim, tem-se que o valor positivo da faixa é expresso por $MGg = (LimNDeslo, LimTDeslo)$ e o limite mínimo da faixa é expresso por $-MGg$.

No intuito de comparar se o valor do Grau de Gratidão está contido dentro dos limites da faixa, devem ser definidas as operações maior, igual e menor entre eles, assim têm-se:

1. Maior, $Gg_{pr} > MGg$. O Grau de Gratidão será maior que o limite máximo quando ocorrerem uma das duas condições: a) $NDeslo_{pr} > LimNDeslo$ ou b) $NDeslo_{pr} = LimNDeslo$ e $TDeslo_{pr} > LimTDeslo$;
2. Igual, $Gg_{pr} = MGg$. O Grau de Gratidão será igual ao limite máximo quando ocorrer a condição $NDeslo_{pr} = LimNDeslo$ e $TDeslo_{pr} = LimTDeslo$;
3. Menor, $Gg_{pr} < MGg$. O Grau de Gratidão será menor que o limite máximo quando ocorrerem uma das duas condições: a) $NDeslo_{pr} < LimNDeslo$ ou b) $NDeslo_{pr} = LimNDeslo$ e $TDeslo_{pr} < LimTDeslo$.

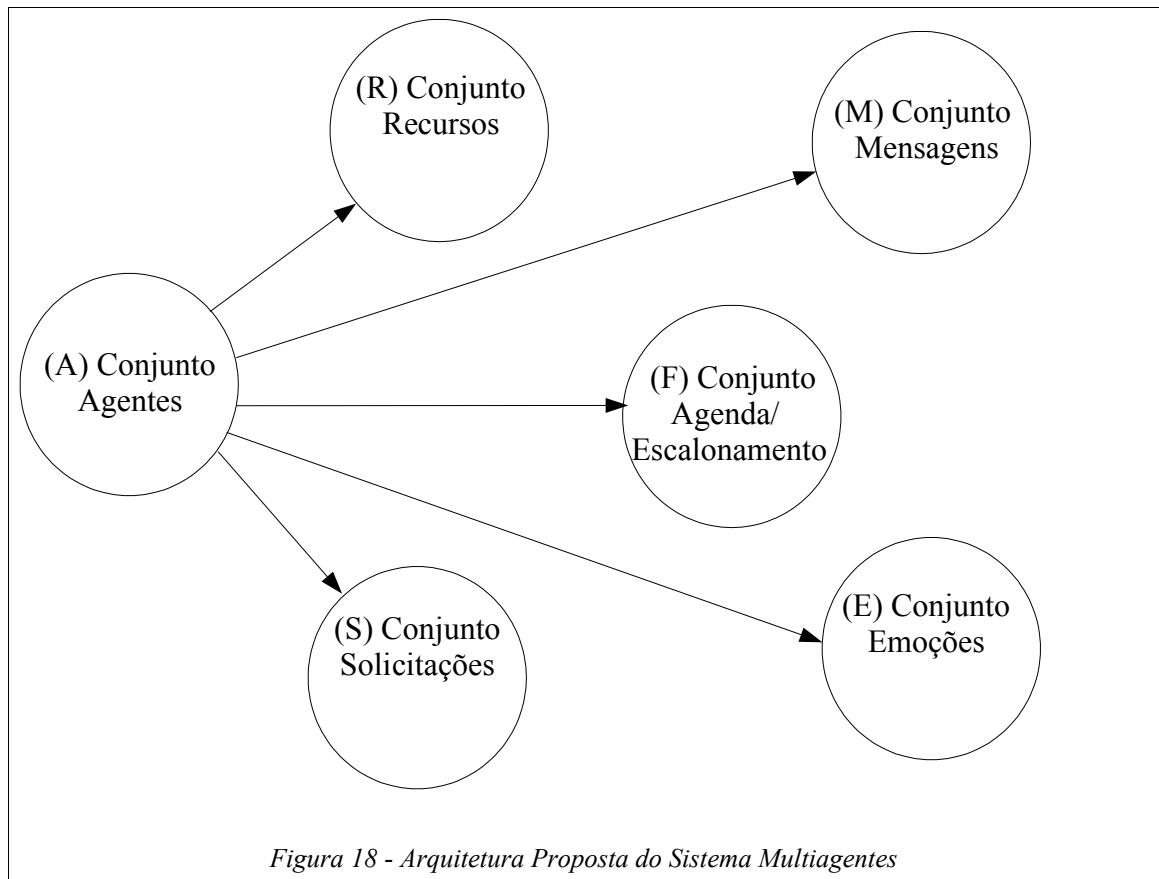
O valor do Grau de Gratidão deve ficar oscilando em valores intermediários aos limites positivos e negativos da faixa, onde os agentes ajudam e são ajudados.

4.6) Sistema Computacional para a Abordagem Proposta

É apresentado nesta seção desenvolvimento do sistema computacional que serve de plataforma de testes e validação da abordagem proposta. Os detalhes da implementação computacional são descritos no Anexo A.

O sistema proposto descreve o sistema multiagentes como uma *n-upla* (A, R, S, M, F, E). Nesta *n-upla* o A representa o conjunto de todos os agentes do sistema, o R representa o conjunto de todos os recursos do sistema, o S representa o conjunto de todas as solicitações recebidas pelo sistema, o M representa o conjunto de todas as mensagens recebidas e enviadas pelos agentes para negociar entre si, o F representa o conjunto das agendas de todos os recursos referente a todas as solicitações do sistema e o E representa as emoções registradas (Figura 18).

Os agentes do sistema multiagentes também são descritos como uma *n-upla* e, cada agente possui: 1) recursos próprios, 2) solicitações recebidas, 3) mensagens recebidas e enviadas pelos agentes para negociar entre si, 4) um conjunto das agendas de cada um dos seus recursos e 5) As emoções registradas em relação a todos os agentes que ele se relaciona.



O sistema computacional tem por objetivo construir um conjunto único de classes que representem todos os agentes do sistema multiagentes e que a diferença entre cada agente seja estabelecida através de arquivos de parâmetros lidos quando o agente é inicializado. Assim sendo, foi desenvolvido um *agente padrão* que recebe e envia somente dois tipos de mensagem: 1) Solicitação e 2) Resposta.

Para implementação do modelo do sistema multiagentes foi usada a ferramenta DECAF (*Distributed Environment Centered Agent Framework*) (GRAHAM, 2001, 2000) (GRAHAM, MERSIC, DECKER, 2000) (MCGEARY, 2001) (MCGEARY, DECKER, 2001) (GRAHAM, DECKER, MERSIC, 2003) e como protocolo/linguagem de comunicação entre os agentes, optou-se pelo KQML (*Knowledge Query and Manipulation Language*) (FININ, 1994) pois é um dos protocolos mais difundidos em termos de protocolo de comunicação entre agentes e é o protocolo nativo da ferramenta DECAF.

Como linguagem de programação foi usada a linguagem Java (KEN, et. al., 2006) (ECKEL, 2002) (LEA, 1997) pois ela se adapta muito bem a ambientes multi-plataformas e distribuídos e suporta todos os quesitos de orientação a objetos. Além disso, ela é a linguagem nativa da ferramenta DECAF.

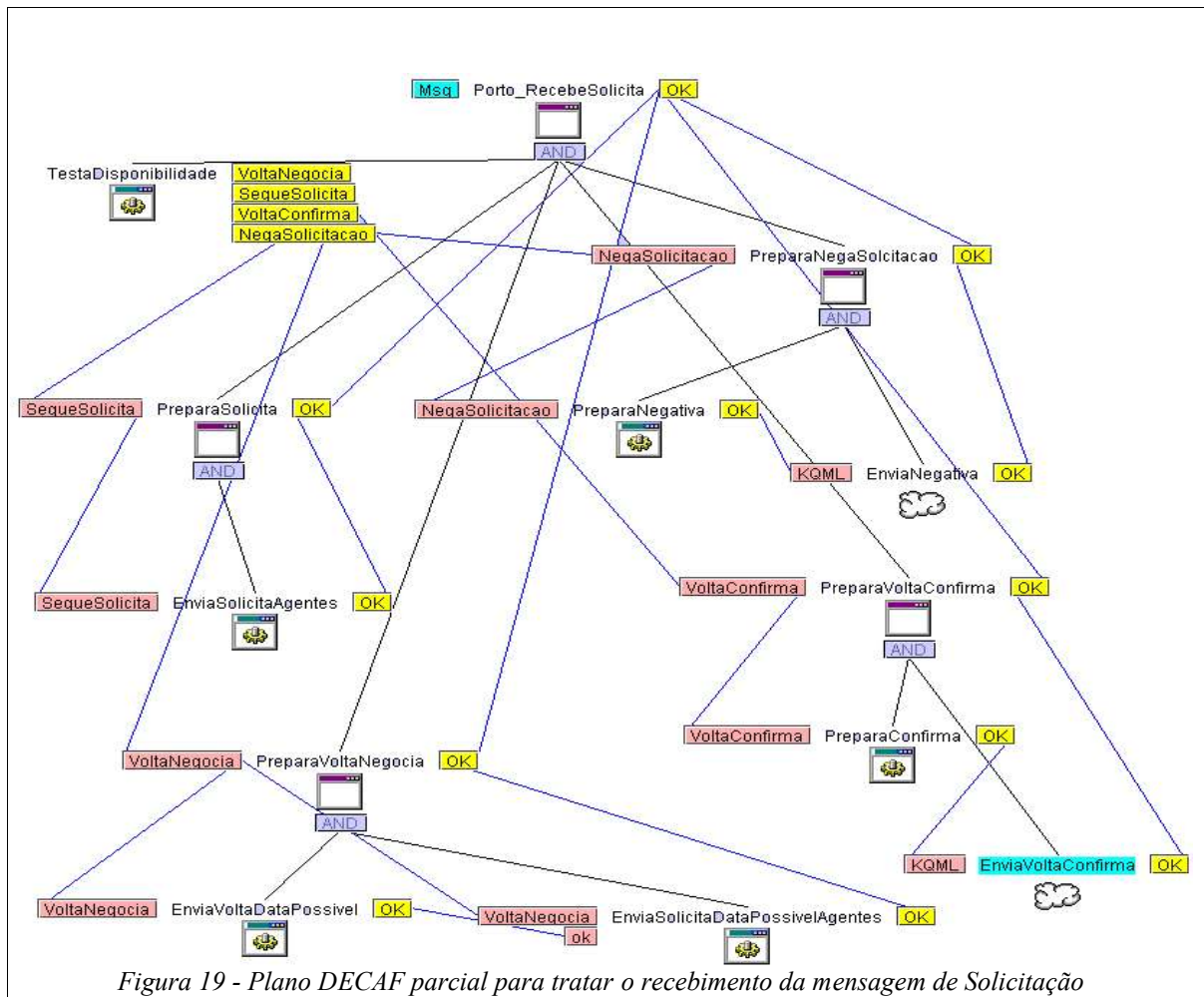
No intuito de criar o agente padrão do sistema multiagentes foram modeladas duas ferramentas no sistema computacional:

1. Um plano padrão para recebimento e envio das mensagens. Este plano foi implementado na ferramenta DECAF. Na Figura 19 é apresentado um plano parcial do agente escrito na ferramenta DECAF;
2. Um conjunto de classes escritas em Java para suportar as ações do agente.

No plano padrão desenvolvido na ferramenta DECAF foram descritos todos os possíveis métodos para recebimento e tratamento das mensagens. O plano DECAF do agente proposto trata somente dois tipos de mensagens: 1) Solicitação; 2) Resposta. Estas mensagens serão os únicos meios dos agentes atuarem no sistema multiagentes.

A mensagem do tipo Solicitação é única e padrão e tem como função receber todos os dados sobre a solicitação. Basicamente, em qualquer sistema de agendamento devem ser informados os seguintes dados: 1) Número da solicitação; 2) Dados do veículo que transporta a carga, pode ser navio, caminhão, trem, etc.; 3) Dados da carga. Por exemplo: quantidade, peso, tamanho e volume; 4) Data/hora estimada de chegada e 5) Número estimado de horas de operação. Estas variáveis podem ser alteradas conforme o problema que está sendo tratado e suas respectivas necessidades.

As mensagens do tipo Resposta possuem quatro subtipos possíveis: 1) Sugestão; 2) Confirmação; 3) Negativa e 4) Gratidão.



As mensagens do subtipo Sugestão são usadas quando o agente não consegue atender a solicitação na data solicitada e, conforme seus recursos e sua agenda ele define uma data mais próxima e posterior à solicitada que gere o menor impacto para sua agenda. Neste ponto, ele avisa a todos os solicitados e ao solicitador esta nova sugestão de data. As mensagens são compostas dos seguintes dados: 1) Tipo da mensagem; 2) Número da solicitação e 3) Data/hora sugerida.

As mensagens do subtipo Confirmação são usadas quando o agente consegue negociar com todos seus solicitados uma data comum para atendimento a solicitação e retorna para o solicitador a confirmação da data solicitada. A mensagem é composta dos seguintes dados: 1) Tipo da mensagem e 2) Número da solicitação.

As mensagens do subtipo Negativa são usadas quando nenhum recurso que o agente possui tem as características mínimas para atender a solicitação, independente da data solicitada. Cita-se como exemplo um navio com comprimento maior do que o maior berço disponível para os navios no porto. A mensagem é composta dos seguintes dados: 1) Tipo da mensagem e 2) Número da solicitação. Todos os agentes que recebem uma mensagem do subtipo Negativa param a negociação imediatamente. Com exceção desta situação, os agentes sempre propõem uma nova data posterior à data solicitada.

As mensagens do subtipo Gratidão servem para enviar os valores dos favores prestados. A mensagem é composta dos seguintes dados: 1) Tipo da mensagem; 2) Valor dos favores prestados, (F_{pr}).

Foi desenvolvido um conjunto de classes escritas na linguagem de programação Java e encapsuladas em uma única interface que é a base para o processo de agendamento de cada recurso de cada agente. Neste conjunto de classes, onde estão as definições de como um agente funciona, destaca-se a classe denominada *pool de recursos*. Na classe *pool de recursos* cada recurso possui uma agenda específica onde ficam registradas as solicitações que estão agendadas em um determinado período.

Como mencionado anteriormente, este conjunto de classes usa arquivos de parâmetros para definir qualquer agente do sistema. Estes arquivos são carregados assim que o agente é inicializado. Dentre os arquivos de parâmetros destaca-se o que representa as relações de conhecimento, *acquaintances*, onde para cada agente fica registrado os seguintes dados: 1) Todos os agentes para os quais devem ser enviadas solicitações (os solicitados) e 2) Todos os agentes dos quais podem ser recebidas solicitações (os solicitadores).

A interface que encapsula o conjunto de classes implementada é denominada de Interface do Agente Geral e é a base de todos os agentes do sistema. Vale ressaltar, que cada agente implementado por esta interface conhece única e exclusivamente o seu mundo, que pode ser definido como sendo: seus recursos e suas respectivas agendas, as solicitações recebidas e enviadas, as mensagens recebidas e enviadas, e o status de negociação, confirmada ou pendente com os outros agentes com os quais ele negocia, solicitador e solicitados e o Grau de Gratidão em relação aos outros agentes.

Para efetivamente testar a abordagem proposta implementada conforme descrito nesta seção, optou-se por aplicar a abordagem e o sistema desenvolvido a um estudo de caso específico que é a elaboração da agenda para o atendimento dos navios para carregar e descarregar em um terminal marítimo de contêiner. Esta agenda é conhecida como fila de navios. Na próxima seção será feita uma explicação sobre o estudo de caso.

4.7) Estudo de Caso: Terminal Marítimo de Contêiner

Um porto pode ser definido como uma área de terra limítrofe com o mar ou rio que possui equipamentos para carregar e descarregar navios ou barcas com segurança para a embarcação e para a carga (UNCTAD, 1996) (AGERSCHOU, 1983) (ROSA, 2006).

A atracação é o processo no qual o navio é conduzido do alto mar para uma área do porto protegida de ondas e marés. Os berços de atracação são os locais protegidos contra ondas e correntes nos quais os navios ficam amarrados para realizar de forma segura as operações de carga e descarga. Cada navio ocupa um berço de atracação quando está operando.

O porto possui áreas contíguas aos berços de atracação denominadas pátio de contêiner que servem para armazenamento dos contêineres.

As operações de um porto podem sofrer atraso em função da quebra de equipamentos e/ou de navios com problemas ou características que dificultem a operação. Acrescenta-se a isto, a possível existência de situações de mau tempo que levem ao atraso das operações. Estas mesmas operações podem ser feitas em menor tempo quando as condições do navio facilitam as operações ou quando os próprios operadores de equipamentos se superam.

A data estimada de chegada do navio é conhecida como ETA, do inglês *estimated time of arrival*. Ela é informada com quinze dias de antecedência da data que o navio vai chegar. No entanto, nestes quinze dias podem ocorrer outras operações de carga/descarga deste navio em outros portos que podem atrasar ou adiantar a chegada do navio no porto, ocasionando assim, mudanças constantes na fila de navio. Assim sendo, necessita-se que sejam feitas alterações na fila de navios para refletir esta mudança da data de chegada do navio fazendo com que a fila de navios seja dinâmica.

Para a elaboração da fila de navios do porto são necessárias as seguintes informações: 1) A previsão de navios por um período (três a seis meses); 2) O tipo de carga que o navio está transportando (contêiner de 20 ou 40 pés); 3) A quantidade de contêineres; 4) Os equipamentos disponíveis no navio; 5) O número de berços disponíveis para o navio ficar atracado; 6) A capacidade de chegada/escoamento de carga do porto através de transporte multimodal (ferrovia, caminhão e navios).

Para que ela possa ser feita, devem-se agendar todos os recursos necessários para operação do navio na data possível de atendimento, isto é, berços de atracação, guindastes para o navio, áreas de pátio, empilhadeiras e pontes rolantes para o pátio entre outros. Só após a reserva de todos os recursos, pode-se considerar o navio na fila de navios.

Os administradores do porto preparam a agenda de atendimento dos navios no porto com base nos dados anteriormente citados. Esta agenda do porto é denominada fila de navios.

A partir da descrição sobre um porto e a fila de navios, foi desenvolvido um estudo de caso para atendimento a fila de navios em um Terminal Marítimo de Contêiner a fim de avaliar o modelo proposto. Neste estudo de caso foram definidos quatro agentes: 1) Agência; 2) Berço; 3) Pátio e 4) Equipamento.

O agente Agência, para poder realizar suas tarefas, precisa de recursos do agente Berço. Para o agente Berço poder realizar suas tarefas, precisa de recursos dos agentes Pátio e Equipamento. O agente Pátio, para poder realizar suas tarefas, precisa de recursos do agente Equipamento. O agente Equipamento não precisa de recursos de nenhum outro agente para poder concluir sua tarefa e, portanto, o agente equipamento não necessita solicitar recursos à outro agente sendo, então, o ponto de retorno da confirmação do sistema.

As mensagens de Solicitação para o problema do Terminal Marítimo de Contêiner podem ser parametrizadas conforme o tipo de problema, e possuem os seguintes dados: Número de referência do Solicitador, Dados do navio, Dados da carga, Data/Hora estimada de chegada, Número estimado de horas de operação.

No próximo capítulo, são realizados os experimentos para avaliar a abordagem proposta.

Capítulo 5 - Os Experimentos Realizados

No Capítulo anterior foi proposta e elaborada uma abordagem inspirada na sociedade humana para solução de problemas dinâmicos em ambientes complexos e, no intuito de avaliar esta abordagem são feitos testes neste capítulo usando o sistema computacional desenvolvido. Estes testes foram divididos em três etapas:

1. Etapa 1 - Validação do código do sistema computacional. O objetivo desta etapa é fazer a validação e conferência do código computacional escrito na linguagem Java que implementa a lógica de cada agente. Para fazer estes testes foi usada a biblioteca JUnit (CLARK, 2000) e um conjunto de testes com as situações limites. Esta etapa é muito importante pois é necessário garantir que a lógica do sistema computacional esteja correta antes de testá-la no sistema multiagentes que é distribuído e *multithread* o que dificulta e às vezes inviabiliza a depuração de erros de código;
2. Etapa 2 - Avaliação do sistema multiagentes proposto. Foram feitos testes de avaliação dos sistema fazendo uso de dados gerados aleatoriamente, simulando uma operação de um Terminal Marítimo de Contêiner.

A principal dificuldade para se desenvolver um conjunto de testes que pudesse ser padronizados e comparados decorre do fato do sistema ser distribuído e utilizar o protocolo TCP/IP, que não possui a garantia de entrega e recebimento de mensagens na mesma seqüências que foram enviadas. Por esta razão, não se pode garantir que um mesmo experimento consiga reproduzir o mesmo resultado final em todos os testes.

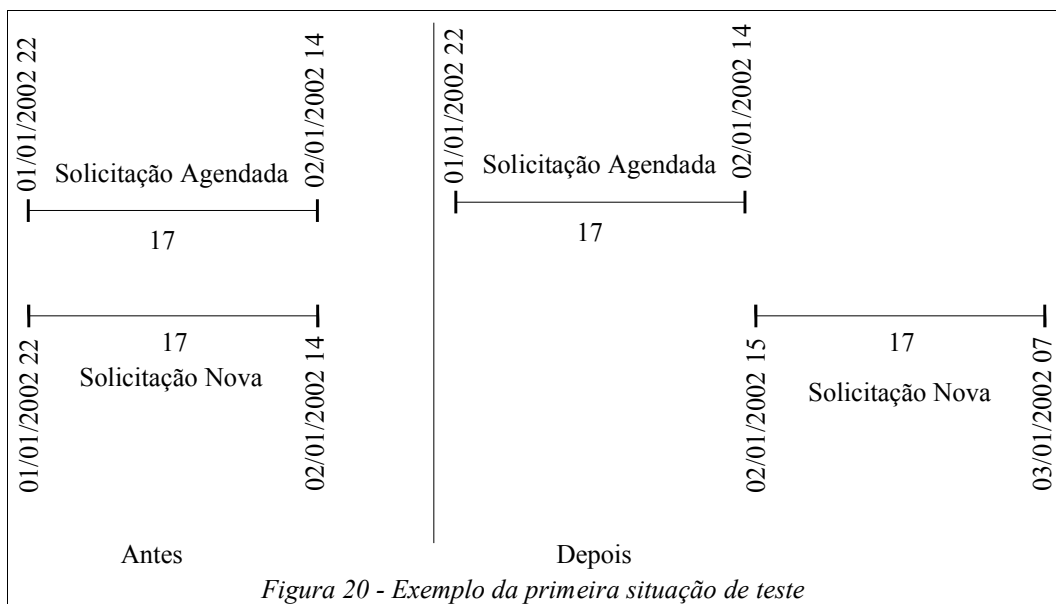
5.1) Validação do sistema computacional

Tendo em vista que o sistema é distribuído e, portanto, com alto grau de complexidade para ser testado, optou-se inicialmente por usar uma biblioteca de testes para testar a lógica computacional desenvolvida para cada agente fora do ambiente distribuído.

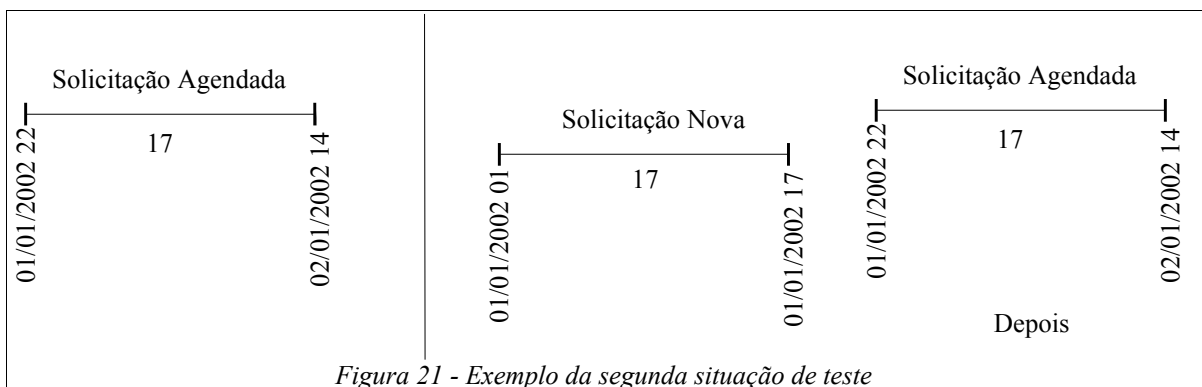
A vantagem de se usar uma biblioteca de testes é que, além de se garantir que o código em sua primeira versão funciona adequadamente, é possível, também, testar o funcionamento do código após o mesmo sofrer modificações e/ou implementações.

Foi escolhida a biblioteca de testes JUnit (CLARK, 2000) que provê de maneira automática e gráfica o resultado dos testes aplicados sobre as classes que devem ser testadas. Para se usar a biblioteca de testes é necessário levantar quais são as situações que podem ocorrer e qual o resultado que é esperado do processamento de cada uma das situações de teste. Assim sendo, foram levantadas treze possíveis situações que podem ocorrer quando a Interface Geral de Agentes é executada. Para estas treze situações foram definidos quais seriam os resultados esperados. As treze situações de teste que podem ocorrer são apresentadas a seguir:

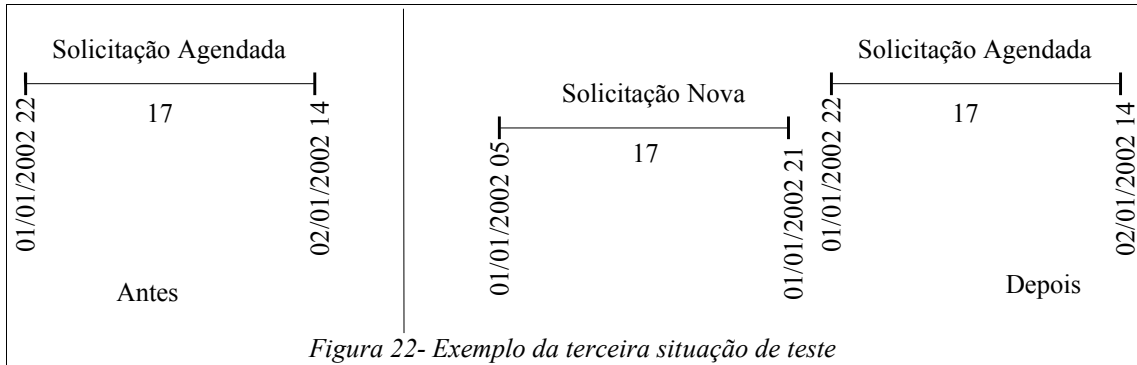
1. Existe uma solicitação já agendada e, uma nova solicitação é feita ao sistema. Esta nova solicitação inicia na mesma data/hora da solicitação já agendada (Figura 20);



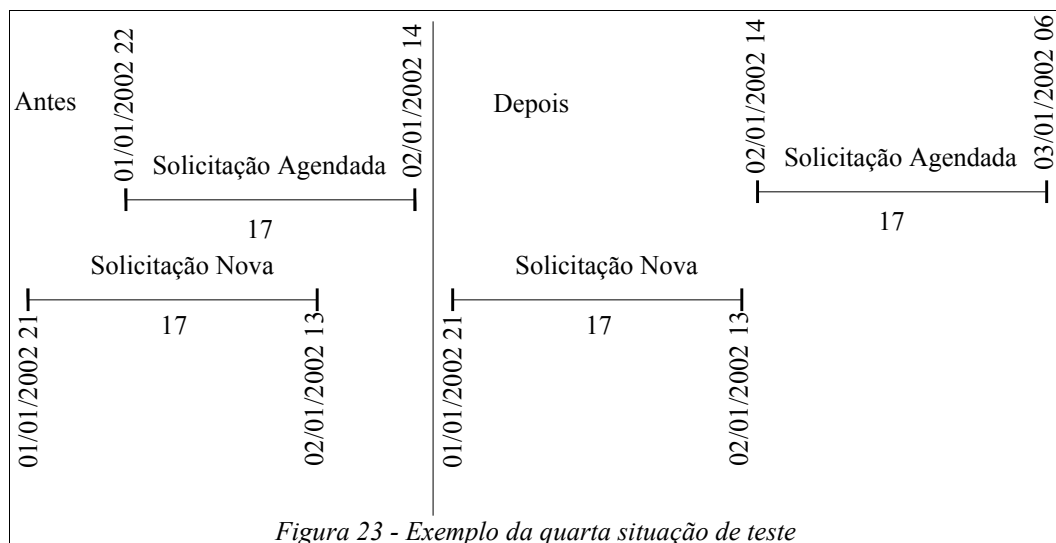
2. Existe uma solicitação já agendada e uma nova solicitação é feita ao sistema. Esta nova solicitação inicia e termina com folga antes da solicitação previamente agendada (Figura 21);



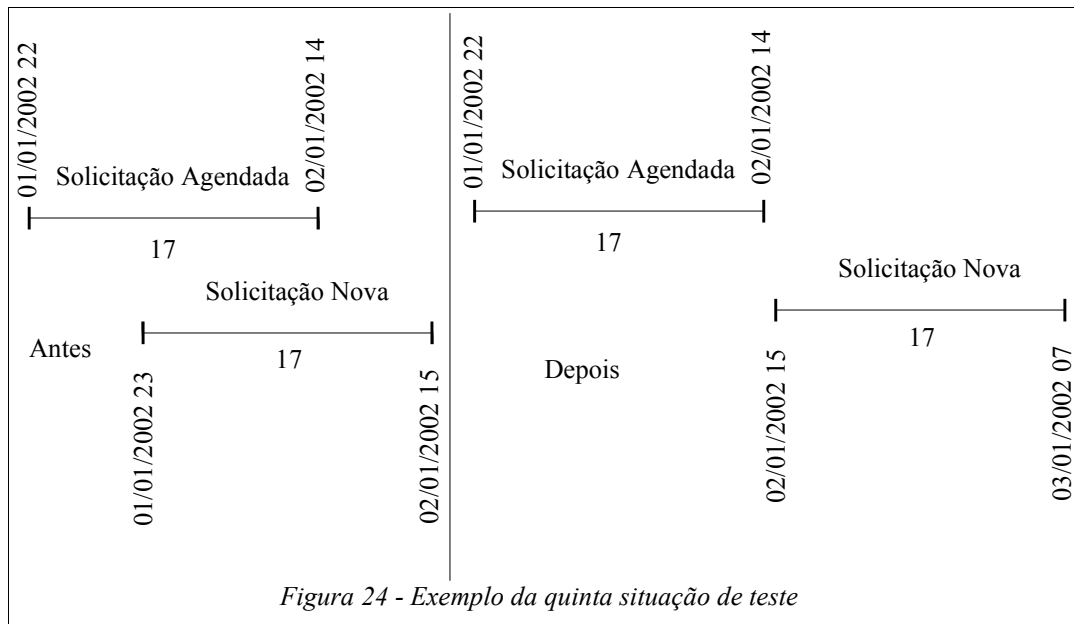
3. Existe uma solicitação já agendada e uma nova solicitação é feita ao sistema. Esta nova solicitação inicia e termina exatamente antes da solicitação previamente agendada (Figura 22);



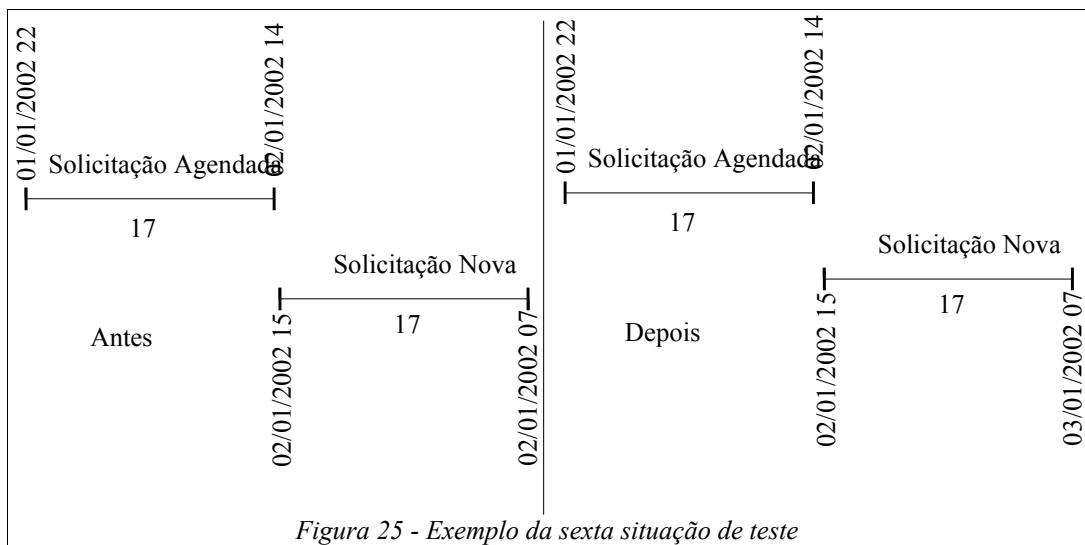
4. Existe uma solicitação já agendada e, uma nova solicitação é feita ao sistema. Esta nova solicitação inicia antes do início da solicitação previamente agendada e termina no intervalo do agendamento dela (Figura 23);



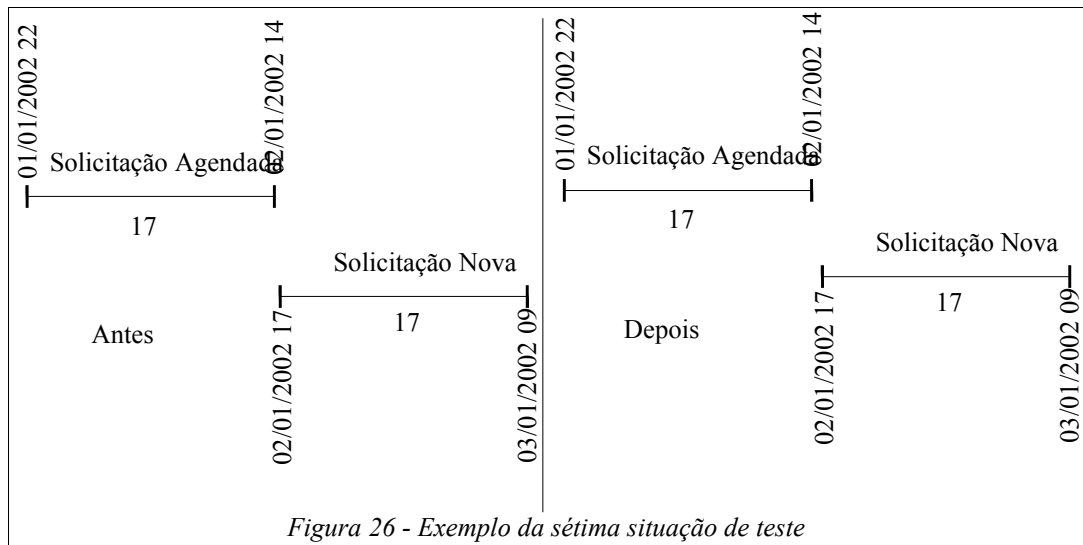
5. Existe uma solicitação já agendada e, uma nova solicitação é feita ao sistema. Esta nova solicitação inicia depois do início da solicitação previamente agendada e termina depois do término da mesma (Figura 24);



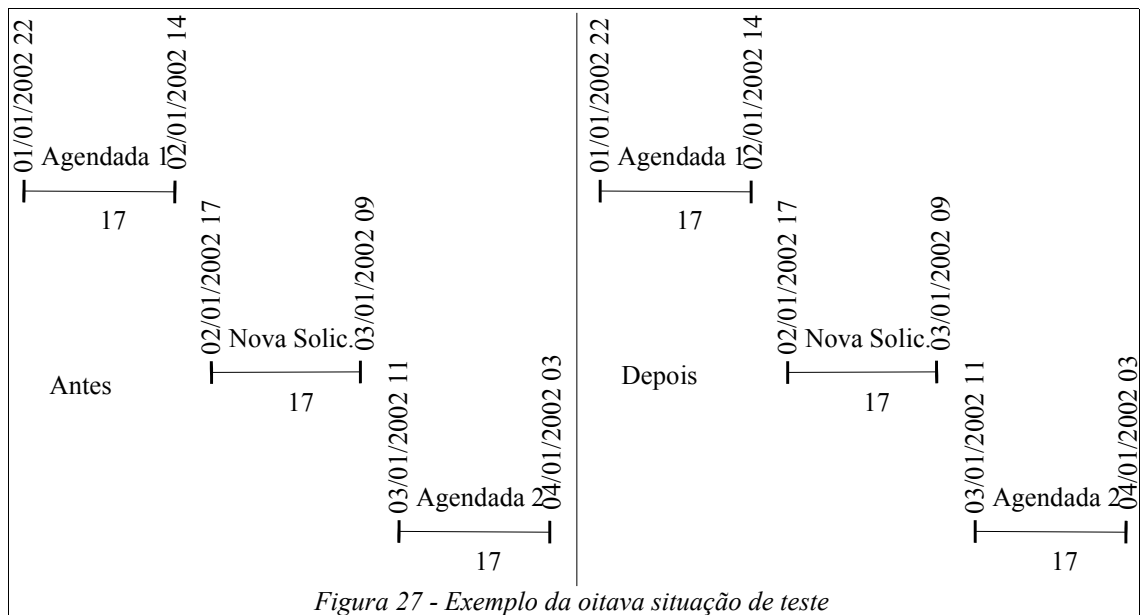
6. Existe uma solicitação já agendada e, uma nova solicitação é feita ao sistema. Esta nova solicitação inicia exatamente depois do fim da solicitação previamente agendada (Figura 25);



7. Existe uma solicitação já agendada e, uma nova solicitação é feita ao sistema. Esta nova solicitação inicia depois do fim da solicitação previamente agendada (Figura 26);



8. Existem duas solicitações já agendadas e, uma nova solicitação é feita ao sistema. Esta nova solicitação inicia e termina no meio do intervalo entre as duas previamente agendadas. Possui folga no início e no término (Figura 27);



9. Existem duas solicitações já agendadas e, uma nova solicitação é feita ao sistema. Esta nova solicitação inicia e termina no intervalo entre as duas previamente agendadas. Não possui folga após a primeira e nem antes da última (Figura 28).

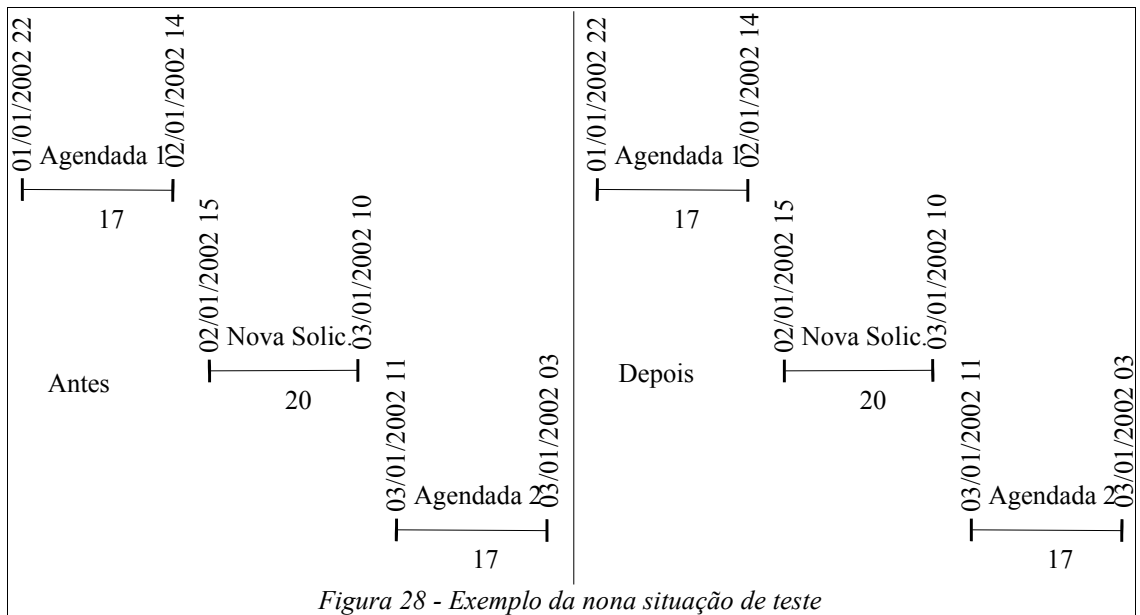


Figura 28 - Exemplo da nona situação de teste

10. Existem duas solicitações já agendadas e, uma nova solicitação é feita ao sistema. Esta nova solicitação inicia no meio do período agendado para a primeira solicitação e, não precisa deslocar o agendamento da segunda, ficando no meio das duas previamente alocadas sem folga (Figura 29);

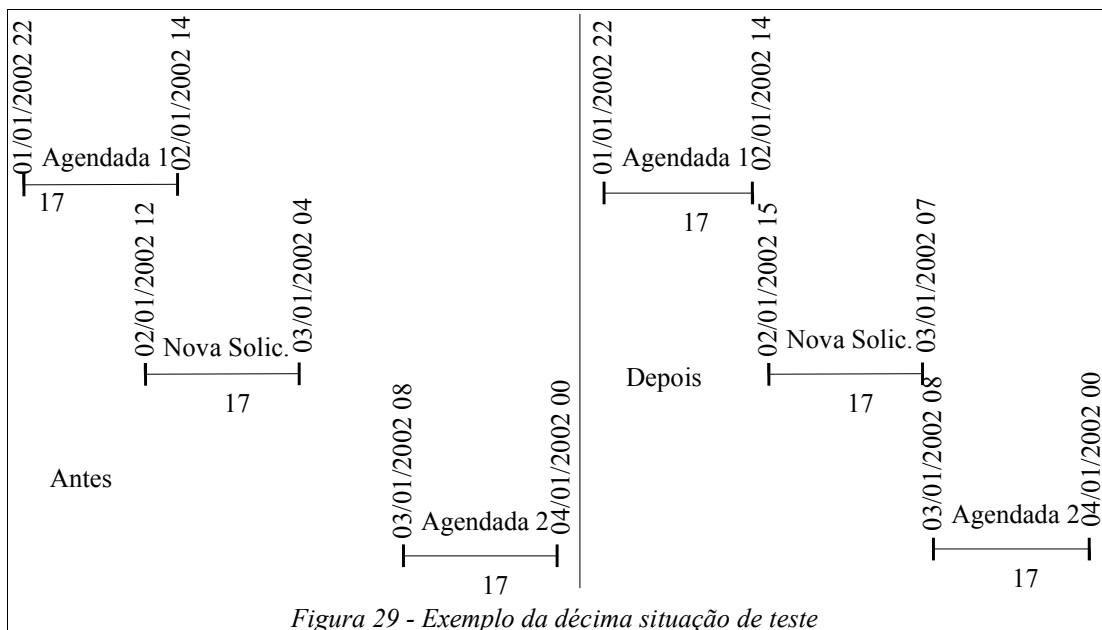


Figura 29 - Exemplo da décima situação de teste

11. Existem duas solicitações já agendadas e, uma nova solicitação é feita ao sistema. Esta nova solicitação inicia antes da primeira solicitação agendada, e provoca, pelo deslocamento do agendamento da primeira, o deslocamento do início da segunda solicitação previamente agendada também (Figura 30);

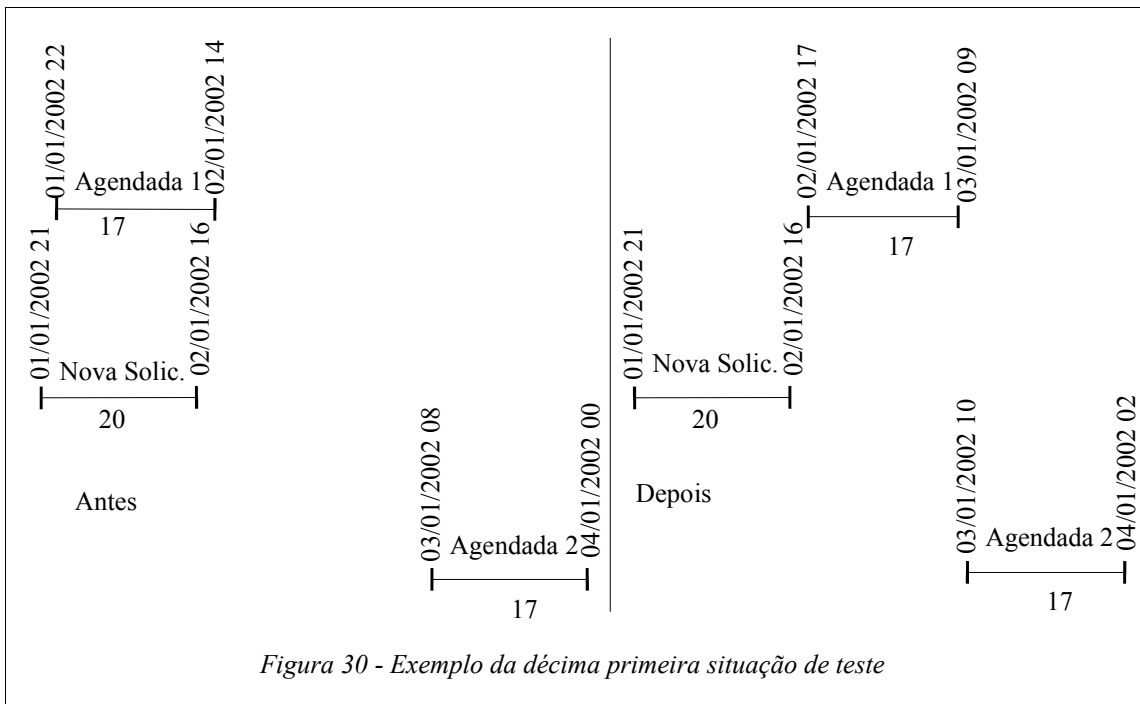


Figura 30 - Exemplo da décima primeira situação de teste

12. Existem duas solicitações já agendadas e, uma nova solicitação é feita ao sistema. Esta nova solicitação inicia antes da primeira solicitação agendada, e provoca o deslocamento do agendamento da primeira e, a faz encostar no início do agendamento da segunda solicitação previamente agendada (Figura 31);

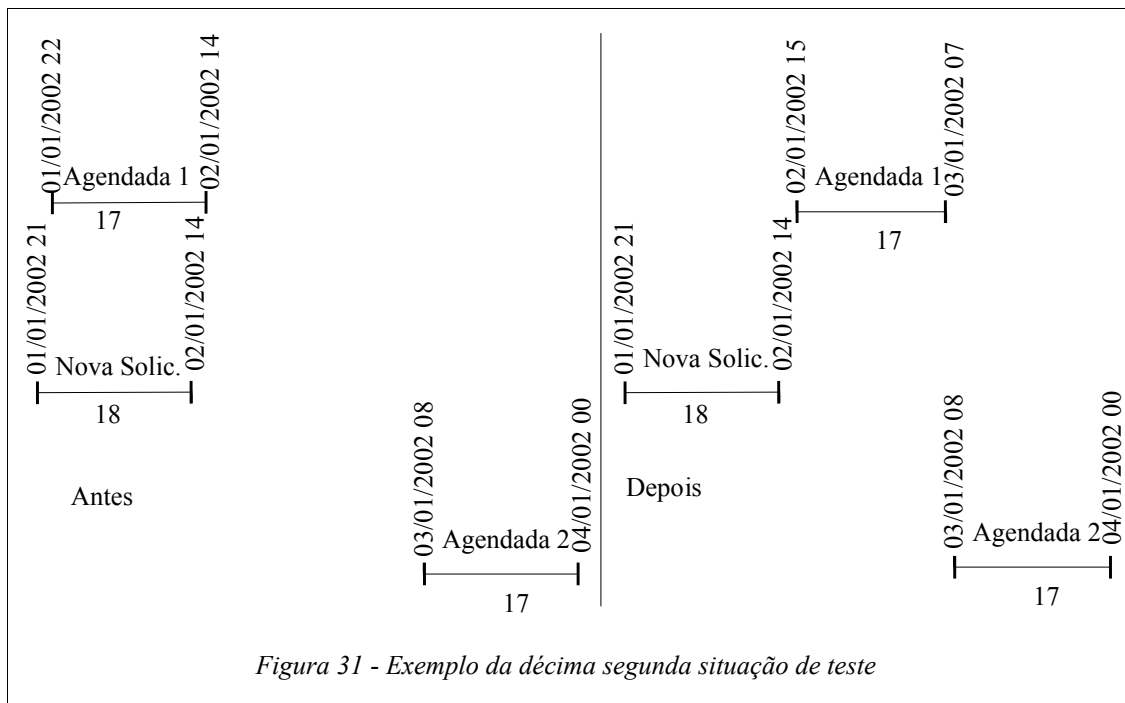
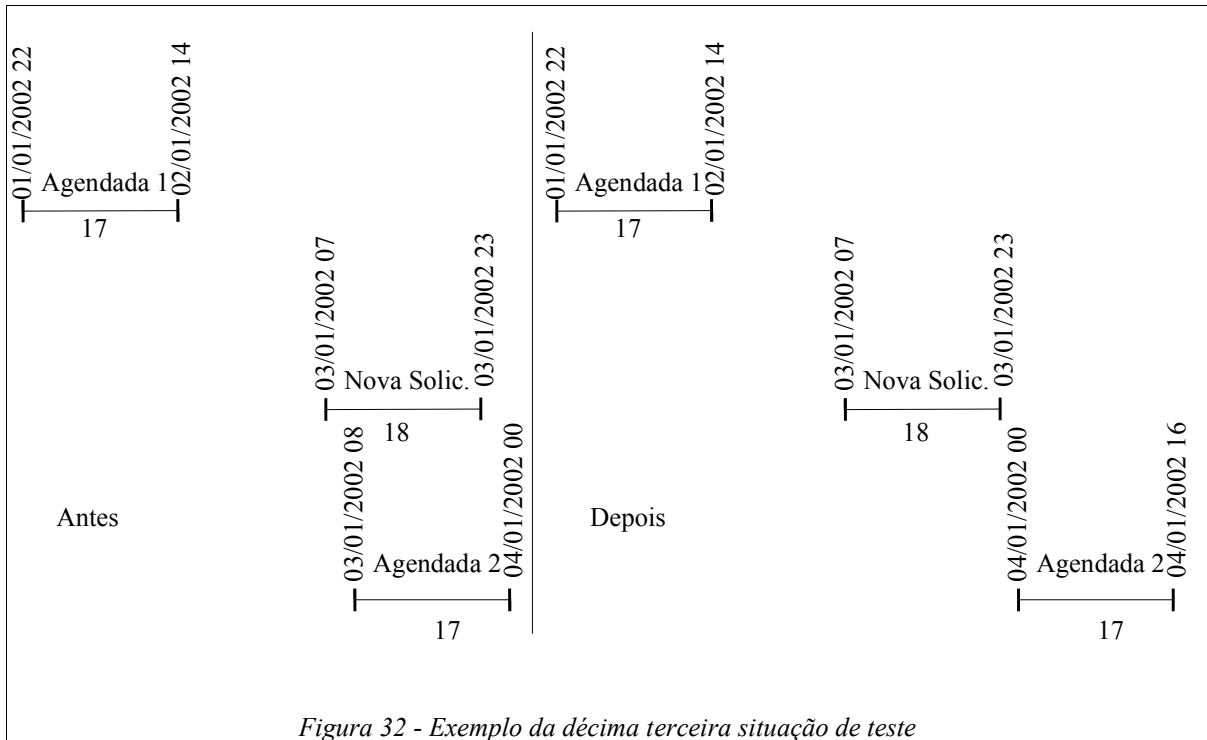


Figura 31 - Exemplo da décima segunda situação de teste

13. Existem duas solicitações já agendadas e, uma nova solicitação é feita ao sistema. Esta nova solicitação inicia após o término da primeira solicitação agendada, e antes do início da segunda já agendada, provocando o deslocamento do agendamento da segunda (Figura 32).



A partir do levantamento das treze situações de teste, foi elaborado um programa de computador que realiza os treze testes anteriormente citados e que verifica se os valores de saída conferem com os valores esperados de saída. Este programa foi escrito na linguagem de programação Java usando a biblioteca JUnit.

Como visto anteriormente, é a biblioteca JUnit que fornece as classes para que efetivamente seja feito a bancada de testes da lógica de programação. O programa desenvolvido apresenta uma tela gráfica gerada a partir da biblioteca JUnit que pode ser vista na Figura 33 onde é dado destaque para os treze testes realizados.

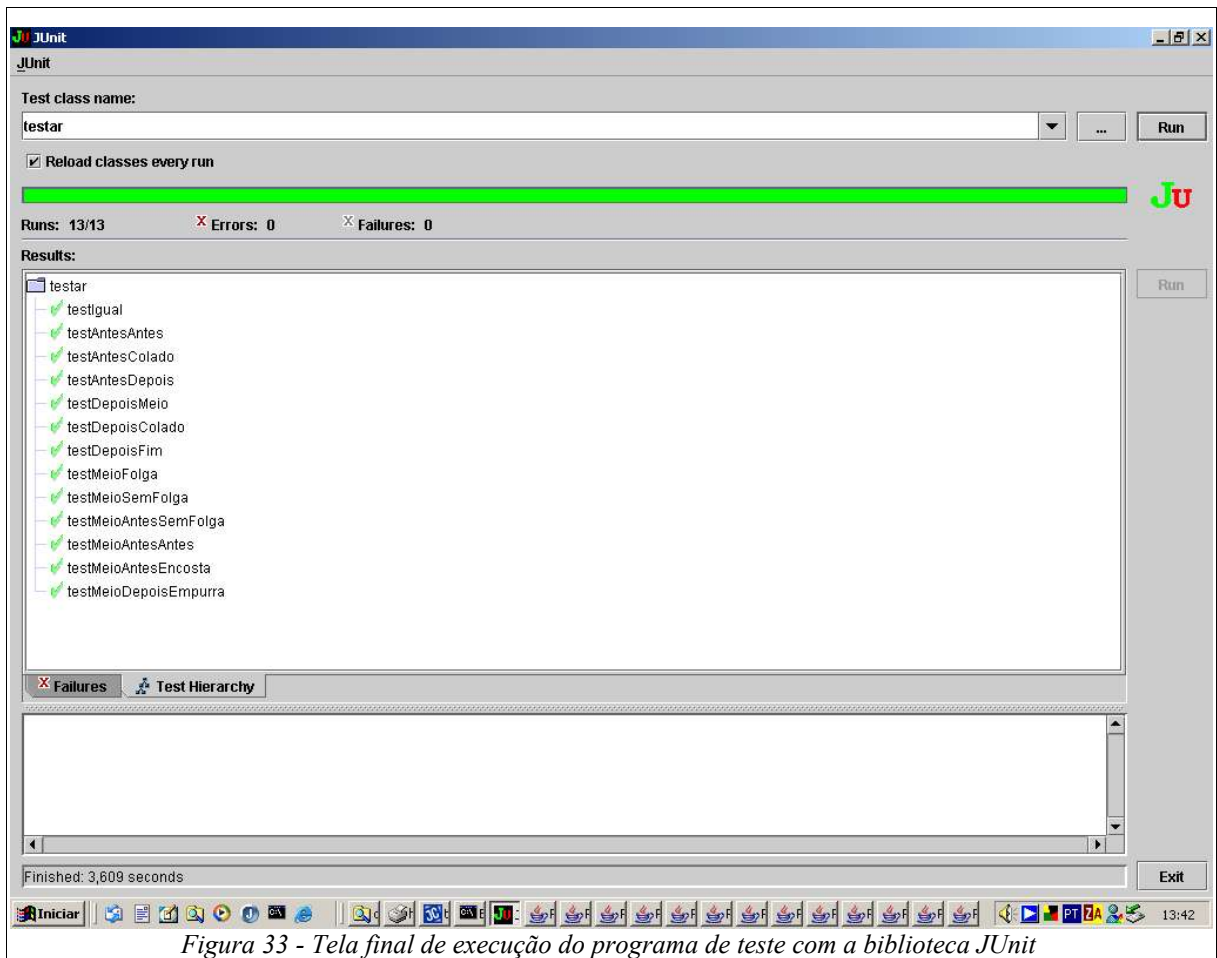


Figura 33 - Tela final de execução do programa de teste com a biblioteca JUnit

A partir destes testes, verificou-se que nas treze situações elencadas, o sistema funcionou em termos de lógica computacional conforme o que se esperava. Na próxima seção, serão apresentados os testes do sistema multiagentes.

5.2) *Validação do funcionamento do sistema multiagentes através do estudo de caso do Terminal Marítimo de Contêiner*

A partir deste ponto, é realizada a validação do funcionamento do sistema distribuído no que tange às questões de negociação e troca de mensagens. Para os testes foram feitas simulações de geração da fila de navios de um Terminal Marítimo de Contêiner e, também, as análises dos resultados de cada um dos dois padrões de negociação e uma comparação entre eles e os resultados alcançados por um Modelo de Referência.

O Modelo de Referência foi elaborado a partir de operações portuárias típicas. Numa operação portuária típica, a fila de navios, que representa a sequência de atendimento aos navios que solicitam atracação e operação no terminal, é gerada a partir da disponibilidade, na data solicitada, de berço de atracação, de áreas de pátio e de equipamentos para operar o berço e o pátio. Caso existam os recursos citados, faz-se a reserva deles de tal maneira que quando o navio chegar para operar, todos os recursos estejam disponíveis para atendê-lo, pois caso contrário ele não poderá iniciar sua operação, quer seja por falta de equipamentos, ou por falta de área no pátio ou de berço de atracação.

Numa operação portuária típica ocorre a possibilidade de um navio ter sua data de chegada alterada em função de uma operação que tenha demorado mais em um porto ou por problemas náuticos e, assim sendo, pode haver a necessidade de alteração da fila de navio. Portanto, a fila de navios é dinâmica.

O Modelo de Referência foi elaborado com base na descrição anteriormente feita de uma operação portuária típica. No entanto, ele não contempla a questão da mesma ser dinâmica, e na sua elaboração foi criado um ambiente de simulação onde foram registrados todos os recursos existentes no porto e suas regras de utilização.

Desta forma, as principais características do Modelo de Referência são, ver Figura 34:

1. Critério *First In-First Served*;
2. Critério baseado na data da solicitação;
3. Estático, não há replanejamento;
4. Só agenda os navios caso todos os recursos necessários possam ser agendados;
5. Descreve o funcionamento adotado por uma operação portuária típica;
6. Minimiza o *flow time*.

Modelo de Referência

- Critério *First In-First Served*
- Critério baseado na data da solicitação
- Estático, não há replanejamento
- Só agenda os navios caso todos os recursos necessários possam ser agendados
- Descreve o funcionamento adotado por uma operação portuária típica
- Minimiza o *flow time*

Figura 34 - Principais características do Modelo de Referência

O Modelo de Referência foi construído e executado fazendo uso do software de simulação Arena (KELTON et. al., 2004) (AMORIM, 2005) que simula de forma estática o recebimento de solicitações para operar no porto.

Tanto para o Modelo de Referência quanto para os dois protocolos de negociação propostos foram elaborados alguns cenários de testes. As características destes cenários são descritas a seguir.

Como características do tamanho do porto, usou-se três configurações de berço de atracação:

1. Um berço, cenário bem restritivo;
2. Três berços;
3. Cinco berços.

Para as três características do tamanho do porto, gerou-se a chegada de navios, solicitações de atracação, com base em uma distribuição exponencial. Foram geradas três seqüências seguindo as seguintes características da distribuição exponencial:

1. Com média de tempo entre dois navios igual a seis horas;
2. Com média de tempo entre dois navios igual a doze horas;
3. Com média de tempo entre dois navios igual a três horas. Esta seqüências é usada somente no Cenário 25 com o objetivo de testar uma saturação muito grande de chegada de navios ao porto.

Para cada umas destas seqüências, foram gerados os tempos de operação de cada solicitação baseados numa distribuição normal. Para tal, foram gerados dois tempos de operação seguindo as seguintes características da distribuição normal:

1. Com média de tempo de operação de dezoito horas e com desvio padrão de seis horas;
2. Com média de tempo de operação de trinta e seis horas e com desvio padrão de doze horas.

Nestes cenários, foi estabelecido um limite máximo de vinte e quatro horas entre a data de solicitação de atracação e a data possível de atendimento. Caso não seja possível atender dentro deste limite, o navio não será agendado para operar no porto.

No caso do protocolo de negociação com o critério de desempate baseado na emoção, os valores do Grau de Gratidão foram gerados aleatoriamente. Esta geração ocorre no momento em que o agente é inicializado.

Para elaboração da base de testes, tanto para os protocolos propostos como para o Modelo de Referência, foram gerados e utilizados vinte e cinco cenários conforme explicado anteriormente, ver Tabela 2.

Cenários de teste	Tamanho do Porto – Número de Berços	Número de Solicitações de Atracação	Seqüência de Chegada Distrib. Exponencial Tempo Médio entre Navios	Tempo de Operação Distribuição Normal	
				Média	Desvio Padrão
1	1	200	6	18	6
2	1	200	6	36	12
3	1	200	12	18	6
4	1	200	12	36	12
5	3	200	6	18	6
6	3	200	6	36	12
7	3	200	12	18	6
8	3	200	12	36	12
9	5	200	6	18	6
10	5	200	6	36	12
11	5	200	12	18	6
12	5	200	12	36	12
13	3	500	6	18	6
14	3	500	6	36	12
15	3	500	12	18	6
16	3	500	12	36	12
17	5	500	6	18	6
18	5	500	6	36	12
19	5	500	12	18	6
20	5	500	12	36	12
21	5	1000	6	18	6
22	5	1000	6	36	12
23	5	1000	12	18	6
24	5	1000	12	36	12
25	5	1000	3	36	12

Tabela 2 - Cenários de Teste para a Abordagem Proposta

Foram realizadas dez rodadas de simulação para cada um dos vinte e cinco cenários, pois os resultados dos testes não são determinísticos. Os resultados dos protocolos propostos foram

comparados com os resultados obtidos pelo Modelo de Referência. Para análise dos resultados, foram definidas três métricas:

1. *Flow time*. O *flow time* avalia o tempo entre a data de solicitação de atracação e a data efetiva de atracação. Em termos práticos, o *flow time* representa o tempo que o navio espera desde sua chegada à área do porto até sua atracação efetiva;
2. Número de solicitações não atendidas. Este valor mede a capacidade do sistema de agendar o máximo possível de navios no porto respeitando o limite máximo de vinte e quatro horas entre a data de solicitação de atracação e a data possível de atendimento;
3. Número de mensagens realizadas. O número de mensagens realizadas representa o número total de mensagens e indica o esforço do sistema proposto para chegar à solução final.

Os resultados alcançados para cada um dos vinte e cinco cenários são apresentados nas tabelas de número três à vinte e sete.

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	14,6	0,8	132,1	0,7	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	15,1	1,1	132,1	0,7	3.368,3	38,1
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	15,4	1,2	132,1	0,7	4.184,6	41,2

Tabela 3 - Resultados do Cenário de Teste 1

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	16,3	0,8	167,4	0,8	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	16,6	1,1	167,4	0,8	3.794,7	37,9
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	16,8	1,3	167,4	0,8	4.642,2	52,2

Tabela 4 - Resultados do Cenário de Teste 2

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	10,1	0,6	79,1	0,2	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	10,3	0,8	79,1	0,2	2.914,3	25,2
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	10,3	0,8	79,1	0,2	3.921,3	39,8

Tabela 5 - Resultados do Cenário de Teste 3

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	15,1	0,9	135,2	0,8	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	15,2	1,0	135,2	0,8	3.520,3	36,1
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	15,3	1,1	135,2	0,8	4.232,2	43,4

Tabela 6 - Resultados do Cenário de Teste 4

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	5,9	0,4	20,1	0,1	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	6,2	0,6	20,1	0,1	1.418,9	18,1
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	6,2	0,6	20,1	0,1	2.037,4	22,1

Tabela 7 - Resultados do Cenário de Teste 5

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	8,8	1,1	96,1	0,2	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	9,1	1,3	96,1	0,2	2.724,8	34,1
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	9,2	1,4	96,1	0,2	3.421,4	40,7

Tabela 8 - Resultados do Cenário de Teste 6

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	4,6	0,2	0,0	0,0	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	4,6	0,2	0,0	0,0	1.840,7	16,1
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	4,6	0,2	0,0	0,0	2.100,2	22,3

Tabela 9 - Resultados do Cenário de Teste 7

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	6,1	0,3	37,1	0,1	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	6,3	0,5	37,1	0,1	2.301,2	27,0
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	6,4	0,5	37,1	0,1	2.947,4	36,3

Tabela 10 - Resultados do Cenário de Teste 8

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	4,8	0,1	0,0	0,0	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	5,0	0,2	0,0	0,0	1.820,2	18,4
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	5,0	0,2	0,0	0,0	1.930,1	23,1

Tabela 11 - Resultados do Cenário de Teste 9

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	6,1	0,2	34,1	0,1	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	6,3	0,4	34,1	0,1	2.399,9	19,2
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	6,4	0,4	34,1	0,1	3.039,8	29,6

Tabela 12 - Resultados do Cenário de Teste 10

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	4,1	0,1	0,0	0,0	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	4,1	0,3	0,0	0,0	1.812,4	22,2
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	4,2	0,3	0,0	0,0	1.903,9	24,7

Tabela 13 - Resultados do Cenário de Teste 11

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	4,4	0,4	1,0	0,0	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	4,6	0,5	1,0	0,0	1.901,4	25,9
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	4,6	0,5	1,0	0,0	2.179,3	32,7

Tabela 14 - Resultados do Cenário de Teste 12

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	6,1	0,4	47,1	0,1	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	6,2	0,5	47,1	0,1	6.369,9	74,3
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	6,2	0,6	47,1	0,1	7.710,5	95,8

Tabela 15 - Resultados do Cenário de Teste 13

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	9,5	1,3	243,8	0,9	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	9,6	1,3	243,8	0,9	8.673,4	68,4
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	10,4	1,7	243,8	0,9	9.996,4	108,1

Tabela 16 - Resultados do Cenário de Teste 14

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	5,1	0,3	1,0	0,0	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	5,4	0,5	1,0	0,0	5.700,1	79,3
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	5,4	0,5	1,0	0,0	6.307,3	91,2

Tabela 17 - Resultados do Cenário de Teste 15

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	7,3	1,1	82,1	0,2	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	7,4	1,2	82,1	0,2	6.679,7	90,3
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	7,5	1,3	82,1	0,2	7.932,4	109,1

Tabela 18 - Resultados do Cenário de Teste 16

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	4,7	0,2	0,0	0,0	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	4,9	0,3	0,0	0,0	5.404,5	49,1
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	4,9	0,3	0,0	0,0	6.391,2	61,2

Tabela 19 - Resultados do Cenário de Teste 17

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	5,4	0,2	88,2	0,4	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	5,6	0,3	88,2	0,4	8.209,3	99,7
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	5,8	0,4	88,2	0,4	9.122,3	101,7

Tabela 20 - Resultados do Cenário de Teste 18

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	4,2	0,1	0,0	0,0	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	4,3	0,3	0,0	0,0	5.101,1	47,2
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	4,3	0,3	0,0	0,0	6.104,1	58,1

Tabela 21 - Resultados do Cenário de Teste 19

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	4,4	0,1	5,0	0,0	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	4,8	0,3	5,0	0,0	6.341,7	81,4
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	4,8	0,3	5,0	0,0	7.797,1	104,6

Tabela 22 - Resultados do Cenário de Teste 20

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	5,1	0,3	0,0	0,0	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	5,2	0,4	0,0	0,0	11.420,0	79,3
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	5,2	0,4	0,0	0,0	12.541,4	89,1

Tabela 23 - Resultados do Cenário de Teste 21

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	9,1	0,8	179,6	0,8	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	10,2	1,0	179,6	0,8	14.081,2	142,1
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	11,4	1,3	179,6	0,8	15.494,3	173,2

Tabela 24 - Resultados do Cenário de Teste 22

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	4,8	0,2	0,0	0,0	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	5,1	0,2	0,0	0,0	11.106,0	77,4
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	5,1	0,2	0,0	0,0	12.207,4	88,3

Tabela 25 - Resultados do Cenário de Teste 23

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	4,9	0,4	7,0	0,0	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	5,2	0,4	7,0	0,0	11.773,2	77,4
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	5,2	0,5	7,0	0,0	12.617,3	107,1

Tabela 26 - Resultados do Cenário de Teste 24

Critério de Agendamento	<i>Flow Time</i>		Numero de Solicitações Não Atendidas		Número de Mensagens Realizadas	
	Média (Horas)	Desvio Padrão	Média (Un.)	Desvio Padrão	Média (Un)	Desvio Padrão
Modelo de Referência	9,9	1,1	577,7	1,4	-----	-----
1- Com a possibilidade de deslocar em função da data de chegada	10,4	1,7	577,7	1,4	16.129,9	178,4
2 - Com a possibilidade de deslocar em função da data de chegada usando como critério de desempate a emoção	12,1	2,1	577,7	1,4	19.122,3	196,5

Tabela 27 - Resultados do Cenário de Teste 25

A partir dos resultados apresentados nas tabelas três a vinte e sete é feita a análise sob a ótica das três métricas definidas anteriormente:

1. *Flow time*;
2. Numero de Solicitações Não Atendidas;
3. Número de Mensagens Realizadas.

Em relação ao *flow chart*, os dois protocolos de negociação propostos se aproximam muito do valor do Modelo de Referência e possuem uma variabilidade muito próxima ao mesmo. Esta variabilidade, medida pelo desvio padrão, tanto no Modelo de Referência como nos dois protocolos propostos, é causada pela distribuição exponencial que gera valores mais perto um dos outros no início e valores mais espaçados no fim da distribuição.

Outro fato importante é que os dois protocolos de negociação obtiveram resultados semelhantes entre si e o Modelo de Referência independente do número de solicitações e, conforme visto anteriormente, só é afetado pelas mesmas condições do Modelo de Referência.

Quanto ao número de solicitações não atendidas, situação imposta pelo limite de vinte quatro horas para os testes, os dois protocolos de negociação alcançaram em todos os testes valores iguais aos encontrados pelo Modelo de Referência, inclusive numa análise feita por cada rodada de simulação, o número de solicitações foi sempre o mesmo. A variabilidade é igual em todos os testes para o Modelo de Referência e para os dois protocolos propostos. Ela é ocasionada em função dos dados de entrada serem diferentes nas dez rodadas de simulação.

Para os dois protocolos propostos, o número de mensagens realizadas, em valores absolutos, apresenta valores que chegam ao máximo de 22.395 para o cenário 25 com 1.000 solicitações, o que aparentemente é um valor alto. No entanto, pode-se perceber que para cada solicitação, o sistema multiagentes deve gerar uma mensagem de solicitação do agente agência para o agente berço, uma mensagem de solicitação do agente berço para o agente pátio, uma mensagem de solicitação do agente berço para o agente equipamento e uma mensagem de solicitação do agente pátio para o agente equipamento. Além disso, para cada uma destas quatro mensagens, deve-se ter uma mensagem de confirmação de atendimento a solicitação, portanto mais quatro mensagens. Então, para cada solicitação no mínimo oito solicitações devem ocorrer.

Soma-se a isso, o fato que a cada mensagem de sugestão, no mínimo mais oito mensagens devem ocorrer. Assim sendo, para os cenários onde ocorrem muita negociação é de se esperar que o valor do número de mensagens realizadas aumente.

Este número aumenta em função da intensidade de negociação que porventura venha ocorrer. No entanto, em todos os testes realizados os dois protocolos propostos convergiram para o resultado final.

Aliado a isso, deve-se ressaltar, que com exceção das mensagens de solicitações, todas as outras mensagens são formadas por poucos caracteres, a de confirmação possui somente o código da solicitação, três caracteres, a de sugestão possui o código da solicitação e a data sugerida, onze caracteres. Portanto, o impacto delas no sistema não é tão grande e, em contrapartida, ganha-se na distribuição dos processos, o que não ocorre no Modelo de Referência.

Vale observar que o valor do desvio padrão para o número de mensagens realizadas é inerente ao fato das entradas de dados serem diferentes para as dez rodadas de simulação. Este fato acarreta variados níveis de negociação em função das datas de solicitação serem diferentes em cada rodada de simulação levando ao aumento do desvio padrão registrado após as dez rodadas de simulação.

Os resultados do protocolo 2 reforçam a idéia que sua maior virtude não é melhorar resultados, mas, sim, ser um elemento de desempate em conflitos que possam ocorrer em um ambiente de negociação. Isto pode ser melhor notado quando é feita a análise caso a caso, onde é observado que apesar do número de solicitações não atendidas serem iguais, as solicitações atendidas não são as mesmas, pois elas dependem do Grau de Gratidão gerados aleatoriamente no início do processamento e, também, do desenrolar das negociações quando vai ocorrendo as modificações do Grau de Gratidão.

Nos cenários 1 a 25, os dois protocolos propostos se apresentam de forma estável nos testes, podendo ser, ambos, alternativas interessantes para resolução desta classe de problema, alcançando, inclusive, resultados muito próximos aos alcançados pelo Modelo de Referência, que utiliza as técnicas de simulação, atualmente, largamente difundidas no setor portuário.

Capítulo 6 - Conclusões

Foi apresentado nesta tese uma proposta de abordagem inspirada na sociedade humana para solução de problemas de agendamento em ambientes dinâmicos e complexos. Além disso, foram propostos três padrões de negociação, dando destaque a proposta inovadora de um padrão de negociação inspirado nas emoções humanas.

Desta proposta foi elaborada uma abordagem e, também, foi desenvolvido um sistema computacional que utiliza a técnica de agentes e sistemas multiagentes. Com este sistema, foram realizados diversos testes experimentais e os resultados apresentados no Capítulo 4.

Este sistema computacional foi aplicado, também, num estudo de caso de um Terminal Marítimo de Contêiner onde se busca uma solução do problema de elaboração da fila de navios do terminal. A aplicação desta abordagem a este estudo de caso, também, é inovadora.

O maior objetivo da abordagem não é obter resultados ótimos, mas sim, obter resultados tão bons ou melhores que os conseguidos na prática proporcionando a elaboração de ferramentas computacionais que possam, eventualmente, substituir o ser humano sem haver perda de qualidade do serviço prestado. Esta substituição pode gerar os seguintes ganhos: 1) Redução de recursos aplicados na negociação; 2) Redução no nível de estresse dos envolvidos nesta negociação; 3) Agilidade no processo de negociação através da automação completa do processo.

Um outro ponto importante que advém desta abordagem é que os padrões de interação e negociação na sociedade humana podem ser usados como fonte de inspiração para abordagens de solução de problemas complexos sem se valer de métodos matemáticos robustos. Além disso, após os testes pode-se observar que a escolha do padrão de negociação é questão fundamental para o bom resultado da abordagem proposta.

Portanto, como a abordagem proposta incorpora padrões de negociação como sendo um complemento dela, é possível inserir nela novos padrões de negociação, além dos três padrões modelados e testados, e tratar outros problemas analisando somente o padrão de negociação utilizado no negócio e inserí-lo na abordagem. Com isto, a abordagem pode ser aplicada aos mais diversos tipos de problema que envolvam agendamento de recursos e negociação para se alcançar este agendamento.

A abordagem proposta alcançou bons resultados quando comparados com o Modelo de referência. Vale ressaltar que os resultados dos dois protocolos propostos na abordagem da tese não teve qualquer interferência humana. Além disso, apesar das incertezas inerentes a um sistema distribuído, garantia de entrega ordenada das mensagens, o sistema computacional se mostrou estável nas diversas repetições dos testes.

Vale ressaltar que muitos destes métodos centralizados e de abordagem matemática nem sempre conseguem resolver os problemas dinâmicos em ambientes complexos. Nestes ambientes a informação para a solução do problema não se apresenta de forma completa e sempre estão sendo alteradas o que dificulta sobremaneira a solução pelos métodos citados. Já no caso da abordagem proposta como a solução emana da negociação e da troca de propostas e contra-propostas entre os agentes, os mesmos não precisam das informações globais e podem ir se adaptando ao longo do tempo e se ajustando às modificações.

A abordagem proposta foi aplicada a outros estudos de caso. Por exemplo, estão concluídos os testes aplicados ao problema de agendamento da recepção de caminhões para carregar e descarregar em um centro de distribuição, apresentando resultados tão bons quanto aos encontrados no mundo real (ROSA, SCHNEEBELI, 2004).

Outros trabalhos em ambientes complexos que envolvem negociação vem sendo estudados e testados, também, em paralelo ao estudo de caso específico desta tese, nas áreas de: 1) Alocação dinâmica de caminhões de uma frota de entrega e coleta, 2) Reposição de estoques, 3) Distribuição dinâmica de reservas de quartos em um sistema de consórcio hoteleiro.

6.1) Trabalhos Futuros

Como trabalho futuro, espera-se analisar outros problemas logísticos e com isso modelar outros padrões de negociação e avaliar os resultados encontrados. Com isto, espera-se criar uma biblioteca de padrões de negociação que possa dar suporte a um grande número de problemas logísticos.

Com base nestas bibliotecas, espera-se criar uma interface gráfica para configuração do problema logístico que se deseja solucionar e, assim, poder distribuir uma ferramenta genérica aplicada a problemas logísticos que possa ser usada por qualquer usuário.

Além disto, deve-se analisar outras emoções que desempenham papel importante nas relações sociais, dentre estas citam-se o medo, a raiva, a vaidade. Com esta análise, pode-se propor mecanismos de negociação que tenham como base mais de uma emoção e avaliar se os mesmos trazem melhores resultados do que mecanismos baseados em somente uma emoção.

Por fim, deve-se buscar um mecanismo interno a cada um agente que consiga perceber, avaliar e decidir por um mecanismo de negociação dinamicamente, visando assim, criar um agente que se assemelha a um ser humano que consegue se adaptar a diversos ambientes de negociação como, por exemplo, sua família, seu emprego, o trânsito no qual ele dirige seu carro entre outros.

Referências

- ADAMATI, D., BAZZAN, A., "A Framework for Simulation of Agents with Emotions", WORKCOMP 2002 (Workshop de Computacao), ITA, São José dos Campos, Sao Paulo, 2002.
- AMORIM, R., "Estudo do Transporte de Produtos Siderúrgicos em Viagem Cíclicas de Comboios Oceânicos", Dissertação de Mestrado, Universidade Federal do Espírito Santo, Programa de Pós-Graduação em Engenharia Civil, UFES, 2005.
- AGERSCHOU, H., "Planning and design of ports and marine terminals", Willey Interscience, 1983.
- BAKER, A., "A Survey of Factory Control Algorithms that can be implemented in a Multi-Agent Heterarchy: Dispatching, scheduling, and pull," Journal of Manufacturing Systems, vol. 17, no. 4, pp. 297-320, 1998.
- BAKER, K., "Introduction to Sequencing and Scheduling", John Wiley and Sons, New York, 1974.
- BLAZEWICZ, J., ECKER, K., PESCH, E., SCHMIDT, W., WEGLARZ, J. "Scheduling Computer and Manufacturing Process", Springer Verlag, Heidelberg, 1996.
- BOND, A., GASSER, L., "Readings in Distributed Artificial Intelligence", ed. Morgan Kaufman, 1988.
- BURKE, P., PROSSER, P., "A Distributed Asynchronous System for Predictive and Reactive Scheduling", Artificial Intelligence in Engineering, 6(3):106-123, 1991.
- CASARE, S., "Uma Ontologia Funcional de Reputação para Agentes", Dissertação de Mestrado, Escola Politécnica, Universidade de São Paulo, USP, 2005.
- CASTILHO, B., DAGANZO, C., "Handling Strategies for Import Containers at Marine Terminals", Transportation Research, 27B-2, 151-166, 1993.
- CHAIB-DRAA, B., "Industrial Applications of Distributed AI" In Huhns, M., Singh, M. "Readings In Agents", Morgan Kaufmann Publishers, USA, 1997

- CHAVEZ, A., MOUKAS, A., MAES, P. "Challenger: A Multi Agent System For Distributed Allocation", ACM, Agents 1997, Conference Proceedings.
- CLARK, M., "Junit Primer", Clarkware Consulting, 2000.
- CLORE, G., COLLINS, A., ORTONY, A., "The Cognitive Structure of Emotions", Cambridge University Press, 1990.
- CRAINIC, T., "Dynamic and stochastic models for the allocation of empty containers", Operations Research, 41-1, 102-126, 1993.
- CRUZ, M., "Uma Contribuição ao Estudo da Dinâmica de Sistemas de Terminais de Contêineres sob o Enfoque Sistêmico", Tese de Doutorado, COPPE, 1997.
- CRUZ, M., "Proposição de Um Sistema de Utilização Automática Racional da Área de Pré Ou Pós Estivagem de Contêineres em Terminais Especializados", Dissertação Mestrado, COPPE, 1993.
- DAMÁSIO, A., "Descartes Error, Emotion, Reason, and the Human Brain", Gosset/Putnam, NY, 1994.
- DAMIANI, E., PARABOSCHI, S., SAMARATI, P., VIOLANTE, F., "A reputation-based approach for choosing reliable resources in peer-to-peer networks", In Proceedings of the 9th ACM Conference on Computer and Communications Security, Pgs: 207-216, 2002.
- DECKER, K., LI, J., "Coordinating Mutually Exclusive Resources using GPGP", Journal of Autonomous Agents and Multi-Agent Systems, 3(2):133-157, Kluwer Academic Publishers, 2000.
- DOSHI, P., GMYTRASIEWICZ, P., "Towards Affect-based Approximation to Rational Planning", Architectures for Modeling Emotion: Cross-Disciplinary Foundations, American Association for Artificial Intelligence, 2004 Spring Symposium Series, Stanford University, Palo Alto, California, AAI-2004.
- DURKHEIN, É., "As regras do método sociológico", 4ª ed. São Paulo, Nacional, 1966.
- DURFEE, E., LESSER, V., CORKILL, D., "Trends in Cooperative Distributed Problem Solving", IEEE Transactions Knowledge Data Engineering, 11(1), 63-83, 1989.
- DURFEE, E., LESSER, V., CORKILL, D., "Coherent Cooperation among Communicating Problem Solvers", IEEE Transactions of Computers C-36(ii), pages 1275-1291, 1987.
- ECKEL, B., "Thinking In Java", 3rd Ed., Prentice Hall PTR, 2002.
- FEBER, J., "Multi Agent Systems An Introduction to Distributed Artificial Intelligence", Addison Wesley, 1999.
- FERREIRA, A. , "Novo Dicionario Aurelio da Lingua Portuguesa", Ed.: Nova Didática Ltda., 2004.

- FININ, T., WIEDERHOLD, G. , “Specification Of The KQML Agent-Communication Language Plus Example Policies And Architectures”, DARPA KSE External Interfaces Working Group Draft., 1994.
- GARCIA, A., SICHTMAN, J., “Agentes e Sistemas Multiagentes” in REZENDE, S., (Org.) “Sistemas Inteligentes, Fundamentos e Aplicações”, Ed. Manole, 2003.
- GASSER, L., “Social Conceptions of Knowledge and Action: DAI Foundations and Open Systems Semantics”, *Artificial Intelligence*, 47, 1991.
- GMYTRASIEWICZ, P., DOSHI, P., “A Framework for Sequential Planning in Multiagent Settings”, *Journal of AI Research (JAIR)*. Vol 24:1-31, 2005.
- GMYTRASIEWICZ , P., LISETTI, C., “Using Decision Theory to Formalize Emotions for Multi-Agent Systems”, *Second ICMAS-2000 Workshop on Game Theoretic and Decision Theoretic Agents*, Boston, 2000.
- GRAHAM, J., “Real-Time Scheduling In Distributed Multi Agent Systems”, PhD Thesis, University of Delaware, 2001.
- GRAHAM, J., “Real-Time Scheduling for Distributed Agents”, In *AAAI-Spring Symposium on Real-Time Autonomous Systems*, Palo Alto, CA, 2000.
- GRAHAM, J., DECKER, K., MERSIC, M., “DECAF - A Flexible Multi Agent System Architecture”, In *Proceedings of Autonomous Agents and Multi-Agent Systems*, 2003
- GRAHAM, J., MERSIC, M., DECKER, K., “Support for Resource Management in Multi Agent Systems”, Technical Report, Department of Information and Computer Science, University of Delaware, 2000.
- GRATCH, J., MARSELLA, S., “Evaluating the Modeling and Use of Emotion in Virtual Humans”, *Third International Joint Conference on Autonomous Agents & Multi Agent System*, New York, USA, 2004.
- HAIDT, J., “The Moral Emotions” In Davidson, R, Scherer, K., Goldsmith, H., (orgs.) “*Handbook of Affective Sciences*” Oxford University Press, 2003.
- HEE, K., HUITINICK, B., LEEGWATER, D., ”Portplan, Decision Support System for Port Terminals”, *European Journal of Operational Research*, 34, 249-261, 1988.
- HEE, K., WIJBRANDS, R., “Decision Support Systems For Container Terminal Planning”, *European Journal of Operational Research*, 34, 262-272, 1988.
- HENESEY, L., WERNSTEDT, F., DAVIDSSON, P., “Market-Driven Control in Container Terminal Management”, *2nd International Conference on Computer Applications and Information Technology in the Maritime Industries*, 2003.
- HENESEY, L., WERNSTEDT, F., DAVIDSSON, P., “A Market-Based Approach to Container Port Terminal Management”, *Proceedings of the ECAI-02 Workshop on Agent Technologies in Logistics*, 2002.

- HUHNS, M., SINGH, M., "Readings In Agents", Morgan Kaufmann Publishers, USA, 1997.
- HUHNS, M., STEPHENS, L., "Multiagent Systems and Societies of Agents", In WEIS, G. (Org.), "Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence", Cambridge, MIT Press, 2000.
- JENNINGS, N., "Controlling Cooperative Problem Solving In Industrial Multi-Agent Systems Using Joint Intentions", *Artificial Intelligence Journal*, 75(2):1-46, 1995.
- JENNINGS, N., SYCARA, K., WOOLDRIDGE, M., "A Roadmap Of Agent Research and Development", *Autonomous Agents And Multi-Agents Systems*, 1, 7-38, Kluwer Academic Publishers, Boston, 1998.
- KELTON, W., SADOWSKI, R., STURROCK, D., "Simulation with Arena", 3rd ed., McGrawHill, USA, 2004.
- KEN, A., GOSLING, J., HOLMES, D., "The Java Programming Language", 4rd ed., Prentice Hall, USA, 2006.
- KLEINGINNA, P., KLEINGINNA, A., "A Categorized List of Emotion Definitions with Suggestions for a Consensual Definition", *Motivation and Emotion*, 5, 345-379. 1981.
- KRAUS, S., "Strategic Negotiation In Multi-Agent Environments", MIT Press, Cambridge, USA, 2001.
- LAKATOS, E., "Sociologia Geral", 6^a Ed., Ed. Atlas, 1991.
- LEA, D., "Concurrent Programming in Java, Design Principles and Patterns", Addison-Wesley, 1997.
- LEWICKI, R., SAUNDERS, D., MINTON, J., "Fundamentos da Negociação", 2^a ed., Ed. Bookman, 2002.
- LIU, J., SYCARA, K., "Exploiting Problem Structure for Distributed Constraint Optimization", In *Proceedings of the First International Conference on Multi-Agent Systems*, San Francisco, CA, 1995.
- LIU, J., SYCARA, K., "Distributed Problem Solving Through Coordination in a Society of Agents", In *Proceedings of the 13th International Conference on Distributed Artificial Intelligence*, Seattle, WA, 1994.
- LIU, J., SYCARA, K., "Distributed Scheduling Through Cooperative Specialists", In *Proceedings of the IJCAI-93, Workshop on Knowledge-Based Production, Planning, Scheduling, and Control*, 1993.
- LIU, L., "Survey on Trust Management", disponível em <http://www.eas.asu.edu/~p2pcom/seminar/060203-lintao.ppt>, acessado em 01/agosto/2006.

- LORENZONI, L., “Problema de Escalonamento com Restrição de Recursos e Múltiplos Modos de Processamento: Novos Métodos de Resolução e uma Aplicação no Contexto Portuário”, UFES, Vitória/ES, setembro de 2003.
- MALONE, T., “Organizing Information Processing Systems: Parallels Between Human Organizations and Computer Systems. In Zachary, W., Robertson, S., (Org.) “Cognition, Computation and Cooperation”, Ablex, Norwood, NJ, 1990.
- MCGEARY, F., DECKER, K., “DECAF Programming: Agents for Undergraduates”, Workshop on Infrastructure for Scalable Multi-Agent Systems, 5th International Conference on Autonomous Agents, pp. 53-60, 2001.
- MCGEARY, F., “DECAF Programming: An Introduction“, Technical Report, Department of Information and Computer Science, University of Delaware, 2001.
- MIR, J., "Trust and Reputation for Agent Societs", Tesi Doctoral, Universita Autônoma de Barcelona, UAB, Escola Técnica Superior D'Enginyeria, Espanha, 2002.
- MOLDT, D., SCHEVE, C., “Emotions and Multimodal Interface-Agents: A Sociological View” In: Conference Proceedings of “Mensch & Computer 2001”. Bad Honnef (Bonn), Germany, Altos, 2001.
- MUI, L., "Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks", PhD Thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, USA, 2002.
- MULLER, J., “Negotiation Principles”, In O’Hare, G., Jennings, N., (Org.), “Foundation of Distributed Artificial Intelligence”, John Wiley & Sons, New York, 1996.
- MURRAY, I., ARNOTT, J., “Towards the Simulation of emotion in Synthetic Speech: A review of the Literature on Human Vocal Emotion“, In Journal of the Acoustic Society of America, pp. 1097-1108, 1993.
- NEIMAN, D., HILDUM, D., LESSER, V., SANDHOLM, T., “Exploiting Meta-Level Information in a Distributed Scheduling System” In Proc. Twelfth National Conference on Artificial Intelligence (AAAI 94), 1994.
- NEMANI, S., ALLAN, V., “Agents and the Algebra of Emotion”, Second International Joint Conference on Autonomous Agents and Multitagent Systems - AAMAS 2003, 2003.
- OLIVEIRA, E., SARMENTO, L., “Emotional Valence-based mechanisms and agent personality”, Proceedings of SBIA 2002.
- O’HARE, G., JENNINGS, N., “Foundations of Distributed Artificial Intelligence”, Ed. John Wiley E Sons, Inc., NY, 1996.
- PARUNACK, H., “Industrial and Pratical Applications of DAI”, In WEIS, G. (Org.), “Multiagent Systems, “A Modern Approach to Distributed Artificial Intelligence”. Cambridge, MIT Press, 2000.

- PARUNACK, H., "Manufacturing Experience with the Contract Net", In HUHNS, M. (Org.), "Distributed Artificial Intelligence", Pitman, Morgan Kaufmann, 1987.
- PERRET, M., "Using constraint logic programming techniques in container port planning", ICL Technical Journal, 7-3, 537-545, 1991.
- PRUITT, D., "Negotiation Behavior", New York: Academic Press, 1981.
- PRUITT, D., RUBIN, J., "Social Conflict: Escalation, Stalemate and Settlement", New York: Random House, 1986.
- REBOLLO, J., CARRASCOSA, C., BOTTI, V., "A MAS Approach for Port Container Terminal Management: The Transtainer Agent". Proceedings SCI 2001, pp. 1-5, 2001.
- REBOLLO, J., CARRASCOSA, C., BOTTI, V., "A Multiagent System for the Automation of a Port Container Terminal." In Proceeding Agents in Industry Workshop. Autonomous Agents 2000 (a).
- REBOLLO, J., CARRASCOSA, C., BOTTI, V., "A MAS Approach for Port Container Terminal Management.", Proceedings of the Third Iberoamerican Workshop on Distributed Artificial Intelligence and Multi-Agent Systems, pp.: 83-94. IBERAMIA 2000(b).
- REBOLLO, J., CARRASCOSA, C., BOTTI, V., "Un sistema multiagente para la automatización de terminales de contenedores.", Actas de la VIII Conferencia de la AEPIA Vol. 2, pp.: 54-62, 1999.
- REILLY, S., BATES, J., "Natural Negotiation for Believable Agents", Technical Report, CMU-CS-95-164, School of Computer Science, 1995.
- REIS, J., MAMEDE, N., "Scheduling, Re-scheduling and Communication in the Multi-agent Extended Enterprise Environment", 10th Portuguese Conference on Artificial Intelligence, EPIA 2001, Porto, Portugal, 2001.
- REIS, J., MAMEDE, N., "Multi-Agent Dynamic Scheduling and Re-Scheduling with Global Temporal Constraints", Proceedings of the 3rd International Conference on Enterprise Information Systems, Setubal, Portugal, 2001.
- ROSA, R., "Portos: Conceitos Essenciais", Instituto Histórico e Geográfico do Espírito Santo, 2006.
- ROSA, R., SCHNEEBELI, H., "Uma Abordagem Humana para Solução do Problema de Escalonamento Dinâmico com Restrição de Recursos. Estudo de Caso: Terminal Portuário de Contêiner", XXXVII SBPO – Congresso da Sociedade Brasileira de Pesquisa Operacional, 2005.
- ROSA, R., SCHNEEBELI, H., "Escalonamento da Descarga e Embarque de Caminhões em um Centro de Distribuição Usando Tecnologia de Agentes", XXXVI SBPO – Congresso da Sociedade Brasileira de Pesquisa Operacional, 2004.

- ROSA, R., SCHNEEBELI, H., “Abordagem Multi-Agentes para Solução do Problema de Escalamento Dinâmico Com Restrição De Recursos”, XXXV SBPO – Congresso da Sociedade Brasileira de Pesquisa Operacional, 2003 (a).
- ROSA, R., SCHNEEBELI, H., “Um Sistema Baseado em Agentes para Alocação de Recursos Aplicado a um Terminal Marítimo de Contêineres”, VI SBAI - Simpósio Brasileiro de Automação Inteligente, 2003 (b).
- RUDNICK, S., “Letting Computers Organize The Container Terminal”, *Engineering & Automation*, 13-4, 26-29, 1991.
- SANTOS, R., “Emotion Recognition in Speech Signal”, PhD. Thesis, Technical University of Madrid, Spain E.T.S.I. Telecomunicación, Departamento de Ingeniería Electrónica, 2001 .
- SHAW, M., WHINSTON, A., “Distributed Planning in Cellular Flexible Manufacturing Systems”, Technical Report, Management Information Research Center, Purdue University, 1983.
- SHAW, M., “Dynamic Scheduling in Cellular Manufacturing Systems: A Framework for Networked Decision Making”, *Journal of Manufacturing Systems*, 7(2), 83-94, 1988.
- SHEHORY, O., KRAUS, S., “Methods For Task Allocation Via Agent Coalition Formation”, Elsevier Science, 1998.
- SHEN, J., ZHANG, X., LESSER, V., “Degree of Local Cooperation and its Implication on Global Utility”, Third International Joint Conference on Autonomous Agents & Multi Agent System, New York, USA, 2004.
- SHEN, W., NORRIE, D., “Agent-Based Systems for Intelligent Manufacturing: A State of the Art Survey”, *Knowledge and Information Systems, International Journal*, 1(2),129-156, 1999.
- SINGH, R., SALAM, A., LAKSHMI, I., “Agents in E-Supply Chains”, *Communications of the ACM*, June, Vol 48(6), 2005.
- SMITH, R., “The Contract Net Protocol: High Level Communications and Control in a Distributed Problem Solver”, *IEEE Transactions on Computers*, C-29:12, 1104-1113, 1980.
- SMITH, R., DAVIS, R., “Frameworks for Cooperation in Distributed Problem Solving”, *IEEE Transactions on Systems, Man, and Cybernetics*, vol SMC-11, 1, 61-70, 1981.
- STIBBARD, R., “Vocal Expression of Emotions in Non-laboratory Speech: An Investigation of the Reading/Leeds Emotion in Speech Project Annotation Data”, PhD thesis, University of Reading, UK. 2001.
- SUBRAHMANIAN, V., BONATTI, P., DIX, J., EITER, T., KRAUS, S., OZCAN F., ROSS R., “Heterogeneous Agent Systems: Theory And Implementation”, MIT Press, Cambridge, USA, 2000.

- SUTHERLAND, J., HEUVEL, W., "Enterprise application integration and complex adaptive systems", *Communications of the ACM*, June, Vol 45(10), 2002.
- SYCARA, K., ROTH, S., SADEH, N., FOX, M. "Resource Allocation in Distributed Factory Scheduling" *IEEE Expert*, Vol. 6, No. 1, pp. 29-40, 1991.
- TALEB-IBRAHIMI, "Modeling And Analysis Of Container Handling In Ports", Ph.D. Thesis, Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA, 1989.
- THURSTON, T., HU, H., "Distributed Agent Architecture for Port Automation", *Proceedings of the 26th Annual Int. Computer Software and Applications Conference*, England, 26-29, 2002.
- UNCTAD, "Gestion Informatizada de las Terminales de Contenedores", 1994 (a).
- UNCTAD, "Intercambio de Datos Portuários Informatizado", 1994 (b).
- UNCTAD, "Planejamento de Terminais de Containeres Seminário IPP-2", 1996.
- VELÁSQUEZ, J., "Modeling Emotion-based decision making". In Dolores Cañanero (Editor), *Emotional and Intelligent: The tangled knot of cognition*, pg 164-169, 1998.
- VENTURA, R., "Emotion-based agents", M.Sc. Thesis, Instituto Superior Técnico, Lisboa, Portugal, 2001.
- VENTURA, R., Pinto-Ferreira, C., "Emotion-based Agents: Three approaches to implementation", Instituto de Sistemas e Robótica, 1999.
- WANG, Y., VASSILEVA, J., "Trust-Based Community Formation in Peer-to-Peer File Sharing Networks", In *Proceedings of the Web Intelligence, IEEE/WIC/ACM International Conference on (WI'04)*, Vol.00, Pgs: 341-348, 2004.
- WOOLDRIDGE, M., "Intelligent Agents", In WEIS, G. (Org.), "Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence", Cambridge, MIT Press, 2000.
- WRIGHT, I., "Emotional Agents", PhD Thesis, School of Computer Science, The University of Birmingham, 1997.
- WERNER, E., "Cooperating Agents: A Unified Theory of Communication and Social Structure", In Gasser, E., Huns, M., "Distributed Artificial Intelligence", Vol. II, Morgan Kaufmann, London, 1989.
- ZWEBEN, M., FOX, M., "Intelligent Scheduling", Morgan Kaufman Publishers, 1994.

Anexo A - Sistema Computacional para a Abordagem Proposta

Neste anexo são descritas as ferramentas computacionais empregadas no desenvolvimento do sistema computacional que suporta a abordagem proposta.

Para implementação do modelo do sistema multiagentes foi usada a ferramenta DECAF (*Distributed Environment Centered Agent Framework*) (Graham, 2001, 2000) (McGeary, 2001) (McGeary, Decker, 2001) (Graham, Decker, Mersic, 2003). O DECAF é uma ferramenta para desenvolvimento de agentes e sistemas multiagentes implementada em linguagem Java que provê algumas funcionalidades que agilizam o desenvolvimento e a execução dos testes dos agentes e do sistema multiagentes. Dentre estas funcionalidades, citam-se as três funções a seguir:

1. Desenhar graficamente a estrutura funcional do agente, plano do agente;
2. Disponibilizar a infra-estrutura para a execução de agentes, fluxo de métodos que devem ser executados pelo agente em função das mensagens recebidas e dos métodos;
3. Prover o sistema de um servidor de nomes de agentes.

O desenho da estrutura funcional do agente é representado na ferramenta DECAF como uma rede de tarefas hierarquizadas denominada *HTN* (*Hierarchical task network*) (Graham, 2001) que provê:

1. Uma estrutura útil para a coordenação do agente através da definição de como o agente funciona e como as tarefas que ele executa estão relacionadas;
2. Conceitos de adaptabilidade permitindo que vários resultados de saída e vários fluxos de dados possam ser modelados.

O DECAF disponibiliza uma ferramenta visual denominada *PlanEditor* para a representação da rede de tarefas. Esta ferramenta permite o desenvolvimento da definição do funcionamento do agente, de suas tarefas e seus relacionamentos com as outras tarefas e dos resultados e dos fluxos de dados dentro do agente, Figura 35, (Graham, 2001).

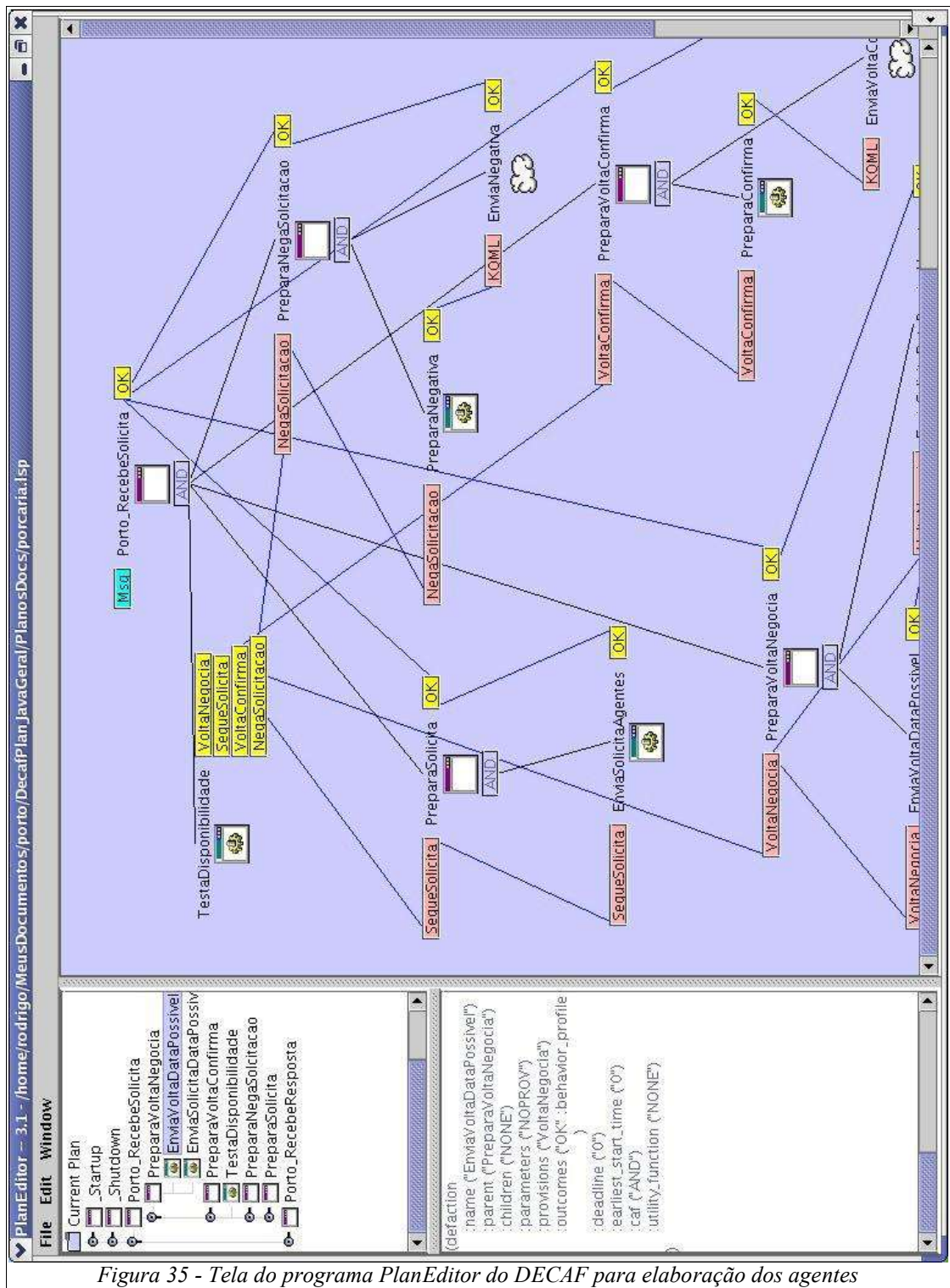


Figura 35 - Tela do programa PlanEditor do DECAF para elaboração dos agentes

Na tela do *PlanEditor* da Figura 35 podem ser vistas as seguintes áreas:




1. No canto superior esquerdo da tela está representada textualmente a hierarquia das tarefas;

2. No canto inferior esquerdo da tela é apresentada a linguagem que define o funcionamento de cada agente e suas tarefas. Esta linguagem é gerada automaticamente à medida que as tarefas são desenhadas graficamente no lado direito da tela.
3. No lado direito da tela é feita a representação gráfica do funcionamento do agente e que gera automaticamente o resultado textual no canto superior direito e o conteúdo da linguagem no canto esquerdo inferior.

O *PlanEditor* oferece também a possibilidade de gerar automaticamente o esqueleto do código em linguagem Java para implementação da lógica de cada uma das tarefas.

Para cada agente é definido um plano onde todo o funcionamento do agente está escrito e todas as tarefas e as interrelações entre elas estão definidas. O plano é de fato o arquivo texto gerado pela soma das definições de cada uma das tarefas descritas pela linguagem vista no canto inferior esquerdo da tela.

As figuras Figura 36 e Figura 37 representam um único plano DECAF. Elas foram divididas em duas figuras visando uma melhor visualização nesta tese. No plano DECAF existem três estruturas visuais importantes:

1. Tarefas, . As tarefas podem receber parâmetros que são variáveis representadas a esquerda do símbolo da tarefa e as variáveis de saída que estão representadas do lado direito do símbolo;
2. Ações, . As ações podem receber parâmetros que são variáveis representadas a esquerda do símbolo da ação e as variáveis de saída que estão representadas do lado direito do símbolo;
3. Tarefas não locais, . As tarefas não locais podem receber parâmetros que são variáveis representadas a esquerda do símbolo da tarefa não local e as variáveis de saída que estão representadas do lado direito do símbolo;

As linhas nas Figuras 36 e 37 representam as possíveis seqüências de processamento e que são ativadas através da passagem das variáveis de saída de uma tarefa ou ação ou tarefas não locais para os parâmetros de entrada das tarefas ou das ações ou das tarefas não locais que a linha esteja conectando.

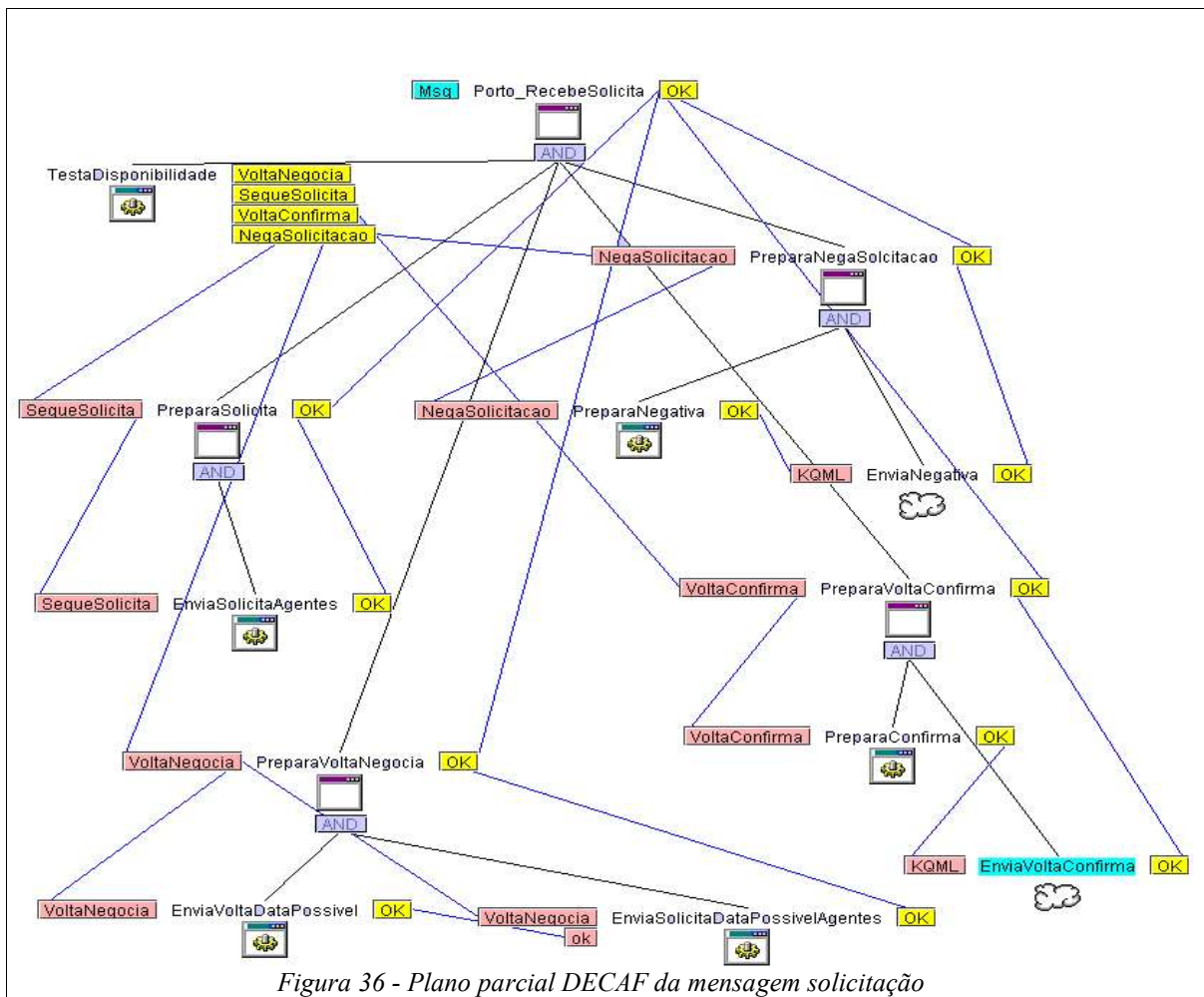
Toda tarefa deve ter pelo menos uma variável de saída e pode ou não ter uma variável de entrada. No entanto, caso ela possua variável de entrada, ela só será processada caso todas as variáveis de entrada tenham recebido valores das variáveis de saída nos quais estão conectados. Todo este mecanismo é controlado pelo DECAF em tempo de execução do agente.

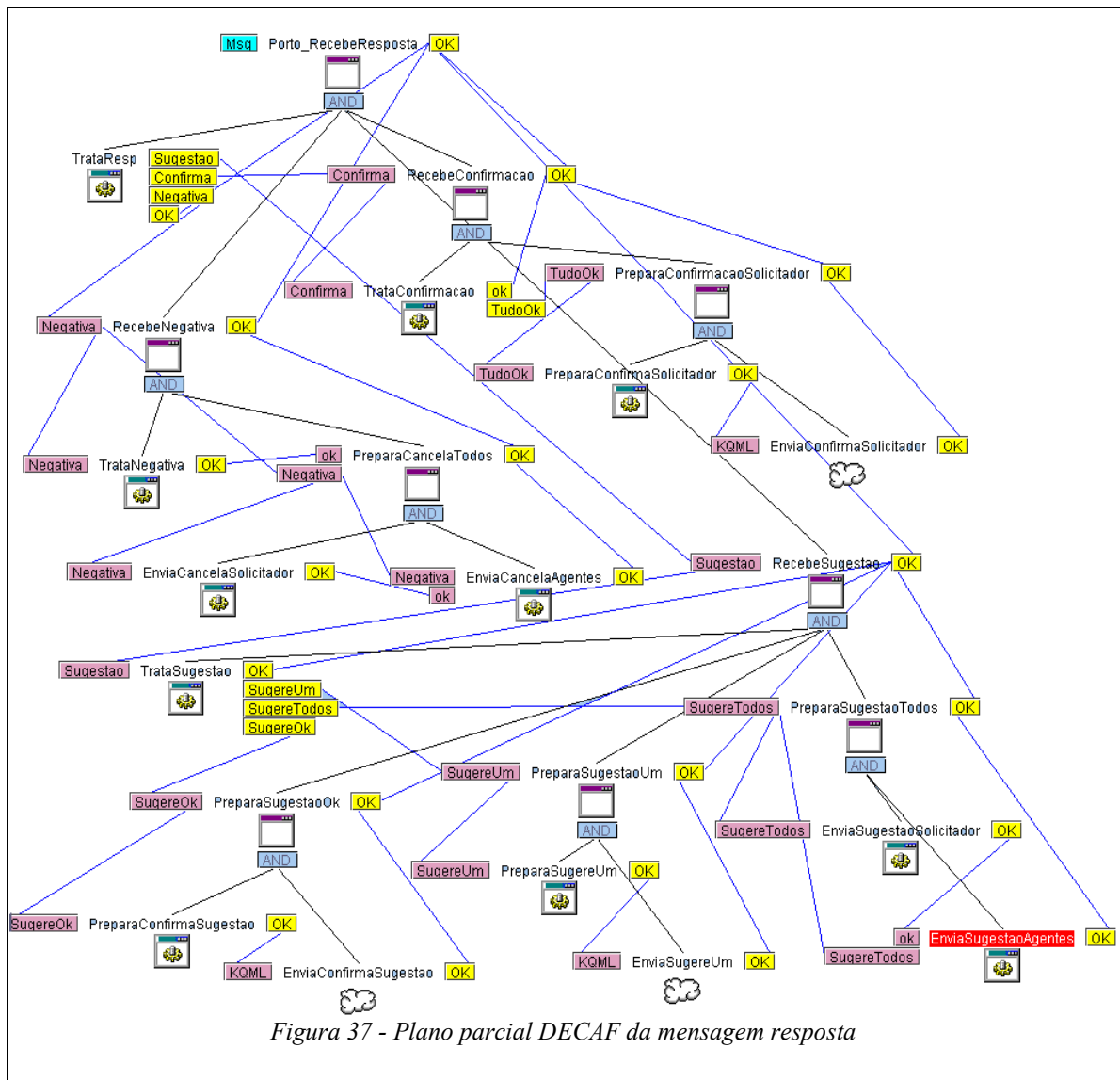
O nome da tarefa mais no topo das Figuras 36 e 37 é composto de dois segmentos, um a direita do traço de *underscore* e outra à esquerda. O segmento à esquerda é a ontologia que está sendo usada e o à direita é o nome da mensagem que está sendo recebida. A mensagem enviada de um agente para outro agente deve conter, então, a ontologia e o nome da mensagem. O nome completo para a mensagem no plano DECAF é então *Ontologia_NomeMensagem*.

Existem somente dois tipos de mensagem que não podem possuir ontologia no DECAF: Uma de inicialização, *_Startup* e uma de término do agente, *_Shutdown* (McGeary, 2001). Ambas são internas ao funcionamento do DECAF.

Foi desenvolvido um único modelo de agente denominado agente padrão e que é a base para todos os agentes do sistema. O plano DECAF do agente padrão trata somente de dois tipos de mensagens: 1) Solicitação, *Porto_RecebeSolicita* e 2) Resposta, *Porto_RecebeResposta*. Estas serão as únicas maneiras dos agentes atuarem no sistema multiagentes, ora enviando e recebendo solicitação e ora enviando e recebendo respostas às solicitações.

Cada tarefa no DECAF gera a implementação de uma classe Java e cada ação vinculada a ela através da linha no plano representa um método dentro desta classe.





Foi desenvolvido um conjunto de classes escritas em Java e encapsuladas em uma única interface para descrever a lógica do agendamento dos recursos do agente. Esta interface é a base para o processo de agendamento dos recursos de cada agente, Figura 38. Neste conjunto de classes estão as definições de como um agente faz efetivamente o agendamento de seus recursos. Dentre as várias classes deste conjunto, destacam-se duas classes: 1) *Pool* de solicitações recebidas e 2) *Pool* de recursos que o agente pode utilizar para executar as tarefas.

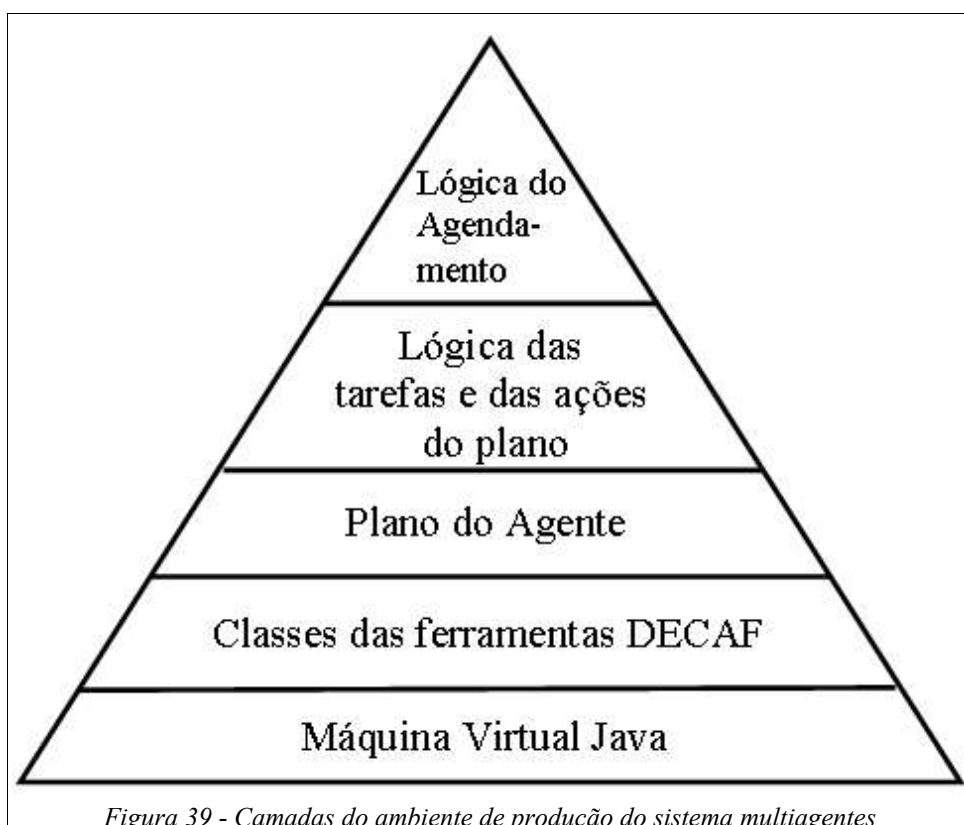
A classe *AgenteGeral*, Figura 38, é a única utilizada no desenvolvimento dos agentes e serve como interface e encapsulamento de todas as outras classes desenvolvidas. Portanto, cria-se em cada agente um único objeto da classe *AgenteGeral* que terá todos os métodos de todas as classes para executar o agendamento.

Vale ressaltar, que cada agente implementado por esta interface conhece única e exclusivamente o seu mundo, que pode ser definido como sendo: seus recursos e suas respectivas agendas, as solicitações recebidas e enviadas, as mensagens recebidas e enviadas, e o status da negociação.

O ambiente DECAF tem, também, a função de prover o ambiente computacional para executar o sistema proposto. O roteamento das mensagens é feito através do programa ANS (*Agent Name Service*) e do programa ANSQUERY (*Agent Name Service Query*), ambos são parte integrante da biblioteca DECAF.

O ambiente de produção dos agentes pode ser visualizado em cinco camadas, Figura 39:

1. Máquina virtual Java;
2. Classes das ferramentas DECAF;
3. Plano do Agente escrito no *PlanEditor*;
4. Classes que representam a lógica de agendamento das tarefas e das ações descritas no plano do agente;
5. Classes que representam a lógica do agendamento dos recursos do agente.



Como todas as classes DECAF são escritas em Java, faz-se necessário para executar o agente que a máquina virtual Java esteja rodando. Em cima da máquina virtual Java são executadas as classes DECAF. Estas classes carregam os planos do agente. Na medida que o agente recebe uma solicitação, o DECAF percorre o plano do agente para ver qual a tarefa ou ação ele tem que executar e, então, carrega a classe necessária. Por fim, ao executar as classes que representam o funcionamento das tarefas e ações, o agente deve fazer efetivamente o agendamento dos seus recursos e, para tanto, ele carrega as classes da lógica de agendamento.

Para o funcionamento do sistema multiagentes deve-se primeiramente executar duas classes Java denominadas: 1) ANS e 2) ANSQUERY. A classe ANS é um servidor de nomes aplicados aos agentes do sistema multiagentes e a classe ANSQUERY faz a representação visual da classe ANS. Todos os agentes que são inicializados no sistema multiagentes devem se cadastrar no servidor de nomes ANS no intuito de serem vistos pelo sistema multiagentes.

Uma vez que o servidor de nomes esteja operacional, pode-se inicializar os agentes. Para tanto, deve-se executar classe *Agent* que é a responsável por interpretar os planos dos agentes. A tela de inicialização do agente padrão é apresentada na Figura 40 e é a mesma da classe *Agent*.

A tela de inicialização do agente DECAF apresenta dados do servidor de mensagens, os dados do agente, como nome, qual a máquina que ele está sendo executado e o nome do plano do agente, Figura 40. O botão *Start Agent* carrega o plano do agente e o botão *Register* faz a identificação do agente ao servidor de nomes.

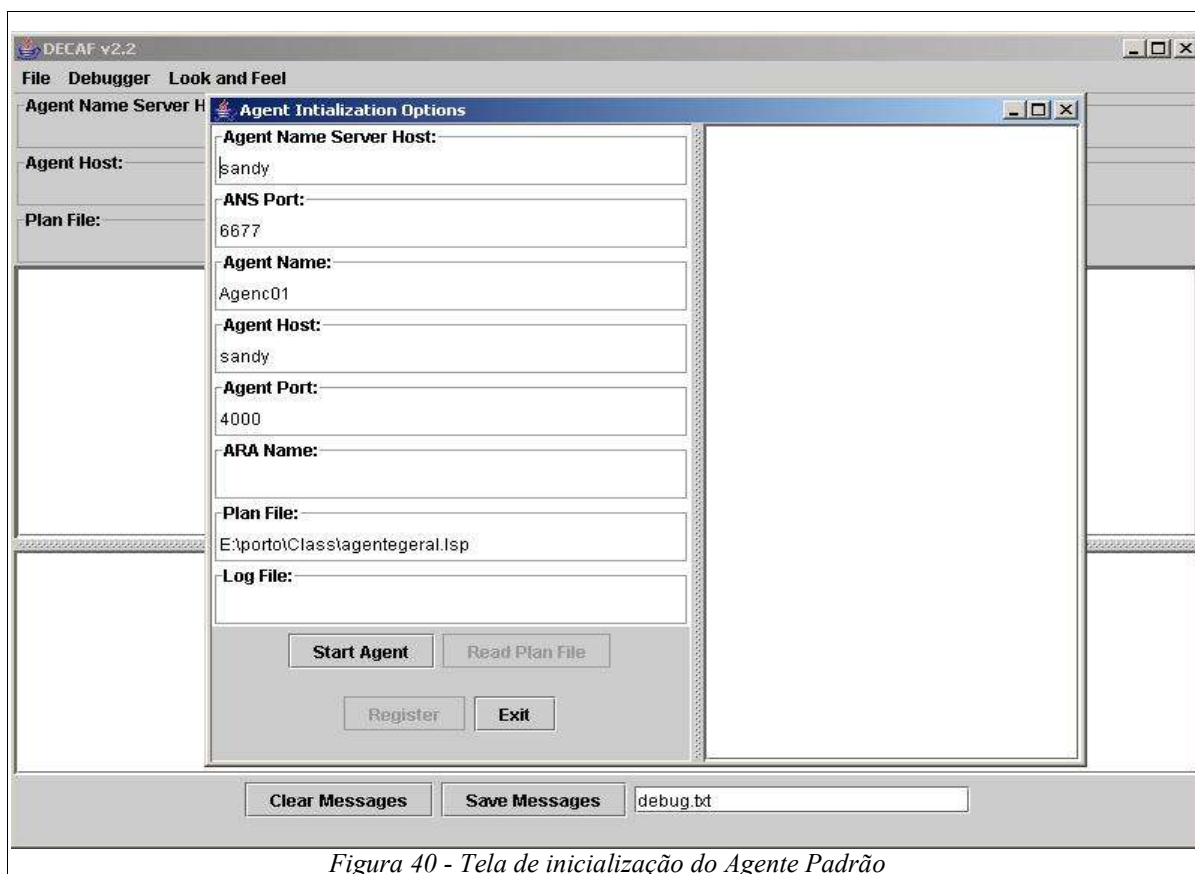


Figura 40 - Tela de inicialização do Agente Padrão

A Figura 41 apresenta uma tela do agente em execução. Na parte branca no canto superior/esquerdo são postadas as mensagens enviadas e no canto superior/direito são postadas as mensagens recebidas. Na parte inferior é mostrado o status do funcionamento do agente.

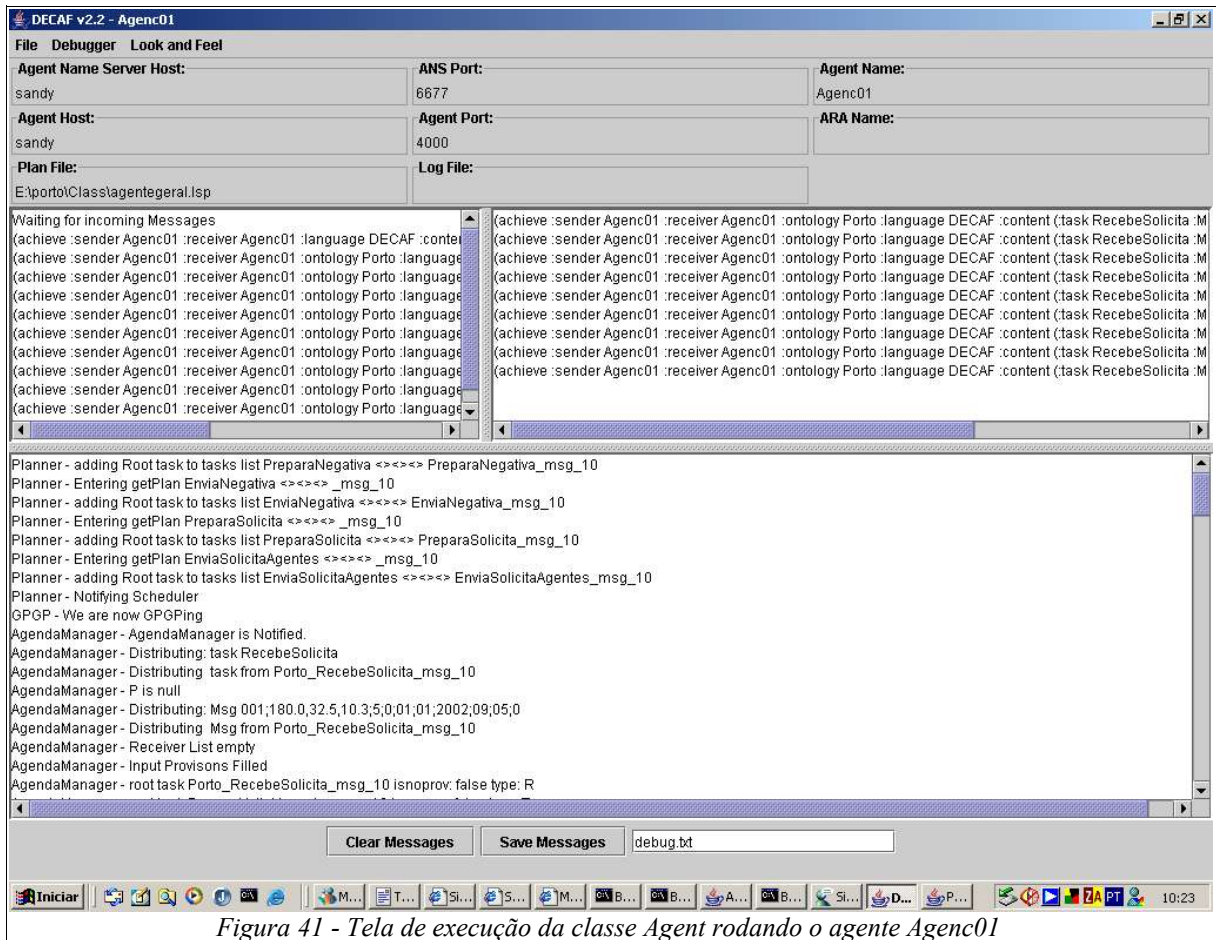


Figura 41 - Tela de execução da classe Agent rodando o agente Agenc01

O processo apresentado deve ser executado para cada um dos agentes do sistema multiagentes. Após todos os agentes terem sido inicializados, simula-se o envio de uma mensagem do mundo externo para o agente Agenc01 e este começa a enviar e receber mensagens dos outros agentes até que se chegue numa data de acordo para todas as solicitações que tenham sido feitas ao sistema multiagentes.