

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
COORDENAÇÃO DOS CURSOS DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA

Dissertação de Mestrado

**Modelagem de um Sistema Multiagente para
Aplicação em Ambientes Industriais**

José Pedrosa Barreto Neto

Campina Grande – PB
Março - 2007

Modelagem de um Sistema Multiagente para Aplicação em Ambientes Industriais

José Pedrosa Barreto Neto

Dissertação de mestrado submetida à Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande – Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciências no domínio da Engenharia Elétrica.

Área de Concentração
Processamento da Informação

Orientadores:
Maria de Fátima Queiroz Vieira Turnell, PhD
José Sérgio da Rocha Neto, D.Sc.

Campina Grande, Paraíba
Março de 2007



FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCCG

B273m Barreto Neto, José Pedrosa

Modelagem de um sistema multiagente para aplicação em ambientes industriais / José Pedrosa Barreto Neto . — Campina Grande, 2007.
71. : il.

Inclui Bibliografia.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Tecnologia e Recursos Naturais.

Orientadores: Maria de Fátima Q. V. Turnell, José Sérgio da R. Neto.

1— Inteligência Artificial-Sistemas Multiagentes 2— Sistemas Industriais-Supervisão 3— Sistemas de Apoio-Decisão I— Título

CDU 004.896

**MODELAGEM DE UM SISTEMA MULTIAGENTE PARA APLICAÇÃO EM
AMBIENTES INDUSTRIAIS**

JOSÉ PEDROSA BARRETO NETO

Dissertação Aprovada em 19.03.2007

Maria de Fatima Q. Vieira Turnell
MARIA DE FÁTIMA QUEIROZ VIEIRA TURNELL, Ph.D., UFCG
Orientadora

José
JOSÉ SÉRGIO DA ROCHA NETO, D.Sc., UFCG
Orientador

Benemar Alencar
BENEMAR ALENCAR DE SOUZA, D.Sc., UFCG
Componente da Banca

Angelo Perkusich
ANGELO PERKUSICH, D.Sc., UFCG
Componente da Banca

CAMPINA GRANDE - PB
MARÇO - 2007

Resumo

Neste trabalho é detalhada uma proposta de estratégia descentralizada para auxiliar operadores de sistemas industriais no processo de tomada de decisão, após a ocorrência de contingências no sistema. Essa estratégia é baseada na abordagem multiagentes. Nela os agentes são responsáveis por avaliar o estado do sistema e executar um conjunto de ações corretivas a partir de negociações fundamentadas em um conjunto de regras

O processo de modelagem do sistema multiagentes que implementa a estratégia é apresentado no transcorrer do texto. O contexto de aplicação escolhido para exemplificar a aplicabilidade da estratégia foi o de sistemas elétricos de potência, mais especificamente a subtransmissão.

O trabalho está relacionado à investigação da adequação de uma estratégia descentralizada à supervisão de processos industriais, onde tradicionalmente se adota a estratégia centralizada. Como resultado da investigação é proposta a arquitetura de uma ferramenta computacional baseada em agentes que viabilize a aplicação desta estratégia em ambientes industriais.

Palavras chave: sistemas multiagentes, supervisão de sistemas industriais, sistemas de apoio à decisão.

Abstract

In this work a proposal of a decentralized strategy to aid industrial systems operators in the decision making process after the occurrence of contingencies in the system is detailed. This strategy is based on the multiagent approach. Those agents are responsible for evaluating the state of the system and perform a set of corrective actions based on a set of rules and negotiations between themselves.

The modeling process of the multiagent system that implements the strategy is presented in the text. The application context chosen to exemplify the applicability of the strategy was that of an electrical power system, more specifically the subtransmission part.

The work is related to the investigation of a decentralized strategy adequacy to the supervision of industrial processes, where traditionally it is employed a centralized approach. As result of this work it is proposed the architecture of a multiagent-based computational tool that supports the strategy's application in industrial environments.

Key words: multiagent systems, industrial systems supervision, decision support systems

Agradecimentos

Ao CNPq por ter disponibilizado uma bolsa que possibilitou a realização dos meus estudos.

À Prof^a Maria de Fátima Q. V. Turnell pela amizade, pelos ensinamentos teóricos e de vida, e pelo apoio concedido.

À comunidade acadêmica do LSIS, representados pelo Prof. Bernard Espinasse, pelos ensinamentos e atenção dispensados durante minha estadia em Marseille.

Aos companheiros do LIHM por todas as conversas, incentivos e auxílios em todos os momentos do desenvolvimento deste trabalho.

Aos colegas e professores da UFCG que estiveram sempre presentes para incentivar e ajudar na superação de dificuldades que surgiram no transcorrer deste trabalho.

Aos meus amigos que estiverem ao meu lado e me dando forças para vencer todos os obstáculos durante os últimos dois anos.

Aos meus pais, Maria do Perpétuo Socorro Leite Barreto e José Pedrosa Barreto Filho, pelo amor, apoio, compreensão e, mais do que tudo, pela formação que me deram. Às minhas irmãs Cristiane, Larissa, Carol e Gabriela por me apoiarem sempre.

Especialmente à minha esposa, Monique Ribeiro de Araújo, pelo amor e dedicação. Por suportar minhas variações de humor sempre com muita paciência e compreensão nos momentos difíceis destes dois anos que estamos casados, e por estar ao meu lado apoiando todas as decisões que tomo.

Sumário

1 - Introdução	1
1.1 – Motivação	1
1.2 – Objetivos.....	5
1.3 – Estrutura do Trabalho	6
2 - Revisão Bibliográfica.....	8
2.1 – Inteligência Artificial Distribuída (IAD).....	8
2.1.1 – Sistemas Multiagentes	10
2.2 – Sistemas de Apoio à Decisão (SAD).....	16
2.2.1 – SAD’s Aplicados a Sistemas Elétricos.....	20
3 - Especificação da Estratégia Descentralizada para Tomada de Decisões	23
3.1 – Aplicação da Estratégia a um Sistema Elétrico	27
3.1.1 – Fase 1 – Resolução Interna.....	34
3.1.2 – Fase 2 – Propagação em Mesmo Nível	36
3.1.3 – Fase 3 – Propagação para Nível Superior (I).....	39
3.1.4 – Fase 4 – Propagação para Nível Superior (II)	41
3.1.5 – Processo de Rejeição de Carga.....	44
4 - Modelagem do Sistema Multiagentes	47
4.1 – A Plataforma MAJORCA.....	53
4.2 – A Plataforma JADE	55
4.3 - Discussão sobre a Implementação.....	59
5 - Considerações Finais.....	61
5.1 – Trabalhos Futuros	63
6 - Referências Bibliográficas	64

Índice de Figuras

Figura 2.1 - Representação de um agente e sua interação com o ambiente em que está inserido [Russel & Norvig, 2002]	11
Figura 2.2 - Quadro Ilustrativo das Diferenças entre Agentes Cognitivos e Reativos [Demazeau, 1994]	12
Figura 2.3 - Estrutura da Mensagem KQML Padrão [Weiss, 1999].....	14
Figura 2.4 - Estrutura da Mensagem FIPA-ACL Padrão	15
Figura 3.1 – Diagrama de Blocos Representando um Sistema Dividido em Níveis Hierárquicos	25
Figura 3.2 - Arquitetura para Implementação da Estratégia	26
Figura 3.3 - Estrutura da Rede Considerada no Estudo de Caso.....	28
Figura 3.4 - Representação do Sistema Contendo Uma Perturbação.....	29
Figura 3.5 - Fluxo de Execução do Processo	31
Figura 3.6 - Diagrama de Seqüência de Mensagens para os Agentes Envolvidos na Fase 1.....	35
Figura 3.7 - Grafo RCA para elemento SS3_1 na Fase 1.....	36
Figura 3.8 - Grafo RCA para o elemento SS2_2 na Fase 1	36
Figura 3.9 - Grafo RCA para o elemento LT na Fase 1	36
Figura 3.10 - Diagrama de Seqüência de Mensagens para os Agentes Envolvidos na Fase 2.....	38
Figura 3.11 - Grafo RCA para o elemento SS3_1 na Fase 2	38
Figura 3.12 - Grafo RCA para o elemento SS3_2 na Fase 2	38
Figura 3.13 - Grafo RCA para o elemento LT nas Fases 2, 3 e 4	39
Figura 3.14 - Diagrama de Seqüência de Mensagens para os Agentes Envolvidos na Fase 3.....	40
Figura 3.15 - Grafo RCA para o elemento SS3_1 na Fase 3	41
Figura 3.16 - Grafo RCA para o elemento SS2_2 na Fase 3	41
Figura 3.17 - Grafo RCA para o elemento SS2_2 na Fase 4	43
Figura 3.18 - Grafo RCA para o elemento SEx na Fase 4	43

Figura 3.19 - Diagrama de Seqüência de Mensagens para os Agentes Envolvidos na Fase 4.....	44
Figura 3.20 - Diagrama de Seqüência de Mensagens para os Agentes Envolvidos no Processo de Rejeição de Carga.....	45
Figura 3.21 - Grafo RCA para o elemento SS3_1 no processo de rejeição de carga.....	46
Figura 3.22 - Grafo RCA para o elemento SD1 no processo de rejeição de carga.....	46
Figura 4.1 - Convenções Gráficas do Formalismo RCA	49
Figura 4.2 - Metamodelo Representando as Relações entre Estados e Transições do RCA [Tranvouez, 2001].....	50
Figura 4.3 - Arquitetura dos Agentes Cognitivos	52
Figura 4.4 - Modelo Padrão de Plataforma de Agentes Definido pela FIPA [Silva, 2003]	56
Figura 4.5 - Plataforma de Agentes JADE Distribuída em Vários Containeres	57
Figura 4.6 - Síntese do Processo de Desenvolvimento	58

Índice de Quadros

Quadro 1 - Representação dos conhecimentos de cada tipo de agente	33
Quadro 2 - Pseudocódigo Referente à Dinâmica da Fase 1	34
Quadro 3 - Pseudocódigo Referente à Dinâmica da Fase 2	37
Quadro 4 - Pseudocódigo Referente à Dinâmica da Fase 3	40
Quadro 5 - Pseudocódigo Referente à Dinâmica da Fase 4	43
Quadro 6 - Pseudocódigo Referente à Dinâmica do Processo de Rejeição de Carga	45
Quadro 7 - Trecho de Código Gerado pela Ferramenta XML2Jess	55

Capítulo 1

Introdução

1.1 – Motivação

O setor industrial ocupa uma posição de destaque na sociedade devido ao constante aumento de produção e à diversificação de seus processos. Algumas atividades deste setor impactam diretamente no bem-estar da sociedade, de forma que a operação de sistemas relacionados a estas atividades é considerada crítica. Neste trabalho, sistemas são considerados críticos se a ocorrência de uma falha operacional pode ocasionar perdas consideráveis de recursos financeiros, materiais e /ou humanos.

A operação dos sistemas industriais envolve um processo contínuo de supervisão e controle tanto local quanto global. Este controle é realizado pelos operadores do sistema geralmente apoiados por *software* do tipo SCADA (Supervisory Control and Data Acquisition systems). Na execução de suas atividades, o operador tem acesso a diversas informações sobre o sistema, e ao fazê-lo se submete a uma sobrecarga de informações que pode causar desconforto e levá-lo a cometer erros.

A situação se torna mais complexa por conta do caráter centralizado da operação que força a propagação do problema até uma unidade centralizadora. Esta propagação, além de implicar no aumento do tempo de solução dos problemas, pode resultar em uma decisão não-otimizada devido ao foco global dado pela unidade centralizadora não levar em consideração especificidades do problema, agravando as conseqüências de um erro. Esse aumento no tempo de implementação de uma solução muitas vezes é decorrente da necessidade de comunicação entre os envolvidos, a qual poderia ser minimizada uma

vez que, em alguns sistemas industriais, soluções locais ótimas podem levar a uma solução global ótima [Talukdar, 2002]. Desta forma, um processo de gestão descentralizado com base em uma estratégia de resolução local parece bastante apropriado para lidar com supervisão e controle de sistemas críticos.

Este processo de gestão pode ser auxiliado pelos chamados sistemas de apoio à decisão (SAD). Sistemas deste tipo são encontrados em aplicações diversas e em etapas que vão desde o planejamento de atividades até operação em tempo real. A importância destes SAD's é particularmente grande quando o operador lida com uma situação em que o sistema sofre uma falta e precisa ser restabelecido o quanto antes, devido aos impactos desta interrupção da operação dos sistemas.

A interrupção em sistemas de produção produz impactos, principalmente, de ordem financeira para as empresas que terá uma diminuição da produção total e conseqüente diminuição das vendas possíveis. Do ponto de vista do consumidor, caso o produto seja de primeira necessidade, ele será privado do acesso a um item indispensável ao seu bem-estar. Alguns exemplos de sistemas que fornecem itens de primeira necessidade são os sistemas de provisão de água tratada e energia elétrica.

Qualquer interrupção no fornecimento destes itens afeta de forma negativa tanto os consumidores quanto as empresas que provém o serviço. No caso da energia elétrica, por exemplo, o impacto óbvio para os consumidores é a privação da possibilidade de utilização de equipamentos e dispositivos elétricos para os mais diversos fins, enquanto que as empresas fornecedoras estão sujeitas a sanções econômicas proporcionais ao efeito negativo causado aos consumidores.

Por estas razões, esforços são feitos no sentido de minimizar as interrupções do fornecimento de energia elétrica e que se vierem a ocorrer sejam interrupções de curta duração, uma vez que a satisfação do usuário tem relação direta com a frequência e a duração da interrupção do serviço [Fukuyama, Endo & Nakanishi, 1996].

Hoje em dia, a maior parte dos sistemas já é automatizada e opera com o auxílio de *software* tipo SCADA. A adoção deste tipo de *software* para a operação dos sistemas industriais representou um grande avanço, do ponto de vista de supervisão dos sistemas,

para as empresas, pois ele permite que os operadores tenham acesso às mais diversas informações sobre o sistema, tais como valores de grandezas, propriedades dos equipamentos, dados sobre interligação de subsistemas, alarmes gerados devido a quaisquer anormalidades no sistema, entre outras.

Entretanto, todas essas informações não constituem elementos suficientes para que seja possível impedir a ocorrência de falhas operacionais que resultam em faltas no sistema, nem assegura que estas sejam resolvidas rapidamente. Por esta razão, a utilização de SAD's associados a sistemas supervisórios, com o intuito de garantir que a operação do sistema industrial seja segura e possua apenas interrupções de duração mínima, aparece como uma opção viável e passível de apresentar bons resultados do ponto de vista prático.

Especificamente na operação de sistemas elétricos, que é o contexto de aplicação utilizado no decorrer desta dissertação, a automação está presente com o intuito de garantir a operação maximamente ininterrupta e minimizar o tempo necessário para a correção de problemas. Entretanto, no caso de dispositivos de religamento automático não resolverem o problema de interrupção do fornecimento, as etapas de diagnóstico e correção do problema são feitas pelos operadores de subestações e centros com base em regras definidas pelas empresas do setor elétrico.

Normalmente, o processo de restabelecimento do fornecimento de energia a uma determinada área afetada tem o tempo como um recurso escasso (o que impede, por exemplo, medidas de aumento da geração, pois isso demanda um tempo significativo) e dispendioso, e muitas vezes o operador tem que lidar com situações em que as informações são incompletas e/ou não-confiáveis.

O problema é que a tarefa de restauração do fornecimento de energia tem múltiplos objetivos os quais podem ser contraditórios [Ćurčić et al, 1996], além disso, este é um problema combinatório, não-linear e restritivo [Nagata et al, 2000]. O restabelecimento deve ser feito satisfazendo objetivos como garantir que o mínimo de clientes fique sem o fornecimento, o mínimo de operações de chaveamento sejam feitas visto que elas têm um custo associado do ponto de vista da qualidade da energia fornecida e de manutenção das chaves, que a operação seja executada o mais

rapidamente possível, e que nenhum elemento do sistema opere fora dos limites operacionais estabelecidos (acima da máxima sobrecarga tolerável).

Por ser um problema de otimização onde é necessário atender múltiplos objetivos, torna-se explícita a necessidade de elaboração e desenvolvimento de estratégias e ferramentas computacionais para auxiliar a tomada de decisão dos operadores de subestação e de centros de controle na tarefa de restauração do fornecimento de energia elétrica às áreas afetadas pela ocorrência de uma contingência.

Estas estratégias e ferramentas computacionais são exatamente os sistemas de apoio à decisão (SAD), mencionados anteriormente. Ultimamente, pode-se perceber na literatura científica uma tendência na utilização de técnicas e conceitos de inteligência artificial (IA), sejam sistemas especialistas, redes neurais, algoritmos genéticos, agentes de software ou qualquer outra vertente, no desenvolvimento destas ferramentas. São encontradas aplicações de SAD's em atividades que vão desde o diagnóstico médico [Ramnarayan & Britto, 2002] até o gerenciamento de contingências em complexos petrolíferos [Sheremetov, Contreras & Valencia, 2004], passando por atividades logísticas [Koutsoukis et al, 2000].

Especificamente em sistemas elétricos de potência, o uso de SAD's baseados em IA visam, principalmente, auxiliar o processo de restauração do sistema. Entretanto, existem aplicações que abrangem desde processamento de alarmes, diagnósticos de faltas, determinação de esquemas de proteção, escalonamento de atividades até planejamento para expansão de sistemas.

No que se refere ao processo de restauração do sistema, existem dois enfoques distintos. O primeiro em sistemas de distribuição que faz a suposição de que o sistema considerado tem a radialidade como uma de suas características intrínsecas, facilitando a modelagem dinâmica do mesmo. E o segundo em sistemas de transmissão, que descartam a consideração de que o sistema é radial, uma vez que, por questões operacionais, é necessário garantir que existam caminhos alternativos para que a energia elétrica atinja os pontos finais (empresas de distribuição de energia elétrica), tornando mais complexa a modelagem dinâmica devido à existência de malhas.

Neste trabalho, sugerimos a adoção de uma estratégia descentralizada para auxílio à tomada de decisão que servirá como base para a implementação de uma ferramenta computacional. Essa ferramenta será fundamentada na abordagem de agentes de software, para auxílio à tomada de decisões diante de contingências. A estratégia aqui proposta é um refinamento daquela proposta originalmente em [Aventini, 2004], que será descrita em detalhes posteriormente, e tem como princípio um método de restauração com base em uma “solução de reparo” (*repair solution*) já utilizado no re-escalonamento de sistemas de manufatura [Tranvouez, 2001]. Tal solução, por sua vez, é baseada nas interações entre diferentes entidades a fim de minimizar os efeitos de perturbações no sistema.

A partir do refinamento da estratégia será proposta a uma arquitetura para a ferramenta de apoio à decisão, a qual consiste em duas camadas. A camada inferior é um modelo que representa a parte operacional do sistema, e a camada superior contém os agentes de software associados ao modelo operacional do sistema.

Estas duas camadas se comunicam através de trocas de mensagens a fim de que o modelo operacional seja atualizado com as decisões tomadas pelos agentes. O objetivo é que os agentes tenham sempre os dados mais recentes do sistema para “raciocinar” sobre quais decisões devem ser tomadas. Sempre que houver alguma alteração na condição operacional do sistema, os valores serão atualizados. Importante destacar que a proposta da ferramenta não é substituir o elemento humano no processo e, sim, fornecer-lhe subsídios para uma melhor tomada de decisão.

1.2 – Objetivos

Nosso objetivo principal com este trabalho é aplicar a estratégia descentralizada a um contexto de operação de um sistema crítico.

Como objetivos específicos:

- Refinar a estratégia proposta em [Aventini, 2004; Turnell, Espinasse & Aventini, 2005] para um contexto industrial caracterizado por uma estrutura de decisão centralizada;
- Especificar uma arquitetura de agentes que possibilite a descentralização do processo de tomada de decisões;
- Determinar formas de implementação para uma solução computacional com base na estratégia proposta e em conceitos de sistemas multiagentes;

1.3 – Estrutura do Trabalho

O texto desta dissertação está estruturado como é descrito a seguir.

No Capítulo 2 é feita uma breve revisão bibliográfica com relação ao uso de sistemas de apoio à decisão (SAD) em sistemas industriais. Além disso, como são utilizados conceitos de inteligência artificial (IA), mais especificamente de sistemas multiagentes, na abordagem proposta neste trabalho, alguns conceitos e aspectos relacionados com esta técnica são discutidos.

O Capítulo 3 é destinado à apresentação da estratégia proposta, para lidar com o problema de perturbações em tempo de operação, bem como é apresentada a aplicação da estratégia a um cenário de falta em um sistema de transmissão. A estratégia descentralizada é baseada em *repair solution* e tem como objetivo resolver o problema da forma mais local possível, evitando sua propagação pelo sistema.

A seguir, no Capítulo 4, são apresentadas características das plataformas, a arquitetura dos agentes cognitivos concebida para implementação dessa estratégia, e as ferramentas utilizadas no desenvolvimento da camada de agentes da ferramenta a ser desenvolvida. Além disso, são discutidos aspectos relacionados com a implementação da estratégia proposta tais como a escolha de ferramentas de modelagem e formas de integração entre as camadas da ferramenta.

Por fim, as considerações sobre limitações e possibilidades de trabalhos futuros são apresentadas no Capítulo 5.

Capítulo 2

Revisão Bibliográfica

Este capítulo é dedicado à apresentação da revisão bibliográfica pertinente ao escopo do trabalho. Uma vez que este trabalho engloba dois campos de pesquisa distintos, faz-se necessário o desenvolvimento de uma revisão bibliográfica tanto do uso de técnicas de inteligência artificial distribuída, mais especificamente sistemas multiagentes, quanto da utilização de sistemas de apoio à decisão (SAD) em sistemas industriais.

2.1 – Inteligência Artificial Distribuída (IAD)

Inteligência Artificial Distribuída (IAD) é uma subárea da Inteligência Artificial (IA), que surgiu na década de 80 devido a uma necessidade de certos tipos de problemas que possuem solução inerentemente distribuída seja funcional ou geograficamente [Gasser, 1988]. Esta subárea estuda o conhecimento e os métodos de raciocínio que podem ser necessários ou úteis para que agentes computacionais participem de sociedades de agentes.

Diferentemente da IA clássica que tem seu foco no comportamento humano individual, a IAD tem foco no estudo das entidades computacionais, seja no aspecto de suas capacidades internas, seja no aspecto de sua sociabilidade incluindo cooperação, levando à realização conjunta de algum objetivo global. Ou seja, os sistemas computacionais são vistos como sociedades de agentes inteligentes.

Essa abordagem coloca os sistemas muito mais próximos da realidade de sociedades humanas, onde a solução de problemas muitas vezes é resultado da cooperação de vários indivíduos. É esperado que estes sistemas apresentem algumas características interessantes, tais como adaptabilidade, diminuição dos custos de desenvolvimento e manutenção, aumento da eficiência e da velocidade, distribuição de problemas de forma mais natural, entre outras.

Esta subárea caracteriza-se por ter dois enfoques diferentes, que são a Solução Distribuída de Problemas (SDP) e os Sistemas Multiagentes (SMA). O primeiro tem foco no problema e estuda técnicas para resolver problemas específicos dividindo o trabalho entre muitos módulos que cooperam interagindo e trocando conhecimentos sobre o problema e a solução. Este enfoque aplica agentes pré-programados para cooperar, com métodos que visam garantir uma cooperação coerente, robusta e eficiente. A qualidade de um sistema de SDP é medida pelo desempenho global do sistema na solução de um problema específico.

O segundo enfoque, que foi utilizado no desenvolvimento deste trabalho, tem foco no agente e estuda o comportamento inteligente em uma sociedade de agentes autônomos, isto é, como coordenar seus conhecimentos, metas e planos para resolver problemas. Em outras palavras, estuda as pressuposições básicas sobre agentes que garantam a possibilidade de ação cooperativa em sociedade [Bittencourt, 2005].

A relação entre SDP e SMA não é clara e modifica-se continuamente. Na comunidade científica existem três visões distintas, que são [Bittencourt, 2005], [Durfee & Rosenschein, 1994]:

- SDP é um subconjunto de SMA: sistemas para SDP são SMA em que se verificam certas hipóteses sobre o comportamento dos agentes, como benevolência, compartilhamento de objetivos globais, existência de um projetista centralizador que pode introduzir os aspectos mencionados nos agentes sob sua responsabilidade;
 - SMA fornecem a base para SDP: assume-se que em SDP os agentes comportam-se cooperativamente e em SMA, como isso não é tão
-

simples, buscam-se meios para que isto possa realmente realizado. Então, um SMA estuda como os agentes irão agir cooperativamente e um SDP, usando essa cooperação, estuda como solucionar um problema externo.

- SMA e SDP são linhas de pesquisa complementares: a diferença entre os dois enfoques deve-se aos interesses teóricos subjacentes aos sistemas desenvolvidos. Pesquisadores de SDP preocupam-se com questões como eficiência e robustez da comunicação, enquanto pesquisadores de SMA preocupam-se com a possibilidade de cooperação.

Para o nosso trabalho adotamos a primeira visão, uma vez que um sistema multiagentes para apoiar a resolução de problemas por parte de operadores industriais é modelado. Desta forma, a subseção seguinte é destinada à apresentação de aspectos teóricos destes sistemas.

2.1.1 – Sistemas Multiagentes

A abordagem de sistemas multiagentes é utilizada no desenvolvimento de sistemas complexos, devido à possibilidade de divisão do problema em subproblemas de mais simples resolução. Segundo [Jennings & Wooldridge, 1999], essa abordagem pode reduzir o custo e a complexidade associados ao desenvolvimento de sistemas distribuídos. De forma simples, pode-se definir um sistema multiagentes como um conjunto de agentes que interagem em um ambiente comum, mas surge a questão do que é exatamente um agente.

Estabelecer uma definição de agente amplamente aceita ainda é um desafio para a comunidade científica de inteligência artificial. Entretanto, existem algumas definições que possuem uma aceitação parcial e são interessantes. Por exemplo, em [Weiss, 1999] encontramos que um agente é um sistema computacional que está situado

em algum ambiente, e que é capaz de atuar de forma autônoma neste ambiente a fim de alcançar seus objetivos de projeto.

A autonomia referida na definição encontrada em [Weiss, 1999] é, segundo [Wooldridge & Jennings, 1995a], é a capacidade que um agente tem de operar sem a intervenção direta de humanos ou outros, e de possuir algum tipo de controle sobre suas ações e estados internos.

Em [Russel & Norvig, 2002], um agente é definido como qualquer entidade que pode perceber um ambiente através de sensores e agir através de atuadores. Tal definição pode ser representada pela Figura 2.1.

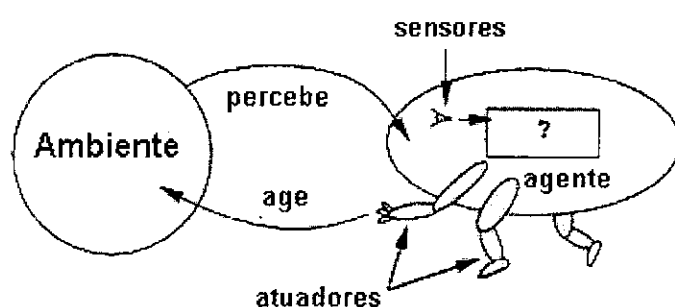


Figura 2.1 - Representação de um agente e sua interação com o ambiente em que está inserido [Russel & Norvig, 2002]

Em [Nareyek, 2000], um agente é considerado uma entidade que tem objetivos, que pode sentir certas propriedades do ambiente e que pode executar ações específicas. Em [Franklin & Graesser, 1996], é dito que agentes autônomos estão situados em um ambiente, no qual eles sentem e agem, sobre o tempo, mantendo sua própria agenda e sentem os efeitos de suas ações no futuro.

Uma outra abordagem é a de encontrada em [Wooldridge & Jennings, 1995b], onde é proposta a distinção de duas noções gerais para agentes. A primeira delas chamada de noção fraca é utilizada para denotar qualquer *hardware* ou sistema computacional que apresente as propriedades de autonomia, habilidade social, reatividade e pró-atividade. Enquanto a segunda noção, chamada de noção forte, que defende que, além das características mencionadas anteriormente, o agente deve ser definido utilizando conceitos que usualmente são aplicáveis a seres humanos, tais como mobilidade, veracidade, benevolência e racionalidade.

Quanto aos tipos de agentes que podem existir em um sistema multiagentes, é possível definir que, basicamente, eles podem ser reativos ou cognitivos. Os agentes reativos têm seu comportamento baseado no par estímulo-resposta e, em geral, não apresentam memória, não planejam sua ação futura e não se comunicam com outros agentes. Por sua vez, os agentes cognitivos possuem uma representação explícita do ambiente e de outros agentes, podem planejar suas ações futuras devido à presença de memória, e podem se comunicar com outros agentes. As diferenças básicas entre esses dois tipos podem ser sumarizadas no quadro apresentado na Figura 2.2.

<i>Agentes Cognitivos</i>	<i>Agentes Reativos</i>
• representação explícita	• representação implícita
• têm histórico	• não têm histórico
• comunicação direta	• comunicação indireta
• controle deliberativo	• controle não deliberativo
• organização social	• organização etológica
• poucos agentes	• muitos agentes

Figura 2.2 - Quadro Ilustrativo das Diferenças entre Agentes Cognitivos e Reativos [Demazeau, 1994]

Esses agentes utilizam seu conhecimento para realizar tomadas de decisão e, assim, gerar os SMA deliberativos. Quando se pretende unir agentes deste tipo, a presença dos aspectos a seguir é muito relevante [Siebra, 2000]:

- **Organização:** refere-se à organização das atividades que são necessárias para o processo de solução do problema;
- **Cooperação:** ocorre quando há o envolvimento de mais de um agente na resolução de uma dada tarefa;
- **Negociação:** acontece quando algum agente tenta obter auxílio para a realização de seus objetivos através do oferecimento de alguma vantagem;
- **Comunicação:** refere-se à forma de comunicação, qual o protocolo será utilizado para troca de informações entre os agentes;

Este último aspecto, especificamente, é de suma importância para o funcionamento do sistema. Diversas estratégias de comunicação são possíveis para

efetuar a troca de mensagens entre os agentes. Algumas delas são comunicação direta, comunicação assistida, difusão de mensagens (*broadcast*) e através de quadro negro (*blackboard*).

Na comunicação direta, cada agente tem conhecimento da existência dos outros e da forma de endereçamento que deve ser utilizada para que a mensagem os atinja. A comunicação é feita sem intermediários. A principal vantagem é o fato de não existir um agente que coordena a comunicação, pois isso representa um gargalo na comunicação podendo até causar o bloqueio do sistema.

Na comunicação assistida, também conhecida como comunicação por sistemas federados, existe uma estrutura hierárquica de agentes e a troca de mensagens é feita por meio de agentes chamados mediadores ou facilitadores. Essa opção diminui consideravelmente o custo e a complexidade dos agentes individuais na realização da comunicação, mas insere uma centralização na mesma que pode causar um bloqueio do sistema.

A comunicação por difusão (*broadcast*) deve ser utilizada apenas em situações onde a mensagem deve ser enviada para todos os agentes ou quando o agente remetente não conhece o agente destinatário ou seu endereço. Neste tipo de comunicação, todos os agentes presentes no sistema recebem a mensagem.

A comunicação utilizando a estrutura de quadro negro (*blackboard*) é um modelo de estrutura com memória compartilhada que é bastante utilizado em projetos de IA. Ela consiste simplesmente de um repositório onde os agentes escrevem mensagens para outros agentes e obtêm informações sobre o ambiente.

Porém, tão importante quanto a definição da forma de comunicação entre os agentes é a definição de uma linguagem que seja compreendida por todos eles, uma vez que não há a garantia inicial de que todos os projetistas utilizem os mesmos termos para os mesmos conceitos. Em outras palavras, é necessário que os agentes inseridos em um mesmo ambiente compartilhem de uma ontologia comum. Neste sentido, os esforços mais relevantes foram na definição da *Knowledge Query and Manipulation Language* (KQML) e da *FIPA-Agent Communication Language* (FIPA-ACL).

KQML foi desenvolvida pelo consórcio *ARPA Knowledge Sharing Effort* com o intuito de ser uma linguagem e um protocolo de comunicação de alto nível para troca de mensagens independente de conteúdo e da ontologia aplicável. Segundo [Weiss, 1999], a elegância da KQML é que toda informação para compreensão do conteúdo da mensagem está incluída na própria comunicação.

Conceitualmente, a linguagem KQML é dividida em três níveis [Finin, Labrou & Mayfield, 1994], que são:

- Conteúdo: é o conteúdo da mensagem propriamente dita, na linguagem de representação própria dos agentes – KQML pode incluir qualquer linguagem de representação, sendo o conteúdo ignorado no que diz respeito à sua interpretação, que é função do receptor;
- Mensagem: cobre o conteúdo da mensagem que um agente quer transmitir a outro – este nível forma o cerne da KQML e determina os tipos de interação que os agentes podem ter;
- Comunicação: acrescenta alguns parâmetros à mensagem relacionada com a comunicação de baixo nível, como a identidade do remetente e do destinatário e um identificador único associado à comunicação;

Uma mensagem KQML consiste numa performativa e nos seus argumentos, incluindo o conteúdo da mensagem e um grupo de argumentos opcionais que o descrevem, como pode ser observado na Figura 2.3.

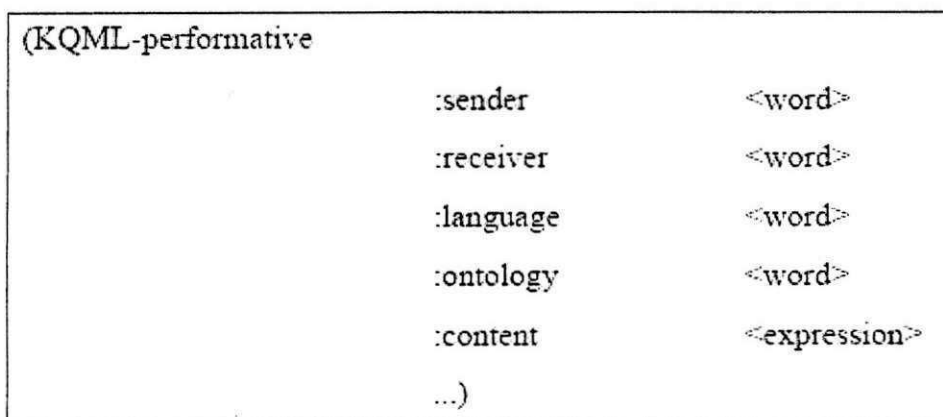


Figura 2.3 - Estrutura da Mensagem KQML Padrão [Weiss, 1999]

Apesar de a KQML ter representado um avanço, no sentido de obter uma padronização para a comunicação de agentes, alguns autores enumeram problemas relacionados à sua utilização. Algumas das reclamações mais constantes são que existe ambigüidade nos termos, que existem performativas com nomes inadequados e que existem atos comunicativos que não estão representados entre as performativas disponíveis.

Por sua vez, FIPA-ACL é uma linguagem que também é baseada em atos de fala. Ela possui sintaxe semelhante a da KQML, porém as performativas que são utilizadas são diferentes. Por sinal, as primitivas de comunicação são chamadas de atos comunicativos em vez de performativas. Sua especificação consiste em um conjunto de tipos de mensagens e descrições dos efeitos da mensagem sobre os agentes que a enviam e recebem. A estrutura da mensagem em FIPA-ACL é bastante similar à de uma mensagem KQML, como pode ser percebido na Figura 2.4, mas aspectos como gerenciamento de múltiplas soluções, por exemplo, são tratados de forma diferente pelas duas linguagens.

```
(communicative act
  :sender <valor>
  :receiver <valor>
  :content <valor>
  :language <valor>
  :ontology <valor>
  :conversation-id<valor>
  ...)
```

Figura 2.4 - Estrutura da Mensagem FIPA-ACL Padrão

A determinação de uma linguagem para servir como padrão na comunicação entre agentes foi apenas uma das iniciativas da *Foundation for Intelligent Physical Agents* (FIPA) que foi fundada ao perceber-se o grande potencial existente no uso da abordagem de sistemas multiagentes e ao verificar-se todos os problemas existentes em termos de padronização. Ela é uma organização sem fins lucrativos voltada para o estabelecimento de padrões para interoperabilidade de agentes heterogêneos e interativos e sistemas baseados em agentes. Sua missão é “A promoção de tecnologia e especificações de interoperabilidade que facilitem a comunicação entre sistemas de agentes inteligentes no contexto comercial e industrial moderno” [FIPA, 2003]. A FIPA

possui dois princípios básicos que são que o tempo necessário para se estabelecer uma padronização não deve ser longo de modo a impedir o progresso nessa área de pesquisa, e que apenas os comportamentos externos dos componentes é que podem ser especificados, deixando detalhes de implementação e arquiteturas internas para os desenvolvedores de agentes.

Com a adoção de padrões estabelecidos pela FIPA torna-se mais fácil o desenvolvimento e a integração de sistemas baseados em agentes por parte de diversos setores sendo que, principalmente, do setor industrial que elimina a preocupação de lidar com padrões proprietários estabelecidos por fabricantes de componentes e subsistemas. Hoje em dia, aplicações de sistemas baseados em agentes podem ser encontradas nas mais diversas áreas como, por exemplo, mercado financeiro, internet, robótica, sistemas de potência, jogos computacionais, problemas educacionais e de simulação [Monclair, 2001 apud Lopes Jr. et al, 2001].

Ainda é possível, mais especificamente, citar as principais aplicações para agentes ditos reativos e cognitivos separadamente. O uso de agentes reativos é associado a atividades como cartografia, simulação de partículas, simulação de veículos autônomos e manipuladores robóticos em geral, enquanto o uso de agentes cognitivos é mais freqüentemente associado a reconhecimento de linguagem natural, tutores inteligentes, visão computacional, sistemas de automação industrial e de telecomunicações, sistemas de apoio à decisão, entre outros.

2.2 – Sistemas de Apoio à Decisão (SAD)

Os sistemas de apoio à decisão (SAD) são sistemas computacionais que visam sistematizar e apoiar os processos decisórios em ambientes empresariais [Teixeira Jr., Fernandes & Pereira, 2006]. Eles são projetados para interagir ativamente com o usuário a fim de ajudá-lo a tomar melhores decisões com base nas informações obtidas [Sprague & Carlson, 1982 apud Adla & Zarate, 2006]. Estes sistemas trazem benefícios no

sentido de tornar a operação de sistemas produtivos mais eficiente e de minimizar os impactos de falhas que ocorram durante a operação do sistema.

Outro aspecto interessante relativo ao uso de SAD's é a possibilidade de reunir informações relativas à operação dos sistemas e, assim, possibilitar a manutenção do conhecimento especialista de uma organização [Serapião et al, 2003]. Tipicamente, os SAD's são compostos de pelos mesmos componentes básicos que são os módulos de interface com o usuário, o módulo de banco de dados e de modelos e o módulo decisório, que visam fornecer apoio em decisões semi-estruturadas (apenas partes do problema possuem uma resposta definida fornecida por uma metodologia aceita) e desestruturadas (não existe resposta fornecida por uma metodologia aceita para nenhuma parte do problema) a fim de aumentar a efetividade das decisões.

O processo de tomada de decisões envolve seis etapas distintas, que são executadas pelos SAD's. Elas são:

- Análise e identificação da situação;
- Desenvolvimento de alternativas;
- Comparação entre as alternativas;
- Classificação dos riscos de cada alternativa;
- Escolha da melhor alternativa;
- Sugestão ou execução e avaliação.

A utilização de conceitos e técnicas de inteligência artificial (IA) no desenvolvimento destas ferramentas de auxílio à tomada de decisão é uma constante. Dentre as técnicas mais aplicadas, podem ser citadas as redes neurais, a lógica *fuzzy*, os algoritmos genéticos e os sistemas multiagentes como os mais utilizados. Estas técnicas são utilizadas para avaliação de regras que determinam a forma de execução de um sistema especialista.

Exemplos de aplicação de SAD's podem ser encontrados em diversas áreas do conhecimento, que vão desde diagnósticos médicos [Ramnarayan & Britto, 2002] até sistemas de produção de petróleo [Guerra et al, 2003], dentre outras. Além da área de

aplicação, os SAD's também diferem quanto ao seu enfoque. Alguns deles são voltados para as etapas de planejamento da operação, enquanto outros são destinados à operação propriamente dita.

No contexto de planejamento da operação, alguns trabalhos podem ser mencionados. Em [Nicholson, Gunn & Nahavandi, 2005] é apresentada uma ferramenta de apoio à decisão para planejamento da produção através de uma melhor alocação de recursos. Esta ferramenta visa minimizar o número de operadores e a diferença entre os tempos de ciclo mínimo e máximo de cada grupo de máquinas de uma linha de produção, de modo a aproveitar melhor os recursos da produção.

Em [Dey, 2004] é proposto um SAD para inspeção e manutenção de dutos de petróleo. Ele visa minimizar o tempo gasto com a inspeção dos dutos, e planeja as manutenções com base em fatores de risco e suas probabilidades. A implementação deste sistema foi feita em um arcabouço de processo analítico hierárquico (AHP) e possibilitou uma redução de cerca de 60% nos custos provenientes de atividades de inspeção de dutos.

Um sistema de apoio à decisão para centros de controle de tráfego é apresentado em [Hegyi et al, 2001]. Este sistema combina medidas de controle com base em situações de tráfego apresentadas. A decisão sobre quais medidas devem ser tomadas é feita com base em interpolações *fuzzy* e visam reduzir o índice de congestionamentos não-recorrentes. O sistema também é dotado de um mecanismo de aprendizado adaptativo, de modo que seja aperfeiçoado com o passar do tempo.

Em [Teixeira Jr., Fernandes & Pereira, 2006] é apresentado um SAD para programação da produção em fundições de mercado. Este sistema é utiliza três metodologias para obtenção da solução, a saber, modelo de programação matemática, modelos de busca heurística e modelo com a abordagem de algoritmos genéticos. Os resultados apresentados são comparados e considerados satisfatórios, mas ainda devem ser validados em um sistema real representativo.

Outras diversas aplicações de SAD's ao planejamento de operações podem ser encontradas na literatura. Em [Wen, Wang & Wang, 2005], por exemplo, é apresentado

um sistema baseado em regras para o planejamento do orçamento destinado à defesa nacional da China. Em [Tsang, Yung & Li, 2004] é apresentado um SAD que visa, através de previsões com base no comportamento de mercado, auxiliar a escolha de melhores opções em operações com ações no mercado financeiro. Em [Ricci, 2002] é apresentado um sistema para recomendação de viagens com base em informações fornecidas pelo usuário, com relação a seus interesses.

Aplicações relacionadas ao planejamento de operações em sistemas elétricos também são frequentemente encontradas e serão listadas na subseção seguinte, juntamente com aplicações de SAD's à operação dos sistemas elétricos.

No que se refere à operação de sistemas, a utilização de SAD's também contempla diversas áreas. Em [Carvalho, Carvalho & Yovanovic, 2006] é apresentado um sistema inteligente para predição do comprimento de eletrodos em um forno elétrico a arco submerso, a fim de diminuir a frequência de abertura do forno e aumentar sua produtividade. Para o desenvolvimento deste trabalho foi empregada uma técnica com base em funções de *kernel*, chamada *Least Squares Support Vector Machine* (LS-SVM) e os resultados obtidos foram superiores a 70% de acerto.

Em [Koumboulis, Tzamtzi, Pavlovič, 2006] é proposta a utilização de um SAD baseado em agentes de software para automação de agronegócios. Os agentes são responsáveis por tarefas como monitoramento, detecção de faltas e logística, dentre outras. Os agentes são integrados através de OPC e utilizam ActiveX para se comunicar com sistemas de aquisição de dados e de controle supervisão.

Um sistema para gerenciamento de resíduos líquidos (água não-tratada) é proposto em [Ceccaroni, Cortés & Sánchez-Marré, 2004]. Este sistema é baseado em regras e utiliza ontologias para identificar situações de falta na planta de tratamento de água e, também, para prover soluções para estas situações. A provisão de soluções, com a integração de regras e ontologia, foi facilitada devido à possibilidade de fazer associações de causa e efeito.

Em [Farinelli et al, 2003] é proposto um SAD para operações de resgate. Este sistema é baseado em sistemas multiagentes e é dividido em dois aspectos que são o

desenvolvimento cognitivo, responsável pela fusão de informações, planejamento e coordenação, e a avaliação, responsável pela eficiência e robustez, mediante mudanças no ambiente. O sistema proposto ainda não foi validado em uma situação real.

Em [Sheremetov, Contreras & Valencia, 2004] é proposto um sistema híbrido para lidar com o gerenciamento de contingências em complexos petrolíferos existentes em zonas marítimas. Este sistema utiliza agentes de *software* que realizam inferências baseadas em lógica *fuzzy*. Na época do artigo, o trabalho estava em uma etapa de prototipagem.

2.2.1 – SAD's Aplicados a Sistemas Elétricos

Assim como em outras áreas, aplicações de SAD's a sistemas elétricos são encontradas tanto com ênfase no planejamento das operações quanto nas atividades de operação propriamente ditas.

No que se refere ao planejamento das operações, [Campitelli et al, 2004] propõe um sistema baseado em algoritmos genéticos para determinar alocação ótima e coordenação da proteção de alimentadores aéreos e radiais de distribuição de energia. A técnica utilizada no desenvolvimento do trabalho foi a de busca heurística. O foco do trabalho é nas faltas do tipo fase-terra, o que diminui um pouco sua generalidade.

Em [Xiao, Zhang & Wang, 2002] é apresentado um SAD para previsão de cargas elétricas utilizado no planejamento de sistemas de distribuição. O sistema utilizado é baseado em regras e em processo analítico hierárquico (AHP), mas seu desenvolvimento não se encontrava finalizado na época da publicação, de modo que resultados não foram apresentados.

Em [West et al, 2001] é apresentado um sistema baseado em regras para projeto e aplicação de esquemas de proteção para sistemas de transmissão de energia elétrica. A troca de informações dentro deste sistema é feita através da *web* para melhor disseminar

o conhecimento dentro da organização. A utilização deste SAD resultou em sugestões de modificações em esquemas anteriores.

Um sistema para determinação do preço da energia elétrica é apresentado em [Ribeiro et al, 2004]. Este sistema utiliza a lógica *fuzzy* para minimizar o risco na incerteza dos preços após a desregulamentação do setor elétrico, com o processo de privatizações e a implantação do livre mercado. Então, o objetivo aqui é a maximização dos lucros através da elaboração de planos de operação e investimentos que dêem poder de barganha às empresas em negociações de preços.

Com relação à operação, em [Mota, Mota & França, 2004] é proposta a utilização de lógica *fuzzy* e busca heurística para auxiliar o operador na recomposição do sistema após a ocorrência de um blecaute. Para se alcançar o resultado desejado são utilizadas apenas restrições operacionais de regime permanente e a solução é voltada para operação em tempo real.

Em [Canôas & Murari, 2006] é proposta a utilização de lógica *fuzzy* para criação de um sistema especialista para resolver o problema de controle de tensão e de potência reativa. A utilização deste sistema apresentou benefícios tais como melhoria no perfil de tensão, aumento na capacidade de transmissão disponível devido à diminuição do carregamento de equipamentos, aumento na segurança do sistema e melhoria da operação.

Outra aplicação de SAD's é apresentada em [Niu et al, 2005] que propõe um sistema baseado na utilização de mineração de dados associada a regras *fuzzy* para otimização da operação em uma planta de geração de energia. A mineração de dados é utilizada a fim de encontrar o ótimo com base em dados históricos. Alguns resultados experimentais satisfatórios foram obtidos.

Em [Duarte et al, 2004] é apresentada uma proposta de utilização de um sistema especialista para identificar e diagnosticar faltas, filtrando os alarmes e fornecendo ao operador apenas as informações relevantes. Além disso, em [Aoki et al, 2003] é proposto um sistema multiagentes para otimizar o deslocamento de viaturas no

atendimento de ocorrências na rede de distribuição de energia elétrica e, assim, minimizar o tempo gasto na restauração da normalidade do sistema.

Em [McArthur et al, 2005] é proposto um sistema baseado em agentes de *software* para monitoramento do sistema elétrico e detecção *online* de anomalias. O sistema foi implementado em JADE e aplicado a um estudo de caso de um gerador a diesel de 2,5 MW, apresentando bons resultados.

No caso de interrupção de serviço em um sistema elétrico, o processo de restauração passa por três etapas que são o restabelecimento das fontes de energia iniciais, integração de geração e transmissão, e minimização de cargas não alimentadas. Nossa proposta, instanciada para o sistema de transmissão, é uma estratégia de restauração cooperativa com foco na terceira fase, que é a minimização das cargas não atendidas. A estratégia é baseada na filosofia de *repair solution*, que já foi utilizada em outros contextos e será explicada com detalhes no capítulo posterior. A implementação desta estratégia é a base de um sistema de apoio à decisão para uso em ambientes industriais.

Capítulo 3

Especificação da Estratégia Descentralizada para Tomada de Decisões

Neste capítulo é apresentada uma estratégia, baseada na filosofia de *repair solution* [Zweben et al, 1993 apud Tranvouez, 2001], proposta com o intuito de fornecer uma alternativa descentralizada para auxílio à tomada de decisões. São apresentados, também, detalhes referentes à filosofia adotada, bem como, são explicadas todas as etapas que compõem o processo de solução do problema.

Esta estratégia, da forma como será apresentada na seqüência do texto, pode ser aplicada a qualquer sistema que possa ser dividido hierarquicamente. A divisão em níveis hierárquicos vem facilitar a utilização da filosofia de *repair solution* que visa alcançar uma solução para um dado problema da forma mais local possível, como será discutido posteriormente. Apesar de a estratégia considerar sistemas divididos hierarquicamente, não há restrições quanto à forma de interação entre os níveis dessa hierarquia. Um exemplo de sistema onde a estratégia poderia ser aplicada pode ser visto na Figura 3.1.

A filosofia de *repair solution*, denominada *iterative repair* em [Zweben et al, 1993 apud Tranvouez, 2001], propõe o ajuste iterativo de planos de ação de um dado sistema em vistas de restrições operacionais que venham a se apresentar. Essas modificações recorrentes dos planos acontecem até que uma situação operacional satisfatória seja alcançada e não sejam necessárias modificações adicionais.

Mantendo a mesma filosofia, mas mudando o enfoque dado, pode-se dizer que a filosofia visa minimizar a propagação de perturbações ou problemas através da

resolução deles da forma mais local possível. A filosofia é de aplicar, inicialmente, uma tentativa de resolução interna que, caso não seja suficiente para resolver o problema, é descartada e é iniciada uma tentativa de cooperação com os vizinhos mais próximos. Caso esta primeira cooperação não forneça os resultados desejados, o problema é propagado para o nível seguinte da circunvizinhança, e assim sucessivamente. Estas operações de cooperação permitem tentar resolver o problema da forma mais local possível e possuindo apenas um conhecimento parcial dele.

Aplicações diversas do uso desta filosofia podem ser encontradas na literatura. Alguns exemplos são citados a seguir. Em [Volpe et al, 2000] é proposta a utilização desta filosofia para o planejamento otimizado de rotas de navegação para robôs atuando em terrenos desconhecidos. Em [Jung et al, 2002] são propostas técnicas distribuídas de satisfação automática de restrições para coordenação e resolução de conflitos entre planos de execução de plataformas espaciais operando conjuntamente. Em [Chiu et al, 2002] é proposto um sistema computacional para gerenciamento do sistema ferroviário Chinês através do re-escalonamento de trens. Em [Zhang & Dietterich, 1995] é proposta a associação desta filosofia com o uso de redes neurais para criação de um sistema de escalonamento de tarefas em células de sistemas de manufatura (*job shops*).

Em [Tranvouez, 2001] é proposta a utilização de uma outra estratégia baseada em *repair solution* para lidar com problemas de gerenciamento de recursos em chão de fábrica. Mais especificamente, a abordagem propõe uma estratégia de re-escalonamento de tarefas para as máquinas existentes nas células de produção de um sistema industrial, levando em consideração as restrições de tempo para execução das mesmas. Ele desenvolveu um protótipo que permite, através de interações entre agentes de software, garantir um re-escalonamento cooperativo progressivo, onde o problema é dividido em partes e cada parte é resolvida completamente antes de passar para a parte seguinte. Isto é feito até que o sistema encontre uma configuração que garante o alcance de seus objetivos de produção.

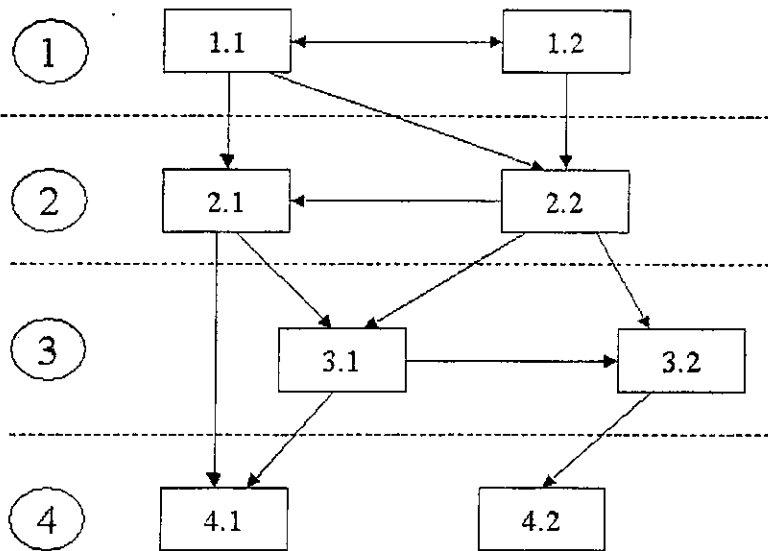


Figura 3.1 – Diagrama de Blocos Representando um Sistema Dividido em Níveis Hierárquicos

Como a estratégia descentralizada não está vinculada a um tipo de sistema específico, ela pode ser aplicada aos mais diversos sistemas industriais seja no nível de gerenciamento de recursos no chão de fábrica ou no gerenciamento estratégico dos elementos do sistema em um nível mais alto. A implementação dessa estratégia é baseada em agentes de *software* que estão associados aos elementos ou grupos de elementos presentes no sistema.

A solução é baseada em agentes que, em situações em que há a ocorrência de perturbações, têm o intuito de fornecer soluções para absorvê-las. Para que sejam modeladas estas situações, é necessário que existam regras bem definidas que ditem o comportamento destes agentes de software que participam do processo. Estes agentes têm um conhecimento apenas parcial do seu ambiente e restrito às propriedades associadas ao elemento ou grupo de elementos do sistema que ele representa. Eles podem interagir com outros agentes localizados em todos os níveis da hierarquia. Desta forma, é possível construir um sistema capaz de resolver os problemas de restauração, localmente, após um incidente no sistema [Turnell, Espinasse & Aventini, 2005]. Isto é interessante, visto que em grandes sistemas se as decisões são centralizadas há uma penalidade no que se refere, principalmente, à comunicação e ao tempo necessário para efetua-la, acarretando assim custos para a empresa.

A arquitetura proposta para implementação da estratégia é dividida em dois níveis, sendo um físico e outro de decisão, como pode ser visto na Figura 3.2.

No nível físico existe o modelo do sistema de onde provêm os valores dos parâmetros utilizados como base para as tomadas de decisões. Estes valores são passados, por meio de mensagens, para os agentes associados aos elementos correspondentes. Estes agentes têm conhecimento de informações referentes às capacidades e restrições físicas do próprio sistema, dentre outras.

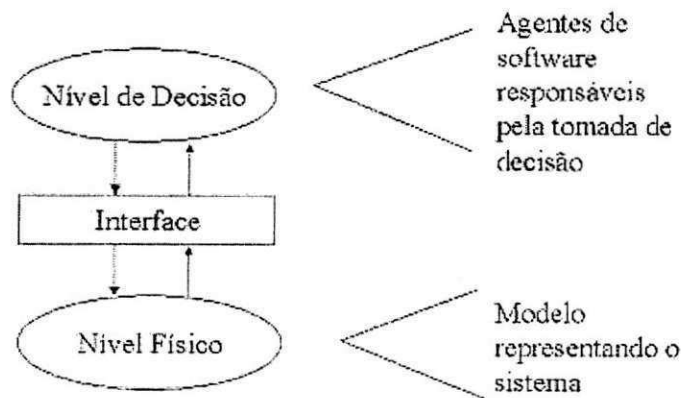


Figura 3.2 - Arquitetura para Implementação da Estratégia

No nível de decisão, os agentes são responsáveis pela avaliação das informações fornecidas pelo nível físico e pelo fornecimento de soluções para os problemas impostos pelas condições atuais do sistema. Esta avaliação é feita com base nas regras que definem o comportamento dos agentes.

A interface existente entre estes dois níveis tem a finalidade de facilitar o estabelecimento da comunicação entre eles. Ela pode ser implementada através da utilização de um agente que distribui as mensagens para o nível superior. Este tipo de solução elimina a dificuldade de endereçamento para estabelecer a comunicação entre o modelo físico do sistema (um aplicativo externo) e os agentes associados a ele. Entretanto, como será visto em uma seção posterior do texto, existem outras possibilidades para se implementar esta camada de interface.

Em sistemas industriais reais, a operação é feita, tipicamente, com uma pequena margem de segurança, à exceção de alguns sistemas que operam sob demanda, como o

sistema elétrico, por exemplo, de modo que na ocorrência de uma contingência simples, quando possível, o sistema encontra automaticamente um novo ponto de operação. Essa possibilidade de encontrar um novo ponto de operação sem que medidas drásticas sejam necessárias cria a possibilidade de resolução do problema através de interações entre agentes e da utilização da filosofia de *repair solution*.

Então, com base nesta filosofia definimos as fases componentes de nossa estratégia de descentralização da tomada de decisão. Na seção seguinte são descritas as quatro fases componentes da estratégia aplicadas a um contexto de sistemas elétricos de potência.

3.1 – Aplicação da Estratégia a um Sistema Elétrico

Antes de discutir a aplicação da estratégia, é necessário definir como o sistema é estruturado. Sabemos que para facilitar, dentre outras coisas, o estudo de sistemas de potência, eles são divididos em subsistemas de geração, transmissão e distribuição. Conseqüentemente, quando se trata de subestações existentes no sistema existem as subestações elevadoras, as de subtransmissão e as de distribuição que podem ser classificadas hierarquicamente.

Para o escopo da estratégia aqui proposta apenas os três níveis superiores, que contém as subestações elevadoras e de subtransmissão, são considerados. As subestações de distribuição representam apenas a carga do sistema concentrada que serve para estabelecer a demanda do sistema.

O modelo do sistema elétrico considerado no estudo de caso é uma visão simplificada, tanto em termos de tamanho quanto em termos de complexidade, de um sistema real. Como a intenção é de avaliar a estratégia, um sistema simples contendo apenas oito instalações divididas nos quatro níveis hierárquicos do sistema de transmissão foi utilizado. Esta visão simplificada é mostrada na Figura 3.3.

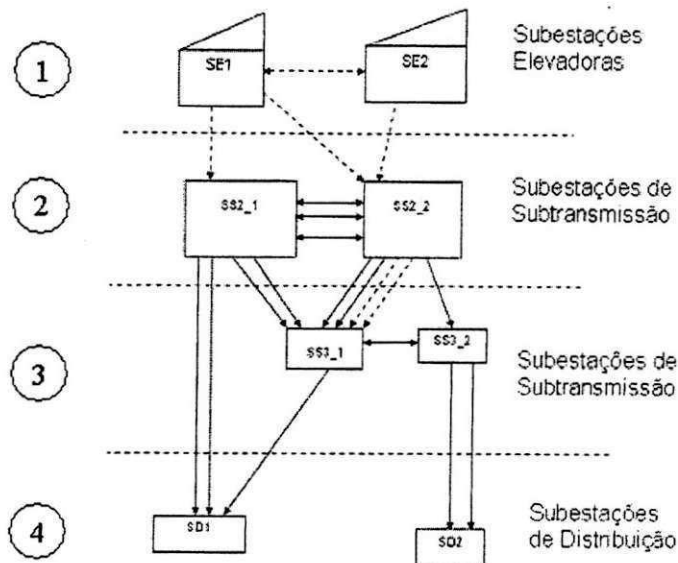


Figura 3.3 - Estrutura da Rede Considerada no Estudo de Caso

A seguir, é apresentada uma breve descrição do que representa cada um dos níveis desta hierarquia.

- Nível 1: é o nível mais elevado e contém as subestações elevadoras (SE) que estão conectadas diretamente às usinas geradoras e possuem o papel de elevar o nível de tensão para fins de transmissão da energia elétrica;
- Nível 2: este nível contém as subestações de subtransmissão (SS2) que estão diretamente conectadas às subestações elevadoras (SE). Estas subestações podem alimentar outras localizadas nos níveis 3 e 4;
- Nível 3: também contém subestações de subtransmissão (SS3), mas estas estão diretamente conectadas ao sistema de distribuição e são alimentadas pelas subestações de subtransmissão (SS2) do segundo nível;
- Nível 4: aqui se localizam as subestações de distribuição (SD) que são, em nossa abordagem, os consumidores finais. São estes elementos que definem a demanda do sistema.

Uma vez que nossa proposta não está interessada no nível de distribuição da rede, consideram-se aqui apenas as situações onde há a perda de algum elemento do sistema de geração/transmissão que force a existência de comunicação entre os operadores e o centro de controle. Em outras palavras, consideram-se apenas as situações onde não é possível efetuar um simples religamento automático. Esta comunicação visa determinar caminhos alternativos para o fluxo de potência e/ou estratégias para rejeição de carga.

Este estudo de caso refere-se à existência de uma falta no *link* que une as subestações SS2_1 e SS3_1, como mostrado na Figura 3.4.

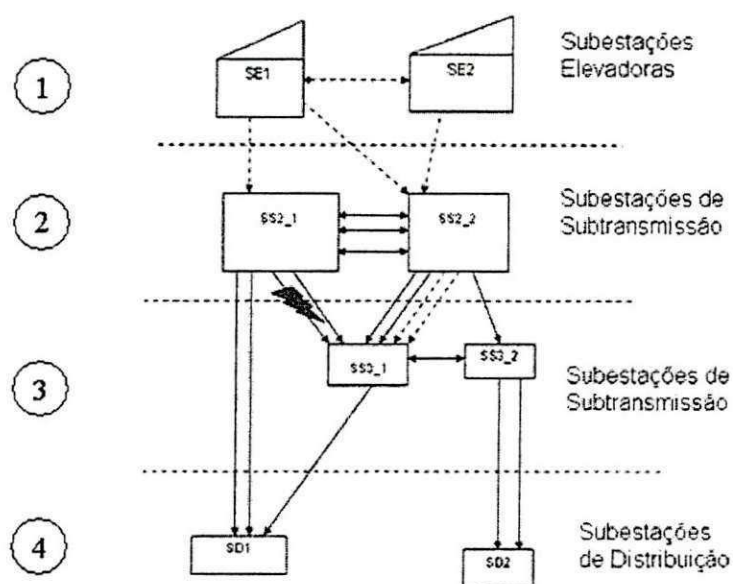


Figura 3.4 - Representação do Sistema Contendo Uma Perturbação

Então, aplicando a estratégia para restauração do sistema, as seguintes fases estão presentes:

- **Primeira Fase:** Se existe alguma outra linha conectando as duas subestações envolvidas, ambas irão negociar a fim de distribuir a energia entre as linhas remanescentes. Uma vez que há energia disponível no lado do alimentador, esta avaliação é feita com base na capacidade operacional das linhas. Se não for possível transmitir

a energia através das linhas remanescentes, passa-se para a segunda fase;

- **Segunda Fase:** Aqui, tenta-se resolver o problema no mesmo nível que a perturbação ocorreu. Os elementos localizados neste nível negociam e fornecem o excedente de energia existente a fim de encontrar localmente uma possível solução para o problema. Para esta negociação, o destinatário da requisição avalia se a energia solicitada é menor que 50% de seu excedente disponível. Se sim, então é possível fornece-la. Caso contrário, não existe solução possível nesta fase e o problema é propagado para um nível superior;
- **Terceira Fase:** Esta fase pode ser repetida tantas vezes quantos forem os níveis hierárquicos ainda não atingidos pelo processo de obtenção da solução e é dividida em dois passos, como explicado abaixo.
 - **Passo 1:** O problema é propagado, como uma solicitação de excedente de energia, para um nível acima na hierarquia, mas envolvendo apenas as subestações diretamente conectadas àquela que foi afetada. Se existem mais de uma, uma função custo é usada para determinar qual é a melhor opção. Aqui, a mesma avaliação realizada na fase 2 é válida. Isto é feito para garantir que sempre haverá algum excedente de energia no caso de um problema local.
 - **Passo 2:** Se o problema não for resolvido no passo 1, então ele será propagado para as outras subestações como uma solicitação de cooperação enviada pelas subestações envolvidas no primeiro passo;
- **Quarta Fase:** Se não for possível resolver o problema nas três primeiras fases, então o problema é propagado para o nível mais alto na hierarquia, que são as subestações elevadoras. Esta fase, assim como a terceira, pode ser dividida em dois passos que são similares àqueles apresentados na terceira fase. Nesta fase a avaliação é feita com relação à capacidade máxima de produção. Se a demanda total for menor que a capacidade de

produção, o problema pode ser resolvido sem nenhuma rejeição de carga. Se não for, a subestação elevadora solicitará cooperação e, apenas se não houver como prover aquilo que foi demandado, será iniciado um processo de rejeição de carga.

Na Figura 3.5 é apresentado o fluxo de execução do processo.

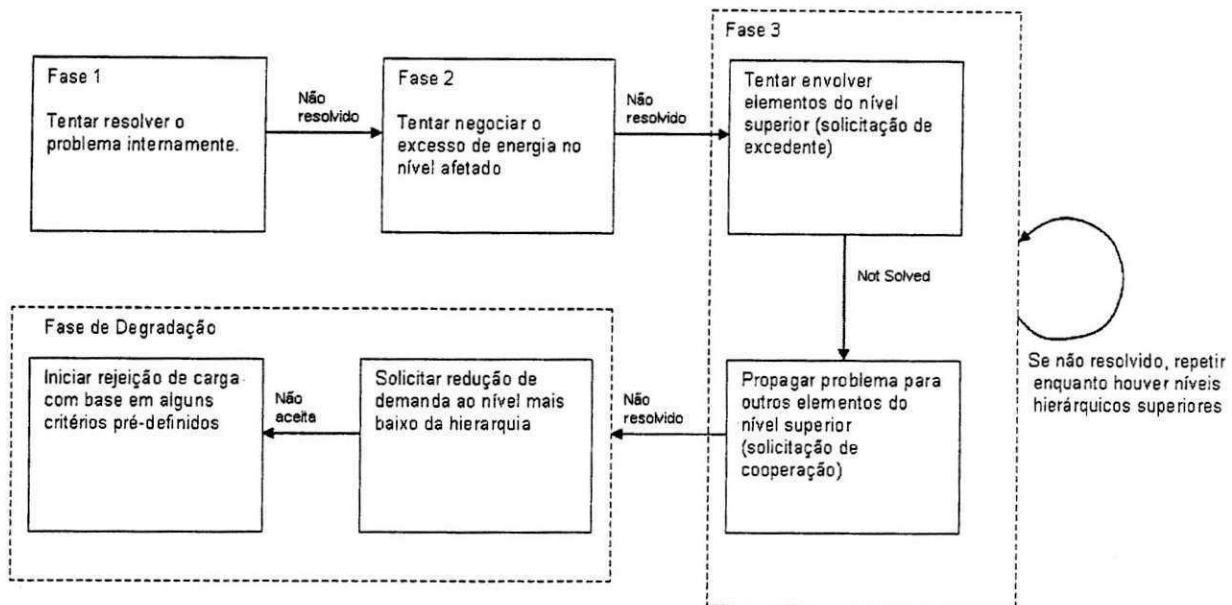


Figura 3.5 - Fluxo de Execução do Processo

Para implementar a estratégia é necessário definir tipos diferentes de agentes que interajam a fim de propiciar a realização das tarefas de interesse. No caso específico de sistemas de potência foram definidos seis tipos de agentes que são:

- Agente Subestação Elevadora (SE) – Este agente, como o próprio nome diz, representa as subestações elevadoras que estão localizadas junto das usinas e têm o propósito de aumentar o nível de tensão para diminuir perdas de transmissão. Este agente é capaz de avaliar a possibilidade de fornecer consumidores através da comparação entre a demanda e sua capacidade máxima de produção. Ele pode, também, cooperar com outros agentes do mesmo tipo para prover algum excedente de energia, no caso de solicitação;

- Agente Subestação de Subtransmissão Nível 2 (SS2) – Este tipo de agente representa as subestações de subtransmissão alimentadas diretamente pelas subestações elevadoras. Estes agentes são aptos a realizar tarefas como avaliar capacidade de fornecimento, solicitar cooperação tanto de agentes do mesmo tipo quanto de agentes tipo SE, e solicitar redução de demanda aos agentes do tipo SS3 durante o processo de rejeição de carga;
- Agente Subestação de Subtransmissão Nível 3 (SS3) – Agentes deste tipo representam as subestações de subtransmissão que não estão conectadas diretamente às subestações elevadoras. Estas subestações podem estar conectadas ou não às subestações de distribuição. Estes agentes devem ser capazes a avaliar a possibilidade de cooperação de acordo com excedente de energia disponível e iniciar o processo de rejeição de carga emitindo solicitação de redução de demanda;
- Agente Subestação de Distribuição (SD) – Estes agentes são responsáveis por definir a demanda efetiva do sistema (excetuando-se as perdas) e, também, devem ser capazes de avaliar quais cargas devem ser rejeitadas no caso de um processo de rejeição de carga ser iniciado;
- Agente Linha de Transmissão (LT) – Este tipo de agente é responsável por avaliar a capacidade de transmissão comparando sua máxima capacidade de transmissão à soma de demanda e perdas calculadas. O cálculo das perdas de potência é realizado com base em alguns parâmetros elétricos tais como, nível de tensão, resistência e indutância da linha;
- Agente Perturbador (AP) – Este agente é usado para fins de simulação apenas e é usado para modelar algumas faltas no sistema. Ele deve ser capaz de enviar mensagens para todos os outros agentes presentes no sistema e o conteúdo destas mensagens irá inserir algum mau funcionamento na rede.

Cada tipo de agente, que representa um tipo de elemento do sistema, possui um conjunto de propriedades através das quais a dinâmica do sistema é ditada. No Quadro 1 são exibidas as propriedades associadas a cada tipo de agente.

SE	SS3/SS2
Nome	Nome
Capacidade Máxima de Produção (CPMax)	Capacidade Máxima de Transmissão (CTMax)
Consumidores [1,...,n]	Consumidores [1,...,n]
Linhas [1,...,n]	Fornecedores [1,...,n]
Produção (P)	Linhas [1,...,n]
Linha Nível [1,...,n]	Excedente
	Linha Nível [1,...,n]
SD	LT
Nome	ID
Capacidade Máxima (CM)	Capacidade Máxima (Cap_Max)
Fornecedores [1,...,n]	Subestações [1,...,n]
Linhas [1,...,n]	Distância (D)
Linha Nível [1,...,n]	Resistência (R)
Consumidores Prioridade 1 (Cons1)	Restância (X)
Consumidores Prioridade 2 (Cons2)	Links
Consumidores Prioridade 3 (Cons3)	Custo (D, R, X)
Demanda (D) = Cons1+Cons2+Cons3	

Quadro 1 - Representação dos conhecimentos de cada tipo de agente

Na seqüência, para cada fase da estratégia, são mostradas quais regras são avaliadas pelos agentes, bem como um diagrama de seqüência de mensagens representando as interações entre os agentes envolvidos em cada fase e os grafos RCA gerados para cada um destes agentes.

O RCA é uma notação gráfica semi-formal composta por grafos de estados fortemente tipados, tanto no nível de estados quanto no nível de transições, para a definição de diferentes tipos de estados e transições adaptados a um contexto multiagente [Tranvouez, 2001]. Maiores detalhes sobre o RCA serão apresentados no Capítulo 4.

3.1.1 – Fase 1 – Resolução Interna

Para a primeira fase, a transferência do fluxo de potência através de uma linha de transmissão existente no caso de falta em outra linha do mesmo *link* deve ser feita sem troca de mensagens. Entretanto, para fins de representação, nós consideramos uma solicitação e sua resposta dependendo da capacidade das linhas de transmissão.

No que se refere às regras que regem esta fase, é possível representá-las por meio dos pseudocódigos mostrados no Quadro 2.

```
*/em LT/*
Recebe Msg de AP
Se (Msg = "out") então
    Informa Subestações
Fim

*/em SS3_1/*
Recebe Msg de LT
Se (links(LT) ≠ 0) então
    Solicita Fornecimento
Senão Se (Linha Nível ≠ 0) então
    Inicia Fase 2
    Senão
        Inicia Fase 3
Fim
Fim

*/em LT, ao receber solicitação de fornecimento/*
Recebe Msg de SS3_1
Calcula Nova Carga
Se (Nova Carga < Cap_Max) então
    Confirma Subestações
Senão
    Informa Subestações "Acima da Capacidade"
Fim
```

Quadro 2 - Pseudocódigo Referente à Dinâmica da Fase 1

Então, as trocas de mensagens entre os elementos envolvidos na fase 1 são representadas no diagrama de seqüência de mensagens mostrado na Figura 3.6.

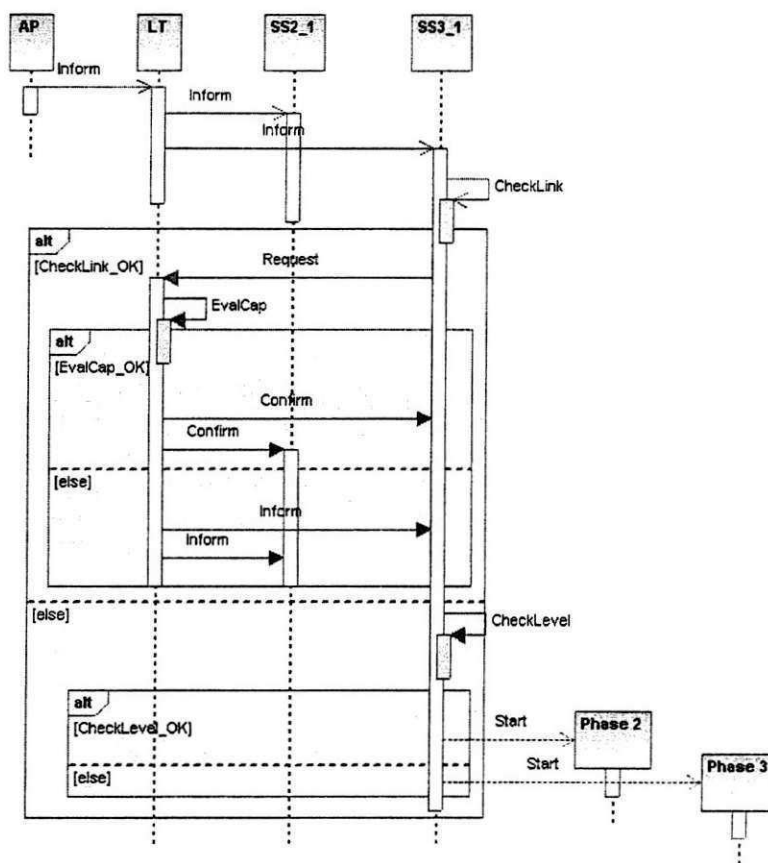


Figura 3.6 - Diagrama de Seqüência de Mensagens para os Agentes Envolvidos na Fase 1

Os grafos RCA para os elementos envolvidos nesta fase são mostrados nas Figuras 3.7, 3.8 e 3.9. Na Figura 3.7 é mostrado o grafo para o elemento SS3_1. O grafo para o elemento SS2_2 é mostrado na Figura 3.8 e, por fim, o grafo da linha que os interconecta é exibido na Figura 3.9.

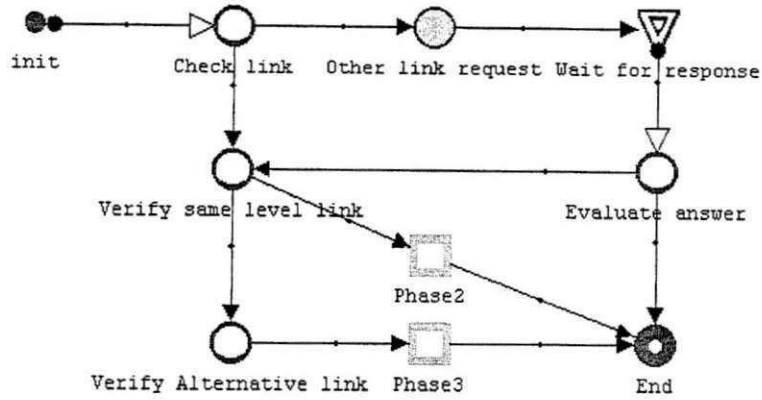


Figura 3.7 - Grafo RCA para elemento SS3_1 na Fase 1

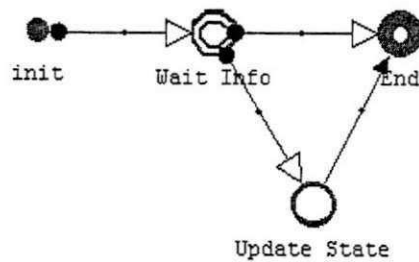


Figura 3.8 - Grafo RCA para o elemento SS2_2 na Fase 1

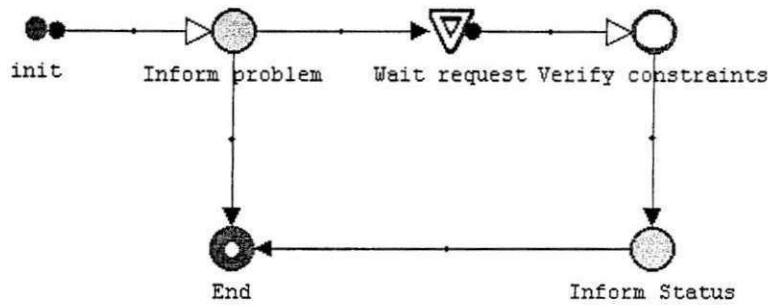


Figura 3.9 - Grafo RCA para o elemento LT na Fase 1

3.1.2 – Fase 2 – Propagação em Mesmo Nível

Para a segunda fase, tenta-se resolver o problema no mesmo nível em que ele ocorre. Então, considerando a rede dada, todos os elementos do tipo SS3 negociam a

fim de tentar resolver o problema sem propagá-lo. As regras são avaliadas com a chegada de uma mensagem de SS3_1 para SS3_2, como pode ser visto no Quadro 3.

```
*/em SS3_1/*
Solicita Auxílio
Aguarda Resposta

*/em LT/*
Recebe Msg de SS3_1
Calcula Nova Carga
Se (Nova Carga < Cap_Max) então
    Repassa Msg
Senão
    Nega Auxílio
Fim

*/em SS3_2/*
Recebe Msg de LT
Se (Auxílio ≤ Excedente/2) então
    Confirma Auxílio_OK
Senão
    Inicia Fase 3
    Se (Auxílio ≤ (Excedente/2) + Resultado Fase 3) então
        Confirma Auxílio_OK
    Senão
        Confirma Auxílio_Parcial
    Fim
Fim

*/em SS3_1/*
Recebe Msg
Se (Remetente = LT) então
    Inicia Fase 3
Senão Se (Msg = "Auxílio_Parcial") então
    Atualiza Auxílio
    Inicia fase 3
    Senão
        Fim
Fim
```

Quadro 3 - Pseudocódigo Referente à Dinâmica da Fase 2

As possíveis interações dos elementos da rede para a fase 2 são mostradas na Figura 3.10. Assim como, os grafos que representam o comportamento dos elementos envolvidos nesta fase são mostrados nas Figuras 3.11, 3.12 e 3.13.

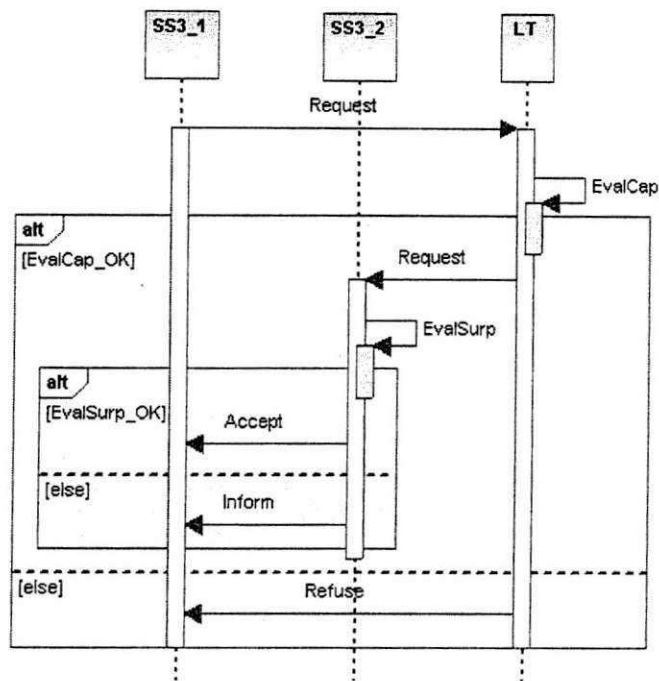


Figura 3.10 - Diagrama de Seqüência de Mensagens para os Agentes Envolvidos na Fase 2

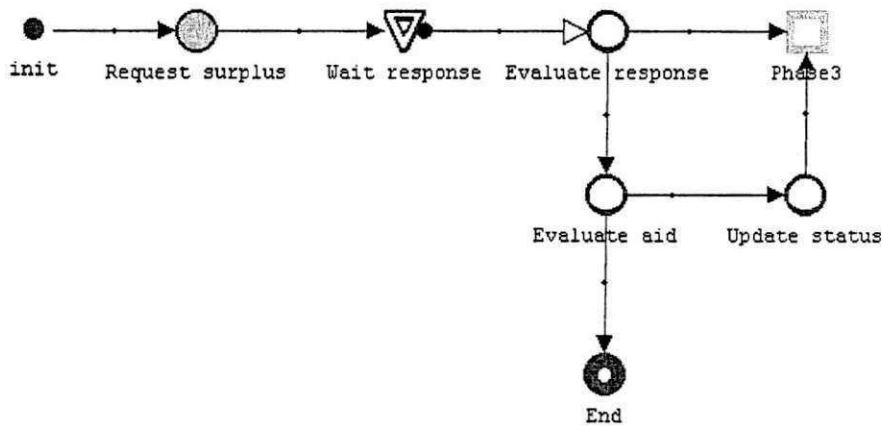


Figura 3.11 - Grafo RCA para o elemento SS3_1 na Fase 2

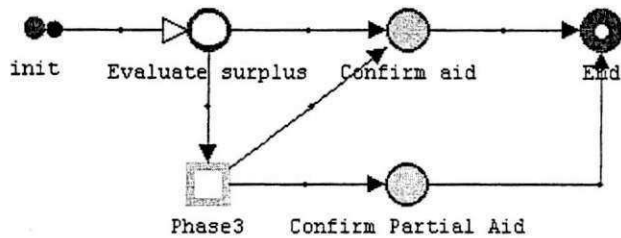


Figura 3.12 - Grafo RCA para o elemento SS3_2 na Fase 2

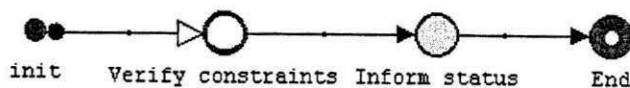


Figura 3.13 - Grafo RCA para o elemento LT nas Fases 2, 3 e 4

3.1.3 – Fase 3 – Propagação para Nível Superior (I)

Assim como nas fases anteriores, as regras avaliadas nesta terceira fase são representadas por meio de um pseudocódigo, como exibido no Quadro 4.

```

*/em SS3_1/*
Solicita Auxílio
Aguarda Resposta

*/em LT/*
Recebe Msg de SS3_1
Calcula Nova Carga
Se (Nova Carga < Cap_Max) então
    Repassa Msg
Senão
    Nega Auxílio
Fim

*/em SS2_2/*
Recebe Msg de LT
Se (Auxílio ≤ Excedente/2) então
    Confirma Auxílio_OK
Senão
    Inicia Fase 2, sendo SS2_2 a solicitante
    N_Excedente = (Excedente/2) + Resultado Fase 2
    Se (Auxílio ≤ N_Excedente) então
        Confirma Auxílio_OK
    Senão
        Inicia Fase 4
Fim
Fim

```

```

*/em SS3_1/*
Recebe Msg
Se (Remetente = LT) então
  Inicia Rejeição
Senão Se (Msg = "Auxílio_Parcial") então
  Atualiza Auxílio
  Inicia Rejeição
Senão
  Fim
Fim

```

Quadro 4 - Pseudocódigo Referente à Dinâmica da Fase 3

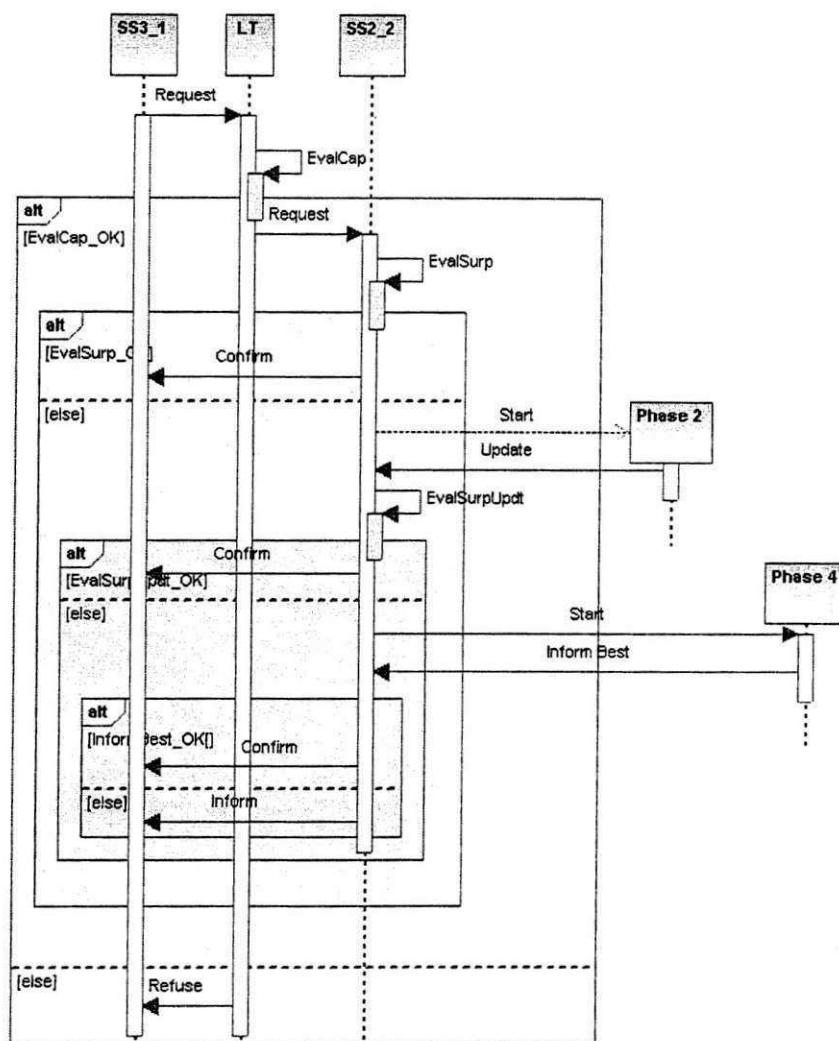


Figura 3.14 - Diagrama de Seqüência de Mensagens para os Agentes Envolvidos na Fase 3

O mesmo tipo de diagrama de seqüência é exibido para a terceira fase, sendo que os dois passos desta fase estão representados no mesmo diagrama. O diagrama é mostrado na Figura 3.14. Nas Figuras 3.15 e 3.16 são mostrados os grafos RCA

relativos aos elementos SS3_1 e SS2_2, que participam desta fase do processo. O grafo RCA que representa o comportamento do elemento LT é exatamente igual ao mostrado na Figura 3.13.

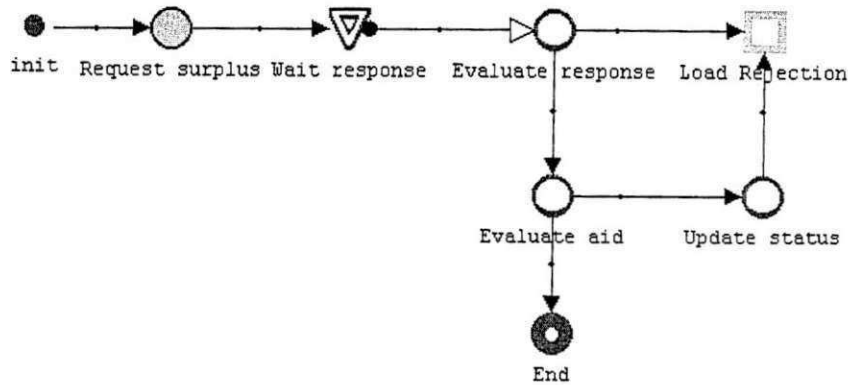


Figura 3.15 - Grafo RCA para o elemento SS3_1 na Fase 3

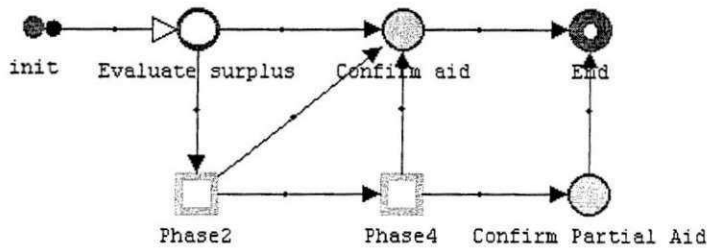


Figura 3.16 - Grafo RCA para o elemento SS2_2 na Fase 3

3.1.4 – Fase 4 – Propagação para Nível Superior (II)

A quarta fase consiste de uma repetição da terceira fase da estratégia envolvendo o nível mais alto da hierarquia, que contém as subestações elevadoras. As regras para o estudo de caso, relacionadas a esta fase, são mostradas no Quadro 5.

/em SS2_2/

Resto = Auxílio - N_Excedente

Solicita Resto

Aguarda Resposta

/Em LTx/

Recebe Msg de SS2_2

Calcula Nova Carga

Se (Nova Carga < Cap_Max) então

 Repassa Msg

Senão

 Nega Auxílio

Fim

/em SEx/

Recebe Msg de LT

Folga = CPMax - P

Se (Resto ≤ Folga) então

 Confirma Auxílio_OK

Senão

 Inicia Fase 2 sendo a solicitante

 Folga_Max = Folga + Resultado Fase 2

 Se (Resto ≤ Folga_Max) então

 Confirma Auxílio_OK

 Senão

 Confirma Auxílio_Parcial

 Fim

Fim

/em SS2_2/

Para i = 1 até Número de Linhas

 Recebe Msg (i)

 Se (Remetente (i) = LT) então

 Senão

 Registra Oferta (i)

 Fim

 i = i+1

Fim

Peso = infinito

Para j=1 até Número de Ofertas

 Se (Oferta (j) = "Auxílio_OK") então

 Flag = 1

 Temp = Custo (j)

 Se (Peso > Temp) então

 Peso = Temp

 Melhor = Oferta (j)

 Senão

 Fim

 Senão Se (Flag ≠ 1) então

```

Temp = Custo (j)
Se (Peso > Temp) então
    Peso = Temp
    Melhor = Oferta (j)
Senão
    Fim
Senão
    Fim
Fim

```

Quadro 5 - Pseudocódigo Referente à Dinâmica da Fase 4

Na seqüência, é possível observar nas Figuras 3.17 e 3.18, respectivamente, os grafos referentes aos comportamentos dos elementos SS2_2 e SEx na Fase 4. O grafo que representa o comportamento do elemento LTx é o mesmo apresentado na Figura 3.13. Enquanto que, na Figura 3.19, é possível observar o diagrama referente à quarta fase, onde a negociação envolve também as subestações elevadoras.

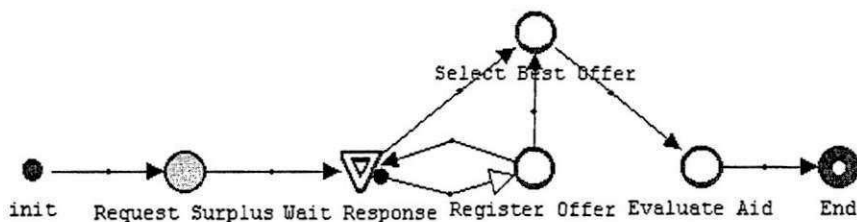


Figura 3.17 - Grafo RCA para o elemento SS2_2 na Fase 4

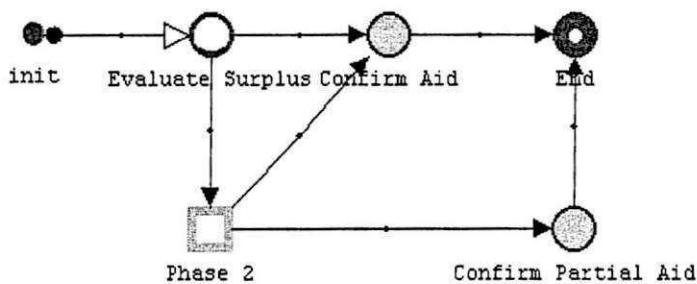


Figura 3.18 - Grafo RCA para o elemento SEx na Fase 4

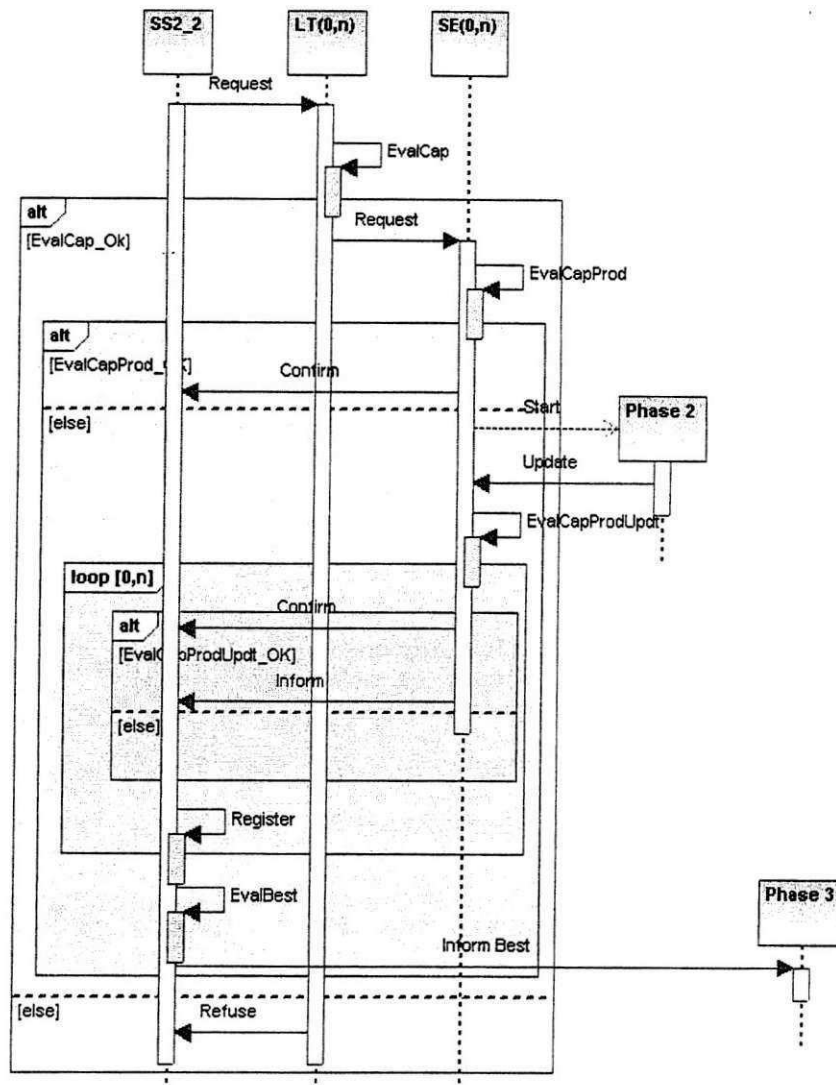


Figura 3.19 - Diagrama de Seqüência de Mensagens para os Agentes Envolvidos na Fase 4

3.1.5 – Processo de Rejeição de Carga

Caso nenhuma das fases anteriores seja suficientemente para a obtenção de uma solução, é iniciado um processo de rejeição de carga no sistema. Para este sistema, especificamente, esse processo não é complexo visto que apenas um elemento SD está diretamente ligado ao elemento SS3_1, inicialmente afetado pela perturbação. No Quadro 6 estão listadas as regras que são disparadas nesse processo. A Figura 3.20 mostra o diagrama de seqüência de mensagens para este processo.

```

*/em SS3_1/*
Solicita Redução de Demanda
Aguarda Resposta

*/em SD1/*
Se (Redução de Demanda > Demanda (Cons1)) então
    Informa Redução_Ref
Senão
    Informa Redução_OK
Fim

*/em SS3_1/*
Recebe Msg
Se (Msg = "Redução_OK") então
    Confirma Recebimento
Senão
    Informa "No Solution"
Fim

```

Quadro 6 - Pseudocódigo Referente à Dinâmica do Processo de Rejeição de Carga

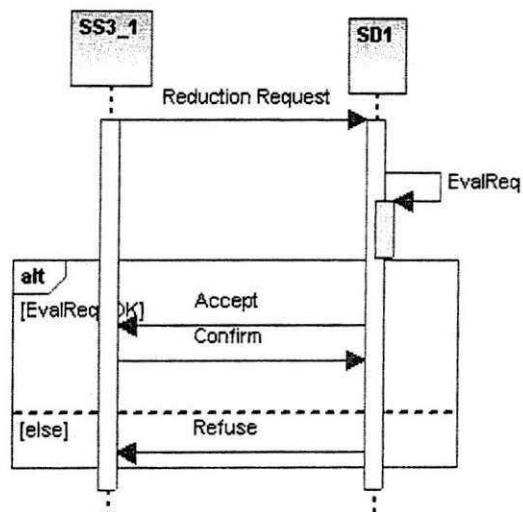


Figura 3.20 - Diagrama de Seqüência de Mensagens para os Agentes Envolvidos no Processo de Rejeição de Carga

Na seqüência, são mostrados, nas Figuras 3.21 e 3.22, os grafos RCA que representam o comportamento dos agentes SS3_1 e SD1, envolvidos neste processo de rejeição de carga.

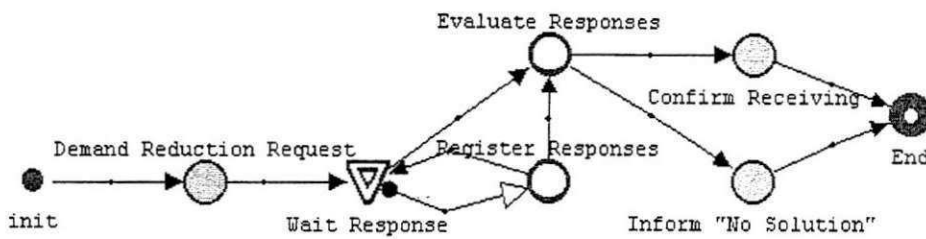


Figura 3.21 - Grafo RCA para o elemento SS3_1 no processo de rejeição de carga

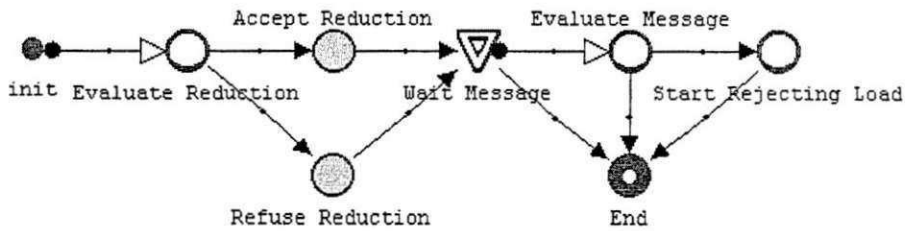


Figura 3.22 - Grafo RCA para o elemento SD1 no processo de rejeição de carga

Por fim, é importante mencionar que esta descentralização proposta deve ser feita sem ônus para as empresas ou seus funcionários, visto que a finalidade da proposta é dar mais dinamismo e segurança às operações que, tipicamente, envolvem uma grande carga de comunicação direta entre centro de controle e elemento componente do sistema e estão associadas ao risco de falha operacional com graves conseqüências. Tem-se o intuito apenas de se fornecer mais um recurso ferramental ao operador para que o mesmo se sinta mais confortável na resolução de problemas que venham a ocorrer durante sua supervisão.

No próximo capítulo, são descritas tanto a plataforma quanto as ferramentas utilizadas para modelagem da camada multiagentes que implementa a estratégia descrita neste capítulo.

Capítulo 4

Modelagem do Sistema Multiagentes

Este capítulo é dedicado à apresentação da plataforma e do processo de modelagem da camada multiagentes utilizada para implementar a estratégia descrita no Capítulo 3. O foco é dado nas características existentes na camada de desenvolvimento dos agentes de software, ou seja, são apresentados detalhes das ferramentas utilizadas durante a criação da estrutura de tomada de decisão. A camada mais baixa da arquitetura deve ser tal que seja capaz de representar o sistema a ser estudado da melhor forma possível e, por isso, dependendo da natureza do sistema, diversas ferramentas podem ser apropriadas.

Como explicado anteriormente, a camada superior da arquitetura proposta, vide Figura 3.2, é responsável pelas tomadas de decisão que impactam no comportamento do sistema. Para se construir essa camada faz-se uso de algumas ferramentas para determinação do comportamento (concepção) e implementação dos agentes de software, que fazem parte de uma plataforma lógica já existente chamada MAJORCA (*Moteur Agents Jess Orienté Représentation de Comportement d'Agents*), proposta em [Tranvouez, 2001]. Esta plataforma fornece um ambiente de desenvolvimento de sistemas multiagentes baseados na noção de planos comportamentais especificados com o auxílio do formalismo RCA (*Représentation des Comportements d'Agents*) [Labarthe, 2006], e atendem os requisitos desejados para implementação de nossa estratégia dando suporte à criação das regras que serão avaliadas durante a execução da mesma.

A utilização destas ferramentas é uma etapa posterior à especificação do comportamento dos agentes que fazem parte do sistema. Esta especificação corresponde

à determinação dos planos comportamentais dos agentes tanto de um ponto de vista local quanto de um ponto de vista social. No ponto de vista local se refere aos comportamentos individuais do agente a fim de alcançar seus objetivos próprios. No ponto de vista social está associado aos comportamentos decorrentes das interações com outros agentes para a realização de objetivos comuns. Cada um destes níveis de representação implica em uma especificação detalhada de comportamentos, funções e protocolos de interação do agente. Estes diferentes planos comportamentais são divididos em:

- Plano Local (PL) – é um plano que descreve um comportamento isolado, isto é, representa o comportamento que não necessita de interações para ser cumprido. Deve-se notar que essa ausência de comunicação não implica dizer que o agente é egoísta;
- Protocolo de Interação (PI) – é um plano que representa o comportamento de um agente em interação com outro. A existência de um PI pressupõe a existência de ao menos um plano dual, que juntos formam uma conversação [Tranvouez, 2001];

Como dito anteriormente, para representação destes comportamentos é utilizado o formalismo RCA. Verdadeiramente, os grafos RCA, assim como outros formalismos gráficos para descrição de protocolos de interação entre agentes encontrados na literatura, não é dotado de uma semântica formal [Tranvouez, 2001 apud Mokhati, Badri & Badri, 2006]. Estes formalismos gráficos oferecem apenas uma especificação semi-formal das interações entre agentes [Paurobally, Cunnigham & Jennings, 2003].

A utilização do RCA deve-se à fácil compreensão decorrente de sua natureza gráfica e, também, devido à existência de uma ferramenta que fornece suporte para a utilização do formalismo. O RCA utiliza a notação mostrada na Figura 4.1 para representar os planos definidos.

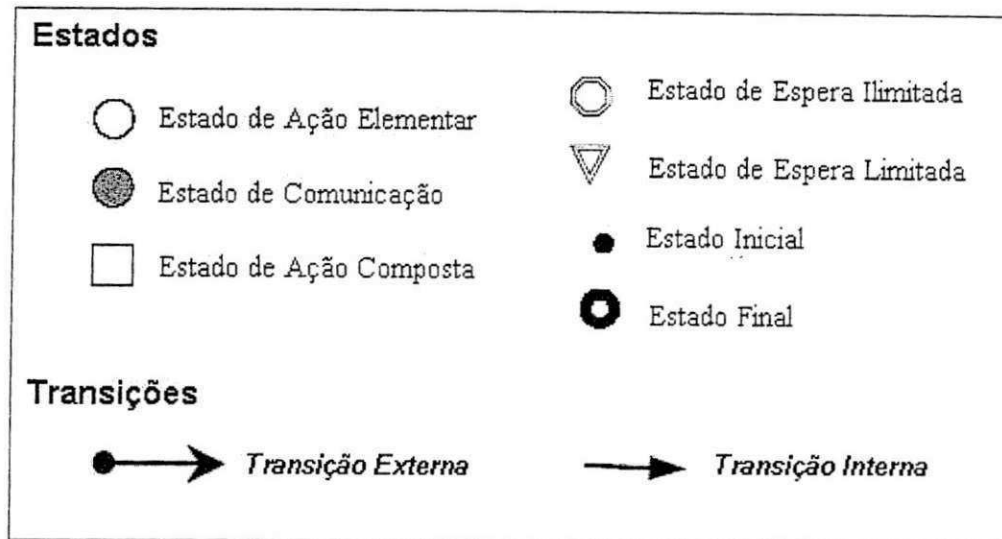


Figura 4.1 - Convenções Gráficas do Formalismo RCA

Então, conforme mostrado na Figura 4.1, no formalismo existem três estados de ação e dois estados de espera, além dos estados inicial e final. Os estados de ação são:

- Estado de ação elementar – corresponde a toda ação atômica (exceto envio de mensagens) que não necessita de procedimentos complexos para ser realizada;
- Estado de ação composta – corresponde a toda ação que pode ser decomposta em ações elementares reagrupadas em um plano comportamental. A ativação de um estado de ação composta corresponde à inicialização do plano comportamental associado a ele;
- Estado de comunicação – corresponde à ação de envio de mensagens. Este estado, juntamente com as transições externas, corresponde aos pontos de entrada e saída do fluxo de informações entre agentes;

Com relação aos estados de espera, eles podem ser apenas limitados ou ilimitados. As diferenças são:

- Estado de espera ilimitada – corresponde a um estado de bloqueio, ou seja, a saída deste estado está condicionada a uma determinada condição. Neste caso, à chegada de um evento externo;
- Estado de espera limitada – corresponde a um estado de espera que possui uma condição temporizada associada. Assim, como no estado

de espera ilimitada, o agente aguarda a ocorrência de um evento externo, mas apenas durante um intervalo de tempo definido. Caso não ocorra durante este intervalo, o agente passa para um outro estado;

Quanto às transições, existem as internas e as externas cujas diferenças são descritas a seguir:

- Transição interna – ocorre se o evento que origina a mudança de estado é interno ao agente, como resultado de um cálculo, por exemplo;
- Transição externa – ocorre se o evento que origina a mudança de estado é externo ao agente, como a chegada de uma mensagem, por exemplo;

Na Figura 4.2, é apresentado um metamodelo que representa as relações entre os estados e transições que compõem o formalismo RCA.

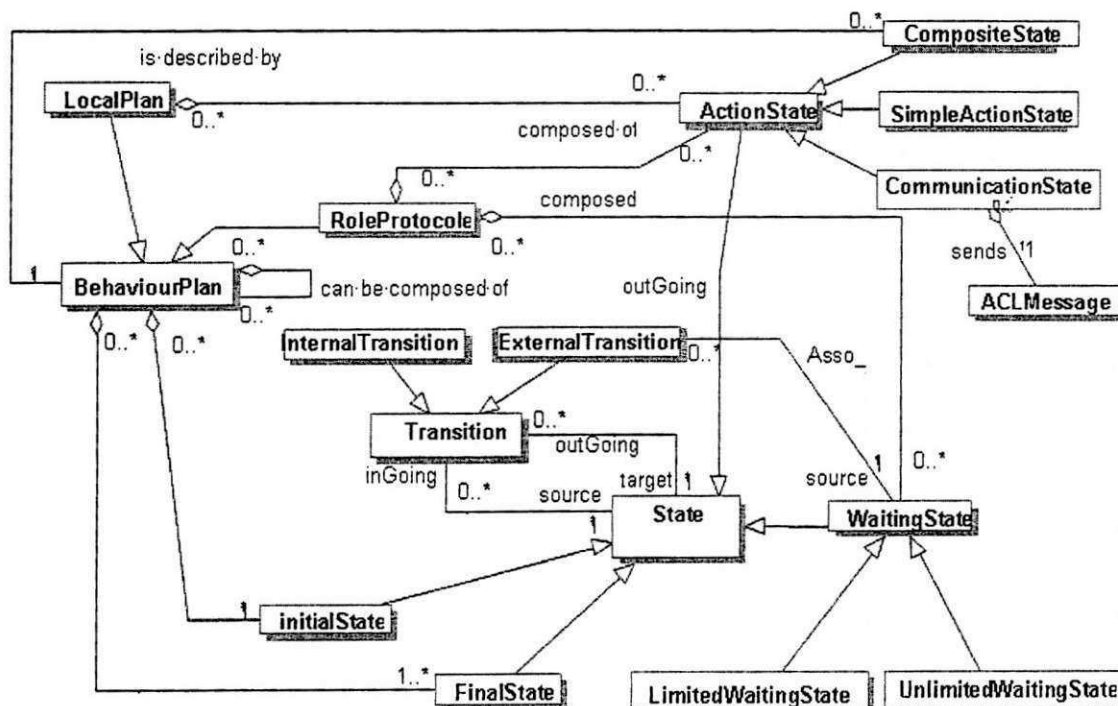


Figura 4.2 - Metamodelo Representando as Relações entre Estados e Transições do RCA [Tranvouez, 2001]

Associada a estes estados e transições que compõem o formalismo existe um conjunto de nove regras que determina algumas características importantes do mesmo. Estas regras são [Moyaux, 2000]:

- Regra 1 – Um grafo é uma seqüência de estados e transições; dois estados são separados por, pelo menos, uma transição;
- Regra 2 – O grafo começa por um único estado inicial e termina em um único estado final;
- Regra 3 – Apenas uma transição sai do estado inicial. Ela pode ser interna ou externa;
- Regra 4.a – Nenhuma transição chega ao estado inicial;
- Regra 4.b – Nenhuma transição sai do estado final;
- Regra 5 – As transições internas se encontram em qualquer número não nulo depois de estados de ação;
- Regra 6 – Uma ou duas transições internas podem sair de um estado de comunicação;
- Regra 7 – Apenas uma transição interna sai de um estado de espera limitada;
- Regra 8 – As regras 5, 6 e 7 mencionam as únicas transições internas que podem sair de um estado;
- Regra 9 – As transições externas se localizam em quantidade indefinida não nula apenas depois de estados de espera (limitada ou ilimitada) e nula em qualquer outro local;

Estas regras auxiliam na etapa de especificação conceitual com o uso de ferramentas que serão detalhadas na seqüência do texto. Após esta etapa, é necessário criar o código que garantirá que o comportamento desejado será alcançado. Esta é uma etapa que consome bastante tempo, mas que pode ser facilitada de certa forma através do uso de ferramentas que geram uma estrutura básica de código de forma automática. Em outras palavras, o uso de uma ferramenta destas pode tornar a geração de códigos para representação de agentes de software uma tarefa semi-automática, e é esta a proposta de utilização da plataforma MAJORCA, que será descrita na seção seguinte.

Para que os agentes de *software* possam implementar a estratégia definida no Capítulo 3, é necessário definir os tipos de agentes e como será a interação entre eles a fim de propiciar a realização das tarefas de interesse. Esta etapa de especificação dos tipos de agentes está diretamente relacionada ao tipo de sistema em que esta estratégia será aplicada.

É extremamente importante que a definição dos papéis de cada agente seja feita de forma cuidadosa e inequívoca, pois eles são responsáveis por realizar os passos definidos na estratégia e, assim, proporcionar a descentralização de alguns aspectos da tomada de decisão em sistemas tipicamente centralizados. Tais agentes possuem arquitetura dividida em quatro módulos básicos que são os de comunicação, conhecimentos, avaliação e decisão, como mostrado na Figura 4.3.

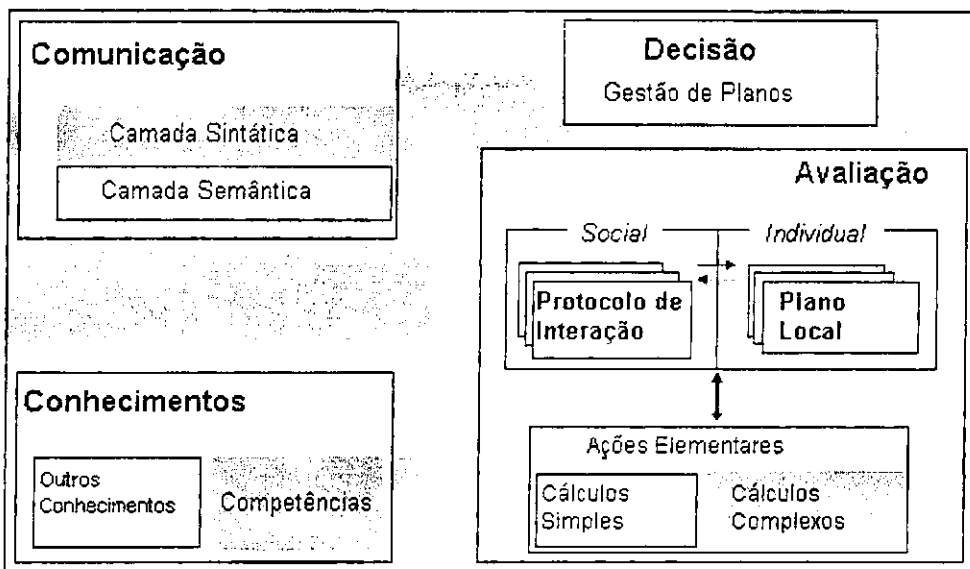


Figura 4.3 - Arquitetura dos Agentes Cognitivos

A seguir é fornecida uma breve descrição a respeito de cada um dos módulos que compõem os agentes.

- **Módulo de Comunicação:** tem a função de gerenciar a troca de mensagens entre os agentes. Verifica se a mensagem está sintática (respeita regras gramaticais da linguagem ACL) e semanticamente (verifica se o contexto da mensagem é conhecido, através do nome do protocolo utilizado) correta;

- **Módulo de Conhecimentos:** armazena informações que o agente possui sobre si próprio (conhecimento individual) e sobre outros agentes (conhecimento social). Possui dentre outras informações, as competências físicas e cognitivas do agente;
- **Módulo de Avaliação:** enquanto as competências, presentes no módulo de conhecimentos, descrevem o que um agente sabe fazer, este módulo possui um detalhamento de como fazer. Esse detalhamento está presente em planos locais e protocolos de interação, que serão detalhados na seqüência do texto;
- **Módulo de Decisão:** este módulo controla a execução das ações tomadas no nível do módulo de avaliação. Isto é feito por meio de um gestor de planos;

4.1 – A Plataforma MAJORCA

A plataforma MAJORCA é composta basicamente das ferramentas JAPHA e XML2Jess. Esta plataforma foi desenvolvida pelos pesquisadores do Laboratório de Ciências da Informação e Sistemas, da Université Paul Cézanne, situada em Marseille, França.

A ferramenta JAPHA é uma ferramenta gráfica que permite a utilização de alguns formalismos, tais como autômatos, portas lógicas e o próprio RCA, para especificar o comportamento dos agentes. Esta ferramenta é baseada na linguagem de programação Java e, associada com outra ferramenta chamada XML2Jess provê um gerador semi-automático de códigos que descrevem os comportamentos dos agentes que foram definidos em RCA. A ferramenta JAPHA é inteiramente gráfica e não exige mais do que o conhecimento sobre o que os agentes devem fazer para ser utilizada.

Após a definição do comportamento do agente em RCA, o arquivo é salvo pela ferramenta JAPHA no formato XML (*eXtensible Markup Language*). Em seguida, outra ferramenta chamada XML2Jess faz o papel de tradutor e gera um código em linguagem CLIPS (*C Language Integrated Production System*) que contém a estrutura básica das

regras que serão usadas para garantir que o comportamento definido em JAPHIA será alcançado, ou seja, que contém os planos comportamentais dos agentes. Estes planos são realizados em uma plataforma para desenvolvimento de sistemas especialistas chamada JESS (*Java Expert System Shell*).

JESS é um motor de inferência que permite a execução de programas CLIPS diretamente integráveis ao código Java. Ele foi desenvolvido completamente em linguagem Java e foi inspirado originalmente no arcabouço para sistemas especialistas CLIPS [Menken, 2002], desenvolvido pela NASA na década de 80, porém existem diversas diferenças entre as duas iniciativas.

Dentre as diferenças principais que podem ser citadas entre JESS e CLIPS estão o fato que JESS foi desenvolvido completamente em Java e, por isso, possui a capacidade de lidar com objetos Java e atuar sobre eles. Outras diferenças são citadas em [Friedman-Hill, 2003] que diz que existem algumas operações que existem em ambas possuem semântica diferente e, mais do que isso, existem algumas características que são implementadas em JESS e não são em CLIPS, e vice-versa. Então, apesar de semelhantes na origem, existem alguns pontos que as diferenciam.

Existem três formas básicas de representação do conhecimento em JESS, elas são Fatos, Funções e Programação Orientada a Objetos, devido à sua natureza Java [Menken, 2002]. Então, um código pode ser gerado utilizando-se apenas um tipo de representação, ou mesmo um misto delas.

Enfim, o código gerado em JESS não contém as regras que regem o comportamento dos agentes, mas apenas a estrutura destas regras. É necessário que diversos parâmetros específicos, como conteúdo das mensagens trocadas, por exemplo, sejam inseridos manualmente.

Além destes parâmetros, todas as funções a serem avaliadas durante a execução devem ser criadas manualmente. Essas funções e alterações no código devem seguir a sintaxe Lisp (*List Processing*) e podem ser efetuadas em qualquer editor de texto.

Editores como o ConTEXT, por exemplo, trazem a vantagem de serem sensíveis às palavras reservadas e à sintaxe facilitando o trabalho de edição. No Quadro 7, é apresentado um trecho de código gerado pelo XML2Jess onde os símbolos “<” e “>” delimitam as informações que devem ser inseridas para completar as regras.

```
(assert
  (resultat
    (plan AGT1)
    (plan-id ?plan-id)
    (etat-id 3)
    (valeur
      (send-message
        <perf>
        <content>
        ?conv-id
        <enveloppe>
        <in-r-to>
        CLIPS
        <ontology>
        <protocol >
        <receiver>
        <reply-by>
        <reply-with>
        ?*myself*
      ) ; fin send-message
    )
  ); fin resultat
): fin assert resultat
```

Quadro 7 - Trecho de Código Gerado pela Ferramenta XML2Jess

1.2 – A Plataforma JADE

A etapa posterior à definição dos comportamentos dos agentes é a implementação em uma plataforma que dê suporte a este fim. A plataforma escolhida foi a JADE (*Java Agent Development Framework*), por estar em conformidade com as especificações da FIPA (*Foundation for Intelligent Physical Agents*) e por ser difundida tanto no meio acadêmico quanto no meio industrial.

JADE é um *software* livre desenvolvido pelo CSELT, da Universidade de Parma, situada na Itália, e distribuído pelo Tilab em código aberto. Ele tem por objetivo

simplificar o trabalho de desenvolvimento de sistemas multiagentes fornecendo um conjunto completo de serviços em conformidade com as especificações FIPA [Bellifemine, Poggi & Rimassa, 1999].

A plataforma inclui todos os componentes obrigatórios que controlam um sistema multiagentes. Estes componentes são:

- AMS (*Agent Management System*) – tem a função de supervisionar o registro dos agentes, sua autenticação, acesso e utilização do sistema. Apenas um MAS existe em cada plataforma;
- DF (*Directory Facilitator*) – tem a função de fornecer um serviço de páginas amarelas para a plataforma;
- ACC (*Agent Communication Channel*) – tem a função de gerir a comunicação entre os agentes. Todos os agentes, inclusive o AMS e o DF utilizam esse canal;

A representação do modelo padrão da plataforma de agentes definido pela FIPA, que é seguido pela plataforma JADE, pode ser vista na Figura 4.4.

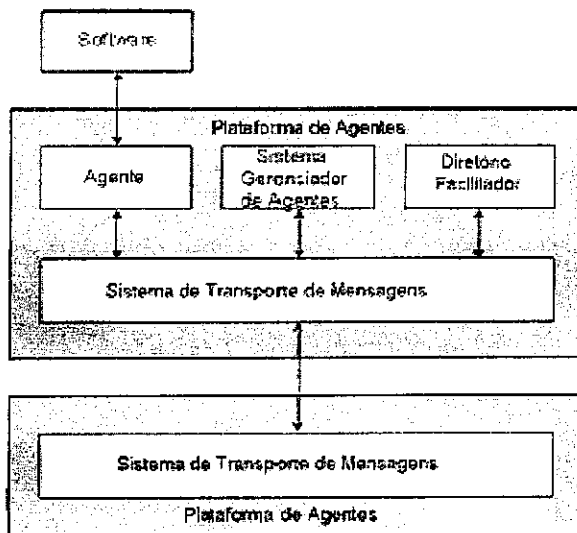


Figura 4.4 - Modelo Padrão de Plataforma de Agentes Definido pela FIPA [Silva, 2003]

A plataforma de agentes pode ser repartida entre diversos servidores. Apenas na aplicação Java e, portanto apenas uma máquina virtual Java (JVM) é executada em cada servidor. Cada JVM é um container de agentes que fornece um ambiente completo

para a execução dos agentes e permite que diversos agentes sejam executados paralelamente no mesmo servidor, como pode ser visto na Figura 4.5.

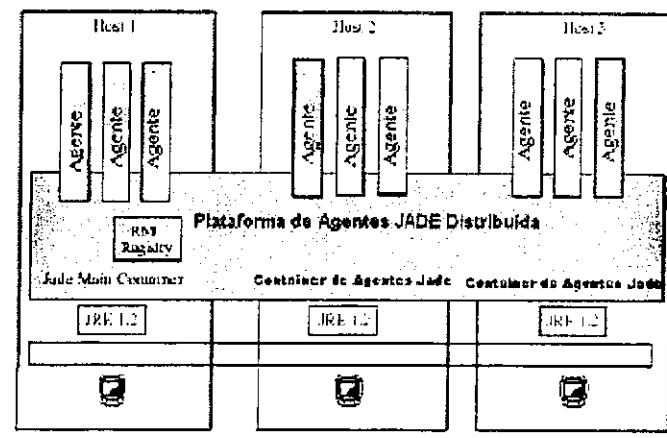


Figura 4.5 - Plataforma de Agentes JADE Distribuída em Vários Containers

Em nosso caso, os agentes utilizam a plataforma JADE, e seus comportamentos, devido à capacidade de integração simples de um software externo em uma das tarefas dos agentes, são baseados no *JessBehaviour* que usa o JESS como motor de raciocínio dos agentes. Isso faz com que JESS seja responsável pelo papel deliberativo e os comportamentos JADE sejam responsáveis pelos papéis reativos na plataforma [Bellifemine, Poggi & Rimassa, 1999].

Para executar a plataforma, é possível utilizar a ferramenta Eclipse, onde são especificados os parâmetros desejados para a execução e JADE é inicializada de acordo com eles, ou pode-se iniciá-la através de linhas de comando. Uma outra opção é executar a plataforma a partir de linhas de comando do DOS. Quando a plataforma está ativa, existe a possibilidade de executar um agente chamado RMA (*Remote Agent Management*), que é uma interface gráfica que facilita a interação do usuário com a plataforma e fornece acesso a outros agentes que podem atuar como ferramentas, como é o caso dos agentes *Sniffer* e *Dummy*.

Através do RMA é possível efetuar atividades como criar e destruir agentes, coloca-los em estado suspenso, criar clones, enviar mensagens, entre outras. Além disso, uma função interessante é a do agente *Sniffer* que permite que toda troca de

mensagens entre os agentes presentes na plataforma seja acompanhada por meio de um diagrama de seqüência de mensagens.

Então, quando o desenvolvimento atinge este ponto, o projeto está completamente realizado e já pode ser utilizado em estudos de caso, onde as regras definirão o comportamento dos agentes cognitivos que serão associadas às características reativas dos agentes existentes na plataforma de desenvolvimento. São definidos arquivos de inicialização para cada agente onde são listados todos os planos que podem ser instanciados por esse eles.

Para se ter uma visão mais completa do processo de desenvolvimento dos agentes dentro da plataforma utilizada, a Figura 4.6 ilustra essa seqüência e fornece algumas informações sobre as etapas.

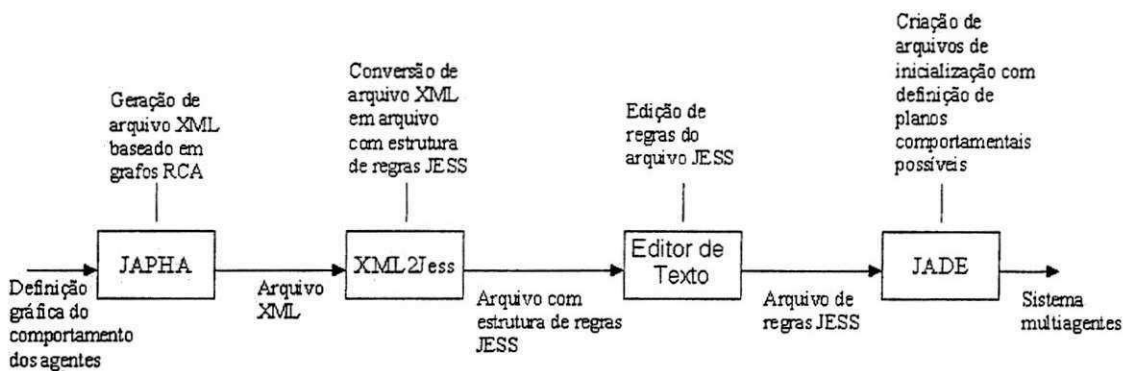


Figura 4.6 - Síntese do Processo de Desenvolvimento

Assim, após a criação do sistema multiagente deve-se proceder com a modelagem do sistema em estudo, no caso de simulações, e a integração de ambos. Neste sentido, a seção seguinte traz considerações a respeito desta integração e composição de uma ferramenta computacional para auxílio à tomada de decisões.

4.3 - Discussão sobre a Implementação

São apresentados até este ponto, no texto desta dissertação, apenas aspectos relacionados ao processo de especificação e criação dos agentes de *software* que estão presentes na camada superior (nível de decisão) da arquitetura de uma ferramenta de apoio à decisão. Para realizar a implementação, isto é, especificar o nível físico e a forma de comunicação entre estes dois níveis, algumas possibilidades são viáveis do ponto de vista prático. Alguns esforços foram feitos no sentido de viabilizar essa complementação do trabalho, porém, houve alguns fatores, tais como restrições de tempo, que impossibilitaram tal feito.

Dentre os esforços realizados, é válido mencionar a tentativa de criação de um modelo elétrico simplificado para representação e simulação do comportamento de um sistema elétrico de subtransmissão através da utilização da ferramenta computacional AnylogicTM, desenvolvida pela *XJ Technologies*. A ferramenta AnylogicTM agrega as características de interface gráfica, simulador e gerador automático de código, possibilitando, teoricamente, seu uso para modelagem de sistemas discretos, híbridos ou contínuos. Além disso, possui a funcionalidade de criação de animações interativas baseadas no modelo [Scaico, 2004].

De posse do modelo físico, implementado no AnylogicTM, e considerando que é apenas uma visão simplificada do sistema, o cálculo de fluxo de carga baseado no método de Newton-Raphson deve ser aplicado a fim de simular o comportamento do sistema e de prover uma atualização das variáveis do sistema em resposta às variações provocadas pela atuação dos agentes no processo de minimização de perturbações. A integração do modelo operacional, em AnylogicTM, com o modelo decisional, em JESS, deve ser feita através de um agente que traduz os sinais emitidos pelo nível operacional em mensagens que possam ser processadas pelos agentes cognitivos. Essas mensagens são tratadas como fatos pelo JESS e como *strings* pelo AnylogicTM.

Entretanto, durante a modelagem do nível físico, camada inferior da arquitetura da ferramenta, a utilização de derivadas parciais para implementação do cálculo de fluxo de carga baseado no método de Newton-Raphson é imprescindível e a ferramenta Anylogic™ não suporta tal operação matemática, sendo necessária a criação de uma biblioteca adicional que acrescente esta funcionalidade. Este fato associado a uma limitação de tempo para conclusão da pesquisa fez com que a utilização de outra ferramenta fosse escolhida em detrimento ao Anylogic™.

Outra possibilidade para implementação é a utilização da ferramenta Matlab® para modelar o sistema elétrico. Apesar da mesma não possibilitar a criação de animações para visualização do sistema, ela possui o suporte matemático necessário para a implementação de um modelo do comportamento do sistema. No caso de utilizar-se apenas cálculo de fluxo de carga para representar a dinâmica do sistema, uma opção é a utilização do pacote MATPOWER, desenvolvido pelo PSERC (Power Systems Engineering Research Center) da Cornell University, que contém um conjunto de arquivos .m para resolução de problemas de fluxo de carga e fluxo de carga ótimo [MATPOWER, 2007].

Uma vez que o pacote MATPOWER utiliza arquivos de texto como entradas, a manipulação dos dados deveria ser feita através de leitura e escrita em arquivos. Neste caso de utilização do Matlab® para implementação do modelo operacional do sistema, duas estratégias que poderiam ser utilizadas para efetuar a integração entre as camadas da ferramenta são o uso de *sockets* e o uso da JMatLink, que é uma API que permite utilizar os recursos do Matlab® dentro de aplicações java, *applets* e *servlets* desenvolvidos por usuários finais [JMATLINK, 2005].

A utilização da ferramenta Matlab® para implementação da dinâmica do sistema traz benefícios no sentido de possibilitar a verificação da estratégia descentralizada descrita no Capítulo 3. Entretanto, para que a estratégia possa ser validada, é necessário que o mesmo sistema seja implementado e avaliado com técnicas já consagradas na literatura e, em seguida, que os resultados obtidos sejam comparados a partir de alguns parâmetros bem definidos.

Capítulo 5

Considerações Finais

No decorrer deste trabalho, foram apresentadas uma estratégia descentralizada para auxílio à tomada de decisões e a modelagem de um sistema multiagentes capaz de implementar esta estratégia. Após o desenvolvimento e a instanciação a um contexto de aplicação com fortes características de centralização, acredita-se que o emprego desta estratégia pode trazer benefícios quando utilizada em ambientes industriais, no sentido de minimizar o tempo de solução de problemas decorrentes de falhas operacionais e, portanto, reduzir os custos decorrentes de faltas originadas por estas falhas.

O objetivo principal no desenvolvimento deste trabalho foi propor uma estratégia descentralizada que torne mais ágil a realização de operações que tipicamente envolvem uma grande carga de comunicação direta entre centro de controle e as unidades componentes do sistema. A verificação de qual estratégia é mais apropriada para ambientes industriais decorrente da comparação dos resultados de ambas fornecerá subsídios para a oferta de um recurso computacional ao operador dos sistemas para apoiá-lo na resolução de problemas de operação.

O refinamento da estratégia original, proposta em [Turnell, Espinasse & Aventini, 2005], consistiu na especificação do comportamento apresentado pelos agentes de *software*, que são responsáveis pelas tomadas de decisão diante de uma contingência, e na revisão da divisão e do estabelecimento das fases da estratégia original.

É importante destacar que apesar da estratégia ter sido apresentada de forma instanciada para sistemas de potência, alguns aspectos de sua dinâmica são dependentes

da aplicação e do grau de fidelidade do modelo do sistema utilizado. Modelos mais complexos, que levam em consideração diversos aspectos operacionais do sistema, podem exigir a adaptação da estratégia a fim de cobrir especificidades.

No que se refere à arquitetura dos agentes definida, sua modularidade propicia uma separação lógica para tratamento de aspectos distintos dos agentes (comunicação, decisão, avaliação e conhecimentos) e permite que haja uma comunicação entre os módulos os quais se tornam interdependentes.

Após a definição e implementação da arquitetura de agentes foi possível verificar que do ponto de vista conceitual a estratégia é adequada aos propósitos de apoiar a supervisão de sistemas industriais em geral, através de sua instanciação a um contexto de aplicação industrial. No entanto, ainda é necessário avaliar sua eficácia a partir de testes que integrem as camadas física e de decisão da ferramenta e a apliquem a um estudo de caso. A aplicação da estratégia a situações reais ainda demanda ajustes, pois, o modelo do sistema elétrico utilizado foi uma visão extremamente simplificada e poucas restrições operacionais prática foram levadas em consideração.

A validação desta ferramenta consistirá da comparação de seus resultados com outros obtidos a partir de outras técnicas já consagradas na literatura científica ou na indústria como, por exemplo, algoritmos genéticos e lógica *fuzzy*, entre outros.

No Capítulo 4, mais especificamente na Seção 4.3, são apresentadas algumas possibilidades de implementação da estratégia proposta. Outras possibilidades, certamente, ainda não foram exploradas. A avaliação destas possibilidades pode constituir o foco de um trabalho futuro.

Enfim, a proposta de descentralização da tomada de decisões em sistemas industriais, críticos ou não, oferece a possibilidade de minimização de perdas quando da ocorrência de falhas operacionais nestes sistemas. Com a continuidade deste trabalho pretende-se explorar essa possibilidade.

5.1 – Trabalhos Futuros

Dentre os trabalhos propostos se encontram o estudo e a avaliação de modelos mais apropriados do sistema adotado, a exemplo de um contexto de aplicação com posterior integração à camada de decisão, visando maior fidelidade dos resultados e, conseqüentemente, minimizando a probabilidade de ações desnecessárias ou inapropriadas durante a operação.

Especificamente para sistemas elétricos, a utilização de modelos que levem em consideração restrições operacionais como limites de tensão e surtos de chaveamento, por exemplo, podem fornecer subsídios para justificar a aplicação da estratégia em um sistema real. Uma possibilidade seria a integração com ferramentas típicas do âmbito de sistemas de potência como a ATP (*Alternative Transients Program*), por exemplo.

Outro aspecto a ser melhorado na ferramenta é o fornecimento ao operador da sugestão de solução acompanhada da descrição do procedimento para implantar a solução. Isto pode ser alcançado através da implementação de um agente cuja função é registrar os passos necessários para obtenção da solução em um arquivo. Este *log* do sistema facilitaria a consulta ao procedimento adotado para se chegar à solução e auxiliaria no caso de auditorias.

Apesar de ainda não ser possível apresentar resultados conclusivos sobre o impacto que a utilização desta estratégia pode causar em sistemas com estrutura centralizada, é possível antever resultados potencialmente interessantes do ponto de vista de execução de tarefas de restauração em processos industriais.

6 - Referências Bibliográficas

ADLA, A.; ZARATE, P. **A cooperative intelligent decision support system**. In IEEE International Conference on Service Systems and Service Management, pp. 763-769, 2006.

AOKI, A. R.; LAMBERT-TORRES, G.; MORAES, C. H. V.; COSTA, B. R.; BARBOSA, J. A. **Sistema multi-agente para o atendimento de ocorrências na rede de distribuição de energia elétrica**. In Simpósio Brasileiro de Automação Inteligente (SBAI), pp. 1049-1054, 2003.

AVENTINI, G. **Un Système d'aide à la supervision de réseau de distribution électrique à base d'agents**. Étude Bibliographique, March 2004.

BELLIFEMINE, F.; POGGI, A.; RIMASSA, G. **JADE – A FIPA-compliant agent framework**. In PAAM'99, pp. 97-108, London, UK, 1999.

BITTENCOURT, G. **Introdução aos sistemas multiagentes: Histórico e fundamentos**. Disponível em <http://www.das.ufsc.br/~gb/pg-ic/tra-hist.pdf>. Acessado em 29/01/2007, 2005.

CAMPITELLI, E. M.; DA SILVA, L. G. W.; PEREIRA, R. A. F.; MANTOVANI, J. R. S. **Sistema inteligente para alocação e coordenação de proteção contra sobrecorrentes em redes radiais de distribuição de energia elétrica**. In Congresso Brasileiro de Automática, pp- 821-826, 2006.

CANÔAS, A. C. G.; MURARI, C. A. F. **Uma metodologia para controle de tensão e de potência reativa utilizando regras nebulosas**. In Congresso Brasileiro de Automática, pp. 2141-2146, 2006.

CARVALHO, B. P. R.; CARVALHO, D. H. D.; YOVANOVIC, A. P. **Sistema inteligente para predição do comprimento de eletrodos em um forno elétrico a arco submerso**. In Congresso Brasileiro de Automática, pp. 953-958, 2006.

CECCARONI, L.; CORTÉS, U.; SÀNCHEZ-MARRÉ, M. **OntoWEDSS: augmenting environmental decision-support systems with ontologies**. In Environmental Modelling & Software, Vol. 19, Issue 9, pp. 785-797, 2004.

CHIU, C. K.; CHOU, C. M. LEE, J. H.; LEUNG, H. F.; LEUNG, Y. W. **A constraint-based interactive train rescheduling tool**. In Constraints, Vol.7, No. 2, pp. 167-198, 2002.

ĆURČIĆ, S.; ÖZVEREN, C. S.; CROWE, L.; LO, P. K. L. **Electric power distribution network restoration: a survey of papers and a review of the restoration problem**. In Electric Power Systems Research 35, pp. 73-86, 1996.

DEMAZEAU, Y. **Distributed AI and multi-agent systems**. Lecture Notes, Lindö Center for Applied Mathematics, University of Odense, Denmark, 1994.

DEY, P. K. **Decision Support System for Inspection and Maintenance: A Case Study of Oil Pipelines**. In IEEE Transactions On Engineering Management, Vol. 51, No. 1, pp. 47-56, 2004.

DUARTE, A. N.; SAUVÉ, J. P.; CIRNE, W.; FIGUEIREDO, J. C. A.; SAMPAIO, M. C.; NETO, E. R.; DA SILVA, M. A. H.; ARAÚJO, A. S.; MELO, M. S. C. **Tratamento de eventos em rededs elétrticas: uma ferramenta**. In Anais do Congresso Brasileiro de Automática, 2004.

DURFEE, E. H.; ROSENSCHEIN, J. S. **Distributed problem solving and multi-agent systems: comparisons and examples**. In Proceedings of the Thirteenth International Distributed Artificial Intelligence Workshop, pp. 94-104, 1994.

FARINELLI, A.; GRISSETTI, G.; IOCCHI, L.; LO CASCIO, S.; NARDI, D. **Design and evaluation of multi agent systems for rescue operations**. In Proceedings of IEEE International Conference on Intelligent Robots and Systems, pp. 3138-3143, 2003.

FININ, T.; LABROU, Y.; MAYFIELD, J. **KQML as an agent communication language**. In Software Agents, J. M. Bradshaw ed., pages 291-319. AAAI Press/The MIT Press, 1997.

FIPA. **Statement of Intent**. <http://www.fipa.org/about/mission.html>. Acessado em 11/01/2007, 2003.

FRANKLIN S.; GRAESSER, A. **Is it an agent, or just a program?: A taxonomy for autonomous agents**. In Proceedings of the Third International Workshop on Agent Theories, pp. 21-36, Springer-Verlag, 1996.

FRIEDMAN-HILL, E. **Jess in Action – Rule-Based Systems in Java**. Manning, Greenwich, 2003.

FUKUYAMA, Y.; ENDO, H.; NAKANISHI, Y. **A hybrid system for service restoration using expert system and genetic algorithm**. Proceedings of IEEE Intelligent Systems Applications to Power Systems, pp. 394-398, 1996.

GASSER, L. **Distribution and coordination of tasks among intelligent agents**. In Proceedings of the JCAI'88. Scandinavian Conference on AI. Amsterdam: Springfield, PP.189-204, 1988.

GUERRA, F. A.; TOVAR, F. T. R.; RAMOS, M. P.; CORRÊA, L. A. D.; COELHO, L. S.; ANDRADE, C.; BAPTISTA, W. **Sistema especialista para monitoramento e controle de processos: a experiência do projeto Injex**. Simpósio Brasileiro de Automação Inteligente (SBAI), pp. 450-455, 2003.

HEGYI, A.; DE SCHUTTER, B.; HOOGENDOORNT, S.; BABUSKA, R.; VAN ZUYLENT, H.; SCHUURMANT, H. **A Fuzzy Decision Support System for Traffic Control Centers**. In IEEE Intelligent Transportation Systems Conference Proceedings, pp. 358-363, 2001.

JMATLINK. **JMatLink**. <http://www.held-mueller.de/JMatLink/>. Acessado em 28/02/2007, 2005.

JENNINGS, N. R.; WOOLDRIDGE M. **Agent-Oriented Software Engineering**. Proceeding of the 9th European Workshop on Modeling Autonomous Agents in a Multi-Agent World, pp. 1-7, 1999.

JUNG, H.; TAMBE, M.; BARRET, A.; CLEMENT, B. **Enabling efficient conflict resolution in multiple spacecraft missions via DCSP**. In Proceedings of the NASA workshop on planning and scheduling, 2002.

KOUMBOULIS, F. N.; TZAMTZI, M. P.; PAVLOVIC, M. **Decision support systems in agribusiness**. In IEEE 3rd International Conference on Mechatronics, pp. 457-461, 2006.

KOUTSOUKIS, N. S.; DOMINGUEZ-BALLESTEROS, B.; LUCAS, C. A.; MITRA, G. **A Prototype Decision Support System for Strategic Planning under Uncertainty**. In International Journal of Physical Distribution & Logistics Management, Vol. 30, pp. 640-661, 2000

LABARTHE, O. **Modélisation et Simulation Orientées Agents de Chaînes Logistiques dans un Contexte de Personnalisation de Masse : Modèles et Cadre Méthodologique**, Ph.D. Thesis, University of Aix-Marseille, 2006.

LOPES JR., C. R.; AOKI, A. R.; ESMIN, A. A. A.; LAMBERT-TORRES, G. **Multi-Agent model for power substation restoration**. In IASTED International Conference on Power and Energy Systems, Tampa, pp. 26-32, FL, 2001.

MATPOWER. MATPOWER Home Page. <http://www.pserc.cornell.edu/matpower/>. Acessado em 14/03/2007, 2007.

MCARTHUR, S. D. J.; BOOTH, C. D.; MCDONALD, J. R.; MCFADYEN, I. T. **An agent-based anomaly detection architecture for condition monitoring**. In IEEE Transactions on Power Systems, Vol. 20, No. 4, pp. 1675- 1682, 2005.

MENKEN, M. **Jess Tutorial**. Vrije Universiteit, Amsterdam, Netherlands, December, 2002.

MOKHATI, F.; BADRI, M.; BADRI, L. **A formal framework supporting the specification of the interactions between agents**. In Informatica, Vol. 30, Issue 1, pp. 97-110, 2006.

MOTA, A. A.; MOTA, L. T. M.; FRANÇA, A. L. M. **Metodologia de recomposição coordenada de sistemas elétricos com base em técnicas de lógica nebulosa e busca heurística**. In Anais do Congresso Brasileiro de Automática, 2004.

MOYAU, T. **Specification de comportements d'agents dans des systemes multi-agents**. DEA de Productique et Informatique, University of Aix-Marseille III, 2000.

NAGATA, T.; HATAKEYAMA, S.; YASUOKA, M.; SASAKI, H. **An efficient method for power distribution system restoration based on mathematical programming and operation strategy**. In Proceedings of the IEEE International Conference on Power System Technology, pp. 1545-1550, Vol. 3, 2000.

NAREYEK, A. **Intelligent agents for computer games**. In Proceedings of the Second International Conference on Computers and Games, pp. 414-422, 2000.

NICHOLSON, D.K.; GUNN, B.M.; NAHAVANDI, S. **A decision support tool for resource allocation in batch manufacturing**, In IEEE International Conference on Systems, Man and Cybernetics, Vol. 4, pp. 3018-3023, 2005.

NIU, C.; YU, X.; LI, J.; SUN, W. **The application of operation optimization decision support system based on data mining in power plant.** In Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, pp. 1830-1834, 2005.

PAUROBALLY, S.; CUNNINGHAM, J.; JENNINGS, N. R. **Developing agent interaction protocols using graphical and logical methodologies.** In Proceedings of 1st Workshop on Programming Multi-Agent Systems, 149-168, 2003.

RAMNARAYAN, P.; BRITTO, J. **Paediatric clinical decision support systems.** In Archives of Disease in Childhood, 87, pp. 361-362, 2002.

RIBEIRO, A. M.; BIONDI NETO, L.; COELHO, P. H. G.; CHIGANER, L.; MEZA, L. A.; COUTINHO, L. H. S. A. **Sistema de apoio à decisão para determinação do preço da energia, usando-se lógica nebulosa.** In ENGEVISTA, Vol. 6, No. 2, pp. 4-14, 2004.

RICCI, F. **Travel recommender systems.** In IEEE Intelligent Systems, pp. 55-57, 2002.

RUSSEL, S. J.; NORVIG, P. **Artificial intelligence: A modern approach.** 2nd edition, Prentice Hall, 2002.

SCAICO, A. **Métodos Formais no Projeto de Interfaces para Sistemas Industriais Críticos.** Proposta de Tese, UFCG, 2004.

SERAPIÃO, A. B. S.; GUILHERME, I. R.; MOROOKA, C. K.; MENDES, J. R. P.; FRANCO, K. P. M. **Um sistema inteligente para escolha dentre alternativas de sistemas marítimos de produção.** Simpósio Brasileiro de Automação Inteligente (SBAI), pp. 272-277, 2003.

SHEREMETOV, L. B.; CONTRERAS, M.; VALENCIA, C. **Intelligent multi-agent support for contingency management system.** In Expert Systems with Applications, Vol. 26, Issue 1, pp. 57-71, 2004.

SIEBRA, C. A. Uma arquitetura para suporte de atores sintéticos em ambientes virtuais – Uma aplicação em jogos de estratégia. Dissertação de Mestrado, UFPE, Recife, 2000.

SILVA, L. A. M. Estudo e desenvolvimento de sistemas multiagentes usando JADE: Java Agent Development Framework. Monografia de Conclusão de Curso, Universidade de Fortaleza, Fortaleza, 2003.

TALUKDAR, S. The equitable distribution of the benefits from decentralization: a challenge for power system designers. In IEEE Power Engineering Society Summer Meeting, Vol. 3, pp. 1691-1692, 2002.

TEIXEIRA JR., R. F.; FERNANDES, F. C. F.; PEREIRA, N. A. Sistema de apoio à decisão para programação da produção em fundições de mercado. In Gestão & Produção, Vol. 13, No. 2, pp. 205 - 221, 2006.

TRANVOUEZ, E. IAD et ordonnancement: une approche coopérative du reordonnancement d'ateliers, Ph.D. Thesis, University of Aix-Marseille, 2001.

TSANG, E.; YUNG, P.; LI, J. EDDIE-automation, a decision support tool for financial forecasting. In Decision Support Systems, Vol. 37, Issue 4, pp. 559-565, 2004.

TURNELL, M. F. Q. V.; ESPINASSE, B.; AVENTINI, G. Supply restoration in electric distribution networks: a multi-agent approach. In: I3M-Conceptual Modeling and Simulation Conference, pp. 129-138, October 2005.

VOLPE, R.; ESTLIN, T.; LAUBACH S.; OLSON, C.; BALARAM, J. Enhanced mars rover navigation techniques. In Proceedings of the 2000 IEEE International Conference in Robotics and Automation, pp. 926-931, San Francisco, CA, 2000.

WEISS, G. Multiagents systems: A modern approach to distributed artificial intelligence. MIT Press, Cambridge, 1999.

WEN, W.; WANG, W. K.; WANG, C. H. **A knowledge-based intelligent decision support system for national defense budget planning.** In Expert Systems with Applications, Vol. 28, Issue 1, pp. 55-66, 2005.

WEST, G. M.; STRACHAN, S. M.; MOYES, A.; MCDONALD, J. R. **Intelligent decision support for protection scheme design associated with electrical transmission systems.** In 22nd IEEE Power Engineering Society International Conference on Innovative Computing for Power, pp. 128-133, 2001.

WOOLDRIDGE, M. J.; JENNINGS, N. R. **Agent theories, architectures, and languages.** In Intelligent Agents, Springer Verlag, pp. 1-22, 1995a.

WOOLDRIDGE, M. J.; JENNINGS, N. R. **Intelligent agents: Theory and practice.** In Knowledge Engineering Review, pp. 115-152, Vol. 10, No. 2, Cambridge University Press, 1995b.

XIAO, J.; ZHANG, Y.; WANG, C. **A study and implementation of an intelligent load forecast support system.** In Proceedings of IEEE International Conference on Power System Technology, pp. 858-862, 2002.

ZHANG, W.; DIETTERICH, T. G. **A reinforcement learning approach to job-shop scheduling.** In Proceedings of the International Joint Conference on Artificial Intelligence, pp. 1114-1120, 1995.