

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ  
CENTRO DE CIÊNCIAS BIOLÓGICAS E DA SAÚDE

DENNYS ANDREI MACHADO

Uma contribuição à concepção e modelagem de  
sistemas de agentes para o atendimento pré-hospitalar

DISSERTAÇÃO DE MESTRADO

PROGRAMA DE PÓS-GRADUAÇÃO

EM TECNOLOGIA DA SAÚDE

CURITIBA

2005

DENNYS ANDREI MACHADO

Uma contribuição à concepção e modelagem de  
sistemas de agentes para o atendimento pré-hospitalar

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Tecnologia da Saúde área de concentração de Sistemas de Informação da Pontifícia Universidade do Paraná como pré-requisito para obtenção do título de Mestre.

Orientador: Prof. Dr. Edson E. Scalabrin  
Co-orientador: Prof. Dr. João da Silva Dias

CURITIBA

2005

# FOLHA DE APROVAÇÃO

Dennys Andrei Machado.

Uma contribuição à concepção e modelagem de sistemas de agentes para o atendimento pré-hospitalar.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Tecnologia da Saúde da Pontifícia Universidade Católica do Paraná como pré-requisito para obtenção do título de Mestre.

Aprovado em 20 de abril de 2005.

Banca examinadora:

Prof. Dr. \_\_\_\_\_ Instituição: \_\_\_\_\_

Julgamento: \_\_\_\_\_ Assinatura: \_\_\_\_\_

Prof. Dr. \_\_\_\_\_ Instituição: \_\_\_\_\_

Julgamento: \_\_\_\_\_ Assinatura: \_\_\_\_\_

Prof. Dr. \_\_\_\_\_ Instituição: \_\_\_\_\_

Julgamento: \_\_\_\_\_ Assinatura: \_\_\_\_\_

Algo só é impossível até que alguém duvide e acabe  
provando o contrário!

Albert Einstein (1879 - 1955)

A Deus,  
Aos meus Pais,  
A minha amada esposa.

## Agradecimentos

Se as pessoas tivessem idéia do quanto um gesto, uma atitude ou mesmo uma simples palavra de apoio faz diferença na concretização de uma dissertação de mestrado, certamente estas ficariam honradas em serem lembradas neste momento.

Durante o período de realização deste estudo inúmeras pessoas me incentivaram, amigos, familiares, colegas, alunos, orientadores, professores e inclusive pessoas que acabara de conhecer. Estas pessoas reconhecem a importância de uma dissertação e trazem forças para a realização desta etapa. Porém, existem pessoas que agüentam constantemente, mal humor, irritações, alegrias por acabar um pedaço de texto de forma satisfatória, comemoração por um resultado esperado alcançado, decepção por um teste frustrado, alegria pela descoberta de uma nova solução, e depois de tudo isto ainda tem a incumbência de ler um texto e dizer que está bom ou ruim, ou de escutar todos os “sonhos” e ambições colocadas em um estudo sem parar de prestar a atenção um só minuto, ou ainda ajudar com seu conhecimento técnico de sua área, afinal este é um estudo multi-disciplinar.

Estas pessoas que participaram destes momentos são os pilares da minha vida, pessoas que estão comigo o tempo todo e que realmente me fazem crescer, pessoas que acreditam em mim e pelas quais vou buscar nunca decepcionar.

Muito obrigado orientadores, pai, mãe, sogro, sogra, irmã e principalmente minha esposa, que neste período me acompanhou, entendeu e me apoiou tornando-se vital a conclusão deste trabalho.

# Sumário

<b>Lista de Figuras .....</b>	<b>IX</b>
<b>Resumo .....</b>	<b>XI</b>
<b>Abstract .....</b>	<b>XII</b>
<b>Capítulo 1 - Introdução.....</b>	<b>1</b>
1.1    Objetivos .....	4
1.1.1    Objetivo Geral .....	4
1.1.2    Objetivos Específicos .....	4
1.2    Proposta.....	4
1.3    Estrutura do Documento.....	5
<b>Capítulo 2 – Revisão Bibliográfica .....</b>	<b>7</b>
2.1    Atendimento pré-hospitalar .....	7
2.2    Trajetória dos sistemas para o atendimento pré-hospitalar na PUCPR .....	10
2.2.1    Gerenciamento de Emergências Médicas.....	10
2.2.2    Sistema de Informação e Comunicação no Atendimento Pré-Hospitalar.....	10
2.2.3    Integração de sistemas na área de saúde.....	11
2.2.4    Uma arquitetura multi-agente para o atendimento pré-hospitalar .....	12
<b>Capítulo 3 – Agentes de Software .....</b>	<b>13</b>
3.1    Características de um agente .....	14
3.2    Sistemas abertos .....	16
3.3    Sistemas multi-agente .....	17
3.4    Organização de um sistema multi-agente.....	18
3.4.1    Modelo de agente reativo e cognitivo .....	21
3.4.2    Arquitetura Reativa .....	22
3.4.3    Arquitetura Cognitiva.....	23
3.4.4    Tipos de organização em um sistema multi-agente .....	24
3.5    Modelo de agente .....	26
3.5.1    Representação conceitual de um agente .....	26
3.5.2    Interação via troca de mensagens .....	29
3.5.3    Componentes do modelo de agente.....	33
3.6    Linguagem de comunicação agente .....	34
3.7    Redes Contratuais.....	38
<b>Capítulo 4 – Linguagem de modelagem de agentes .....</b>	<b>40</b>
4.1    Introdução .....	40
4.2    Metodologias.....	41
4.2.1    UML.....	42
4.2.2    AUML.....	44
4.3    Utilização de Padrões – Nível 1 .....	46
4.3.1    Pacotes .....	46
4.3.2 <i>Templates</i> .....	47
4.4    Interações entre Agentes – Nível 2.....	48
4.4.1    Diagramas de Seqüência .....	49
4.4.2    Diagramas de Colaboração.....	51
4.5    Representação de processos internos de agentes – Nível 3 .....	55
4.5.1    Diagramas de Atividade .....	55
4.5.2    Diagramas de estado.....	56
<b>Capítulo 5 - Modelagem do sistema SAAP para o atendimento pré-hospitalar.....</b>	<b>57</b>
5.1    Atendimento pré-hospitalar .....	57
5.2    Modelagem do atendimento pré-hospitalar usando AUML .....	64
5.3    Arquitetura e organização da comunicação entre agentes.....	74
5.3.1    Arquitetura e organização do sistema SAAP. ....	77
5.3.2    Arquitetura e organização do agente <i>Central de regulação</i> .....	80
5.3.3    Implementação da arquitetura SAAP .....	82
<b>Capítulo 6 – Resultados e Discussão.....</b>	<b>86</b>
6.1    Relato de Acidente de Trânsito .....	94

6.2 Comentários .....	97
<b>Capítulo 7 - Conclusão .....</b>	<b>98</b>
7.1 Trabalhos Futuros.....	100
7.1.1 Web Service multi-Agente para integração de sistemas legados no SUS .....	100
7.1.2 Utilização de <i>guidelines</i> para auxílio no atendimento pré-hospitalar.....	101
<b>Referências Bibliográficas .....</b>	<b>102</b>



## Lista de Figuras

Figura 2: Modelo de agente reativo .....	22
Figura 3: Modelo de agente cognitivo.....	23
Figura 4: Visão centrada nos agentes e visão centrada na organização.....	24
Figura 5: Representação conceitual de um agente definido por Ferber [Ferber, 1995]....	27
Figura 6: Modelo de agente definido por Bussmann e Demazeau [Bussman & Demazeau, 1994].....	28
Figura 7: Modelo de agente definido por Hayes-Roth [Hayes-Roth, 1990]. .....	28
Figura 8: Modelo de agente ARCHON definido por Jennings e Wittig [Jennings & Wittig, 1992].....	30
Figura 10: Modelo de agente definido por Scalabrin et al. [Scalabrin et al, 1996]. .....	32
Figura 11: Modelo de agente utilizado.....	33
Figura 12: Elementos para a comunicação entre agentes. ....	36
Figura 13: Camadas da linguagem de comunicação KQML.....	37
Figura: 14 Protocolo de Interação de Agente, mostrando as interações para o protocolo Contract Net.....	45
Figura 17: Usando pacotes para expressar protocolos.....	47
Figura 18: Instância do modelo apresentado na figura 13 .....	48
Figura 19: Identificação de um novo pacote .....	48
Figura 20: Formato básico para comunicação de agente. ....	49
Figura 21: Transmissão simultânea de mensagens.....	50
Figura 22: Múltiplas técnicas para representar comunicação simultânea entre agentes, utilizando múltiplos papéis.....	51
Figura 23: Um exemplo de um diagrama de colaboração mostrando uma interação entre agentes com papéis múltiplos. ....	52
Figura 24: Uma versão do diagrama de seqüência de figura 23.....	53
Figura 25: Um diagrama de atividade que mostra um protocolo de venda com interação de vários agentes.....	54
Figura 26: Um diagrama de estados indicando os estados e transições de Pedido .....	55
Figura 27: Um diagrama de atividade que especifica e ordena o comportamento do processo para um agente <i>Unidade móvel</i> . ....	56
Figura 28. Diagrama de estados que especifica processo interno do agente <i>unidade móvel</i> . ....	56
Figura 29: Diagrama de Caso de Uso das atividades da <i>Central de regulação</i> .....	62

Figura 30: Diagrama de Caso de Uso das atividades da <i>Unidade móvel</i> .....	63
Figura 31: Diagrama de Caso de Uso das atividades dos Hospitais/Clínicas .....	63
Figura 32: Diagrama de Caso de Uso das atividades dos Serviços de Apoio.....	64
Figura 33: Diagrama de seqüência do template Contract Net Protocol.....	66
Figura 34: Diagrama de seqüência que ilustra algumas conexões entre agentes, utilizando o pacote Contract Net definido. ....	67
Figura 35: Diagrama de seqüência que ilustra algumas conexões entre agentes na realização de um atendimento pré-hospitalar.....	69
Figura 36: Diagrama de colaboração com algumas situações do atendimento pré- hospitalar .....	71
Figura 37: Diagrama de atividades com algumas atribuições do atendimento pré- hospitalar .....	72
Figura 38: Diagrama de estados de um hospital .....	73
Figura 39: Diagrama de estados da <i>central de regulação</i> .....	73
Figura 40: Diagrama de estados da <i>unidade móvel</i> .....	74
Figura 41: Modelo de Agente .....	78
Figura 42: Estrutura básica da rede do atendimento pré-hospitalar.....	79
Figura 43: Estrutura do Agente <i>Central de regulação</i> .....	81
Figura 44: Camadas envolvidas na troca dmensagens via <i>web service</i> .....	87
Figura 45: Interface do protótipo de comunicação da <i>central de regulação</i> .....	88
Figura 46: Interface do protótipo de configuração de um agente .....	89
Figura 47: Implementação da classe de comunicação.....	90
Figura 48: Console de visualização das mensagens trocadas pela rede de atendimento pré-hospitalar.....	91

## Resumo

Este trabalho propõe o uso de técnicas clássicas da engenharia de software orientada a objetos para a modelagem de sistemas de agentes cognitivos independentes, bem como mostrar a relevância de uma arquitetura distribuída e aberta, baseada no modelo agente, no atendimento pré-hospitalar. Neste contexto, a escolha de tal arquitetura baseou-se nos seguintes critérios: a) trata-se de um problema cuja decomposição em subproblemas é natural, à medida que cada subproblema (Central Regulação, Unidade Móvel, etc) pode ser representado e implementado como um agente autônomo independente; b) a interação entre os agentes pode tornar o sistema de atendimento pré-hospitalar mais confiável, à medida que as trocas de informações entre os agentes podem resolver situações difíceis pelo número de especialistas e recursos exigidos de forma dinâmica (ou sobre demanda); c) o paralelismo natural dos sistemas multi-agente podem contribuir de forma considerável na implementação de um sistema de alta performance e disponibilidade. Deve-se enfatizar que tais critérios, no contexto do atendimento pré-hospitalar, facilitam a decomposição do problema em entidades fracamente acopladas e fortemente coesas que são os princípios básicos para se construir sistemas eficientes e mais fáceis de mantê-los. Finalmente, a proposta envolve um exame sobre o conceito de agente, um estudo sobre linguagem UML para a modelagem de sistemas multi-agente, assim como a demonstração da aplicação das técnicas de engenharia de software orientado a objetos na modelagem de um sistema de agentes. Unidas estas tarefas definiram a implementação de uma arquitetura de agentes, a aplicação de um modelo de agente, a concretização na utilização de web service para comunicação de agentes e a construção de protótipos de agentes que testaram o desenvolvimento realizado.

**Palavras-Chave:** Atendimento pré-hospitalar, agente, integração, sistema de agente.

# Abstract

This work considers the use of classic techniques of the guided engineering of software the objects for the modeling of systems of independent cognitivos agents, as well as showing the relevance of a distributed and opened architecture, based in the agent model, in the daily first aid attendance. On this context, the choice of such architecture was based on the following criteria: a) one is about a problem whose decomposition in sub problems is natural, to the measure that each sub problem (Central Regulation, Ambulance, etc) can be represented and be implemented as an independent independent agent; b) the interaction enters the agents can become the system of more trustworthy daily first aid attendance, to the measure that the exchanges of information between the agents can decide difficult situations for the number of specialists and demanded resources of dynamic form (or on demand); c) the natural parallelism of the systems multi-agent can contribute of considerable form in the implementation of a system of high performance and availability. It must be emphasized that such criteria, in the context of the daily first aid attendance, facilitate the decomposition of the problem in entities weakly connected and strong coesas that are the basic principles to construct efficient and more easy systems to keep them. Finally, the proposal involves an examination on the agent concept, a study on language UML for the modeling of systems multi-agent, as well as the demonstration of the application of the techniques of engineering of guided software the objects in the modeling of a system of agents. Joined these tasks they had defined the implementation of an architecture of agents, the application of an agent model, the concretion in the use of web service for communication of agents and the construction of archetypes of agents who had tested the development carried through.

**Palavras-Chave:** first aid attendance, agent, integration, multi-agent.

# Capítulo 1 - Introdução

A informática aplicada na área de saúde busca através de técnicas computacionais contribuir para o desenvolvimento dos diversos segmentos da saúde. O atendimento pré-hospitalar é um segmento multidisciplinar que envolve entidades de naturezas distintas como: hospital, polícia, central de regulação, unidade móvel, entre outras; com objetivo de prestar atendimento de urgência e emergência médica às vítimas clínicas e de traumas [Machado, 2000].

O atendimento pré-hospitalar existe devido à necessidade de se prover um atendimento de urgência/emergência sem a necessidade de remoção da vítima. Este tipo de serviço vem assumindo papel importante na sociedade devido ao considerável aumento de usuários [Brasil, 2002b].

O serviço de atendimento pré-hospitalar é viabilizado pela implantação de centrais de regulação que visam fornecer um meio de comunicação de serviços de urgência e emergência para a população, além de organizar e regular os atendimentos realizados. Como as centrais de regulação estarão atendendo às necessidades imediatas da população, a atuação em um atendimento acontece sempre em situações de alto stress e onde são exigidas respostas rápidas. Um serviço de atendimento pré-hospitalar deve possuir integração com outras organizações. Com isso, este serviço deve ser capaz de encaminhar uma vítima para o local mais adequado à continuidade de seu tratamento, de realizar procedimentos de alta complexidade (se necessário), consultas especializadas, exames complementares, internações e atendimentos domiciliares, consulta na rede básica de saúde, informações de outros serviços de organizações como polícia, bombeiros, etc. É previsto na Portaria 2048 [Brasil, 2002a] que a central de regulação de um serviço de atendimento deve ser conduzida e orientada pela presença de um médico regulador. A estes profissionais devem ser oferecidos os meios e recursos necessários à sua atuação, além de estabelecer previamente as responsabilidades hierárquicas, afim de conceder-lhe poder de decisão.

As competências técnicas do médico regulador são sintetizadas em sua capacidade de julgar, definindo prioridades de atendimento. Este julgamento deve ser realizado com base nas informações disponíveis sobre o atendimento. O médico ainda pode contar com o auxílio das demais entidades que prestam assistência ao atendimento pré-hospitalar, visando

possibilitar o melhor atendimento às necessidades das vítimas. De acordo com a Portaria 2048 o médico regulador é responsável por:

*“- julgar e decidir sobre a gravidade de um caso que lhe está sendo comunicado por rádio ou telefone, estabelecendo uma gravidade presumida; “*

*“- enviar os recursos necessários ao atendimento, considerando necessidades e ofertas disponíveis;”*

Com a viabilidade de unir centrais de regulação de diversos órgãos as possibilidades médicas aumentam, e uma ferramenta de gestão pode ser útil. A Portaria 2048 também identifica esta necessidade, conforme apresentado a seguir.

*“ monitorar e orientar o atendimento feito por outro profissional de saúde habilitado (médico intervencionista, enfermeiro, técnico ou auxiliar de enfermagem), por profissional da área de segurança ou bombeiro militar (no limite das competências desses profissionais) ou ainda por leigo que se encontre no local da situação de urgência“*

Sendo a orientação médica de vital importância em um atendimento, quanto maior for a qualidade informacional que o mesmo possua, melhor será a qualidade de sua orientação. Com a presença de demais entidades médicas, o auxílio ao médico regulador torna-se viável.

*“ definir e acionar o serviço de destino do paciente, informando-o sobre as condições e previsão de chegada do mesmo, sugerindo os meios necessários ao seu acolhimento”*

Conforme exposto, o encaminhamento da vítima também pode ser auxiliado através das informações dos hospitais ou clínicas que estão atuando normalmente, inclusive com relação a equipe médica e equipamentos adequados a um determinado atendimento.

O atendimento pré-hospitalar pode assim ser caracterizado pelo envolvimento de várias entidades fisicamente distribuídas. Neste sentido, o intuito é unir as organizações que prestam serviços ao atendimento pré-hospitalar coordenando as participações destas organizações e fornecendo meios de multiplicar as informações que apoiam o médico regulador, auxiliando assim as decisões que esse deve tomar.

*“ estabelecer claramente, em protocolo de regulação, os limites do telefonista auxiliar de regulação médica, o qual não pode, em hipótese alguma, substituir a prerrogativa de decisão médica e seus desdobramentos, sob pena de responsabilização posterior do médico regulador;*

*- definir e pactuar a implantação de protocolos de intervenção médica pré-hospitalar, garantindo perfeito entendimento entre o médico regulador e o intervencionista, quanto aos elementos de decisão e intervenção, objetividade nas comunicações e precisão nos encaminhamentos decorrentes;*

*- monitorar o conjunto das missões de atendimento e as demandas pendentes; “.*

Para atender a necessidade explicitada, a solução que deve ser aplicada, nesse caso, tem o papel de auxiliar a comunicação fornecendo segurança e regularidade às informações trocadas entre, por exemplo, médicos intervencionistas e médicos reguladores. De certa forma, esta solução visa facilitar os atendimentos de demandas concorrentes, documentar recomendações e intervenções solicitadas e realizadas, definir os encaminhamentos solicitados, colocar à disposição do médico regulador todas as opções de organizações que possam auxiliar o atendimento.

As necessidades do atendimento pré-hospitalar podem ultrapassar as barreiras de atuação dos órgãos públicos, estaduais e/ou municipais. Dessa forma, um médico regulador possui autoridade para intervir junto à órgãos particulares de atendimento, assim como exigir que um atendimento tenha continuidade em uma determinada organização hospitalar, independente das condições de atendimento que essa esteja atuando. Definitivamente, a integração dos hospitais, clínicas, serviços públicos e outros, a nível informacional é relevante. Um médico regulador pode possuir em suas mãos, a qualquer momento, todas as informações das melhores condições de atendimento a um determinado paciente. Impondo o poder que lhe é conferido e com um conteúdo informacional relevante ao seu alcance, as decisões de continuidade de um atendimento pré-hospitalar poderão ser, de certa forma, apoiadas em dados atualizados e pertinentes. O atendimento pré-hospitalar é o início de um atendimento. Dessa forma, pode-se enfatizar que a condução do paciente ao lugar mais apropriado e com as melhores condições para o caso específico pode trazer benefícios à recuperação do mesmo. [Brasil, 2001].

A presença dos serviços particulares de atendimento pré-hospitalar deve ser regulamentada de acordo com as especificações da Portaria 2048. Essas organizações devem possuir centrais de regulação, médicos reguladores, médicos intervenientes, equipe de enfermagem e assistência farmacêutica. As centrais de regulação particulares, sempre que necessário, devem ser submetidas a regulação pública. Dessa forma, definem-se as entidades que respondem pelas centrais de regulação privadas, que também poderão participar de um atendimento junto às organizações públicas de regulação. É regulamentado também a presença dos órgãos de apoio, que serão descritos mais tarde, tais como polícia rodoviária, corpo de bombeiros, defesa civil e outros.

A presença de uma quantidade indefinida de organizações ativas no atendimento pré-hospitalar é inevitável. A atuação de algumas organizações, fisicamente distribuídas, deve acontecer em todos os chamados. A troca de informações entre estas organizações é a própria

rotina de um atendimento pré-hospitalar. Este quadro identifica a necessidade de uma solução capaz de acolher uma flexibilidade não encontrada em aplicações convencionais.

## 1.1 Objetivos

### 1.1.1 Objetivo Geral

Este trabalho visa conceber e modelar um sistema distribuído e aberto capaz de integrar os diferentes agentes da sociedade que atuam no atendimento pré-hospitalar.

### 1.1.2 Objetivos Específicos

- Definir uma arquitetura de sistema que permita conceber um sistema de agentes aberto.
- Identificar e estudar uma linguagem de comunicação para agentes trocarem dados e intenções.
- Modelar os agentes envolvidos no atendimento pré-hospitalar usando uma linguagem de modelagem de sistema de agentes.
- Definir e implementar um protótipo do sistema.

## 1.2 Proposta

A natureza operacional do atendimento pré-hospitalar é segmentada entre competências de organizações distintas e distribuídas fisicamente, evidenciando a necessidade de comunicação e colaboração mútua, a fim de prover assistência às vítimas. A organização do atendimento pré-hospitalar caracteriza-se como um sistema distribuído e aberto, por envolver competências multidisciplinar de organizações dinâmicas e distribuídas [Machado et al, 2000]. Toda esta dinâmica, complexidade e diversidade nos remete a pensar em uma arquitetura de sistema multi-agente para aumentar as chances de sucesso de um sistema computacional de apoio ao atendimento pré-hospitalar.

O conceito de sistema multi-agente pertence ao domínio da inteligência artificial distribuída [Bond & Gasser, 1988, 1992], que se interessa pela problemática da distribuição da *expertise* sobre um grupo de entidades denominadas agentes, cujo controle e dados são distribuídos. Cada agente possui suas próprias competências, mas tem a necessidade de interagir com os outros para resolver problemas que requerem seu domínio de *expertise*, porém transpõem suas competências [Scalabrin, 1996].

Em Ferber [Ferber, 1995] também se discute algumas razões para a distribuição da inteligência de um sistema, que são:



- os problemas complexos são efetivamente, na maior parte do tempo, fisicamente distribuídos;
- os problemas são funcionalmente bem distribuídos e heterogêneos;
- as redes de computadores impõem uma visão distribuída;
- a complexidade dos problemas impõe uma visão local, ou seja, quando um problema é muito grande para ser analisado globalmente, as soluções baseadas em abordagens locais permitem freqüentemente resolvê-los mais rapidamente;
- os sistemas devem poder se adaptar às modificações de estrutura e de ambiente. Ou seja, saber conceber sistemas computacionais eficientes, confiáveis e corretos não é mais suficiente. Os sistemas de agentes, pela sua natureza distribuída que envolvem um raciocínio local e permitem a integração dinâmica de novos agentes, são arquiteturas aptas a levar em conta as evoluções e adaptações necessárias para o funcionamento de um sistema;
- um sistema deve possuir a capacidade de evoluir frente a certas necessidades, tais como: adição de novas funcionalidades, modificações sobre sua utilização, integração de um novo software etc.; e
- a engenharia de software caminha para uma concepção em termos de unidades autônomas que interagem.

### **1.3 Estrutura do Documento**

Este documento apresenta em seu primeiro capítulo a sua introdução e seus objetivos. Visando contextualizar o estudo apresentado, o segundo capítulo mostra uma revisão sobre atendimento pré-hospitalar, assim como, alguns elementos importantes para se chegar a uma solução baseada em agentes. No terceiro capítulo é apresentado textos de outros autores sobre as técnicas e tecnologias utilizadas no desenvolvimento desta pesquisa. O protótipo que foi construído para firmar o estudo apresentado foi modelado usando a linguagem de modelagem de agentes AUML. No quarto capítulo são apresentados alguns exemplos e casos de uso desta linguagem no contexto do atendimento pré-hospitalar. No quinto capítulo é apresentado o desenvolvimento da pesquisa, mostrando a arquitetura desenvolvida e a modelagem do sistema de agentes do atendimento pré-hospitalar. O sexto capítulo apresenta os resultados e discussões considerando o protótipo que foi concebido e construído. Aqui também é disponibilizado um relato de um atendimento, que pode ser instanciado no protótipo. Este relato tem a intenção de possibilitar que o leitor possa melhor contextualizar o funcionamento

de um sistema de agente para um atendimento pré-hospitalar. No sétimo capítulo é apresentada a conclusão do trabalho acompanhada de um breve relato de trabalhos que podem complementar e validar a arquitetura aqui proposta, seguido das referências bibliográficas.

## Capítulo 2 – Revisão Bibliográfica

### 2.1 Atendimento pré-hospitalar

Um atendimento pré-hospitalar é um atendimento que procura chegar precocemente a uma vítima após ter ocorrido um agravo a sua saúde (de natureza clínica, cirúrgica, traumática, inclusive as psiquiátricas), que possa levar a sofrimento, seqüelas ou à morte, sendo assim, tem como objetivo principal prover um atendimento ou transporte de qualidade da vítima para um serviço de saúde que possa prestar-lhe a devida assistência. Chama-se atendimento pré-hospitalar primário quando o pedido de socorro é oriundo de um cidadão e de atendimento pré-hospitalar secundário quando o pedido é realizado a partir de um serviço de saúde, no qual o paciente já tenha recebido o primeiro atendimento necessário à estabilização do quadro de urgência, mas necessite de um atendimento de maior complexidade em um outro serviço de saúde especializado [Finin & Labrou, 1992a].

Através da Portaria 814/GM de julho de 2001, o Sistema Único de Saúde passa a ter um serviço de atendimento pré-hospitalar. Este serviço tem por finalidade atender a todos os cidadãos que forem acometidos por agravos de urgência em espaços públicos ou estabelecimentos privados, possibilitando atendimento médico sempre que o médico regulador julgar necessário. Será o *médico regulador de urgências* a autoridade sanitária pública que, por delegação do Sistema Único de Saúde, irá ordenar e coordenar o uso de todos os recursos envolvidos no atendimento de saúde às urgências, devendo, para isto, contar com recursos de outros órgãos que prestam socorro a população [Brasil, 2001].

A portaria 814/GM esclarece a necessidade de normatizar a estrutura e funcionamento dos serviços de atendimento pré-hospitalar, apoiando-se ao crescimento do número de serviços de atendimento pré-hospitalar móvel no setor privado de assistência à saúde e nas estradas, sem o necessário ordenamento técnico e integração aos princípios do Sistema Único de Saúde, disponibilizando serviços sem o controle adequado da formação, capacitação e educação dos profissionais que atuam na área de urgência. [Brasil, 2001].

O atendimento pré-hospitalar encontra apoio legal através das portarias 2048/GM de 5 de novembro de 2002 [Brasil, 2002b], 814/GM de 01 de junho de 2001 [Brasil, 2001], 1864/GM de 29 de setembro de 2003 [Brasil, 2003], além da resolução 1529/98 do Conselho Federal de Medicina e da regulamentação técnica e resoluções elaboradas pela ANVISA. A

citação a seguir é parte da portaria do Gabinete Ministerial 2048 e apresenta a preocupação com a implantação de serviços de atendimento pré-hospitalar.

*“Sistema Estadual de Urgência e Emergência deve ser implementado dentro de uma estratégia de Promoção da Qualidade de Vida” como forma de enfrentamento das causas das urgências. Deve valorizar a prevenção dos agravos e a proteção da vida, gerando uma mudança de perspectiva assistencial – de uma visão centrada nas conseqüências dos agravos que geram as urgências, para uma visão integral e integrada, com uma abordagem totalizante e que busque gerar autonomia para indivíduos e coletividades. Assim, deve ser englobada na estratégia promocional a proteção da vida, a educação para a saúde e a prevenção de agravos e doenças, além de se dar novo significado à assistência e à reabilitação. As urgências por causas externas são as mais sensíveis a este enfoque, mas não exclusivamente. As urgências clínicas de todas as ordens também se beneficiam da estratégia promocional” [Brasil, 2002a].*

*“Estas centrais, obrigatoriamente interligadas entre si, constituem um verdadeiro complexo regulador da assistência, ordenador dos fluxos gerais de necessidade/resposta, que garante ao usuário do SUS a multiplicidade de respostas necessárias à satisfação de suas necessidades. As Centrais de Regulação Médica de Urgências devem ser implantadas, de acordo com o definido no Anexo II da Portaria SAS/MS nº 356, de 22 de setembro de 2000” [Brasil, 2002b].*

*“O aumento nos casos de acidentes e violência tem forte impacto sobre o Sistema Único de Saúde e o conjunto da sociedade. Na assistência, este impacto, pode ser medido diretamente pelo aumento dos gastos realizados com internações hospitalares, assistência em UTI e a alta taxa de permanência hospitalar deste perfil de paciente. Na questão social, pode ser verificado pelo aumento de 30% no índice APVP (Anos Potenciais de Vida Perdido) em relação a acidentes e violências nos últimos anos, enquanto que por causas naturais este dado encontra-se em queda.” [Brasil, 2002b].*

Os serviços específicos de atendimento pré-hospitalar são os responsáveis pela maior parte dos encaminhamentos aos serviços de saúde. Em Curitiba estes serviços são realizados através dos números 192 e 193, respectivamente para o Serviço de Atendimento Móvel de Urgência (SAMU) e para o Serviço Integrado de Atendimento ao Trauma de Emergência (SIATE).

Em se tratando de serviço de atendimento pré-hospitalar, pode-se afirmar que os sistemas manuais funcionam, prova disto é o estudo de caso realizado nesta instituição [Machado & Grein, 1999]. Neste exemplo, todos os passos são realizados de maneira manual, sem a interação de um sistema informatizado auxiliando a integração dos atendimentos, das entidades participantes e das ocorrências [Grupo pré-hospitalar, 2001].

O SIATE (Serviço Integrado de Atendimento ao Trauma em Emergência), disponível à população pelo fone 193, existe para prestar atendimento pré-hospitalar ao traumatizado no perímetro urbano de cidades com mais de 150.000 habitantes [Grupo pré-hospitalar, 2001]. Através da portaria interministerial nº 18 de 25/05/1987, foi implantado um projeto piloto de atendimento ao trauma na região metropolitana de Curitiba - Projeto Piloto de Atenção ao Acidentado de Tráfego, o que determinou a criação do SIATE. As parcerias participantes na ocasião da implantação eram a Secretaria Estadual de Saúde (SESA), a Secretaria Municipal de Saúde e a Secretaria Estadual de Segurança Pública (SESP). Operando em fase experimental os serviços do SIATE tiveram início em maio de 1990 [SIATE, 2002].

O objetivo do SIATE é prestar socorro de emergência às vítimas de acidentes ocorridos em vias e logradouros públicos, em ambientes profissionais e domiciliares, garantindo o suporte básico e avançado de vida. O transporte das vítimas atendidas é feito em ambulâncias, em condições ideais e com equipamentos adequados e indispensáveis a manter o suporte de vida, evitando o agravamento das lesões.

*“O atendimento do SIATE é exclusivo ao trauma, sendo que em torno de 70% das ocorrências são relacionadas à acidentes de trânsito, como atropelamento, colisão, capotamento, queda de bicicleta, 10% a acidentes interpessoais como agressão, ferimento por arma de fogo, ferimento por arma branca, 10% devido a quedas e os 10% restante dos atendimentos são de causas variadas como queimaduras, soterramento, acidente de trabalho ou ainda problemas clínicos com risco iminente de vida.” [SIATE, 2002]*

Em se tratando de SAMU, os primeiros trabalhos realizados na área do socorro a vítimas era o transporte de pacientes. Este serviço começou a ser realizado na década de 90 por motoristas sem capacitação específica. Iniciativas futuras vieram a partir de 1991 por intermédio de grupos de voluntários, secretaria de estado e hospitais municipais. Estas iniciativas tinham a idéia de integrar-se ao resgate do corpo de bombeiros a fim de estruturar e fornecer um serviço de atendimento pré-hospitalar integrado, garantindo uma assistência pré-hospitalar de qualidade às vítimas de urgência clínica e traumática [SAMU, 2002a].

Atualmente o SAMU é regulamentado pelo Governo Federal através de suas portarias como o serviço que deve atender as necessidades de todo o território nacional [Brasil, 2003]. Similar ao serviço de atendimento do SIATE e SAMU realiza atendimento pré-hospitalar à vítima de agravo súbito à saúde de origem clínica ou traumática, utilizando-se de uma frota de ambulâncias devidamente equipadas e profissionais capacitados, possibilitando maiores

chances de sobrevivência, diminuição das seqüelas, e o transporte seguro até o serviço de saúde mais adequado para continuidade do tratamento [SAMU, 2002b].

Atividades desenvolvidas pelo SAMU

- atendimentos de urgência / emergência;
- atendimentos sob agendamento prévio para pacientes crônicos e transportes intermunicipais eletivos;
- cobertura de eventos de risco;
- cobertura de acidentes de grandes proporções;
- ações educativas para a comunidade;
- atividades político-científicas com outros SAMUs do Brasil e da França [SAMU Franca, 2002].

## **2.2 Trajetória dos sistemas para o atendimento pré-hospitalar na PUCPR**

### **2.2.1 Gerenciamento de Emergências Médicas**

Um dos primeiros trabalhos desenvolvidos para atender a área do atendimento pré-hospitalar foi contemplado durante o ano 1999. O GEM (**G**erenciamento de **E**mergências **M**édicas) é um software de gerenciamento de emergências médicas possui em sua essência as seguintes atribuições:

- a) identificação de chamadas;
- b) triagem de chamados;
- c) triagem das ocorrências;
- d) orientação durante o atendimento pré-hospitalar;
- e) controle das unidades móveis envolvidas no atendimento, e;
- f) auxílio na definição do serviço de saúde a encaminhar a vítima.

O GEM foi desenvolvido com o intuito de auxiliar o atendimento prestado pelo Sistema Integrado de Atendimento de Emergência atuando junto a regulação das chamadas atendidas pelo órgão. Esta aplicação fora desenvolvida sobre uma plataforma Cliente / Servidor usando a linguagem de programação Delphi e o Sistema Gerenciador de Banco de Dados Microsoft SQL Server [Machado & Grein, 1999].

### **2.2.2 Sistema de Informação e Comunicação no Atendimento Pré-Hospitalar**

O SICAPH (**S**istema de **I**nformação e **C**omunicação no **A**tendimento **P**ré-**H**ospitalar) visa o desenvolvimento de um sistema de informação de emergências médicas para

atendimento pré-hospitalar, com o controle e triagem de chamados, despacho de ambulâncias e outras viaturas, manipulação de equipamentos de socorro e serviços realizados [Dias et al, 2002]. Neste tipo de sistema, a informação georeferenciada é muito importante. As ferramentas de geoprocessamento facilitaram o gerenciamento do chamado, a localização das ocorrências, o envio do socorro necessário e acompanhamento das viaturas e ocorrências, permitindo um suporte adequado para todos os tipos de informações e tomadas de decisões durante a ocorrência. Com este intuito a PUCPR tem desenvolvido algumas soluções para a informatização do atendimento pré-hospitalar [Machado & Grein, 1999; Parolin, 1998a; Parolin, 1998b; Parolin, 1999] já há algum tempo. O projeto SICAPH, começou a ser desenvolvido em janeiro de 2001, objetivando integrar as pesquisas já desenvolvidas, novas tecnologias e conhecimentos, e minimizar os problemas de atendimentos de urgência e emergência médica. O sistema apresenta-se modularizado de acordo com a Figura 1, indicando os passos que geralmente ocorrem durante o atendimento de emergência. Este produto foi desenvolvido sob a plataforma *web* utilizando a linguagem de programação Java e o SGDB (Sistema Gerenciador de Banco de Dados) da Oracle. [Dias et al, 2002].

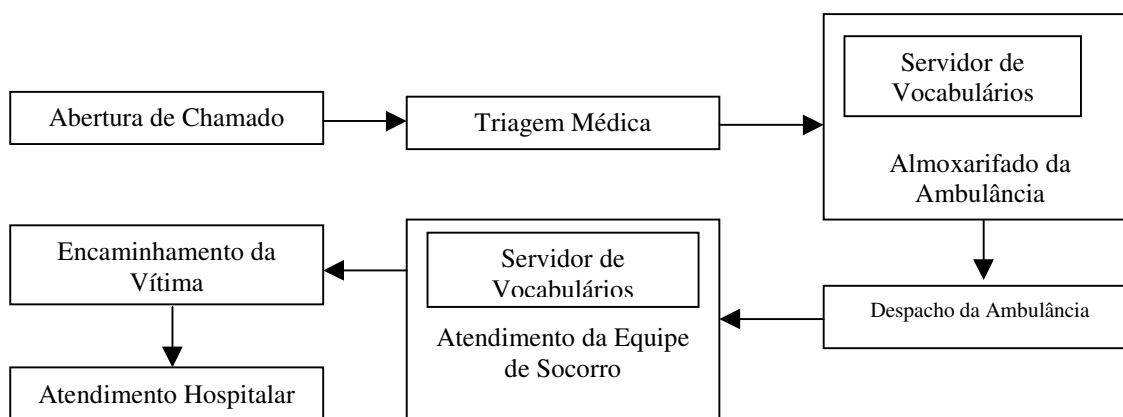


Figura 1 - Módulos do Sistema [Dias et al, 2002]

### 2.2.3 Integração de sistemas na área de saúde

As pesquisas desenvolvidas acerca da integração de sistemas trouxeram a necessidade de considerar a participação de várias aplicações para atender um contexto pré-definido. O desenvolvimento de uma pesquisa considerando uma forma de integrar aplicações na área de saúde traz uma importância com relação a necessidade da troca de informações entre as instituições da área de saúde [Grein & Machado, 2002].

Este primeiro trabalho de pesquisa integrou o SICAPH ao Cartão Nacional de Saúde, visando promover de forma facilitada e rápida a identificação dos pacientes em estado de

emergência. Assim como a participação de outros órgãos, a contribuição do Cartão Nacional de Saúde é pontual, especificamente para a identificação de um paciente.

#### **2.2.4 Uma arquitetura multi-agente para o atendimento pré-hospitalar**

Os estudos de uma solução multi-agente começaram com a definição de uma arquitetura de agentes cognitivos independentes para o atendimento pré-hospitalar. Neste sentido, a opção por uma arquitetura distribuída e aberta baseou-se nos seguintes critérios:

- a) trata-se de um problema naturalmente subdividido, sendo que cada subproblema (*Central de Regulação, Unidade Móvel, etc*) pode ser representado como um agente autônomo e independente;
- b) O uso de agentes viabiliza a utilização de especialistas distintos para solucionar, através das trocas de informações, situações difíceis de forma dinâmica (ou sobre demanda);
- c) o paralelismo existente nos sistemas multi-agente contribui na implementação de um sistema de alto desempenho.

O atendimento pré-hospitalar apresenta em seu contexto os critérios que atendem as especificações de uma aplicação multi-agente. A fácil decomposição das necessidades do atendimento pré-hospitalar em entidades que possuem uma atividade complexa e que necessariamente devem trocar informações com outras sempre visando a solução de um atendimento, convergem para o desenvolvimento para sistemas eficientes e que possuam fácil manutenibilidade. [Machado et al, 2004]

Uma das características de uma solução multi-agente é o reuso de várias tecnologias e padrões definidos. Algumas necessidades deste tipo de sistema podem ser acompanhadas no próximo capítulo.



## Capítulo 3 – Agentes de Software

A inteligência artificial distribuída estuda sociedades de agentes de software para trabalharem juntos resolvendo problemas de natureza distribuída, ou seja, problemas complexos que não podem ser resolvidos por um único programa [Scalabrin et al, 1996]. Não há uma única definição para o conceito de agente. Na literatura encontram-se conceitos distintos dependendo do foco do autor [Smith et al, 1994]. Contudo, tais definições não se anulam. Elas colaboram mutuamente para um entendimento mais amplo da concepção de um agente. Um agente é um software persistente que permite a interoperabilidade entre sistemas computacionais de organizações distintas. Tipicamente devem possuir capacidade adaptativa entre os ambientes heterogêneos das organizações, caracterizando a necessidade de sistemas abertos [Genesereth & Ketchpel, 1994].

Agente é uma entidade de software persistente dedicada a um propósito específico. A capacidade de persistência do software distingue agentes de sub-rotinas: o agente tem suas próprias concepções de como efetuar tarefas [Smith et al, 1994]. Percepção das condições dinâmicas do ambiente, ação para modificar as condições no ambiente e raciocínio para interpretar percepções, resolver problemas, fazer inferências e determinar ações, ou seja, um agente pode agir sobre ele mesmo ou sobre o ambiente [Ferber, 1994; Hayes-Roth, 1995] . Agentes podem agir autonomamente, de maneira racional ou intencional, para cumprir um conjunto de objetivos [Maes, 1995; Shoham, 1993; Demazeau & Müller, 1991]. Um agente interface, por exemplo, pode ser definido como um personagem desempenhado pelo computador para representar o usuário em um ambiente virtual. Ao "pensar" e comunicar comportamentos, os agentes interfaces baseiam-se em metáforas de organismos vivos em termos de acessibilidade cognitiva e estilo de comunicação [Laurel, 1990; Sycara et al, 1996; Maes, 1994b]. Um agente ainda pode ser mantenedor de informações ou responsável por tarefas simples, cujo principal intuito é a filtragem da informação para reduzir a sobrecarga cognitiva do usuário [Wooldridge & Jennings, 1995; Maes, 1994a ;Mitchell et al, 1994] . Na comunidade de inteligência artificial, os agentes são constantemente entendidos como entidades autônomas [Luck, 1995].

"Um agente pode ser visto como algo que observa o ambiente através de sensores e age nesse ambiente através de atuadores, compondo sistemas computacionais que habitam algum ambiente dinâmico complexo, percebem e agem de forma autônoma nesse ambiente e,

fazendo isto, realizam um conjunto de metas e tarefas para as quais foram projetados. Estes agentes percebem o ambiente que estão inseridos e atuam continuamente para atender a seus objetivos" [Russel & Norvig, 1995; Franklin & Graesser, 1997; Maes, 1995].

Olhando para o contexto da inteligência artificial pode-se afirmar que a inteligência artificial distribuída baseia-se no comportamento social, considerando sociedades de agentes inteligentes, autônomas e dotadas de capacidades cognitivas. Os agentes especialistas nas atividades que desempenham não trabalham isolados, de forma cooperativa eles tentam resolver um problema da melhor forma possível, caracterizando uma área da inteligência artificial distribuída, denominada sistemas multi-agente [Demazeau, 1995].

Partindo deste pressuposto, começa-se a colocar a visão do agente como parte de um contexto, assim como será tratado no atendimento pré-hospitalar. Wayer [Wayer, 1995] e Cheong [Cheong, 1995] afirmam que um agente encontra-se imerso em uma sociedade de agentes. Nesta sociedade, a interação e a coordenação das metas e dos planos de ações destes agentes têm como objetivo a resolução cooperativa de uma determinada tarefa.

### **3.1 Características de um agente**

Um agente possui um conjunto de propriedades que ajudam a sua atuação no ambiente que está inserido. Desta forma, um agente pode ser ainda um sistema computacional com habilidade de autonomia, capacidade social, reatividade e pró-atividade [Wooldridge & Jennings, 1995]. As propriedades ou habilidades dos agentes diferenciam a atuação de um agente com relação a outro. Abaixo estão apresentadas algumas das características que poderão ser encontradas em um agente:

Reatividade: propriedade que permite que os agentes percebam seu ambiente e responda adequadamente às mudanças nele ocorrida. O ambiente aqui referido pode ser o mundo físico, uma interface gráfica, uma coleção de outros agentes, a Internet ou ainda um contexto que pode ser definido com a combinação de mais de um desses elementos [Wooldridge & Jennings, 1995].

Autonomia: capacidade de tomar iniciativa e ter um controle sobre as suas próprias ações ou ainda a capacidade de tomar ações para realizar algumas tarefas e objetivos, sem a interferência do usuário final. Deve haver um elemento de independência no agente [Franklin & Graesser, 1997; Foner, 2004], ou seja, os agentes operam sem a intervenção direta de seres humanos ou outros agentes [Castelfranchi, 1995; Davidsson, 1995; Nissen, 1995].

Habilidade social: Habilidade que os agentes possuem de interagir com os outros agentes ou pessoas no momento adequado, para concluir suas tarefas ou ajudar outros agentes [Krulwich, 1997; Paraiso, 1996].

Confiabilidade: Os agentes devem demonstrar veracidade e benevolência, ou seja, os usuários precisam ter certeza de que os agentes serão fidedignos nas ações e informações e que irão agir em seu benefício [Wooldridge & Jennings, 1995].

Comportamento adaptativo: Capacidade do agente de modificar seu comportamento em função de experiências anteriores [Wooldridge & Jennings, 1995].

Mobilidade: Capacidade do agente de transportar-se de uma máquina à outra, ou ainda, habilidade do agente movimentar-se na rede [Franklin & Graesser, 1997].

Capacidade de aprendizagem: Capacidade que um agente deve possuir para executar uma tarefa com maior eficiência do que em execuções anteriores [Auer, 1995; Belgrave, 1995].

Flexibilidade: Habilidade dos agentes de escolher dinamicamente as ações e a seqüência de execução das mesmas, em resposta a um estado do ambiente [Franklin & Graesser, 1997], [Auer, 1995].

Comunicabilidade: Capacidade dos agentes de comunicar através de troca de mensagens em uma expressiva e específica linguagem para conversação [Franklin & Graesser, 1997].

Pró-atividade: Capacidade manifestada pelo agente de exibir um comportamento direcionado a objetivos. Desta forma, o agente não age simplesmente em resposta ao ambiente, mas de acordo com um propósito. Para tanto, deve exibir um comportamento oportunista, voltado para a realização de seus objetivos [Wooldridge & Jennings, 1995].

Racionalidade: Os agentes devem ter racionalidade para agir de forma lógica [Galliers, 1988]. A capacidade de raciocínio é talvez o aspecto mais importante que distingue um agente inteligente dos outros agentes. Dizer que um agente inteligente tem raciocínio significa dizer que ele tem a habilidade de inferir e extrapolar, baseado no conhecimento atual e nas experiências, de uma maneira racional e reproduzível [O'Connor et al, 1995; Belgrave, 1995].

Um agente não deve necessariamente possuir todas as habilidades descritas. As habilidades do agente dependerão do contexto a que o mesmo está inserido e do papel que está representando. Um agente sempre estará interagindo com o meio externo. Desta forma, o agente deverá estar inserido em um contexto previamente definido, onde a troca de dados, informações ou conhecimento ocorram através de um protocolo conhecido [Paraiso, 1996].

As propriedades de um agente são caracterizadas pela arquitetura do sistema. Através dessa arquitetura são fornecidos e gerenciados os recursos primitivos de um conjunto de agentes [Maes, 1995].

### 3.2 Sistemas abertos

A integração dinâmica de agentes implica na capacidade de aprendizagem do agente sobre tarefas em que estão envolvidos. Este conceito é usado em várias áreas da ciência da computação: a) interconexão entre arquiteturas de diferentes máquinas (modelo OSI); b) ligação dinâmica entre cliente e servidor (modelo CORBA); c) reconfiguração dinâmica e seleção mútua (modelo *Contract-Net*) e d) integração dinâmica de um agente em um contexto de trabalho (modelo agente). A complexidade e a dinâmica dos sistemas levam a uma nova situação onde saber como projetar sistemas computacionais eficientes, confiáveis e corretos não é mais suficiente [Ferber, 1995].

Jacques Ferber argumenta que os novos softwares devem facilitar adaptações, em particular às mudanças no contexto do trabalho, tais como: a) mudança do sistema operacional; b) mudança do sistema gerenciador de banco de dados; c) mudança da interface gráfica e d) inserção de um novo software. O novo sistema computacional deve ter capacidade que envolva situações como: a) inclusão de novas funcionalidades; b) alteração do modo de funcionamento do software; c) integração do novo software. Neste sentido, pode-se dizer que sistemas abertos são fortemente adaptativos.

Sistemas multi-agentes são candidatos promissores para implementação de arquiteturas distribuídas, heterogêneas, flexíveis e abertas, capazes de oferecer uma grande quantidade de serviço em ambientes de trabalho coletivo sem uma estrutura prévia definida. Sistemas multi-agentes podem ser classificados de acordo com a arquitetura que possuem (organização global), o grau de autonomia de cada agente, o tipo de protocolo usado para comunicação, ou a sua complexidade. A principal distinção está relacionada aos agentes autônomos. Agentes reativos são muito simples sem qualquer representação do seu ambiente. Eles interagem com um tipo de comportamento estímulo-resposta [Ferber & Drogoul, 1992]. Assim, comportamentos inteligentes podem emergir de uma população de numerosos agentes [Brooks, 1991; Scalabrin, 1996; Beer, 1992; Demazeau, 1990; Castelfranchi, 1990; Wooldridge & Jennings, 1994].

Uma condição essencial para o desenvolvimento de um sistema aberto é que o ambiente permita gerir a adição e a supressão de um agente de uma aplicação sem a necessidade de parar e reinicializar o sistema. Porque, quando um novo serviço é adicionado, a necessidade de recompilar todos os programas sobre todas as máquinas da rede é claramente inaceitável.

Um ponto comum entre todas as arquiteturas baseadas no modelo cliente-servidor é a presença de uma entidade mediadora entre o requisitante e o fornecedor de um serviço. As abordagens se distinguem pelo nível de conhecimentos modelados sobre os objetos servidores ou agentes serviços. Em certos casos, essas entidades intermediárias recobrem perfeitamente a noção de agente autônomo.

Com relação ao modelo cliente-servidor, o modelo de coordenação para agentes assíncronos é mais complexo. Porque os agentes são:

- de um lado independentes para levar em conta ou rejeitar uma solicitação e,
- de outro, obrigados (eles mesmos) a gerir as interações com os outros agentes fornecedores de serviços.

Os agentes autônomos utilizam o serviço de uma entidade intermediária somente para o transporte das mensagens. Os agentes são os próprios responsáveis da negociação de um serviço e da gestão de suas relações com os outros agentes [Scalabrin, 1996].

### **3.3 Sistemas multi-agente**

A área de sistemas multi-agente estuda o comportamento de um grupo *organizado* de agentes *autônomos* que cooperam na resolução de problemas que vão além das capacidades de resolução de cada um individualmente, ou ainda, um sistema composto de múltiplos agentes inter-atuantes [Durfee et al, 1994]. Duas propriedades, aparentemente contraditórias, são fundamentais para os sistemas multi-agente: a autonomia dos agentes e sua organização [Briot & Demazeau, 2002]. Conforme já mencionado neste trabalho, autônomo significa “capacidade de tomar iniciativa e ter um controle sobre as suas próprias ações ou ainda a capacidade de tomar ações para realizar algumas tarefas e objetivos, ou seja, para funcionar, um agente não precisa de outros agentes, mesmo que para alcançar seus objetivos ele eventualmente precise da ajuda de outros”. Por outro lado, a organização coloca restrições aos comportamentos dos agentes, procurando estabelecer um comportamento de grupo coeso [Jennings & Wooldridge, 1998; Wooldridge, 2002].

Um sistema multi-agente surge em um cenário de entidades diversificadas onde cada uma possui uma tarefa dedicada, ou seja, a necessidade de cumprir seus objetivos é suficiente para tomar boa parte de seu tempo e processamento. Como a recuperação de informações do atendimento pré-hospitalar é dependente do funcionamento destas diversas entidades, um modelo de arquitetura onde cada entidade continua a realizar suas atribuições, e ao mesmo tempo, possa compartilhar seus conhecimentos e dividir suas atribuições, é capaz de

solucionar de maneira elegante uma integração que, nos dias atuais, ainda não existe. Dada esta situação, a abordagem destinada ao sistema multi-agente traz as seguintes características [Hubner, 2003].

- os agentes são concebidos independentemente de um problema particular;
- a interação entre os agentes não é projetada anteriormente, busca-se definir protocolos que possam ser utilizados em situações genéricas;
- a decomposição de tarefas para solucionar um dado problema pode ser feita pelos próprios agentes;
- não existe um controle centralizado da resolução de um problema.

As propriedades dos sistemas multi-agente apresentam algumas vantagens:

- viabilizar sistemas *adaptativos* e *evolutivos*: o sistema multi-agente tem capacidade de adaptação a novas situações, tanto pela eliminação e/ou inclusão de novos agentes ao sistema quanto pela mudança da sua organização [Hubner, 2003].
- modelar sistemas complexos e distribuídos: em muitas situações, o conhecimento está distribuído, o controle é distribuído, os recursos estão distribuídos. E, quanto à modelagem do sistema, a decomposição de um problema e a atribuição dos sub-problemas a agentes permite um alto nível de abstração e independência entre as partes do sistema [Jennings & Wooldridge, 1998];
- tirar proveito de ambientes *heterogêneos* e *distribuídos*: agentes com arquiteturas diferentes, que funcionam em plataformas diferentes, distribuídas em uma rede de computadores, podem cooperar na resolução de problemas. Isto permite o uso das potencialidades particulares de cada arquitetura e, pela distribuição, melhora o desempenho do sistema;
- permitir conceber *sistemas abertos*: os agentes podem migrar entre sociedades, isto é, agentes podem sair e entrar em sociedades, mesmo que desenvolvidos por projetistas e objetivos distintos.

### 3.4 Organização de um sistema multi-agente

Uma organização é um conjunto de unidades funcionais (operacionais e administrativas) relacionadas segundo uma estrutura que permite, através do desempenho das atividades, alcançarem os objetivos globais da organização [Shmeil, 1999].

Cada indivíduo inserido dentro de uma organização caracteriza-se como uma unidade funcional, como especifica Shmeil, que podem ser vistas como centros cognitivos, dotados de percepção, raciocínio e memória, o que possibilita o desenvolvimento de capacidades como comunicação, planejamento, coordenação, execução de suas especialidades e a aprendizagem. A organização como um todo passa a ser considerada como um conjunto composto de centros cognitivos e de uma base de conhecimentos corporativa.

Para a formação de uma organização, dois aspectos devem ser observados. A divisão da organização em partes menores (Unidade Funcional ou Centro Cognitivo) e a coordenação destas partes. Estas duas atividades são de vital importância para uma organização e devem ser realizadas seguindo alguns princípios que primam pela melhor aplicação de ambos. Marcos Shmeil (1999) sugere que a divisão das atividades organizacionais, ou seja, a caracterização de uma unidade funcional pode ser realizada seguindo alguns critérios:

- Função: diferenciando as atividades que devem ser realizadas pelas unidades funcionais.
- Produto: união de tarefas realizadas por unidades funcionais capazes de gerar um bem ou produto.
- Cliente e mercado: divisão baseada na satisfação do cliente ou para atender a uma necessidade de mercado.
- Geográfico: quando a distância entre as unidades passa a ser objeto divisor.
- Projeto: quando o produto envolve grande concentração de recursos ou as organizações são fortemente influenciadas pelo desenvolvimento tecnológico.
- Processo: quando a seqüência do processo produtivo define o agrupamento.

Após a aplicação destes critérios de divisão, o conjunto das unidades produzidas pode ser classificado em duas classes:

- Unidades administrativas.
- Unidades executoras.

As unidades administrativas são responsáveis pelo planejamento e coordenação das atividades diferenciadas pela divisão do trabalho, pela estrutura física e conceitual. As unidades executoras são responsáveis pela execução das atividades planejadas [Shmeil, 1999 apud Simon, 1960].

Toda divisão de organização em unidades funcionais caracteriza uma disposição hierárquica, levando em consideração uma visão sistêmica. Um exemplo mencionado por Simon [Simon, 1960] trata de dois relojoeiros responsáveis pela montagem de relógios. Um deles adota a montagem dos relógios em partes, em módulos; o outro monta o relógio de forma monolítica. A diferença de produtividade acontece no momento em que o trabalho de ambos é interrompido. Ao retornar ao trabalho, o primeiro relojoeiro, que realiza a montagem em partes, tem que repensar apenas na parte exata que está montando, enquanto o segundo deve retomar seu trabalho estudando toda a montagem do relógio. Aproximando este exemplo ao atendimento pré-hospitalar, a divisão das tarefas, realizadas por estruturas distintas, permitirá que o controle do módulo venha a ser feito apenas dentro do mesmo, não levando para toda a rede a organização de seu módulo. Desta forma, as informações trafegadas na organização do atendimento pré-hospitalar são apenas as informações necessárias ao funcionamento de cada agente.

Assim como em outras arquiteturas, a arquitetura do atendimento pré-hospitalar possui em suas unidades funcionais uma relação de objetivos, ou tarefas. Cada agente participante da rede possui suas atribuições e participam do planejamento das atividades da rede como se estas tarefas fossem compromissos assumidos. Como as tarefas dos agentes são planejadas em seu contexto, a arquitetura também trabalha com seus objetivos e suas tarefas, que, neste caso, pode-se resumir como prestar atendimento pré-hospitalar. Em suma, deve haver a preocupação por parte de uma unidade funcional participante em organizar ou mesmo realizar a gestão da arquitetura. A principal responsabilidade desta estrutura será a preocupação com um atendimento, manterão a relação das tarefas e atividades das unidades funcionais (agentes) ativos e, principalmente, os relacionamentos existentes entre estas unidades, com o intuito de guiar a participação de cada entidade no momento apropriado.

Segundo Marcos Shmeil (1999), existem dois canais distintos de comunicação entre as unidades de uma arquitetura: os formais e os informais.

A distribuição formal de comunicação diz respeito ao fluxo hierárquico definido pela arquitetura, ou seja, uma unidade funcional (agente) irá se relacionar exclusivamente com unidades funcionais imediatamente superiores e imediatamente inferiores definidos na estrutura organizacional da arquitetura.

A distribuição informal apresenta a comunicação de uma unidade funcional independente do nível apresentado pela estrutura organizacional.

A coordenação mencionada acima passa a ser necessária pelo interesse de unir os objetivos de cada unidade em prol do objetivo da arquitetura. A coordenação possibilita a



integração e o ajustamento de esforços do trabalho individual para o cumprimento de um objetivo [Singh & Rein, 1992]; é a ação de trabalhar em conjunto [Malone et al, 1991].

Segundo Marcos Shmeil (1999), para a ciência da computação, a coordenação está associada à partilha dos recursos computacionais, à segmentação e atribuição de tarefas para os processadores e à gestão dos fluxos de informação, particularmente tratados pela inteligência artificial distribuída.

Existem 3 mecanismos que podem ser utilizados para a coordenação de uma arquitetura:

- a) Coordenação por ajustamento mútuo: neste caso, a responsabilidade da coordenação recai sobre os agentes que estão diretamente relacionados. Os conflitos que porventura venham a aparecer devem ser solucionados entre as entidades participantes.
- b) Coordenação por supervisão direta: nesse caso, existe uma unidade exclusiva para a resolução da comunicação entre as demais entidades, o que caracteriza uma comunicação através de canais formais.
- c) Coordenação por padronização: neste caso, cada unidade da arquitetura possui conhecimento exato das tarefas das demais unidades de relacionamento e sabe o momento de interagir na arquitetura. Neste caso também é utilizado o canal formal de comunicação. Neste tipo de coordenação existe a possibilidade de manter três padrões distintos: processos de trabalho, resultados e qualificações. A normalização por processos diz respeito a como determinada atividade deve ser desenvolvida. A normalização por resultado diz respeito ao resultado que cada agente deverá produzir para a arquitetura e a normalização por qualificação diz respeito a que tipo de atividade o agente é qualificado a realizar.

Desta forma, o contexto interno de uma arquitetura é organizado segundo a divisão das tarefas, a coordenação e a comunicação dos agentes envolvidos, podendo possuir mais de uma estrutura de coordenação, assim como utilizar dois canais de comunicação possíveis.

### **3.4.1 Modelo de agente reativo e cognitivo**

A forma de funcionamento de um agente deve estar implícita em sua arquitetura. A arquitetura de um agente pode ser definida como uma metodologia particular para definir agentes. Ela especifica como o agente é decomposto na construção de um conjunto de módulos (componentes) e como estes módulos podem interagir. O conjunto dos módulos e suas interações devem prover uma resposta para a questão de como os sensores de dados e o estado interno corrente do agente determinam suas ações e futuros estados internos. Uma

arquitetura abrange técnicas e algoritmos para suportar esta metodologia [Maes, 1995; Wooldridge & Jennings, 1995].

As arquiteturas de agentes podem ser divididas em duas áreas: a arquitetura deliberativa (abordagem clássica do paradigma de IA) e a arquitetura reativa (abordagem alternativa) [Wooldridge & Jennings, 1995].

### 3.4.2 Arquitetura Reativa

O modelo reativo não apresenta pré-definições, não apresenta nenhum modelo do mundo, possui apenas mapeamentos de situações e respostas associadas, responde ao que lhe é questionado através de conhecimento obtido pelas situações que já vivenciou e não usa argumentos simbólicos complexos. Assim, quando uma mudança no ambiente ocorre, o agente executa a ação correspondente [Franklin & Graesser, 1997; Heilmann, 1995; Russel & Norvig, 1995; Holland, 1975].

Os sistemas de agentes reativos são constituídos por um grande número de agentes de granularidade fina, bastante simples, sem inteligência e sem representação de seu ambiente. Eles podem modelar, por exemplo, uma sociedade de formigas, ou ainda os clientes e os servidores na abordagem OMG CORBA. Estes agentes são fortemente acoplados. Eles interagem utilizando um comportamento do tipo estímulo/resposta [Ferber & Drogoul, 1992]. Um comportamento inteligente emerge a partir das interações entre estes agentes e seu ambiente [Minsky, 1986]. Ou seja, os agentes não são individualmente inteligentes, mas seu comportamento global é inteligente.

Na figura 2 é apresentada a arquitetura definida por Russel e Norvig [Russel & Norvig, 1995] para o modelo reativo de agente. A percepção do ambiente e um conjunto de regras são o que determinam o comportamento de um agente. Um agente reativo por definição percebe um estímulo e executa uma tarefa, pelo fato que a primeira é responsável pela ação do agente para resolver a segunda. Neste caso, o agente normalmente age por ter recebido no ambiente uma interação que o fez agir de tal maneira.

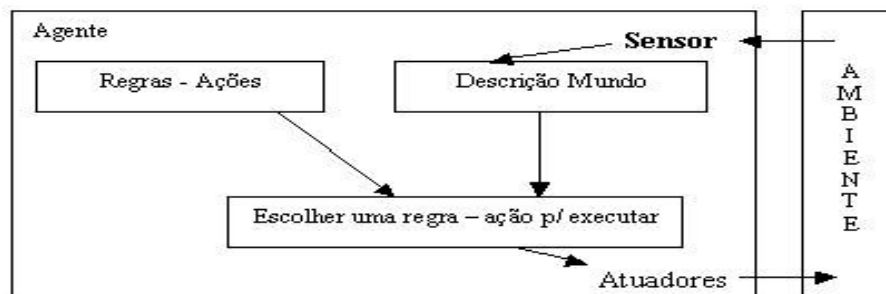


Figura 2: Modelo de agente reativo

Na literatura, as aplicações mais presentes concernem aos robôs. Por exemplo, o comportamento complexo de um robô pode ser decomposto em comportamentos individuais simples. Esses comportamentos são colocados em contato com o mundo real; o robô capta continuamente informações sobre o ambiente que o cerca. Deste modo, o robô faz parte do ambiente e reage baseado em um comportamento simples, ditado pelos estímulos/respostas que podem chegar [Scalabrin, 1996]. Dentre os trabalhos sobre os agentes reativos, pode-se citar Brooks [Brooks, 1986], que realizou os primeiros estudos sobre os agentes reativos, onde, a idéia central é que a concepção de um robô inteligente e autônomo recai na criação de um conjunto de camadas que agem como pequenos robôs reativos. Ferber & Jacopin [Ferber & Jacopin, 1990] e Ferber & Drogoul [Ferber & Drogoul, 1992] utilizam a eco resolução, uma abordagem particular de resolução distribuída de problemas. Eles consideram que a resolução distribuída de um problema é uma série de interações bastante simples dentro de uma população de agentes. A solução do problema emerge das interações entre os agentes. Steels [Steels, 1989] tentou resolver o problema da coleta de minério por robôs em ambiente desconhecido.

### 3.4.3 Arquitetura Cognitiva

Também chamados de agentes cognitivos, estes agentes possuem um modelo do mundo, que possivelmente os inclui também. Este modelo é de certa forma pré-concebido, porém seu estado pode ser alterado através da comunicação com os demais agentes da rede e através da percepção do ambiente realizada pelos sensores do agente. O agente estima que ações serão necessárias para alcançar um determinado objetivo através da interpretação deste modelo, e então executa ações que levarão à sua realização. [Franklin & Graesser, 1997; Heilmann, 1995; Russel & Norvig, 1995].

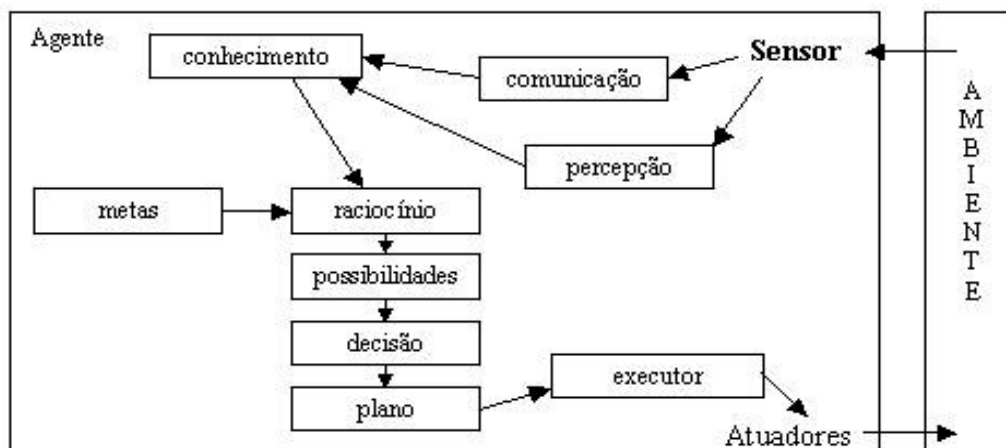


Figura 3: Modelo de agente cognitivo

Em Hubner [Hubner, 2003] encontra-se o modelo cognitivo, que apresenta na figura 3 a arquitetura proposta por [Beer, 1992]: o conhecimento que o agente possui é formado a partir da sua percepção do ambiente e da comunicação com outros agentes, ou seja, o conhecimento é atribuído ao agente através de seus relacionamentos, seja com o ambiente ou com os demais agentes da rede. Dado este conhecimento e uma meta, o agente gera um conjunto de possíveis planos que atingem esta meta. Desta forma, o agente utiliza o conhecimento para rotear as possíveis soluções para o problema que o mesmo foi incumbido de resolver. Dadas estas possibilidades ou soluções, o agente analisa qual a melhor solução a ser executada. Neste caso, pode-se afirmar que o agente tomou uma determinada ação porque esta ação faz parte do processo que leva o agente a cumprir uma determinada tarefa. Obviamente, esta diferença entre as arquiteturas somente pode ser percebida olhando-se “dentro” dos agentes. Somente pela observação do comportamento não se pode dizer se um agente é reativo ou cognitivo.

#### 3.4.4 Tipos de organização em um sistema multi-agente

Existem duas formas de classificar uma organização quanto a seu tipo: visão centrada nos agentes e visão centrada na organização, como apresentado na Figura 4.

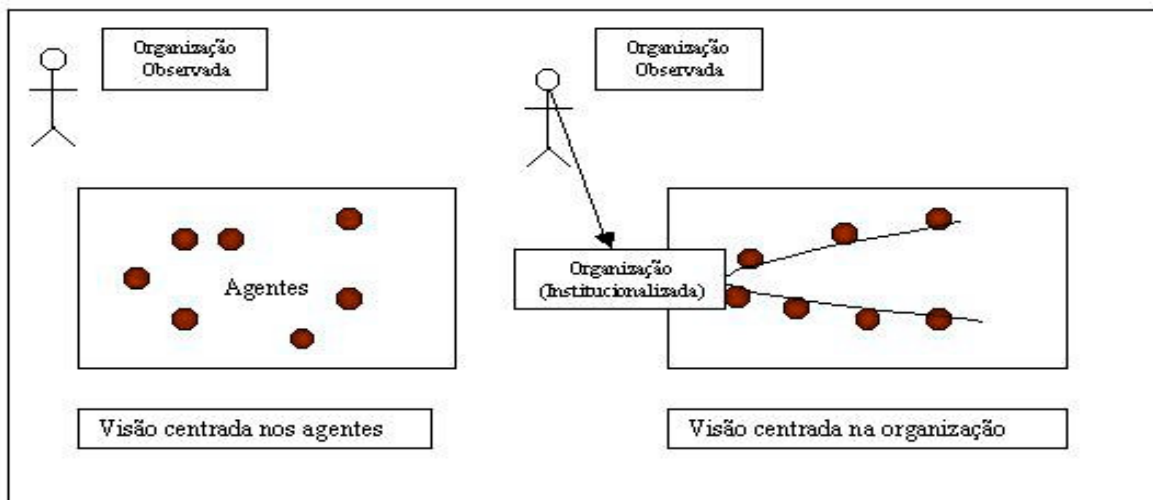


Figura 4: Visão centrada nos agentes e visão centrada na organização

Na visão centrada nos agentes existe apenas a organização dentro dos agentes, um observador do sistema multi-agente não tem acesso a esta visão e somente pode criar uma visão subjetiva da organização a partir do comportamento do sistema. Já na visão centrada na organização, existe uma definição da organização em que os agentes estão envolvidos e esta organização está fora dos mesmos. Neste caso, o observador do sistema pode conhecer a

organização do sistema analisando tanto a organização como observando o comportamento do sistema.

As duas visões de organização de sistemas multi-agentes vistas por [Lemaitre & Excelente, 1998] implicam na existência ou não de uma organização dentro da arquitetura. O fato de observar a arquitetura ilustra de forma melhor as duas visões. A visão centrada no agente mostra ao observador um sistema confuso onde a organização é presente apenas quando se observa a cada agente constante na rede. Ou seja, a organização existe apenas dentro de um agente e os mesmos agem segundo a sua organização sem necessariamente uma dependência de organização da rede em que estão envolvidos [Hubner, 2003].

No outro ponto de vista, o da arquitetura centrada na organização, o mesmo observador consegue uma descrição da organização que a sociedade está utilizando, sem necessariamente ter que observar o comportamento de cada agente separadamente. Esta organização é a chamada institucionalizada. Este tipo de organização pode ser encontrada, por exemplo, em uma empresa, indústria ou escola, ou seja, qualquer organização que seja regida por normas, manuais e organogramas. Um observador, ao deparar-se com uma estrutura destas, consegue obter a descrição da sociedade em que a mesma está envolvida. Obviamente que, ao observar um agente específico deste tipo de organização, é possível se construir uma descrição subjetiva da organização. Esta característica contribui para a complexidade das organizações, já que o processo de construção da organização observada é subjetivo e, portanto, não é único, o que indica que diferentes agentes podem construir diferentes visões sobre a mesma organização, ou seja, independente de estarem na mesma organização, agentes podem construir visões distintas da organização e o observador que atenta a um agente poderá possuir descrições distintas da mesma organização. Esta razão faz com que vários sistemas criem uma descrição objetiva (institucionalizada), assim como o caso do sistema multi-agente do atendimento pré-hospitalar, que tende a minimizar as diferenças das observações de cada agente.

Se para o observador existem duas situações para a mesma organização, do lado do agente também podem ser concebidas mais de uma situação: na primeira, o agente possui a capacidade de explicitar sobre a organização em que está inserido e na outra onde o agente não é capaz disto. O fato dos agentes terem ou não uma representação interna da organização onde estão envolvidos não implica que sua sociedade tenha ou não uma organização institucionalizada. Desta forma, existem, segundo [Lemaitre & Excelente, 1998], quatro tipos de sociedade:

- a) Agentes que não representam e não raciocinam sobre a organização observada – visão centrada nos agentes, sendo os agentes incapazes de gerar uma representação interna da sociedade.
- b) Agentes que representam e raciocinam sobre a organização observada – visão centrada nos agentes, tendo os agentes a capacidade de gerar a representação da sociedade.
- c) Agentes que não representam a organização institucionalizada - visão centrada na organização com agentes sem capacidade de gerar uma representação de sua organização.
- d) Agentes que representam a organização, seja observada ou institucionalizada - visão centrada na organização com agentes com capacidade de gerar uma representação de sua organização.

Esta divisão coloca um agente sob alguns aspectos. Se o agente possui capacidade de raciocinar sobre a organização em que está inserido, ele consegue tirar melhor proveito da organização. Se o agente está inserido em uma organização centrada no agente e pode raciocinar sobre a organização, o agente poderá colocar suas ações de acordo com o entendimento que obteve da organização. Se o agente não raciocina sobre a organização e este está envolvido em uma organização centrada no agente, ele terá suas ações colocadas apenas segundo suas necessidades, podendo a organização auxiliar segundo uma demanda solicitada por ele. Se o agente não raciocina sobre a organização e este está envolvido em uma sociedade centrada na organização, é esta última que determinará as ações do agente dentro da sociedade. Neste caso, quem usa e faz proveito da organização é o projetista do sistema [Baeijs, 1998]. Se o agente raciocina sobre a organização e esta inserido em uma sociedade centrada na organização, ele pode tirar proveito da organização da sociedade e elaborar melhor suas ações e determinações [Sichman, 1998].

### **3.5 Modelo de agente**

Algumas das definições mais importantes no desenvolvimento do sistema multi-agente do atendimento pré-hospitalar foi o modelo de agente que seria utilizado e a arquitetura que o mesmo estaria utilizando.

#### **3.5.1 Representação conceitual de um agente**

A figura 5 mostra as principais propriedades que devem existir em um agente (percepção, comunicação, ações, etc). Todavia, a maior parte dos modelos encontrados na

literatura são modelos que utilizam a percepção e a ação para as atividades de interações ou a comunicação direta entre os agentes.



Figura 5: Representação conceitual de um agente definido por Ferber [Ferber, 1995].

Antes de continuar a análise, é importante lembrar que a ambição das pesquisas em inteligência artificial é construir agentes autônomos para um ambiente real [Hayes-Roth, 1990].

Esta classe de sistemas computacionais é conhecida como agentes inteligentes e cujas tarefas requerem uma interação com as entidades dinâmicas do ambiente e o raciocínio baseado em conhecimentos. Esse tipo de agente executa tarefas complexas. Essas tarefas existem em vários outros domínios: controle de processos, monitoramento de tratamentos intensivos etc. Deve-se observar, neste exemplo, que a comunicação direta entre os agentes não é mencionada. Isto mostra que, neste tipo de situação, as pesquisas privilegiaram as comunicações indiretas através do ambiente.

A ambição de criar agentes autônomos capazes de viver em um ambiente real requer a construção de modelos bastante complexos de interações com o ambiente (percepção, cognição, ação). As Figura 6 e 7 mostram dois modelos de agentes que foram especificamente desenvolvidos para tais situações. Nestes modelos, os conhecimentos representados são essencialmente:

- os estados mentais internos de um agente (planos, ações); e
- o ambiente.

Os mecanismos implementados são:

- o raciocínio ou a inferência;

- o planejamento;
- a execução de um plano;
- a integração de um novo evento no mecanismo de raciocínio; e
- a adaptação do plano.

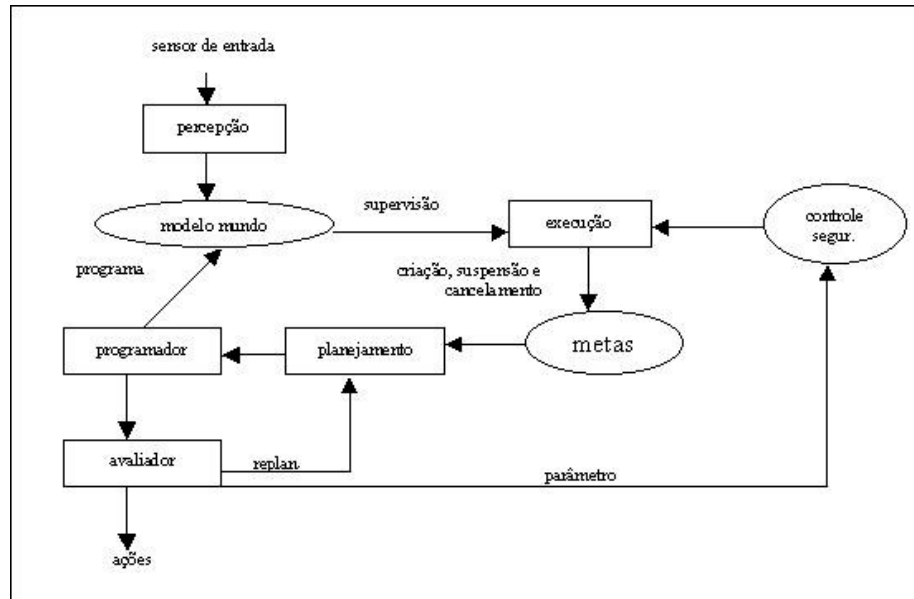


Figura 6: Modelo de agente definido por Bussmann e Demazeau [Bussman & Demazeau, 1994].

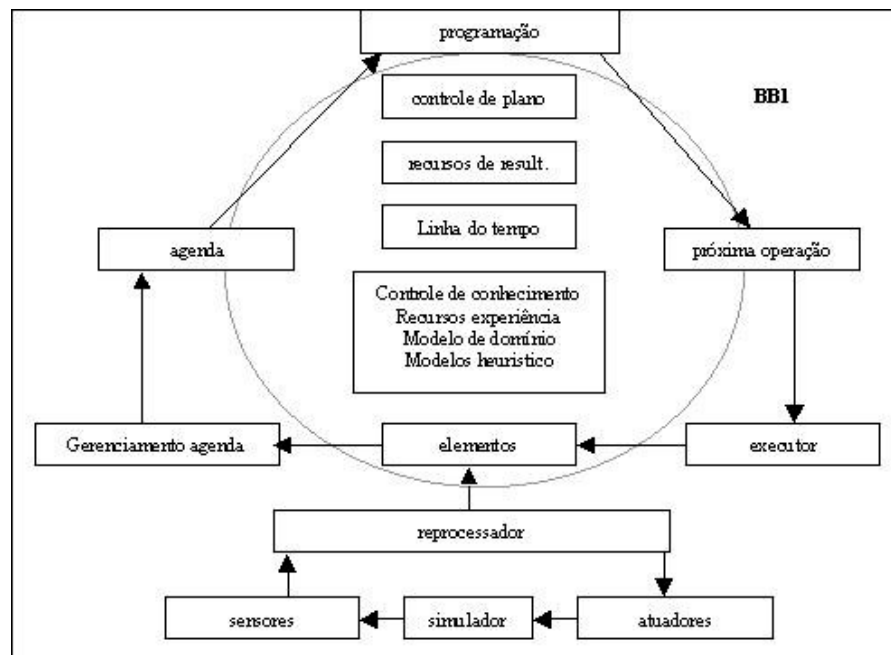


Figura 7: Modelo de agente definido por Hayes-Roth [Hayes-Roth, 1990].

As Figuras 6 e 7 representam modelos de agentes que estabelecem a integração de modelos reativos e cognitivos a fim de realizar sistemas do tipo percepção/cognição/ação. Eles concernem à organização dos processos internos de um agente e especificam as



interações entre os modelos que, no contexto de uma implementação real, devem assegurar um comportamento tanto reativo quando cognitivo.

A tarefa principal do modelo de um agente é descrever como os dados de entrada são tratados a fim de produzir uma saída tal como: sensor de entrada e ações (Figura 6). Com esse mesmo princípio, a percepção e a execução fazem referência ao processo necessário para interagir com o ambiente. Na Figura 6, o núcleo do agente é o modelo do mundo, o qual compreende conhecimentos sobre o ambiente, o estado mental dos outros agentes (não é dito como o agente que recupera o estado mental dos outros), e seu próprio estado mental. Este último inclui em particular os planos que devem ser executados. Finalmente, para assegurar a reatividade do agente, um processo de avaliação examina de modo contínuo o modelo do mundo, detectando assim as situações que um agente deve reagir, bem como avaliar e decidir as ações a executar [Scalabrin, 1996].

Pode-se observar que, mesmo nestes modelos específicos (a uma classe de problemas), existem certas diferenças, uma delas é o encadeamento dos módulos no interior dos modelos. Ou seja, o laço que leva em conta os eventos internos e externos, bem como a adaptação do comportamento de um agente em função desses eventos não é a mesma. Na Figura 6, um evento externo é levado em conta primeiramente pelo módulo *executor* e somente na seqüência pelos módulos planejamento e programador, o que não é o caso na Figura 7.

### 3.5.2 Interação via troca de mensagens

Tome-se como ponto de partida o modelo da Figura 8, que representa ARCHON [Jennings & Wittig, 1992]. Este último é um dos primeiros modelos de agentes utilizado no meio industrial. Este modelo é interessante porque ele separa claramente a aplicação do sistema de serviços que correspondem respectivamente aos blocos inferior e superior da Figura 8.

O sistema de serviços corresponde ao conjunto de mecanismos da arquitetura que não faz parte de nenhum domínio de aplicação. Ele fornece facilidades de comunicação, tais como:

- o endereçamento de mensagens;
- a filtragem; e
- a difusão de informações dirigida por interesses (Publish/Subscribe).

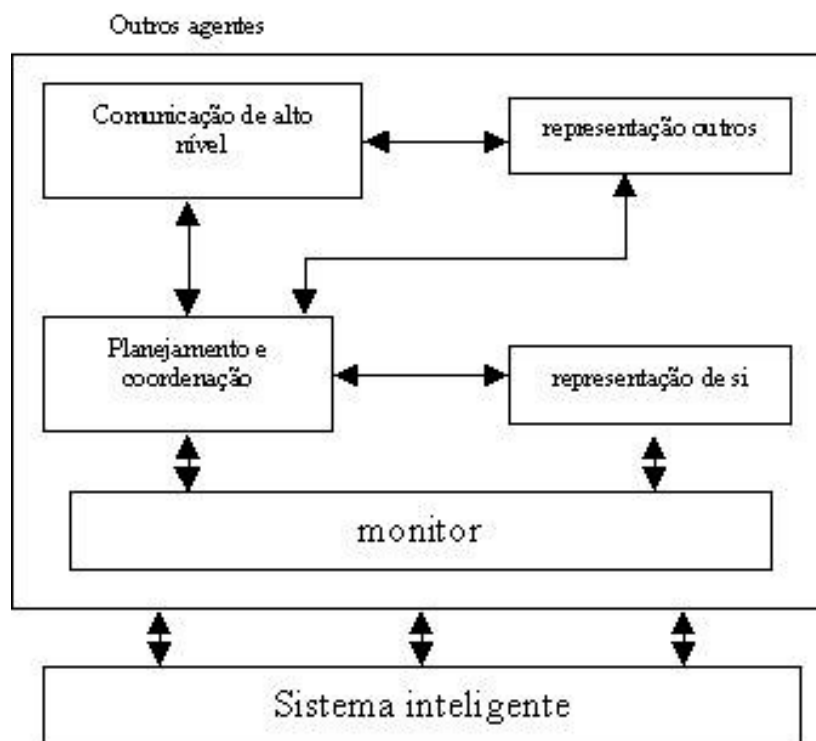


Figura 8: Modelo de agente ARCHON definido por Jennings e Wittig [Jennings & Wittig, 1992].

Foram incluídos também certos modelos de representação e de interação:

- a representação de si;
- a representação dos outros ("acquaintances"); e
- a interação via protocolo cliente-servidor ou redes contratuais (*Contract-Net*).

A representação dos outros contém modelos para efetuar a comunicação em termos de competências e de interesses. Esses modelos são pré-requisitos para todas as atividades de coordenação entre agentes. Os serviços de difusão de informações utilizam esses modelos para encontrar os agentes interessados por um determinado resultado. Da mesma maneira, o modelo de si representa, de modo abstrato, o domínio do agente ou da aplicação. A parte aplicação contém, antes de tudo, informações sobre o estado corrente do sistema (tarefas em execução). Ela encapsula também os planos pré-compilados para a parte reativa do controle. Esses planos são acessíveis pelo monitor, que é responsável pelo controle da aplicação e das trocas de informações entre as partes. O subsistema de planejamento e coordenação representa a principal parte reflexiva do sistema de serviços. Se uma exceção ocorre, é tarefa desse subsistema raciocinar e encontrar uma solução. Sua influência sobre o monitor ocorre principalmente através do comportamento armazenado no modelo de si. Este contém também partes reativas a respeitar no momento da inicialização da cooperação. Por exemplo, o agente

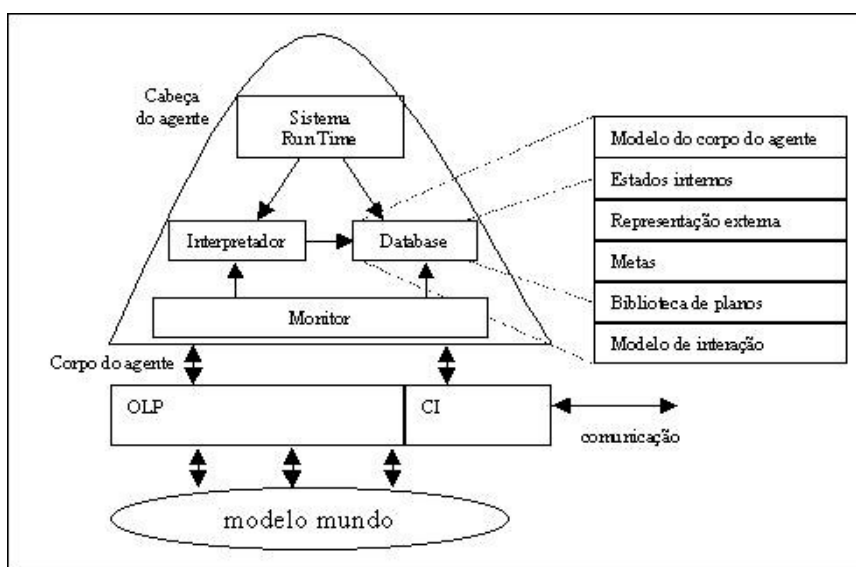
detecta a necessidade de cooperação e determina qual tarefa ele deve confiar aos outros agentes. Deve-se frisar que a seleção do protocolo de cooperação a utilizar é feita em função da carga de trabalho do sistema de raciocínio. O processo é o mesmo quando um agente recebe uma demanda de cooperação da parte de um outro agente [Scalabrin, 1996].

A figura 9 ilustra uma arquitetura similar à arquitetura ARCHON, exceto que a separação entre o sistema de serviços e a aplicação é realizada pelo módulo de comunicação. No entanto, a modelagem inclui um conjunto de modelos bastante completo em relação às outras arquiteturas. Ou seja, ela possui uma base de dados onde são armazenados:

- estados internos;
- representação externa;
- metas (parecidas com "scripts");
- biblioteca de planos; e
- modelo de intenções.

Ela gerencia uma biblioteca de planos, os quais podem ser: privados, não privados e a especificar o executor. Cada plano contém um campo onde é inserido o nome do executor do plano. Assim, se o valor desse campo é:

- *self*, trata-se de um plano privado;
- *uma constante* (nome de um outro agente), trata-se de um plano não privado;
- *uma variável*, trata-se de um plano em que se deve determinar o seu executor.



.Figura 9: Modelo de agente definido por Hägg & Ygge [Hagg & Ygge, 1994].

Quando um objetivo é criado, um mecanismo de raciocínio procedimental é ativado a fim de ligar esse objetivo ao plano de execução correspondente. Em caso de sucesso, o interpretador invoca o plano [Scalabrin, 1996].

A Figura 10 mostra um modelo de agente (denominado GA<sub>g</sub>), defendido por Scalabrin [Scalabrin et al, 1996], construído para ser o mais genérico possível. Este agente possui um conjunto mínimo de características independentes do domínio de aplicação, e realiza algumas operações elementares. Ele fornece mecanismos de gestão das entradas/saídas e de eventos assíncronos, bem como um conjunto de modelos. Este último permite que um agente construa, de maneira incremental, uma visão de seu mundo externo. Um conjunto de comportamentos por "defaults" também é previsto. Estes últimos concernem à política de difusão de informações e às reações do agente com alguns eventos, tais como o recebimento de mensagens.

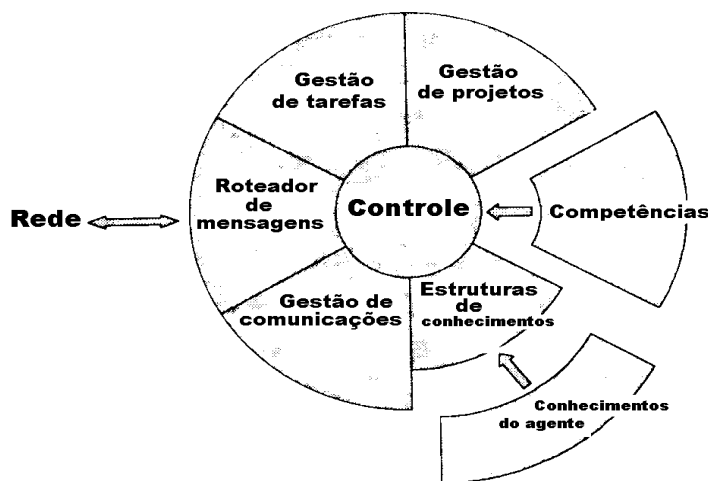


Figura 10: Modelo de agente definido por Scalabrin et al. [Scalabrin et al, 1996].

Como visto, um modelo de agente pode ser concebido com uma divisão operacional de uma estrutura capaz de suportar uma base de conhecimento, uma camada de comunicação, uma memória externa, uma memória interna e uma central de processamento. A maioria dos modelos conhecidos, já citados anteriormente, utiliza uma divisão parecida, pois a necessidade de operação de um agente é sempre a mesma. Ou seja, sempre um agente deverá interpretar as mensagens que recebe e formatar as mensagens que envia, deverá processar as solicitações recebidas pelas mensagens e consultando a uma base de conhecimentos, pois um agente existe para uma determinada função e poderá armazenar informações sobre si mesmo e informações sobre a rede ou sociedade em que está envolvido. Desta forma, todas as propostas de arquitetura de agente não fogem muito deste modelo, pois o funcionamento de um agente depende no mínimo destes quesitos.

O modelo de agente da Figura 11 é baseado nos modelos apresentados anteriormente e traz o modelo escolhido para o desenvolvimento do sistema de agentes do atendimento pré-hospitalar, apenas com a retirada da camada de interface com o usuário, que, para este caso, será realizada através de aplicativos externos.

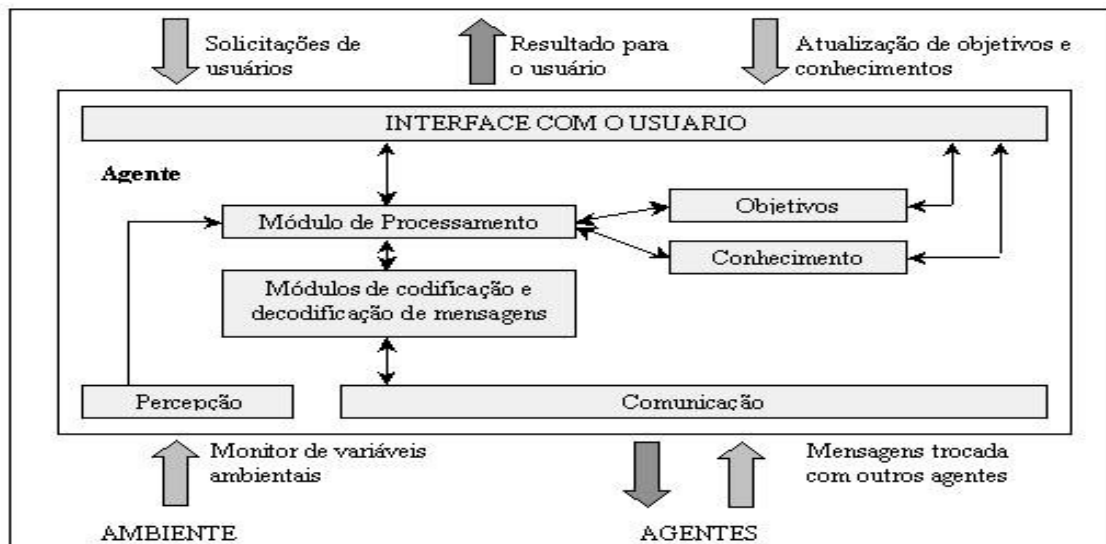


Figura 11: Modelo de agente utilizado

### 3.5.3 Componentes do modelo de agente

Interface com o Usuário: Responsável pelo recebimento das solicitações do usuário. Após a recepção, as solicitações devem ser encaminhadas ao módulo de processamento para, depois, apresentar os resultados ao usuário. Uma de suas funcionalidades é prover meios que permitam ao usuário atualizar os objetivos e os conhecimentos do agente.

Módulo de Processamento: Está constantemente ligado à base de conhecimento do agente. É o módulo responsável por responder todas as solicitações da camada de comunicação e da interface com o usuário. Dentro de seu trabalho está a necessidade de avaliar as diferentes alternativas de solução e selecionar a melhor opção. Pode, com base nas mensagens recebidas de outros agentes e através de sua capacidade de percepção das variáveis ambientais, atualizar sua base de conhecimento. O comportamento reflete todos os estados possíveis na interação entre agentes.

Objetivos: Todo agente possui um objetivo definido. Este objetivo traz a responsabilidade que possui dentro da sociedade em que esta envolvido.

Base de Conhecimentos: Todo conhecimento do agente está armazenado em sua base de conhecimento. É de onde o módulo de processamento tirará apoio para apresentar as soluções exigidas pela sociedade.

Módulo de Codificação e Decodificação de Mensagens: Todas as mensagens recebidas e enviadas pelos agentes estarão em um determinado formato, dependendo de qual padrão ou ferramenta de comunicação a rede está utilizando. Desta forma, o módulo de codificação e decodificação de mensagens é o módulo responsável por receber e enviar as mesmas dentro do padrão destinado. As mensagens são trocadas com os demais agentes da rede. Como exemplo de linguagem de comunicação entre agentes, pode-se citar a KQML, que será abordada com maiores detalhes na próxima sessão.

Módulo de Percepção: Refere-se aos meios disponíveis para o agente monitorar variáveis do ambiente.

Módulo de Comunicação: Encarrega-se de enviar e receber mensagens de outros agentes mediante protocolos de transporte (p.e. na Internet, via TCP e HTTP). Este módulo pode desenvolver-se mediante KAPI (KQML Application Programmer's Interface), que provê uma interface abstrata entre os protocolos de transporte [Torres & Constantino, 1996].

### **3.6 Linguagem de comunicação agente**

O pressuposto de uma arquitetura diz que deve existir um meio de comunicação entre as entidades que a compõem. Para a arquitetura do atendimento pré-hospitalar foi utilizado a linguagem KQML (Knowledge Query Manipulation Language), responsável por realizar as trocas de informações e conhecimentos entre os agentes, caracterizando-se tanto por uma formatação de mensagens como por um protocolo para manipular essas mensagens que suportam, em tempo de execução, o compartilhamento de conhecimento e informações entre agentes participantes da arquitetura [Finin & Labrou, 1997].

A definição do KQML traz um conjunto de mensagens pré-definidas, denominadas *performativas*, as quais definem um conjunto de operações a serem executadas dentro da arquitetura. Para tanto, todo agente participante deverá necessariamente entender e manipular os padrões definidos pelo KQML, respondendo assim toda solicitação expedida, independente da estrutura do seu emissor. A necessidade da utilização de um protocolo independe da forma que o agente trabalha internamente é a característica que possibilita a integração de entidades distintas em sua construção, porém que possuem informações que possam auxiliar o funcionamento de uma rede única.

Em suma, o KQML é um protocolo formado por mensagens pré-formatadas com significado e funções bem específicas. Um agente, ao receber uma mensagem KQML, toma uma ação com base no seu significado e conteúdo [Finin & Labrou, 1996].

Na especificação do KQML, os agentes participam através da troca de mensagens e podem ser comparados a entidades que possuem comportamento autônomo, como o são, agindo em sociedade através da interação com outros agentes que também agem segundo suas crenças. As crenças dos agentes representam as informações sobre eles próprios e sobre o ambiente externo, incluindo as bases de conhecimentos de outras entidades [Finin & Labrou, 1992b].

O KQML não soluciona ou sugere como deve estar estruturada a implementação dos agentes, ele apenas identifica-se como uma linguagem de propósito geral, onde, independente do ambiente ou estrutura dos agentes, é possível usá-la para prover a comunicação em um sistema multi-agente [Finin & Labrou, 1996; Finin & Fritzson, 1992a].

O conjunto de mensagens KQML pode ser estendido desde que às novas performativas criadas obedeçam à mesma forma da especificação original da linguagem. Os agentes que estão de acordo com o KQML não precisam reconhecer todas as mensagens, de forma que um pequeno subconjunto pode ser suficiente num determinado sistema multi-agente, ou seja, dependendo da necessidade, pode-se escolher somente algumas performativas a serem utilizadas na comunicação [Finin & Labrou, 1992a].

O KQML, em sua estrutura, possibilita o encapsulamento de uma linguagem conhecida pelos demais agentes, deixando livre para a arquitetura definir de que forma as informações serão trocadas. Esta liberdade torna este protocolo potencialmente utilizável na arquitetura do atendimento pré-hospitalar, onde existe uma terminologia particular.

A estrutura do KQML está ligada ao formato de suas mensagens. Toda mensagem KQML possui uma área reservada ao conteúdo que está sendo transmitido. Este conteúdo possui o formato que os agentes da arquitetura se adaptam melhor. O conteúdo de uma *performativa* pode ser baseado, por exemplo, em linguagem natural. É importante salientar que todos os agentes da arquitetura devem necessariamente entender, ou melhor, interpretar o conteúdo das mensagens que são trocadas [Finin & Labrou, 1996].

O conteúdo sempre diz respeito a uma determinada ação, informação ou conhecimento do agente. Porém, toda mensagem possui um sentido ao qual deve ser aplicado, ou seja, sempre que um agente recebe uma mensagem de outro, esta pode ser facilmente identificada pelo seu tipo. De acordo com o tipo da mensagem, um agente pode identificar rapidamente qual a ação solicitada. Como uma *performativa* se associa a um conteúdo variável, são necessários outros parâmetros para tornar possível uma comunicação, tais como a linguagem com a qual o conteúdo foi escrito e uma ontologia responsável pelo domínio que a linguagem utiliza. Para a concretização da mensagem, ainda são necessários alguns parâmetros onde

devem estar identificados o emissor, o receptor e os identificadores de cada mensagem trocada.

Para simplificar o estudo da comunicação entre os agentes, serão considerados apenas três níveis distintos, como ilustra a Figura 12. Ao criar uma relação com o modelo de referência OSI (Open Systems Interconnection), os dois níveis mais altos se situam na camada aplicação.

<b>Nível 3</b>	<b>Protocolo de interação</b>
<b>Nível 2</b>	<b>Linguagem de comunicação</b>
<b>Nível 1</b>	<b>Protocolo de transporte</b>

Figura 12: Elementos para a comunicação entre agentes.

O protocolo de comunicação (nível três) corresponde à estratégia de alto nível utilizada por um agente a fim de gerir suas interações com os outros. Um protocolo como este pode abrigar um esquema de negociação ou protocolos derivados da teoria de jogos. A linguagem de comunicação (nível dois) é o meio pelo qual um agente comunica sua atitude ao conteúdo de uma comunicação. É a linguagem de comunicação que "diz" se o conteúdo da comunicação é uma asserção ou uma requisição. O protocolo de transporte (nível um) é o mecanismo real de transporte utilizado para a comunicação, por exemplo, ELM, TCP, UDP, HTTP, SMTP.

Existem várias linguagens de comunicação. ACL (Agent Communication Language) de Mayfield et al. (1996) é um exemplo. Ela permite que os agentes troquem conhecimentos utilizando interfaces declarativas. Uma mensagem ACL é uma mensagem KQML, composta de uma diretiva de comunicação (ou ato da fala) e de um conteúdo semântico expresso em termos de um vocabulário codificado em um formato de troca. Cada mensagem ACL tem três componentes:

- um vocabulário ligado à semântica própria de um domínio (ontologia),
- uma linguagem para codificar o conteúdo da mensagem, e
- uma linguagem de comunicação, por exemplo, KQML.

De forma simples, a estrutura do KQML ainda pode ser dividida em três camadas [Finin, 1996]:



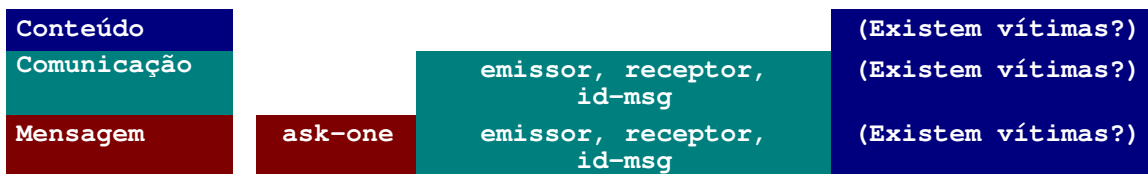


Figura 13: Camadas da linguagem de comunicação KQML.

A camada de conteúdo possui a mensagem e pode ter no seu conteúdo qualquer representação de linguagem. A camada de comunicação codifica um conjunto de características para aceitar parâmetros de alto nível, tais como a identidade de quem envia e recebe a mensagem e também um identificador único associado com a mensagem. A camada de mensagem codifica a mensagem que um agente deseja transmitir. Este nível forma o núcleo do sistema KQML. Ele determina o tipo de interação que um agente pode ter com seu correspondente. A função principal desse nível é identificar o protocolo a ser utilizado para expedir uma mensagem e fornecer uma *performativa* (e.g., “ask-one”), que será anexada ao conteúdo. O conteúdo não é visível pela linguagem de comunicação do agente. Este nível inclui também características operacionais que descrevem a linguagem do conteúdo e a ontologia. Uma vez que a mensagem é codificada, ela é passada ao sistema de transporte para ser expedida (Figura 12). Deve-se salientar que KQML não é uma linguagem homogênea, porque ela fornece uma linguagem para o conteúdo e não para os demais parâmetros da mensagem.

O KQML, como já mencionado, é projetado para ser usado com vários mecanismos (atualmente há implementações que usam TCP/IP, SMTP (email), HTTP e CORBA). No caso do atendimento pré-hospitalar, se utilizará HTTP. O ambiente operacional para agentes KQML pode ser altamente distribuído, heterogêneo e extremamente dinâmico. Para satisfazer as exigências de tal ambiente, a linguagem KQML provê ferramentas formais que trabalham com outras linguagens e protocolos [Finin & Labrou, 1996].

Apesar de atender às principais características de uma linguagem de comunicação agente, KQML ainda não tratou adequadamente as questões de segurança e autenticação em suas mensagens.

Com estas observações, as mensagens KQML tornam-se capazes de atender aos requisitos impostos pela arquitetura do atendimento pré-hospitalar, tornando a comunicação eficiente e principalmente possível entre os diversos órgãos envolvidos.

### 3.7 Redes Contratuais

Um dos protocolos de maior influência na área de inteligência artificial distribuída é o *Contract Net* [Smith, 1980; Smith, 1981].

No *Contract Net* não há um controle centralizado e os agentes podem cooperar através do compartilhamento de tarefas [Marietto, 2000]. É um protocolo para cooperação entre os agentes e para a solução de problemas por comunicação. É modelado sobre os mecanismos que governam trocas de mercadorias e serviços. Provê uma solução para o chamado problema da conexão: encontrar um agente apropriado para fazer uma determinada tarefa [Eudemia, 2001].

Os agentes compartilham tarefas quando a tarefa é muito extensa ou o agente não tem conhecimento necessário para realizar uma determinada tarefa. Uma solução para tarefas extensas é o particionamento em subtarefas, para que se possa alocar estas subtarefas a outros agentes. Se o agente não tem conhecimento necessário, ele procura outros agentes que estão aptos a executar ou gerenciar a tarefa a ser realizada.

Para que se possa compartilhar as tarefas, é necessário que estas tarefas sejam distribuídas entre os agentes, e para isso o *Contract Net* utiliza um processo de negociação e estabelecimento de contratos. Como definido em [Smith, 1980], um contrato é um acordo explícito entre um agente que gera uma tarefa (chamado *Manager*) e um agente que executará a tarefa (chamado *Contractor*). O *Manager* é responsável por monitorar a execução de uma tarefa. Os agentes não são designados a priori como *Manager* ou *Contractor*, gerindo assim um sistema sem controle hierárquico.

Para que se possa determinar qual agente é o *Contractor*, é necessário estabelecer um contrato após um processo de negociação, onde o *Manager* de uma determinada tarefa avisa a existência de uma tarefa a outros agentes. Aqueles agentes que estão livres de qualquer tarefa avaliam os anúncios feitos pelos *Manager*. Se algum *Contractor* tiver potencial para realizar uma ou mais tarefas que foram anunciadas pelo *Manager*, é enviada uma proposta a este *Manager*, que tem como uma de suas funções analisar todas as propostas e escolher o agente mais rápido para realizar a tarefa.

Do ponto de vista do *Manager*, o processo é:

- anunciar uma tarefa que precisa ser executada;
- receber e avaliar as ofertas dos *Contractor* potenciais;
- decidir por um *Contractor* conveniente e envia o contrato para ele;
- receber e sintetizar os resultados.

Do ponto de vista do *Contractor*, o processo é:

- receber o anúncio de uma tarefa;
- avaliar sua própria capacidade de respondê-la;
- responder negando ou fazendo uma oferta;
- executar a tarefa se sua oferta é aceita;
- enviar os resultados.

O início de uma comunicação entre organizações é o princípio de um sistema multi-agente. Não se integra organizações distintas de maneira dinâmica sem se estabelecer uma forma de cooperação entre estas. A construção de um sistema multi-agente necessita de um conjunto de ferramentas e padrões que sejam capazes de possibilitar a integração das organizações que venham a participar de um atendimento. Além das ferramentas e padrões descritos, existe a forma que estas organizações estarão atuando. Para modelar esta arquitetura será utilizada uma linguagem de modelagem de agente que será apresentada no próximo capítulo.

# Capítulo 4 – Linguagem de modelagem de agentes

## 4.1 Introdução

O desenvolvimento de aplicações com o uso da tecnologia agente requer técnicas que reduzam o risco inerente em qualquer tecnologia nova. Dois modos de reduzir os riscos potenciais são:

- apresentar uma nova tecnologia com base em métodos já existentes, conhecidos e aceitos; ou
- suportar métodos que sejam utilizados atualmente.

A tecnologia presente nos agentes deve possuir a preocupação de tratar os agentes como objetos ativos, que possuam a capacidade de autonomia (capacidade de tomar iniciativa e ter um controle sobre as suas próprias ações ou ainda a capacidade de tomar ações para realizar algumas tarefas e objetivos, sem a interferência do usuário final). Deve haver um elemento de independência no agente [Franklin & Graesser, 1997]. Um agente autônomo deve manter uma agenda independentemente de seu usuário. Isto requer aspectos de ação periódica, execução espontânea e iniciativa, na qual o agente deve ser capaz de tomar ações independentes que irão eventualmente beneficiar o usuário [Foner, 2004], ou seja, operam sem a intervenção direta de seres humanos ou outros agentes [Castelfranchi, 1995]. "Um sistema capaz de interagir independente e efetivamente com seu ambiente através de seus próprios sensores, com o objetivo de realizar alguma tarefa externa ou gerada por ele próprio." [Davidsson, 1995]. Este controle deve ser relacionado somente ao usuário final quando define que, um agente inteligente deve possuir a habilidade de praticar ações para desenvolver tarefas ou alcançar objetivos, sem necessitar da interferência do usuário final [Nissen, 1995]. Além de possuir controle sobre seu comportamento, deve também possuir controle sobre seu estado interno [Franklin & Graesser, 1997]. Além da possibilidade de ser um agente autônomo, a tecnologia deve permitir que um agente possua mobilidade, raciocínio, inteligência, habilidade social, entre outras. Estas capacidades são extensões de um agente básico.

Os métodos para desenvolvimento de software dependem de representações e normas que sustentem a análise, especificação e projeto de um software agente. Estas três fases de

desenvolvimento de software exigem que este desenvolvimento seja baseado no ciclo de vida de um software. As habilidades dos desenvolvedores são enfocadas mais em metodologia de desenvolvimento que em acompanhamento das técnicas de agente mais recentes.

A UML é a linguagem amplamente aceita para a concepção de software orientado a objeto. O princípio básico da linguagem proposta para agentes é acomodar os requisitos dos agentes dentro da UML.

Estarão sendo apresentados neste capítulo as abordagens necessárias para adaptar os modelos da UML em uma metodologia que tratará da modelagem de um sistema de agentes.

## 4.2 Metodologias

A AUML (Agent UML) sintetiza alguns elementos importantes de uma metodologia baseadas em agentes.

A comunidade de pesquisa e desenvolvimento de agente está interessada em métodos e ferramentas representacionais para apoiar os softwares desenvolvidos com a utilização de agentes. Este tema pode ser consultado em [Burmeister, 1996; Iglesias, 1998; Garijo, 1999; Parunak, 1998; Kinny, 1996a; Kinny, 1996b; Bryson, 1998; Martin & Odell, 1998; Nodline et al, 1998; Wooldridge et al, 2000].

Vários grupos têm divulgado metodologias para projetar agentes, baseados sempre em mecanismos representacionais. O paralelo que é realizado entre as metodologias de projeto para agentes e para objetos é compartilhado por vários autores, por exemplo [Burmeister, 1996; Bryson, 1998].

A metodologia aplicada por Wooldridge et al (2000) inclui recomendações específicas para especificações que sustentam o resumo de um protocolo de alto nível, como uma unidade atômica, uma especificação é refletida nesta metodologia. O programa em andamento na *Universidade Livre de Amsterdã* em metodologias composicionais para requisitos [Herlea, 1999], projeto [Brazier, 1998], e verificação [Jonker, 1997] usa representações gráficas com vínculos fortes para diagramas de colaboração da UML. A discussão da composição de protocolos é antecipada no trabalho de Burmeister et al. [Burmeister et al, 1993]. Os gráficos de Dooley facilitam a identificação de um agente desempenhando um papel específico [Parunak, 1996; Singh, 1998a].

Esta larga atividade é um sinal saudável de que sistemas baseados em agentes estão tendo uma evolução significativa. Esta demanda por metodologias e softwares reflete a importância industrial crescente desta tecnologia. O objetivo não é competir com quaisquer

destes esforços, mas estender e aplicar um formalismo de modelagem extensamente aceito e representacional.

#### 4.2.1 UML

Durante os anos setenta, a programação estruturada era a abordagem dominante para desenvolvimento de software. Junto com isto, tecnologias de engenharia de software eram desenvolvidas a fim de aliviar e formalizar o desenvolvimento de sistemas conforme um ciclo de vida definido: planejamento, análise e projeto, e finalmente a construção de um sistema, teste e manutenção. Na década de oitenta, as linguagens orientadas a objetos surgiram, trazendo com elas novos conceitos, como encapsulamento de dados, herança, mensagens, polimorfismo e outros. Ao final dos anos oitenta e início dos anos noventa, uma grande quantidade de abordagens de modelagem surgiram e passaram a sustentar a esfera comercial de softwares orientados a objetos. Para unificar estas várias abordagens, uma força tarefa de Análise e Projeto era estabelecida em 29 de junho 1995 dentro do OMG. Por volta de novembro de 1997, um padrão era adotado pela OMG, chamada pelos seus membros de Linguagem de Modelagem Unificada (UML).

A UML unifica e formaliza os métodos de muitas abordagens para o ciclo de vida de desenvolvimento de software orientado a objetos, inclusive para Booch, Rumbaugh (OMT), Jacobson, e Odell [Odell, 1999; Bauer, 1999]. A UML suporta os seguintes tipos de modelos:

- **Diagramas Estruturais:** existem para visualizar, especificar, construir e documentar os aspectos estáticos de um sistema. Considerem-se os aspectos estáticos como uma representação de seu esqueleto relativamente estáveis. Assim como os aspectos estáticos de uma casa incluem a existência e a colocação de itens como paredes, portas, janelas, canos, fios e outros, também os aspectos estáticos de um sistema de software abrangem a existência e a colocação de itens como classes, interfaces, colaborações, componentes e nós. Os diagramas estáticos da UML são organizados em função dos principais grupos de itens encontrados na modelagem de um sistema. 1. Diagrama de classe: mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos. Os diagramas de classes são os diagramas mais encontrados em sistema de orientação a objetos; 2. Diagrama de objetos: mostra um conjunto de Objetos e seus relacionamentos. Usa-se este diagrama para ilustrar as estruturas de dados, registros estáticos de instâncias dos itens encontrados no diagrama de classes; 3. Diagrama de componentes: mostra um conjunto de componentes e seus relacionamentos. Usa-se este diagrama para representar a visão estática da

implementação de um sistema; 4. Diagrama de implantação: mostra um conjunto de nós e seus relacionamentos. Usa-se este diagrama para ilustrar a visão estática da implantação de uma arquitetura [Rumbaugh & Jacobson & Boock, 1999].

- **Diagramas Comportamentais:** Os cinco diagramas comportamentais da UML são utilizados para visualizar, especificar, construir e documentar os aspectos dinâmicos de um sistema. Considerem-se os aspectos dinâmicos de um sistema como uma representação de suas partes que sofrem alterações. Assim como os aspectos dinâmicos de uma casa abrangem a circulação de ar e a passagem de pessoas pelos cômodos da casa, também os aspectos dinâmicos de um sistema de software envolvem itens como o fluxo de mensagens ao longo do tempo e a movimentação física de componentes em uma rede. Os diagramas comportamentais são basicamente organizados a partir das principais maneiras disponíveis para se fazer a modelagem da dinâmica de um sistema. 1. Diagrama de caso de uso: mostra um conjunto de casos de uso, atores e seus relacionamentos. Aplica-se este diagrama para ilustrar a visão estática do caso de uso de um sistema; 2. Diagrama de seqüência: é um diagrama de interação que dá ênfase à ordenação temporal das mensagens. Um diagrama de seqüência mostra o conjunto de objetos e as mensagens enviadas e recebidas por estes objetos. 3. Diagrama de colaboração: é um diagrama de interação que dá ênfase à organização estrutural dos objetos que enviam e recebem mensagem. Um diagrama de colaboração mostra um conjunto de objetos, as ligações existentes entre estes objetos e as mensagens enviadas e recebidas pelos objetos. 4. Diagrama de Estados: mostra uma máquina de estados, que consiste de estados, transições, eventos e atividades. Usa-se o diagrama de estados para ilustrar a visão dinâmica de um sistema. 5. Diagrama de atividades: mostra o fluxo de uma atividade para outra em um objeto. Um diagrama de atividade mostra um conjunto de atividades, o fluxo seqüencial ou ramificado de uma atividade para outra e os objetos que realizam ou sofrem ações [Rumbaugh & Jacobson & Boock, 1999].

A proposta aqui é uma extensão baseada em UML para representar agentes com os seguintes produtos: pacotes, templates, diagrama de seqüência, diagramas de colaboração, diagramas de atividade e diagrama de estados [Brazier, 1998; Herlea, 1999; Jonker, 1997; Kinny, 1996a; Kinny, 1996b; Wooldridge et al, 2000].

### 4.2.2 AUML

Comparados à abordagem tradicional de objetos, agentes são autônomos e interativos. Baseados em estados internos, suas atividades incluem planos e condições que guiam à execução de tarefas definidas. Enquanto as necessidades dos objetos são atendidas pelos seus métodos, os agentes percebem as condições e planejam suas ações, tratando suas necessidades com maior responsabilidade. Além disso, agentes podem agir só e com outros agentes. Os sistemas multi-agente podem se assemelhar a uma comunidade social de membros interdependentes que agem individualmente. Porém, ainda não existe nenhum formalismo para especificar o desenvolvimento deste tipo de sistema. Para empregar a programação baseada em agente, deve haver uma técnica de especificação que sustente o processo de engenharia de software inteiro, desde o planejamento, análise, projeto, construção, implantação e manutenção. A FIPA e o OMG Agent Work Group estão explorando o uso e recomendando extensões da UML [Bauer, 1999; Odell, 1999]. Depke et al [Depke et al, 2000] discute transformações de gráficos e papéis em uma base para modelagem UML de agente. Nesta proposta, a UML pode ser usada como um Protocolo de Interação de Agente (AIP), como mostra a Figura 14.

Este subconjunto foi escolhido porque protocolos de interação são complexos o suficiente para ilustrar o uso de AUML. Os protocolos de interação de agente são um bom exemplo de padrões de software em um contexto prático. Uma especificação de um protocolo de interação de agente provê um exemplo ou analogia para resolução de problemas em análise de sistema e projeto. A AUML é uma técnica para especificação de protocolo de interação de agente com a semântica formal e intuitiva e uma anotação gráfica. A semântica permite uma definição precisa, que também é utilizável no processo de engenharia de software. A anotação gráfica provê uma linguagem comum para comunicação de protocolo de interação de agente — um protocolo de interação de agente descreve um padrão de comunicação como uma seqüência permitida de mensagens entre agentes e as restrições no conteúdo destas mensagens [Odell, 1999].



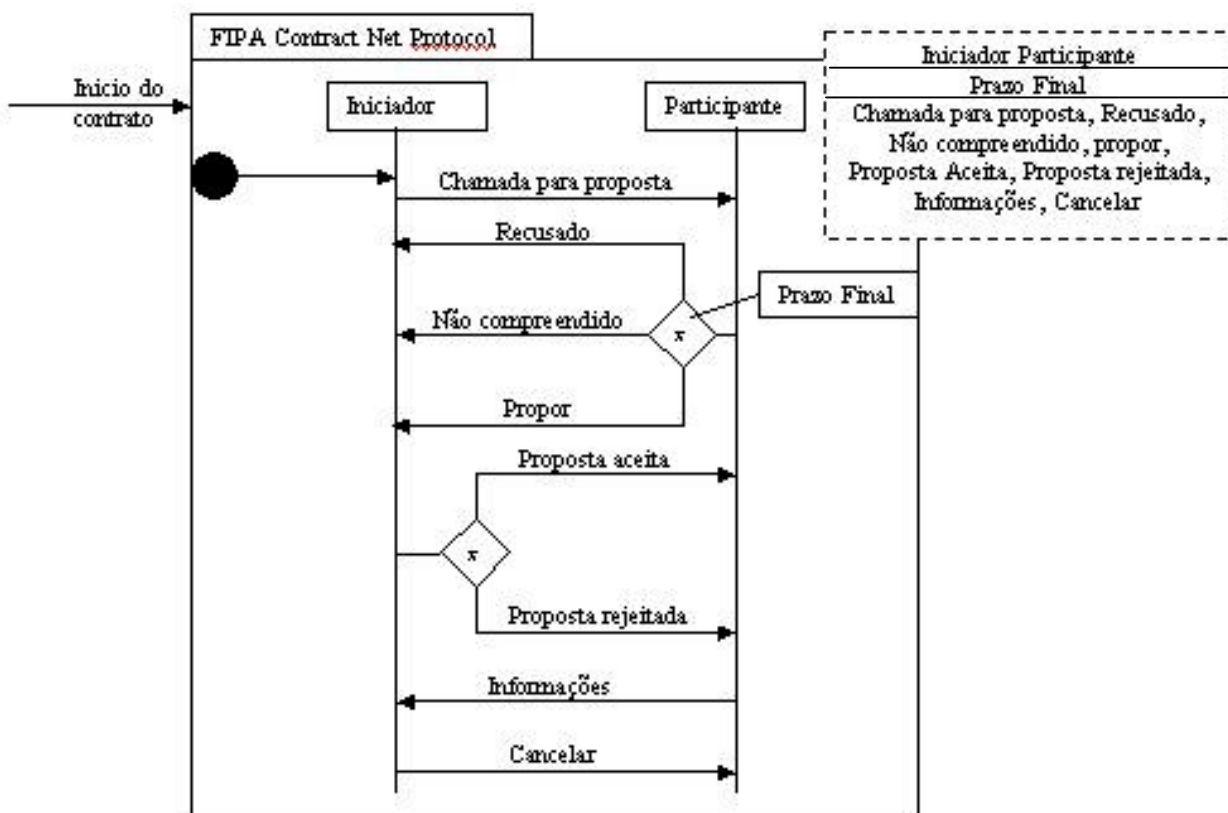


Figura: 14 Protocolo de Interação de Agente, mostrando as interações para o protocolo Contract Net.

A Figura 14 mostra um protocolo expresso como em diagrama de seqüência AUMML para o Contract Net Protocol. Quando invocado, um agente Iniciador envia a proposta para um agente que está disposto a participar e prover uma proposta. O agente Participante pode então escolher o que responder para o Iniciador antes de um prazo final: recusando, submetendo uma proposta, ou indicando que não entendeu. (O símbolo de diamante indica uma decisão que pode resultar em zero ou mais comunicações). O “x” dentro do diamante indica que deverá existir apenas uma resposta. Se uma proposta é oferecida, o Iniciador tem a escolha de aceitar ou rejeitar a proposta. Quando o Participante receber uma aceitação de proposta, informará o Iniciador sobre a execução da proposta. Adicionalmente, o Iniciador pode cancelar a execução da proposta em qualquer hora.

A Figura 14 também expressa mais dois conceitos representados no nível superior do quadro. Primeiro, o protocolo como um todo é tratado como uma entidade. A pasta de anotação na parte superior indica que o protocolo é um *pacote*, uma agregação conceitual de interação. Segundo, o *pacote* protocolo pode ser tratado como um padrão que pode ser customizado para domínios de problemas análogos. A caixa no canto superior direito expressa este padrão como uma *especificação de template* que identifica as entidades dentro do *pacote*

que precisam estar descritas quando o template estiver sendo instanciado.

Em UML, o protocolo de interação de agente serve como pacote e template. Um template é um módulo parametrizado que pode ser instanciado por outro produto em uma outra relação de tempo. Na Figura 14, a caixa pontilhada no lado superior direito indica que o pacote é um template. Os parâmetros desconectados na caixa são divididos pelas linhas horizontais em três categorias: parâmetros de papel, restrições e atos de comunicação.

### 4.3 Utilização de Padrões – Nível 1

Padrões são idéias reaproveitadas em um contexto prático e que podem provavelmente ser utilizadas em outros. Como tal, fornecem exemplos ou analogias que podem ser usadas como soluções para problemas em análise de sistema e projeto. Protocolos de interação agente são, então, projetados como soluções reutilizáveis que podem ser aplicadas aos vários tipos de seqüência de mensagem que são encontradas entre os agentes. Existem duas técnicas de UML que melhor expressam as soluções de reuso: *Pacotes* e *Templates* [Rumbaugh & Jacobson & Boock, 1999].

#### 4.3.1 Pacotes

Protocolos de interação são padrões. A UML descreve dois modos de expressar agregação, que para a orientação a objetos, estrutura o comportamento dos objetos: Componentes e Pacotes. Os componentes são agregações físicas que compõem classes para propósitos de implementação. Pacotes são elementos da modelagem que agregam conjuntos conceituais. Aqui, classes podem estar conceitualmente agrupadas para qualquer propósito arbitrário.

Protocolos de interação agente podem ser classificados como padrões e tornam-se módulos reutilizáveis. Por exemplo, a Figura 17 mostra dois pacotes. O pacote encaminhamento expressa um protocolo simples entre uma *unidade móvel* e uma *central de regulação*. Aqui, a *unidade móvel* envia a solicitação de encaminhamento de vítima para uma *central de regulação* e a *central de regulação* responde com informações sobre o local que a vítima deve ser encaminhada, a *central de regulação* poderia também verificar a disponibilidade de um hospital. Baseadas nas informações de retorno, a *central de regulação* pode prover uma resposta mais precisa.

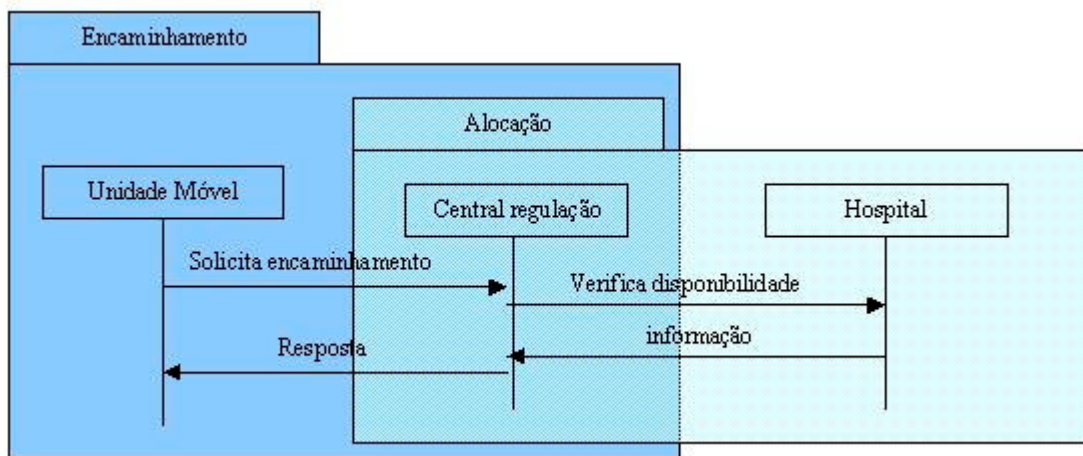


Figura 17: Usando pacotes para expressar protocolos

Burmeister et al. sugere uma construção similar, descrita por um complexo *protocolo de cooperação* [Burmeister et al, 1993]. São três protocolos primitivos: Oferecer, Solicitar e Propor —“são gerais o suficiente para serem usados em um grande número de situações de interação.” Sua abordagem “permite a construção de aplicações ou protocolos de tarefas.” Além de seus três protocolos primitivos, aqui é defendida uma abordagem pragmática onde o analista pode estender e incluir quaisquer protocolos que poderiam ser reusados em uma próxima especificação usando AUML.

#### 4.3.2 Templates

Na Figura 14 ilustrou-se um comportamento genérico que pode servir como uma solução em domínios de problemas análogos. Na Figura 17, o comportamento do *hospital* é exatamente reusado como definido pelo pacote alocação. Porém, um verdadeiro padrão (em vez de só um pacote de componente reutilizável) customizado deve ser sustentado. Por exemplo, na Figura 18 se aplica o FIPA *Contract Net Protocol* para um argumento particular envolvendo *central de regulação* e *unidade móvel*. Note que os agentes Iniciador e Participante se tornaram agentes *central de regulação* e *unidade móvel*. Também neste argumento existem duas formas de recusa pela *unidade móvel*: Recusa-1 e Recusa-2. Ultimamente, um prazo final real foi fornecido para uma resposta pela *unidade móvel*.

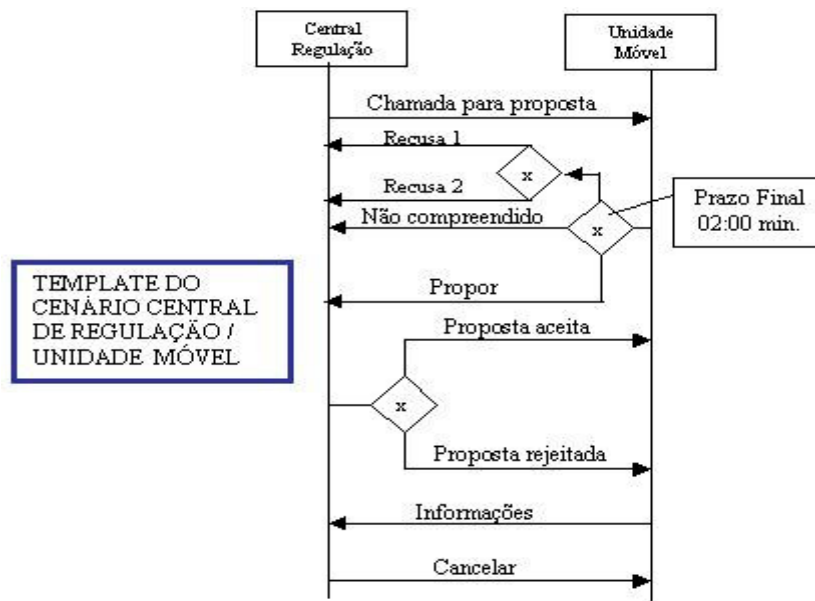


Figura 18: Instância do modelo apresentado na figura 13

A Figura 19 ilustra como o novo pacote na Figura 18 é produzido usando a definição de template da Figura 14. Wooldridge et al (2000) sugere uma forma semelhante de definição sobre pacote e template “um padrão de interação tem estado formalmente definido e abstrato longe de qualquer seqüência particular de passos de execução. Visualizar interações deste modo significa que a atenção é centrada na natureza e propósito essencial de interação em lugar das trocas ordenadas de mensagens particulares.” Em vez da anotação ilustrada por Wooldridge et al., a abordagem gráfica apresentada se assemelha mais a UML, expressando a mesma semântica.

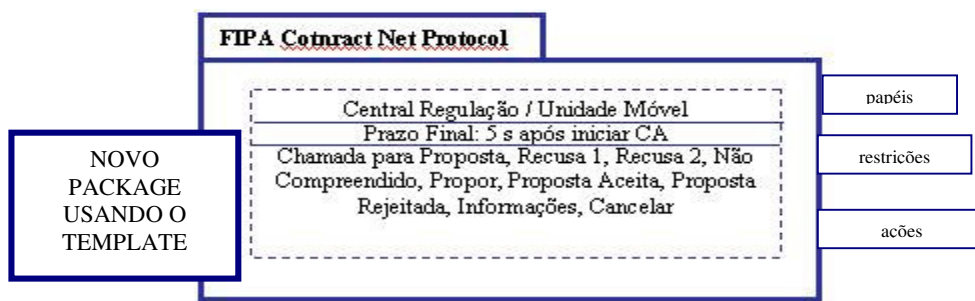


Figura 19: Identificação de um novo pacote

#### 4.4 Interações entre Agentes – Nível 2

Os modelos dinâmicos da UML são úteis para expressar interações de agentes. Os diagramas de interação capturam os padrões estruturais de interações de objetos. Os diagramas de seqüência e os diagramas de colaboração fazem parte do modelo dinâmico. Os dois diagramas contêm as mesmas informações. O plano gráfico do diagrama de seqüência

ênfatisa a seqüência cronológica de comunicações, enquanto o plano gráfico do diagrama de colaboração ênfatisa as associações entre os agentes. Os diagramas de atividade e de estado capturam o fluxo de processo na comunidade de agente.

#### 4.4.1 Diagramas de Seqüência

O diagrama de seqüência definido para a AUML apresenta algumas evoluções com relação ao diagrama definido pela UML (para uma discussão mais detalhada de diagramas de seqüência, veja Rumbaugh [Rumbaugh & Jacobson & Boock, 1999]). Nesta seção, serão discutidas estas extensões possíveis para UML, que também podem modelar protocolos de interação de agente.

A Figura 20 mostra alguns elementos básicos para comunicação de agente. O retângulo pode expressar agentes individuais ou fixar papéis e classes de agentes. Por exemplo, um agente individual poderia ser denominado como Bob/Cliente. Aqui, Bob é uma instância de agente que representa o papel do cliente. Bob poderia também representar o papel de provedor, empregado e proprietário. Para indicar que Bob é uma Pessoa (independente de qualquer papel que ele tenha), Bob poderia ser expresso como Bob:Pessoa. O formato básico para a etiqueta é nomedoagente/papel:classe. Então, podem-se expressar todas as várias situações para Bob, como Bob/Cliente:Pessoa e Bob/Empregado:Pessoa.

A caixa retangular também pode indicar um conjunto geral de agentes desempenhando um papel específico. Aqui, apenas a palavra cliente ou provedor apareceriam. Para especificar que o papel é para ser protagonizado por uma classe específica de agente, o nome de classe seria anexada (por exemplo, Empregado:Pessoa). Em outras palavras, o nomedoagente/papel:classe é a sintaxe usada para especificar o nome do agente individual.

A sintaxe nomedoagente/papel:classe já é herdada da UML (a sintaxe da UML indica um nome de objeto em vez de um agente). A Figura. 20 estende a UML, denominando a seta com um ato de comunicação agente (CA), em vez de uma mensagem.

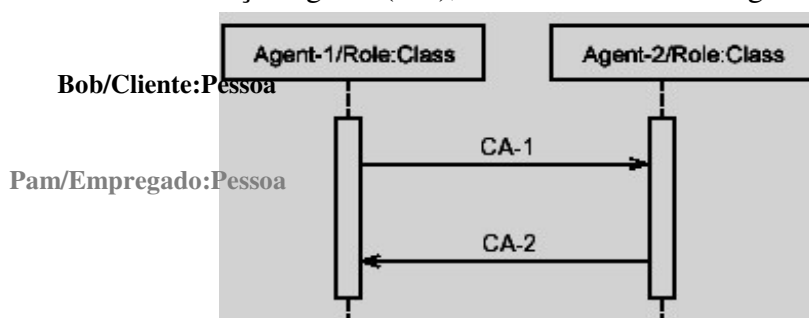


Figura 20: Formato básico para comunicação de agente.

Outra recomendação que a extensão UML sustenta são as linhas simultâneas de interação. A Figura 21 mostra três modos de expressar linhas múltiplas. A Figura 21(a) indica que todas as linhas CA-1 para CA-n são concorrentemente enviadas. Na Figura 21(b) foi incluída uma caixa de decisão indicando que deverá haver uma decisão antes do envio das CA`s, zero ou mais, neste caso. Se mais de uma CA é enviada, a comunicação é simultânea. A Figura 21(c) indica um “ou” exclusivo, de forma que exatamente uma CA será enviada. Desta forma, a Figura 21(a) indica um “e” na comunicação.

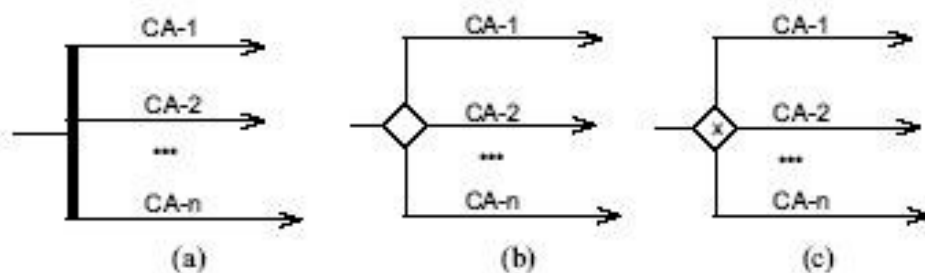


Figura 21: Transmissão simultânea de mensagens

A Figura 22 ilustra uma maneira de usar as linhas simultâneas de interação mostradas na Figura 21. As Figuras 22(a) e (b) retratam dois modos de expressar linhas simultâneas, do agente-1 até o agente-2. As múltiplas barras verticais indicam que o agente receptor é um processo com várias linhas de comunicação concorrente. As barras de ativação da Figura 22(a) são exibidas em paralelo. Na Figura 22(b), as barras de ativação aparecem uma em cima da outra. Algumas observações podem ser notadas sobre estas duas variações [Bauer, 1999; Odell, 1999]:

- O significado semântico é equivalente; a escolha é baseada em facilidade e clareza de aparência visual.
- Cada barra de ativação pode indicar que o agente está usando um papel diferente ou que meramente está empregando uma linha de processo diferente para sustentar o ato da comunicação. Se o agente está usando um papel diferente, a barra de ativação pode ser apropriadamente anotada. Por exemplo, na figura 22(a) e (b), CA-n é manipulada pelo agente debaixo de seu papel-1.

Estas figuras indicam que um único agente possui múltiplos processos de CA`s concorrentemente. Porém, as CA`s simultâneas poderiam ser enviadas para agentes diferentes, por exemplo, CA-1 para agente-2, CA-2 para agente-3, e assim por diante. Tal comportamento de protocolo já está sustentado pela UML; a anotação na Figura 21, por outro

lado, é uma extensão recomendada para UML (para tratamento mais detalhado destas extensões para a UML para diagrama de seqüência, veja [Bauer, 1999])

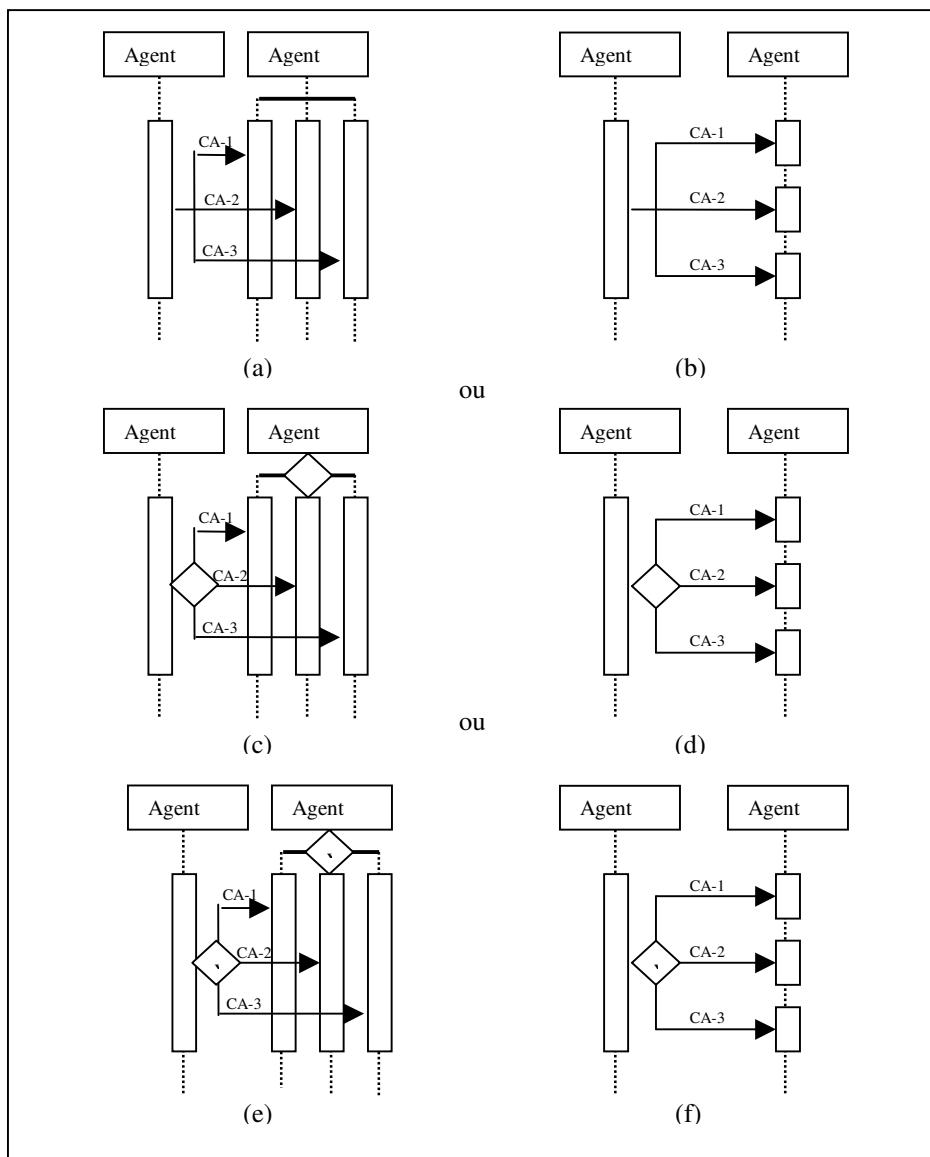


Figura 22: Múltiplas técnicas para representar comunicação simultânea entre agentes, utilizando múltiplos papéis

#### 4.4.2 Diagramas de Colaboração

A Figura 23 é um exemplo de um diagrama de colaboração e mostra um padrão de interação entre agentes. Uma das distinções primárias do diagrama de colaboração é que os agentes (os retângulos) podem ser dispostos em qualquer lugar no diagrama; considerando que, em um diagrama de seqüência, os agentes são colocados em uma camada horizontal na parte superior do diagrama. Semanticamente, eles são quase equivalentes; graficamente eles são semelhantes. Por exemplo, na Figura 24, onde é mostrado o diagrama de seqüência, ficam explícitos basicamente os mesmos significados subjacente da Figura 23. A experiência

monstra que no modelo baseado em agente ambos os diagramas são úteis.

A Figura 23 ilustra uma situação de atendimento pré-hospitalar. A *central de regulação* após receber um chamado aciona uma *unidade móvel* que para chegar ao local do atendimento faz uso de um agente *analizador de rotas*. Ao chegar no local do atendimento a *unidade móvel* troca informações com uma *clínica especialista* e aciona os serviços da *polícia*. Ao informar a *central de regulação* sobre o atendimento que está prestando, esta consulta o agente *analizador de disponibilidade* para buscar um *hospital* que possa atender a vítima. A troca de papéis é ilustrada quando a *central de regulação* e a *unidade móvel* assumem um papel que pode ser visto como um outro agente.

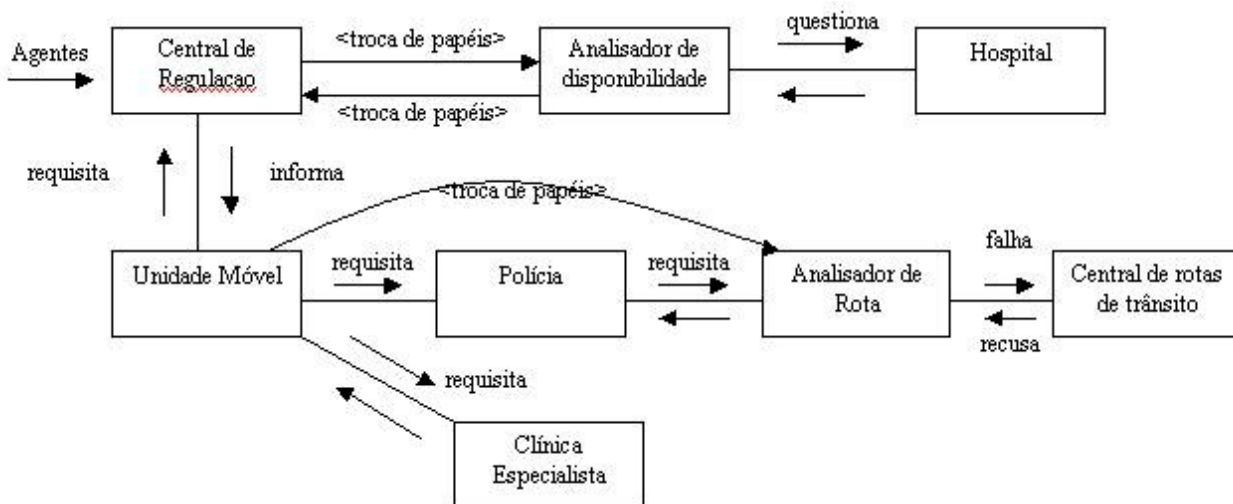


Figura 23: Um exemplo de um diagrama de colaboração mostrando uma interação entre agentes com papéis múltiplos.

A intuição na terminologia é que um caráter é um agente específico desempenhando um papel específico. O papel é uma abstração acima de vários personagens com padrões semelhantes de interação. Inversamente, cada nó é um agente em um papel específico, onde o "papel" é definido estreitamente.

Em se tendo um papel preciso ou uma definição de papéis, é possível construir um diagrama de colaboração que tem conteúdo semântico como um gráfico de Dooley.

Na Figura 24 é ilustrada uma representação diferente para a situação da Figura 23. O diagrama de seqüência é uma forma de representação utilizada para explicitar a seqüência das operações, como já foi mencionado, de certa forma a seqüência estabelecida em uma determinada situação necessita da participação de vários agentes, o que caracteriza uma colaboração entre eles. Sendo assim, a Figura 24 usou o mesmo caso da Figura 23, porém o foco dado é a seqüência das ações colocadas no diagrama de colaboração da Figura 23.



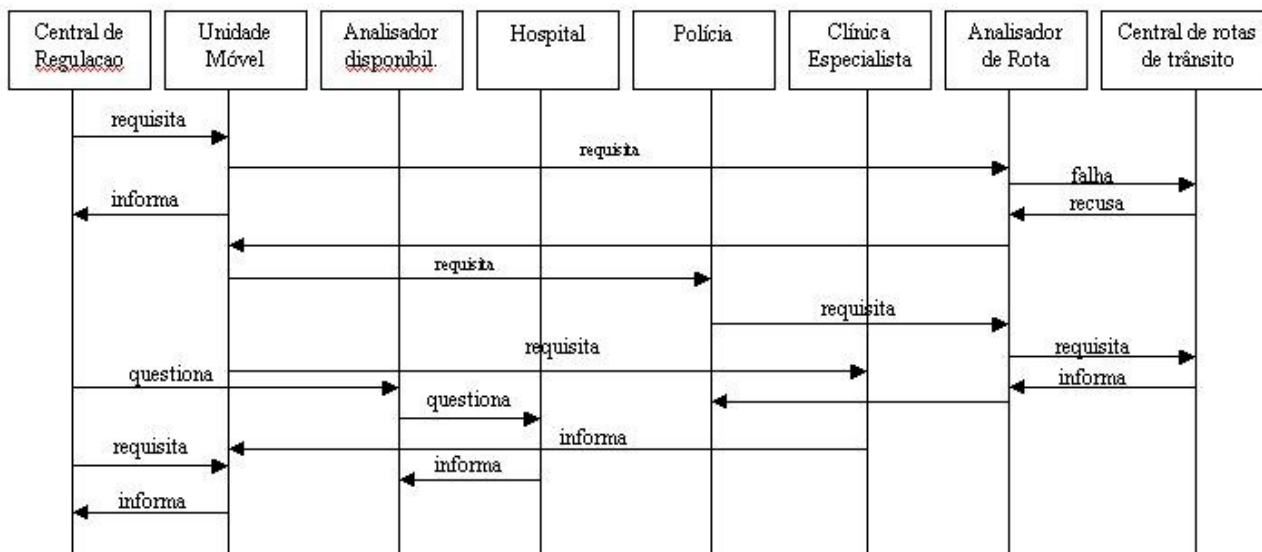


Figura 24: Uma versão do diagrama de seqüência de figura 23

#### 4.4.3 Diagramas de atividades

Os protocolos de interação agente podem, às vezes, exigir especificações muito claras sobre semântica do processo. O diagrama de atividade expressa operações e eventos que os ativam (para um tratamento mais detalhado, veja descrição do Odell de diagramas de atividade em [Martin & Odell, 1998]). O exemplo na Figura 25 mostra um atendimento pré-hospitalar envolvendo vários agentes. Aqui, um agente *central de regulação* inicia um atendimento. Este processo resulta em um pedido do evento que ativa a *unidade móvel* para atender a solicitação, que é então aceita por uma rede de atendimento. A rede de atendimento pré-hospitalar só pode associar um atendimento com uma orientação quando ambos forem aceitos. Uma vez que isto acontece, o *hospital* e a *unidade móvel* estão notificados que o atendimento foi realizado. O diagrama de atividades difere de diagramas de interação porque provê uma linha explícita de controle. Isto é particularmente útil para protocolos de interação complexos que envolvem processos simultâneos.

Os diagramas de atividade são semelhantes em redes de natureza Petri de vários modos. Primeiro, diagramas de atividade provêm uma representação gráfica que possibilita a visualização de processos simples, facilitando assim o projeto e comunicação dos modelos. Diagramas de atividade podem representar processos simultâneos assíncronos. Finalmente, eles podem expressar comunicações simultâneas com vários correspondentes. A diferença primária entre as duas abordagens é aquela em que os diagramas de atividade são formalmente baseados e estendidos na máquina de estados modelada pela UML [Rumbaugh & Jacobson & Boock, 1999]. Ferber na formalização de BRIC [Ferber, 1999] estende redes de

Petri para sistemas baseados em agentes. Esta proposta estende diagramas de atividade de UML para o mesmo propósito.

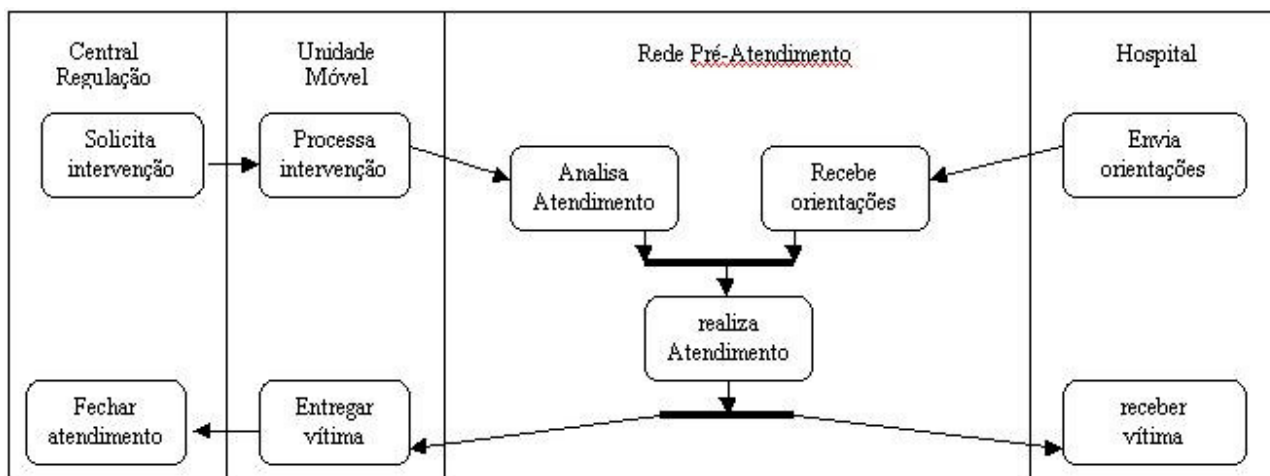


Figura 25: Um diagrama de atividade que mostra um protocolo de venda com interação de vários agentes.

#### 4.4.4 Diagrama de Estados

Outro processo relacionado a diagramas da UML é o diagrama de estados. Um diagrama de estados é um gráfico que representa uma máquina de estados. Os estados são representados como retângulos com cantos arredondados, enquanto transições estão geralmente representadas por arcos orientados que interconectam os estados. A Figura 26 mostra um exemplo de um diagrama de estados que controla uma *unidade móvel*. Aqui, se uma *unidade móvel* está em um estado “na central”, um agente pode alterar para “em trânsito (atend. s/ paciente)” — resultando em uma transição para “em atendimento”. Além disso, este diagrama indica que um agente passa o estado “em assepsia” apenas quando a *unidade móvel* está no estado “na central”. O estado “na central” tem duas outras ações possíveis além do “em trânsito (atend. s/ paciente)”, que são: “na central”, “em atendimento”, “em assepsia”, “em manutenção” e “fora de operação”.

O diagrama de estados não é comumente usado para expressar protocolo de interação porque é uma visão centrada no estado, em lugar de uma visão centrada no agente ou processo. A visão centrada no agente retratada por diagramas de interação enfatiza o primeiro e o segundo agente da interação. A visão centrada no processo enfatiza o fluxo do processo (por agente). A visão centrada no estado enfatiza os estados permissíveis mais proeminentes que o processo de transição de agente. A força primária do diagrama de estados em protocolos de interação agente é como um mecanismo de restrição para o protocolo. O diagrama de estados e seus estados estão tipicamente não implementados diretamente como agentes. Porém, um agente *unidade móvel* poderia solicitar as restrições da transição de estado.

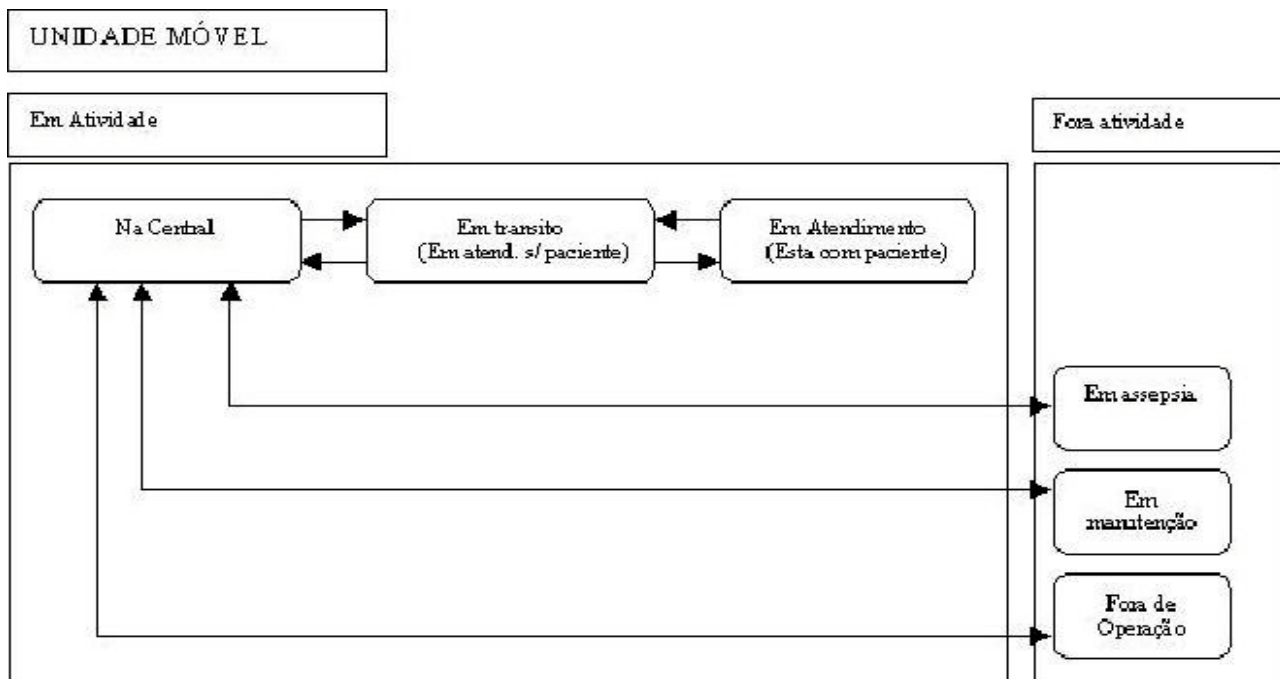


Figura 26: Um diagrama de estados indicando os estados e transições de Pedido

## 4.5 Representação de processos internos de agentes – Nível 3

No nível mais baixo da especificação de um protocolo de agente exige-se que um processo que acontece dentro de um agente seja detalhado a fim de implementar o protocolo. O comportamento interno de um agente pode, deste modo, ser descrito usando quaisquer representações do Nível 2. Além de que diagramas de atividade e estados também podem especificar o processo interno de agentes, como é ilustrado anteriormente.

### 4.5.1 Diagramas de Atividade

A Figura 27 mostra o processo detalhado que acontece dentro de um agente *unidade móvel*. Aqui, um diagrama de seqüência indicou que o processo do agente é ativado por uma CA e finalizado com um atendimento concluído. O processo interno pela *unidade móvel* é expresso como um diagrama de atividade, onde a *unidade móvel* analisa a viabilidade de um atendimento, realiza o atendimento, encaminha a vítima, entrega a vítima e encerra o atendimento. As caixas de operação destacadas e indicadas na linha mais abaixo representam interfaces para os processos executados por agentes externos — como também é ilustrado no diagrama de seqüência. Por exemplo, o diagrama indica isto quando a *unidade móvel* está prestes a realizar um atendimento. Ambos, solicitar orientações e necessidades da vítima são ações ativadas simultaneamente.

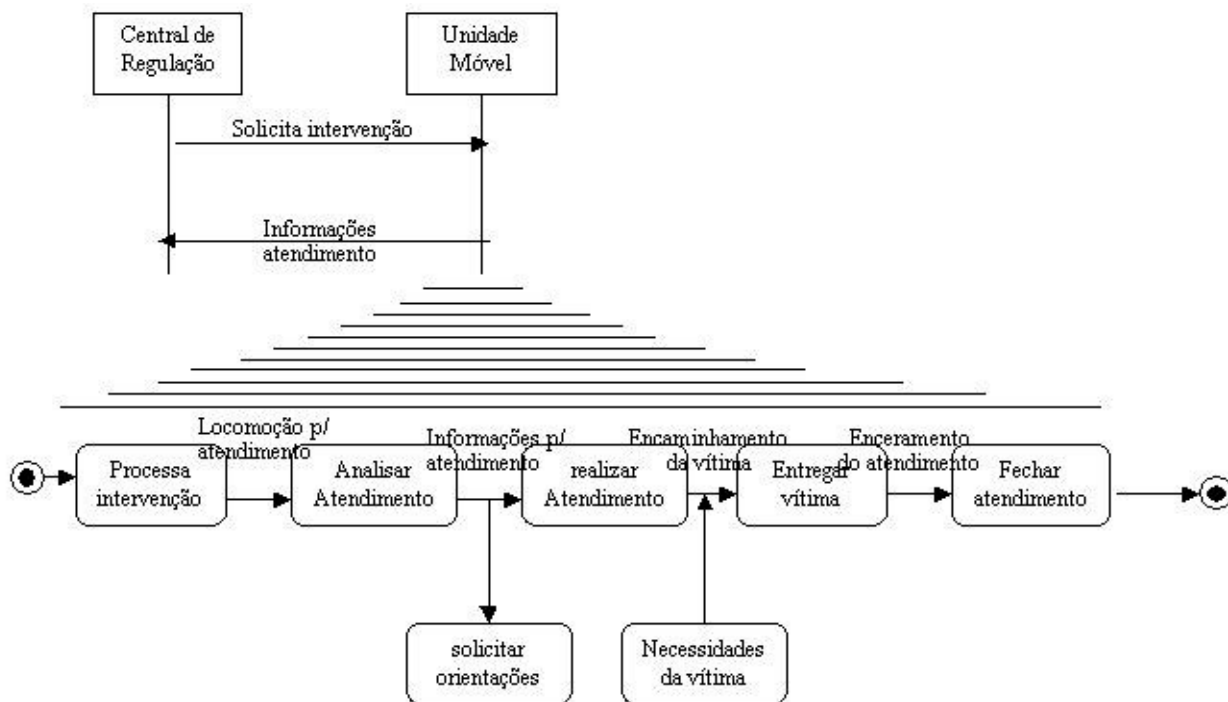


Figura 27: Um diagrama de atividade que especifica e ordena o comportamento do processo para um agente *Unidade móvel*.

#### 4.5.2 Diagramas de estado

O processo interno de um agente único também pode ser expresso como diagrama de estado. A Figura 28 mostra os estados e transições internas da *unidade móvel*. Este uso do diagrama de estados da UML dentro do agente sustenta a noção do Singh de esqueletos de agente [Singh, 1998b].



Figura 28. Diagrama de estados que especifica processo interno do agente *unidade móvel*.

A AUML foi utilizada nesta pesquisa com o intuito de modelar a arquitetura do sistema multi-agente do atendimento pré-hospitalar. O desenvolvimento da proposta segue baseado nos princípios tecnológicos e metodológicos discutidos até o momento.

# Capítulo 5 - Modelagem do sistema SAAP para o atendimento pré-hospitalar

## 5.1 Atendimento pré-hospitalar

Na busca pela melhor utilização das informações, as evoluções no desenvolvimento de sistemas passaram do mundo dos mantenedores de dados para a geração de informações. Há não muito tempo, era possível acompanhar o desenvolvimento de aplicações distintas dentro de uma organização onde cada aplicação preocupava-se em resolver e manipular dados apenas de seu contexto pré-definido, sem a interferência ou a integração com qualquer outra área da mesma organização. Com o passar do tempo, as necessidades de uma visão mais global de uma instituição passaram a ser questão de sobrevivência. Os aplicativos desenvolvidos para as diversas áreas deveriam então tratar das informações com o intuito de produzir resultados acerca da instituição como um todo, e não mais apenas sobre o contexto ao qual foram inicialmente projetados. Começaram então os desenvolvimentos centralizados, onde os contextos das aplicações passaram a tratar da totalidade das informações. Neste período, o controle passou a ser dedicado à organização como um todo, atendendo à solicitação de uma visão integrada. A evolução continua e mostra mais uma vez seu interesse em ver as diversas organizações unidas para resolver determinadas situações. Viu-se que as instituições, um hospital, por exemplo, sozinhas resolviam seus próprios problemas, porém, o relacionamento entre estas passou a ser inevitável. As soluções passaram a acontecer com a utilização de mais de uma instituição e a integração passou dos limites pré-definidos.

Uma instituição, sob o ponto de vista de suas soluções, gera e utiliza informações para uma rede de relacionamentos. Esta rede de relacionamentos, por sua vez, possui seus objetivos, suas informações e também gera respostas ao meio. O caso do atendimento pré-hospitalar se insere neste contexto. Com o controle das organizações presentes na rede do atendimento pré-hospitalar, ficou evidente a necessidade de integração dentro da rede. Um hospital, por exemplo, pode ser organizado, possuir um controle a contento, trazer todas as informações necessárias para seus gestores. Porém, quando entra em uma rede onde o contexto passa a ser, por exemplo, o atendimento pré-hospitalar, este possui informações extremamente úteis a outros órgãos, assim como também passa a fazer uso de informações

que foram registradas em organizações externas. A integração, ou ainda a formação da rede de interesse, nesse caso, criou-se devido à necessidade de um atendimento à população. O atendimento pré-hospitalar ultrapassou as barreiras das organizações e, para ser realizado com sucesso, necessita da interação de mais de uma organização.

A necessidade torna-se ainda mais evidente quando parte de um trabalho é realizado por uma organização e parte por outra. Nesse contexto, uma organização pode ser denominada de *unidade funcional* e deve necessariamente utilizar e fornecer informações a outras que compartilham o mesmo contexto. Essa troca de informações atualmente acontece por diversos outros meios, menos por meio de uma aplicação informatizada. Por exemplo, quando acontece um atendimento pré-hospitalar hoje, as organizações como *hospital, clínicas, central de regulação, unidade móvel e serviços de apoio* (polícia, bombeiro, etc) são comunicados. O desenvolvimento de uma única aplicação para atender a todas as necessidades de todos estes órgãos já foi inclusive experimentado [Machado et al, 2000]. Alguns projetos trataram dessa situação há algum tempo, porém o custo de se informatizar um número variável de entidades distintas acaba por ser alto, sem contar com treinamentos, novas inserções de entidades e integrações que este produto deverá possuir com cada organização. A solução converge para aproveitar o desenvolvimento de cada organização, tornando possível o encapsulamento de aplicações já existentes dentro de uma estrutura que passa a ter um comportamento de agente ou unidade funcional.

Cada agente trata as informações pertinentes ao seu negócio. Cada agente possui as suas tarefas pré-definidas. Cada agente possui informações que possam ser compartilhadas com os demais agentes. Todo agente comunica em uma determinada linguagem entendida por todos que fazem parte da rede. Cada agente pode fornecer informações que sejam utilizadas por outros. Com essa forma de tornar informações disponíveis, basta apenas que cada agente acople ao seu sistema uma camada de comunicação que permita a realização da comunicação, a troca de mensagens com qualquer outro agente da rede.

As razões que justificam a distribuição de uma aplicação podem ser apresentadas a partir de uma analogia das formas de organizações conhecidas entres os seres humanos, como mostra Fox [Fox, 1988]. Um sistema distribuído é como uma organização particular — decomposição de tarefas e regime de controle —, resultado da distribuição de um conjunto de tarefas sobre um conjunto de elementos disjuntos de processamento lógico ou físico. Com um sistema distribuído pode-se reduzir: a) a complexidade das informações (volume de informações que um ser humano ou um processador pode tratar em um determinado intervalo de tempo); b) a complexidade das tarefas (volume de ações necessárias para a execução de

uma tarefa); c) a complexidade da coordenação (coordenação da execução de uma tarefa decomposta). Além destes itens é importante identificar no atendimento pré-hospitalar alguns pontos como: a) a localização física dos conhecimentos; b) a localização da equipe de concepção/programação dos sistemas; c) o nível de conhecimento ou de domínio das interações entre os módulos de uma aplicação. O sistema distribuído será visto como uma organização capaz de integrar as entidades que participam da rede e de resolver os problemas do atendimento pré-hospitalar.

Uma organização é um conjunto de agentes (operacionais e administrativos) relacionados segundo uma estrutura que permite, através do desempenho das atividades, alcançar os objetivos globais dessa organização [Shmeil, 1999]. Relembrando essa definição de organização, tem-se o início da distribuição de papéis e de tarefas dentro da organização do atendimento pré-hospitalar. Como em qualquer organização, o atendimento pré-hospitalar é regido por normas e regras, e os agentes que participam dessa organização possuem atividades pré-definidas. Estão listadas abaixo algumas atribuições que alguns agentes devem possuir, lembrando que um sistema aberto deve possibilitar a integração de novos agentes de forma dinâmica.

**Central de Regulação:** Dentro dessa estrutura devem estar armazenadas todas as informações de todas as ocorrências atendidas, o controle de todas as vítimas e de todas as unidades que podem realizar atendimento, o controle de todas as equipes e todos os médicos que possam vir a ser convocados, o apoio que as unidades precisarão em um momento de atendimento. Todas as determinações serão tomadas na *Central de regulação*. Entre as suas principais tarefas estão:

- Atender chamada telefônica;
- validar a chamada;
- captar dados iniciais do atendimento (Endereço, veículos envolvidos e outros);
- captar dados das vítimas (Quantidade, estado, e outros);
- passar instruções ao informante da chamada;
- liberar *unidade móvel* para atendimento;
- liberar equipe de socorro para o atendimento;
- decidir sobre a intervenção médica;
- indicar *hospital* para encaminhamento da vítima;
- instruir os paramédicos que foram deslocados ao atendimento, tendo a possibilidade de disponibilizar a conduta adequada ao atendimento corrente;

- buscar informações sobre o melhor atendimento dentro da rede do atendimento pré-hospitalar (caso haja clínicas ou hospitais na rede); e
- acionar órgãos de apoio.

**Unidade Móvel:** É através da unidade móvel que o atendimento será realizado. Estas unidades são dotadas de equipe de socorro, equipamentos e medicamentos para a melhor realização de um atendimento pré-hospitalar. É importante ressaltar que as atividades das unidades móveis são classificadas quanto a seu tipo, ou seja, ao tipo de atendimento que a unidade pode realizar. Estes tipos de unidade móvel são definidos pela Portaria 814 [Brasil, 2001]. Algumas características podem ser atribuídas a maioria dos tipos de unidade, estas são:

- definir melhor caminho para chegar ao atendimento;
- captar informações sobre o atendimento (endereço correto, veículos envolvidos e outros);
- captar informações médicas das vítimas;
- captar sinais vitais das vítimas;
- indicar estado de cada vítima, podendo atender a mesma com a utilização de consultas especializadas;
- indicar *hospital* a ser encaminhada a vítima;
- informar sobre equipamentos e equipe médica necessária para atendimento à vítima em *hospital* ou *clínica*, e
- acionar órgãos de apoio.

**Hospitais ou Clínicas:** Independente da quantidade de clínicas e hospitais presentes na organização do atendimento pré-hospitalar, a presença dos mesmos é imprescindível. O *hospital* é o terceiro órgão vital ao funcionamento de um atendimento. Além da necessidade de receber uma vítima que está sendo atendida por uma *unidade móvel*, as *clínicas* e *hospitais* têm papel vital na orientação da equipe que está realizando um atendimento, seja diretamente na *unidade móvel* ou através da *central de regulação*. A invocação deste agente torna-se mais evidente quando o atendimento é realizado em região isolada, onde o deslocamento até um hospital pode demorar certo tempo. Dentre suas tarefas estão:

- informar equipe médica disponível;
- informar equipamentos médicos disponíveis;
- informar sobre leitos disponíveis;



- analisar sinais vitais das vítimas;
- analisar estado da vítima que está sendo atendida na *unidade móvel*;
- orientar os atendimentos;
- acionar especialista para atendimento de urgência/emergência;
- receber vítima do atendimento pré-hospitalar (preparar instalações e equipes para o mesmo).

**Serviços de apoio:** Os serviços de apoio são os diversos órgãos que também possuem papel importante dentro do atendimento pré-hospitalar, sempre fornecendo apoio às operações. Órgãos como polícia, DETRAN (BPTRAN), DIRETRAN, IML e outros são acionados para possibilitar melhores condições em um atendimento. Dentro da organização do atendimento pré-hospitalar, estas entidades terão os seguintes papéis:

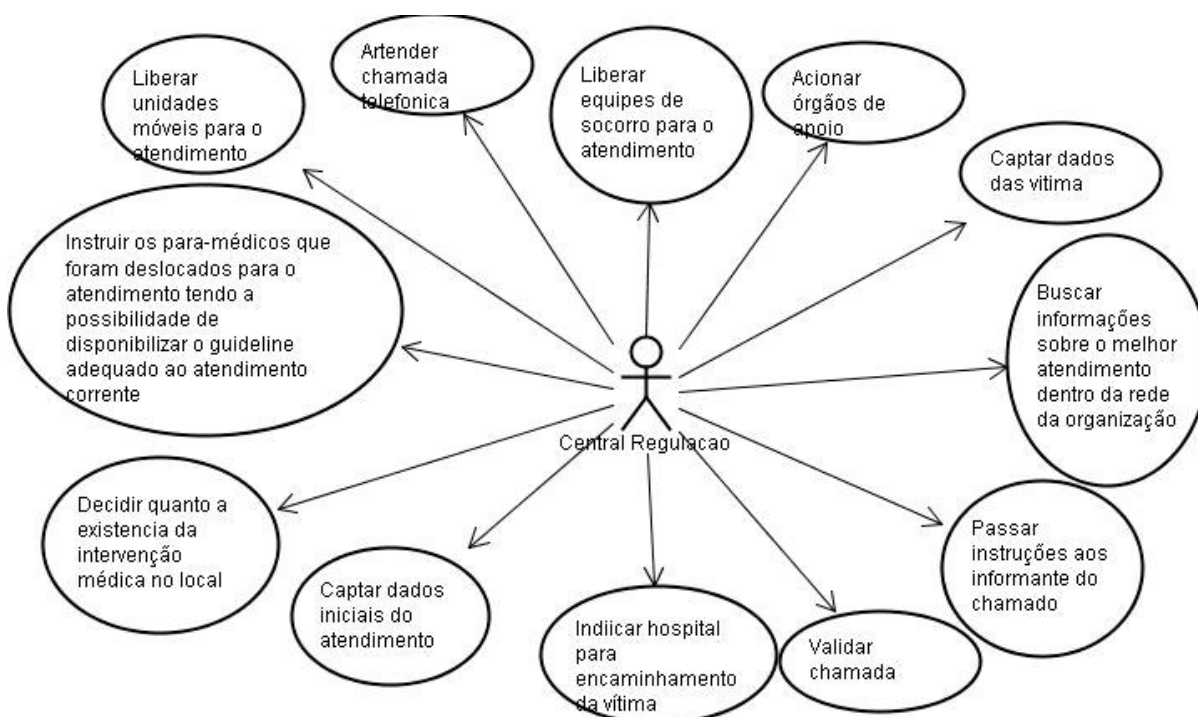
- registrar atendimento;
- fornecer informações à *unidade móvel*;
- fornecer informações às clínicas e hospitalares;
- fornecer informações à *central de regulação*;
- prestar auxílio no atendimento pré-hospitalar;
- encaminhar vítimas.

Por ser o universo médico tão amplo, é impossível listar todas as atividades que um agente pode fornecer dentro do atendimento pré-hospitalar. Os atendimentos podem ocorrer em diversas especialidades médicas e para cada uma delas há uma série de rotinas e práticas que podem ser executadas por um agente e por outro. De certa forma, um mínimo de informações deve ser trafegado pela rede para um atendimento, mas a hipótese deste volume de dados e de informações ser ampliado existe e a organização da arquitetura proposta permite isso.

Pode haver atendimento em que não exista a necessidade da utilização das organizações ditas vitais para o atendimento pré-hospitalar, assim como podem existir atendimentos que exijam a presença de inúmeros agentes.

Para melhor exemplificar, e com o intuito de começar a modelar um exemplo de utilização da estrutura do atendimento pré-hospitalar, serão apresentados abaixo os gráficos que ilustram as atividades que estas instituições listadas acima estarão realizando, através da representação UML de casos de uso.

O diagrama de caso de uso da Figura 29 ilustra as tarefas que a *central de regulação* estará realizando dentro dos atendimentos. O agente *central de regulação* poderá ter sua lista de tarefas ampliada de acordo com as necessidades, assim como outros agentes que auxiliem a *central de regulação*, ou nela busquem informações que poderão ser anexados à organização.



**Figura 29: Diagrama de Caso de Uso das atividades da *Central de regulação*.**

No diagrama de caso de uso da Figura 30 estão dispostas as tarefas executadas pelo agente *unidade móvel*. As tarefas aqui ilustradas poderão ser utilizadas em qualquer atendimento, assim como poderão surgir novas tarefas de acordo com a utilização do agente.



Figura 30: Diagrama de Caso de Uso das atividades da *Unidade móvel*.

No diagrama de caso de uso da Figura 31 são apresentadas as tarefas iniciais das clínicas e hospitais que poderão compor a organização do atendimento pré-hospitalar. Não necessariamente todas as clínicas e hospitais terão as tarefas aqui descritas, e muito menos terão apenas estas tarefas.

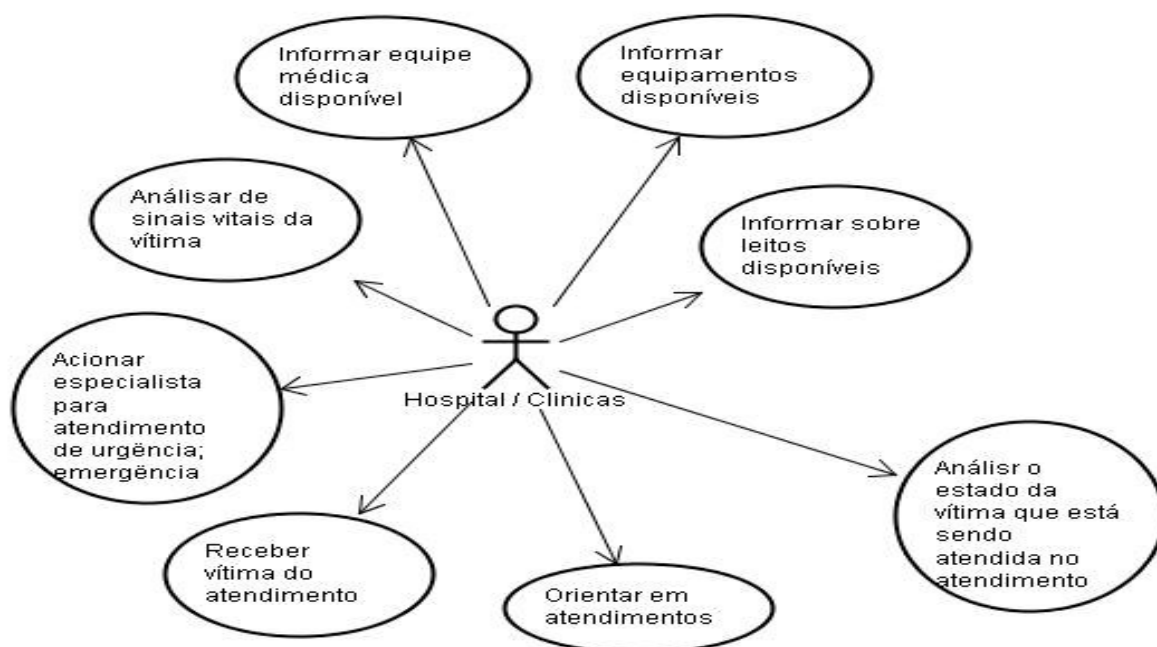
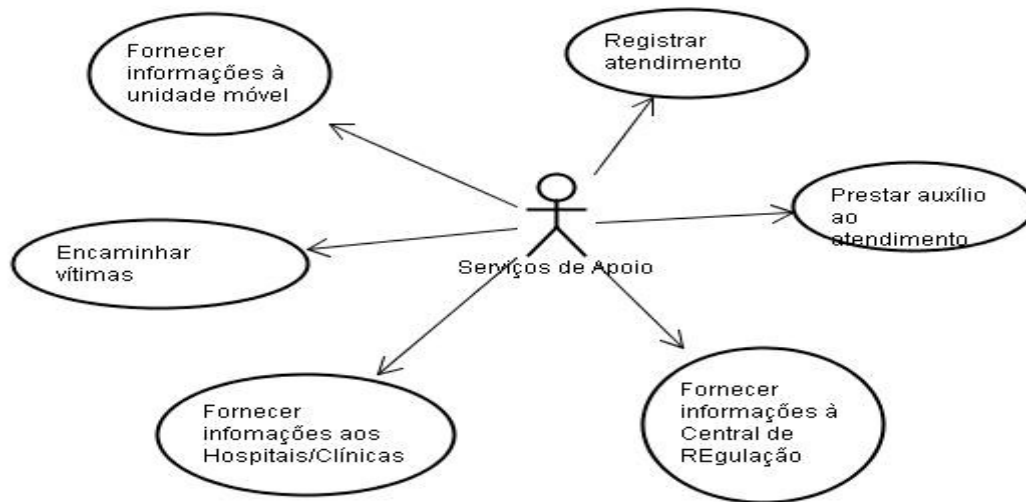


Figura 31: Diagrama de Caso de Uso das atividades dos Hospitais/Clinicas

No diagrama de caso de uso da Figura 32 estão explicitadas as tarefas dos serviços de apoio. A quantidade de agentes de serviços de apoio é variável.



**Figura 32: Diagrama de Caso de Uso das atividades dos Serviços de Apoio**

Os exemplos mostrados nas figuras 29, 30, 31 e 32 são apenas a título de apresentação. Não significa que as tarefas aqui dispostas necessariamente façam parte destes agentes, assim como, não as pode fixar como sendo suas únicas tarefas.

## 5.2 Modelagem do atendimento pré-hospitalar usando AUML

Baseado em um conjunto mínimo de agentes participantes, foi realizada a modelagem do sistema SAAP para atender ao atendimento pré-hospitalar.

Serão apresentados os diagramas de seqüência, de colaboração, de atividades e de estados do sistema SAAP baseados na AUML. A primeira identificação de modelagem de estrutura, realizada utilizando AUML, foi a modelagem de um pacote evidenciando a necessidade de reuso.

A troca de informações entre os agentes de um sistema multi-agente deve ser regulamentada por uma estrutura que possa permitir a interface com os demais agentes do sistema. Essa interface diz respeito a um conjunto de regras que possibilitam a comunicação entre os mesmos. Existem alguns protocolos distintos para a realização deste tipo de controle. O *Contract Net* é um protocolo que possui como principal objetivo realizar a integração de agentes, possibilitando que esses agentes possam trocar informações, baseados em um contrato que é estabelecido pelo próprio protocolo [Smith, 1980]. O atendimento pré-hospitalar estará fazendo uso do *Contract Net Protocol* para a comunicação entre os agentes

envolvidos. Para tanto, a AUMML prevê a formação de estruturas chamadas pacotes e templates. Essas estruturas possuem como objetivo a especificação de componentes que possam ser reusados dentro ou fora de um contexto.

Transformar o Contract Net Protocol em um pacote na aplicação do atendimento pré-hospitalar é uma solução apenas de modelagem, como é exibido na Figura 33. Toda interação que acontece dentro de um atendimento pré-hospitalar é iniciada com a utilização deste protocolo. Um pacote, e, neste caso, também um template, pode ser reutilizado por outra estrutura em uma outra razão de tempo. No caso do atendimento pré-hospitalar, esse template será utilizado por todos os agentes que desejarem iniciar uma comunicação para solucionar um atendimento.

No diagrama de seqüência da Figura 33 ilustra-se a utilização desse pacote quando a *central de regulação*, que em um primeiro momento é quem recebe a solicitação de atendimento provinda de uma interface externa, inicia a comunicação com uma *unidade móvel* que possa realizar o atendimento. A facilidade do uso dos agentes nesse tipo de aplicação pode ser melhor visualizada quando, em um atendimento onde sejam necessárias interações de mais de uma *unidade móvel*, a *central de regulação* possa ter a capacidade de iniciar atendimentos com um número N de unidades móveis. Aplicando-se em uma situação de catástrofe ou em um horário de pico, por exemplo, numa sexta-feira de chuva, das 17 horas às 23 horas, já que a incidência de atendimentos nessas condições e horários é maior que em outros períodos, a *central de regulação* é capaz de receber mais de uma solicitação ao mesmo tempo e, conseqüentemente, poderá iniciar a comunicação com mais de uma *unidade móvel* simultaneamente. Essa facilidade é encontrada em uma arquitetura multi-agente, e está sendo representado no diagrama da Figura 33. Em cada atendimento que a central consegue disponibilizar a uma unidade móvel, um “canal” de comunicação é efetivado, ou seja, segundo as especificações da AUMML, uma comunicação entre agentes é realizada. Conseqüentemente, como se pode perceber pela descrição de um atendimento pré-hospitalar, existe ainda outros agentes envolvidos e para cada agente estará se estabelecendo um “canal” de comunicação. Para cada comunicação iniciada, uma instância do *template Contract Net Protocol* é realizada.

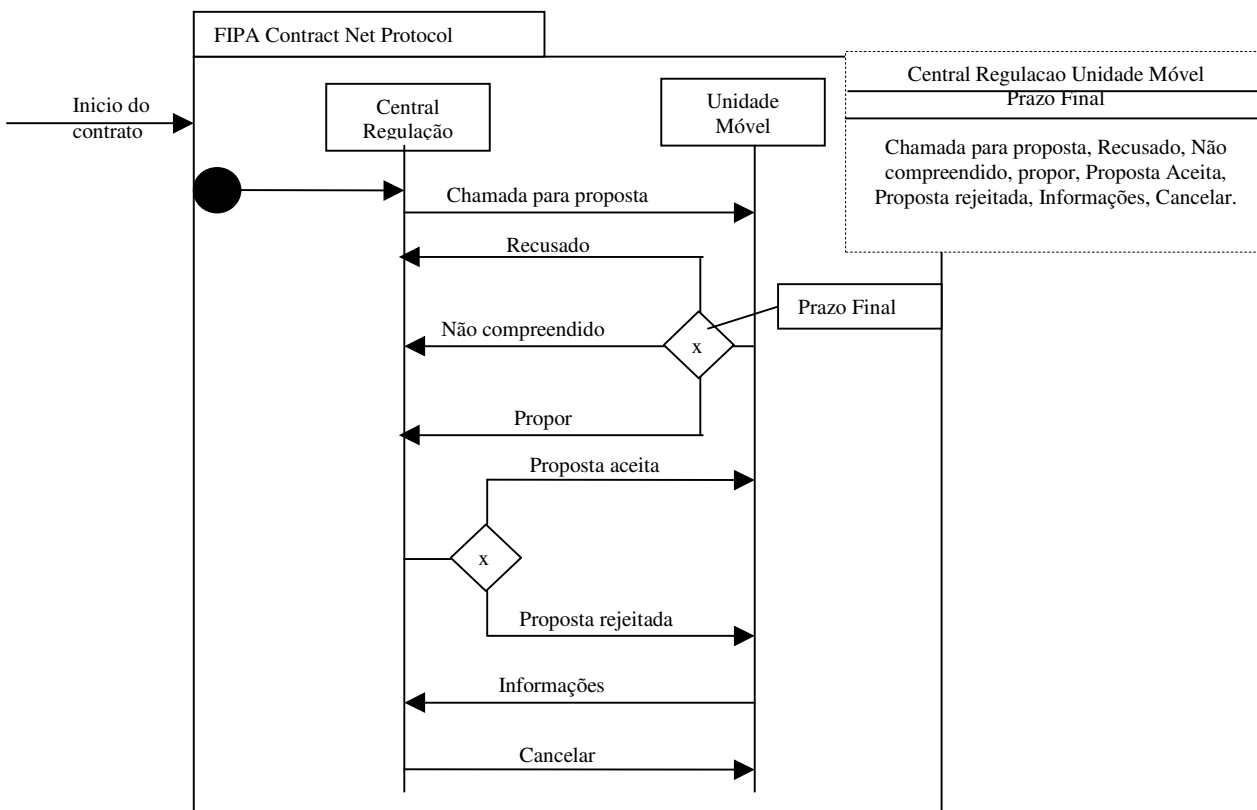


Figura 33: Diagrama de seqüência do template Contract Net Protocol

A definição do template Contract Net Protocol como pacote é evidenciado dentro da AUML pela representação da figura 33, onde são apresentados os agentes que estão instanciando o pacote, o tempo máximo para a comunicação, também chamado de *deadline* de comunicação, e os eventos que são chamados pelo pacote.

O quadro pontilhado no canto superior direito da figura indica a que agentes o template está sendo aplicado, o *deadline* que o agente *unidade móvel*, neste exemplo, possui para responder a solicitação de contrato e, na parte inferior, todas as operações que são realizadas dentro do template. O nome fixado no quadro acima da figura indica o nome do pacote.

O Diagrama de Seqüência AUML da Figura 34 ilustra na prática todos os contratos que poderão ser realizados por uma *central de regulação*. São explícitas aqui apenas as comunicações para registro da rede. Como já foi mencionado, existe um conjunto mínimo de agentes que devem participar de um atendimento pré-hospitalar. Esse conjunto mínimo, composto pela *central de regulação*, *unidade móvel* e hospital, deve necessariamente possuir pelo menos um contrato ativo. A Figura 34 ilustra a série de possíveis conexões entre agentes que atendam também ao atendimento pré-hospitalar.

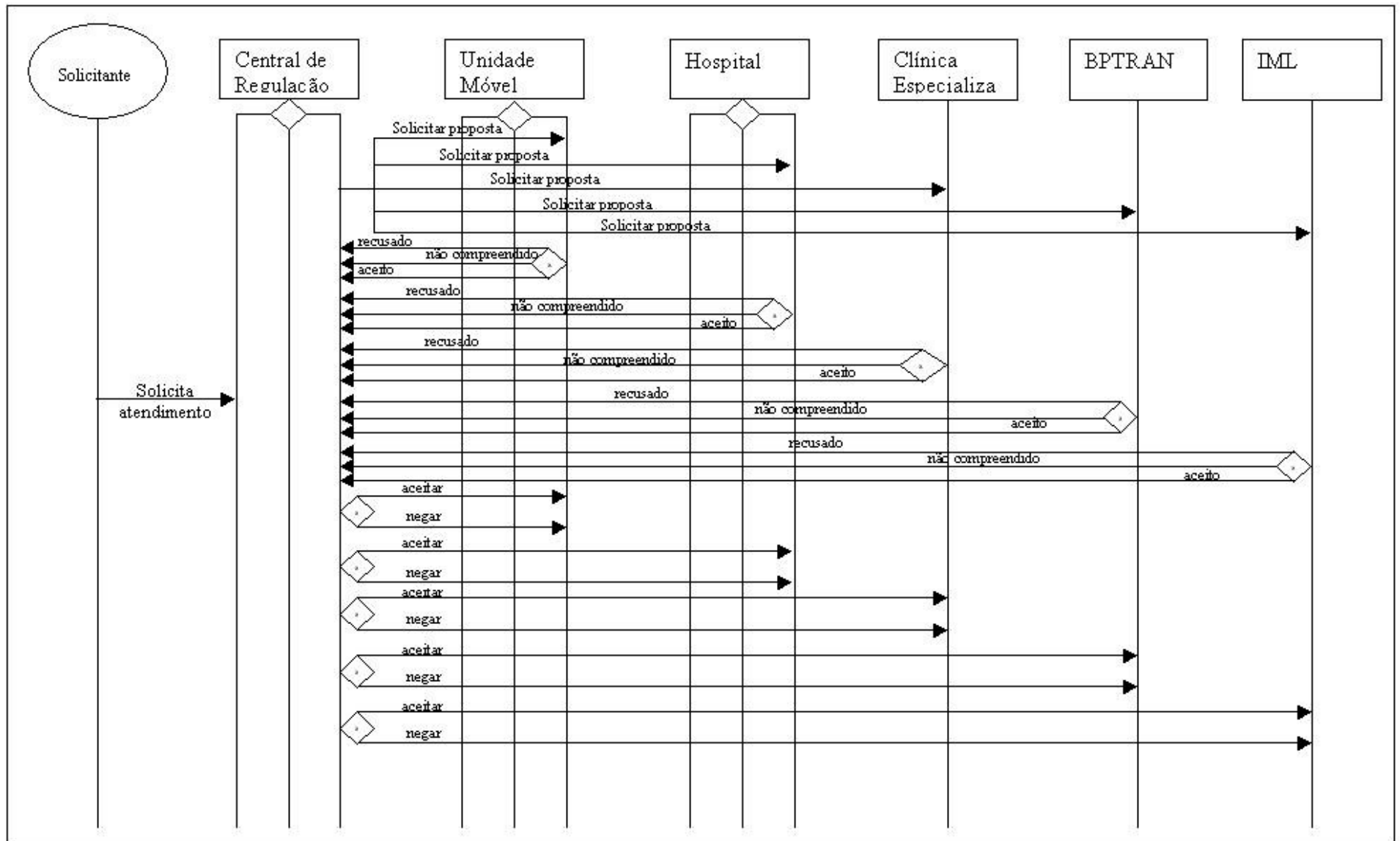


Figura 34: Diagrama de seqüência que ilustra algumas conexões entre agentes, utilizando o pacote Contract Net definido.

Neste momento, estão estabelecidas todas as conexões que o agente *central de regulação* estará utilizando para realizar um atendimento pré-hospitalar. No diagrama de seqüência da Figura 35, ilustrar-se-á a continuação deste e, principalmente, a utilização destes agentes dentro do atendimento pré-hospitalar.

É interessante salientar que não deverá necessariamente ser estabelecido um contrato para cada atendimento com todos os agentes participantes do sistema. Um contrato pode ainda ser mantido com mais de uma organização e esta pode vir a atender a mais de um atendimento. Necessariamente para o atendimento pré-hospitalar deverá haver a contratação de uma *unidade móvel*, um *hospital* ou *clínica* e a *central de regulação*, que formam o conjunto mínimo de agentes responsáveis para que um atendimento possa ocorrer. Os demais agentes participantes da arquitetura poderão participar em vários atendimentos com o mesmo contrato, já que o relacionamento já está estabelecido. Pode-se, por exemplo, buscar uma clínica especializada em ortopedia em um determinado momento em que a *central de regulação* está realizando o atendimento a 10 chamados aproximadamente. Destes 10 chamados, 50% são ditos acidentes de trânsito e demandam uma grande quantidade de recursos da clínica citada. O mesmo agente, poderá prestar auxílio a todos os chamados onde seja detectada a sua necessidade, sem necessariamente ter que estabelecer para cada atendimento um novo contrato de relacionamento.

No diagrama de seqüência da figura 35 é ilustrada as demais trocas de mensagens entre os agentes que possuem um contrato estabelecido com a *central de regulação*. Após a contratação dos serviços primários do atendimento, o sistema poderá contar com uma série de outros agentes que podem registrar-se na rede a qualquer momento. No diagrama de seqüência a seguir são ilustradas algumas mensagens que podem ser trocadas na solução de um atendimento. Estas mensagens são variáveis e podem envolver os demais agentes. As extensões que a AUML realizou nos diagramas da UML ilustram a possibilidade de possuir a representação de mais de um papel para um agente. Uma outra representação visível no diagrama abaixo é a possibilidade do agente poder transmitir mais de uma mensagem simultaneamente. Essas duas extensões são visíveis na Figura 35 nos agentes *central de regulação*, *unidade móvel* e *hospital*. Nestes três agentes foi estipulado um número de três papéis para cada um. Vale lembrar que a metodologia possui flexibilidade suficiente para que esse tipo de configuração seja alterada sem que isso venha a prejudicar o funcionamento da rede. Essa representação possibilitou que um diagrama de seqüência de objetos pudesse ser também utilizado para representar agentes.



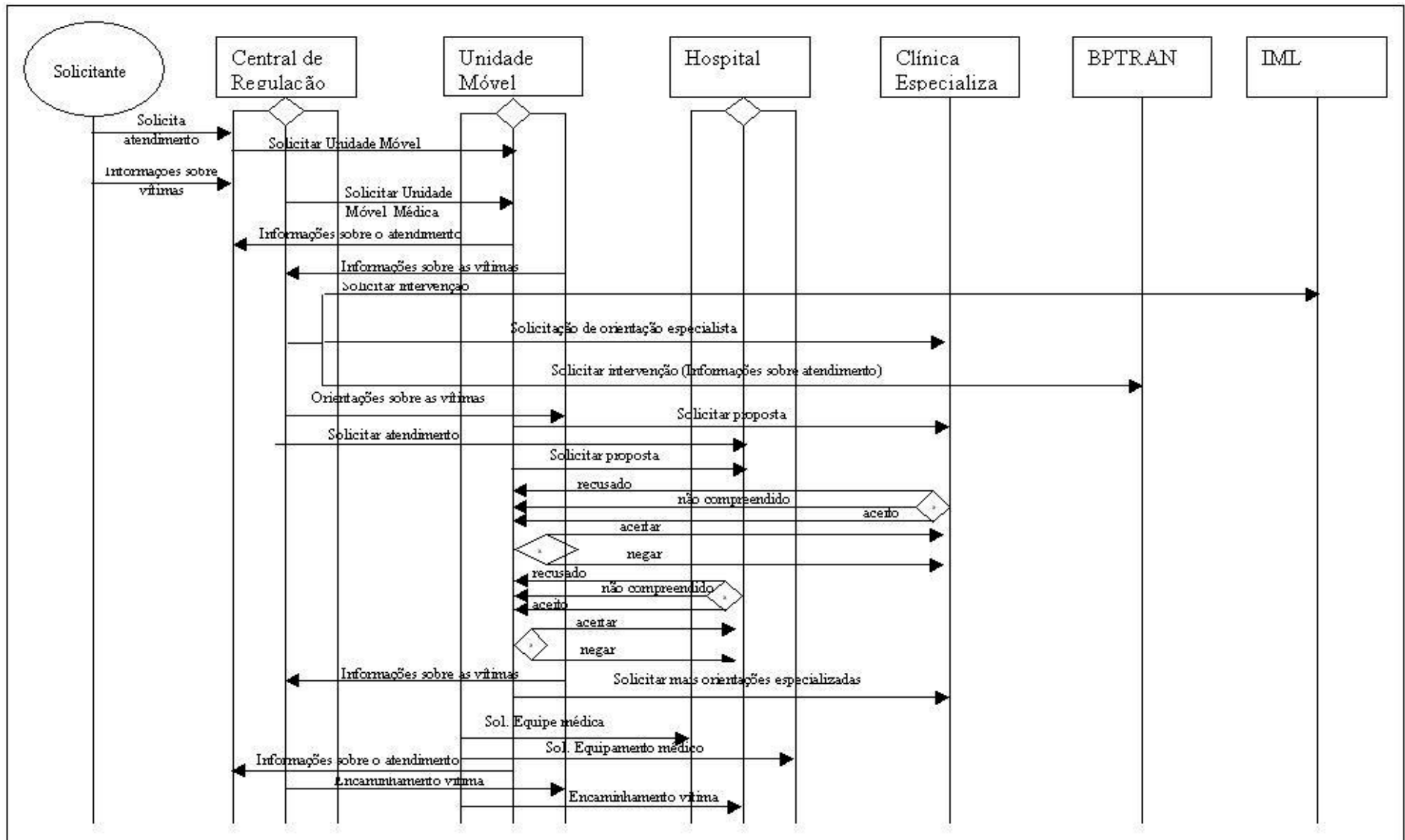


Figura 35: Diagrama de seqüência que ilustra algumas conexões entre agentes na realização de um atendimento pré-hospitalar

O diagrama de colaboração é uma representação gráfica que visa mostrar os relacionamentos existentes entre os agentes de um sistema. No diagrama a seguir são ilustrados alguns dos possíveis agentes envolvidos em um atendimento pré-hospitalar. Uma particularidade interessante que se torna explícita neste diagrama é a troca de papéis que ocorre entre os agentes *unidade móvel* e *central de regulação*. Em um dado momento se torna mais simples a comunicação direta da *unidade móvel*, que está prestando o atendimento, e o agente *hospital*, por exemplo. Essa mobilidade negocial fica também explícita através das trocas de mensagens ilustradas no diagrama de seqüência da Figura 35, quando para um atendimento em específico, existe a utilização do template de contrato, ou seja, a contratação de uma competência para intervir junto ao atendimento que está acontecendo. Os questionamentos realizados pelo agente *unidade móvel* podem ser os mesmos que o agente *central de regulação* realiza, porém a troca de mensagens direta é sempre uma solução mais rápida, pois independe do tráfego que pode ser gerado pelo acúmulo de solicitações de atendimentos para o agente *central de regulação*. Mais uma vez é melhor avaliar essa situação em extremos, ou em um atendimento de catástrofe ou em horários de pico, aliados ao clima propício a acidentes de trânsito. Essas situações trazem à *central de regulação* um acúmulo de atendimentos, podendo gerar uma sobrecarga em todo o sistema. Esta sobrecarga pode ainda gerar uma demora na solução das solicitações, o que poderia prejudicar os atendimentos que estão em curso. A ilustração mostra uma das características da arquitetura que está sendo proposta, que é a facilidade da utilização do agente *central de regulação* como regulador de um atendimento pré-hospitalar ou a eficiência de um agente poder se dirigir a outro diretamente, resolvendo seu problema sem alocar recursos do agente *central de regulação*. Dessa forma, pode ser realizada uma analogia com a Figura de um agente *black board* (agente de ambiente) dentro do contexto de um atendimento pré-hospitalar, devido ao fato de que uma *central de regulação* sempre é a entidade que necessita ter o controle dos atendimentos que estão acontecendo.

Em tempo é importante salientar que o sistema pode contar ainda com o apoio de outros recursos para sanar suas dificuldades, como por exemplo, a falta de comunicação. *Guidelines*, neste caso, é um recurso que pode auxiliar um determinado agente em um momento em que a comunicação é escassa. Desta forma, o agente munido deste recurso poderá ter acesso a informações e procedimentos que tragam auxílio ao atendimento que está sendo prestado.

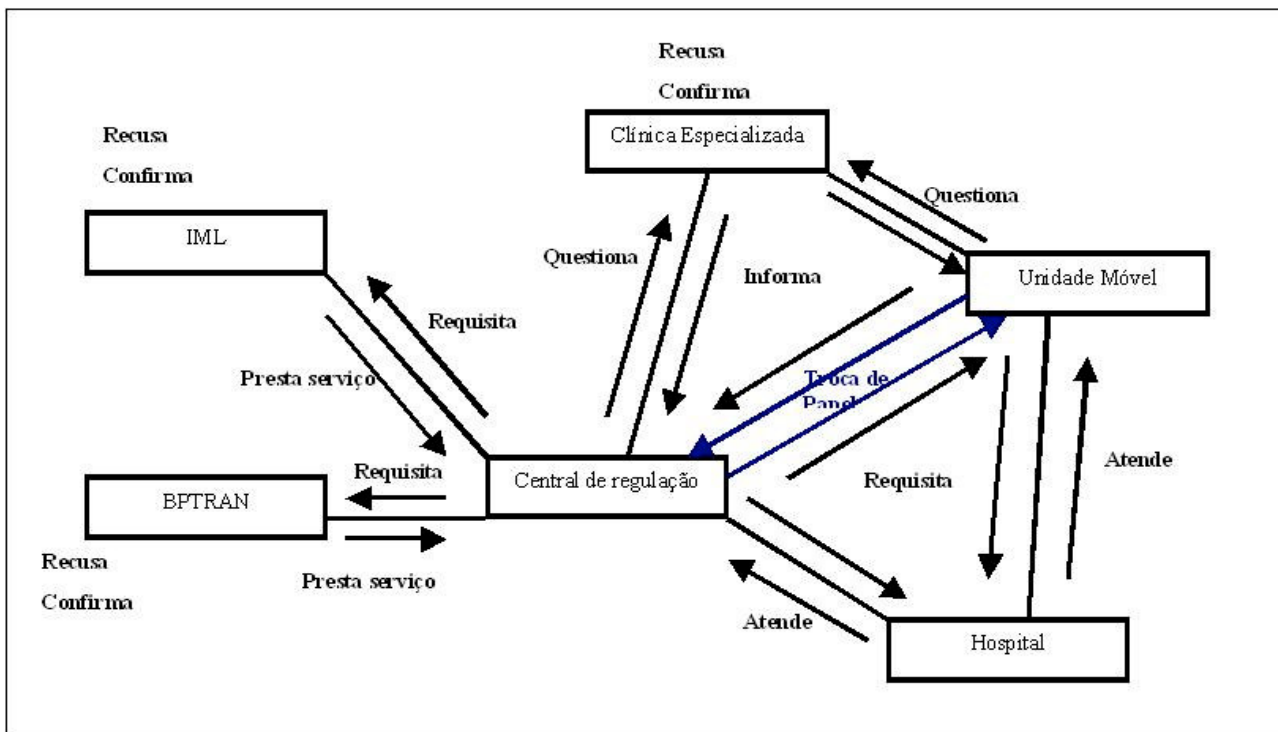


Figura 36: Diagrama de colaboração com algumas situações do atendimento pré-hospitalar

O diagrama de atividades é uma estrutura que ilustra os processos que são executados e as interações desses processos entre os agentes. Essa interação permite a visualização da execução das tarefas que são de responsabilidade de cada entidade participante. Uma particularidade interessante é a possibilidade, com este tipo de visualização, de se conhecer de quais agentes existe a necessidade de participação e, principalmente, conhecer em qual momento é mais ou menos importante a participação de uma determinada entidade.

O diagrama de atividade, independente do diagrama de seqüência, não estipula uma seqüência cronológica para o acontecimento dos relacionamentos, ele apenas coloca as atividades que deverão ser executadas divididas para cada agente. A estrutura prevista pela AUML para o diagrama de atividades é bem similar à apresentada pela UML. A diferença básica destes dois diagramas é a presença do agente na AUML, ao invés de um objeto, como é o caso da UML. As atividades do agente podem ser extensas e complexas, sendo que o agente poderá possuir apenas uma atividade, ou ainda particionada, quando o agente apresenta um volume maior de atividades. Para se entender melhor a extensão da UML, um objeto possuirá suas atividades selecionadas dentro de uma mesma estrutura negocial. A apresentação dos agentes acontece na mesma disposição do diagrama de seqüência, ou seja, na linha superior do diagrama, as atividades devem necessariamente estar dispostas uma embaixo da outra na mesma coluna do agente que a possui. O relacionamento das atividades acontece diretamente

pele apontamento de uma seta ligando uma atividade a outra. Desta forma, a visualização da integração dos agentes de um sistema passa a acontecer no nível das tarefas executadas por cada agente. Desta maneira, os processos também se tornam mais simples de serem analisados e pensados, podendo definir de forma mais precisa quais os papéis e as necessidades de um determinado agente dentro da organização proposta.

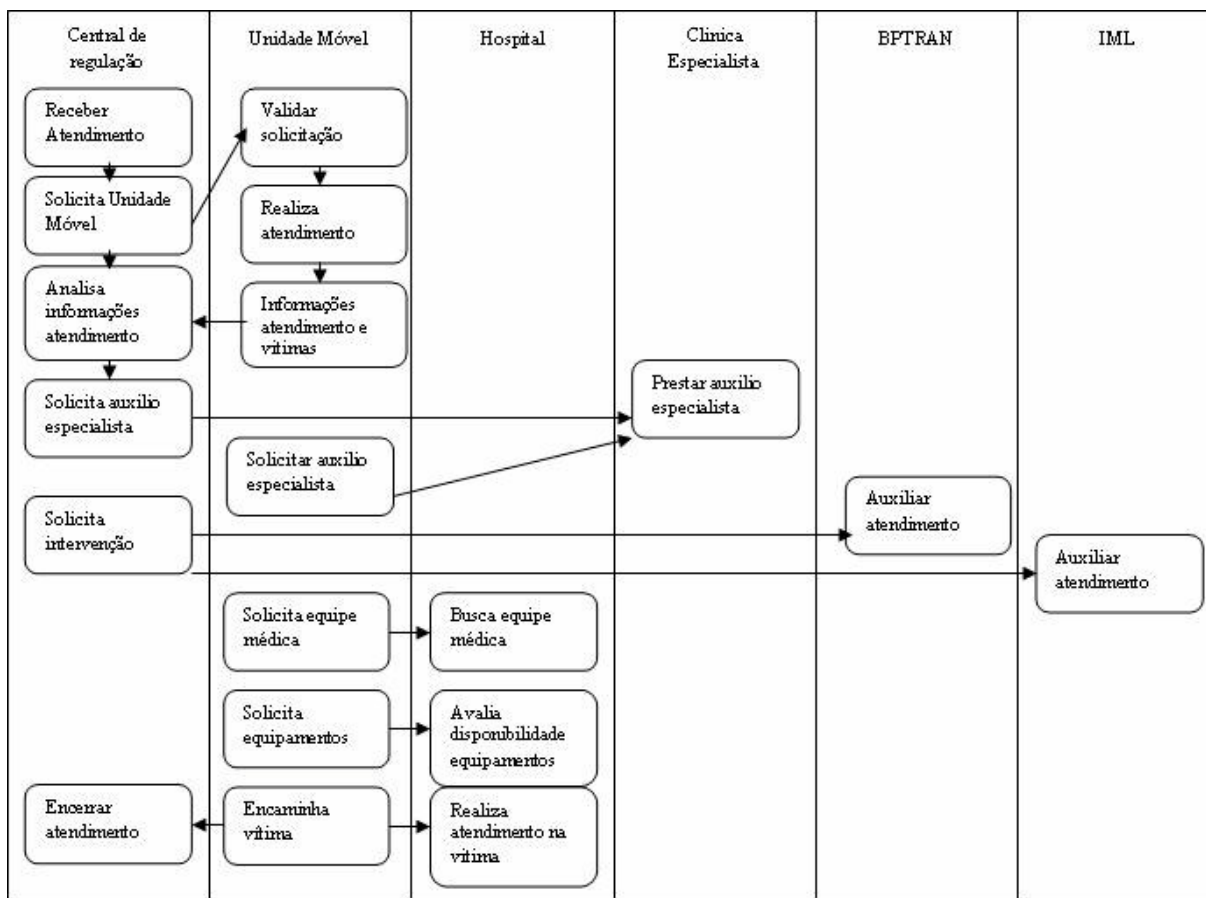


Figura 37: Diagrama de atividades com algumas atribuições do atendimento pré-hospitalar

O diagrama de estados da AUML tem basicamente o mesmo apelo gráfico do diagrama de estado da UML. Na AUML ele representa os estados possíveis de um agente; na UML, o mesmo representa os estados possíveis de um objeto. Quando se fala de sistemas multi-agente, é importante estabelecer que um estado pode significar uma restrição de uso para o agente. Todo agente constituinte de um sistema possui seu conjunto de estados. Esses estados devem trazer informações a respeito da atuação do agente, facilitando sua interação com o sistema. Dessa forma, os estados funcionam como uma restrição estabelecida pelo agente à sua operação dentro do sistema. Dos diagramas apresentados na AUML, o diagrama de estados é o responsável por mostrar as disponibilidades de um agente. Na figuras 38, 39 e 40

são apresentados os diagramas de estados do hospital, *central de regulação* e *unidade móvel*, respectivamente.

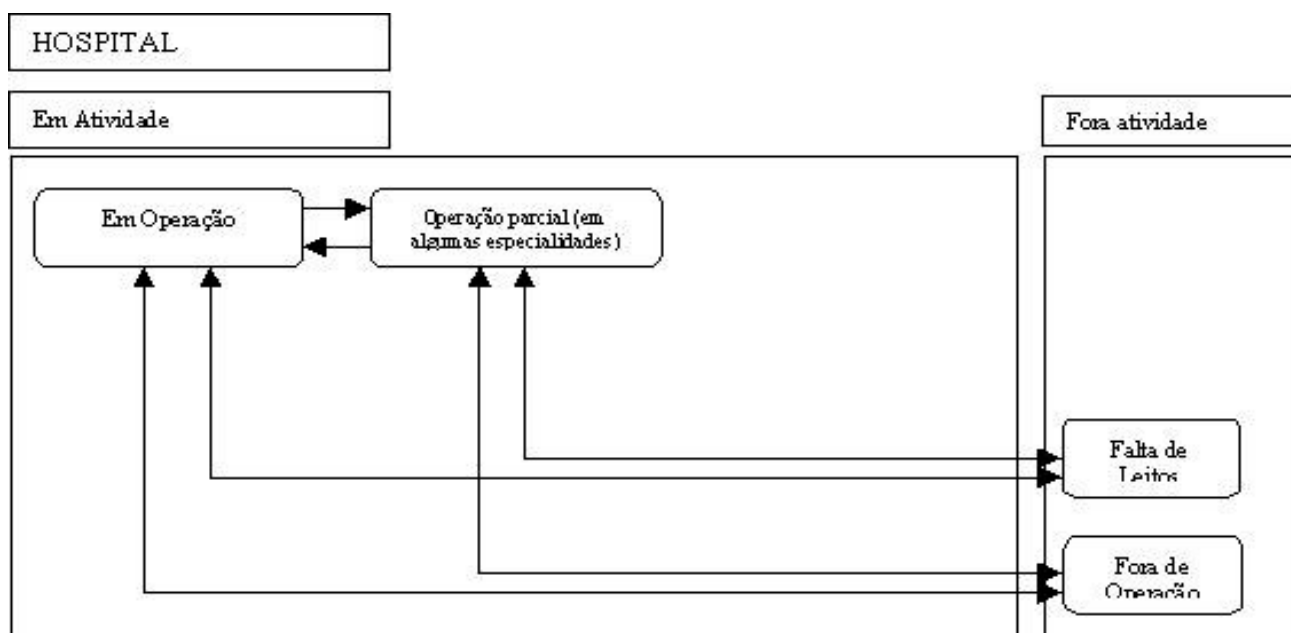


Figura 38: Diagrama de estados de um hospital



Figura 39: Diagrama de estados da *central de regulação*

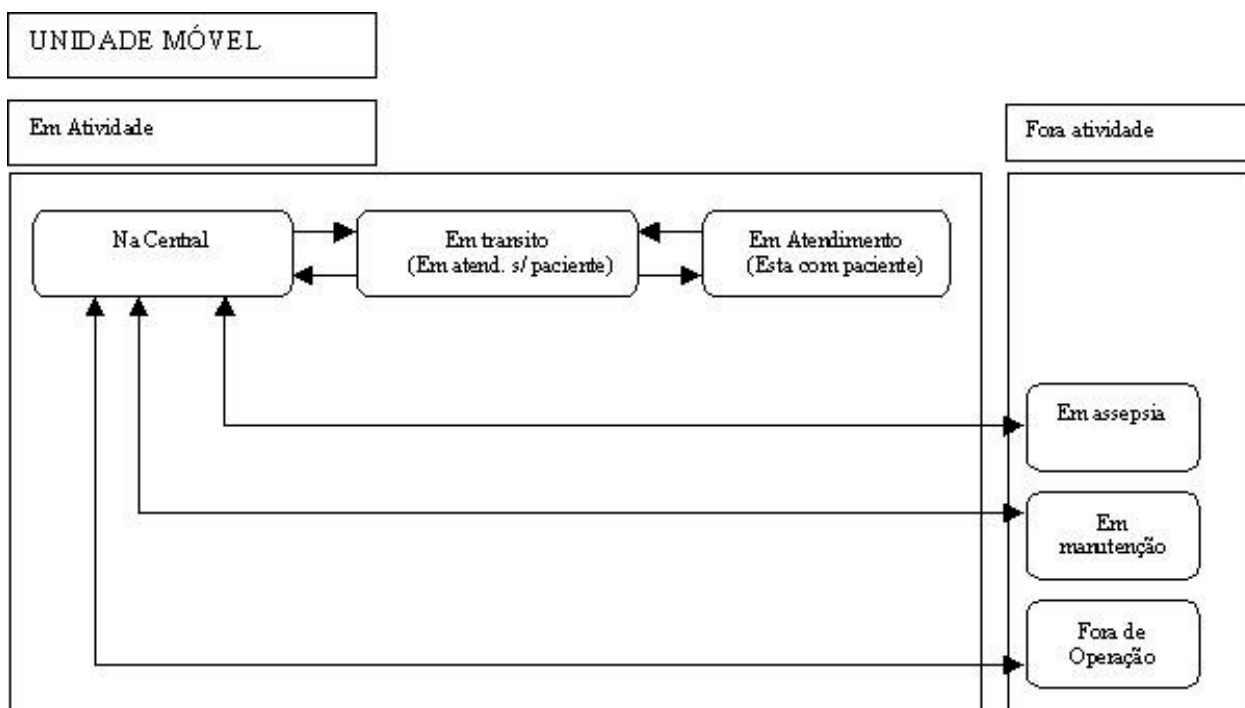


Figura 40: Diagrama de estados da *unidade móvel*

Nos diagramas de estados propostos estão selecionados alguns estados que os agentes do atendimento pré-hospitalar podem possuir. Pode-se observar que os estados indicam uma situação em que o agente está sendo submetido no momento da consulta. A divisão dos estados em que o agente encontra-se em atividade e fora de atividade deixa claro o objetivo do diagrama de estados em atuar como um levantamento de situações que restrinjam a atuação de um agente no sistema proposto.

É importante ressaltar mais uma vez que o universo do atendimento pré-hospitalar pode ser compreendido por situações mais amplas que as que foram expostas nessa modelagem, devido ao número maior de agentes que possam vir a fazer parte do sistema SAAP.

### 5.3 Arquitetura e organização da comunicação entre agentes

Independente da arquitetura, os agentes devem possuir informações de sua localização e de suas competências. A diferença é a forma de armazenamento e recuperação dessas informações. Os modelos de arquitetura expostos a seguir preocupam-se em prover uma forma de comunicação entre dois ou mais agentes:

Quadro Negro -- Neste tipo de arquitetura, todos os agentes de um determinado contexto estão publicados em um único lugar de acesso. Todos os agentes, antes de solicitarem comunicação com os demais, devem necessariamente buscar este local. No quadro

negro encontram-se registrados o nome do agente, a localização e a competência de cada agente integrante da arquitetura. A única forma de um determinado agente conseguir se comunicar dentro da arquitetura é estando registrado no quadro negro. A desvantagem dessa arquitetura ocorre quando o quadro negro está indisponível (sem comunicação). Neste caso, nenhum agente da rede poderá se comunicar. Outro problema é com relação à concorrência de acesso ao quadro negro, pois todos os agentes estarão concorrendo com solicitações a um mesmo local. Esse problema pode ser agravado quando a rede possui um grande número de agentes com muitas transações, ocasionando problemas de desempenho [Paraiso, 1996]. A troca (ou compartilhamento) de informações, por meio de um quadro-negro, pode ser relativa ao artefato ou ao controle da solução de um problema. Os fatos iniciais e os diferentes resultados parciais representam as versões sucessivas do artefato (atendimento em andamento) que é armazenado neste tipo de estrutura de dados. O quadro-negro é o único meio de troca de informações. Esta arquitetura possui a vantagem de assegurar a coerência das trocas, mas tem o inconveniente de ser potencialmente um gargalo de estrangulamento. Este modelo foi utilizado em inúmeras aplicações, notadamente na concepção em engenharia. Shen & Barthes [Shen & Barthes, 1996] apresentam uma longa lista de projetos que utilizam a arquitetura quadro-negro no domínio da concepção. Pode-se também encontrar uma discussão detalhada sobre a abordagem quadro-negro em Hayes-Roth [Hayes-Roth, 1988] e em Corkill et al. [Corkill et al, 1986].

Na estrutura de sistemas multi-agente, os agentes complexos, chamados de facilitadores, organizam o trabalho entre os agentes mais simples [Genesereth & Ketchpel, 1994]. O processo de coordenação é conduzido a um facilitador; os agentes de serviços notificam o facilitador das tarefas que eles podem tratar e, quando um agente envia um pedido ao facilitador, posteriormente é localizado um agente competente para executar a tarefa, então, o facilitador retorna o resultado para ser transmitido ao agente que o solicitou. A arquitetura facilitadora racionaliza recursos de comunicação. Porém, devido ao facilitador operar como uma ponte entre agentes, seu fracasso pode impedir a comunicação entre os mesmos.

Uma outra abordagem é dada por Bouron [Bouron, 1993], quando concebe o ambiente como sendo um agente, ou seja, o *black board* passou a ser denominado agente de ambiente. Considerando o exemplo dos robôs futebolistas, a modelagem do ambiente seria facilmente realizável na forma de um agente. Isto teria a vantagem de simplificar certos aspectos da implementação. Por exemplo, um jogador pode solicitar ao agente ambiente onde se encontram seus colegas de equipes, a bola etc. Porém, a centralização causada por esta estrutura torna o ambiente um gargalo de estrangulamento. Guessoum [Guessoum, 1996]

salienta que, se o ambiente é um agente (uma entidade ativa e, sobretudo autônoma), a ação e a percepção se tornam mais complicadas. Já para Scalabrin [Scalabrin, 1996], o ambiente deve ser independente e os agentes devem possuir mecanismos suficientemente robustos para superar as possíveis dificuldades. Além disso, o ambiente de um sistema multi-agente pode ser representado como um agente ativo e autônomo de uma maneira natural. O problema da percepção e da ação sobre o ambiente não deve aparecer quando as condições ambientais são respeitadas. Os agentes, residindo em um ambiente, percebem os eventos através das perturbações que o ambiente provoca sobre eles.

A troca direta de mensagem ocorre quando um agente possui o endereço e a competência de cada agente integrante da arquitetura. Todos os agentes participantes dispõem de uma área reservada para o armazenamento das informações dos demais agentes. Para a entrada ou saída de um novo agente na rede, é necessário que o agente comunique aos outros agentes a sua identificação e as suas competências.

Os agentes da rede deverão manter o endereço e as competências de cada agente. Cada agente poderá comunicar diretamente com outros agentes sem intervenção de outra estrutura. Caso um determinado agente esteja sem comunicação, apenas este não trocará informações, os demais estarão realizando a comunicação normalmente. Cada agente constrói seu próprio modelo da solução em andamento para um problema através da aquisição de informações provenientes de outros agentes. Este modelo de troca requer a especificação de um protocolo e de um formato de mensagem (linguagem comum) para expressar demandas e fornecer resultados. Cada agente armazena a solução em andamento do problema em questão, ou uma parte desta solução, em sua base de fatos local. Geralmente, em um sistema multi-agente, seus componentes são heterogêneos e não existe controle global; a comunicação entre eles pode ser <sup>1</sup>síncrona ou <sup>2</sup>assíncrona; ela pode ser <sup>3</sup>ponto-a-ponto, <sup>4</sup>broadcast ou <sup>5</sup>multicast; e, finalmente, a semântica das mensagens trocadas entre os agentes é de alto nível. Certo número de projetos, tais como MARS [Abriat, 1991], PACTO [Cutkosky et al, 1993], DIDE [Shen & Barthes, 1995], utilizam ou têm utilizado este tipo de arquitetura. Shen & Barthès [Shen & Barthes, 1996] apresentam igualmente uma longa lista de projetos no domínio da engenharia. Eles indicam, em particular, que a comunicação assíncrona, presente nos sistemas multi-agente, convém a um ambiente distribuído no contexto de grandes projetos. Já a comunicação

---

<sup>1</sup> síncrona = em tempos iguais

<sup>2</sup> assíncrona = em tempos diferentes

<sup>3</sup> ponto-a-ponto = comunicar de um lugar *a* para um lugar *b* definido

<sup>4</sup> broadcast = enviar mensagens a todos os agentes da rede

<sup>5</sup> multicast = enviar mensagens a todos os agentes inclusive os de rede externa a que o agente está registrado



síncrona (estilo quadro-negro), bastante utilizada entre agentes que devem trabalhar simultaneamente, não é apropriada aos problemas encontrados no escopo de projetos de grande porte, para os quais o período de concepção é freqüentemente longo [Shen & Barthes, 1995].

A arquitetura desenvolvida para o atendimento pré-hospitalar pode ser definida como uma arquitetura híbrida. Foi desenvolvido, em caráter de teste, um agente ambiente que é responsável em manter informações dos agentes pertencentes à rede. Este agente é o que necessariamente detém todos os endereços e as competências dos agentes que pertencem ao sistema de agentes do atendimento pré-hospitalar. Os agentes do atendimento pré-hospitalar, por se tratar de uma rede onde o quesito desempenho é crítico, possuem também em seu modelo uma memória responsável por armazenar os endereços dos agentes que participam da rede. Dessa forma, toda comunicação poderá acontecer de forma direta e, toda vez que for necessário, um agente poderá realizar um processo de atualização dos agentes da rede através de uma consulta simples ao agente de ambiente. A intenção é possibilitar que as duas formas de arquitetura estejam vigentes e disponíveis ao sistema multi-agente do atendimento pré-hospitalar.

### **5.3.1 Arquitetura e organização do sistema SAAP.**

O modelo interno de um agente também é uma característica importante [Scalabrin, 1996]. O modelo de um agente estrutura mecanismos que podem permitir o agente: a) planejar e coordenar suas ações; b) argumentar sobre as convicções e intenções de outros agentes (desta forma, escolhendo o que comunicar com os agentes) [Genesereth et al, 1988]; c) coordenar suas ações sem comunicação [Smith, & Davis, 1981]; d) tomar decisões locais (controle descentralizado); e) reduzir comunicação (somente informação útil é comunicada, com assinatura KQML [Finin et al, 1993]); f) saber quem faz o quê e para quem é útil pedir informação [Halfpap & Belli, 1997].

Desta forma, todo agente constante de uma arquitetura possui um modelo de funcionamento. No modelo definido neste protótipo, o agente é subdividido em algumas camadas:

- Comunicação e Percepção: camada onde é implementada a forma do agente comunicar com os demais, na rede em que está registrado. Toda comunicação do agente será realizada por recebimento e envio de mensagens, que, neste caso, serão *performativas* KQML. Este modelo de agente tem a capacidade de perceber o ambiente e postar as respostas ao que foi questionado.

- Núcleo de Processamento: o processamento é responsável pelas respostas que o agente estará fornecendo. O núcleo de processamento estará utilizando para elaboração das solicitações mais dois módulos, descritos abaixo:
- Representação de si: este módulo guarda os dados e informações utilizadas pelo agente para processar as respostas necessárias. São armazenadas aqui também as capacidades e as tarefas do agente.
- Representação externa: este módulo guarda as informações do ambiente que o agente está envolvido, principalmente das capacidades e tarefas (objetivos e conhecimento) dos demais agentes.

O modelo de agente adotado para a construção do sistema SAAP inspirado nos modelos vistos anteriormente, conforme ilustra a figura 41.

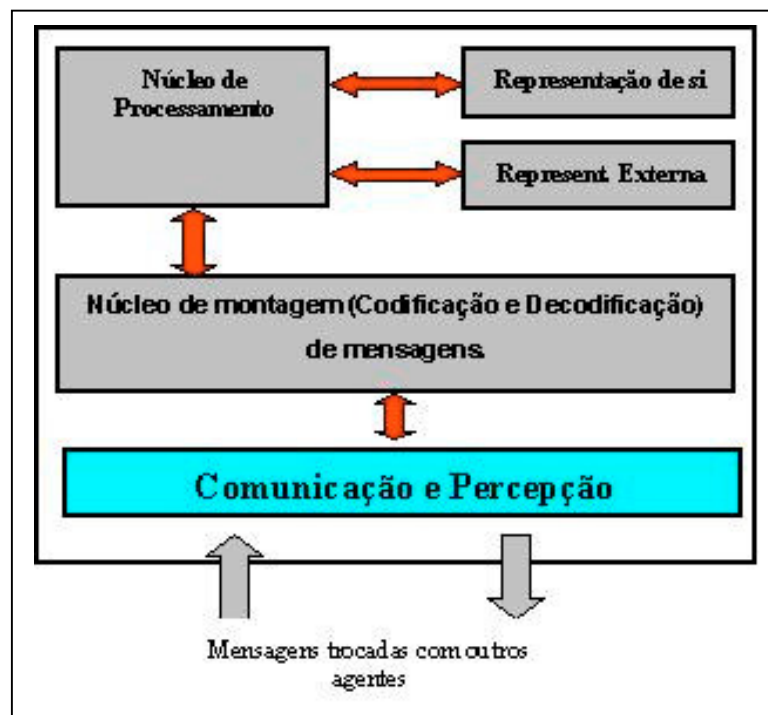


Figura 41: Modelo de Agente

Após a definição do agente, é importante definir como estes agentes estarão interagindo com o ambiente. O ambiente definido para operação dos agentes que são necessários a um atendimento pré-hospitalar pode ser alterado, independente de quais instituições estejam envolvidas. Nesse contexto, será necessária a colaboração de três agentes importantes no atendimento pré-hospitalar:

- central de regulação*;
- unidade móvel*;

c) hospital ou clínica.

Dependendo da alteração do ambiente, outros agentes podem fazer parte do atendimento, por exemplo, em uma catástrofe, podem ser alocadas quantas unidades móveis, centrais de regulação, hospitais e demais entidades forem necessárias.

As entidades colaboram no atendimento por meio de uma solicitação ou por iniciativa própria. A rede de agentes do atendimento pré-hospitalar ainda possui a característica de registrar e desregistrar agentes participantes, o que traz a possibilidade de contribuições de outras instituições a qualquer ocorrência que esteja sendo atendida. Esta característica de rede aberta é ilustrada na Figura 42, junto com alguns agentes que podem fazer parte de um atendimento. Por se tratar de uma rede aberta, a ilustração não representa todos os agentes que poderão participar de um atendimento, apresenta apenas alguns dos agentes para o atendimento pré-hospitalar.

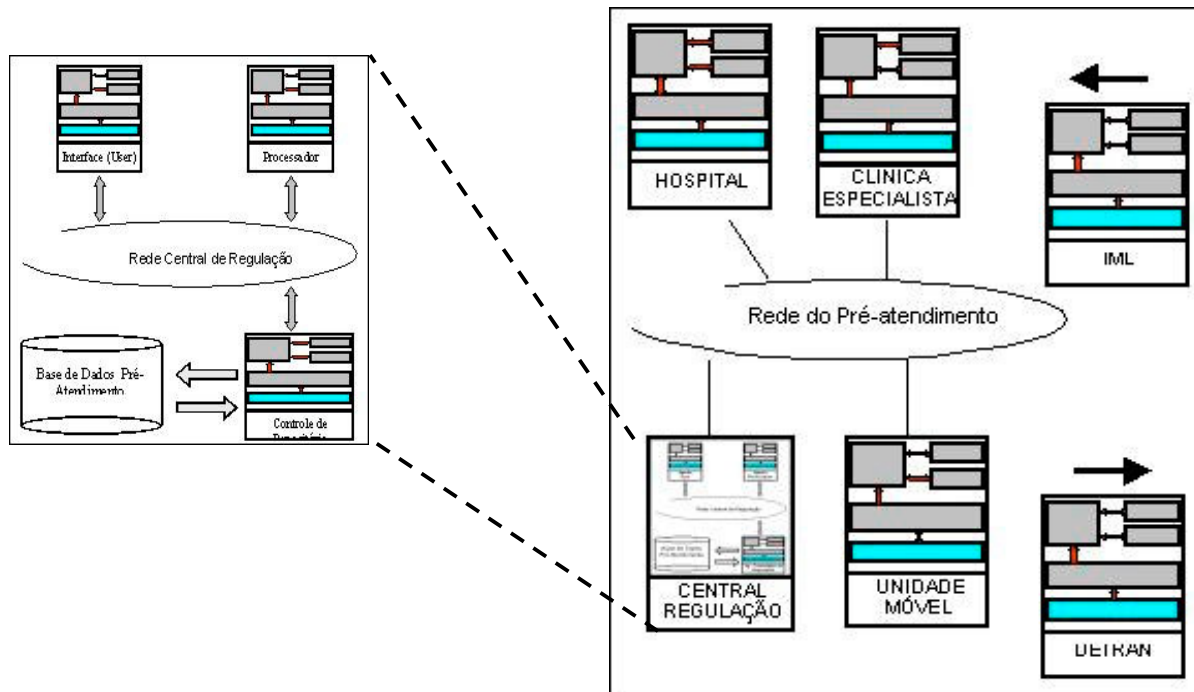


Figura 42: Estrutura básica da rede do atendimento pré-hospitalar

Cada agente envolvido na rede do atendimento pré-hospitalar pode ser integrante de outras redes, como é o caso da *central de regulação*. A construção de uma arquitetura de agentes visa facilitar o acesso e a atualização das informações decorrentes do atendimento pré-hospitalar. A proposta para atender ao atendimento pré-hospitalar é a utilização de uma rede interna para controlar as informações específicas. Essa rede seria abastecida por todos os agentes que estiverem participando do atendimento. Cada agente do sistema possuirá uma

configuração específica para sua rede interna, não sendo do intuito do sistema SAAP controlar a organização interna das entidades participantes.

Para que um agente possa participar do atendimento pré-hospitalar, ele deverá utilizar o protocolo e a linguagem de comunicação definidos. A partir do momento da apresentação de um novo agente para a rede, esse agente passará a contribuir com suas informações e competências.

### **5.3.2 Arquitetura e organização do agente *Central de regulação***

Para atender a essa flexibilidade, o agente *central de regulação* necessitará de uma estrutura onde deverão ser definidos papéis distintos a cada membro participante. Esta estrutura é exclusiva para atender ao atendimento pré-hospitalar e não necessariamente deverá ser a mesma de outros agentes participantes. A intenção de criar estes conceitos e definições (papéis) é de organização. Cada agente interno da estrutura possuirá suas atribuições para que todas as informações de um atendimento pré-hospitalar possam ser mantidas e gerenciadas. Esses papéis podem ser melhor visualizados no diagrama de seqüência da Figura 35, onde ficam explícitos os possíveis papéis para os agentes *unidade móvel* e hospital.

### **Controle de Repositório**

Todas as solicitações de informações serão dirigidas a este agente, assim como solicitações de inclusão e manutenção das informações. É necessária a existência desta estrutura sempre que se desejar controlar as informações de um determinado contexto. Nesse caso, é o atendimento pré-hospitalar que estará sendo controlado. A arquitetura deverá possibilitar que as informações do atendimento pré-hospitalar sejam utilizadas por qualquer agente integrante da rede.

### **Interface**

Essa atribuição, que também pode ser incumbência de um agente, é responsável em realizar a integração com as demais estruturas que fazem parte do atendimento pré-hospitalar. Essas Interfaces podem ser hardware, software ou outros agentes. Esta arquitetura existe para possibilitar a portabilidade de um atendimento pré-hospitalar. A comunicação com demais agentes passa a ser imprescindível quando o sistema é aberto para a utilização por órgãos que trabalhem de maneira distinta, inclusive em locais físicos diferentes.

## Processamento

Será responsável pelos processamentos necessários para o atendimento pré-hospitalar, operando em conjunto com o controle de repositório. O processamento estará trabalhando para o atendimento pré-hospitalar e os mantenedores do negócio poderão lhe solicitar as informações mantidas por todos os participantes da rede.

Uma flexibilidade deste modelo é a possibilidade de que a rede seja implementada em qualquer outro contexto. A contextualização do negócio passa a ficar ao encargo dessas entidades, e, se elas estiverem trabalhando, por exemplo, para uma rede que atenda aos sistemas de um hospital, o negócio a que estará tratando será de um hospital. Isto torna portátil uma arquitetura. A figura 43 mostra a estrutura de funcionamento da rede interna do agente *central de regulação*

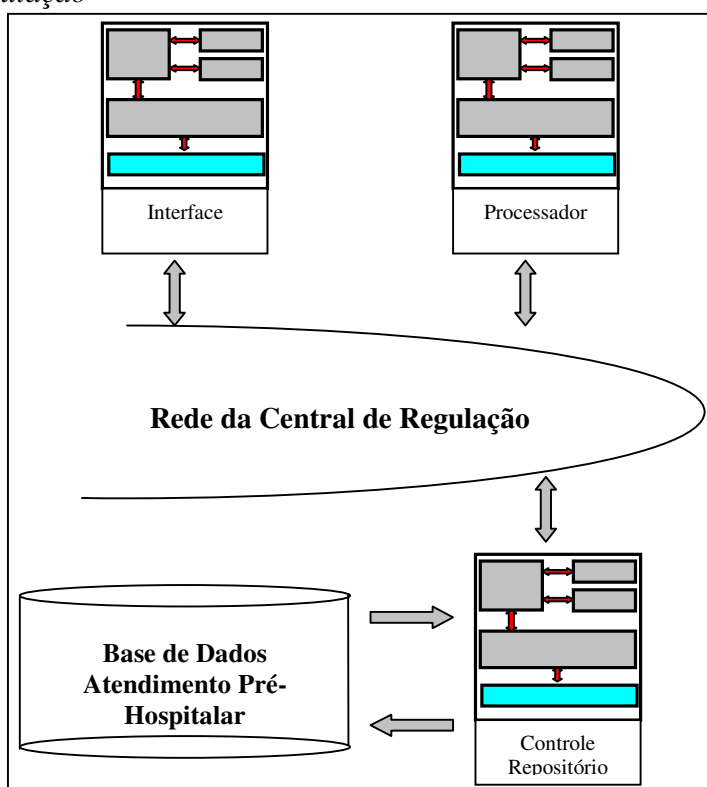


Figura 43: Estrutura do Agente *Central de regulação*

A arquitetura desenvolvida para o atendimento pré-hospitalar conta com a presença de um agente *central de regulação* responsável por conter as informações dos chamados que regula, conforme mostra a Figura 43. Como exemplo de solução utilizou-se também agente. Ao contrário do contexto do atendimento pré-hospitalar, o agente *central de regulação* poderá conter em sua estrutura uma aplicação que não necessariamente utilize um sistema multi-agente. O modelo de agente e os modelos de arquitetura já são conhecidos, e, neste caso, procurou-se organizar os agentes para disponibilizar as informações do atendimento pré-

hospitalar. Dessa forma, foi criada uma estrutura para acesso à base de dados, que estará sendo mantida e consultada pelos agentes participantes da rede. A idéia é que toda solicitação de informação e toda manipulação de dados sejam realizadas através desta estrutura.

O fluxo das informações ocorrerá conforme indicado. As solicitações da rede serão captadas pela área de comunicação e percepção da interface, cuja principal funcionalidade é justamente realizar esta interface. Após identificadas e filtradas, no sentido de responder se a rede poderá ser atendida, as solicitações são encaminhadas à área de processamento, responsável por identificar qual conjunto de dados é necessário para responder a solicitação que lhe foi encaminhada pela interface. O processamento deverá repassar ao controle de repositório todos os dados necessários para atender à solicitação. O controle de repositório elaborará as consultas necessárias à base de dados da rede do atendimento pré-hospitalar, assim como as manipulações de dados exigidas pelo negócio. O fluxo de informações retorna à sua origem, passando novamente pelos agentes que fizeram o seu encaminhamento. Dessa forma, o controle de repositório devolve os dados solicitados pelo processamento, que monta a solução com as informações solicitadas e devolve à interface, que disponibiliza a informação montando a resposta ao agente da rede que a solicitou.

Esta forma de aplicar agentes visa apresentar uma utilização diferente para a mesma arquitetura de agente proposta para as demais organizações do atendimento pré-hospitalar. É importante ressaltar que existem dois níveis no sistema proposto. Um que atende ao negócio atendimento pré-hospitalar e outro que atende ao negócio *central de regulação*. Enquanto que na *central de regulação* proposta um agente de *processamento* realizará comunicação apenas com um agente *interface* e um agente *controlador de repositórios*, um agente *hospital*, por exemplo, atua em outro nível e poderá realizar comunicação com agentes que não foram previstos neste trabalho. Porém, a hipótese de criação de novos agentes dentro do agente *central de regulação* não pode ser descartada. A cada nova necessidade de inclusão de funcionalidade em uma *central de regulação*, um novo agente poderá ser acoplado ao sistema, não alterando o funcionamento dos demais agentes desta rede.

### **5.3.3 Implementação da arquitetura SAAP**

O atendimento pré-hospitalar é um contexto multidisciplinar que envolve entidades de naturezas distintas, evidenciando a necessidade de comunicação e colaboração mútua entre as entidades. Nesse caso, a ação coletiva entre os agentes é suportada pela capacidade de comunicação dos agentes [Bond & Gasser, 1988; Demazeau, 1990; Wooldridge & Jennings,

1994]. O processo de comunicação entre agentes complexos é normalmente assíncrono, e é executado por meio de vários protocolos.

Na arquitetura proposta foram utilizadas *performativas* KQML como forma de comunicação entre os agentes participantes e adotado o *Contract Net Protocol* para protocolo de negociação. Dessa forma, um agente que desejar participar de um atendimento pré-hospitalar deverá adicionar a camada de comunicação implementada, suportando a linguagem KQML.

O desenvolvimento da arquitetura teve como agentes de teste a *central de regulação* e a *unidade móvel*. Como se optou pelo desenvolvimento dos agentes e de toda a infra-estrutura, a preocupação ficou mais concentrada na possibilidade de se apresentar a comunicação, a disponibilidade desta arquitetura no ambiente Web e a troca de mensagens significativas entre os protótipos de agentes. Partindo do princípio que cada entidade constante do atendimento pré-hospitalar possui a sua estrutura já concebida, as suas interfaces, o seu conjunto de informações sendo tratado, e, principalmente, a possibilidade de se utilizar em um atendimento uma organização qualquer, o desenvolvimento da arquitetura baseou-se em tornar possível esta comunicação de maneira que o administrador das aplicações que desejarem compartilhar da rede de atendimento pré-hospitalar tenha que despender pouco ou nenhum recurso de manutenção em suas próprias aplicações. Foi desenvolvida uma camada KQML com algumas performativas, que deverá ser acoplada em todas as aplicações que desejarem participar do atendimento pré-hospitalar e ainda não possuam a sua própria camada desenvolvida.

Atualmente, todas as organizações que participam do atendimento pré-hospitalar possuem um sistema mantenedor de dados que atende às necessidades mínimas de registro de informações. Essas aplicações, por mais simples que sejam, atendem a alguns requisitos: o registro das ocorrências realizado no órgão responsável pelo registro e acompanhamento dos chamados, aqui denominado de agente *central de regulação*, um sistema capaz de registrar algumas informações das unidades móveis que realizam um atendimento pré-hospitalar, aqui denominado de agente *unidade móvel*, um sistema de informações hospitalares, que atende de maneira diferente, os indefinidos hospitais públicos ou particulares que poderão participar de um atendimento, aqui denominado genericamente de agente *hospital*, clínicas particulares ou públicas com suas aplicações aqui denominadas de agente *clínica especialista*, serviços de apoio, exemplificados aqui como agentes *BPTRAN* (representando o apoio necessário às intervenções no trânsito causadas por um atendimento), agente *IML* (serviço de apoio que poderá ser instanciado em qualquer atendimento). Todos os agentes denominados aqui

possuem, em suas organizações, necessidades que são mais bem atendidas por sistemas de informações locais. O atendimento pré-hospitalar necessita das habilidades de todas estas organizações, necessita compartilhar informações para melhorar a eficiência e a eficácia dos atendimentos, precisa recuperar as informações dos atendimentos ocorridos para identificar melhoramentos, apontar falhas de processo que possam ser corrigidas. O custo operacional de um atendimento onde exista a necessidade de deslocamento médico é alto. Os serviços de atendimento pré-hospitalar particulares crescem devido a uma necessidade que não consegue ser suprida [Brasil, 2002b]. Estes serviços de atendimento pré-hospitalar particulares podem contemplar a população em geral.

A arquitetura desenvolvida visa unir as forças de todas as organizações citadas e muitas outras que possam, de certa forma, cooperar com o auxílio a vítimas em situações de emergência. Dentro da arquitetura desenvolvida, não importa a qual serviço à *unidade móvel* pertence, mas sim qual atendimento será realizado com uma estrutura capaz de atender as solicitações e aos casos mais adversos, sem a necessidade de se alterar ou manipular aplicações construídas. A integração destas organizações vai além de uma comunicação via rádio. As informações que trafegam em um atendimento podem ser utilizadas para auxiliar em outros atendimentos, como na necessidade de preparar um ambiente em um *hospital* ou *clínica* que atenderá um paciente de atendimento pré-hospitalar, ou na realização de um procedimento especialista devidamente orientado por uma junta de especialistas dentro de uma *unidade móvel* (como, por exemplo, unidades móveis do serviço das concessionárias de pedágio em acidentes que ocorrem a uma distância maior de um hospital ou clínica).

A possibilidade de inclusão de novas funcionalidades em uma aplicação sem a real necessidade de se alterar os códigos já implementados e em produção é perseguida pelos projetistas da área de engenharia de software. Como o contexto do atendimento pré-hospitalar tem a particularidade de não fornecer todos os requisitos que atenderão as suas necessidades, a arquitetura possibilita que sejam inseridas funcionalidades, ou módulos inteiros, no formato de agente, sem ter-se que alterar códigos das soluções que já estão atuando. Um agente específico para definição de rotas para as unidades móveis, ou um agente responsável por auxiliar os médicos na tomada de decisão, ou ainda um agente específico para controlar as disponibilidades de cada hospital, levando em consideração suas especialidades. Estes são apenas exemplos de novas funcionalidades, e alguns deles já foram inclusive citados neste trabalho. A hipótese de inclusão de novas funcionalidades é latente neste contexto e a arquitetura foi desenvolvida com a capacidade de suportar estas situações.



As necessidades da aplicação da arquitetura são grandes e, para tanto, a implementação realizada possibilita que organizações em distintos locais sejam interligadas pela Web. Atualmente, a solução parte de um *Web Service* que disponibiliza as funcionalidades dos diferentes agentes e consegue integrar uma aplicação local com um agente, independente da disposição física destes órgãos.

## Capítulo 6 – Resultados e Discussão

Com o intuito de mostrar o funcionamento e a organização do sistema proposto, foi construído um protótipo capaz de realizar a comunicação e trocar informações entre os agentes que pertencem ao atendimento pré-hospitalar. As informações pertinentes a um atendimento, que fazem parte exclusivamente do domínio de um agente específico, puderam ser acessadas por outro. Este contexto distribuído a que o sistema SAAP está submetido exige a presença de uma tecnologia de comunicação que viabilize a troca de mensagens entre os agentes participantes da arquitetura. A tecnologia utilizada para resolver esta questão no protótipo desenvolvido foi o *Web Service*.

Um *Web Service* é “uma aplicação que é acessível por programas através de protocolos *web* de uma maneira independente de plataforma” [Kalani, 2002]. A aplicação desta tecnologia é baseada na premissa da utilização de padrões conhecidos e testados. O grupo de empresas que atua junto aos padrões que mantém a tecnologia *Web Service* trabalha junto ao W3C<sup>6</sup>, entidade responsável pela padronização de diversas linguagens e protocolos de Internet, em uma proposta visando à padronização do SOAP (Simple Object Access Protocol) e de outros padrões que dão suporte ao *Web Service* [Hendricks, 2002]. Uma aplicação que possa ser acessada por outras através de protocolos *web* é uma definição que padrões como CORBA e DCOM também compartilham. Ser independente de plataforma é a parte da definição de *Web Service* que interessa ao atendimento pré-hospitalar. *Web Service* torna pública uma funcionalidade que a partir de então pode ser acessada por qualquer aplicação, independente de sua implementação, para tanto é necessário a utilização de mais alguns padrões igualmente conhecidos: o XML (eXtensible Markup Language): responsável pelo conteúdo que será transmitido; HTTP (Hyper Text Transfer Protocol): protocolo de pergunta – resposta (request - response) por onde trafegam os conteúdos; SOAP: responsável pela padronização da mensagem que será transportada; WSDL (Web Services Description Language): linguagem responsável por esclarecer sobre a sintaxe, parâmetros, funções ou métodos que serão trocados na comunicação; UDDI (Universal Description, Discovery and Integration Services): diretório onde estão descritos os serviços disponíveis para um *Web Service* [Brasiura, 2003].

---

<sup>6</sup> W3C = Grupo responsável em manter os padrões utilizados no ambiente *web*.

As mensagens trocadas pelos agentes são baseadas em *performativas* KQML, como será visto mais adiante. Porém, estas *performativas* devem ser transferidas de uma organização a outra. Para que esta comunicação seja possível o *Web Service* utiliza o protocolo SOAP, que é um formato de mensagens baseado em XML, usado para passar informações entre duas localidades ou “extremidades”, onde estas informações são armazenadas em um tipo de envelope [Amundsen & Litwin, 2002]. Este conjunto de elementos XML (um documento XML) realiza a tarefa de descrever as formas usadas para a transmissão da mensagem. Ou seja, SOAP é um protocolo padrão para troca de mensagens e, com esta propriedade, ele possibilita que qualquer aplicação invoque e responda a qualquer outra aplicação. O *Web Service* desenvolvido para o protótipo do sistema do atendimento pré-hospitalar foi implementado utilizando a linguagem Delphi 7, ferramenta que encapsula a utilização do SOAP via protocolo HTTP de maneira rápida e eficiente [Wildt, 2002]. Os componentes desta aplicação criam os arquivos XML que serão trafegados dentro do protocolo SOAP com todas as informações que deverão ser transmitidas, neste caso as informações contidas dentro das *performativas* KQML. Para tanto, as informações que deverão ser transmitidas devem ser devidamente tratadas e informadas a ferramenta do compilador que realiza este encapsulamento. Desta forma, em uma comunicação entre o agente *central de regulação* e o agente *unidade móvel*, por exemplo, aconteceriam os seguintes passos: 1) O agente *central de regulação* empacota dentro do XML do protocolo SOAP uma mensagem que contém uma *performativa* KQML. 2) A mensagem SOAP é transportada via protocolo HTTP para o *Web Service* do agente *unidade móvel*. 3) O *Web Service* do agente *unidade móvel* desempacota o XML da mensagem SOAP recebida e encaminha a *performativa* para a camada de comunicação KQML. 4) A camada de comunicação realiza sua inferência sobre o agente, em busca de uma solução. 5) O agente *unidade móvel* monta o XML da mensagem SOAP com a resposta da solicitação do agente *central de regulação* e encaminha esta resposta via protocolo HTTP.

Para que um agente consiga entender o que o outro está disponibilizando o *Web Service* possui uma linguagem específica. O WSDL é uma linguagem de definição de interface que possibilita a descrição das funcionalidades, métodos e atributos disponibilizados pelo *Web Service* [Amundsen & Litwin, 2002]. Dentro do protótipo desenvolvido, os métodos disponibilizados no WSDL são os correspondentes as *performativas* KQML

A Figura 44 ilustra a forma que as mensagens KQML serão transmitidas de um agente

---

<sup>7</sup> Ferramenta de desenvolvimento de aplicativos de 4ª geração.

*unidade móvel* a um agente *central de regulação*, apresentando os padrões necessários para que esta comunicação possa acontecer.

A estrutura do *Web Service* possibilitou a comunicação dos agentes do atendimento pré-hospitalar independente da plataforma que o mesmo tenha sido desenvolvido. Esta premissa é importante quando não se conhece as estruturas das organizações que possam vir a fazer parte de um atendimento.

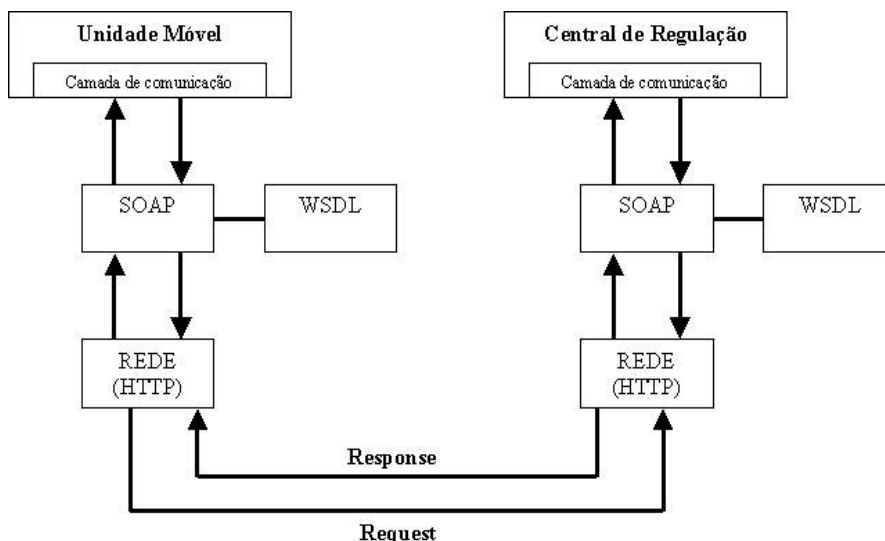


Figura 44: Camadas envolvidas na troca de mensagens via *web service*

Para a utilização da camada de comunicação, descrita na Figura 44, foi desenvolvido uma interface com o intuito de viabilizar a comunicação entre as organizações. A interface apresentada na Figura 45 ilustra a camada de comunicação de uma *central de regulação*. Esta camada estará trocando informações do sistema local que atende à *central de regulação* com os demais sistemas das organizações ativas no momento.

A Figura 45 apresenta uma simulação de comunicação de uma *central de regulação* com uma *unidade móvel* que está realizando um atendimento pré-hospitalar. A performativa “ask-about” retorna todas as informações que o agente possui sobre determinada operação, nesse caso, o atendimento pré-hospitalar. No campo “Language” deve-se informar a linguagem em que a rede realiza suas comunicações, esta linguagem é pré-estabelecida pela rede, ou seja, os agentes que deverão trocar suas mensagens baseadas nas definições desta linguagem; no campo “Ontology”, deve ser inserida a ontologia aceita pela rede. Esta ontologia traz o conjunto de termos que farão a comunicação. É baseado na ontologia que um agente entende a solicitação dos demais agentes da rede; no campo “Content”, deve ser informado sobre qual conteúdo deseja-se saber (o conteúdo pode variar de acordo com a performativa escolhida); no campo “Receiver”, deve ser informado o endereço do agente que

se deseja comunicar; no campo “Sender”, para qual agente deve ser respondida a solicitação, que deve ser identificada no campo “In Reply To”; o campo “Comment” é utilizado em algumas *performativas* para o complemento das informações que devem ser comunicadas. Nesse protótipo ainda existe a possibilidade de informar qual a *performativa* que se deseja utilizar. As repostas dos agentes são ilustradas no quadro situado na parte inferior da Figura 45.

Figura 45: Interface do protótipo de comunicação da *central de regulação*

Um agente não precisa necessariamente conhecer apenas uma linguagem e uma ontologia, ao contrário, quanto maior o número de linguagens e ontologias que o agente conheça, mais amplo é seu universo de relacionamentos. Um mesmo agente pode estar inserido em mais de uma rede e trocar informações com os agentes destas redes. O cuidado que deve-se ter nestes casos é que o agente deve conhecer a linguagem e a ontologia que está se trocando mensagens dentro do contexto que ele está inserido no momento.

Todos os parâmetros apresentados, e que foram preenchidos, como mostra a Figura 45, fazem parte dos parâmetros definidos pela linguagem KQML. Desta forma, o protótipo apresentou uma forma de comunicação e a utilização das *performativas* para troca de informações.

O desenvolvimento da arquitetura deixou livre a escolha da ontologia e da linguagem a serem utilizadas. Atualmente, o protótipo desenvolvido aceita apenas a pseudo-linguagem

Natural com uma ontologia denominada pré-atendimento. Vale a ressalva de que a linguagem e a ontologia citadas foram utilizadas apenas para o contexto do protótipo.

A interface apresentada é preenchida desta forma apenas para facilitar o teste das camadas de comunicação e da integração com os demais agentes.

Foi construída também neste protótipo uma interface para configuração do agente, já que não há a presença de um sistema de informação que gere conteúdo para a rede. Estas informações foram inseridas para que os testes pudessem ser concluídos. Através das *performativas* que foram desenvolvidas, é possível também inserir conhecimentos em outros agentes participantes da rede. A *performativa Insert* possui esta função. A Figura 46 ilustra a interface de configuração dos agentes desenvolvidos para o protótipo.

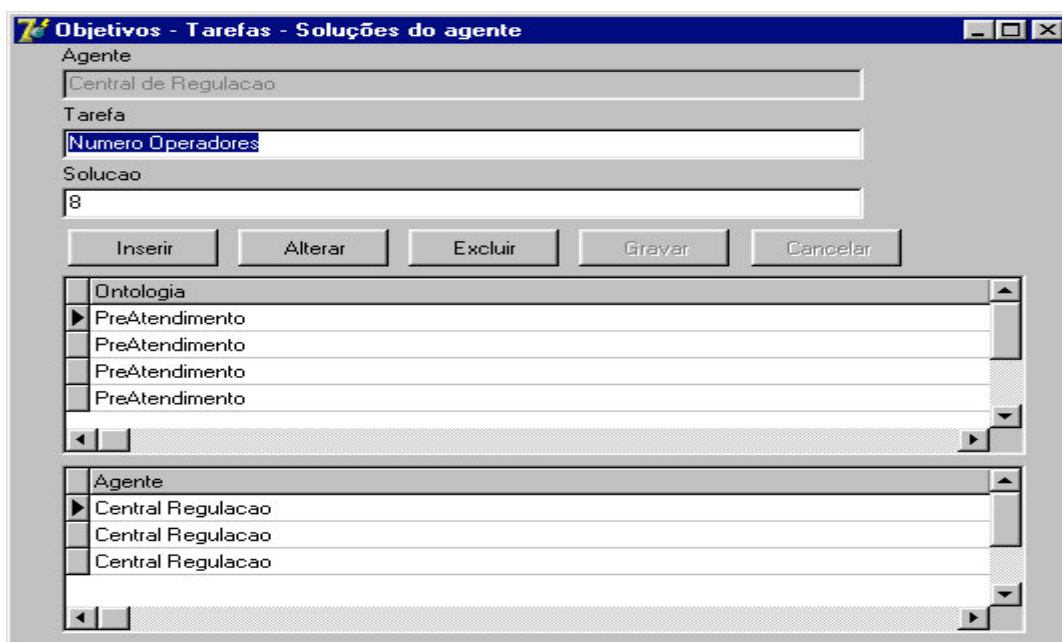
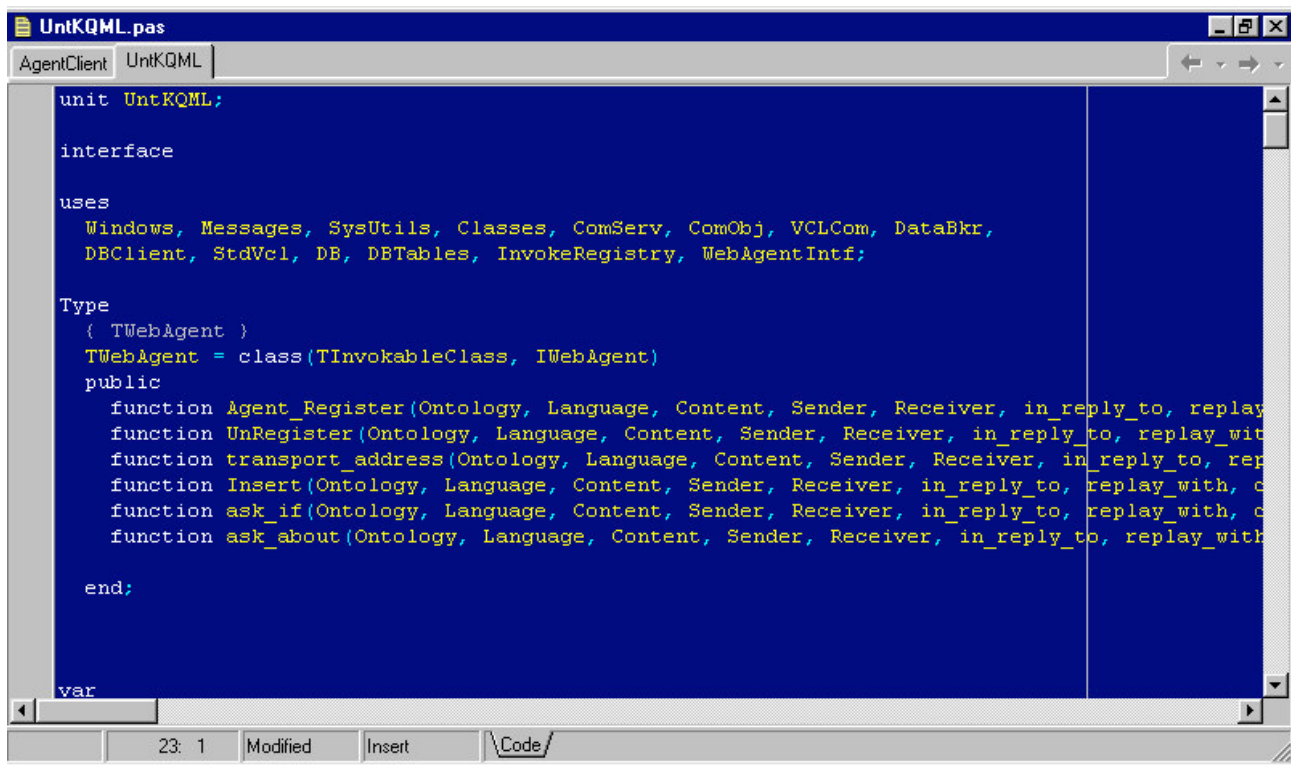


Figura 46: Interface do protótipo de configuração de um agente

Na Figura 47 estão às declarações das *performativas* desenvolvidas. Como se pode observar também a camada de comunicação é apenas uma classe que pode ser acoplada nas aplicações que desejarem fazer uso da arquitetura construída. A partir da inserção desta classe em um projeto de um novo agente, o mesmo terá a capacidade de realizar a comunicação com os demais agentes da rede, bastando para isso o preenchimento dos campos descritos na interface da Figura 45. Dessa forma, a comunicação entre agentes passou a ser uma chamada de função que é capaz de realizar uma comunicação e obter as informações necessárias e possíveis com a utilização do pacote definido pelas *performativas* KQML. As funções disponibilizadas são as mesmas para todos os agentes, já que a camada de comunicação é uma

para todas as organizações. As mudanças ficam a cargo da interpretação que cada agente realiza da camada de comunicação e da inteligência que cada um utiliza para a busca das informações solicitadas pela rede. A troca das informações através das performativa do protocolo KQML, da linguagem e ontologia aceitas pela rede permitem que qualquer informação seja retirada de um agente, bastando, para tanto, que o agente detenha essa informação.



```

unit UntKQML;

interface

uses
  Windows, Messages, SysUtils, Classes, ComServ, ComObj, VCLCom, DataBkr,
  DBClient, StdVcl, DB, DBTables, InvokeRegistry, WebAgentIntf;

Type
{ TWebAgent }
TWebAgent = class(TInvokableClass, IWebAgent)
public
  function Agent_Register(Ontology, Language, Content, Sender, Receiver, in_reply_to, replay_with,
  function UnRegister(Ontology, Language, Content, Sender, Receiver, in_reply_to, replay_with,
  function transport_address(Ontology, Language, Content, Sender, Receiver, in_reply_to, replay_with,
  function Insert(Ontology, Language, Content, Sender, Receiver, in_reply_to, replay_with,
  function ask_if(Ontology, Language, Content, Sender, Receiver, in_reply_to, replay_with,
  function ask_about(Ontology, Language, Content, Sender, Receiver, in_reply_to, replay_with,
end;

var

```

Figura 47: Implementação da classe de comunicação.

Para o acompanhamento do tráfego das mensagens trocadas dentro da rede, foi desenvolvido um console capaz de mapear e mostrar de forma gráfica as trocas de mensagens entre os agentes da rede. Esse mapeamento tornou-se possível devido a um controle que cada agente grava no momento em que está enviando uma mensagem. O controle possui a informação do agente que enviou a mensagem, o agente que recebeu e o conteúdo dessa mensagem. A Figura 48 ilustra a interface de visualização das mensagens trocadas. No topo da tela encontram-se todos os agentes registrados na rede no momento com a sua respectiva situação, informando quanto ao seu estado (0 - Ativo ou 1 - Inativo). O controle da atividade do agente é realizado através do envio de uma mensagem, que, ao receber resposta positiva, confirma a atividade de um agente.

O console, como é denominada esta interface, faz uso das informações armazenadas pelas próprias *performativas* KQML. O usuário ainda possui a opção de atualizar as informações apresentadas no console. A cada vez que o controle atualizar é pressionado uma nova posição a respeito das mensagens trocadas é mostrada através da movimentação dos “envelopes”, indicando o caminho que uma mensagem realizou, desde sua origem até seu destino.

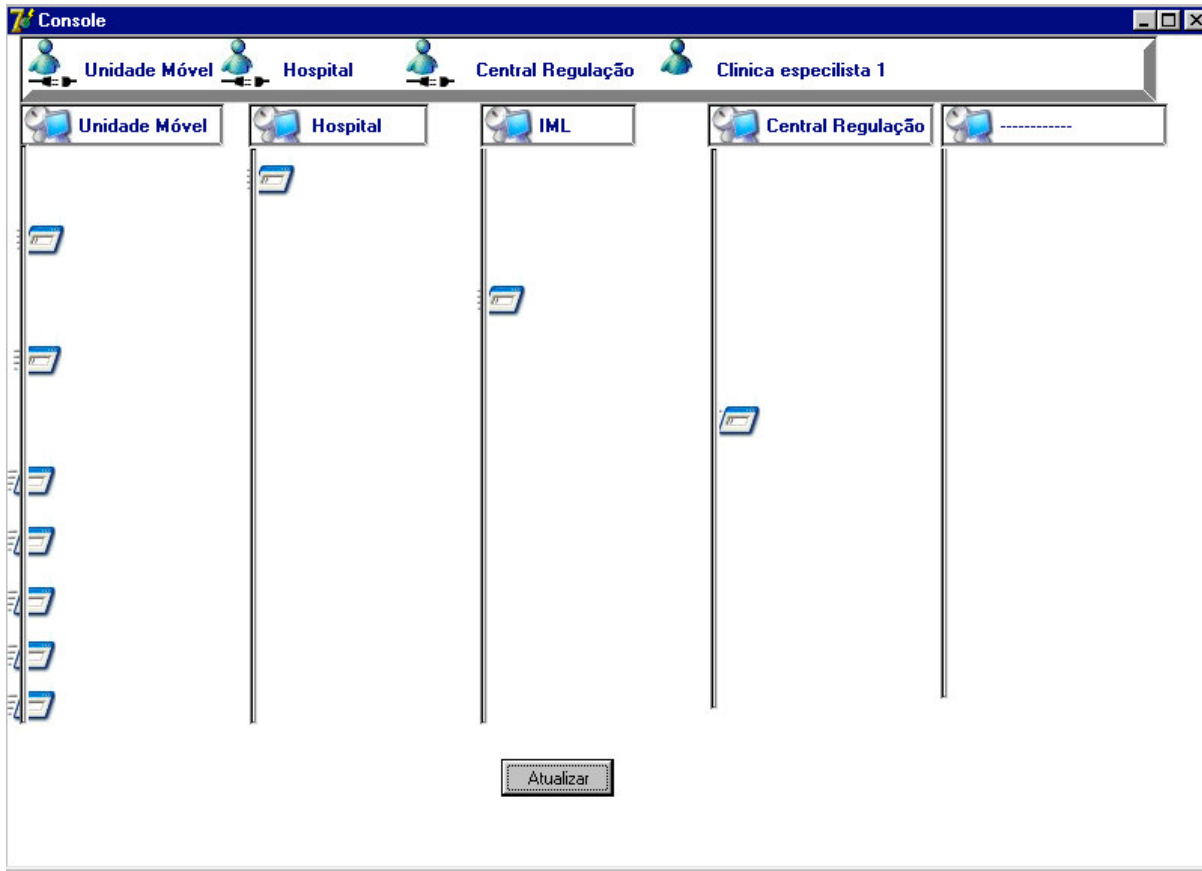


Figura 48: Console de visualização das mensagens trocadas pela rede de atendimento pré-hospitalar

A consolidação do protótipo possibilitou dizer que a arquitetura pode ser testada em situações onde sejam caracterizadas uma estrutura para atender a um único chamado, envolvendo apenas uma *unidade móvel*, uma *central de regulação* e um *hospital* ou clínica, assim como, ter capacidade de atender, por exemplo, a uma catástrofe, onde estariam presentes várias unidades móveis, integradas a muitas centrais de regulação, utilizando diversas clínicas e hospitais, além dos demais órgãos que prestam apoio nas ocorrências.

Foram realizados dois tipos de testes na arquitetura utilizando os protótipos de dois agentes. O primeiro teste foi realizado em uma rede fechada, ou seja, sem a utilização da internet. Este teste foi apenas para medir a viabilidade da troca de mensagens e validar se a forma de comunicação atende a necessidade. Este teste ocorreu em um laboratório com 2



máquinas onde cada uma representava um agente, neste caso a *central de regulação* e a *unidade móvel*. Além de verificar a viabilidade da aplicação do KQML para troca de mensagens foi avaliado o tempo e a eficiência das respostas dadas pelos agentes. Por se tratar de uma rede fechada a comunicação aconteceu a 100 Mbits. Nestas condições os tempos de resposta foram praticamente imediatos e sempre atendendo a expectativa de conteúdo. Obviamente que o volume de informações que cada agente possuía nos testes eram reduzidos, dentro da realidade do atendimento pré-hospitalar pode haver ainda uma certa demora nas respostas decorrente do processamento que a informação irá necessitar para ser gerada. Uma vantagem apresentada pela arquitetura é que o processamento de extração de uma informação será sempre dentro de um agente, sem depender, a maioria das vezes, de acessos externos para recuperar informações. Pode-se fazer uma analogia com os procedimentos de *trigger* e *store procedure* (funções processadas dentro de um servidor de banco de dados, e não em um cliente fora do servidor) de um banco de dados em aplicações cliente/servidor típicas, assim como o Gerenciador de Emergências Médicas [Machado & Grein, 1999], onde o banco de dados, normalmente, encontra-se em um servidor e o processamento encontra-se em um cliente. A cada solicitação de informação por um cliente, o servidor deve ser acessado em busca dos dados que gerarão no cliente as informações desejadas. Esta busca de dados demanda tempo e recurso, e só após estes dados estarem na máquina cliente, é que serão processados e apresentados ao usuário. Uma solução utilizada neste tipo de aplicação, em casos específicos que dependam de performance, é colocar o processamento em procedimentos dentro do banco de dados, como mencionado. O cliente, neste caso, receberá a informação pronta para ser apresentada ao usuário. Esta forma de implementação apresentou resultados melhores durante os testes realizados no GEM. De forma análoga a um procedimento que apresenta a informação pronta a seu cliente, um agente também apresenta a resposta de uma solicitação prontamente. Isto explica o tempo imediato nas respostas dadas durante esta primeira bateria de testes.

O segundo teste realizado visou validar a comunicação a distância, o envolvimento do ambiente *web* é inevitável neste tipo de configuração. Desta forma, foi disponibilizado em uma máquina um protótipo de agente que era acessado via protocolo HTTP por qualquer outra máquina conectada a internet. O teste de comunicação foi realizado em dois lugares físicos distintos que apresentavam configurações de acessos deferentes. O primeiro deles, realizado dentro de uma LAN, obteve tempo de resposta próximo ao imediato. O destaque do teste realizado a distância ficou no tempo de resposta apresentado pelo acesso realizado através de ADSL de 256 Kbps, neste caso, os tempos de respostas foram pouco superiores aos

demais testes realizados, ainda dentro do satisfatório. A análise realizada resulta em afirmar que o conteúdo das mensagens trocadas é o responsável pelos bons tempos de respostas obtidos. Em todos os casos eram comunicados apenas caracteres e isto possibilitou a boa performance apresentada.

É importante lembrar que nem sempre as comunicações acontecerão nestas condições, ou seja, em casos específicos serão trafegadas informações que diferem de uma seqüência de caracteres, como por exemplo, imagem e som. Nestes casos, a comunicação poderá apresentar uma perda de performance. Porém, os chamados de atendimento pré-hospitalar podem ser atendidos sem necessariamente a utilização de imagens e sons, e quando se fizerem imprescindíveis estes benefícios podem ainda utilizar formas de comunicação alternativa.

Nos projetos realizados na PUCPR desde 1999, foram utilizadas plataformas cliente/servidor e web, sempre visando um sistema de atendimento pré-hospitalar integrado, porém, fechado. Neste tipo de desenvolvimento é previsto apenas uma interface de integração com os demais sistemas que podem vir a surgir. Isto sempre ocasionará um re-desenvolvimento em pelo menos uma das pontas, isto quando não se fizer necessário nas duas. Uma arquitetura aberta é uma evolução dos modelos já desenvolvidos. Esta arquitetura foi idealizada após uma pesquisa que apresentava uma quantidade indefinida de órgãos necessários para o acontecimento de um atendimento pré-hospitalar. Em se tendo, no momento do desenvolvimento de um projeto, uma estrutura de entidades participativas indefinida, a forma mais adequada de projeto é a que permita que estas entidades, não previstas a princípio, possam ser incorporadas ao sistema sem causar impacto. A arquitetura de um sistema de agentes surgiu para solucionar um problema tecnológico e negocial. Em se tratando de negócio o atendimento pré-hospitalar é uma área que passa por mudanças legais, necessita de novas entidades muitas vezes imprevisíveis, necessita de um banco de dados que possa ser facilmente acessado e disponibilizado para pesquisas e necessita considerar a existência de um conjunto de sistemas complexos em suas entidades (*hospital, unidade móvel, etc*) fisicamente distantes. Estas variáveis levam a uma tecnológica diferente da aplicada nos projetos dentro da PUCPR [Machado & Grein, 1999; Dias et al, 2002].

É importante ressaltar que o controle do atendimento pré-hospitalar pode fornecer dados para área de saúde e para a gestão pública, dados estes que transformados em informações, podem trazer benefícios a sociedade que necessita deste tipo de atendimento, além de criar uma rede de relacionamentos que pode ser utilizada para solucionar problemas que transcendam as barreiras do atendimento pré-hospitalar. Para viabilizar estas reivindicações negociais o apelo tecnológico teve que ser revisto, e os estudos apontaram à troca de

mensagens, ao conhecimento distribuído de entidades complexas, a aceitar alterações de ambientes e a suportar inclusão de novas funcionalidades não previstas. Estas características levaram os estudos ao SAAP (Sistema de Agentes do Atendimento Pré-hospitalar).

As mensagens trocadas em um atendimento pré-hospitalar podem variar muito. Em um determinado momento, por exemplo, podem existir inúmeros agentes auxiliando, em outro apenas o mínimo necessário (*unidade móvel, central de regulação e hospital ou clínica*). Desta forma, um relato de um atendimento pode ser utilizado para acompanhar a evolução das trocas de mensagens e da integração dos agentes envolvidos. Este relato visa fornecer uma linha de raciocínio que possa vir a facilitar o entendimento da proposta apresentada nesta dissertação.

A seguir, segue um relato de uma ocorrência para que seja validada a modelagem da aplicação e os demais produtos desenvolvidos. A contextualização de uma situação pode auxiliar no desenvolvimento de um caso para mostrar as necessidades exploradas neste trabalho. No decorrer do relato serão apresentadas algumas das mensagens que podem ser trocadas entre os agentes. As mensagens possuem um padrão definido, onde os parâmetros estão dispostos da seguinte forma: `tipo_mensagem("Emissor", "Receptor", "Linguagem", "Ontologia", "Conteúdo", "Identificação da mensagem", "Parâmetro de resposta esperada", "Comentário")`.

## 6.1 Relato de Acidente de Trânsito

“Um acidente de trânsito acontece, um automóvel em alta velocidade choca-se contra uma árvore. No carro, um condutor e três passageiros, todos ficam presos nas ferragens.

O acidente ocorre por volta das 4 horas da madrugada de sábado. Imediatamente, pessoas da localidade acordadas pelo barulho acionam o serviço de atendimento pré-hospitalar. As informações são captadas pelo atendente responsável pelo atendimento. Neste momento, o que se sabe é que há quatro pessoas, todas presas nas ferragens, o endereço e as características da ocorrência (veículo de passeio conduzido em alta velocidade choca-se contra uma árvore). Após esta prévia coleta de informações realizadas pelo atendente, o médico é acionado, recebendo a ligação. As informações, neste caso, são outras.”

```
ask_if("Central de Regulacao", "Unidade Movel", "Pre-Atendimento",
"Natural", "Solicitar_Contrato","","",""); // Central contrata unidade
Tell("Central de Regulacao", "Unidade Movel", "Pre-Atendimento", "Natural",
"Confirmar_Solicitacao_contrato","","",""); // Central confirma contrato
com unidade
```

```
Tell("Central de Regulacao", "Unidade Move1", "Pre-Atendimento", "Natural",
"Primeiras_informações_ocorrendia", "" ,"" , ""); // Central repassa endereço
e informações da ocorrência para unidade
```

“Paralelo ao atendimento do médico, a *unidade móvel* já está se conduzindo ao local da ocorrência. O médico capta novas informações com relação às vítimas.”

```
Tell("Central de Regulacao", "Unidade Move1", "Pre-Atendimento", "Natural",
"informações_Medicadas_ocorrendia", "" ,"" , ""); // Central repassa
informações médicas da ocorrência para unidade
```

“Estas informações são capazes de conduzir à próxima ação a ser tomada. Informações como escoriações externas, possibilidade de óbito no local, grande número de vítimas presas nas ferragens, são definitivas na decisão de encaminhamento do médico até o local da ocorrência. Tais informações são coletadas e o médico se dirige para o atendimento.”

“No momento da liberação da *unidade móvel*, que ocorreu antes da decisão da presença médica, já se sabia que seria necessária a utilização de equipamentos especiais para recortar as ferragens do veículo. Os paramédicos que chegaram primeiro ao local do acidente realizaram o trabalho de remoção das vítimas de dentro do veículo, sempre apoiados pelo atendimento médico da *central de regulação* responsável pelo atendimento.”

```
Tell("Unidade Move1", "Central de Regulacao", "Pre-Atendimento", "Natural",
"informações_ocorrendia", "" ,"" , ""); // Unidade repassa informações gerais
(vítimas e ocorrência) da ocorrência
ask_about("Unidade Move1", "Central de Regulacao", "Pre-Atendimento",
"Natural", "Auxilio_Medico", "" ,"" , ""); // Unidade solicita procedimento
médico a central
```

“Quando a presença do médico de atendimento foi requerida, ele estava presente para estabilizar a situação dos feridos. Durante o atendimento, existe a necessidade de consultar especialistas médicos. A solicitação da *unidade móvel* foi prontamente atendida por uma *clínica especializada*.”

```
ask_if("Unidade Move1", "Clinica Especialista", "Pre-Atendimento",
"Natural", "Solicitar_Contrato", "" ,"" , ""); // Unidade solicita contrato
com clínica especialista
Tell("Unidade Move1", "Clinica Especialista", "Pre-Atendimento", "Natural",
"Confirmar_Solciitacao_contrato", "" ,"" , ""); // Unidade confirma contrato
com clínica especialista
```

```

Tell("Unidade Moveel", "Clinica Especialista", "Pre-Atendimento", "Natural",
"Quadro_Geral_vitimas", "" ,"" ,"" ); // Unidade informa quadro geral de
vítimas
ask_About("Unidade Moveel", "Clinica Especialista", "Pre-Atendimento",
"Natural", "Fraturas_multiplas", "" ,"" ,"" ); // Unidade solicita
informações sobre poli-fraturados
Tell("Clínica Especialista", "Unidade Móvel", "Pre-Atendimento", "Natural",
"informações_poli_fratuados", "" ,"" ,"" ); // Clínica especialista
responde Unidade Móvel
Tell("Unidade Moveel", "Central de Regulacao", "Pre-Atendimento", "Natural",
"informações_ocorrencia", "" ,"" ,"" ); // Unidade repassa informações
complementares s/ atendimento para central

```

“O atendimento recebeu o auxílio dos oficiais que respondem ao Departamento de Trânsito.”

```

ask_if("Central regulacao", "BPTRAN", "Pre-Atendimento", "Natural",
"Solicitar_Contrato", "" ,"" ,"" ); // Central solicita contrato com BPTRAN
Tell("Central regulacao", "BPTRAN", "Pre-Atendimento", "Natural",
"Confirmar_Solciitacao_contrato", "" ,"" ,"" ); // Central confirma contrato
com BPTRAN
Tell("Central regulacao", "BPTRAN", "Pre-Atendimento", "Natural",
"Informações_ocorrencia", "" ,"" ,"" ); // Central informa BPTRAN sobre
endereço e condições do acidente

```

“Duas unidades móveis do tipo UTI foram utilizadas para a remoção dos feridos, além da *unidade móvel* médica. Os três passageiros do veículo foram encaminhados ao hospital e receberam os primeiros atendimentos logo após a colisão, o que evitou o óbito dos três.”

```

ask_if("Unidade Moveel", "Hospital", "Pre-Atendimento", "Natural",
"Solicitar_Contrato", "" ,"" ,"" ); // Unidade verifica contrato com
hospital
Tell("Unidade Moveel", "Hospital", "Pre-Atendimento", "Natural",
"Confirmar_Solciitacao_contrato", "" ,"" ,"" ); // unidade confirma contrato
com hospital
Tell("Unidade Moveel", "Hospital", "Pre-Atendimento", "Natural",
"Informacoes_Vitimas", "" ,"" ,"" ); // Unidade repassa informações das
vítimas ao hospital
Ask_if("Unidade Moveel", "Hospital", "Pre-Atendimento", "Natural",
"Encaminhamento", "" ,"" ,"" ); // Unidade verifica se hospital pode atender
vítimas

```

“Já o condutor do veículo entrou em óbito no local do acidente antes mesmo da equipe de socorro prestar atendimento. Devido a tal fato, este foi atendido pelos profissionais do Instituto Médico Legal.”

```
ask_if("Central regulacao", "IML", "Pre-Atendimento", "Natural",
"Solicitar_Contrato", "", "", ""); // Central solicita contrato com IML
Tell("Central regulacao", "IML", "Pre-Atendimento", "Natural",
"Confirmar_Solciitacao_contrato", "", "", ""); // Central confirma contrato
com IML
Tell("Central regulacao", "IML", "Pre-Atendimento", "Natural",
"Informações_ocorrencia", "", "", ""); // Central informa IML sobre
endereço e condições do acidente
```

## 6.2 Comentários

A ação dos agentes dentro de um atendimento pré-hospitalar é variável, ou seja, em cada chamada de atendimento pré-hospitalar a atuação pode existir ou não, assim como a própria tarefa que o agente executa pode ser diferenciada. No caso descrito acima, por exemplo, o agente *clínica especializada* prestou auxílio ao médico que se deslocara ao local da ocorrência com relação a traumas generalizados, o que não impede que, em uma próxima ocorrência, o mesmo agente *clínica especializada* venha a fornecer auxílio a um paramédico sobre um caso clínico.

O atendimento pré-hospitalar exige em seus requisitos mais básicos um grande número de agentes distintos com atitudes diferentes e que possam agir correspondendo às necessidades de cada atendimento. Este modelo é fortemente inspirado nas relações humanas que atualmente atendem às necessidades das ocorrências.

Com essa arquitetura, almeja-se conseguir coordenar os diversos órgãos quando estes se fizerem necessários dentro de uma ocorrência, unir uma grande massa de dados e informações capaz de auxiliar nos mais variados métodos de atendimento pré-hospitalar de urgência/emergência, fornecer meios de criação de melhores técnicas de atendimento pré-hospitalar e, principalmente, fornecer um melhor auxílio aos usuários dos serviços de atendimento pré-hospitalar de uma determinada região.

## Capítulo 7 - Conclusão

As chamadas causas externas, onde estão incluídos os acidentes de trânsito, as violências, as quedas, os acidentes domiciliares e de trabalho, constituem hoje a principal causa de morte na população de 05 a 49 anos nos centros urbanos de médio e grande porte do país. No Paraná, os acidentes de trânsito são responsáveis por 43,6% dos óbitos por causas externas, seguidos dos homicídios com 19,7%. Por outro lado, mesmo não ocorrendo mortes, uma proporção relevante de pessoas são feridas, diminuindo ou perdendo a sua capacidade produtiva [SIATE, 2002].

A modelagem de um sistema de agentes serviu para apresentar uma metodologia em prol de um contexto de relevância para a sociedade. A possibilidade de utilizar uma estrutura de agentes modelados pode trazer benefícios ainda não computados no contexto do atendimento pré-hospitalar, tais como: a) o atendimento pré-hospitalar é um contexto complexo e fisicamente distribuído; b) o atendimento pré-hospitalar possui um conjunto de entidades indefinido e complexo para ser tratado como uma solução única, integrada e fechada. O SAAP contempla soluções baseadas em abordagens locais onde os problemas são freqüentemente resolvidos mais rapidamente; c) o SAAP pode se adaptar às modificações de estrutura e de ambiente através de uma arquitetura apta a levar em conta as evoluções e adaptações necessárias para o funcionamento de um sistema; d) a adição de novas funcionalidades, modificações sobre sua utilização, integração de um novo software podem acontecer a cada novo atendimento pré-hospitalar. Além de possibilitar novas integrações a arquitetura desenvolvida permite a adaptação de softwares que viabilizem *feedback* para gestores da administração pública, inclusão de estudos evolutivos para auxílio a tomada de decisões nos atendimentos, entre outras soluções que não foram previstas e nem imaginadas para o contexto atual.

A solução utilizada no SAAP foi construída com a aplicação de uma linguagem de modelagem de agentes. A AUML proporciona um conjunto de modelos que permitem inferir sobre um conjunto de agentes capazes de trocar intenções para a solução de um determinado problema. O SAAP foi modelado com base em um conjunto mínimo de agentes participantes nos atendimentos pré-hospitalares, porém esta modelagem não especifica um conjunto máximo de agentes, já que se trata de uma arquitetura aberta. Os agentes que podem vir a

compor o atendimento pré-hospitalar estarão trocando informações entre si com um objetivo definido. Este objetivo é o papel que cada agente deve efetuar dentro da arquitetura proposta.

Atendendo aos requisitos de comunicação qualquer novo software ou agente que seja desenvolvido é capaz de trocar informações com os demais agentes constantes na rede. O desenvolvimento de protótipos de agente possibilitou examinar a utilização de troca de mensagens que devem existir entre as instituições que fazem parte do atendimento pré-hospitalar.

O desenvolvimento de sistemas multi-agente pode trazer ainda vantagens que vão além da solução de um negócio. A construção de unidades capazes de armazenar dado é a melhor solução para resolução de problemas futuros. Com esta estrutura alimentada, torna-se possível à implementação de agentes específicos para pesquisas dentro do contexto do sistema. Esta aplicabilidade de sistemas de agentes exemplifica a característica de agregação de novas funcionalidades a um sistema já idealizado, construído e em produção, sem a necessidade de se realizar qualquer alteração nos aplicativos dos agentes já envolvidos.

O contexto do atendimento pré-hospitalar apresenta características que o enquadram como um sistema distribuído. Sendo assim, o SAAP é apresentado seguindo as seguintes diretrizes: a) problemas complexos; b) entidades fisicamente distribuídas; c) envolvimento de aplicações heterogêneas; d) ambientes heterogêneos, e; e) grande potencialidade de evoluções e agregações de novas entidades. Para todo contexto há uma necessidade, um negócio, um motivo. As regras de acesso são definidas, as formas de comunicações também, as regras negociais serão contempladas para cada agente envolvido. Ou seja, para se controlar qualquer ambiente, deve-se apenas preocupar-se com a sua organização.

Com um ambiente multi-agente modelado para atendimento pré-hospitalar, o protótipo de um sistema aberto desenvolvido, uma linguagem e um meio de comunicação testados e um conjunto de agentes definidos para atuar em atendimentos, o SAAP apresenta uma proposta capaz de viabilizar um contexto distribuído. O SAAP pode ainda aproveitar estudos e implementações já concluídos e implantados, integrando-os e, da mesma forma, permitir que novos estudos e implementações também venham a participar e contribuir para o atendimento pré-hospitalar.

Como afirma Ferber [Ferber, 1995] “a engenharia de software caminha para uma concepção em termos de unidades autônomas inter-atuantes” e a aplicação deste modelo no atendimento pré-hospitalar possibilitou a idealização de uma solução capaz de organizar e integrar de forma transparente as diversas e imprevisíveis entidades que podem prestar um atendimento pré-hospitalar à sociedade.



## **7.1 Trabalhos Futuros**

Devido à facilidade de se adicionar soluções a uma arquitetura de agentes, esta pode ser utilizada em outros ambientes que não só o atendimento pré-hospitalar. Uma arquitetura aberta como a proposta pelo SAAP ainda cria possibilidades de novas implementações que venham a melhorar o sistema a que a arquitetura é aplicada. A seguir são apresentados estudos que compartilham as diretrizes e definições do SAAP.

### **7.1.1 Web Service multi-Agente para integração de sistemas legados no SUS**

Desde o momento que foi instituído o SUS em 1988, criou-se um sistema com conhecimento distribuído em diversas instâncias, a saber: municipais, estaduais e federais. Neste contexto, a NOB 01/96 (Normas de Operacionais Básicas) amparada pela Constituição Brasileira de 1988 Lei 8.080/90, preocupa-se expressamente em estabelecer diretrizes para gestão descentralizada do SUS [Brasil, 1996].

No entanto, mesmo tratando-se de um sistema que busca a unificação da saúde pública, os gestores de áreas específicas do SUS, instituíram sistemas de informações para atender objetivos específicos, suprimindo as necessidades pontuais de determinadas competências, sem considerar a organização unificada do SUS, ou seja, mesmo o SUS sendo descentralizado devido a sua amplitude, os sistemas de informação devem buscar a integração em todos os níveis de competências.

O resultado desta fragmentação (ou falta de integração) é facilmente percebida nos sistemas de informação legados desenvolvidos pelos gestores do SUS, com objetivos específicos como [Datusus, 2004]: mortalidade, morbidade, natalidade, hospitalar, ambulatorial, imunização, laboratorial, farmacêutica, etc. Observa-se nestes sistemas que as entidades envolvidas no processo (estabelecimentos de saúde, usuários da saúde, profissionais da saúde, ações de saúde, etc) deveriam ser correlacionados, porém, os sistemas produzem informações sem manter a relação entre estes elementos ou entidades, desconsiderando aspectos de integridade e consistência dos dados muito menos dos processos de cada segmento da saúde pública.

Almejando a integração do conhecimento gerado nos sistemas de informações do SUS, será utilizada a mesma arquitetura prevista no SAAP, com o intuito de encapsular os sistemas legados existentes dentro de uma estrutura de sistemas multi-agente, onde cada instância do sistema passará a ser um agente, promovendo a integração e colaboração entre os sistemas. As

definições de linguagem de comunicação e tecnologia de troca de mensagens serão compartilharão das definições apresentadas no SAAP.

### **7.1.2 Utilização de *guidelines* para auxílio no atendimento pré-hospitalar**

A proposta deste trabalho consiste em conceber um modelo para permitir a incorporação/integração dos recursos institucionais (humanos e matérias) no acompanhamento de *guidelines* com a flexibilidade do conhecimento médico, possibilitando o incremento de novas informações para complementar o que está definido na *guideline*. Poderá permitir também, o acompanhamento do atendimento do paciente através das informações fornecidas pelo profissional da organização de acordo com a evolução do quadro clínico do paciente.

A aplicação de *guidelines* dentro do atendimento pré-hospitalar pode seguir duas linhas de implementação: a) utilização de *guidelines* compondo a construção de um agente específico da rede de atendimento pré-hospitalar, ou seja, possuir um agente denominado *guideline* onde os demais agentes buscariam soluções para apoiar as decisões que deverão tomar dentro de um atendimento. b) possibilidade de implementação de *guidelines* dentro dos agentes, ou seja, como cada agente possui uma área de atuação, este *guideline* seria específico para atender as solicitações deste agente. Em ambas as situações os *guidelines* teriam o papel de auxiliar o atendimento compartilhando as informações e as soluções que o competem com os demais agentes da rede.

A utilização de *guideline* deixa a possibilidade de construir-se aplicações, ou agentes, com a possibilidade de permitir que novos conhecimentos possam ser adicionados garantindo a evolução da rede do atendimento pré-hospitalar, ganhando com isto uma capacidade maior de tomada de decisões.

## Referências Bibliográficas

[Abriat, 1991] Abriat P.; Conception et Realisation d'un Système Multi-Agents de Robotique Permettant de Récupérer les Erreurs dans les Cellules Flexibles, Thèse de Doctorat, Division Informatique, U.R.A. C.N.R.S. n° 817 HEUDIASYC, Université de Technologie de Compiègne, Compiègne, 1991.

[Amundsen & Litwin, 2002] AMUNDSEN, Michael e LITWIN, Paul. ASP.NET para desenvolvedores de Web Sites 319- 60 p. Rio de Janeiro: Editora Ciência Moderna Ltda., 2002.

[Auer, 1995] AUER, Karl. Agentes. URL: <http://www.pcug.org.au/~kauer/project/main.htm>, acessado em 10/2003, 1995.

[Baeijs, 1998] Baeijs, C.; Fonctionnalite Emergente dans une Societe d'Agents Autonomes, Etude des Aspects Organisationnels dans les Systemes Multi-Agents Reactifs. Tese doutoradp, 1998.

[Bauer, 1999] Bauer, B., Extending UML for the Specification of Interaction Protocols, submitted for the 6th Call for Proposal of FIPA, 1999.

[Beer, 1992] Beer R.D.; A Dynamical Systems Perspective on Autonomous Agents. In: Special Issue of the AI Journal on Computational Theories of Interaction and Agency, 1992.

[Demazeau, 1990] Demazeau Y., Müller J-P.; Decentralized Artificial Intelligence. In Demazeau Y., Müller J-P., Decentralized AI, Elsevier Science Publishers, 1990.

[Belgrave, 1995] Belgrave, Marc, The unified agent architecture. [URL: [http://www.ee.mcgill.ca/~belmarc/uaa\\_paper.html](http://www.ee.mcgill.ca/~belmarc/uaa_paper.html)] acessado em 02/2004, 1995.

[Bond & Gasser, 1988] - Bond H. and Gasser L., (1988), What is DAI ?, In : Reading in Distributed Artificial Intelligence, Morgan Kaufman Publishers, Los Altos, CA, 1988.

[Bond & Gasser, 1988] Bond H., Gasser L., (), What is DAI?. In Reading in Distributed Artificial Intellingence. Morgan Kaufman Publishers, 1988.

[Bond & Gasser, 1992] - Bond A. H. and L. Gasser, A Subjet-Indexed Bibliography of Distributed Artificial Intelligence, IEEE Transactions on Systems, Man, And Cybernetics, 22 , n° 6, November/December, pp 1260-1281, 1992.

[Bouron, 1993] Bouron T.; Structures de Communication et d'Organisation pour la Coopération dans un Univers Multi-Agents, LAFORIA 93.04, Février, 1993.

[Brasil, 1996] Normas Operacionais Básicas; Ministério da Saúde, 1996.

[Brasil, 2001] BRASIL, Portaria 814/GM de 01 de junho de 2001. normatização dos Serviços de Atendimento Pré-hospitalar Móvel de Urgências e define os princípios e as diretrizes da Regulação Médica das Urgências. Documento disponibilizado no endereço [http://dtr2001.saude.gov.br/samu/legislacao/index\\_leg.htm](http://dtr2001.saude.gov.br/samu/legislacao/index_leg.htm) e acessado em novembro de 2004.

[Brasil, 2002a] BRASIL, Portaria 2048/GM de 5 de novembro de 2002. Regulamenta o atendimento das urgências e emergências. Capítulo I pag. 77. Documento disponibilizado no endereço [http://dtr2001.saude.gov.br/samu/legislacao/index\\_leg.htm](http://dtr2001.saude.gov.br/samu/legislacao/index_leg.htm) e acessado em novembro de 2004.

[Brasil, 2002b] BRASIL, Portaria 2048/GM de 5 de novembro de 2002. Regulamenta o atendimento das urgências e emergências. Capítulo II e III - . Documento disponível no endereço [http://dtr2001.saude.gov.br/samu/legislacao/index\\_leg.htm](http://dtr2001.saude.gov.br/samu/legislacao/index_leg.htm), 2002

[Brasil, 2003] Portaria N° 1864/GM Em 29 de setembro de 2003. que regulamente o atendimento móvel de urgência e emergência. Documento disponibilizado no endereço [http://dtr2001.saude.gov.br/samu/legislacao/index\\_leg.htm](http://dtr2001.saude.gov.br/samu/legislacao/index_leg.htm) e acessado em dezembro de 2004.

[Brasiura, 2003] BASIURA, Russ et al. Professional ASP.NET Web Service. São Paulo: Pearson Education, 2003.

[Brazier, 1998] Brazier, Frances M.T., Catholijn M. Jonkers, and Jan Treur, ed., Principles of Compositional Multi-Agent System Development Chapman and Hall, 1998.

[Briot & Demazeau, 2002] Briot, J.-P. e Demazeau, Y., editors; Principes et architecture des syst`emes multiagents. Hermes, Paris, 2002.

[Brooks, 1986] Brooks R. A.; A Robust Layered Control System for a Mobile Robot, In : IEEE Journal of Robotics and Automation, 2(1), pp. 14-33, March, 1986.

[Brooks, 1991] Brooks R. A.; Intelligence without representation. Vol.47. Artificial Intelligence, 139-159, 1991.

[Bryson, 1998] Bryson, Joanna, and Brendan McGonigle, "Agent Architecture as Object Oriented Design," *Intelligent Agents IV: Agent Theories, Architectures, and Languages. Proceedings of ATAL'97.*, ed., Springer, Berlin, 1998.

[Burmeister et al, 1993] Burmeister, Birgit, Afsaneh Haddadi, and Kurt Sundermeyer, ed., *Generic, Configurable, Cooperation Protocols for Multi-Agent Systems* Springer, Neuchâtel, Switzerland, 1993.

[Burmeister, 1996] Burmeister, B., ed., *Models and Methodology for Agent-Oriented Analysis and Design* 1996.

[Bussman & Demazeau, 1994] Bussman Demazeau Y.; *An Agent Model Combining Reactive and Cognitive Capabilities*, In : *IEEE International Conference on Intelligent Robots and Systems, IROS*, 1994.

[Canestrado, 2004] Um modelo para alocação, controle e acompanhamento de Diretrizes aplicada a Unidade de Dor Torácica, Pré-dissertação de mestrado apresentada ao Programa de Pós-Graduação em Tecnologia da Saúde da PUCPR, 2004.

[Castelfranchi, 1990] Castelfranchi C.; *A Pont Missed in Multi-Agent, DAI and HCI. Decentralized AI*, Y. Demazeau & J-P. Müller (Eds), Elsevier Science Publisher B.V. (North-Holland), 49-62, 1990.

[Castelfranchi, 1995] CASTELFRANCHI, C. *Guarantees for Autonomy in Cognitive Agent Architecture*. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages* volume 890 , 1995.

[Cheong, 1995] CHEONG, F. *Internet agents*. Indianapolis: New Riders Publishing, 1995.

[Corkill et al, 1986] Corkill D.D., Gallagher K.Q., Murray K.E., *GBB : A Generic Blackboard Development System*, In *AAAI*, 2, pp. 1008-1014, 1986.

[Cutkosky et al, 1993] Cutkosky M.R., Englemore R.S., Fikes R.E., Genesereth M.R., Gruber T.R., Mark W.S., Tenenbaum J.M., Weber J.C.; *PACT : An experiment in Integrating Concurrent Engineering Systems*, *IEEE Computer*, January, pp. 23-37, 1993.

[Datusus, 2004] DataSus; Ministério da Saúde; *Catálogo de Aplicações*, disponível em <http://www.datusus.gov.br/catalogo/catprod.htm>, 2004.

[Davidsson, 1995] DAVIDSSON, Paul. On the Concept of the Concept in the Context of Autonomous Agents. II World Conference on the Fundamentals of Artificial Intelligence. Pg. 85-96, 1995.

[Demazeau & Müller, 1991] Demazeau Y., Müller J.P., From reactive to intentional agents, In: Decentralized Artificial Intelligence, (Y. Demazeau and J.P. Müller, Ed.), Vol. 2, pp. 3-10, Elsevier Science Publishers., North Holland, 1991.

[Demazeau, 1995] DEMAZEAU, Y. From interactions to collective behaviour in Agent-Based System. Saint Malo: European Conference on Cognitive Science, 1995.

[Depke et al, 2000] Depke, Ralph, Reiko Heckel, Jochen Malte Küster, "Requirement Specification and Design of Agent-Based Systems with Graph Transformation, Roles, and UML," in this volume, 2000.

[Dias et al, 2002] Dias, João da Silva; Satasiu Raquel K; Malucelli, Andréia. Sistema de Informação e Comunicação para Atendimento Pré-hospitalar. Documento da web [<http://www.objetos6000.com.br>], 2002.

[Durfee et al, 1994] Durfee, E.H.; Rosenschein, J. S., Distributed Problem Solving and Multi-Agent Systems: Comparisons and Examples. Proceeding of The International Workshop on Distributed Artificial Intelligence, Julho 1994.

[Eudemia, 2001] Eudemia, X. M.. Contract net. <http://www.ime.usp.br/eudemia/jaia/texto/mode25.html>, 2001

[Ferber & Drogoul, 1992] Ferber J., Drogoul A.,; Using Reactive Multi-Agent Systems in Simulation and Problem Solving, In : Avouris N.M., Gasser L., eds., Distributed Artificial Intelligence : Theory and Praxis, Kluwer Academic Publishers, pp. 53-80, 1992.

[Ferber & Jacopin, 1990] Ferber J., Jacopin E.; The Framework of Eco-Problem Solving, In: Proceedings of the second MAAMAW, Saint-Quentin en Yvelines, August, 13-16, 1990.

[Ferber, 1994] Ferber J., (1994), La Kenetique : des systemes multi-agents a une science de linteraction, In : Revue Internationale de systémique, Vol. 8, N° 1, pp. 13-27, 1994.

[Ferber, 1995] Ferber J.; Les Systemes Multi-Agents. Interéditions, Paris, 1995.

[Ferber, 1999] Ferber, Jacques, Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence, Addison Wesley Longman, Harlow, UK, 1999.

[Finin & Fritzson, 1992a] FININ, Tim; FRITZON, Rich; MCKAY, Don. A Language and Protocol to Support Intelligent Agent Interoperability. University of Maryland Baltimore, 1992.

[Finin & Labrou, 1992a] - FININ, Tim; LABROU, Yannis. A semantics approach for KQML - a general purpose communication language for software agents. University of Maryland Baltimore County., 1992.

[Finin & Labrou, 1992b] FININ, Tim; LABROU Yannis; MAYFIELD, James. Evaluation of KQML as an Agent Communication Language. Computer Science and Electrical Engineering Department. University of Maryland Baltimore County. Baltimore, USA, 1992.

[Finin & Labrou, 1996] FININ, Tim, LABROU, Yannis. A Proposal for a new KQML Specification. Universidade de Maryland Baltimore Count (UMBC), 1996.

[Finin & Labrou, 1997] - FININ, Tim; LABROU, Yannis. A Proposal for a new KQML Specification. University of Maryland Baltimore Count(UMBC), 1997.

[Finin et al, 1993] Finin T., Weber J., Wiederhold G., Genesereth M., Fritzson F., McKay D., McGuire J., Pelavin P., Shapiro S.; Specification of the KQML Agent-Communication Language, Technical Report EIT TR 92-04, Enterprise Integration Technologies, Palo Alto, CA, julho, 1993.

[Finin, 1996] FININ, Tim. UMBC KQML Web. Lab for Advanced Information Technology, (<http://www.cs.umbc.edu/kqml>), 1996

[Foner, 2004] Foner, Lenny, "What's an Agent, Anyway? A Sociological Case Study", [URL: [http://foner.www.media.mit.edu/people/foner/julia/subssection3\\_4.html](http://foner.www.media.mit.edu/people/foner/julia/subssection3_4.html) ], acessado em 26/02/2004, 2004.

[Fox, 1988] Fox M. S., ; An Organizational View of Distributed Systems, In : Reading Distributed Artificial Intelligence, A.H..Bond & L. Gasser (editors), Morgan Kaufman Publishers, San Mateo, California, 1988.

[Franklin & Graesser, 1997] Franklin, Stan; Graesser, Art Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, published as Intelligent Agents III, pp 21-35, Springer-Verlag, 1997

[Freitas, 1998] Freitas, A.A. Lavington, S.H.; Mining Very Large Databases with Parallel Processing, MA: Kluwer Academic Publishers, 1998.

[Galliers, 1988] GALLIERS, J. R. A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent conflict. PhD Thesis, Open University, UK, 1988.

[Garijo, 1999] Garijo, Francisco J., and Magnus Boman ed., Multi-Agent System Engineering: Proceedings of MAAMAW'99, Springer, Berlin, Germany, 1999.

[Genesereth & Ketchpel, 1994] Genesereth M.R., Ketchpel S. P., Software Agents. Communications of the ACM. Vol. 37 (7). July 48-53, 1994.

[Genesereth et al, 1988] Genesereth M. R., Ginsberg M.L., Rosenschein J.S.; Cooperation without Communication. In: Readings in Distributed Artificial Intelligence, A. H. & L. Gasser, editors, Morgan Kaufmann Publishers, San Mateo, California 220-226, 1988.

[Grein & Machado, 2002] Grein, Dirceu; Machado, Dennys; Dias, João da Silva. Proposta de Integração dos Sistemas de Informações em Saúde com o Cartão Nacional de Saúde. CBIS – Congresso Brasileiro de Informática em Saúde, 2002

[Grupo pré-hospitalar, 2001] – Grupo de estudos de atendimento pré-hospitalar. Disponível em <<http://www.aph.com.br>>. Primeiro acesso em: 10 de outubro de 2001, 2001.

[Guessoum, 1996] Guessoum Z.; Un environnement opérationnel de conception et réalisation de systèmes multi-agents, Thèse de doctorat, Université Pierre et Marie Curie, avril, 1996.

[Hagg & Ygge, 1994] Hagg S., Ygge F.; An Architecture for Agent-Oriented Programming with a Programmable Model of Interaction, In : Proceedings of the Seventh Annual Conference on AI and Cognitive Science, September 8-9, 1994.

[Halfpap & Belli, 1997] Halfpap, Dulce M., Belli, Mauro J.(), O Papel de Sistemas Inteligentes para o desenvolvimento de Organizações Virtuais no ambiente Internet, Universidade Federal de Santa Catarina. [[www.ufsc.br](http://www.ufsc.br)], Acessado em 10/2003, 1997.

[Hayes-Roth, 1988] Hayes-Roth B.; A Blackboard Architecture for Control, In : Bond A., Gasser L., eds., Readings in Distributed Artificial Intelligence, Morgan Kaufman, 1988.

[Hayes-Roth, 1990] Hayes-Roth B.; Architectural Foundations for Real-Time Performance in Intelligent Agents, In : Real-Time Systems, may, 1990.

[Hayes-Roth, 1995] Hayes-Roth, B. "An Architecture for Adaptive Intelligent Systems", Artificial Intelligence: Special Issue on Agent and Interactivity, vol. 72, 1995



[Heilmann, 1995] HEILMANN, Kathryn et al. "Intelligent Agents: A Technology and Business Application Analysis", Documento web <http://haas.berkeley.edu/~heilmann/agents/index.html>, acessado em 12/01/2003, 1995.

[Hendricks, 2002] HENDRICKS, Mack et al. Professional Java Web Service. Rio de Janeiro: Editora Alta Books, 2002.

[Herlea, 1999] Herlea, Daniela E., Catholijun M. Jonker, Jan Treur, and Niek J.E. Wijngaards, ed., Specification of Behavioural Requirements within Compositional Multi-Agent System Design Springer, Valencia, Spain, 1999.

[Holland, 1975] Holland, J.H. Adaptation In Natural And Artificial Systems. University Of Michigan Press, Ann Arbor. 1975.

[Hubner, 2003] Hubner, Jomi Fred, Um Modelo de Reorganização de Sistemas Multiagentes, Tese de Doutorado apresentado a Escola Politécnica da Universidade de São Paulo, 2003.

[Iglesias, 1998] Iglesias, Carlos A., Mercedes Garijo, and José C. González, ed., A Survey of Agent-Oriented Methodologies University Pierre et Marie Curie, Paris, FR, 1998.

[Jennings & Wittig, 1992] Jennings N.R., Wittig T.; ARCHON : Theory and Practice, In : Distributed Artificial Intelligence : Theory and Praxis (eds. N.M. Avouris and L.Gasser), Kluwer Academic Press, 179-195, 1992.

[Jennings & Wooldridge, 1998] Jennings, N. R. e Wooldridge, M. J. Agent Technology: foundations, applications, and markets. Springer Verlag, London, 1998.

[Jonker, 1997] Jonker, Catholijn M., and Jan Treur, ed., Compositional Verification of Multi-Agent Systems: a Formal Analysis of Pro-activeness and Reactiveness Springer, 1997.

[Kalani, 2002] KALANI, Amit et. al. ASP.NET 1.0 with C#. Namespace Reference. Wrox, 2002.

[Kinny, 1996a] Kinny, David, and Michael Georgeff, "Modelling and Design of Multi-Agent Systems," Intelligent Agents III: Proceedings of the Third International workshop on Agent Theories, Architectures, and Languages (ATAL'96), ed., Springer, Heidelberg, 1996.

[Kinny, 1996b] Kinny, David, Michael Georgeff, and Anand Rao, "A Methodology and Modelling Technique for Systems of BDI Agents," Agents Breaking Away. 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96), Walter VandeVelde and John W. Perram ed., Springer, Berlin, 1996, pp. 56-71, 1996.

[Krulwich, 1997] KRULWICH, Bruce, "Automating The Internet: Agents as User Surrogates", IEEE Internet Computing, July-August , <http://computer.org/internet>, acessado em 26/02/2004, 1997.

[Laurel, 1990] LAUREL, Brenda. The Art Of Human- Computer Interface Design, 1990

[Lemaitre & Excelente, 1998] Lemaitre, C. e Excelente, C. B.; Multi-agent organization approach. In Garijo, F. J. e Lemaitre, C., editors, Proceedings of II Iberoamerican Workshop on DAI and MAS, 1998.

[Luck, 1995] LUCK, Michael, D'INVERNO, Mark. A Formal Framework for Agency and Autonomy In: ICMAS-95, 1995.

[Machado & Grein, 1999] Machado, Dennys; Grein, Dirceu. GEM – Gerenciamento de Emergências Médicas. I Workshop de Informática em Saúde, PUCPR – Hospital Cajuru. Curitiba, 1999.

[Machado et al, 2000] Machado, Dennys Grein, Dirceu Rutz, Linnyer Dias, João D., (2000), Gerenciamento de Emergências Médicas, Workshop de Informática em Saúde - Hospital Universitário Cajuru - HUC, Curitiba, 2000.

[Machado et al, 2004] Machado, Dennys; Grein Dirceu; Scalabrin, Edson; Dias, João. Uma Arquitetura Aberta Multi-Agente para o Pré-Atendimento Hospitalar. CBIS – Congresso Brasileiro de Informática em Saúde. Ribeirão Preto, 2004.

[Maes, 1994a] - Maes P., Agents that reduce work and information overload, In : Communications of the ACM, 37(7), July, 1994.

[Maes, 1994b] Maes P., (1994b), Social interface agents : Acquiring competence by learning from users and other agents, In : O. Etzioni (eds.) Software Agents - Papers from the Spring Symposium (Technical Report SS-94-03), pp. 71-78, AAAI Press, 1994.

[Maes, 1995] MAES, Pattie. "Artificial Life Meets Entertainment: Life Like Autonomous Agents", Communications of the ACM, vol. 38, 1995.

[Malone et al, 1991] Malone, T. W.; Crownton, K., Toward an Interdisciplinary Theory of Coordination. Tech. Rep. No 120. Massachusetts Institute of Technology, Center for Coordination Science, Cambridge, Mass, 1991.

[Marietto, 2000] Marietto, M.G.B.. Definição dinâmica de estratégias Instrucionais em Sistemas de Tutoria Inteligente: Uma Abordagem Multiagentes na WWW. Tese de doutorado de 1 INPG, ITA, Brasil, 2000

[Martin & Odell, 1998] Martin, James, and James J. Odell, Object-Oriented Methods: A Foundation, (UML edition), Prentice Hall, Englewood Cliffs, NJ, 1998.

[Mayfield et al, 1996] MAYFIELD, James; LABROU, Yannis; FININ, Tim. Evaluation of KQML as an Agent Communication Language. In: ECAI 95 Workshop on Agent Theories, Architectures & Languages, aug. 95, Montreal, Canada. Proceedings... Berlin : Springer-Verlag , 1996.

[Minsky, 1986] Minsky M.; The Society of Mind, Simon and Schuster, New York, 1986.

[Mitchell et al, 1994] Mitchell T., Caruana R., Freitag D., McDermott J., Zabowski D., Experience with a learning personal assistant, In : Communications of the ACM, 37(7), Julho, 1994.

[Nissen, 1995] NISSEN, Mark. Intelligent agents: A technology and business application. URL: <http://haas.berkeley.edu/~heilmann/agents>, 1995.

[Nodine et al, 1998] Nodine, Marian H., and Amy Unruh, "Facilitating Open Communication in Agent Systems: the InfoSleuth Infrastructure," Intelligent Agents IV: Agent Theories, Architectures, and Languages, Munindar P. Singh et al. ed., Springer, Berlin, pp. 281-296, 1998.

[O'Connor et al, 1995] Gilbert, D., Aparicio, M., Atkinson, B., Brady, S., Ciccarino, J., Grosof, B., O'Connor, P., Osisek, D., Pritko, S., Spagna, R., and Wilson, L. "IBM Intelligent Agent Strategy," working paper, IBM Corporation, 1995.

[Odell, 1999] Odell, James ed., Agent Technology, OMG, green paper produced by the OMG Agent `Working Group, 1999.

[Paraiso, 1996] PARAISO, Emerson C. MASC – Sistema Multi-Agente para Monitoração e Controle de Processo. International ICSC Symposium on Engineering of Intelligent Systems - Tenerife, Espanha, 1996.

[Parolin, 1998a] PAROLIN, M.K.F, ABRAHÃO, M.T.F., VILAR, G. Sistema de Gerenciamento de Emergências Médicas – GEM, FNTCS – Fórum Nacional de Ciência e Tecnologia em Saúde. Curitiba PR., 1998.

[Parolin, 1998b] PAROLIN, M.K.F. Desenvolvimento de um Sistema de Gerenciamento e Controle de Viaturas em um Serviço de Emergências Médicas – GEM. 1o Congresso Latino-americano de Engenharia Biomédica, Mazatlan-México, 1998.

[Parolin, 1999] PAROLIN, M.K.F. Gerenciamento de Emergência Médica num Serviço de Atendimento Pré-hospitalar. III Congresso da Sociedade Brasileira de Atendimento Integrado ao Traumatizado. São Paulo, SP, 1999.

[Parunak, 1996] Parunak, H. Van Dyke, ed., Visualizing Agent Conversations: Using Enhanced Dooley Graphs for Agent Design and Analysis 1996.

[Parunak, 1998] Parunak, H. Van Dyke, John Sauter, and Steven J. Clark, "Toward the Specification and Design of Industrial Synthetic Ecosystems," Intelligent Agents IV: Agent Theories, Architectures, and Languages, Munindar P. Singh et al. ed., Springer, Berlin, 1998, pp. 45-59, 1998.

[Rumbaugh & Jacobson & Boock, 1999] Rumbaugh, James, Ivar Jacobson, and Grady Booch, The Unified Modeling Language Reference Manual, Addison-Wesley, Reading, MA, 1999.

[Russel & Norvig, 1995] Russel, J. Stuart e Norvig, P., Artificial Intelligence: A Modern Approach, Prentice Hall, EUA, 1995.

[SAMU Franca, 2002] – Apresentação do SAMU da França. Disponível em <<http://www.uvp5.univ-paris5.fr/SAMU/Present/PresentPO.asp>> . Acesso em março de 2002.

[SAMU, 2002a] – Histórico SAMU. Disponível em <[http://www.campinas.sp.gov.br/saude/unidades/samu/samu\\_hist.htm](http://www.campinas.sp.gov.br/saude/unidades/samu/samu_hist.htm)>. Acesso em março de 2002.

[SAMU, 2002b]- SAMU atendimento de origem clínica. Disponível em <[http://www.campinas.sp.gov.br/saude/unidades/samu/samu\\_missao.htm](http://www.campinas.sp.gov.br/saude/unidades/samu/samu_missao.htm)> Acesso em março de 2002.

[Scalabrin et al, 1996] Scalabrin, E., Carvalho, D., Angelotti, E., Azevedo, H., Ramos, M. An Open Multiagent Architecture to Improve Reliability and Adaptability of Systems, 1996.

[Scalabrin, 1996] Scalabrin, E. E., Conception et Realisation d'environnement de développement de systemes d'agents cognitifs. PhD Thesis. Université de Technologie de Compiègne, France, 169, 1996.

[Shen & Barthes, 1995] Shen W., Barthes J-P.; DIDE : A Multi-agent environnement for Enginnering Design, In : Proceedings First International Conference on Multi-Agent Systems, AAAI Press/The MIT Press, june 12-14, San Francisco, California, pp. 344-351, 1995

[Shen & Barthes, 1996] Shen W., Barthes J-P.; An Experimental Multi-Agent Environnement for Engineering Design, In : International Journal of Cooperative Information Systems, Ed. World Scientific Publishing Company, Vol. 5, Nos. 2 & 3, pp. 131-151, 1996.

[Shmeil, 1999] Shmeil, Marcos Augusto Hochuli., Sistemas Multiagente na Modelação da Estrutura e Relações de Contratação de Organizações. Tese de Doutorado, 1999.

[Shoham, 1993] Shoham Y., Agent-oriented programming, In : Artificial Intelligence, 60(1) :51-92, 1993.

[SIATE, 2002] – SIATE de Curitiba. Disponível em <[http://www.aph.com.br/siate\\_curitiba.htm](http://www.aph.com.br/siate_curitiba.htm)> . Acesso em março de 2002.

[Sichman, 1995] Sichman, J. S.; Du Raisonment Social Chez les Agents: Une Approche Fondée sur la Theorie de la D'épendance. These (doctorat), Institut National Polytechnique de Grenoble, 1995.

[Sichman, 1998] Sichman, J. S.; DEPINT: Dependence-based coalition formation in an open multiagent scenario. Journal of Artificial Societies and Social Simulation, 1(2), 1998.Doc Web <http://jasss.soc.surrey.ac.uk/1/2/3.html>, 1998.

[Sichman, et al 1994] Sichman, J. S., Conte, R., Demazeau, Y., e Castelfranchi, C.; A social reasoning mechanism based on dependence networks. In Cohn, T., editor, Proceedings of the 11th European Conference on Artificial Intelligence, pages 188–192, 1994.

[Simon, 1960] Simon, H.A., Decision Making and Organizational Design, em Organization Theory, Edited by D.S. Pugh, Penguin Books, 1960.

[Singh & Rein, 1992] Singh, B., Rein, G. L., Role interaction nets (RIN): A process description formalism. Tech. Rep. CT-083-92. Microelectronics and Computer Tecnology Corp., Autin, Text, 1992

[Singh, 1998a] Singh, Munindar P., "A Customizable Coordination Service for Autonomous Agents," *Intelligent Agents IV: Agent Theories, Architectures, and Languages*, Munindar P. Singh et al. ed., Springer, Berlin, pp. 93-106, 1998.

[Singh, 1998b] Singh, Munindar P., ed., *Developing Formal Specifications to Coordinate Heterogeneous Autonomous Agents* IEEE Computer Society, Paris, FR, 1998.

[Smith et al, 1994] SMITH, D. C., CIFRA e J. SPOHER. "KidSim: Programming Agents Without a Programming Language", *Communications of the ACM*, 1994, vol. 37, 1994.

[Smith, & Davis, 1981] Smith R.G., Davis R.; *Frameworks for Cooperation in Distributed Problem Solving*. IEEE Transactions on Systems, Man and Cybernetics, 1981.

[Smith, 1980] R. G. Smith *The Contract net Protocol: High Level Communication and Control in a Distributed Problem Solver: IEEE Transaction and Computer*, 1980

[Smith, 1981] D. R. Smith, R.G. *Framework for cooperation in Distributed Problem Solving*. IEEE Transaction on System, Man, and Cybernetic, 1981

[Steels, 1989] Steels L.; *Cooperation between Distributed Agents Through Self-Organization*, In : *Proceedings of the First MAAMAW*, Cambridge, England, August, pp. 16-18, 1989.

[Sycara et al, 1996] Sycara K., Decker K., Williamson M., Pannu A., (1996), *Distributed Intelligent Agents*, In : *IEEE Expert*, July, 1996.

[Torres & Constantino, 1996] TORRES M. CONSTANTINO; *Polveri da fiuto allucinogene nel Cile precolombiano*, *Altrove*, Torino, vol. 3, pp. 29-39, 1996.

[Wayer, 1995] WAYER, P. *Agents unleashed*. USA: AP Professional, 1995.

[Wildt, 2002] Wildt, Daniel; *Web Services Interoperabilidade e integração entre sistemas no Delphi 7 e Delphi 8 for .NET*, 2002.

[Wooldridge & Jennings, 1994] Wooldridge M. J., Jennings N. R.; *Agent Theories, Architectures, and Languages: A Survey*, *Workshop on Agent Theories, Architectures and Languages*, ECAI94, Amsterdam, 1994.

[Wooldridge & Jennings, 1995] Wooldridge M., Jennings N., (1995), *Intelligent agents : theory and practice*, In : *The knowledge Engineering Review*, Vol. 10 (2), pp. 115-152, 1995.

[Wooldridge et al, 2000] Wooldridge, Michael, Nicholas R. Jennings, and David Kinny, "The Gaia Methodology for Agent-Oriented Analysis and Design," *International Journal of Autonomous Agents and Multi-Agent Systems*, 3:Forthcoming, 2000.

[Wooldridge, 2002] Wooldridge, M.; *An Introduction to MultiAgent Systems*. John Wiley and Sons, 2002.