

DIRCEU GREIN

**Uma contribuição para a integração de sistemas
legados da Saúde Pública do Brasil usando agentes
de software**

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

CURITIBA

2005

DIRCEU GREIN

**Uma contribuição para a integração de sistemas
legados da Saúde Pública do Brasil usando agentes
de software**

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

Área de Concentração: Informática em Saúde

Orientador: Prof. Dr. Eng. João da Silva Dias

Co-orientador: Prof. Dr. Edson E. Scalabrin

CURITIBA

2005

Grein, Dirceu

Uma contribuição para a integração de sistemas legados da Saúde Pública do Brasil usando agentes de software. Curitiba, 2005. 146p.

Dissertação – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Tecnologia em Saúde.

1. Saúde Pública 2. Sistema Multi-Agente 3. Integração de Sistema Legado
4. Web Semântica.

I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Biológicas e da Saúde. Programa de Pós-Graduação em Tecnologia em Saúde. II. Sistema de Informação.

FOLHA DE APROVAÇÃO

*"São fúteis e cheias de erros as ciências
que não nasceram da experimentação,
mãe de todo conhecimento".*

Leonardo da Vinci (1452-1519)

A Deus,
Aos meus queridos Pais,
A minha amada Esposa.

Agradecimentos

Deus nos deu a vida, sobre responsabilidade dos Pais, construiu as condições necessárias para o nosso desenvolvimento, pôs em nós o amor com a liberdade para podermos descobrir o universo, onde cada um de nós faz a diferença.

Gostaria de agradecer a todas as pessoas que de alguma forma fizeram a diferença durante o desenvolvimento deste trabalho, colegas, amigos, familiares, professores e alunos. Sou muito grato aos orientadores por terem compartilhado o conhecimento, dando às condições necessárias para construção desta pesquisa.

Em especial gostaria de agradecer as pessoas que compartilharam diretamente seu tempo e dedicação, muito obrigado mãe, pai, irmãos, e principalmente a minha esposa, que soube tolerar a falta de tempo, sempre me apoiando e incentivando até a conclusão da dissertação.

Publicações

GREIN, D.; MACHADO, D.A.; DIAS, J.S. *Proposta de Integração de Sistema de Informação em Saúde com o Cartão Nacional de Saúde*. VIII Congresso Brasileiro de Informática em Saúde, Natal, 2002.

MACHADO, D.A.; GREIN, D.; SCALABRIN, E.E.; DIAS, J.S. *Uma Arquitetura Aberta Multi-Agente para o Pré-Atendimento Hospitalar*. IX Congresso Brasileiro de Informática em Saúde, Ribeirão Preto, 2004.

MACHADO, D.A.; GREIN, D.; DIAS, J.S. *Sistema de Informação para Unidades Móveis de Pré-Atendimento Hospitalar*. VIII Congresso Brasileiro de Informática em Saúde, Natal, 2002.

PEREIRA, P.; GREIN, D.; MACHADO, D.; NOHAMA, P. *Leitor Automatizado para Cartões de Testes Imunohematológicos*. Congresso Brasileiro de Engenharia Biomédica, 2002.

Sumário

Agradecimentos	v
Publicações	vi
Sumário	vii
Lista de Figuras	x
Lista de Tabelas	xii
Lista de Tabelas	xii
Lista de Abreviaturas	xiii
Resumo	xiv
Abstract	xv

Capítulo 1 - Introdução **1**

1.1. Objetivos	3
1.1.1. <i>Objetivo Geral</i>	3
1.1.2. <i>Objetivo Específico</i>	3
1.2. Proposta de Trabalho	3
1.3. Estrutura do Documento	4

Capítulo 2 - Revisão Bibliográfica **7**

2.1. Sistema de Saúde Pública do Brasil	7
2.1.1. <i>Evolução do Sistema de Saúde Pública do Brasil</i>	7
2.1.2. <i>Sistema Único de Saúde</i>	8
2.2. Descentralização dos Sistemas de Informação do SUS	11
2.3. Sistemas de Informação Legados do SUS	17
2.3.1. <i>Área de Vigilância Epidemiológica</i>	17
2.3.2. <i>Área de Assistência em Saúde</i>	19
2.4. Sistema Cartão Nacional de Saúde	20
2.4.1. <i>Objetivos do Sistema</i>	21
2.4.2. <i>Arquitetura do Sistema</i>	21
2.4.3. <i>Situação atual do Sistema</i>	23
2.5. Arquitetura de Sistemas de Informação	25
2.5.1. <i>Princípios de Arquitetura de Software</i>	26
2.5.2. <i>Estilos Arquiteturais de Software</i>	27
2.6. Considerações Finais do Capítulo	28

Capítulo 3 - Agentes de Software	29
3.1. Conceitos de Agentes de Software	29
3.2. Propriedades de Agentes	31
3.3. Sistemas Multi-Agente	33
3.3.1. <i>Sistemas Reativos e Cognitivos</i>	35
3.4. Ambiente do Sistema.....	36
3.5. Organizações de Agentes	37
3.6. Modelo de Agentes.....	37
3.7. Linguagem de Comunicação de Agentes	41
3.7.1. <i>Linguagem de comunicação KQML</i>	43
3.8. Protocolo de Interação.....	46
3.8.1. <i>Protocolo de Interação: FIPA-Request Protocol e FIPA-Cancel Meta Protocol</i>	47
3.9. Linguagem de Conteúdo	49
3.10. Considerações Finais do Capítulo	50
Capítulo 4 - Ontologias	51
4.1. Tipos de Ontologias.....	53
4.2. Níveis de abordagem de ontologia	54
4.3. Benefícios e problemas relacionado a ontologias.....	55
4.4. Aplicações de Ontologias em Agentes de Software	57
4.5. Considerações Finais do Capítulo	59
Capítulo 5 - Web Semântica	61
5.1. Extensible Markup Language (XML)	62
5.2. Document Type Definition (DTD).....	64
5.3. XML Schema	66
5.4. Resource Description Framework (RDF).....	68
Capítulo 6 - Arquitetura Web Service	71
6.1. Estrutura Funcional de <i>Web Service</i>	72
6.2. Padrões ligados a <i>Arquitetura Web Service</i>	74
6.2.1. <i>Rede de Transporte TCP/IP</i>	75
6.2.2. <i>Camada de Comunicação</i>	76
6.2.3. <i>Simple Object Access Protocol (SOAP)</i>	77
6.2.4. <i>Descrição de Serviços</i>	79
6.2.5. <i>Processos de Descoberta, Agregação e Coordenação</i>	83
6.2.6. <i>Processo de Descoberta de Serviços</i>	83
6.2.7. <i>Universal Description, Discovery and Integration (UDDI)</i>	86
6.2.8. <i>Processo de Agregação e Coordenação</i>	90

Capítulo 7 - Proposta de um Sistema de Agentes Orientado a Serviço para o SUS 93

7.1. AUML - Linguagem de Modelagem Unificada para Agente	93
7.1.1. Modelagem UML estendida para agentes.....	94
7.1.2. Modelagem sobre novo aspecto no contexto de agentes	94
7.2. Modelagem do contexto Epidemiológico e de Assistência em Saúde.....	95
7.2.1. Vigilância Epidemiológica	96
7.2.2. Assistência em Saúde.....	107
7.3. Arquitetura do Sistema de Agentes aplicado no Perfil de Saúde.....	116
7.3.1. Estrutura de um Agente	117
7.3.2. Estrutura de Comunicação.....	118
7.3.3. Integridade dos Dados Interoperados	119
7.4. Implementação do Sistema de Agentes Orientados a Serviços para o SUS	121

Capítulo 8 - Conclusão 127

8.1. Resultados e Trabalhos Futuros.....	128
--	-----

Referências Bibliográficas 131

Lista de Figuras

Figura 2.1: Organização dos Gestores do Sistema Único de Saúde.	10
Figura 2.2: Arquitetura do sistema SCNS [LEA02].	22
Figura 3.1: Representação conceitual de um agente [FER95].	38
Figura 3.2: Modelo do agente GAg (<i>Generic Agent</i>) [SCA96].	38
Figura 3.3: Modelo de agente ARCHON [JEN92].	39
Figura 3.4: Exemplo de Mensagem KQML	45
Figura 3.5: Protocolo <i>FIPA Request Interaction Protocol</i>	47
Figura 3.6: Exemplo de mensagem utilizando o protocolo <i>FIPA-Request Protocol</i>	48
Figura 3.7: Protocolo <i>FIPA Cancel Meta Protocol</i>	48
Figura 4.1: Conceituação e ontologia na representação de domínios.	52
Figura 4.2: Classificação de ontologias conforme o grau de generalidade.	54
Figura 5.1: A arquitetura de camadas da Web Semântica [BER01].	61
Figura 6.1: Estrutura Funcional de <i>Web Service</i>	73
Figura 6.2: Arquitetura <i>Web Service</i> [BOO04].	74
Figura 6.3: Mensagem SOAP para uma solicitação (<i>Request</i>).	78
Figura 6.4: Mensagem SOAP para uma Resposta (<i>Response</i>)	78
Figura 6.5: Documento WSDL descrevendo o serviço <i>AssistenciaSaude</i>	82
Figura 6.6: Estrutura do Registro UDDI de uma organização.	88
Figura 7.1: Fluxo operacional do sistema SINAN.	97
Figura 7.2: Diagrama de caso de uso do controle de morbidade (SINAN).	98
Figura 7.3: Diagrama de classe do agente SINAN.	99
Figura 7.4: Fluxo operacional de Declaração de Nascido Vivo.	101
Figura 7.5: Diagrama de caso de uso do registro de natalidade (SINASC).	101
Figura 7.6: Diagrama de classe do agente SINASC	102
Figura 7.7: Fluxo operacional de Declaração de Óbito	104
Figura 7.8: Diagrama de caso de uso do registro de óbito (SIM).	105
Figura 7.9: Diagrama de classe do agente SIM.	106
Figura 7.10: Diagrama de caso de uso da assistência ambulatorial	109
Figura 7.11: Diagrama de classe do agente Unidade Básica de Saúde.	110
Figura 7.12: Diagrama de caso de uso da assistência hospitalar.	112
Figura 7.13: Diagrama de classe do agente Hospital.	113
Figura 7.14: Diagrama de caso de uso da assistência laboratorial.	114
Figura 7.15: Diagrama de classe do agente Laboratório.	115
Figura 7.16: Estrutura de um Agente.	118
Figura 7.17: Estrutura de Comunicação	119
Figura 7.18: Troca de mensagens KQML	120

Figura 7.19: Camadas da Arquitetura de Comunicação dos Agentes.....	122
Figura 7.20: Esquema de Comunicação do Sistema de Agentes Orientado a Serviços.....	123
Figura 7.21: Interface para Configuração e Monitoramento de Agente (Identificação).....	124
Figura 7.22: Interface para Configuração e Monitoramento de Agente (Serviços).....	124
Figura 7.23: Estrutura de Funcionamento do Sistema de Agentes para o SUS.	125

Lista de Tabelas

Tabela 2.1: Distribuição de estabelecimentos de saúde segundo o tipo de unidade de saúde [MIN05a].	14
Tabela 2.2: Distribuição de estabelecimentos de saúde segundo a natureza da organização [MIN05a].	15
Tabela 3.1: Representação dos Níveis da Linguagem de Comunicação de Agentes.	42
Tabela 3.2: Estrutura de parâmetros de mensagem KQML.	44
Tabela 5.1: Exemplo de conteúdo representado em XML.	63
Tabela 5.2: Exemplo de definição de esquema em DTD.	65
Tabela 5.3: Exemplo de definição de esquema em XML Schema.	67
Tabela 5.4: Exemplo de dados em formato RDF.	69
Tabela 6.1: Camadas da Arquitetura TCP/IP.	75
Tabela 7.1: Nascidos por local de ocorrência e região referente ao ano 2002 [MIN05b].	100
Tabela 7.2: Quantidade de procedimentos ambulatoriais por tipo de prestador e ano [MIN05c].	107
Tabela 7.3: Quantidade de internações hospitalares por tipo de regime do prestador [MIN05d].	107

Lista de Abreviaturas

AUML - Linguagem de Modelagem Unificada para Agente
CNES - Cadastro Nacional de Estabelecimentos de Saúde
SCNS – Sistema Cartão Nacional de Saúde
COM - Common Object Model
CORBA - Common Object Request Broker Architecture
DTD - Document Type Definition
HTTP - Hypertext Transfer Protocol
CID – Código Internacional de Doença
IDL - Interface Definition Language
IEEE - - Institute of Electrical and Electronics Engineers
ISO - International Organization for Standardization
JMS - Java Message Service
MS - Ministério da Saúde
NOB - Norma Operacional Básica do SUS.
RMI - Remote Method Invocation
RNIS - Rede Nacional de Informação em Saúde
SES - Secretaria Estadual de Saúde
SMS - Secretaria Municipal de Saúde
SOAP - Simple Object Access Protocol
SUS - Sistema Único de Saúde
TCP/IP - Transmission Control Protocol / Internet Protocol
UDDI - Universal Description, Discovery and Integration
UML - Linguagem de Modelagem Unificada
W3C - World Wide Web Consortium
WSDL - Web Services Description Language
XML - eXtensible Markup Language

Resumo

A distribuição física e segmentada das áreas de saúde, entre um grande número de organizações, caracteriza a Saúde Pública como um sistema distribuído com diversas fontes de informações. Este trabalho apresenta uma abordagem de agentes de software para integrar as segmentações do conhecimento da Saúde Pública, buscando recursos de interoperabilidade entre os sistemas de informação, particularmente aplicados nas áreas de epidemiologia e de assistência de saúde. Desta forma, baseado em um modelo de agente, é constituída uma arquitetura de agentes que encapsula os sistemas legados da saúde como agentes de serviços, criando mecanismos de interação entre os sistemas, tendo como objetivo promover: a) flexibilidade operacional entre os sistemas; b) intercâmbio de informações estruturadas de forma íntegra; c) interoperabilidade dos sistemas; d) possibilidade de extensão de competências dos agentes (serviços). A arquitetura proposta é suportada pela plataforma *Web Service* utilizando o protocolo SOAP como transporte de mensagens entre os agentes e empregando a linguagem KQML como linguagem de comunicação de agentes. O compartilhamento do conhecimento é expresso pelo modelo semântico RDF e o conteúdo das fontes de dados é representado no formato XML e validado por meio de estruturas definidas na linguagem de esquema DTD. Por fim, a arquitetura proposta é aplicada em um protótipo para monitoração do perfil de saúde no Estado do Paraná, almejando fornecer os mecanismos necessários para interação e colaboração entre os setores competentes pelas ações de Saúde Pública, disponibilizando as informações e serviços providos pelos sistemas de informações das áreas afins.

Palavras-chave: Saúde Pública, Sistema Multi-Agente, Integração de Sistema Legado, Web Semântica.

Abstract

The physical and segmented distribution of the areas of health, among a great number of organizations, it characterizes the public health as a system distributed with several sources of information. This work presents an approach of software agents to integrate the segmentations of the knowledge of the Public Health, looking for interoperability resources among the systems of information, particularly applied in the epidemiology areas and of attendance of health. This way, based on an agent model, it is constituted an agents architecture that encapsulates the systems legacies of the health as agents of services, creating interaction mechanisms among the systems, tends as objective to promote: the) operational flexibility among the systems; b) exchange of structured information in an entire way; c) interoperability of the systems; d) possibility of extension of the agents' competences (services). THE architecture proposal is supported by the platform *Web Service* using the protocol SOAP as transport of messages among the agents and using the language KQML as language of agents' communication. The sharing of the knowledge is expressed by the semantic model RDF and the content of the sources of data is represented in the format XML and validated by means of structures defined in the outline language DTD. Finally, the architecture proposal is applied in a prototype for monitoring of the profile of health in the State of Paraná, longing for to supply the necessary mechanisms for interaction and collaboration among the competent sections for the actions of Public Health, provider the information and services by the systems of information of the kindred areas.

Key-words: Public Health, Multi-Agent System, Integration of Legacy System, Semantic Web.

Capítulo 1 - Introdução

A informática é um meio que possibilita a abstração do mundo real de forma simbólica ou numérica, representada pelos sistemas de informação em forma de dados [YOU90]. Ela, aliada aos meios de comunicação modernos, tem possibilitado a concepção e aplicação de novas tecnologias, reduzindo cada vez mais à distância entre as pessoas e organizações, revitalizando as diversas áreas da ciência.

Na área da saúde, a aplicação da informática busca no meio dos recursos computacionais, desenvolver sistemáticas que possibilitem a representação do conhecimento de forma a contribuir com o meio econômico, social e assistencial, aplicando técnicas e tecnologias que flexibilizem a eficiência na área de saúde.

A área da saúde é um dos campos do conhecimento mais amplos e segmentados, e mesmo sendo uma das ciências mais antiga, continua sendo inovadora até os dias atuais [DEG96]. A amplitude da área de saúde cria a necessidade de segmentar o conhecimento, dividindo-o em setores específicos (especializados). A segmentação do conhecimento facilita o estudo e compreensão aumentando a eficiência das competências. Esta segmentação molda as organizações de saúde com competências específicas, para atuar com ações focalizadas sobre determinados objetivos de cada segmento [LOU03].

Este mesmo modelo é percebido na Saúde Pública do Brasil, distribuindo os conhecimentos entre os setores com competências específicas. A segmentação do conhecimento em competências fortalece a eficiência da organização, mas acaba-se criando estruturas multidisciplinares e organizações distribuídas, mesmo fazendo parte de um sistema único organizacional, o Sistema Único de Saúde (SUS) [NOB96].

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 Norma Operacional Básica do Sistema Único de Saúde (NOB 01/96) amparada pela Lei 8.080/90 da Constituição Brasileira de 1988, preocupa-se expressamente em estabelecer diretrizes para gestão descentralizada do SUS.

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 informação para

atender objetivos específicos, suprindo 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 entre 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 tais como: mortalidade, morbidade, natalidade, hospitalar, ambulatorial, imunização, laboratorial, farmacêutica, etc [DAT04a].

Além dos sistemas legados do SUS, atualmente a área da saúde é composta por uma diversidade de sistemas, desenvolvidos por vários fornecedores e com tecnologias diferentes. Deste modo, pela dificuldade de comunicação entre estes sistemas, acabam constituindo sistemas isolados, restringindo o potencial da organização ([DAT04a], [ILH93], [RNIS04]). A necessidade de comunicação e interação é essencial em diversas áreas da saúde como hospitais, clínicas, unidades de saúde, laboratórios, hemocentros, etc.

Busca-se através de uma arquitetura de agentes autônomos independentes, suportados pela plataforma Web, integrar as segmentações do conhecimento no SUS, oferecendo um mecanismo de interação entre os sistemas legados existentes, proporcionando a ação colaborativa entre as diversas entidades que compõem o SUS, promovendo uma unificação do sistema de forma flexível ([SHO90], [GEN94], [FER99]).

A arquitetura baseada em agentes elaborada neste trabalho, tem como fundamentação os seguintes princípios: a) flexibilidade operacional: cada sistema de informação pode representar as informações dentro do contexto de sua aplicação, apresentando os dados que representem seus conhecimentos e competências de forma independente; b) integridade dos dados: os dados devem ser validados conforme suas estruturas, para manter íntegro os dados dentro de um contexto; c) interoperabilidade dos sistemas: os sistemas devem possuir mecanismos para comunicação (compartilhamento) de dados e conhecimentos.

Almejando a integração do conhecimento gerado nos sistemas de informação do SUS, com base nos princípios citados, foi concebida uma arquitetura que pode encapsular os sistemas legados existentes dentro de uma estrutura de sistemas de agentes de software, onde cada instância do sistema passará a ser um agente, proporcionando a integração e colaboração entre os sistemas.

Com esse intuito, a estruturação dos sistemas de informação de forma distribuída e integrada através de sistemas de agentes, fornecerá mecanismos de interação e colaboração entre os setores competentes pelas ações de Saúde Pública.

1.1. Objetivos

1.1.1. Objetivo Geral

O objetivo deste trabalho é propor uma forma de integração dos sistemas legados da Saúde Pública do Brasil, apresentando mecanismos de interoperabilidade entre sistemas de informação, focalizando na gestão integrada da área de assistência em saúde e vigilância epidemiológica.

1.1.2. Objetivo Específico

1. Apresentar o sistema de Saúde Pública do Brasil, exibindo a gestão descentralizada do Sistema Único de Saúde;
2. Identificar e modelar os principais sistemas de informação aplicados na Saúde Pública do Brasil com gestão descentralizada;
3. Definir os mecanismos de interoperabilidade entre sistemas de informação computacionais, em particular os agentes de software;
4. Identificar as formas de representação do conhecimento relacionadas a tecnologias emergentes orientadas a serviço (*Web Service* e *Web Semântica*);
5. Modelar um sistema de agentes para gestão integrada da assistência de saúde e vigilância epidemiológica; e
6. Elaborar um protótipo para monitoração do perfil de saúde no Estado do Paraná.

1.2. Proposta de Trabalho

Os sistemas de informação em Saúde Pública são conhecidos pela geração contínua de informações aplicadas em contextos diferentes, tais como: assistência em saúde, vigilância epidemiológica e vigilância sanitária. Com a evolução estratégica da Saúde Pública,

descentralizando cada vez mais os serviços de saúde, surgem novos sistemas e fontes de informações, pois neste caso trata-se de organizações multi-institucionais (públicas e privadas). Sobre este aspecto, denota-se um crescente número de sistemas legados, aumentando a quantidade de bases de dados, cada uma com seus objetivos e finalidades. No entanto, o conhecimento gerado pelos sistemas é sub utilizado, pois a estrutura atual dos sistemas não comporta o compartilhamento das informações e conhecimentos produzidos. Em alguns casos, apresenta-se uma integração estática e rígida, através da transferência de arquivos de dados, na maioria dos casos transferidos de forma manual. Desta forma, acaba-se restringindo a evolução contínua dos sistemas em cada contexto da Saúde Pública.

A distribuição física e segmentada das áreas de saúde entre um grande número de organizações, caracteriza a necessidade de evoluir os sistemas de informação que suportem um ambiente descentralizado, dinâmico e heterogêneo. Neste trabalho, é proposta uma abordagem para integração dos sistemas legados da Saúde Pública, tendo como objetivo prover: a) flexibilidade operacional entre os sistemas; b) intercâmbio de informações estruturadas; c) interoperabilidade dos sistemas; d) possibilidade de evolução das competências em cada contexto da saúde.

A arquitetura proposta tem como base à aplicação de sistemas de agentes orientados a serviços, constituídos para proporcionar a colaboração entre estas organizações de saúde, criando mecanismos de integração do conhecimento. Este modelo é disposto, encapsulando os sistemas da Saúde Pública em agentes de software, abstraindo as capacidades do agente como serviços que o agente fornece. Para colaboração entre os agentes é proposto o uso de uma ontologia que represente através de uma semântica o conhecimento a ser compartilhado. A ontologia abstrai os detalhes sintáticos e evidencia os aspectos semânticos dos relacionamentos que constitui a informação, podendo potencializar a integração entre os sistemas legados do SUS.

1.3. Estrutura do Documento

Este documento está disposto em oito capítulos para melhor compreensão do trabalho. No Capítulo 1, descrevem-se os objetivos, especificando as metas da dissertação, sintetizando a proposta do trabalho. No Capítulo 2, constitui-se a revisão bibliográfica sobre o sistema da Saúde Pública e arquitetura de sistemas de informação, exibindo os fundamentos para

desenvolvimento da pesquisa. No Capítulo 3, focalizam-se os aspectos de agentes de software, contextualizando as características aplicadas em sistemas de agentes. No Capítulo 4, é abordada as formas de representação do conhecimento através de ontologias. No Capítulo 5, fundamenta-se a representação do conhecimento através de ontologias para Web denominada Web Semântica. No Capítulo 6, apresenta-se a arquitetura orientada a serviços (*Web Service*). No Capítulo 7, é concebida a arquitetura de agentes orientados a serviços proposta para suportar a integração de sistemas legados da Saúde Pública, como também são apresentados os resultados obtidos. No Capítulo 8, constitui-se a conclusão do trabalho, avaliando os resultados obtidos e propondo trabalhos futuros para serem agregados ao conhecimento desenvolvido.

Capítulo 2 - Revisão Bibliográfica

Neste capítulo será apresentado o sistema de Saúde Pública do Brasil, explanando de forma evolutiva as condições que levaram a constituir o Sistema Único de Saúde (SUS), observando a forma que está estruturado o atual sistema de Saúde Pública. Também serão abordados os sistemas de informações legados do SUS, os quais oferecem suporte operacional, identificados como fonte de informações, particularmente relacionados às áreas de vigilância epidemiológica e de assistência de saúde. Por fim, serão apresentados os fundamentos desta pesquisa sobre a necessidade de constituir uma arquitetura de sistema de informação para prover a integração dos sistemas legados do SUS.

2.1. Sistema de Saúde Pública do Brasil

Nesta seção será apresentado um breve histórico da evolução do sistema de Saúde Pública do Brasil, exibindo as características que moldaram o Sistema Único de Saúde, caracterizando-o como um sistema distribuído entre organizações gestoras da esfera municipal, estadual e federal.

2.1.1. Evolução do Sistema de Saúde Pública do Brasil

O sistema de Saúde Pública do Brasil sofreu muitas mudanças políticas e estratégicas nas últimas décadas. Em meados do século passado até os anos sessenta, o sistema de Saúde Pública caracterizou-se pelo combate em massa de doenças, com atenção à saúde coletiva. Este período foi caracterizado pela gestão centralizada da Saúde Pública. A partir dos anos setenta, iniciou-se um processo de atenção à assistência médica curativa e individual. A expansão deste processo foi intensificada pela reforma sanitária brasileira, cujo objetivo era a “*construção de uma nova política de saúde efetivamente democrática*”, considerando a descentralização, universalização e unificação do sistema de Saúde Pública [CAR98].

Pela Lei 6.229 de 1975, foi criado o Sistema Nacional de Saúde, separando as ações de Saúde Pública (vigilância em saúde) e as ações de atenção a assistência médica individual

(assistência de saúde). Em 1977, foi criado o Instituto Nacional de Assistência Médica da Previdência Social (INAMPS), com objetivo de atender as pessoas em regime CLT (Consolidações das Leis do Trabalho), as demais pessoas eram de responsabilidade das secretarias estaduais de saúde e dos serviços públicos federais [MED94].

A partir da década de oitenta, começou a ser direcionado o modelo de saúde voltado a atenção primária à saúde (assistência de saúde), buscando criar um novo modelo. Neste contexto, muitas propostas de implantação de rede de serviços foram surgindo: Programa de Interiorização das Ações de Saúde e Saneamento (PIASS); Programa Nacional de Serviços Básicos de Saúde (PREV-SAÚDE), Ações Integradas de Saúde (AIS). Este último contribuiu muito para a descentralização da Saúde Pública [MED94].

Após a 8ª Conferência Nacional de Saúde em 1986, foram fortalecidos os princípios da Reforma Sanitária. E em 1987, criou-se o Sistema Unificado e Descentralizado de Saúde (SUDS), consolidando as Ações Integradas de Saúde e definindo diretrizes de universalização, descentralização e igualdade no acesso aos serviços [COS98]. Neste momento, foram descentralizadas as atividades do Instituto Nacional de Assistência Médica da Previdência Social para as Secretarias Estaduais de Saúde.

Através da Constituição Federal de 1988, foram incorporadas de forma parcial as propostas estabelecidas pelo movimento da reforma sanitária, criando o Sistema Único de Saúde [COS98]. Posteriormente em 1993, o Instituto Nacional de Assistência Médica da Previdência Social foi extinto pela Lei 8.689 [MED94].

2.1.2. Sistema Único de Saúde

O Sistema Único de Saúde foi criado pela Constituição Federal de 1988 e regulamentado pelas Leis n.º 8080/90 (Lei Orgânica da Saúde) e n.º 8.142/90, com a finalidade de alterar a situação de desigualdade na assistência à saúde da população, tornando obrigatório o atendimento público a qualquer cidadão [SEN88a]. O SUS representa o sistema de Saúde Pública do Brasil, constituindo a gestão e execução das ações de saúde em todas as esferas do governo.

Segundo a Lei Nº 8.080 de 1990, “o Sistema Único de Saúde representa o conjunto de ações e serviços de saúde, prestados por órgãos e instituições públicas federais,

estaduais e municipais, da administração direta e indireta e das fundações mantidas pelo poder público, incluídas as instituições públicas federais, estaduais e municipais de controle de qualidade, pesquisa e produção de insumos, medicamentos, inclusive de sangue e hemoderivados, e de equipamentos para saúde". O setor privado de saúde participa como forma complementar conforme contratos e convênios com o SUS [MIN04a].

Existe uma diversidade de estabelecimentos de saúde que fazem parte do Sistema Único de Saúde, como: centros e postos de saúde, hospitais, laboratórios, hemocentros, farmácias, unidade móvel, centrais de regulação, além das fundações e institutos de pesquisa, como a Fundação Oswaldo Cruz (FIOCRUZ) e o Instituto Vital Brasil [LOU03].

Através do Sistema Único de Saúde, todos os cidadãos têm direito a consultas, exames, internações e tratamentos nas unidades de saúde vinculadas ao SUS, sejam públicas municipais, estaduais, federais, ou privadas, contratadas pelo gestor público de saúde. O Sistema Único de Saúde tem como objetivo promover a equidade no atendimento das necessidades de saúde da população, ofertando serviços com qualidade adequados às necessidades. No entanto, para garantir a qualidade dos serviços de saúde, as ações do SUS são classificadas em áreas que compreendem [LOU03]:

- Vigilância Epidemiológica, que trata do acompanhamento de doenças (epidemias), da promoção à saúde, do monitoramento da mortalidade, natalidade e imunização;
- Vigilância Sanitária, que trata do controle da qualidade de remédios, de exames, de alimentos, da higiene, da adequação de instalações que atendem ao público (equipamentos e procedimentos), do controle de hemoderivados (sangue); e
- Assistência de Saúde, que trata da assistência médica hospitalar e ambulatorial, da assistência pré-hospitalar, da assistência farmacêutica e da assistência laboratorial.

O SUS se propõe a promover a saúde, priorizando as ações preventivas, democratizando as informações relevantes para que a população conheça seus direitos e os riscos à sua saúde. Ou seja, além de fornecer assistência médica, deve-se prover mecanismos de coleta, geração e divulgação de informações sobre saúde, buscando estratégias para execução de ações preventivas [MAN01]. A gestão do SUS é realizada através das três esferas de governo: Município, por meio das Secretarias Municipais de Saúde; Estado, pelas

Secretarias Estaduais de Saúde; Federal, pelo Ministério da Saúde em parcerias com as secretarias e agências nacionais de saúde [SEN88b].

A população participa da gestão do Sistema Único de Saúde através dos Conselhos de Saúde, onde os representantes da comunidade podem propor ações, acompanhar e fiscalizar as ações de saúde [SEN90]. O planejamento e avaliação das ações de saúde são elaborados através das Conferências de Saúde, executadas a nível municipal, estadual e nacional. Nas Conferências de Saúde, reúnem-se os representantes da sociedade designados de usuários do SUS, do governo, designado gestores do SUS, dos profissionais de saúde e dos prestadores de serviços, instituições que prestam serviço ao SUS como laboratórios, indústrias farmacêuticas, estabelecimentos de saúde, etc. Com objetivo de prover maior integração entre os gestores do SUS, foram criadas comissões intergestoras, conforme Figura 2.1 [NOB93]:

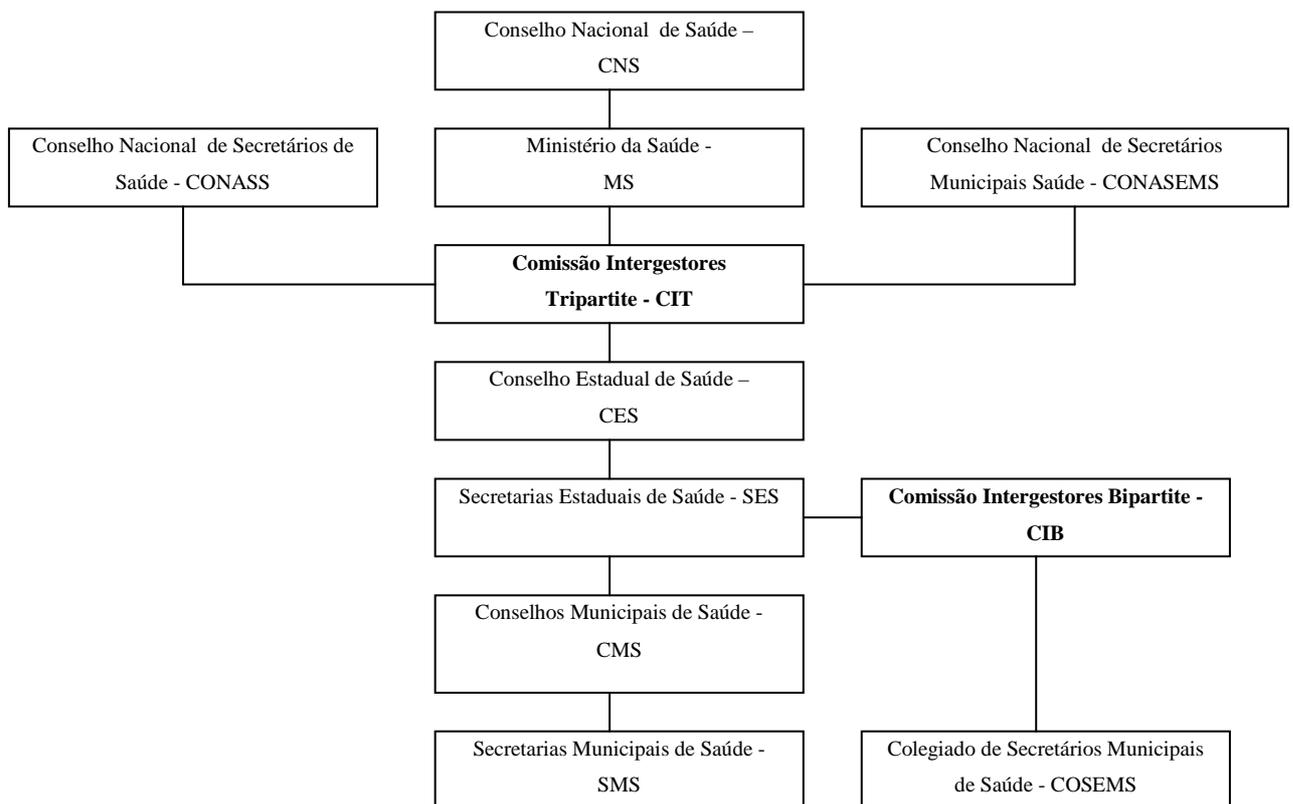


Figura 2.1: Organização dos Gestores do Sistema Único de Saúde.

- Comissão Intergestores Tripartite, formada por representantes dos gestores municipais através do Conselho Nacional de Secretários Municipais de Saúde (CONASEMS), dos

gestores estaduais representado pelo Conselho Nacional de Secretários de Saúde (CONASS) e do gestor federal pelo Ministério da Saúde.

- Comissão Intergestores Bipartite, formada por representantes dos gestores municipais por meio do Colegiado de Secretários Municipais de Saúde (COSEMS) e gestor estadual pela Secretaria Estadual de Saúde (SES).

Estas comissões são responsáveis pelo gerenciamento do Sistema Único de Saúde, constituindo entre outros assuntos, os pactos e programações entre gestores, buscando a integração e descentralização dos serviços entre as esferas de governo [NOB96].

2.2. Descentralização dos Sistemas de Informação do SUS

Desde que foi regulamentado o Sistema Único de Saúde em 1990, criou-se uma estrutura com conhecimento distribuído entre organizações municipais, estaduais e federais. Neste contexto, a NOB 01/96 amparada pela Constituição Brasileira de 1988 Lei 8.080/90, preocupa-se expressamente em estabelecer diretrizes para gestão descentralizada do SUS [NOB96].

A partir deste momento, o sistema de saúde brasileiro iniciou um processo gradual para formalização de procedimentos e competências dos diversos órgãos gestores da Saúde Pública. Sobre este aspecto constata-se uma evolução na descentralização das competências dos gestores do Sistema Único de Saúde [MAN01]. 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 informação para atender objetivos específicos, suprimindo necessidades pontuais de determinadas competências, sem considerar a organização unificada do SUS, ou seja, mesmo o SUS sendo descentralizado, os Sistemas de Informação deve buscar a integração em todos os níveis de competências [LOU03].

Uma das áreas onde a falta de integração dos sistemas é relevante, discutida em outros trabalhos, é a relação entre a área de *Assistência de Saúde* em conjunto com as áreas de *Vigilância Epidemiológica* e *Vigilância Sanitária* ([LOU03], [MAN01]). Estas competências são correlacionadas com objetivo de promoção da saúde, prevenção e controle de doenças, assim como assegurar a qualidade na assistência à saúde da população. Estes objetivos

justificam a busca da *equidade no atendimento*¹ das necessidades de saúde da população, meta estabelecida pelo Ministério da Saúde conforme:

NOAS 01/02: *“Todos os níveis de governo devem avaliar o funcionamento do sistema de saúde, no que diz respeito ao desempenho nos processos de gestão, formas de organização e modelo de atenção, tendo como eixo orientador a promoção da equidade no acesso na alocação dos recursos, e como instrumento básico para o acompanhamento e avaliação dos sistemas de saúde o Relatório de Gestão”* [NOA02].

NOB 01/96 *“A ameaça da ocorrência de gastos exagerados, em decorrência de um processo de incorporação tecnológica acrítico e desregulado, é um risco que pode ser minimizado pela radicalização na reorganização do SUS: um Sistema regido pelo interesse público e balizado, por um lado, pela exigência da universalização e integralidade com equidade e, por outro, pela própria limitação de recursos, que deve ser programaticamente respeitada”* [NOB96].

A liberdade de ação entre os níveis de gestão, propiciada pelas normativas do SUS, projetou como resultado uma fragmentação (ou falta de integração) facilmente percebida nos sistemas de informação legados desenvolvidos pelos gestores do SUS, com objetivos específicos tais como ([DAT04a], [LOU03]):

- SIM - Sistema de Informação sobre Mortalidade;
- SINAN - Sistema de Informação de Agravos de Notificação;
- SINASC - Sistema de Informação sobre Nascidos Vivos;
- SIH - Sistema de Informação Hospitalar do SUS;
- SIA - Sistema de Informação Ambulatorial do SUS;
- SILAB - Sistema de Informação Laboratorial do SUS;

¹ Equidade no atendimento, significa igualdade e equivalência no atendimento das necessidades da população independente das condições sócio-econômicas.

- SIFAB - Sistema de Acompanhamento do Incentivo a Assistência Farmacêutica Básica;
- SIAB – Sistema de Informação de Atenção Básica;
- SISVAN - Sistema de Vigilância Alimentar e Nutricional;
- SICLOM - Sistema de Controle Logístico de Medicamentos;
- SIOPS - Sistema de Informação sobre Orçamentos Público em Saúde;
- API - Avaliação do Programa de Imunização;
- SINITOX - Sistema Nacional de Informação Tóxico – Farmacológicas;

Observa-se nestes sistemas que os processos e serviços são tratados de forma isolada, mesmo sendo relacionados aos mesmos elementos ou entidades como: usuários da saúde, profissionais da saúde, estabelecimentos de saúde, ações de saúde, etc. Os objetivos especificados, nestes sistemas, focalizam determinadas competências isoladamente, e na maioria dos casos os sistemas de informação são estruturados em ambientes heterogêneos com arquiteturas diferentes, dificultando ainda mais integração do conhecimento entre as áreas.

Outro ponto a ser observado, é a descentralização entre os níveis de gestão federal, estadual e municipal. Como as competências de cada nível de gestão são colaborativas, ou seja, buscam atender as metas do Sistema Único de Saúde em conjunto, aumenta a complexidade do sistema. A dificuldade de colaboração entre os gestores é intensificada a medida em que as descentralizações entre os níveis são geograficamente distantes. Este problema é amenizado com a regionalização dos níveis, por outro lado aumentando ainda mais a distribuição do sistema, assim como sua complexidade. Os princípios da regionalização são subsidiados conforme a Lei no 8.080/90 onde estabelece:

Lei no 8.080/90: *“As ações e os serviços de saúde devem ser regionalizados e hierarquizados conforme o nível de complexidade do sistema”*, consoante o disposto no artigo 198 da Constituição Federal de 1988 [SEN88b].

Além da distribuição entre os gestores e suas regionais, o sistema de saúde brasileiro está organizado em uma rede de estabelecimentos públicos e em um conjunto de

estabelecimentos privados complementar, com objetivo de prestar serviços de assistência à saúde da população [MIN04a], como por exemplo, serviços hospitalares, ambulatoriais, laboratoriais, etc. Neste contexto, as ações do SUS são ainda mais distribuídas, pois envolvem um grande número de estabelecimentos de saúde conforme apresentados na Tabela 2.1:

Tabela 2.1: Distribuição de estabelecimentos de saúde segundo o tipo de unidade de saúde [MIN05a].

DESCRIÇÃO	TOTAL
Centro de parto normal – isolado	48
Centro de saúde / unidade básica	25297
Clinica especializada / ambulatório de especialidade	10531
Consultório isolado	23379
Cooperativa	53
Farmácia	150
Hospital especializado	1099
Hospital geral	5053
Hospital/dia - isolado	186
Policlínica	2877
Posto de saúde	11729
Pronto socorro especializado	106
Pronto socorro geral	481
Unidade autorizada	110
Unidade de apoio diagnose e terapia	10068
Unidade de saúde da família – isolado	16
Unidade de vigilância sanitária / epidemiologia – isolado	2225
Unidade mista	920
Unidade móvel de nível pré-hospitalar - urgência / emergência	166
Unidade móvel fluvial	11
Unidade móvel terrestre	736
Total	95241

Estes estabelecimentos de saúde são categorizados conforme a abrangência dos serviços que oferecem, sendo classificados da seguinte forma:

- Serviços de atenção básica de saúde: envolvem os centros de saúde/unidade básica; consultórios isolados; postos de saúde; unidades de saúde da família; unidades móveis fluviais e unidades móveis terrestres;

- Serviços de atenção de média complexidade: envolvem os centros de parto normal; clínicas especializadas/ambulatórios de especialidade; unidades mistas; policlínicas; hospitais-dia; pronto-socorros geral; hospitais geral; unidades móveis de nível pré-hospitalar (urgência e emergência); unidades de apoio de diagnose e terapia.
- Serviços de atenção de alta complexidade: envolvem os hospitais especializados; pronto-socorros especializados; farmácias (medicamentos especiais e excepcionais).

As competências destes serviços são estabelecidas e distribuídas entre as esferas municipal, estadual e federal, conforme as regras definidas atualmente na Norma Operacional da Assistência à Saúde (NOAS 01/02) [NOA02].

Segundo dados extraídos em janeiro de 2005 do Cadastro Nacional de Estabelecimentos de Saúde (CNES), existem aproximadamente 95.284 estabelecimentos de saúde no Brasil, conforme apresentado na Tabela 2.2. A diversidade de ações da Saúde Pública em conjunto com a grade extensão territorial do Brasil potencializa um alto número de estabelecimentos de saúde, compreendidos em maior parte entre estabelecimentos públicos de administração direta ou indireta, e complementares privados ou mistos.

Tabela 2.2: Distribuição de estabelecimentos de saúde segundo a natureza da organização [MIN05a].

Descrição	Total
Administração direta da saúde (MS, SES e SMS)	47.992
Administração direta de outros órgãos.	329
Administração indireta – Autarquias	245
Administração indireta - Empresa Pública	96
Administração indireta - Fundação Pública	332
Administração indireta - Organização Social Pública	63
Cooperativa	133
Economia mista	8
Empresa privada	41.785
Entidade beneficente sem fins lucrativos	2.484
Fundação privada	406

Serviço social autônomo	846
Sindicato	565
Total	95.284

Percebe-se o quanto é distribuído à estrutura do sistema de saúde público do Brasil, não poderia ser de outra forma, a complexidade das ações de saúde sobre uma população estimada em 179.108.134 de pessoas em 2004 e distribuída geograficamente entre estabelecimentos de saúde públicos e complementares, direciona a descentralização das ações de saúde entre os gestores, a fim de diminuir a complexidade e aumentar a eficiência dos serviços ([DAT04b], [MIN04a]).

Sobre esta condição, os sistemas de informação computacionais são aplicados de forma distribuída. Ou seja, eles são utilizados em diversas instituições da esfera municipal, estadual e federal. Porém, sem prover nenhum mecanismo de coordenação entre estes sistemas. A maior parte dos sistemas é utilizada de forma isolada e a única ligação entre eles é a transferências de dados de um nível ao outro por meio de arquivos persistentes (arquivo texto ou dados estruturados) transferidos de forma manual por meio de disquetes, e-mail, etc. A estrutura isolada dos sistemas de informação não possibilita nenhuma interação ou coordenação funcional, não permitem uma gestão das competências de cada esfera do governo, formando uma estrutura de sistema de informação distribuída, porém não integrada.

Os ambientes heterogêneos deixam mais distante a possibilidade de interação funcional entre os sistemas, esta condição é difícil de ser modificada, pois cada nível das esferas do governo tem autonomia sobre sua projeção tecnológica e estratégica de ação ([DAT04a], [NOB96]). Este problema é potencializado entre os estabelecimentos complementares, estabelecimentos privados contratados pelo SUS, pois entre estes estabelecimentos a diversidade dos ambientes é ainda maior [LOU03].

Outro fator relevante é a estrutura multidisciplinar da saúde, sobre uma diversidade de segmentações do conhecimento, as informações ganham maior ou menor relevância dependendo da área de atuação, o que dificulta estabelecer como interação entre os sistemas a simples transferência de arquivos, como é utilizada na maior parte dos sistemas da rede pública, onde a estrutura de dados é rígida para possibilitar a interpretação do dado.

O Sistema Único de Saúde necessita de mecanismos para promover a gestão descentralizada da Saúde Pública. Neste contexto, algumas iniciativas estão sendo abordadas com intuito de prover a integração entre os sistemas de informação da saúde, como por

exemplo, o projeto piloto do Sistema do Cartão Nacional Saúde proposto pelo Ministério da Saúde, almejando maior integração das informações para os gestores do SUS [LEA02]. Este projeto piloto iniciou em 1999 abrangendo 44 municípios, devido a arquitetura de funcionamento ser constituída sobre uma rede privativa e com tecnologias fechadas, estão sendo avaliadas formas alternativas para expansão deste projeto. ([BAR02], [GRE02]).

2.3. Sistemas de Informação Legados do SUS

Foram apresentados anteriormente os sistemas de informação desenvolvidos para suprir as necessidades de gestão e operação do SUS. Entre estes sistemas, serão selecionados três sistemas legados da área epidemiológica e dois sistemas da área de assistência de saúde, os quais compreende maior relevância de dados para gestão do SUS, conforme definido pela Fundação Nacional de Saúde/MS [MIN98].

2.3.1. Área de Vigilância Epidemiológica

Compete a área de vigilância epidemiológica o conjunto de ações que proporciona o conhecimento, a detecção ou prevenção sobre os fatores determinantes ou condicionantes da saúde individual ou coletiva, com a finalidade de recomendar e adotar as medidas de prevenção e controle das doenças ou agravos. Neste segmento, os sistemas SIM, SINASC e SINAN oferecem suporte operacional para geração dos dados epidemiológicos de interesse municipal, estadual ou federal.

Sistema de Informação sobre Mortalidade - SIM

O Sistema de Informações sobre Mortalidade foi criado pelo Ministério da Saúde em 1975 para a obtenção regular de dados sobre mortalidade. Este sistema tem como objetivo captar os registros de óbitos, formando um instrumento de monitoramento para subsidiar as esferas de gestão do sistema (federal, estadual e municipal) para a análise de situação, planejamento e avaliação das ações e programas de saúde [MIN98]. Com base nos dados de mortalidade é possível realizar análises que orientem a adoção de medidas preventivas, assim como a tomada de decisão no processo de gestão do sistema de saúde, possibilitando avaliar

as ações implementadas que tenham impacto sobre as causas de morte. Proporciona a produção de estatísticas de mortalidade e a construção dos principais indicadores de saúde, permitindo estudos não apenas do ponto de vista estatístico e epidemiológico, mas também sócio-demográfico.

O documento oficial para captação de dados sobre mortalidade é a Declaração de Óbito (DO). Este sistema está constituído com amparo legal segundo a Lei nº 6.015, de 31 de dezembro de 1973, lei dos registros públicos artigo 78. O fluxo operacional é definido atualmente conforme os decretos e portarias do Ministério da Saúde [MIN04b].

Sistema de Informação sobre Nascidos Vivos - SINASC

O Sistema de Informação sobre Nascidos Vivos, implantado oficialmente a partir de 1990, foi desenvolvido com o objetivo de coletar dados sobre os nascimentos informados em todo território nacional e o fornecimento de dados sobre natalidade para todos os níveis do sistema de saúde (federal, estadual e municipal) [MIN98]. Este sistema visa obter informações referentes aos nascimentos em face da importância do conhecimento de características do recém-nascido e da mãe, assim como da gravidez e do parto, para as políticas e as ações de saúde [MIN04a].

A declaração do nascimento é obrigatória no serviço de saúde em que ocorreu o parto, tendo como propósito caracterizar as condições de nascimento de acordo com alguns fatores como peso ao nascer, duração da gestação, tipo de parto, idade da mãe e paridade, que, reconhecidamente, influenciam o estado de saúde da criança.

O documento oficial para captação dos dados é a Declaração de Nascido Vivo (DN). Este sistema está constituído com amparo legal segundo a Lei nº 6.015, de 31 de dezembro de 1973, lei dos registros públicos artigos 51 a 67. O fluxo operacional é definido atualmente conforme os decretos e portarias do Ministério da Saúde [MIN04c].

Sistema de Informação de Agravos de Notificação - SINAN

O Sistema de Informação de Agravos de Notificação (SINAN) tem por objetivo o registro e processamento dos dados sobre agravos de notificação em todo o território nacional, fornecendo informações para análise do perfil da morbidade e contribuindo desta forma para a

tomada de decisões ao nível municipal, estadual e federal [MIN98]. Este sistema foi desenvolvido para ser operacionalizado em nível de unidade de saúde. Porém, caso o município não disponha de infra-estrutura nas suas unidades, o mesmo poderá ser operacionalizado a partir das Secretarias Municipais, Regionais e Estaduais de Saúde. Este sistema possibilita uma análise global e integrada dos agravos de saúde, gerando informações nos níveis acima referidos, além de distrito e bairro.

Historicamente, a notificação compulsória tem sido a principal fonte usada pela Vigilância Epidemiológica para desencadear as medidas de controle. O Sistema Nacional de Agravos Notificáveis (SINAN), é o principal instrumento de coleta de dados das doenças de notificação compulsória e outros agravos. Instituído em 1996, seu objetivo é dotar municípios e estados de uma infra-estrutura tecnológica básica para a transferência de informações dentro do Sistema de Informação em Saúde.

2.3.2. Área de Assistência em Saúde

A área de assistência de saúde contempla o controle e gestão dos serviços de atenção a saúde individual curativa e preventiva, trata-se da assistência básica de saúde, da assistência médica hospitalar e ambulatorial, da assistência pré-hospitalar, da assistência farmacêutica, da assistência laboratorial e dos serviços de apoio a diagnose e terapia. Neste contexto, os sistemas SIA e SIH oferecem controle operacional e de gestão aos recursos de assistência de Saúde Pública.

Sistema de Informação Ambulatorial do SUS - SIA

O Sistema de Informações Ambulatoriais do SUS (SIA-SUS) foi formalmente implantado em 1991, sendo o instrumento de ordenação do pagamento dos serviços ambulatoriais públicos e privados conveniados [MIN04e]. Este sistema provê informações aos gestores municipais, estaduais e federais, sobre assistência básica de saúde. O sistema SIA utiliza como instrumento de informação o Boletim de Produção Ambulatorial (BPA) e a Autorização de Procedimento de Alta Complexidade/Custo (APAC) [MIN04e]. Através destes instrumentos são processados e validados os dados com a finalidade de regular a aprovação ou não dos procedimentos informados.

Sistema de Informação Hospitalar do SUS – SIH

O Sistema de Informação Hospitalar do SUS foi concebido com o propósito de operar o sistema de pagamento de internação dos hospitais contratados pelo Ministério da Saúde. Os estabelecimentos hospitalares compreendem hospitais públicos (municipal, estadual ou federal), filantrópicos, universitários e privados conveniados. Seu principal objetivo é propiciar ao gestor municipal, estadual e federal, mecanismos para avaliação e controle de internações hospitalares [MIN04d].

Utiliza como instrumento de informação a Autorização de Internação Hospitalar (AIH), onde são registrados os dados do usuário do SUS, do profissional de saúde e dos procedimentos médicos. O sistema possibilita submeter críticas de consistência das informações, a fim de aprovação ou glosa de Autorização de Internação Hospitalar (AIH) [MIN04d].

2.4. Sistema Cartão Nacional de Saúde

O Sistema Cartão Nacional de Saúde (SCNS) é um sistema informatizado de base nacional, que possibilita correlacionar os procedimentos realizados pelo Sistema Único de Saúde ao usuário do SUS, mantendo o vínculo com o profissional de saúde e o estabelecimento de assistência de saúde [BRA01].

Segundo a NOB de 1996, como “*forma de identificar a clientela do Sistema Único de Saúde, explicitando ao mesmo tempo sua vinculação a um gestor² e a um conjunto de serviços bem definido, cujas atividades devem cobrir, integralmente, todo o escopo de atenção à saúde do cidadão, como estipula a Constituição*”. O Cartão deve ter uma numeração única dentro do território nacional, de modo a identificar o cidadão nos diversos sistemas de informação de saúde [NOB96].

² Na NOB 96, gestão é a atividade e a responsabilidade de dirigir um sistema de saúde (municipal, estadual ou nacional), mediante o exercício de funções de coordenação, articulação, negociação, planejamento, acompanhamento, controle, avaliação e auditoria. São gestores do SUS os secretários municipais e estaduais de saúde e o ministro da saúde.

2.4.1. Objetivos do Sistema

Segundo as Portarias: MS/SE n.º 17 e 57 de 2001, o Sistema do Cartão Nacional de Saúde tem como objetivo a modernização dos instrumentos de gerenciamento da atenção à saúde. A utilização de tecnologias de ampla difusão, conjugando a informática e telecomunicações, permitindo dotar o SUS de uma rede integrada para a realização de uma variada gama de operações e captura de informações. Sua característica fundamental será possibilitar ao SUS a capacidade de identificação individualizada dos usuários. Desta forma, será possível identificar o paciente e toda sua evolução, locais onde consultou, exames que executou, diagnósticos, medicações, etc. Entre os objetivos definidos pelo projeto do SCNS, são contemplados:

- construção de uma base de dados de histórico clínico;
- imediata identificação do usuário, facilitando o atendimento;
- ampliação e melhoria de acesso da população a medicamentos;
- possibilidade de revisão do processo de compra de medicamentos;
- integração de sistemas de informação;
- acompanhamento dos fluxos assistências, ou seja, acompanhamento do processo de referência e contra-referência³ dos pacientes;
- revisão dos critérios de financiamento e racionalização dos custos;
- acompanhamento, controle, avaliação e auditoria do sistema e serviços de saúde;
- gestão e avaliação de recursos humanos.

2.4.2. Arquitetura do Sistema

O SCNS criou a infra-estrutura de telecomunicações para os 44 (quarenta e quatro) municípios do projeto piloto e para os 27 (vinte e sete) estados da federação. Os 2 (dois) equipamentos servidores federais (redundância de site), os 27 (vinte e sete) equipamentos servidores estaduais e os equipamentos servidores concentradores estão conectados via rede

³ Referência e contra-referência é o processo de encaminhamento do paciente para os serviços competentes de saúde, buscando seguir um fluxo operacional (triagem) com objetivo de proporcionar um melhor nível de assistência e de regulação dos recursos [NOB96].

permanente (frame-relay). A rede trabalha com o protocolo TCPIP caracterizando uma Intranet corporativa. A comunicação a partir do local de atendimento até o servidor concentrador é feita através de linha discada através dos Terminais de Atendimento do SUS (TAS), conforme apresentado na Figura 2.2 ([REV02], [LEA01]):

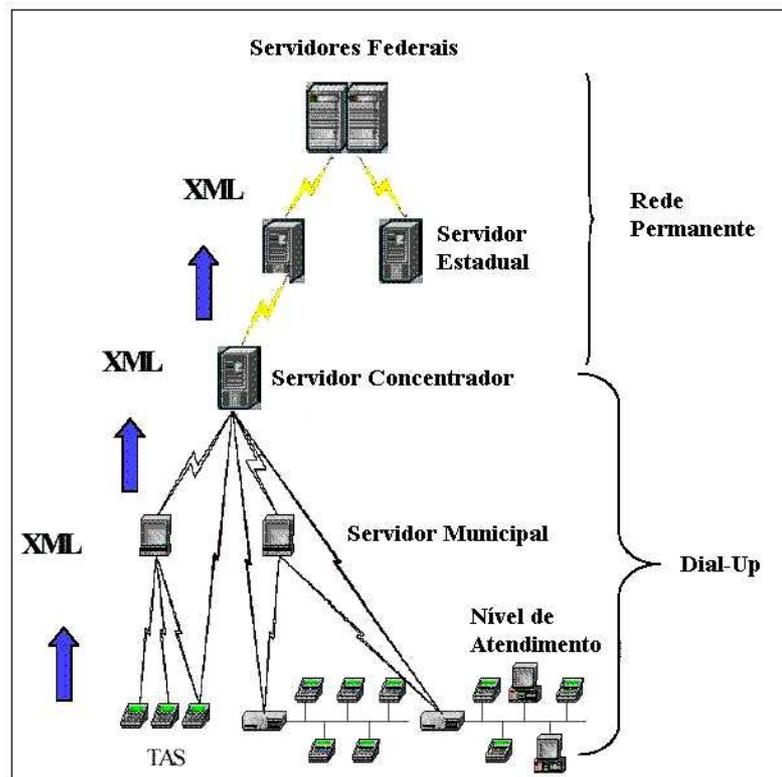


Figura 2.2: Arquitetura do sistema SCNS [LEA02].

- Nível de atendimento: compostos pelos terminais de atendimento instalados as unidades de saúde e conectados através de rede *dial-up* com o nível municipal;
- Nível municipal: compostos pelos servidores municipais responsáveis pelo processamento e armazenamento dos dados de todos os atendimentos realizados no município, conectado através de rede *dial-up* com o nível concentrador;
- Nível concentrador: compostos por servidores conectados on-line com os níveis estadual e federal através de rede *frame-relay*, sendo responsável pelo armazenamento dos dados de atendimentos de todos os municípios a ele

conectados para fins de uma maior segurança dos dados, com possibilidade de acesso dos níveis estadual e federal;

- Nível estadual: compostos por servidores conectados à rede SUS (nível federal) através de rede *frame-relay*, sendo responsável pelo armazenamento das informações totalizadas dos atendimentos realizados no Estado;
- Nível federal: compostos por servidores conectados à rede SUS, sendo responsável pelo armazenamento das informações totalizadas dos atendimentos realizados no País.

2.4.3. Situação atual do Sistema

O projeto piloto do Sistema Cartão Nacional de Saúde encontra-se em processo de implantação desde o ano 2.000. Este projeto piloto abrange 44 (quarenta e quatro) municípios em todo país, dos quais 31 (trinta e um) estão localizados no Estado do Paraná, onde atualmente estão em fase de finalização [LOP04].

Este projeto tem como objetivo dotar os Estados e Municípios de um mecanismo para monitoramento e avaliação dos recursos do SUS, bem como prover informações em Saúde Pública. No entanto, no momento foram concentrados esforços para manter o funcionamento operacional do sistema, com poucos avanços quanto a prover informações para avaliação da Saúde Pública, ou seja, concentrou-se em suprir a parte operacional da assistência de saúde que ainda apresenta dificuldades [LOP04]. Com objetivo de avaliar este projeto no Estado do Paraná, podem-se considerar algumas dificuldades identificadas na avaliação realizada por Lopes em alguns municípios do Paraná:

- Dificuldade de operar o sistema através do Terminal de Atendimento do SUS (TAS);
- Duplicidade de trabalho, mantendo o registro das informações no Terminal de Atendimento do SUS e no prontuário médico do paciente;
- O sistema não apresenta uma interface amigável, aumentando as dificuldades no registro de atendimento;
- O sistema não oferece um prontuário eletrônico para facilitar a assistência de saúde;

- O sistema não está interligado com as centrais de marcação de consultas especializadas e exames, assim como reserva de leitos, causando duplicidade de informações nos sistemas operacionais de controle da assistência de saúde;
- O SCNS não é integrado com os demais sistemas de informações de base nacional do Ministério da Saúde, ou seja, a área operacional da assistência de saúde não é integrada com os sistemas da área epidemiológica, e mesmo com as áreas de controle e avaliação da assistência de saúde apresenta poucos avanços de interoperabilidade.
- O SCNS está estruturado para funcionamento em um terminal do SUS (TAS), não possibilitando utilizar o sistema através de estações de trabalho;
- Devido à arquitetura do sistema estar projetada em uma rede fechada (Intranet), não possibilita comunicar com o sistema utilizando-se de outras redes privadas. Nestes casos, as conexões são realizadas via linha discada (*dial-up*), desperdiçando a infra-estrutura das redes privadas já existentes nos estabelecimentos de assistência de saúde [HEX02].

Entre as atividades desenvolvidas com objetivo de suprir algumas destas dificuldades, podem-se destacar os esforços do projeto para desenvolver um mecanismo de interoperabilidade entre o SCNS com os demais sistemas externos pertencentes aos estabelecimentos de assistência de saúde. Com este intuito foi desenvolvido um aplicativo denominado Agente Cartão. Através deste agente, os estabelecimentos de saúde que possuem sistema de informação poderão se integrar ao sistema do Cartão Nacional de Saúde através de troca de arquivos XML padronizados de acordo com as DTDs (“*Document Type Definitions*”), definidas pelo Ministério da Saúde [LEAO02]. O Agente Cartão que tem como objetivo estabelecer a interoperabilidade entre a rede SCNS, TAS ou Servidores Municipais, e sistemas externos. O sistema externo pode estar sendo executado no âmbito municipal ou local (do estabelecimento assistencial de saúde).

A partir da implementação do Sistema do Cartão Nacional de Saúde, as transformações do modelo e a reorganização dos sistemas e redes de atenção à saúde no Brasil está sendo um foco de estudo aprofundado [REV02]. O projeto piloto do SCNS está sobre avaliação para definir a expansão do projeto para os demais Estados do Brasil [CNS04].

Relacionado a arquitetura do sistema de informação do projeto SCNS, pode-se concluir que ainda existe uma necessidade quanto à integração do SCNS com os demais sistemas da área de controle e avaliação da assistência de saúde (estadual e municipal), assim como com a área epidemiológica e outros sistemas de base nacional, pois até o momento concentrou-se na integração com os sistemas operacionais de atendimento da assistência de saúde. Outro ponto importante está relacionado à assistência de saúde complementar (estabelecimentos particulares conveniados ao SUS), pois neste caso geralmente existem sistemas com legados próprios de cada estabelecimento, estruturados em uma rede privada da qual somente possibilitará a integração com o SCNS através do Agente Cartão. Porém, utilizando acesso via linha discada (*dial-up*) com baixa velocidade, desperdiçando a infraestrutura de rede que possa estar disponível no estabelecimento de saúde.

2.5. Arquitetura de Sistemas de Informação

A complexidade dos sistemas de informação é delineada entorno de soluções de problemas do mundo real, cujos aspectos modelam os requisitos da arquitetura de software, indo além das questões de algoritmos e da estrutura de dados. A arquitetura de software inclui aspectos estruturais como [GAR94]:

- a organização total do sistema e a criação de uma estrutura de controle;
- a existência de protocolos de comunicação, sincronização e acesso aos dados;
- a atribuição de funcionalidades aos elementos do projeto;
- a sua distribuição física;
- a composição dos elementos do projeto;
- o seu escalonamento e desempenho;
- a seleção entre alternativas de projeto.

Segundo HOHMANN, “*uma arquitetura de software é a soma de módulos não triviais, processos e dados, as suas estruturas e relacionamentos, fornecendo uma visão de como estes podem ser estendidos e modificados, conhecendo as tecnologias sobre as quais dependem, tudo isso com o intuito de deduzir as suas exatas capacidades e flexibilidade, possibilitando a formação de um plano para a implementação ou modificação do sistema*” [HOH03].

2.5.1. Princípios de Arquitetura de Software

A escolha de uma arquitetura é um processo cauteloso, porque uma boa arquitetura é um elemento chave para o sucesso do sistema em longo prazo. Entre as características desejadas em uma arquitetura pode-se citar [HOH03]:

- fomentar a longevidade: a maioria das arquiteturas vivem mais que as equipes que a criaram;
- estabilidade: assegura-se que uma mínima quantidade de re-trabalho seja necessária quando as funcionalidades do sistema precisem ser estendidas;
- suportar uma mudança gradativa e natural: atender as mudanças naturais do sistema;
- favorecer a limitação do escopo: influenciando naquilo que deve estar “dentro” e “fora do sistema”.

Sobre estes requisitos, são adicionados princípios de arquiteturas de software condicionando os aspectos do desenho da arquitetura:

- encapsulamento: a arquitetura é organizada ao redor de peças separadas e relativamente independentes que ocultam detalhes internos de implementação, um do outro;
- interfaces: deve-se definir a forma em que os subsistemas interagem;
- acoplamento fraco: refere-se ao grau de interconexões entre as diferentes partes do sistema;
- granularidade apropriada: é determinada pelas tarefas associadas ao componente, esta relacionado ao tamanho do componente;
- coesão alta: descreve as atividades relacionadas dentro de um componente ou de um módulo;
- parametrização: o número de parâmetros deve ser apropriado, assim como os seus tipos, permitindo ao usuário ajustar as operações do sistema;
- deferimento: adiar decisões arquiteturais quando exista algum tipo de incerteza.

2.5.2. Estilos Arquiteturais de Software

Embora não exista uma classificação para arquiteturas de software, podem-se identificar alguns estilos de arquiteturas tais como [GAR94]:

- a) Tubos e Filtros: Cada componente (filtro) possui um conjunto de entradas e de saídas. Cada componente recebe uma série de dados de entrada, processa estes dados, envia os dados resultantes para os tubos de saída. Os conectores são os tubos de dados (pipes). Especializações deste estilo são os pipelines;
- b) Organização Orientada a Objetos e à Abstração de Dados: Este estilo possui como característica forte o fato dos dados e das operações estarem encapsulados em um componente, o qual chama-se de objeto. A comunicação entre componentes ocorre a partir da troca de mensagens entre esses;
- c) Invocações Implícitas, baseadas em eventos: Neste estilo em vez de invocar um procedimento, o componente pode anunciar (broadcast) um ou mais eventos, os quais implicitamente causam a invocação de procedimentos em outros módulos;
- d) Camadas: O estilo camada é organizado de maneira hierárquica. Cada camada tem o objetivo de prover serviços bem definidos para a camada seguinte, a qual funciona como um cliente para a camada anterior. Cada camada pode possuir vários outros componentes. A alteração de uma camada não pode atingir mais de duas outras camadas, visto que uma camada se comunica, no mínimo, com uma camada e, no máximo, com duas. Esta característica torna este estilo forte para reuso;
- e) Repositórios: No estilo repositórios, existem dois elementos distintos: a base central de dados (blackboard) e os componentes que executam operações na referida base, inclusive suas intercomunicações. Os componentes são chamados de fontes de informação (knowledge source). Os estados assumidos pela base central de dados é o principal motivador de disparo de ações nos componentes que executam tarefas. Uma categoria alternativa é quando dado um fluxo de entradas de transações, estas acionam a seleção de processos a executar-se, neste caso o repositório pode ser um banco de dados tradicional;

- f) Interpretadores: Inclui o pseudoprograma que está sendo interpretado e a máquina de interpretação. O pseudoprograma inclui por sua vez o programa e o estado da sua execução que está sendo simulado;
- g) Outras arquiteturas: Entre estas se têm: Processos distribuídos (anel e estrela), organizações de programa principal e sub-rotinas, específicas ao domínio, de transição de estado e de controle de processos.

2.6. Considerações Finais do Capítulo

Neste capítulo foi apresentada a estrutura do sistema de Saúde Pública do Brasil, caracterizando como um sistema descentralizado com informações distribuídas em níveis municipais, estaduais e federais. Observou-se a fragmentação do conhecimento entre as áreas de saúde, identificadas pelos sistemas legados das áreas epidemiológica e de assistência de saúde que será o foco do trabalho. Identificou-se também a necessidade de buscar mecanismos de interoperabilidade entre os sistemas externos (de outras redes ou áreas de serviços) com o SCNS com comunicação não restrita somente aos dados (XML/DTD), como no caso do Agente Cartão.

Considera-se o interesse de promover a integração dos sistemas legados do SUS por meio de uma arquitetura de sistema de agentes, sendo necessário definir um protocolo de interação suportado através de uma linguagem de comunicação para interoperabilidade entre os sistemas de base nacional (área epidemiológica e de assistência de saúde), e principalmente proporcionar uma arquitetura aberta entre os sistemas para compartilhar os serviços, inclusive relacionado ao SCNS, assim como com assistência complementar (estabelecimentos privados conveniados ao SUS), potencializando interoperabilidade com os sistemas de informações do SUS.

Capítulo 3 - Agentes de Software

Neste capítulo serão apresentados conceitos relacionados a agentes de software, descrevendo as propriedades e características relacionadas aos agentes. Serão apresentados modelos de arquiteturas de agentes, assim como aspectos de linguagem de comunicação e protocolos de interação, os quais possibilitam a comunicação entre os agentes de software. Também será abordada a representação do conhecimento através de linguagens de conteúdo que propiciam a troca do conhecimento entre os agentes, proporcionando mecanismos de interoperabilidade entre os sistemas de informações.

3.1. Conceitos de Agentes de Software

A construção de software esta diretamente relacionada à arquitetura concebida para atender os requisitos de sistema, ou seja, as características de um produto de software são definidas conforme a arquitetura prevista para o sistema. A evolução dos sistemas trouxe para o meio computacional a aplicação de arquiteturas concebidas com propósitos de prover interoperabilidade entre os sistemas (troca de informações e serviços entre softwares) [GEN94]. Sobre este aspecto, a concepção de arquitetura orientada a agente de software vêm fornecendo mecanismos que proporcionam esta interoperabilidade.

A abordagem de agentes tem sido desenvolvida com certa ênfase no campo da Inteligência Artificial e recentemente na Engenharia de Software. No entanto, não há uma única definição para o conceito de agente. Na literatura encontram-se conceitos distintos dependendo do foco do autor, pois cada autor define sob sua própria ótica e, portanto, de forma diferente. Contudo, tais definições não se anulam, elas colaboram para um entendimento mais amplo da concepção de um agente.

Segundo Wooldridge & Jennings [WOO95], “um agente é um sistema de computação encapsulado situado em um dado ambiente e capaz de executar ações autônomas e flexíveis para alcançar os objetivos de projeto”. Um agente é um software persistente que permite a interoperabilidade entre sistemas computacionais distintos, tipicamente possuem capacidade adaptativa entre os ambientes heterogêneos [GEN94].

O interesse sobre agentes de software é focalizado pela comunidade de inteligência artificial observando o comportamento cognitivo, ou seja, capacidade de inteligência e raciocínio do agente. Neste sentido, algumas definições são apresentadas:

"É uma entidade física ou virtual que é capaz de agir sobre ela mesma e sobre seu ambiente, que é capaz de perceber seu ambiente, que dispõe apenas de uma representação parcial desse ambiente (e às vezes nenhuma), que pode comunicar com outros agentes, que persegue um objetivo individual, que possui competências e pode eventualmente se reproduzir, e cujo comportamento é a consequência de seus objetivos, de sua percepção, de suas representações e de suas competências e comunicações que ela pode ter com os outros agentes" [FER94].

"Um agente é uma entidade autônoma (software) que tem conhecimentos especializados, possui uma representação desses conhecimentos, possui uma representação do problema a tratar, possui uma representação parcial do ambiente, possui protocolos de comunicação especializados e às vezes um modelo de suas intenções" [SCA96].

"Um agente é uma entidade inteligente, que age de maneira racional e intencional com relação a seus objetivos e o estado corrente de seus conhecimentos" [DEM91].

Sobre outro aspecto, a *engenharia de software* focaliza a concepção de sistemas computacionais com informação/conhecimento distribuídos:

"Um agente interface é um software que emprega técnicas de inteligência artificial a fim de fornecer uma assistência a um usuário negociando com uma aplicação particular" [MAE94a, MAE94b].

"Um agente informação é uma entidade que tem acesso pelo menos a uma, e potencialmente a várias fontes de informações, que é capaz de coletar e manipular as informações obtidas nestas fontes, que responde solicitações feitas por usuários e/ou por outros agentes de informação" [WOO95].

"Um agente serviço é um agente simples com conhecimentos e capacidades simples de resolução de problemas e cuja tarefa principal é a filtragem da informação para reduzir a sobrecarga cognitiva do usuário" ([MAE94a], [MIT94]).

3.2. Propriedades de Agentes

Um agente pode ser uma pessoa, uma máquina, um pedaço de software, ou uma variedade de outras coisas. A definição básica de agente no dicionário é *"aquele que age"*. No entanto, para o meio computacional, esta definição é muito generalizada: agentes relacionados à tecnologia de informação precisam de propriedades adicionais. Um agente de software possui algumas propriedades que caracterizam suas capacidades, diferenciando um agente de outro conforme as habilidades que possui. Estas propriedades são apresentadas por alguns autores, constituindo um maior ou menor nível de complexidade dos agentes de software:

- autonomia: o agente opera sem intervenção direta do ser humano ou de uma outra entidade externa, e possui certos tipos de controle sobre suas ações e seu estado interno ([WOO94], [ODE00]);
- reatividade: o agente percebe seu ambiente e responde de maneira oportuna as mudanças em seu ambiente [WOO94];
- comportamento social (ou interatividade): o agente interage com o ambiente e com outros agentes ou pessoas por meio de uma linguagem de comunicação, buscando atingir seus objetivos ou colaborar com os demais agentes ([WOO94], [RUS95]).
- pró-atividade: o agente age em resposta a eventos externos, porém é capaz de exibir um comportamento direcionado por objetivos tomando a iniciativa [WOO94].
- mobilidade: habilidade do agente se transportar para outro ambiente onde será executado, sem perder seu estado mental [RUS95].
- adaptável: (ou aprendizado): habilidade do agente acumular conhecimento baseado nas experiências passadas e conseqüentemente modificar seu comportamento em resposta as mudanças no ambiente [WOO94].

- inteligência: a inteligência de um agente é a quantidade de conhecimento aprendido (crenças, objetivos, planos, deduções) e a possível capacidade de raciocínio que o agente possa derivar deste conhecimento ([WOO94], [RUS95]).

As habilidades ou características dos agentes aqui apresentadas não são restrições para classificar um software agente ou não agente, pois dependendo da aplicação apenas algumas destas habilidades estarão presentes. Segundo a FIPA, algumas propriedades de agentes de software são indicadas para justificar a aplicação de sistemas baseados em agentes de software, tais como ([ODE00], [FIP03b]):

Agentes autônomos: quando um agente tem certa independência de controle externo, ele é considerado autônomo. Um agente pode ser autônomo até certo nível, onde consegue interagir sem invocação ou intervenção externa direta, ou seja, possui uma característica dinâmica exercendo certo grau de atividade sobre ações passivas até inteiramente pró-ativas. Desta forma, além de reagirem a solicitações, também deve ativar ações observando eventos que ocorrem em seu ambiente.

Agentes interativos: o agente deve possuir formas de interação com o ambiente e com outros agentes ou pessoas. Esta interação é propiciada por meio de uma linguagem de comunicação de agente, caracterizando um comportamento social do agente. Evidenciando a necessidade de coordenação através de mecanismos cooperativos e/ou competitivos, tais como: negociações e planejamentos.

Agentes adaptáveis: um agente é considerado adaptável se for capaz de responder a outros agentes e ao seu ambiente, até certo nível. Isto quer dizer, no mínimo, que um agente deve ser capaz de responder a um estímulo simples, dando uma resposta direta, a um evento específico ou um sinal do ambiente. Formas mais avançadas de adaptação incluem a capacidade de raciocinar, aprender e desenvolver. Estes agentes podem mudar o seu comportamento baseado em suas experiências, esta aprendizagem é propiciada por técnicas de Inteligência Artificial como: redes neurais, redes Bayesianas, raciocínio baseado em casos, etc.

3.3. Sistemas Multi-Agente

O segmento de sistemas multi-agente pertence ao domínio da inteligência artificial distribuída ([BON88], [BON92]). Tem como objetivo encontrar uma solução para problemas globais baseado-se em um conjunto de entidades distribuídas que trabalham em cooperação ([ERC91], [FER95]), mas cujo comportamento destas entidades pode ser bastante simples. Um sistema multi-agente pode ser definido em função da autonomia de cada agente e dos meios que os mesmos dispõem para gerar suas interações, por exemplo, na auto atribuição de subproblemas através de procedimentos mais ou menos sofisticados de negociação e coordenação.

De modo geral, os agentes buscam atingir seus objetivos individuais e a interação entre eles passa a existir apenas quando surgem conflitos, que podem ocorrer devido à falta de um determinado recurso local ou simplesmente da necessidade de articulação de ações individuais. A ausência de um controle global e de dados globalmente acessíveis e coerentes é compensada por mecanismos locais de coordenação definidos sobre o modelo dos outros agentes, de seus objetivos, de suas intenções e sobre os procedimentos de cooperação [SCA96].

Uma das principais razões para concepção de sistemas multi-agente refere-se aos sistemas baseados em conhecimentos que se tornaram bastante complexos (em tamanho e em número) e difíceis de manter por causa da localização do especialista e da *expertise*. Ferber apresenta algumas razões para a distribuição da inteligência de um sistema, que são [FER95]:

- os problemas complexos são efetivamente na maior parte do tempo fisicamente distribuídos (e.g., refinaria, linha de montagem, transporte de energia);
- os problemas são funcionalmente bem distribuídos e heterogêneos (e.g., a concepção de um carro envolve especialistas em: motor, chassi, pneumático, eletricista);
- as redes de computadores impõem uma visão distribuída (e.g., Internet, sistemas abertos “à la Hewitt”);
- a complexidade dos problemas impõe uma visão local; 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 as 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 envolvendo um raciocínio local e permitindo 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 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 interagentes.

A FIPA apresenta outras razões como [ODE00]:

- poderia ser construído um agente que faz tudo sozinho, mas agentes robustos tem baixa performance e difícil manutenção. Dividir as funcionalidades entre agentes competentes proporciona modularidade, flexibilidade, adaptabilidade e extensão;
- geralmente conhecimentos específicos não estão à disposição a partir de um único agente. O conhecimento que está espalhado em diversas fontes (agentes) poderá ser integrado para uma visão mais ampla quando necessário;
- aplicativos que exigem computação distribuída são suportados mais facilmente por sistemas multi-agente. Desta forma, os agentes podem ser elaborados como componentes autônomos refinados que agem paralelamente para resolução de problemas;

Para sustentar sistemas multi-agente, deve ser estabelecido um ambiente adequado caracterizado da seguinte forma:

- deve prover uma infra-estrutura específica em linguagem de comunicação e protocolos de interação;
- a arquitetura do sistema é tipicamente classificada como sistema aberto, estruturado de forma não centralizada;

- deve possuir agentes que sejam autônomos, adaptáveis e coordenadores.

Contudo, ambientes de agentes solitários são muito mais simples, pois a arquitetura do agente não precisa lidar com questões como cooperação, negociação, e assim por diante.

3.3.1. Sistemas Reativos e Cognitivos

Segundo Scalabrin [SCA96], os sistemas multi-agente são divididos basicamente em duas classes de arquiteturas:

- sistemas reativos; e
- sistemas cognitivos.

Os sistemas de *agentes de reativos* são constituídos por um grande número de agentes, bastante simples, sem inteligência e sem representação de seu ambiente. Na maioria dos casos, estes agentes são fortemente acoplados. Eles interagem utilizando um comportamento do tipo estímulo/resposta [FER92]. Um comportamento inteligente emerge a partir das interações entre estes agentes e seu ambiente [MIN86]. Ou seja, os agentes não são individualmente inteligentes, mas seu comportamento global é inteligente. Existem alguns trabalhos que abordam agentes reativos como:

- Brooks considera a concepção de um robô inteligente e autônomo dividido em camadas que agem como pequenos robôs reativos [BRO86];
- Ferber & Jacopin [FER90] e Ferber & Drogoul [FER92], utilizam a eco-resolução, uma abordagem particular de resolução distribuída de problemas, onde a solução do problema emerge das interações entre os agentes; e
- Steels tentou resolver o problema da coleta de minério por robôs em ambiente desconhecido [STE89].

Os sistemas de *agentes cognitivos* são geralmente constituídos de um pequeno número. Estes agentes são *inteligentes*, possuem uma representação parcial e explícita de seu ambiente. Eles podem representar, por exemplo, um pequeno grupo de especialistas. Esses agentes possuem uma capacidade local de decisão, podem negociar uma informação ou um serviço. Eles são em geral dotados de conhecimentos, competências, intenções e planos, o que

os permitem coordenar suas ações na resolução de um problema. Em agentes cognitivos, busca-se representar o modelo de si (do próprio agente), o modelo dos outros agentes, os atos da fala (comunicação) e mecanismos de cooperação. Contemplando neste modelo, as intenções, as crenças (conhecimento), os objetivos ou planos do agente [SCA96].

Alguns trabalhos preocupam-se em modelar o conhecimento considerando o domínio de aplicação (a contratação de um serviço, a alocação de tarefas, etc.) assim como estabelecer mecanismos de cooperação, como por exemplo:

- Smith & Davis modelam as competências dos outros agentes de um ambiente [SMI88];
- Durfee representa os recursos que dispõem um agente e suas necessidades [DUR87];
- Cohen & Perrault modela o conhecimento das responsabilidades dos outros agentes, os conhecimentos para comunicar (linguagem e protocolo), os conhecimentos sobre as crenças [COH79];
- Georgeff representa as ações do agente [GEO83];
- Cammarata e Steeb modelam os objetivos e planos ([CAM83], [STE81]); e
- Davis e Smith propõem um protocolo de alto nível para negociação de contrato [DAV83].

3.4. Ambiente do Sistema

Em sistemas multi-agente os agentes estão imersos em um ambiente onde possam interagir com outros agentes ou mesmo com o próprio ambiente. Este ambiente pode ser considerado físico ou virtual representando uma organização. Na prática o ambiente é inserido pelo projetista ao mesmo tempo em que os agentes [HAY90]. Este ambiente é uma entidade estrutural que pode resultar de processos dinâmicos. Guessoum concebe o ambiente como sendo um agente ativo e autônomo [GUE96]. Os agentes percebem e agem sobre o ambiente através das perturbações que o ambiente provoca sobre eles. Contudo, os agentes devem possuir capacidade social sobre o ambiente para representar determinada organização.

3.5. Organizações de Agentes

Um agente é uma entidade social que está inserida em uma organização. Neste sentido, a noção de grupo e de interação entre os elementos de um grupo é fundamental para obter a solução de um problema. Os agentes se organizam a fim de aumentar sua eficiência. Eles negociam para resolver conflitos. A *inteligência artificial distribuída* propõe assim organizar os agentes em uma sociedade flexível e eficaz [SCA96].

A existência de uma sociedade de agentes cognitivos implica necessariamente na existência de uma linguagem para comunicações explícitas, ou seja, uma linguagem declarativa e expressiva suficiente para compartilhar o conhecimento dos agentes ([GRU91], [GEN92], [SCA96]).

3.6. Modelo de Agentes

Existe uma diversidade de propostas de modelos de agentes sendo aplicadas em problemas específicos. Estas propostas divergem quanto à forma de representação dos elementos que constituem um agente, principalmente dependendo da modalidade do agente (agente autônomo, agente móvel, agente cognitivo, agente de informação, agente de coordenação/negociação, etc...), ou seja, dependendo da abordagem do agente, os mecanismos que compõem a arquitetura do agente pode constituir maior ou menor grau de complexidade do modelo do agente.

Na Figura 3.1, são apresentadas as principais propriedades que devem existir em um agente, tais como: a representação do próprio agente, com seus recursos, objetivos e competências, a representação dos outros agentes, e por fim, a representação do ambiente onde estão localizados. As propriedades de “percepção e ação” são facilmente identificadas em sistema que possuem interação com meio físico, como na área industrial. No entanto, em sistemas de informação estas propriedades são referenciadas como “atividades de interação ou comunicação” entre os agentes da rede. Estas propriedades almejadas em agentes de software são abstraídas em arquiteturas propostas por alguns autores ([FER95], [SCA96], [JEN92]). As funcionalidades do agente são organizadas internamente através de camadas ou módulos com funções específicas [FAL98].



Figura 3.1: Representação conceitual de um agente [FER95].

Na Figura 3.2 é apresentado um modelo de agente denominado *GAg* (*Generic Agent*) proposto por Scalabrin [SCA96]. Este modelo de agente foi construído para ser o mais genérico possível, abstraindo um conjunto mínimo de características independentes do domínio de aplicação, com propósito de realizar algumas operações elementares. Este arquétipo permite que um agente se construa, de maneira incremental, uma visão de seu mundo externo, podendo evoluir através de mecanismos de raciocínio interno. Esta arquitetura possui mecanismos que possibilita acondicionar novos conhecimentos e competências ao agente conforme o domínio da aplicação.

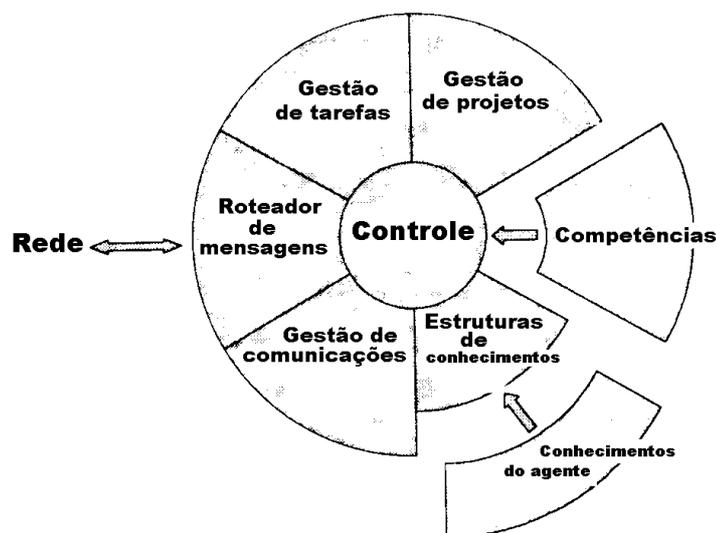


Figura 3.2: Modelo do agente *GAg* (*Generic Agent*) [SCA96].

Os módulos que compõem este modelo compreendem as seguintes funções:

- roteador de mensagens: controle sobre os mecanismos de interação entre os agentes da rede, possibilita a localização dos agentes na rede;
- gestão de comunicações: definição de mensagens aceita e competências para receber, enviar e tratar mensagens;
- gestão de tarefas: processo de execução das tarefas relacionado às competências que o agente possui;
- gestão de projetos: mecanismos que permitem distinguir as diferentes tarefas que os agentes podem estar envolvidos com propósito de resolver um problema particular;
- estruturas de conhecimento e de competências: compreende o conhecimento obtido pelo agente e as capacidades que o agente dispõem, sendo diretamente relacionado ao domínio de aplicação;
- controle: núcleo de processamento do agente onde são relacionados os demais módulos, ou seja, controle desde o roteamento da mensagem e comunicação até o processo de execução de tarefas para satisfazer um objetivo definido em projeto, utilizando-se das competências e conhecimentos do agente.

Um outro modelo de agente é proposto por Jennings e Wittig denominado *ARCHON* [JEN92]. Este modelo de agente separa claramente os módulos que compõem o agente (bloco superior) do módulo da aplicação de determinado domínio (bloco inferior), conforme apresentado na Figura 3.3:

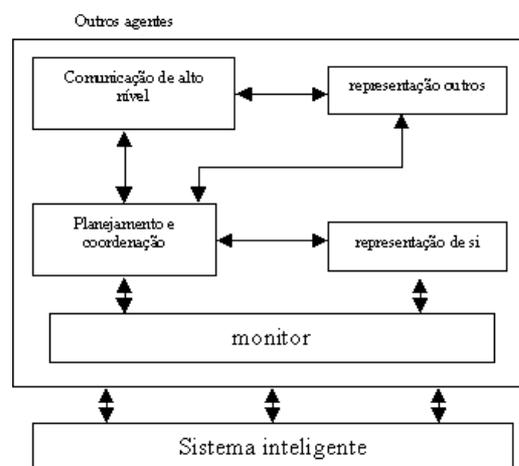


Figura 3.3: Modelo de agente ARCHON [JEN92]

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 (facilidades) de difusão de informações utilizam esses modelos para encontrar os agentes interessados por um determinado resultado, ou mesmo para identificar as competências de outros agentes. Da mesma maneira, o **modelo de si** representa de modo abstrato o domínio do agente ou da aplicação [JEN92].

A **aplicação do domínio** 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**, o qual é responsável do controle da aplicação e das trocas de informações entre as partes [JEN92].

O subsistema de **coordenação e planejamento** 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; devemos 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 [JEN92].

Nesta arquitetura, o agente representa um sistema de serviços com mecanismos que não fazem parte da aplicação propriamente dita. Fornece serviços que suportam a comunicação/interação, representação do conhecimento do agente, planejamento e execução de tarefas, tais como:

- o endereçamento de mensagens;
- a filtragem;
- a difusão de informações dirigida por interesses (*Publish/Subscribe*);
- a representação de si;
- a representação dos outros ("acquaintances"); e
- a interação por meio de protocolo cliente-servidor ou redes contratuais (*Contract-Net*) [SMI80].

Existe diferentes modelos de agentes propostos por outros autores como Hayes-Roth (1990), Bussmann e Demazeau (1994), Hägg & Ygge (1994) [SCA96]. Neste trabalho, será aplicada uma arquitetura de agente utilizando como base os modelos Gag e ARCHON apresentados.

3.7. Linguagem de Comunicação de Agentes

A comunicação entre agentes de software é essencial em sistemas computacionais que utilizam agentes para controle organizacional ou mesmo para suporte operacional de informação como neste trabalho. Em sistemas multi-agente a comunicação é um requisito para proporcionar a cooperação entre os agentes de uma rede. O papel fundamental da comunicação é oferecer um meio de troca de informações entre os agentes: planos, resultados parciais, objetivos, etc.. Porque, sem comunicação, é impossível que os agentes possam manter seus conhecimentos de forma coerente e consistente [SCA96].

O objetivo da linguagem de comunicação de agentes é prover mecanismos para interação entre os agentes de software. Sobre este aspecto, a característica relevante da linguagem usada por agentes é sua expressividade, permitindo a troca de dados e informação lógica, comandos individuais ou scripts (e.g. programas).

A linguagem de comunicação é o meio pelo qual um agente comunica sua intenção ou atitude sobre um conteúdo que está sendo comunicado, ou seja, não busca representar o conteúdo, e sim apresentar o conteúdo ao agente receptor, indicando através de performativas a finalidade da conversação. Por exemplo, a linguagem de comunicação pode “dizer” se o conteúdo da comunicação é uma asserção ou uma requisição [SCA96].

A linguagem de comunicação deve possuir uma semântica comum independente do agente para possibilitar a interação entre os agentes de software, capacitando ao agente um *comportamento social*. No entanto, a comunicação entre os agentes ainda é um problema devido à ausência de normas neste domínio [SCA96].

Para possibilitar o estudo da comunicação entre agentes de software, serão considerados três níveis que estruturam a comunicação conforme apresentado na Tabela 3.1 Em relação ao modelo OSI (*Open Systems Interconnection*), os dois níveis mais altos se situam na camada de aplicação.

Tabela 3.1: Representação dos Níveis da Linguagem de Comunicação de Agentes.

Protocolo de interação	Nível 3: 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, assim como definir os mecanismos de acesso aos serviços do agente.
Linguagem de comunicação	Nível 2: é o meio pelo qual um agente comunica sua atitude ou intenção sobre o conteúdo que está sendo comunicado na mensagem, por exemplo: a linguagem de comunicação que "diz" se o conteúdo da comunicação é uma asserção ou uma requisição.
Protocolo de transporte	Nível 1: mecanismo real de transporte utilizado para a comunicação, por exemplo: TCP, UDP, HTTP, SMTP, etc...

Existem algumas linguagens de comunicação sendo propostas por instituições de interesse sobre agentes de software, tais como:

- KQML (*Knowledge Query for Manipulation Language*), esta linguagem é proposta pelo grupo de trabalho *External Interfaces Working Group* do *DARPA Knowledge Sharing Initiative*. Esta linguagem define um conjunto de diretivas de comunicação (performativas) para interação entre agentes de software. Possui uma estrutura onde são representados os parâmetros das mensagens, possibilitando a troca de informação entre agentes de forma declarativa. A linguagem KQML é independente do conteúdo transmitido, no entanto, é comum a utilização da linguagem de conteúdo KIF (*Knowledge Interchange Format*) em sistemas que aplicam a linguagem de comunicação KQML. A linguagem KIF também é proposta pelo mesmo grupo de trabalho ([MAY95], [GEN94]).
- FIPA ACL (*FIPA Agent Communication Language*), esta linguagem é proposta pela organização FIPA (*Foundation for Intelligent Physical Agents*), possui performativas semelhantes à linguagem KQML. Porém, restrita a um conjunto menor de performativas [FIP02a]. A estrutura da mensagem FIPA ACL

compreende os mesmo parâmetros da mensagem KQML, com exceção de *reply-to*, *encoding* e *reply-by* [MCG93b].

- IACL é a linguagem de comunicação utilizada na arquitetura OSACA (*Open System for Asynchronous Cognitive Agents*) [SCA96]. Esta linguagem compreende um conjunto de tipos de mensagens (e.g., bid, execute, result, inform, etc.), onde cada mensagem é um objeto. Este objeto define um conjunto de ações que determina o comportamento do agente perante uma mensagem comunicada (e.g. o que fazer se não obter uma resposta para um questionamento antes de um "timeout"). As performativas definidas pela IACL são adequadas apenas à implementação de protocolos de mais alto nível, tais como as redes contratuais [SCA96].
- ICL (*Interagent Communication Language*), esta linguagem é proposta pela organização AIC (*Artificial Intelligence Center*), compreende performativas de comunicação, assim como mecanismos de coordenação de tarefas. A linguagem ICL é utilizada em uma arquitetura aberta denominada OAA (*Open Agent Architecture*), possibilita a troca de informações, consultas, solicitação de serviços, execução de ações, configuração de procedimentos e manipulação e compartilhamento de dados ([CHE01], [MAR99]). Esta arquitetura também utiliza uma linguagem de conteúdo desenvolvida em linguagem de programação PROLOG, semelhante à linguagem KIF.
- Existem outras linguagens como Agent-0, Agent-K e COOL que buscam aprimorar os recursos da linguagem KQML [SCA96].

3.7.1. Linguagem de comunicação KQML

Na arquitetura proposta neste trabalho, com objetivo de integração do dos sistemas legados do SUS, será abordada a aplicação da linguagem KQML devido à completude das primitivas de comunicação fornecidas pela linguagem, assim como se mostrou suficiente em projetos como OSACA, SHADE, SAAP, ABSE entre outros ([SCA96], [MCG93a], [MAC05], [MCG93b]).

A linguagem de comunicação KQML (Knowledge Query for Manipulation Language) é 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 KQML é composta de parâmetros classificados em três níveis conforme Tabela 3.2 ([FIP02b], [MCG93b]):

Tabela 3.2: Estrutura de parâmetros de mensagem KQML

Parâmetro	Descrição do parâmetro	Categoria de parâmetro
Mensagem		
performativa	Performativa da Linguagem de Comunicação onde são encapsulados os parâmetros de comunicação e de conteúdo da mensagem.	Diretiva de comunicação
Comunicação		
sender	Agente de origem que está enviando a mensagem.	Participantes da comunicação
receiver	Agente de destino que receberá a mensagem.	
reply-to	Agente que aguarda a resposta, para o qual deve ser respondido (proposta pela FIPA ACL).	
protocol	Protocolo de interação que é suportado pelos serviços do agente.	Controle da conversação
conversation-id	Identificação da seqüência da conversação.	
reply-with	Identificação da solicitação para a qual a resposta estará relacionada: “Responder para a solicitação n”.	
in-reply-to	Identificação da solicitação que está sendo respondida: “Em resposta a solicitação n”.	
reply-by	Tempo esperado para resposta (proposta pela FIPA ACL).	
Conteúdo		
content	Conteúdo comunicado na mensagem, descreve a informação comunicada.	Conteúdo da informação
language	Linguagem da descrição do conteúdo, a qual define a sintaxe e semântica para interpretar o conteúdo.	Descrição do conteúdo
encoding	Codificação do conteúdo da mensagem (proposta pela FIPA ACL).	
ontology	Ontologia que define os elementos comunicados no conteúdo.	

O **nível conteúdo** diz respeito ao conteúdo da informação real de uma mensagem. Este conteúdo é expresso em alguma linguagem de representação (e.g., KIF, FIPA SL, RDF), utilizando os termos de um domínio definido sob a forma de uma ontologia ([GEN92], [FIP02d], [BER01]).

O **nível comunicação** codifica o conjunto de características da mensagem e descreve os parâmetros de mais baixo nível, tais como: a identificação do emissor e do receptor, controle de seqüência da conversação, tempo esperado de resposta, definição de protocolo de interação. Trata-se de identificadores únicos associados à mensagem que permitem o controle da comunicação ([FIP02b], [MCG93b]).

O **nível mensagem** codifica a mensagem que um agente deseja transmitir. Define a diretiva/performativa de comunicação, ou seja, determina o tipo de interação que um agente pode ter com seu correspondente, independentemente da sintaxe e do contexto ontológico [MCG93b]. A função principal desse nível é identificar o protocolo a ser utilizado para expedir uma mensagem e fornecer uma performativa (e.g., *ask-one*, *tell*, *ask-about*), que será anexada ao conteúdo ([FIP02a], [FIP02c]). 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 [GEN94]. Na

Figura 3.4 é apresentado um exemplo de mensagem KQML sendo comunicada, a performativa utilizada neste exemplo é “ask-if” que tem como intenção um questionamento a respeito do conteúdo informado, neste caso o conteúdo solicitado é a informação relacionada ao usuário do SUS cujo Cartão Nacional de Saúde corresponde ao número “2222222222222224”.

```
ask-if :language XML :ontology SUS-CartaoNacionalSaude
      :reply-with q1 :content (request(<usuario cns=2222222222222224></usuario>))
```

Figura 3.4: Exemplo de Mensagem KQML

A linguagem KQML não possui o nível de protocolo de interação que permite gerenciar a cooperação entre agentes (planejamento/negociação), ele apenas fornece um conjunto de primitivas para prover a comunicação. Existem alguns protocolos de interação

propostos com objetivo de suprir esta necessidade. Porém, não são robustos suficientes para atender todos domínios particulares de aplicação. Sendo assim, é comum a elaboração de novos protocolos de interação para determinados domínios.

3.8. Protocolo de Interação

O protocolo de interação é transmitido no conteúdo da mensagem com objetivo de definir uma estratégia de comunicação de alto nível, utilizada por um agente a fim de gerir suas interações com os outros de forma colaborativa. Um protocolo de interação pode abrigar um esquema de negociação, assim como definir os mecanismos de acesso aos serviços do agente. Sobre o aspecto de negociação em sistemas multi-agente, este protocolo pode ser conjugado de forma centralizada ou distribuída.

A **negociação centralizada** pressupõe a existência de uma visão global do plano. Um agente central raciocina sobre o conjunto de ações dos diferentes agentes, resolvendo os conflitos entre os agentes e estabelecendo um plano de ação para cada agente de forma mandatária [CAM83].

Na **negociação distribuída**, tal agente não existe. Cada agente cria seu próprio plano. Os agentes trocam seus planos parciais para detectar e evitar conflitos. Como exemplo de negociação distribuída, podem-se citar os trabalhos *Contract-Net* [DAV83], alocação dinâmica de tarefas [SCA96], a resolução de conflitos de recursos e de objetivos [CON88], problema de transporte de cargas [SAN93], a negociação da compra/venda de produtos ou serviços ([SHM97], [SHM99]).

Deve-se salientar, para que os agentes KQML cooperem, existe a necessidade de requerer os serviços de um agente especializado, que funciona como um mediador/facilitador entre os agentes clientes e os agentes provedores de serviços. Pode-se citar, como exemplo de *facilitador*, o sistema de serviço *Matchmaker* desenvolvido por Kuokka & Harada [KUO95b], no contexto do projeto SHADE. No entanto, esta funcionalidade pode ser encapsulada em cada agente com objetivo de manter uma base com as competências dos demais agentes da rede, ficando desta forma independente do agente *facilitador*, ou seja, não centralizando a comunicação em um único ponto, em contrapartida, acaba aumentando o tráfego na rede.

A primeira abordagem, utilizando um *facilitador*, ainda continua sendo mais utilizada (e.g., OMG CORBA [OMG99]). Existem alguns trabalhos que procuram acrescentar ao

KQML mecanismos de coordenação através de protocolos de interação, tais como: Agent-K [DAV94], COOL [BAR95] e protocolo de negociação ([SHM97], [SHM99]).

3.8.1. Protocolo de Interação: FIPA-Request Protocol e FIPA-Cancel Meta Protocol

Na arquitetura proposta para integração dos sistemas legados do SUS, serão utilizados os protocolos *FIPA-Request Protocol* e *FIPA-Cancel Meta Protocol* definidos pela FIPA (*Foundation for Intelligent Physical Agents*) [FIP02C]. Sendo derivados destes modelos de protocolos (*template*) três novos protocolos de interação: *Assistência-Ambulatorial*, *Assistência-Hospitalar* e *Vigilância-Epidemiológica*.

O protocolo *FIPA-Request Protocol* define seis interações disponíveis para coordenação e execução de atividades ou serviços dos agentes, tais como apresentado na Figura 3.5:

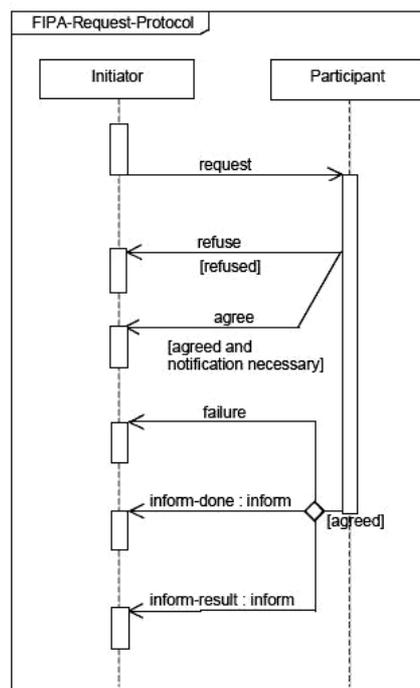


Figura 3.5: Protocolo *FIPA Request Interaction Protocol*

request: solicita uma informação ou execução de um serviço;

refuse: recusa a solicitação;

agree: aceita a solicitação;

failure: falha na tentativa de responder a solicitação;

inform-done: informa que completou a solicitação (realizou o serviço);
 inform-result: informa que completou a solicitação e apresenta os resultados.

Na

Figura 3.6 é apresentado um exemplo de mensagem KQML utilizando o protocolo *FIPA-Request Protocol*, caracterizando o protocolo “failure” que significa a falha na tentativa de responder a solicitação “q1”.

```

reply :language XML :ontology SUS-CartaoNacionalSaude
      :in-reply-with q1 :content(failure(<mensagem>“Usuário Não
                                  Localizado”</mensagem>))
  
```

Figura 3.6: Exemplo de mensagem utilizando o protocolo *FIPA-Request Protocol*

O protocolo *FIPA-Cancel Meta Protocol*, define três interações disponíveis para coordenação e execução de atividades ou serviços dos agentes, tais como apresentado na Figura 3.7:

cancel: cancela uma solicitação de informação ou serviço (request);
 inform-done: informa que completou a solicitação (realizou o serviço);
 failure: falha na tentativa de responder a solicitação;

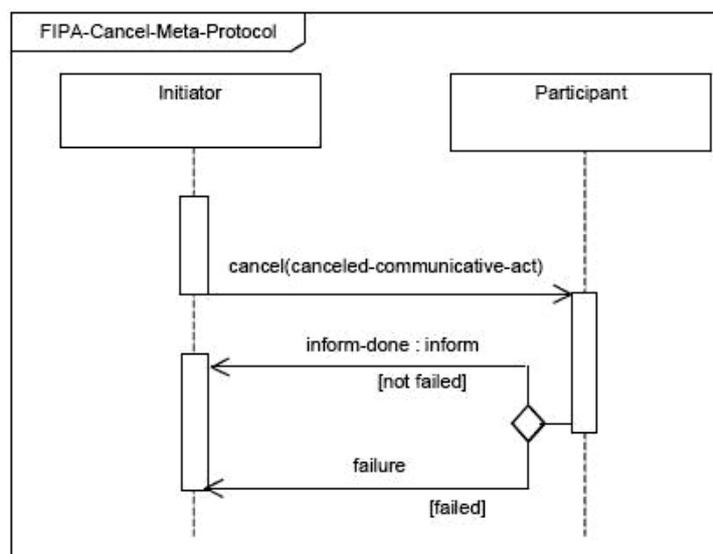


Figura 3.7: Protocolo *FIPA Cancel Meta Protocol*.

3.9. Linguagem de Conteúdo

A linguagem de conteúdo é a linguagem utilizada para descrever do conteúdo comunicado nas mensagens entre os agentes de software. Cada linguagem de conteúdo define um *formato de troca*, ou seja, possui uma sintaxe e semântica para descrever o conhecimento, de forma a possibilitar a interpretação do conteúdo pelos agentes participantes (emissor e receptor) [GEN92]. Obviamente, para que os agentes interpretem os conteúdos de comunicados, devem possuir mecanismos de inferência sobre a linguagem em questão.

A linguagem de conteúdo é avaliada quanto à sua possibilidade de expressão, ou seja, as linguagens diferem quanto ao compromisso ontológico que buscam representar. A princípio, qualquer linguagem de representação de conhecimento formal, ou até mesmo informal, pode ser usada para expressar ontologias. Na prática, no entanto, poucas linguagens têm sido utilizadas para esse fim, entre elas [VAL95]:

- **Lógica de Primeira Ordem:** é comumente usada por ser uma linguagem geral, bem conhecida e expressiva, e por adicionar relativamente poucos compromissos ontológicos. Uma ontologia expressa em lógica é a declaração de uma teoria lógica.
- **KIF (*Knowledge Interchange Format*)** [GRU92]: é uma linguagem formal construída para trabalhar como um meio de comunicação de conhecimento entre bases construídas usando diferentes linguagens. KIF possui uma semântica declarativa (i.e., o significado de uma expressão pode ser compreendido sem a necessidade de recorrer a um interpretador externo); ela é logicamente compreensível (i.e., ela permite escrever expressões arbitrárias no cálculo de predicados de primeira ordem); ela permite definir objetos, funções e relações ([GEN91], [GEN92]).
- **Ontolingua** [GRU92]: é uma linguagem formal e um sistema projetado para o propósito específico de expressar ontologias. Ontolingua foi construída sobre KIF, adicionando mecanismos para expressar classes, relações e hierarquias de classe.
- **CML (*Conceptual Modelling Language*)** [BRE94]: é uma linguagem semi-formal proposta como um formalismo de representação dentro de CommonKADS. CML é largamente inspirada em KL-ONE [BRA85], com construções adicionais para

expressar tarefas, inferências e conhecimento de resolução de problema, de acordo com a infra-estrutura epistemológica adotada por KADS.

- *Description Logic*: é uma lógica projetada para focar categorias e suas definições. Seus principais mecanismos de inferência visam verificar se uma categoria é um sub-conjunto de outra, ou se um objeto pertence a uma categoria [RUS95].
- CORBA IDL (*Interface Definition Language*) é uma linguagem para descrever interfaces de serviços em ambientes heterogêneos desenvolvida pelo grupo OMG. A interface especifica um nome e um conjunto de métodos para que os clientes possam acessar remotamente os serviços. O arquivo que contém as definições gera através de compilação específica os módulos auxiliares para a implementação dos objetos remotos e do cliente [OMG99].
- Web Semântica: é um conjunto de protocolos que definem uma linguagem para descrever e representar o conhecimento, proporcionando a interoperabilidade entre sistemas suportados pela plataforma Web. Os principais elementos que constituem sua estrutura são DTD, XML Schema e RDF [KOI01].

Cada uma dessas linguagens embute um número de compromissos ontológicos e a escolha de uma particular linguagem deve ser feita com base na sua adequação aos propósitos de representação da ontologia.

3.10. Considerações Finais do Capítulo

Neste trabalho, a aplicação de agentes de software tem como objetivo proporcionar a interoperabilidade entre os sistemas que suportam as atividades do Sistema Único de Saúde do Brasil, almejando fornecer mecanismos para integração das fontes de informações produzidas pelos sistemas, possibilitando um compartilhamento do conhecimento entre as áreas de saúde. Para proporcionar um alto grau de expressividade na comunicação entre os agentes, deve-se buscar um nível de ontologia que possibilite representar o conhecimento compartilhado entre os agentes. Com este intuito, no próximo capítulo serão apresentados tipos de ontologias aplicadas para integração de sistemas de informações a fim de definir um nível de ontologia que será empregado na arquitetura desenvolvida.

Capítulo 4 - Ontologias

A evolução da informática está diretamente relacionada com a elevação do nível de abstração das soluções que representam contextos do mundo real. São as abstrações que possibilitam representar genericamente soluções de problemas, possibilitando o reuso em problemas similares. As abstrações aplicadas no meio computacional deram origem ao paradigma declarativo lógico, possibilitando expressar o conhecimento de forma explícita, através de uma especificação formalizada. O paradigma declarativo lógico pode ser visto como uma abstração da tradicional programação imperativa ou procedimental, deixando de descrever os processos e suas nuances e passando a descrever os fatos e entidades de um determinado domínio, ou seja, explicitando o conhecimento sobre um contexto específico [NEW82].

A motivação principal do paradigma declarativo é modelar sistemas num nível mais elevado. O modelo declarativo consiste numa forma mais flexível de descrever um domínio, uma vez que, no nível de conhecimento não há compromisso com a implementação. Desta forma, o conhecimento passa a ser portátil, enquanto conhecimento e não enquanto código procedimental, e, portanto, independente de linguagem de programação ou de ambiente de máquina [NEW82].

O modelo declarativo está diretamente relacionado com a aplicação de ontologia. Através da aplicação de determinada ontologia possibilita-se expressar ou comunicar o conhecimento entre agentes de software. Para tanto, são necessárias a definição de conceitos e a formalização dos termos que expressam o conhecimento sobre determinado domínio ou área do conhecimento. A justificativa para a utilização de ontologias é que esta permite a reutilização desses conceitos, facilitando ao mesmo tempo o seu compartilhamento e disseminação [VAN96].

O termo ontologia teve sua origem fortemente aplicada em filosofia, representando uma teoria sobre a natureza do ser ou existência. Atualmente, a abordagem de ontologias tem sido aplicada em outras áreas como inteligência artificial, lingüística computacional, teoria de banco de dados, etc. A importância da ontologia no meio computacional tem sido reconhecida em segmentos de pesquisa como: engenharia e representação do conhecimento, modelagem

qualitativa, projeto de banco de dados, integração da informação, análise orientada a objetos, integração de informações corporativas, tradução de linguagem natural, sistemas de informações geográficas, sistemas de comércio eletrônico, entre outros [GUA98b].

Segundo Gruber, “*uma ontologia é uma especificação explícita de uma conceituação*” [GRU03]. Uma ontologia é uma especificação explícita de algum tópico, uma representação formal e declarativa que inclui o vocabulário (ou nomes) para se referir aos termos da área em foco e as sentenças lógicas que descrevem o que os termos são, como eles são relacionados com os outros termos e como eles podem ou não podem ser relacionados com os demais termos [GRU91]. As ontologias proporcionam um vocabulário para representar e comunicar conhecimento sobre algum tópico e um conjunto de relacionamentos que existe entre os termos desse vocabulário [GRU04].

Studer, complementa a definição de Gruber atribuindo que “*ontologia é uma especificação explícita e formal de uma conceituação compartilhada*” [STU98]. A conceituação representa o modelo abstrato de uma área do conhecimento, a especificação explícita determina os conceitos, propriedades, restrições, axiomas e relacionamentos definidos através de termos explícitos, por formal entende-se a representação declarativa que pode ser interpretada no meio computacional e, a palavra compartilhada, simboliza a terminologia comum da área modelada ou conhecimento consensual [STU98].

Através da conceituação da visão de um domínio, busca-se especificar ou descrever o domínio, definindo uma ontologia que representada de forma semântica o conhecimento, derivando em um ou mais vocabulários. A Figura 4.1 ilustra este modelo [GUA98b]:



Figura 4.1: Conceituação e ontologia na representação de domínios.

Em sistema de agentes de software, busca-se por meio de determinada ontologia possibilitar a troca de conhecimento entre agentes, ou seja, os agentes devem ter a capacidade

de interpretar semanticamente as informações declaradas durante a comunicação com os demais agentes do sistema.

4.1. Tipos de Ontologias

As ontologias podem ser classificadas de acordo com o seu conteúdo e grau de generalidade. A seguir são apresentados cinco tipos de ontologias:

1. Ontologias genéricas: preocupa-se com a definição de termos gerais de uso comum como espaço, tempo, matéria, objeto, evento, comportamento, função, ação, etc., são conceitos que não dependem de um problema ou domínio específico [MIZ95].
2. Ontologia de domínio: aplica-se a uma área específica do conhecimento, focando nos conceitos e objetos que representam um determinado domínio, como por exemplo área de assistência de saúde, epidemiologia, vigilância sanitárias, etc ([GOM99], [GUA98b]).
3. Ontologia de tarefa: expressa conceituações sobre a resolução de problemas independentes do domínio em que ocorrem, preocupa-se com a definição de termos relevantes de uma atividade específica, descreve-se tarefas como: processos, planos, metas, escalonamentos, etc., com uma visão mais funcional de um domínio ([GOM99], [GUA98b]).
4. Ontologia de aplicação: procura-se solucionar um problema específico de um domínio, ou seja, é uma especialização de uma ontologia dependente do domínio e de tarefa [VAN97]. Tem como objetivo representar um papel específico relacionado a uma ontologia de domínio. Por exemplo, o acompanhamento de mortalidade, a partir de uma ontologia do domínio de epidemiologia.
5. Ontologia de representação: apresenta conceituações que fundamentam os formalismos de representação de conhecimento, tem como objetivo definir primitivas de representação abstraindo os formalismos de representação, por exemplo, recuperação do conhecimento através de frames, axiomas e atributos de forma declarativa. Um exemplo deste tipo de ontologia é a linguagem KIF (*Knowledge Interchange Format*) que tem como propósito ser uma linguagem de conteúdo em

agentes de software e servir como uma *interlingua* na tradução entre outras linguagens de formalismos de representação ([BAA93], [GOM99], [GEN94]).

Guariano, representa as dependências que podem existir entre as ontologias reusáveis de acordo com o grau de generalidade [GUA98b]. Nesta ilustração foi acrescentada a ontologia de representação do conhecimento, utilizada em primeiro nível, independente da aplicação, do domínio ou da tarefa, conforme ilustrado na Figura 4.2:



Figura 4.2: Classificação de ontologias conforme o grau de generalidade.

4.2. Níveis de abordagem de ontologia

Uma ontologia pode ser desenvolvida com níveis mais complexos ou menos elaborada, dependendo da necessidade do problema. Guarino e Welty definem quatro níveis de abordagem de ontologias [GUA98a]:

- a. **Vocabulário:** representa uma forma simples de ontologia, constituindo uma estrutura de elementos que formam um vocabulário. Nesse sentido, uma DTD ou um XML-Schema pode definir uma ontologia. Se diferentes parceiros combinarem com um XML-Schema, eles também concordam com a ontologia definida através do Schema, pois os nomes de tag declarados no DTD definem um vocabulário comum. Quando, por exemplo, dois agentes concordam em compartilhar tal ontologia, eles podem trocar mensagens usando o vocabulário da ontologia.

- b. Taxonomia: classifica os elementos de um vocabulário, atribuindo significado aos termos através da definição de relacionamentos entre eles. Um dos relacionamentos mais naturais é a classificação, onde se estabelecem relacionamentos entre objetos e classes, subclasses e classes-pai. Esses sistemas são denominados taxonomia, cujas relações são conhecidas como relacionamentos “é-um”. Esse tipo de ontologia normalmente é estabelecido por sistemas orientados a objetos. Muitas ontologias existentes são definidas usando-se apenas esses relacionamentos hierárquicos.
- c. Sistema Relacional: estas ontologias tratam de relacionamentos não hierárquicos. Por exemplo, o relacionamento “*é dirigido por*” entre carro e motorista não é um relacionamento hierárquico, mas importante o suficiente para ser descrito em uma ontologia. Tais relacionamentos são típicos nos diagramas de relacionamento de entidades e nos bancos de dados relacionais, e, por conseguinte, cada esquema de banco de dados relacional define sua própria ontologia.
- d. Teoria Axiomática: além de escreverem relacionamentos, as ontologias também podem impor restrições. As restrições são definidas como axiomas. Um axioma é uma afirmação lógica que não pode ser provada a partir de outras afirmações, mas outras afirmações podem ser derivadas da afirmação lógica. Na matemática, teorias inteiras (como a teoria de conjuntos) são derivadas de conjuntos de axiomas relativamente pequenos. Por exemplo, na ontologia de carro citada, a afirmação “um carro tem pelo menos três rodas” é uma restrição desse tipo.

O nível de profundidade na abordagem de ontologia depende do objetivo da aplicação ou problema. O primeiro nível, vocabulário, representa uma abordagem mais específica e menos complexa, porém limitando a capacidade de expressão e representação do conhecimento. Por conseguinte, o último nível (axioma) é mais complexo, possibilitando fazer inferência ao conhecimento representado por sentenças lógicas dando significado a informação, no entanto, são poucas as aplicações práticas desenvolvidas [GUA98a].

4.3. Benefícios e problemas relacionado a ontologias

A aplicação de ontologia em sistemas computacionais facilita o intercambio de informações em um domínio específico, possibilitando expressar o conhecimento de forma

colaborativa entre as aplicações. O uso ontologias trás alguns benefícios para que justificam a sua aplicação, tais como:

- a. Compartilhar a mesma estrutura de informação entre pessoas e agentes de software ([NOY00a], [GRU93]). Por exemplo, se agentes de software distribuídos na rede compartilharem e publicarem as informações usando uma mesma ontologia, possibilitará a extração e agregação das informações entre organizações ou aplicações de determinado domínio.
- b. Permitir o reuso do conhecimento do domínio. Dependendo do grau de generalização de uma ontologia, a abstração de conceituações de um domínio pode ser estendida para outros domínios mais específicos ou mesmo para ontologias de aplicações do domínio [NOY00a].
- c. Possibilidade de tradução entre outras linguagens e formalismos de representação do conhecimento. Devido à existência de tipos diferentes de linguagens declarativas que representam o conhecimento, a evolução da aplicação de ontologias trouxe a necessidade de disponibilizar ferramentas que para construção de ontologias designadas *ontolinguas*, com capacidade de gerar ontologias em diversas linguagens como Prolog, KIF, CLIPS, etc... [NOY00b].
- d. Separar o conhecimento do domínio do conhecimento operacional. Como o conhecimento é representado abstraindo conceitos que representam o conhecimento, o mesmo é desvinculado do processo ou atividade. Desta forma, o conhecimento não é acoplado a determinada atividade funcional [MCG98].
- e. Interação e comunicação do conhecimento. A aplicação de ontologias promove um mecanismo facilitado de comunicação entre pessoas ou agentes de software, possibilitando raciocinar e entender o domínio do conhecimento. A abordagem declarativa da ontologia, constitui um excelente mecanismo de interação e recuperação do conhecimento entre diversas fontes [STU98].
- f. Analisar o conhecimento do domínio. A ontologia define termos explícitos que podem ser expressos de forma declarativa, possibilitando a análise de interpretação do conhecimento sendo uma característica importante para reuso e extensão de ontologias [MCG00].

- g. Formalização do conhecimento. Conforme o grau de formalização do conhecimento da notação utilizada, possibilita-se validar ou restringir o conteúdo, mantendo íntegro os conceitos, evitando contradições ou inconsistências.

A aplicação de ontologia no meio computacional trouxe muitos benefícios como citado anteriormente. No entanto, existem dificuldades e deficiências que devem ser levadas em consideração no uso de ontologias. O'Leary cita alguns problemas como [O'LE97]:

- a. A escolha de uma ontologia é um processo político, principalmente em ontologia de domínios específicos do conhecimento, pois nenhuma ontologia será totalmente adequada a todos os indivíduos ou grupos, ou seja, necessita de um consenso para definição ou adoção de determinada ontologia.
- b. Ontologias não são necessariamente estacionárias ou estáticas, ou seja, necessitam evoluir. Poucos trabalhos têm estudado a evolução de ontologias.
- c. Estender ontologias não é um processo direto. Ontologias geralmente são estruturadas de forma precisa com forte relacionamento, acentuando a complexidade e precisão das definições, o que dificulta a derivação para outras ontologias dependendo o grau de generalidade.
- d. Ontologias desenvolvidas independentemente podem não se integrar efetivamente com outras por vários motivos, desde similaridade de vocabulário até visões conflitantes do mundo.
- e. A falta de metodologia padronizada para construção de ontologias, acaba divergindo em diferentes abordagens de métodos para desenvolvimento de ontologias, e ainda assim muito incipiente, com ausência de ciclos de vida e métodos sistemáticos.
- f. Segundo SWARTOUT, muitas ontologias são distribuídas em um formato de código fonte na linguagem de representação, o que não permite que usuários naveguem através da ontologia para compreender seu escopo, estrutura e conteúdo [SWA97].

4.4. Aplicações de Ontologias em Agentes de Software

Ontologias pré-construídas sobre domínios restritos têm sido bastante reutilizadas e podem vir a representar um papel fundamental como fornecedoras de conhecimento para a inferência dinâmica realizada por agentes inteligentes.

Existem algumas aplicações desenvolvidas sobre arquiteturas que utilizam ontologias para modelar sistemas. Seguem abaixo três exemplos de aplicações que utilizam agentes de software baseados em ontologias:

SHADE (*SHARed Dependency Engineering*)

SHADE é um ambiente que permitiu explorar a utilização de KQML. Seu objetivo era de definir uma infra-estrutura agente para a engenharia colaborativa na manufatura ([KUU95a], [MCG93a]). SHADE inclui técnicas para a definição e a utilização de ontologias, e oferece uma interface de programação de aplicações para KQML. SHADE define agentes mediadores, esses agentes assistem o usuário final no compartilhamento de informações. O mediador mais importante definido para SHADE se denomina matchmaker [KUU95b].

PACT (*Palo Alto Collaboration Testbed*)

O PACT é um ambiente de teste de colaboração aplicado na manufatura [CUT93], aplica ontologia de vocabulário para comunicação entre agentes distribuídos. Este projeto tinha como objetivo a resolução negociada de problemas de projeto, fabricação e revisão de manipuladores robóticos. Os agentes tinham como vocabulário comum duas ontologias, uma matemática para engenharia, constando classes como quantidades, unidades, dimensões, matrizes e funções; e outra de projeto e configurações, com as classes de parâmetros, componentes, restrições, etc.

MASTERWeb (*Multi-Agent System for Text Extraction and Retrieval over the Web*)

Tem como objetivo a manipulação integrada de informação através de ontologias [FRE03]. Propõem uma arquitetura de sistemas multi-agente para manipular a informação referente a um conjunto de classes sobre um mesmo grupo, como, por exemplo, um grupo científico, com classes como: artigos científicos, eventos, pesquisadores, etc. A arquitetura visa recuperar, classificar e extrair dados de páginas pertencentes às classes do grupo, beneficiando-se dos relacionamentos entre as classes. Utiliza-se a ontologia não só como

vocabulário de comunicação entre agentes, como também na definição e organização apropriadas de conceitos, relações, e restrições.

4.5. Considerações Finais do Capítulo

O advento da ontologia no meio computacional teve uma evolução considerável, mesmo com alguns problemas ou dificuldades, o desenvolvimento de sistemas com capacidade de compartilhar o conhecimento de forma declarativa, colaboraram para justificar a aplicação de ontologias. Esta evolução emergiu para aplicação da ontologia na Web, derivando em estudos para elaboração e padronização de mecanismos que possibilitasse a troca de conhecimentos neste contexto. Atribuindo a Web elementos úteis e legíveis que possam ser interpretados e compartilhados por agentes de software, constituindo uma semântica para Internet denominada *Web Semântica*.

Capítulo 5 - Web Semântica

A *Web Semântica*⁴ é uma extensão da Web atual na qual é dado à informação um significado bem definido, permitindo com que computadores e pessoas trabalhem em cooperação [BER01]. Tem como objetivo a construção de novos paradigmas para a representação de conhecimento, de forma a facilitar a localização, comparação e integração de recursos através da Web. A Web Semântica organiza o conteúdo de forma a dar significado as informações representadas através na Web, criando um ambiente no qual os dados podem ser compartilhados e processados tanto por ferramentas automáticas como também por humanos. A Web Semântica permitirá que comunidades publiquem seus dados de forma que um programa não tenha que percorrer uma página e remover os comandos de formatação para obter a informação relevante. Possibilitará definir o relacionamento entre os diferentes conjuntos de dados ou entidades representadas na Web, facilitando a integração de diversas fontes de dados. Dessa forma, através da Web Semântica poderá ser atribuído um sentido bem definido a cada informação, habilitando computadores e pessoas a trabalharem em cooperação. Ela se baseia na idéia de se terem dados na Web definidos e interligados de forma a facilitar a descoberta, automação, integração e o reuso através de várias aplicações [HEN02]. Na Figura 5.1 são apresentadas as camadas que constituem a Web Semântica conforme definido por Berners-Lee [BER01]:

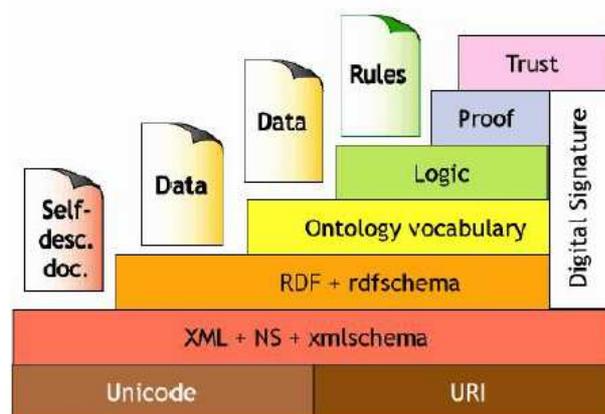


Figura 5.1: A arquitetura de camadas da Web Semântica [BER01]

⁴ Semântica é uma entidade abstrata que representa ou modela o significado termos que podem ser interpretados, por exemplo, em uma linguagem de programação [MOS92].

- URI: o *Uniform Resource Identifier* já é um componente essencial na Web atual, qualquer recurso disponível na Web é identificado por um URI. Seu tipo mais conhecido, a URL, permanece como localizador de fontes de informação.
- XML e XML Schema: na Web semântica, XML é base para a representação dos dados em um formato que facilite a interoperabilidade entre aplicativos. XML Schema consiste na recomendação do W3C para dar estrutura aos documentos XML.
- RDF e RDF Schema: define como descrever recursos em termos de suas propriedades e valores. RDF Schema, por sua vez, define propriedades específicas que podem ser utilizadas para definir esquemas. RDF-S permite verificar se um documento RDF está coerentemente preenchido e ainda checar se, por exemplo, a carteira de um motorista tem o nome de uma pessoa e não um modelo de carro como seu "nome". Documentos RDF neste nível não têm grande poder, mas se espera que esses dados, enquanto limitados dentro de uma aplicação, sejam combinados, mais tarde, com dados de outras aplicações dentro da Web. Essas aplicações serão capazes de usar uma estrutura comum para combinar sua informação.
- OWL: o próximo nível é o nível lógico. É necessário que haja caminhos de escrita lógica dentro de documentos que permitam recursos, tais como a checagem de um documento contra um conjunto de regras de autoconsistência. Esta camada necessita de uma linguagem para representação de ontologias. Nesse sentido, o W3C⁵ está trabalhando na especificação da linguagem OWL (*Web Ontology Language*).

5.1. Extensible Markup Language (XML)

A linguagem XML vem se estabelecendo como um padrão para representação de dados semi-estruturados, e potencialmente sendo utilizado para representar dados

⁵ W3C é uma organização denominada *World Wide Web Consortium*, tem como objetivo prover a especificação de padrões utilizados na Web, como linguagens, protocolos e arquiteturas.

manipulados por aplicações ou transmitidos através da Web. XML é uma linguagem computacional que possibilita descrever informações de forma estrutura [MCG98]. Também é denominada linguagem de marcação apropriada à representação de dados, documentos e demais entidades cuja essência fundamenta-se na capacidade de agregar informações. Desta forma, é possível representar os dados com uma semântica definida para que possa ser interpretado o dado em um momento oportuno, considerando aspectos de relação e de estrutura do dado.

O objetivo da linguagem de esquema XML é definir uma sintaxe para validar a estrutura do conteúdo dos dados representados através da linguagem XML, ou seja, através da definição de esquemas XML é possível avaliar a semântica das informações descritas em documentos XML [BRA04]. Um esquema XML tem como propósito definir e descrever uma classe de documentos XML usando regras para restringir e documentar o significado, uso e relações das partes que constitui o conteúdo do documento, tais como: tipos de dados, elementos, atributos, anotações, entidades, etc. Desta forma, a linguagem de esquema XML pode ser usada para definir, descrever e catalogar vocabulários para classes de documentos XML.

Comparado aos esquemas de banco de dados, pode-se definir que um esquema XML é um modelo para descrever a estrutura da informação, definindo a estrutura dos dados dispostos conforme o arranjo de *tags* e textos dentro de um documento válido.

Atualmente existem algumas propostas de linguagem de esquema submetidas ao *Grupo de Trabalho XML Schema* da W3C, entre as propostas aceitas pela W3C pode-se citar duas linguagens que estão sendo aplicadas: *Document Type Definition* e *XML Schema* [BRA04].

Na Tabela 5.1 é apresentado um conteúdo de dados representados em XML, descrevendo informações relacionadas a usuários e medicamentos:

Tabela 5.1: Exemplo de conteúdo representado em XML

```
<?xml version="1.0"?>
<usuario cns = 1>
  <nome>Jose da Silva</nome>
  <datanasc>03/08/1970</datanasc>
  <sexo>M</sexo>
  <medicamento>
```

```

    <principioativo>Ácido Acetilsalicílico</principioativo>
    <posologia>100mg/dia</posologia>
  </medicamento>
  <medicamento>
    <principioativo>Paracetamol</principioativo>
    <posologia>25mg/dia</posologia>
  </medicamento>
  <medicamento>
    <principioativo>Diclofenaco de Potássico</principioativo>
    <posologia>50mg/dia</posologia>
  </medicamento>
</usuario>
<usuario cns = 2>
  <nome>Maria da Silva</nome>
  <datanasc>03/09/1971</datanasc>
  <sexo>F</sexo>
  <medicamento>
    <principioativo>Diclofenaco de Potássico</principioativo>
    <posologia>50mg/dia</posologia>
  </medicamento>
</usuario>

```

5.2. Document Type Definition (DTD)

A linguagem DTD é derivada da SGML (*Standard Generalized Markup Language*), e tem como objetivo prover mecanismos para definição de estrutura de documentos em XML [BRA04]. Um documento DTD aplica-se sobre o conteúdo de documentos XML como uma gramática, onde é consistido a semântica das informações descritas no documento.

A linguagem DTD é composta por um conjunto de declarações, as quais podem ser elementos, atributos, entidades e notações. Através de regras de estruturais são incluídas restrições aos documentos XML, que podem depois ser verificadas pelo processador, a fim de determinar a validade ou não de um documento. Na Tabela 5.2 é apresentado um exemplo de documento DTD definindo a estrutura de dados que representam a prescrição médica de um paciente.

Tabela 5.2: Exemplo de definição de esquema em DTD

Linha	Descrição
1.	<code><!ELEMENT usuario (nome, datanasc, sexo, medicamento+)></code>
2.	<code><!ATTLIST usuario cns ID #required></code>
3.	<code><!ELEMENT id (#PCDATA)></code>
4.	<code><!ELEMENT nome (#PCDATA)></code>
5.	<code><!ELEMENT datanasc (#PCDATA)></code>
6.	<code><!ELEMENT sexo (M F)></code>
7.	<code><!ELEMENT medicamento (principioativo, posologia)></code>
8.	<code><!ELEMENT principioativo (#PCDATA)></code>
9.	<code><!ELEMENT posologia (#PCDATA)></code>

As declarações são definidas com uma sintaxe específica com objetivo de descrever a estrutura dos dados, sendo representado a seguinte forma [BRA04]:

Declaração de elementos: a declaração dos elementos é especificada através do nome e o tipo de conteúdo que eles podem armazenar. Uma declaração de elementos segue o formato “`<!ELEMENT ...> nome modelo-deconteúdo`”. O modelo de conteúdo especifica que tipo de conteúdo o elemento pode conter. Em DTD, existem alguns tipos de dados pré-definidos. O tipo `#PCDATA` representa uma seqüência de caracteres.

Declaração de atributos: a declaração de atributos identifica quais atributos um elemento pode possuir, o seu tipo e valor default, representado por “`<!ATTLIST...>`”. Na linha 2, pode-se notar que o elemento “usuário” possui um atributo chamado “cns”. Esse atributo pode receber valores do tipo ID, este tipo defini que o valor deve ser único para a atributo em cada ocorrência do elemento “usuário”. Além disso, o atributo é `#required`, o que indica ser sempre necessário atribuir algum valor. Na linha 6, é restringido os valores do elemento “sexo” como masculino ou feminino. Existem outros qualificadores de obrigatoriedade e de valor do conteúdo que podem ser definidos para os de atributos.

Seqüência: os sub-elementos que pertencem a um elemento principal, como no caso dos sub-elementos que pertencem ao “*usuário*” obedecem a ordem de apresentação conforme o conector lógico: “;” (conector de seqüência) e “|” (conector de escolha).

Cardinalidade: no exemplo acima, a linha 1 diz que o elemento “*usuário*” pode conter um ou mais elementos do tipo “*medicamento*”, e também é composto pelos elementos “nome, datanasc, sexo”. O sinal de mais (+) que aparece após o nome medicamento é um qualificador, que indica que esse item pode ser repetido uma ou mais vezes. Outros qualificadores que podem ser utilizados são o ponto de interrogação (?), que indica que o item pode ocorrer zero ou uma vez, e o asterisco (*), que indica que um item pode ocorrer zero ou mais vezes.

Notação: indica que o valor do atributo deve ser uma das notações definidas através da declaração “<!NOTATION ...>”, permitindo que um elemento contenha dados não XML, desde que estes dados estejam de acordo com o formato definido na notação.

Entidade: Uma entidade é utilizada para reaproveitamento de declarações. O uso de entidades pode diminuir o montante do trabalho necessário para o projeto de uma DTD e diminuir erros provenientes deste trabalho, podem ser entidades gerais ou de parâmetro. A declaração de uma entidade geral é representada por “<!ENTITY nomeentidade ...>” e pode ser referenciada através da expressão “&nomeentidde”. No caso da entidade de parâmetro é representada por “<!ENTITY %nomeentidade ...>” e pode ser reutilizada na declaração de outros elementos referenciando através de “%nomeentidade”.

5.3. XML Schema

O W3C propôs o XML Schema para servir como linguagem de definição de tipos de documentos XML, com objetivo semelhante aos esquemas definidos em DTD, porém, com propósito de estabelecer alguns tipos de dados mais específicos como: *string*, *integer*, *binary*, etc., assim como fornecer os mecanismos para criar novos tipos de dados. XML Schema satisfaz os requisitos das aplicações que exigem uma tipagem mais rica do conteúdo por ser

uma linguagem de esquema fortemente tipada⁶. Herdou muitas características das linguagens orientadas a objeto (por exemplo, definição de tipos e herança). Por “tipagem mais rica ou forte”, entende-se por criação de tipos que atendam a necessidade real do conteúdo e extensão de tipos pré-definidos na linguagem [MCG98].

Com relação à validação de documentos, espera-se que com a utilização de esquemas se consiga testar tanto a validação de estruturas de conteúdo quanto a validação do conteúdo. A primeira testa se a ordem e aninhamento dos elementos está correta, enquanto que a segunda verifica se um determinado valor é do tipo correto; ou seja, se está de acordo com as regras de validação daquele tipo.

As características principais do XML Schema são: definição de elementos e atributos, definição de tipos complexos (*complexType*) ou derivação de tipos simples já existentes (*simpleType*), definição de grupos, herança de tipos por restrição ou extensão, criação de grupos de substituição, definição de tipos abstratos e restrições de integridade. Na Tabela 5.3 é apresentado um XML Schema com a mesma funcionalidade de validação do conteúdo XML representada anteriormente através do esquema DTD [BRA04]:

Tabela 5.3: Exemplo de definição de esquema em XML Schema

Linha	Descrição
1.	<code><xsd:element name="usuario" type="tUsuario"/></code>
2.	<code><xsd:complexType name="tUsuario"></code>
3.	<code> <xsd:attribute name="cns"/></code>
4.	<code> <xsd:sequence></code>
5.	<code> <xsd:element name="nome" ref="nome"/></code>
6.	<code> <xsd:element name="datanasc" ref="datanasc"/></code>
7.	<code> <xsd:element name="sexo" ref="sexo"/></code>
8.	<code> <xsd:element name="medicamento" ref="medicamento" minOccurs="1"</code>
9.	<code> maxOccurs="unbounded"/></code>
10.	<code> </xsd:sequence></code>
11.	<code></xsd:complexType></code>
12.	
13.	<code><xsd:element name="nome"/></code>
14.	<code><xsd:element name="datanasc"/></code>
15.	<code><xsd:element name="sexo"/></code>

⁶ Linguagem de esquema tipada, representa uma estrutura de dados bem definida conforme um domínio de valores possíveis de serem interpretados como, por exemplo, inteiro, caractere, data, literal, etc.

16.	
17.	<code><xsd:element name="medicamento" type="tMedicamento"/></code>
18.	<code><xsd:complexType name="tMedicamento"></code>
19.	<code><xsd:sequence></code>
20.	<code><xsd:element name="principioativo"/></code>
21.	<code><xsd:element name="posologia"/></code>
22.	<code></xsd:sequence></code>
23.	<code></xsd:complexType></code>

5.4. Resource Description Framework (RDF)

A flexibilidade de XML para criar dialetos, que podem ser formalizados por um esquema DTD ou *XML Schema*, leva naturalmente à explosão no número dessas gramáticas, ou “padrões” para troca de dados em formato XML em diferentes áreas de negócio. Para comércio eletrônico, por exemplo, tem-se os “padrões” cXML e ebXML. Na verdade, setores que já evoluíram para um pequeno número de diferentes “padrões” são os melhores casos. Em algumas áreas de negócio, não existem padrões de fato estabelecidos, de modo que formatos de troca de informações são definidos sob demanda, muitas vezes sem maiores análises quando de suas criações.

A existência de várias possibilidades para codificação de um determinado domínio traz dificuldades em situações em que se necessita entender documentos de terceiros. Considere um conjunto de parceiros que trocam documentos XML com uma determinada finalidade de negócio. Se dois parceiros utilizam formatos diferentes para seus documentos, um passo de transformação XSLT tem que ser configurado e incorporado ao processo de troca. Se um parceiro que utiliza um novo formato é introduzido, passos de transformação para todos os outros formatos existentes precisam ser configurados, o que demanda um esforço considerável. Para evitar esse tipo de situação, é necessário uma camada de abstração em cima de XML, na qual se possa representar conceitos.

Tal camada para representação de conhecimento na Web deve levar em consideração alguns requisitos essenciais. Inicialmente, seguindo o modelo da própria Web, o conhecimento deve ser distribuído e conectável. Segundo, não se pode trabalhar com modelos de mundo fechado, isto é, não podem haver verdades absolutas. Cada aplicação deve ter liberdade para construir suas verdades, e neste processo se basear, na forma que decidir, em

conceitos de outras partes, que podem em alguns casos ser inclusive antagônicos. Finalmente, deve-se prezar pela simplicidade e extensibilidade, para que possam ser derivadas soluções para diferentes comunidades de usuários.

RDF é a recomendação W3C para a definição e uso de meta-dados. Diferentemente de XML Schema ou DTD, voltados para tipagem e definição estrutural, a proposta de RDF envolve meta-dados conceituais, que têm por objetivo tornar informações disponíveis para aplicações diferentes daquelas para as quais estas informações foram originalmente criadas [LAS99].

O modelo de dados de RDF consiste de três tipos de objetos [LAS99]:

- Um recurso é qualquer coisa que possa ser referenciada por uma URI. Isto inclui páginas Web, assim como elementos individuais em um documento XML.
- Uma propriedade é um aspecto específico, característica, atributo ou relação usada para descrever o recurso. Uma propriedade pode e deve ser um recurso, referenciada por uma URI e podendo ter suas próprias propriedades.
- Uma sentença consiste da combinação de um recurso, uma propriedade e um valor. Essas partes são chamadas, respectivamente, de sujeito, predicado e objeto da sentença. O objeto pode ser um literal, ou um outro recurso.

Na Tabela 5.4 é apresentado um exemplo de dados estruturados no formato RDF representando os dados de identificação do endereço de um estabelecimento de saúde.

Tabela 5.4: Exemplo de dados em formato RDF

```
<?xml version="1.0"?>
<rdf:Description rdf:about="http://www.saude.org#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://www.saude.org #">
  <rdf:type rdf:resource="http:// http://www.saude.org#"/>
  <tipoEstabelecimento>Hospital</tipoEstabelecimento>
  <ruaEndereco>Rua das Pinheiros</ruaEndereco >
  <estadoEndereco>East China Sea</estadoEndereco>
</rdf:Description>
```


Capítulo 6 - Arquitetura *Web Service*

A Internet é um dos mecanismos de comunicação mais utilizado no mundo, seu diferencial em relação aos outros meios de comunicação é interação que proporciona às pessoas e organizações.

No princípio a Internet era voltada muito mais para divulgação de informações representadas através de textos e imagens moldados em HTML (*Hyper Text Markup Language*) e suportados através do protocolo HTTP (*Hypertext Transfer Protocol*).

Atualmente, com a evolução de padrões voltados para Internet, cada vez mais é possível sistematizar operações das organizações através da Internet. Isso é facilmente percebido pela diversidade de sistemas de informação construídos sobre a plataforma da Internet. Um exemplo típico são os serviços disponíveis para população como serviços bancários, serviços do governo, entre outros. Os serviços atualmente disponíveis intensificam a interação entre as organizações com as pessoas.

Como um processo evolutivo da interação que a Internet proporciona, os *Web Services* buscam a interoperabilidade entre aplicações heterogêneas de forma facilitada por utilizar como principal meio de transporte o protocolo HTTP. No entanto, o objetivo de interoperabilidade entre programas já estava presente em outras plataformas e tecnologias como CORBA, COM e Java RMI ([CZE03], [RAJ98]).

Web Service se propõem a prover maior interação entre as aplicações de organizações distintas, independente da plataforma ou linguagem da aplicação. Esta interoperabilidade entre as aplicações é suportada por interfaces definidas através da linguagem XML (*eXtensible Markup Language*) [BRA04]. Desta forma, a comunicação entre os programas distribuídos na rede é proporcionada por padrões que definem a tecnologia de *Web Service*. Atualmente os principais elementos que constituem esta tecnologia são: SOAP (*Simple Object Access Protocol*), WSDL (*Web Services Description Language*), UDDI (*Universal Discovery Description and Integration*) definidos com propósitos específicos, todos baseados em XML e DTD (*Document Type Definition*) ([CZE03], [BRA04]).

Estas tecnologias e outras que estão em desenvolvimento como processo de descoberta, agregação e coordenação (orquestração) de serviços Web, segurança e gerenciamento de transação, serão apresentadas a seguir.

6.1. Estrutura Funcional de *Web Service*

Web Service originou-se de propostas de padrões de comunicação submetidos por empresas de tecnologias a W3C, entidade responsável por estabelecer padrões para Internet [BOO04]. Atualmente existem algumas tecnologias desenvolvidas para suportar arquitetura de *Web Service* aprovadas pela W3C e outras propostas em estudo submetidas a organizações como OASIS, WS-I, Liberty Alliance e DAML ([OAS04], [LIB04], [DAR04], [WSI04]).

Web Service é um componente de software ou programa que pode ser acessado por diferentes sistemas por meio do uso de linguagens e protocolos padronizados, tais como XML, SOAP e WSDL ([BER96], [BRA00], [BOX00]).

Através de um endereço URI (*Uniform Resource Identifiers*) como por exemplo <http://www.sitesaude.br/webservice.exe/wsdl>, pode ser identificado um *Web Service* ([NEW02], [BER98]).

A estrutura de funcionamento de um *Web Service* com objetivo de interoperabilidade entre aplicações caracteriza-se como sistema distribuído, sendo de fundamental importância estabelecer padrões para possibilitar a comunicação e operação entre sistemas. Para que uma aplicação possa acessar um serviço ou programa de um *Web Service*, a aplicação cliente deve suportar SOAP e XML [BOX00].

As aplicações clientes localizam e identificam a interface de serviços disponíveis no *Web Service* através da linguagem WSDL. Por meio desta linguagem são descritos os métodos e parâmetros dos serviços disponíveis. Desta forma, a aplicação cliente conhecendo a descrição dos métodos publicados nos documentos WSDL, poderá utilizá-los passando os parâmetros necessários para acessar o serviço [CHR01].

Para facilitar a localização dos serviços disponíveis na rede, está sendo proposto um diretório de registro e pesquisa dos serviços *Web Service* chamado UDDI (*Universal Description Discovery Integration*). Informações sobre os serviços disponíveis poderão ser extraídas dos documentos WSDL publicados neste diretório.

A estrutura de funcionamento de *Web Service* consiste em três componentes conforme Figura 4 [MYE04]:

- *Service Provider*: elaboram os *Web Services* e publicam na rede registrando os serviços no *Service Broker*;
- *Service Broker*: mantém os registros dos serviços publicados e fornece mecanismos de pesquisa;
- *Service Requester*: procuram os serviços registrados no *Service Broker* e conectam aos serviços disponíveis no *Service Provider*.

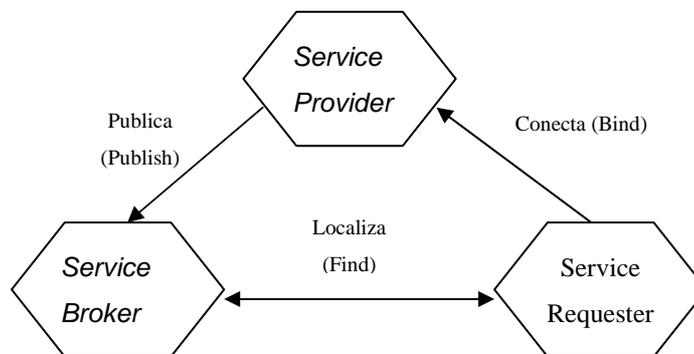


Figura 6.1: Estrutura Funcional de *Web Service*.

Para que o *Service Requester* que representa o cliente comunique-se com o *Service Provider* que representa o servidor, deve conhecer a interface e a localização do *Service Provider*. Esta informação pode ser obtida através do documento WSDL solicitando diretamente do *Service Provider* ou localizado através do *Service Broker* que representa o diretório de publicação de serviços, neste caso utilizando o protocolo UDDI [CLE04]. De posse destas informações, o *Service Requester* pode conectar-se ao *Service Provider*, utilizando o protocolo SOAP para interação entre as aplicações.

Desta forma, os seguintes passos são necessários para que ocorra a interação de uma aplicação cliente com um *Web Service*:

1. O servidor do negócio, representado pelo *Service Provider*, desenvolve um serviço ou aplicação e gera um documento WSDL que descreve os serviços com seus parâmetros de entrada e saída.

2. O documento WSDL pode ser repassado diretamente aos clientes do servidor do negócio, representado pelo *Service Requester*, ou pode ser registrado em um diretório de serviços para facilitar a busca pelos clientes. Neste último caso o documento WSDL será publicado no diretório de serviço UDDI, que representa o *Service Broker*.
3. As aplicações clientes (*Service Requester*) podem consultar os serviços disponíveis no diretório *Service Broker* e recuperar o documento WSDL sobre o resultado da consulta.
4. Após adquirirem o documento WSDL, os clientes podem interagir diretamente com o servidor do negócio utilizando para isso a transmissão de mensagens SOAP sobre o protocolo de comunicação identificado, por exemplo HTTP.

6.2. Padrões ligados a Arquitetura *Web Service*

Os padrões e tecnologias utilizados para suportar a interoperabilidade dos serviços de *Web Service* podem ser classificados em camadas. Todas as tecnologias desenvolvidas utilizam como referência à linguagem XML, DTD e XML Schema, ilustradas na Figura 6.2 [BOO04]:

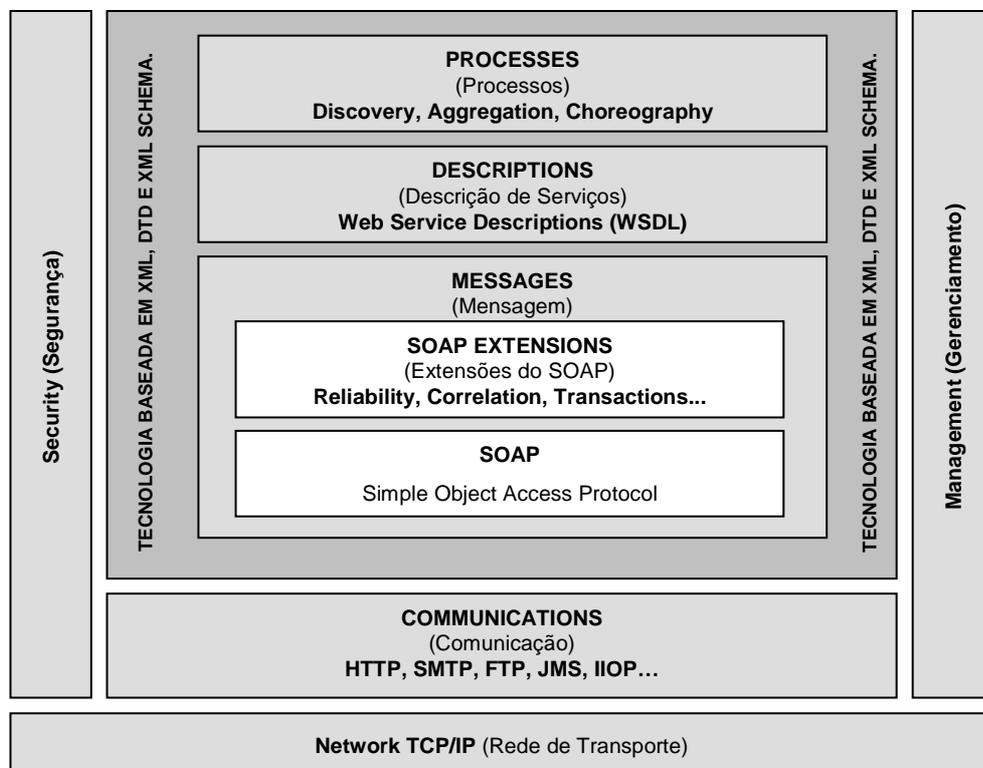


Figura 6.2: Arquitetura *Web Service* [BOO04].

6.2.1. Rede de Transporte TCP/IP

O serviço de rede responsável pela transmissão de dados utilizado na arquitetura *Web Service* é o protocolo TCP/IP (Transmission Control Protocol / Internet Protocol), representam um conjunto de protocolos de comunicação utilizados para troca de dados entre computadores em ambiente LAN (rede local) e WAN (rede longa distância). A Internet utiliza o TCP/IP como protocolo padrão para comunicação, da mesma forma a arquitetura *Web Service* também utiliza este protocolo para prover a transmissão de dados de baixo nível [CYC96].

O modelo OSI (*Open Systems Interconnections*) especificado pela ISO (*International Organization for Standardization*) define sete camadas das quais quatro é implementada pelo protocolo TCP/IP, conforme apresentado na Tabela 6.1 ([OSI89], [BOU81]):

Tabela 6.1: Camadas da Arquitetura TCP/IP.

Modelo OSI	Protocolo TCP/IP
Aplicação	Application Protocol (TELNET, FTP, HTTP, SMTP, etc...)
Apresentação	
Sessão	
Transporte	Service Protocol (TCP / UDP)
Rede	Network Routing (IP)
Enlace	Data-Link (Ethernet, Token Ring, PPP, etc...)
Física	

- Camada de Enlace (*Data-Link*): Especifica como os pacotes são transportados sobre a camada física, identificando o início e o fim dos pacotes de dados, sendo responsável pelo controle de erros de transmissão, controle do sincronismo da transmissão e controle de tráfego. São utilizados nesta camada protocolos como Ethernet, Token Ring, PPP, X.25, Frame Relay, etc. Esta camada é especificada no TCP/IP também chamada de *Physical Network* ou *Host/Rede*.

- Camada de Rede (Network): Esta camada é responsável pelo roteamento e retransmissão de mensagens até chegar a mensagem no destino desejado. Esta camada é especificada pelo TCP/IP através do protocolo IP (*Internet Protocol*), também denominada como *Routing* ou *Inter-Rede*.
- Camada de Transporte (*Service Protocol*): Responsável pela transferência segura de mensagens entre a origem e o destino, executando o controle de fluxo e seqüência de transmissão. É também responsável por dividir os dados em unidade menores como “pacotes”, e garantir a integridade destas unidades até chegarem no destino. Esta camada é especificada pelo TCP/IP através dos protocolos *TCP (Transmission Control Protocol)* e *UDP (User Datagram Protocol)*, também denominada como *Transport*.
- Camada de Aplicação: Responsável por serviços específicos da aplicação, representa uma diversidade de protocolos como *TELNET (Terminal Emulation)*, *FTP (File Transfer Protocol)*, *HTTP (Hypertext Transfer Protocol)*, *SMTP (Simple Mail Transfer Protocol)*, etc. Também é denominada como *Application Protocol*.

As camadas, Física, Sessão e Apresentação, são especificadas pelo modelo OSI, porém não são implementadas pelo TCP/IP.

6.2.2. Camada de Comunicação

São os protocolos suportados pela arquitetura TCP/IP. Estes protocolos são considerados de alto nível por pertencerem a última camada da arquitetura TCP/IP denominada camada de aplicação ([OSI89], [BOO04]).

Estes protocolos são chamados de *Application Protocols*, pois provêm funcionalidades específicas como, por exemplo, HTTP para transferência de hipertexto, SMTP para transferência de e-mail, FTP para transferência de arquivos, JMS como serviço de mensagem de Java, IIOP para comunicação entre objetos, etc. O protocolo HTTP e HTTPS (*Secure Hypertext Transfer Protocol*) são os protocolos mais utilizados em *Web Service* por proporcionar recursos que facilitam a interação entre aplicações.

6.2.3. Simple Object Access Protocol (SOAP)

SOAP é um protocolo utilizado para troca de mensagens entre aplicações utilizando a linguagem XML para estruturar o formato da mensagem. A especificação de SOAP define um envelope para transmissão das mensagens, estabelece a estrutura para codificação dos dados e fornece mecanismos para chamada de procedimento remoto (RPC) ([ROY01], [GUD03]).

As mensagens de SOAP são transmitidas de uma origem a um destino suportado por protocolos de alto nível como HTTP, SMTP, entre outros [POS82, FIE99]. O protocolo mais utilizado é o HTTP por facilitar a transmissão sem ser interrompido por softwares que restringem acesso a execução de procedimentos como *firewalls*.

A especificação SOAP define uma estrutura da mensagem constituída por um cabeçalho opcional, um envelope e um corpo de mensagem descrito em XML. Através do protocolo SOAP pode ser enviada uma mensagem de uma aplicação para outra por meio de um protocolo de comunicação / transmissão, solicitando ou respondendo por serviços ([CZE03], [W3C]).

A Figura 6.3 apresenta um exemplo de uma solicitação SOAP transmitida através do HTTP. No exemplo, a mensagem envia a solicitação do CID (*Código Internacional de Doença*) do diagnóstico de determinado paciente evocando o método *getCIDPaciente* e passando como parâmetro a identificação do paciente *idPaciente* através da mensagem *getCIDPacienteRequest* ([SCH04a], [SCH04b]).

<pre>POST / SITESAUDE HTTP/1.1 Host: www.sitesaude.br Content-Type: text/xml; Charset = "utf-8" Content-Length: nnnn SOAPAction: "www.sitesaude.br/webservice/assistenciasaude.htm"</pre>	Cabeçalho
<pre><SOAP-ENV: Envelope xmlns: SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" SOAP-ENV: encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/" Xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"></pre>	Envelope (início)
<pre><SOAP-ENV: Body> <m: getCIDPacienteRequest xmlns:m="www.sitesaude.br/webservice/assistenciasaude.htm"> <idPaciente xsi:type="xsd:cartaoSaude">99001</idPaciente> </m: getCIDPacienteRequest></pre>	Corpo da Mensagem

<SOAP-ENV: Body>	
</SOAP-ENV: Envelope>	Envelope (fim)

Figura 6.3: Mensagem SOAP para uma solicitação (*Request*).

O cabeçalho da mensagem apresenta informações relacionadas à segurança, formato do conteúdo da mensagem, endereçamento ou roteamento e informações sobre o tratamento da mensagem ao destinatário. No envelope da mensagem é identificada a codificação do envelope e do esquema XML (*XMLSchema*) utilizado como referência para descrever a mensagem. Dentro do envelope é localizado o corpo da mensagem onde são descritos os métodos e parâmetros de entrada ou resposta ([NOV04a], [W3C04]).

A Figura 6.4 apresenta um exemplo de resposta da mensagem de solicitação, descrita de forma semelhante. Porém exibindo no envelope uma resposta com os parâmetros esperados conforme retorno do método, no caso, a informação do CID é enviado como resposta.

POST / SITESAUDE HTTP/1.1 Content-Type: text/xml; Charset = "utf-8" Content-Length: nnnn	Cabeçalho
<SOAP-ENV: Envelope xmlns: SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" SOAP-ENV: encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/" Xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance">	Envelope (início)
<SOAP-ENV: Body> <m:getCIDPacienteResponse xmlns:m="www.sitesaude.br/webservice/assistenciasaude.htm"> <cidDiagnostico xsi:type="xsd:cid10">G30</ cidDiagnostico> </m: getCIDPacienteResponse> </SOAP-ENV: Body>	Corpo da Mensagem
</SOAP-ENV: Envelope>	Envelope (fim)

Figura 6.4: Mensagem SOAP para uma Resposta (*Response*)

6.2.4. Descrição de Serviços

Uma descrição de serviço contém os detalhes da interface e, potencialmente, o comportamento esperado do serviço. Isto inclui tipo de dados, operações, protocolo de transporte da informação e endereço onde está localizado o serviço. Também pode incluir categorização e meta-dado para facilitar a interpretação. A descrição completa pode ser feita através da descrição de documentos XML.

Existem potenciais usos de descrições de serviço como: usados para facilitar a construção e desenvolvimento de serviços, usados por pessoas para localizar serviços apropriados e também podem ser usados por agentes solicitantes para descobrir automaticamente outros agentes provedores de serviços, possibilitando uma escolha satisfatória do serviço [BOO04].

Linguagem de Descrição de Serviços Web (WSDL)

WSDL é uma linguagem de descrição de serviços utilizada para definição de interface de sistemas. A estrutura do documento WSDL é definida usando como referência XML, DTD e XML Schema, possibilitando a descrição das funcionalidades disponibilizadas pelo *Web Service* em um formato padrão, para que outras aplicações possam interpretar os serviços [CZE03, BOO04].

A linguagem WSDL descreve o serviço Web para possibilitar a troca de mensagens entre agentes solicitantes e provedores de serviços. As mensagens são descrições abstratas transportadas sobre um protocolo de rede em no formato de mensagem SOAP [BOO04].

O princípio é similar a uma interface RMI de Java, ou a uma IDL de CORBA. Porém, WSDL é independente da linguagem de programação e é suportado pela plataforma Web [CZE03].

Por meio da linguagem WSDL são descritos os métodos e os atributos suportados pelo *Web Service*, definindo os tipos de dados e os protocolos que podem ser utilizados para o envio e recebimento de mensagens entre as aplicações. No momento da comunicação entre as aplicações, o conteúdo das mensagens SOAP é interpretado através de elementos (palavras-

chaves) da linguagem WSDL, tais como: type, message, portType, operation, binding e service.

A linguagem WSDL define somente as interfaces necessárias para a chamada dos métodos disponíveis no *Web Service*. Porém, toda a lógica do negócio e a linguagem de implementação dos métodos não fazem parte da definição. Desta forma, os serviços definidos por meio da linguagem WSDL podem ser implementados por vários tipos de linguagens de programação, plataforma, modelo de objeto ou sistema de mensagem. A simples extensão da infra-estrutura existente de Internet podem implementar *Web Service* para interação por browsers ou diretamente dentro de uma aplicação.

As aplicações podem ser implementadas usando COM, JMS, CORBA, COBOL (COMmon Business Oriented Language), ou qualquer outra solução proprietária de integração, contanto que remetente e receptor concordem na descrição do serviço (por exemplo: WSDL) [BOO04].

A Figura 6.5 apresenta um exemplo de um documento WSDL para descrever um serviço Web chamado AssistenciaSaude. Este documento será utilizado para consistir a troca de mensagens entre as aplicações, como apresentado no exemplo da anterior nas Figura 6.3 e Figura 6.4, onde foram exibidas trocas de mensagens SOAP.

<pre><?xml version="1.0"?> <definitions name="AssistenciaSaude" targetNamespace="http:// www.sitesaude.br/webservice/assistenciasaude.wsdl" xmlns:tns="http:// www.sitesaude.br/webservice/assistenciasaude.wsdl" xmlns:xsdSaude="http:// www.sitesaude.br/webservice/assistenciasaude.xsd" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns="http://schemas.xmlsoap.org/wsdl/"></pre>	Cabeçalho do documento WSDL
<pre><types> <schema targetNamespace=" http:// www.sitesaude.br/webservice/assistenciasaude.xsd" xmlns="http://www.w3.org/2000/10/XMLSchema"> <element name="CartaoSaude"> <complexType> <all> <element name="NumeroCartao" type="int"/> </all> </complexType> </element></pre>	Tipos de dados

<pre> <element name="CID10"> <complexType> <all> <element name="CodigoCid10" type="string"/> </all> </complexType> </element> </schema> </types> </pre>	
<pre> <message name="getCidPacienteRequest"> <part name="idPaciente" element="xsdSaude:CartaoSaude"/> </message> <message name="getCidPacienteResponse"> <part name="cidDiagnostico" element="xsdSaude:CID10"/> </message> </pre>	Mensagem
<pre> <portType name="AgentWebAssistenciaSaude"> <operation name="getCidPaciente"> <input message="tns:getCidPacienteRequest"/> <output message="tns:getCidPacienteResponse"/> </operation> </portType> </pre>	Tipo de Porta e Operação
<pre> <binding name="AgentWebAssistenciaSaudeBinding" type="tns:AgentWebAssistenciaSaude"> <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/> <operation name="getCidPaciente"> <soap:operation soapAction="http://www.sitesaude.br/webservice/assistenciasaude/ getCidPaciente"> <input> <soap:body use="literal"/> </input> <output> <soap:body use="literal"/> </output> </operation> </binding> </pre>	Conexão Binding
<pre> <service name="AssistenciaSaude"> <documentation>WebAgentAssistenciaSaude</documentation> <port name="AssistenciaSaudePort" </pre>	Serviço e Porta

<pre> binding="tns:AgentWebAssistenciaSaudeBinding "> <soap:address location="http://www.sitesaude.br/webservice/assistenciasaude"/> </port> </service> </pre>	
<pre> </definitions> </pre>	Fim do documento WSDL

Figura 6.5: Documento WSDL descrevendo o serviço AssistenciaSaude.

O documento WSDL é composto dos seguintes elementos utilizados para definição do serviço Web [CHR01, FAL01]:

- Cabeçalho do documento WSDL: no cabeçalho do documento é definido o nome do serviço e os documentos utilizados como referência para compor o documento WSDL;
- Tipos de dados: representa um conjunto para definições de tipos de dados utilizando algum sistema de tipos, como por exemplo, XSD (*XML Schema Definition*);
- Mensagem: defini a estrutura abstrata dos dados que serão comunicados. Uma mensagem é formada por uma ou mais partes, sendo que cada parte representa um parâmetro tipado⁷;
- Tipo de Porta e Operação: defini um conjunto abstrato de operações que são oferecidas pelo serviço e quais mensagens são suportadas pelo método, ou seja, quais mensagens de entrada e saída são utilizadas em cada operação definida;
- Conexão (Binding): especifica um protocolo de comunicação que será utilizado, por exemplo HTTP, e formatos de dados para cada tipo de porta disponível no serviço Web.
- Porta: representa um *endpoint*⁸, porta de acesso, definido com a combinação de uma conexão (*binding*) com um endereço de rede onde está localizado o serviço;

⁷ Parâmetro tipado representa uma estrutura de dados bem definida conforme um domínio de valores possíveis de serem interpretados como, por exemplo, inteiro, caractere, data, literal, etc.

⁸ *Endpoint* representa uma porta de acesso a um serviço em Web Service, identificado através de uma conexão binding e um endereço de rede (Universal Resource Identifier - URI) [BER98].

- Serviço: representa um conjunto de *endpoint* relacionados e a localização de cada um deles.

6.2.5. Processos de Descoberta, Agregação e Coordenação

Os processos de descoberta, agregação e coordenação de serviços Web ainda estão em estudo e elaboração. Existem algumas especificações que buscam definir os mecanismos destas funcionalidades, porém ainda muito difuso e sem um padrão definido. A seguir serão apresentadas as principais propostas que estão sendo aceitas por algumas das instituições de padrões e pesquisa como W3C, OASIS, WS-I, Liberty Alliance e DAML⁹.

6.2.6. Processo de Descoberta de Serviços

O processo de descoberta de serviço é o ato de localizar uma descrição de uma máquina processadora de um serviço Web relacionado a um recurso que pode ser desconhecido em um primeiro momento, mas que possua determinados critérios funcionais. Envolve combinar critérios funcionais requisitados com as descrições dos recursos externos. O objetivo é encontrar um serviço Web relacionado ao recurso de interesse [BOO04].

Esse processo de descoberta é realizado por entidades solicitantes que podem ser pessoas (usuário final) ou agentes de software, porém para que isso ocorra de forma efetiva, deve ser especificado e aceito um conjunto de padrões ainda em elaboração. A semântica da descrição das funcionalidades de um serviço pode ser tão simples como uma URI, e tão complexa quanto a descrição de meta-dados utilizados em UDDI [BOO04].

Um serviço de descoberta é usado para publicar e procurar descrições que possuem a funcionalidade correta de acordo com um critério, para que entidades solicitantes possam usufruir dos agentes provedores apropriados para realizar determinada tarefa. Para que o

⁹ Estas organizações são instituições de pesquisa de padrões aplicados em *Web Service*:

W3C: *World Wide Web Consortium* (www.w3.org)

OASIS: *Organization for the Advancement of Structured Information Standards* (www.oasis-open.org)

WS-I: *Web Service Interoperability Organization* (www.ws-i.org)

Liberty Alliance: *Liberty Alliance Consortium* (www.projectliberty.org)

DAML: *DARPA Agent Markup Language* (www.daml.org)

agente solicitante usufrua um serviço deve de alguma forma obter o endereço do agente de provedor, tipicamente pode-se obter o endereço do provedor de serviço das seguintes formas:

- a entidade solicitante pode obter o endereço do agente de provedor diretamente da entidade provedora;
- a entidade solicitante pode usar de um serviço de descoberta para localizar uma descrição de serviço que satisfaça determinado requisito;
- a entidade solicitante pode obter a partir de um parceiro ou conhecido.

A segunda opção pode ser realizada de forma manual, com interação humana, ou por seleção autônoma onde o próprio agente solicitante procura por um agente provedor de determinado serviço sem interação humana, confiando suas informações ao agente provedor. No entanto, a seleção autônoma propicia uma fragilidade de segurança, pois a própria máquina decide sobre a aceitação ou não de determinado serviço provedor.

Existem três formas conhecidas de políticas de descoberta de serviços como, Registro, Índice e Par-a-Par [BOO04]:

- A abordagem por Registro tem como objetivo centralizar em um único local ou diretório os registros de entidades e serviços, requerendo um papel ativo da entidade provedora para manter atualizado os registros de seus serviços. Embora um provedor pode dar permissão para determinadas entidades provedoras atualizarem informações em seu registro, outras entidades externas não podem publicar informações sobre os mesmos serviços, independente de ter uma melhor descrição para determinado serviço. Um exemplo de abordagem por registro é o diretório de serviços UDDI.
 - A abordagem por Índice tem como objetivo compilar informações de vários locais como um guia de serviços, as informações compiladas são adquiridas dos serviços publicados pelas próprias entidades provedoras, sem necessidade de um registro centralizado. Desta forma, as entidades provedoras de serviços podem publicar seus serviços além de poderem criar seus próprios índices. Diferentes índices poderão prover informações diferentes com melhor ou pior qualidade. As próprias entidades solicitantes (pessoas ou agentes de software) poderão utilizar qualquer índice de
-

procura. Um exemplo de abordagem por índice é o disponibilizado no Site Google (www.google.com).

- A abordagem Par-a-Par, também conhecida como P2P, tem como objetivo localizar determinado serviço, comunicando uma solicitação para cada máquina da rede (nó de rede) de forma dinâmica até encontrar um provedor do serviço que satisfaça os requisitos definidos pela entidade solicitante. Neste caso, cada entidade (nó de rede) deve prover mecanismos de comunicação para solicitar ou responder por solicitações de serviços. Como a pesquisa de determinado serviço é realizada em cada nó da rede e caso não encontrado é propagado para outros nós da rede, acaba sendo ineficiente devido ao grande número de conexões necessárias na tentativa de localizar um serviço, diminuindo o desempenho da rede.

As abordagens de localização de serviços Web diferem em seus mecanismos de ação, a abordagem de registro busca um controle maior na descrição de serviços, sendo de responsabilidade do proprietário do registro atualizar e manter as informações, enquanto a abordagem de índice busca a flexibilidade de escolha entre diversos índices permitindo a seleção pelas entidades solicitantes, por fim a abordagem Par-a-Par oferece como vantagem localizar um serviço com uma descrição atualizada pois se comunica diretamente com a entidade provedora, além de não possuir um único meio de localização como ocorre com a abordagem de registro com informações centralizadas, porém, com baixo desempenho devido a grande quantidade de conexões que ocorrem entre os nós de rede durante a localização de um serviço.

Atualmente a abordagem de registro é a mais utilizada, considerando o desempenho e o controle dos serviços disponíveis, sendo um dos principais mecanismos utilizados atualmente para localização de serviços Web chamado UDDI [CLE04]. A seguir serão apresentados a forma de registro UDDI e os elementos que compõem este registro. No entanto, existem outros mecanismos de localização de serviços como RDF (*Resource Description Framework*) especificado pela organização W3C e OWL-S (*Ontology Web Language - Semantic*) também chamado de DAML-S (*DARPA Agent Markup Language - Semantic*) especificado pela organização DARPA (*Defense Advanced Research Projects Agency*) ([MAN04], [DAR04]).

6.2.7. Universal Description, Discovery and Integration (UDDI)

O UDDI é um tipo de abordagem de registro onde implementa um diretório de serviços Web com mecanismos de registro e publicação de *Web Service*, possibilitando descobrir e localizar os serviços registrados por organizações diferentes, fornecendo informações sobre o serviço e sua interface [CLE04]. O principal objetivo do diretório UDDI (www.uddi.org) é facilitar a localização de serviços que podem ser integrados a outras aplicações [NOV04b].

A especificação de UDDI descreve como os dados devem ser armazenados em um registro, os métodos possíveis para publicação e mecanismos de busca e categorização desses registros. Os dados incluem os contatos da organização, como se comunicar, lista de serviços disponíveis e como usá-los. As interações com os registros são realizadas utilizando SOAP como protocolo para troca das mensagens.

Os registros UDDI funcionam de forma análoga a uma “lista telefônica”, com páginas brancas onde estão informações da organização, páginas amarelas com a categorização dos tipos de serviços e páginas verdes com especificações detalhadas dos serviços [UDD01]. O funcionamento é similar ao “repositório de objetos” utilizado em CORBA e LDAP (*Lightweight Directory Access Protocol*) definido pela IETF (*Internet Engineering Task Force*).

Desta forma, após a localização de um serviço e identificação da estrutura do serviço, pode ser acoplado na aplicação cliente os serviços disponíveis no provedor *Web Service*.

A combinação de WSDL e UDDI permite publicar um serviço para que seja descoberto por possíveis usuários. No entanto, a publicação do serviço é opcional e ainda é pouco utilizado. A maior parte dos *Web Services* presentes na rede estão apenas disponíveis para um seleto número de usuários, isso porquê na maioria dos casos os serviços são destinados aos parceiros de negócio ou clientes e não ao público em geral [CZE03].

A estrutura do registro UDDI é constituída de quatro elementos principais: *businessEntity*, *businessServices*, *bindingTemplate* e *tModel*. Sendo que o conjunto destas partes define o modelo de dados do UDDI. A Figura 6.6 apresenta um exemplo de registro UDDI para publicar um serviço de *AssistenciaSaude* de uma determinada organização saúde [NOV04b].

<pre> <businessEntity businessKey="0076b468-eb27-42e5-ac99-9955cff462a3" operator="Microsoft Corporation" authorizedName="Martin Kholleppel"> <name>MicrosoftCorporation</name> <description xml:lang="en">Empoweringpeoplethroughgreatsoftware–anytime, anyplaceandonanydeviceis Microsoft’s vision. </description> <businessServices> ... </businessServices> <identifierBag>... </identifierBag> <contacts> <contact useType="CorporateAddressesandtelephone"> <descriptionxml:lang="en">CorporateMailingAddresses</description> <personName>Microsoft</personName> <phone useType="CorporateHeadquarters">(425) 882-8080</phone> </contact> </contacts> <categoryBag> </categoryBag> </businessEntity> </pre>	<ul style="list-style-type: none"> • Business Entity
<pre> <businessService serviceKey="d5921160-3e16-11d5-98bf-002035229c64" businessKey="ba744ed0-3aaf-11d5-80dc-002035229c64" <name>XmethodsDelayedStockQuotes </name> <description xml:lang="en">20-minute delayedstockquotes </description> <bindingTemplates> <bindingTemplate serviceKey="d5921160-3e16-11d5-98bf-002035229c64" bindingKey="d594a970-3e16-11d5-98bf-002035229c64" <description xml:lang="en">SOAP bindingfor delayedstockquotesservice </description> <accessPoint URLType="http">http://services.xmethods.net:80/soap </accessPoint> <tModelInstanceDetails> <tModelInstanceInfo tModelKey="uuid:0e727db0-3e14-11d5-98bf- 002035229c64"/> </tModelInstanceDetails> </bindingTemplate> </bindingTemplates> <categoryBag> </categoryBag> </businessService> </pre>	<ul style="list-style-type: none"> • Business Service • Binding Template
<pre> <tModel tModelKey="uuid:0e727db0-3e14-11d5-98bf-002035229c64" </pre>	<ul style="list-style-type: none"> • Tmodel

<pre> operator="www.ibm.com/services/uddi" authorizedName="0100001QS1"> <name>XmethodsSimpleStockQuote </name> <description xml:lang="en">Simplestockquoteinterface </description> <overviewDoc> <description xml:lang="en">wsdlink </description> <overviewURL>http://www.xmethods.net/tmodels/SimpleStockQutoe.wsdl </overviewURL> </overviewDoc> <identifierBag>... </identifierBag> <categoryBag> <keyedReference tModelKey="uuid:c1acf26d-4404-9d70-39b756e62ab4" keyName="uddi-org:types" keyValue="wsdlSpec"/> </categoryBag> </tModel> </pre>	
--	--

Figura 6.6: Estrutura do Registro UDDI de uma organização.

- *Business Entity*: é o primeiro elemento da estrutura de dados UDDI, ele identifica a organização que provê serviços na rede através de um identificador único (*BusinessKey*). Neste escopo são descritas as informações da organização e os serviços que elas oferecem, fornecendo também os nomes dos negócios em outros idiomas e detalhes de contato.
- *Business Service*: representa uma coleção de serviços disponíveis na Internet fornecidos por uma organização ou entidade. Este elemento pode ser usado para se agrupar e categorizar serviços baseados no tipo de serviços oferecidos. Provém também a identificação do serviço (*serviceKey*) que é um identificador único para o serviço, o nome do serviço e sua descrição. Cada serviço deve pertencer a uma entidade organizacional.
- *Binding Template*: codificam informações técnicas necessárias para acesso ao serviço, provendo a conexão entre a aplicação cliente e o provedor do serviço. Cada *BindingTemplate* representa um ponto de acesso ao serviço Web (*bindingKey*) e pertence a uma entidade de serviço que por sua vez pertence a uma entidade organizacional.
- *Tmodel*: tem como objetivo fornecer informações técnicas de um *Web Service*, direcionando a aplicação cliente para o *Web Service* através de uma URL, para obter informações técnicas do meta-dados e a chave de acesso ao serviço. Tmodel são usados

para descrever um modelo técnico que represente um conceito reutilizável como um tipo de serviço, um protocolo de serviço ou categoria de sistema, utilizando-se de identificadores de esquema para classificação baseado em uma taxonomia. Tmodel são aplicados na própria especificação UDDI. Cada *Tmodel* possui um identificador único (*TModelKey*).

Cada um dos elementos da estrutura do registro UDDI possui um identificador único chamado UUID (*Universally Unique Identifier*). Estes identificadores são chaves únicas constituídas de caracteres hexadecimais. O objetivo é localizar uma organização, um serviço, um modelo de conexão ou de meta-dado, de forma unívoca.

Estrutura de Funcionamento para Invocação através do UDDI

Um *Web Service* publicado é representado por um modelo com uma estrutura *bindingTemplate* registrado em um diretório UDDI. Após a aplicação cliente obter o *bindingTemplate* do *Web Service* e armazená-lo localmente, a invocação ao serviço é realizada utilizando como referência os dados da estrutura, procedendo da seguinte forma ([CLE04], [NOV04b]):

1. Localiza o registro do UDDI por meio de alguma ferramenta com interface UDDI e obtém as informações da organização (*businessEntity*).
2. Identifica o serviço desejado (*businessService*) e armazena localmente.
3. Através do *tModel* contido no *bindingTemplate*, o programa cliente pode preparar para invocar os serviços publicados.
4. No momento da invocação do *Web Service*, é feita a invocação do serviço utilizando as informações do *bindingTemplate* armazenado localmente.

Em caso de falha, através do *bindingKey* pode ser obtida uma cópia atualizada do *bindingTemplate* do serviço para comparar a informação nova com a antiga, se for diferente, substituir a informação antiga pelo novo *bindingTemplate* obtido.

6.2.8. Processo de Agregação e Coordenação

O processo de agregação e coordenação tem como objetivo descrever o fluxo de mensagens entre serviços Web que participam de um processo (*workflow*), definindo a semântica do serviço, independente de sua implementação [BOO04]. Descreve como o serviço Web participa da colaboração, quais estados podem alcançar, além dos serviços que fornece, considerando a lógica do negócio e a ordem de execução de atividades, a fim de controlar as interações entre os serviços que participam de determinada tarefa [DAM04].

O processo de agregação e coordenação define uma sucessão de condições sobre os quais diversos agentes trocam mensagens para atingir determinado objetivo:

- utiliza-se de um ou mais serviços conectados;
- define-se as interações possíveis entre os agentes;
- expressa-se a coordenação em uma determinada Linguagem de Descrição de Coordenação;
- define-se a relação entre as mensagens trocadas e as tarefas de determinado serviço;
- descreve-se os papéis e as associações entre os serviços Web;
- estabelece-se um acoplamento entre os serviços disponíveis na Web.

O processo de agregação e coordenação descreve como *Web Services* podem interagir com outros agentes por meio de mensagens, incluindo a lógica de negócio e a seqüência de execução das interações entre eles. Isto possibilita estender aplicações e organizações num modelo de processo transacional.

A coordenação de serviços Web pode ser usada para descrever a criação de processos de negócios que executam ou colaboram para executar determinada tarefa [PEL03]. O processo de coordenação deve ser dinâmico e flexível, separando a lógica do processo dos serviços usados. Essa separação pode ser realizada pelo mecanismo de coordenação que manipula o fluxo do processo, invocando os serviços apropriados e determinando os próximos passos a serem completados.

A especificação das mensagens de interação relacionadas com uma atividade deve suportar operações para chamar, suspender, recomeçar, parar e terminar um processo de negócio, assim como consultar o estado em que o mesmo se encontra [SNE02].

Existem algumas propostas de linguagem de descrição de agregação e coordenação de serviços como WSCI (*Web Service Choreography Interface*) proposto pela Sun, SAP, BEA e Intalio, WSFL (*Web Service Flow Language*) proposto pela IBM e BPEL4WS (*Business Process Execution Language for Web Services*) proposto pela Microsoft, BEA e IBM ([ARK02], [LEY01], [EBP04]).

Um aspecto importante da especificação da interface de um serviço é o padrão de troca de mensagens que o serviço suporta. Para casos simples, este padrão normalmente interação direta. Porém, para a maioria dos casos reais o processo de coordenação de serviços pode ser muito complexo.

Com objetivo de promover mecanismos de agregação e coordenação de serviços Web está sendo elaborada uma semântica Web. Estudos e propostas estão sendo realizados, mas ainda não tem um padrão definido e aceito para agregação e coordenação de serviços Web. A W3C criou um grupo *Web Services Choreography Working Group* para avaliar as propostas de especificações a fim de definir um padrão [KAV04].

Capítulo 7 - Proposta de um Sistema de Agentes Orientado a Serviço para o SUS

A modelagem da arquitetura do sistema será representada através da linguagem AUML (Linguagem de Modelagem Unificada para Agente), buscando representar os agentes de software para o contexto da saúde pública.

7.1. A UML - Linguagem de Modelagem Unificada para Agente

A modelagem de sistema é um dos primeiros passos na construção de um produto de software. Desta forma, através do processo de análise são definidos os requisitos para solução de um problema. Estes requisitos são os subsídios para projetar a lógica da solução do problema, através da representação simbólica descrita em uma determinada metodologia [LAR00].

A Linguagem UML (Linguagem de Modelagem Unificada), é uma notação para modelagem de sistemas que utiliza conceitos da metodologia orientada a objetos [OMG05, LAR00]. Como a arquitetura proposta tem como abordagem sistemas de agentes de software, serão modelados os agentes desta arquitetura através da linguagem A UML (Linguagem de Modelagem Unificada para Agente). Esta linguagem está sendo avaliada e especificada, utilizando como principal referência os conceitos e definições da linguagem UML.

A FIFA (*Foundation for Intelligent Physical Agents*) criou um grupo de trabalho para avaliar e estabelecer um padrão de modelagem de agentes de software, utilizando como referência à linguagem UML [FIP03a]. Este grupo de trabalho está subdividido em áreas de trabalho com objetivos específicos conforme apresentado logo abaixo, a fim de constituir uma extensão da linguagem UML denominada A UML (Agent Unified Modeling Language) [FIP05].

7.1.1. Modelagem UML estendida para agentes

Os diagramas que serão apresentados na sequência são derivados da linguagem UML e aplicados na modelagem de agentes, apresentam-se poucas modificações em relação ao padrão especificado em UML [MAC05]:

- Diagrama de Classe – especificação do comportamento interno e externo de um agente;
- Diagramas de Interação – representação das interações entre os agentes e objetos, modelado através dos diagramas:
 - Diagrama de Colaboração;
 - Diagrama de Seqüência;
- Diagrama de Componente - especificado conforme o Diagrama de Classe;
- Caso de Uso – representado conforme especificado na UML;
- Planejamento e Fluxo de Trabalho (*workflow*) – representação do fluxo de planejamento de execução, representado pela extensão do Diagrama de Atividades.

7.1.2. Modelagem sobre novo aspecto no contexto de agentes

Os aspectos citados a seguir não estão contemplados na linguagem UML, buscam representar características específicas no contexto de agentes que não é o foco da linguagem UML, pois o principal contexto da linguagem UML é a metodologia orientada a objetos [FIP03a, FIP05]:

- Multi-Agente e Agente Individual – nível multi-agente, representando as interações e relacionamentos entre os agentes; e nível individual representando a estrutura interna do agente através de tarefas e comportamento.
- Objetivos / Metas de Software – representação das metas dos agentes dentro de um contexto.
- Aspectos Sociais – rerepresentação dos aspectos sociais do agente em uma comunidade de agentes, representação da estrutura de organizações autônomas (espontâneas).
- Diagrama de Ambiente – definição das propriedades do ambiente considerando aspectos físicos e de comunicação.

- Serviços – representação dos serviços de uma organização, conjugando os agentes no contexto do ambiente.
- Níveis de abstração do Agente – nível de abstração do agente podendo considerar a concepção do agente no ambiente, a composição atual ou forma de implementação do agente (classes e estruturas de dados). Abordando também o relacionamento entre os níveis de abstração do agente.
- Regras temporais: Definição de agenda e limites de tempo para execução de interações.
- Diagrama de Mobilidade – Modelagem e implementação da mobilidade de agentes abrangendo análise e projeto.

A linguagem AUML está sendo elaborada com objetivo de criar mecanismos padrões para modelagem de sistemas no contexto de agentes de software. No entanto, os diagramas e documentos sugeridos pelo grupo de trabalho AUML ainda estão sendo propostos a avaliados, podendo ser aplicados conforme a necessidade do contexto, não sendo obrigatória à representação de todos os diagramas [FIP05].

Os agentes da arquitetura proposta serão representados através dos diagramas de caso de uso, diagrama de seqüência, diagrama de colaboração, diagramas de atividades e diagrama de estados, conforme especificado pela AUML ([FIP05], [MAC05]).

7.2. Modelagem do contexto Epidemiológico e de Assistência em Saúde.

Foram apresentados anteriormente os sistemas de informação desenvolvidos para suprir as necessidades de gestão e operação do SUS. Entre os sistemas existentes, serão abordados os sistemas legados que compreendem maior relevância na gestão do SUS, abrangendo as áreas de assistência em saúde e epidemiológica [MIN98]. Um dos objetivos básicos do sistema de informação em saúde na concepção do SUS é possibilitar a vigilância, planejamento, avaliação e controle sobre o processo de saúde-doença da população. Neste contexto, observa-se a amplitude e complexidade da Saúde Pública, sendo necessário segmentar o conhecimento e as ações de saúde em áreas específicas, suportadas por sistemas de informação caracterizados a área a fim, ou seja, mesmo tratando-se de informações que

estão correlacionadas, o conhecimento deve ser representado em contextos específicos de cada setor da Saúde Pública.

7.2.1. Vigilância Epidemiológica

Compete a vigilância epidemiológica o acompanhamento de doenças e agravos de saúde, monitoramento da mortalidade, natalidade e imunização, a fim de promover ações para promoção da saúde coletiva da população [LOU03]. Os sistemas de gestão, utilizados na área epidemiológica, são estruturados para atender as três esferas de governo: Município, por meio das Secretarias Municipais de Saúde; Estado, pelas Secretarias Estaduais de Saúde; e Federal, pelo Ministério da Saúde em parcerias com as secretarias e agências nacionais de saúde [SEN88b].

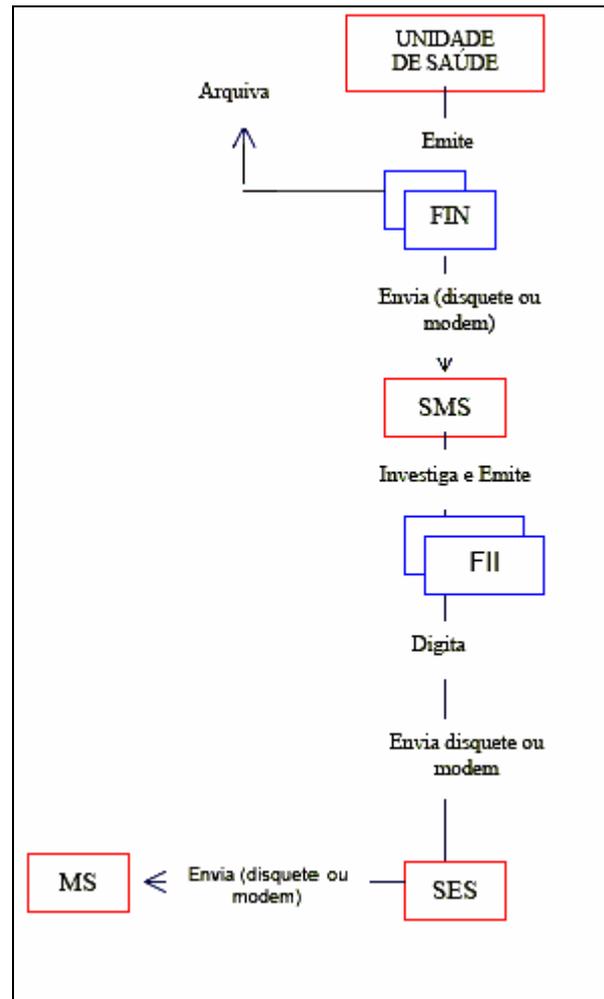
Os sistemas da área epidemiológica que serão abordados são:

- SINAN – Sistema de Informação de Agravos de Notificação
- SINASC – Sistema de Informação de Nascidos Vivos
- SIM – Sistema de Informação de Mortalidade

SINAN – Sistema de Informação de Agravos de Notificação

O sistema SINAN é um sistema da gestão epidemiológica utilizado para notificar agravos de saúde (doenças). Constituído com objetivo de análise e acompanhamento epidemiológico sobre agravos de interesse, ou seja, agravos de notificação obrigatória, definidos pelas três esferas de governo (municipal, estadual e federal).

As informações sobre agravos de saúde originam-se nas unidades de saúde que prestam atendimento à população, públicas ou privadas, as quais encaminham uma ficha individual de notificação (FIN) para a secretaria municipal de saúde, que tem a competência de avaliar e investigar a notificação, utilizando uma ficha de investigação individual (FII), com objetivo de confirmar o agravo de saúde. Posteriormente, são repassados os dados para secretaria estadual de saúde e ministério da saúde, conforme apresentado no diagrama da Figura 7.1:



Fonte: CENEP\FNS\MS - 1998

Figura 7.1: Fluxo operacional do sistema SINAN.

Na atividade de controle epidemiológico utiliza-se o sistema SINAN como instrumento de coleta e de análise de dados. Este sistema está relacionado aos serviços de assistência de saúde executados pelas unidades ambulatoriais e hospitalares, pois os dados de notificação são oriundos dos atendimentos da assistência à população. Porém, atualmente o sistema está estruturado de forma isolada, sem interação com os sistemas que oferecem suporte operacional para assistência de saúde, restringindo controle efetivo de notificações e aumentando a possibilidade de inconsistência das informações.

Com objetivo de integrar os segmentos da Saúde Pública, foi encapsulado o sistema SINAN como um Agente de Software, representando as competências do sistema e propondo uma nova abordagem de interação com os demais sistemas. No diagrama de caso de uso apresentado na

Figura 7.2, observa-se as interações entre os sistemas no contexto epidemiológico para controle de morbidade (doenças):

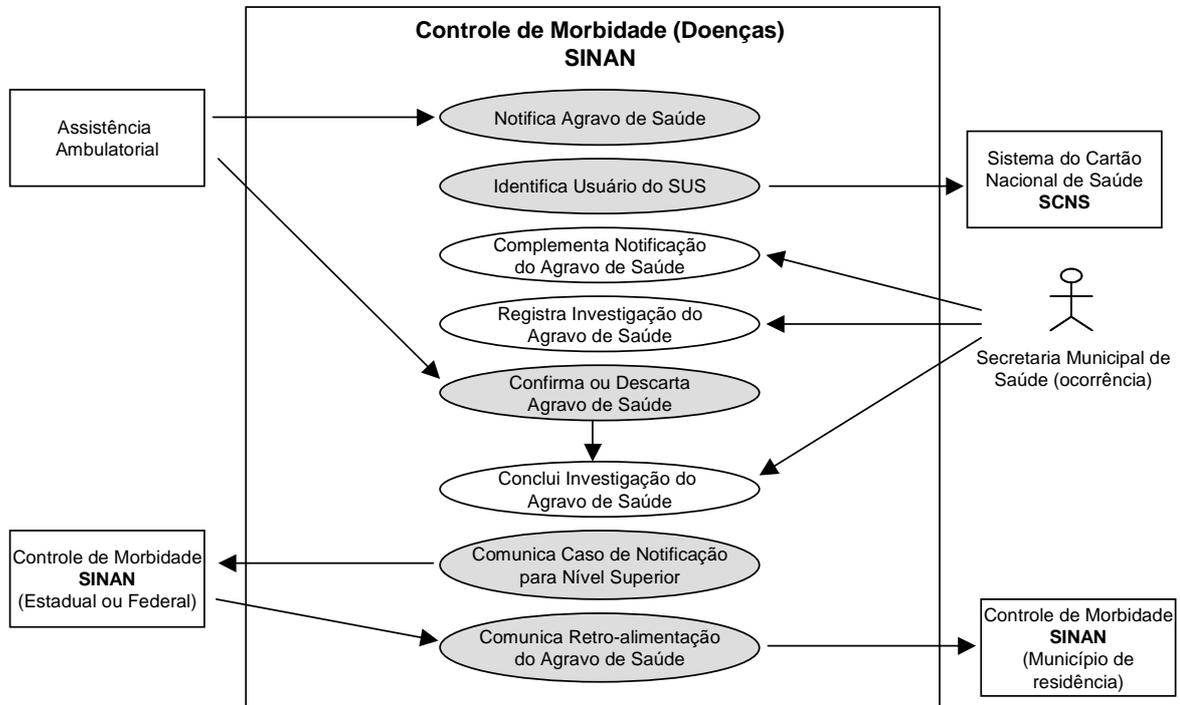


Figura 7.2: Diagrama de caso de uso do controle de morbidade (SINAN).

Neste diagrama observa-se às interações necessárias entre os agentes para realizar o acompanhamento de notificações de morbidade. As competências do agente de controle de morbidade são especificadas considerando a ação colaborativa com os demais agentes, sintetizando as capacidades sob responsabilidade de cada agente. As competências definidas para o Agente SINAN são abordadas como serviços que o agente oferece ou suporta, conforme apresentado na Figura 7.3 onde é representado o protocolo definido para o contexto epidemiológico do SUS.

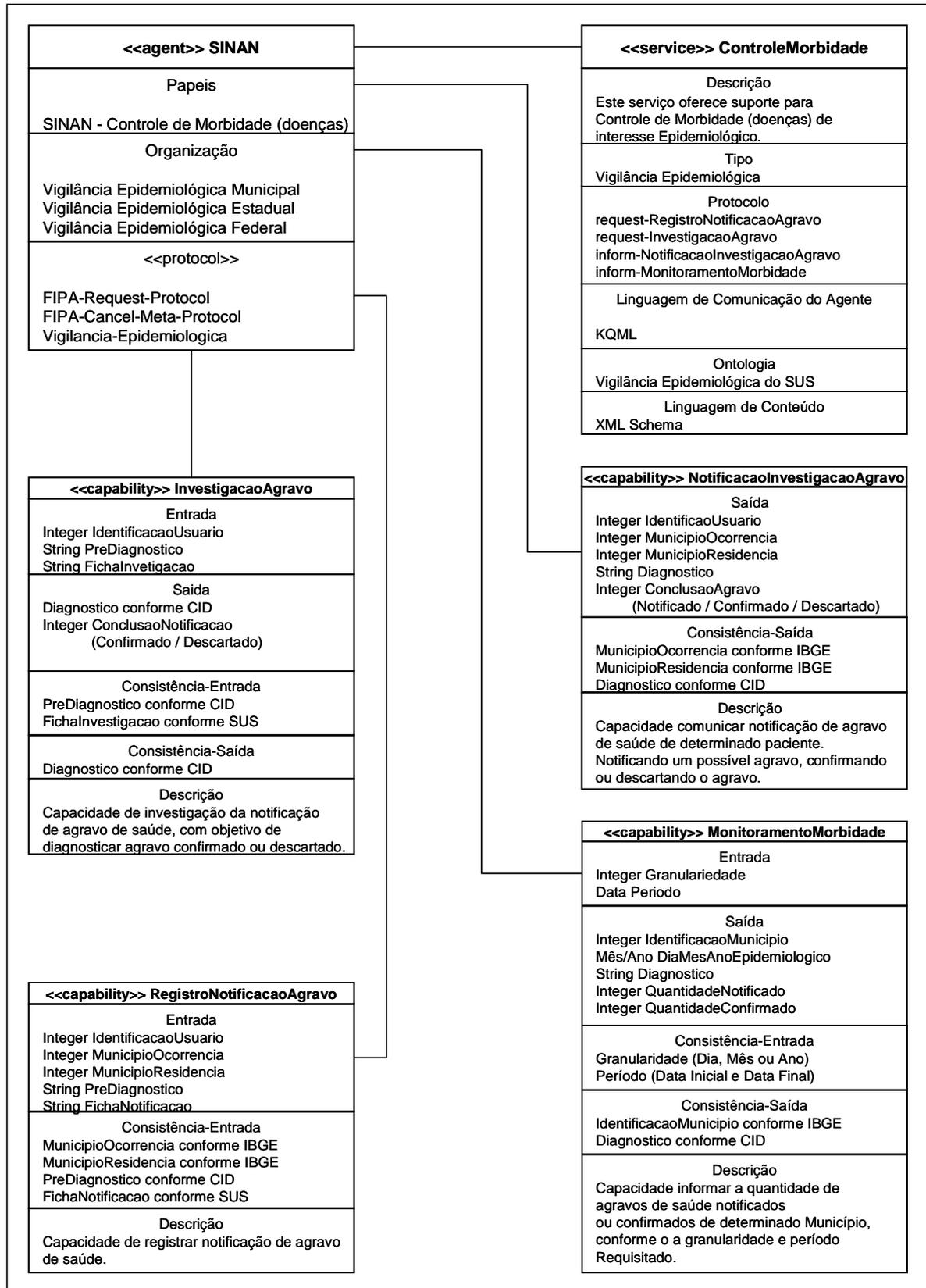


Figura 7.3: Diagrama de classe do agente SINAN.

No diagrama de classe do agente SINAN, observam-se as seguintes capacidades:

- complementar e registrar notificação de agravo de saúde informada pela assistência ambulatorial;
- investigar o agravo notificado, confirmando ou descartando o caso;
- comunicar a notificação ou investigação do agravo de saúde para o nível superior (estadual ou federal) ou para o nível inferior como retro-alimentação (municipal);
- informar sobre o monitoramento de morbidade (doenças).

SINASC – Sistema de Informação de Nascidos Vivos

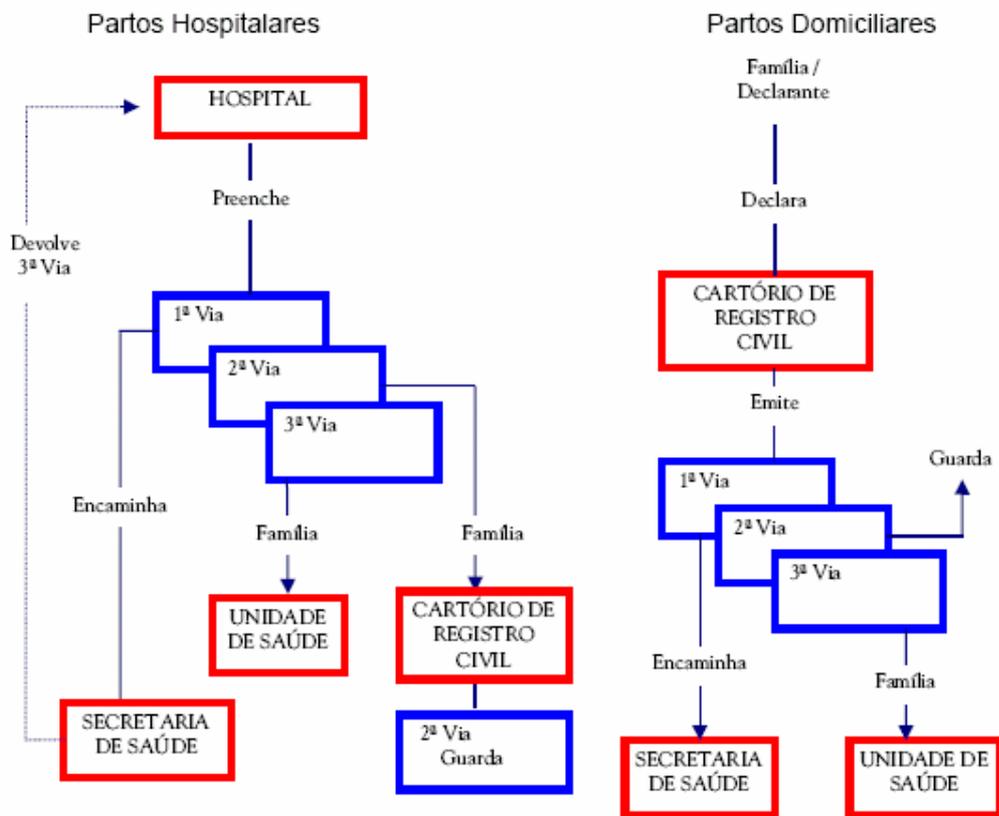
O sistema SINASC é um sistema da gestão epidemiológica, constituído com objetivo de acompanhamento da natalidade, provendo a avaliação das ações de saúde através da análise de indicadores epidemiológicos que representam informações sobre mortalidade infantil, mortalidade materna, assistência no pré-natal, prematuridade, fecundidade, gravidez precoce, condições do parto, etc.

A principal fonte de declaração de nascimento é oriunda da assistência hospitalar, sendo responsável por aproximadamente 97% do total de declarações de nascidos, conforme apresentado na Tabela 7.1:

Tabela 7.1: Nascidos por local de ocorrência e região referente ao ano 2002 [MIN05b].

Local de Ocorrência	Norte	Nordeste	Sudeste	Sul	C. Oeste	Total
Hospital	274.291	869.785	1.185.593	402.067	226.137	2.957.873
Outro Estabelecimento de Saúde	10.843	35.768	4.934	2.012	590	54.147
Domicílio	15.520	22.944	3.369	1.734	811	44.378
Outro	206	1.343	475	236	128	2.388
Ignorado	6	170	424	8	8	616
Total	300.866	930.010	1.194.795	406.057	227.674	3.059.402

As ocorrências de nascimento em outros locais como domicílio, por exemplo, representam um número pequeno, neste caso a fonte de declaração utilizada é o cartório de registro civil. Na Figura 7.4 é apresentado o fluxo atual de operação do sistema, utilizado para manter as informações pertinentes ao sistema SINASC:



Fonte: Ministério da Saúde, Manual de Instruções para o Preenchimento da Declaração de Nascido Vivo, 1996

Figura 7.4: Fluxo operacional de Declaração de Nascido Vivo.

Constituindo o sistema SINASC como um agente de software, pode ser aprimorada a interação com os demais agentes de assistência hospitalar, possibilitando mecanismos mais eficientes para acompanhamento da natalidade.

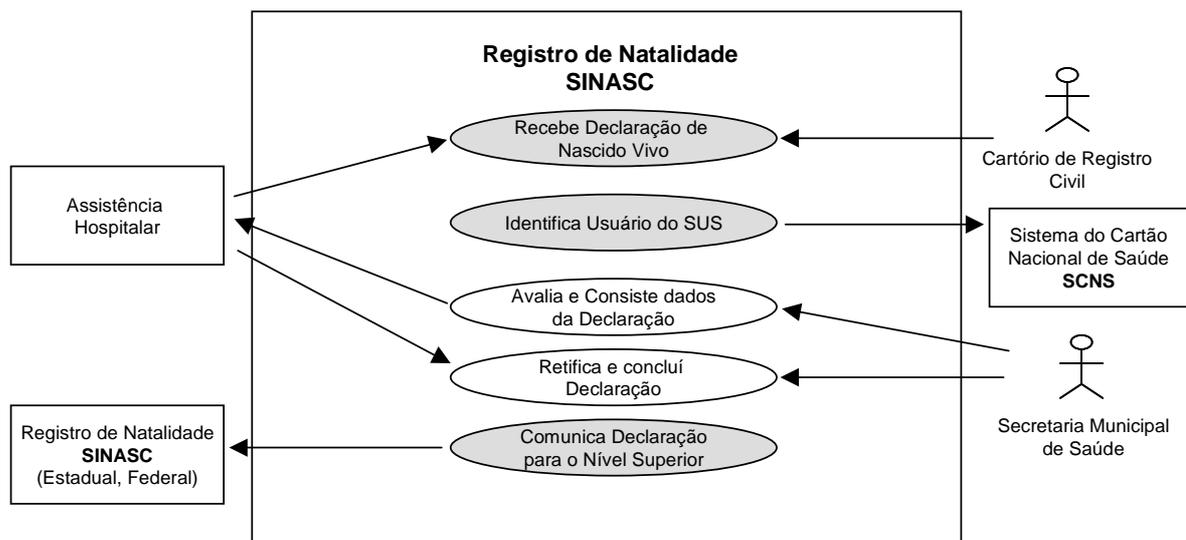


Figura 7.5: Diagrama de caso de uso do registro de natalidade (SINASC).

O diagrama de caso de uso apresentado anteriormente na Figura 7.5, aborda as interações necessárias entre os agentes no acompanhamento de natalidade.

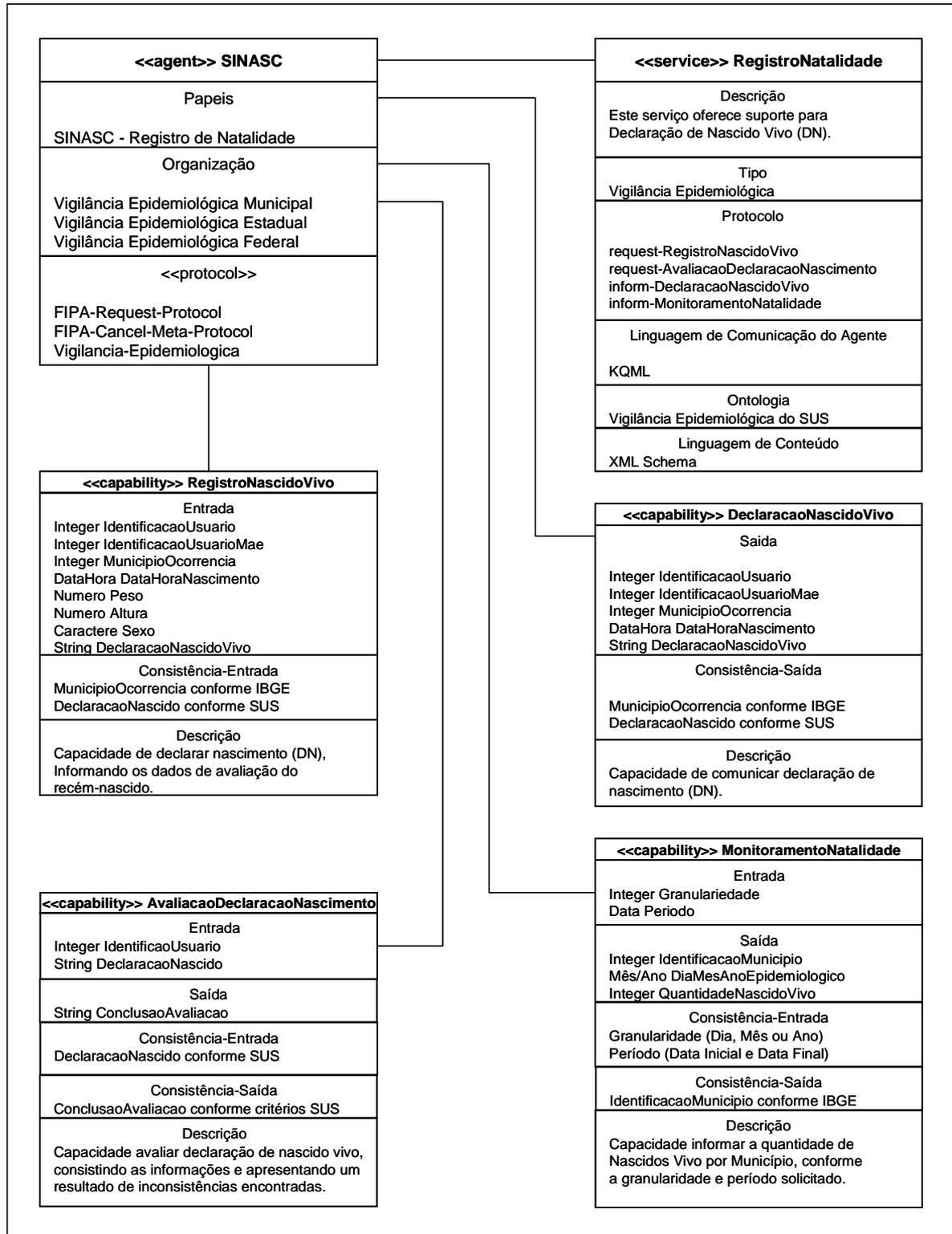


Figura 7.6: Diagrama de classe do agente SINASC

As competências do agente SINASC são definidas conforme modelado no diagrama de classe do agente SINASC apresentado na Figura 7.6. Entre as atividades realizadas neste contexto, observam-se as seguintes capacidades do agente SINASC:

- registrar declaração de nascimento (DN), informando os dados de avaliação do recém-nascido;
- avaliar declaração de nascido vivo, consistindo as informações e apresentando um resultado de inconsistências encontradas;
- comunicar declaração de nascimento (DN);
- informar sobre o monitoramento de nascidos vivos e as condições do nascimento.

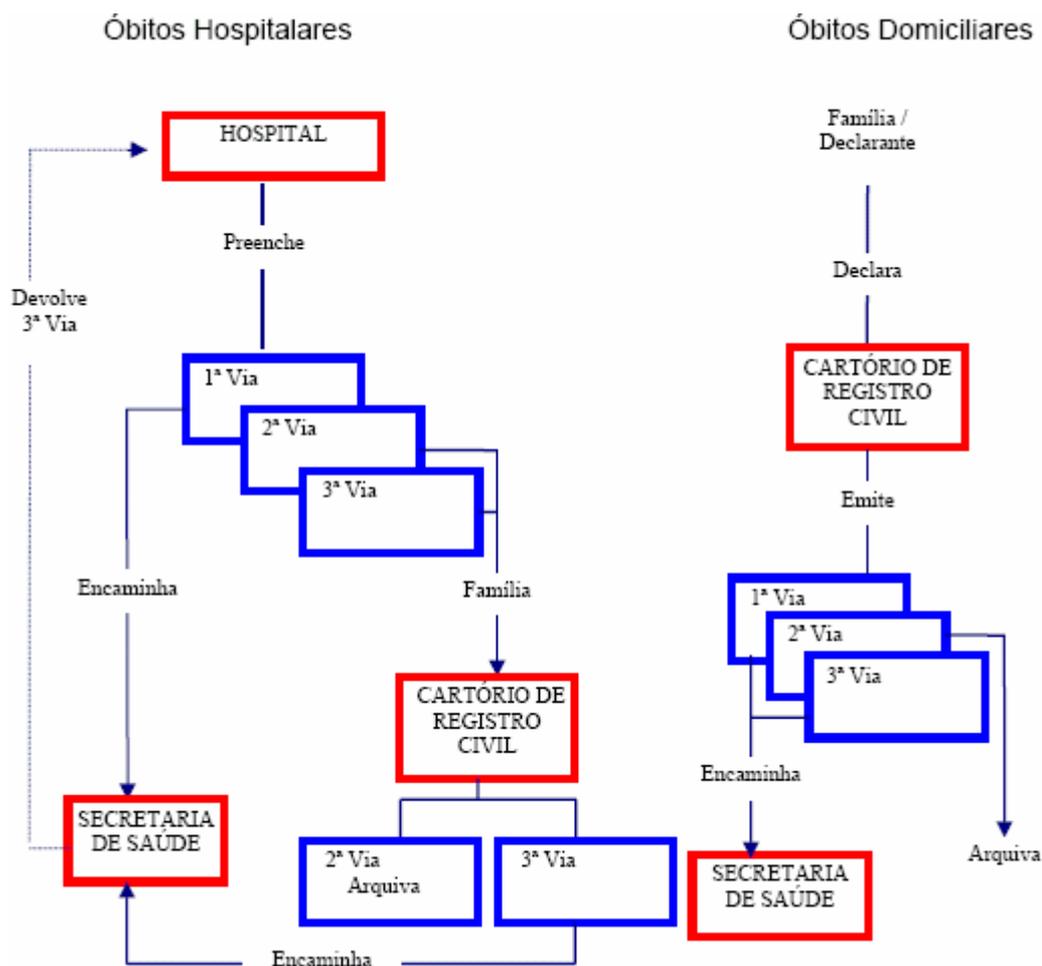
SIM – Sistema de Informação de Mortalidade

O sistema SIM é um sistema da gestão epidemiológica, constituído com objetivo de acompanhamento de mortalidade, servindo como meio de avaliação sobre as causas de óbitos. Este sistema oferece suporte para análise de indicadores epidemiológicos, oferecendo subsídio as diversas esferas de gestão para a análise de situação, planejamento e avaliação das ações e programas de saúde, conduzindo ao conhecimento do perfil saúde-doença da população.

As fontes de dados do sistema são oriundas da assistência de saúde, principalmente hospitalar, e em casos de óbito por causas externas acidentais ou violentas e sem assistência médica, utiliza-se do Instituto Médico Legal (IML) e nos demais casos de óbitos naturais sem assistência médica são comunicados através de cartório de registro civil.

Na

Figura 7.7 apresenta-se o fluxo de operação atual do sistema, proposto pelo ministério da saúde. No entanto, alguns estados ou municípios podem utilizar um fluxo alternativo ao proposto, dependendo das condições de operação de cada região.



Fonte: Ministério da Saúde, Manual de Instruções para o Preenchimento da Declaração de Óbito, 1996

Figura 7.7: Fluxo operacional de Declaração de Óbito

A partir deste fluxo, foram definidas as principais interações que ocorrem no processo de registro de mortalidade, constituindo as interações necessárias entre os agentes da assistência em saúde e da vigilância epidemiológica, conforme o diagrama de caso de uso apresentado na Figura 7.8:

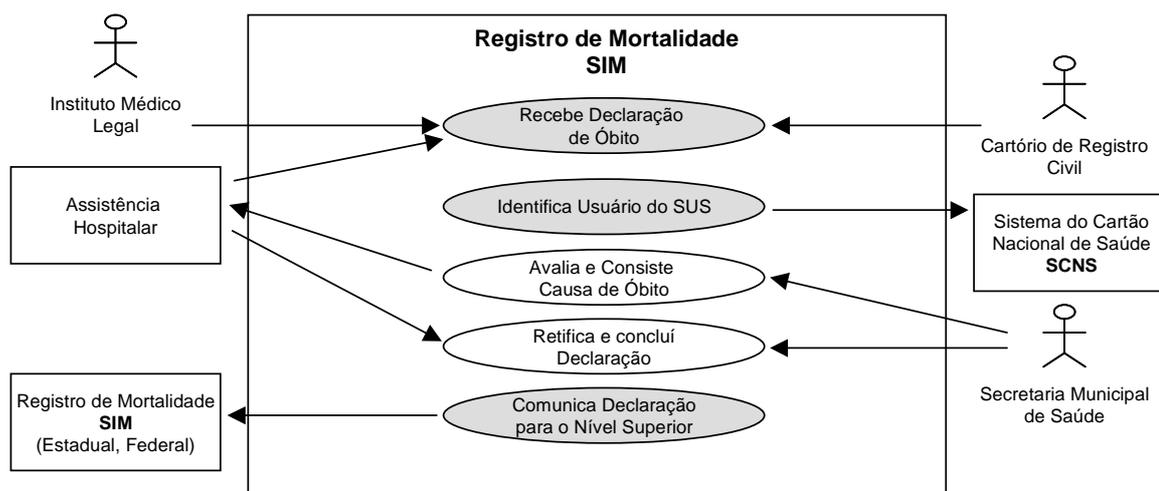


Figura 7.8: Diagrama de caso de uso do registro de óbito (SIM).

As capacidades do agente SIM que pertence rede da vigilância epidemiológica são especificadas considerando as interações com os demais agentes da rede de assistência em saúde (hospitalar) e de suporte a gestão (SCNS).

Neste contexto, observa-se as capacidades do agente SIM, conforme especificadas no diagrama de classe de agente apresentado na

Figura 7.9, constituindo as seguintes competências:

- registrar declaração de óbito (DO), informando os dados da ocorrência do óbito;
- avaliar declaração de óbito, consistindo as informações (causa óbito) apresentando um resultado de inconsistências encontradas;
- comunicar declaração de óbito (DO);
- informar sobre o monitoramento de óbitos e as condições do registro do óbito.

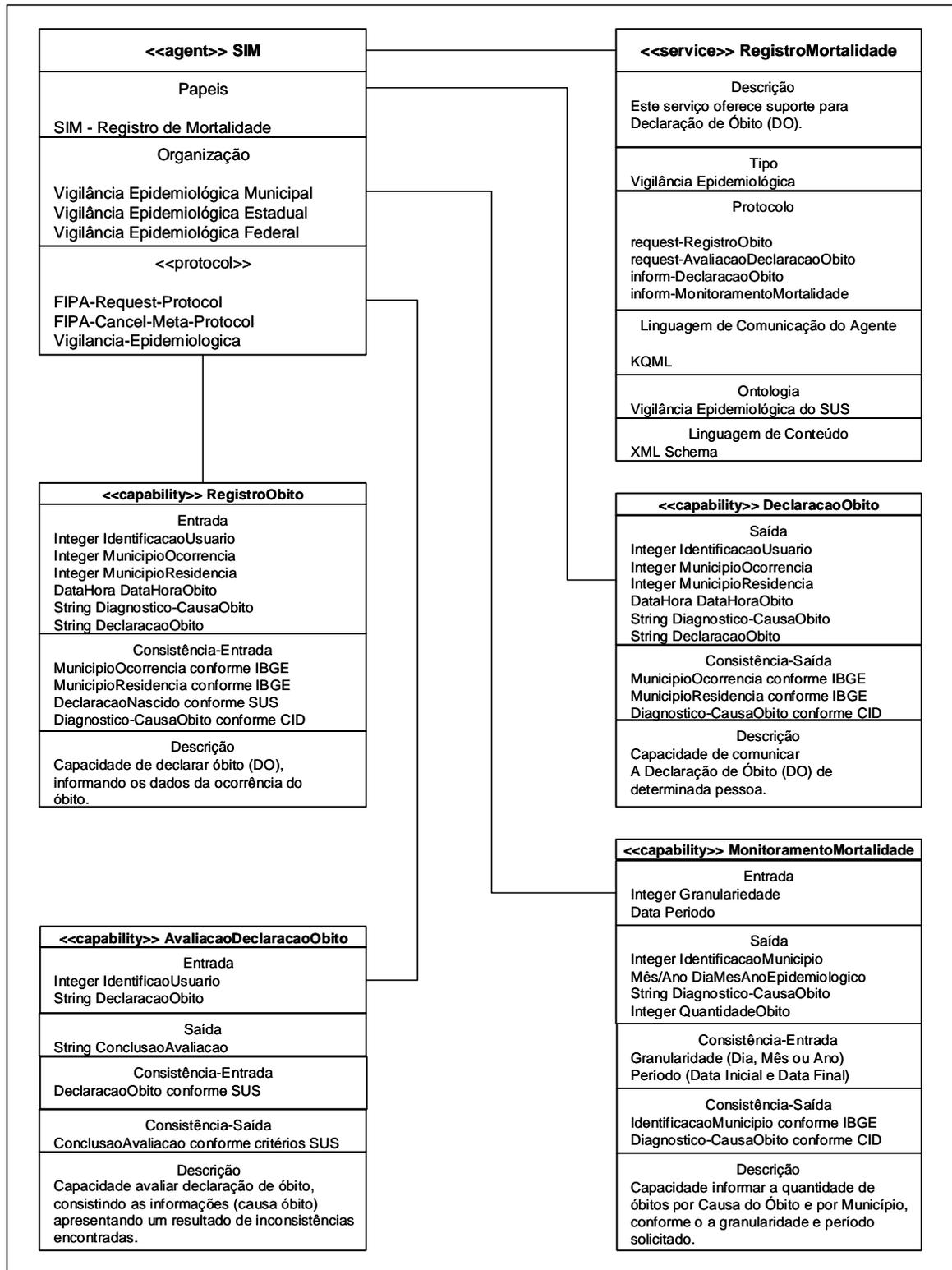


Figura 7.9: Diagrama de classe do agente SIM.

7.2.2. Assistência em Saúde

Compete à área de assistência em saúde prover serviços de atendimento a saúde curativa e preventiva das pessoas. Desta forma, são organizados os serviços de saúde entre organizações públicas e privadas (conveniadas) ao SUS, com objetivo de prover os serviços de assistência médica hospitalar e ambulatorial, assistência pré-hospitalar, assistência farmacêutica e assistência laboratorial.

Em média, aproximadamente 13% dos procedimentos ambulatoriais são realizados por estabelecimentos de saúde privados, isso porque a maior parte dos estabelecimentos de saúde que prestam serviços ambulatoriais, como por exemplo às unidades básicas de saúde, pertence ao SUS. No entanto, aproximadamente 58% das internações hospitalares são realizadas por estabelecimentos de saúde privados conveniados ao SUS, isso porque a rede pública de saúde dispõe de poucos hospitais, neste caso utiliza-se bastante da assistência complementar ao SUS, conforme apresentado na Tabela 7.2 e Tabela 7.3 [MIN05c]:

Tabela 7.2: Quantidade de procedimentos ambulatoriais por tipo de prestador e ano [MIN05c].

TIPO PRESTADOR	ANO			
	2.000	2.001	2.002	2.003
Público Federal	13.597.160	13.356.401	12.177.383	29.158.185
Público Estadual	209.531.268	241.522.872	245.169.507	304.561.354
Público Municipal	1.058.274.940	1.173.727.926	1.249.097.968	1.364.930.636
Universitários Públicos	63.135.678	67.989.633	71.046.836	38.058.244
Privado com fins lucrativos	119.721.264	125.128.316	126.996.771	135.631.244
Privado optante pelo SIMPLES	-	-	-	4.360.836
Privado sem fins lucrativos	108.469.888	114.793.206	116.356.796	77.262.828
Universitários Privados	6.599.499	7.989.592	12.152.672	7.933.920
Filantropico com CNAS válido	-	-	-	62.601.535
Sindicatos	4.514.500	4.067.792	3.392.405	3.362.176
Não Identificados	14	-	-	2.636.282
Total	1.583.844.211	1.748.575.738	1.836.390.338	2.030.497.240

Tabela 7.3: Quantidade de internações hospitalares por tipo de regime do prestador [MIN05d].

REGIME	ANO			
	2.000	2.001	2.002	2.003
Público	3.159.062	3.423.251	3.667.205	4.100.057
Privado	7.152.175	6.891.869	6.671.025	6.739.864
Universitário	1.626.086	1.441.234	1.375.519	798.273
Total	11.937.323	11.756.354	11.713.749	11.638.194

Nos casos de estabelecimentos de saúde privados, os sistemas de informação são ainda mais específicos a cada organização, e mesmo entre os estabelecimentos da rede pública existe uma diversidade de sistemas para suporte operacional das unidades de saúde. Por este motivo, será apresentado as principais competências relacionadas à assistência ambulatorial, hospitalar e laboratorial, com objetivo de evidenciar as interações entre estes sistemas e os sistemas gestores da assistência em saúde relacionados abaixo:

- SIA – Sistema de Informação Ambulatorial
- SIH – Sistema de Informação Hospitalar

Assistência ambulatorial

A área de assistência ambulatorial compreende os procedimentos médicos realizados na maioria dos casos por unidades básicas de saúde. No entanto, existem procedimentos que são executados por hospitais e unidades especializadas.

A assistência ambulatorial compreende: consultas médicas em clínicas básicas; consultas médicas especializadas; atendimento de urgência/emergência; cirurgias ambulatoriais; procedimentos de patologia clínica; ações executadas por profissionais de enfermagem; ações básicas de odontologia; procedimentos médicos especializados; procedimentos em ortopedia; terapias especializadas e procedimentos de diagnose e terapia.

Na Figura 7.10 é composto um caso de uso para representar as principais interações que ocorrem no contexto da assistência ambulatorial, apresentando as entidades de agentes que contribuem para prover os serviços de assistência ambulatorial. Os agentes que fazem parte desta organização buscam interagir em prol de seus objetivos e também colaborar para suprir as necessidades dos demais agentes.

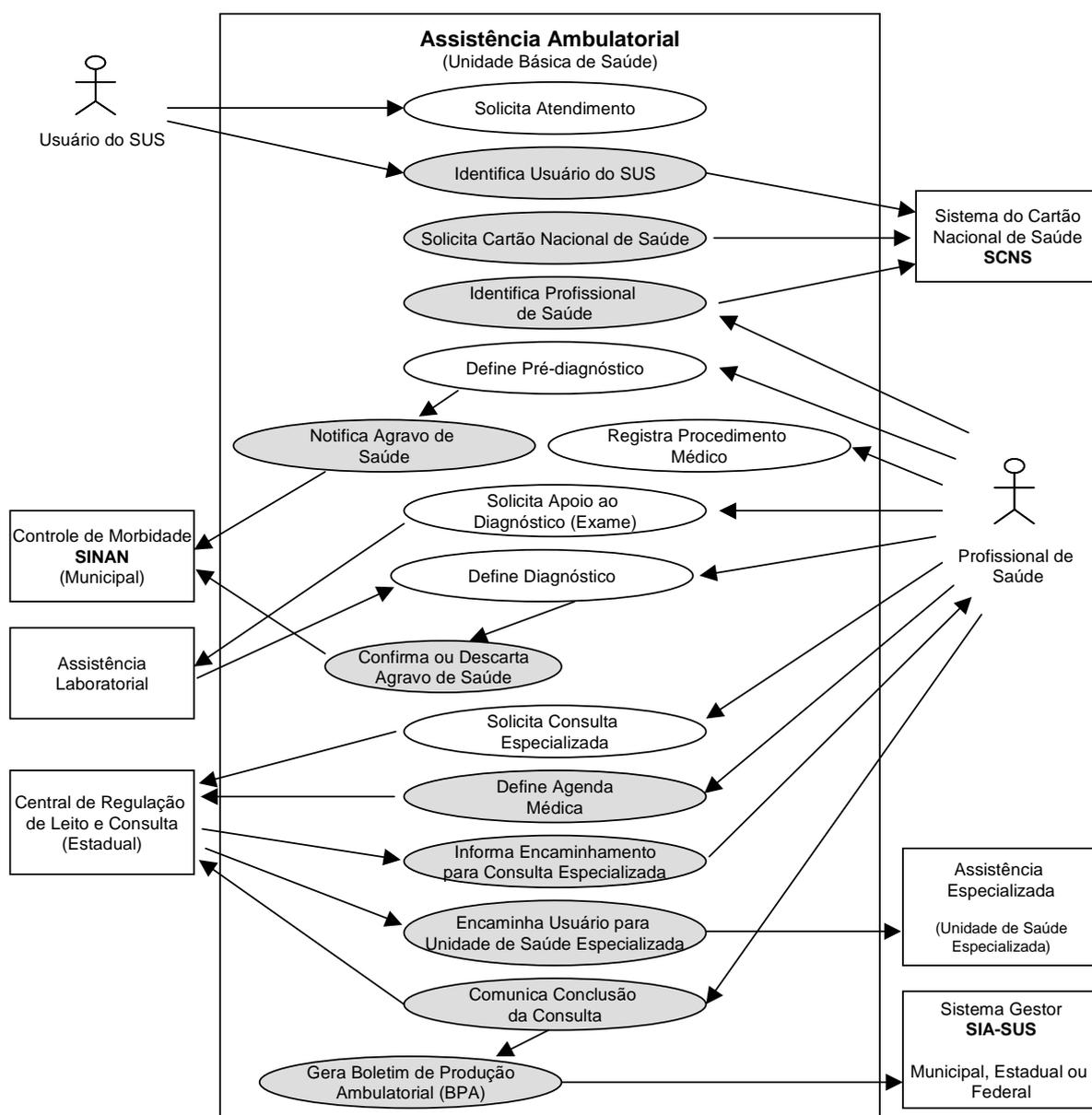


Figura 7.10: Diagrama de caso de uso da assistência ambulatorial

As capacidades do agente Unidade Básica de Saúde, são sintetizadas no diagrama de classe do agente, conforme apresentado na Figura 7.11. Neste diagrama pode-se observar os serviços que o agente provê no contexto da assistência em saúde.

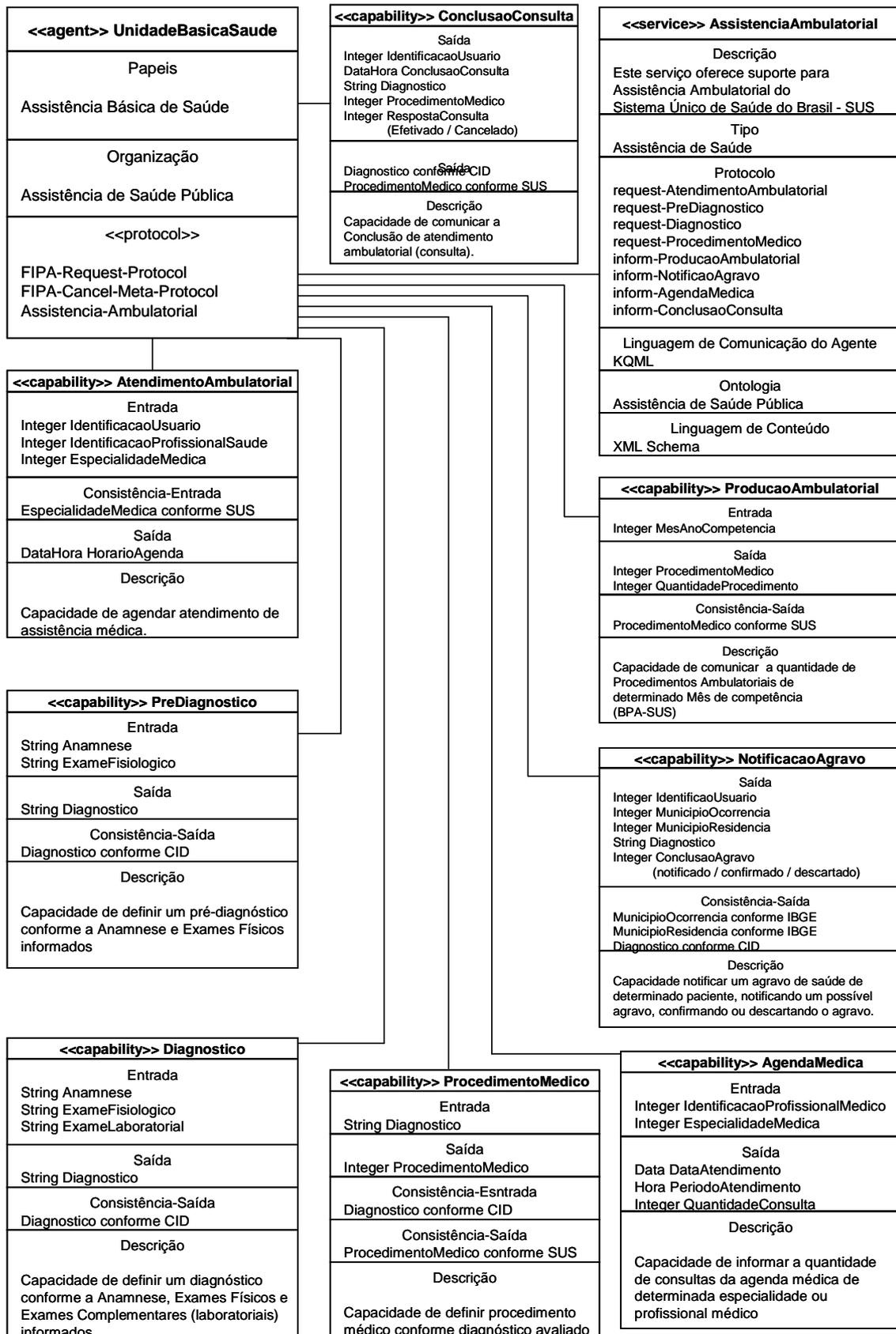


Figura 7.11: Diagrama de classe do agente Unidade Básica de Saúde.

Conforme apresentado no diagrama de classe, o agente Unidade de Saúde Básica, oferece serviço de assistência ambulatorial, constituindo as seguintes capacidades:

- agendar atendimento de assistência médica;
- definir um pré-diagnóstico conforme a anamnese e exames físicos informados;
- definir um diagnóstico conforme a anamnese, exames físicos e exames complementares (laboratoriais) informados;
- definir procedimento médico conforme diagnóstico avaliado;
- informar a quantidade de consultas da agenda médica de determinada especialidade ou profissional médico;
- notificar um agravo de saúde de determinado paciente, notificando um possível agravo, confirmando ou descartando o agravo;
- comunicar a quantidade de procedimentos ambulatoriais de determinado mês de competência ao Gestor do SUS (BPA-SUS);
- comunicar a conclusão de atendimento ambulatorial (consulta).

Assistência Hospitalar

A assistência hospitalar compreende os procedimentos médicos de média ou alta complexidade, realizados por hospitais e clínicas especializada em regime de internamento. Neste contexto, as atividades são realizadas em conjunto com os agentes gestores (Central de Regulação de Leitos e Consultas, Sistema do Cartão Nacional de Saúde e Sistema de Informação Hospitalar Gestor). O diagrama de caso de uso apresentado na Figura 7.12, representa as atividades realizadas no contexto hospitalar, identificando as principais interações entre os agentes no processo de assistência hospitalar:

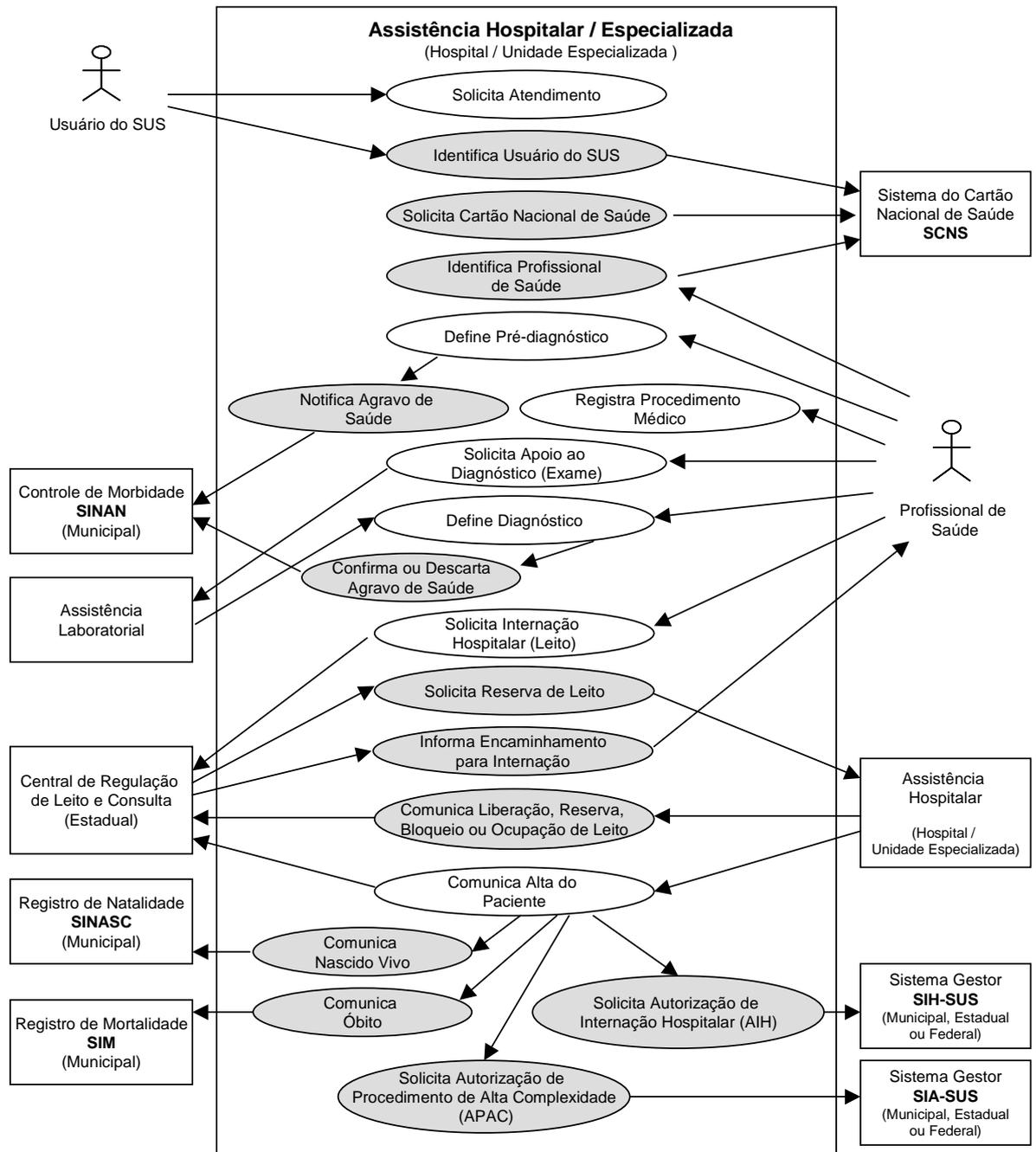


Figura 7.12: Diagrama de caso de uso da assistência hospitalar.

As capacidades do agente Hospital são definidas no diagrama classe do agente, conforme Figura 7.13, onde são contemplados os serviços que o agente oferece para a organização da assistência em saúde.

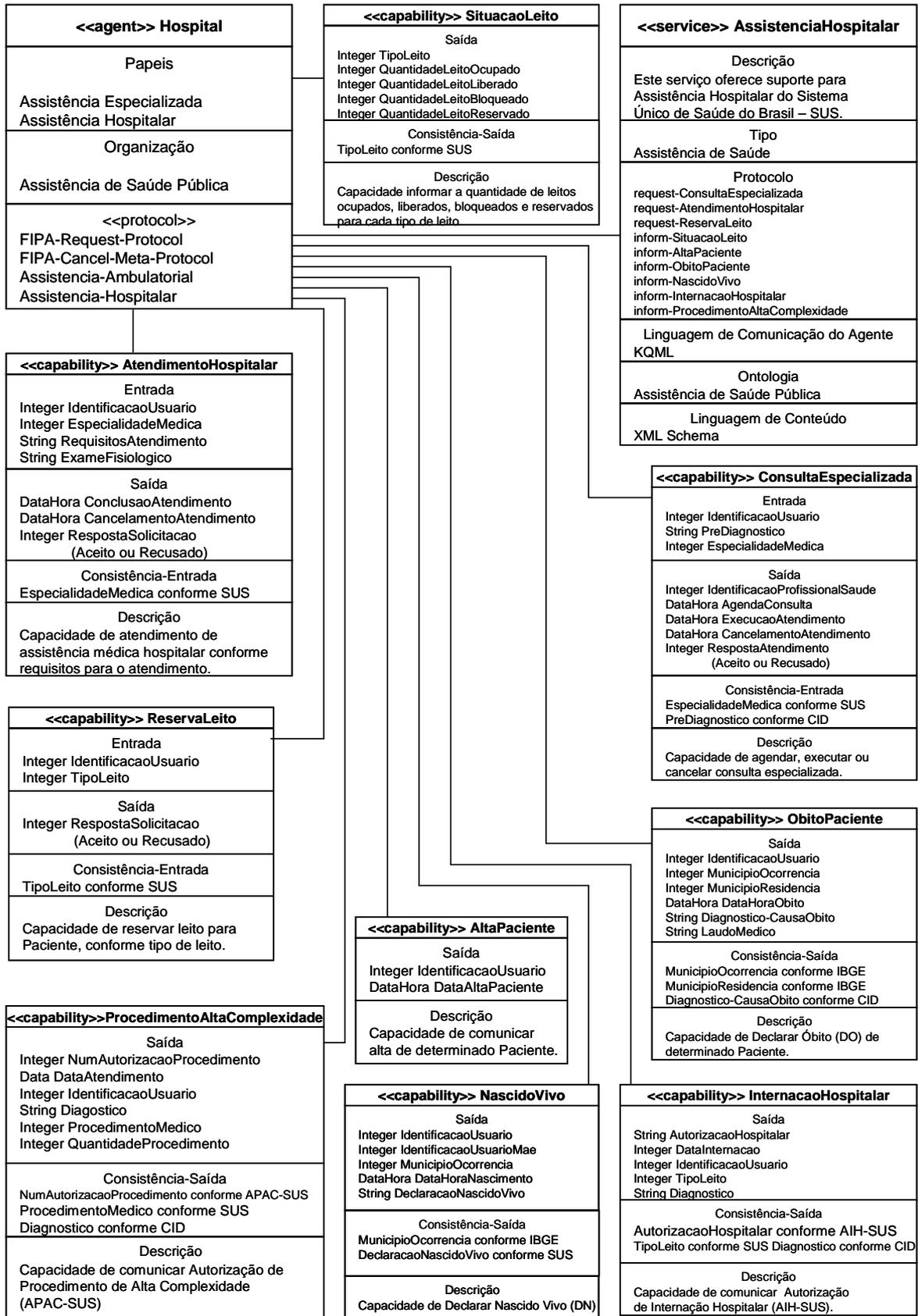


Figura 7.13: Diagrama de classe do agente Hospital.

As capacidades do agente Hospital são modeladas de acordo com o diagrama de classe e representadas como serviços suportados pelo agente, tais como:

- prestar atendimento de assistência médica hospitalar;
- reservar leito para paciente, conforme requisitos do tipo de leito;
- solicitar autorização de procedimento de alta complexidade (APAC-SUS);
- comunicar alta de determinado paciente;
- declarar Nascido Vivo (DN);
- agendar, executar ou cancelar consulta especializada;
- declarar Óbito (DO) de determinado Paciente;
- solicitar autorização de internação hospitalar ao gestor do SUS (AIH-SUS);
- informar a quantidade de leitos ocupados, liberados, bloqueados e reservados para cada tipo de leito.

Assistência Laboratorial

A assistência laboratorial compreende as atividades executadas como apoio ao diagnóstico médico em forma de exames laboratoriais. Neste contexto, o processo das atividades são realizados em conjunto com os agentes que utilizam-se dos serviços laboratoriais, tais como os agentes de assistência Ambulatorial e Hospitalar. No diagrama de caso de uso apresentado na Figura 7.14 é representado a interação entre estes agentes:

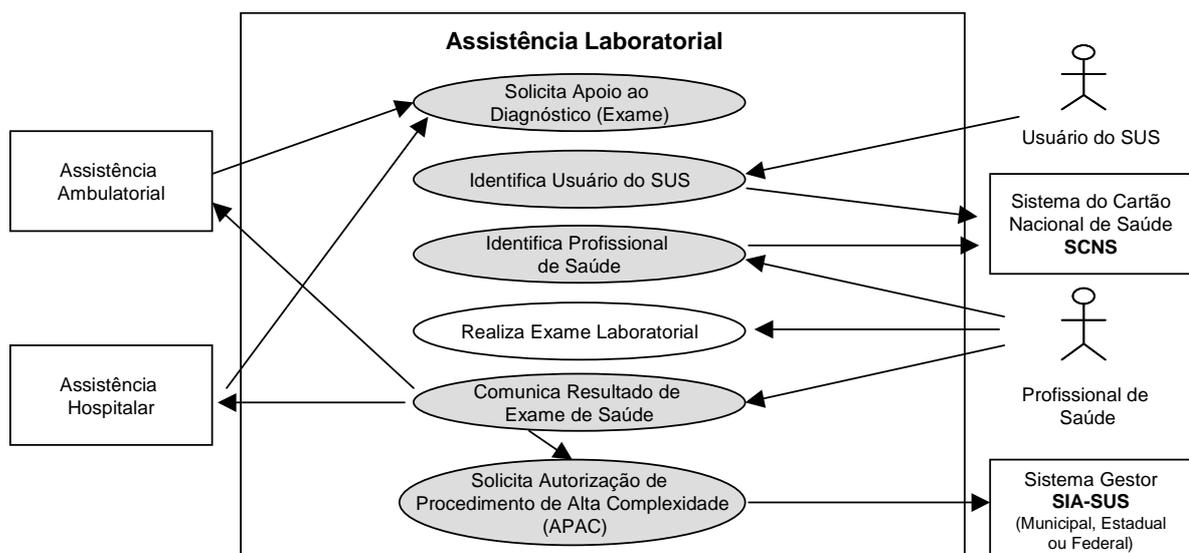


Figura 7.14: Diagrama de caso de uso da assistência laboratorial.

Para as atividades realizadas no contexto laboratorial são identificadas as principais interações entre os agentes envolvidos neste processo. As capacidades do agente Laboratório são modeladas de acordo com diagrama de classe e representadas como serviços suportados pelo agente, tais como:

- prover exame laboratorial atendendo as solicitações das áreas de assistência;
- comunicar resultado de exame laboratorial conforme protocolos laboratoriais do SUS;
- comunicar Autorização de Procedimento de Alta Complexidade (APAC-SUS).

As capacidades do agente Laboratório são definidas no diagrama classe do agente, conforme Figura 7.15, onde são contemplados os serviços que o agente oferece para a organização da assistência em saúde.

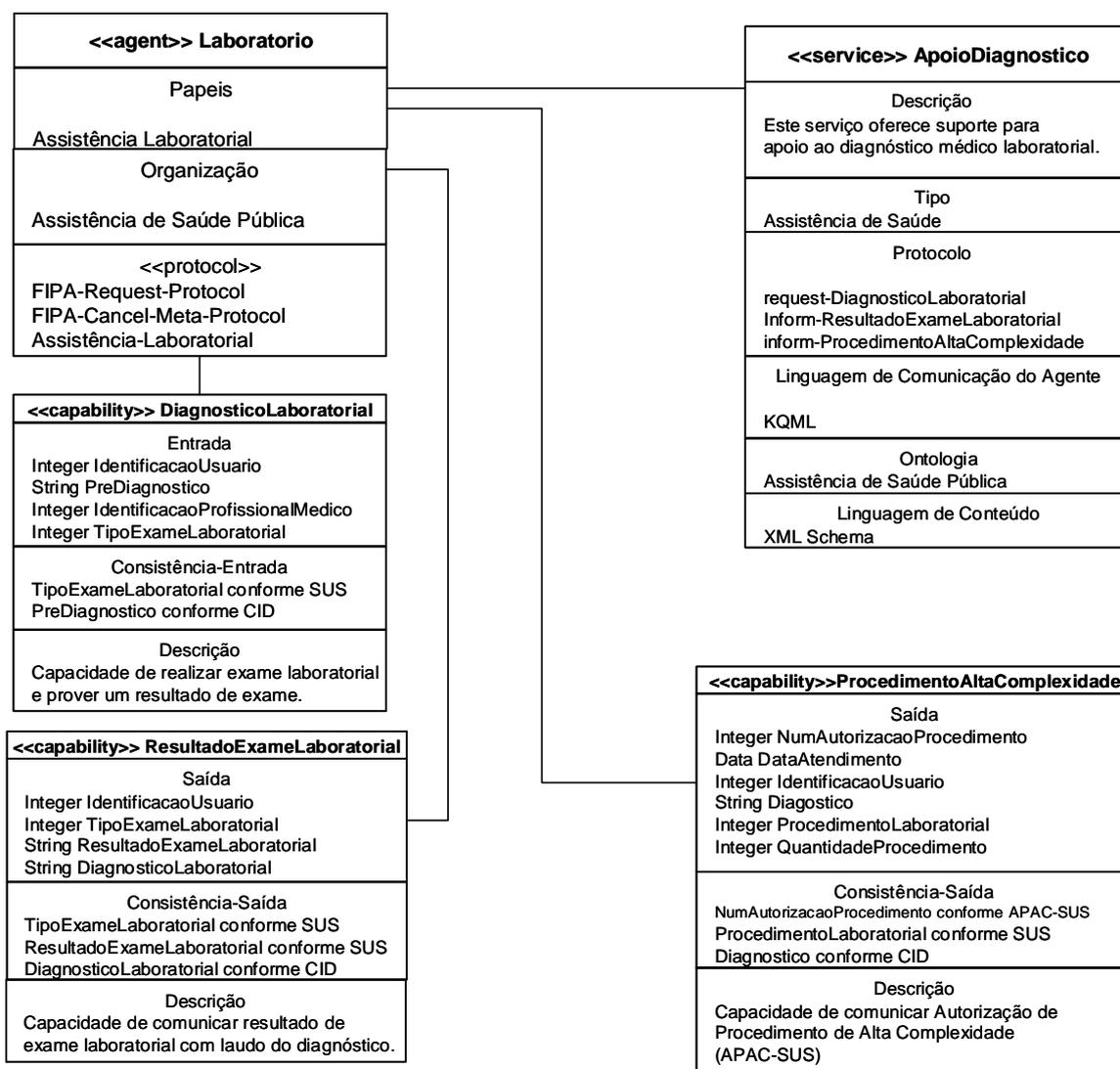


Figura 7.15: Diagrama de classe do agente Laboratório.

7.3. Arquitetura do Sistema de Agentes aplicado no Perfil de Saúde

As informações epidemiológicas originam-se da área de assistência de saúde (ambulatorial e hospitalar), ou seja, a informação sobre determinada ocorrência de doenças, nascimentos ou mortalidade são oriundas principalmente da área de assistência de saúde. No caso de mortalidade e natalidade existem outras fontes de informações como cartório de registro civil, porém, com uma escala bem menor de ocorrências.

Sobre este aspecto, é importante proporcionar aos sistemas o compartilhamento do conhecimento produzido na área de assistência de saúde para com as demais áreas, tais como a vigilância epidemiológica. Considerando esta necessidade de integração, foi estruturado um sistema de agentes capaz de prover o compartilhamento das informações produzidas pelos sistemas de cada área, assim como utilizar os serviços de competência de cada sistema, atribuindo aos sistemas maior interação do conhecimento.

Na arquitetura proposta, busca-se através de agentes de software sustentar a integração entre os segmentos da saúde, criando mecanismos para interoperar os sistemas existentes. A arquitetura baseada em agentes elaborada neste trabalho, tem como fundamentação os seguintes princípios: a) **flexibilidade operacional**: cada sistema de informação pode representar as informações dentro do contexto de sua aplicação, apresentando os dados que representem seus conhecimentos e competências de forma independente; b) **integridade dos dados**: os dados devem ser validados conforme suas estruturas, para manter íntegro os dados dentro de um contexto; c) **interoperabilidade dos sistemas**: os sistemas devem possuir mecanismos para comunicação (compartilhamento) de dados e conhecimentos.

Almejando a integração do conhecimento gerado nos sistemas de informações do SUS, com base nos princípios citados, foi concebida uma arquitetura que pode encapsular os sistemas legados existentes dentro de uma estrutura de sistema de agentes, onde cada instância do sistema passará a ser um agente, promovendo a integração e colaboração entre os sistemas.

A interação entre os sistemas (agentes) é proporcionada através da linguagem KQML (Knowledge Query Manipulation Language), que foi criada pelo grupo de trabalho *External Interfaces Working Group* da *DARPA Knowledge Sharing Effort*. KQML é uma linguagem de comunicação que define mensagens, denominadas performativas, com funções específicas para expressar a intenção do agente durante a comunicação com outros agentes [FIN94].

Para comunicar o conhecimento entre os sistemas é utilizado no conteúdo da mensagem KQML, protocolos que representam o contexto de cada aplicação, estes protocolos utilizam como referência os protocolos *FIPA-Request-Protocol* e *FIPA-Cancel-Meta-Protocol* definidos pela FIFA (*Foundation For Intelligent Physical Agents*) [FIP02c].

O conteúdo que representa o conhecimento é transmitido através dos protocolos expresso em uma linguagem de conteúdo. Foi utilizado como linguagem de conteúdo a linguagem XML Schema (*eXtensible Markup Language Schema*) ([BRA04],) [NAR02]. Este padrão foi definido pela W3C (*World Wide Web Consortium*) com objetivo de descrever dados estruturados ou semi-estruturados.

Os dados em XML são validados através de DTD (*Document Type Definition*) que especifica a estrutura dos dados formatados em XML, possibilitando a interpretação dos dados [BRA04]. Deste modo, pode-se representar o conhecimento de forma estruturada, possibilitando uma integração dos dados entre os sistemas proprietários dos SUS, tornando-se flexível o intercâmbio de dados entre sistemas distintos e heterogêneos.

Para possibilitar a interoperabilidade dos sistemas mantendo a flexibilidade operacional de cada um, foram implementadas algumas performativas KQML utilizando a Internet como meio de comunicação, ou seja, a comunicação entre os sistemas é proporcionada através do protocolo SOAP (*Simple Object Access Protocol*). Operando desta forma, o transporte de mensagens utiliza-se como meio de comunicação o protocolo HTTP (*Hypertext Transfer Protocol*), suportando a interoperabilidade dos sistemas através da Internet [GUD03]. A arquitetura construída sobre *Web Service* tem como objetivo promover a integração de aplicações em ambientes heterogêneos.

Desta forma, organizações distintas que trabalham com informações em comum podem ser integradas. A integração dos negócios, das funcionalidades e operações dos sistemas legados do SUS, proporcionará uma unificação colaborativa, evidenciando o SUS como um Sistema Único de Saúde Integrado.

7.3.1. Estrutura de um Agente

Existem alguns modelos de agente propostos na literatura como nos projetos SHADE, ARCHON e OSACA ([MCG93a], [JEN92], [SCA96b]). Alguns conceitos são comuns entre os modelos propostos. O modelo de agente adotado é estruturado com base no

modelo ARCHON e utilizando a arquitetura de agente desenvolvido por Machado 2005 [MAC05]. Neste modelo representa um agente através de camadas com funcionalidades específicas conforme apresentado na Figura 7.16:

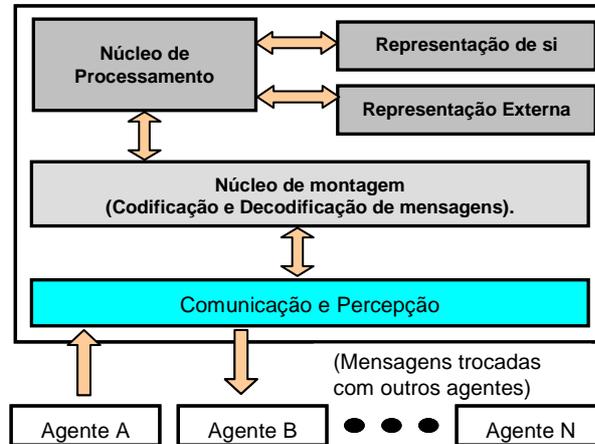


Figura 7.16: Estrutura de um Agente

No modelo apresentado, o agente é subdividido em algumas camadas com as seguintes funcionalidades:

- comunicação e Percepção: Camada onde é implementada a forma de um agente comunicar com os demais agentes da rede. A comunicação do agente é suportada pela troca de mensagens KQML.
- núcleo de Processamento: Implementa a inteligência do agente, trabalhando em conjunto com memórias de representação do próprio agente (representação de si) e de outros agentes (representação externa).
- representação de si: Armazena os conhecimentos utilizados pelo agente para processar as respostas que representam o conhecimento do agente. Nesta camada também são armazenadas as capacidades e as tarefas do agente.
- representação Externa: Armazena os conhecimentos sobre o ambiente externo, principalmente das capacidades e tarefas dos demais agentes, formando uma representação das competências de outros agentes.

7.3.2. Estrutura de Comunicação

O funcionamento da estrutura de comunicação pode ser observado na Figura 7.17, onde são apresentados os protocolos utilizados na comunicação. Cada sistema encapsulado

como um agente, será provido de comunicação com outros agentes (sistemas) através dos protocolos HTTP e SOAP citados anteriormente, ambos identificados na camada de aplicação do modelo OSI [OSI89].

Esta estrutura de comunicação possibilita a troca de mensagens KQML entre os agentes, suportando a interação entre os sistemas legados. O modelo OSI é apresentado na Figura 7.17, para facilitar a compreensão, sendo utilizado o protocolo TCP/IP nesta estrutura de comunicação.

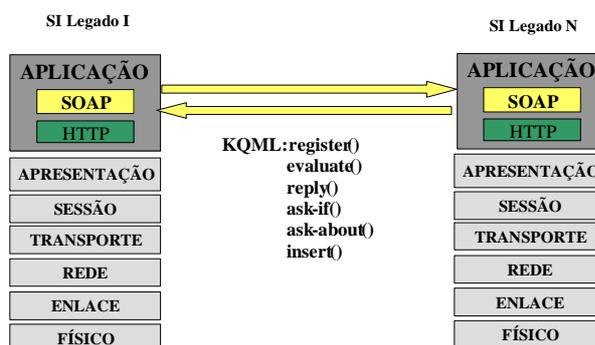


Figura 7.17: Estrutura de Comunicação

7.3.3. Integridade dos Dados Interoperados

Através das mensagens KQML é provida a interoperabilidade, possibilitando a troca de informações entre os sistemas legados. Como cada área de aplicação manipulam dados em um determinado contexto, a troca de conhecimento entre os agentes será realizada por meio do conteúdo das mensagens KQML. Para expressar o conhecimento, é utilizado XML Schema como linguagem de conteúdo das mensagens, possibilitando comunicar estrutura de dados entre os agentes. Para consistir a integridade dos dados interoperados é utilizado DTD. Na seqüência é apresentado um documento DTD e XML exemplificando a comunicação entre dois agentes.

a) Exemplo de um documento DTD:

```
<!ELEMENT usuario (nome, datanasc, sexo, medicamento*)>
<!ATTLIST usuario cns ID #required>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT datanasc (#PCDATA)>
<!ELEMENT sexo (M | F)>
```

```

<!ELEMENT medicamento (principioativo, posologia)>
<!ELEMENT principioativo (#PCDATA)>
<!ELEMENT posologia (#PCDATA)>

```

b) Exemplo de um documento XML:

```

<usuario cns = 1>
  <nome>Pedro</nome>
  <datanasc>03/08/1970</datanasc>
  <sexo>M</sexo>
  <medicamento>
    <principioativo>Ácido Acetilsalicilico</principioativo>
    <posologia>100mg / dia</posologia>
  </medicamento>
  <medicamento>
    <principioativo>Paracetamol</principioativo>
    <posologia>25mg / dia</posologia>
  </medicamento>
  <medicamento>
    <principioativo>Diclofenaco de Potássico</principioativo>
    <posologia>50mg / dia</posologia>
  </medicamento>
</usuario>

```

A seguir é apresentado um exemplo de troca de mensagens entre sistemas legado encapsulado em agentes, conforme Figura 7.18:

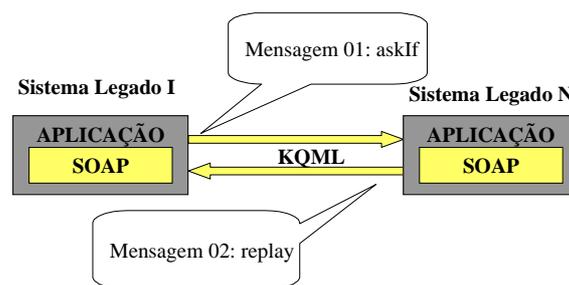


Figura 7.18: Troca de mensagens KQML

Mensagem 01:

```

ask-if :language XMLSchema :ontology SUS-Medicamento
      :reply-with q1 :content (request(<usuario cns=1></usuario>))

```

Mensagem 02:

```

reply :language XML Schema :ontology SUS-Medicamento
      :in-reply-with q1 :content(inform(
        <usuario cns = 1>
          <nome>Pedro</nome>
          <datanasc>03/08/1970</datanasc>
          <sexo>M</sexo>
        <medicamento>
          <principioativo>Ácido Acetilsalicilico</principioativo>
          <posologia>100mg / dia</posologia>
        </medicamento>
        <medicamento>
          <principioativo>Paracetamol</principioativo>
          <posologia>25mg / dia</posologia>
        </medicamento>
        <medicamento>
          <principioativo>Diclofenaco de Potássico</principioativo>
          <posologia>50mg / dia</posologia>
        </medicamento>
      </usuario>))

```

Neste exemplo, observa-se uma solicitação de informação sobre determinado usuário do SUS, onde a resposta devolvida é dados do paciente e dos medicamentos em uso. Através do arquivo DTD, pode ser representado o modelo relacional, como neste exemplo, um paciente pode ter muitos medicamentos relacionados a ele. Desta forma, no momento da interpretação dos dados do arquivo XML é realizada a validação dos mesmos, de acordo com a estrutura definida no arquivo DTD, mantendo a integridade e consistência dos dados.

7.4. Implementação do Sistema de Agentes Orientados a Serviços para o SUS

O sistema de agentes orientados a serviços foi desenvolvido com intuito de prover os mecanismos necessários para representar o Perfil de Saúde do Estado do Paraná. Para tanto, foram utilizadas as técnicas apresentadas ao longo deste trabalho e aplicadas no

desenvolvimento da arquitetura proposta. A Figura 7.19 ilustra as camadas que suportam a comunicação entre os agentes:

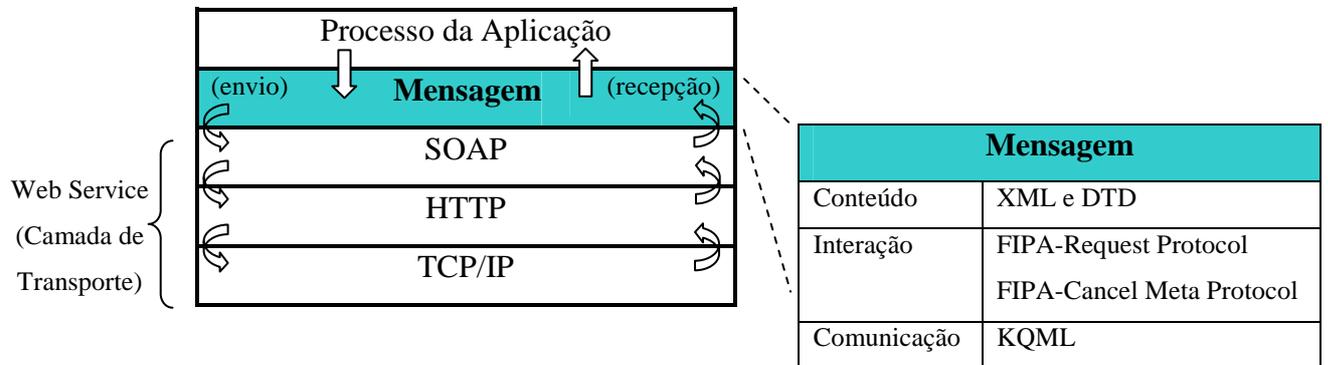


Figura 7.19: Camadas da Arquitetura de Comunicação dos Agentes

As primeiras três camadas representam os protocolos utilizados como transporte da informação comunicada entre os agentes. Estas camadas foram estruturadas utilizando a arquitetura *Web Service*, onde os dados são transmitidos sobre o protocolo SOAP, proporcionando a comunicação entre os sistemas que serão encapsulados em forma de agentes de software, sendo suportados pela plataforma Web.

A quarta camada é responsável pela codificação e decodificação utilizando-se a linguagem KQML e dos protocolos FIPA-Request Protocol e FIPA-Cancel Meta Protocol para interação entre os agentes, e comunicando o conteúdo da informação descritos em XML e validados pelo esquema DTD.

As aplicações que suportam *Web Service* podem ser integradas diretamente com os serviços publicados no agente. Atualmente muitas linguagens de programação suportam *Web Service* como Java, C++, Delphi, ASP, etc. A especificação da arquitetura *Web Service* foi disponibilizada pela W3C e pode ser implementada para linguagem de programação que se deseja. No entanto, para as aplicações que não suportam *Web Service* ou que não tem disponibilidade de manutenção, foi desenvolvido uma agente denominado Wrapper Agent. Este agente tem como propósito facilitar a interface entre as fontes de dados compartilhadas e os demais agentes, ficando limitado aos dados. Para as informações processadas será necessária a elaboração de módulos para tratamento da informação, conforme apresentado na Figura 7.20:

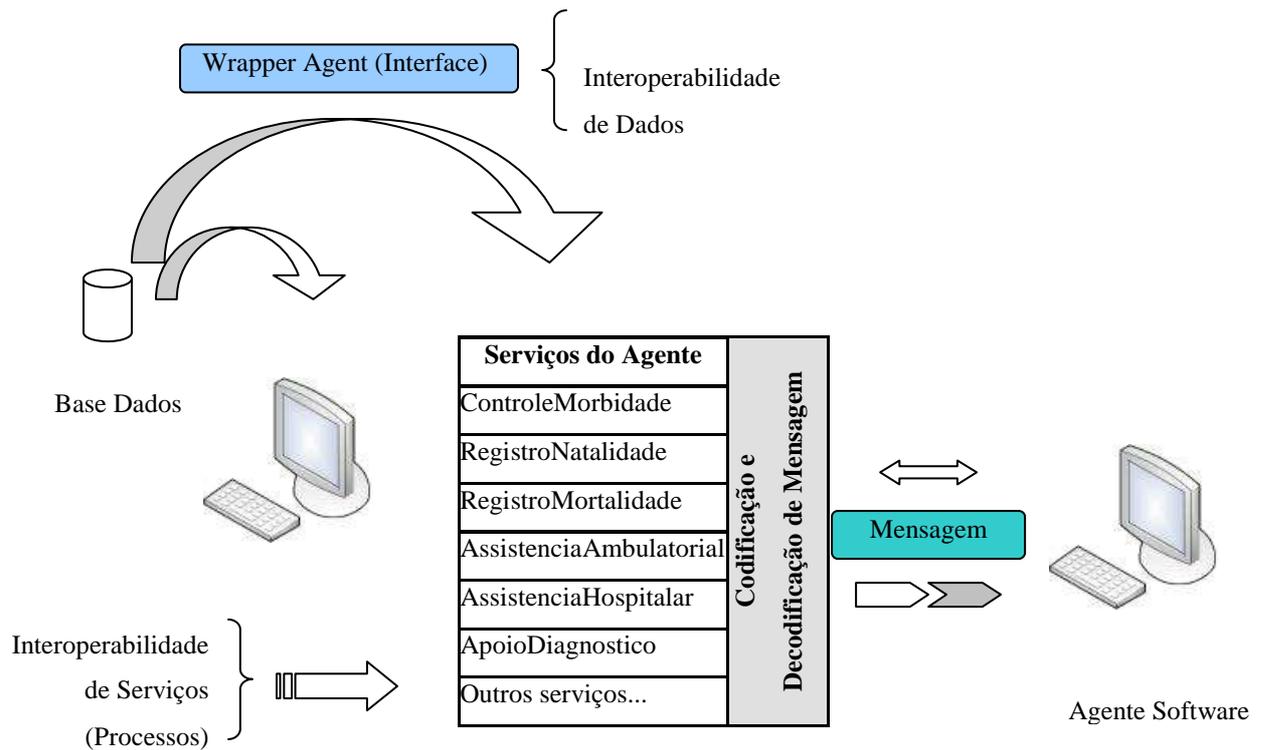


Figura 7.20: Esquema de Comunicação do Sistema de Agentes Orientado a Serviços.

Para configuração e monitoramento dos agentes foi desenvolvido uma interface de usuário. Esta interface é utilizada para configuração dos serviços e competências do agente, assim como o acompanhamento de atividades em execução ou agendamento de novos serviços. Para cada competência do agente de acordo com a aplicação que está sendo encapsulada, é definida um esquema DTD representando a estrutura de dados de entrada e de saída, conforme apresentado na Figura 7.21 e Figura 7.22:

The screenshot shows the 'Configuração e Monitoramento de Agente' window with the 'Identificação' tab selected. The window is divided into several sections:

- Agentes:**
 - ID:
 - IP:
 - Porta:
- Facilitador:**
 - ID:
 - IP:
 - Porta:
- Descrição do Agente:**

Este serviço oferece suporte para Controle de Morbidade (doenças) de interesse Epidemiológico.
- Ontologia:**

Vigilância Epidemiológica do SUS
- Linguagem de Conteúdo:**

XML/DTD
- Esquema de Dados (XML/DTD):**

\\documento\Epidemiologia.dtd

Figura 7.21: Interface para Configuração e Monitoramento de Agente (Identificação).

The screenshot shows the 'Configuração e Monitoramento de Agente' window with the 'Serviços' tab selected. The window displays the following configuration options:

- Serviços:**
 - Adicionar...
 - ControleMorbidade
 - RegistroNatalidade
 - RegistroMortalidade
 - AssistenciaAmbulatorial
 - AssistenciaHospitalar
 - ApoioDiagnostico
- Capacidades (Protocolo Interação):**
 - Adicionar...
 - RegistroNotificacaoAgravado
 - InvestigacaoAgravado
 - NotificacaoInvestigacaoAgravado
 - MonitoramentoMorbidade
- Estrutura Entrada (DTD):**

```
<!ELEMENT UsuarioMedicamento (nome, datanasc, sexo, medicamento+)>
<!-- ATTLIST UsuarioMedicamento cns ID #REQUIRED -->
<!ELEMENT nome (#PCDATA)>
<!ELEMENT datanasc (#PCDATA)>
<!ELEMENT sexo ("M" | "F")>
<!ELEMENT medicamento (principioativo, posologia)>
<!ELEMENT principioativo (#PCDATA)>
<!ELEMENT posologia (#PCDATA)>
```
- Estrutura Saída (DTD):**

(Empty text area)

Figura 7.22: Interface para Configuração e Monitoramento de Agente (Serviços).

Cada um dos sistemas é encapsulado como um agente, representando suas informações em determinado contexto. Para gerenciamento da rede esta sendo utilizado um agente facilitador que mantém as informações dos demais agentes da rede, assim como possibilita inserção ou exclusão dos agentes da rede, conforme apresentado na Figura 7.23:

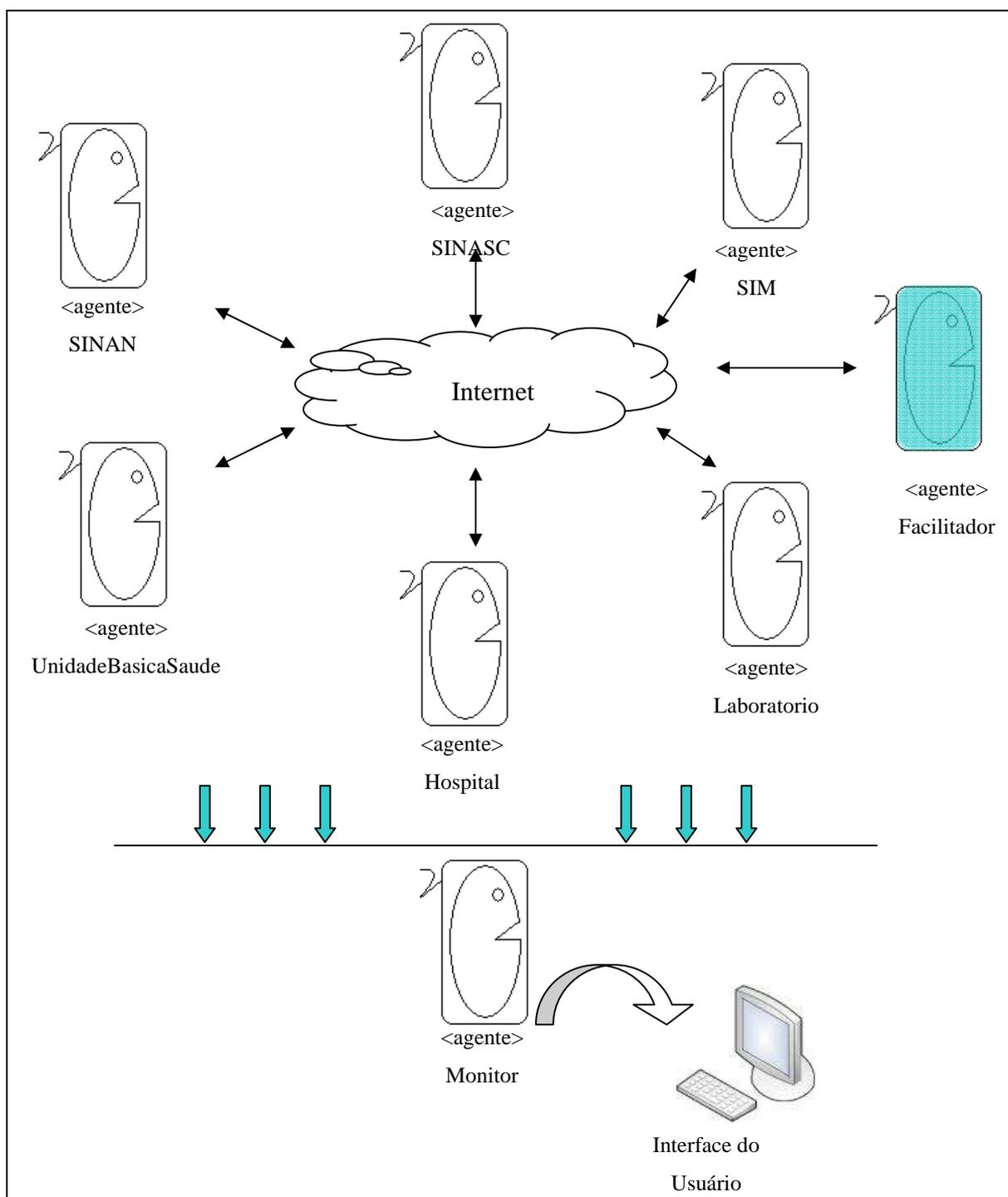


Figura 7.23: Estrutura de Funcionamento do Sistema de Agentes para o SUS.

Capítulo 8 - Conclusão

A integração de sistemas de informação discutida neste trabalho fortaleceu a possibilidade da execução do mesmo. Vindo a ajudar na composição e estudos das novas tecnologias utilizadas com este objetivo. Observa-se que cada vez mais se busca a união entre os sistemas de informação, de modo a possibilitar uma interoperabilidade funcional dos sistemas. Sendo assim, muito mais do que utilizar a infra-estrutura da Internet como fonte de propaganda ou meio de comunicação trivial como: correio eletrônico, sites de venda ou marketing. É necessário unir funcionalidades, tornar as instituições integradas de modo a permitir a interoperabilidade entre os sistemas de informação.

A fraca integração entre os sistemas de saúde do SUS pode ser reduzida através de um sistema de agentes independentes racionais cooperativos. Um dos setores dispendiosos para a Saúde Pública é a Vigilância Epidemiológica, pois é utilizada uma diversidade de fontes de dados não sincronizadas e não integradas para gerar informações tanto de cunho operacional como estratégico conforme citado no neste trabalho.

Através da implementação do sistema de agentes de software para o SUS, ou seja, da integração das aplicações desta organização, torna-se viável o compartilhamento do conhecimento entre as áreas de Assistência em Saúde e Vigilância Epidemiológica, pois são áreas correlacionadas. Proporcionando como resultado cooperativo o monitoramento do perfil de saúde dos municípios de forma rápida e flexível, diminuindo as inconsistências e facilitando a divulgação dos dados do perfil de saúde.

A integração dos sistemas de informações do SUS pode ser proporcionada pela arquitetura de agentes de software suportada pelas tecnologias *Web Service* e *Web Semântica*, conforme apresentado neste trabalho, encapsulando os sistemas legados do SUS em agentes que possibilitem a interoperabilidade. Por outro lado, foram identificadas algumas limitações, como por exemplo: a) dificuldade de representação XML de dados relacionais em estruturas complexas; b) baixa performance de comunicação para interatividade de grande quantidade de dados; c) a interação entre os agentes não pode ser restringida pelas performativas KQML existentes, ou seja, é necessário criar mecanismos para extensão da linguagem KQML caso venha a ser necessário em contextos específicos.

Estas limitações podem ser reduzidas, quanto a possibilitar a criação de novas performativas da linguagem KQML, foi testado com sucesso o uso da própria estrutura *Web Service* para codificar as performativas KQML. Desta forma, mantendo uma linguagem comum de fácil atualização, sem limitar as aplicações em criar novas performativas para atender necessidades específicas de determinados contextos.

A extensão de linguagem pode ser mantida através da publicação das novas performativas KQML entre os agentes da rede, utilizando como meio de divulgação documento WSDL (*Web Services Description Language*) que possibilita a publicação de serviços (métodos ou processos).

A aplicação da Web Semântica, em particular do modelo semântico RDF, no contexto de sistemas de agentes forneceu uma linguagem comunicação expressiva, proporcionando uma maior flexibilidade na integração de dados e processos do Sistema Único de Saúde (SUS). A definição de novos protocolos de interação que serão comunicados através da linguagem KQML possibilitará representar as competências de contextos específicos da Saúde Pública de forma evolutiva.

8.1. Resultados e Trabalhos Futuros

O trabalho desenvolvido com objetivo de definir mecanismos de interoperabilidade entre sistemas de informação proporcionou a aplicação de tecnologias emergentes no desenvolvimento de agentes de software, sendo efetivo quanto à utilização de Web Semântica (XML, DTD, RDF) para representação de dados e conhecimento compartilhados entre os sistemas de informação. Criando condições para aplicação da arquitetura do sistema de agentes como proposta para monitoramento do perfil de saúde do Estado do Paraná. Neste trabalho, focalizou-se de forma experimental a um número limitado de sistemas e fonte de informações. Sendo que, para prover a totalidade de indicadores de saúde definidos pelo Ministério da Saúde, é necessário estender este trabalho a um conjunto maior de outros sistemas utilizados como instrumentos da Saúde Pública.

O monitoramento integrado do perfil de saúde possibilitará um acompanhamento epidemiológico regionalizado, através do compartilhamento do conhecimento de indicadores de interesse da Saúde Pública, possibilitando a detecção precoce de epidemias, assim como avaliação das ações conjuntas dos níveis de gestão.

O sistema de agentes orientado a serviços possibilitará a avaliação monitorada do impacto de ações, programas e políticas de saúde de forma emergente, percebendo no tempo certo à situação de saúde da população. Proporcionando a retroalimentação (divulgação da informação entre os níveis de gestão) entre os sistemas de informação, proporcionando um forma de reavaliar os processos e ações da Saúde Pública, potencializando o processo de decisão gerencial do Sistema Único de Saúde Pública do Brasil (SUS).

O trabalho elaborado consistiu a aplicação XML para representação do conhecimento comunicado entre os agentes. Ainda assim, pode-se explorar a possibilidade de mapeamento dos serviços disponíveis dos agentes e a representação de objetos em linguagens orientada a objeto.

Sobre o aspecto de coordenação e planejamento de ações executadas em conjunto entre os agentes, existem propostas e estudos sendo elaborados com objetivo de proporcionar mecanismos de controle e coordenação de atividades dos agentes. Sobre este aspecto, podem ser explorados novos mecanismos como Schema RDF (Schema Resource Description Framework) e Web Ontologia ainda em estudos.

Neste trabalho buscou-se avaliar e representar os agentes de software em uma arquitetura suportada pela Web (*Web Service*), propiciando a interoperabilidade entre os sistemas da área de Saúde Pública. Contudo, como a áreas de Saúde Pública, outras áreas correlacionadas podem ser projetadas com este mesmo objetivo.

Referências Bibliográficas

- [ARK02] Arkin, Assaf, *Web Service Choreography Interface (WSCI) 1.0*, World Wide Web Consortium. Agosto de 2002.
- [BAA93] Van Baalen, J., Fikes, R.; 1993. *The Role of Reversible Grammars in Translating Between Representation Languages*. Stanford University, CA.
- [BAR02] BARROS, R.S.M.; FERREIRA, S.M.G.; HEXSEL, R.A. *Desenvolvimento de Solução Única de Software para o Sistema Cartão Nacional de Saúde*. VIII Congresso Brasileiro de Informática em Saúde, Natal, 2002.
- [BAR95] BARBUCEANU M.; FOX M. *A Language for Describing Coordination in Multi Agent Systems*. *Proceedings First International Conference on Multi-Agent Systems (ICMAS'95)*, 1995.
- [BER01] BENNERS-LEE, T.; LASSILA, O. HENDLER, J. *The Semantic web*. *Scientific American*, In: www.scientificamerican.com/2001/0501issue/0501berners-lee.html, 2001.
- [BIR04] BIRON, P.V.; MALHOTRA, A. *XML Schema Part 2: Datatypes Second Edition*. World Wide Web Consortium (W3C), In: <http://www.w3.org/TR/xmlschema-2/>, outubro, 2004.
- [BON88] BOND H. & GASSER L. *What is DAI ? Reading in Distributed Artificial Intelligence*, Morgan Kaufman Publishers, Los Altos, CA, 1988.
- [BON92] BOND H. & GASSER L. *A Subject-Indexed Bibliography of Distributed Artificial Intelligence*, *IEEE Transactions on Systems, Man, And Cybernetics*, 22, no.6, November/December 1992, p.1260-1281.
- [BOO04] Booth, David, *Web Services Architecture*, World Wide Web Consortium. Fevereiro 2004.

- [BOU81] Boulevard, Wilson, Transmission Control Protocol, Internet Engineering Task Force, RFC793. Setembro 1981.
- [BRA01] BRASIL, MINISTÉRIO DA SAÚDE, SECRETARIA EXECUTIVA, Cartão Nacional de Saúde, Cadastramento de Usuários do SUS: Portarias: MS/SE n.º 17, MS/SE/SIS n.º 57, de 2001. – Brasília: Ministério da Saúde, 2001.
- [BRA04] BRAY, T.; PAOLI, J.; SPERBERG, C.M.; MALER, E.; YERGEAU, F. Extensible Markup Language (XML) 1.0 (Third Edition). World Wide Web Consortium (W3C), In: <http://www.w3.org/TR/REC-xml/>, fevereiro, 2004.
- [BRA85] BRACHMAN, R.; SCHMOLZE, J. An Overview of the KL-ONE Knowledge Representation System, *Cognitive Science*, no.9, 1985.
- [BRE94] BREUKER, J.; VANDEVELDE, W. CommonKADS Library for Expertise Modelling, IOS Press, 1994.
- [BRO86] BROOKS R. A. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, 2(1), March 1986, p.14-33.
- [CAM83] CAMMARATA, S.; ARTHUR, M. D.; STEEB, R. Strategies of Cooperation in Distributed Problem Solving, *Proceedings of the Eighth IJCAI*, vol. 2, Karlsruhe, August 8-12, 1983, p.767-770.
- [CAR98] CARVALHO, A. I. Política de Saúde e Organização Setorial no País. Curso de Especialização à Distância em Autogestão em Saúde, Escola Nacional de Saúde Pública FIOCRUZ, 1998.
- [CHE01] CHEYER, A.J.; MARTIN, D.L. The Open Agent Architecture. *Journal of Autonomous Agents and Multi-Agent Systems*, vol.4, no.1, p.143-148, March 2001.
- [CHR01] CHRISTENSEN, Erik, CURBERA, Francisco, MEREDITH, Greg, WEERAWARANA, Sanjiva, *Web Services Description Language (WSDL) 1.1*, World Wide Web Consortium. Março 2001.

- [CLE04] Clement, Luc, UDDI Spec Technical Committee Draft Version .3.0.2, Organization for the Advancement of Structured Information Standards (OASIS). Outubro 2004.
- [CNS04] CNS-PR. Expansão do CNS: Resultado do Encontro de Consultores. Secretaria Estadual de Saúde do Paraná, Curitiba, 2004.
- [COH79] COHEN P.R.; PERRAULT C.R. Elements of Plan-Based Theory of Speech Acts. *Cognitive Science*, 3(3), 1979, p.177-212.
- [CON88] CONRY S.E.; MAYER R.A.; LESSER V.R. Multistage Negotiation in Distributed Planning. *Readings in Distributed Artificial Intelligence*, Bond and Gasser (eds) Morgan Kaufman, 1988, p.367-383.
- [COS98] COSTA, Nilson do Rosário, 1998. Políticas Públicas, Justiça Distributiva e Inovação: saúde e saneamento na agenda social. São Paulo: HUCITEC.
- [CUT93] Cutovsky, M., Engelmores, R., Fikes, R., Genesereth, M., Gruber, T., Mark, W., Tenenbaum, J. 1993. PACT: An experiment in integrating concurrent engineering systems. *IEEE Computer*, January 1993. In special issue on computer-supported concurrent engineering.
- [CYC96] Cyclades, Guia Internet de Conectividade, User's Guide, 1996.
- [CZE03] Czerveny, Arno, *Revista Mundo Java*, Número 02 Ano 1, 2003
- [DAM04] Damjanović, Violeta, Web Agent's Enclaves – A New Opportunity for the Semantic *Web Services*, DAM04 Workshop Application Design, Development and Implementation Issues in the Semantic Web, Central EUROpe Workshop Proceedings. <http://CEUR-WS.org> Maio de 2004.
- [DAR04] Defense Advanced Research Projects Agency, Agent Markup Language, *Seman/Ontology Web Language Semantic (DAML/OWL-S)*, In: www.daml.org/services/owl-s/ 20/12/2004.

- [DAT04a] DATASUS. Catálogo de Aplicações. In: www.datasus.gov.br/catalogo/catprod.htm, dezembro, 2004.
- [DAT04b] DataSus, Ministério da Saúde, População Residente por Unidade de Federação segundo a Região - Ano 2004; tabnet.datasus.gov.br/cgi/ibge/popmap.htm 10/01/2005.
- [DAV83] DAVIS R.; SMITH R.G. Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence*, 20(1), 1983, p.63-109.
- [DAV94] DAVIES W.; EDWARDS P. Agent K: An Integration de AOP and KQML. Technical Report AUCS/TR9406, University of Aberdeen, Scotland, UK, 1994.
- [DEG96] DEGOULET, P., FIESCHI, M.; Introduction to Clinical Informatics. Springer-Verlag, New York, 1996.
- [DEM91] DEMAZEAU Y.; MÜLLER J. P. From reactive to intentional agents, *Decentralized Artificial Intelligence*, Vol.2, p.3-10, Elsevier Science Publishers, North Holland, 1991.
- [DUR87] DURFEE E.; LESSER V.R.; CORKILL D.D. Cooperation through Communication in a Distributed Problem Solving Network. *IEEE Transactions on Computers*, Vol.36 (11), 1987.
- [EBP04] Business Process Modeling Languages, Business Process Execution Language for *Web Services*, www.ebpm.org/bpel4ws.htm 27/12/2004.
- [ERC91] ERCEAU J.; FERBER J. L'Intelligence Artificielle Distribuée. *La Recherche*, Juin 1991.
- [FAL01] Fallside, David C., XML Schema Part 0: Primer Second Edition, World Wide Web Consortium. Outubro de 2004.
- [FAL98] FALBO, R. A. Integração de Conhecimento em um Ambiente de Desenvolvimento de Software. Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ, Rio de Janeiro, Dezembro de 1998.

- [FER90] FERBER J.; JACOPIN E. The Framework of Eco-Problem Solving. Proceedings of the second MAAMAW, Saint-Quentin en Yvelines, August, 1990, 13-16.
- [FER92] FERBER J.; DROGOUL A. Using Reactive Multi-Agent Systems in Simulation and Problem Solving. Avouris N.M., Gasser L., eds., Distributed Artificial Intelligence: Theory and Praxis, Kluwer Academic Publishers, 1992, p.53-80.
- [FER94] FERBER J. La Kénétique: des systèmes multi-agents à une science de l'interaction. Revue Internationale de systémique, Vol.8, no.1, 1994, p.13-27.
- [FER95] FERBER J. Les Systèmes Multi-Agents. Interéditions, Paris, 1995.
- [FER99] Ferber J., Multi-Agent System: An Introduction to Distributed Artificial Intelligence, Ed. Longman Publishing Co., Boston, EUA 1999.
- [FIE99] Fielding, R. et al., Hypertext Transfer Protocol, Internet Engineering Task Force RFC 2616 Junho 1999.
- [FIN94] FININ, T.; FRITZON, R.; MCKAY, D.; MCENTIRE, R. KQML - A Language and Protocol for Knowledge and Information Exchange. Computer Science Department. Universit of Maryland Baltimore, 1994.
- [FIP02a] FIPA TC COMMUNICATION. FIPA Communicative Act Library Specification SC00037J, Foundation for Intelligent Physical Agents. Dezembro 2002.
- [FIP02b] FIPA TC COMMUNICATION. FIPA ACL Message Structure Specification SC00061G, Foundation for Intelligent Physical Agents. Dezembro 2002.
- [FIP02c] FIPA TC COMMUNICATION. FIPA Request Interaction Protocol Specification SC00026H. Foundation for Intelligent Physical Agents. Dezembro 2002.
- [FIP02d] FIPA TC COMMUNICATION. FIPA SL Content Language Specification SC00008I. Foundation for Intelligent Physical Agents, dezembro, 2002.

- [FIP03a] FIPA Modeling Technical Committee, Modeling Work Plan F-WP-00022, Foundation for Intelligent Physical Agents. Fevereiro de 2003.
- [FIP03b] FIPA Modeling Technical Committee. *FIPA Modeling: Agent Class Diagrams*. Foundation for Intelligent Physical Agents, fevereiro, 2003.
- [FIP05] FIPA Modeling Technical Committee, Agent UML, Foundation for Intelligent Physical Agents, www.auml.org 05/01/2005.
- [FRE03] Freitas, F.; Bittencourt, G. 2003. An Ontology-based Architecture for Cooperative Information Agents. A ser publicado nos Proceedings of the Internacional Joint Conference on Artificial Intelligence - IJCAI'2003. Acapulco, México.
- [GAR94] GARLAN, D.; SHAW, M. An Introduction to Software Architecture. School of Computer Science. Carnigie Mellon University. Pittsburgh, 1994.
- [GEN91] GENESERETH M.R. Knowledge Interchange Format. Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference. Morgan Kaufmann, p.599-600.
- [GEN92] GENESERETH M.R., FIKES R.E. Knowledge Interchange Format, version 3.0 reference manual. Rapport Technique n° Logic-92-1, Computer Science Department, Stanford University, 1992.
- [GEN94] GENESERETH M.R.; KETCHPEL S.P. *Software Agents*. Communications of the ACM. Vol.37, no.7, julho, 1994, p.48-53.
- [GEO83] GEORGEFF M.P. Communication and Interaction in Multi-Agent Planing. Proceedings of the Third National Conference on Artificial Intelligence, AAAI-83, 1983, p.125-129.
- [GOM99] GÓMEZ-PÉREZ, A. Tutorial on Ontological Engineering. Proceedings IJCAI'99. Sixteenth International Joint Conference on Artificial Intelligence 1999.

- [GRE02] GREIN, D.; MACHADO, D.A.; DIAS, J.S. Proposta de Integração de Sistema de Informação em Saúde com o Cartão Nacional de Saúde. VIII Congresso Brasileiro de Informática em Saúde, Natal, 2002.
- [GRU04] GRUBER, Thomas R. What is an Ontology?. Disponível em: <<http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>>, Novembro 2004.
- [GRU91] GRUBER, Thomas R. The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases. Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning; San Mateo, CA, 1991.
- [GRU92] GRUBER, T.R. Ontolingua: A mechanism to support portable ontologies - version 3.0. Technical Report, Knowledge Systems Laboratory, Stanford University, California, 1992.
- [GRU93] GRUBER, Thomas R. A Translation Approach to Portable Ontology Specifications. Technical Reporter KSL 92-71. In: Knowledge Acquisition. Academic Press, 1993, pp 199-220.
- [GUA98a] GUARINO, N.; WELTY, C. Conceptual Modeling and ontological analysis. Padova: LADSEB-CNR, 1998.
- [GUA98b] GUARINO, N. Formal Ontology and Information Systems. In: Proceedings 1st International Conference Formal Ontology in Information Systems, June 1998. Trento, Italy.
- [GUD03] Gudgin, Martin et al., Simple Object Access Protocol - SOAP 1.2, World Wide Web Consortium. Junho 2003.
- [GUE96] GUESSOUM Z. Un environnement opérationnel de conception et réalisation de systèmes multi-agents, Thèse de doctorat, Université Pierre et Marie Curie, avril, 1996.
- [HAY90] HAYES-ROTH B. Architectural Foundations for Real-Time Performance in Intelligent Agents. Real-Time Systems, may, 1990.

- [HEN02] HENDLER, J. Integrating Applications on the Semantic Web. Journal of the Institute of Electrical Engineers of Japan, v. 122(10), outubro, 2002, p.676-680.
- [HEX02] HEXSEL, R.A.; URBAN, A.; BARROS, R.S.M. Arquitetura do Sistema Cartão Nacional de Saúde. VIII Congresso Brasileiro de Informática em Saúde, Natal, 2002.
- [HOH03] HOHMANN, L. Beyond Software Architecture. Addison Wesley Professional; 1st Ed., ISBN: 0201775948, Janeiro, 2003
- [ILH93] ILHA JAIME DE O.; O Registro Clínico Computadorizado: Padronização e Codificação. Revista Informática, 1 (5): 5-8, 1993. www.epub.org.br/informed;28/03/2002.
- [JEN92] Jennings N.R.; Wittig T. ARCHON: Theory and Practice. Distributed Artificial Intelligence: Theory and Praxis (eds. N.M. Avouris and L.Gasser), Kluwer Academic Press, 1992, p.179-195.
- [KAV04] Kavantzas, Nickolas, *Web Services Choreography Description Language Version 1.0*, World Wide Web Consortium. Dezembro de 2004.
- [KOI01] KOIVUNEN, M., MILLER, E. *W3C Semantic Web Activity*. In: <http://www.w3.org/2001/12/semweb-fin/w3csw>, 2001.
- [KUO95] KUOKKA D.; HARADA L. Matchmaking for information agents. Proceedings of the Joint Conference on Artificial Intelligence, 1995.
- [KUO95a] Kuokka D., Harada L., (1995a), A Communication infrastructure for concurrent engineering, In : Journal of Artificial Intelligence in Engineering, Design, Analysis, and Manufacturing.
- [KUO95b] Kuokka D., Harada L., (1995b), Matchmaking for information agents, Proceedings of the Joint Conference on Artificial Intelligence.
- [LAR00] Larman, Craig, Utilizando UML e Padrões: uma introdução à análise e projeto orientados a objetos, Ed. Bookman, Porto Alegre 2000.

- [LAS99] LASSILA, O.; SWICK, R.R. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, In: <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>, fevereiro, 1999.
- [LEA01] LEÃO, B.F. The Brazilian National Health Informatics Strategy. Proceedings MEDINFO 2001, Londres, outubro, 2001.
- [LEA02] LEÃO, B. F.; SILVA, J. Z.; TOLEDO, F. B.; CUNHA, R. E. Agente Cartão: Integração de Sistemas Externos ao Sistema Cartão Nacional de Saúde. VIII Congresso Brasileiro de Informática em Saúde, Natal, 2002.
- [LEY01] Leymann, Frank, *Web Services Flow Language (WSFL 1.0)*, IBM Software Group. Maio de 2001.
- [LIB04] Liberty Alliance Project, www.projectliberty.org 03/01/2005.
- [LOP04] LOPES, J.P.; PINHEIRO, L.F.R.; CARVALHO, N.L.; VALASKI, M.R.; DOMINGUEZ, O.; MORAES, J.F.T.; MELLO E SILVA, A.T.G. Cartão Nacional de Saúde: Projeto Piloto no Estado do Paraná Avaliação do Grau de Satisfação dos Gestores Municipais. IX Congresso Brasileiro de Informática em Saúde, Ribeirão Preto, 2004.
- [LOU03] LOUREIRO, S. Sistema Único de Informação em Saúde: Integração dos dados da Assistência Suplementar à Saúde ao Sistema SUS. Agencia Nacional de Saúde, julho de 2003.
- [MAC05] MACHADO, D. A. *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, Pontifícia Universidade Católica do Paraná, Curitiba, 2005.
- [MAE94a] MAES P. Agents that reduce work and information overload. *Communications of the ACM*, 37(7). Julho 1994.

- [MAE94b] MAES P. Social interface agents: Acquiring competence by learning from users and other agents. O. Etzioni (eds.) Software Agents, Spring Symposium (Technical Report SS-94-03), AAAI Press, 1994, p.71-78.
- [MAN01] MANSUR, M. C.; COSTA, N. R. O Financiamento Federal da Saúde no Brasil. Ministério da Saúde, Fundação Oswaldo Cruz, Escola Nacional de Saúde Pública. Rio de Janeiro, agosto de 2001.
- [MAN04] Manola, Frank, Resource Description Framework, World Wide Web Consortium. Fevereiro 2004.
- [MAR99] MARTIN, D.L.; CHEYER, A.J.; MORAN, D.B. The Open Agent Architecture: A Framework for Building Distributed Software Systems. Applied Artificial Intelligence, vol.13, no.1-2, 1999, p.91-128. Disponível em <http://www.ai.sri.com/oa/>.
- [MAY95] MAYFIELD J.; LABROU Y.; FININ T. Evaluation of KQML as an Agent Communication Language. Intelligent Agents. Proc. of the 1995 Workshop on Agents Theories, Vol.2, Architecture, and Languages, M. Wooldridge, J-P. Müller and M. Tambe (eds), Lecture Notes in Artificial Intelligence, Springer-Verlag, 1995.
- [MCG00] McGuinness, D., Fikes, R., Rice, J., Wilder, J.. The Chimaera Ontology Environment. Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI 2000). Austin, Texas. July 30 - August 3, 2000.
- [MCG93a] McGuire J., Kuokka D., Weber L., Tenenbaum J., Gruber T., Olsen G, (1993), SHADE : Technology for knowledge-based Collaborative Engineering, In : Concurrent Engineering Research and Application, vol 1(3).
- [MCG93b] MCGUIRE, J.; PELAVIN, R.; SHAPIRO, S.; BECK, C.; FININ, T.; WEBER, J., WIEDERHOLD, G.; GENESERETH, M.; FRITZSON, R.; MCKAY, D. Draft: Specification of the KQML Agent-Communication Language, 1993.

- [MCG98] McGUINNESS, D.L.; WRIGHT, J. Conceptual modeling for configuration: A description logic-based approach. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing – special issue on Configuration*, 1998.
- [MCG98] MCGRATH, S. *XML By Example*. Prentice Hall PTR, 1st ed., 1998.
- [MED94] MÉDICI, A. C. & MARQUES, R. M.. Saúde: entre gastos e resultados. *Boletim de Conjuntura Política Social*, 1994.
- [MIN04a] MINISTÉRIO DA SAÚDE. *Saúde Brasil 2004 - Uma Análise da Situação de Saúde*. Ministério da Saúde, Brasília, maio, 2004.
- [MIN04b] MINISTÉRIO DA SAÚDE. Sistema de Informação sobre Mortalidade (SIM). In: www.saude.gov.br/svs/sis/sis00_sim.htm, Secretaria de Vigilância em Saúde, agosto, 2004.
- [MIN04c] MINISTÉRIO DA SAÚDE. Sistema de Informações sobre Nascidos Vivos (SINASC). Secretaria de Vigilância em Saúde, In: www.saude.gov.br/svs/sis/sis00_sim.htm, agosto, 2004.
- [MIN04d] MINISTÉRIO DA SAÚDE. *Manual do Sistema de Informações Hospitalares do SUS*. Secretaria de Atenção à Saúde, Brasília, dezembro, 2004.
- [MIN04e] MINISTÉRIO DA SAÚDE. *Manual de Instrumentação do SIA*. In: <http://dtr2001.saude.gov.br/sas/decas/apresent.mansia.htm>, Secretaria de Atenção à Saúde, janeiro, 2004.
- [MIN05a] Ministério da Saúde. DATASUS. *Cadastro Nacional de Estabelecimentos de Saúde*. In: <http://cnes.datasus.gov.br>, janeiro, 2005.
- [MIN05b] Ministério da Saúde. MS/SVS/DASIS. *Sistema de Informações sobre Nascidos Vivos (SINASC)*, dezembro, 2005.
- [MIN05c] Ministério da Saúde. DATASUS. *Sistema de Informações Ambulatoriais do SUS (SIA/SUS)*, março, 2005.

- [MIN05d] Ministério da Saúde. DATASUS. Sistema de Informações Hospitalares do SUS (SIH/SUS), março, 2005.
- [MIN86] MINSKY M. The Society of Mind, Simon and Schuster, New York, 1986.
- [MIN98] MINISTÉRIO DA SAÚDE. Guia de Vigilância Epidemiológica. Fundação Nacional de Saúde, Centro Nacional de Epidemiologia, 1998.
- [MIT94] Mitchell T.; Caruana R.; Freitag D.; McDermott J.; Zabowski D. Experience with a learning personal assistant, Communications of the ACM, 37(7), Julho 1994.
- [MIZ95] MIZOGUCHI, R.; VANWELKENHUYSEN, J.; IKEDA, M. Task Ontology for Reuse of Problem Solving Knowledge. Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing. IOS Press. 1995. 46-59.
- [Mos92] MOSSES, P. Action Semantics. Cambridge: Cambridge University Press, 1992.
- [MYE04] Myerson, Judith M, Testing for SOAP Interoperability & Testing *Web Services* Clients. *Web Services Architect*.
www.webservicesarchitect.com/content/articles/myerson02.asp 14/12/04.
- [NAR02] NARDON, F.B. Introdução ao XML; INCOR - HCFMUSP - Instituto do Coração do Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo. www.incor.usp.br/spdweb/xml/xml.pdf; Março 2002.
- [NEW82] Newell, A.; 1982. The Knowledge Level. Artificial Intelligence 18(1):87-127.
- [NOA02] Norma Operacional da Assistência à Saúde, NOAS01/02, Portaria MS/GM n.º 373, de 27 de fevereiro de 2002.
- [NOB93] NORMA OPERACIONAL BÁSICA DO SISTEMA ÚNICO DE SAÚDE. Ministério da Saúde, Brasília, 1993.
- [NOB96] NORMA OPERACIONAL BÁSICA DO SISTEMA ÚNICO DE SAÚDE. Ministério da Saúde, Brasília 1996.

- [NOV04a] Novell Company, *Understanding Web Services*. In: www.novell.com/documentation/workbench41/docs/help/books/devUnderstandingServices.html, 15/12/2004.
- [NOV04b] Novell Company, *UDDI Server 2.1 Administration Guide* www.novell.com/documentation/nw65/uddi/data/aio19.html 15/12/2004
- [NOY00a] NOY, Natalya F.; McGUINNESS, Deborah L. *Ontology development 101: a guide to creating your first ontology*. Stanford University, CA, 2000.
- [NOY00b] Noy, N., Ferguson, R., Musen., M. 2000. The knowledge model of Protege-2000: Combining interoperability and flexibility. 2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), Juan-les-Pins, France.
- [O'LE97] O'LEARY, D.E., 1997, "Impediments in the use of explicit ontologies for KBS development", *Int. J. Human-Computer Studies*, v. 46, n. 2/3.
- [OAS04] Organization for the Advancement of Structured Information Standards <http://www.oasis-open.org/> 03/01/2005.
- [ODE00] ODELL, J. *Agent Technology Green Paper Version 1.0*. Agent Working Group OMG Document, Agosto, 2000.
- [OMG05] Object Management Group, *UML - Unified Modeling Language*, www.uml.org 10/01/2005.
- [OMG99] OMG CORBA. *The Common Object Request Broker: Architecture and Specification*. Object Management Group, Cap.3-17, 1999. Disponível em: <http://www.omg.org/docs/formal/99-10-07.pdf>
- [OSI89] Open Systems Interconnection, International Organization for Standardization ICS 35.100.01, 1989.
- [PEL03] Peltz, Chris, *Web Services Orchestration*, Hewlett Packard, Co, Janeiro 2003.

- [POS82] Postel, Jonathan B., Simple Mail Transfer Protocol, Internet Engineering Task Force RFC821 Agosto de 1982.
- [RAJ98] Raj, Gopalan S., A Detailed Comparison of CORBA, DCOM, and Java/RMI, Object Management Group (OMG). Setembro de 1998.
- [REV02] Revista Saúde Pública vol.34 n.5 São Paulo Oct. 2000. www.fsp.usp.br/~rsp/ - 28/03/2002.
- [RNIS04] Rede Nacional de Informações em Saúde – DataSus - www.datasus.gov.br/rnis; 05/12/2004.
- [RUS95] RUSSEL, S.; NORVIG, P. Artificial Intelligence: A Modern Approach, Cap.12.1-12.3, Prentice Hall, 1995, p.367-380.
- [SAN93] SANDHOLM T. An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations. Proceedings of the Eleventh National Conference on AI, Washington, DC, July II-15, 1993, p.256-262.
- [SCA96] SCALABRIN, E. E. Conception et Réalisation d'environnement de développement de systèmes d'agents cognitifs. Tese de Doutorado, Université de Technologie de Compiègne, France, 1996.
- [SCA96b] SCALABRIN, E., CARVALHO, D., ANGELOTTI, E., AZEVEDO, H., RAMOS, M. An Open Multiagent Architecture to Improve Reliability and Adaptability of Systems, 1996.
- [SCH04a] Schema for the SOAP 1.1 encoding, Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique <http://schemas.xmlsoap.org/soap/encoding/> 27/12/2004.
- [SCH04b] Schema for the SOAP/1.1 envelope, Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique <http://schemas.xmlsoap.org/soap/envelope/> 27/12/2004.

- [SEN88a] SENADO FEDERAL. Constituição Federal de 1988. Senado Federal, Brasília, 1988.
- [SEN88b] SENADO FEDERAL. Inciso I do art. 198 da Constituição Federal de 1988, Senado Federal, Brasília, 1988.
- [SEN90] SENADO FEDERAL. Lei Nº 8.142 de 28 de dezembro de 1990. Senado Federal, Brasília, 1990.
- [SHM97] SHMEIL, M.A.H.; OLIVEIRA, E. The establishment of partnerships to create Virtual Organizations: a Multi-agent approach. Camarinha-Matos (ed.), Re-engineering for sustainable industrial Process, Chapman & Hall, 1997, p.284-295.
- [SHM99] SHMEIL, M.A.H. Sistemas Multi-Agente na Modelação da estrutura e relações de contratação de Organizações, Tese de Doutorado, Universidade do Porto, 1999.
- [SHO90] Shortliffe E. H. Wiederhold G., Perreault I. E. Fagan I. M.; Medical Informatics; Computer Applications in Health Care. New York: Addison-Wesley Publishing Company, 1990. I – 2, 4, II - 6, 7.
- [SMI80] SMITH, R. The contract-net protocol. IEEE Trans on Computers, C29(12), 1980, p.1104-1113.
- [SMI88] SMITH R. G.; DAVIS R. Frameworks for Cooperation in Distributed Problem Solving. Readings in Distributed Artificial Intelligence, A.H. Bond & L. Gasser, editors, Morgan Kaufmann Publishers, San Mateo, California, 1988, p.61-70.
- [SNE02] Snell, James, Business Process Modeling with WSFL, IBM. Junho 2001. www-106.ibm.com/developerworks/webservices/library/ws-ref4 27/12/2004.
- [STE81] STEEB R.; CAMMARATA, S.; HAYES-ROTH F.A.; THORNDYKE, P.W.; WESSON, R.B. Architectures for Distributed Intelligence Air Fleet Control. Rapport technique n° R-2728-ARPA, Rand Corporation, Santa Monica, 1981.

- [STE89] STEELS L. Cooperation between Distributed Agents Through Self-Organization. Proceedings of the First MAAMAW, Cambridge, England, August 1989, p.16-18.
- [STU98] STUDER, R.; BENJAMINS, V.R.; FENSEL, D. "Knowledge Engineering: Principles and Methods". Data and Knowledge Engineering. 25 (1998) 161-197.
- [SWA97] SWARTOUT, B., PATIL, R., KNIGHT, K., RUSS, T., 1997, "Toward Distributed Use of Large-Scale Ontologies". Ontological Engineering - Working Notes, Stanford, California, March.
- [THO04] THOMPSON, H.S.; BEECH, D.; MALONEY, M.; MENDELSON, N. XML Schema Part 1: Structures Second Edition. World Wide Web Consortium (W3C), In: <http://www.w3.org/TR/xmlschema-1/>, outubro, 2004.
- [VAL95] VALENTE, A. Legal Knowledge Engineering - A Modelling Approach. IOS Press, 1995.
- [VAN96] VAN HEIJST, G.; VAN DER SPEK, R.; KRUIZINGA, E. Organizing Corporate Memories. In B, Gaines, M. Musen Eds. Proceedings of Knowledge Acquisition Workshop'96, Banff, Canada, November, pp. 42-1 42-17. 1996.
- [VAN97] VAN HEIJST, G.; SCHREIBER, T.; WIELINGA, B. Using Explicit Ontologies in KBS. International Journal of Human-Computer Studies. Vol. 46 (2/3). 183-292. 1997.
- [WOO94] WOOLDRIDGE M.J., JENNINGS N.R. Agent Theories, Architectures, and Languages: A Survey. ECAI'94, Workshop on Agent Theories, Architectures and Languages, Amsterdam, The Netherlands, 1994, p.1-32.
- [WOO95] WOOLDRIDGE M.; JENNINGS N. Intelligent agents: theory and practice. The knowledge Engineering Review, Vol.10, no.2, 1995, p. 115-152.
- [WSI04] *Web Services Interoperability Organization* <http://www.ws-i.org/> 03/01/2005.
- [YOU90] YOURDON, E. Análise estruturada moderna. Rio de Janeiro: Campus, 1990.