



UNIVERSIDADE SALVADOR – UNIFACS
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E
COMPUTAÇÃO
MESTRADO ACADÊMICO EM SISTEMAS E COMPUTAÇÃO

HERBERT MONTEIRO SOUZA

UM SERVIÇO DE PUBLICAÇÃO E DESCOBERTA DE SERVIÇOS
PARA INFRA-ESTRUTURAS DE MONITORAMENTO DE REDE

Salvador
2007

HERBERT MONTEIRO SOUZA

**UM SERVIÇO DE PUBLICAÇÃO E DESCOBERTA DE SERVIÇOS
PARA INFRA-ESTRUTURAS DE MONITORAMENTO DE REDE**

Dissertação apresentada ao Programa de Pós-Graduação em Sistemas e Computação da Universidade Salvador – UNIFACS, como requisito parcial para obtenção do título de Mestre.

Orientador: Prof. Dr. José Augusto Suruagy Monteiro

Co-orientador: Prof. Leobino Nascimento Sampaio

Salvador
2007

FICHA CATALOGRÁFICA

(Elaborada pelo Sistema de Bibliotecas da Universidade Salvador - UNIFACS)

Souza, Herbert Monteiro

Um serviço de publicação e descoberta de serviços para infra-estrutura de monitoramento de rede / Herbert Monteiro Souza. - 2007.

164 f. : il.

Dissertação (Mestrado) - Universidade Salvador – UNIFACS.

Mestrado em Sistemas de Computação, 2007.

Orientadora: Prof. Dr. José Augusto Suruagy Monteiro

1 .Serviços. Web 2. XML. 3. Redes e Sistemas Distribuídos. 4. Arquitetura de Redes e Computadores. I. Monteiro, José Augusto Suruagy, orient. II. Título.

CDD: 004.048

TERMO DE APROVAÇÃO

HERBERT MONTEIRO SOUZA

UM SERVIÇO DE PUBLICAÇÃO E DESCOBERTA DE
SERVIÇOS PARA INFRA-ESTRUTURAS DE
MONITORAMENTO DE REDES

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Sistemas e Computação, Universidade Salvador – UNIFACS, pela seguinte banca examinadora:

José Augusto Suruagy Monteiro – Orientador _____
Ph. D. em Ciência da Computação, University of California - UCLA
Universidade Salvador – UNIFACS

Joberto Sérgio Barbosa Martins _____
Doutor em Ciência da Computação, Université Paris VI
Universidade Salvador – UNIFACS

Daniela Barreiro Claro _____
Doutora em Informática, Université Angers, UA, França
Universidade Federal da Bahia – UFBA

Salvador, 20 de Dezembro de 2007

Dedico este trabalho aos meus pais, Gildásio e Delenice, a minha namorada Maria Margarete, aos meus irmãos Gilbert e Isabele e aos meus sobrinhos Diego, Duda e Rayssa

AGRADECIMENTOS

Muitas pessoas contribuíram para que eu superasse mais uma etapa importante da minha carreira. No entanto, eu não poderia deixar de agradecer primeiro ao alicerce da minha vida. O homem ao qual eu procuro ser a sua semelhança, e que motivado por essa procura eu continuo a vencer. Obrigado Deus!

Ao meu orientador Prof. Suruagy, que desde a época da graduação se tornou para mim um exemplo de profissional, pela confiança depositada para realizar este trabalho e muitos outros em nosso grupo de pesquisa, pela orientação dada neste trabalho e por me inserir no meio científico da pesquisa.

Ao Prof. Leobino, pela co-orientação valiosa neste trabalho, pelo incentivo que me fez lembrar as minhas responsabilidades, pela companhia na elaboração dessa dissertação e pelo apoio realizado na pesquisa.

A minha namorada Margarete, pela compreensão em relação à ausência no período de construção deste trabalho.

A RNP e FAPESB, pelo apoio financeiro, pela infra-estrutura de equipamentos e pela importância dada ao nosso grupo de pesquisa.

Ao meu amigo e colega de mestrado Ivo Kenji, que pelo fato de conduzir seus trabalhos em paralelo me ajudou muito na tomada de decisões e elaboração desta dissertação.

Ao meu amigo e colega de pesquisa Rafael Costa, pela ajuda na correção deste trabalho e pelo companheirismo na pesquisa.

Aos meus colegas de pesquisa e amigos Dimitri, Cayo, Marcelo, Priscilla e Walter, pelo apoio no laboratório, pela descontração nos momentos de tensão e pelo apoio nas pesquisas realizadas no grupo.

A toda equipe do NUPERC, que manteve toda infra-estrutura em funcionamento para a realização deste trabalho. Gostaria de agradecer também a Jane e Izabel, pelo acompanhamento nas questões acadêmicas durante o período do mestrado.

Finalmente a todos os meus amigos e membros da minha família que sempre estiveram e estão ao meu lado.

Disse-lhe Jesus: Porque me viste, Tomé, creste;
bem-aventurados os que não viram e creram.
—cf. João 20, 29

RESUMO

As infra-estruturas de monitoramento em redes de computadores vêm adotando como padrão a arquitetura orientada a serviços (SOA) com a intenção de oferecer serviços de medição em múltiplos domínios administrativos. Para usufruir de todas as vantagens do SOA é necessário também utilizar todos os três componentes fundamentais desta arquitetura, ou seja, o consumidor, o provedor de serviços e também a tecnologia de publicação e descoberta para sanar o problema de localização e acesso aos serviços. Neste sentido existe atualmente uma forte tendência para a utilização de Serviços Web como a tecnologia para a implantação dos serviços nas infra-estruturas de monitoramento. E é neste aspecto que este trabalho está focado, ou seja, ele propõe um Serviço de Publicação e Descoberta voltado para Serviços Web de monitoramento de redes. Antes desta proposta foi necessário verificar os padrões e tecnologias disponíveis que possivelmente podem auxiliar a construção do serviço proposto. Portanto são apresentadas as arquiteturas orientadas a serviços, enfatizando a publicação e descoberta. Também é mostrado alguns padrões que definem serviços de publicação e descoberta como o UDDI e o ebXML, bem como as tecnologia para desenvolvimento e implantação da SOA. O UDDI é um componente crucial para um bom uso de Serviços Web, pois cria uma interface padrão que possibilita que as aplicações localizem rapidamente, facilmente e dinamicamente os Serviços Web na Internet. Este trabalho usa na sua proposta este padrão como base do serviço de publicação e descoberta e incorpora nas funcionalidades do UDDI as características dos Serviços Web e das infra-estruturas de monitoramento. Assim, características como: nomenclaturas, tipos, funcionalidades e etc., podem ser usadas como taxonomias para facilitar a interação com o Serviço de Publicação e Descoberta proposto. Pesquisando todas as características que envolvem as infra-estruturas de monitoramento em relação à publicação e descoberta de serviços e verificando as peculiaridades que envolvem os Serviços Web de monitoramento, foi possível definir, desenvolver e testar um Serviço de Publicação e Descoberta para Serviços Web de medição. Este serviço faz parte dos componentes da infra-estrutura de monitoramento piPEs-BR/GFD e pode ser utilizada por qualquer aplicação que deseja localizar e descobrir os Serviço Web dessa infra-estrutura.

Palavras-chave: Serviços Web. XML. Redes e Sistemas Distribuídos. Arquitetura de redes e computadores.

ABSTRACT

The network performance monitoring infrastructures have been adopting the service oriented architecture (SOA) standard with the intention of offering measurement services in multiple administrative domains. To enjoy all the benefits of SOA it is also necessary to use the three components of this architecture, in other words, the service provider, the consumer and the publish and discovery technology to solve the problem of services search and access. In this way there is currently a strong trend towards the use of Web Services as a technology for deployment of services in network monitoring infrastructures. And this is where this work is focused, in other words, it proposes a Publish and Discovery Service to the network monitoring Web Services. Before of this proposal it was necessary to verify the standards and technologies available that could make possible the construction of the proposed service. Therefore we present the service oriented architectures, emphasizing the publication and discovery. It is also shown some standards that define services, publishing and discovery as UDDI and ebXML, and the technologies for development and deployment of SOA. The UDDI is a key component for a good use of Web services, because it creates standard interfaces that enables the applications quickly, easily and dynamically locate Web Services on the Internet. This work uses in its proposal this standard as the basis of the service publication and discovery and incorporates the UDDI features in the network monitoring Web services characteristics and the monitoring infrastructure characteristics. Thus, characteristics such as: classifications, types, features and so on, can be used as taxonomy to facilitate interaction with the Discovery and Publication Service proposed. Researching of all the features that involve the monitoring infrastructure in relation to the publication and discovery of services and verifying the peculiarities that involving the Web Services Web, it was possible to define, develop and test a service for Publishing and Discovery of network monitoring Web Services. This service is part of the of the piPEs-BR/GFD infrastructure and can be used by any application that want discover or publish the piPEs-BR/GFD Web Services.

Keywords: Web Service. XML. Network Computer and Distributed Systems. Computer network architecture.

LISTA DE FIGURAS

Figura 1.1 - Arquitetura do piPEs-BR	18
Figura 2.1 - Arquitetura Orientada a Serviços	28
Figura 2.2 - Funcionamento do ebXML	35
Figura 2.3 - Componentes do UDDI	37
Figura 3.1 - Arquitetura do GFD	48
Figura 3.2 - Cenário de medições com a infra-estrutura pingER	53
Figura 3.3 - de visualização da MonALISA	54
Figura 3.4 - Interação entre os diversos componentes do MonALISA	55
Figura 3.5 - Componentes do protótipo perfSONAR 1.0	57
Figura 3.6 - Estrutura da mensagem NMWG	58
Figura 3.7 - Arquitetura do piPEs-BR/GFD	61
Figura 4.1 - Classificação dos serviços pelo tipo	69
Figura 4.2 - Características de rede usadas para descrever o comportamento dos dispositivos de redes	71
Figura 4.3 - Modelo de dados	75
Figura 4.4 - Exemplo de relacionamento de taxonomias	77
Figura 4.5 - Exemplo das taxonomias criadas	77
Figura 4.6 - Estrutura macro do Serviço de Publicação e Descoberta proposto	82
Figura 5.1 - Arquitetura do Serviço de Publicação e Descoberta	87
Figura 5.2 - Diagrama de componentes do padrão perfSONAR	88
Figura 5.3 - de seqüência do Serviço de Publicação e Descoberta	92
Figura 5.4 - Mapeamento da mensagem NMWG nos componentes do UDDI	93
Figura 5.5 - Diagrama de classe básico para a biblioteca de acesso ao UDDI	95
Figura 5.6 - Arquitetura da biblioteca de acesso ao UDDI	96
Figura 5.7 - Cenário de testes n° 1	97
Figura 5.8 - Código usado na aplicação de testes do cenário n° 1	98
Figura 5.9 - Cenário de testes n° 2	99
Figura 5.10 - Comparação de desempenho entre o LS-UDDI e o LS-XML	101
Figura 9.1 - Tela de acesso ao Serviço de Publicação e Descoberta	116
Figura 9.2 - Tela de configuração	117
Figura 9.3 - Tela do <i>UDDI Browser</i>	118

LISTA DE TABELAS

Tabela 10.1 - Resultado dos testes realizados com o LS-UDDI.	119
Tabela 10.2 - Resultado dos testes realizados com o LS-XML.	119

LISTA DE QUADROS

Quadro 2.1 - Exemplo de conjunto de características de um serviço	39
Quadro 4.1 - Informações de tipos de serviços do piPEs-BR/GFD	68
Quadro 4.2 - Nomenclatura padrão de várias combinações de características	70
Quadro 4.3 - Componentes do UDDI usados no serviço de publicação e descoberta do piPEs-BR/GFD	74
Quadro 5.1 - Níveis de <i>Log</i> do sistema	90

SUMÁRIO

1	INTRODUÇÃO	16
1.1	CONTEXTO	16
1.2	JUSTIFICATIVAS	23
1.3	PROBLEMA E OBJETIVO	24
1.4	APRESENTAÇÃO	25
2	ARQUITETURA ORIENTADA A SERVIÇOS – SOA	27
2.1	INTRODUÇÃO	27
2.2	SOA: PUBLICANDO E DESCOBRINDO SERVIÇOS	29
2.3	SERVIÇOS WEB	30
2.3.1	Web Service Definition Language – WSDL	32
2.4	PADRÕES PARA PUBLICAÇÃO E DESCOBERTA DE SERVIÇOS	32
2.4.1	ebXML - Comércio Eletrônico usando XML	33
2.4.2	Universal Description Discovery and Integration – UDDI	35
2.4.2.1	Componentes do UDDI	37
2.4.2.2	Categorização de serviços	39
2.4.2.3	UDDI versão três	41
2.5	TECNOLOGIAS PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOA	42
2.5.1	Apache Axis	42
2.5.2	JUDDI	43
2.5.3	UDDI4J	44
2.5.4	Web Sphere	44
2.5.5	Apache Tomcat	45
2.6	CONSIDERAÇÕES	45
3	INFRA-ESTRUTURAS DE MONITORAMENTO	47
3.1	GENERAL FRAMEWORK DESIGN (GFD)	47
3.2	INFRA-ESTRUTURAS DE MONITORAMENTO	51
3.2.1	PingER	51
3.2.2	MonALISA	53
3.2.3	perfSONAR	55
3.2.4	piPEs-BR/GFD	59

3.2.4.1	Descrição dos módulos	61
3.3	CONSIDERAÇÕES	62
4	PUBLICAÇÃO E DESCOBERTA DE SERVIÇOS DE MONITORAMENTO	66
4.1	INTRODUÇÃO	66
4.2	CARACTERÍSTICAS DOS SERVIÇOS DE MONITORAMENTO	67
4.2.1	Classificação por tipo de serviço	68
4.2.2	Usando a hierarquia das características de medições em redes	69
4.3	PADRÕES E TECNOLOGIAS A SEREM UTILIZADOS	72
4.4	MODELO DE DADOS	73
4.5	TAXONOMIAS NOS SERVIÇOS DE MONITORAMENTO	75
4.6	SERVIÇO DE PUBLICAÇÃO E DESCOBERTA PROPOSTO	78
4.6.1	Requisitos do piPEs-BR/GFD	78
4.6.2	Funcionalidades	80
4.6.3	Projeto do Serviço de Publicação e Descoberta	82
4.6.4	Integração com outras infra-estruturas	83
4.7	CONSIDERAÇÕES	83
5	PROTÓTIPO DESENVOLVIDO	85
5.1	INTRODUÇÃO	85
5.2	DESCRIÇÃO E DESENVOLVIMENTO DO PROTÓTIPO	87
5.2.1	Lookup Service UDDI Type - LS-UDDI	88
5.2.2	Biblioteca de acesso ao UDDI	92
5.3	TESTES	97
5.3.1	Cenário de teste nº 1	97
5.3.2	Cenário de teste nº 2	98
5.4	AVALIAÇÃO DOS RESULTADOS	100
6	CONSIDERAÇÕES FINAIS	104
6.1	CONCLUSÃO	104
6.2	CONTRIBUIÇÕES	106
6.3	TRABALHOS FUTUROS	107
	REFERÊNCIAS	109
	APÊNDICE A - EXEMPLO DE UTILIZAÇÃO DO UDDI4J	112

APÊNDICE B – MENSAGEM UTILIZADA NO TESTE DO CENÁRIO Nº 1	114
APÊNDICE C – CLIENTES DE ACESSO AO PROTÓTIPO DESENVOLVIDO	116
APÊNDICE D – RESULTADOS OBTIDOS NOS TESTES DO CENÁRIO Nº 1	119

1 INTRODUÇÃO

Este capítulo apresenta o contexto em que o trabalho está inserido; as justificativas para a realização do trabalho; o problema identificado e os objetivos a serem alcançados.

1.1 CONTEXTO

O uso de mecanismos de monitoramento no auxílio ao gerenciamento de redes de computadores, bem como no suporte de usuários para executarem suas aplicações; é uma realidade, pois cada vez mais a demanda por informações sobre as condições da rede vem aumentando. Esses mecanismos de averiguação e acompanhamento, inicialmente se limitavam a ferramentas de medição e ferramentas para análise dos dados medidos. Ao longo do tempo, as ferramentas de medições foram bastante difundidas e usadas. Logo, o cenário de medições nas redes de computadores se resume ao uso de um conjunto de ferramentas de medição que visam suprir demandas específicas de usuários, obtendo somente medidas fim-a-fim. No monitoramento fim-a-fim as informações coletadas são referentes a um caminho (conexão) entre dois hosts distintos, sem levar em consideração os dispositivos que compõem o caminho como: enlaces, roteadores, *switches* e etc.

Só que o uso de ferramentas para a coleta de dados fim-a-fim, geralmente não fornece as informações necessárias para identificar e solucionar problemas no caminho entre os pontos fim-a-fim. Logo, os usuários continuam sofrendo com problemas de conectividade e desempenho que podem ocorrer na sua própria máquina, na rede local ou nos dispositivos de rede que fazem parte do caminho fim-a-fim. Contudo, é necessário implementar mecanismos que possibilitem também fornecer informações sobre pontos intermediários de interesse que compõem um caminho fim-a-fim.

Assim, a partir desta necessidade de se obter o máximo possível de informações ao longo de um caminho fim-a-fim e fornecê-las a um amplo conjunto de aplicações e usuários, tornou-se necessária a criação de serviços de monitoração. Estes serviços fariam uso de diversas ferramentas e técnicas de medições existentes em vários pontos das redes, sendo que todo o

processo de medição é abstraído pelos usuários. Assim, tanto um usuário com pouco conhecimento sobre redes de computadores (usuário leigo), como um usuário avançado (administrador de rede), poderiam usufruir das funcionalidades desses serviços de monitoramento. Logo, esses serviços teriam a difícil tarefa de receber solicitações de dados de medição, manipular as ferramentas para coletar os dados solicitados, tratar os dados e fornecer os mesmos da forma desejada ao requisitante, sendo que todo esse processo deve ser realizado com segurança, dinamicidade e de forma controlada.

Com esta finalidade, apareceram no meio acadêmico algumas iniciativas para propor e desenvolver infra-estruturas de monitoramento que agregassem as funcionalidades citadas. Essas infra-estruturas são caracterizadas por operar de forma distribuída, pois para reunir o máximo de informação é necessário abranger o maior número possível de pontos das redes. No cenário internacional é possível citar a iniciativa de desempenho fim-a-fim *E2Epi-End to End performance initiative* da Internet2¹, a atividade de pesquisa em Medição e Monitoração de Desempenho (JRA1) da Géant2² e os estudo realizado pelo Grupo de Trabalho de Medições em Redes de Computadores (GT-Medições³) da Rede Nacional de Ensino e Pesquisa (RNP⁴), que buscam utilizar e desenvolver mecanismos de averiguação e acompanhamento que auxiliem no diagnóstico mais preciso dos problemas, bem como na sua caracterização para detecção futuras.

Numa tentativa inicial de disponibilizar uma infra-estrutura desse tipo, a Internet2 em 2003 definiu e iniciou o desenvolvimento do *performance initiative Performance Environment system* (piPEs) (BOYD, 2002) que utiliza diversas ferramentas para coletar e disponibilizar dados de medições. Já no cenário nacional, o GT-Medições, em paralelo, iniciou suas pesquisas no desenvolvimento de uma infra-estrutura de monitoramento para os seus usuários, pois o grupo já possuía uma grande experiência com as técnicas, métricas, ferramentas e dados de medições. O GT-Medições ao verificar um interesse em comum com a iniciativa do piPEs e ao verificar também que o protótipo do piPEs desenvolvido pela Internet2 não ficaria

¹ *Internet2 End-to-End Performance Initiative* é um grupo de pesquisa da rede acadêmica norte americana (Internet2) que procura solucionar problemas de desempenho nas redes.

² GN2 - JRA1 - *Performance Measurement and Monitoring*, projeto iniciado pela rede acadêmica européia (Géant2) para pesquisar sobre medições em redes.

³ Grupo de Trabalho apoiado pela RNP para disponibilizar mecanismo de monitoramento em sua rede.

⁴ Rede acadêmica brasileira que apóia o GT-Medições.

pronto dentro do cronograma do GT-Medições, resolveu então, desenvolver uma versão da infra-estrutura piPEs, de acordo com as necessidades da RNP e de acordo com a experiência do grupo com o cenário de medições. Assim surgiu a versão denominada piPEs-BR (SAMPAIO e outros, 2006), cuja arquitetura pode ser visualizada na Figura 1.1.

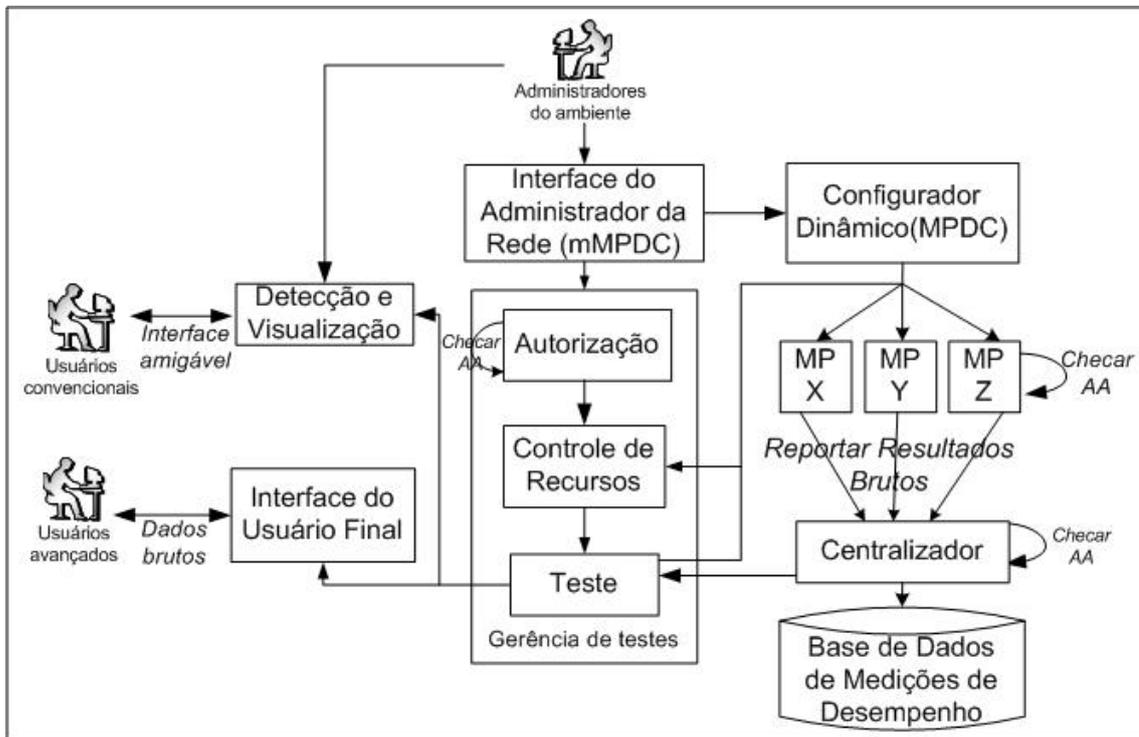


Figura 1.1 - Arquitetura do piPEs-BR
Fonte: Sampaio e outros (2006)

A Figura 1.1 mostra a versão inicial da arquitetura do piPEs-BR onde é possível visualizar cada um de seus componentes constituintes. No lado esquerdo da figura encontram-se as interfaces de acesso à infra-estrutura e os possíveis usuários. Na parte central estão os componentes que gerenciam a infra-estrutura e mais à direita encontram-se os componentes que manipulam as ferramentas de medição e armazenam dados. Assim, o piPEs-BR disponibiliza para a RNP uma infra-estrutura que realiza as medições através das ferramentas, armazena os dados medidos e disponibiliza esses dados para os usuários. No piPEs-BR a comunicação entre componentes é feita preferencialmente via Serviços Web (do inglês *Web Service* (ALONSO e outros, 2004), o que demonstra que o GT-Medições já se preocupava com interoperabilidade entre tecnologias. Um exemplo que é possível citar é a interface com o cliente de visualização que disponibilizava Serviços Web para que qualquer cliente de visualização (com qualquer tecnologia) pudesse interagir com a infra-estrutura.

As infra-estruturas piPEs e piPEs-BR além de fornecer informações fim-a-fim pretendem fornecer também informações em pontos estratégicos ao longo desses caminhos fim-a-fim. Esse tipo de abordagem pode envolver vários domínios administrativos diferentes, pois a origem e destino de um caminho fim-a-fim podem estar em domínios diferentes. Assim, essas duas infra-estruturas são caracterizadas pelo uso de pontos de medição (*Measurement Point - MP*), onde são obtidos os dados de medição, seja através da execução de várias ferramentas de medição ou através da coleta de dados. Essa abordagem do uso de pontos de medições ajuda no alcance de múltiplos domínios com as medições, pois os pontos podem estar situados em domínios diferentes. No entanto, as questões que envolvem medições em múltiplos domínios vão além, pois, cada domínio possui suas peculiaridades e restrições, modificando assim o cenário em que a medição está inserida. Logo, existe uma carência por infra-estruturas de monitoramento padronizadas que sejam flexíveis e adaptáveis aos: domínios administrativos diferentes, às aplicações dos usuários finais, bem como às ferramentas que fazem uso dos dados de medições, tais como ferramentas de execução de testes, gerenciamento, visualização, etc.

O interesse em alcançar múltiplos domínios com as medições e utilizar componentes distribuídos para obter informações ao longo de um caminho fim-a-fim, provocou uma forte tendência para que as infra-estruturas de monitoramento adotassem o conceito de uma Arquitetura Orientada a Serviços (do inglês *Service Oriented Architecture - SOA*) (PAPAZOGLU, 2003), onde as funcionalidades são fornecidas através de serviços bem definidos. O uso da arquitetura SOA possibilita a criação de componentes (ou módulos) que disponibilizam suas funcionalidades através de serviços. Esses componentes podem ser distribuídos ao longo das redes, viabilizando assim a coleta de informações em pontos específicos de um caminho fim-a-fim.

O uso de uma arquitetura baseada em serviços facilita o alcance de múltiplos domínios, pois componentes de domínios diferentes podem simplesmente interagir através de serviços padronizados, independentemente da tecnologia utilizada. Existe uma forte tendência para a utilização de Serviços Web como a tecnologia para a implantação desses serviços nas infra-estruturas de monitoramento. Isso com o intuito de reduzir os problemas de interoperabilidade das aplicações, fazendo com que cada componente exponha as suas funcionalidades em forma de Serviços Web bem definidos, e conseqüentemente utilizando a Web como o principal meio de comunicação.

Os Serviços Web geralmente são descritos por meio de informações técnicas, em documentos escritos em XML (*eXtensible Markup Language*⁵) que usam uma linguagem específica chamada de WSDL (*Web Services Description Language*) (ROY, 2001). Essas informações que são descritas através da linguagem WSDL possibilitam adquirir o conhecimento necessário para acessar o serviço, pois a descrição possui as informações de acesso ao Serviço Web tais como: endereço de conexão, parâmetros de entrada e saída, e etc. O uso do XML para descrever as informações do serviço e realizar a comunicação entre os Serviços Web e seus usuários, possibilita a padronização do acesso e a independência de tecnologias semelhantes entre a comunicação dos envolvidos.

Com os Serviços Web sendo uma promessa para integração entre as ferramentas de medição e na interoperabilidade entre componentes das infra-estruturas de múltiplos domínios, surgiram diversos trabalhos na área. Como exemplo é possível citar Sampaio e Suruagy (SAMPAIO; SURUAGY, 2004) que propuseram um modelo de medições por fluxo de tráfego orientado a serviços em que foram definidos e implementados serviços de manipulação de medidas relacionadas aos fluxos de aplicações em redes IP. E ainda, Williamson (WILLIAMSON, 2001) que publicou um estudo relatando a importância da medição e relatando algumas questões importantes no cenário de medições, como o alcance em múltiplos domínios e a necessidade de padrões.

Assim a tendência das infra-estruturas em seguir as definições para uma Arquitetura Orientada a Serviços pode ser comprovada pelo fato de que a Internet2 e a Géant2 em 2005, à procura de soluções inovadoras, iniciaram um trabalho conjunto no sentido de definir um ambiente comum de medição que pudesse ser implantado nas redes nacionais de ensino e pesquisa (*National Research and Education Networks - NRENs*) associadas. Dentro deste esforço, foi elaborado um documento, chamado de *General Framework Design – GFD* (BOOTE, 2005), que detalha uma arquitetura orientada a serviços (SOA) para uma infra-estrutura de medições de desempenho em redes IP. Atualmente, está sendo desenvolvido um protótipo batizado de *PERformance Service Oriented Network monitoring ARchitecture* (perfSONAR) (HANEMANN, 2005) com a finalidade de testar e validar concretamente, os serviços definidos no GFD e que servirá de base para a infra-estrutura a ser implantada nas NRENs.

⁵ Linguagem de marcação que descreve objetos chamados XML.

No cenário nacional, o GT-Medições também vislumbrou a necessidade de transformar sua infra-estrutura de monitoramento em orientada a serviços e multidomínio, pois a infra-estrutura piPEs-BR já era formada por vários componentes desenvolvidos e utilizados separadamente que se comunicavam entre si, necessitando assim de uma padronização na comunicação e a transformação das funcionalidades de cada componente em serviços para facilitar a comunicação e o uso dessas funcionalidades. Outro fator que levou ao GT-Medições também seguir essa tendência é o fato de que, num futuro próximo, as infra-estruturas de domínios diferentes pudessem se comunicar e compartilhar suas funcionalidades. Logo, o GT-Medições decidiu adaptar sua infra-estrutura, o piPEs-BR, para seguir as definições contidas no documento GFD criando assim uma nova infra-estrutura que passou a ser chamada de piPES-BR/GFD.

Ao planejar o desenvolvimento do piPEs-BR/GFD, o GT-Medições expôs para os grupos de pesquisadores do perfSONAR as suas intenções. Essa iniciativa do GT-Medições foi bem recebida, e logo o grupo foi convidado para participar do desenvolvimento do perfSONAR⁶. Como o GT-Medições teve a preocupação em não ser um esforço concorrente, definiu e desenvolveu componentes que não eram previstos no perfSONAR, agregando assim mais funcionalidades às duas infra-estruturas.

A arquitetura GFD provê vários módulos que formam a infra-estrutura de monitoramento e disponibilizam suas funcionalidades através de serviços bem definidos. Dentre esses componentes existe o Serviço de Publicação e Descoberta que será utilizado como repositório de informações sobre todos os Serviços Web da infra-estrutura. O uso de Serviços Web basicamente envolve um provedor de serviços, que disponibiliza suas funcionalidades em forma de serviços, e pelo menos um consumidor dos serviços prestados, que através das descrições disponibilizadas pelo provedor, obtém as informações técnicas de como acessá-los. No entanto, a fim de se obter um maior dinamismo nestas soluções, é necessário desenvolver os três componentes da arquitetura orientada a serviços (SOA), ou seja, não somente o consumidor e provedor de serviços como também um Serviço de Publicação e Descoberta (ou *Service Broker*).

⁶ Desenvolvimento dos serviços de monitoramento especificados no documento GFD.

O Serviço de Publicação e Descoberta possibilita aos usuários de Serviços Web uma busca por um ou mais serviços desejados de acordo com suas necessidades. Essa busca envolve por parte do Serviço de Publicação e Descoberta, o fornecimento das informações técnicas necessárias para a identificação e acesso ao Serviço Web desejado. Além da localização (descoberta) o Serviço de Publicação e Descoberta possibilita que os Serviços Web se registrem (publiquem), possibilitando assim a descoberta. Geralmente o Serviço de Publicação e Descoberta é implantado através do *Universal Description Discovery and Integration* (UDDI) (CURBERA, 2002) que faz parte de um conjunto de padrões para Serviços Web definidos por organizações internacionais.

UDDI é um padrão criado pela *Organization for the Advancement of Structured Information Standards* (OASIS⁷), que utiliza outros padrões como XML e HTTP suportados pelo *World Wide Web Consortium's* (W3C⁸) e pelo *Internet Engineering Task Force* (IETF⁹). O UDDI é formado por normas que definem um Serviço de Publicação e Descoberta de Serviços Web. Esse padrão também fornece definições para *Application Programming Interfaces* (APIs) que podem ser utilizadas por diversos sistemas para interagir com o serviço e acessar as funcionalidades do registro UDDI. De forma resumida, UDDI é um conjunto de tecnologias e funcionalidades para armazenar e prover informações (principalmente de acesso) sobre Serviços Web, seguindo uma especificação em comum. Na grande maioria dos estudos disponíveis, ao se falar em publicação e descoberta usa-se o UDDI como padrão. Assim, o fato da tecnologia ser bastante difundida, possuir uma grande aceitação, existir trabalhos acadêmicos utilizando a mesma e prometendo realizar a publicação e descoberta de qualquer tipo de Serviço Web, foram alguns dos fatores que contribuíram para a escolha dessa tecnologia para ser usada neste trabalho.

Mas só esses fatores não justificam o uso da publicação e descoberta de Serviços Web nas infra-estruturas de monitoramento, tampouco o uso do UDDI como padrão para esse tipo de serviço. Portanto, é preciso pesquisar e verificar os benefícios da publicação e descoberta e, conseqüentemente, do UDDI no escopo das infra-estruturas de medição. Logo, na próxima seção serão apresentadas algumas justificativas para este fim.

⁷ Organização internacional sem fins lucrativos de padronização de serviços.

⁸ Consórcio que desenvolve padrões usados na criação e interpretação de conteúdos para a Web.

⁹ Comunidade internacional de projetistas, fabricantes, fornecedores, operadores e pesquisadores de redes de computadores, interessados com a evolução da arquitetura da Internet e o seu funcionamento perfeito.

1.2 JUSTIFICATIVAS

A tendência das infra-estruturas de monitoramento é de operar em redes de padrões abertos o que implica na relação entre domínios diferentes (multidomínio). Devido a isto, essas infra-estruturas estão seguindo as definições de uma arquitetura orientada a serviços (SOA) e adotando o uso de Serviços Web para abstrair detalhes técnicos, que dificultam ou impedem a comunicação entre ferramentas, técnicas de medições e clientes de visualização. Como a arquitetura SOA possui um Serviço de Publicação e Descoberta (*Service Broker*) como um de seus componentes, fica evidente que para um bom funcionamento das infra-estruturas de monitoramento é necessário propor esse componente para realizar a localização dos Serviços Web. Um serviço desse tipo torna “visível” aos usuários, os Serviços Web das infra-estruturas de monitoramento, possibilitando assim dinamizar o acesso aos Serviços Web e localizar um serviço desejado.

Assim, a experiência do GT-Medições com o uso de seu primeiro protótipo de infra-estrutura de medições o piPEs-BR, o qual já disponibilizava alguns Serviços Web, mostrou que para facilitar o acesso dos clientes aos Serviços Web seria necessário criar um mecanismo que dinamizasse esse acesso. Pois o uso de Serviços Web sem o auxílio da Publicação e Descoberta implica na definição estática das formas de acesso aos Serviços Web, ou seja, em arquivos de configurações ou no código fonte da aplicação usuária (SAMPAIO e outros, 2006). Assim, ao usar as informações de acesso aos Serviços Web de forma estática, pode ocorrer que as informações que estão sendo utilizadas não sejam compatíveis, pois esses tipos de informação mudam constantemente. Por exemplo, ao disponibilizar uma nova versão de um Serviço Web, as informações de acesso como: parâmetros, nome do serviço ou endereço de acesso geralmente são modificadas. Logo, essa abordagem implica em que toda vez que as informações de acesso forem alteradas, o cliente fica impossibilitado de acessar o Serviço Web, sendo que para voltar a acessar o Serviço Web, será preciso descobrir as novas informações e alterar informações desatualizadas. Por outro lado, a partir do uso da publicação e descoberta as informações são buscadas em um componente independente, possibilitando assim que essas informações possam ser facilmente atualizadas.

Outra justificativa para a realização desse trabalho é que geralmente antes de usar um Serviço Web, os usuários normalmente desejam saber algumas informações tais como: se esse serviço

existe, se o serviço atende às suas exigências, onde esse serviço está localizado e os requisitos para usar esse serviço. Ainda assim é possível que exista mais de um Serviço Web que atenda as exigências do usuário ou atenda parcialmente essas exigências. Para fornecer essas informações, realizar a localização, ajudar na escolha e uso do Serviço Web é necessário usar uma tecnologia que considere todas essas informações e que de certa forma classifique esses Serviços Web. Neste contexto, o serviço de Publicação e Descoberta tem a finalidade de fornecer as informações desejadas pelos usuários antes que este escolha e use determinado Serviço Web.

O padrão UDDI possibilita a publicação e descoberta de qualquer tipo de Serviço Web, abrangendo questões tais como a troca de mensagens, definição de funcionalidades, armazenamento de informações e a interação com outros componentes. Ou seja, ele disponibiliza um Serviço de publicação e Descoberta, confiável, padronizado, estável que também pode ser utilizado para fornecer informações dos Serviços Web de monitoramento da infra-estrutura piPEs-BR/GFD (MONTEIRO e outros, 2007).

Na próxima seção será abordado o problema a ser atacado neste trabalho, bem como o principal objetivo deste estudo.

1.3 PROBLEMA E OBJETIVO

O principal problema atacado neste trabalho é como utilizar o conceito de Publicação e Descoberta de Serviços da arquitetura SOA e o padrão UDDI no escopo das infra-estruturas de monitoramento de redes de computadores. Atualmente o uso das normas UDDI se restringe ao cenário do comércio eletrônico, ocasionando uma falta de material na literatura que envolva a publicação e descoberta em outros escopos. Na literatura, ao se falar em publicação e descoberta, geralmente o UDDI é citado como o padrão para Serviços Web que provê essas funcionalidades. Outro ponto importante é que, para usar a publicação e descoberta nas infra-estruturas de monitoramento, é necessário considerar as características dos Serviços Web utilizados e os requisitos dos clientes dos mesmos.

Portanto, o objetivo principal deste trabalho é propor e desenvolver um Serviço de Publicação e Descoberta para a infra-estrutura de monitoramento piPEs-BR/GFD. Para este

fim será necessário: desenvolver o serviço usando a tecnologia padrão de Serviços Web UDDI, levar em consideração as necessidades da infra-estrutura piPEs-BR/GFD, especificar um modelo de dados sobre o padrão UDDI para Serviços Web de medições, desenvolver uma biblioteca de acesso ao serviço que facilite o acesso ao mesmo, possibilitar a inter-operação com outras infra-estruturas e produzir um estudo comparativo entre as implementações disponíveis e a realizada neste trabalho.

Para conseguir esse objetivo foi elaborada uma estrutura de tópicos neste documento que tem o intuito de facilitar o entendimento deste trabalho. Logo, essa estrutura pretende abordar a Arquitetura Orientada a Serviços, a qual o Serviço de Publicação e Descoberta está inserido, mostrar as infra-estruturas de monitoramento levando em consideração a Publicação e Descoberta de Serviço Web, descrever as tecnologias disponíveis, apresentar a proposta de um Serviço de Publicação e Descoberta para Serviço Web de medições, relatar o desenvolvimento do serviço proposto e observar os resultados obtidos com a pesquisa. Portanto, na próxima seção será apresentada essa estrutura, bem como o que se pretende com cada capítulo.

1.4 APRESENTAÇÃO

Este trabalho está organizado em seis capítulos, este é o primeiro deles e contém o conteúdo introdutório para ambientar e informar ao leitor qual é a proposta do trabalho.

No segundo capítulo serão abordados os aspectos que envolvem a Arquitetura Orientada a Serviços (SOA), pois, a Publicação e Descoberta de Serviços Web faz parte do contexto desse tipo de arquitetura. Assim, o segundo capítulo explicará o que é SOA e abordará algumas tecnologias e padrões que implementam e dão suporte a este tipo de arquitetura. É importante frisar que é neste capítulo que será apresentado um dos assuntos principais desse trabalho que é o padrão UDDI.

Segue-se o terceiro capítulo, onde serão abordadas as características que envolvem as infra-estruturas de monitoramento, enfatizando como as mais importantes infra-estruturas utilizam, no seu escopo, um serviço de Publicação e Descoberta de Serviços Web.

O quarto capítulo apresenta a proposta para a solução do problema abordado neste trabalho que é a definição de um Serviço de Publicação e Descoberta de Serviços Web de monitoramento. Nesse capítulo serão apresentadas dentre outras, as características, a estrutura, e as funcionalidades do serviço.

Já no capítulo cinco será apresentado o protótipo desenvolvido do serviço proposto neste trabalho, que tem como finalidade validar o serviço proposto e possibilitar a publicação e descoberta dos Serviços Web da infra-estrutura de monitoramento piPEs-BR/GFD.

Finalmente, o capítulo seis aborda as questões conclusivas tiradas da pesquisa reportada nesta dissertação, bem como são identificados e propostos trabalhos futuros utilizando o serviço proposto.

2 ARQUITETURA ORIENTADA A SERVIÇOS – SOA

Este capítulo aborda a Arquitetura Orientada a Serviços (*Service Oriented Architecture - SOA*), onde a Publicação e Descoberta está inserida. O intuito deste capítulo é abordar as tecnologias e padrões que estão envolvidos com SOA.

2.1 INTRODUÇÃO

O uso das definições contidas na arquitetura SOA para transformar funcionalidades em serviços, atualmente é uma realidade. Há algum tempo atrás se iniciavam na literatura algumas vertentes tentando dar mais importância ao uso das funcionalidades dos softwares (TUNER e outros, 2002) (BENNETT e outros, 2000) e, conseqüentemente, assumindo que os mesmos são provedores de serviços. Hoje em dia é possível ver novas vertentes que utilizam as definições de SOA para criar novos conceitos como a *Consumer-Centric Service-Oriented Architecture - CCSOA* (TSAI e outros, 2006) e é possível ver também que a arquitetura SOA está sendo aplicada até em dispositivos móveis (*Mobile SOA: Service Orientation on Lightweight Mobile Devices*) (TERGUJEFF e outros, 2007).

A Arquitetura Orientada a Serviços é um tipo de arquitetura de sistemas distribuídos que tem como base tornar as aplicações provedoras de serviços (PAPAZOGLU, 2003). Basicamente, a arquitetura possui três elementos: um cliente de serviços, um provedor de serviços e um repositório de informações sobre os mesmos. A funcionalidade de cada elemento está descrita a seguir. Na Figura 2.1 é possível ver a interação entre esses elementos, onde existe uma seqüência lógica de acontecimentos.

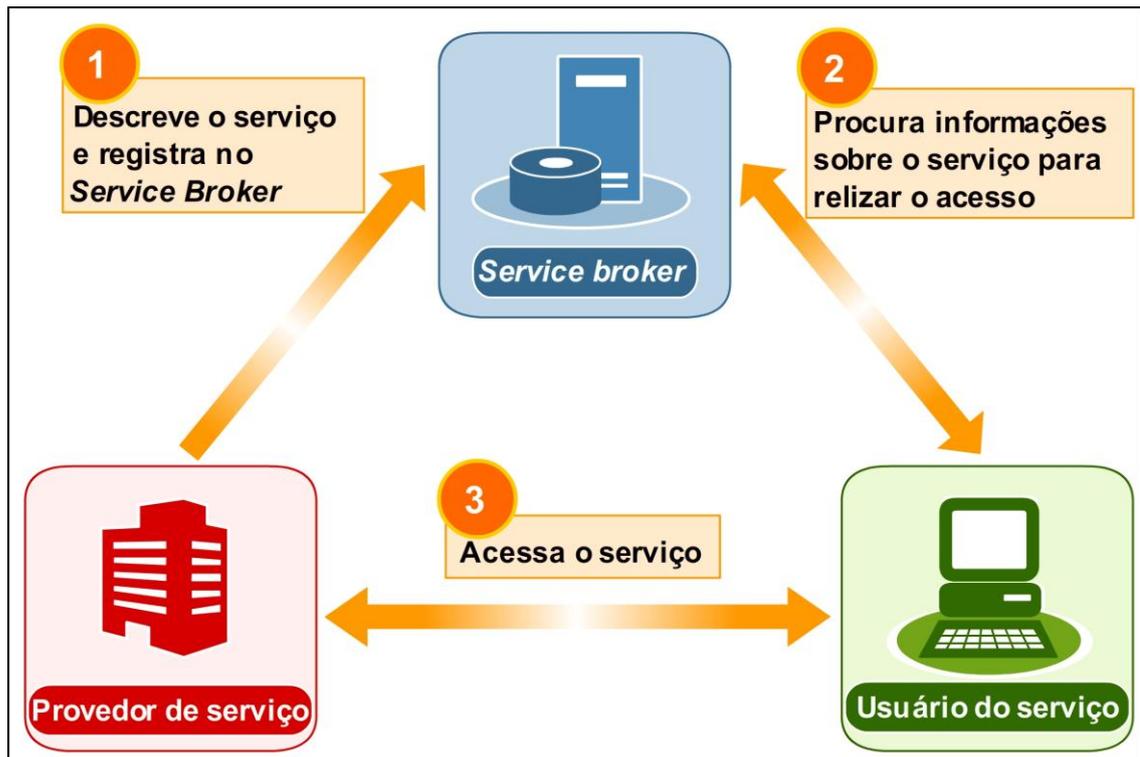


Figura 2.1 - Arquitetura Orientada a Serviços
 Nota: Elaboração do autor (2007).

As funcionalidades dos três principais elementos da SOA são:

- cliente** - São os usuários do serviço disponibilizado. Podem ser quaisquer aplicações, até mesmo uma aplicação que disponibiliza outros serviços. Envia requisições ao serviço iniciando o ciclo de comunicação, esperando como resposta o resultado do que foi solicitado;
- provedor de serviços** - É o fornecedor de serviços que pode conter um ou mais serviços que são disponibilizados para os clientes. Por sua vez recebe as solicitações, interpreta e responde ao solicitante;
- publicação e descoberta (*Service Broker*)** - O *Service Broker* armazena informações sobre os serviços, possibilitando assim para os clientes conhecer o serviço antes de utilizá-lo. O *Service Broker* tem um papel fundamental no funcionamento da arquitetura, pois facilita a descoberta dinâmica dos serviços por parte dos clientes.

Um dos requisitos da SOA é que os elementos que a compõem devem inter-operar, mesmo que sejam sistemas ou linguagens de programação diferentes (PAPAZOGLU, 2003). Isso

deve ser feito via comunicação com protocolos padrão, introduzindo assim a tecnologia de Serviços Web, comumente usada para fazer essa integração, pois a mesma utiliza os protocolos da Internet que são independentes de plataforma e linguagens de programação. Já para o *Service Broker* existe um padrão Web chamado UDDI e que é bem utilizado. Para finalizar, abordando o lado do cliente, a tecnologia bastante utilizada é o *Apache Axis*¹⁰ que disponibiliza um conjunto de soluções para a construção de Serviços Web e clientes de acesso.

2.2 SOA: PUBLICANDO E DESCOBRINDO SERVIÇOS

Um dos componentes da SOA que vem sendo bastante desenvolvido e utilizado é o *Service Broker*, chamado neste trabalho de Serviço de Publicação e Descoberta. Esse aumento no foco do Serviço de Publicação e Descoberta se deve à aceitação da abordagem de transformar as funcionalidades dos softwares em serviços, pois, assim que o número de serviços aumentou a necessidade, por parte dos consumidores, de descobrir o serviço ficou mais evidente. Outro fator que contribuiu bastante para o aumento no interesse para a publicação de serviços é a necessidade de dinamizar o acesso aos serviços.

Desenvolver uma aplicação que disponibiliza suas funcionalidades através de serviços sem o uso da publicação e descoberta é possível e até aceito em alguns casos como: uma aplicação que possui poucos serviços, não existir freqüentes mudanças no acesso aos serviços e se todos os serviços forem de um mesmo proprietário, pois nesses três casos os serviços são bem conhecidos e existe pouca possibilidade de mudança no cenário. Como essa combinação de fatores é pouco provável, um Serviço de Publicação e Descoberta torna-se indispensável em longo prazo. Assim, a tendência de disponibilizar mais Serviços Web é crescente, bem como a mudança nas formas de acesso aos serviços e o uso de serviços de outras estruturas é inevitável (ex. mudança de URL (*Uniform Resource Locator*) de acesso que é freqüente e o alcance de domínios diferentes).

¹⁰ Implementação do protocolo *Simple Object Access Protocol* (SOAP) descrito pelo w3c.

Atualmente esse cenário de aplicações com muitos serviços disponibilizados para uma grande quantidade de usuários espalhados pelas redes é tão presente, que as principais pesquisas sobre a publicação e descoberta estão envolvendo a utilização de mais de um Serviço de Publicação e Descoberta interagindo entre si (ZURAWSKI e outros, 2007). A utilização de vários Serviços de Publicação e Descoberta possui o intuito de abranger mais usuários (ou domínios), aumentar a disponibilidade e melhorar o desempenho. Um exemplo de serviço que possibilita o uso de múltiplos Serviços de Publicação e Descoberta é o definido no padrão UDDI em sua versão três.

2.3 SERVIÇOS WEB

Para melhor definir o que são Serviços Web, primeiro se faz necessário definir serviços em computação. Serviço é um programa ou componente de software em um dado ambiente que provê e gerencia acesso a recursos que são essenciais para outras entidades. É importante ressaltar que a noção de recurso nessa definição é bastante genérica. O recurso pode ser um hardware ou um software. Já o Serviço Web faz parte de uma classe especial de serviços onde a principal característica é a interoperabilidade (APTE; MEHTA, 2002).

Os Serviços Web funcionam em redes de computadores e são suportados por um *framework* específico. O *framework* provê meios de descrever e descobrir o serviço, interagir com o serviço e numa forma mais sofisticada integrar Serviços Web com outros Serviços Web.

Uma das características principais dos Serviços Web é que as tecnologias relacionadas têm como base o uso da linguagem XML, que é utilizada para a comunicação entre as aplicações, descrição e definição de tipos complexos de dados, descrição dos serviços e publicação e descoberta dos mesmos. Essas aplicabilidades do XML geralmente são implantadas pelas seguintes tecnologias: a linguagem *Web Services Description Language* (WSDL); o protocolo *Simple Object Access Protocol* (SOAP) (BOX e outros, 2000); e o padrão UDDI. Essas tecnologias fazem parte do *framework* de Serviços Web que por sua vez visa possibilitar o acesso aos mesmos.

Contudo, o acesso a um Serviço Web pode ser feito de duas maneiras. O primeiro tipo de acesso é via passagem de parâmetros, ou seja, o cliente realiza uma requisição a um determinado serviço passando os parâmetros de entrada e esperando assim os parâmetros de saída. O segundo tipo de acesso é feito via troca de documentos padronizados. Nesse caso o cliente irá fornecer ao serviço um documento padronizado que geralmente contém a funcionalidade desejada e os parâmetros necessários. Nesse tipo de abordagem é possível que uma aplicação disponibilize apenas um serviço padrão, que recebe os documentos padrões, verifica a solicitação e realiza a operação. Um exemplo claro dos dois tipos pode ser descrito assim:

- a) **passagem de parâmetros** - Um usuário deseja obter a média de suas notas escolares. Esse usuário precisa utilizar um serviço que realize a soma de suas notas, esse serviço recebe como parâmetros as notas. Depois o usuário terá que usar um serviço que realiza a divisão entre o total de pontos com a quantidade, logo esse serviço recebe como parâmetros o somatório das notas e a quantidade de notas;
- b) **troca de documentos** - O mesmo usuário que deseja obter a média de suas notas escolares pode utilizar um serviço chamado média, que recebe um documento que contém no seu interior a solicitação para a soma de todas as notas (que também estão no documento) e que depois realize a divisão pela quantidade de notas. É importante frisar que nesse caso somente um serviço é disponibilizado.

Outro componente importante do Serviço Web é o ponto de acesso (do inglês *Access Point*), como é comumente chamado. O ponto de acesso é um URI (*Uniform Resource Identifier*) que nomeia o endereço onde deve ser realizada a conexão com o serviço.

Existe um mecanismo no cenário de Serviços Web que é de extrema importância para a Publicação e Descoberta. Esse mecanismo é o WSDL que descreve as características técnicas do Serviço Web. Essa descrição pode ser usada para conhecer previamente o serviço, o que é semelhante a uma das funcionalidades do Serviço de Publicação e Descoberta que tem a finalidade de fornecer informações sobre os Serviços Web.

2.3.1 Web Service Definition Language – WSDL

O WSDL é uma linguagem baseada em XML que descreve em documentos, os Serviços Web disponibilizados por uma aplicação. Ou seja, é recomendado que ao disponibilizar um Serviço Web, seja disponibilizado também um documento WSDL que descreve como os usuários podem realizar requisições ao serviço. Essas descrições fornecem a forma de interação e os requisitos necessários para a comunicação, o que as deixam transparentes no que diz respeito à plataforma e linguagem de programação para o cliente.

Outro uso bastante conhecido do documento WSDL é na criação dinâmica de clientes de acesso ao serviço. Dependendo da estrutura e forma de acesso compatível com o serviço é possível, através de um documento WSDL, construir automaticamente uma aplicação que possa usar o Serviço Web normalmente. Isso é possível devido ao fato de que no WSDL são descritas as informações de acesso ao serviço como: o URI do ponto de acesso, os parâmetros de entrada e saída e etc.

O WSDL é importante para publicação e descoberta, pois as informações contidas no documento WSDL fazem parte das informações fornecidas por um Serviço de Publicação e Descoberta aos usuários. Essas informações são consideradas básicas para o acesso ao serviço. Alguns padrões de Serviço de Publicação e Descoberta como o UDDI possuem elementos que fazem analogia ao documento WSDL e até mesmo usam mecanismos para armazenar as informações de um documento WSDL (APTE; MEHTA, 2002). Contudo, o UDDI e o WSDL têm funções semelhantes quando o assunto é fornecer informações de acesso ao Serviço Web. No entanto, essa semelhança não passa deste ponto, pois o WSDL não possibilita a localização dinâmica do serviço.

2.4 PADRÕES PARA PUBLICAÇÃO E DESCOBERTA DE SERVIÇOS

A Publicação e Descoberta de Serviços é conhecida como *Service Broker* na arquitetura orientada a serviços e, às vezes, é comumente chamada por UDDI, pois é possível se confundir entre UDDI, que é um padrão que utiliza normas para definir um Serviço de

Publicação e Descoberta, com o próprio *Service Broker*, que é um conceito. Essa confusão se dá pelo fato da especificação UDDI ser considerada a tecnologia padrão Web na Publicação e Descoberta de Serviços Web, fazendo com que ao se falar na Publicação e Descoberta de Serviços, automaticamente se fale em UDDI.

Contudo, os Serviços de Publicação e Descoberta gerenciam uma base de dados com as informações de acesso a Serviços Web, independentemente se esses serviços fazem parte da mesma aplicação ou de várias aplicações em contextos diferentes. Essas informações são usadas na identificação e acesso aos Serviços Web. Portanto, um Serviço de Publicação e Descoberta armazena informações do tipo: o URI do ponto de acesso ao serviço web, os parâmetros de entrada e os parâmetros de saída. Outras informações podem também ser adquiridas no serviço de publicação e descoberta, isso vai depender de quem especifica o serviço e com que finalidade.

Na literatura existem alguns padrões que são usados para a publicação e descoberta de serviços. Alguns desses padrões têm um foco específico, por exemplo, o ebXML (HOFREITER e outros, 2002) que é usado especificamente para a publicação e descoberta de serviços de comércio eletrônico (do inglês *eBusiness*). Existem também outras definições que foram elaboradas para realizar a publicação e descoberta de forma genérica, ou seja, para qualquer tipo de Serviço Web. A seguir serão descritas duas especificações existentes.

2.4.1 ebXML - Comércio Eletrônico usando XML

Geralmente no cenário do comércio eletrônico, as aplicações disponibilizam suas funcionalidades através de serviços possibilitando a realização de transações comerciais. Essa abordagem facilita o processo, pois nesse tipo de transação existe uma interação entre diversos elementos, por exemplo, em uma simples venda por cartão, onde é preciso a autorização da operadora, verificação da disponibilidade do produto dentre outros. Assim, dentro do escopo de comércio eletrônico há um grande interesse em disponibilizar serviços para que o cliente e outras empresas possam realizar transações através de serviços padronizados e bem definidos.

O *Electronic Business using eXtensible Markup Language* (ebXML) é um padrão para Serviços Web definido pela OASIS, que fornece um conjunto de especificações que permitem às empresas administrar seus negócios na Internet. Dentre essas especificações existe uma que define como publicar e descobrir Serviços Web automaticamente para serem utilizados nos processos de negócios. Outra funcionalidade que faz parte deste conjunto de especificações é a padronização de mensagens e tipos de dados através de um *Schema XML* próprio (HOFREITER e outros, 2002). Isso auxilia na troca mensagens entre o registro ebXML e os usuário do mesmo que podem ser os Serviços Web de comércio eletrônico ou os usuários comuns. Uma característica que também pode ser encontrada nas especificações do ebXML é que todo o processo envolvendo o registro ebXML deve ser realizado de forma segura e confiável. Atualmente a especificação do ebXML se encontra na versão três.

O padrão ebXML também define um protocolo para a troca de mensagens XML baseadas em serviços de comércio eletrônico. Essas mensagens contêm informações empresariais necessárias para acessar um serviço de comércio eletrônico disponibilizado por uma empresa. Basicamente o protocolo definido no ebXML é um conjunto de camadas estendidas do SOAP. No entanto, essas camadas tentam prover duas funcionalidades não suportadas pelo SOAP que é a confiabilidade e a segurança (HOFREITER e outros, 2002).

Na Figura 2.2 é possível ver a utilização do ebXML, onde estão enumerados os passos de um exemplo clássico. Na figura temos a Empresa A que deseja desenvolver um serviço e ao mesmo tempo utilizar um serviço de ebXML existente. No passo 1 a Empresa A para desenvolver um serviço precisa saber os detalhes necessários para desenvolver um serviço de acordo com as especificações de negócio do registro ebXML. Logo, a Empresa requisita esses detalhes. Com os detalhes em mãos, a Empresa A desenvolve o serviço no passo 2 e consecutivamente registra as informações sobre esse serviço no passo 3. Neste ponto, o serviço disponibilizado pela Empresa A se torna visível para todos os usuários do mesmo registro ebXML. Continuando no exemplo da Figura 2.2, existe agora uma Empresa B que, por exemplo, ao acessar o registro ebXML verificou que existe um serviço fornecido pela Empresa A que satisfaz suas necessidades. Logo, no passo 4, a Empresa B consulta sobre a Empresa A e recupera as informações sobre o serviço oferecido pela Empresa A. Depois de manipular e verificar as informações sobre o serviço oferecido, a Empresa B concorda com o esquema de negócios da Empresa A através de identificações e validações no passo 5. E para

finalizar, de posse das informações adquiridas no registro ebXML, a Empresa B pode então usar o serviço da Empresa A, normalmente, no passo 6.

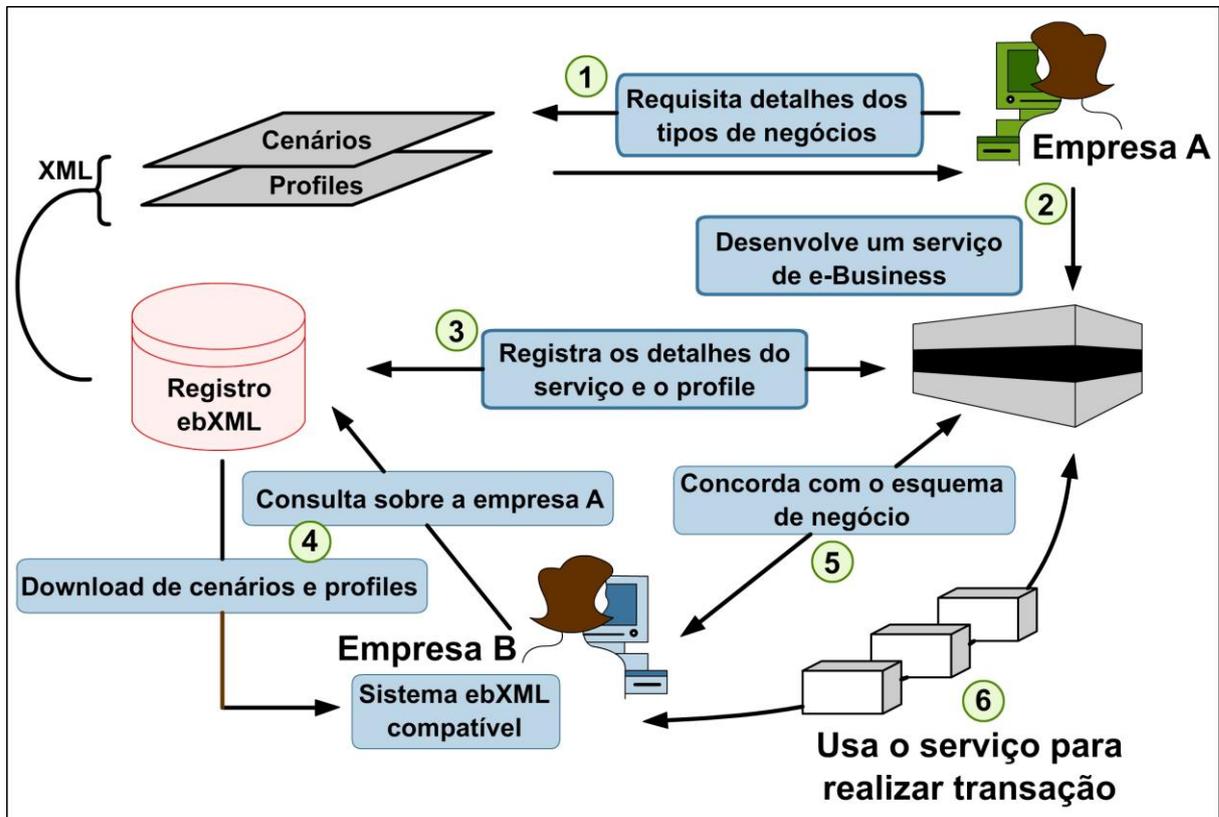


Figura 2.2 - Funcionamento do ebXML

Nota: Elaboração do autor (2007).

2.4.2 Universal Description Discovery and Integration – UDDI

O UDDI é um componente chave para um bom uso de Serviços Web, pois cria uma interface padrão que possibilita que as aplicações localizem rapidamente, facilmente e dinamicamente os Serviços Web na Internet (APTE; MEHTA, 2002). O UDDI pode ser utilizado em diferentes contextos, pois as suas definições procuram não especificar o tipo de serviço a ser publicado.

O UDDI é uma iniciativa que foi implantada como projeto de pesquisa por várias companhias interessadas em discutir questões relacionadas à integração entre sistemas de computação. Logo, essa iniciativa definiu um padrão Web que provê uma especificação para um Serviço de Publicação e Descoberta de Serviços Web. Esse padrão também definiu APIs (*Application Programming Interfaces*) que são usadas como interfaces de acesso ao UDDI e podem ser

utilizadas por diversos sistemas para interagir com o UDDI e acessar suas funcionalidades. Essa interação ocorre da seguinte forma: os Serviços Web acessam o UDDI para registrar suas informações de acesso e identificação; já os usuários desses serviços acessam o UDDI para localizar essas informações.

Numa visão de alto nível, as informações sobre os Serviços Web que são registradas no UDDI são classificadas em três categorias principais:

- a) **páginas brancas** (*White Pages*) - São as informações gerais sobre o proprietário dos serviços, como por exemplo, nome do proprietário, descrição desse proprietário, informações de contato, endereço e outros. É importante lembrar que o proprietário pode ser uma empresa, uma pessoa, uma instituição de ensino e etc;
- b) **páginas amarelas** (*Yellow Pages*) - Aqui são armazenadas as informações genéricas dos serviços, por exemplo, nome, descrição, tipo e etc. Aqui é possível usar a classificação do serviço através de taxonomias;
- c) **páginas verdes** (*Green Pages*) - Esta categoria armazena as informações técnicas sobre o serviço, como por exemplo, a URI de acesso ao serviço, o tipo de protocolo, os parâmetros necessários e etc.

A primeira versão do UDDI, a 1.0, foi desenvolvida pela Microsoft, IBM e Ariba, e foi anunciada em setembro de 2000. Desde o anúncio inicial a tecnologia cresceu e, em maio de 2001, foi lançada a primeira implementação, o UDDI Registry. Em junho de 2001 foi anunciada a versão 2.0 das normas UDDI, incluindo novas características e funcionalidades. Em agosto de 2003 foi lançada a versão 3 das especificações UDDI, a qual se encontra em utilização atualmente. Algumas funcionalidades da versão 2 e 3 serão detalhadas no decorrer deste trabalho.

A especificação do padrão UDDI consiste em um *Schema* XML que define uma estrutura de dados que armazena as informações sobre os Serviços Web. Sendo que o UDDI também descreve uma especificação para o desenvolvimento de uma API de acesso. Juntos, o *Schema* e a API formam um modelo básico para a publicação de informação sobre um grande número de Serviços Web.

Contudo, as informações contidas no UDDI que estão distribuídas nas páginas branca, verde e amarela, são implementadas, na prática, em formato XML através de quatros componentes básicos: o *businessEntity*, *businessService*, *bindingTemplate* e o *tModel*. Esses componentes serão descritos na próxima subseção.

2.4.2.1 Componentes do UDDI

O UDDI possui um conjunto de componentes onde as informações sobre os Serviços Web são distribuídas. Alguns autores chamam esse conjunto de componentes de estrutura de dados. Realmente esse conjunto de componentes remete a uma estrutura de dados de uma base de dados relacional, mas geralmente esses componentes são demonstrados visualmente sem levar em consideração as características de uma estrutura de dados como os relacionamentos, índices, atributos e outros. Na Figura 2.3 é possível ver os componentes principais e seus relacionamentos.

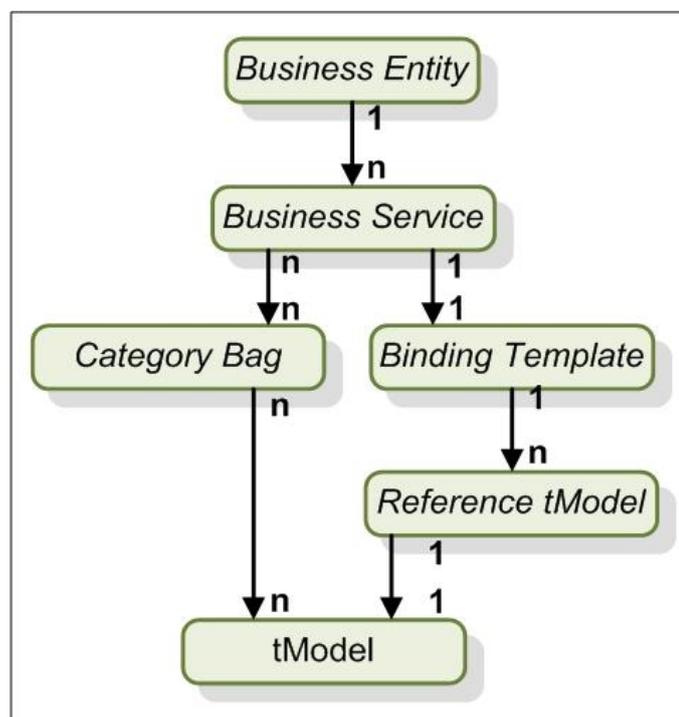


Figura 2.3 - Componentes do UDDI
Nota: Elaboração do autor (2007).

- a) **business entity** - Componente que armazena as informações sobre o proprietário de um conjunto de serviços. Um registro UDDI pode conter mais de um *Business Entity*. Logo, ao consultar sobre um determinado serviço web é necessário informar também a identificação do *Business Entity*. Por exemplo, se em um registro UDDI existirem dois *Business Entities* (UNIFACS e UFBA), ao consultar um serviço nesse registro é necessário informar na solicitação, que o serviço procurado pertence à entidade UNIFACS ou UFBA. Um *Business Entity* armazena informações como: nome, descrição, contato, endereço e etc.;
- b) **business service** - O Business Service é o serviço propriamente dito. Nele são armazenadas informações genéricas do serviço tais como nome e descrição;
- c) **binding template** - O *Binding Template* armazena informações técnicas do serviço como: ponto de acesso, URI do WSDL, tipo da URL do ponto de acesso (ex. HTTP ou FTP), parâmetros e etc. Essas informações são consideradas as informações de acesso, pois para acessar um serviço é imprescindível saber a URI do ponto de acesso do serviço, os parâmetros de entrada e os parâmetros de saída, caso houver;
- d) **tModel** - Fornecer informações sobre os Serviços Web também é só uma das funcionalidades do UDDI. Outra funcionalidade é a possibilidade de armazenar informações que sejam significativamente específicas e diversificadas, para serem bastante úteis durante a busca. O *tModel* é o componente do UDDI que possui diversas funcionalidades, sendo que a principal é identificar o serviço através de referências e armazenamento de informações específicas, fazendo com que o mesmo seja único ou faça parte de um grupo. O *tModel* é utilizado como um tipo de marcador que identifica os serviços através de uma característica do mesmo. Por exemplo, um serviço de vendas pode ser categorizado por diversas características tais como: venda a varejo ou atacado, venda a vista ou a prazo, venda de roupas ou de comidas e etc. Com essa variedade de características é possível com o *tModel* criar uma estrutura de categorização ou taxonomia onde uma ou mais dessas características identificam o serviço;
- e) **category bag** - Usado para armazenar um conjunto de características que categorizam um serviço. É composto por um par de informações que são o nome da característica e o valor. Como exemplo, é possível citar um serviço de

lavagem de carros, que possui as seguintes características: lava os carros com um jato de água, realiza polimento com cera cristal, limpa o interior do carro com aspirador e etc. Nesse caso esse conjunto de características seria armazenado no *Category Bag* do serviço de lavagens de carro, como pode ser visto na Quadro 2.1.

<i>Category Bag</i>	
Características	Valor
Lavagem	Jato de água
Polimento	Cera cristal
Limpeza de interior	Aspirador

Quadro 2.1 - Exemplo de conjunto de características de um serviço
Nota: elaboração do autor (2007).

Contudo, é possível dizer que esses componentes são divididos em dois grupos: um que armazena as informações genéricas dos serviços e outro que armazena as informações específicas dos Serviços Web. Assim, num registro UDDI podem existir vários serviços que são semelhantes em relação às informações genéricas, sendo que com o auxílio dos componentes *tModel* e *Category Bag* existe a possibilidade de diferenciar ou até relacionar esses serviços. Em nosso exemplo é possível que no registro UDDI existam vários serviços que realizem a lavagem de carros, mas dentre esses serviços também é possível pesquisar, por um que seja identificado pelo fato de possuir em seu *Category Bag* referências a *tModels* e valores que o diferenciam dentre os outros. Como, o fato de realizar polimento com cera cristal, lavagem com jato de água ou utilizar o aspirador.

2.4.2.2 Categorização de serviços

As informações presentes nos componentes *tModel* e *Category Bag* do UDDI dão suporte para o uso das taxonomias. Taxonomia é a ciência que estuda a classificação ou categorização de uma determinada coisa (XIMENES, 1998). No UDDI a taxonomia classifica Serviços Web de acordo com determinadas características específicas do serviço, possibilitando que a pesquisa do mesmo seja baseada em uma determinada categoria (ex. de produto, ou de localização geográfica). Essas taxonomias podem ser normatizadas, que são definidas através de normas internacionais ou privadas que são criadas para suprir uma necessidade própria,

permitindo que sejam desenvolvidas taxonomias para quaisquer tipos de categorização (PINTO e outros, 2003).

Assim, qualquer estrutura no uso dos *tModels* para categorizar um serviço pode ser considerada uma taxonomia. Por exemplo, ao categorizar um serviço pelo seu “tipo” e “subtipo”, é possível considerar essa categorização uma taxonomia de tipo, onde um serviço possui um tipo genérico e um subtipo que especifica o serviço. Portanto, a categorização citada é uma taxonomia privada que foi criada pelo operador do registro UDDI para categorizar um serviço de acordo com o seu tipo. Essa taxonomia é considerada hierárquica, onde o “tipo” é o ancestral do “subtipo”. Assim, um *tModel* considerado “filho” faz referência ao seu “pai”, possibilitando saber que existe uma ou mais características específicas que podem ajudar na escolha do serviço. É possível citar como exemplo, o Serviço de Armazenamento definido no GFD (*Measurement Archive - MA*), que pode ser caracterizado pelo *tModel* “tipo” com valor MA, que por sua vez é referenciado pelo seu “filho” um *tModel* “subtipo”. Como um MA é um serviço de armazenamento, podemos considerar o seu “subtipo” o formato em que os dados são armazenados, por exemplo, RRD ou XML.

Contudo, na especificação do UDDI existem algumas taxonomias normatizadas que são conhecidas no cenário de classificação de indústrias ou produtos. Para demonstrar um tipo de taxonomia normatizada que o padrão UDDI utiliza é possível citar a taxonomia ISO 3166 (também conhecida como “geo” taxonomia) para a categorização geográfica de serviços. Essa taxonomia possibilita associar serviços a localizações cuja granularidade varia entre países e dependentes (que podem ser estados). Por exemplo, o número de identificação do Brasil na taxonomia ISO 3166 é 076. Ao se pretender usar a taxonomia ISO 3166 para os serviços de um registro UDDI é preciso criar um *tModel* para a norma ISO e armazenar o valor correspondente à norma ISO no serviço. Portanto, ao categorizar um serviço situado no Brasil, é necessário armazenar o valor referente ao Brasil na norma ISO 3166 que é o número 076 e fazer com que o serviço aponte para o *tModel* referente à norma ISO. Ao tentar localizar um serviço pela norma ISO 3166 o usuário deverá buscar todos os serviços que apontam para o *tModel* da norma e que possuem o valor desejado.

2.4.2.3 UDDI versão três

Todas as características citadas anteriormente estão inseridas na versão dois do padrão UDDI. Mas, é preciso também mencionar neste trabalho as funcionalidades existentes na versão três do padrão UDDI, pois essa versão será bastante útil para trabalhos futuros que podem ser realizados com os resultados desta dissertação.

A versão três do padrão UDDI possui como principal característica o incentivo ao uso do UDDI bem como ao dos Serviços Web. Pois essa versão traz alguns recursos de segurança e uso de múltiplos registros. Ou seja, serviços de publicação e descoberta interagindo entre si, trocando informações e localizando serviços situados em outros registros.

Em relação à segurança o UDDI versão três introduz assinaturas digitais para fornecer segurança adicional. Cada serviço de publicação e descoberta pode ser assinado digitalmente, aprimorando a integridade e a confiança dos dados de UDDI. Lembrando que já na versão dois o UDDI utiliza um mecanismo de autenticação baseado em usuário e senha.

Já em relação ao uso de vários serviços de publicação e descoberta interagindo entre si, o UDDI versão três implementa funcionalidades que possibilitam a replicação de dados entre as bases. Assim, dentro de um domínio, é possível utilizar vários serviços do UDDI que interagem entre si replicando suas informações. Essa replicação é formada por um anel lógico de serviços de publicação e descoberta, onde qualquer membro desse anel, ao verificar que seus registros foram alterados, replica essa informação ao longo do anel até que todos possuam a mesma informação.

Outro avanço no UDDI versão três é que esse anel citado pode interagir com outros serviços. Por exemplo, um conjunto de serviços UDDI da RNP que formam um anel lógico de replicação é capaz de eleger um líder entre eles e fazer com que esse líder também replique suas informações com outro líder que está implantado na Internet2. Assim, um usuário do serviço de publicação e descoberta da Internet2, poderia descobrir serviços publicados nos serviços de publicação e descoberta instalados na RNP. Sendo que essa funcionalidade também possibilita a restrição na replicação dos dados, ou seja, um UDDI pode somente replicar parte das informações, ou replicar um conjunto reduzido de Serviços Web.

2.5 TECNOLOGIAS PARA DESENVOLVIMENTO E IMPLANTAÇÃO DE SOA

Existem algumas tecnologias proprietárias ou *open source* que dão suporte para desenvolver e implantar aplicações orientadas a serviços. Essas tecnologias são importantes para este trabalho, pois algumas delas serão usadas no protótipo desenvolvido. Essas tecnologias também são importantes pelo fato de serem as mais usadas no meio acadêmico, fazendo parte assim do contexto em que o trabalho se encontra.

2.5.1 Apache Axis

Apache Axis (ALMAER, 2002) é um *framework* para desenvolvimento e implantação de Serviços Web. Dentre as funcionalidades do *Apache Axis* a principal é a implementação do *Simple Object Access Protocol* (SOAP) especificado pelo W3C.

Segundo Gudgin (2007, p.6), o SOAP é:

SOAP is a lightweight protocol for exchanging structured information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses.

Desenvolvido pela *Apache Foundation*, o *Apache Axis* é comumente usado no suporte e desenvolvimento de Serviços Web. Usado juntamente com um servidor de aplicativos disponibiliza interfaces capazes de receber mensagens via protocolo SOAP, que são usadas como requisições de serviços. Essa tecnologia é de extrema importância, pois a grande maioria dos softwares de código livre relacionados com a arquitetura SOA a usam. Por exemplo, as infra-estruturas citadas anteriormente o piPEs-BR/GFD e o perfSONAR, ou até mesmo a implementação do padrão UDDI, o JUDDI, que será descrito na próxima subseção.

O *Apache Axis* também disponibiliza bibliotecas que viabilizam funcionalidades para desenvolver clientes e realizar requisições aos serviços. Ou seja, essa tecnologia disponibiliza um conjunto de soluções para desenvolver, implantar e acessar Serviços Web.

2.5.2 JUDDI

JUDDI é uma implementação de código aberto em JAVA das especificações contidas no padrão UDDI para Serviços Web. Desenvolvido pela *Apache Foundation* se encontra na versão 0.9, a qual implementa a versão dois do padrão UDDI. Desenvolvida especificamente para Web, pode ser usada com qualquer servidor de aplicativos que suporte a versão 2.1 ou anterior do *servlet* API¹¹. Contudo, o JUDDI é um serviço de publicação e descoberta que disponibiliza suas funcionalidades através de Serviços Web, utiliza mensagens XML para comunicação e implementa seus Serviços Web via tecnologia *Apache Axis*, citada anteriormente.

Como o Serviço de Publicação e Descoberta também é conhecido como registro ou base de informações de Serviços Web, o JUDDI também utiliza uma base de dados externa, para armazenar as informações dos serviços. Portanto, é possível dizer que o JUDDI suporta as mais conhecidas bases de dados relacionais *open source* do mercado.

O ponto chave do armazenamento de informações no JUDDI é que, mesmo recebendo as informações a serem armazenadas em formato XML, o JUDDI consegue, através do *Schema* XML definido no UDDI, transformar os elementos das mensagens em objetos JAVA que facilitam a manipulação e a inserção das informações em uma base de dados relacional.

Essa implementação é bem relacionada na literatura, pois se trata de uma solução gratuita que é estável e contemplam todas as funcionalidades do UDDI versão 2.0, que podem ser consideradas as funcionalidades básicas para um serviço de publicação e descoberta. Outra vantagem do JUDDI é o fato de ser *open source* e possuir uma boa documentação, o que possibilita modificações.

¹¹Protocolos e servidor de componentes independente de plataforma escrito em JAVA. Sun Microsystems, *The Java servlet API Whitepaper*.

2.5.3 UDDI4J

O UDDI4J¹² é uma biblioteca JAVA de código aberto que provê acesso ao UDDI. Desenvolvida pela IBM, é uma biblioteca de fácil manuseio que envia requisições a um registro UDDI solicitando as funcionalidades oferecidas por esse registro.

O UDDI4J utiliza o protocolo SOAP baseado em XML (via tecnologia *Apache Axis*) para enviar mensagens e transformar os elementos das mensagens XML em objetos JAVA. Um usuário que deseja localizar um serviço, ou um próprio serviço que deseja se registrar no UDDI pode usar a biblioteca UDDI4J e realizar essas operações.

Outro exemplo de uso para a biblioteca UDDI4J é a possibilidade de se construir uma interface sobre essa biblioteca de acordo com as características do serviço. Por exemplo, o desenvolvedor para facilitar o acesso ao UDDI para os usuários pode criar novos objetos (tipo, descrição, nome e etc.) que abstraem os nomes e características dos objetos do UDDI que são análogos aos elementos do UDDI. Logo, o usuário não precisa saber detalhes do UDDI como: *bindingTemplates*, *tModel* e etc.

2.5.4 Web Sphere

O *Web Sphere* (COLGRAVE, 2004) é um conjunto de softwares da IBM que oferece: uma infra-estrutura para desenvolver e disponibilizar aplicações, a integração entre essas aplicações, a possibilidade de reuso e de estender funcionalidades de outras aplicações, gerência de processos de negócios, automação e integração para comércio eletrônico e soluções para dispositivos móveis. Muitas dessas funcionalidades oferecidas são baseadas na arquitetura SOA ou são usadas para implantar arquiteturas SOA. Um exemplo é que o *Web Sphere* possui uma implementação das normas UDDI.

O *Web Sphere* foi desenvolvido para tentar suprir todas as questões que envolvem aplicações Web. O *Web Sphere* possui seus próprios servidores, suas ferramentas para desenvolver

¹² Biblioteca JAVA de acesso ao registro UDDI.

aplicações e bibliotecas, ferramentas para desenvolver serviços, barramento de serviços e ferramentas que dão suporte a arquitetura SOA. O UDDI foi inserido no *Web Sphere* a partir da sua versão 6.0 e já essa versão implementa o UDDI na sua versão mais atual. Assim o *Web Sphere* se torna uma solução completa para a implementação, disponibilização, descoberta e publicação de Serviços Web.

2.5.5 Apache Tomcat

Desenvolvido pela *Apache Foundation*, o *Tomcat* é um servidor de aplicativos JAVA para Web. É uma implementação de software livre que possui código aberto e basicamente suporta as tecnologias *JavaServer Pages*.

Atualmente grande parte das aplicações que disponibilizam Serviços Web utiliza o *Tomcat* como Servidor Web para disponibilizar os serviços e receber as requisições. Um dos motivos pra essa escolha é sua estabilidade, pois ao longo dos anos o software vem sendo utilizado e recomendado pelo fato de já ser utilizados em ambientes de produção de forma satisfatória.

É importante citar o *Tomcat*, pois ele será o servidor que irá disponibilizar o protótipo desenvolvido neste trabalho bem como a implementação JUDDI.

2.6 CONSIDERAÇÕES

Neste capítulo foram apresentados alguns padrões e tecnologias que estão envolvidos com as arquiteturas orientadas a serviços. Nele foi possível observar o componente da SOA, o *Service Broker*, e a sua importância para a arquitetura. Foram apresentados também alguns padrões que definem o *Service Broker* e que podem ser usados no escopo deste trabalho. Portanto, este capítulo também apresentou o padrão usado neste trabalho, o UDDI, bem como as suas funcionalidades e características.

Para pesquisar sobre as peculiaridades dos Serviços Web de monitoramento, é preciso previamente saber as características dos serviços Web de forma genérica. Sendo assim, esse capítulo mostrou o assunto Serviços Web, descrevendo suas características e definições, bem como exemplos de seu uso. Outro ponto abordado nos Serviços Web foi a utilização do WSDL na publicação e descoberta. O WSDL assim como o serviço de publicação e descoberta possui informações de acesso ao serviço, logo o WSDL pode ser usado no auxílio a publicação e descoberta com suas informações.

Outro ponto abordado neste capítulo, é que como este trabalho propõe um Serviço de Publicação e Descoberta utilizando o padrão UDDI; é necessário então apresentar as possíveis tecnologias a serem usadas no escopo desse serviço.

Com a leitura do capítulo de Arquitetura Orientada a Serviços – SOA, é possível observar que existem padrões e tecnologias que podem ser utilizados na criação de um Serviço de Publicação e Descoberta de Serviços Web. Sendo que, para disponibilizar a publicação e descoberta para os Serviços Web de monitoramento é necessário um desenvolvimento a parte, que considere as características dos Serviços Web de monitoramento, bem como as características dos usuários dos Serviços Web de monitoramento.

3 INFRA-ESTRUTURAS DE MONITORAMENTO

Atualmente existem diversas infra-estruturas de monitoramento de redes e, como já mencionado, existe uma forte tendência para o uso da Arquitetura Orientada a Serviços (SOA) na estrutura dessas infra-estruturas. Existem também algumas infra-estruturas que não seguem as especificações de uma arquitetura orientada a serviços, mas que usam Serviços Web para fornecer algumas de suas funcionalidades.

Este capítulo aborda um ponto importante no cenário de medições em redes de computadores que é o *General Framework Design* (GFD). Neste capítulo também serão descritas algumas funcionalidades, estrutura e aplicação das mais importantes infra-estruturas de monitoramento, onde também serão analisados nessas infra-estruturas, aspectos tais como o uso da arquitetura orientada a serviços e a publicação e descoberta, caso se aplique. Essa abordagem é extremamente necessária, pois, a publicação e descoberta proposta neste trabalho está inserida no contexto das infra-estruturas de monitoramento, mais precisamente no piPEs-BR/GFD e perfSONAR.

3.1 GENERAL FRAMEWORK DESIGN (GFD)

Hoje em dia as redes empregam diversas ferramentas para monitorar uma variedade de características da rede, desde uma simples conexão a uma análise detalhada de um protocolo. Assim, os mecanismos usados para realizar testes, coletar dados e até mesmo apresentar esses dados para os usuários diferem entre si de acordo com o domínio em que está sendo utilizado. O modo de acesso, assim como as políticas de acesso a esses dados também diferem em domínios administrativos diferentes.

Contudo, um gerente de rede que deseja detectar o baixo desempenho em uma conexão fim-a-fim geralmente não possui condições de detectar o problema ao longo do caminho, pois é necessária a utilização de vários mecanismos ou pontos que possam desdobrar esse caminho para detectar com precisão o local do problema. Outro fator é que esse caminho pode

percorrer domínios administrativos diferentes e logicamente o gerente de um domínio pode não ter acesso aos dados de outro domínio. Isso faz com que o gerente não tenha uma visão global do trajeto fim-a-fim e sim uma visão somente do seu domínio.

Para tentar solucionar problemas como os citados anteriormente, a Internet2 e Géant2 se uniram para definir uma infra-estrutura de monitoramento para que as medições atinjam múltiplos domínios. Essa definição está contida em um documento chamado de *General Framework Design* (GFD). O objetivo principal do GFD é detalhar e definir uma camada de *middleware* entre as ferramentas de visualização e as ferramentas de medições existentes entre os domínios. Na Figura 3.1 é possível ver essa camada. Essa camada seria disponibilizada por uma arquitetura orientada a serviços (vide Capítulo de Arquitetura Orientada a Serviços – SOA).

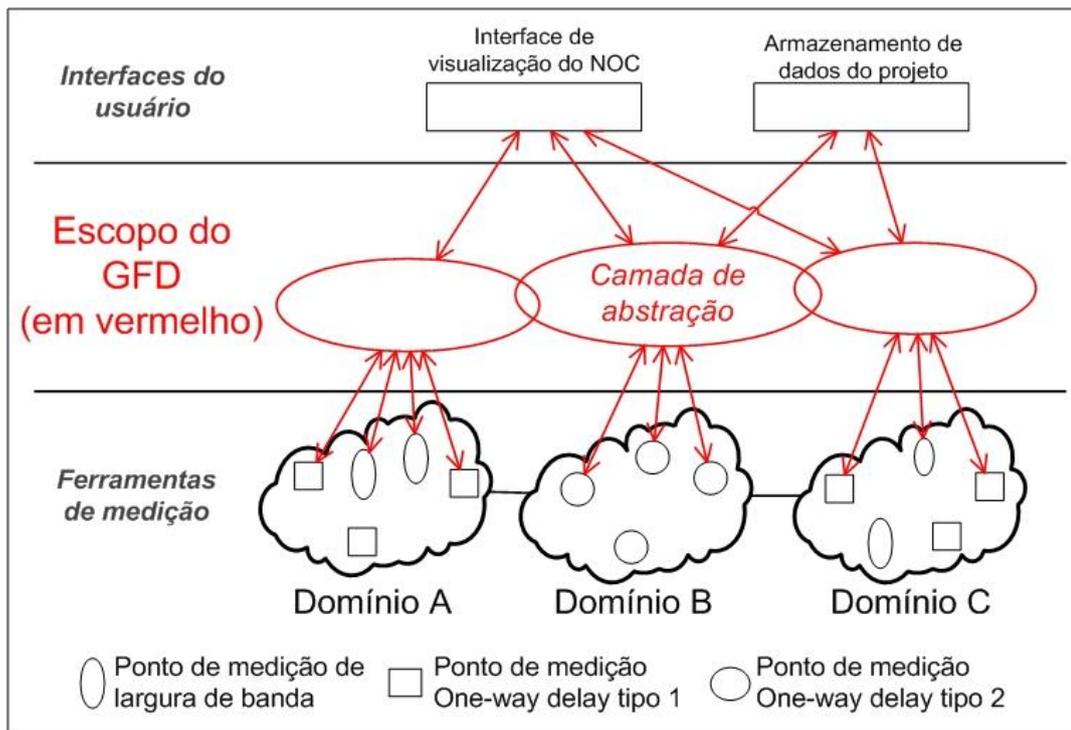


Figura 3.1 - Arquitetura do GFD
Fonte: HANEMANN (2005).

Num cenário onde um administrador de rede deseja monitorar o status e o desempenho das redes e, por sua vez, um usuário comum também possui o mesmo desejo, mas com finalidade diferente, pode receber como mecanismo para o fornecimento das informações necessárias, uma infra-estrutura implementada de acordo com as definições contidas no GFD. Pois essa infra-estrutura deve ser capaz de fornecer informações localizadas em pontos específicos da rede possibilitando administradores identificar possíveis problemas, bem como informações

que auxiliem os usuários no uso de suas aplicações com o auxílio de dados de medições fim-a-fim. Contudo, a infra-estrutura de monitoramento deve ser capaz de receber vários tipos de requisições como: execução de testes, armazenamento de dados, recuperação de dados e outros. Essas requisições são solicitações aos serviços disponibilizados pela infra-estrutura que podem estar aglomerados em um único componente ou em vários, como exemplo um serviço de armazenamento de dados que armazena e recupera dados.

Os principais serviços descritos no GFD são: Ponto de Medição (*Measurement Point* - MP), serviço de armazenamento (*Measurement Archive* - MA), serviço de publicação e descoberta (*Lookup Service* - LS), serviço de transformação (*Transformation Service* - TS), serviço de topologia (*Topology Service* - TopS) e o protetor de recursos (*Resource Protector* - RP).

- a) **armazenamento de medições** - O Serviço de Armazenamento (*Measurement Archive* - MA) é usado para guardar os dados históricos de medição em um arquivo ou em uma base de dados. Um arquivo pode ser, por exemplo, dados armazenados em um arquivo com a extensão “txt”, ou uma base de dados pode ser *Round Robin Database* (RRD). Geralmente o MA pode armazenar dados medido por um ponto de medição ou dados transformados pelo serviço de transformação. O MA pode simplesmente também armazenar dados que foram medidos por um dispositivo que não faz parte da infra-estrutura. Por exemplo, dados coletados de um roteador que são armazenados em uma base RRD e que é acessada pelo MA, disponibilizando assim esses dados.

Este serviço deve ser projetado para possuir uma alta disponibilidade para poder disponibilizar os dados para os clientes. Isso pode ser feito de muitas maneiras, incluído a escolha de softwares e hardware de ótima confiança, fazendo backup das informações, proporcionando redundância dos dados e recuperação de falhas;

- b) **Publicação e Descoberta** - É altamente incomodo e difícil descobrir a localização e as funcionalidade de um serviço que pode estar até em domínios diferentes. O *Lookup Service* (LS) permite os usuários descobrirem os serviços (MA, MP, TS, etc.), além de descobrir também informações sobre as funcionalidades do serviço e suas características. O LS fornece para o usuário todas as informações necessárias para acessar o serviço. Essencialmente o LS age como um diretório de serviços, onde os serviços possam se publicar e os

usuários possam localizar esses serviços. Algumas dessas informações podem ser escondidas dependendo da permissão do usuário.

O *Lookup Service* é um componente chave da infra-estrutura de monitoramento porque ele torna possível o uso dos serviços de forma dinâmica e independente, tornando o Serviço Web uma parte visível da infra-estrutura;

- c) **Autenticação** - Alguns domínios podem restringir ou oferecer uma forma diferenciada no acesso a algumas funcionalidades dos serviços, para um grupo de usuários. O Serviço de Autenticação (*Authentication Service* - AS) provê a funcionalidade de autenticação (AuthN) para a infra-estrutura, assim como a autorização para usar um determinado serviço ou somente algumas de suas funcionalidades. Com as informações fornecidas pelo LS, o serviço local pode permitir acesso a um conjunto de suas funcionalidades para um usuário de acordo com o grupo ao qual ele faz parte;
- d) **Transformação** - O Serviço de Transformação (*Transformation Service* - TS) é usado para fazer a tradução, correlação e filtragem dos dados entre os outros serviços da infra-estrutura. Pode ser usado para executar qualquer função em cima dos dados. Pode atuar nos dados dos “produtores”, por exemplos os MPs, ou nos consumidores, por exemplo, clientes de visualização. Existem várias funcionalidades que podem ser inseridas no serviço de transformação. Um exemplo é a agregação de dados. Por exemplo, um serviço de transformação pode comprimir uma série de dados atuais coletados de forma bem detalhada e transformá-los em dados históricos que podem ser uma média de um intervalo maior;
- e) **Topologia** - O Serviço de Topologia (*Topology Service* - TopS) é responsável por tornar as informações de topologia da rede disponíveis para a infra-estrutura de monitoramento. Geralmente os clientes de visualização fazem uso dessa informação de topologia para criar mapas das redes. Num primeiro estágio o TopS pode focalizar na camada do IP. Uma versão mais refinada que pode se aprofundar ainda mais na topologia da rede pode ser desenvolvida gradativamente.
O TopS irá coletar informações de topologia de diversas fontes como, por exemplo, de dispositivos de redes e usa um algoritmo para deduzir essa topologia. O serviço de rede pode também refletir a camada de rede, isso uma

descrição no nível de domínios. Essas informações de topologia também podem conter informações geográficas como coordenadas de GPS;

O serviço de topologia também deve ser capaz de informar dados de proximidades entre as peças que compõem a topologia ou até a distância entre serviços da infra-estrutura. Por exemplo, um usuário gostaria de saber qual o ponto de medição mais próximo do seu computador para realizar testes.

- f) **Protetor de Recursos** - O Protetor de Recursos (*Resource Protector* - RP) atua como um árbitro, limitando o uso de dados de medições nas redes. É o RP quem libera a execução de qualquer teste pelos componentes da infra-estrutura. Por exemplo, em um canal de 1Mbps é possível atribuir no RP que desse total somente 100kbps serão disponibilizados para dados de medição. Se esse teto for atingido, o RP não liberará a execução de novos testes naquele canal ou testes que passam por aquele canal. O RP deve somente limitar o uso de recursos em enlaces pequenos ou enlaces sobrecarregados.

3.2 INFRA-ESTRUTURAS DE MONITORAMENTO

3.2.1 PingER

A infra-estrutura *Ping End-to-end Reporting* (pingER) (MATTHEWS; COTTRELL, 2000) é baseada na ferramenta ping, que é uma ferramenta muito familiar para os administradores de rede e que também é bastante usada por usuários de redes. Basicamente a infra-estrutura gerencia um conjunto de medidores (servidores), que dependendo da configuração enviam pacotes ICMP (*Internet Control Message Protocol*) (POSTEL, 1981) para pontos (sistemas finais ou roteadores) específicos da rede usando a ferramenta ping, para monitorar o desempenho entre domínios.

O pingER é capaz de medir o percentual de pacotes perdidos e o tempo de resposta (*Round Trip Time* - RTT)¹³ entre dois ou mais pontos. Atualmente o projeto pingER possui vários

¹³ RTT é o tempo total para que um pacote (ou datagrama) seja enviado e retorne à origem.

hosts e medidores espalhados pelas rede de computadores em vários países, medindo constantemente. Essas informações são disponibilizadas na Internet para que usuários das redes possam obtê-las.

A arquitetura do pingER é composta por três componentes. Na Figura 3.2 esses componentes podem ser visualizados, bem como a relação entre os mesmo. Já a descrição de cada componente pode ser feita assim:

- a) **medidor remoto** - É um simples ponto que de forma passiva responde aos *pings*. Esse *host* só precisa possuir alguns requisitos como disponibilidade e permissão para responder à solicitação de ecos do ICMP;
- b) **medidor** - Possui uma ferramenta chamada *PinGER Monitoring Tool*, a qual executa a ferramenta ping, para fazer testes até os Medidores Remotos. Ela também envia para o armazenador os resultados das medições. O Medidor também faz algumas análises dos dados medidos, isso de acordo com as configurações. Essas análises consistem em: geração de gráficos diários, geração de gráfico 3D composto pela hora do dia, o host que está sendo testado e a métrica medida. A análise também realiza médias mensais que enfatizam os horários de pico das medidas e previsões com base em dados históricos. É importante salientar que o envio das informações para o armazenador é feito via Serviços Web;
- c) **armazenador** - Esse componente é responsável por armazenar os dados coletados pelos medidores e por realizar operações sobre esses dados, sendo que numa visão mais global como: compor os dados, analisar dados e etc. O armazenador disponibiliza serviços web para que os medidores possam armazenar os dados coletados. E, por sua vez, disponibilizam essas informações via web.

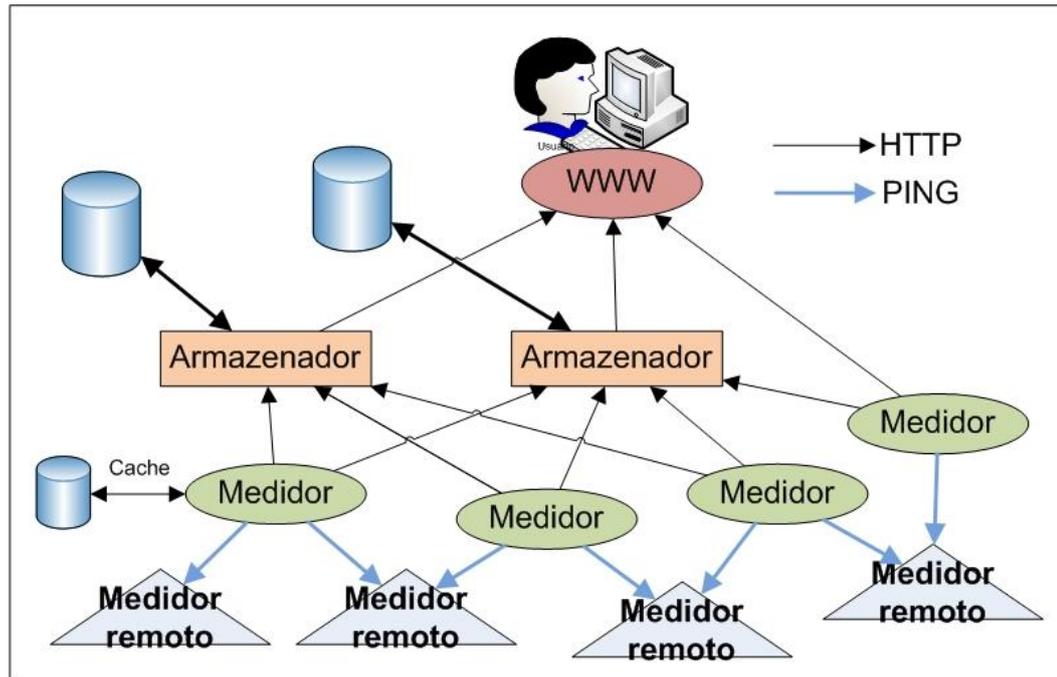


Figura 3.2 - Cenário de medições com a infra-estrutura pingER
 Fonte: MATTHEWS (2000).

3.2.2 MonALISA

A MonALISA (*Monitoring Agents using a Large Integrated Services Architecture*) é um *framework* que provê um sistema de serviços de monitoração distribuído, usando as tecnologias JINI/JAVA¹⁴ e WSDL/SOAP e está baseado em uma Arquitetura Dinâmica de Serviços Distribuídos (do inglês *Dynamic Distributed Services Architecture* - DDSA) (NEWMAN e outros, 2001). A MonALISA é formada por componentes que agem como um sistema independente, provendo informações a serem descobertas e utilizadas pelos demais usuários clientes. Esse sistema é composto por dois componentes principais: o serviço MonALISA (*MonALISA Service*) e o cliente MonALISA (*MonALISA Client*) onde o serviço é responsável pelas medições e publicação dos dados e o cliente é encarregado de realizar a visualização dos dados fornecidos pelo serviço.

Assim, o serviço MonALISA contém módulos que realizam coleta de dados ou testes com ferramentas de medição. Apesar de ser uma ferramenta proprietária, existe ainda a possibilidade de desenvolver módulos para disponibilizar outros tipos de dados. Ao se

¹⁴ Página Web do JINI, disponível em <http://www.jini.org>

desenvolver módulos adicionais para o MonALISA, é necessário que estes sejam desenvolvidos em linguagem Java. Neste caso, esses módulos são classes que herdam funcionalidades de outras classes do próprio serviço MonALISA.

Basicamente, cada serviço MonALISA pode ser composto por uma ou mais “*Farms*”, que é o nome dado a um conjunto de módulos, onde essas “*Farms*” realizam a coleta de dados em dispositivos ou executa ferramentas de medição tais como o Ping e o IPerf¹⁵. Essas “*Farms*” são visualizadas no cliente MonALISA como pontos na superfície de um globo terrestre, e que quando são selecionados fornece as informações do serviço MonALISA correspondente. A Figura 3.3 mostra a tela de um cliente MonALISA.

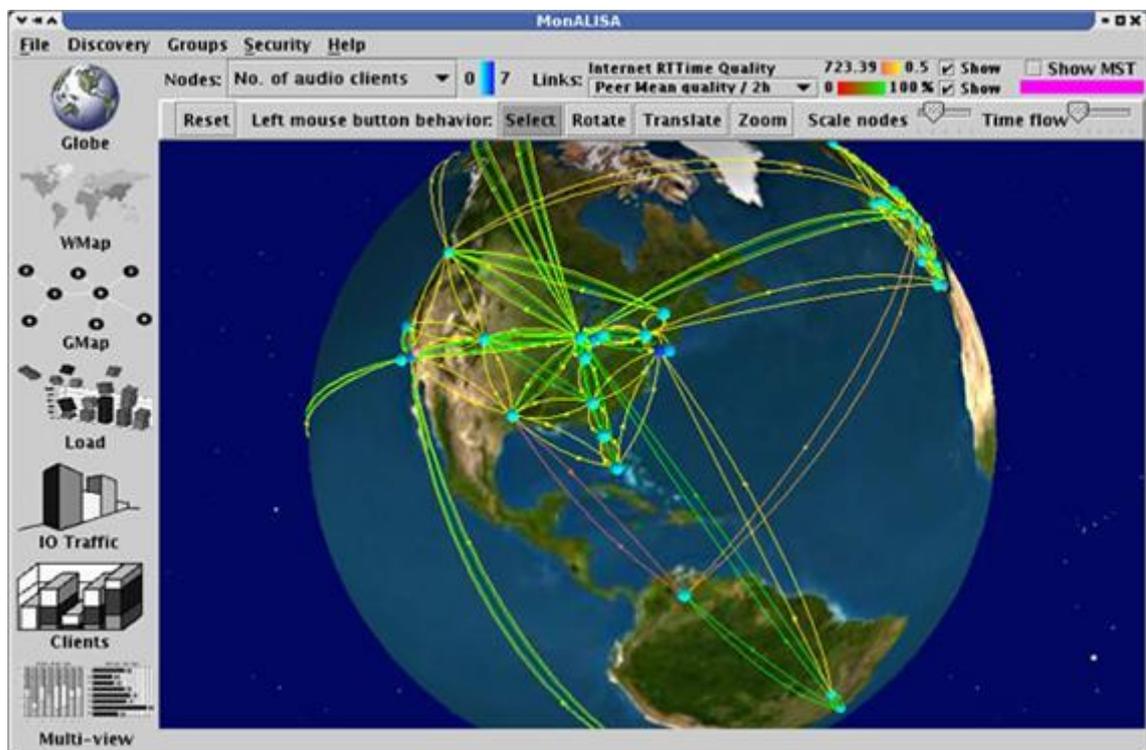


Figura 3.3 - de visualização da MonALISA
Nota: elaboração do autor (2007).

Na Figura 3.4 é possível ver a interação entre todos os elementos envolvidos com o MonALISA. Ao lado direito é possível ver o cliente de visualização (cliente MonALISA), que como passo inicial conecta em um serviço de busca e coleta as informações necessárias para acessar e exibir no globo as “*Farms*” existentes. Com esse procedimento completo, o cliente pode conectar em cada “*Farm*” (serviço monALISA) e requisitar os dados disponíveis. No

¹⁵ Ferramenta de monitoramento que mensura a largura de banda disponível entre dois computadores.

centro da figura, é possível ver o serviço MonALISA e como ele interage com as ferramentas e dispositivos através de seu módulos.

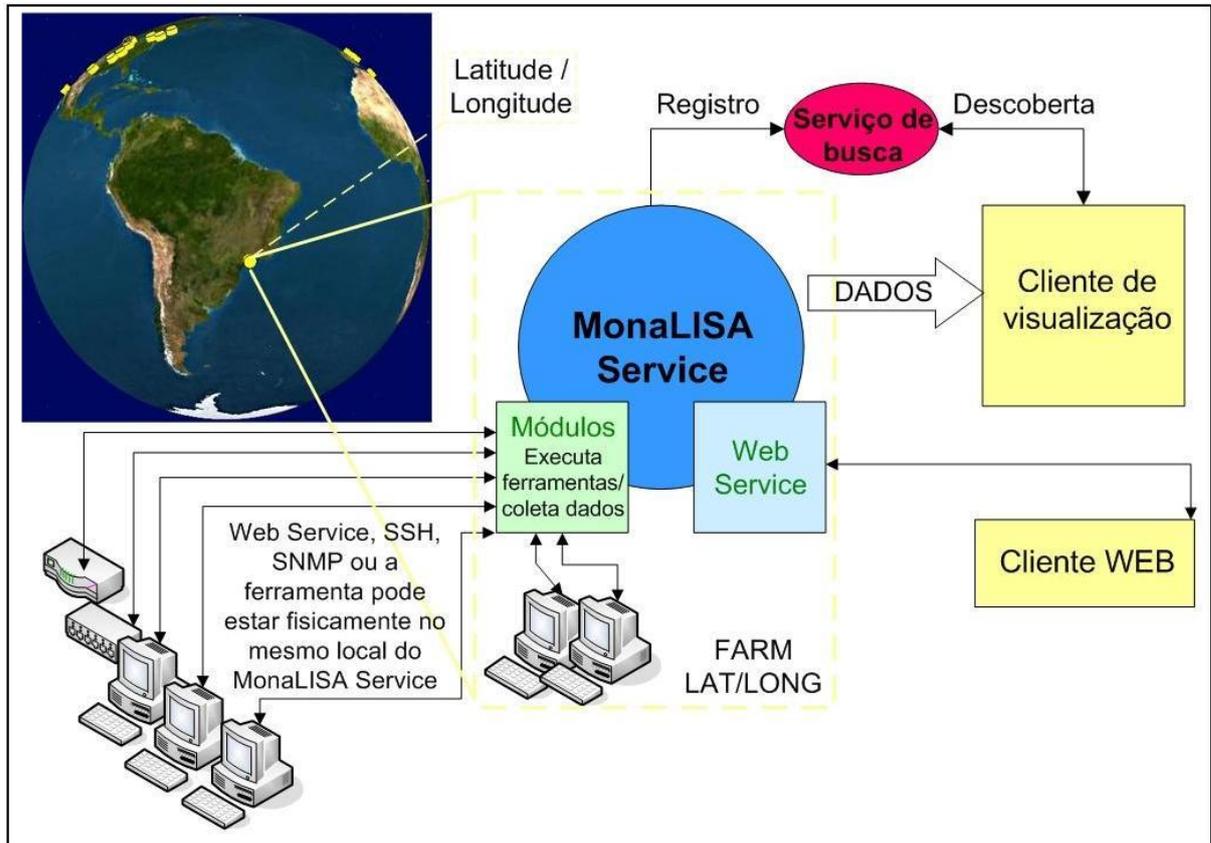


Figura 3.4 - Interação entre os diversos componentes do MonALISA
Nota: elaboração do autor (2007).

3.2.3 perfSONAR

O perfSONAR (*Performance focused Service Oriented Network monitoring Architecture*) é o protótipo de validação do *framework* GFD. O perfSONAR está sendo desenvolvido pela Internet2, Géant2 e recentemente juntaram-se ao grupo pesquisadores da RNP e ESnet. Numa abordagem inicial, o grupo definiu que para a primeira versão do protótipo seriam disponibilizados apenas dois serviços e um cliente de visualização. Os elementos disponibilizados na versão 1.0 foram:

- a) **Measurement Archive RRD** - Serviço que armazena dados coletado por dispositivos da rede (roteadores) no formato RRD. Esses dados são referentes a

utilização das interfaces desses dispositivos, assim o MA-RRD disponibiliza o nível de utilização das interfaces dos dispositivos de rede;

- b) **Lookup Service XML** - Serviço de publicação e descoberta de Serviços Web, que utiliza base de dados XML para armazenar as informações sobre os Serviços Web;
- c) **perfSONARUI** - Cliente de visualização que faz acesso ao MA-RRD e disponibiliza gráficos com os dados das interfaces.

Com a chegada do GT-Medições da RNP e pesquisadores da ESnet para juntar esforços, a versão 2 do perfSONAR incorporou vários pontos de medição, dentre eles o MP de linha de comando (CLMP - *Command Line Measurement Point*) desenvolvido pelo GT-Medições. Outro novo serviço do perfSONAR 2.0 é o *Measurement Archive SQL* onde são armazenados os dados medidos pelo CLMP. Na Figura 3.5 é possível ver um exemplo da estrutura e interação dos componentes do perfSONAR na sua versão 1.0. Nessa figura, o elemento mais à esquerda é formado pelos dispositivos onde são coletados os dados, que no caso da versão 1.0 do perfSONAR se trata dos dados de utilização das interfaces de equipamentos de redes. O próximo componente é uma base de dados que armazena os dados de medições no formato RRD, que por sua vez fornece esses dados para o MA. Todavia, o perfSONARUI, que é o cliente de visualização pode localizar o MA utilizando o Serviços LS (publicação e descoberta) e assim recuperar os dados medidos via MA.

O perfSONAR usa um protocolo próprio para a troca de mensagens na comunicação interna e externa. Esse protocolo é baseado em mensagens XML SOAP que seguem os padrões definidos pelo *Network Measurement Working Group* (NMWG¹⁶) do *Open Grid Forum* (OGF¹⁷), para medições em redes de computadores. Logo, as mensagens de comunicação usada no perfSONAR são chamadas de mensagens do tipo NMWG.

¹⁶ Grupo de pesquisa internacional em redes de computadores.

¹⁷ Comunidade de padronização para *grid* computacionais.

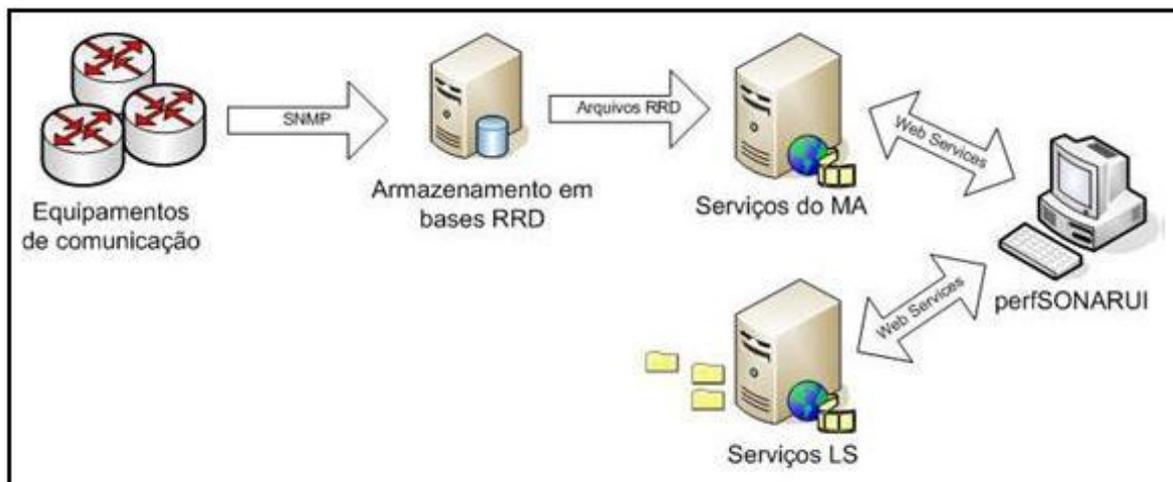


Figura 3.5 - Componentes do protótipo perfSONAR 1.0

Nota: elaboração do autor (2007).

O esquema NMWG distribui as informações dos dados de medições em elementos básicos chamados de *Metadata* e *Data*, onde o *Metadata* contém informações consideradas genéricas que identificam um conjunto de dados de medições. Já o objeto *Data* armazena o dado propriamente dito, ou seja, pode ser um valor ou uma informação específica. Esses elementos podem ser combinados formando um conjunto de informações que podem ser analisadas facilmente. Na Figura 3.6 é possível ver esses dois elementos e suas relações.

Como exemplo é possível utilizar uma mensagem que contém o resultado de uma requisição de dados de medição unidirecional entre dois pontos. Onde o *Metadata*, por exemplo, pode conter as informações entres os dois pontos, o intervalo de tempo medido, a ferramenta usada e etc. Já a informação contida no elemento *Data* seria o valor da medição. Caso o resultado contenha mais de um valor de medição, estes valores estarão distribuídos em mais de um elemento *Data*.

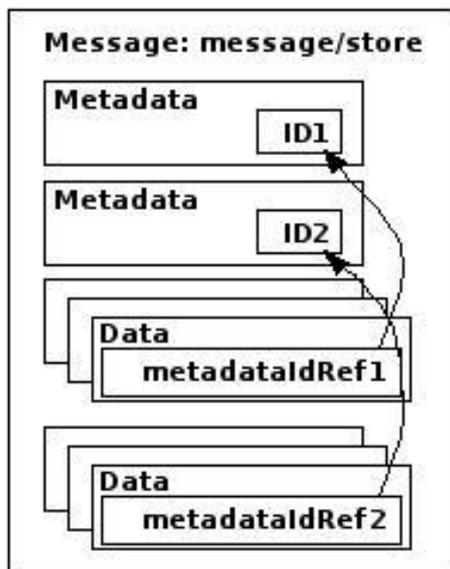


Figura 3.6 - Estrutura da mensagem NMWG
Nota: elaboração do autor (2007).

A seguir serão descritos os componentes da versão 1.0 do perfSONAR:

- a) **Measurement Archive RRD (MA RRD)** - O MA RRD foi inicialmente desenvolvido para armazenar e disponibilizar dados de utilização de interfaces de dispositivos de redes, especificamente roteadores. Essa abordagem foi adotada pelo simples fato de que uma grande parte das redes acadêmicas possui uma base de dados em RRD com os dados de utilização das interfaces dos roteadores que compõem as redes acadêmicas. O MA RRD basicamente faz a interface de acesso padrão do perfSONAR, disponibilizando alguns serviços para recuperar esses dados

Atualmente existe versões da base RRD do perfSONAR que trabalham de forma mais genérica. Ou seja, possibilitam o armazenamento e o fornecimento de dados RRD independentemente da sua origem. Esses dados podem ser obtidos através de uma ferramenta de medição que deseja utilizar as vantagens de uma base RRD. Podem ser também dados de uma ferramenta de medição passiva que geralmente se relacionam bem com as bases RRD devido à quantidade de informação e outros;

- b) **Lookup Service XML Type (LS-XML)** - O LS-XML realiza a publicação e descoberta de Serviços Web do perfSONAR. O LS pode receber mensagens solicitando o registro, consulta, alteração e remoção de informações sobre serviços web. Cada Serviço Web possui um conjunto de informações que são

armazenadas periodicamente pelos próprios Serviços Web. Essas informações são armazenadas em uma base de dados XML, onde a implementação usada dessa base é denominada de eXist¹⁸. A escolha desse tipo de base de dados se deu pelo simples fato de que as mensagens contendo as informações sobre o Serviço Web já são “escritas” em XML. Desse modo o armazenamento se torna simples, pois é possível extrair parte da própria mensagem e armazenar diretamente na base de dados. Esse tipo de abordagem dispensa a manipulação dos dados, o que facilita a programação em vários aspectos. Mas, também traz alguns problemas para o serviço, como a não verificação se os dados estão corretos;

- c) *perfSONARUI* - A PerfSONARUI (HANEMANN, 2006) é uma ferramenta de visualização desenvolvida em JAVA, que teve como finalidade inicial demonstrar as funcionalidades do perfSONAR. Basicamente a ferramenta recupera dados do serviço de armazenamento MA e os disponibiliza graficamente para os usuários. Vale ressaltar que como protótipo inicial o perfSONAR possuía apenas o MA-RRD, logo o perfSONARUI possuía apenas a visualização para esses tipos de dados. Sendo que existem previsões de disponibilização de outros tipos de informações.

Atualmente o perfSONARUI não usa os *Lookup Service* para localizar os serviços usados. O perfSONARUI possui as informações de acesso aos serviços armazenadas em arquivos de configuração ou na própria codificação. Isto quer dizer que qualquer alteração nas informações de acesso aos serviços a ferramenta deixa de ser funcional a menos que seja reinstalada com as novas informações.

3.2.4 piPEs-BR/GFD

A infra-estrutura de monitoramento de redes de computadores piPEs-BR/GFD é uma evolução da arquitetura piPEs-BR desenvolvida no primeiro ano do GT-Meidições. A idéia por trás do piPEs-BR/GFD é disponibilizar um conjunto de softwares que siga as

¹⁸ Implementação de uma base de dados XML nativo.

especificações contidas no *General Framework Design* (GFD). É possível dizer também que o piPEs-BR/GFD é a versão brasileira do perfSONAR, que leva em consideração a experiência do GT-Medições na RNP. Outro fator que justifica a afirmação de que o piPEs-BR/GFD é uma versão brasileira do perfSONAR é que o GT-Medições ao vislumbrar um interesse em comum com os grupos de pesquisa da Internet2 e Géant2 em desenvolver uma infra-estrutura com base no GFD decidiu desenvolver em sua infra-estrutura de monitoramento componentes diferentes dos previstos no perfSONAR. Tendo essa meta em vista o GT-Medições expôs suas intenções aos grupos desenvolvedores do perfSONAR. Deste modo, o GT foi convidado para participar no desenvolvimento do perfSONAR, agregando assim seus módulos desenvolvidos ao protótipo. Assim, o perfSONAR e o piPEs-BR/GFD são infra-estruturas que se completam, onde uma pode usar os módulos da outra, devido à padronização das mensagens e ao uso de serviços Web assim como definido na SOA.

O piPEs-BR/GFD é composto por um conjunto de módulos que visam contemplar as funcionalidades de: agendamento de testes, execução de testes, armazenamento, autorização e visualização de dados medidos. Inicialmente, em termos de protótipo, o GT-Medições desenvolveu para a RNP um módulo de armazenamento, um serviço de publicação e descoberta, um ponto de medição e um cliente de visualização. Estes módulos interagem entre si através de serviços Web, disponibilizando e requisitando serviços com funcionalidades específicas e bem definidas. A arquitetura do piPEs-BR/GFD pode ser vista na Figura 3.7, bem como a interação entre os módulos. Os principais módulos do piPEs-BR/GFD são: o Ponto de Medição de Linha de Comando (*Command Line Measurement Point - CLMP*), na figura representado apenas por MP (X, Y e Z), o módulo de armazenamento de dados de medição (*Measurement Archive Sql*) que na figura é representado pela sigla MA e uma base de dados de medição, o serviço de publicação e descoberta UDDI (*Lookup Service UDDI Type - LS-UDDI*) e a ferramenta de visualização *Internet Computer Eye - ICE* (KOGA e outros, 2007). Cada um desses componentes serão descritos a seguir.

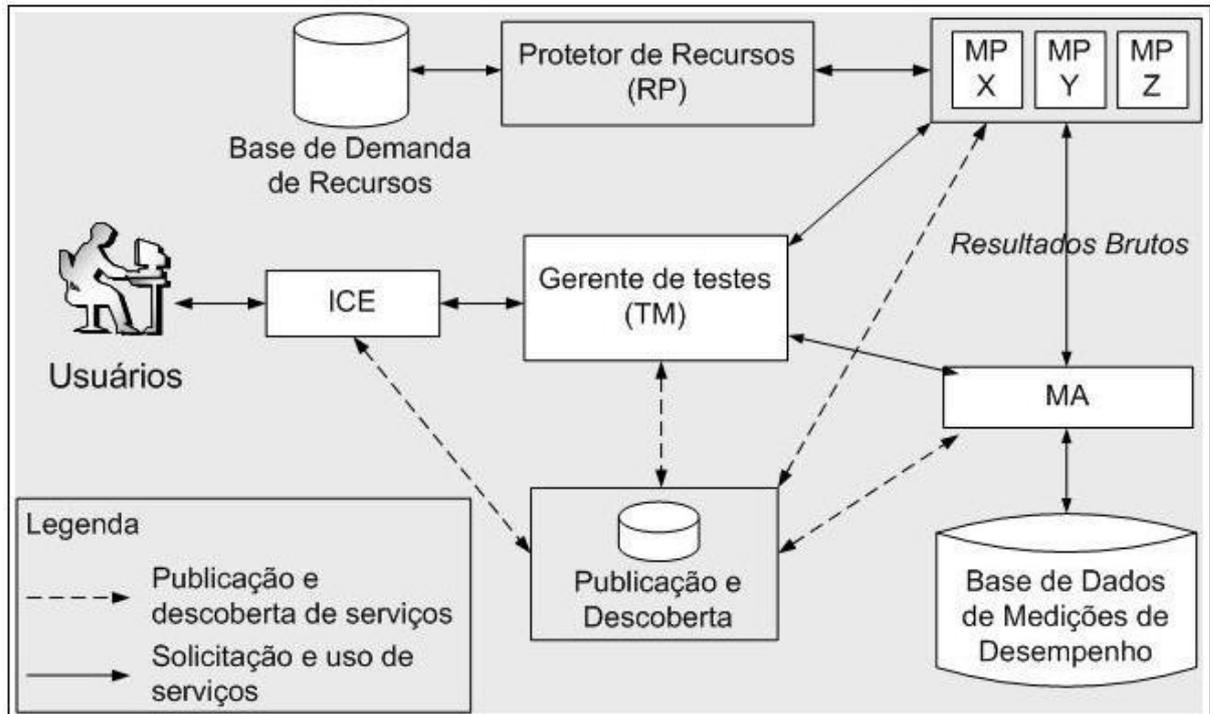


Figura 3.7 - Arquitetura do piPEs-BR/GFD
 Fonte: SAMPAIO e outros (2007).

3.2.4.1 Descrição dos módulos

- a) **CLMP - Ponto de Medição de Linha de Comando** - O ponto de medição cria os dados de monitoramento a partir da realização de testes de medições ativa ou passiva¹⁹. No caso do CLMP ele disponibiliza uma interface para realizar teste de medições ativas através das ferramentas de linha comando, ou seja, ferramentas cuja interação se dá através da linha de comando da interface *shell*²⁰ do sistema operacional. OWAMP²¹, IPERF e PING são exemplos de ferramentas de linha de comando.

O CLMP é capaz de receber requisições de testes sob demanda e requisições para agendamento de testes. Esse agendamento pode ser usado para configurar

¹⁹ Nas medições passivas são coletadas informações sobre todos os pacotes que trafegam na rede sem provocar nenhuma interferência no tráfego. Nas medições ativas são gerados pacotes de teste e é monitorado o desempenho para os mesmos através da rede.

²⁰ Software que faz o intermédio entre o Sistema Operacional e o usuário.

²¹ Ferramenta que implementa o *One-Way Active Measurement Protocol* e fornece perda de pacotes, variação do atraso e o atraso entre dois computadores. O OWAMP fornece dados de medição unidirecional, o qual para ser coletado; precisa de sincronismo entre os envolvidos.

testes regulares ou executar teste em um horário específico desejado pelo usuário. Outra funcionalidade do CLMP é armazenar no módulo de armazenamento, o MA, os resultados dos testes agendados. Assim como os outros módulos do piPEs-BR/GFD, o CLMP realiza o registro de suas informações de acesso no Serviço de Publicação e Descoberta;

- b) **armazenamento de medições com Sql - MA-Sql** - Módulo responsável por armazenar dados de medição. A origem desses dados é diversificada o que torna a abrangência do MA em relação ao armazenamento bastante ampla. O MA pode armazenar os dados medidos pelos pontos de medição, pode armazenar dados coletado de dispositivos de redes, pode armazenar dados provenientes de ferramentas de monitoramento e pode armazenar dados modificados por um serviço de transformação;
- c) **serviço de publicação e descoberta** - O serviço de publicação e descoberta do piPEs-BR/GFD é proposto neste trabalho, logo maiores informações podem ser obtidas no Capítulo de Publicação e Descoberta de Serviços de Monitoramento;
- d) **Internet Computer network Eye - ICE** - A idéia inicial do ICE era somente de prover acesso a diferentes métricas de desempenho, prover acesso aos serviços do piPEs-BR e demonstrar visualmente os dados provenientes desse acessos. Com a intenção de tornar possível que outras aplicações também pudessem ter acesso ao piPEs-BR/GFD, o ICE incorporou a arquitetura FLAVOR (KOGA e outros, 2007) que permite o desenvolvimento de *bundles* OSGI (MARPLES; KRIENS, 2001) que podem ser reusados e estendidos por outras aplicações que desejam incorporar as funcionalidades do ICE (KOGA e outros, 2007).

3.3 CONSIDERAÇÕES

O uso de uma determinada infra-estrutura depende diretamente da demanda de informação por parte dos usuários e administradores das redes, pois existem no mercado diversas opções, para diversas finalidades. No entanto, é possível afirmar que devido a um interesse maior por parte dos usuários em informações de medições diversificadas, as infra-estruturas que abrangem várias métricas, usam várias ferramenta e que atingem vários domínios fazem parte

das soluções inovadoras em medições em redes de computadores. Isso pode ser explicado também pelo interesse de várias redes acadêmicas e de pesquisa em apoiar o projeto do perfSONAR.

A infra-estrutura pingER pelo fato de usar somente uma ferramenta, o *ping*, só atinge uma parte dos usuários de redes. Por exemplo, os usuários de *grids* computacionais que desejam saber principalmente o atraso entre os computadores. Isso difere de usuários de voz sobre IP que além do atraso desejam saber também qual é a banda disponível. Outro fator que pode restringir o número de usuários para o pingER é o fato de a ferramenta *ping* só fornecer informações bidirecionais, o que pode não satisfazer um administrador de rede que deseja saber em que sentido do link está ocorrendo perdas, por exemplo.

No entanto na infra-estrutura pingER é possível observar o uso de Serviços Web nos componentes que são distribuídos em domínios diferentes, mas essa infra-estrutura não utiliza os três componentes integrantes da arquitetura orientada a serviços. Para auxiliar a localização dos Serviços Web por parte dos usuários é necessário utilizar um Serviço de Publicação e Descoberta. O pingER faz o que comumente infra-estruturas que não implementa os três componentes da SOA fazem: armazena em um arquivo XML as informações de acesso aos Serviços Web. Esse tipo de abordagem faz com que parte da infra-estrutura pare de funcionar caso as informações de acesso aos Serviços Web mudem. Por exemplo, o Medidor ao tentar armazenar as informações coletadas no armazenador via Serviços Web pode não obter êxito, isso porque a URI de acesso ao Serviço Web do medidor pode ter mudado (o que é comum) e as informações de acesso contidas no arquivo XML estão desatualizadas. Com o uso do UDDI na infra-estrutura pingER, problemas como esse seriam sanados, pois essa operação iria ser automatizada.

Já na infra-estrutura MonALISA ocorre um fato diferente em relação à publicação e descoberta. Como visto na Figura 3.4, a MonALISA possui um serviço de publicação e descoberta, mas esse serviço é baseado na tecnologia JINI. O JINI é um conjunto de mecanismos para a construção de arquiteturas orientadas a serviços, disponibilizando ferramentas, bibliotecas JAVA e definindo um modelo de programação para desenvolver aplicações seguras e distribuídas. Resumindo, o JINI provê um conjunto de soluções para a construção de uma arquitetura orientada a serviços, contemplando os três componentes da SOA. Assim, o JINI possibilita o desenvolvimento de serviços, proporciona a publicação e

descoberta dos mesmos e possibilita também o desenvolvimento de clientes que os utilizem. É importante salientar que os serviços disponibilizados pelo JINI não são Serviços Web.

Além do mais, o serviço de publicação e descoberta da MonALISA é uma solução proprietária e só pode ser utilizado por clientes e serviços da própria infra-estrutura MonALISA. Isso implica em que nenhuma outra aplicação pode utilizar as funcionalidades da MonALISA. Outro ponto a ser ressaltado é que no serviço de publicação e descoberta do JINI não existe a possibilidade de utilizar múltiplos serviços de publicação e descoberta. O uso de múltiplos serviços de publicação e descoberta possibilita maior disponibilidade do serviço, pois existem mais de uma opção. Outra vantagem é que serviços de publicação e descoberta em domínios diferentes podem interagir e trocar informações, possibilitando a localização de serviços em domínios diferentes.

Por outro lado, os desenvolvedores do perfSONAR ao decidir qual a melhor opção para a publicação e descoberta para sua infra-estrutura, optaram por desenvolver um serviço próprio. Nos seus estudos outra tecnologia poderia ser usada, o UDDI, mas o receio de que futuramente o UDDI não suportasse novas funcionalidades e características, fez com que a sua escolha não fosse recomendada. Isso aconteceu mesmo existindo implementações do padrão UDDI de código aberto que possibilitam modificações.

Assim, o serviço de publicação e descoberta do perfSONAR é chamado de *Lookup Service XML (LS-XML)*. O LS-XML implementa as funcionalidades básicas de um serviço de publicação e descoberta, o que pode ser considerado como uma nova implementação de um Serviço de Publicação e Descoberta. Sendo que o LS-XML não implementa a verificação das informações, o controle de inserção e o uso de índices, que são características implícitas nas base de dados relacionais. Ou seja, várias funcionalidades que a UDDI provê, não são implementadas no LS-XML, o que lhe deixa em desvantagem em relação ao UDDI. Outro fator que é importante relatar no LS-XML é o fato de usar uma base de dados XML. A base de dados XML por não possuir mecanismos de indexação, relacionamento e outros pode obter um desempenho baixo nas consultas de serviços. No entanto, o uso de base de dados XML torna o armazenamento de dados flexível, pois com a utilização de *Schemas XML* é possível criar variações no formato dos dados a serem armazenados. Por exemplo, no caso do LS-XML do perfSONAR existe a possibilidade de criar estruturas com o uso de vários objetos *Metadata* e *Data*.

Estas considerações reforçam o objetivo principal deste trabalho que é o de propor e desenvolver um Serviço de Publicação e Descoberta para a infra-estrutura piPEs-BR/GFD utilizando uma implementação da tecnologia UDDI. Pois é possível ver que as soluções existentes de publicação e descoberta para as infra-estruturas de monitoramento apresentam algumas desvantagens em relação a uma solução usando o padrão UDDI, como o uso de uma base de dados XML que possui baixo desempenho, a não verificação dos dados contidos nas requisições e o difícil acesso. Sendo que o UDDI é um padrão para Serviços Web para esse tipo de finalidade, e que no seu desenvolvimento já estão incluídas inúmeras funcionalidades envolvidas na publicação e descoberta de Serviços Web.

É preciso salientar também que para existir a interação do serviço de publicação e descoberta proposto com outras infra-estruturas de monitoramento, é desejável que o mesmo possua uma interface padrão para que a comunicação entre as infra-estruturas possa ocorrer independente da tecnologia utilizada. Assim, essa interface de comunicação deve seguir o padrão proposto pelo perfSONAR, o qual define que a comunicação feita entre os componentes é realizada através de mensagens XML definidas por um esquema padrão que, neste caso, é o NMWG. Logo, este trabalho também tem como objetivo fazer com que o serviço de publicação e descoberta proposto utilize essa interface padrão com troca de mensagens NMWG.

4 PUBLICAÇÃO E DESCOBERTA DE SERVIÇOS DE MONITORAMENTO

Este capítulo aborda questões que envolvem a Publicação e Descoberta de Serviços no âmbito de medições em redes de computadores, pois existem características pertinentes à área de medições como: nomenclaturas existentes, tipo de serviços, forma de acesso e etc., que devem ser levadas em consideração e que diferem de uma simples Publicação e Descoberta de Serviços. Essas questões serão usadas no protótipo desenvolvido e são de extrema importância para o entendimento do mesmo. Sendo que como o protótipo desenvolvido utiliza o padrão UDDI para Publicação e Descoberta de Serviços Web, algumas das questões abordadas abrangem também aspectos relacionados ao UDDI.

4.1 INTRODUÇÃO

Como visto nos capítulos anteriores, o Serviço de Publicação e Descoberta é um componente importante para as infra-estruturas de monitoramento. Mas, para disponibilizar um Serviço de Publicação e Descoberta eficiente, disponível, confiável e que possa ser utilizado em vários domínios diferentes, é necessário ir além de simplesmente implantar uma das tecnologias de Publicação e Descoberta existentes. Muitas “variáveis” envolvem os Serviços Web de monitoramento e essas variáveis devem ser levadas em consideração na definição, desenvolvimento e implantação de um Serviço de Publicação e Descoberta. Para isso, é necessário pesquisar as características que envolvem os Serviços Web de monitoramento, verificar as necessidades dos usuários dos Serviços Web de monitoramento em relação à descoberta desses serviços, determinar as tecnologias a serem usadas e definir o Serviço de Publicação e Descoberta de acordo com os tipos de usuário.

Uma questão principal que envolve os Serviços Web de monitoramento é sua classificação. A classificação facilita a identificação do serviço e ajuda a distinguir um único serviço ou um grupo de serviços com características semelhantes ou que por algum motivo estão relacionados. Essa classificação é de extrema importância para a Publicação e Descoberta, pois através dessa classificação é possível criar uma estrutura de informações que facilite a

identificação dos serviços e, conseqüentemente, a descoberta dos mesmos. Como exemplo, é possível citar o UDDI que possibilita usar informações características dos serviços para criar taxonomias que são usadas como índice nas descobertas dos serviços.

Já no cenário dos Serviços Web de monitoramento, o *Network Measurement Working Group* (NMWG), definiu um padrão que propõe uma nomenclatura entre as características de medições em redes de computadores (LOWEKAMP, 2004). A motivação para a criação dessa nomenclatura foi facilitar a comunicação entre componentes. Esse tipo de trabalho facilita a identificação dos serviços de monitoramento, pois alguns serviços possuem como característica principal uma dessas informações contidas na hierarquia. Um exemplo que pode ser citado é um serviço que fornece dados de medição unidirecional (*oneway*), logo a característica *oneway* que está contida na hierarquia pode ser usada como um índice na pesquisa em um Serviço de Publicação e Descoberta.

Outro ponto que envolve a publicação e descoberta de serviços de monitoramento é quais tecnologias podem ser usadas para o seu desenvolvimento e implantação. Em resumo, este trabalho usou as tecnologias consideradas padrões de Serviços Web, seja para disponibilizar serviços, desenvolver clientes e publicar e descobrir esses serviços. Ao longo deste trabalho é possível verificar justificativas implícitas para o uso destas tecnologias e ao longo deste capítulo também serão abordadas algumas das vantagens de usar essas tecnologias.

4.2 CARACTERÍSTICAS DOS SERVIÇOS DE MONITORAMENTO

Além das características definidas pelo NMWG, existem outras informações que categorizam o Serviço Web de monitoramento. E mais especificamente os Serviços Web da infra-estrutura piPEs-BR/GFD. Nesta seção serão descritas as características dos Serviços Web de monitoramento e de que forma essas características podem ser utilizadas na publicação e descoberta.

4.2.1 Classificação por tipo de serviço

Foi verificado que nos serviços disponibilizados pelos piPEs-BR/GFD existe em comum uma classificação lógica que identifica o serviço. Essa classificação pode ser considerada como o “tipo” do serviço. No Quadro 4.1 é apresentada a lista desses tipos e seus respectivos serviços. Logo, essas informações foram usadas para categorizar os Serviços Web do piPEs-BR/GFD no Serviço de Publicação e Descoberta proposto neste trabalho.

Tipos de serviços do piPEs-BR/GFD e perfSONAR		
Tipo	Sigla	Descrição
<i>Measurement Point</i>	MP	Ponto de medição
<i>Measurement Archive</i>	MA	Módulo de armazenamento de dados
<i>Round Robin Database</i>	RRD	Tipo de base de dados que usa o conceito
<i>Structured Query Language</i>	SQL	Linguagem de pesquisa declarativa para banco de dados relacional
<i>SNMP Measurement Point</i>	SNMP MP	Disponibiliza Serviços Web para acessar dispositivos que suportam SNMP
<i>Alcatel NMS Measurement Point</i>	NMS MP	Recupera dados medidos pelo Alcatel's Network Management System
<i>Command Line Measurement Point</i>	CL MP	Ponto de medição que executa ferramentas via linha de comando
<i>Telnet/SSH Measurement Point</i>	TELNET MP	Usado para conectar de forma segura em dispositivos para recuperar informações
<i>Ping Measurement Point</i>	PING MP	Ponto de medição que usa a ferramenta ping
<i>BWCTL Measurement Point</i>	BWCTL MP	Ponto de medição que utiliza a ferramenta BWCTL
<i>E2EMon Measurement Point</i>	E2EMon MP	MP que coleta dados do <i>E2E Monitoring System</i>
<i>Command Measurement Point</i>	C MP	Usado para executar qualquer tipo de comando em dispositivos remotos
<i>Hades Measurement Point</i>	HADES MP	Disponibiliza os dados de medições do Hades (IPPM).
<i>G3 Measurement Archive</i>	G3 MA	Semelhante ao RRD MA

Quadro 4.1 - Informações de tipos de serviços do piPEs-BR/GFD

Nota: elaboração do autor (2007).

A idéia é construir uma hierarquia com as informações de tipo. Onde o nível dessa hierarquia depende da especificação da informação. Um exemplo a ser citado é o serviço *Measurement Point*, que como informação raiz possui o tipo “MP” que é a abreviação de seu tipo propriamente dito. Como existem várias implementações do MP, para várias ferramentas de medição e métricas diferentes, é possível utilizar mais níveis na hierarquia. Assim, o próximo nível do MP nessa hierarquia se refere basicamente à tecnologia do MP, por exemplo, o

Command Line Measurement Point (CLMP) que é o MP que executa ferramentas de medição via linha de comando. E sendo ainda mais específico na classificação do MP, é possível usar mais um nível de identificação, como a ferramenta utilizada nesse MP, que no exemplo do CLMP poderia ser identificado pela ferramenta OWAMP. Na Figura 4.1 é possível ver um exemplo dessa classificação.

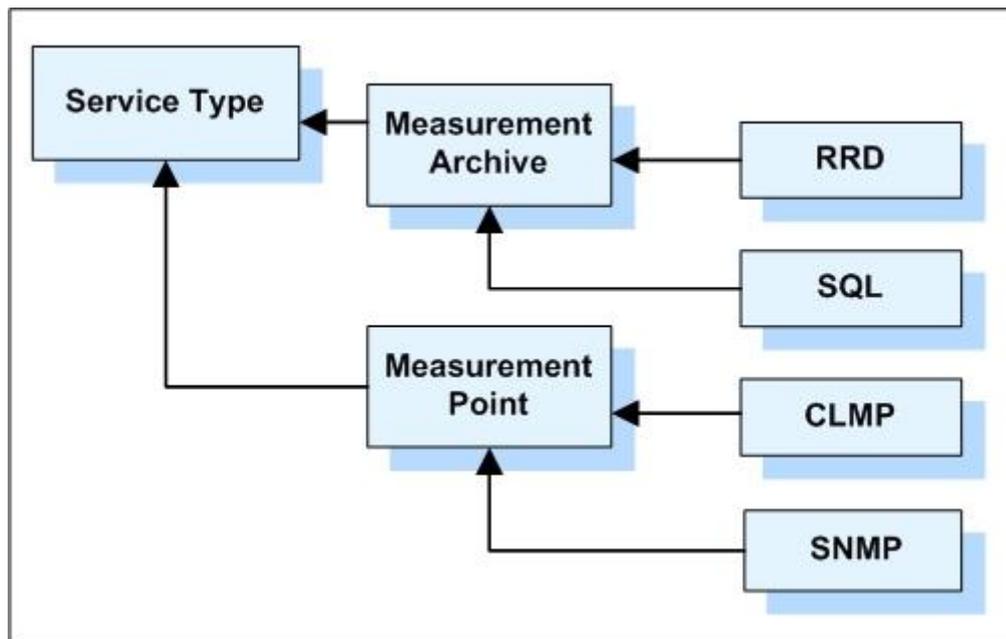


Figura 4.1 - Classificação dos serviços pelo tipo
Nota: elaborado pelo autor (2007).

4.2.2 Usando a hierarquia das características de medições em redes

O *Network Measurement Working Group* (NMWG) propôs um conjunto padronizado de características de redes e a classificação hierárquica dessas características. A motivação do NMWG para realizar este trabalho foi a necessidade de interação entre vários sistemas de Grid que manipulam informações de medição e que ao trocarem essas informações precisavam que as mesmas fossem identificadas. A padronização dessas características também facilita a criação de um esquema para a descrição dos dados de monitoramento.

Portanto, o NMWG identificou vários tipos de características que envolvem medições em redes, definiu uma nomenclatura e estabeleceu uma hierarquia entre essas informações. O resultado desse trabalho pode ser visto na Figura 4.2, onde estão representadas as características identificadas e a hierarquia dessas características. É possível observar na figura que essas

características abrangem vários aspectos dos dispositivos de redes, desde as métricas que influem sobre os dispositivos, até parâmetros de confiabilidade como o período médio entre falhas (do inglês *Mean time between failures* - MTBF²²) dos dispositivos ou enlaces.

Para a nomenclatura ficou definido que essas características devem ser escritas sempre com letras minúsculas, exceto quando o nome da característica possuir mais de uma palavra. As nomenclaturas que englobam as características da Figura 4.2 podem ser vistas no Quadro 4.2.

path.delay.roundTrip	path.delay.oneWay
path.loss.oneWay	path.packetReordering
path.loss.roundTrip	hop.packetReordering
path.bandwidth.Utilized	node.queue.length
hostToHostPath.bandwidth.achievable	router.queue.discipline
hostToHostPath.hoplist	switch.forwarding.forwardingTable
router.bandwidth.utilized	router.bandwidth.capacity
hop.delay.oneWay.jitter	autonomousSystem.delay.oneWay

Quadro 4.2 - Nomenclatura padrão de várias combinações de características

Nota: elaboração do autor (2007).

Essa padronização, através das características de redes, faz com que componentes possam interagir utilizando a mesma “linguagem”. Um exemplo bastante demonstrativo que pode ser citado é o serviço de MP que pode fornecer informações de atraso (*delay*). Logo, um cliente que deseja requisitar a um MP que realize testes de atraso unidirecional pode simplesmente na requisição enviar uma mensagem contendo a solicitação de dados do tipo *path.delay.oneWay*.

²² Parâmetro utilizado para descrever a confiabilidade.

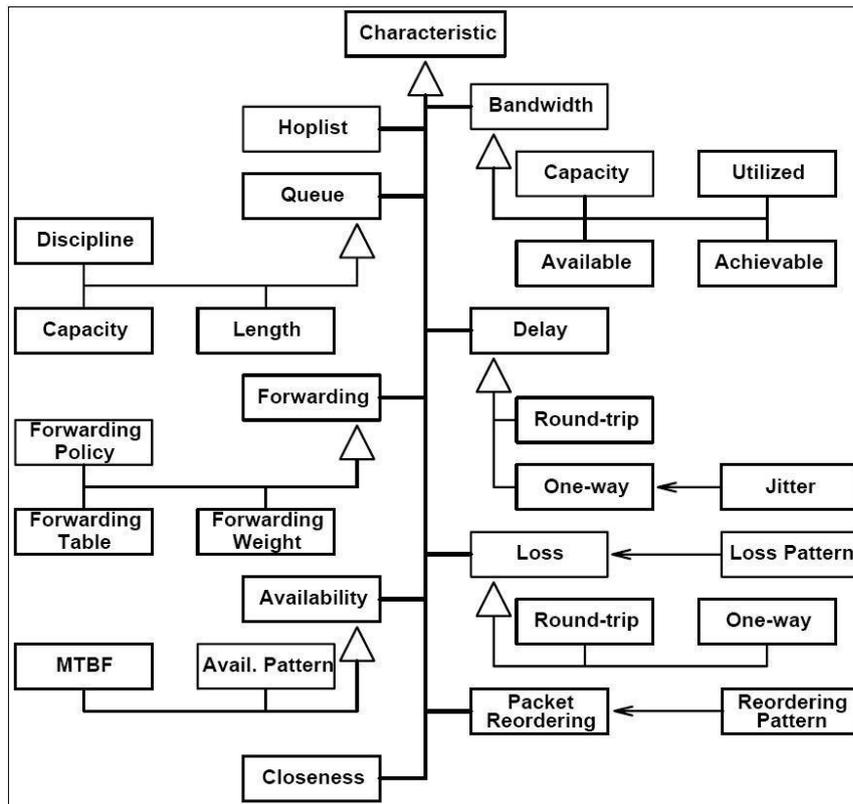


Figura 4.2 - Características de rede usadas para descrever o comportamento dos dispositivos de redes
Fonte: LOWEKAMP (2004).

Ao deparar com essa hierarquia e nomenclatura, fica claro que é possível usar algumas dessas características para identificar os Serviços Web de monitoramento. Logo, foi decidido usar essas características no Serviço de Publicação e Descoberta proposto como mais uma opção para identificar os Serviços Web e assim facilitar a descoberta de serviços.

É possível também usar o trabalho realizado pelo NMWG para criar taxonomias que categorizem um grupo de serviços, como os serviços do tipo *delay*, que é um conjunto de serviços que fornecem também o atraso do tipo *Round Trip*, *One Way* e *Jitter*. Assim se um usuário deseja utilizar Serviços Web que fornecem todos os tipos de *delay*, a única característica a ser buscada no Serviço de Publicação e Descoberta é o próprio *delay*. Mais informações sobre as taxonomias criadas para o serviço proposto neste trabalho poderá ser visto nas seções seguintes.

As características citadas anteriormente também podem ser usadas como índices para identificar um ou mais serviços. Assim, é possível identificar um grupo de serviços que, por exemplo, pode estar relacionado com uma ferramenta de medição específica ou relacionado a uma métrica.

4.3 PADRÕES E TECNOLOGIAS A SEREM UTILIZADOS

Para disponibilizar um Serviço de Publicação e Descoberta é preciso verificar se as soluções disponíveis (padrões, tecnologias, etc.) podem contemplar os requisitos desejados de acordo com o escopo em que o serviço será disponibilizado. Por exemplo, a tecnologia ebXML (vide capítulo de Arquitetura Orientada a Serviços – SOA) não se aplica neste trabalho devido ao fato de ser direcionada somente ao comércio eletrônico. Portanto este trabalho, como já relatado anteriormente, usa como base o padrão Web para Publicação e Descoberta, o UDDI (vide capítulo de Arquitetura Orientada a Serviços – SOA). As motivações para a escolha do UDDI podem ser enumeradas da seguinte forma:

1. O UDDI é a tecnologia padrão para publicação e descoberta de serviços;
2. O UDDI define um Serviço de Publicação e Descoberta para Serviços Web de forma genérica. Ou seja, o UDDI não especifica o cenário de utilização, podendo ser utilizado para publicar e descobrir qualquer tipo de Serviço Web em qualquer contexto;
3. Na literatura, UDDI é “sinônimo” de publicação e descoberta. Muitos trabalhos usam o UDDI em detrimento de outras soluções.
4. Existem muitos trabalhos envolvendo o UDDI em diferentes escopos (ex. (PINTO e outros, 2003), (ZHOU e outros, 2003) e (KAWAMURA e outros, 2004);
5. O uso do UDDI no escopo de medições é uma iniciativa inovadora;
6. Um estudo prévio sobre o UDDI mostrou que o mesmo pode ser utilizado no cenário de medições, em específico no piPEs-BR/GFD (MONTEIRO e outros, 2007).

Outras tecnologias e padrões são usados na proposta deste trabalho. Dentre elas é possível citar os Serviços Web que irão realizar interface com o exterior, JAVA que será utilizada como linguagem de desenvolvimento e as padronizações definidas pelo NMWG em relação a mensagens de comunicação e classificação hierárquica das características de medições.

4.4 MODELO DE DADOS

Para utilizar o UDDI no cenário de medições em redes de computadores, é necessário definir quais componentes do UDDI serão usados e quais informações sobre os Serviços Web de medição esses componentes irão armazenar. Alguns desses componentes possuem o tipo de informação a ser armazenada predefinida, por exemplo, o componente *Binding Template* (vide seção Componentes do UDDI) que armazena o Ponto de Acesso do serviço. Logo, as informações que são comuns a qualquer Serviço Web serão armazenadas nos locais designados pelos componentes do UDDI e outras informações específicas dos serviços de medições devem ser distribuídas nos componentes de forma a possibilitar o entendimento e a descoberta dos serviços. Neste trabalho estamos chamando essa atividade de modelagem de dados, já que esse tipo de ação é semelhante à modelagem de dados de bases de dados.

Basicamente algumas informações podem ser consideradas padrão, ou seja, comum para qualquer serviço, são elas: nome, ponto de acesso, descrição, parâmetros e etc. Mas existem informações específicas que diferem de serviço para serviço e que devem ser distribuídas nos componentes do UDDI. O componente *Category Bag* do UDDI foi concebido para armazenar essas informações específicas. O próprio nome do componente explica o seu uso, pois as informações específicas de um serviço geralmente o categorizam. No Quadro 4.3 estão listados os componentes que serão usados no Serviço de Publicação e Descoberta do piPEs-BR/GFD. Nela é possível observar também as informações que serão armazenadas e exemplos dessa informação.

Componentes	Campos	Descrição	Exemplo
<i>Business Entity</i>	Nome	Nome da entidade proprietária dos serviços	UNIFACS, RNP
	Operador	Usuário responsável pela gerência dos serviços	Herbert
	URL	URL onde o serviço está disponível	uddi.unifacs.br
	Descrição	Descrição do registro UDDI	UDDI responsável por publicar e descobrir os serviços web da UNIFACS
<i>Business Service</i>	Nome	Armazena o nome do serviço do piPEs-BR/GFD e pode ser usado como identificador	MA UNIFACS, MP UFSC
	Descrição	Descrição sobre o serviço	Serviço de armazenamento de dados de utilização

Componentes	Campos	Descrição	Exemplo
<i>Binding Template</i>	Ponto de acesso	URL do ponto de acesso ao serviço. No piPEs-BR/GFD essa é a principal informação de acesso	http://mu.dante.org.uk:8080/axis/services
	Descrição	Descrição técnica do serviço	Esse serviço funciona através de troca de mensagens NMWG.
<i>Category Bag</i>	Indefinido	Esse componente possui o número de informações armazenadas variável. A quantidade de informação depende do tipo do serviço. Aqui são armazenadas informações adicionais	No caso de um Serviço do tipo MA-RRD as informações são: IP das interfaces dos roteadores, capacidade da interface, o tipo da interface e o nome da interface.
<i>tModel</i>	Nome	Como o <i>tModel</i> é um índice seu nome é usado como chave para identificar um serviço. Foram criados <i>tModels</i> de acordo com as características dos serviços	MA, MP, CLMP, SQL, RRD, PATH, DELAY, ONEWAY, ROUTER

Quadro 4.3 - Componentes do UDDI usados no serviço de publicação e descoberta do piPEs-BR/GFD
Nota: elaboração do autor (2007).

A Figura 4.3 melhora o entendimento sobre os componentes do UDDI e os seus relacionamentos, pois mostra a modelagem usada na proposta deste trabalho. Nesta modelagem é possível ver a entidade proprietária de serviços (*Business Entity*) a qual pode possuir um ou mais Serviços Web em seu registro. Já um Serviço Web pode ser relacionado com um ou mais componentes do tipo *tModel* possibilitando assim o uso de identificadores (*tModel*) para categorizar o serviço. O Serviço Web propriamente dito (*Business Service*) possui apenas um componente do tipo *Binding Template*, pois as informações de acesso a um serviço no caso do piPEs-BR/GFD são únicas. Já o *Binding Template* pode possuir um ou mais *Category Bags* o que permite que um serviço também possa ser categorizado por mais de um conjunto de informações específicas.

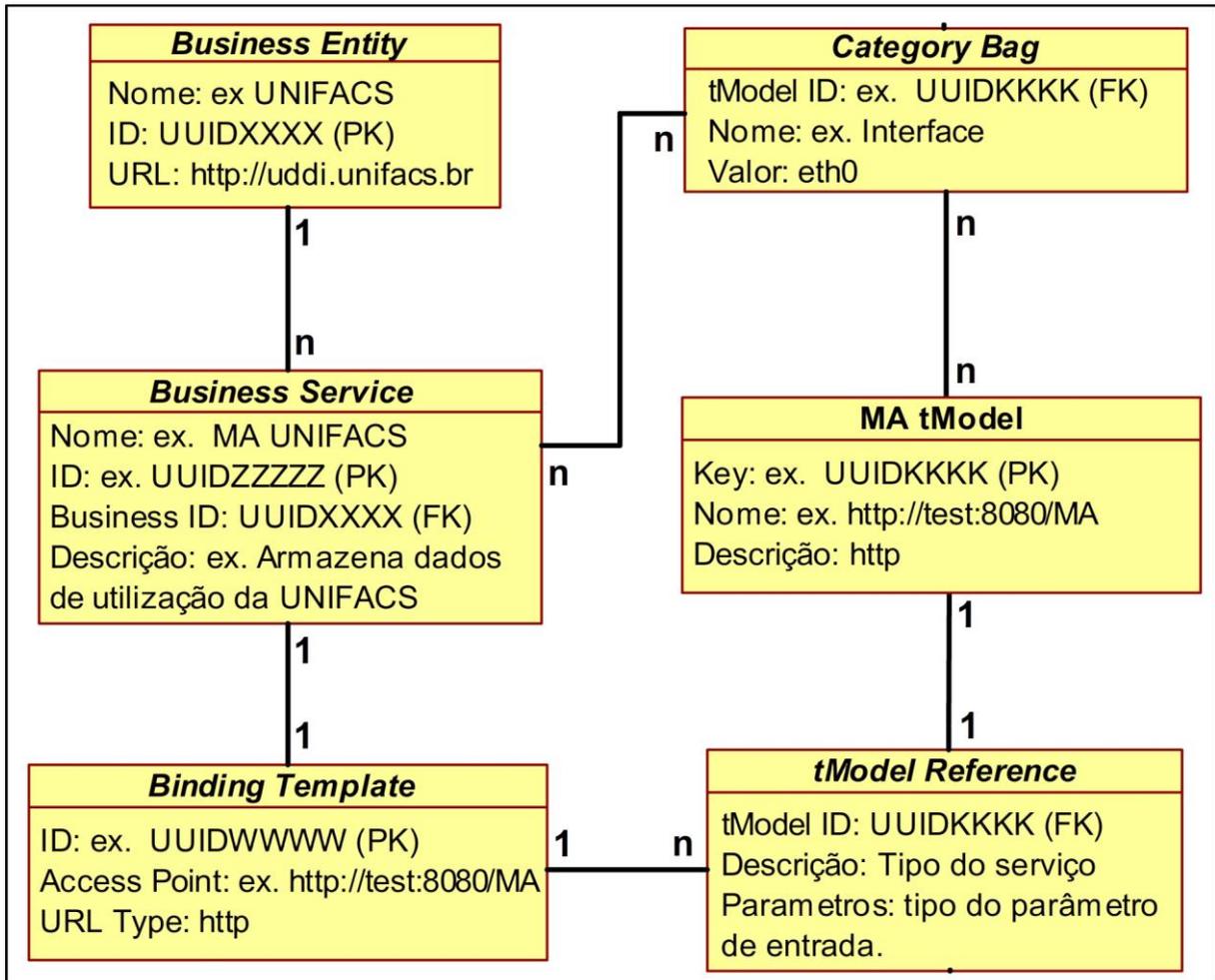


Figura 4.3 - Modelo de dados
Nota: elaborado pelo autor (2007).

4.5 TAXONOMIAS NOS SERVIÇOS DE MONITORAMENTO

Para Apte e Mehta (APT; MEHTA, 2002) taxonomia é um sistema de categorização. A qual denota uma abordagem formal usada para categorizar qualquer conjunto de entidades. Alguns exemplos de taxonomia são:

- a) sistema Norte Americano de Classificação Industrial (do inglês *North American Industry Classification System - NAICS*);
- b) padrão de classificação universal para serviços e produtos (do inglês *Universal Standard Products and Service Classification - UNSPSC*);

- c) classificação com base em características geográficas (do inglês *Geographical - Geo*²³).

As taxonomias citadas são normatizadas, ou seja, foram concebidas em forma de normas internacionais. Elas são comumente usadas para classificar serviços disponibilizados por indústrias, pois foram desenvolvidas especialmente para identificar produtos que na maioria das vezes são fabricados ou vendidos por essas indústrias. Nos serviços de monitoramento é possível usar algumas das taxonomias normatizadas como a *Geographical* que é normatizada na ISO 3166 e identifica geograficamente onde o serviço está localizado. Assim instituições de localizações diferentes podem realizar medições e localizar serviços situados em localizações diferentes. Essa abordagem pode ser uma analogia ao uso de medições e múltiplos domínios. Assim, num ambiente mais refinado seria possível utilizar a taxonomia geográfica junto com uma taxonomia própria que categoriza o Serviço Web de acordo com o domínio a que o mesmo faz parte.

As taxonomias no UDDI são representadas por um conjunto de objetos *tModel* que formam uma hierarquia de informações, onde um *tModel* pode estar relacionado com outros possibilitando assim localizar informações em qualquer nível da hierarquia. Na Figura 4.4 é possível visualizar um exemplo simples de uma taxonomia rerepresentadas por *tModels*. É possível observar também na figura as informações contidas nesses *tModels*. Assim, se um usuário deseja localizar todos os serviços pertencentes ao domínio do NUPERC, é só localizar todos os serviços que fazem referência a esse *tModel*. É possível ainda saber quais outros serviços estão relacionados com os serviços do domínio do NUPERC, pois ao localizar o *tModel* NUPERC será verificado que o mesmo pertence (ou faz referência) ao *tModel* RNP. Esse tipo de abordagem facilita, por exemplo, a medição em múltiplos domínios, pois em uma medição fim-a-fim que passa por mais de um domínio, é possível localizar os serviços envolvidos nesse caminho, solicitando assim a cada domínio as informações necessárias para compor a medição fim-a-fim.

²³ Geralmente são usadas as coordenadas geográficas ou a norma ISO 3166, vide subseção de Categorização de serviços.

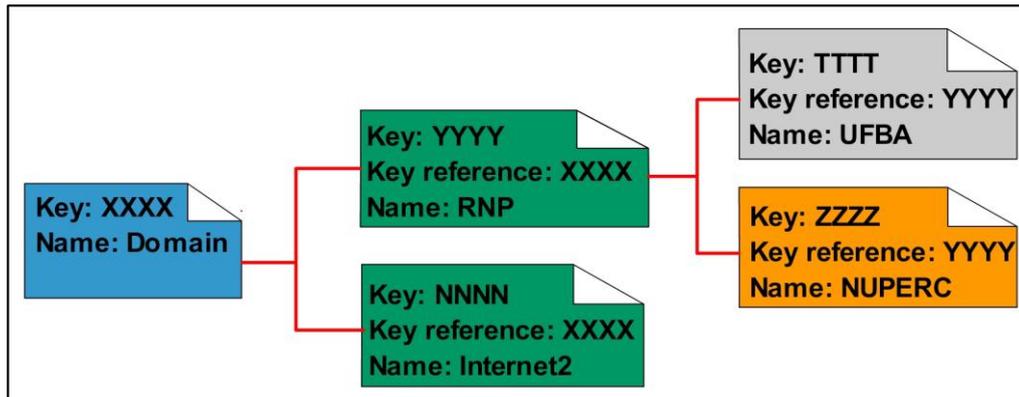


Figura 4.4 - Exemplo de relacionamento de taxonomias
Nota: elaborado pelo autor (2007).

Para o Serviço de Publicação e Descoberta proposto, foram criadas taxonomias privadas baseadas nas informações sobre os Serviços Web. Assim foram criadas duas taxonomias, uma com base na classificação por tipo (seção 4.2.1) e outra com base na hierarquia das características de medições em redes de computadores do NMWG (seção 4.2.2). Ao consultar um serviço, o usuário pode utilizar as informações da classificação do NMWG ou as características de tipo dos serviços. Por exemplo, é possível localizar um ou mais serviços que fornecem a perda de pacotes em um caminho buscando pelos serviços que fazem referência aos *tModels path*, *loss* e *oneWay*. Na Figura 4.5 é possível visualizar as taxonomias criadas e as informações que as mesmas contêm e o relacionamento entre as informações.

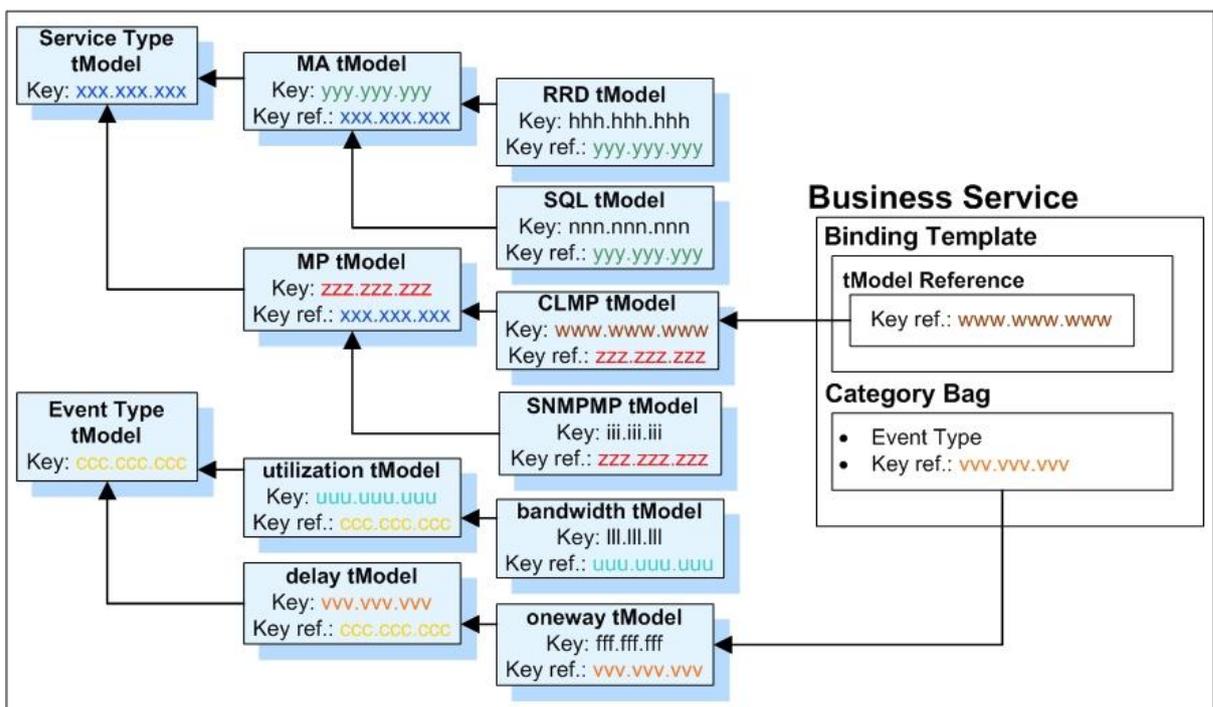


Figura 4.5 - Exemplo das taxonomias criadas
Nota: elaboração do autor (2007).

4.6 SERVIÇO DE PUBLICAÇÃO E DESCOBERTA PROPOSTO

As informações apresentadas nas seções anteriores possibilitam realizar a definição de um Serviço de Publicação e Descoberta para infra-estruturas de monitoramento. Essas informações também possibilitam observar que o serviço proposto irá utilizar o padrão para a Publicação e Descoberta de Serviços Web (UDDI). O serviço proposto irá disponibilizar duas taxonomias para ajudar na consulta de serviços e distribuirá nos componentes do UDDI as informações necessárias para acessar e identificar os Serviços Web da infra-estrutura de monitoramento piPEs-BR/GFD.

Contudo, outros pontos devem ser relatados, como os requisitos do piPEs-BR/GFD, a definição clara das funcionalidades do serviço, o *design* interno do serviço e a interface de acesso ao serviço. Alguns desses pontos também estão citados na introdução deste trabalho, pois os mesmos fazem parte das motivações, justificativas e objetivos.

4.6.1 Requisitos do piPEs-BR/GFD

Com a experiência no desenvolvimento da infra-estrutura de monitoramento piPEs-BR e a adaptação do piPEs-BR para seguir as definições do documento GFD, alguns requisitos foram demandados para a definição do Serviço de Publicação e Descoberta do piPEs-BR/GFD. Esses requisitos vão desde os requisitos básicos como a própria publicação e descoberta até requisitos como: facilidade de acesso, bom desempenho, interfaces amigáveis, interoperabilidade com outras infra-estruturas e utilização em domínios administrativos diferentes.

O principal requisito a ser atendido pelo Serviço de Publicação e Descoberta é o que se pode chamar de requisitos básicos que são: registrar, consultar, remover e alterar os Serviços Web disponibilizados pela infra-estrutura piPEs-BR/GFD. Para atender essas demandas pode ser utilizado o padrão UDDI, pois o padrão define essas funcionalidades e tantas outras mais.

Um problema observado no uso do padrão UDDI é o fato de os usuários terem que possuir um conhecimento interno da tecnologia para poder utilizar suas funcionalidades. Para solucionar esse problema é preciso elaborar uma biblioteca de acesso ao UDDI que facilite para os usuários o acesso ao mesmo. O UDDI pelo fato de ter sua nomenclatura própria de seus componentes como: *bindingTemplate*, *serviceDetail*, *InstanceDetails*, *InstanceParms*, *TModelInstanceDetails*, *BusinessService* e *BusinessList*, pode “assustar” os usuários dos serviços de medição. Os usuários de medições estão acostumados com nomes de ferramentas, métricas e técnicas de medição como: *Access point*, *network characteristic*, *event type*, *interface*, *parameters*, *delay*, *oneway* e etc. Contudo, é preciso elaborar uma biblioteca que abstraia os nomes dos componentes do UDDI e apresente para os usuários termos já conhecidos.

Outra funcionalidade dessa biblioteca é a criação de mecanismos de consultas que também vão abstrair os métodos de consultas disponibilizados pelas tecnologias que implementam o UDDI. Ou seja, serão criados diversos tipos de consultas para tentar atender todas as possibilidades de combinação de dados e utilização das taxonomias disponíveis. Com essa biblioteca o usuário será capaz de registrar serviços, consultar serviços e remover serviços registrados de uma forma mais amigável.

Outro requisito do piPes-BR/GFD é de um bom desempenho em relação ao tempo de resposta pelo Serviço de Publicação e Descoberta, pois os principais usuários do serviço de Publicação e Descoberta serão clientes de visualização de dados e esse tipo de aplicação geralmente é usada pelos usuários das redes. Logo, um cliente de visualização que interage com a infraestrutura piPEs-BR-GFD além dos acessos ao Serviço de Publicação e Descoberta irá utilizar outros serviços da infra-estrutura. Contudo, se o serviço possuir um baixo desempenho poderá ocasionar um descontentamento por parte do usuário.

Finalizando os requisitos do piPEs-BR/GFD é possível citar a operação em domínios diferentes, ou seja os Serviços de Publicação e Descoberta implantados em múltiplos domínios devem interagir uns com os outros, possibilitando a localização de serviços em domínios diferentes e conseqüentemente possibilitando realizar medições num caminho fim-a-fim que passa por um ou mais domínios. O próprio fato de estar seguindo as definições do GFD, o qual transforma em módulos os componentes da infra-estrutura e transforma as funcionalidades da infra-estrutura de monitoramento em serviços, faz com que esse requisito

seja atendido. Outra recomendação para possibilitar o alcance em múltiplos domínios é também seguir as definições de comunicações adotadas pelo perfSONAR, a qual utiliza mensagens XML padronizadas pelo esquema definido pelo NMWG e que foi relatado na seção 3.2.3. O fato de seguir as definições do perfSONAR possibilita a criação de um novo Serviço de Publicação e Descoberta que pode ser usado tanto pelo perfSONAR quanto pelo piPEs-BR/GFD. Assim esse serviço é chamado de *Lookup Service UDDI Type*. Logo, as infra-estruturas de monitoramento podem interagir entre si trocando informações sobre os seus Serviços Web, possibilitando acessar serviços de medição disponíveis em múltiplos domínios administrativos (MONTEIRO e outros, 2007).

4.6.2 Funcionalidades

Para atender os requisitos citados anteriormente, o Serviço de Publicação e Descoberta deve possuir funcionalidades bem definidas. Essas funcionalidades são:

- a) **registro de serviços** - O serviço deve receber solicitações de registro de Serviços Web. Para prover essa funcionalidade é preciso: verificar se o serviço já existe, verificar se a solicitação contém a quantidade mínima de informação para registro, registrar o serviço e responder à solicitação com sucesso ou falha. O registro de serviço também pode ser considerado uma alteração, pois ao verificar que o serviço já existe, o mesmo é considerado uma alteração;
- b) **consulta de serviços** - O serviço deve receber solicitações de consultas de serviços. A consulta deve ser realizada por informações predefinidas que são utilizadas como identificadores dos serviços como: as taxonomias, nomes, ponto de acesso e etc.;
- c) **alteração de serviços** - A alteração do serviço é considerada um novo registro de serviço, ou seja, ao receber uma solicitação de alteração, será consultado se o serviço realmente existe. Caso o serviço exista, o registro antigo é removido e é feito um registro novo;
- d) **remoção de serviços** - O Serviço de publicação e descoberta deve ser capaz de receber uma solicitação de remoção, identificar o serviço a ser removido e remover o mesmo. Essa funcionalidade assim como a de consulta deve utilizar

índices para identificar o serviço a ser removido. Logo, se a remoção for apenas de um serviço, o identificador a ser utilizado deve ser único. Ao pretender remover um grupo de serviços é possível utilizar um identificador que rotule, por exemplo, um grupo de serviços. Para isso assim como na consulta serão usados os mecanismos de identificação como taxonomias e informações específicas;

- e) **disponibilidade dos serviços** - Para manter uma base atualizada de informações sobre serviços Web, é necessário verificar se as informações registradas pertencem a Serviços Web que estão funcionando. Assim, um usuário ao consultar informações sobre um Serviço Web terá a garantia de que pelo menos o serviço está disponível. Isso pode ser proporcionado mantendo um tempo de vida (do inglês *lifetime*) para os registros, obrigando a todos os Serviços Web se registrarem automaticamente de tempos em tempos. Ou seja, ao se registrar no Serviço de Publicação e descoberta, o Serviço Web recebe um tempo de vida. Assim, para que suas informações sejam mantidas no registro, o serviço Web deve realizar um novo registro antes que seu tempo de vida esgote. O serviço de publicação e descoberta, por sua vez, deve verificar periodicamente os tempos de vida esgotados. E ao verificar que um tempo de vida se esgotou, deve remover as informações referentes ao respectivo serviço Web;
- f) **verificação de status** - Os usuários do serviço de publicação e descoberta desejam saber se o serviço está funcionando a contento. Logo, o serviço de publicação e descoberta deve receber requisições de informações de status. O status do serviço de publicação e descoberta pode ser verificado de diversas maneiras. Numa versão inicial, o status será considerado a disponibilidade da base de dados e a disponibilidade do próprio serviço de publicação e descoberta em receber requisições e responder à mesma. Assim será feita uma pequena consulta de serviço e, ao verificar o sucesso, o status do serviço será considerado OK.

4.6.3 Projeto do Serviço de Publicação e Descoberta

Como já citado anteriormente, o Serviço de Publicação e Descoberta será composto por uma aplicação principal que é o serviço propriamente dito, duas interfaces de acesso, o padrão web UDDI e uma base de dados relacional. As interfaces de acesso serão formadas pela biblioteca de acesso ao serviço e a interface de Serviço Web no padrão perfSONAR. Na Figura 4.6 é possível ter uma visão geral da estrutura do serviço proposto de publicação e descoberta. Sendo que a interface de Serviços Web faz parte da aplicação principal e aciona os métodos da aplicação para executar a funcionalidade solicitada na requisição.

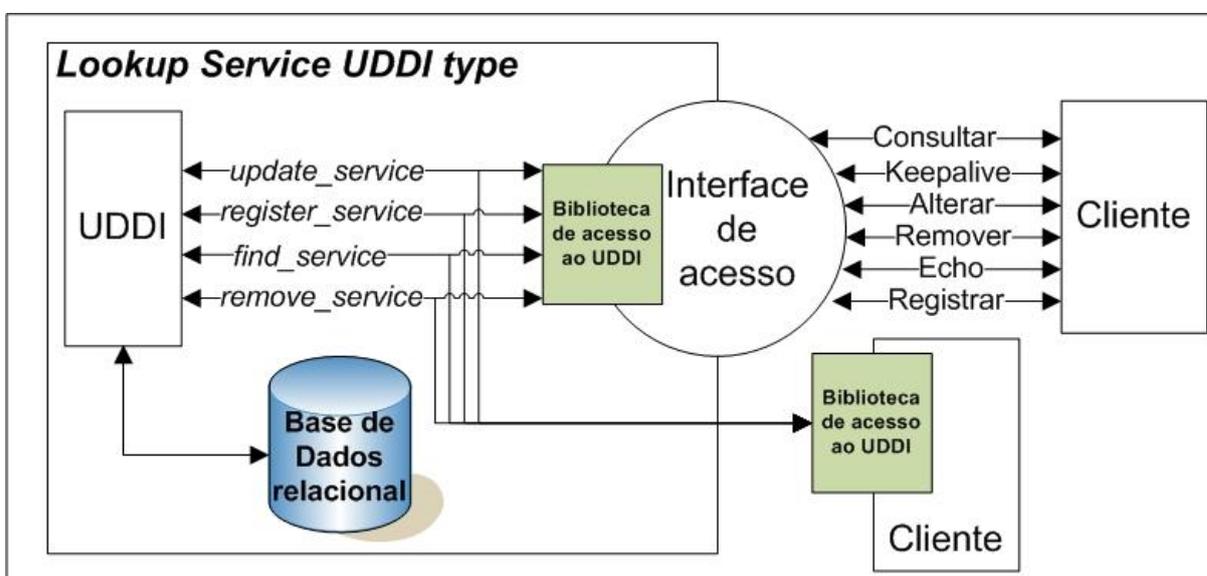


Figura 4.6 - Estrutura macro do Serviço de Publicação e Descoberta proposto
Nota: elaboração do autor (2007).

Por ser uma visão macro, na figura não é possível visualizar o processo que é realizado dentro do serviço de publicação e descoberta, ou seja, todo o processo que é feito antes de realizar as operações no UDDI. Esse processo consiste em: tratamento das mensagens, identificação da funcionalidade solicitada, criação de objeto, execução de funções, geração de *log*, manipulação de dados e construção de mensagens. Sendo que no próximo capítulo, serão apresentadas informações técnicas do desenvolvimento que detalham melhor esse processo. Outra questão que pode ser observada na figura é o acesso direto ao UDDI via biblioteca de acesso.

A biblioteca de acesso ao UDDI que também faz parte dos resultados deste trabalho tem a intenção de facilitar e melhorar o desempenho no acesso ao UDDI, pois o acesso via

biblioteca possibilita uma comunicação direta do UDDI, onde algumas operações que são feitas no serviço de publicação e descoberta passam a ser realizadas no usuário. Outra vantagem é a simplificação do acesso, pois a biblioteca também abstrai os detalhes técnicos do acesso via interface de Serviços Web. Esses detalhes técnicos são dentre outros, as mensagens XML e a padronização do NMWG.

4.6.4 Integração com outras infra-estruturas

Um dos objetivos deste trabalho e uma grande vantagem do serviço de publicação e descoberta proposto, é o fato de ele ser inter-operável com outras infra-estruturas de monitoramento, ou melhor, inter-operável com qualquer outra aplicação que deseje consultar informações sobre os serviços Web do piPEs-BR/GFD. Mas, em relação às infra-estruturas piPEs-BR/GFD e perfSONAR, a intenção é que os usuários das duas infra-estruturas usem o serviço proposto indiferentemente da tecnologia utilizada. Assim não importa para o usuário se o serviço de publicação e descoberta usa uma base de dados XML ou usa o padrão UDDI.

Essa inter-operação pode ser alcançada utilizando um padrão no acesso aos serviços, ou seja, na interface de acesso. Contudo, existe uma padronização que define mensagens de requisição e resposta criadas pelos desenvolvedores do perfSONAR. Essa padronização utiliza o esquema criado pelo NMWG para facilitar a manipulação das informações de medições. Logo, um serviço bem definido tem que ser hábil de reconhecer as mensagens de requisição referente às suas funcionalidades. Para exemplificar utilizando as mensagens padrão do perfSONAR, o serviço de publicação e descoberta proposto poderá inter-operar com clientes tanto do piPEs-BR/GFD como do perfSONAR, pois as mensagens de requisição e respostas são as mesmas.

4.7 CONSIDERAÇÕES

O uso do padrão UDDI como base do Serviço de Publicação e Descoberta neste trabalho poupa a definição e desenvolvimento de muitas questões relacionadas. Como exemplo é

possível citar o controle na inserção dos dados, autenticação, controle de erros e etc. Mas, por outro lado, existe bastante trabalho a se fazer nas questões que envolvem quais informações sobre os serviços serão armazenadas, na distribuição das informações, na criação das taxonomias e etc. Logo, o trabalho é mais intelectual em relação à definição do que técnico em relação a escolhas de tecnologias e desenvolvimento.

Contudo, com as questões abordadas neste capítulo é possível visualizar as funcionalidades e características do serviço proposto. Para concluir o entendimento sobre o serviço podemos defini-lo como um Serviço de Publicação e Descoberta que utiliza o padrão UDDI e contará com duas formas de acesso. Sendo forma de acesso através de uma biblioteca que abstrai detalhes técnicos e outra via uma interface de Serviço Web. Onde a interface Web expõe as funcionalidades do serviço via troca de mensagens XML padronizadas pelo perfSONAR.

Mas, para que essa proposta tornasse realidade foi necessário desenvolver um protótipo que implementasse as definições contidas neste capítulo. Esse protótipo tem o intuito de validar a proposta deste trabalho e integra os componentes do piPEs-BR/GFD usado como protótipo na rede da RNP. Assim, no próximo capítulo serão relatados os detalhes técnicos contidos no desenvolvimento do protótipo do serviço de publicação e descoberta proposto neste trabalho.

5 PROTÓTIPO DESENVOLVIDO

Este trabalho após pesquisar assuntos que envolvem a publicação e descoberta dos Serviços Web de monitoramento, elaborou no capítulo anterior, um Serviço de Publicação e Descoberta para Serviços Web de monitoramento. Porém, para finalizar o estudo e disponibilizar uma pesquisa consistente, foi desenvolvido um protótipo com base nas definições estabelecidas. Neste capítulo serão apresentados os detalhes técnicos do protótipo desenvolvido, bem como a motivação para o seu desenvolvimento, as tecnologias utilizadas, testes realizados com o protótipo e as considerações sobre o que foi desenvolvido e avaliado.

5.1 INTRODUÇÃO

Como resultado da pesquisa sobre Publicação e Descoberta de Serviços Web de monitoração de redes, foi desenvolvido um protótipo denominado atualmente por *Lookup Service UDDI Type*. Esse protótipo tem como finalidade validar os estudos realizados e comprovar que é possível realizar a publicação e descoberta de Serviços Web para as infra-estruturas piPEs-BR/GFD e perfSONAR utilizando o padrão UDDI. Outro objetivo desse protótipo é integrar o conjunto de componentes desenvolvidos pelo GT-Medições da RNP da infra-estrutura piPEs-BR/GFD. Assim, o GT-Medições pode disponibilizar para a RNP uma versão inicial do piPEs-BR/GFD, para que possa ser utilizado como protótipo na sua rede.

Contudo, para desenvolver esse protótipo é preciso escolher as tecnologias a serem utilizadas, definir um modelo de dados básico e desenvolver a aplicação. Algumas tecnologias já foram abordadas neste trabalho, mas as tecnologias que serão utilizadas foram escolhidas pelo fato de atender os requisitos do serviço proposto. Assim, é possível citar as principais tecnologias utilizadas tais como: o JUDDI, UDDI4J, JAVA, *Apache Axis*, *Apache Tomcat* e outras. É bom lembrar também que o desenvolvimento seguirá as definições usadas para o desenvolvimento do perfSONAR. Ou seja, será utilizada a comunicação via mensagens padronizadas e a utilização de funcionalidades semelhantes, onde a complexidade do protótipo será mais acentuada no tratamento das mensagens e interação com o UDDI.

O protótipo desenvolvido é composto por duas implementações, uma é o serviço de publicação e descoberta propriamente dito e a outra é a biblioteca de acesso ao UDDI. Logo, o resultado são duas aplicações distintas, mas com as mesmas finalidades. Como linguagem de programação para desenvolvimento dessas duas aplicações foi escolhida JAVA, devido a existirem implementações como bibliotecas, que facilitam e dão suporte ao seu desenvolvimento. Sendo que para o acesso via interface de Serviços Web, é indiferente para o usuário qual linguagem de programação e plataforma sejam utilizadas.

A Figura 5.1 apresenta a arquitetura do serviço de publicação e descoberta em camadas, onde a primeira camada é a interface com o usuário que é feita via Serviços Web. Já a segunda camada são as funcionalidades do serviço, que no capítulo anterior foram bem definidas. Na figura é possível ver também que na camada de Ações, existe a possibilidade de se adicionar novas funcionalidades, ou seja, mais uma ação. A terceira camada é outro produto deste trabalho, a biblioteca de acesso ao UDDI desenvolvida em JAVA. É bom lembrar que uma aplicação pode utilizar diretamente a biblioteca de acesso ao UDDI e eliminar as duas primeiras camadas. Mas isso implica num acesso sem o padrão perfSONAR. As três camadas citadas anteriormente foram elaboradas e desenvolvidas neste trabalho e fazem parte do Serviço de Publicação e Descoberta proposto. Sendo que as camadas seguintes são tecnologias existentes que fazem parte e dão suporte ao serviço proposto. A quarta camada é a implementação do UDDI da *Apache Foundation*, o JUDDI e na quinta camada está à base de dados relacional proporcionando todas as funcionalidades de armazenamento de dados.

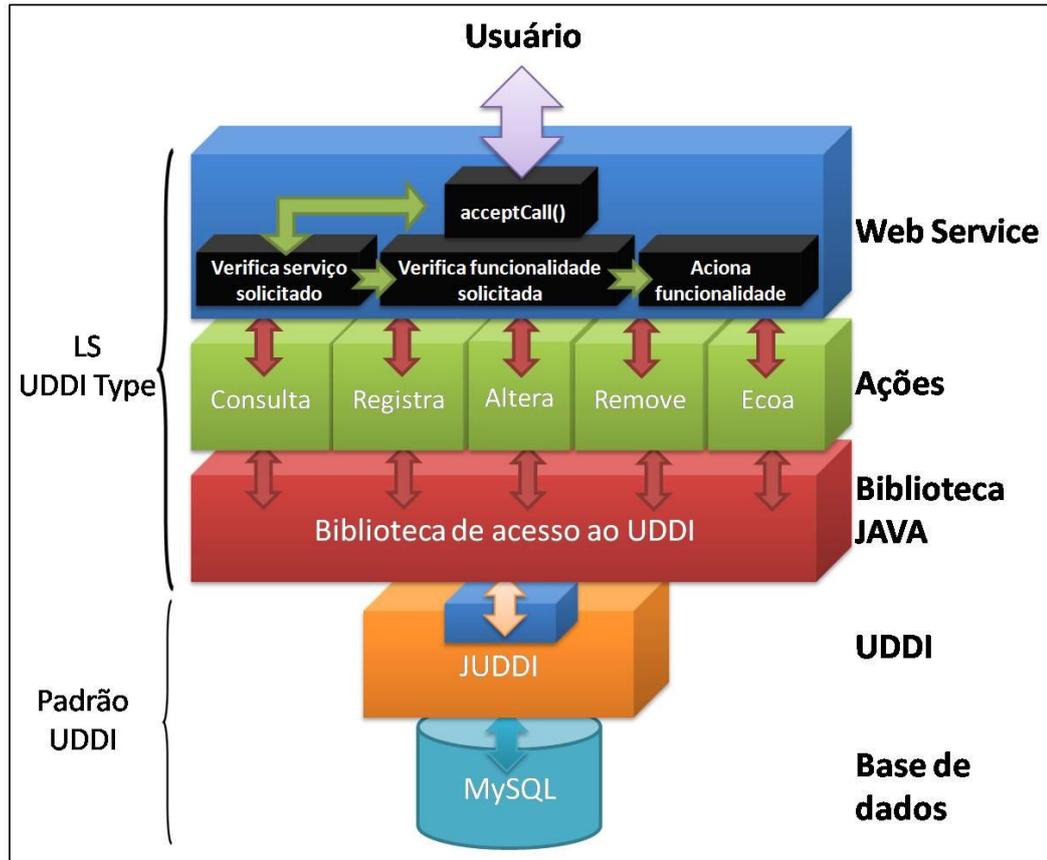


Figura 5.1 - Arquitetura do Serviço de Publicação e Descoberta
Nota: elaboração do autor (2007).

5.2 DESCRIÇÃO E DESENVOLVIMENTO DO PROTÓTIPO

Nesta seção é apresentada a descrição do desenvolvimento do protótipo proposto. Esse protótipo está dividido em duas implementações, uma é o serviço propriamente dito (*Lookup Service UDDI Type* - LS-UDDI) e a outra é a biblioteca de acesso ao UDDI. Essa seção também nomeia o Serviço de Publicação e Descoberta desenvolvido de *Lookup Service UDDI Type* (LS-UDDI), essa atitude é para significar que o Serviço de Publicação e Descoberta também pode ser utilizado na infra-estrutura perfSONAR, que nomeou o seu serviço de *Lookup Service XML Type* (LS-XML). Isso é possível, pois ambos os serviços seguem o mesmo padrão de comunicação e utilizam as mesmas mensagens de comunicação, além de possuir também as mesmas funcionalidades.

5.2.1 Lookup Service UDDI Type - LS-UDDI

O desenvolvimento seguindo o padrão utilizado no perfSONAR consiste na construção de um serviço genérico. A intenção de se desenvolver um serviço genérico é de criar um serviço que possa ser implementado e estendido para qualquer tipo de serviço específico como o Ponto de Medição, Serviço de Armazenamento e o próprio Serviço de Publicação e Descoberta. Assim, os desenvolvedores dos serviços específicos podem reutilizar o código, o que facilita e acelera o seu desenvolvimento.

Contudo, a estrutura de qualquer componente do perfSONAR ou piPEs-BR/GFD é semelhante, só diferindo nas ações realizadas. Na Figura 5.2 é possível ver o diagrama de componentes do serviço genérico definido no perfSONAR. A figura mostra seis componentes, sendo que quatro são considerados principais e dois como componentes de ajuda.

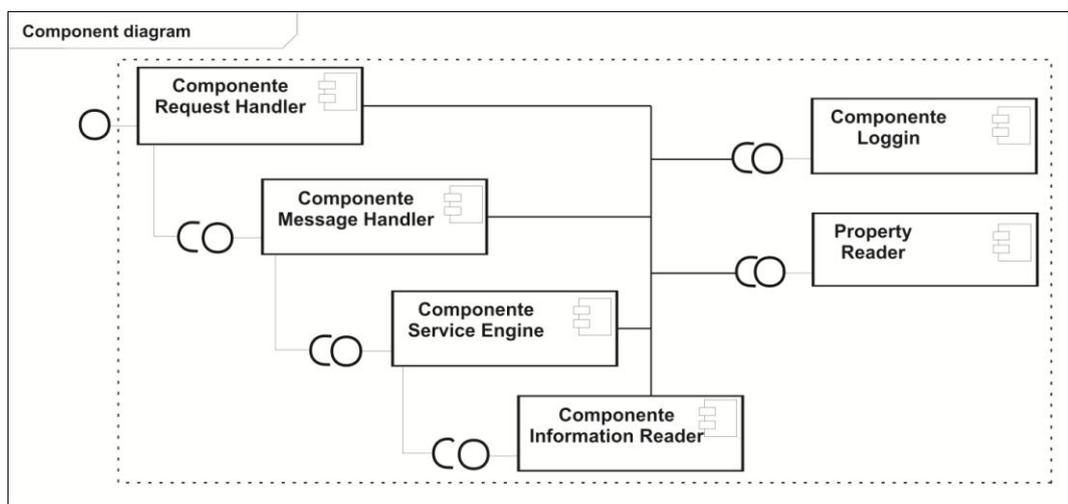


Figura 5.2 - Diagrama de componentes do padrão perfSONAR
Nota: elaboração do autor (2007).

Os componentes principais são:

- a) ***RequestHandler*** - É uma classe que implementa o serviço Web propriamente dito. Ela possui um método chamado *acceptCall()*, que recebe os Documentos XML (*org.w3c.DOM.Document*) que é enviado pelos usuários. Essa classe é responsável por acionar os componentes seguintes que realizam as operações do Serviço de Publicação e Descoberta. No final, esse componente retorna para

o usuário também um Documento XML (*org.w3c.DOM.Document*) com a resposta da requisição;

- b) ***MessageHandler*** - É um componente utilizado para construir um objeto com as características do tipo de serviço que está sendo solicitado na mensagem. Como já dito anteriormente, um serviço no perfSONAR é considerado de certa forma genérico, ou seja, ele é comum para qualquer módulo (ponto de medição, módulo de armazenamento, publicação e descoberta e etc.) do perfSONAR. Sendo que durante a seqüência lógica do software, é identificado que tipo de módulo a requisição está solicitando. Para ser mais claro, na mensagem de requisição existe um item que informa que serviço está sendo solicitado, que, por exemplo, podem ser publicação e descoberta, ponto de medição, armazenamento e etc. E o *MessageHandler* é responsável por criar um objeto (mensagem) que possui as características para o tipo de serviço solicitado. Lembrando que esse objeto é construído com o auxílio do esquema NMWG;
- c) ***ServiceEngine*** - A principal atribuição do *ServiceEngine* é acionar a funcionalidade que está sendo solicitada na mensagem. Após a identificação de que serviço (ponto de medição, módulo de armazenamento, publicação e descoberta e etc.) está sendo requisitado, é necessário identificar qual funcionalidade desse serviço também está sendo solicitada. O *ServiceEngine* faz essa identificação e aciona a funcionalidade desejada;
- d) ***InformationReader*** - É um componente utilizado para consultar em base de dados algumas informações referentes à funcionalidade. Ou seja, se existe uma característica específica que é utilizada durante o processo do serviço, essa informação pode ser armazenada e consultada pelo componente *InformationReader*.

E os componentes de ajuda são:

- a) ***Property Reader*** - é responsável por ler em arquivos de configuração as propriedades do respectivo serviço, ou seja, recuperar em arquivos informações do tipo: localização de bibliotecas, informação e conexão a base de dados e etc;

- b) **Logging** - Realiza a construção de arquivos de *Log* onde são gerados registros de eventuais acontecimentos durante a execução do software. O *Log* é dividido em cinco níveis. No Quadro 5.1 é possível ver esses níveis e suas informações.

Nível	Nome	Descrição
1	<i>Info</i>	Usado para orientar os usuário e desenvolvedores em relação às operações realizadas pela aplicação
2	<i>Debug</i>	Utilizado pelos desenvolvedores no auxílio ao desenvolvimento. Serve como artifício de programação e ajuda na localização de falhas nos algoritmos, para verificar a seqüência de um algoritmo e etc.
3	<i>Warn</i>	São sinais de avisos que não são considerados erro. Exemplo, uma tentativa de conexão inválida, que pode ser uma tentativa de invasão
4	<i>Erro</i>	Relata os erros ocorridos, como: falha na conexão da base de dados, falha no serviço, falha nas informações de configuração e etc.
5	<i>Fatal</i>	Erro fatal é registro de quando a aplicação parou inesperadamente, ou teve que ser finalizada.

Quadro 5.1 - Níveis de *Log* do sistema

Nota: elaboração do autor (2007).

A Figura 5.3 proporciona um melhor entendimento das funcionalidades de cada componente. Essa figura traz um diagrama de seqüência referente à estrutura do Serviço de Publicação e descoberta proposto com uma analogia ao serviço genérico. Contudo, é preciso salientar que esse diagrama foi construído com base no diagrama do serviço genérico, logo para cada serviço específico, o mesmo pode ser alterado de acordo com as características do serviço específico. O fluxo representado na figura pode ser descrito assim:

- a) a requisição é recebida pelo componente *requestHandle*. Sendo que a requisição é um documento XML (`org.w3c.DOM.Document`) que é convertido em objetos através de *Parser XML*²⁴. Então, o *requestHandle* utiliza o dispositivo *MessageHandle Factory* para identificar o tipo de mensagem (armazenamento, ponto de medição, armazenamento e etc.) e construir um objeto do tipo *MessageHandler* com as informações contidas na mensagem. Finalizando o componente *requestHandle* aciona o *MessageHandle* passando o objeto *MessageHandler*. No cenário deste trabalho é possível dizer que o *requestHandle* identificou que a mensagem é para o serviço de publicação e descoberta;

²⁴ O *Parser XML* tem como função principal validar um documento XML, podendo também utilizar como parâmetro de validação um *XML Schema*. É utilizado também para criação de objetos de acordo com estrutura do documento XML.

- b) agora é necessário identificar que tipo de funcionalidade do Serviço de Publicação e Descoberta a requisição está solicitando. Assim, o componente *MessageHandler* ao manipular a mensagem e identificar o tipo da requisição (consulta, inserção, remoção e etc.), utiliza o dispositivo *ServiceEngine Factory* que faz parte do componentes *ServiceEngine*, para criar um objeto do tipo *ServiceEngine* apropriado (com as informações específicas contidas na solicitação) e acionar o componente *ServiceEngine* apropriado. Para exemplificar é possível dizer que é nesse passo onde os serviços específicos (armazenamento, ponto de medição, armazenamento e etc.) começam a aparecer. Pois, ao identificar o tipo de solicitação é criado o objeto *ServiceEngine* referente ao serviço específico. E é o *ServiceEngine* que executará as funções necessárias para atender à funcionalidade solicitada;
- c) o Componente *ServiceEgine* por sua vez tenta invocar as funções referentes (consulta, inserção, remoção e etc.) à informação contida no objeto que foi recebido o *ServiceEngine*. O Componente *ServiceEngine* também pode acionar o Componente *InformationReader*, para saber se existe alguma informação específica a ser passada na hora de invocar as funcionalidades desejadas. É o Componente *ServiceEngine* que também irá decidir se a resposta a ser enviada para o usuário é o resultado da solicitação ou uma mensagem de erro;
- d) ao tentar invocar uma ação o *ServiceEgine* aciona as classes de acesso ao UDDI de acordo com a ação solicitada. Essas classes utilizam a biblioteca de acesso desenvolvida para acessar de fato o UDDI através da função *Inquiry()*.

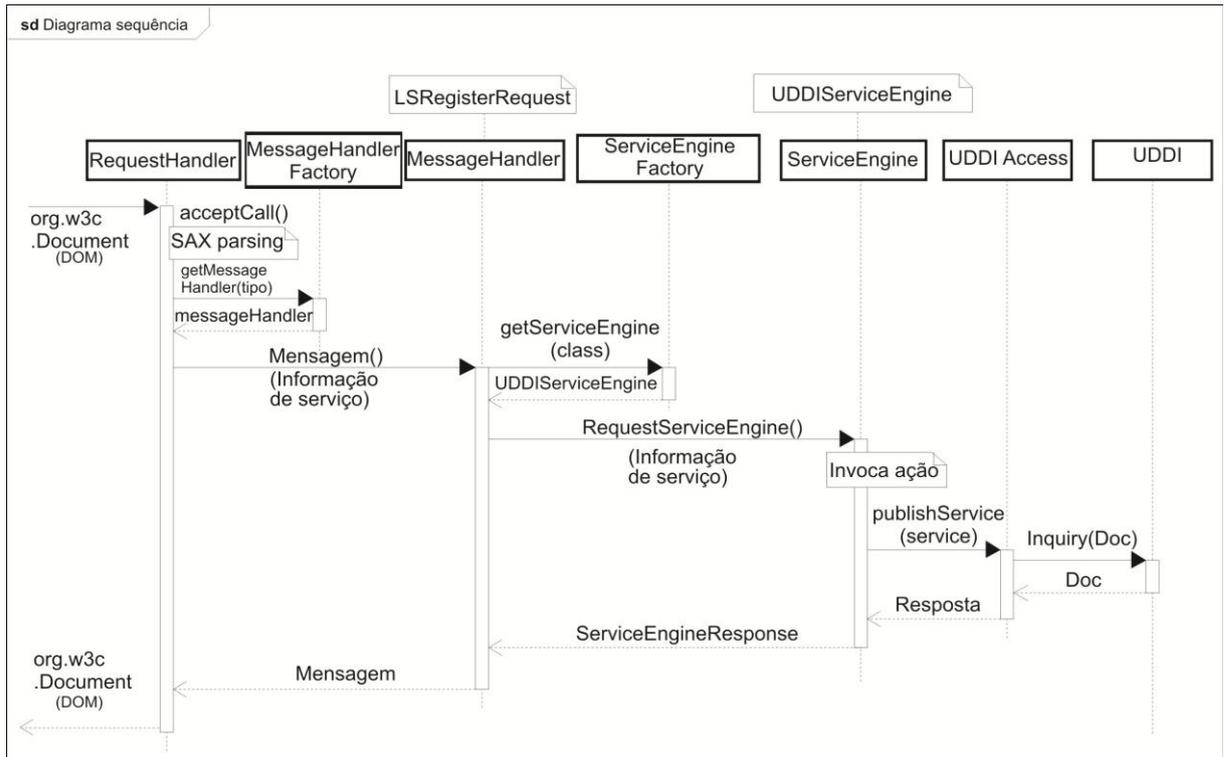


Figura 5.3 - de seqüência do Serviço de Publicação e Descoberta
 Fonte: ZURAWSKI e outros (2007).

Para finalizar, os componentes que envolvem o serviço, com o resultado de suas solicitações, compõem a mensagem que será enviada como resposta para o usuário. Assim, cada componente incorpora na mensagem as informações que fazem parte de suas funcionalidades. No final, essa mensagem é convertida em um documento XML para que possa ser enviada para o usuário.

5.2.2 Biblioteca de acesso ao UDDI

A biblioteca de acesso ao UDDI foi desenvolvida em JAVA utilizando alguns dos mecanismos das linguagens de programação orientada a objetos. Isso tudo para facilitar a manipulação das informações por parte dos usuários dos serviços de medições. A biblioteca possui uma estrutura formada por objetos que fazem um mapeamento das informações contidas no esquema do NMWG para informações de medições em redes de computadores. Na Figura 5.4 é possível ver esse mapeamento, onde à esquerda se encontra um exemplo de uma mensagem definida com o esquema NMWG e à direita essas informações armazenadas

na estrutura de componentes do padrão UDDI. Logo, os principais objetos da biblioteca são: *service*, *metadata* e *data*.

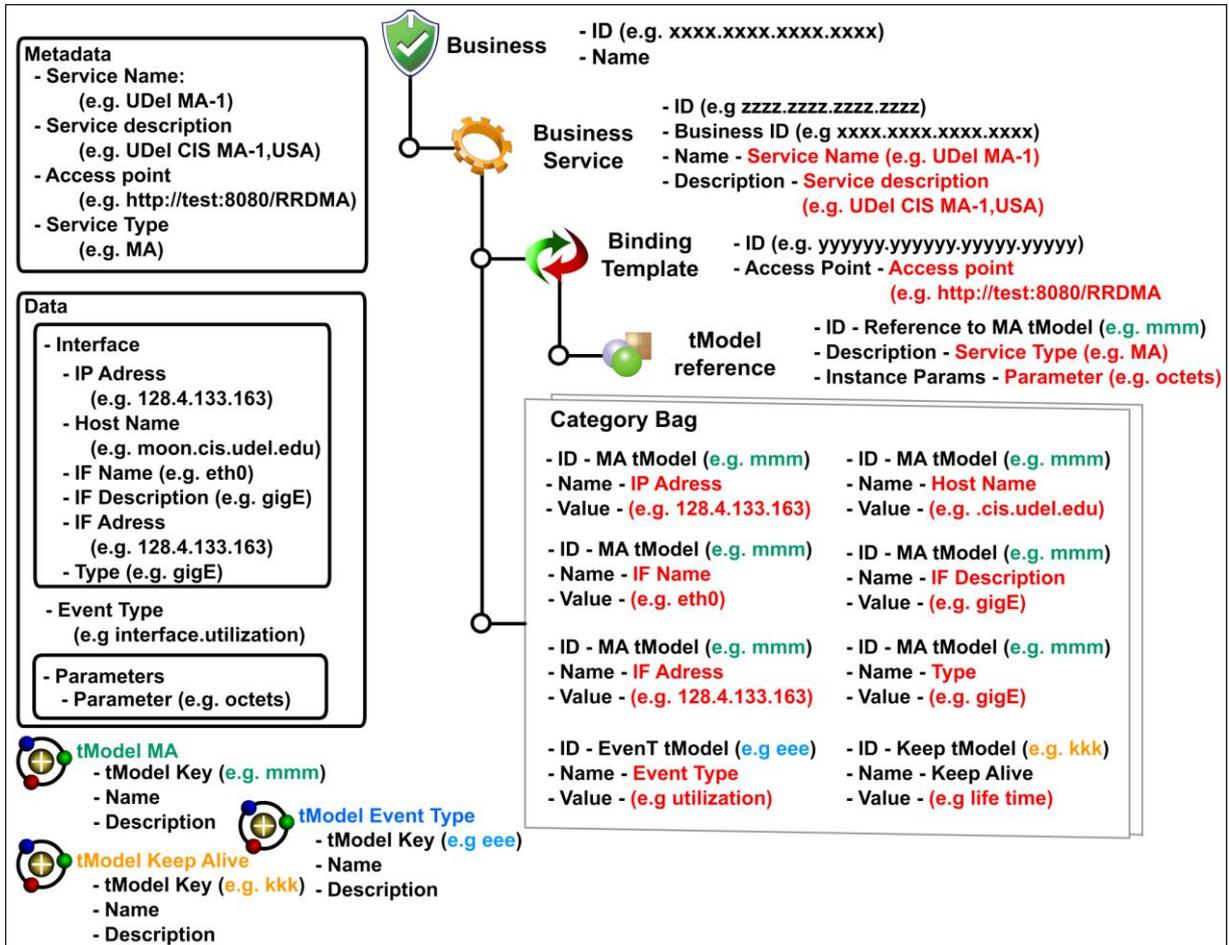


Figura 5.4 - Mapeamento da mensagem NMWG nos componentes do UDDI
Nota: elaboração do autor (2007).

Assim, como o Serviço de publicação e Descoberta, a biblioteca de acesso disponibiliza as seguintes funcionalidades:

- a) **publish** - Conhecida como Registro no LS UDDI *Type*, é responsável por publicar os serviços no UDDI;
- b) **unpublish** - É a remoção dos serviços publicados;
- c) **find** - Localização dos serviços;
- d) **change** - Alteração de um serviço já publicado.

Para acionar as funcionalidades do JUDDI, a Biblioteca de acesso utiliza outra biblioteca denominada UDDI4J. A UDDI4J disponibiliza uma interface que abstrai a conexão com o UDDI, mas ela segue o padrão definido pelo UDDI. Ou seja, utiliza os componentes do

UDDI, junto com a nomenclatura desses componentes. Isso pode dificultar o entendimento por parte do usuário. Logo, a biblioteca proposta neste trabalho utiliza termos conhecido pelos usuários de serviços de medições como nome de ferramentas e métricas.

Como a biblioteca foi desenvolvida para simplificar para os usuários, foi decidido desenvolver somente três métodos que podem ser acessados pelos usuários. O primeiro método é o mais importante e é denominado de *findService*. O método *findService* como a tradução de seu nome já diz, é utilizado para realizar a descoberta de Serviços Web. Para realizar a descoberta o *findService* recebe como parâmetros até 6 valores que são:

- a) **lsKey** - Possibilita a consulta pelo identificador do perfSONAR;
- b) **uddiKey** - Possibilita a localização de um serviço através da chave fornecida pelo UDDI;
- c) **type** - Possibilita a busca por tipo de serviço. Recorda-se que essa busca utiliza os objetos *tModel* do padrão UDDI;
- d) **name** - Possibilita a busca pelo nome do serviço;
- e) **id** - Possibilita a busca por informações contidas em taxonomias como: característica de rede e outras;
- f) **keepAlive** - Possibilita a busca de serviço com o tempo de vida inválido.

Ainda falando sobre o método *findService*, se o mesmo não receber nenhum parâmetro assume que a consulta deve retornar todos o serviços contidos no repositório. Outro método importante é o *publishService*, que recebe como parâmetro um objeto do tipo Service que está descrito na Figura 5.5. O *publishService* fornece as funcionalidades de publicação e alteração do serviço. Assim, se o serviço que o usuário pretende publicar já existir no registro, o método *publishService* considerará que se trata de uma alteração.

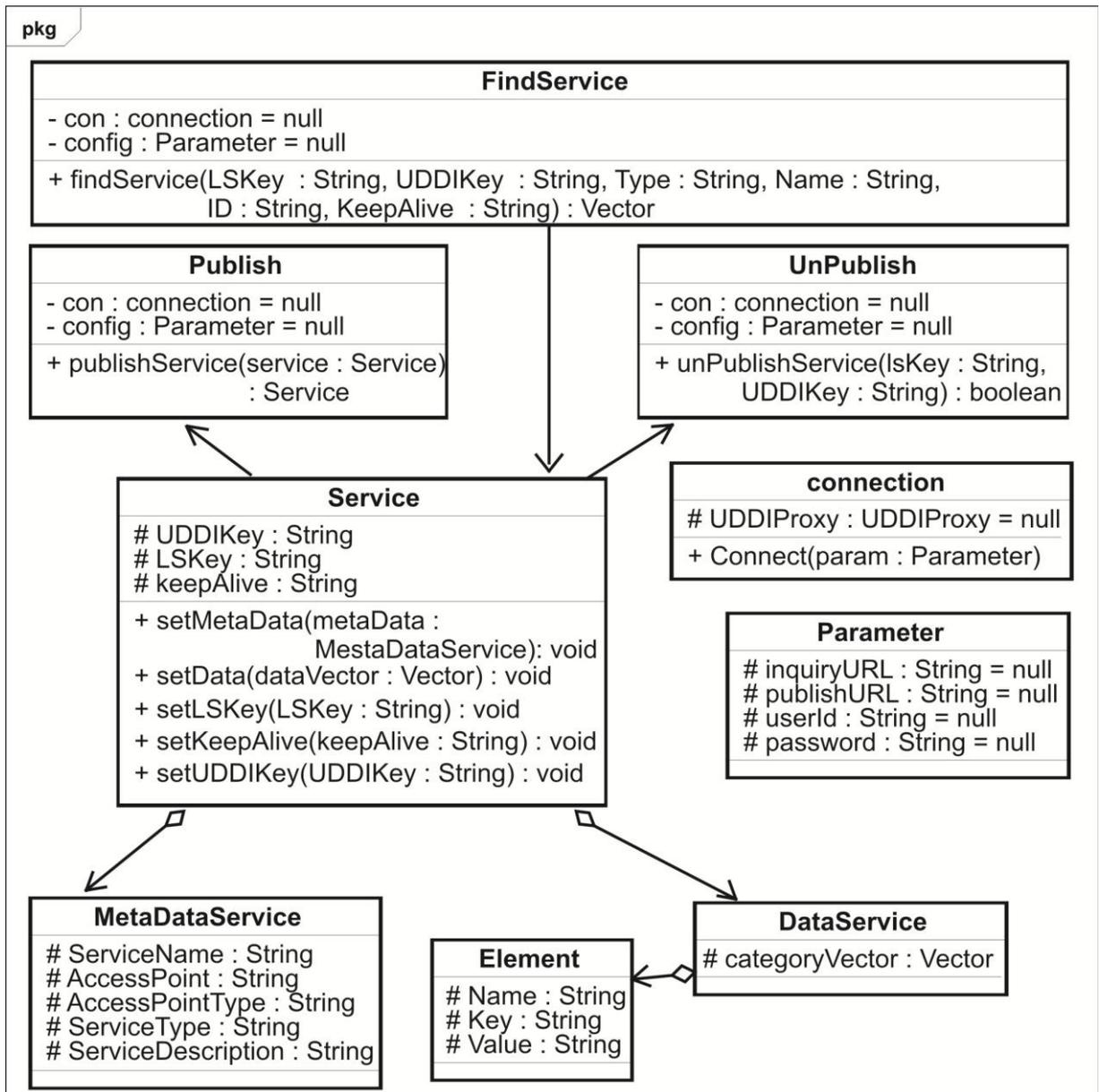


Figura 5.5 - Diagrama de classe básico para a biblioteca de acesso ao UDDI

Nota: elaboração do autor (2007).

O terceiro e último método é o *unPublishService*, que recebe como parâmetro um identificador do UDDI ou do perfSONAR para realizar a remoção de um serviço específico. Ou seja, para remover um Serviço Web é preciso saber a sua identificação. Caso o usuário não possua a identificação, o usuário deve realizar uma consulta prévia para receber essa identificação.

Na Figura 5.6 é possível ver o acesso ao serviço utilizando a biblioteca, bem como a arquitetura dessa biblioteca. Já a Figura 5.5 mostra o diagrama de classes da biblioteca. Essas duas figuras possibilitam o entendimento do funcionamento da biblioteca. Outro exemplo que pode ajudar

no entendimento do uso da biblioteca é o trecho de código no APÊNDICE A que mostra a utilização da biblioteca na publicação de um serviço.

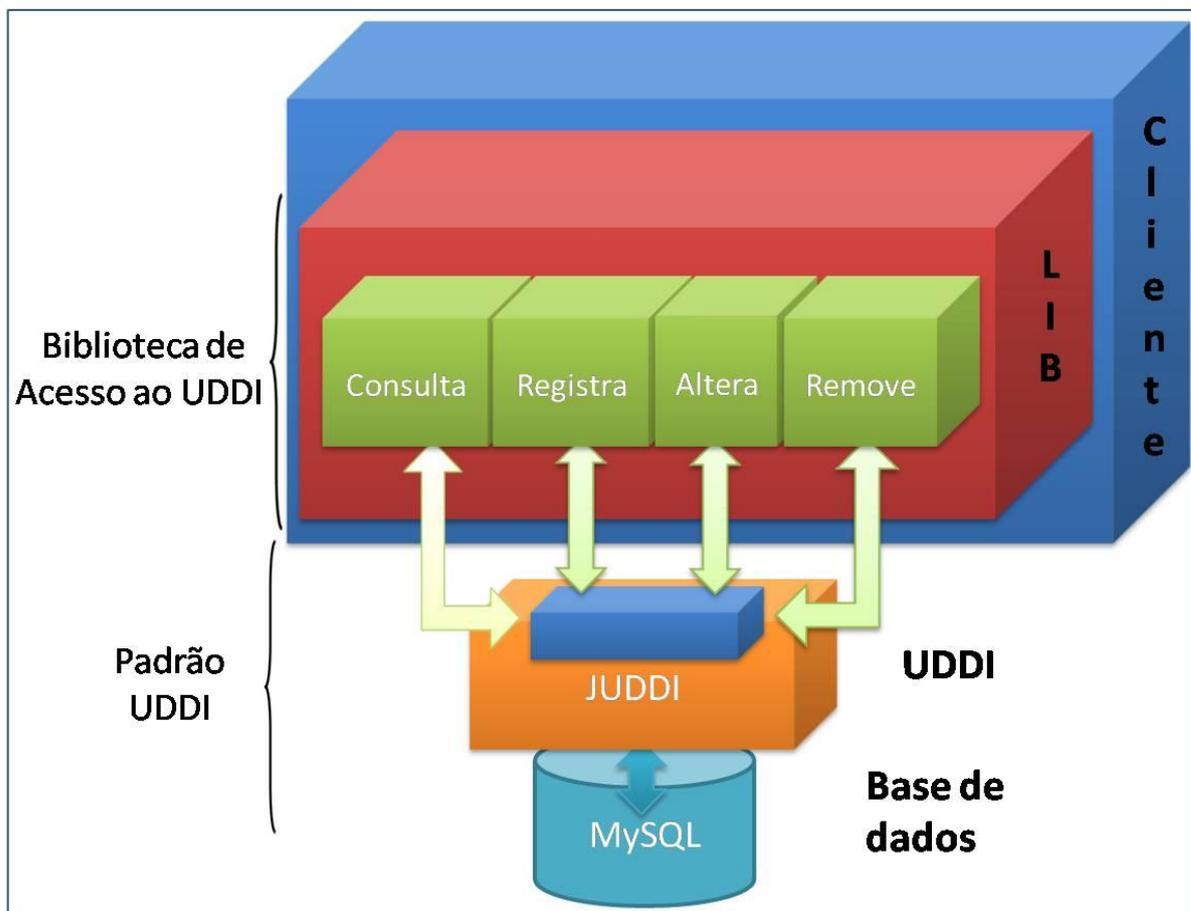


Figura 5.6 - Arquitetura da biblioteca de acesso ao UDDI

Nota: elaboração do autor (2007).

O Serviço de Publicação e Descoberta foi desenvolvido e pode ser obtido no site do GT-Medições. Onde é possível encontrar também explicações sobre o mesmo. É possível também obter no site o código fonte da primeira versão do serviço e assim desenvolver novas funcionalidades ou até mesmo usar como método de aprendizagem no uso do UDDI e perfSONAR.

Finalizando, para validar o protótipo desenvolvido é necessário realizar testes. Com esses testes é possível verificar se todas as funcionalidades desenvolvidas estão sendo implementadas a contento, além de verificar também o comportamento do software em uma possível utilização em produção, pois o intuito também desse protótipo é integrá-lo à infraestrutura que será implantada na RNP. Na próxima seção serão apresentados os testes realizados no protótipo do Serviço de Publicação e Descoberta proposto.

5.3 TESTES

5.3.1 Cenário de teste nº 1

Para testar o protótipo desenvolvido, foram usados dois cenários. O primeiro cenário foi implantado num laboratório em uma rede local sem muita interferência de tráfego externo. Esse cenário foi escolhido por dois motivos. O primeiro motivo é o fato de testar o protótipo numa ambiente controlado. O segundo motivo é de existir a possibilidades de realizar os mesmos testes com o serviço de publicação e descoberta do perfSONAR o *Lookup Service XML Type*, que utiliza outros métodos e assim poder comparar o resultado entre os dois serviços. Na Figura 5.7 é possível ver o cenário de testes 1.

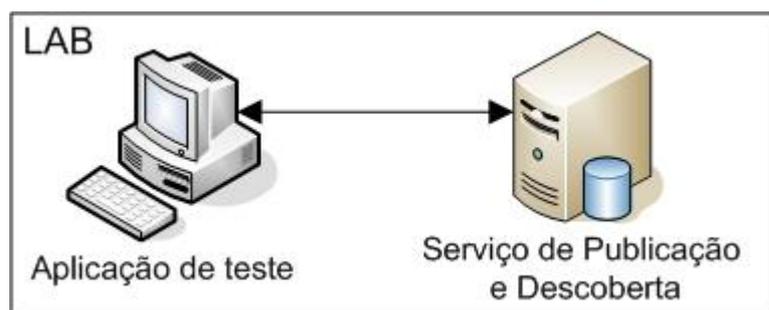


Figura 5.7 - Cenário de testes nº 1
Nota: elaboração do autor (2007).

Ainda sobre o cenário de testes nº 1, é possível dizer que o mesmo consistiu em testar as funcionalidades de remoção, publicação e alteração. A consulta não foi relacionada devido ao fato de que nas três funcionalidades citadas a consulta é realizada automaticamente, pois é preciso localizar o serviço antes para poder remover e alterar. Já na inserção, a consulta é realizada para verificar se o serviço já existe. Assim, foi desenvolvida uma aplicação cliente que requisita essas funcionalidades no serviço de publicação e descoberta utilizando a interface de Serviços Web no padrão perfSONAR.

Essa aplicação é bastante simples, antes de realizar a solicitação é marcado o instante de tempo inicial e ao receber a resposta do serviço é marcado o instante de tempo final. Com os dois resultados em mãos é só verificar o tempo que levou para que a aplicação fosse atendida

pelo Serviço de Publicação e Descoberta. No Figura 5.8 é possível ver na linguagem de programação JAVA a simplicidade do teste.

```
//MARCA O TEMPO INICIAL
double startTime = new Long(System.currentTimeMillis()).doubleValue();
//REQUISITA AO SERVIÇO DE PUBLICAÇÃO E DESCOBERTA
Object resultObject = call.invoke(new Object[] {requestMessage});
//MARCA O TEMPO FINAL
double endTime = new Long(System.currentTimeMillis()).doubleValue();
//IMPRIME O TEMPO DE RESPOSTA PARA A SOLICITAÇÃO
System.out.println("Resultado :"+(tempoFinal-tempoInicial)/1000d)+" secs");
```

Figura 5.8 - Código usado na aplicação de testes do cenário n° 1
Nota: elaboração do autor (2007).

Outro ponto importante a ser citado no cenário n° 1 é em relação às informações que foram usadas nas requisições. O teste que é relatado neste trabalho utilizou as informações referentes ao serviço de armazenamento de dados de medições o *Measurement Archive* (MA). Para ser mais específico, o tipo de MA utilizado foi o MA-RRD que armazena dados de utilização de enlaces dos dispositivos de redes (ex. roteador) em uma base de dados RRD. Assim, no caso específico dos dados de utilização, quanto mais enlaces, maior a quantidade de informação a ser armazenada, pois no esquema NMWG para dados de utilização, cada interface de rede dos dispositivos deve estar representada na mensagem. A mensagem utilizada no teste possui um objeto *Metadata* que descreve as informações genéricas do serviço e um objeto *Data* para cada interface de rede que faz parte da coleta de dados de medição. Ou seja, caso um MA esteja armazenando dados de utilização de dez roteadores e cada roteador possuir cem interfaces, a mensagem de registro que deve ser enviada para o Serviço de Publicação e Descoberta deve conter um *Metadata* e mil objetos *Data* representando as interfaces. A mensagem utilizada no teste pode ser visualizada no APÊNDICE A.

5.3.2 Cenário de teste n° 2

O segundo cenário testa o Serviço de Publicação e Descoberta na rede da RNP e tem o intuito de verificar como o serviço se comporta, publicando e descobrindo Serviços Web funcionais implantados na RNP. Assim é possível verificar com os usuários o nível de satisfação em relação à publicação e descoberta. Para o teste no *backbone* da RNP foi utilizada a infra-

estrutura disponibilizada por projetos pilotos como o infra-estrutura compartilhada entre Europa e América Latina (EELA, do nome em inglês). O GT-Medições foi requisitado pela RNP para implantar parte da sua infra-estrutura de monitoramento para fornecer alguns dados de medições nas redes que participam do projeto EELA. Na Figura 5.9 é possível ver detalhes do cenário n° 2, nesta figura estão representados os Pontos de Medição implantados, o Serviço de Publicação e Descoberta e o Serviço de Armazenamento dos dados de medição. Nesta figura também é possível visualizar as redes envolvidas nos testes e quem fazem parte do projeto piloto do EELA, envolvendo redes européias e latino americanas (RedIRIS, CLARA, CUDI, UNAM RNP e CIEMAT).

Assim, todos os Serviços Web disponibilizados pela infra-estrutura de medição piPES-BR/GFD instalada na RNP foram configurados para se registrarem periodicamente no serviço instalado também no *backbone* da RNP.

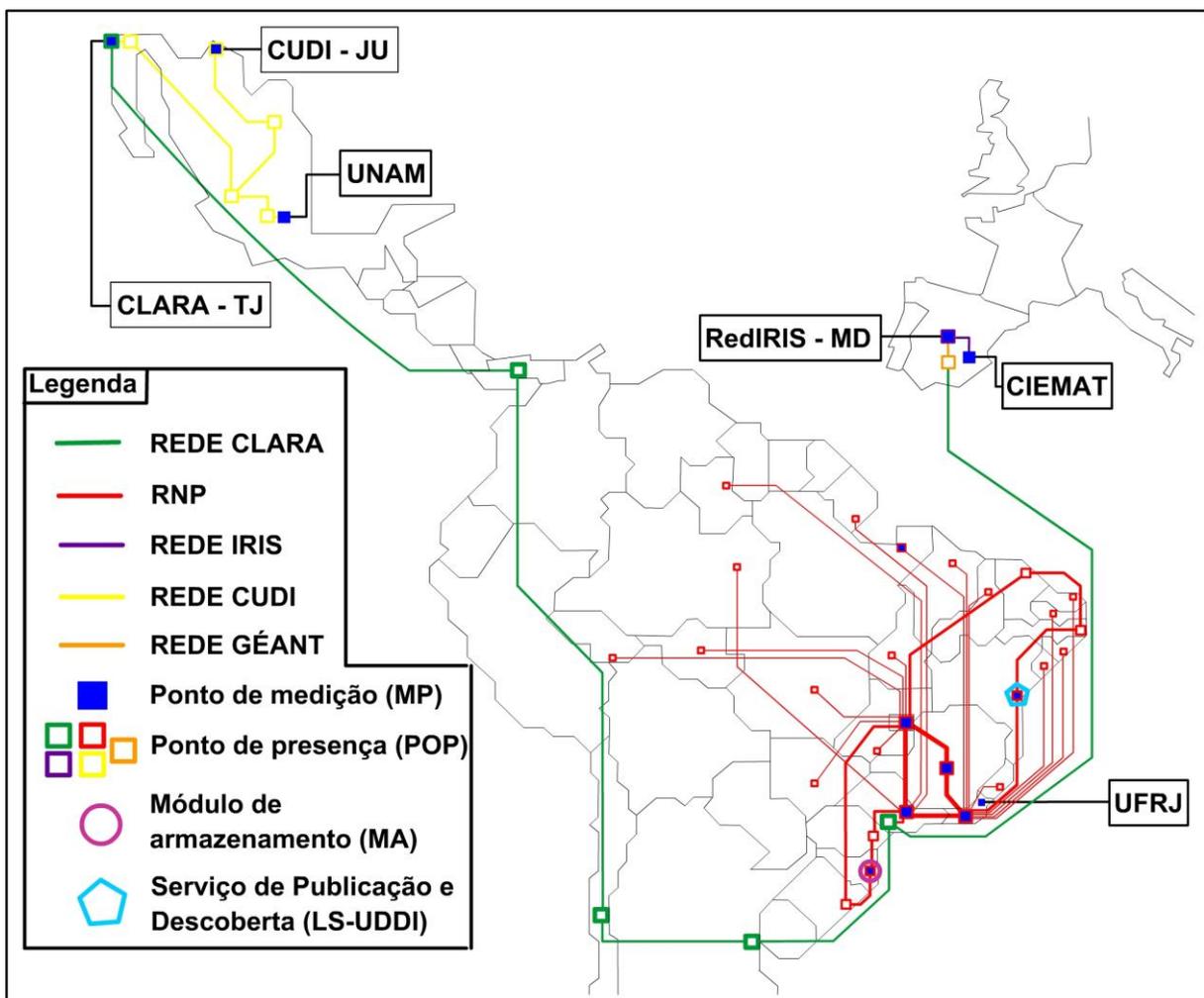


Figura 5.9 - Cenário de testes n° 2
Nota: elaboração do autor (2007).

Contudo o segundo cenário proporciona verificar se as funcionalidades do Serviço de Publicação e Descoberta estão sendo fornecidas, desde as funcionalidades principais como o registro, remoção, alteração, consulta e *keepalive*, até as funcionalidades como gerenciamento, sistemas de *log* e outros. Esse teste também possui a finalidade de validar a versão desenvolvida, mostrando que ela pode ser usada. Outro benefício desse teste é ajustar a configuração do serviço e observar coisas do tipo: qual o intervalo de tempo que deve ser utilizado para verificar os tempos de vida dos serviços registrados e realizar a remoção dos serviços inativos, verificar se os serviços estão se registrando continuamente e em um tempo satisfatório para que seu registro não seja removido e etc.

Na Figura 5.9 é possível ver também que os componentes da infra-estrutura piPEs-BR/GFD estão espalhados pelas redes onde o maior número de componentes são os pontos de medição do tipo de linha de comando o CL-MP. Outros componentes são os módulos de armazenamento que armazenam os dados medidos por esses pontos. Assim, todos esses componentes foram configurados para realizar a publicação de suas informações de acesso no Serviço de Publicação e Descoberta situado em Salvador. Assim, foram obtidos resultados que serão explicados e comentado na próxima seção.

5.4 AVALIAÇÃO DOS RESULTADOS

Uma das motivações de realizar os testes no cenário nº1, foi o fato de já existir um teste desse tipo com o *Lookup Service XML Type* do perfSONAR (LS-XML). Nesse teste os desenvolvedores do perfSONAR identificaram um problema que pode ser considerado grave. O desempenho do LS-XML com uma quantidade pequena de informações se demonstrou satisfatório, mas à medida que a quantidade de informação foi crescendo, o desempenho do serviço caiu drasticamente. Lembrando que o teste foi realizado com mensagens com informações sobre o serviço de armazenamento de dados de utilização dos dispositivos de redes (MA-RRD) e que o aumento de informação é causado pelo aumento do número de interfaces dos dispositivos monitorados, o que é comum em um *backbone* com muitos roteadores.

Assim que esses resultados foram divulgados, os desenvolvedores direcionaram sua atenção para o desempenho da outra aplicação para a Publicação e Descoberta que pode ser utilizada no perfSONAR e que também é o resultado deste trabalho. Assim, os mesmos testes foram realizados no Serviço de Publicação e Descoberta do piPEs-BR/GFD, o *Lookup Service UDDI Type* (LS-UDDI). E os resultados, apesar de serem esperados, comprovaram uma das justificativas para a realização deste trabalho que é o bom desempenho do serviço. A Figura 5.10 mostra graficamente os resultados dos testes realizados. Foram realizados sete testes para as funcionalidades de publicação, alteração e remoção, onde a cada teste ocorreu uma elevação na quantidade de informação. O primeiro teste foi realizado com uma mensagem de MA-RRD com apenas uma interface e consecutivamente com um objeto *Metadata* e um objeto *Data*. Já no segundo teste a quantidade de interfaces foi aumentada para dez e conseqüentemente o número de objetos *Data* mudou para dez. O terceiro teste com 50 interfaces e assim por diante, até chegar ao sétimo teste que utilizou mil interfaces. Os dados coletados podem ser vistos no apêndice dD.

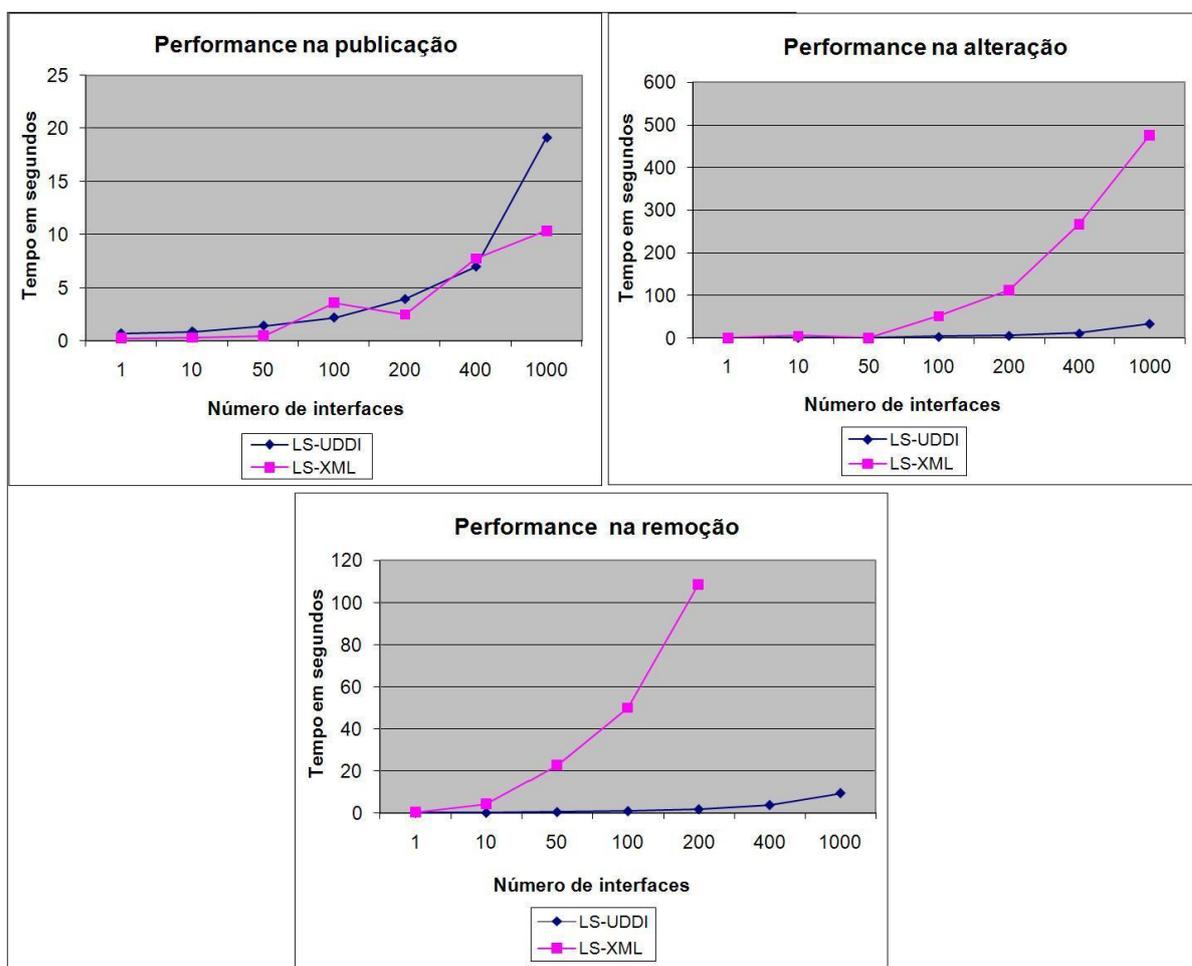


Figura 5.10 - Comparação de desempenho entre o LS-UDDI e o LS-XML

Nota: elaboração do autor (2007).

O testes mostram que na publicação de serviços ambas as implementações obtiveram resultados semelhantes, mas nas operações de alteração e remoção, o LS-UDDI à medida que a quantidade de informações aumenta se demonstrou mais veloz do que o LS-XML. Existem duas explicações que se complementam para esses resultados. A primeira é o fato das bases de dados relacionais serem comprovadamente mais rápidas do que as bases XML, isso é devido ao uso de mecanismos como índices e relacionamentos das bases relacionais. A segunda explicação é que a partir do momento em que se envolve a consulta de serviços na operação, o LS-UDDI com sua base de dados relacional possui um desempenho melhor. Ou seja, na publicação de serviço existe uma consulta que é para verificar se o serviço já existe, mas essa consulta é relativamente simples. Já na remoção e alteração de um serviço também existe a consulta de serviços, mas a complexidade aumenta à medida que a quantidade de objetos *Data* aumenta.

Ao remover ou alterar um serviço com o LS-XML, a base de dados XML ao localizar o *Metadata* correspondente ao serviço que será alterado ou removido, precisa localizar também os objetos *Datas* que pertencem também ao serviço. No caso de somente uma interface, a busca realizada pela base XML é por apenas um objeto *Data* na base de dados. No caso de mil interfaces, a base XML tem que localizar mil objetos *Datas*, o que é extremamente dispendioso para esse tipo de base que não possui os mecanismos utilizados por uma base de dados relacional. Como pode ser visto no Apêndice bB, a base XML apenas usa uma informação de identificação (ID) para relacionar os objetos *Data* com os *Metadatas*. Assim, toda vez que for necessário localizar um objeto *Data* é preciso percorrer toda a base de dados à procura do objeto com o ID correspondente.

Já os testes realizados no cenário nº 2 comprovam que todas as funcionalidades desenvolvidas no Serviço de Publicação e Descoberta proposto estão sendo cumpridas, pois os usuários das ferramentas de medição ao tentarem localizar os serviços das infra-estruturas piPEs-BR/GFD reportaram sucesso na operação, comprovando assim a dinamização no uso dos serviços Web. Outro resultado obtido com os testes realizados foi o perfeito funcionamento do sistema de *Log*, o qual reportou satisfatoriamente todas as vezes que foi acionado. O sistema de *Log* gerou um histórico das operações realizadas sobre o Serviço de Publicação e Descoberta, reportou também os erros ocorridos o que ajudou na configuração dos clientes em relação ao uso errados das mensagens ou na forma de acesso.

Utilizando ferramentas de visualização e até mesmo consultando a base de dados foi possível verificar que todos os serviços web da infra-estrutura pipEs-BR/GFD instalados no *backbone* da RNP se registraram com todas as informações desejada no serviço. Foi possível verificar também a eficiência do sistema de *keepalive*, pois por problemas de hardware alguns pontos de medição durante o período de teste pararam de se registrar no Serviço de Publicação e Descoberta e foram automaticamente removidos após o término dos seus tempos de vida.

Um teste importante que foi realizado, mas não utilizou nenhum tipo de cenário foi o teste da biblioteca de acesso ao UDDI. A ferramenta de visualização *Internet Computer Eye* (ICE) também desenvolvida pelo GT-Medições da RNP realiza a descoberta de serviço através das duas interfaces ficando a escolha a critério do usuário. Os testes de acesso na localização dos serviços através da biblioteca também foram satisfatórios, pois a ferramenta ICE não apresentou nenhum problema na sua utilização. No apêndice cC são apresentadas algumas telas que mostram a interface de acesso ao LS-UDDI no ICE.

Em relação à avaliação das tecnologias utilizadas, é importante começar com a aplicação base o JUDDI. O JUDDI se demonstrou uma aplicação bastante estável, pois nos testes realizados após a sua instalação não ocorreu nenhum problema. Outro fato importante de ressaltar é que o JUDDI implementa realmente todas as funcionalidades do padrão UDDI na sua versão número dois, pois utilizamos um cliente de acesso a qualquer implementação do padrão UDDI chamado de *UDDI Browser*²⁵ que mostra todas as funcionalidades disponíveis na implementação do UDDI ao qual está sendo feito o acesso. Foi testado no JUDDI também o sistema de autenticação o qual é feito por usuário e senha registrados no próprio JUDDI. Logo, o JUDDI apesar de só implementar a versão dois do padrão UDDI, pode ser utilizado tranqüilamente para a Publicação e Descoberta de Serviços Web de monitoramento. As outras tecnologias são bastante conhecidas nos cenários de medição e dispensam avaliações como: a *Apache Axis*, *Apache Tomcat*, JAVA e outras.

²⁵ Interface gráfica de acesso ao UDDI desenvolvida em JAVA.

6 CONSIDERAÇÕES FINAIS

Neste capítulo são apresentadas as conclusões obtidas com a realização deste trabalho. Portanto é possível dizer que este trabalho contribuiu para a descoberta de como utilizar o conceito de Publicação e Descoberta de Serviços da arquitetura SOA e o padrão UDDI no escopo das infra-estruturas de monitoramento de redes de computadores. Este capítulo também apresenta as contribuições efetivas deste trabalho, bem como as possibilidades de continuação da pesquisa como trabalhos futuros.

6.1 CONCLUSÃO

Na introdução deste trabalho foram levantadas questões que comprovam a necessidade de um Serviço de Publicação e Descoberta para auxiliar os usuários de Serviços Web das infra-estruturas de monitoramento. Pois os usuários dos Serviços Web possuem as seguintes demandas: acessar dinamicamente os Serviços Web, alcançar Serviços Web em domínios administrativos diferentes e possibilidade de realizar a escolha do Serviço Web a ser utilizado. Essas questões instigam a realização de uma pesquisa para saber como utilizar a Publicação e Descoberta no cenário dos Serviços Web de monitoramento. O que direciona a pesquisa para a tentativa de utilizar o padrão web UDDI para Publicação e Descoberta de Serviços Web nas infra-estruturas de medições para auxiliar os seus usuários. Assim, no Capítulo 2 foram apresentadas as possíveis tecnologias e padrões que podem ser usados na solução dos problemas relatados. Logo, este trabalho focou seus esforços na solução desses problemas com o intuito de planejar, desenvolver e testar um Serviço de Publicação e Descoberta com essas características.

Mas antes do planejamento, foi preciso verificar as principais infra-estruturas de monitoramento e observar o que essas infra-estruturas já possuem ou pretendem em relação à Publicação e Descoberta de Serviços. Assim, no Capítulo 3 foi verificado que poucas infra-estruturas de monitoramento utilizam um serviço para disponibilizar informações sobre Serviços Web e as que os possuem optaram ou por desenvolver seu próprio serviço ou utilizar

uma solução que resolva somente as necessidades internas da infra-estrutura. Como exemplo podemos citar o perfSONAR e a MonALISA onde, respectivamente, uma desenvolveu um novo serviço e a outra utilizou uma tecnologia que não possibilita que os serviços sejam acessados por aplicações externas.

Contudo, com os resultados obtidos na pesquisa, ficou comprovada a possibilidade de se usar o padrão UDDI no cenário de medições em redes de computadores utilizando a implementação JUDDI. Um dos receios em relação ao uso do UDDI para publicação e descoberta de serviço de monitoramento é o fato de ter que modificar o código fonte da aplicação que implementa o UDDI (JUDDI), caso o mesmo não contemplasse todas as funcionalidades previstas ou funcionalidades futuras. Realmente em relação a novas funcionalidades que podem aparecer no futuro, o padrão pode não contemplar e a necessidade de se alterar o código fonte da implementação seria inevitável. Assim, essa é uma flexibilidade que o UDDI não possui, mas os estudos comprovam que todas as funcionalidades previstas para publicação e descoberta de serviço web de monitoramento foram atendidas.

Entretanto, o serviço de publicação e descoberta utilizando o padrão UDDI demonstrou, nos testes realizados neste trabalho, que possui um melhor desempenho em relação à solução com base de dados XML. Isso é um fator importante no caso de Serviços Web de monitoramento, pois a tendência em relação à quantidade de informações sobre os serviços é sempre de aumentar, dado que as redes de computadores estão sempre em evolução seja no aumento da quantidade de dispositivos que na evolução nos *hardwares* instalados. Logo, é possível prever que a quantidade de informações que serão armazenadas no serviço de publicação e descoberta irá aumentar à medida que as redes evoluem.

Os resultados deste trabalho também verificaram que o Serviço de Publicação e Descoberta além de tornar dinâmico o acesso aos Serviços Web, faz com que os serviços web se tornem visíveis para qualquer usuário. Ou seja, o Serviço de Publicação e Descoberta se transforma em uma vitrine de serviços, onde os usuários escolhem o serviço que mais lhes agrada. Isso pôde ser verificado a partir das aplicações cliente com o ICE (vide Apêndice A), onde foi desenvolvido um módulo para acesso ao UDDI que possibilita uma visualização dos serviços publicados em uma estrutura de diretórios com as informações dos serviços. Assim, o usuário poderia escolher nessa estrutura o serviço que atende as suas necessidades.

6.2 CONTRIBUIÇÕES

A principal contribuição deste trabalho é a solução do problema de como utilizar o conceito de Publicação e Descoberta de Serviços da arquitetura SOA e a tecnologia UDDI no escopo das infra-estruturas de monitoramento. Neste trabalho, foram descritas as características que envolvem essa solução, tendo sido proposto um Serviço de Publicação e Descoberta de Serviços Web que atende os requisitos específicos dos Serviços Web de monitoramento e o desenvolvimento de um protótipo que além de validar o serviço proposto, pode ser utilizado em produção. Um serviço de publicação e descoberta com um desempenho no atendimento das requisições bastante satisfatório, abrangendo as funcionalidades definidas e que proporciona outras funcionalidades que melhoram o processo.

Mas esse trabalho contribuiu também com outros fatores que serão descritos a seguir. Assim, outra contribuição é a biblioteca de acesso ao UDDI que também segue o padrão perfSONAR e torna o acesso mais amigável para os usuários. A biblioteca pode ser obtida na página Web do GT-Medições e utilizada por qualquer aplicação JAVA que deseja localizar os Serviço Web da infra-estrutura piPEs-BR/GFD.

Contudo, o fato que será descrito a seguir pode até não ser considerado como uma contribuição, mas é correto afirmar que este trabalho é inovador no que diz respeito à Publicação e Descoberta de Serviços Web de monitoramento, pois na literatura não existe registro de pesquisas semelhantes, onde o padrão UDDI é inserido no cenário de medições, o que demonstra a importância deste trabalho. Esse fato também comprova que este trabalho irá servir como base para outros estudos na área e até mesmo para dar suporte a outras tecnologias. Assim, serão relatadas na próxima seção, as possibilidades de continuação da pesquisa na área bem como a utilização do material pesquisado junto com outras tecnologias que dão suporte às infra-estruturas de monitoramento.

6.3 TRABALHOS FUTUROS

Como trabalho futuro é possível realizar verificação da versão três do padrão UDDI que possibilita múltiplos registros do UDDI, ou seja, possibilita montar uma estrutura de registros UDDI, que descentralizam o serviço, aumentando assim a disponibilidade e facilita também a utilização das infra-estruturas em múltiplos domínios. O uso do UDDI em domínios diferentes com a sua versão dois é possível, mas ao tentar localizar um serviço que faz parte de um domínio diferente, o usuário teria que utilizar o UDDI desse domínio diferente. Com o uso da versão três, os registros UDDI replicam suas informações entre si possibilitando que o usuário de um UDDI de um determinado domínio possa localizar o serviço de outro domínio simplesmente perguntando ao UDDI de seu domínio. Logo ele não precisa conhecer os outros registros que fazem parte das infra-estruturas de monitoramento.

Contudo, como a intenção das infra-estruturas de monitoramento é a de operar em domínios administrativos diferentes, o estudo da versão três do padrão UDDI poderia ser realizado para verificar se o mesmo atende os requisitos em relação ao uso multidomínios das infra-estruturas de monitoramento. E assim torna possível a evolução do Serviço de Publicação e Descoberta proposto neste trabalho.

Atualmente a Internet também está sendo povoada por uma grande quantidade de serviços, assim também surgiu a necessidade de que esses serviços interoperem entre si. Outra necessidade que aparece com essa grande quantidade de serviços é a escolha correta do serviço, que atende as necessidades do usuário. A localização de serviços remete a um problema de semântica, pois ao se tentar localizar um serviço, geralmente se abstraem as características semânticas que envolvem os serviços e que os diferenciam. Utilizando somente o padrão UDDI, não é possível atribuir valores semânticos aos serviços. Logo existe um esforço na literatura para estudar a descoberta de serviços com base nas funcionalidades providas pelos mesmos. Paolucci (PAOLUCCI, 2002) afirma que para solucionar este problema é necessário utilizar uma linguagem que expresse as funcionalidades do serviço, incorporando valores semânticos aos serviços. Conseqüentemente ele propôs a utilização da linguagem de descrição de serviços DAML-S junto com o padrão UDDI para atribuir valores semânticos nas descrições dos Serviços Web.

Logo, como trabalho futuro também é possível verificar a utilização de mecanismos que atribuem valores semânticos na descoberta de serviços para o cenário dos Serviços Web de monitoramento, melhorando assim a localização dos serviços de acordo com as necessidades dos usuários. Pois, como já visto neste trabalho, os Serviços Web de monitoramento possuem muitas características que o identificam e que podem ser utilizadas.

Assim como a semântica, existem outros mecanismos que podem ser utilizados para otimizar e facilitar o acesso aos serviços Web. Um desses mecanismos é a Composição de Serviços (SRIVASTAVA; KOEHLER, 2003). A composição de serviços basicamente possibilita a construção de um novo serviço a partir de outros serviços. Um exemplo clássico para explicar a composição de serviço faz uma analogia a um agente de viagens, o qual compõe vários serviços para proporcionar apenas um serviço para o cliente que é a viagem. Um agente de viagens fornece pacotes de viagens para os usuários, por trás desses pacotes o agente de viagens precisa acionar outros serviços para compor o serviço final. Assim, o agente reúne serviços como: reserva de hotéis, compra de passagens, reserva de guias turísticos e passeio, etc.

Um exemplo da utilização da composição de serviços nas infra-estruturas de monitoramento pode ser descrito da seguinte maneira: a infra-estrutura de monitoramento proporciona informações de medição entre vários pontos. Um usuário deseja saber a informação de medição entre dois pontos e no caminho entre esse dois pontos existem vários serviços que fornecem informações em partes deste caminho. A infra-estrutura de monitoramento poderá compor com esses serviços um serviço final que utiliza esses serviços intermediários, para que o usuário não tenha que consultar todos esses serviços no caminho. Logo uma pesquisa sobre a utilização de composição de serviço nas infra-estruturas de monitoramento faz parte dos possíveis trabalhos futuros dessa dissertação.

REFERÊNCIAS

- ALMAER, D. **Creating Web Services with Apache Axis**. mai. 2002. Disponível em: <[Http://www.onjava.com/pub/a/onjava/2002/06/05/axis.html](http://www.onjava.com/pub/a/onjava/2002/06/05/axis.html)>. Acesso em: 14 set. 2006.
- ALONSO, G. et al. **Web Services: concepts, architectures and applications**. 1. ed. Heidelberg: Springer, 2004, 354 p.
- APTE, N.; MEHTA, T. **UDDI: Building Registry-based Web Services Solutions**. 1. ed. Upper Saddle River: Prentice Hall, 2002. 448 p. HP Professional Series.
- BENNETT, K. et al. Service-based software: the future for flexible software. In: ASIA-PACIFIC SOFTWARE ENGINEERING CONFERENCE, 7, 2000, Singapore. **Anais...** p. 214 - 221.
- BOOTE, J. et al. **Deliverable DJ.1.2.1: GEANT2 General Monitoring Framework Design, v.5**, jun. 2005. Disponível em: <<https://wiki.man.poznan.pl/perfsonar-mdm/images/perfsonar-mdm/9/95/GN2-05-057v5.pdf>>. Acesso em: 15 set. 2005.
- BOX, D. et al. **Simple object access protocol (SOAP) 1.1**, mai. 2000. Disponível em: <http://www.w3.org/TR/SOAP/>>. Acesso em: 10 jan. 2006.
- BOYD, E. L. et al. **E2E piPEline: End-to-End Performance Initiative Performance Environment System Architecture**. jul. 2002. Disponível em: <<http://e2epi.internet2.edu/e2epipe11.shtml>>. Acesso em: 20 ago. 2006.
- COLGRAVE, J.; AKKIRAJU, R.; GOODWIN, R. External matching in uddi. In: IEEE INTERNATIONAL CONFERENCE ON WEB SERVICES, 2, 2004, San Diego. **Anais...** p. 226-233.
- CURBERA, F. et al. Unraveling the web services web: an introduction to soap, wsdl, and uddi. **IEEE Internet Computing**, Los Alamitos, v. 6, n. 2, p. 86-93, mar./abr. 2002.
- GUDGIN, M. et al. **SOAP Version 1.2 part 1: Messaging Framework**. Abr. 2007. Disponível em: <<http://www.w3.org/tr/soap12-part1>> Acesso em: 15 mai. 2007.
- HANEMANN, A. et al. Perfsonar: A service oriented architecture for multi-domain network monitoring. In: INTERNATIONAL CONFERENCE ON SERVICE ORIENTED COMPUTING (ICSOC), 3, 2005, Amsterdam. **Anais...** Heidelberg: Springer, 2005. p. 241-254.
- HANEMANN, A. et al. Complementary visualization of perfsonar network performance measurements. In: INTERNATIONAL CONFERENCE ON INTERNET SURVEILLANCE AND PROTECTION (ICISP), 1, 2006, Côte d'Azur. **Anais...** p. 6-6.
- HOFREITER, B.; HUEMER, C.; KLAS, W. ebxml: Status, research issues, and obstacles. In: INTERNATIONAL WORKSHOP ON RESEARCH ISSUES IN DATA ENGINEERING, 12, 2002, San Jose. **Anais...** p. 57-67.

KARN, P.; PARTRIDGE, C. Improving round-trip time estimates in reliable transport protocols. **ACM SIGCOMM Computer Communication Review**, New York, v. 25, n. 1, p. 66-74, jan. 1994.

KAWAMURA, T. et al. Public Deployment of Semantic Service Matchmaker with UDDI Business Registry. In: INTERNATIONAL SEMANTIC WEB CONFERENCE (ISWC), 3, 2004, Hiroshima. **Anais...** Heidelberg: Springer, 2004. p. 752-766.

KOGA, I. K.; SAMPAIO, L.; MONTEIRO, J. A. S. Flavor: A dynamic and open framework for the development of network measurement access and visualization tools. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 25, 2007, Belém. **Anais...** p. 665-678

KOGA, I. K. et al. Ice: A flexible network monitoring access environment. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 25, 2007, Belém. **Anais...** p. 1181-188.

LOWEKAMP, B et al. **A hierarchy of network performance characteristics for grid applications and services**. Disponível em: < <http://www.ggf.org/documents/GFD.23.pdf>>. Acesso em: mai. 2005.

MARPLES, D.; KRIENS, P. The Open Services Gateway Initiative: An Introductory Overview. **IEEE Communication Magazine**. Los Alamitos, v. 39, n. 12, p. 110-114, dez. 2001.

MATTHEWS, W.; COTTREL, L. The PingER project: Active internet performance monitoring for the HENP community. **IEEE Communications Magazine**. Los Alamitos, v. 38, n. 5, p. 130-136, mai. 2000.

SOUZA, H. M. et al. Uma infra-estrutura para publicação e descoberta de serviços de monitoramento de rede. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 25, 2007, Belém. **Anais...** p. 679-692.

NEWMAN, H. B.; LEGRAND, I. C.; BUNN, J. J. A distributed agent-based architecture for dynamic services. In: COMPUTING IN HIGH ENERGY AND NUCLEAR PHYSICS (CHEP), 1, 2001, Beijing. **Anais...** p. 3-7.

NEWMAN, H. et al. Monalisa: An agent based, dynamic service system to monitor, control and optimize grid based applications. In: COMPUTING IN HIGH ENERGY AND NUCLEAR PHYSICS (CHEP), 4, 2004, Interlaken. **Anais...** p. 87-92.

PAOLUCCI, M. et al. Importing the semantic web in uddi. In: INTERNATIONAL WORKSHOP ON WEB SERVICES, E-BUSINESS, AND THE SEMANTIC WEB, 2, 2002, Toronto. **Anais...** Heidelberg: Springer, 2002 p. 225 – 236.

PAPAZOGLU, M. Service-oriented computing: Concepts, characteristics and directions. In: INTERNATIONAL CONFERENCE ON WEB INFORMATION SYSTEMS ENGINEERING (WISE), 4, 2003, Rome. **Anais...** p. 3-12.

PINTO, H.; BOAS, N. V.; JOSÉ, R. Utilização do uddi no suporte na descoberta de serviços baseados na localização. In: XML: APLICAÇÕES E TECNOLOGIAS ASSOCIADAS (XATA), 1, 2003, Braga. **Anais...** p. 13-21.

POSTEL, J. **Internet Control Message Protocol**, mai. 1981. Disponível em: <ftp://ftp.ietf.org/rfc/rfc0791.txt>. Acesso em: 16 jun. 2006.

ROY, J.; RAMANUJAN, A. Understanding web services. **IT Professional**, Los Alamitos. v. 3, n. 6, p. 69-73, nov. 2001.

SAMPAIO, L. et al. pipes-br: Uma arquitetura para a medição de desempenho em redes ip. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 24, 2006, Curitiba. **Anais...** p. 32-43.

SAMPAIO, L.; SURUAGY, J. A. M. A service-based flow traffic measurement management model for ip networks. In: WORKSHOP ON END-TO-END MONITORING TECHNIQUES AND SERVICES (E2EMON), 2, 2004, San Diego. **Anais...** p. 55-61.

SRIVASTAVA, B.; KOEHLER, J. Web service composition: Current solutions and open problems. In: INTERNATIONAL CONFERENCE ON AUTOMATED PLANNING AND SCHEDULING (ICAPS), 13, 2003, Trento. **Anais...** p. 155-161.

TERGUJEFF, R. et al. Mobile soa: Service orientation on lightweight mobile devices. In: IEEE INTERNATIONAL CONFERENCE ON WEB SERVICES (ICWS), 5, 2007, Salt Lake City. **Anais...** p. 1224-1225.

TSAI, W. T. et al. Consumer-Centric Service-Oriented Architecture: A New Approach. In: INTERNATIONAL WORKSHOP ON COLLABORATIVE COMPUTING, INTEGRATION, AND ASSURANCE, 2, 2006, Gyeongju. **Anais...** Washington, DC: IEEE Computer Society, p. 175-180

TURNER, M.; BUDGEN, D.; BRERETON, P. Turning software into a service. In: PENNINE RESEARCH FORUM, 1, 2002, Manchester. **Anais...** p.38-44.

WILLIAMSON, C. Internet traffic measurement. **Internet Computing IEEE**, Manchester, v. 5, n. 1, p. 70-74, nov. 2001.

XIMENES, S. B. **Dicionário da língua portuguesa**. 3. ed. São Paulo: Ediouro Publicações, 1998.

ZHOU, C. et al. UX - An Architecture Providing QoS-Aware and Federated Support for UDDI. In: INT. CONF. ON WEB SERVICES, 1, 2003, Las Vegas. **Anais...** p.104-109.

ZURAWSKI, J. et al. Hierarchically federated registration and lookup within the perfsonar framework. In: INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT, 10, 2007, Munich. **Anais...** p. 705-708

APÊNDICE A - EXEMPLO DE UTILIZAÇÃO DO UDDI4J

PUBLICAÇÃO DE UM SERVIÇO

```

\\Cria o objeto Parameters com as informações de conexão no UDDI
Parameters para = new Parameters();
para.setInquiryURL("http://200.128.80.179:8080/juddi/inquiry");
para.setPublishURL("http://200.128.80.179:8080/juddi/publish");
para.setUserid("herbert");
para.setPassword("xxxxxxx");
para.setTransportClassName("org.uddi4j.transport.ApacheAxisTransport");
para.setLogEnabled("false");
para.setHandlerPackageName("com.sun.net.ssl.internal.www.protocol");
para.setSecurityClassName("com.sun.net.ssl.internal.ssl.Provider");
para.setBusinessName("Sample Business");
para.setServiceName("Sample Service");
para.setTmodelName("Sample TModel");
para.setAssertionRelationship("peer-peer");
para.setBusiness("LS");

\\Cria o objeto Publish para executar a publicação
Publish app = new Publish(para);

\\Cria o objeto principal Service
Service service = new Service();

\\Cria o objeto Metadata que será inserido no objeto Service
MetadataService metaData= new MetadataService();

\\Cria o objeto Data que será inserido no objeto Service
DataService data= new DataService();

\\Como existe a possibilidade de existir mais de um objeto Data
Vector dataVector = new Vector();

\\Cria os elementos que fazem parte do objeto Data
Element ele = new Element();
Element ele1 = new Element();
Element ele2 = new Element();
Element ele3 = new Element();

\\Insere os parâmetros no objeto Metadata
metaData.setAccessPoint("http://200.237.193.2:8080/axis/services/CLMPService","http");
metaData.setServiceDescription("MP comand line");
metaData.setServiceName("mpcl-pop-sc");
metaData.setServiceType("CLMP");

\\Insere os parâmetros nos objetos Element

```

```
ele.setElement("Host Name");
ele.setValue("gtmed-ndt.pop-sc.rnp.br");
ele1.setElement("If Name");
ele1.setValue("eth0");
ele2.setElement("Capacity");
ele2.setValue("100000000");
ele3.setElement("Event Type");
ele3.setValue("utilization");
```

```
\\Insere os objetos Element no objeto Data
data.setDataElements(ele);
data.setDataElements(ele1);
data.setDataElements(ele2);
data.setDataElements(ele3);
```

```
\\Insere o objeto Data no vetor de Data
dataVector.add(data);
```

```
\\Insere os objetos Metadata e Data no objeto Service
servi.setDataService(dataVector);
servi.setMetadataService(metaData);
servi.setLocalizationService("POP-SC");
servi.setKeepAlive("1212121");
```

```
\\Executa o método de publicação
app2.publishService(servi);
```

APÊNDICE B – MENSAGEM UTILIZADA NO TESTE DO CENÁRIO Nº 1

MENSAGEM DE REGISTRO DE UM SERVIÇO DO TIPO MA-RRD

```

<!-- Purpose: -->
<!-- Version: $Id: LSRegisterRequest.xml 792 2006-02-21 -->

<nmwg:message type="LSRegisterRequest"
  id="msg1"
  xmlns:perfsonar="http://ggf.org/ns/nmwg/tools/
    org/perfsonar/1.0/"
  xmlns:nmwg="http://ggf.org/ns/nmwg/base/2.0/"
  xmlns:psservice="http://ggf.org/ns/nmwg/tools/
    org/perfsonar/service/1.0/"
  xmlns:nmwgt="http://ggf.org/ns/nmwg/topology/2.0/"
  xmlns:netutil="http://ggf.org/ns/nmwg/
    characteristic/utilization/2.0/">

  <nmwg:metadata id="serviceLookupInfo">
    <perfsonar:subject id="commonParameters" xmlns:perfsonar=
      "http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/">
      <psservice:service id="serviceParameters"
        xmlns:psservice="http://ggf.org/ns/nmwg/tools/
          org/perfsonar/service/1.0/">
        <psservice:serviceName>
          My_test_MA
        </psservice:serviceName>
        <psservice:accessPoint>
          http://reed.man.poznan.pl:8080/axis/services/MA
        </psservice:accessPoint>
        <psservice:serviceType>MA</psservice:serviceType>
        <psservice:serviceDescription>
          This is my testing MA
        </psservice:serviceDescription>
        </psservice:service>
      </perfsonar:subject>
    </nmwg:metadata>

    <nmwg:data id="data0" metadataIdRef="serviceLookupInfo">
      <nmwg:metadata id="meta1">
        <perfsonar:subject id="subj1" xmlns:perfsonar=
          "http://ggf.org/ns/nmwg/tools/org/perfsonar/1.0/">
          <nmwgt:interface xmlns:nmwgt=
            "http://ggf.org/ns/nmwg/topology/2.0/">
            <nmwgt:hostName>
              atlang-hstnng.abilene.ucaid.edu
            </nmwgt:hostName>
            <nmwgt:ifName>unknown</nmwgt:ifName>
          </perfsonar:subject>
        </nmwg:metadata>
      </nmwg:data>
    </nmwg:message>
  
```

```
<nmwgt:ifDescription>
  hstn:oc192(p2p)::show:intracloud
</nmwgt:ifDescription>
<nmwgt:ifAddress type="ipv4">
  198.32.8.34
</nmwgt:ifAddress>
<nmwgt:direction>in</nmwgt:direction>
<nmwgt:capacity>10000000000</nmwgt:capacity>
</nmwgt:interface>
</perfsonar:subject>
<nmwgt:eventType>utilization</nmwgt:eventType>
</nmwgt:metadata>
</nmwgt:data>

</nmwgt:message>
```

APÊNDICE C – CLIENTES DE ACESSO AO PROTÓTIPO DESENVOLVIDO

INTERNET COMPUTER EYE (ICE)

O ICE possui a funcionalidade de acessar o Serviço de Publicação e Descoberta desenvolvido através da biblioteca de acesso ao UDDI. Essa funcionalidade permite a visualização e remoção dos Serviços Web que estão registrados no Serviço de Publicação e Descoberta. A visualização é disponibilizada em forma de estrutura de diretórios, que segue a estrutura de informações do Serviço Web. É possível também adicionar o Serviço de Publicação e Descoberta que será acessado. Essas funcionalidades podem ser vistas na Figura 9.1 e Figura 9.2.

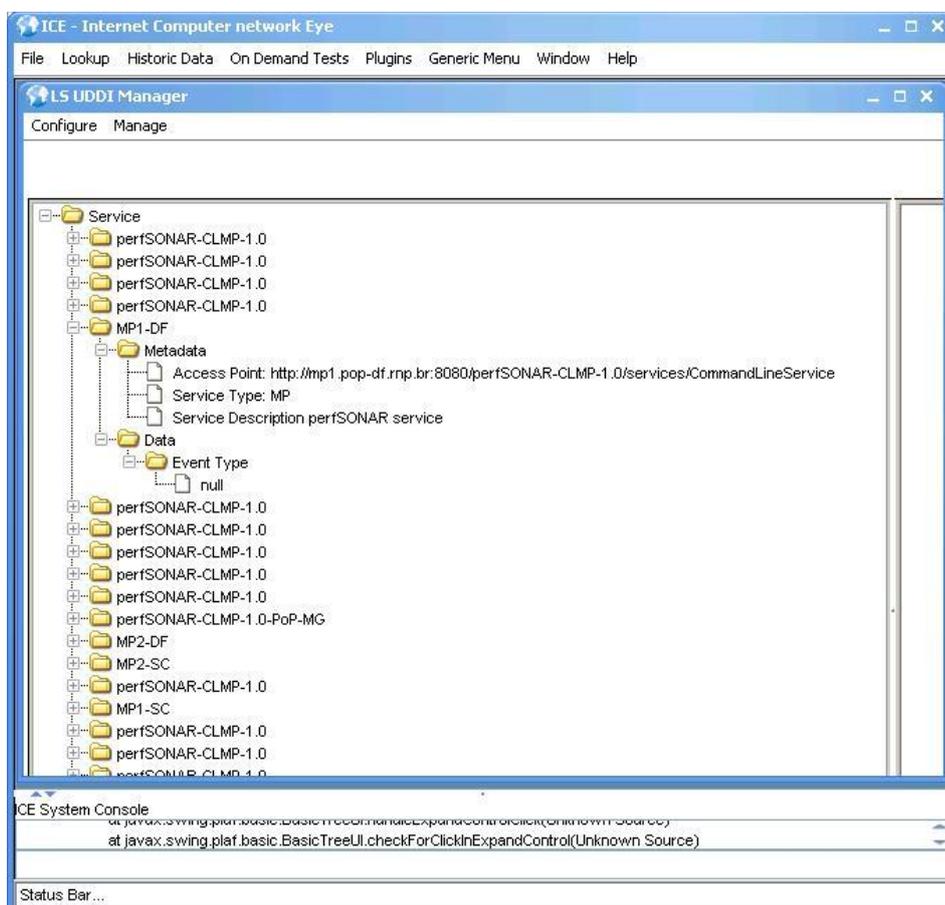


Figura 9.1 - Tela de acesso ao Serviço de Publicação e Descoberta

Nota: elaboração do autor (2007).

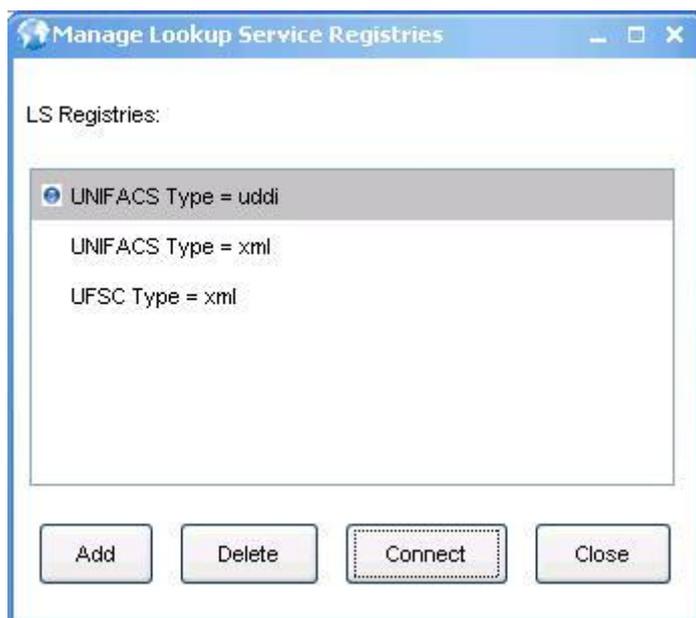


Figura 9.2 - Tela de configuração
Nota: elaboração do autor (2007).

UDDI BROWSER

Além do ICE, outro *software* foi utilizado para acessar o Serviço de Publicação e Descoberta desenvolvido. O *software* utilizado foi o *UDDI browser*, que realiza todas as operações disponíveis para um UDDI na sua versão dois. Essas operações são: consulta, inserção, remoção e alteração. Na Figura 9.3 é possível ver o *UDDI Browser* acessando o Serviço de Publicação e Descoberta.

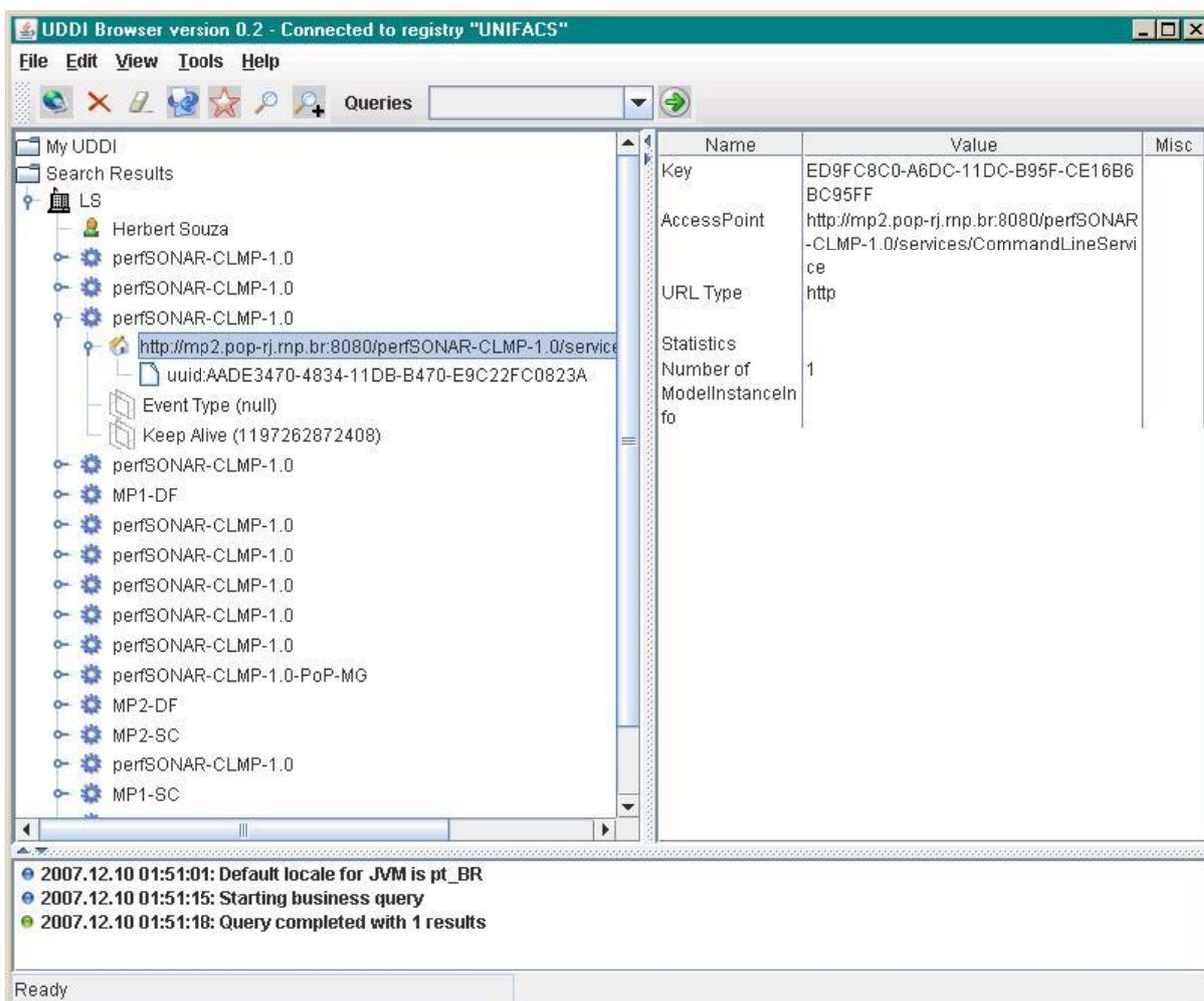


Figura 9.3 - Tela do *UDDI Browser*
 Nota: elaboração do autor (2007).

APÊNDICE D – RESULTADOS OBTIDOS NOS TESTES DO CENÁRIO Nº 1

SERVIÇO DE PUBLICAÇÃO E DESCOBERTA LS-UDDI E LS-XML

Tabela 10.1 - Resultado dos testes realizados com o LS-UDDI.

<i>LS-UDDI</i>			
INTERFACES	REGISTRAR	ALTERAR	REMOVER
1	0,685	0,775	0,195
10	0,862	0,941	& 0,272
50	1,423	1,89	0,635
100	2,186	3,033	1,057
200	3,923	5,686	1,922
400	6,998	10,999	3,904
1000	19,18	33,893	9,475

Nota: elaboração do autor (2007)

A

Tabela 10.1 e a Tabela 10.2, ajudam no entendimento dos testes realizados no cenário de testes nº 1. Pois aqui são apresentados os dados coletados nos testes de desempenho entre o LS-UDDI e o LS-XML.

Tabela 10.2 - Resultado dos testes realizados com o LS-XML.

<i>LS-XML</i>			
INTERFACES	REGISTRAR	ALTERAR	REMOVER
1	0,25	0,68	0,25
10	0,3	4,92	4,34
50	0,52	23,49	22,75
100	3,61	51,69	49,94
200	2,49	112,17	108,83
400	7,77	266,95	X
1000	10,38	476,85	X

Nota: elaboração do autor (2007)