

WESLEY DOS REIS BEZERRA

**SLM PARA WEB SERVICES, UMA PROPOSTA DE MODELO
AUTOMATIZÁVEL**

Florianópolis - 2006

WESLEY DOS REIS BEZERRA

**SLM PARA WEB SERVICES, UMA PROPOSTA DE MODELO
AUTOMATIZÁVEL**

Trabalho de Conclusão de Curso
para o Bacharelado em Sistemas
de Informação da Universidade
Federal de Santa Catarina,
Departamento de Informática e
Estatística.

Florianópolis - 2006

Sumário

Resumo	5
1. Introdução	6
2. SOA e Webservices	8
2.1 Web Service.....	8
2.1.1 SOAP(Simple Object Access Protocol)	8
2.1.2 WSDL(Web Service Description Language)	9
2.1.3 UDDI (Universal Description, Discovery and Integration).....	11
2.2 SOA	13
3. Governança em TI	15
3.1 ITIL (IT Infrastructure Library)	15
3.2 Gerenciamento de nível de serviço.....	18
SLA – Service Level Agreement.....	22
Proposta de Arquitetura	26
Coletores.....	28
Transporte	30
Entidade Gerente	31
Entidade Repositório	32
Entidade de Persistência	33
Entidade de Registro.....	34
Conclusão	35
Referências	36
Anexos	40
Relatórios.....	41
Coletores.....	42
Transporte	45
Painel	49
Gerencia de Redes Distribuídas – Estudo do uso do Apache/Axis na camada de transporte.	50

Lista de Ilustrações

Ilustração 1 - The service-oriented architecture[1].....	14
Ilustração 2 - Responsabilidades do Service Level Management	18
Ilustração 3 – Atividades do processo de SLM	20
Ilustração 4 – Fases do SLM	21
Ilustração 5 – Arquitetura proposta	26
Ilustração 6 - Coletores.....	28
Ilustração 7 – Coletor Abstrato.....	28
Ilustração 8 - Camada de transporte	30
Ilustração 9 - Responsabilidades da entidade gerente	31
Ilustração 10 - O repositório de SLAs	32
Ilustração 11 - Entidade de Persistência.....	33
Ilustração 12 - Entidade de registro	34

Resumo

Uma proposta de infra-estrutura para automatizar o processo de gerenciamento de nível de serviço, focando principalmente a forma como os dados são obtidos e transportados devido a característica principal dos web services de poderem estar geograficamente distribuídos.

É exposto uma especificação de um coletor assim como uma implementação padrão.

É exposta também a proposta da camada de transporte que se adequa as características dos serviços que estamos gerenciando.

1. Introdução

Atualmente a TI tem se tornado parte fundamental das empresas, mesmo de empresas onde a tecnologia da informação não é um fim e sim um meio.

Profissionais de TI têm ganhado espaço dentro da estrutura organizacional das empresas e cada vez mais o departamento de informática deixa de ser uma caixinha no cantinho do organograma que é vista como algo necessário e não como uma oportunidade ou algo integrante do processo.

Junto com o reconhecimento surge também a cobrança, e a necessidade maior de comunicação entre o setor de TI e os outros setores, comunicação interna. E como TI tem se tornado em muitos casos parte da solução em alguns setores ou uma fornecedora de meios para essa solução, surge a necessidade de uma melhoria na comunicação com fornecedores dos insumos necessários a TI e também no fornecimento, quando é o caso, de insumos a outras empresas, dos quais os que mais nos interessam neste trabalho são os serviços.

Também surge a necessidade de estabelecer padrões de qualidade tanto nos serviços fornecidos internamente ou externamente, como nos adquiridos de outros fornecedores.

Neste novo cenário surge a necessidade de formalizar os processos de comunicação e estabelecimento de padrões de qualidade tanto interna como externa a empresa na qual a TI esta atuando. Esta formalização pode acontecer de duas formas, criando-se um processo formal entre todas as partes envolvidas, ou adotando um modelo existente que atenda as necessidades de comunicação presentes.

Devido ao custo menor de adotar um processo já existente, e vários destes processos utilizarem como ferramenta de estabelecimento de padrões de qualidade o SLA (Service Level Agreement) este trabalho o adotou como foco de

estudo e como solução aos problemas relacionados ao nível dos serviços fornecidos ou adquiridos pelas empresas.

Os principais objetivos deste trabalho são:

Apresentar um framework de governança sucintamente para melhor visualização de em qual parte do processo de governança de TI se encaixa a criação e gerencia dos SLA's

Introduzir alguns conceitos básicos do nosso tipo de serviço a ser utilizado neste trabalho, no caso Web Services.

Introduzir um conceito que atualmente esta em bastante uso e do qual o nosso tipo de serviço é uma das partes principais, o SOA.

Fazer uma proposta de framework de desenvolvimento para gerenciar o nível de serviço em um nível mais abstratos expondo os requisitos necessários a implantação da mesma, também especificando os arquitetura física que deve comportar este gerenciamento.

Alguns pontos desta arquitetura abstrata tem um proposta de implementação. Esta proposta tem o intuito de fazer uso dos recursos fornecidos pelas características existentes na natureza dos serviços a serem gerenciados. Sendo assim existe uma proposta da camada de transporte e também dos coletores que a utilizarão.

2. SOA e Webservices

2.1 Web Service

É uma parte lógica do negócio, alocada em algum lugar na internet, que é acessível através de protocolos baseados em padrões da Internet como http e smtp.

Por causa da promessa de interoperabilidade cross-plataform feita pelo SOAP e Web Services, nos podemos prover soluções de negócio práticas a problemas que, até agora, tem sido somente um sonho para a computação distribuída.

Tem como características:

XML-based:

Baixo acoplamento:

Grão grosso:

Habilidade de ser síncrono ou assíncrono;

Suporte a RPC(Remote Procedure Call);

Suporte a troca de documentos;

Principais tecnologias relacionadas:

2.1.1 SOAP(Simple Object Access Protocol)

Provê uma arquitetura de empacotamento padrão para transportar XML documentos sobre uma variedade de tecnologias padrões na Internet, incluindo SMTP,http e FTP.

Exemplo de Envelope SOAP

(<http://www.intertwingly.net/stories/2002/03/16/aGentleIntroductionToSoap.html>)

```
<SOAP:Envelope
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP:Body>
    <m:getStateName xmlns:m="http://www.soapware.org/">
      <statenum>41</statenum>
    </m:getStateName>
  </SOAP:Body>
</SOAP:Envelope>
```

2.1.2 WSDL(Web Service Description Language)

É uma tecnologia XML que descreve a interface de um Web Service de modo a padronizá-lo;

Exemplo de WSDL

(http://www.capeclear.com/support/manuals/help/new_wSDL_files/example_wSDL_file.htm)

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="Stock"
targetNamespace="http://www.capeclear.com/Stock.wsdl"
xmlns:tns="http://www.capeclear.com/Stock.wsdl"
xmlns:xsd1="http://www.capeclear.com/Stock.xsd"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <import
location="http://www.capescience.com/registry/schemas/SchemaRegistry.xsd"
namespace="http://www.capescience.com/registry/schemas"/>
  <types>
    <xsd:schema targetNamespace="http://www.capeclear.com/Stock.xsd"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      <xsd:complexType name="ArrayOfstring">
        <xsd:complexContent>
          <xsd:restriction base="SOAP-ENC:Array">
            <xsd:sequence>
              <xsd:element name="item" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>
    </types>
  </definitions>
```

```

<xsd:attribute ref="SOAP-ENC:arrayType"
wsdl:arrayType="xsd:string[]" />
</xsd:restriction>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>
</types>
<message name="getPrice">
<part name="symbol" type="xsd:string" />
</message>
<message name="getPriceResponse">
<part name="return" type="xsd:int" />
</message>
<message name="getQuotedStocks" />
<message name="getQuotedStocksResponse">
<part name="return" type="xsd1:ArrayOfstring" />
</message>
<message name="getHighestPrice">
<part name="stocks" type="xsd1:ArrayOfstring" />
</message>
<message name="getHighestPriceResponse">
<part name="return" type="xsd:int" />
</message>
<portType name="Stock">
<operation name="getPrice">
<input message="tns:getPrice" />
<output message="tns:getPriceResponse" />
</operation>
<operation name="getQuotedStocks">
<input message="tns:getQuotedStocks" />
<output message="tns:getQuotedStocksResponse" />
</operation>
<operation name="getHighestPrice">
<input message="tns:getHighestPrice" />
<output message="tns:getHighestPriceResponse" />
</operation>
</portType>
<binding name="StockBinding" type="tns:Stock">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
<operation name="getPrice">
<soap:operation soapAction="capeconnect:Stock:Stock#getPrice" />
<input>
<soap:body use="encoded" namespace="capeconnect:Stock:Stock"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding" />
</input>
<output>
<soap:body use="encoded" namespace="capeconnect:Stock:Stock"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding" />
</output>
</operation>
<operation name="getQuotedStocks">
<soap:operation soapAction="capeconnect:Stock:Stock#getQuotedStocks" />

```

```

<input>
<soap:body use="encoded" namespace="capeconnect:Stock:Stock"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="capeconnect:Stock:Stock"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
<operation name="getHighestPrice">
<soap:operation soapAction="capeconnect:Stock:Stock#getHighestPrice" />
<input>
<soap:body use="encoded" namespace="capeconnect:Stock:Stock"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</input>
<output>
<soap:body use="encoded" namespace="capeconnect:Stock:Stock"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
</output>
</operation>
</binding>
<service name="Stock">
<documentation>Stock</documentation>
<port name="Stock" binding="tns:StockBinding">
<soap:address location="http://localhost:8000/ccx/Stock" />
</port>
</service>
<!--Created by CapeConnect on Tue Jul 03 13:33:03 GMT+01:00 2001
See http://www.capeclear.com for more details-->
</definitions>

```

2.1.3 UDDI (Universal Description, Discovery and Integration)

Provê um world wide registro de Web Services para propósitos de propaganda, descoberta e integração.

OracleAS Service Registry – é a implementação da Oracle para UDDI, é uma plataforma que suporta a versão 3 da especificação. Permite publicar, categorizar e descobrir Web Services e Recursos através da empresa.

jUDDI – implementação da ASF(Apache Software Foundation) para UDDI v2, baseado em Java, independente de plataforma, utiliza container servlet

que implementem a especificação 2.3, e utiliza qualquer banco de dados relacional que implemente ANSI SQL.

2.2 SOA

É uma nova tecnologia que provê uma poderosa infra-estrutura para construção da nova geração de aplicações;

A arquitetura que usa a distribuição, ambiente de execução baseada em descoberta para expor e gerenciar uma coleção de recursos de software orientados a serviços é chamada Composite Computing Model. A Composite Computing Model define uma visão do que deve ser computado. SOA representa o modo para conseguir esta visão usando o conjunto de tecnologias que compõe a pilha da tecnologia de Web Services.

Cada Web Service é composto de duas partes:

Serviço;

Descrição do Serviço;

Os três principais participantes do são (figura 01):

Service provider – que fornece o serviço;

Service requestor – o interessado na utilização do serviço;

Service broker – registra os serviços;

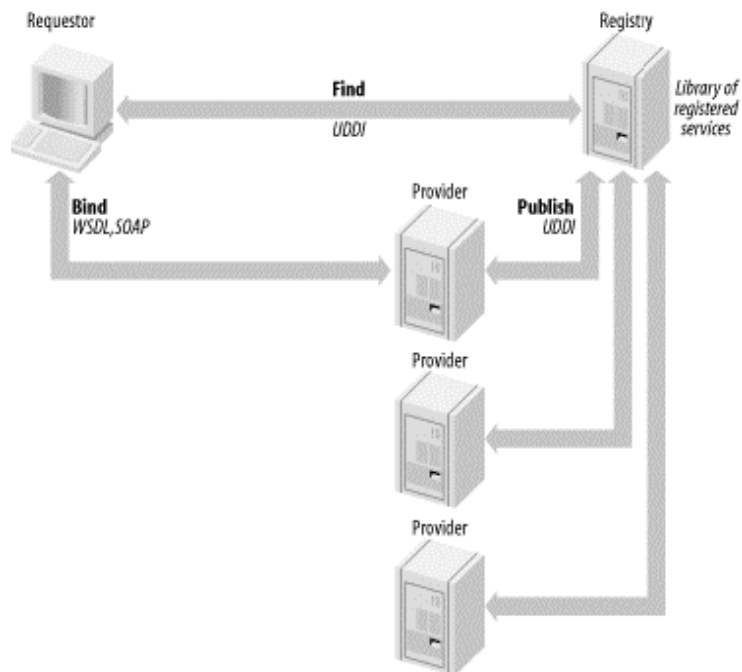


Ilustração 1 - The service-oriented architecture[1]

Service provider

“Um negócio que vê a si mesmo realizando algum passo de um serviço eletrônico”[1]. O service provider vê o SOA como um framework para expor seus serviços web. Estes serviços são ilhas de códigos projetadas para resolver um aspecto da totalidade dos problemas do negócio.

Service requestor

Vê o SOA como algo que ele usa para acessar os serviços web.

3. Governança em TI

3.1 ITIL (*IT Infrastructure Library*)

É um framework de governança em TI que fornece um conjunto de documentos com as melhores práticas na área de serviço. Tem forte foco nos processos, no entanto é mais limitado no desenvolvimento e na segurança.

Com o uso de um framework de governança podemos confrontar alguns principais desafios das organizações de TI, como por exemplo:

- Contribuir para resolver desafios do negócio;

- Uma mensurável contribuição para a cadeia de valor do negócio;

- Prover serviços como oposição á entrega de produtos de TI;

- Um negócio como relacionamento;

- Um serviço estável e consistente;

Menos ênfase em tecnologia;

O ITIL apresenta facilidades como:

- A descrição e objetivos dos vários serviços em um ambiente de TI.

- Uma representação de como estes serviços se inter-relacionam.

- Um guia para implantação com sucesso destes serviços.

É dividido em duas áreas:

- Service Management;

Service Delivery;

Pode ser dividido em 10 processos e uma função:

Gerência de configuração

Service Desk (função)

Gerência de incidentes

Gerência de problemas

Gerência de mudanças

Gerência de “release”

Gerência de capacidade

Gerência de avaliabilidade

Gerência de continuidade do serviço de TI

Gerenciamento financeiro de serviços de TI

Gerenciamento de Nível de Serviço

Os objetivos chave do gerenciamento de serviço em TI é alcançado através da utilização dos três P's:

Pessoas – clientes, usuários e equipe de TI;

Processos – ITSM / ITIL;

Produtos – ferramentas e tecnologia;

Service Desk:

Cliente – quem financia o serviço;

Usuários – quem usa o serviço;

Usuários Experts – linha de frente no suporte;

3.2 Gerenciamento de nível de serviço

“SLM (Service Level Management) é uma metodologia disciplinada e proativa e procedimentos aplicados para assegurar que os níveis adequados de serviço sejam prestados a todos os usuários da TI, de acordo com as prioridades empresariais e a um custo razoável.” [21] .

O acordo de nível de serviço é uma ferramenta que vem se tornando bastante usada no fornecimento de serviços através da definição de parâmetros para garantir e monitorar a qualidade do serviço fornecido. Este processo no qual do qual o SLA faz parte tem diversas responsabilidades(vide figura 02).

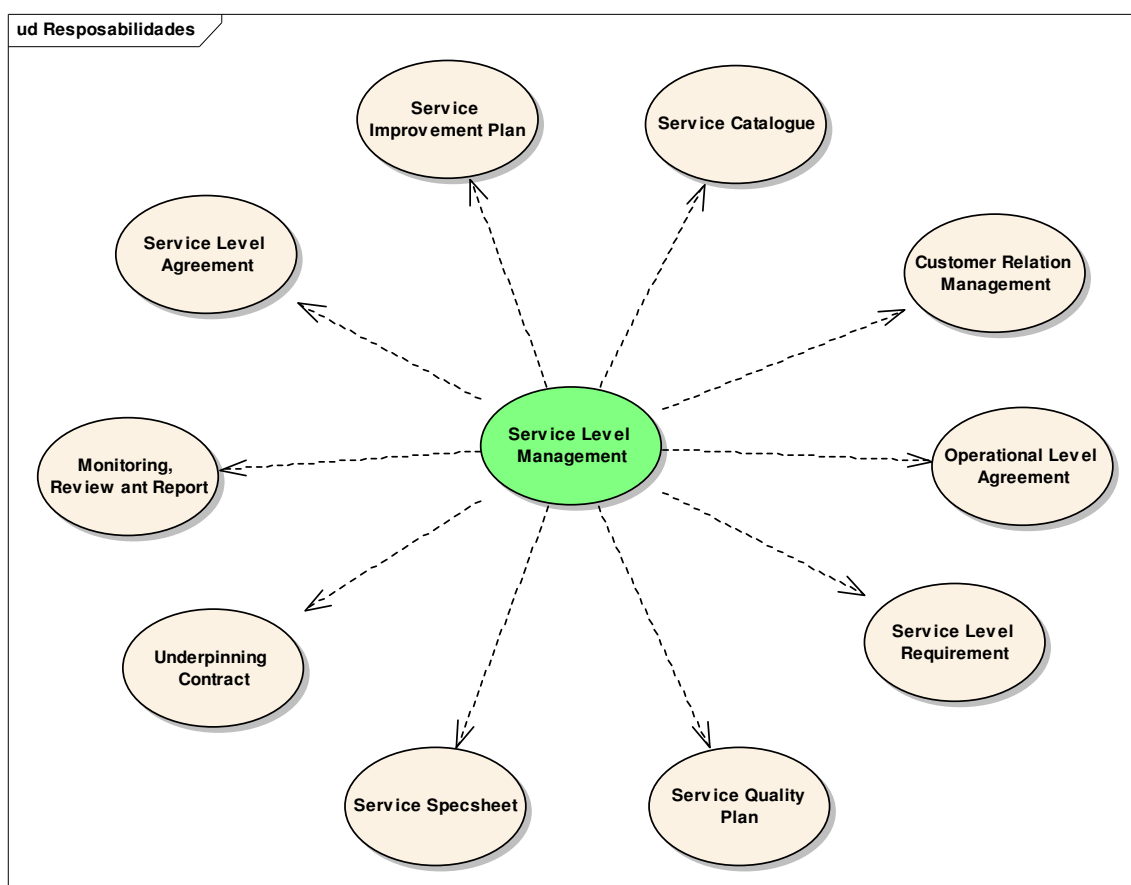


Ilustração 2 - Responsabilidades do Service Level Management

Responsabilidade do Gerenciamento de Nível de serviço

Service Catalogue.

Service Level Agreement

Service Level Requirement

Operational Level Agreement e Underpinning Contract

Service Specsheet

Service Quality Plan

Monitoring, Review and Report

Service Improvement Plan

Customer Relationship Management

Processo de Gerenciamento de Nível de Serviço

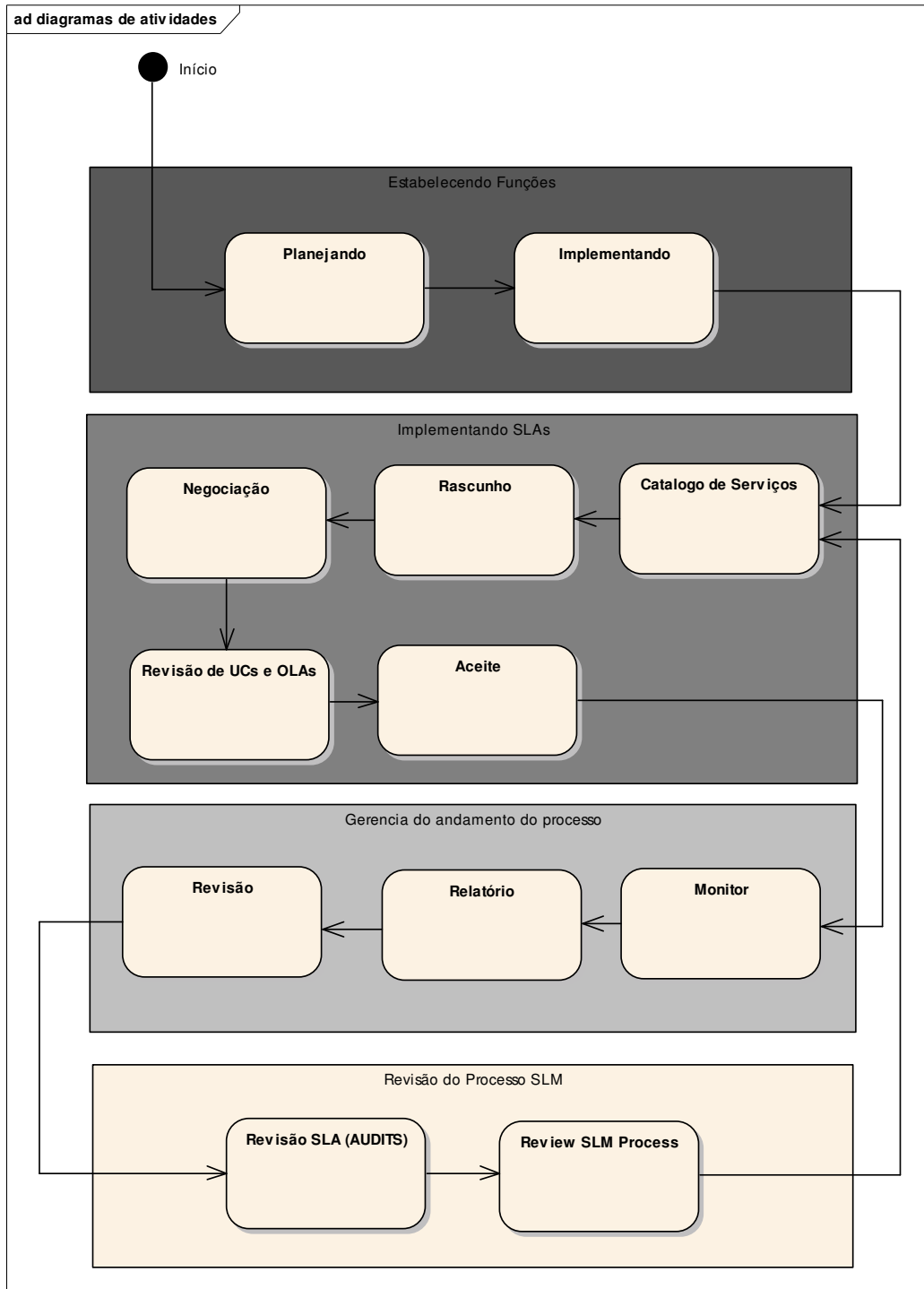


Ilustração 3 – Atividades do processo de SLM

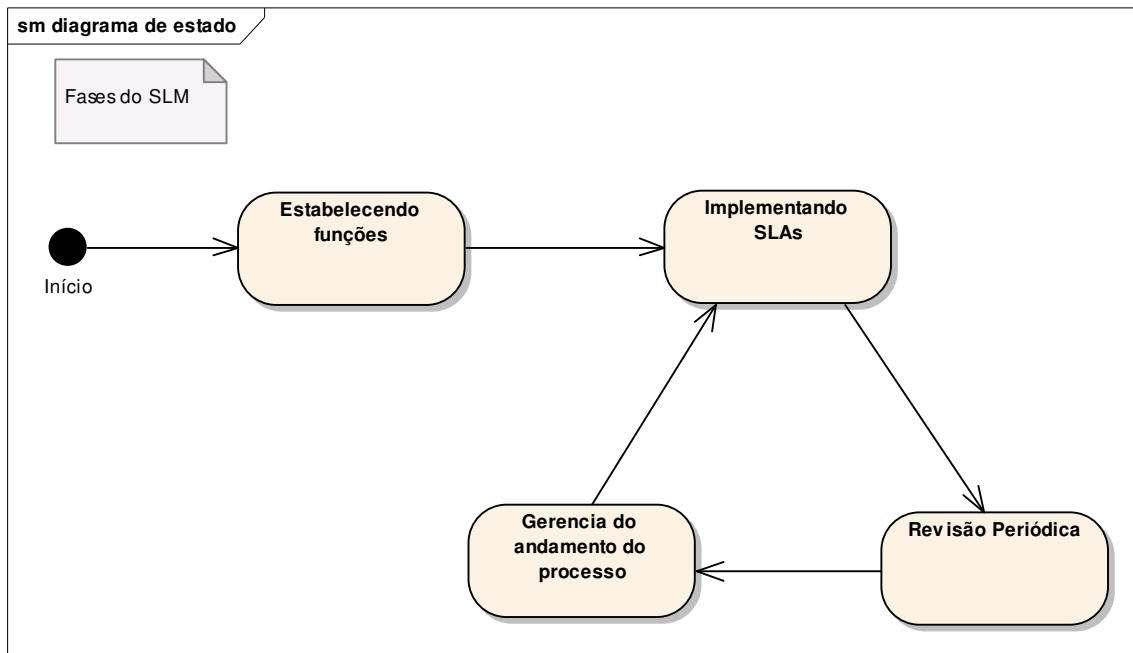


Ilustração 4 – Fases do SLM

Estabelecendo Funções

Plano inicial de atividades

Plano de monitoramento de capacidades

Estabelecendo percepção inicial dos serviços

Configurando / Validando contrato sub-fixado e acordo de nível operacional

Gerenciando o andamento dos processos

Monitorando e relatando.

Reunião de revisão dos serviços.

Revisões periódicas

Reuniões periódicas de revisão de serviços.

Criação de programa de melhoria de serviços.

Manutenção dos SLAs contratados e OLAs.

SLA – Service Level Agreement

Porque especificar um SLA é importante?

Problemas do SLA [6]

Verificação do esforço vs especificação do resultado;

Especificação dos serviços não clara;

Especificação incompleta dos serviços;

Gerencia de custos insuficientes

“dead-end” SLA documents;

Para diminuição dos problemas na especificação de um SLA é necessário o conhecimento prévio de alguns aspectos da utilização de um SLA. São estes:

Um SLA é mais objetivo quando ele reflete as necessidades de um cliente em termos inteligíveis.

Um SLA pode ser especificado em termos de eficiência de negócio.

Um SLA que é especificado em termos de mensuração irá aumentar a inteligibilidade e o consenso no processo de construção de várias partes envolvidas.

Um SLA deve ser baseado em componentes bem definidos do SLA.

Diferentes grupos de usuários têm diferentes necessidades de negócio.

Projetar a estrutura do SLA irá aumentar a qualidade do mesmo e diminuirá o tempo necessário para o desenvolvimento.

Implementação do SLA.

Produzir catalogo de serviços

Gerenciar expectativas

Plano de estruturação do SLA

Estabelecendo Requisitos de Nível de Serviço

Fraseando os SLAs.

Buscando acordos.

Estabelecendo e monitorando capacidades.

Revisando contratos sub-fixados e acordo de nível de operacional.

Definindo relatórios e revisando processos.

Tornar o público o SLA.

O SLA tipicamente tem os seguintes componentes:

Uma descrição do serviço que deve ser provido;

A expectativa de performance esperada pelo serviço;

Um detalhado procedimento de resolução dos problemas com o serviço;

Um procedimento para monitorar e reportar o nível de serviço ao cliente;

As conseqüências para o não cumprimento dos níveis de serviço pelo provedor de serviço;

Uma descrição de sobre quais circunstancias o SLA não se aplica;

Ciclo de vida do SLA

Fase de criação;

Fase de operação;

Fase de remoção;

Parâmetros propostos para serviços baseados em IP[3]:

Tempo de delay

Jitter

Proporção de perda

Spurious packet rate

Troughput

Avaliability

Algumas ferramentas de verificação utilizadas em SLA:

- Netranet – é uma implementação do METER MIB[RFC 2720] definido pela IETF Real-time Traffic Flow Measunrement Working Group;

- NTOP – é uma ferramenta integrada que pode fazer vários tipos de mensurações e análises do tráfico IP;
- MIB II – é a MIB que todo equipamento IP deve implementar;
- RMON – padrão definido pelo IETF que permite o monitoramento completo de um segmento de rede composto de vários hosts;
- MRTG (Multi Router Traffic Grapher) – é uma interface que pode monitorar enviando e recebendo tráfico nas interfaces do router, apresentar os resultados de forma gráfica em uma pagina web.

Proposta de Arquitetura

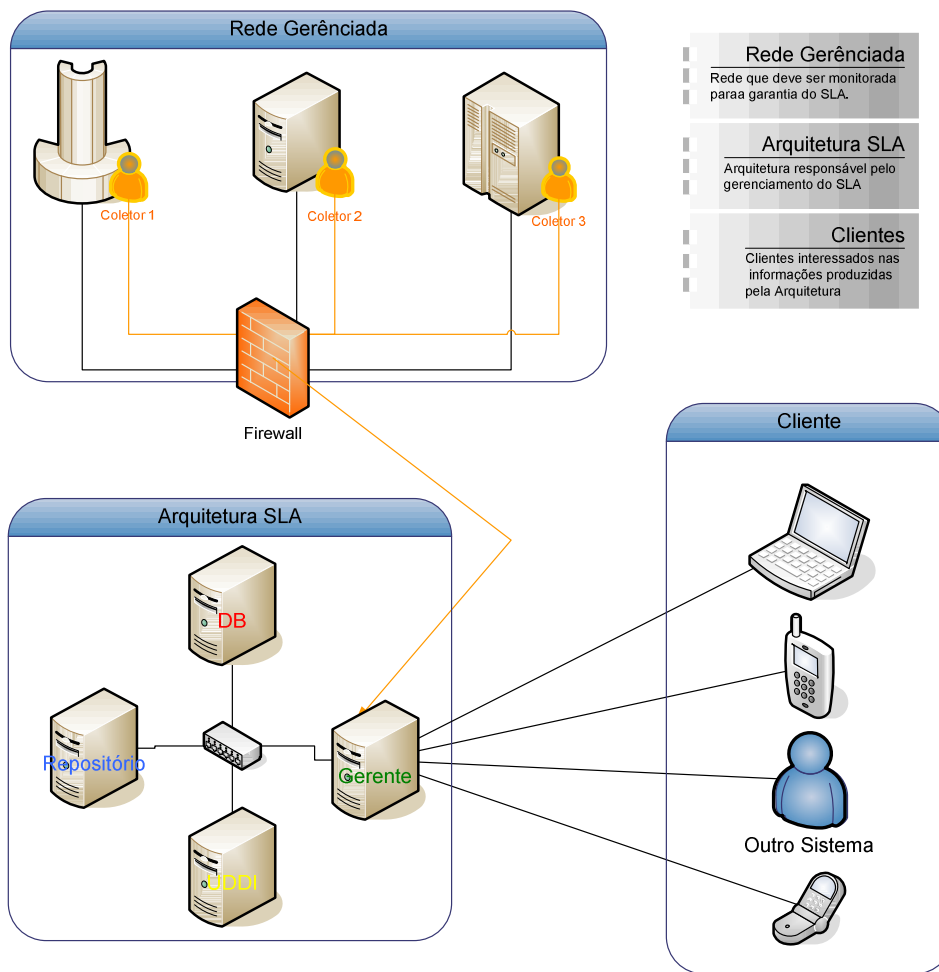


Ilustração 5 – Arquitetura proposta

A proposta se baseia em uma entidade gerente, uma entidade que persistirá as coletas, um repositório de SLA, uma entidade responsável pela publicação dos serviços. Externamente auxiliando o monitoramento existem os coletores específicos.

O gerente que é responsável por se comunicar com os coletores através da camada de transporte feita sobre SOAP, posteriormente detalhada, também é responsável por fornecer uma interface com suporte a vários dispositivos. No entanto a principal função do gerente é fazer Monitoramento, Revisão e Relatos sobre o cumprimento ou não dos Requisitos de Nível de Serviço.

A entidade de persistência fornece o serviço de persistência ao gerente e também deve garantir a integridade e segurança dos dados removendo assim as responsabilidades sobre os dados do gerente.

A entidade de repositório fornece as informações sobre os SLAs assim como os SLRs de cada SLA gerenciado, no entanto ela não pode fazer alterações nos mesmo, sendo somente estes utilizados como fonte de consulta para verificação da conformidade dos serviços com o padrão estabelecido.

A entidade de publicação permite o acesso de maneira facilitada aos serviços que são gerenciados pela arquitetura.

Coletores

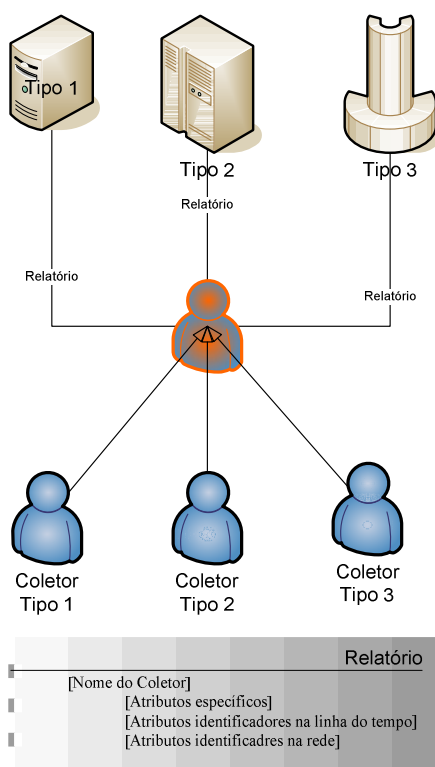


Ilustração 6 - Coletores

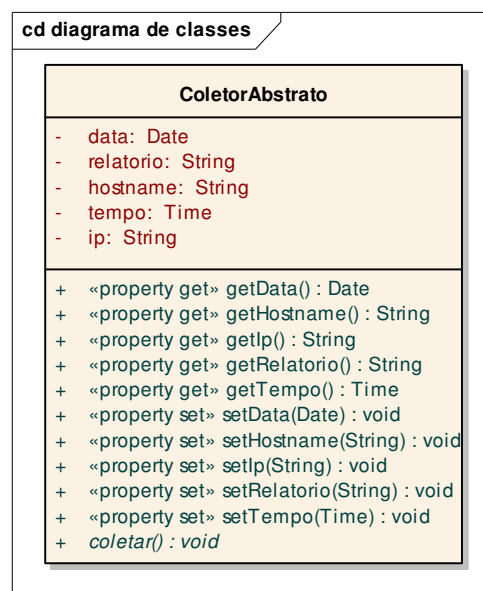


Ilustração 7 – Coletor Abstrato

O serviço de coleta é implementado através de uma especificação, que na tecnologia orientada a objeto utilizada é expressa por uma classe abstrata que deve ser estendida conforme cada necessidade específica.

Essa interface especifica que um coletor deve implementar o método coletar. Este método é responsável pelo preenchimento dos atributos do objeto coletor, é neste método que é montado o relatório para envio ao interessado na informação coletada, no nosso caso o gerente.

O relatório é montado em formato textual facilitando assim seu entendimento até mesmo a um humano, e por seguir um formato estabelecido pode ser facilmente interpretado por uma entidade computacional.

Os coletores não tem uma dependência das outras camadas, sendo proposta sua implementação na forma de componentes, pois devido a esta independência estes coletores possuem um alto grau de reuso.

Transporte

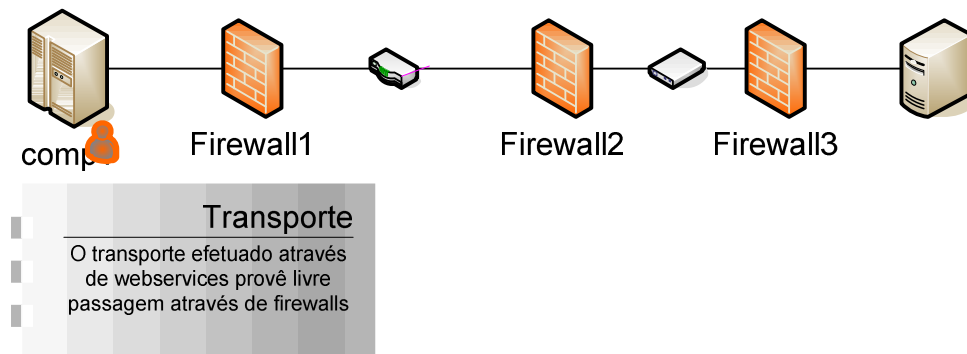


Ilustração 8 - Camada de transporte

O transporte através de web services permite, por não fazer uso de um protocolo binário, livre acesso através de diversos dispositivos de segurança e através de diversas redes diferentes (figura 07).

Como os serviços a serem gerenciados, que no caso são os web services possuem características que facilitam seu transporte, foi proposto que o transporte das informações coletadas aconteça através de web services também, pois pode ser aproveitada a infra-estrutura onde o serviço está alojado, por exemplo, um container servlet como tomcat.

Entidade Gerente

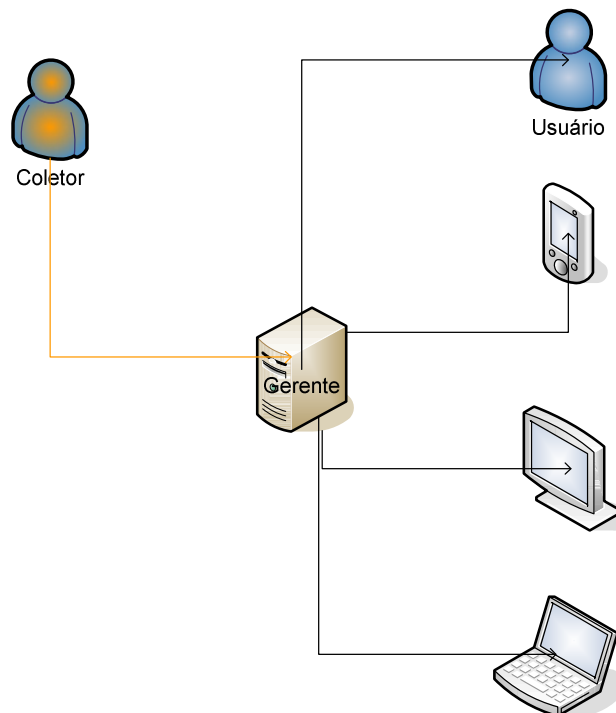


Ilustração 9 - Responsabilidades da entidade gerente

È a entidade responsável por coletar as métricas é solicitar o seu armazenamento por parte da entidade de persistência.

Esta entidade também acumula a responsabilidade de fazer as comparações entre as métricas e o SLA especificado para o serviço. Assim como também tem a responsabilidade de prover acesso por aos reports do processo de gerencia do nível de serviço.

Entidade Repositório

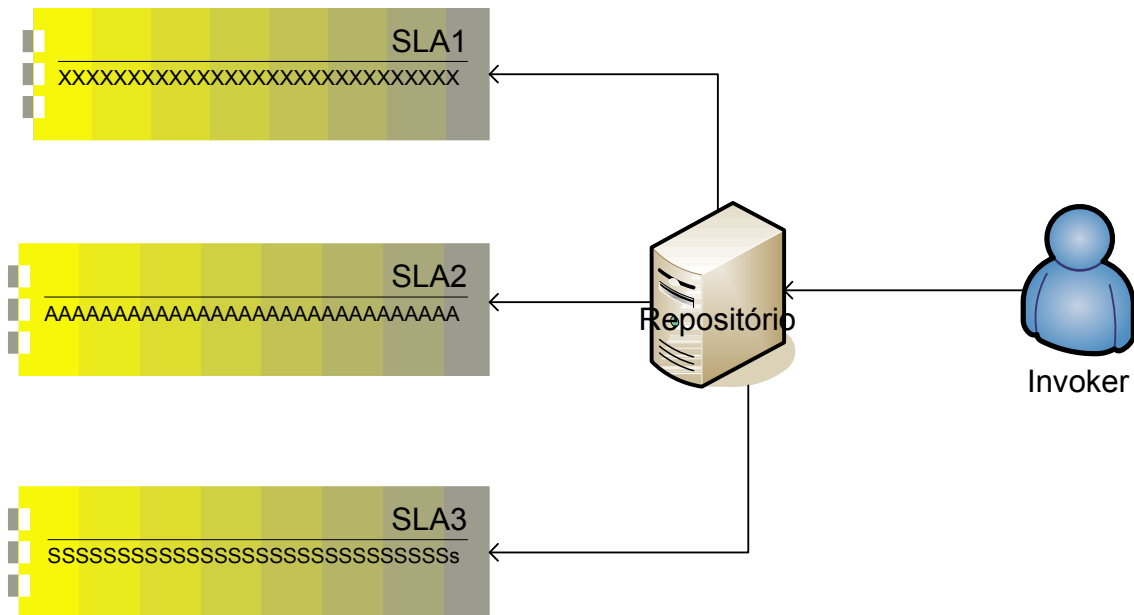


Ilustração 10 - O repositório de SLAs

O repositório de SLAs tem a responsabilidade de armazenar os SLAs especificados. Para sua implantação pode ser utilizada qualquer tecnologia que atenda ao requisito de persistir os dados. Por exemplo: sistema de arquivos, banco de dados relacional, banco de dados XML.

Deve prover meios para a obtenção dos SLA especificados utilizando como chave o serviço.

Entidade de Persistência

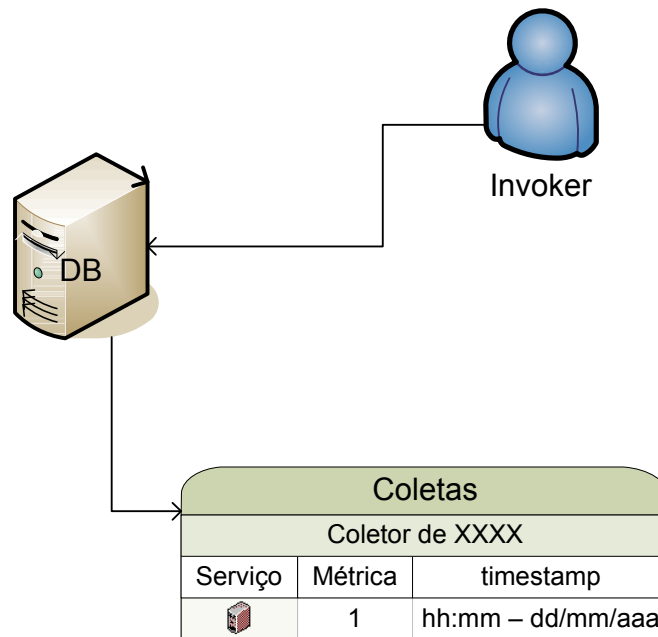


Ilustração 11 - Entidade de Persistência

Garante a persistência das métricas coletas.

É indicado o uso de um banco de dados relacional, pois o mesmo é amplamente utilizado em outras áreas da computação e o seu uso facilita uma transformação mais fácil do modelo operacional para um modelo analítico que facilitaria consultar mais elaboradas.

Entidade de Registro

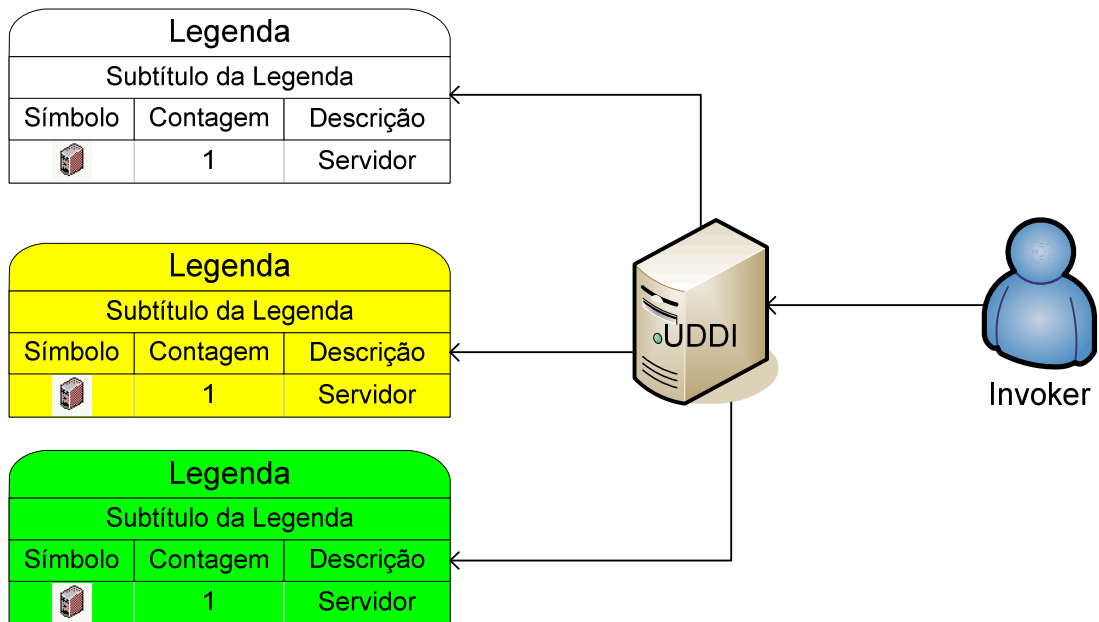


Ilustração 12 - Entidade de registro

Fornecer um catálogo com os serviços disponíveis.

Parte fundamental na utilização de uma arquitetura orientada a serviços, pois permite ao service requestor abstrair a localização real onde seu problema está sendo resolvido.

Conclusão

O gerenciamento de serviços se mostra necessário e deve ser implantado de forma madura nas empresas tanto internamente como externamente no fornecimento de serviços. O SLA permite estabelecer um acordo que visa um fornecimento adequado tanto ao cliente como ao fornecedor, estabelecendo as necessidades do cliente assim como também expondo os casos onde o SLA não se aplica.

Este trabalho serviu para estabelecer guia para quem desejar ter uma idéia inicial de como implementar uma ferramenta para gerenciar o nível de serviço em sua empresa. Não abordando pontos como persistência dos dados, pois o modo como este acontece não influencia diretamente na gerencia dos serviços.

O maior foco da proposta foi na tentativa de aproveitar os recursos que os web services tem e utiliza-los também no gerenciamento do seu nível de serviço.

Referências

[1] CHAPPELL, David , JEWEL, Tyler. **Java Web Services**. O'Reilly. First Edition March 2002

[2]Axis User Guide - Webservices Axis - <http://ws.apache.org/axis/java/user-guide.html>

[3]On service level agreements for IP networks Martin, J.; Nilsson, A. INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE Volume 2, Date: 2002, Pages: 855 - 863 vol.2 Digital Object Identifier 10.1109/INFOCOM.2002.1019332

[4] Analytical definition of SLA parameters in a video-on-demand service. Martin, I.V.; Juan, F.; Alins-Delgado, J.; Aguilar-Igartua, M.; Mata-Diaz, J. Networks, 2004. (ICON 2004). Proceedings. 12th IEEE International Conference on Volume 1, Date: 16-19 Nov. 2004, Pages: 427 - 431 vol.1 Digital Object Identifier 10.1109/ICON.2004.1409190

[5]Service management architecture and information model for next generation network with dynamic service level agreement management Yin, Z.M.; Yang, F.C.; Liu, Y.Z. Networks, 2004. (ICON 2004). Proceedings. 12th IEEE International Conference on Volume 1, Date: 16-19 Nov. 2004, Pages: 437 - 441 vol.1 Digital Object Identifier 10.1109/ICON.2004.1409192

[6]Specification of service level agreements, clarifying concepts on the basis of practical research .Bouman, J.; Trienekens, J.; van der Zwan, M. Software Technology and Engineering Practice, 1999. STEP '99. Proceedings Date: 1999, Pages: 169 – 178 Digital Object Identifier 10.1109/STEP.1999.798790

[7] Service Level Agreement Negotiation: A Theory-based Exploratory Study as a Starting Point for Identifying Negotiation Support System Requirements Demirkan, H.; Goul, M.; Soper, D.S. System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on Date: 03-06 Jan. 2005, Pages: 37b - 37b Digital Object Identifier 10.1109/HICSS.2005.540

[8] A Differentiated Service Mechanism Considering SLA for Heterogeneous Cluster Web Systems Changhoon Kang; Kiejun Park; Sungsoo Kim Software Technologies for Future Embedded and Ubiquitous Systems, 2006 and the 2006 Second International Workshop on Collaborative Computing, Integration, and Assurance. SEUS 2006/WCCIA 2006. The Fourth IEEE Workshop On Date: 27-28 April 2006, Pages: 245 – 250 Digital Object Identifier 10.1109/SEUS-WCCIA.2006.3

[9] Specifying and monitoring guarantees in commercial grids through SLA Sahai, A.; Graupner, S.; Machiraju, V.; van Moorsel, A. Cluster Computing and the Grid, 2003. Proceedings. CCGRID 2003. 3rd IEEE/ACM International Symposium on Date: 12-15 May 2003, Pages: 292 – 299 Digital Object Identifier 10.1109/CCGRID.2003.1199380

[10] On architecture for SLA-aware workflows in grid environments Quan, D.M.; Kao, O. Advanced Information Networking and Applications, 2005. AINA 2005. 19th

International Conference on Volume 1, Date: 28-30 March 2005, Pages: 287 - 292 vol.1 Digital Object Identifier 10.1109/AINA.2005.260

[12] Requirements for service level agreement management Marilly, E.; Martinot, O.; Betge-Brezetz, S.; Delegue, G. IP Operations and Management, 2002 IEEE Workshop on Date: 2002, Pages: 57 – 62 Digital Object Identifier 10.1109/IPOM.2002.1045756

[13] Automatic SLA management in SLA-aware architecture D' Arienzo, M.; Esposito, M.; Romano, S.P.; Ventre, G. Telecommunications, 2003. ICT 2003. 10th International Conference on Volume 2, Date: 23 Feb.-1 March 2003, Pages: 1402 - 1406 vol.2 Digital Object Identifier 10.1109/ICTEL.2003.1191639

[14] Risk management in availability SLA Clemente, R.; Bartoli, M.; Bossi, M.C.; D'Orazio, G.; Cosmo, G. Design of Reliable Communication Networks, 2005. (DRCN 2005). Proceedings.5th International Workshop on Date: 16-19 Oct. 2005, Pages: 8 pp. Digital Object Identifier 10.1109/DRCN.2005.1563900

[15] Efficient resource allocation for SLA based wireless/wireline interworking Cheng, Y.; Zhuang, W.; Leon-Garcia, A.; Hu, R.Q. Broadband Networks, 2005 2nd International Conference on Date: 3-7 Oct. 2005, Pages: 561 - 570 Vol. 1 Digital Object Identifier 10.1109/ICBN.2005.1589661

[16] Managing eBusiness on demand SLA contracts in business terms using the cross-SLA execution manager SAM Buco, M.; Rong Chang; Luan, L.; Ward, C.; Wolf, J.; Yu, P. Autonomous Decentralized Systems, 2003. ISADS 2003. The Sixth International Symposium on Date: 9-11 April 2003, Pages: 157 - 164

[17] The structure and management of service level agreements in

networks Bouillet, E.; Mitra, D.; Ramakrishnan, K.G. Selected Areas in Communications, IEEE Journal on Volume 20, Issue 4, Date: May 2002, Pages: 691 – 699 Digital Object Identifier 10.1109/JSAC.2002.1003036

[18] Fresco: a Web services based framework for configuring extensible

SLA management systems Ward, C.; Bucu, M.J.; Chang, R.N.; Luan, L.Z.; So, E.; Chunqiang Tang .Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International

Conference on Date: 11-15 July 2005, Pages: 237 - 245 vol.1 Digital Object Identifier 10.1109/ICWS.2005.141

[19] The Study on SLA Metrics and System Architecture for IP TV

Seong Hwan Kim; Lee, G.H. Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference Volume 3, Date: 20-22 Feb. 2006, Pages: 2114 – 2117

[20]CHAPPEL, David A.; JEWELL, Tyler. **Java Web Services**: Using Java in Service-Oriented Architecture. Usa: O’rielly, 2002. 276 p.

[21] STURM, Rick. Service Level Management :Fundamentos do nível de serviço. Rick Sturm, Wayne Morris, Mary Jander; tradução de Teresa Cristina Félix de Souza – Rio de Janeiro : Campus, 2001.

[22] Hoekstra, Angeli; Conradie, Nicolette. CobiT, ITIL and ISO17799

How to use them in conjunction. PriceWaterHouseCooper.

Anexos

Relatórios

Coletores

Os coletores são responsáveis pela extração de informações gerencias nos computadores remotos.

Todos os coletores devem estender uma subclasse chamada ColetorAbstrato

Classe ColetorAbstrato

```
private String relatório;  
  
private Date data;  
  
private Time tempo;  
  
private String hostname,ip;
```

Atributos

Identificadores na linha do tempo:

tempo – o horário em que foi recolhida a amostra;

data – a data em que foi recolhida a amostra;

Identificadores na rede:

hostname – nome do equipamento onde ocorre a extração;

ip – endereço ip do equipamento onde ocorre a extração;

Relatório sumarizado

relatório – responsável pelo relatório sumarizado da coleta. É estruturada da seguinte forma:

[Nome do Coletor]

[Atributos específicos]

[Atributos identificadores na linha do tempo]

[Atributos identificadores na rede]

Detalhamento:

Nome do coletor – o nome atribuído a classe usada na coleta, usado para identificar o coletor usado.

Atributos específicos – mensurações específicas de cada coletor implementado.

Atributos identificadores na linha do tempo – identificam o instante de tempo da amostra.

Atributos identificadores na rede – identificam o equipamento.

Observação: atributo e valor são separados por “:”.

Justificativa

Identificadores na linha do tempo

A coleta pode acontecer de modo assíncrono em diversos dispositivos, o que nos traz a necessidade de ter uma marca na linha do tempo para a coleta.

Esta abordagem permite que utilizemos o histórico das amostras para análises estatísticas, previsões entre outras.

Transporte

Tecnologia

Webservice usando Axis, uma iniciativa da Apache Foundation implementada em Java que permite um desenvolvimento rápido e fácil de serviços web (Webservices).

Proposta

Implementar uma camada de transporte que possibilite o tráfego de informações gerencias da rede de maneira fácil e transparente para os outros componentes. Fazer com que as mensagens trocadas sejam em linguagem de alto nível, próximas a linguagem natural, diminuindo a necessidade do uso de interpretadores para essas informações.

Usar um protocolo que possa trafegar em um canal seguro, passando assim a responsabilidade de controle da segurança para a camada de rede o que conceitualmente é mais correto, pois, a segurança do trafego da informação deve estar no canal onde ela trafega e não na própria informação ou nos interlocutores, essa decisão de projeto não exclui as demais abordagens.

Serviço de transporte simples

Um serviço de transporte simples foi implementado para validação da proposta. Este serviço oferece dois métodos básicos, um com a listagem das coletas, obtem a lista de coletas e encapsula em um envelope SOAP; e

outro é o método de listagem de relatórios, obtém a lista das coletas e o relatório de cada uma e encapsula em um envelope SOAP.

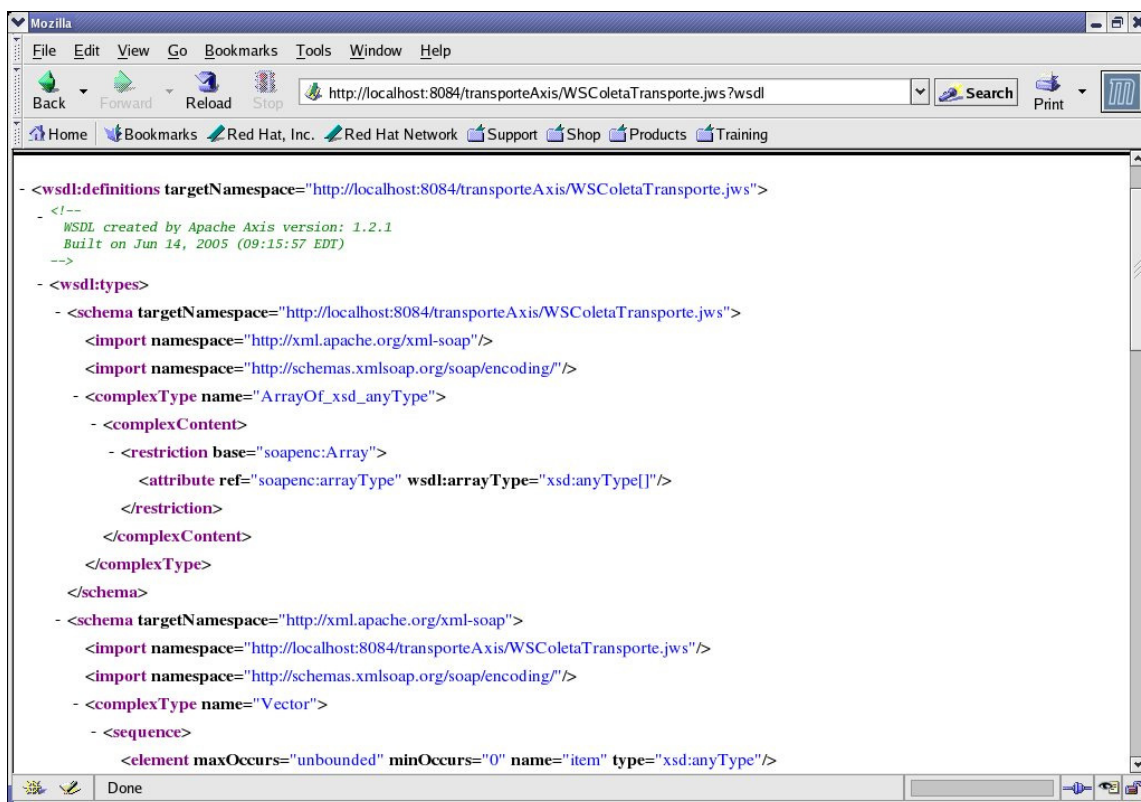
Justificativa

O Axis possibilita um rápido desenvolvimento e uma baixa curva de aprendizado o que se mostra como uma vantagem em nosso cenário onde o foco não são aplicações distribuídas mas a comunicação entre entidades responsáveis pela gerencia, podendo ser esses atores humanos ou entidades computacionais.

O protocolo SOAP por ser textual permite uma passagem mais facilitada através de dispositivos de segurança na rede (exemplo: firewalls). O SOAP é uma extensão da linguagem XML que nos permite estruturar os dados de maneira hierárquica aproximando um pouco mais a maneira de estrutura da informação aos modelos mentais propostos pela cognição.

Resultados parciais

Implementação de um webservice de transporte usando Axis de acordo com a proposta apresentada. O WSDL que descreve o serviço, e que esta anexo, foi gerado a partir do próprio Axis.



```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://localhost:8084/transporteAxis/WSColetaTransporte.jws">
  <!--
    WSDL created by Apache Axis version: 1.2.1
    Built on Jun 14, 2005 (09:15:57 EDT)
  -->
  <wsdl:types>
    <schema targetNamespace="http://localhost:8084/transporteAxis/WSColetaTransporte.jws">
      <import namespace="http://xml.apache.org/xml-soap"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="ArrayOf_xsd_anyType">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:anyType[]"/>
          </restriction>
        </complexContent>
      </complexType>
    </schema>
    <schema targetNamespace="http://xml.apache.org/xml-soap">
      <import namespace="http://localhost:8084/transporteAxis/WSColetaTransporte.jws"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="Vector">
        <sequence>
          <element maxOccurs="unbounded" minOccurs="0" name="item" type="xsd:anyType"/>
        </sequence>
      </complexType>
    </schema>
  </wsdl:types>
</wsdl:definitions>
```

Figura 01 – WSDL gerado pelo axis, visualização no mozilla

Foi usado o browser Mozilla para acessar o webservice e fazer os testes.

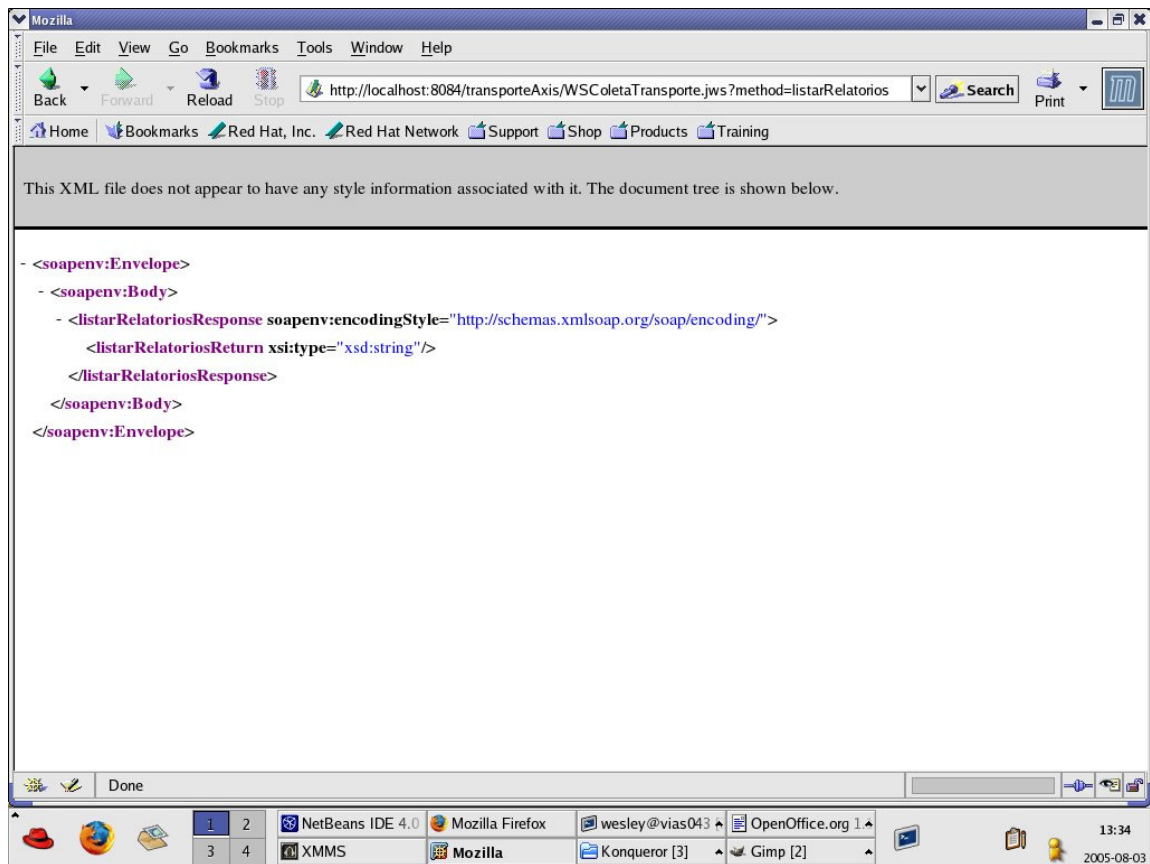


Figura 02 – Método ListarRelatorios, visualização no mozilla

Painel

Gerencia de Redes Distribuídas – Estudo do uso do Apache/Axis na camada de transporte.

Wesley dos Reis Bezerra (UFSC)

wesley@inf.ufsc.br

Carlos Becker Westphall (UFSC)

westphal@lrg.ufsc.br

1. Introdução

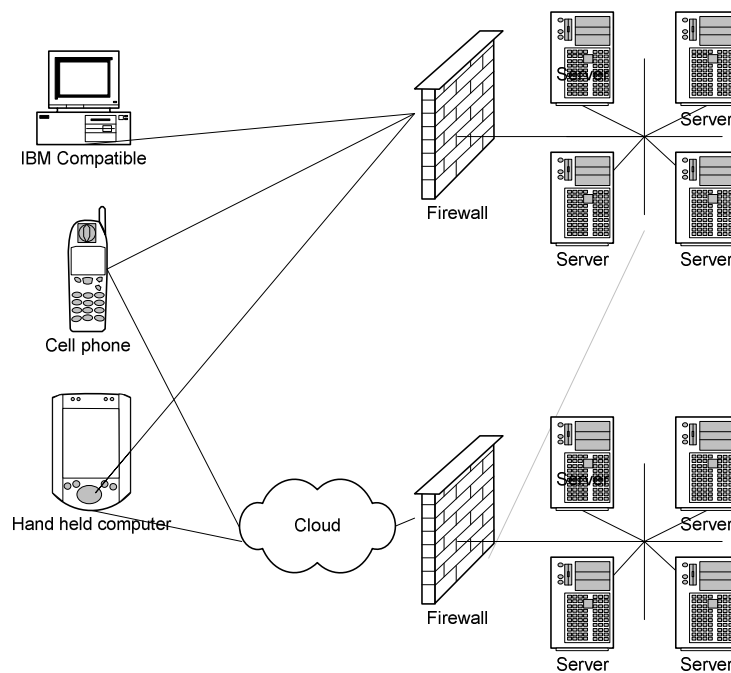


Figura 01 – Rede Distribuída

A internet tem aberto portas para empresas se lançarem ao redor do mundo em busca de conhecimento, mas as vezes essas portas podem ser uma ameaça a segurança dos dados na empresa. Como solução tem-se hoje em dia utilizado uma ferramenta de proteção que se tornou um item básico de segurança, o firewall. O firewall protege a rede de acessos indevidos agindo através de filtragem ou bloqueio de portas. Uma abordagem muito utilizada é bloquear todas as portas por padrão e liberar somente as portas baixas de serviços principais. Isso constitui um problema quando falamos em gerencia de redes distribuída, que usa coletores remotos e que precisam transmitir informação pela rede, as vezes através de portas altas utilizadas pela aplicação desenvolvida. Outro problema que é bem comum é que as portas do SNMP, por motivos de segurança, são barradas no firewall. A utilização de outros protocolos de computação distribuída para este caso também não é possível pois portas que não são serviço são bloqueadas.

Os webservises são serviços web que usam um protocolo textual (SOAP) na sua comunicação e utilizam portas de serviços padrões como http (80) e SMTP(25) que são abertas ou somente filtradas na maioria dos servidores. Configura-se assim uma possibilidade de uso desta facilidade de trafego de mensagens SOAP no contexto da gerencia de redes distribuída. Esta solução agrega mais valores além de apenas um tráfego facilitado através de firewall, o protocolo SOAP por ser textual e por ser uma extensão do XML traz as trocas de mensagens uma linguagem mais inteligível por nós humanos e que possibilita uma aproximação do administrador de dados do equipamento, pois a mensagem não precisa utilizar protocolos de baixo nível.

2. Objetivos

Estudo da aplicação de webservices na gerencia de redes de computadores.

Implementação de coletores padrões para validação da proposta.

Implementação de serviços de transporte para trafego das informações de gerencia coletadas para validação da proposta.

Trazer a troca de mensagens de gerencia a um nível mais inteligível ao ser humano ou mesmo a agentes computacionais.

Usar o conceito de desenvolvimento em camadas possibilitando a abstração por parte dos outros componentes quanto ao transporte.

3.1. Webservices

“A web service is a piece of business logic, located somewhere on the Internet” [1]. Um serviço web pode estar alocado em qualquer parte do mundo e pode ser uma peça dentro do seu negócio ou mesmo a maior parte do mesmo. Na nossa arquitetura ele é a peça chave na camada de transporte permitindo um trafego de dados baseado em um protocolo textual e através de portas de livre passagem pelos firewalls

3.2. Webservices com Apache Axis

“Axis is essentially a *SOAP engine* -- a framework for constructing SOAP processors such as clients, servers, gateways, etc”[2]. O Axis prove uma ferramenta de controle de requisições para os serviços nele publicados. Proporciona um desenvolvimento de serviços simples em poucos minutos.

4. Metodologia

Estudo formal sobre a utilização do Apache/Axis. Fazer um estudo da tecnologia levando em conta seus pontos fortes e fracos ao ser aplicada no escopo da gerencia de redes.

Montar uma arquitetura para validação da proposta. Criar uma estrutura com coletores padrões para extração de dados de gerencia a serem transportados pelos webservices.

Definir serviço de transporte. Definir a assinatura dos métodos disponíveis através da interface web. Implementar um protótipo para validação.

A modelagem de um modelo de persistência para os dados não será abordada.

5. Resultados

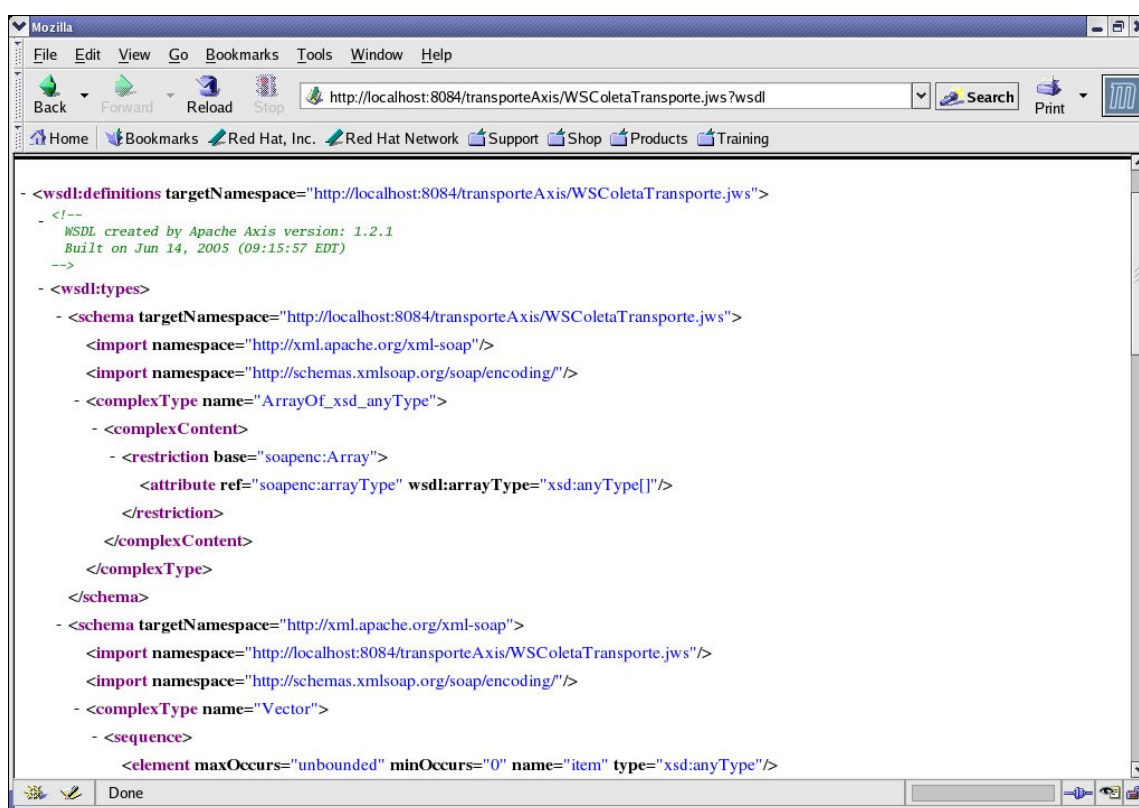
5.1. Conclusões do estudo

O Axis possibilita um rápido desenvolvimento e uma baixa curva de aprendizado o que se mostra como uma vantagem em nosso cenário onde o foco não são aplicações distribuídas mas a comunicação entre entidades responsáveis pela gerencia, podendo ser esses atores humanos ou entidades computacionais.

O protocolo SOAP por ser textual permite uma passagem mais facilitada através de dispositivos de segurança na rede (exemplo: firewalls). O SOAP é uma extensão da linguagem XML que nos permite estruturar os dados de maneira hierárquica aproximando um pouco mais a maneira de estrutura da informação aos modelos mentais propostos pela cognição.

5.2. Protótipo

Implementação de um webservice de transporte usando Axis de acordo com a proposta apresentada. O WSDL que descreve o serviço, e que esta anexo, foi gerado a partir do próprio Axis.



```
<?xml version="1.0" encoding="UTF-8" ?>
<wsc:definitions targetNamespace="http://localhost:8084/transporteAxis/WSColetaTransporte.jws">
  <!--
  WSDL created by Apache Axis version: 1.2.1
  Built on Jun 14, 2005 (09:15:57 EDT)
  -->
  <wsc:types>
    <schema targetNamespace="http://localhost:8084/transporteAxis/WSColetaTransporte.jws">
      <import namespace="http://xml.apache.org/xml-soap"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="ArrayOf_xsd_anyType">
        <complexContent>
          <restriction base="soapenc:Array">
            <attribute ref="soapenc:arrayType" wsdl:arrayType="xsd:anyType[]"/>
          </restriction>
        </complexContent>
      </complexType>
    </schema>
    <schema targetNamespace="http://xml.apache.org/xml-soap">
      <import namespace="http://localhost:8084/transporteAxis/WSColetaTransporte.jws"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      <complexType name="Vector">
        <sequence>
          <element maxOccurs="unbounded" minOccurs="0" name="item" type="xsd:anyType"/>
        </sequence>
      </complexType>
    </schema>
  </wsc:types>
  <wsc:binding targetNamespace="http://localhost:8084/transporteAxis/WSColetaTransporte.jws">
    <wsc:operation name="getTransporte">
      <input wsdl:required="true" encoding="http://schemas.xmlsoap.org/soap/encoding/" type="xsd:string"/>
      <output wsdl:required="true" encoding="http://schemas.xmlsoap.org/soap/encoding/" type="xsd:string"/>
    </wsc:operation>
  </wsc:binding>
  <wsc:service name="WSColetaTransporte" targetNamespace="http://localhost:8084/transporteAxis/WSColetaTransporte.jws">
    <wsc:binding value="getTransporte" targetNamespace="http://localhost:8084/transporteAxis/WSColetaTransporte.jws"/>
  </wsc:service>
</wsc:definitions>
```

Figura 02 – wsdl gerado pelo Axis – visualização no Mozilla Firefox.

Foi usado o browser Mozilla para acessar o webservice e fazer os testes

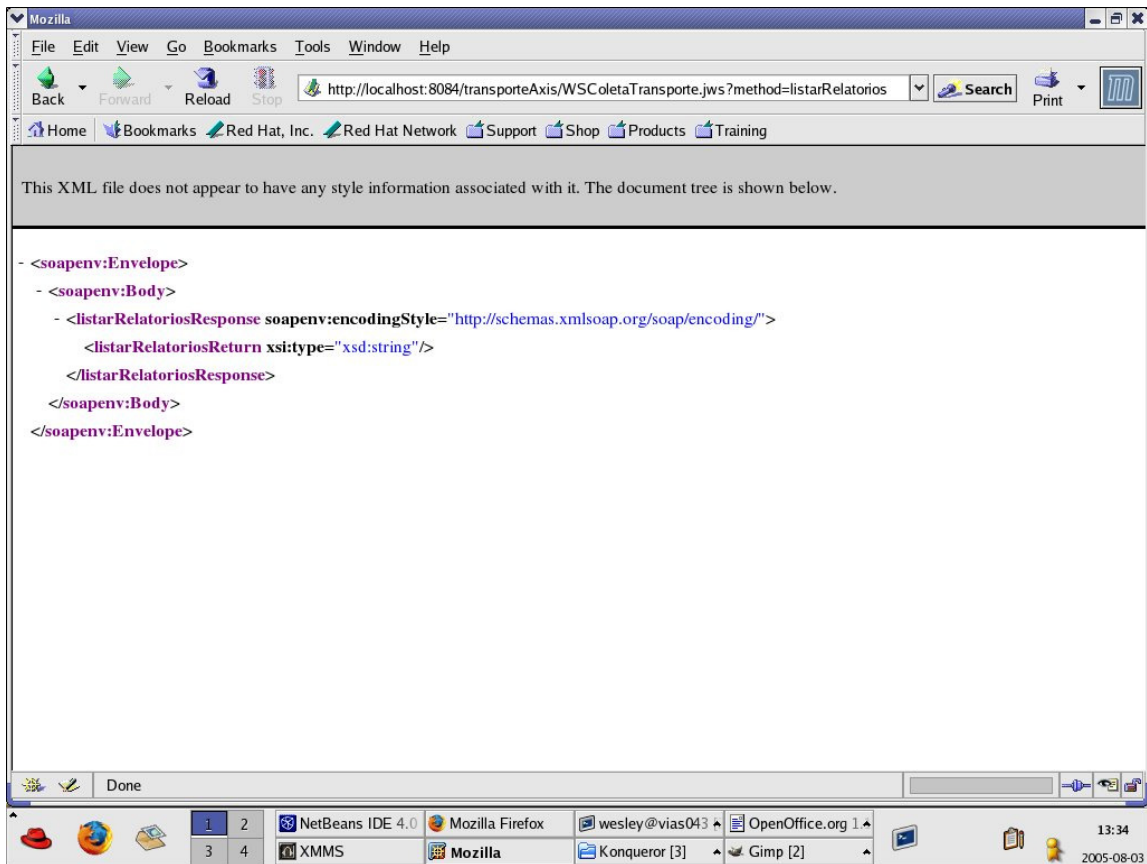


Figura 03 – envelope SOAP gerado pelo Axis – Método listar relatórios.

Foi implementado um serviço de transporte, para teste e validação, usando Axis e como ferramenta de acesso ao webservice foi utilizado o browser Mozilla Firefox.

