

PROPOSTA DE UM PADRÃO PARA A COMUNICAÇÃO DE DADOS EM SISTEMAS DE TELEMETRIA BASEADO EM WEB SERVICES

José Morelli Neto

neto@univali.br

Rodrigo Becke Cabral

cabral@univali.br

Universidade do Vale do Itajaí - UNIVALI

Rua Uruguai, 458 - 88302-202 - Itajaí - SC - Brasil

RESUMO

Telemetria é a técnica que mede quantidades, transmitindo os resultados para um centro de controle que é responsável pela interpretação, apresentação e/ou armazenamento dos valores medidos. Os meios de transmissão de dados utilizados podem ser muito variados, abrangendo desde linhas telefônicas convencionais até a transmissão por meio da Internet. A forma de transação dos dados de telemetria tem sido padronizada por diversas organizações com o objetivo de tornar as indústrias que fabricam equipamentos de telemetria compatíveis entre si. Com o advento de novas tecnologias, existe uma lacuna na padronização das transações de telemetria em nível de camada de aplicação. Este trabalho visa preencher esta lacuna com a proposta de um protocolo de transações de dados de telemetria através da tecnologia de Web Services. Essa normatização tem aplicação imediata no sistema de rastreamento de embarcações arrendadas do Centro de Ciências Tecnológicas da Terra e do Mar da Universidade do Vale do Itajaí.

ABSTRACT

Telemetry is the technique that measure quantities, transmitting the results to a control center responsible for the interpretation, presentation and/or storage of these measurements. The modes of data transmission can vary, ranging from conventional telephone lines to the Internet. How the transactions must occur has been standardized by numerous organizations, with the objective of turning fully compliant the industry that manufactures telemetry equipments. With the new technologies, there is room for creating more standards for telemetry transactions in software application layer. This work fullfills this need by proposing a protocol for telemetry data transactions using the technology of Web Services. This standardization has immediate application in the system for tracking vessels of the Centro de Ciências Tecnológicas da Terra e do Mar of the Universidade do Vale do Itajaí.

1. Introdução

Com o desenvolvimento da tecnologia e de sua portabilidade para sistemas embarcados, tem-se observado uma crescente demanda de comunicação entre sensores e os sistemas de armazenamento dos dados coletados. Para que se possa efetuar essa coleta remotamente em tempo real, são utilizadas técnicas de telemetria, que é o processo pelos quais os parâmetros de

um objeto são medidos e os resultados transmitidos para uma estação distante, que os armazena e apresenta para análise.

O processo da telemetria envolve agrupar medidas (tais como temperatura, velocidade e posição geográfica) em uma estrutura que possa ser transmitida em um único fluxo de dados. Uma vez recebido, o fluxo de dados é separado nos componentes medidos originalmente para que possa então ser analisado.

É importante ressaltar o sistema RASTRO, que foi utilizado como estudo de caso, sendo o primeiro sistema a implementar o padrão proposto. O RASTRO é um sistema acessado via *web* que apresenta as informações na forma de mapas de navegação contendo as últimas posições e os dados traçados de uma ou várias embarcações. Por trás desse sistema, encontra-se um módulo que é responsável por receber os dados de empresas de rastreamento, que acompanham via satélite a localização das embarcações na costa brasileira.

Em um sistema de telemetria, aquisição dos dados inicia-se quando os sensores medem uma quantidade de atributos físicos e transformam essas medidas em uma unidade de valor planejado. Esses sensores estão ligados a um condicionador de sinal, que fornece energia para que eles funcionem ou modifiquem o tipo do sinal para o próximo estágio da aquisição. Um multiplexador é utilizado para capturar os valores medidos em série de forma análoga e emití-los para uma saída (output) como um único fluxo de dados (L-3 COMMUNICATIONS, 2003).

A telemetria permite a coleta de dados em locais de difícil acesso ou perigosos para os seres humanos. Esses pontos de coleta podem ser próximos dos centros de controles, como normalmente aplicados, podem estar a centenas de quilômetros (IN4MA REMOTE MONITORING, 2003). Os sistemas de telemetria podem ser caracterizados de várias formas. Como exemplo serão citados dois modelos: a telemetria sem fio e a telemetria via Internet.

De nada adiantaria coletar dados se não fosse possível armazená-los e tratá-los de forma que os tornassem informações úteis para alguma aplicação. Dessa forma, poder-se-ia propor uma solução em que esses dados coletados fossem transmitidos utilizando uma tecnologia bastante inovadora para troca de informações: os Web Services.

A idéia dos Web Services é criar componentes de software que podem ser ativados a distância via Internet, trocando informações pela rede na forma de arquivos no padrão XML (Extensible Markup Language). Cada componente se autodescreve e eles podem ser listados em diretórios para que sejam encontrados mais facilmente na Internet (GREGO, 2002).

Atualmente são vários os sistemas de telemetria disponíveis no mercado, porém, esses sistemas possuem um protocolo proprietário, ou fechado para a troca de informações, como é o caso do Padrão NMEA (National Marine Electronics Association) que serve para a troca de informações entre equipamentos marítimos (NMEA, 2003). Este projeto pretende abordar de forma diferente a troca de informações utilizando para isso a tecnologia de Web Service. Este modelo será incorporado ao Sistema Rastro, mostrando, dessa forma, que além de inovador, é aplicável a sistemas já existentes.

Também é importante ressaltar que essa troca de informações será efetuada de forma segura, o que significa que esse modelo poderá ser utilizado na esfera comercial. Para assegurar a troca dos dados, deverão ser utilizadas técnicas como o HTTPS (Secure Hyper Text Transfer Protocol), que apresenta o uso de HTTP (Hyper Text Transfer Protocol) encapsulado sobre SSL (Secure Sockets Layer) permitindo o envio e recebimento de mensagens cifradas. Também será utilizada uma autenticação prévia, permitindo o acesso ao Web Service somente a sistemas credenciados.

Várias organizações têm definido Web Service de formas diferentes, mas em sua essência Web Service é um componente de software que interage com outro componente de software distinto através de protocolos de rede conhecidos. Com o uso de XML, o acesso aos serviços tornou-se independente das suas implementações. Assim, um Web Service oferece tanto o benefício do componente de software quanto do uso da Web para a transferência dos dados (AYALA et al. 2002).

Os componentes representam uma funcionalidade implementada em uma "caixa-preta", a qual pode ser reutilizada sem a preocupação de como o serviço foi implementado. As aplicações acessam os Web Services através de protocolos padrão, como HTTP e SOAP (Simple Object Access Protocol) e transferem os dados no formato XML, diferentemente das tecnologias anteriores, que utilizam protocolos específicos (DCOM - Distributed Component Object Model; RMI - Remote Method Invocation; ou IIOP - Internet Inter-Operability Protocol) (OPV, 2003).

Conforme apresentado na Figura 1, o Web Service pode se registrar em um repositório central, onde softwares clientes que necessitam dos serviços oferecidos pelo Web Service poderão pesquisá-lo. Uma vez localizado, o software cliente receberá uma referência para o serviço encontrado, podendo usufruir dos serviços apenas executando os vários métodos implementados com a ajuda das interfaces públicas. Assim, cada serviço precisa publicar sua interface para que os softwares clientes a utilizem (AYALA et al., 2002).

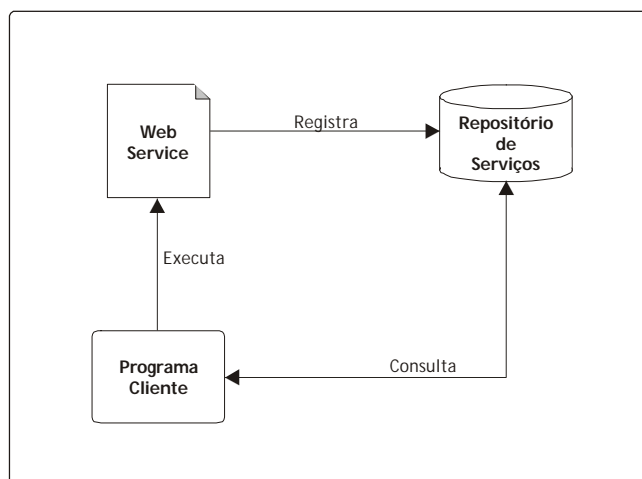


Figura 1: Funcionamento Simplificado de um Web Service.

Fonte: Ayala et al. (2002).

Austin et al. (2002) define um Web Service como sendo um sistema identificado por um URI (Uniform Resource Identifiers) que publica uma interface e um meio de comunicação lógico que são definidos e descritos usando XML. Essa definição pode ser descoberta por outros sistemas, que então podem interagir com o Web Service conforme descrito pela sua definição, usando mensagens baseadas em XML que são transportadas sob protocolos de Internet.

1.1 UDDI

O UDDI (Universal Description, Discovery and Integration) é a definição de um conjunto de serviços apoiando a descrição e descoberta de: (i) negócios, organizações e outros provedores de Web Services; (ii) Web Services disponibilizados pelos provedores; e (iii) interface técnica que pode ser utilizada para acessar esses serviços. Baseada em um conjunto comum de padrões industriais, incluindo HTTP, XML, XML Schema e SOAP, a UDDI fornece a interoperabilidade e a infra-estrutura necessárias para softwares baseados em Web Services de ambos os ambientes: serviços disponíveis publicamente e serviços disponíveis apenas internamente dentro da organização (BELLWOOD et al., 2002).

1.1.1 Função da UDDI

Segundo Ayala et al. (2002), a função da UDDI é fornecer os seguintes serviços básicos:

Publicar: Trata de como o provedor de Web Service registra a si mesmo e seu serviço no UDDI;

Descobrir: Trata de como o cliente pode localizar o Web Service desejado;

Ligar: Trata de como o cliente pode conectar e utilizar o Web Service localizado.

As associações que disponibilizam esses serviços e hospedam o registro global UDDI são chamadas de Operadores UDDI (UDDI Operators). Elas são responsáveis por sincronizar a informação de registro, sendo que a sincronização das informações de registro é chamada de replicação (replication). As empresas publicam seus serviços e as informações relacionadas utilizando esses operadores. Os dois operadores mais conhecidos são a Microsoft, que fornece o serviço de UDDI em <http://uddi.microsoft.com>, e a IBM, que fornece o mesmo tipo de serviço em <http://uddi.ibm.com> (AYALA et al., 2002).

1.2 WSDL

1.2.1 Introdução

Segundo Ayala et al. (2002), WSDL (Web Services Description Language) é um formato XML para descrever a interface de um Web Service definindo um conjunto de operações suportadas pelo servidor e o formato que os clientes necessitam utilizar quando requisitarem um serviço. Um arquivo WSDL atua como sendo um contrato entre o cliente e o serviço, pois efetiva a comunicação entre ambas as partes. O cliente requisitará o serviço enviando uma solicitação propriamente formatada em SOAP.

Cerami (2002) ressalta que o WSDL é uma plataforma independente de linguagem e, inicialmente, é utilizada (embora não exclusivamente) para descrever um serviço SOAP. Esse autor também enfatiza que, com o uso do WSDL, o cliente pode localizar um Web Service invocando qualquer uma das funções públicas disponíveis. Dessa forma, pode-se dizer que o WSDL representa uma base fundamental para a arquitetura de Web Service, pois disponibiliza uma linguagem comum para descrever os serviços e a plataforma de modo a haver a integração automática desses serviços.

Exemplificando o funcionamento do WSDL, pode-se imaginar a criação de um Web Service que ofereça as cotações na bolsa de valores para os compradores. Dessa forma, será necessária a criação de um arquivo WSDL que descreverá o serviço. Esse arquivo será alocado no provedor do Web Service ou publicado em um repositório UDDI. Os clientes interessados no serviço, primeiramente, terão que obter uma referência desse arquivo utilizando uma pesquisa ao repositório ou pegando-a diretamente do provedor. Após compreender a referência, deverá ser criado um Web Service solicitante que invocará o serviço fazendo uma requisição baseada em SOAP utilizando HTTP post. O servidor validará a requisição executando-a ou passando-a para o serviço adequado. O resultado, que é o preço das ações na bolsa de valores, é então retornado para o cliente como uma resposta no formato SOAP. Na

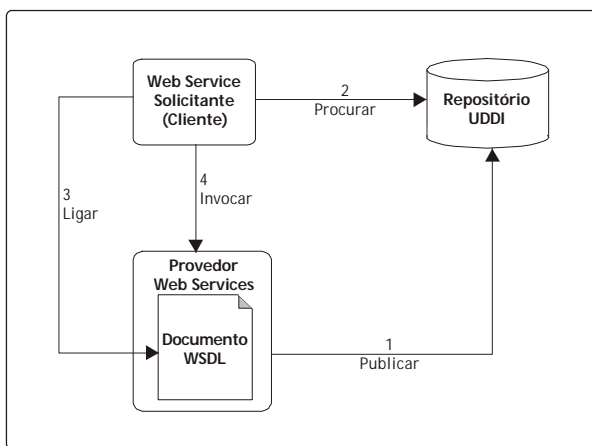


Figura 2, pode-se observar o funcionamento do WSDL (AYALA et al., 2002).

Conforme Cerami (2002), o WSDL apresenta quatro partes importantes de dados:

- ?? Informação da interface descrevendo todas as funcionalidades disponíveis publicamente;
- ?? Informações de tipos de dados para todas as mensagens de requisição ou mensagens de respostas;
- ?? Informação obrigatória sobre o protocolo de transporte a ser utilizado;
- ?? Informações de endereçamento para localizar um serviço especificado.

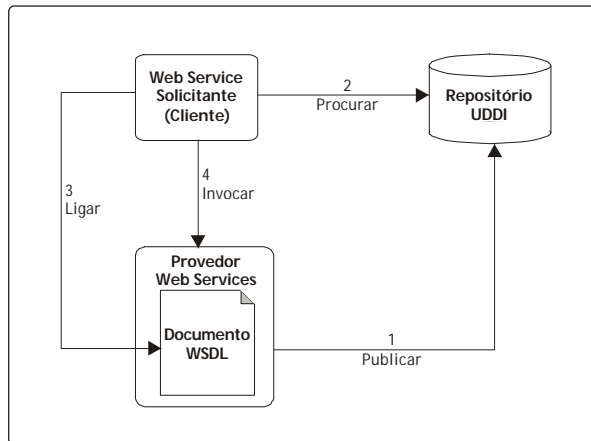


Figura 2: Funcionamento do WSDL.

Fonte: Ayala et al. (2002, p.40).

1.2.2 Estrutura de um documento WSDL

A estrutura de um documento WSDL é implementada utilizando XML Schema em conjunto com seus tipos suportados. A especificação para o documento WSDL fornece um conjunto com seis definições de elementos XML. O elemento raiz ou básico no documento WSDL é o elemento definitions. Os seis elementos aninhados são: types, message, portType, binding, service e port. (AYALA et al., 2002). A Figura 3 apresenta de forma simples a estrutura de um documento WSDL contendo todos seus elementos, os quais são detalhados nas subseções a seguir.

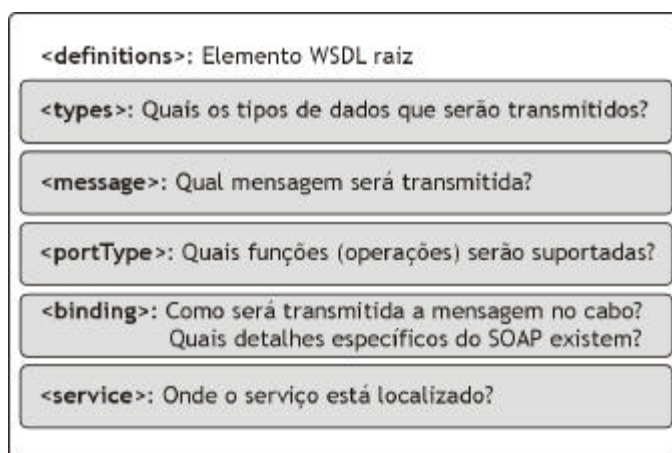


Figura 3: Estrutura de um arquivo WSDL.

2. Justificativa

Em Ciência da Computação são contempladas diversas áreas de conhecimento, entre elas: Sistemas Operacionais, Redes de Computadores, Programação e Modelagem de Sistemas. Para o

desenvolvimento desta proposta, que aborda a comunicação entre sistemas de forma automatizada, todas estas áreas foram colocadas em prática. Como exemplo de aplicação da proposta em uma sub-área de sistemas de telemetria - o rastreamento de veículos tem sido explorado por diversas empresas que utilizam a transferência de dados de telemetria por meio de satélite. Essas empresas fornecem, além do equipamento utilizado para a coleta e transmissão dos dados, um sistema que controla os dados coletados, de forma a apresentá-los e tratá-los no centro de controle. Isso torna-se uma grande desvantagem para o cliente, pois uma vez adquirido esse sistema, o cliente estará preso a ele, não podendo personalizar a aplicação existente, ou mesmo substituí-la por uma aplicação melhor. Dessa forma, ao padronizar a comunicação entre a empresa de rastreamento e o cliente, permite-se a escolha de qualquer sistema para o tratamento dos dados, independente da empresa de rastreamento que será utilizada, ou mesmo o desenvolvimento interno de uma aplicação que supra as necessidades da empresa.

3. Metodologia

A metodologia seguida no desenvolvimento do presente projeto baseia-se nos seguintes tópicos:

1. Levantamento bibliográfico sobre telemetria;
2. Estudo da linguagem de marcação XML (Extensible Markup Language);
3. Levantamento bibliográfico sobre Web Services;
4. Pesquisa de formas para a transmissão dos dados no nível de aplicação;
5. Análise e comparação as formas de transmissão de dados no nível de aplicação;
6. Pesquisa das formas de segurança disponíveis pelos protocolos usados para assegurar a transferência dos dados;
7. Modelagem do sistema de Web Service utilizando UML (Unified Modeling Language);
8. Implementação de um Web Service complementando o Sistema Rastro;
9. Validação do sistema;
10. Documentação.

4. Resultados

Para exemplificar melhor o padrão proposto, foi elaborado um protótipo que possui todas as especificações definidas. Esse protótipo possui a entrada de dados de forma manual (o serviço

em si é executado sem interação com o usuário) e foi elaborado utilizando a linguagem PHP e o Banco de Dados MySQL em conjunto com a biblioteca PEAR::SOAP. Essas tecnologias foram abordadas anteriormente neste capítulo.

Esse protótipo, aparentemente simples, apresenta recursos avançados, e muitas vezes de difícil compreensão. Um dos fatores mais relevantes na dificuldade da implementação foi a falta de documentação explicando de forma prática a implementação. A ausência de documentação deriva do fato de que a biblioteca ainda encontra-se em fase de desenvolvimento, não possuindo uma documentação formal. O material utilizado foi baseado em exemplos já existentes (embora muitos exemplos não funcionassem de acordo) e extensivas consultas ao Fórum de desenvolvedores da biblioteca SOAP.

Superados estes obstáculos, foi possível concluir este protótipo totalmente funcional de acordo com o padrão proposto. O Banco de Dados MySQL foi gerado automaticamente pela Ferramenta *Powerdesigner DataArchitect* versão 9.5.

A implementação foi dividida em dois módulos principais: o módulo servidor do *Web Service* e o módulo cliente. A seguir serão apresentadas algumas particularidades de cada módulo.

1.2.2.1 Módulo Servidor

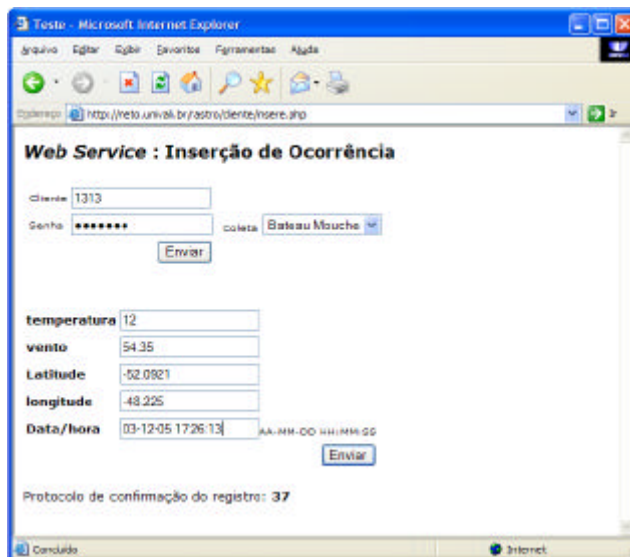
O módulo servidor é o responsável por receber as requisições SOAP dos clientes, processá-las de acordo com os métodos e retornar os resultados. É esse módulo que interage com o Banco de Dados onde encontram-se todas as informações registradas, e foi implementado utilizando todos os métodos propostos. O servidor possui apenas uma *interface* com o usuário que apresenta o arquivo WSDL responsável por descrever o serviço, e que pode ser visualizado no Anexo III.

1.2.2.2 Módulo Cliente

O módulo cliente está dividido em três telas principais, que serão apresentadas e descritas a seguir. A primeira tela é a tela inicial do protótipo. Nela existem duas opções de uso: Inserir Ocorrência e Consultar Registros e Sensores.

A Figura 4 apresenta a tela responsável por inserir ocorrências no sistema. Ao entrar nessa tela serão apresentados inicialmente os campos Cliente e Senha. Após preenchidos corretamente

esses campos, será executado um método via *Web Service* que retornará todos os pontos de coleta pertencentes ao cliente informado.



The image shows a screenshot of a web browser window titled "Teste - Microsoft Internet Explorer". The address bar shows the URL "http://Web0.univalk.br/astro/cliente/insere.php". The page content is titled "Web Service : Inserção de Ocorrência". It features a form with the following fields and values: "cliente" (1313), "Senha" (masked with asterisks), "coleta" (Bateau Mouche), "temperatura" (12), "vento" (54.35), "Latitude" (-52.0921), "longitude" (-48.225), and "Data/hora" (03-12-05 17:26:13). There are two "Enviar" buttons. At the bottom, it displays "Protocolo de confirmação do registro: 37".

Figura 4: Tela de inserção de ocorrências.

Após selecionado um dos pontos de coleta disponíveis será automaticamente executado outro método, que retornará todos os sensores da embarcação selecionada, mais a latitude, longitude e um campo para acrescentar a data/hora. Em seguida, deverão ser preenchidos os valores dos sensores, latitude e longitude, e pressionado o botão enviar. Nesse momento, será executado o método responsável por cadastrar uma ocorrência, que enviará para o *Web Service* todos os dados preenchidos e retornará um número de protocolo de confirmação do registro. Essa tela implementa três dos quatro métodos propostos: Registrar Ocorrência, Solicitar Pontos de Coleta e Solicitar Sensores.

Por fim, a Figura 5 apresenta a tela responsável apenas por consultas, e ela implementa o último método proposto: o Solicitar Ocorrências. Semelhante a tela anterior, essa também faz-se necessário informar o Cliente e a Senha. Após pressionado o botão Enviar, serão retornados todos os Pontos de Coleta disponíveis para determinado cliente. A diferença básica é a existência de dois *Checkbox* por onde é possível escolher quais os métodos que serão executados. Nesse exemplo, foram selecionados os dois *checkbox*, o que retornou os sensores do ponto de coletas escolhido e os registros dos últimos 10 dias desse ponto de coletas.

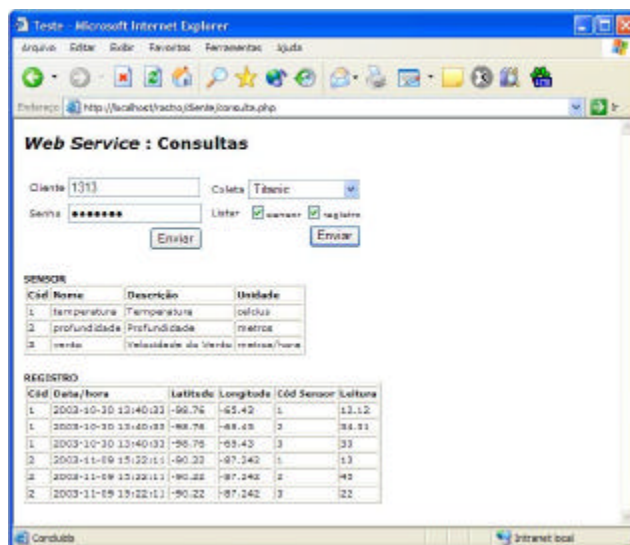


Figura 5: Tela de consultas do Módulo Cliente.

O sistema RASTRO, como apresentado no Capítulo 5, é dividido em dois módulos: Recepção de Dados e Mapa de navegação via *web*. O funcionamento inicial seguia as seguintes etapas:

?? A cada 15 minutos o agendador de tarefas do Linux, o *crontab*, executa um *shell script* que dispara a execução do *fetchmail*, responsável por conectar no servidor de *e-mail* via POP3 e copiar as mensagens para repassá-las a outro programa: chamado MDA (*Mail Delivery Agent*);

?? O MDA analisa todas as mensagens baixadas, uma a uma, coletando de cada mensagem informações como o posicionamento (Latitude e Longitude) e o nome da embarcação que enviou a informação.

?? De acordo com os parametros passados na execução do programa, o MDA grava os registros ou em um arquivo texto, ou diretamente no Banco de Dados Oracle.

A Figura 5 ilustra o funcionamento do módulo de Recepção de Dados do sistema RASTRO antes da implementação proposta.

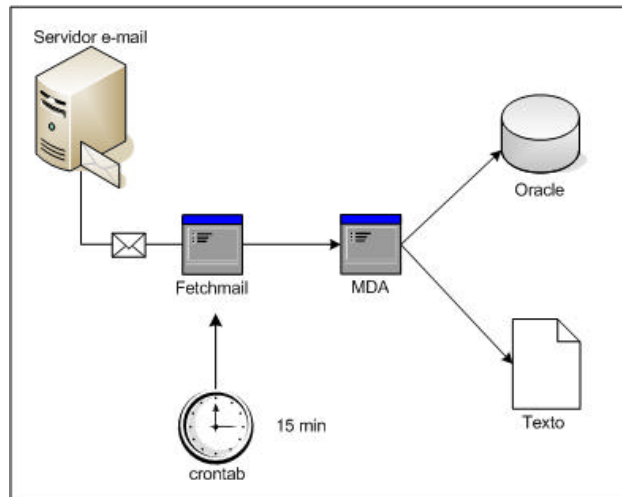


Figura 5: Funcionamento inicial do módulo de Recepção de Dados.

A implementação no sistema RASTRO não necessita de nenhuma interação humana – os dados são recebidos por e-mail, e todo o processo é automático. Para permitir ao RASTRO usufruir dos recursos do *Web Service* proposto e para validar esse modelo, a implementação realizada substituiu o programa MDA por duas aplicações *Web Services*: um cliente e um servidor.

A aplicação cliente é responsável por se conectar ao servidor POP3 (*e-mail*) a cada 15 minutos, baixando as mensagens uma a uma, analisando-as e coletando o posicionamento (latitude e longitude), data e hora de coleta e o nome da embarcação. Caso não exista uma posição válida, o valor *null* será repassado como latitude e longitude. Em seguida uma operação via *Web Service* é executada para consultar se a embarcação está ou não cadastrada no banco de dados do RASTRO. Caso esteja as informações são repassadas via *Web Service* para a aplicação servidor, que então armazena os dados no banco de dados Oracle ou em um arquivo texto. No caso da embarcação não existir no RASTRO, a aplicação cliente salvará as informações em um arquivo de *Log* para constar a tentativa de transmissão.

A Figura 6 apresenta a alteração realizada no módulo de Recepção de Dados, para compatibilizar o sistema RASTRO ao padrão proposto.

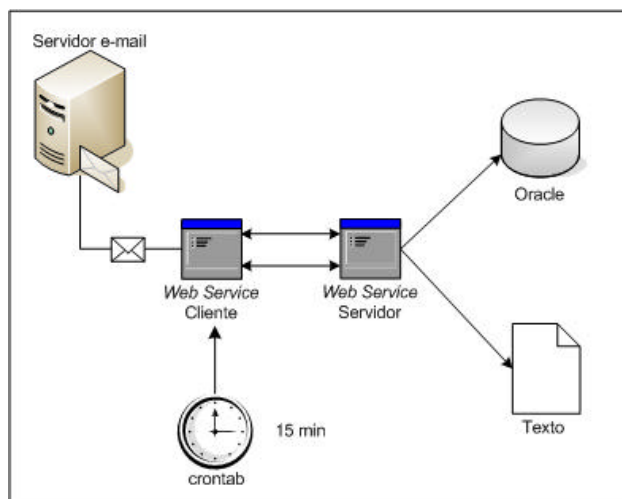


Figura 6: Alteração efetuada no módulo de Recepção de Dados.

Devido ao fato do sistema RASTRO ainda não suportar o armazenamento de valores coletados a partir de sensores, a implementação realizada limita-se apenas a aceitar informações referentes ao posicionamento e ponto de coleta.

5. Conclusão

Em termos de Telemetria, os estudos foram voltados à apresentação do conceito, exemplos de classificação e algumas vantagens e desvantagens. O objetivo desse capítulo foi apenas a apresentação conceitual do tema, uma vez que esse projeto não visa propor alterações na forma de comunicação existente hoje, e sim apresentar os benefícios do uso de *Web Services* para o transporte desses dados no nível de aplicação.

Uma vez delineado os estudos, percebeu-se a necessidade de apresentar uma proposta de padrão que exemplifique e apresente de forma minuciosa o objetivo do projeto. Para isso, foi necessário efetuar uma revisão bibliográfica que contemple o tema Padrões, apresentando a sua importância, definição, algumas entidades de Normalização, exemplos e uma introdução de como se constrói um padrão.

Em seguida, é apresentada uma seção que aborda os *Web Services*, descrevendo seu funcionamento, vantagens de utilização, e alguns conceitos acerca desse tema, como XML, UDDI e WSDL – uma linguagem elaborada para descrever um serviço. Existem dois protocolos distintos utilizados para a implementação de um *Web Service*: SOAP e XML-RPC. Ambos são

apresentados de forma profunda, e por fim é feito um comparativo entre os dois permitindo a escolha do mais viável para o projeto.

Como estudo de caso foi utilizado um sistema de monitoramento de embarcações - O sistema RASTRO. Foi dedicado um capítulo a esse sistema para permitir um melhor entendimento dos processos de um sistema de monitoramento. Uma das finalidades desse trabalho foi substituir um módulo do sistema RASTRO para adequá-lo ao padrão proposto.

Após finalizado o levantamento bibliográfico, iniciou-se o processo de desenvolvimento, onde foi elaborado a proposta de padrão nos moldes de uma RFC (*Request for Comment*) utilizando *Web Services*, que foi empregado para a elaboração de um protótipo de testes utilizado para apresentar todas as funcionalidades do modelo proposto e, em seguida, foi alterado um dos módulos do sistema RASTRO, permitindo a ele usufruir dessa proposta.

Este trabalho ajuda a contornar um problema comum entre as empresas de rastreamento e seus clientes, pois inicialmente não é possível optar pelo *software* de monitoramento de uma empresa e o serviço de rastreamento de outra. Uma vez padronizado esse canal de dados, um cliente poderá possuir apenas um sistema de monitoramento desenvolvido por qualquer empresa e utilizar o serviço de rastreamento de uma ou várias empresas de rastreamento diferentes.

Sendo o objetivo deste trabalho atender a necessidade de padronização no intercâmbio de informações de telemetria entre sistemas automatizados em nível de aplicação, foi estabelecido um padrão na forma de um documento, abordando as operações propostas e os tipos de dados utilizados. A descrição é apresentada na forma de um documento WSDL, que apresenta o serviço. Os dados são transacionados por meio do protocolo SOAP sob o HTTPS (*HyperText Transfer Protocol Secure*), o que garante a criptografia dos dados transportados.

É importante ressaltar a distinção deste tema enquanto um Trabalho de Conclusão de Curso de Ciência da Computação, ao considerar a inovação no tratamento de um problema não através da produção de um sistema, mas sim por meio de uma padronização das transações entre sistemas, ou seja, na criação de um protocolo de transações de dados de telemetria em camada de aplicação.

Como recomendação para o sistema RASTRO, pode-se citar a implementação de um suporte a

coletas de valores de sensores nas embarcações, a codificação de programas "exemplos" que poderão ser repassados a empresas de rastreamento que desejam se adequar ao sistema, e a remoção por completo do recebimento de dados telemétricos por meio de *e-mail*, utilizando apenas *Web Services* para esse fim.

Já para o padrão proposto incorporado nesse trabalho na forma de um anexo, é recomendado a definição de novas operações permitindo maior flexibilidade inter-sistemas, e o uso de estruturas de PKI (*Public Key Infrastructure*), XKMS (*XML Key Management Specification*) para assegurar as transações efetuadas entre empresas de rastreamento e seus clientes.

6. Referências Bibliográficas

AUSTIN, Daniel (Ed.) et al. **Web Services Architecture Requirements**. 2002. Disponível em: <<http://www.w3.org/TR/wsa-reqs>>. Acesso em: 05 mai. 2003.

AYALA, Dietrich et al. **Professional Open Source Web Services**. UK: Wrox Press Ltd, 2002. ISBN 1-861007-46-9.

BELLWOOD, Tom et al. **UDDI Version 3.0**. 2002. Disponível em: <<http://uddi.org/pubs/uddi-v3.00-published-20020719.htm>>. Acesso em: 20 mai. 2003.

BIRON, Paul V.; MALHOTRA, Ashok. **XML Schema Part 2: Datatypes**. 2001. Disponível em: <<http://www.w3.org/TR/xmlschema-2/>>. Acesso em: 06 jun. 2003.

BRAY, Tim (Ed.); HOLLANDER, Dave (Ed.); LAYMAN, Andrew (Ed.). **Namespaces in XML**. 1999. Disponível em: <<http://www.w3.org/TR/1999/REC-xml-names-19990114>>. Acesso em: 06 jun. 2003.

BRAY, Tim (Ed.) et al. **Extensible Markup Language (XML) 1.0**. 2 ed., 2000. Disponível em: <<http://www.w3.org/TR/REC-xml>>. Acesso em: 05 jun. 2003.

CABRAL, Rodrigo B.; SPERB, Rafael M.; TRIPODI, Rodrigo Z. **Tecnologia Opensource para Rastreamento Online de Veículos Monitorados via Satélite**. Itajaí. 2002. 11p. Disponível em: <<http://g10.cttmar.univali.br/publicacao-01.pdf>>. Acesso em: 20 mai. 2003.

CABRAL, Rodrigo B.; SPERB, Rafael M.; TRIPODI, Rodrigo Z.; WAHRLICH, Roberto; SOUZA, Jesiel de. **RASTRO: Internet Based Tracking System for Fisheries Control**. In: FIFTH INTERNATIONAL SYMPOSIUM ON GIS AND COMPUTER CARTOGRAPHY FOR COASTAL ZONE MANAGEMENT, October 2003, Genova, Italy. CD-ROM Proceedings of the Fifth International Symposium on GIS and Computer Cartography for Coastal Zone Management. 2003.

CERAMI, Ethan. **Web Services Essentials**. CA. O'Reilly, 2002. ISBN 0-596-00224-6.

GREGO, Maurício. De carona com Web Services. **Revista INFO Corporate**, São Paulo: Abril, n.1, p. 28-37, dez. 2002.

IN4MA REMOTE MONITORING SOLUTIONS. **What is Telemetry**. Disponível em: <<http://www.in4ma.co.uk/faq/telemetry.html>>. Acesso em: 15 abr. 2003.

L-3 COMMUNICATIONS TELEMETRY & INSTRUMENTATION. **Telemetry Tutorial**. Disponível em: <<http://www.ti.l-3com.com/tutorial/preface.html>>. Acesso em: 09 mar. 2003.

NMEA - National Marine Electronics Association. **NMEA 2000 Standard**. Disponível em: <<http://www.nmea.org/>>. Acesso em: 11 mar. 2003.

OPV - Oficina Projeto Virtual. **Web Services**. Disponível em: <http://www.iweb.com.br/opv/tec_web.php3>. Acesso em: 05 mai. 2003.

TESCH, José R. **XML Schema**. Florianópolis: Ed. Visual Books, 2002. ISBN 85-7502-073-0.

THOMPSON, Henry S. et al. **XML Schema Part 1: Structures**. 2001. Disponível em: <<http://www.w3.org/TR/xmlschema-1/>>. Acesso em: 06 jun. 2003.

TIDWELL, Doug. **Programming Web Services with SOAP**. Ed. O'Reilly. 2001. ISBN 0-596-00095-2.