

Alexsander Lopes de Oliveira

**Web Service Para Geração de VRS no Contexto de
Posicionamento GPS**

Alexsander Lopes de Oliveira

***Web Service Para Geração de VRS no Contexto de
Posicionamento GPS***

Monografia apresentada ao Departamento de Matemática, Estatística e Computação da Faculdade de Ciências e Tecnologia da Universidade Estadual Paulista “JÚLIO DE MESQUITA FILHO” – UNESP, como requisito obrigatório para a conclusão do curso de Bacharelado em Ciência da Computação.

Orientador: Prof. Dr. Milton Hirokazu Shimabukuro

Presidente Prudente

2010

Termo de Aprovação

Alexsander Lopes de Oliveira

Monografia sob o título **“Web Service Para Geração de VRS no Contexto de Posicionamento GPS”**, defendida por Alexsander Lopes de Oliveira e aprovada em 10 de Dezembro de 2010, em Presidente Prudente, Estado de São Paulo, pela banca examinadora constituída pelos doutores:

Dr^a. Daniele Barroca Marra Alves
Pós-Doutoranda/FAPESP - Departamento de Cartografia – FCT/UNESP

Prof. Dr. Messias Meneguette Junior
Departamento de Matemática, Estatística e Computação – FCT/UNESP

Prof. Dr. Milton Hirokazu Shimabukuro
Departamento de Matemática, Estatística e Computação – FCT/UNESP

Presidente Prudente, 10 de Dezembro de 2010

Agradecimentos

Agradeço a Deus, que sempre me surpreendeu mostrando que seus planos são maiores que os meus.

É com alegria que presto a minha gratidão a toda a minha família, em especial aos meus pais Cida e José e à minha irmã Andressa.

Os colegas de curso tiveram papel fundamental na minha formação, e a eles devo meus agradecimentos, bem como a todos os funcionários e professores da FCT, em especial ao Milton Shimabukuro, pela paciência e ajuda na orientação deste trabalho e à Daniele Alves, que com a mesma paciência e compreensão forneceu o código fonte do sistema legado utilizado neste trabalho e esclareceu todas as minhas dúvidas.

Resumo

A internet está totalmente inserida na sociedade contemporânea notadamente, no que se refere aos serviços de entretenimento e comércio. Seu alcance ultrapassou os tradicionais modelos *desktop* de computador chegando aos dispositivos móveis como celulares e receptores GPS (*Global Positioning System*). Não obstante, a comunidade científica utiliza-se de seus benefícios, seja para publicação de trabalhos, ou para comunicação entre clusters processando informações físicas, como as do LHC (*Large Hadron Collider*), localizado na Suíça.

No que se refere ao posicionamento geodésico, a pesquisa na área apresenta o conceito de Estações de Referência Virtual (*Virtual Reference Station – VRS*), no qual é necessária uma via de comunicação entre as estações de referência reais e um sistema central, bem como entre este e o solicitante do serviço.

Neste trabalho, buscamos analisar as soluções atuais para geração de VRS no que diz respeito à disponibilização dos dados ao usuário requisitante do serviço e apresentamos uma solução baseada em *Web Services* como uma alternativa ao modelo que vinha sendo desenvolvido no GEGE - Grupo de Estudos em Geodésia Espacial da FCT/UNESP.

Comparando as soluções, pôde-se verificar a potencialidade do uso de *Web Services* para auxiliar nas pesquisas de posicionamento geodésico utilizando VRS, pois, por meio desta tecnologia, obtém-se interoperabilidade, proporcionando maior flexibilidade para desenvolver aplicativos clientes, seja este desenvolvimento realizado por pesquisadores da universidade ou por qualquer pessoa física ou jurídica que deseje utilizar o serviço.

Palavras-chave: Aplicações Web. *Web Services*. Estação de Referência Virtual. Posicionamento GPS em Rede.

Abstract

Internet is fully inserted in contemporary society, specially in relation to entertainment services and trading. Its reach has transposed the traditional desktop computer models coming to mobile devices like cell phones and GPS receivers. Likewise, the scientific community takes its benefits, both for publication of studies and for communication between clusters processing information, such as at LHC, located in Switzerland.

Concerning geodetic positioning, researches in the area present the concept of Virtual Reference Stations - VRS, in which is necessary a communication way between the real reference stations and a central system as well as between central system and a service requester.

In this work, we analyze the current solutions for generation of VRS with regard to data delivery for the service requester and present a solution based on Web Services as an alternative to the model being developed by Spatial Geodesy Study Group – GEGE/FCT/UNESP.

Comparing solutions, it was verified the potential of Web Services to aid in researches of geodetic positioning using VRS. Using such technology, it is obtained interoperability, providing greater flexibility to develop client applications, both development carried out by researchers of the university or by any person or enterprise wishing to use the service.

Keywords: Web Applications. Web Services. Virtual Reference Station. Network Based Positioning.

Lista de Figuras

Figura 01 - Elipsóide Esferóide sendo $b = c$	06
Figura 02 - Distância $D = n * \lambda + F$	09
Figura 03 - Variações do centro de fase.....	11
Figura 04 - Área de cobertura do Trimble VRS Now TM na Europa	18
Figura 05 - Área de cobertura do Trimble VRS Now TM nos Estados Unidos	18
Figura 06 - Cobertura dos produtos da empresa Leica Geosystems	19
Figura 07 - Solicitação de VRS em formato RINEX.....	20
Figura 08 - Página Web para solicitação de dados VRS no e-mail	23
Figura 09 - Web Service provedor de serviço e um consumidor do mesmo	26
Figura 10 - Web Service no modelo SOA	27
Figura 11 - Web Service como um “Wrapper”	28
Figura 12 - Hierarquia WSDL	29
Figura 13 - Hierarquia SOAP.....	33
Figura 14 - Arquitetura da solução	35
Figura 15 - Software consumidor do serviço	38

Lista de Códigos Fonte

Listagem 01 - Elemento <i>import</i>	30
Listagem 02 - Elemento <i>types</i>	30
Listagem 03 - Elementos <i>message</i>	30
Listagem 04 - Elemento <i>portType</i>	31
Listagem 05 - Elemento <i>binding</i>	31
Listagem 06 - Elemento <i>service</i>	32
Listagem 07 - SOAP sobre HTTP	34
Listagem 08 - Operação no Web Service.....	37
Listagem 09 - WSDL	A-2
Listagem 10 - XSD.....	B-1
Listagem 11 - Requisição usando SOAP	C-1
Listagem 12 - Resposta usando SOAP.....	D-1

Sumário

1	INTRODUÇÃO.....	1
1.1	CONTEXTUALIZAÇÃO.....	1
1.2	OBJETIVOS	2
1.2.1	OBJETIVO GERAL.....	3
1.2.2	OBJETIVOS ESPECÍFICOS	3
1.3	JUSTIFICATIVA	4
1.4	ORGANIZAÇÃO.....	4
2	GPS – GLOBAL POSITIONING SYSTEM	6
2.1	CONSIDERAÇÕES INICIAIS	6
2.2	SEGMENTOS DO SISTEMA.....	6
2.2.1	SEGMENTO ESPACIAL	7
2.2.2	SEGMENTO DE CONTROLE.....	7
2.2.3	SEGMENTO DO USUÁRIO	8
2.3	ERROS.....	9
2.3.1	ERROS ORIGINÁRIOS DOS SATÉLITES	10
2.3.2	ERROS ORIGINÁRIOS DO RECEPTOR	10
2.3.3	ERROS ORIGINÁRIOS DA PROPAGAÇÃO DOS SINAIS	11
2.4	MÉTODOS DE POSICIONAMENTO	12
2.4.1	POSICIONAMENTO POR PONTO SIMPLES	13
2.4.2	DGPS	14
2.4.3	RTK	15
2.4.4	POSICIONAMENTO EM REDES - VRS	15
3	SOLUÇÕES COMPUTACIONAIS PARA REDE DE ESTAÇÕES DE REFERÊNCIA.....	18
3.1	TRIMBLE VRS NOW TM	18
3.2	LEICA SmartNet e SpiderWeb.....	19
3.3	GNSMART.....	21
3.4	SERVIR.....	21
3.5	VRS UNESP.....	22
4	WEB SERVICES	24
4.1	CONSIDERAÇÕES INICIAIS	24
4.2	ARQUITETURAS	25
4.2.1	PROVEDOR - CONSUMIDOR.....	25
4.2.2	SOA	26
4.2.3	WRAPPER	27
4.3	PADRÕES	28
4.3.1	WSDL.....	28
4.3.2	SOAP.....	32
5	SOLUÇÃO PROPOSTA	35
5.1	ALTERAÇÕES NO SISTEMA LEGADO.....	35
5.2	WEB SERVICE.....	36

5.3	<i>CONSUMIDOR</i>	37
6	CONSIDERAÇÕES FINAIS	39
6.1	<i>TRABALHOS FUTUROS</i>	40
	Referências	41
	Apêndice A – Arquivo WSDL	A-1
	Apêndice B – Arquivo XSD	B-1
	Apêndice C – Requisição SOAP ao Web Service	C-1
	Apêndice D – Resposta SOAP do Web Service	D-1
	Apêndice E – Instalação/Configuração do Sistema	E-1

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

A busca pelo posicionamento GPS com alta acurácia é de grande importância para uma grande variedade de aplicações, por exemplo, nas áreas de sismologia, vulcanologia, topografia e geodésia. Porém apenas a utilização dos satélites para obter o posicionamento não é suficiente para obter a precisão desejada, isto se deve a diversos fatores como os efeitos atmosféricos produzidos sobre o sinal além de outros.

A forma mais simples de se obter o posicionamento consiste em obter a distância entre receptor e satélites por meio da fórmula $D = VT$ (Distância = Velocidade * Tempo) observando-se o tempo que leva para um sinal ser repetido pelo satélite a uma velocidade já conhecida. Após obter a distância, o posicionamento do receptor é calculado utilizando-se a equação da distância geométrica aplicada a 04 ou mais satélites em um sistema de equações lineares, o que caracteriza o posicionamento por ponto. O Capítulo 2 tratará deste assunto.

Outra forma de obter a distância é utilizar a fórmula $D = n*\lambda + F$, onde λ é o comprimento de onda do sinal, n é número de comprimentos e F é a diferença de fase observada. Mais detalhes sobre as observáveis no sistema GPS e os principais métodos de posicionamento são apresentados no Capítulo 2.

O posicionamento por ponto simples não produz resultados precisos, como também será descrito. Como solução para o posicionamento GPS de alta precisão, surgiram os métodos de posicionamento baseados em redes, nos quais medições de receptores fixos em locais de coordenadas conhecidas são utilizados para mitigar os erros que ocorrem na determinação do posicionamento. As estações fixas formam o que se chama de Rede de Estações de Referência, as quais estão presentes em países como Austrália, Singapura, Finlândia, Brasil, dentre outros.

Neste contexto surge a Estação de Referência Virtual (*Virtual Reference Station - VRS*) obtida pela combinação dos dados de uma rede de estações de referência.

No conceito de VRS, uma estação base é gerada nas proximidades do receptor móvel (usuário). Assim, o usuário tem a possibilidade de utilizar um receptor de simples frequência para realizar o posicionamento com uma linha de base curta (ALVES, 2008).

São várias as etapas a serem cumpridas ao construir uma ferramenta para gerar uma VRS e disponibilizar seus dados ao usuário, mas pode-se dividi-las em algumas principais as quais são: Obtenção de dados das estações de referência reais, geração da VRS utilizando modelos matemáticos (modelos que por vezes utilizam dados atmosféricos relacionados à ionosfera e troposfera), e transmissão dos resultados ao receptor do usuário, como descreve Alves (2008).

Em relação à geração da VRS, vários modelos matemáticos foram desenvolvidos pela comunidade científica e os resultados obtidos satisfazem a bons critérios de precisão. Para esta etapa pode ser usado um sistema de processamento central.

Para a transmissão de dados entre o sistema central e o usuário, pode-se utilizar a internet, sendo o *Web Service* uma boa opção de implementação, a qual proporciona acoplamento fraco entre o servidor e o software cliente.

Neste trabalho, busca-se analisar as soluções existentes para a geração de VRS, principalmente no que se refere à disponibilização dos dados e comunicação entre o sistema de processamento central e o requisitante do serviço. Também foi desenvolvida e apresentada uma solução baseada em *Web Services* e comparada com as existentes.

Web Service é uma solução para integração de sistemas que proporciona interoperabilidade. O objetivo é que as aplicações possam se comunicar independentemente da plataforma de desenvolvimento por meio de mensagens padronizadas.

Além da interoperabilidade, o uso desta tecnologia proporciona o baixo acoplamento entre as aplicações.

Mais detalhes sobre o projeto e implementação de *Web Services*, bem como a linguagem de comunicação serão apresentados no Capítulo 4.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

O objetivo deste projeto é implementar um *Web Service* para geração de VRS no contexto da Rede GPS Ativa do Estado de São Paulo bem como um software cliente para se comunicar com o mesmo e comparar com as soluções existentes, notadamente com o software desenvolvido por Alves (2008) o qual está sendo modificado para utilizar uma arquitetura web baseada em formulários para disponibilizar os resultados aos usuários no projeto *Desenvolvimento e Implantação do RTK em Rede Para Posicionamento Geodésico no Estado de São Paulo* (Alves, 2010).

Como consequência, pretende-se contribuir para a evolução das pesquisas na área de posicionamento geodésico utilizando VRS, proporcionar disponibilidade de serviço e alta precisão para usuários GPS, bem como demonstrar a adequabilidade do uso de *Web Services* para aplicações GPS.

1.2.2 OBJETIVOS ESPECÍFICOS

Para alcançar a meta supracitada, diversas etapas intermediárias deverão ser cumpridas tais como segue abaixo:

- Analisar os softwares proprietários e *freeware* já existentes para uso de Redes de Estações de Referência e VRS, bem como os desenvolvidos por Alves (2008 e 2010), verificando a possibilidade de integração destes ao *Web Service*;
- Implementar as funcionalidades indisponíveis para geração de VRS no que concerne à obtenção de dados da Rede GPS Ativa do Estado de São Paulo e comunicação com o usuário;
- Definir o padrão de comunicação, mensagens e parâmetros que serão utilizados entre o *Web Service* e os softwares clientes;
- Implementar o *Web Service*;
- Criar um software cliente que servirá de exemplo para o uso do *Web Service*;
- Comparar com as soluções existentes.

1.3 JUSTIFICATIVA

A utilização de VRS tem sido muito estudada pela comunidade científica, modelos matemáticos tem sido desenvolvidos e os resultados tem sido satisfatórios, porém ainda falta infra-estrutura de software para o completo desenvolvimento das pesquisas.

A característica de distribuição inerente aos receptores GPS e a complexidade envolvida na geração das VRS nos abre precedente para a utilização de um grande recurso computacional, o *Web Service*, o qual pode ser utilizado como sendo o sistema central para geração da VRS, com a vantagem de um acoplamento fraco entre este e o software cliente.

Além disso, a presença da Rede GPS Ativa do Estado de São Paulo e das pesquisas desenvolvidas por Alves (2008 e 2010) nos indica a necessidade da infra-estrutura proporcionada pelos *web services* nesta área de interesse.

Finalizando, o uso de VRS aliado às vantagens proporcionadas por um *web service* gera boas perspectivas para que este serviço seja muito utilizado no futuro, tendo em vista o crescente número de aplicações GPS e a necessidade de alta precisão em muitas delas.

1.4 ORGANIZAÇÃO

No Capítulo 2, os aspectos gerais do posicionamento por GPS são abordados, conceitos básicos são apresentados assim como os principais métodos de posicionamento até se chegar ao método que utiliza o conceito de VRS.

No Capítulo 3 são apresentadas as principais soluções comerciais e não comerciais para o posicionamento utilizando rede de estações de referência focando os aspectos computacionais das mesmas e os métodos de comunicação com o usuário.

O conceito de *Web Services* é apresentado no Capítulo 4, assim como uma breve discussão sobre as vantagens da tecnologia, arquiteturas e padrões normalmente utilizados.

A solução proposta neste trabalho é apresentada no Capítulo 5, com a integração do software existente escolhido para realização dos cálculos com um *web service* desenvolvido, assim como um software consumidor do serviço que ilustra o seu funcionamento.

Finalmente, o Capítulo 6 traz as considerações finais, elencando as vantagens e desvantagens da solução proposta em relação às existentes e sugerindo trabalhos futuros.

2 GPS – GLOBAL POSITIONING SYSTEM

2.1 CONSIDERAÇÕES INICIAIS

O *Global Positioning System* (GPS) ou Sistema de Posicionamento Global foi desenvolvido pelo Departamento de Defesa dos Estados Unidos e declarado operacional em 1995. Por meio deste pode-se obter as coordenadas de um ponto a um tempo em relação ao sistema de coordenadas geodésicas WGS 84. Nesse sistema a superfície terrestre é considerada uma aproximação de um elipsóide do tipo esferóide com valor $a = 6.378.137$ metros no equador e $b = 6.356.752,3$ metros nos pólos conforme descreve EL-Rabbany (2002). A Figura 01 ilustra o elipsóide.

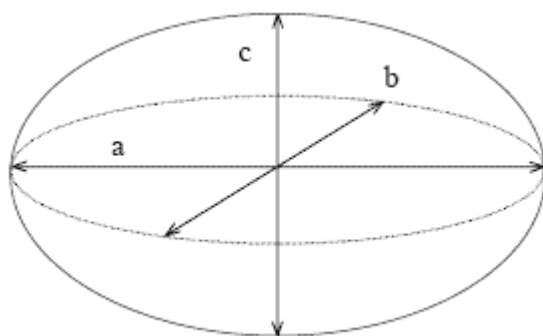


Figura 01: Elipsóide Esferóide sendo $b = c$.
Fonte: Adaptada de Afonso Júnior et al (2002).

2.2 SEGMENTOS DO SISTEMA

O sistema divide-se em três segmentos que são o segmento espacial, segmento de controle e segmento do usuário. Tais segmentos são descritos a seguir.

2.2.1 SEGMENTO ESPACIAL

O segmento espacial inicialmente consistia em 24 satélites divididos em 06 órbitas planas com 04 satélites cada uma, porém hoje existem mais satélites de diferentes gerações em órbita e este número está sempre mudando conforme descreve Seeber (2003).

Um receptor pode obter sua posição desde que tenha pelo menos quatro satélites visíveis de acordo com os métodos de posicionamento que serão apresentados na Seção 2.4.

Os satélites transmitem os códigos binários chamados código C/A e código P além da mensagem de navegação.

Os códigos são transmitidos em portadoras usualmente chamadas de L1 e L2 com as respectivas frequências de 1575,42 MHz e 1227,60 MHz sendo que a L1 é modulada pelo código C/A, código P e mensagem de navegação enquanto a L2 é modulada apenas pelo código P e mensagem de navegação. (Seeber, 2003).

O código C/A é um código único o qual identifica o satélite e trata-se de uma *stream* de 1023 bits repetida a cada milissegundo sendo sua taxa de transmissão de 1,023 Mbps. São utilizados 37 códigos diferentes para os 24 ou mais satélites em órbita.

O código P tem taxa de transmissão de 10,23 Mbps sendo composto por cerca de $2,35 \times 10^{14}$ bits divididos em 38 segmentos que levam exatamente uma semana para serem transmitidos. Mais preciso que o código C/A, o código P é criptografado gerando o código Y sendo utilizado apenas por usuários autorizados. (Monico, 2008).

A mensagem de navegação é composta por 25 frames de 1500 bits modulados na razão de 50 bps. Cada frame é composto de 5 sub-frames com 10 palavras de 30 bits cada uma contendo informações sobre o posicionamento do satélite, data atual e tempo, dados atmosféricos, dados de almanaque (informações orbitais para aquele e outros satélites do sistema) dentre outras.

Mais detalhes sobre o assunto podem ser encontrados em Monico (2008) e Seeber (2003).

2.2.2 SEGMENTO DE CONTROLE

O segmento de controle tem a função de monitorar os satélites e enviar-lhes atualizações de suas mensagens de navegação. Consiste em Estações de Monitoramento ao redor do planeta que capturam a informação da distância, altura e velocidade dos satélites visíveis além de informações atmosféricas e as transmitem para a Estação Mestre localizada em Colorado Springs, EUA, a qual calcula as novas órbitas e também é responsável pela sincronização do tempo. (Seeber, 2003).

Algumas das estações também tem a capacidade transmitir dados aos satélites para a atualização de sua mensagem de navegação bem como a realização de seu reposicionamento.

2.2.3 SEGMENTO DO USUÁRIO

O segmento do usuário consiste no equipamento receptor conectado a uma antena para determinar o seu posicionamento.

Utilizando-se dos dados obtidos neste segmento, o usuário implementa os diferentes métodos de posicionamento através de diferentes algoritmos, softwares e técnicas para correções de erros, dentre elas a utilização de redes GPS (Monico, 2008), tendo resultados distintos como consequência, os quais podem variar a precisão obtida no posicionamento da ordem de poucos milímetros a metros em determinadas circunstâncias.

Os receptores se diferenciam pelo oscilador, código e frequência suportados, número de canais, etc.

Em relação à frequência, receptor de simples frequência é aquele que utiliza apenas L1 e de dupla frequência é aquele que utiliza L1 e L2 conforme classifica EL-Rabbany (2002).

Monico (2008) traz uma classificação quanto aos tipos de dados proporcionados pelo equipamento, os quais podem ser os códigos C/A, P1, P2 além dos sinais L1 e L2. Assim um receptor pode proporcionar acesso aos códigos e não às portadoras, bem como aos códigos e portadoras. Diferentes combinações são possíveis como apenas C/A, C/A e L1, C/A, L1 e L2, etc. Cabe ressaltar que L1 e L2 são referências às medidas de fase das referidas portadoras, como será descrito a seguir.

Os códigos são utilizados para a determinação da pseudodistância. Esta é obtida comparando o código repetido pelo satélite com uma réplica no receptor, verificando a

defasagem para assim obter seu valor por meio da fórmula $D = VT$ (Distância = Velocidade * Tempo). A denominação pseudodistância provém do não sincronismo entre o relógio do receptor e o relógio do satélite, causando erro na determinação do tempo conforme descreve Monico (2008). Consequentemente, o valor obtido para a distância também não é exato.

A fase da onda portadora, mais precisa que a pseudodistância é a outra observável que pode ser utilizada. Utilizando esta medida, a distância do receptor ao satélite é obtida pelo número de ciclos recebidos multiplicados pelo comprimento de onda somando-se ao final a diferença de fase conforme ilustra a Figura 02.

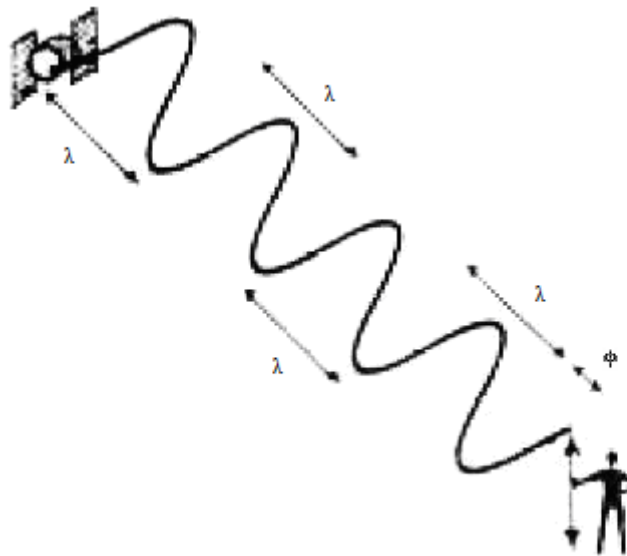


Figura 02: Distância $D = n * \lambda + F$.
Adaptada de Timbó (2000).

O valor n não é conhecido inicialmente, sendo seu cálculo efetuado em diferentes instantes. Ele é chamado como parâmetro inicial inteiro de ambigüidade, sendo os valores diferentes para L1 e L2. Os métodos para a sua solução não serão apresentados neste trabalho. Mais detalhes podem ser obtidos em Monico (2008) e Seeber (2003).

2.3 ERROS

A acurácia no posicionamento GPS depende do tratamento de diversos erros sistemáticos, os quais podem ter suas fontes nos satélites, receptores, propagação do sinal e estações conforme descreve Monico (2008). As seções a seguir descrevem os principais erros que devem ser modelados para a realização do posicionamento com alto grau de precisão. Os erros relacionados às estações não são apresentados neste trabalho, mas podem ser encontrados em Monico (2008).

2.3.1 ERROS ORIGINÁRIOS DOS SATÉLITES

Os satélites sofrem a ação de forças físicas como as radiações solares e forças gravitacionais da Terra, do Sol e da Lua, as quais alteram a sua órbita conforme descreve Seeber (2003).

EL-Rabbany (2002) salienta que as estações de controle solicitam aos satélites a correção de suas órbitas, mas no intervalo de tempo em que os erros orbitais permanecem, a acurácia das efemérides transmitidas é de 1 a 3 metros de acordo com Monico (2008).

Outra fonte de erros é o clock dos satélites que é impreciso na faixa de 8,64 a 17,28 ns por dia. O segmento de controle faz o monitoramento deste erro e envia informação aos satélites para a correção.

Uma terceira causa de erros com origem nos satélites é a chamada Disponibilidade Seletiva (SA) e trata-se da adição intencional de erros pelo governo americano nos relógios e nas coordenadas do satélite limitando a acurácia do GPS no segmento civil a cerca de 100 metros. A SA foi desativada no ano 2000, melhorando o posicionamento em torno de 10 vezes de acordo com Monico (2008).

Outros erros relacionados ao satélite dizem respeito à relatividade, atraso entre as duas portadoras no hardware do satélite, centro de fase da antena do satélite, e fase *wind-up*. Mais detalhes sobre os mesmos podem ser encontrados em Monico (2008) e Seeber (2003).

2.3.2 ERROS ORIGINÁRIOS DO RECEPTOR

Assim como acontece no segmento espacial, o relógio do receptor também tem um grau de imprecisão que gera erro, já que a medida do tempo é utilizada na determinação das distâncias. Monico (2008) afirma que o erro associado pode ser praticamente eliminado quando se usa o posicionamento relativo desde que o erro do relógio de cada receptor não seja superior a 01 microssegundo. Técnicas de posicionamento relativo serão apresentadas na Seção 2.4.

Outra fonte de erros está no fato de que o centro de fase da antena receptora não coincide com o centro mecânico ou geométrico da mesma, ele depende do ângulo de elevação do satélite e intensidade do sinal observado levando a uma imprecisão da ordem de centímetros, dependente do tamanho da antena. A Figura 03 ilustra tal situação.

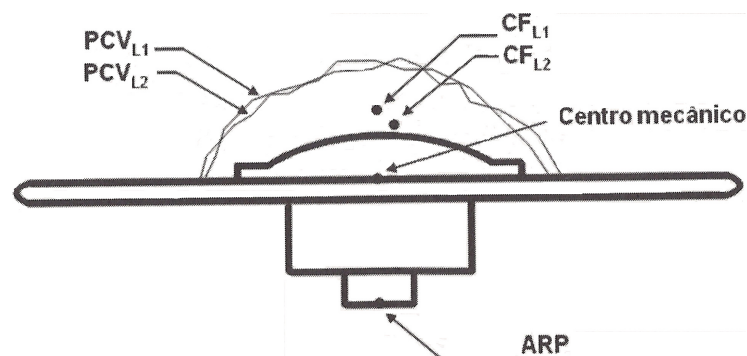


Figura 03: Variações do centro de fase.
Fonte: Retirada da referência (Monico, 2008).

Na Figura 03, CF_{L1} e CF_{L2} são os centros de fase das portadoras L1 e L2, *Phase Center Variation*, ou PCV é a variação do centro de fase e *Antenna Reference Point*, ou ARP é o ponto de referência da antena acessível a medidas que é relacionado com os centros de fase.

Monico (2008) também escreve sobre o erro entre canais do receptor e fase *wind-up*.

2.3.3 ERROS ORIGINÁRIOS DA PROPAGAÇÃO DOS SINAIS

As imprecisões originárias da propagação dos sinais são causadas principalmente pelo atraso ionosférico, refração troposférica, e efeito de multicaminho.

A ionosfera é um meio dispersivo que causa atraso no sinal. Este é proporcional ao número de elétrons livres no caminho da onda, que por sua vez depende da hora do dia (sendo

maior durante a tarde), da época do ano (sendo maior no verão, no hemisfério sul), do ciclo solar de 11 anos (sendo maior no período de maior quantidade de explosões solares) e da localização geográfica (sendo os níveis irregulares em regiões equatoriais, como grande parte do Brasil). O efeito é diferente para as portadoras L1 e L2, podendo ser reduzido usando uma combinação destas conforme descreve EL-Rabbany (2002).

A troposfera é um meio não dispersivo e independente de frequência, assim a combinação de L1 e L2 não pode ser usada para minimizar seus efeitos.

O atraso troposférico depende da temperatura, pressão e umidade. Sinais de satélite com um baixo ângulo de elevação têm um caminho maior para atravessar na troposfera causando maiores erros. Efeito semelhante ocorre na ionosfera.

É importante frisar que o atraso troposférico é atribuído a dois componentes: A componente seca que corresponde a cerca de 90% da imprecisão (existem modelos matemáticos que geram bons resultados para tratá-la) e a componente úmida, dependente do vapor d'água presente, sendo necessários dados meteorológicos para modelagem, neste caso mais complexa.

O efeito de multicaminho acontece quando o sinal reflete em um objeto e assim chega ao receptor por um caminho alternativo ao caminho direto.

Monico (2008) apresenta com detalhes os erros discutidos nesta seção e ainda os relacionados à perda de ciclos e rotação da terra.

2.4 MÉTODOS DE POSICIONAMENTO

Um método de posicionamento é classificado em pós-processado ou tempo real, relativo, diferencial, por ponto ou baseado em redes. A Tabela 01 apresenta os principais métodos de posicionamento utilizados.

Tabela 01 - Métodos de Posicionamento

MÉTODO	TIPO	TEMPO DE RESPOSTA
<i>Por Ponto</i>	Por ponto	Pós-Processado/Tempo Real
<i>Por Ponto Preciso</i>	Por ponto	Pós-Processado/Tempo Real
<i>DGPS</i>	Por ponto	Pós-Processado/Tempo Real

<i>RTK</i>	Relativo	Pós-Processado/Tempo Real
<i>VRS</i>	Em redes	Pós-Processado/Tempo Real

Os métodos ainda podem ser classificados em estáticos ou cinemáticos, com as variações estático rápido e semi-estático apresentadas por Monico (2008), porém estas classificações não são importantes para o desenvolvimento deste trabalho.

O posicionamento por ponto tem este nome porque é utilizado apenas um receptor, o qual tem sua localização calculada desde que haja quatro satélites visíveis. A Seção 2.4.1 descreve os princípios do posicionamento por ponto simples.

No posicionamento relativo, são utilizados no mínimo dois receptores, sendo um chamado receptor base localizado em um ponto de coordenadas conhecidas. Este é usado para auxiliar na correção de erros das coordenadas do receptor móvel após a estimação do vetor que liga a estação base e a estação móvel. Diferentes técnicas tem sido utilizadas no posicionamento relativo sendo possível sua determinação de maneira estática e cinemática, dentre outras conforme apresenta Monico (2008). A precisão alcançada pode ser subcentimétrica ou de poucos metros conforme descrevem Freiburger Junior e Krueger (2007).

No DGPS, assim como no posicionamento relativo, utiliza-se de uma estação fixa, porém esta transmite ao usuário as discrepâncias observadas nas coordenadas ou pseudodistâncias obtidas conforme descreve Monico (2008).

Embora alguns textos apresentem o DGPS como um método relativo, Monico (2008) afirma que o mesmo pode ser considerado como um método de posicionamento por ponto, já que, em termos de referencial, proporciona as coordenadas com relação ao geocentro.

A Seção 2.4.2 apresenta o DGPS e a Seção 2.4.3 apresenta o método de posicionamento relativo RTK.

Na Seção 2.4.4 é apresentado o método de posicionamento baseado em redes utilizando uma Estação de Referência Virtual, uma promissora área de pesquisa.

2.4.1 POSICIONAMENTO POR PONTO SIMPLES

Na realização do posicionamento por ponto existem quatro incógnitas a serem determinadas, as quais são as coordenadas x , y e z do receptor e o erro associado à imprecisão do seu relógio. A solução é obtida utilizando as coordenadas conhecidas X , Y e Z dos quatro satélites e as respectivas distâncias dos satélites ao receptor.

A pseudodistância poderia ser substituída na equação da distância geométrica conforme a Equação (1):

$$p_i = \sqrt{(X - x)^2 + (Y - y)^2 + (Z - z)^2} \quad (1)$$

Onde:

- p_i é a pseudodistância até o satélite i ;

A utilização do quarto satélite é necessária também à determinação do erro associado à imprecisão do relógio.

Esta equação pode ser linearizada e resolvida por meio de métodos algébricos ou numéricos. Monico (2008) apresenta detalhes da linearização da equação descrita acima em função do tempo das observações bem como detalhes do posicionamento por ponto.

Monico (2008) também informa sobre a possibilidade da utilização das medidas de fase da onda portadora no processamento, mas argumenta que tal procedimento não é normalmente utilizado, já que não proporciona refinamento da precisão em uma única época.

A precisão alcançada é de cerca de 10 a 15 metros de acordo com Seeber (2003).

O posicionamento por ponto preciso (PPP) gera erros de magnitude centimétrica ou decimétrica. O método de posicionamento por ponto preciso não será apresentado neste trabalho. Mais informações podem ser encontradas em (Monico, 2008).

2.4.2 DGPS

O DGPS ou *Differential GPS* é um método de posicionamento baseado em código utilizando uma estação base e um receptor móvel monitorando o mesmo conjunto de satélites (no mínimo quatro). A estação base de referência determina as discrepâncias nas coordenadas ou pseudodistâncias obtidas, tendo por base as coordenadas conhecidas previamente. A comparação é feita por software e utilizada para modelar os erros. Então as correções são

enviadas ao usuário que determinará as corretas coordenadas ou pseudodistâncias. Monico (2008) apresenta detalhes do DGPS.

Geralmente a informação entre a base e o móvel é enviada em tempo real via RTCM (Radio Technical Commission for Maritime Service)

A acurácia obtida é de cerca de 0,5 a 3 metros segundo Monico (2008).

2.4.3 RTK

Métodos cinemáticos são aqueles em que a posição do receptor é obtida enquanto este se movimenta. Geralmente utilizam a fase da portadora para a realização do posicionamento, sendo necessária inicialmente a resolução das ambigüidades.

RTK (*Real Time Kinematic*) é um método cinemático e relativo que, semelhantemente ao DGPS utiliza uma estação base com coordenadas conhecidas para gerar as correções e enviar ao usuário utilizando RTCM.

A solução de ambigüidades é realizada utilizando a técnica denominada *on-the-fly* não apresentada neste trabalho. A precisão esperada é de poucos centímetros (Monico, 2008), porém a técnica é utilizada quando a linha de base (distância entre o receptor e a estação base de referência) não é superior a 10 Km (Freiberger Junior e Krueger, 2007).

A observável utilizada é a dupla diferença (DD) obtida pela simultaneidade da observação de dois satélites por dois receptores e o posicionamento é obtido em relação à estação base. Detalhes podem ser encontrados em Monico (2008).

2.4.4 POSICIONAMENTO EM REDES - VRS

Alves (2008) cita os principais métodos de posicionamento baseado em redes de estações de referência como sendo os que utilizam algoritmos de derivadas parciais, algoritmos de interpolação, algoritmo de ajustamento condicional e o algoritmo de estação de referência virtual. Nesta seção apresentamos o último deles.

O conceito de VRS (*Virtual Reference Station* ou Estação de Referência Virtual) consiste na geração de uma estação virtual nas proximidades do usuário criada a partir de dados de estações reais dispostas em rede. Após a determinação dos dados da estação virtual, pode-se calcular o posicionamento do receptor móvel utilizando o posicionamento relativo.

A distância entre o receptor e as estações de referência reais pode ser de mais de uma centena de quilômetros sendo esta a grande vantagem em relação ao posicionamento RTK, que limita esta distância a aproximadamente 10 km tendo seus resultados prejudicados caso esta seja ultrapassada conforme descrevem Freiburger Junior e Krueger (2007). Isto resulta em um menor custo, devido ao uso de um menor número de estações reais.

As estações de referência são ligadas a um centro de controle. Lá elas são monitoradas para a modelagem dos erros sistemáticos decorrentes da ionosfera e troposfera, bem como a solução das ambigüidades para L1 e L2.

Para realizar o posicionamento, o usuário envia sua localização aproximada (a qual pode ser obtida por posicionamento por ponto) ao centro de controle, então este envia os dados da VRS ao usuário para que este faça o posicionamento relativo. Sendo assim percebe-se a necessidade de softwares e infra-estrutura de comunicação entre o receptor e o centro de controle e entre o centro de controle e as estações da rede.

Alves (2008) descreve uma maneira de gerar os dados da VRS que consiste em selecionar uma estação mais próxima do usuário como estação base. Os dados da VRS são gerados a partir das observações da fase e pseudodistância da estação base. Posteriormente é calculada a distância geométrica entre o satélite e a VRS e entre o satélite e a estação base. O deslocamento geométrico Δp , calculado como a diferença entre estas medidas, é aplicado a todas as observações da estação base deslocando-as para a posição da VRS. Após esta etapa, os dados da VRS são corrigidos dos erros de ionosfera e troposfera obtidos a partir da modelagem realizada com os dados das estações de referência da rede.

Afonso et al (2007) ressaltam que o receptor do usuário pode enviar mensagens ao centro de controle por GSM (*Global System for Mobile Communications*) e o centro de controle pode enviar a resposta no formato CMR (*Compact Measurement Record*) ou RTCM (*Radio Technical Commission for Maritime Services*) em caso de posicionamento em tempo real. Uma outra alternativa é utilizar o formato RINEX quando o posicionamento é realizado no modo pós-processado.

Ressalta-se que a indisponibilidade de uma estação de referência é suprida por outras da região, sendo assim o posicionamento baseado em redes pode prover tolerância a falhas desde que esta característica tenha sido buscada em seu projeto.

No próximo capítulo, apresentamos algumas soluções existentes para o posicionamento baseado em redes de estações de referência buscando enfatizar os aspectos computacionais das mesmas.

3 SOLUÇÕES COMPUTACIONAIS PARA REDE DE ESTAÇÕES DE REFERÊNCIA

Neste capítulo apresentamos as principais soluções computacionais para a realização do posicionamento utilizando redes de estações de referência.

3.1 *TRIMBLE VRS NOWTM*

A empresa Trimble apresenta uma solução baseada em VRS chamada Trimble VRS NowTM. As Figuras 04 e 05 apresentam a área de cobertura do serviço na Europa e Estados Unidos respectivamente.

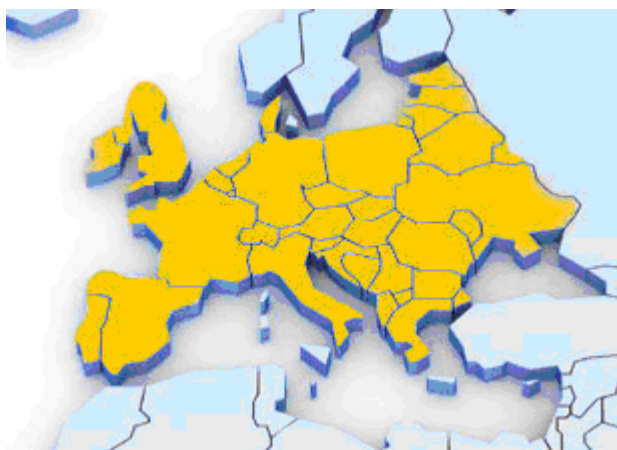


Figura 04: Área de cobertura do Trimble VRS NowTM na Europa.
Disponível no site www.trimble.com.



Figura 05: Área de cobertura do Trimble VRS NowTM nos Estados Unidos.
Disponível no site www.trimble.com.

A própria empresa faz a manutenção das estações de referência para assegurar o correto funcionamento.

São utilizados receptores da própria empresa preparados para fazer a comunicação. Não há muita informação técnica disponível, já que é uma solução comercial.

3.2 LEICA SmartNet e SpiderWeb

A empresa Leica Geosystems tem soluções baseadas em estações de referências, sendo a principal delas o SmartNet. A Figura 06 mostra a área de cobertura dos produtos da empresa na Europa. A rede SmartNet também está sendo expandida para a América do Norte e Austrália.

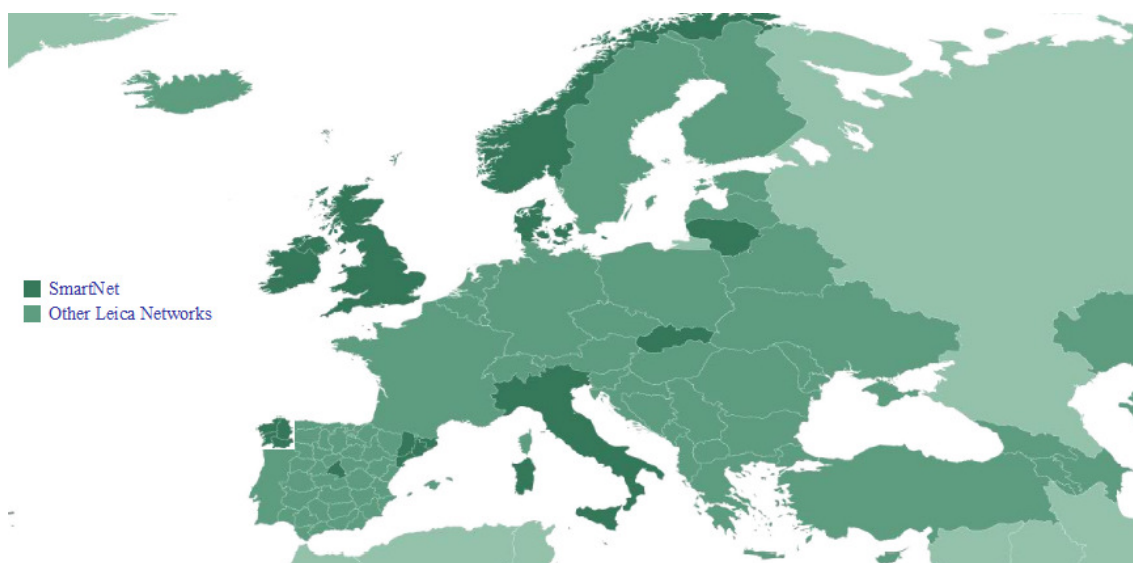


Figura 06: Cobertura dos produtos da empresa Leica Geosystems.
Disponível no site www.leica-geosystems.com.

Alguns dos países possuem soluções baseadas em VRS utilizando produtos da empresa, outros possuem apenas serviços baseados em RTK.

Cabe destaque ao serviço SmartNet, inovador por apresentar a possibilidade da transmissão de informações via GSM (Global System for Mobile Communications) ou GPRS (General Packet Radio Service), sendo necessário que o usuário registre o seu *Sim card*.

Assim como o Trimble VRS NowTM, o SmartNet trata-se de uma solução comercial projetada para ser usada com os receptores da própria empresa e criar dependência do usuário em relação à mesma.

A empresa também possui o produto SpiderWeb para auxiliar na disponibilização dos dados gerados em uma rede para usuários na internet. Entre as opções, pode-se transmitir dados em tempo real via portas TCP/IP, assim como serviço de transmissão de dados VRS no formato RINEX. Sendo um produto comercial, os detalhes de implementação do mesmo não estão disponíveis.

Em relação ao serviço de geração de arquivos RINEX, o mesmo se dá de forma que o usuário solicita a criação da VRS e depois faz o download do arquivo, não havendo qualquer interface para a comunicação direta por meio de software. A Figura 07 mostra como é realizada a solicitação.

Virtual RINEX Request

Date: *	11/17/2010	
Local start time: *	0 : 0	
Duration: *	1 : 0	h:min
Output files: *	☒ Observation ☒ Navigation	
Observation rate: *	1.00	sec
Latitude: *	0 ° 0 ' 0 " °	☒ N ☐ S
Longitude: *	0 ° 0 ' 0 " °	☐ W ☒ E
Height: *	0.0	m
Virtual station site code: *	VREF	
Virtual station site name: *	Virtual Rinex	
Use data from satellite system: *	GPS only	
Select position of site in map:		
Google Map		
Note: You can not set the height component using google maps. Please enter the desired value yourself.		
* Required field		
		Submit

Figura 07: Solicitação de VRS em formato RINEX.
Disponível em www.leica-geosystems.com.

Apesar disso, o SpiderWeb apresenta-se como uma boa ferramenta para disponibilização de dados ao usuário via internet se comparada às existentes devido ao

número de opções que tem, com prestação de serviços para pós processamento ou utilização em tempo real.

3.3 GNSMART

GNSmart é a solução da empresa Geo++. Ela é formada pelos seguintes produtos da empresa: GNREF, GNCOM e GNNET que são responsáveis pelas estações de referência, comunicação, e correções da rede respectivamente.

As correções são realizadas e enviadas ao receptor móvel via RCTM ou RINEX por rádio ou conexões TCP/IP.

O RCTM é suportado por diversos receptores, incluindo os das empresas Trimble e Leica.

Sendo assim, o GNSmart proporciona maior flexibilidade ao usuário no que diz respeito ao uso do receptor.

O INCRA (Instituto Nacional de Reforma Agrária) utiliza dos produtos da empresa Geo++ na operação da RiBaC (Rede INCRA de Bases Comunitárias do GNSS) para geração VRS e processamento de dados GNSS (CIGALA, 2010).

3.4 SERVIR

A rede SERVIR é um projeto português liderado pelo Instituto Geográfico do Exército. A arquitetura foi projetada de forma a disponibilizar dados em tempo real via RTCM para posicionamento RTK além de GSM, GPRS e ainda HTTP (*Hipertext Transfer Protocol*) e FTP (*File Transfer Protocol*) para dados pós-processados pela internet.

Por meio do endereço <http://213.63.136.12/> o usuário deve fazer a autenticação. Na página o mesmo pode solicitar a criação de estações virtuais no formato RINEX e fazer o download utilizando FTP.

Mais detalhes da rede SERVIR são apresentados por Afonso et al (2007).

3.5 VRS UNESP

Alves (2008), como parte de sua tese, implementou na Universidade Estadual Paulista um sistema para a geração de VRS que aqui denominaremos VRS UNESP.

Dadas as coordenadas aproximadas, o software gera uma VRS em formato RINEX de acordo com diferentes métodos de processamento utilizando facultativamente previsão numérica do tempo e o modelo de Hopfield para modelar o atraso troposférico, e ainda mapas globais da ionosfera e modelagem regional da ionosfera.

Para tal atividade o sistema utiliza os dados das estações de referência e as efemérides precisas IGS disponíveis para download via FTP pelo endereço <ftp://igscb.jpl.nasa.gov/pub/product/>. Estas efemérides são resultado da combinação de dados de vários centros de análise e ficam disponíveis com latência de cerca de treze dias de acordo com Monico (2008).

O software foi desenvolvido na linguagem de programação C++ utilizando algumas bibliotecas de funções desenvolvidas na própria FCT/UNESP.

Em Alves (2010) o sistema está sendo adaptado para disponibilizar os resultados ao usuário por e-mail utilizando o protocolo SMTP (*Simple Mail Transfer Protocol*) após o mesmo ter realizado uma solicitação pela internet. A Figura 08 mostra o layout que está sendo desenvolvido.

The screenshot shows a web application for requesting VRS (Vertical Reference Station) data. The page is titled "VRS" and displays a map of São Paulo state with several VRS stations marked: ILHA, SJRP, SPAR, LINS, ROSA, PPTK, OCRT, SPCA, UBAT, and NEIA. The map is centered at -49.33 X -24.28. The interface includes a sidebar with navigation links: Histórico, Colaboradores, Aspectos Teóricos, and Software. The main content area contains a form for entering VRS data, including a section for "Sistema de Coordenadas" (Coordinate System) with options for Phi, Lambda, H and X, Y, Z. Below this, there are input fields for "Entre com as coordenadas da VRS" (Enter the coordinates of the VRS) for Phi, Lambda, and H. There are also input fields for "Entre com o dia no qual a VRS deve ser gerada" (Enter the day on which the VRS should be generated), "Entre o tempo inicial" (Enter the initial time), and "Entre o tempo final" (Enter the final time). The form includes buttons for "Calcular" (Calculate) and "Limpar" (Clear). The footer of the page contains navigation links: Home, Portal UNESP, Unidades, Licitações, Telefones, and WebMaster.

Figura 08: Página Web para solicitação de dados VRS no e-mail.
Em desenvolvimento, cedido pelo GEGE-FCT/UNESP.

No Capítulo 4, são abordados os conceitos, arquiteturas e padrões utilizados na construção de *Web Services*, uma tecnologia pouco explorada nas soluções para disponibilização de dados VRS ao usuário, considerando as soluções pesquisadas.

4 WEB SERVICES

4.1 CONSIDERAÇÕES INICIAIS

O W3C (2004) define *Web Services* da seguinte maneira:

“Um Web service é um sistema de software projetado para suportar interoperabilidade em uma rede. Ele tem uma interface descrita em um formato (especificamente WSDL) processável por máquina. Outros sistemas interagem com o Web service na forma definida na sua descrição usando mensagens SOAP, tipicamente transmitidas usando HTTP com uma serialização XML em conjunto com outros padrões web relacionados” (tradução própria).

A tecnologia de *Web Services* surgiu com a necessidade da integração de aplicações em sistemas heterogêneos. Seu objetivo é prover serviços por meio de uma interface que será conhecida por quem pretende “consumir” o serviço. A interface deve estar descrita em conformidade com um padrão sendo a WSDL (*Web Services Description Language*) a linguagem de definição que se consolidou na área.

Para transmitir as mensagens, utiliza-se o protocolo SOAP (*Simple Object Access Protocol*) sobre HTTP (*Hipertext Transfer Protocol*) ou HTTPS (*Hipertext Transfer Protocol Secure*) como uma implementação XML (*Extensible Markup Language*).

Anteriormente ao surgimento dos *Web Services*, outras tecnologias surgiram na tentativa de resolver o problema da integração de sistemas, entre elas podemos citar o modelo cliente/servidor implementado diretamente usando *sockets*. Este modelo tem como característica o acoplamento forte, o que não é desejado tendo em vista a constante mudança dos softwares.

Outra solução foi o padrão CORBA (*Common Object Request Broker Architecture*) que utiliza o conceito de objetos distribuídos. O CORBA apresentou grandes vantagens, possibilitando toda a infra-estrutura para objetos distribuídos, sendo possível atingir a interoperabilidade e integração de aplicações legadas. Porém a tecnologia vem sendo superada pelo uso de *Web Services*. Newcomer e Lomow (2005) lembram que pessoas com familiaridade com a tecnologia CORBA argumentam que *Web Services* são apenas sistemas CORBA implementados utilizando XML, porém apontam que, apesar de ser possível fazer

em CORBA quase tudo o que é possível fazer com *Web Services*, pode-se implementar um *Web Service* apenas utilizando SOAP, sem a necessidade da WSDL. Já em CORBA, é obrigatório o uso da IDL (*Interface Definition Language*), sendo a arquitetura de CORBA inflexível e difícil de compreender, o que coloca o fator humano como problema para o desenvolvimento da tecnologia.

Albinader Neto e Lins (2006) descrevem fatores técnicos para uso de *Web Services* em detrimento de CORBA, como o acoplamento orientado a conexões. Botto (2004), por sua vez, aponta como um dos fatores de sucesso dos *Web Services* a dose menor de invasão de um sistema legado por meio da criação de uma camada de serviços que se comunique com o mesmo, permitindo a migração gradual e evitando custos maiores. Mais detalhes da integração de *Web Services* com sistemas legados, assim como outras possibilidades de arquitetura, serão discutidos na Seção 4.2.

Na Seção 4.3 é apresentada a linguagem de definição WSDL e o protocolo SOAP, os principais padrões abertos utilizados na construção de *Web Services*.

4.2 ARQUITETURAS

Nesta seção são apresentados os principais modelos arquiteturais utilizados para a implementação de *Web Services*.

4.2.1 PROVEDOR - CONSUMIDOR

Os *Web Services* surgiram com uma arquitetura parecida com o que se emprega no modelo cliente-servidor. Nesta arquitetura o software provedor do serviço disponibiliza uma descrição dos serviços prestados por meio de um arquivo WSDL, Linguagem de Definição de *Web Services*. Por sua vez, o cliente, aqui chamado consumidor do serviço, obtendo uma cópia do arquivo WSDL, constrói mensagens utilizando o protocolo SOAP para fazer a comunicação. A Figura 09 ilustra o exposto acima.



Figura 09: *Web Service* provedor de serviço e um consumidor do mesmo.

Newcomer e Lomow (2005) ainda escrevem sobre a possibilidade de utilizar apenas o protocolo SOAP sem a necessidade da descrição WSDL, como de fato era feito inicialmente.

4.2.2 SOA

A sigla SOA vem do inglês *Service Oriented Architecture*, que significa Arquitetura Orientada a Serviços. Apesar de ser comum a sua implementação com *Web Services*, esta não é uma exigência. No documento *Definition of SOA*, produzido pelo consórcio *The Open Group* (2006), algumas considerações são feitas sobre o item serviço, entre elas podemos destacar que um serviço deve ser uma “caixa preta” para quem o consome e pode ser composto de outros serviços.

Em relação à arquitetura, ela deve ser baseada em serviços, os quais devem ser orquestrados utilizando padrões abertos para obter interoperabilidade e transparência de localização.

A arquitetura citada na Seção 4.2.1 estava próxima de cumprir as recomendações SOA, mas para alcançar transparência de localização, como componente indispensável à orquestração dos serviços, surgiu necessidade de inserir mais um componente ao modelo, como ilustra a Figura 10.

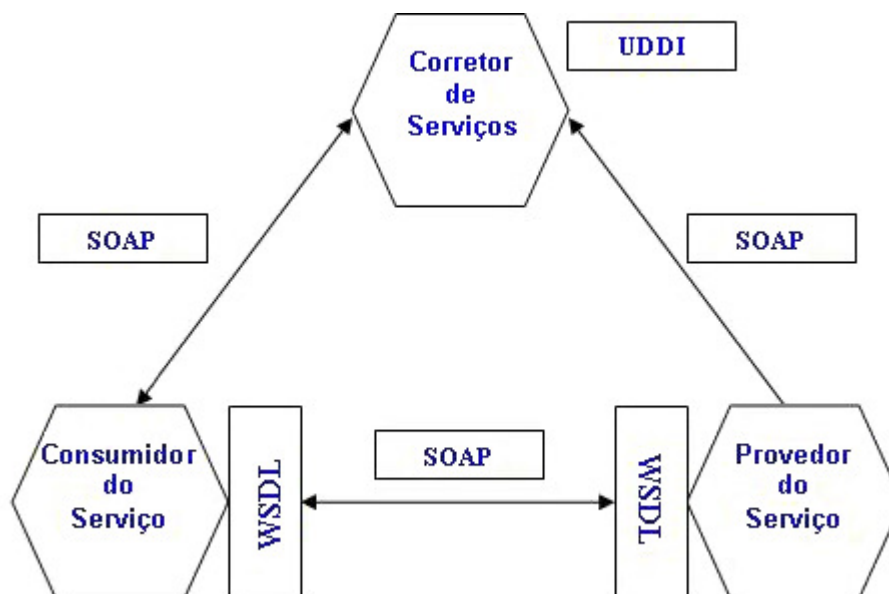


Figura 10: *Web Service* no modelo SOA.

Para conseguir cumprir as recomendações do modelo SOA, utiliza-se um serviço para a descoberta de outros serviços, geralmente chamado Corretor de Serviços ou Registro de Serviços. Ele utiliza o padrão UDDI (*Universal Description, Discovery and Integration*) para realizar tal atividade. O UDDI, assim como WSDL, utiliza XML e foi projetado para possibilitar a descrição e descoberta de organizações, negócios e provedores de serviços.

No funcionamento da arquitetura, o provedor do serviço publica o mesmo no corretor UDDI usando SOAP, por sua vez, o consumidor do serviço obtém o arquivo WSDL com informações do serviço e localização do corretor e o utiliza para se comunicar com o provedor de serviços. Assim obtém-se transparência de localização.

Este tem se tornado um padrão de fato na construção de *Web Services*.

4.2.3 WRAPPER

Uma prática comum na construção de um web service é fazê-lo como uma camada entre um sistema legado e o consumidor do serviço. Sommerville (2007) o denomina de “*wrapper*” que significa empacotador. Em Newcomer e Lomow (2005) encontra-se a denominação *Legacy Service Gateway*. A Figura 11 demonstra um *web service* como um *wrapper*.

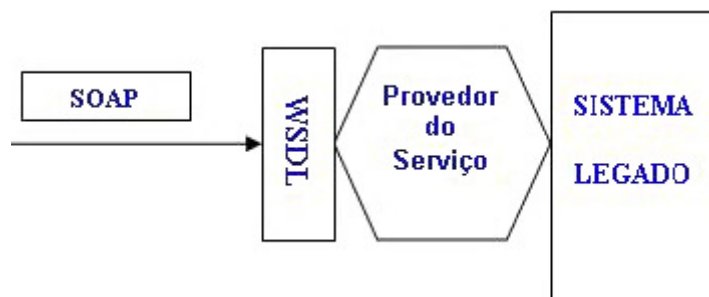


Figura 11: *Web Service* como um “Wrapper”.

Nesta arquitetura, o *web service* “empacota” a requisição SOAP do cliente e a repassa ao sistema legado, seja por CORBA, ou qualquer outro método que esteja acessível à empresa ou organização. Este método pode ser implementado tanto na comunicação com um consumidor do serviço como na comunicação com o corretor de serviços, caso se utilize o modelo SOA.

A grande vantagem está na possibilidade de integrar o sistema legado aos poucos, não alterando muito sua estrutura, já que para grandes sistemas isto é uma dificuldade.

4.3 PADRÕES

A seguir apresentamos os principais padrões abertos utilizados no desenvolvimento de *Web Services*.

4.3.1 WSDL

Um documento WSDL, *Web Services Description Language* ou Linguagem de Definição de *Web Services* trata-se de um formato XML que tem como objetivo informar ao consumidor do serviço o que o *web service* faz, como se comunicar com o mesmo e onde encontrá-lo. Para tal atividade utiliza as estruturas *import*, *types*, *message*, *portType*, *binding* e

service uma posição a baixo da estrutura *definitions*, raiz na hierarquia, conforme ilustra a Figura 12.

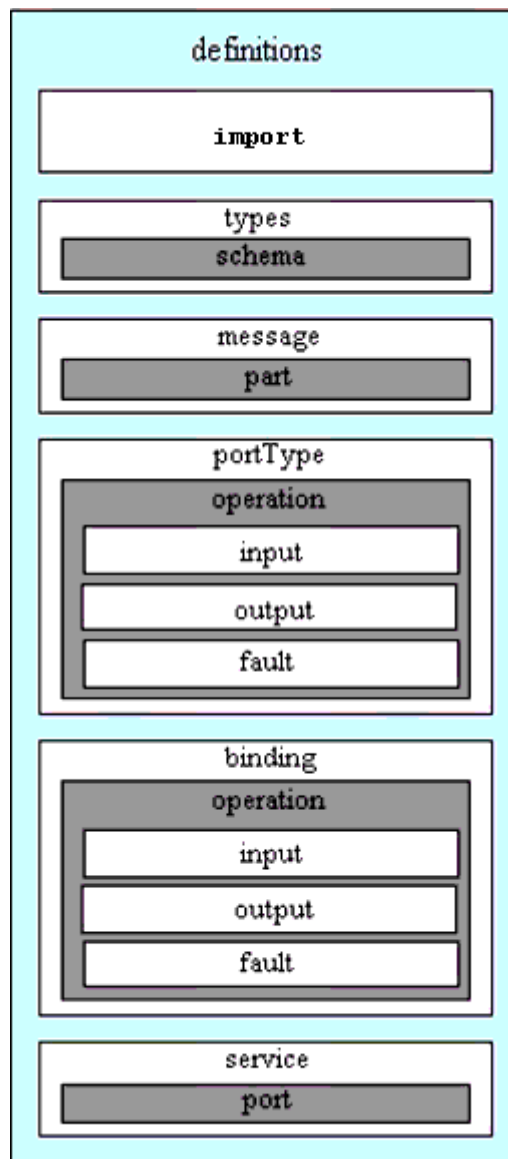


Figura 12: Hierarquia WSDL.
Fonte: Adaptada da referência (IMS, 2005).

Abaixo, discutimos os elementos WSDL de forma fragmentada. Um exemplo integral de WSDL é o que apresentamos no Apêndice A, que descreve o serviço implementado durante o desenvolvimento deste trabalho.

O elemento *import* possibilita a separação das outras estruturas em diferentes documentos, assim pode-se integrá-los ao documento principal no lugar de outros elementos, como por exemplo o elemento *types*. Não se trata de um item obrigatório. Um exemplo do uso de *import* é apresentado na Listagem 01.

```

1      <import namespace="http://example.com/conversorMoedas/schemas"
2      location="http://example.com/conversorMoedas/conversorMoedas.xsd"/>

```

Listagem 01: Elemento *import*.

O elemento *types* é o local onde os esquemas de tipos de dados são definidos. Normalmente utiliza-se o esquema XSD, com as características dos tipos XML, mas extensões podem ser utilizadas. A construção na Listagem 02 demonstra o uso de *types*.

```

1      <types>
2          <schema targetNamespace="http://example.com/conversorMoeda.xsd"
3              xmlns="http://www.w3.org/2000/10/XMLSchema">
4              <element name="pais">
5                  <complexType>
6                      <all>
7                          <element name="nome" type="string"/>
8                      </all>
9                  </complexType>
10             </element>
11         </schema>
12     </types>

```

Listagem 02: Elemento *types*.

A próxima estrutura é *message*. Toda mensagem que possa ocorrer na comunicação entre um provedor e um consumidor deve estar definida com estruturas deste tipo. O trecho na Listagem 03 mostra as mensagens trocadas na implementação desenvolvida neste trabalho.

```

1      <message name="vrsByteRinex">
2          <part name="parameters" element="tns:vrsByteRinex"/>
3      </message>
4      <message name="vrsByteRinexResponse">
5          <part name="parameters" element="tns:vrsByteRinexResponse"/>
6      </message>
7      <message name="Exception">
8          <part name="fault" element="tns:Exception"/>
9      </message>

```

Listagem 03: Elementos *message*.

O item *part* possui um atributo *name* e obrigatoriamente um atributo *type* ou *element*, que associa a mensagem a tipos XSD, ou a tipos definidos no elemento *types*.

A estrutura *portType* define as interações realizadas por meio do elemento *operation*. Cada operação possui entrada, e/ou saída (resposta), e/ou mensagem de falha. As diferentes possibilidades de construção acontecem pelo fato de que a interação pode ser em um único sentido, solicitação-resposta, resposta-solicitação ou notificação. Na Listagem 04 é apresentada a estrutura *portType* desenvolvida neste trabalho:

```

1      <portType name="VRSGenerator">
2          <operation name="vrsByteRinex">
3              <input wsam:Action="http://vrsgenerator.me.org/VRSGenerator
4                  /vrsByteRinexRequest" message="tns:vrsByteRinex"/>
5              <output wsam:Action="http://vrsgenerator.me.org/VRSGenerator
6                  /vrsByteRinexResponse" message="tns:vrsByteRinexResponse"/>
7              <fault message="tns:Exception" name="Exception" wsam:Action=
8                  "http://vrsgenerator.me.org/VRSGenerator/vrsByteRinex/Fault/
9                  Exception"/>
10         </operation>
11     </portType>

```

Listagem 04: Elemento *portType*.

O elemento *binding* define os detalhes de protocolo e formato de mensagens das operações de um elemento *portTypes* os quais serão usados na comunicação. Três tipos de *binding* são definidos no WSDL, o SOAP, o MIME e o HTTP.

```

1      <binding name="conversorMoedaPortBinding" type="tns: conversorMoeda">
2          <soap:binding transport="http://
3              schemas.xmlsoap.org/soap/http" style="document"/>
4          <operation name="alterar">
5              <soap:operation soapAction=""/>
6              <input>
7                  <soap:body use="literal"/>
8              </input>
9          </operation>
10     </binding>

```

Listagem 05: Elemento *binding*.

O elemento *service* define os elementos *port* como locais físicos de acesso onde o *binding* é realizado conforme é demonstrado na Listagem 06.

```

1      <service name="conversorMoeda">
2          <port name="conversorMoedaPort" binding=
3              "tns:conversorMoedaPortBinding">
4              <soap:address location="http://conversor.com/conversorMoeda
5          </port>
6      </service>

```

Listagem 06: Elemento *service*.

Os detalhes da WSDL podem ser encontrados no documento *Web Services Description Language (WSDL) 1.1* do W3C (2001).

4.3.2 SOAP

O SOAP (*Simple Object Access Protocol*), é o protocolo de serviço de mensagens habitualmente utilizado na construção de *Web Services*, ele é baseado em XML e implementado na camada de aplicação sobre o HTTP, HTTPS ou outros como o SMTP.

O protocolo não possui qualquer definição relacionada ao modelo do objeto ou linguagem de programação, apenas como as mensagens devem ser trocadas. Originalmente, não mantém estado ou conexão, mas pode-se obter sincronismo na comunicação pela combinação de mensagens enviadas em sentidos opostos.

Em relação à estrutura, o elemento raiz é o *Envelope*, o qual possui um elemento opcional *Header*, com várias entradas de header e um elemento *Body* obrigatório. Também podem haver anexos. A Figura 13 ilustra a hierarquia de uma mensagem SOAP.

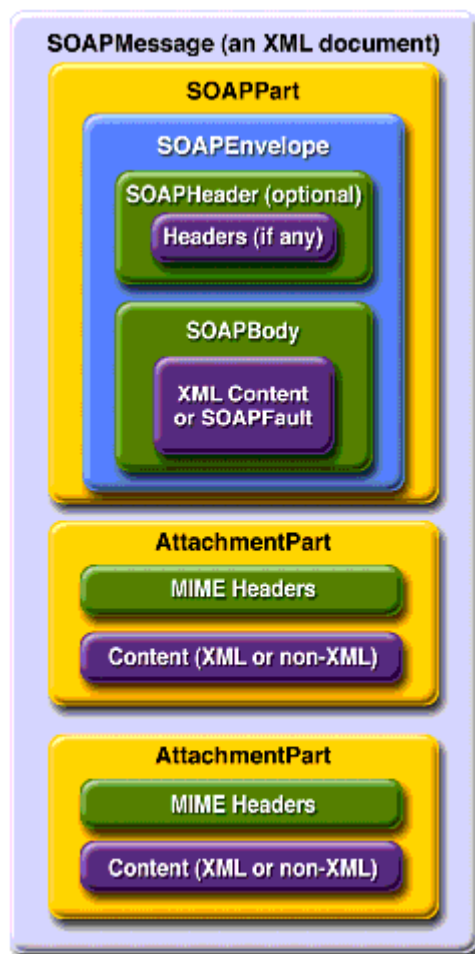


Figura 13: Hierarquia SOAP.
 Fonte: Extraída da referência (WILDE, 2006).

O *Header* é utilizado para incluir meta informações em relação ao *Body*, e principalmente informações de controle ou contexto das mensagens.

O elemento *Body* contém o “*payload*”, que se trata da mensagem propriamente dita e um elemento opcional *Fault* que contém as informações de erro retornadas por um nó ao processar uma mensagem.

A descrição de SOAP pode ser encontrada no documento *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)* do W3C (2007). Neste documento há um exemplo do uso de *Header*. Cunha (2002) complementa este exemplo, demonstrando o seu uso sobre HTTP e incluindo o elemento *Body* conforme apresentado na Listagem 07.

```

1      POST /StockQuote HTTP/1.1
2      Host: www.stockquoteserver.com
3      Content-Type: text/xml; charset="utf-8"
4      Content-Length: nnnn
5      SOAPAction: "Some-URI"
6      <SOAP-ENV:Envelope
7          xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
8          SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
9          <SOAP-ENV:Header>
10             <t:Transaction
11                 xmlns:t="some-URI" SOAP-ENV:mustUnderstand="1">
12                 5
13             </t:Transaction>
14          </SOAP-ENV:Header>
15          <SOAP-ENV:Body>
16             <m:GetLastTradePrice xmlns:m="Some-URI">
17                 <symbol>DIS</symbol>
18             </m:GetLastTradePrice>
19          </SOAP-ENV:Body>
20      </SOAP-ENV:Envelope>

```

Listagem 07: SOAP sobre HTTP.

O elemento *SOAPAction* especifica uma URI que pode ser usada por um firewall para fazer a filtragem. As duas primeiras linhas do envelope indicam o estilo de processamento e codificação.

No *Header*, o atributo *Transaction* define um *namespace* para o elemento e 5 é um valor a ser interpretado pelos nós que processam a mensagem. O cabeçalho deve ser processado pelo receptor da mensagem, isto é especificado com *mustUnderstand=1*.

No elemento *Body*, um método é chamado com seu respectivo parâmetro.

O Capítulo 5 traz a descrição das atividades realizadas neste trabalho, com a implementação de um *web service* como solução para transmitir dados VRS ao usuário.

5 SOLUÇÃO PROPOSTA

Neste capítulo, apresentamos a solução proposta neste trabalho para a disponibilização de dados VRS ao usuário no que diz respeito à comunicação entre as partes. Verificamos a possibilidade de utilizar o software apresentado na Seção 3.5, produzido por Alves (2008 e 2010), como sendo um sistema legado a ser integrado a um *web service*, este construído como um *wrapper* conforme foi discutido na Seção 4.2.3.

Houve necessidade de fazer alterações no sistema legado, estas são apresentadas na Seção 5.1. O *web service*, é descrito na Seção 5.2.

Um software consumidor foi desenvolvido para demonstrar o uso do serviço, este é apresentado na Seção 5.3.

O modelo arquitetural utilizado é ilustrado na Figura 14.

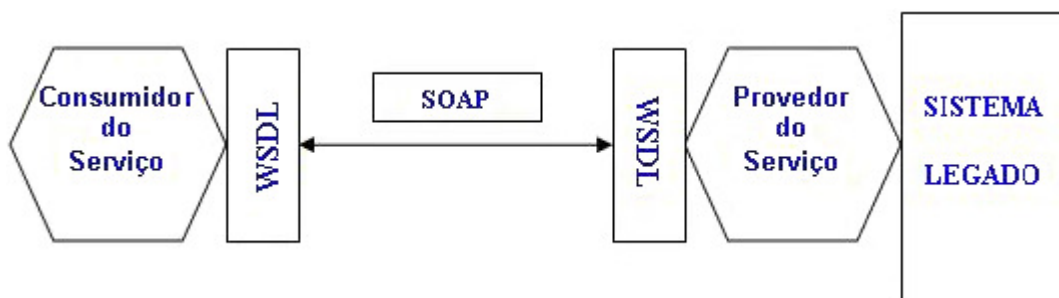


Figura 14: Arquitetura da solução.

5.1 ALTERAÇÕES NO SISTEMA LEGADO

Conforme foi discutido na seção 3.5, uma solução está em fase de desenvolvimento na FCT/UNESP para a disponibilização dos dados via e-mail após uma solicitação pela internet. No entanto utilizamos apenas o sistema desenvolvido inicialmente, o qual realiza os cálculos e salva um arquivo no formato RINEX em uma pasta do software. Este foi alterado de forma a receber os argumentos ao ser executado, sendo possível utilizá-lo para testes como uma

aplicação *desktop* em que os dados de entrada são informados no arquivo fonte ou executado pelo *web service*, que passa os parâmetros via linha de comando.

O software também fora modificado de forma a fazer o download dos arquivos RINEX das estações de referência. Existe a opção de obter estes arquivos no próprio Departamento de Cartografia da FCT/UNESP ou no IBGE. Optou-se pela segunda alternativa, já que o IBGE disponibiliza os dados das estações de referência de todo o Brasil e a FCT/UNESP, apenas os dados das estações do Estado de São Paulo. Sendo assim, em futuras atualizações, o software poderá ser alterado de maneira a prover o serviço em território nacional. Os dados são obtidos no endereço <ftp://geofp.ibge.gov.br/RBMC/dados/>.

Com as alterações realizadas, quando o software é executado pelo *web service*, os dados resultantes não são gravados em arquivo, como era feito anteriormente, mas enviados para a saída padrão, a qual é capturada pelo *web service*.

5.2 WEB SERVICE

O software provedor do serviço foi desenvolvido utilizando a linguagem JAVA e IDE Netbeans, versão 6.9 com auxílio do pacote `javax.jws`, específico para implementação de *Web Services* e com suporte para geração automática de códigos no Netbeans. Como servidor de aplicações foi utilizado o Tomcat 6.0.26 da Apache Foundation.

O *web service* foi desenvolvido como um *wrapper*, o que possibilitou a integração de maneira mais simples, sem muitas alterações no sistema legado, vantagem discutida na Seção 4.2.3. A integração se dá de forma que o *web service* inicia o processo do sistema legado passando os parâmetros de entrada e recebe os dados da saída padrão.

O software disponibiliza apenas uma operação, a que provê uma estação virtual de referência gerada pelo sistema legado em formato RINEX. A interface deste serviço é definida em um arquivo WSDL. A Listagem 08 apresenta a definição da operação, que pode ter mensagem de entrada e mensagem de saída ou mensagem de falha. O Apêndice A apresenta o arquivo completo.

```

1      <operation name="vrsByteRinex">
2          <input wsam:Action="http://vrsgenerator.me.org/VRSGenerator/
3              vrsByteRinexRequest" message="tns:vrsByteRinex"/>
4          <output wsam:Action="http://vrsgenerator.me.org/VRSGenerator/
5              vrsByteRinexResponse" message="tns:vrsByteRinexResponse"/>
6          <fault message="tns:Exception" name="Exception" wsam:Action="http://
7              vrsgenerator.me.org/VRSGenerator/vrsByteRinex/Fault/Exception"/>
8      </operation>

```

Listagem 08: Operação no Web Service.

Os tipos de dados são definidos em um arquivo XSD, importado no WSDL. No Apêndice B encontra-se o arquivo XSD utilizado.

O *web service* recebe os parâmetros relativos à VRS (estilo de processamento, coordenadas, data, horários inicial e final) e também um tempo limite em segundos que o software consumidor do serviço envia para sinalizar o quanto pode esperar. O software, por meio de uma *thread*, verifica se este tempo foi excedido pelo sistema legado, e neste caso envia uma mensagem de falha ao software consumidor.

O limite de tempo pode ser excedido quando o sistema legado não encontra os arquivos necessários ao cálculo, não possui tempo suficiente para fazer o download ou quando ocorre outro tipo de erro interno na aplicação.

Outros tipos de falha são transmitidos ao software cliente. A solicitação de dados com horário inicial inferior a duas horas ou horário final superior a vinte e duas não é possível, pois é necessário fazer interpolação com dados do dia anterior e posterior respectivamente, tarefa ainda não implementada no sistema legado.

Em caso de sucesso, o software consumidor recebe o arquivo RINEX na mensagem de saída.

A comunicação entre provedor o consumidor do serviço se dá utilizando o protocolo SOAP sobre HTTP. Exemplos completos de mensagens SOAP trocadas entre os mesmos podem ser encontrados no Apêndice C e no Apêndice D.

5.3 CONSUMIDOR

O software consumidor do serviço foi implementado com a finalidade de testar o uso do *web service*, trata-se de um aplicativo *desktop* construído utilizando a linguagem JAVA. O usuário preenche um formulário com os parâmetros e envia ao provedor do serviço via SOAP. Caso o provedor encaminhe uma mensagem de falha, esta é apresentada em um campo apropriado. Caso haja sucesso, o arquivo é salvo em uma pasta específica e o conteúdo é exibido ao usuário no campo RINEX. A interface com o usuário do software consumidor é mostrada na Figura 15.

VRS Client: Um consumidor do serviço VRS

Solicitação

Coordenadas Aproximadas: Data da VRS a ser gerada:

X: 3687624.376 Dia: 16

Y: -4620818.73 Mes: 10

Z: -2386880.27 Ano: 2009

Horário Inicial: Horário Final:

Hora: 2 Hora: 22

Minuto: 0 Minuto: 0

Segundo: 0 Segundo: 0

Método de Processamento: 1

Tempo Limite: min seg

4 30

Enviar

RINEX

```

2.10 OBSERVATION DATA G (GPS) RINEX VERSION / TYPE
REDESGPS 1.00 001 FCT/UNESP 16OUT2009 2: 0: 0 PGM / RUN BY / DATE
Virtual Reference Station MARKER NAME
Dani&Luiz FCT/UNESP OBSERVER / AGENCY
4912K34566 TRIMBLE NETR8 4.03 REC # / TYPE / VERS
4844A59869 TRM59800.00 NONE ANT # / TYPE
3687624.3760 -4620818.7327 -2386880.2787 APPROX POSITION XYZ
0.0000 0.0000 0.0000 ANTENNA: DELTA H/E/N
1 1 WAVELENGTH FACT L1/2
4 L1 C1 L2 P2 # / TYPES OF OBSERV
15 INTERVAL
2009 10 16 2 0 0.00000000 GPS TIME OF FIRST OBS
END OF HEADER
09 10 16 1 59 60.00000000 0 5 2 9 12 15 17
113972008.095 21688154.983 88809397.668 21688159.397
114015266.194 21696397.091 88843193.671 21696403.466
120376810.954 22906950.824 93800202.401 22906957.835
116728268.923 22212642.653 90957118.757 22212650.118
123440676.925 23489940.265 96187511.287 23489945.937
09 10 16 2 0 15.00000000 0 5 2 9 12 15 17
113936714.414 21681438.524 88781896.116 21681442.868
113979992.403 21689685.011 88815707.600 21689691.413
120356832.167 22903149.555 93784634.542 22903156.168
116765552.249 22219737.710 90986170.425 22219745.088
123490106.572 23499346.787 96226027.907 23499351.677
09 10 16 2 0 30.00000000 0 5 2 9 12 15 17

```

Retorno

VRS no formato Rinex salva no arquivo c:\Saidalsaida_cliente.txt

Nova Solicitação

Figura 15: Software consumidor do serviço.

No Capítulo 6 são apresentadas as considerações finais sobre o projeto realizado, com a apresentação das vantagens e desvantagens da solução proposta quando comparada com as soluções existentes e sugestão de trabalhos futuros.

6 CONSIDERAÇÕES FINAIS

O uso de VRS para realização de posicionamento geodésico é amplamente dominado por instituições privadas como as empresas Trimble e Leica Geosystems, que utilizam modelos matemáticos que por vezes não se adequam à realidade da atmosfera no território brasileiro. Porém instituições públicas como a UNESP tem realizado esforços no desenvolvimento de modelos adequados à realidade nacional, bem como para que os resultados possam estar disponíveis aos usuários a um custo baixo.

Um dos aspectos importantes na disponibilização destes serviços é a forma como o usuário obtém os dados. Neste trabalho, utilizando do software apresentado na Seção 3.5, desenvolvido na FCT/UNESP, foi apresentada uma nova alternativa à forma como os dados são entregues ao usuário. Para isto, utilizou-se a tecnologia baseada em *Web Services*, o que proporciona interoperabilidade entre sistemas.

Com a solução proposta, o desenvolvedor do software consumidor do serviço, seja ele um pesquisador da unidade ou uma empresa interessada, tem liberdade na implementação. A solução anterior, a qual enviava os dados via e-mail, tinha como vantagem a facilidade com que o usuário fazia a interação, necessitando apenas fazer a requisição pela internet. Por sua vez, a solução atual necessita que um software cliente seja desenvolvido, este pode ser uma página dinâmica feita pela própria unidade, ou, por exemplo, um software embarcado em um receptor móvel, que pode obter o arquivo RINEX e já realizar o posicionamento. De qualquer maneira ainda existe a possibilidade de utilizar o *web service* para interagir com uma página web e disponibilizar os dados da mesma maneira como era feito anteriormente, além das novas opções agora possíveis.

Para ilustrar o uso do serviço, um software consumidor foi desenvolvido. Ele obtém o arquivo RINEX com o *web service*, salva em uma pasta local e exibe os dados ao usuário. Caso fosse implementado em um receptor, assim como utiliza os dados para exibi-los ao usuário, o software poderia utilizá-los para realizar o posicionamento relativo, tarefa não realizada neste trabalho porque não fazia parte do escopo do mesmo, cujo objetivo era o desenvolvimento do *web service*.

A solução da empresa Trimble, apresentada na Seção 3.1, tem como desvantagem a dependência total do usuário a software e hardware desenvolvido pela empresa. Por sua vez, a Empresa Leica Geosystems possui uma solução baseada em GSM bastante interessante assim

como uma via web bastante semelhante à solução em desenvolvimento na FCT/UNESP e ao projeto SERVIR de Portugal. Também possui uma solução baseada na comunicação direta via portas TCP. Não foi possível obter informações sobre esta última, mas sabe-se que a tecnologia baseada em *Web Services* tem a vantagem de utilizar protocolos na camada de aplicação, os quais proporcionam maior independência do desenvolvedor e acoplamento fraco. Já a comunicação via sockets possui como característica o forte acoplamento.

A solução GNSmart, da Geo++, proporciona independência do dispositivo receptor. Mas a comunicação via TCP também depende de pacotes disponibilizados pela empresa.

As soluções proprietárias apresentadas, além de causarem dependência do usuário em relação às empresas, também não tem cobertura no Brasil para todos os serviços prestados ou não apresentam modelos de modelagem da troposfera e ionosfera adequados à realidade nacional.

Do exposto acima, verifica-se que o método de comunicação proposto traz contribuições tanto do ponto de vista técnico como do econômico, visando à disponibilidade dos serviços a um baixo custo para quem tiver interesse em construir o software consumidor do serviço.

6.1 TRABALHOS FUTUROS

A seguir, seguem sugestões de trabalhos futuros, a partir do realizado nesta pesquisa:

- Implementar um servidor de registro UDDI ou registrar o *web service* em um já existente, de modo a obter transparência de localização conforme recomendado no modelo SOA descrito na Seção 4.2.2.
- O sistema legado poderá ser adaptado para aplicações em tempo real, e o *web service* também pode ser utilizado para esta finalidade por meio de protocolos de tempo real como RTP (*Real-Time Transport Protocol*) e RTCP (*Real-Time Transport Control Protocol*).
- Transformar o sistema legado em um *web service*, eliminando a camada *wrapper* da arquitetura de forma a melhorar o desempenho.

Referências

AFONSO, Antonio, DIAS Rui, MARTINS Francisco, MENDES, Virgílio Brito. **O Projecto Servir do IGEOE e suas Aplicações**. 2007? Instituto Geográfico do Exército, Faculdade de Ciências da Universidade de Lisboa.

AFONSO JÚNIOR, P. C, CARDOSO J. B., CORRÊA, P. C, QUEIROZ, D. M, SAMPAIO, C. P. **Variação das dimensões características e da forma dos frutos de café durante o processo de secagem**, Revista Brasileira de Engenharia Agrícola e Ambiental. V. 6. n.3. 2002.

ALBINADER NETO, Jorge Abílio; LINS, Rafael Dueire. **Web Services em Java**. 1ª ed. Rio de Janeiro: Brasport, 2006. 288p.

ALVES, Daniele Barroca Marra. **Posicionamento GPS Utilizando Conceito de Estação Virtual**. 2008. 165f. Tese (Doutorado em Ciências Cartográficas) – Faculdade de Ciências de Tecnologia, Universidade Estadual Paulista – Universidade Estadual Paulista, Presidente Prudente.

ALVES, Daniele Barroca Marra. **Desenvolvimento e Implantação do RTK em Rede Para Posicionamento Geodésico no Estado de São Paulo**. Pós Doutorado. 2010. Em desenvolvimento.

BOTTO, Renato. **Arquitetura Corporativa de TI**. 1ª ed. Rio de Janeiro: Brasport, 2004. 248p.

CIGALA; **CIGALA Industrial Inventory Report**. v1.0. 2010.

CUNHA, Davi. **Web Services, SOAP e Aplicações Web**, 2002. Disponível em http://devedge-temp.mozilla.org/index_en.html. Acesso em novembro/2010.

EL-RABBANY, Ahmed, **Introduction to GPS: The Global Positioning System**. 1ª ed. Norwood-MA : Artech House Inc 2002.

FREIBERGER JUNIOR, Jaime, KRUEGER, Cláudia Pereira, **Posicionamento RTK Empregando Diferentes Estações de Referência**. Revista Brasileira de Cartografia No 59/02. p 137-144. Agosto 2007.

IMS Global Learning Consortium. **IMS General Web Services WSDL Bindings Guidelines**. 2005. Acesso em Novembro de 2010.

MONICO, J. F. G. **Posicionamento pelo GNSS**: Descrição, Fundamentos e Aplicações. 2.ed. São Paulo: Unesp, 2008. 476p.

NEWCOMER, Eric; LOMOW, Greg. **Understanding SOA with Web Services**. 1ª ed. NJ: Pearson Education, 2005. 444p.

SEEBER, Günter. **Satellite Geodesy: Foundations, methods, and applications**. Berlin, New York: Walter de Gruyter, 2003.

SOMMERVILLE, Ian. **Engenharia de Software**. Tradução: Selma S. Melnikoff, Reginaldo Arakaki e Edilson de Andrade Barbosa. 8ª ed. São Paulo: Pearson Addison-Wesley, 2007. 552p.

The Open Group. **Definition of SOA**. Version 1.1. 2006. Disponível em <http://opengroup.org/projects/soa/doc.tpl?gdid=10632>. Acesso em novembro/2010.

TIMBÓ, Marcos A, **Levantamentos Através do Sistema GPS**. UFMG, 2000.

WC3 – World Wide Web Consortium. **SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)**. 2007. Disponível em <http://www.w3.org/TR/soap12-part1/>. Acesso em novembro/2010.

WC3 – World Wide Web Consortium. **Web Services Architecture**. 2004. Disponível em <http://www.w3.org/TR/ws-arch>. Acesso em novembro/2010.

WC3 – World Wide Web Consortium. **Web Services Description Language (WSDL) 1.1**. 2001. Disponível em <http://www.w3.org/TR/wsdl>. Acesso em novembro/2010.

WILDE, Erik. **Web Services – Part I**. UC Berkeley. 2006. Disponível em <http://dret.net/lectures/services-fall06/webservices1>. Acesso em novembro/2010.

Apêndice A – Arquivo WSDL

A Listagem 09 apresenta o arquivo WSDL que descreve o *web service* desenvolvido neste trabalho.

Os tipos de dados são definidos na linha 4, importando um esquema de um arquivo XSD. O Apêndice B apresenta o arquivo importado.

O arquivo define 03 tipos de mensagens trocadas na comunicação com o *web service*, estas estão nas linhas de 8 a 16 e constituem nas mensagens que serão utilizadas para a entrada de dados, saída, e comunicação de falha.

No elemento *portType*, nas linhas 17 a 26, está a única operação realizada, denominada *vrsByteRinex* com sua entrada, saída e falta, mensagens descritas nos elementos *message*.

O elemento *binding*, nas linhas 27 a 41 define o protocolos utilizados na comunicação (SOAP sobre HTTP) e os elementos básicos utilizados nestes para a operação disponível no *web service*.

Finalmente, a localização do *web service* é informada na linha 44, no elemento *service*. Esta deve ser alterada caso o serviço esteja executando em uma máquina disponível para acesso externo.

```

1  <definitions targetNamespace=http://vrsgenerator.me.org/ name= VRSGeneratorService">"
2      <types>
3          <xsd:schema>
4              <xsd:import namespace="http://vrsgenerator.me.org/"
5                  schemaLocation="http://localhost:8080/VRSGenerator/VRSGenerator?xsd=1"/>
6          </xsd:schema>
7      </types>
8      <message name="vrsByteRinex">
9          <part name="parameters" element="tns:vrsByteRinex"/>
10     </message>
11     <message name="vrsByteRinexResponse">
12         <part name="parameters" element="tns:vrsByteRinexResponse"/>
13     </message>
14     <message name="Exception">
15         <part name="fault" element="tns:Exception"/>
16     </message>
17     <portType name="VRSGenerator">
18         <operation name="vrsByteRinex">
19             <input wsam:Action="http://vrsgenerator.me.org/VRSGenerator/
20                 vrsByteRinexRequest" message="tns:vrsByteRinex"/>
21             <output wsam:Action="http://vrsgenerator.me.org/VRSGenerator/
22                 vrsByteRinexResponse" message="tns:vrsByteRinexResponse"/>
23             <fault message="tns:Exception" name="Exception" wsam:Action="http:/
24                 /vrsgenerator.me.org/VRSGenerator/vrsByteRinex/Fault/Exception"/>
25         </operation>
26     </portType>
27     <binding name="VRSGeneratorPortBinding" type="tns:VRSGenerator">
28         <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
29         <operation name="vrsByteRinex">
30             <soap:operation soapAction=""/>
31             <input>
32                 <soap:body use="literal"/>
33             </input>
34             <output>
35                 <soap:body use="literal"/>
36             </output>
37             <fault name="Exception">
38                 <soap:fault name="Exception" use="literal"/>
39             </fault>
40         </operation>
41     </binding>
42     <service name="VRSGeneratorService">
43         <port name="VRSGeneratorPort" binding="tns:VRSGeneratorPortBinding">
44             <soap:address location="http://localhost:8080/VRSGenerator/VRSGenerator"/>
45         </port>
46     </service>
47 </definitions>

```

Listagem 09: WSDL.

Apêndice B – Arquivo XSD

A Listagem 10 define o esquema com os tipos de dados que serão utilizados na comunicação com o *web service*.

Cada mensagem é um tipo, conforme linhas 2 a 4. *vrsByteRinex* é um tipo composto formado pelos elementos usados como parâmetros de entrada que serão passados ao sistema legado (x, y, z, dia, mês, etc) e o tempo limite que também é recebido pelo *web service*.

Exception e *vrsByteRinexResponse* também são tipos complexos (linhas 22 a 33).

```

1  <xs:schema version="1.0" targetNamespace="http://vrsgenerator.me.org/">
2    <xs:element name="Exception" type="tns:Exception"/>
3    <xs:element name="vrsByteRinex" type="tns:vrsByteRinex"/>
4    <xs:element name="vrsByteRinexResponse" type="tns:vrsByteRinexResponse"/>
5    <xs:complexType name="vrsByteRinex">
6      <xs:sequence>
7        <xs:element name="estiloProcessamento" type="xs:int"/>
8        <xs:element name="x" type="xs:double"/>
9        <xs:element name="y" type="xs:double"/>
10       <xs:element name="z" type="xs:double"/>
11       <xs:element name="dia" type="xs:int"/>
12       <xs:element name="mes" type="xs:int"/>
13       <xs:element name="ano" type="xs:int"/>
14       <xs:element name="horaInicial" type="xs:int"/>
15       <xs:element name="minutoInicial" type="xs:int"/>
16       <xs:element name="segundoInicial" type="xs:int"/>
17       <xs:element name="horaFinal" type="xs:int"/>
18       <xs:element name="minutoFinal" type="xs:int"/>
19       <xs:element name="segundoFinal" type="xs:int"/>
20       <xs:element name="tempoLimite" type="xs:int"/>
21     </xs:sequence>
22   </xs:complexType>
23   <xs:complexType name="vrsByteRinexResponse">
24     <xs:sequence>
25       <xs:element name="return" type="xs:base64Binary" nillable="true"
26         minOccurs="0"/>
27     </xs:sequence>
28   </xs:complexType>
29   <xs:complexType name="Exception">
30     <xs:sequence>
31       <xs:element name="message" type="xs:string" minOccurs="0"/>
32     </xs:sequence>
33   </xs:complexType>
34 </xs:schema>

```

Listagem 10: XSD.

Apêndice C – Requisição SOAP ao Web Service

A Listagem 11 demonstra como um consumidor do serviço pode fazer uma requisição ao *web service*.

Neste caso, não é necessário o uso do *Header*. Porém este poderia ser usado para a sincronização das mensagens.

No elemento *Body*, são especificados de maneira hierárquica os valores para cada tipo de dado utilizado na mensagem enviada em conformidade com o que foi definido na descrição WSDL.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
3    <S:Header/>
4    <S:Body>
5      <ns2:vrsByteRinex xmlns:ns2="http://webservice.me.org/">
6        <estiloProcessamento>1</estiloProcessamento>
7        <x>3687624.376</x>
8        <y>-4620818.7327</y>
9        <z>-2386880.2787</z>
10       <dia>16</dia>
11       <mes>9</mes>
12       <ano>2009</ano>
13       <horaInicial>6</horaInicial>
14       <minutoInicial>0</minutoInicial>
15       <segundoInicial>0</segundoInicial>
16       <horaFinal>7</horaFinal>
17       <minutoFinal>0</minutoFinal>
18       <segundoFinal>0</segundoFinal>
19       <tempoLimite>30</tempoLimite>
20     </ns2:vrsByteRinex>
21   </S:Body>
22 </S:Envelope>
```

Listagem 11: Requisição usando SOAP.

Apêndice D – Resposta SOAP do Web Service

A Listagem 12 traz um exemplo de resposta do *web service* a uma requisição. Da mesma forma que na requisição, o elemento *Body* contém a especificação de cada tipo de dado enviado na mensagem.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
3    <S:Body>
4      <ns2:vrsByteRinexResponse xmlns:ns2="http://webservice.me.org/">
5        <return>AQWEUN2348ASDJAKSDJAQYRIWEJHVUEBLJ==</return>
6      </ns2:vrsByteRinexResponse>
7    </S:Body>
8  </S:Envelope>
```

Listagem 12: Resposta usando SOAP.

Apêndice E – Instalação/Configuração do Sistema

Para este trabalho, utilizou-se uma máquina com o sistema operacional Windows Vista Home Premium.

A pasta do sistema legado alterado conforme descreve a Seção 5.1, denominada VRS, deve ser copiada em **C://**.

O Web Service foi implementado utilizando-se a linguagem Java, JDK 1.6.0_05, por meio do IDE Netbeans, versão 6.9. O arquivo **VRSGenerator.war** gerado na pasta **dist** é utilizado pelo servidor de aplicações web para fornecer o serviço.

Como servidor de aplicações web, foi utilizado o Tomcat, versão 6.0.26, da Apache Softwer Foundation. Trata-se de uma ferramenta *free*. Para iniciá-lo é necessário executar o arquivo **startup.bat**, no diretório **bin** da instalação do mesmo.

Para implantar o serviço no Tomcat, é necessário acessar o endereço **http://localhost:8080**, e clicar na opção Tomcat Manager. O acesso somente será permitido mediante o fornecimento do login e senha de um usuário com *role* (papel) “*manager*” previamente definido no arquivo **conf/tomcat-users.xml**.

Obtendo acesso, basta ir à Seção Deploy, clicar no botão “**Enviar Arquivo**”, escolher o arquivo **VRSGenerator.war** e clicar no botão “**Deploy**”.

Para utilizar o software cliente, basta executar o arquivo **VRSClient.jar** na respectiva pasta **dist**. Caso o serviço esteja sendo oferecido em outra máquina, o arquivo WSDL na pasta **VRSGenerator** do cliente deve ser alterado para o endereço correto no elemento *soap:adress*.