



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Disciplina SER 300 - Introdução ao Geoprocessamento.

Trabalho: Implementação da especificação OpenLS, usando TerraLib

Marcio Azeredo - 105953

Junho de 2008

SUMÁRIO

1.	INTRODUÇÃO.....	03
2.	SERVIÇOS PARA INTERCÂMBIO DE DADOS NA WEB.....	05
2.1.	WEB SERVICES E INTEROPERABILIDADE.....	05
2.2.	<i>EXTENSIBLE MARKUP LANGUAGE E OPEN GEOSPATIAL CONSORTIUM.</i>	06
2.3.	OpenGIS® <i>LOCATION SERVICES</i> (OpenLS).....	08
3.	IMPLEMENTAÇÃO MÍNIMA DO SERVIÇO.....	13
3.1.	ARQUITETURA PROPOSTA.....	13
3.2.	REQUISIÇÕES E RESPOSTAS (<i>REQUESTS AND RESPONSES</i>).....	15
4.	CONCLUSÕES E TRABALHOS FUTUROS.....	18
	REFERENCIAS BIBLIOGRÁFICAS.....	19

ÍNDICE DE FIGURAS

Figura 1.1: Interoperable 3D Emergency Routing Based on OpenLS	04
Figura 2.1: Sistemas em um cenário sem interoperabilidade.	05
Figura 2.2: Sistemas em um cenário interoperável.	06
Figura 2.3: Exemplo da estrutura de um arquivo XML.	07
Figura 2.4: Exemplo de uma requisição do <i>Directory Service</i>	10
Figura 2.5: Exemplo de uma requisição ao <i>Presentation Service</i> .	11
Figura 2.6: Modelo de serviços OpenLS.	12
Figura 3.1: Arquitetura proposta.	13
Figura 3.2: Exemplo de classes implementadas.	14
Figura 3.3: Exemplo de requisição do serviço <i>Geocode</i> .	17
Figura 3.4: Exemplo de resposta do serviço <i>Geocode</i> .	17

CAPÍTULO 1

INTRODUÇÃO

O mapeamento, em tempo real, de eventos espacialmente definidos sobre a superfície terrestre, é de grande importância para aplicações de geoprocessamento. Estes sistemas têm um papel fundamental no apoio à tomada de decisão atuando em diferentes níveis de gerenciamento, atendendo interesses públicos e privados. A seguir são apresentadas algumas aplicações:

- Planejamentos de rotas de salvamento para vítimas de catástrofes;
- Planejamento de tráfego rodoviário em centros urbanos;
- Localização, análise e avaliação de riscos em saúde pública;
- Localização e análise de desmatamentos em áreas de proteção ambiental;
- Localização de produtos e serviços comerciais.

Devido à dinâmica atual das informações e muitas vezes, a dependência de variáveis externas que influenciam no problema, há a necessidade do aproveitamento de dados provenientes de outras fontes. Um determinado órgão público de saúde, por exemplo, pode ter a necessidade de considerar em seus estudos, dados sócio-econômicos que não são de sua responsabilidade. Neste caso, é aconselhável que seus sistemas utilizem tecnologias que permitam esse intercâmbio de informações para a correção e direcionamento de suas políticas.

Atualmente, uma solução que vem se mostrando convidativa é a utilização de dispositivos móveis de comunicação, como celulares e PDAs, utilizando redes sem fio. As tecnologias de localização têm crescido e já existem grupos de desenvolvimento envolvidos na elaboração de especificações para padronização desse tipo de serviço.

Um exemplo de sucesso de implementação de um serviço de localização é o *Emergency Route Service 3D* (Figura 1.1). Trata-se de um sistema alemão utilizado para determinar rotas de salvamento baseado no registro das ocorrências de desastres. Estes registros são utilizados para gerar rotas alternativas para as equipes de resgate.

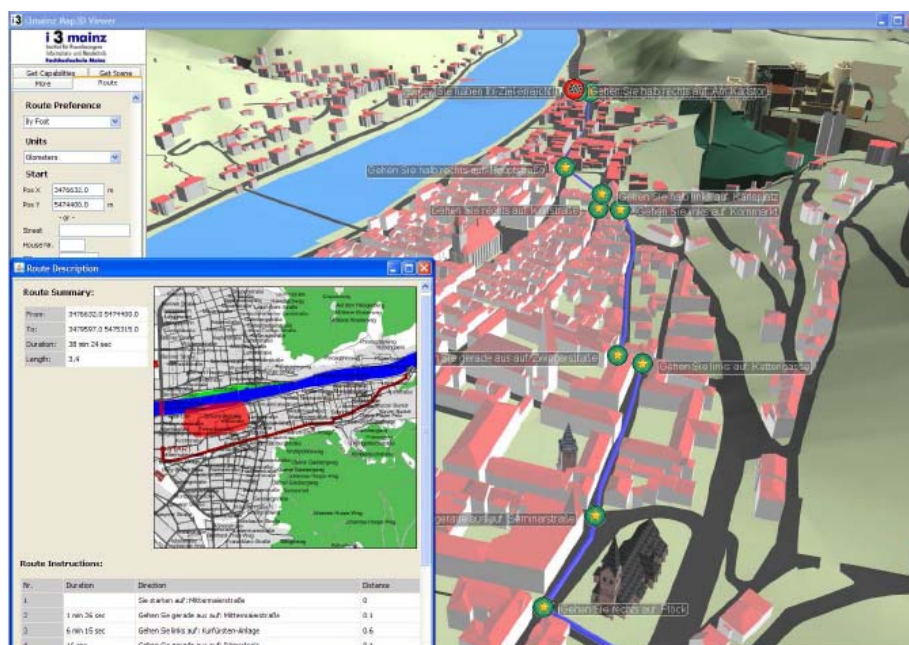


Figura 1.1 – *Interoperable 3D Emergency Routing Based on OpenLS*

Dentro dessa linha de pesquisa, o objetivo deste trabalho é justamente desenvolver uma biblioteca de funções em C++ para prover um serviço mínimo de localização, baseada na especificação *OpenGIS® Location Services* (OpenLS) elaborada pelo *Open Geospatial Consortium* (OGC®).

CAPÍTULO 2

SERVIÇOS PARA INTERCÂMBIO DE DADOS NA WEB

2.1 *Web Services* e Interoperabilidade

Os *Web Services*, ou Serviços Web, são aplicações de software que têm por objetivo permitir, de uma forma interoperável, a comunicação e o intercâmbio de informações entre diferentes sistemas na internet. Em outras palavras, permitir que um sistema “converse” com outro sistema semelhante, ou não, independente de implementações de hardware ou de software.

Na figura 2.1 é possível observar um cenário no qual dois sistemas de computadores (A e B) tentam comunicar-se, mas não conseguem devido à ausência de padrões no intercâmbio das informações. Mesmo que um dos sistemas receba os dados provenientes do outro, não conseguirá processá-los, pois desconhece a estrutura de construção do arquivo recebido.



Figura 2.1 – Sistemas em um cenário sem interoperabilidade.

Observando um novo cenário (figura 2.2), os mesmos sistemas descritos anteriormente, agora apóiam-se no uso de serviços para padronizar os protocolos de comunicação e a forma de encapsulamento dos seus dados, garantindo assim a interoperabilidade.



Figura 2.2 – Sistemas em um cenário interoperável.

Uma grande vantagem de um *Web Service* é que ele fornece interoperabilidade entre componentes de *software* que podem estar em localizações e infra-estruturas diferentes (Xavier *et al*, 2006). A padronização é conseguida a nível de sistemas, e não de dados. Com isso, o Sistema A não precisa conhecer como os dados estão armazenados no Sistema B, ou até mesmo que linguagem de programação foi utilizada no desenvolvimento do serviço. Conhecida a forma com que as informações foram encapsuladas, é possível realizar o processo de extração das informações sem que haja perda de dados.

Atualmente o padrão mais utilizado para intercâmbio de dados entre aplicações que buscam a interoperabilidade, é a XML (*Extensible Markup Language*).

2.2 *eXtensible Markup Language*(XML) e *Open Geospatial Consortium* (OGC®)

A XML surgiu da necessidade de utilização de uma nova linguagem, totalmente extensível e estruturada, que combinasse a flexibilidade da SGML (*Standard Generalized Markup Language*) com a simplicidade da HTML (*HyperText Markup Language*). Tem por objetivo descrever os dados e seus metadados de forma estruturada por intermédio de marcações (Figura 2.3).

```

<complexType name="QualityOfPositionType">
  <sequence>
    <element name="HorizontalAcc" type="xls:HorAccType" />
    <element name="VerticalAcc" type="xls:VerAccType" />
  </sequence>
  <attribute name="responseReq" use="optional" default="Delay_Tol">
    <simpleType>
      <restriction base="string">
        <enumeration value="No_Delay" />
        <enumeration value="Low_Delay" />
        <enumeration value="Delay_Tol" />
      </restriction>
    </simpleType>
  </attribute>
  <attribute name="responseTimer" type="string" use="optional" />
</complexType>

```

Figura 2.3 – Exemplo da estrutura de um arquivo XML.

O número de marcações possíveis de construção é ilimitado, ficando a critério do próprio usuário de acordo com as necessidades de suas aplicações.

Atualmente, a XML é a linguagem recomendada para o intercâmbio de dados pela *World Wide Web Consortium* (W3C), sendo utilizada por outras organizações com o mesmo objetivo de facilitar a interoperabilidade entre sistemas. Um desses exemplos é o *Open Geospatial Consortium* (OGC®).

O OGC® é um consórcio formado por mais de 300 participantes, entre empresas, agências governamentais e universidades. Foi criado com o objetivo de promover o desenvolvimento de tecnologias que facilitem a interoperabilidade entre sistemas envolvendo informação geo-espacial (Gardels, 1996).

Segundo (Aulicino, citado por Xavier et al, 2006), são três as preocupações básicas do OGC® que visam auxiliar a interoperabilidade e extensão das aplicações de SIG:

- Ambiente de aplicação interoperável: uma interface configurável que utiliza ferramentas específicas e dados necessários para resolver um problema.

- Espaço de dados compartilhados: um modelo de dados genérico que suporta uma grande variedade de aplicações analíticas e cartográficas.
- Navegação heterogênea às informações: métodos para acessar e explorar as informações e aplicações analíticas disponíveis na rede.

Para atingir seus objetivos, o OGC® especifica um conjunto de normas (documentos) que padronizam a construção de serviços na internet para diferentes tipos de aplicações que utilizam dados geo-espaciais. Uma dessas especificações em particular (OpenLS) é alvo deste trabalho e descreve como construir sistemas para prover serviços de localização.

2.3 OpenGIS® Location Services (OpenLS)

Nos últimos anos houve um grande avanço tecnológico na área dos dispositivos móveis, como celulares e PDAs, atingindo um grande número de usuários em todo o mundo. De olho nas aplicações que poderiam usufruir dessa tecnologia, o OGC® criou uma especificação para a construção de serviços de localização. Essa especificação permitiria o intercâmbio de informações espaciais de forma interoperável. Estava criada a especificação OpenLS.

A especificação OpenLS é composta por cinco serviços básicos, chamados de *Core Services*, também conhecidos como *GeoMobility Server*(GMS). Seu objetivo principal é definir o acesso aos serviços básicos e aos tipos de dados que os compõem. Os serviços básicos que compõem o núcleo de serviços OpenLS são apresentados a seguir:

Location Utility Service

Este serviço é considerado o coração do OpenLS, sendo subdividido em duas partes: *Geocode Service* e *Reverse Geocode Service*.

O *Geocode Service* tem por objetivo a determinação de uma localização geográfica, com base em um nome, endereço ou código postal. No caso do *Reverse Geocode Service* a tarefa é a inversa. Recebendo um endereço normalizado, retorna a latitude e longitude da localização no Datum WGS84. Ambas as requisições podem retornar zero, uma, ou mais respostas a uma solicitação realizada, dependendo dos parâmetros da requisição e dos algoritmos empregados. Uma vez determinada a localização geográfica pelo *Geocode Service*, com base em um endereço, esta informação pode ser utilizada por outros serviços que serão descritos mais adiante, como o *Directory Service* ou o *Route Service*.

Directory Service

O *Directory Service* fornece aos assinantes com acesso a um diretório on-line, a possibilidade de encontrar a localização mais próxima de um produto ou serviço específico. Para tal, o assinante formula uma requisição com base nos parâmetros previstos na especificação, identificando o local, produto ou serviço desejado por meio do nome, tipo, categoria, palavra-chave ou número de telefone. No caso da procura de um ponto mais próximo, também deverá ser informada a posição, que poderá ser a posição atual do dispositivo móvel determinada por intermédio da *Gateway Service*, ou de alguma outra forma.

Com base no pedido formulado, o *Directory Service* busca os diretórios on-line mais indicados a satisfazer o pedido, e retorna uma, ou mais respostas à consulta, com localizações e descrições do local, produto ou serviço, dependendo do conteúdo do diretório. As respostas são classificadas com base nos critérios utilizados na pesquisa. Um usuário pode requisitar, por exemplo, a localização de um serviço de aluguel de carros, ou dos restaurantes chineses próximos ao seu hotel (Figura 2.4).

```

<DirectoryRequest>
  <POILocation>
    <Nearest>
      <POI ID="1">
        <POIAttributeList>
          <POIInfoList>
            <POIInfo name="POI Name" value="My Hotel"/>
          </POIInfoList>
        </POIAttributeList>
      </POI>
    </Nearest>
  </POILocation>
  <POIProperties directoryType="Yellow Pages">
    <POIProperty name="NAICS_type" value="Restaurant"/>
    <POIProperty name="NAICS_subType" value="Chinese"/>
  </POIProperties>
</DirectoryRequest>

```

Figura 2.4 - Exemplo de uma requisição do *Directory Service*.

Em todos os serviços especificados no OpenLS existem parâmetros de consulta obrigatórios e outros opcionais, de acordo com as necessidades do usuário.

Route Service

Este serviço possibilita a determinação uma rota com base em um ponto inicial (normalmente adquirida por intermédio do *Gateway Service*), e de um ponto final (qualquer localização, como por exemplo, um lugar para o qual o usuário só possui um número de telefone ou um endereço fornecido pelo *Directory Service*). O usuário pode ainda especificar pontos intermediários (*waypoints*), uma via preferencial (mais rápida, mais curta, com menos tráfego, etc), e até o modo de transporte preferido.

Gateway Service

O *Gateway Service* é responsável pela obtenção da posição do terminal móvel do assinante a partir da rede de comunicação da operadora. Uma vez determinada a

posição, o serviço envia o resultado para ser exibido na tela do dispositivo pelo *Presentation Service*.

Presentation Service

Rendeniza a informação geográfica para visualização na tela do dispositivo móvel. Qualquer serviço de localização pode utilizar o *Presentation Service* para obter a representação da área desejada, com uma ou mais camadas de informação. Este serviço também é empregado ainda para a renderizar instruções de rotas. Na requisição são definidos parâmetros de saída do mapa (Figura 2.5).

```
<PortrayMapRequest>
  <Output width="640" height="480" format="image/png">
    <gml:Envelope>
      <gml:pos>-114.342 50.234</gml:pos>
      <gml:pos>-114.123 50.031 </gml:pos>
    </ gml:Envelope >
  </Output>

  <!-- Ask the impl to exclude no layers (thus, create the base map with all your layers) -->
  <Basemap filter="Exclude" />

  <Overlay>
    <Position>
      <!-- This ADT contains the lat/lon of Joe's House -->
    </Position>
  </Overlay>
</ PortrayMapRequest >
```

Figura 2.5 - Exemplo de uma requisição ao *Presentation Service*.

O modelo de serviços OpenLS pode ser observado na figura 2.6.

The OpenLS Information Model

Position ADT	Point location in well-known coordinate system
Address ADT	Street address or intersection
Point of Interest (POI) ADT	The location where someone can find place, product or service
Area of Interest (AOI) ADT	A polygon, bounding box or circle used as a search template
Location ADT	A location (Position, Address or POI)
Map ADT	The portrayal of maps and feature overlays (routes & POI)
Route Summary ADT	Metadata pertaining to a route
Route Geometry ADT	Geometry data for a route
Route Maneuvers ADT	Navigation maneuver data for a route
Route Directions ADT	Turn-by-turn navigation instructions for a route

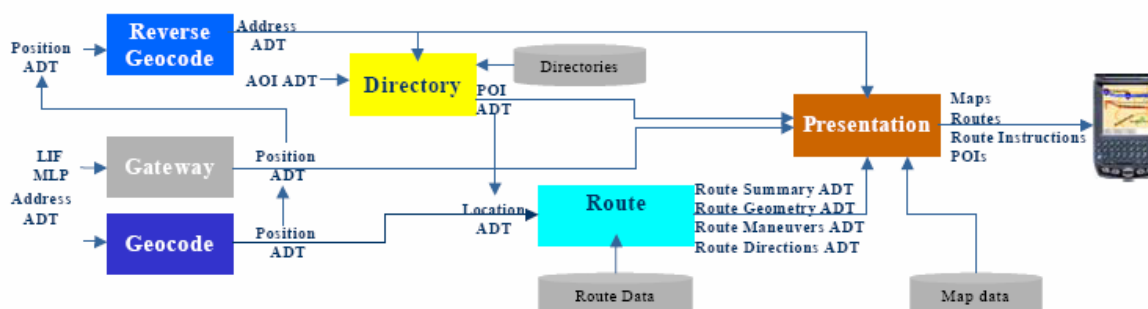


Figura 2.6 – Modelo de serviços OpenLS.

Fonte: OGC Implementation Specification 05-016

CAPÍTULO 3

IMPLEMENTAÇÃO MÍNIMA DO SERVIÇO

Uma implementação mínima da especificação OpenLS pode ser obtida somente trabalhando com o *Location Utility Service* (*Geocode Service* e *Reverse Geocode Service*). Logo, somente esses serviços foram alvos de implementação deste trabalho.

A linguagem de programação escolhida para implementação dos serviços foi a C++, tendo em vista facilitar a integração com a biblioteca de funções TerraLib desenvolvida no INPE, escrita nessa mesma linguagem.

3.1 Arquitetura Proposta

A arquitetura proposta para teste pode ser observada na figura 3.1.

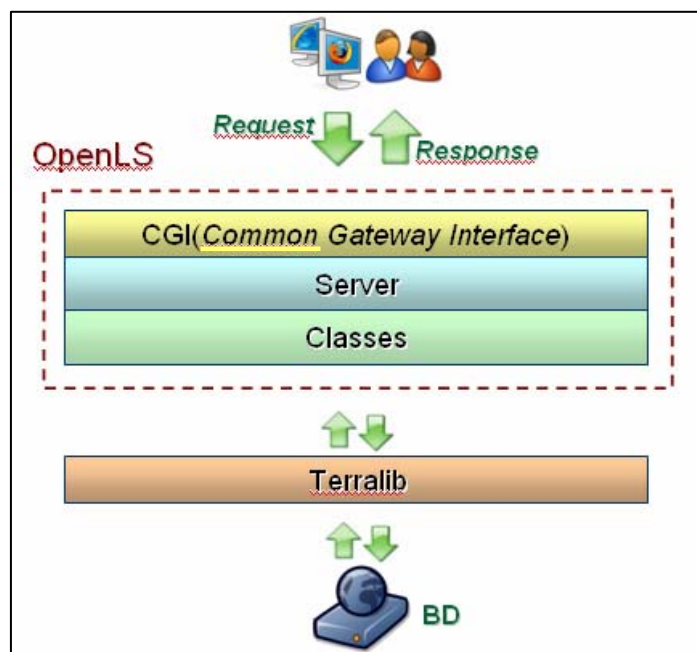


Figura 3.1 – Arquitetura proposta.

Sistema Gerenciador de Banco de Dados (SGBD)

Pode ser utilizado qualquer SGBD que possua *driver* de conexão implementado na TerraLib. Para os testes realizados foram utilizados o PostgreSQL 8.2 e sua extensão espacial PostGIS 1.3.3.

Construção de Classes

A especificação OpenLS é composta por treze esquemas XML. Cada um definindo a forma de construção dos arquivos XML contendo os objetos e respectivos atributos previstos na especificação.

Com base nos esquemas da especificação foram construídas as classes necessárias para o funcionamento da implementação considerada básica, e ainda as classes previstas no *Directory Service* e *Presentation Service*. Um exemplo de classes construídas pode ser observado na figura 3.2.

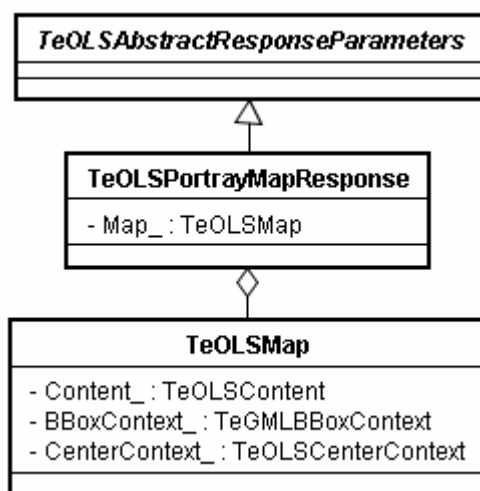


Figura 3.2 – Exemplo de classes implementadas.

Foram construídas aproximadamente 60 classes, sendo que cada uma possui um arquivo de cabeçalho “.h” e um arquivo de implementação “.cpp”.

Camada Server

Nesta camada ficam as implementações das funções que irão efetivamente realizar as operações de requisição e resposta, assim como algumas implementações dos algoritmos de busca e de endereçamento.

Camada CGI

A *Comum Gateway Interface* (CGI) é uma tecnologia que permite gerar páginas web de forma dinâmica e em tempo real. O navegador passa os parâmetros para um programa executável que roda no servidor web, e este executa o código. No caso deste trabalho foi utilizada a linguagem C++ por motivos já citados anteriormente.

Após a criação das classes e da camada *Server*, todo o projeto é compilado e transformado em um arquivo executável. Esse é o arquivo utilizado para receber os parâmetros por intermédio do navegador. No caso específico deste trabalho, o arquivo recebe como parâmetro um arquivo XML contendo uma requisição do serviço.

Camada Cliente

A camada do cliente compreende os processos do sistema que são executados no navegador do usuário. Para este trabalho, foram implementadas funções escritas na linguagem javascript que montam os arquivos XML com base em informações contidas em formulários HTML.

3.2 Requisições e Respostas (*Requestes and Responses*)

A comunicação entre o usuário e o serviço é dada por intermédio de requisições e respostas. O usuário requisita, o sistema processa e responde à requisição. A execução pode ser resumida da seguinte forma:

- 1) O usuário preenche um formulário HTML informando o endereço no caso do *Geocode Service*, ou uma localização no caso do *Reverse Geocode Service*.
- 2) Os dados são transformados em um arquivo XML por funções escritas na linguagem javascript, rodando no próprio navegador do usuário.
- 3) Uma vez transformados os dados de consulta em um arquivo XML, o mesmo é passado como parâmetro para o arquivo executável rodando no servidor (Figura 3.3).
- 4) Dentro do arquivo executável existem funções que verificam qual o tipo de requisição está sendo recebida. O arquivo XML é então decomposto e suas marcações são interpretadas segundo as especificações OpenLS.
- 5) As operações requeridas sobre os dados são executadas pelas classes implementadas e por outras classes já disponíveis na TerraLib. São essas as classes quem possuem os algoritmos de busca.
- 6) Terminadas as operações, é montado no servidor, o arquivo XML de resposta às requisições realizadas (Figura 3.4).
- 7) O arquivo contendo as informações de resposta à requisição é exibido no navegador do usuário.

```

<GeocodeRequest>
  <Address countryCode="BR">
    <StreetAddress>
      <Building number="3020" subdivision="APT 804" />
      <Street typePrefix="AVENIDA" officialName="CRUZEIRO DO SUL"/>
    </StreetAddress>
    <PostalCode>12228-615</PostalCode>
  </Address>
</GeocodeRequest>

```

Figura 3.3 – Exemplo de requisição do serviço Geocode.

```

<GeocodeResponse>
  <GeocodeResponseList numberOfGeocodedAddresses="1">
    <GeocodedAddress>
      <gml:Point>
        <gml:pos>114.324 50.232</gml:pos>
      </gml:Point>
      <Address countryCode="BR">
        <StreetAddress>
          <Building number="3020" subdivision="APT 804" />
          <Street typePrefix="AVENIDA" officialName="CRUZEIRO DO SUL"/>
        </StreetAddress>
        <PostalCode>12228-615</PostalCode>
      </Address>
    </GeocodedAddress>
  </GeocodeResponseList>
</GeocodeResponse>

```

Figura 3.4 – Exemplo de resposta do serviço Geocode.

CAPÍTULO 4

CONCLUSÕES E TRABALHOS FUTUROS

O objetivo do trabalho foi atingido, uma vez que foi possível o desenvolvimento de uma biblioteca de funções em C++ para prover um serviço mínimo de localização, baseada na especificação OpenLS. Vale lembrar que foram implementados somente os serviços *Geocode* e *Reverse Geocode*. Os demais serviços em algum momento podem necessitar de variáveis externas que merecem de um estudo mais detalhado.

Como sugestão para trabalhos futuros, fica a implementação do restante dos serviços OpenLS não implementados neste trabalho (*Directory Service*, *Gateway Service*, *Route Service* e *Presentation Service*), e ainda o melhoramento dos algoritmos de localização com o objetivo da obtenção de respostas mais precisas.

REFERENCIAS BIBLIOGRÁFICAS

- MABROUK, M. **OpenGIS Location Services (OpenLS): Core Services**. Version 1.1, OGC 05-016. Wayland: Open Geospatial Consortium, 2005. 174 p. Disponível em: <http://portal.opengeospatial.org/files/?artifact_id=8836>. Acesso em: 17 jun. 2008.
- COX, S.; DAISEY, P.; LAKE, R.; PORTELE, C.; WHITESIDE, A. **OpenGIS® Geography Markup Language (GML) Implementation Specification**. Version: 3.1.0, OGC 03-105r1. Wayland: Open Geospatial Consortium, 2004. 601 p. Disponível em: <http://portal.opengeospatial.org/files/?artifact_id=4700>. Acesso em: 17 jun. 2008.
- BRAY, T.; PAOLI, J.; SPERBERG-MCQUEEN, C. M.; MALER, E.; YERGEAU, F. **Extensible Markup Language (XML) 1.0**. 4th ed.. [S.l.]: World Wide Web Consortium, 2006. p. Disponível em: <<http://www.w3.org/TR/2006/REC-xml-20060816/>>. Acesso em: 17 jun. 2008.
- BORGES, K. A. V. **Uso de uma Ontologia de Lugar Urbano para Reconhecimento e Extração de Evidências Geo-espaciais na Web**. 195 p. (). Tese (Doutorado em Ciência da Computação) - Universidade Federal de Minas Gerais, Belo Horizonte, 2006.
- NEIS, P.; SCHILLING, A.; ZIPF, A. **Interoperable 3D Emergency Routing Based on OpenLS** Disponível em: < <http://www.geographie.uni-bonn.de/karto/GI4D2007.3DEmergencyRouteService.pdf>>. Acesso em: 17 jun. 2008.
- AULICINO, L. C. M. **WISS - serviço Web para segmentação de imagens: especificação e implementação**. 116 p. (INPE-14609-TDI/1189). Dissertação (Mestrado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2006.
- APACHE XML PROJECT - part of The Apache Software Foundation, <<http://xml.apache.org>> Acesso em: 17 jun. 2008.
- XAVIER, E.; DE SOUZA, VANESSA C. O. (2006). **Implementação de web services no padrão OGC, usando TerraLib**. Computação Aplicada. São José dos Campos, INPE. **Trabalho da Disciplina Introdução ao Geoprocessamento**.
- CÂMARA, G. **Biblioteca TerraLib** – INPE/DPI, 2000. Disponível em <<http://dpi.inpe.br/terralib>>. Acesso em: 17 jun. 2008.
- CASANOVA, M.A., G. Câmara, C. Davis, L. Vinhas, G. Queiroz. **Bancos de Dados Geográficos**. Editora MundoGEO, Curitiba, 2005.
- DAVIS, C.; BORGES, K.A.V.; SOUZA, L.A.; CASANOVA, M.A.; JÚNIOR, P.O.L. **Disseminação de dados geográficos na Internet**. In: CASANOVA, M.A.; CÂMARA, G.; DAVIS, C.A.; VINHAS, L.; QUEIROZ, G.R. Banco de Dados Geográficos. Curitiba: MundoGEO, maio de 2005b. cap. 10
- DAVIS, C.; BORGES, K.A.V.; SOUZA, L.A.; CASANOVA, M.A.; JÚNIOR, P.O.L. **O Open Geospatial Consortium**. In: CASANOVA, M.A.; CÂMARA, G.; DAVIS, C.A.; VINHAS, L.; QUEIROZ, G.R. Banco de Dados Geográficos. Curitiba: MundoGEO, maio de 2005b. cap. 11
- DAVIS, C.; CÂMARA, G. **Arquitetura de Sistemas de Informação Geográfica**. In: CÂMARA, G.; DAVIS, C.; MONTEIRO, A.M.V.(Org). Introdução à ciência da geoinformação. São José dos Campos:INPE, out. 2001. cap.3.