

6 Conclusão e Trabalhos Futuros

6.1. Resultados

O resultado principal deste trabalho consistiu em proporcionar à TerraLib interoperabilidade com outros sistemas através do protocolo OpenGIS WMS.

No capítulo 2, contextualizamos o trabalho e apresentamos uma breve revisão bibliográfica.

No capítulo 3, discutimos a escolha do protocolo OpenGIS e, mais especificamente, a escolha do WMS. Este pode ser considerado o primeiro passo para tornar a TerraLib compatível com o padrão OpenGIS, proporcionando assim interoperabilidade ampla, através de um protocolo relativamente simples e muito bem definido, com vários sistemas de GIS existentes atualmente. Neste capítulo apresentamos ainda a TerraLib e o *framework* TDK, que proporciona um manuseio de mais alto-nível dos dados geográficos.

No capítulo 4, descrevemos em detalhe o protocolo WMS, mostrando desde seus elementos até o seu comportamento para estabelecimento da comunicação entre o cliente e o servidor.

No capítulo 5, cobrimos as questões referentes à implementação do servidor WMS, incluindo requisitos, arquitetura, diagramas e a forma de utilização.

6.2. Trabalhos Futuros

Como sugestão de trabalhos futuros, podemos citar:

- implementar outros Web Services definidos pelo OpenGIS, como WFS e WCS, usando o *framework* OGCWebService e estendendo o WMSTerraLib;
- implementar um cliente WMS;

Como o processamento na TerraLib para criação da resposta à operação GetMap foi o que consumiu maior tempo em todo o processo, sugerimos mais especificamente as seguintes formas de melhorá-lo:

- criar um mecanismo de *cache* no *framework*, evitando acessos desnecessários à base TerraLib;
- criar um mecanismo de *cache* no *canvas*, possibilitando ligar ou desligar os temas visíveis. Em particular, é importante manter a ordem dos temas no *canvas*, ou seja, deve ser possível ligar ou desligar os temas no *canvas*, sem alterar a sua ordem no *canvas*, a menos que seja solicitada explicitamente esta mudança;
- manter um controle de sessão e um *cache* entre o servidor CORBA e o cliente CORBA para que as requisições de um cliente possam, eventualmente, ser aproveitadas por outro;
- manter um controle de sessão e um *cache* entre o servidor do Web Service e o cliente do Web Service, armazenando as requisições e seus resultados, evitando eventualmente acesso ao servidor CORBA. Existem muitas requisições que podem se repetir e algumas que se completam. Por exemplo, em GetFeatureInfo, o usuário requisita atributos de um tema, depois de outro e em seguida dos dois temas juntos. Assim, a última requisição pode ser composta com o resultado das duas primeiras;
- usar outra tecnologia mais próxima de J2EE, além de struts, para realizar o controle do sistema e implementar o *framework*, (atualmente implementado usando Tomcat e servlets).