

Uma Proposta de Interface Operacional entre SIG e SGBDR para Dados Convencionais

ROGÉRIO THOMÉ¹
FERNANDO YAMAGUCHI¹
MARISA DA MOTA¹

¹ INPE - Instituto Nacional de Pesquisas Espaciais
São José dos Campos - SP, Brasil
thome@ltid.inpe.br
{yama,marisa}@dpi.inpe.br

Abstract. This paper approaches the problem of building an interface between Geographical Information Systems-GIS and Relational Data Base Management System-RDBMS for conventional data. The paper proposes an interface to turn a GIS more portable for RDBMS users in general. The background for it come from the development of an interface between the software Spring (a GIS) and Oracle v.7.1. Also it will be shown characteristics of each system and the results obtained up to.

Keywords: GIS, RDBMS, Interface.

1 Introdução

Os sistemas de informações geográficas podem ser implementados usando uma estratégia dual, onde os objetos gráficos tem suas respectivas representações geométricas armazenadas em sistemas de arquivos e seus atributos convencionais armazenados em um sistema gerenciador de banco de dados relacional - SGBDR. A conexão entre os dois sistemas é feita através de um identificador que liga os componentes geométricos e convencionais do objeto gráfico.

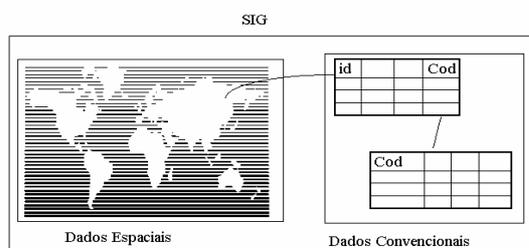


figura 1

Desta forma torna-se possível o uso de diversos SGBDR de mercado, com independência do fabricante.

Este trabalho aborda a construção da interface entre SIG e SGBDR, sob a perspectiva dual, para dados convencionais. Apresenta-se uma solução adotada para a interface Spring (Sistema de Processamento de Informações Georeferenciadas) desenvolvido pelo INPE e o SGBDR Oracle.

2 Modelo de Dados

O Spring tem sua concepção baseado no paradigma orientado por objetos, incorporando os conceitos de classe e objeto, generalização-especialização, todo-parte, herança, polimorfismo, etc.. A figura 2 ilustra este modelo.

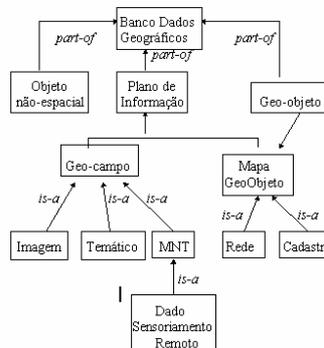


figura 2

Paralelamente, existe a visão de dados expressa em seu modelo Entidade-Relacionamento(E-R) dado na figura 3. Este modelo tem a função de descrever a estrutura do banco de dados utilizado pelo Spring.

3 Classe DataBase

Para garantir maior portabilidade, o software prevê uma classe "DataBase" (Banco de Dados) que encapsula a interface entre o SIG e o SGBDR, escondendo esta conexão do restante do sistema. Nesta classe estarão todos os métodos que devem acessar um determinado banco de dados, fazendo uso das funções primitivas, propostas abaixo.

Criar(): Cria uma tabela ou entidade;
 Inserir(): Inserir uma nova informação em uma tabela;
 Ler(): Carrega uma informação de uma tabela;
 Atualizar(): Atualiza informações de uma tabela;
 Excluir(): Exclui informação de uma tabela.

Assim, ficam agrupadas em uma classe todas as rotinas de interface, tornando o software SIG portátil para qualquer banco de dados.

Dentro da classe, cabe ao desenvolvedor fazer a transição entre os objetos e funções primitivas. Por exemplo, no ato de inserir um objeto, deve-se transformá-lo, afim de que a função primitiva insira seus dados em uma dada tabela no banco de dados relacional.

Propomos alguns passos que descrevem como o desenvolvedor irá construir tal interface.

- 1- Construção das funções primitivas e testes;
- 2- Construir os métodos da classe interface usando as funções primitivas;
- 3- Testes e manutenção.

No Spring existe a classe CODEBASE responsável pela interface Spring-DBASE, onde se encontram diversos métodos que acessam arquivos de dados. Por esta abordagem cada entidade do modelo E-R tornou-se um arquivo em DBASE, onde os métodos da classe CODEBASE tem a função de carregar, inserir, excluir, atualizar registros destes arquivos.

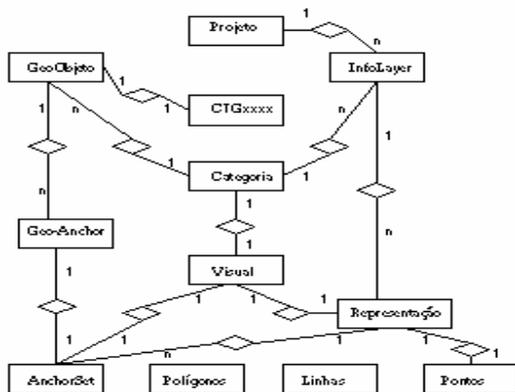


figura 3

4 Spring-Oracle

Pela abordagem Oracle, as entidades de modelo E-R foram mapeadas em forma de tabelas e relações, dentro do banco de dados.

Criou-se uma classe Oracle herdada da classe DATABASE, com a função de acessar estas tabelas.

O acesso as tabelas é realizado diretamente pela linguagem de consulta SQL.

A ligação entre o SQL e a classe Oracle, foi implementada através de um mecanismo, oferecido pela

Oracle, chamado de "Oracle Call Interface"-OCI. Desta forma, os comandos SQL são montados em um programa C, como cadeia de caracteres em uma variável. Esta variável é, então, passada por parâmetro ao OCI, que "traduz" e envia esta instrução SQL ao sistema Oracle.

Em uma primeira fase foram construídas funções primitivas de acesso às tabelas em linguagem C, utilizando OCI. Para cada tabela, derivada do modelo E-R, foi construída uma função de criar tabela, inserir, carregar, atualizar, excluir registro, além da função de criar banco de dados. Após esta fase, essas funções primitivas foram incorporadas à classe Oracle.

Como esta classe Oracle foi derivada da classe DATABASE, todos os seus métodos foram estudados a fim de capturar a sua lógica, para efetuar as mudanças de codificação necessárias, visando incorporar as novas funções primitivas de acesso às tabelas Oracle.

O trabalho mostrou que devido ao poder do mecanismo OCI e a simplicidade dos comandos SQL, na maioria dos métodos ocorreu uma relativa redução na quantidades de linhas codificadas..

A seguir mostramos (figura 4) as camadas de interface Spring-Oracle.

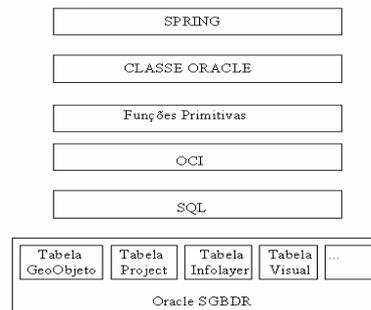


figura 4

5 Conclusão

O desenvolvimento encontra-se em andamento. Com início em setembro de 1995 e previsão de término até o final de dezembro de 1995.

Estamos convencidos de contribuir na questão da portabilidade do Spring em relação aos SGBDR aumentando a sua atuação no mundo do geoprocessamento.

6 Bibliografia

- Burrough, P.A. Principles of Geographical Information Systems for Land Resources Assessment, 1986, pp. 13.
- Câmara, G. Modelos, Linguagens e Arquitetura para Banco de Dados Geográficos, Tese Doutorado, 1995.
- Oracle RDBMS Database Administrator's Guide, v.6.0, 1990.