

2 Fundamentação Teórica

2.1. *Framework*

Um *framework* é um (sub-)sistema de *software* incompleto ou inacabado que é concebido para ser instanciado. Ele define a arquitetura para uma família de (sub-)sistemas de um domínio e fornece os blocos de construção básico para criá-los. Ele também define os lugares onde adaptações para funcionalidades específicas devem ser feitas. Em um ambiente orientado a objeto, um *framework* consiste de classes abstratas e concretas. A instanciação de um *framework* consiste de composição e derivação de classes existentes (Buschmann, 1996).

Por exemplo, um *framework* pode ser apropriado para a construção de interfaces gráficas com o usuário, editores de texto, sistemas operacionais, sistemas hipermídia, entre outros.

Em resumo, um *framework* é um conjunto de classes cooperativas que compõem um projeto re-usável para uma específica classe de *software* (Johnson & Foote, 1988; Deutsch, 1989). De acordo com (Pree, 1994), tal *framework* consiste de *frozen spots* e *hot spots*. De um lado, *frozen spots* definem a arquitetura geral de um sistema, quais são seus componentes básicos e os relacionamentos entre eles. Esses permanecem sem alteração (congelados) em qualquer das instanciações do *framework*. Por outro lado, *hot spots* representam as partes do *framework* que são específicas para um sistema particular. Os *hot spots* são projetados para serem genéricos. Ou seja, eles podem ser adaptados para as necessidades da aplicação que está sendo desenvolvida. Quando instanciamos uma aplicação a partir de um *framework*, os *hot spots* são especializados de acordo com as necessidades e requerimentos específicos desta aplicação. Isto significa que as classes definidas pelo usuário (por exemplo, novas sub-classes), recebem mensagens de classes pré-definidas do *framework*. Essas são geralmente tratadas pela implementação de métodos abstratos das super-classes.

2.2.

Padrão de projeto MVC

O padrão de projeto MVC foi desenvolvido utilizando o ambiente de programação do sistema Smalltalk-80TM para criação de interfaces gráficas de usuário (Krasner & Pope, 1998). O seu objetivo é separar os detalhes da aplicação, no componente Modelo, da forma como eles são apresentados para o usuário, contidos no componente Visão, da forma como o usuário o controla, isolados no componente Controle.

Antes da utilização do padrão de projeto MVC, o projeto de interface do usuário tendia a agrupar tais objetos - Modelo, Visão e Controle - ao mesmo tempo. Com o uso do MVC, estes objetos foram desacoplados, permitindo desta forma uma maior flexibilidade e maior possibilidade de re-uso, além de prover uma maneira eficiente de organizar sistemas que suportam múltiplas representações de uma mesma informação (Wheeler, 1996).

No padrão de projeto MVC, a entrada do usuário, a modelagem do mundo exterior e o retorno visual para o usuário são explicitamente separados e tratados por três tipos de componentes, cada qual especializado em sua função. A Visão gerencia a saída gráfica ou textual para a porção do *canvas* alocada para a aplicação. O Controle interpreta as entradas do usuário via interação com o *mouse* ou teclado, comandando o Modelo ou a Visão para alterá-los adequadamente. Finalmente, o Modelo gerencia o comportamento e os dados do domínio da aplicação, respondendo as solicitações por informações sobre seu estado (geralmente pela Visão), e respondendo as instruções para mudança de estado (geralmente pelo Controle) (Burbeck, 1992).

Atualmente existem várias implementações onde este padrão de projeto é encontrado além de Smalltalk-80TM. Podemos citar exemplos como: Java Server Faces (JSF)¹ e Jakarta Struts², para aplicações em ambiente Web, além de Cocoa³, para aplicações em ambiente MAC.

No contexto deste trabalho a utilização deste padrão de projeto serviu de base para a incorporação de mecanismos de adaptação, conforme será descrito no capítulo 3, de forma a torná-lo mais flexível.

¹ JSF - <http://java.sun.com/j2ee/jspserverfaces>

² Jakarta Struts - <http://struts.apache.org>

2.3.

Sistemas de informação cooperativos

Um dos principais objetivos dos Sistemas de Informação Cooperativos é auxiliar seus usuários na execução de uma determinada tarefa definida sobre um determinado domínio, onde estes sistemas adotam uma postura ativa em relação à tarefa a ser executada, procurando apontar possíveis soluções ou caminhos melhores a seguir, dentre outras possibilidades, comportando-se desta forma como verdadeiros parceiros de seus usuários (Michelis, 1997).

Os Sistemas de Informação Cooperativos baseiam-se na idéia de que, quando um usuário formula uma pergunta, ele não está interessado somente em entender a resposta para aumentar seu conhecimento, mas ele tem a intenção de realizar alguma tarefa e que a resposta contenha informação necessária ou útil para tal (Cuppens, 1989).

Cuppens & Demolombe (1991) apresenta um método e um formalismo para definir transformações de consultas que forneçam ao usuário mais informação do que é explicitamente requisitado pela consulta original. O método é baseado na idéia de que informações interessantes são fortemente dependentes do contexto e que pessoas que tenham muita experiência em prover informação podem explicitar este conhecimento no formalismo proposto. Este formalismo tem duas vantagens importantes: oferecer uma semântica muito precisa e poder ser facilmente implementada usando mecanismos de derivação já existentes como o Prolog (Clocksin, 1981). Além da definição dos conceitos que promovem a transformação das consultas, essa abordagem também adota um modelo de representação do usuário, propiciando um processo de consulta mais personalizado e mais otimizado.

Chu et al. (1996) apresenta o CoBase, um sistema de informação cooperativo escalável e extensível. Implementado sobre uma arquitetura baseada em mediadores que propõe uma abordagem sobre o esquema do banco de dados e características da aplicação, onde os dados estão organizados em *Type Abstraction Hierarchies (TAHs)*. Os níveis mais altos da hierarquia fornecem uma representação dos dados mais abstrata do que os níveis mais baixos. Este conceito

³ Cocoa - <http://homepage.mac.com/redbird/cocoa>

permite que o problema seja representado em vários níveis de abstração. Cada um desses níveis está relacionado com outros por meio de ligações semânticas, que associam elementos pertencentes a diferentes níveis. O processo de consulta é obtido pela reescrita das consultas através da navegação dessa hierarquia de níveis, generalizando e especializando seus elementos por meio das associações semânticas. Além disso, foi desenvolvido um sistema de informação geográfico cooperativo para prover análise e recuperação de informações baseada sobre objetos geográficos.

No contexto deste trabalho, a noção de cooperatividade foi utilizada no sentido mais restrito de adaptação da aplicação a um modelo de usuário previamente definido. Esta adaptação levou em consideração o grau de relacionamento semântico entre as classes de informações (CI's) presentes no domínio utilizado, sendo este grau definido pelo modelo de usuário. Ela se manifesta de forma a contextualizar melhor a consulta original do usuário. Através da identificação de um grau de relacionamento semântico forte entre as classes de informação, o sistema efetua a consulta original e, através de uma consulta adicional, obtém também informações da classe de informação relacionada. Esta noção de adaptação ficará mais clara na seção 3.2.1.

Por se tratar de um domínio com predominância de classes de informações com características geográficas, as relações topológicas (vide seção 1.1) entre as CI's têm papel fundamental nos resultados da consulta, pois dependendo de qual delas for utilizada na consulta submetida, a aplicação pode retornar resultados completamente distintos.

2.4.

Personalização de usuário

Personalização constitui o mecanismo e a tecnologia necessária para adaptar o acesso da informação ao usuário final. Pode ser definida como o ajuste automático do conteúdo da informação, da sua estrutura e da sua apresentação ao usuário (Perugini & Ramakrishnan, 2003).

Gauche et al. (2002) relata o projeto OBIWAN (*Ontology Based Informing Web Agent Navigation*), que faz uso de ontologias como elemento chave para prover acesso à informação personalizada. O objetivo principal do sistema é criar,

automaticamente, perfis de usuários baseados em ontologia. O sistema utiliza estes perfis para personalizar os resultados de uma máquina de busca da *Internet*, utilizando-os também na criação de hierarquias de navegação personalizadas para remoção de *sites* da Web. Este perfil é essencialmente uma ontologia de referência na qual cada conceito tem um peso, indicando o interesse percebido do usuário nesse conceito. Os perfis são gerados analisando o comportamento de navegação do usuário, especificamente o conteúdo, detalhamento, e o tempo gasto em cada página visitada. Estas páginas são classificadas automaticamente nos conceitos contidos na ontologia de referência e os resultados da classificação são acumulados. Isto faz com que os conceitos na ontologia de referência recebam pesos baseados na quantidade de informação relacionada com a navegação do usuário. Nenhum retorno explícito do usuário é necessário.

Em (Abel et al., 2005), é descrito o projeto *Personal Reader* que faz parte da rede da pesquisa do *Excellence REWERSE*. O propósito do projeto é fornecer um *framework* para o desenvolvimento e manutenção do *Web Content-Reader*. Esta aplicação demonstra como fornecer visões personalizadas, sobre dados distribuídos na *Web*, usando tecnologias da *Web* semântica. A aplicação compreende quatro etapas: 1) a etapa de recolhimento da informação, na qual informações de fontes distribuídas e heterogêneas são extraídas e enriquecidas com semântica legível por máquina; 2) a etapa de operação para extrações convenientes e atualizadas; 3) a etapa de raciocínio na qual regras ponderam sobre as descrições semânticas criadas e as bases de conhecimento adicionais como as ontologias e informação do perfil do usuário; e, por último, 4) a etapa de criação da interface do usuário na qual as descrições RDF que resultam da etapa de raciocínio são interpretadas e traduzidas para uma interface de usuário apropriada e personalizada.

No contexto deste trabalho, por se tratar de um universo menor e menos heterogêneo do que o de usuários da *Internet*, a geração automatizada de perfis de usuários não se justifica. No nosso universo os papéis, tarefas e informações relevantes, tanto para seleção, quanto para apresentação, de cada grupo são bem definidos. O procedimento adotado para a sua geração foi o preenchimento prévio das informações relativas a cada grupo de usuário, associando a cada um deles a relação de classes de informações nas quais eles teriam interesse. Como também a definição do grau do meta-relacionamento semântico entre estas classes, para

auxiliar na adaptação da aplicação, e os tipos de consultas de filtro de cada classe de informação. As demais informações, como os *templates* espaciais, atributos das consultas, etc, vieram como uma decorrência das associações com as classes de informações do domínio utilizado.

2.5.

Bancos de dados geográficos

Os Sistemas Gerenciadores de Bancos de Dados Geográficos (SGBDG) tem como objetivos principais o armazenamento, acesso e análise rápida e eficiente de dados espaciais.

O termo dado espacial ou dado geográfico é utilizado na representação de fenômenos do mundo real através de duas componentes:

1. Sua localização geográfica, isto é, sua posição em um sistema de coordenadas conhecido, onde uma coordenada é um número que representa uma posição relativa a um ponto de referência;
2. Seus atributos descritivos, como por exemplo, nome, dimensão, situação, etc.

O SGBDG necessita de estruturas para a representação dos dados espaciais, bem como de operações para sua manipulação. Devido à natureza complexa e multidimensional dos dados espaciais, acrescida do fato dos Sistemas Gerenciadores de Bancos de Dados (SGBD) convencionais não contemplarem os requisitos espaciais, novos tipos de dados e operadores, basicamente topológicos ou métricos, foram construídos (Silva, 2002).

Um dado espacial pode ser representado por dois modelos de dados distintos: vetorial ou matricial (*raster*) (Burrough & McDonnell, 1998), sendo o último não abordado neste trabalho.

Utiliza-se o modelo de dados vetorial para a representação do espaço como um conjunto de entidades discretas (objetos geográficos) definidas por uma unidade (ponto, linha ou polígono) geograficamente referenciada e por seus atributos descritivos, vide Figura 1. Por exemplo, as bacias sedimentares podem ser representadas por objetos geográficos, onde cada bacia é representada por um polígono que define o seu limite, cada rio por uma linha e cada cidade por um ponto. Além da localização, cada objeto geográfico tem seus atributos descritivos,

como o nome de cada bacia, a quilometragem linear dos rios, a população de cada cidade, etc.

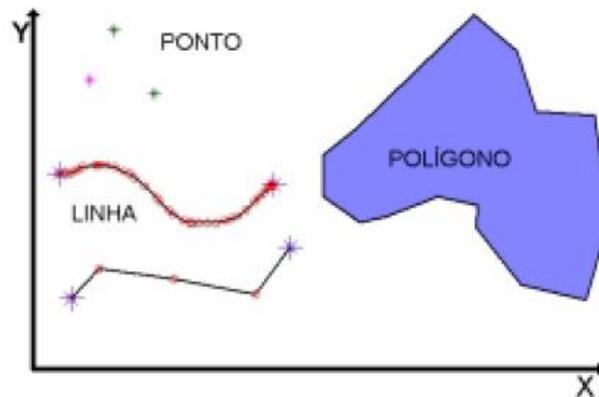


Figura 1: Elementos da representação vetorial

Fonte: Câmara et al., 2003.

Em (Câmara, 2005), são examinados os problemas básicos de representação computacional de dados geométricos, onde questões como a representação dos dados geográficos no computador, de que forma as estruturas de dados geométricas e alfanuméricas se relacionam com o mundo real e quais alternativas de representação computacional existem para os dados geográficos, são esclarecidas. Para abordar tais problemas, foi utilizado o paradigma dos quatro universos proposto inicialmente por (Gomes & Velho, 1995) e adaptado para a geoinformação por (Câmara, 1995). Este paradigma distingue quatro passos entre o mundo real e sua realização computacional, que são, os universos ontológico, formal, estrutural e o de implementação. O uso do paradigma serve para demonstrar que a transposição da realidade para o computador requer uma série complexa de mediações. Inicialmente nomear as entidades da realidade. Depois, gerar modelos formais que as descrevam de forma precisa. A seguir, escolher as estruturas de dados e algoritmos que melhor se adaptam a estes modelos formais. Finalmente, fazer a implementação num suporte computacional adequado.

Em (Ferreira et al., 2005), é feita uma comparação entre os diversos tipos de SGBD's, enfatizando seus requisitos mais importantes e as principais tecnologias para atendê-los, além de abordar a evolução das arquiteturas de SIG. No contexto deste trabalho, dar-se-á ênfase aos SGBDs Objeto-Relacionais (SGBD-OR), que estendem o modelo relacional, entre outras características, com um sistema de

tipos de dados rico e estendível, oferecendo operadores que podem ser utilizados na linguagem de consulta. Estes, possibilitam ainda, a extensão dos mecanismos de indexação sobre os novos tipos, os TDEs (Tipos de Dados Espaciais). Essas características reduzem os problemas ocorridos na simulação de tipos de dados pelos SGBD-R, tornando os SGBD-OR uma solução atrativa para aplicações não convencionais. Segundo os autores, existem basicamente duas principais formas de integração entre os SIGs e os SGBDs, que são a “arquitetura dual” e a “arquitetura integrada”, sendo esta última adotada no contexto deste trabalho, vide Figura 2. Ela consiste em armazenar todos os dados em um SGBD, ou seja, tanto a componente espacial quanto a alfanumérica. Sua principal vantagem é a utilização dos recursos de um SGBD para controle e manipulação de objetos espaciais. Como gerência de transações, controle de integridade, concorrência e linguagens próprias de consulta. Sendo assim, a manutenção de integridade entre a componente espacial e alfanumérica é feita pelo SGBD.



Figura 2: Arquitetura integrada
Fonte: Ferreira et al., 2005.

Em (Queiroz & Ferreira, 2005), são apresentadas duas extensões espaciais de SGBD convencionais, uma da comunidade de software aberto, PostGIS (Ramsey, 2002) e outra comercial, Oracle Spatial® (Ravada & Sharma, 1999; Murray, 2003). No contexto deste trabalho, utilizou-se a extensão Oracle

Spatial®, que estende o modelo objeto-relacional do SGBD Oracle®⁴. Esta extensão baseia-se nas especificações do OpenGIS (OGIS, 1999). Este modelo permite definir novos tipos de dados através da linguagem de definição de dados SQL DDL, vide Figura 3, e implementar operações sobre esses novos tipos, através da linguagem PL/SQL (Urman, 2002) que é uma extensão da SQL (Lassen et al., 1998). Esta extensão é baseada nas especificações do OpenGIS e contém um conjunto de funcionalidades e procedimentos que permitem armazenar, acessar, modificar e consultar dados espaciais de representação vetorial. O Oracle Spatial® é formado pelos seguintes componentes:

- Um modelo próprio de dados chamado MDSYS que define a forma de armazenamento, a sintaxe e semântica dos tipos espaciais suportados.
- Mecanismo de indexação espacial.
- Um conjunto de operadores e funções para representar consultas, junção espacial e outras operações de análise espacial.
- Aplicativos administrativos.

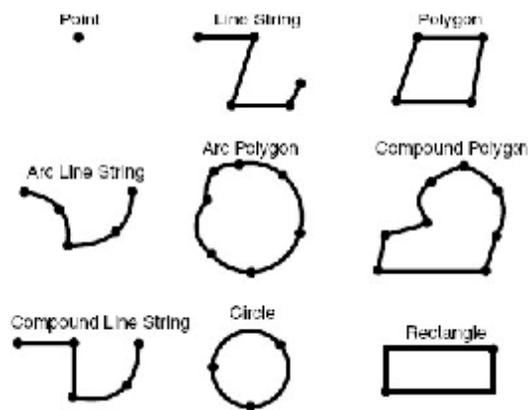


Figura 3: Tipos espaciais primitivos do Oracle Spatial®
Fonte: Queiroz & Ferreira, 2005

⁴ Oracle - <http://www.oracle.com/database/index.html>

2.6. Multi-resolução

As bases de dados que contêm grandes quantidades de informações geográficas tem na exploração interativa destas informações um problema de alto esforço computacional devido ao seu processamento.

Normalmente, uma base de dados geográficos contém milhares de linhas poligonais com milhões de vértices. Esses dados geralmente ficam armazenados em alta resolução, obtidos por digitalização automática ou manual de mapas ou fotografias.

Entretanto, somente uma pequena fração destes dados pode ser exibida na tela. Normalmente a resolução original dos dados não é necessária para gerar os desenhos, porque dependendo da escala atual do mapa, muitos vértices podem ser exibidos em um mesmo *pixel* da tela. Além disso, gerar um desenho usando sua resolução original pode ser proibitivo ou sem sentido numa aplicação interativa.

Uma técnica utilizada para diminuir a quantidade de vértices a serem exibidos é a representação das linhas poligonais em multi-resolução. Trata-se da representação hierárquica da linha que pode ser percorrida somente até o nível onde as características básicas da linha sejam mantidas para uma escala desejada (Günther & Wong, 1990).

Esta representação permite calcular uma aproximação da poligonal que é adequada à resolução desejada, ao mesmo tempo em que mantém intactos os dados originais. Desta maneira, é possível fazer aproximações sucessivas.

Em (Derraik, 1997), foi realizado um estudo comparativo de algumas representações em multi-resolução para linhas poligonais. Foram estudadas as estruturas *strip tree*, *arc tree* e *box tree*, e suas variantes, comparando as velocidades de construção, de percorrimento (*drawing*), de operações de interseção e seleção (*pick*); e custo de armazenagem em memória. A comparação foi feita utilizando bancos de dados cartográficos reais disponíveis na Internet. Seu objetivo principal foi a identificação de técnicas e algoritmos adequados para exploração interativa de grandes bancos de dados cartográficos.

Em (Davis, 1998), foi realizada uma análise comparativa dos algoritmos de simplificação de poligonais, quanto às possibilidades de utilização em generalização dinâmica de objetos geográficos, a partir da seleção de algoritmos

extraídos da literatura. Estes algoritmos se propõem a filtrar o excesso de detalhes da poligonal, produzindo representações mais compactas sem degradar o seu aspecto visual. Sendo que os resultados obtidos por cada algoritmo podem variar bastante quanto à qualidade na sua representação final.

No contexto deste trabalho, a solução adotada em termos de estrutura de armazenamento foi uma estrutura análoga a uma árvore binária, porém implementada em um vetor, onde cada elemento é composto por: um vértice da poligonal, a distância deste vértice à poligonal formada pelos vértices anteriores e o índice que indica a posição onde ele deverá ser inserido na poligonal formada até então. Quanto ao algoritmo, este foi baseado no de (Douglas & Peucker, 1973), porém não há descarte de vértices. Ele é usado somente para identificar os pontos de maior representatividade da poligonal original para auxiliar na criação da estrutura de multi-resolução.

Esta técnica será empregada na exibição das instâncias das CI's que possuem um número muito grande de coordenadas para a sua representação, minimizando o volume de informações a serem tratadas pela Visão. Na seção 4.3.1 o uso desta técnica pelo componente Modelo será detalhado.