

1 Introdução

1.1. Motivação

As bases de dados constituem um dos principais alicerces para o desenvolvimento das atividades técnicas relacionadas com a área de atuação de empresas de grande porte. Exemplos típicos destas empresas são as da área de petróleo, de telefonia, como também de transportes. A correta gestão das bases permite ampliar os níveis de disponibilidade e confiabilidade dos dados.

A adoção de sistemas de informação como ferramenta de apoio aos usuários na tomada de decisão é fundamental para uma maior agilidade e eficácia na localização, seleção e recuperação das informações das bases de dados, contribuindo diretamente para a otimização dos prazos de execução dos projetos empresariais.

Em particular, quando o dado possuir atributo espacial geo-referenciado, existe a possibilidade de se efetuar uma série de operações sobre tal atributo. Podem ser executadas consultas espaciais a partir das relações topológicas, como por exemplo: interseção, toca, contém, etc ..., entre o dado e outros armazenados na base que possuam este mesmo tipo de atributo. Com este tipo de representação, consegue-se reduzir consideravelmente a necessidade de se modelar relações estáticas entre as classes de informação (CI) do domínio, possibilitando-se aumentar a quantidade e a qualidade das informações recuperadas. De fato, dado que as classes de informação possuem atributos espaciais geo-referenciados, algumas relações entre elas podem ser inferidas automaticamente, em lugar de serem explicitamente armazenadas.

Segundo (Câmara, 2005), o termo sistema de informação geográfica (SIG) é aplicado para sistemas que realizam o tratamento computacional de dados geográficos. A principal diferença de um SIG para um sistema de informação convencional é sua capacidade de armazenar tanto os atributos descritivos quanto os atributos espaciais geo-referenciados.

Infelizmente, na etapa de modelagem conceitual de qualquer domínio, não se consegue transpor para as bases de dados, que irão subsidiar uma aplicação, todo o conhecimento dos especialistas do domínio. Isto acaba se tornando um grande problema já que nos códigos dos programas de um sistema de informação podem estar contidas grande parte do conhecimento do domínio que, por vários motivos, não pode ser capturado no projeto da base de dados, separadamente do próprio código. Uma parcela deste conhecimento fica codificada na parte estática dos programas ou na forma de declarações de classes e tipos de dados, nos casos de sistemas orientados a objetos. Outra parcela, a de regras de negócio, é mantida implicitamente no código dos procedimentos das aplicações (Guarino, 1998).

Além disso, operações naturais sobre o modelo conceitual de banco de dados, como a navegação entre classes de informação, não são tão imediatas no modelo lógico e também podem ser encontradas codificadas nos fontes dos programas (Motik, 2002).

Uma maneira de enfrentar esses problemas é reduzir ou eliminar os conflitos conceituais e terminológicos e tornar o conhecimento explícito. Tal conhecimento pode funcionar como um referencial unificado para os diferentes pontos de vista, conforme (Uschold & Gruninger, 1996), servindo como base para:

- Comunicação entre pessoas com diferentes necessidades e pontos de vista que surgem de seus contextos divergentes;
- Interoperabilidade entre sistemas obtidos pela tradução entre diferentes métodos de modelagem, paradigmas, linguagens e ferramentas de *software*;
- Planejamento dos sistemas, particularmente:
 - Re-usabilidade: o conhecimento compartilhado é a base para a formalização da codificação de importantes entidades, atributos e processos e seus inter-relacionamentos no domínio de interesse. Essa representação formal pode ser um componente re-usável e/ou compartilhado em um sistema de *software*.
 - Confiabilidade: uma representação formal também torna possível a automatização da verificação de consistência, resultando em *softwares* mais confiáveis.
 - Especificação: o conhecimento compartilhado pode ajudar no processo de identificação de requisitos e definição de

especificações para um sistema de Tecnologia da Informação. Isto é especialmente verdade quando os requisitos envolvem diferentes grupos usando diferentes terminologias em um mesmo domínio, ou em múltiplos domínios.

Relatados os elementos que nos motivaram para realização deste trabalho, podemos agora definir de que forma pretendemos solucioná-lo.

1.2. Abordagem proposta

O objetivo principal deste trabalho é propor uma arquitetura variante do padrão de projeto Modelo-Visão-Control (MVC) para desenvolver sistemas de informação geográfica adaptativos baseados em modelos.

Modelo é o termo usado para referir-se ao conhecimento explicitado e compartilhável de alguns domínios de interesse, os quais podem ser usados como um referencial unificado na resolução de problemas advindos das áreas de comunicação, interoperabilidade, representação do conhecimento e re-uso (Ushold & Gruninger, 1996).

O uso de modelos, transformados em componentes ativos dos sistemas de informação, nos leva a um MDIS (*Model-Driven Information System*) (Guarino, 1998) e, no caso específico de GIS (*Geographic Information System*), ele nos leva ao que é chamado de MDGIS (*Model-Driven Geographic Information System*) (Fonseca & Egenhofer, 1999).

Em linhas gerais, um sistema baseado na abordagem proposta funcionaria da seguinte maneira: o usuário navega normalmente na aplicação, clicando sobre alguma instância de uma classe de informação solicitando uma consulta. Porém, ao executar a consulta, o sistema identifica a existência de alguma informação associada àquela solicitada e que seja relevante para o usuário. Caso não haja, o sistema funciona da forma tradicional, obtendo somente as informações solicitadas e disponibilizando-as para o usuário. Caso haja, o sistema disponibiliza as informações solicitadas juntamente com todas as informações associadas a ela de uma só vez.

Independente destes dois modos de atuação do sistema, este sempre deverá gerar e apresentar interfaces de seleção e apresentação personalizadas dinamicamente, baseadas nos interesses do usuário, seus requisitos e papéis.

Para projetar sistemas desta natureza, se faz necessário, antes de tudo, a geração de modelos que possuam informações suficientes para auxiliar o sistema, dentre outras atividades, a se tornar adaptativo para o usuário.

Há diversas características peculiares aos sistemas baseados no modelo de desenvolvimento cliente-servidor. Podemos citar o fato de componentes do servidor proverem serviços para múltiplos clientes (*server-side-response*) e de componentes do cliente requisitarem serviços dos componentes do servidor (*client-side-request*). Estas características normalmente acarretam uma grande quantidade de transferências de dados entre o cliente e o servidor. Outra seria possuir um modelo de negócio independente da GUI (*Graphical User Interface*), podendo gerar desta forma uma grande variedade de GUI's para o mesmo modelo de negócio.

As características relatadas acima nos levam a optar pelo uso do padrão de projetos Modelo-Visão-Control (MVC), com adaptações pertinentes ao contexto do uso de modelos. Detalhes do padrão de projeto MVC, além de outros conceitos básicos para o desenvolvimento deste trabalho, serão dados no próximo capítulo.

1.3. Organização da dissertação

O capítulo 2 resume alguns tópicos de fundamentação teórica relativos ao padrão de projeto MVC, ao uso de mecanismos de adaptação e personalização nos sistemas de informações, a bancos de dados geográficos e a multi-resolução.

O capítulo 3 apresenta uma visão geral da arquitetura proposta neste trabalho, descrevendo cada um dos módulos, além da exemplificação de tarefas seguindo tal arquitetura.

O capítulo 4 apresenta uma especificação de implementação da arquitetura proposta no capítulo 3, relacionando as características gerais e requisitos necessários para cada um dos módulos, bem como a descrição de todas as tabelas do modelo.

O capítulo 5 apresenta a descrição de uma aplicação real, segundo a arquitetura desenvolvida para a solução proposta, bem como dois exemplos, um sobre adaptação e outro sobre personalização da aplicação.

Finalmente, o capítulo 6 contém as contribuições do trabalho e sugestões de trabalhos futuros.