

Desenvolvimento de uma Ferramenta CASE para o Modelo UML-GeoFrame com Suporte para Padrões de Análise

JUGURTA LISBOA FILHO

MARCONI DE ARRUDA PEREIRA

UFV - Departamento de Informática, Campus da UFV, 36571-000 Viçosa, MG, Brasil

{jugurta, mdap}@dpi.ufv.br

Abstract. This paper describes the development of CASEGEO, a Computer Aided Software Engineering Tool (CASE tool) for supporting geographic database design, which is based on the UML-GeoFrame model. Rules are presented for mapping from UML-GeoFrame schemes into ArcView's logical representation. Finally, the paper shows as CASEGEO could be used for specification and reuse of analysis patterns.

1 Introdução

Bancos de Dados Geográficos (BDG) são coleções estruturadas de dados geoespaciais e descritivos manipulados por Sistemas de Informação Geográfica (SIG). Um BDG pode ser projetado com base em metodologias tradicionais de projeto de banco de dados [19], ou seja, as fases de projeto conceitual, lógico e físico devem ser contempladas [7].

Na fase de projeto conceitual é necessário definir o modelo de dados a ser utilizado, o que vai facilitar a comunicação entre projetistas e/ou usuários envolvidos. Diversos modelos específicos para modelagem de BDG têm sido propostos e aperfeiçoados nos últimos anos, entre os mais citados estão os modelos GeoOOA [14], MADS [23], OMT-G [4], UML+SpatialPVL [1], OMEGA [24] e UML-GeoFrame [19]. Uma análise comparativa destes modelos pode ser encontrada em [18].

Concluída a modelagem conceitual, o próximo passo - projeto lógico - consiste na transformação do esquema conceitual produzido na etapa anterior em um esquema de dados compatível com os modelos de dados do software de SIG e do Sistema de Gerenciamento de Banco de Dados (SGBD) que serão utilizados. Esta etapa de transformação de esquemas conceituais em esquemas lógicos e sua implantação em um software de SIG comercial (e de SGBD) pode ser feita de forma automática por uma ferramenta CASE (*Computer Aided Software Engineering*). Entre as principais ferramentas CASE, que são específicas para projeto de BDG, pode-se citar:

Perceptory [5] – Ferramenta desenvolvida a partir do sistema Microsoft Visio para suporte à modelagem de dados com base no modelo UML+Spatial PVL (*Plug-in for Visual Languages*) [1]. Esta ferramenta consiste em uma paleta de construtores de desenhos e do dicionário de dados implementado no sistema MS-Access. O sistema gera um esqueleto de código para o SGBD Oracle 8i.

REGIS [13] – Ferramenta CASE desenvolvida em Smalltalk que tem como base o modelo GeoOOA [KOS 97], adaptado para a representação gráfica da linguagem UML. A ferramenta gera automaticamente esquemas para o SIG Smallworld.

AIGLE [15] – Ferramenta CASE para aplicações de SIG desenvolvidas na WEB. Utiliza o modelo orientado a objetos chamado OMEGA [24] e consegue gerar aplicações finais em linguagens como MapBasic e Java.

MADS (editor Java) [23] – Editor de esquemas de dados implementado em Java, tendo como base o modelo MADS. Esta ferramenta é apenas um editor gráfico, não suportando geração automática de esquemas.

Este artigo descreve o desenvolvimento da ferramenta CASEGEO, uma ferramenta CASE que suporta a modelagem conceitual de BDG com base no modelo UML-GeoFrame [19]. A ferramenta provê um módulo de geração automática para os principais SIG comerciais. Atualmente um módulo de geração de esquemas para o ARCVIEW (ESRI) está operacional. Módulos de geração para outros SIG estão sendo desenvolvidos. Uma das características da ferramenta CASEGEO, que a diferencia das outras ferramentas existentes, é o suporte à reutilização de esquemas através do uso de padrões de análise. Um padrão de análise é uma combinação recorrente de elementos de modelagem que ocorrem em algum contexto [8].

O restante deste artigo está organizado da seguinte forma. A seção 2 apresenta um resumo do modelo UML-GeoFrame. A seção 3 descreve as vantagens do uso de padrões de análise na modelagem conceitual de BDG. A seção 4 detalha o desenvolvimento da ferramenta CASEGEO, mostrando sua arquitetura e descrevendo cada módulo implementado. Considerações finais e trabalhos futuros são descritos na seção 5.

2 O Modelo UML-GeoFrame

A modelagem conceitual de BDG com base na linguagem UML [3] e no *framework* GeoFrame [17] produz um esquema de banco de dados de fácil entendimento, melhorando a comunicação entre projetistas e/ou usuários. Além de ser usado na elaboração de esquemas de banco de dados, o modelo UML-GeoFrame é adequado para especificação de padrões de análise [21].

O GeoFrame é um *framework* conceitual que fornece um diagrama de classes básicas para auxiliar o projetista nos primeiros passos da modelagem conceitual de dados de uma nova aplicação de SIG. O uso conjunto do diagrama de classes da linguagem UML e o GeoFrame permite a solução da maioria dos requisitos de modelagem de aplicações de SIG. Um esquema conceitual de dados geográficos construído com base no modelo UML-GeoFrame inclui, por exemplo, a modelagem dos aspectos espaciais da informação geográfica e a diferenciação entre objetos convencionais e objetos/campos geográficos. A especificação desses elementos é feita com base no conjunto de estereótipos mostrados na Figura 1.

Fenômeno geográfico e Objeto convencional	Componente espacial de objetos geográficos	Componente espacial de campos geográficos
<ul style="list-style-type: none"> △ Objeto geográfico △ Campo geográfico △ Objeto não geográfico 	<ul style="list-style-type: none"> □ Ponto ▭ Linha ▭ Polígono ★ Obj. espacial complexo 	<ul style="list-style-type: none"> ● Pontos irregulares ⋯ Grade de pontos ▭ Polígonos adjacentes ▭ Isolinhas ▭ Grade de células ▭ TIN
<<função>> função categórica		

Figura 1 – Estereótipos do Modelo UML-GeoFrame

O primeiro conjunto de estereótipos (*Fenômeno geográfico e Objeto convencional*) é usado para diferenciar os dois principais tipos de objetos pertencentes a um BDG. *Fenômeno geográfico* é especializado em *Objeto geográfico* (△) e *Campo geográfico* (△), segundo as duas formas de percepção dos fenômenos geográficos, descritas por Goodchild [11]. *Objetos não geográficos* são modelados de forma tradicional e são identificados através do estereótipo (△).

Os conjuntos de estereótipos *Componente espacial de objetos geográficos* e *Componente espacial de campos geográficos* são usados para a modelagem do componente espacial de fenômenos segundo as visões de objeto e de campo, respectivamente. A existência de múltiplas representações é modelada através da combinação de dois ou mais estereótipos em uma mesma classe. Por exemplo, uma classe *Município* pode ter duas formas de abstração de seu componente espacial, pontual e poligonal, o que é especificado pelo par de estereótipos (□▭).

Por último, o estereótipo <<função>> é usado para caracterizar um tipo especial de associação que ocorre

quando da modelagem de campos categóricos. Segundo Chrisman [6], numa estrutura de cobertura categórica o espaço é classificado em categorias mutuamente exclusivas, ou seja, uma variável possui um valor do tipo categoria em todos os pontos dentro de uma região (ex.: tipos de solos). A Figura 2 ilustra um diagrama de classes no modelo UML-GeoFrame contendo dois temas: Educação e Meio-Ambiente.

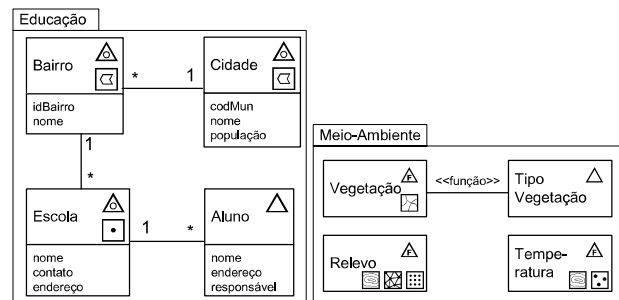


Figura 2 – Exemplo de esquema UML-GeoFrame

3 Padrões de Análise

Um padrão de análise é um mecanismo de reutilização que permite aos projetistas menos experientes reutilizarem o conhecimento de outros especialistas.

Um padrão apresenta a essência de uma solução para um problema recorrente, em um contexto específico [10]. Esta definição compreende as idéias fundamentais de um padrão. A expressão “uma solução para um problema” significa que cada padrão identifica um problema e apresenta uma solução para ele. O termo “essência de uma solução” significa que somente os elementos essenciais são descritos, deixando os aspectos específicos para serem detalhados pelo projetista, dado que aspectos específicos normalmente não são reutilizados. O termo “problema recorrente” significa que os padrões devem ser descritos para problemas que já ocorreram diversas vezes e irão ocorrer novamente. Por último, “em um contexto específico” significa que a solução completa é válida para um contexto particular.

Um padrão de análise descreve um conjunto de classes, possivelmente pertencentes a diferentes hierarquias de classes, e as associações existentes entre elas [9]. Padrões de análise podem ser vistos, portanto, como uma forma de descrever subesquemas de projetos mais complexos, os quais ocorrem com frequência durante o processo de modelagem de muitas aplicações. O uso de padrões melhora, de forma significativa, o tempo de desenvolvimento de novas aplicações, uma vez que a reutilização ocorre através de subesquemas e não através de classes isoladas [12].

Geralmente, um padrão de análise apresenta a solução do problema de uma forma mais sugestiva do que prescritiva, fornecendo um modelo e a discussão do por que a solução é proposta desta forma, suas vantagens e desvantagens. Segundo Fowler [9], a contribuição realmente importante de um padrão não é o modelo fornecido como solução, mas sim, o raciocínio que está por trás desta solução.

A maioria dos padrões de análise publicados até o momento foi projetada, principalmente, para solucionar problemas de aplicações comerciais como, por exemplo, os padrões descritos em [9] e [12]. No entanto, a idéia de padrões de análise pode ser usada para aumentar a qualidade e a produtividade no desenvolvimento de aplicações não-convencionais como as aplicações de SIG. A adequabilidade do uso de padrões de análise na modelagem de BDG é mostrada em [16]. Exemplos de padrões de análise para aplicações de SIG podem ser encontrados em [2] e [21].

4 A Ferramenta CASEGEO

CASEGEO é uma ferramenta CASE que está em desenvolvimento no Departamento de Informática da Universidade Federal de Viçosa, cujo objetivo é dar suporte à modelagem de BDG com base no modelo UML-GeoFrame e na (re)utilização de padrões de análise.

Uma ferramenta CASE é, antes de mais nada, um software de desenho gráfico. Para evitar um grande esforço de programação no sentido de desenvolver uma ferramenta gráfica desde a estaca zero, optou-se por utilizar como ponto de partida algum software gráfico existente. Este software deveria suportar o desenho de diagramas de classes da linguagem UML [3] e ser extensível para suportar os estereótipos definidos no GeoFrame [17].

Os programas analisados foram o Rational ROSE e o Visio Professional [25] (estes dois comerciais), além de alguns softwares livres disponíveis na Internet (ex.: Dia - Gnome). O pacote Visio possui um ambiente de programação acoplado à linguagem Visual Basic for Application (VBA), o que possibilita a implementação de rotinas que podem ser associadas aos elementos gráficos do esquema. Desta forma, o Visio foi escolhido para ser utilizado no desenvolvimento da ferramenta CASEGEO. Esta solução também foi empregada no desenvolvimento das ferramentas Perceptory e AIGLE.

A Figura 3 ilustra a arquitetura da ferramenta CASEGEO, a qual é composta de quatro módulos. O Módulo Gráfico permite ao projetista desenhar o diagrama de classes, fornecendo uma paleta gráfica (*stencil GeoFrame*) que contém os construtores do modelo UML-

GeoFrame. O projetista pode recuperar diagramas de classes pertencentes a padrões de análise disponíveis em um repositório. O Módulo Dicionário de Dados armazena a descrição detalhada dos elementos do diagrama criado pelo projetista. O Módulo Geração Automática permite a transformação de esquemas conceituais armazenados no dicionário de dados em esquemas lógicos correspondentes a alguns modelos utilizados em SIG comerciais. O Módulo de Engenharia Reversa, ainda não implementado, possibilitará ao projetista obter esquemas conceituais a partir de aplicações de SIG existentes.

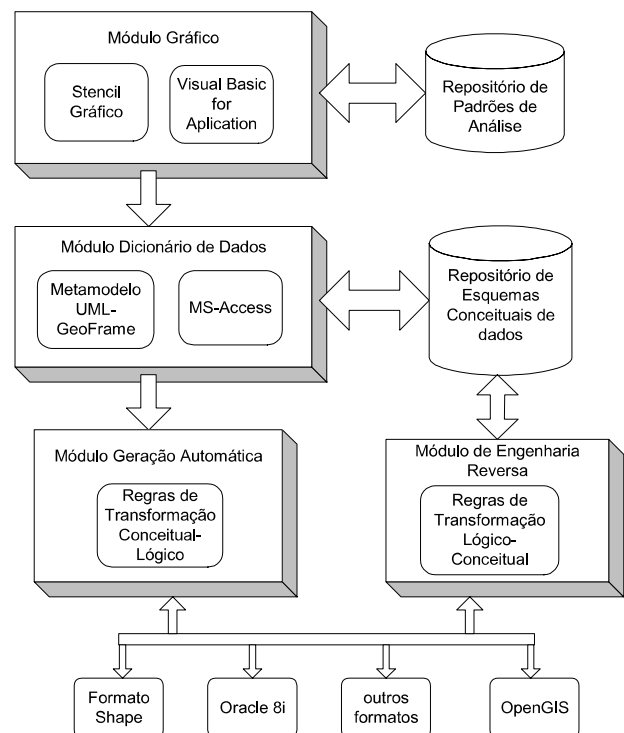


Figura 3 – Arquitetura da Ferramenta CASEGEO

As seções seguintes descrevem detalhadamente cada um destes módulos.

4.1 Módulo Gráfico

A ferramenta CASEGEO disponibiliza o *stencil GEOFRAME*, ou seja, uma paleta gráfica que possui os construtores do modelo UML-GeoFrame. A partir dessa paleta o usuário pode criar o seu esquema conceitual de dados. O Visio permite a criação de novas paletas de desenhos, além dos desenhos (ou construtores) já existentes. Associados a estes desenhos existem várias propriedades que podem ampliar as aplicações às quais estes se destinam. Por exemplo, pode estar associada a um objeto do desenho as propriedades: abrir um formulário;

chamar um procedimento; ou executar um programa em alguma outra linguagem, como C++.

Cada classe do esquema conceitual de dados possui um nome, atributos (com os respectivos domínios), operações e estereótipos GeoFrame. As classes podem estar relacionadas entre si através de uma generalização, agregação, composição ou associação. Num relacionamento do tipo associação podem ser especificados a multiplicidade e o papel que cada classe desempenha no relacionamento. As classes podem estar agrupadas de forma a constituírem um determinado tema, o que é modelado pelo construtor Pacote, da linguagem UML.

A Figura 4 ilustra o Pacote CADASTRO IMOBILIÁRIO, contendo as classes TRECHO_DE_LOGRADOURO, QUADRA, TESTADA_DE_LOTE, LOTE, que são subclasses da classe OBJETO_GEOGRÁFICO (△) e a classe PROPRIETÁRIO, subclasse de OBJETO_NÃO_GEOGRÁFICO (△), isto é, não possui representação espacial. As classes TRECHO_DE_LOGRADOURO e TESTADA_DE_LOTE possuem representação espacial do tipo linha (▣) e as classes QUADRA e LOTE possuem representação espacial do tipo polígono (▣).

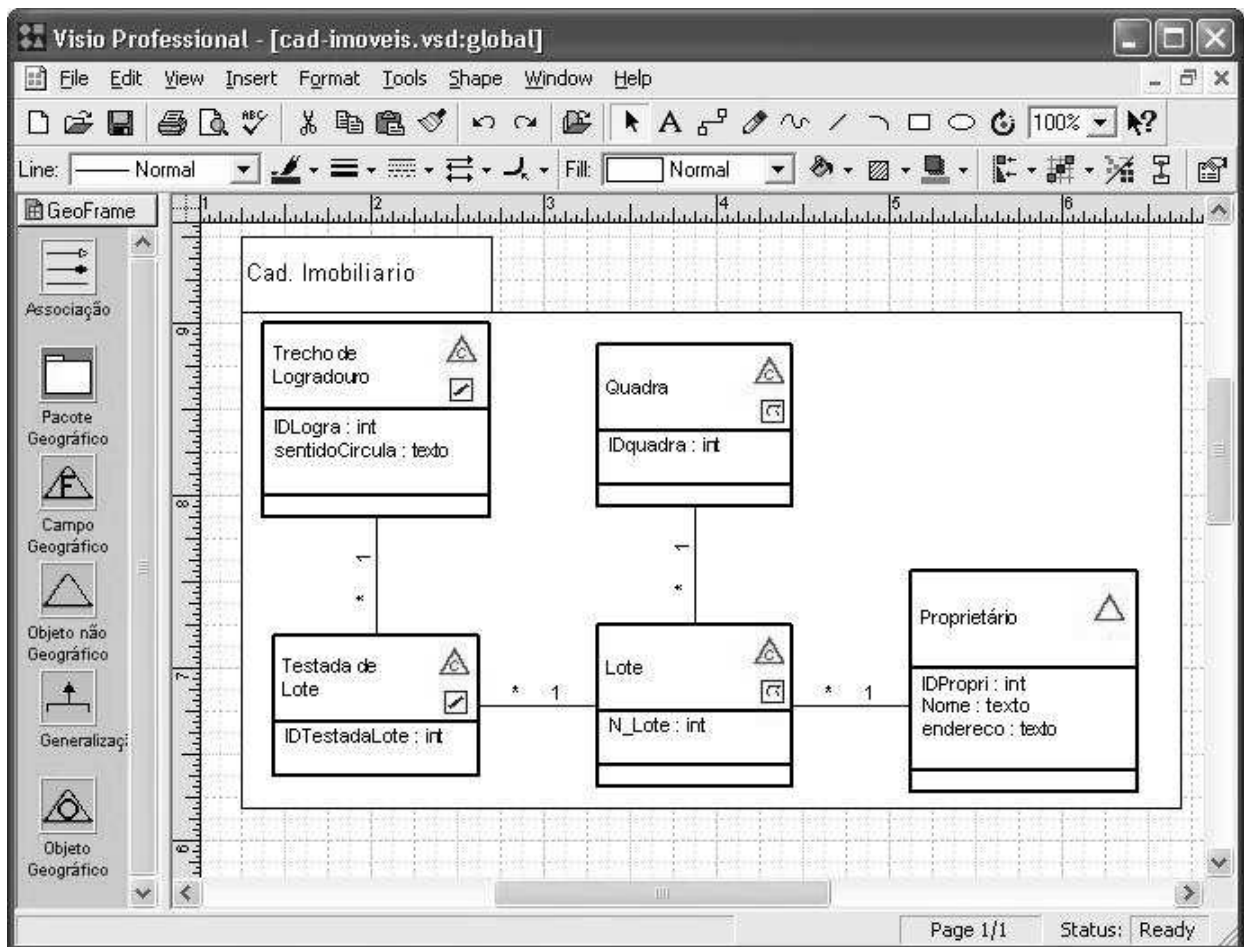


Figura 4 – Ambiente gráfico do Visio com o stencil GEOFRAME e um exemplo de esquema de dados no modelo UML-GeoFrame.

O diagrama de classes de um padrão de análise pode ser armazenado como uma unidade (sub-diagrama de classes), para reutilização posterior. Padrões de análise são armazenados na ferramenta CASEGEO como

qualquer esquema de dados. Porém, os padrões podem ser inseridos na paleta gráfica de forma que o usuário possa recuperá-lo apenas arrastando seu ícone para a área de desenho. Além disso, o projetista pode recuperar diagrama-

mas de classes pertencentes a padrões de análise disponíveis em outros repositórios da própria organização ou, por exemplo, na Internet. Um catálogo de padrões de análise encontra-se disponível no sítio do projeto REUSIG (<http://www.dpi.ufv.br/projetos/reusig/>).

4.2 Módulo Dicionário de Dados

O dicionário armazena o esquema de dados criado pelo usuário. Um esquema possui dois tipos de dados, os dados gráficos (desenho) e os dados semânticos (nomes das classes, dos atributos, multiplicidades das associações, etc). No dicionário de dados são armazenados os dados semânticos, enquanto os dados gráficos são armazenados em um arquivo do próprio Visio. O dicionário de dados é implementado no sistema Access, que é gerenciado diretamente pela ferramenta CASEGEO. A Figura 5 ilustra o diagrama de classes do dicionário de dados, que é, na verdade, o metamodelo UML-GeoFrame.

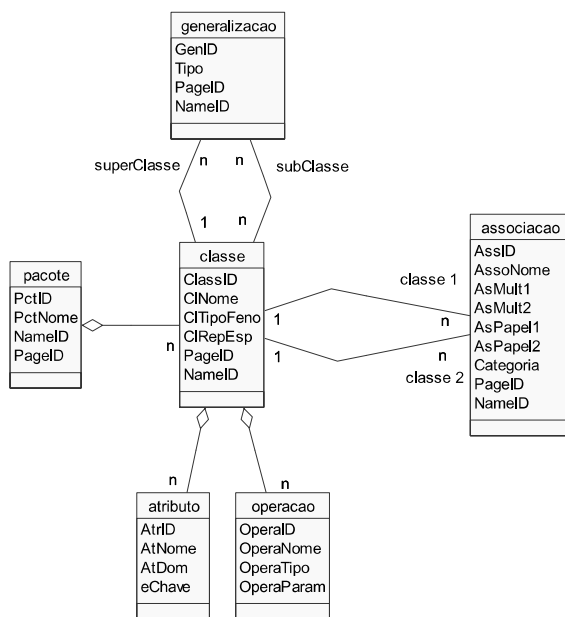


Figura 5 – Metamodelo UML-GeoFrame

A classe CLASSE contém os atributos definidos nas classes do diagrama UML-GeoFrame. Toda classe tem um identificador (ClassID), nome (CINome), o tipo de fenômeno geográfico, ou seja, se é uma subclasse de OBJETO GEOGRÁFICO, CAMPO GEOGRÁFICO ou OBJETO NÃO GEOGRÁFICO (CITipoFeno), o tipo de representação espacial, podendo ter múltiplas (CIRepEsp) e os atributos de ligação com o desenho produzido no Visio (PageID e NameID).

Uma classe pode ter atributos, que são descritos na classe ATRIBUTO. Cada atributo tem um identificador (AtrID), nome do atributo (AtNome), seu domínio (AtDom) e um indicador dizendo se este atributo faz parte da chave de identificação da classe, no modelo relacional (eChave).

Uma classe também pode ter operações, que são definidas na classe OPERAÇÃO. Cada operação especificada para uma classe tem um identificador (OperaID), nome da operação (OperaNome), tipo retornado (operaTipo) e uma lista de parâmetros (operaParam).

As associações especificadas no esquema são mantidas na classe ASSOCIAÇÃO. Cada associação tem, para cada classe envolvida, o nome da associação (assoNome), a multiplicidade (asMult1 e asMult2), o papel desempenhado por cada classe na associação (asPapel1 e asPapel2), a categoria de associação, se é uma agregação, uma composição ou se apenas uma associação (categoria) e os atributos de ligação com o desenho produzido no Visio (PageID e NameID).

Generalizações entre classes são mantidas na classe GENERALIZAÇÃO, que contém a identidade da superclasse (genID), e faz a ligação entre a superclasse e suas subclasses. Herança múltipla é permitida.

Por fim, as definições dos pacotes são mantidas na classe PACOTE. Cada pacote possui um identificador (pctID), um nome (pctNome) e os atributos de ligação com o desenho produzido no Visio (PageID e NameID).

4.3 Módulo Geração Automática

Após a modelagem conceitual, o usuário necessita transformar o esquema elaborado em uma implementação efetiva, o que é feito por meio de um software de SIG específico. Como cada software de SIG possui um modelo lógico de dados proprietário, não é possível estabelecer um conjunto único de regras de transformação para fazer a geração automática para qualquer SIG [20]. Desta forma, para cada SIG comercial, a ferramenta CASEGEO necessita de um módulo de geração automática (MGA). Espera-se que com o desenvolvimento de um modelo de dados padrão, o que vem sendo trabalhado no Consórcio OpenGIS [22], este problema seja resolvido. O primeiro módulo de geração de esquemas da ferramenta CASEGEO transforma esquemas UML-GeoFrame para o SIG ARCVIEW 3.2 (ESRI). Módulos de geração para outros sistemas como o GeoMedia (Intergraph) e o Spring (INPE) estão em desenvolvimento. Além desses, será desenvolvido um módulo de geração para o componente *Spatial* do Oracle 8i.

MGA UML-GeoFrame / ARCVIEW

O módulo de geração automática para o ARCVIEW, implementado na linguagem *Avenue*, tem como entrada a identificação do banco de dados do esquema a ser gerado, ou seja, o dicionário de dados criado pela ferramenta CASEGEO. Os temas gerados no ARCVIEW são armazenados no formato *shape*, enquanto as tabelas associadas a estes temas são armazenadas no formato dBase (dbf).

Para cada elemento do esquema conceitual de dados é aplicada uma determinada regra de transformação. A seguir estão descritas estas regras:

Regra 1 – Pacotes

Um pacote é composto por um conjunto de classes inter-relacionadas. Para cada pacote definido no esquema conceitual é gerada uma nova *view* (uma coleção de temas), recebendo o mesmo nome do pacote. Classes que não estejam definidas dentro de um pacote geram temas em uma *view* padrão.

Regra 2 – Classes do tipo Objeto Geográfico (△)

Cada subclasse de OBJETO_GEOGRÁFICO gera pelo menos um *layer* dentro de sua *view* correspondente, cujo tipo de objeto espacial é definido de acordo com o estereótipo de representação espacial associado à classe.

O conjunto de atributos definidos na classe dá origem a uma tabela (dbf) que fica associada ao *layer* correspondente através de atributos de ligação (*shape, id*).

Se a classe possui múltiplas representações espaciais é gerado um *layer* para cada tipo de representação. Por exemplo, uma classe RIO pode ter dois tipos de representação espacial - linha e polígono. Se isto for especificado no esquema conceitual, o módulo de geração automática gera dois temas (duas *shapes*) com o nome da classe RIO, um *layer* do tipo linha e outro do tipo polígono. Os dois *layers* ficam associados a tabelas contendo apenas os atributos de ligação. Os demais atributos especificados na classe dão origem a uma terceira tabela que pode ser associada a estas tabelas através de operações de junção.

Classes cuja representação espacial seja complexa (☒) devem ser transformadas manualmente.

Regra 3 – Classes do tipo Campo Geográfico (△)

Como o ARCVIEW é um software vetorial, o módulo gera automaticamente apenas as classes de objetos geográficos, ou seja, cuja representação espacial seja ponto, linha ou polígono. Assim, campos geográficos não podem ser mapeados automaticamente para o ARCVIEW. Para isto seria necessário utilizar, por exemplo, uma extensão como o *ArcView Spatial Analyst*. Na fase atual esta opção ainda não foi contemplada. As regras de transformação de

campos geográficos estão sendo definidas no módulo de geração automática para o sistema GeoMedia.

Regra 4 – Classes do tipo Objeto Não Geográfico (△)

Cada subclasse de OBJETO_NÃO_GEOGRÁFICO gera apenas uma tabela (dbf) e não gera um *layer* dentro de uma *view* por este tipo de classe não possuir representação geográfica.

Regra 5 – Associação, Agregação e Composição

As transformações de relacionamentos dos tipos agregação e composição são feitas de forma idêntica à transformação de relacionamentos do tipo associação, ou seja, são realizadas de acordo com a multiplicidade especificada. Existem basicamente três grandes tipos de multiplicidades de uma associação: um-para-um (1..1); um-para-muitos (1..*); e muitos-para-muitos (*..*). As regras de transformação de associações especificadas em um esquema conceitual para um esquema lógico de um SGBD relacional são bem conhecidas. Para maiores detalhes o leitor pode consultar [7].

Regra 6 – Generalização

A transformação de relacionamentos do tipo generalização pode ser feita de três formas distintas, descritas abaixo. Estas opções de generalização são informadas pelo projetista em tempo de criação do esquema conceitual, sendo a opção (a) adotada como padrão. São elas:

- os atributos da superclasse e das subclasses são unidos em uma só tabela. É acrescentado um campo à tabela para dizer a que tipo de subclasse a linha se refere;
- os atributos da superclasse são acrescentados ao conjunto de atributos das subclasses. Posteriormente é criada uma tabela para cada subclasse;
- para cada classe é criada uma tabela contendo os seus atributos. As tabelas referentes às subclasses têm a mesma chave da tabela referente à superclasse.

Para exemplificar o processo de geração automática pode-se tomar como entrada o dicionário de dados do esquema conceitual mostrado na Figura 4 (Pacote Cadastro Imobiliário). O módulo de geração automático irá criar o esquema lógico mostrado na Figura 6.

5 Conclusões

A ferramenta CASEGEO auxilia os projetistas a desenvolverem suas aplicações de SIG com qualidade, pois podem utilizar-se das regras já consagradas em banco de dados para criar o seu esquema lógico espacial, gerar documentação (esquema conceitual, dicionário de dados e os mapas vetoriais) para consulta e visualização

posteriores, o que facilita também uma futura manutenção do sistema e a geração imediata de uma nova versão da aplicação com as atualizações. Uma versão funcional da ferramenta CASEGEO encontra-se disponível no sítio <http://www.dpi.ufv.br/projetos/reusig/casegeo/>.

O uso de ferramentas CASE no processo de desenvolvimento de aplicações de SIG pode fazer com que o tempo de criação do projeto seja reduzido consideravelmente, com conseqüente redução de custos, além de possibilitar a reutilização de esquemas já existentes, sendo este um dos maiores diferenciais deste projeto.

Como trabalhos futuros serão implementados módulos de geração automática para outros softwares de SIG e o desenvolvimento do Módulo de Engenharia Reversa.

Agradecimentos

Este projeto é parcialmente financiado pela FAPEMIG e pelo CNPq.

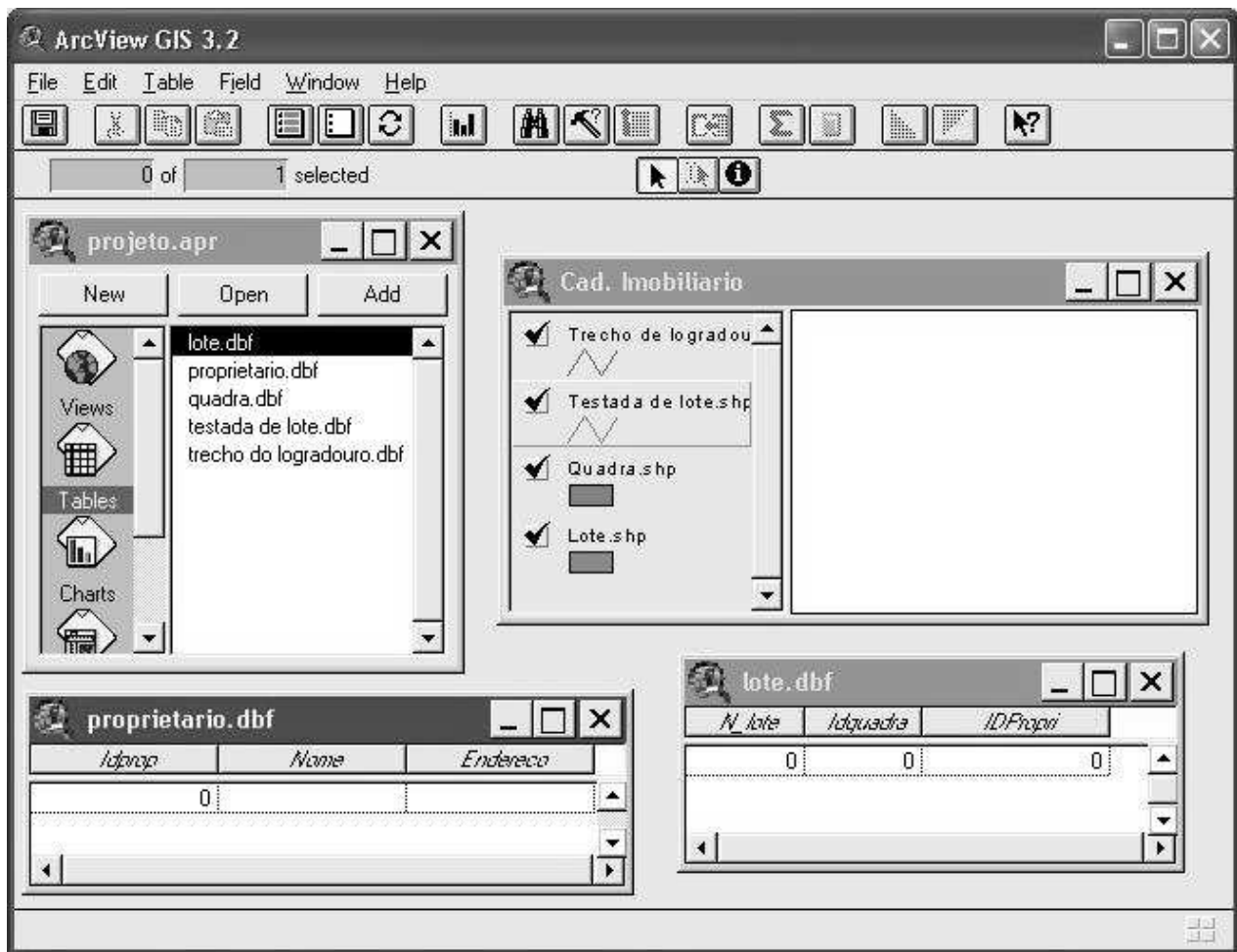


Figura 6 – Parte do esquema do Pacote Cadastro Imobiliário gerado automaticamente no ARCVIEW

Referências bibliográficas

- [1] Bédard, Y. “Visual modeling of spatial databases towards spatial extensions and UML”. *Geomatica*, 53(2) 1999, 169-186.
- [2] Bhering, E. M.; Lisboa Filho, J.; Calijuri, M. L.; Souza, L. A. DE. A. “Sistema de informação da rede de infra-estrutura sanitária de Cachoeiro de Itapemirim-ES”. *Informática Pública*, 4(1), 2002.

- [3] Booch, G.; Jacobson, I.; Rumbaugh, J. *The Unified Modeling Language User Guide*. Addison-Wesley, 1998.
- [4] Borges, K. A. V.; Davis Jr, C. D. Laender, A.H.F. "OMT-G: an object-oriented data model for geographic applications". *GeoInformatica*, 5(3), 2001.
- [5] Brodeur, J; Bérard, Y.; Proulx, M., J.; "Modeling Geospatial Application Databases using UML-based Repositories Aligned with International Standards in Geomatics". In *Proc. 8th ACM GIS*, Washington D.C, 2000.
- [6] Chrisman, N. *Exploring Geographic Information Systems*. John Wiley & Sons, 1997.
- [7] Elmasri, R.; Navathe, S. B. *Fundamentals of Database Systems*. 3.ed. Addison-Wesley, 2000.
- [8] Fernandez, E. B.; Xiaohong, Y.; "Semantic Analysis Patterns". In *Proc. ER2000 Conference*, 2000.
- [9] Fowler, M. *Analysis Patterns: Reusable Object Models*. Addison Wesley Longman, 1997.
- [10] Gamma, E. et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, 1994.
- [11] Goodchild, M. F., "Geographical data modeling". *Computers & Geosciences*, 18, 4, (1992) p.401-408.
- [12] Hay, D. C. *Data Model Patterns: Conventions of Thought*. Dorset House Publishing, 1995.
- [13] Isoware. *CASE-Toll REGIS*. Disponível em <http://www.isoware.de/> Acessado em março de 2002.
- [14] Kösters, G. et al. "GIS-Application Development with GeoOOA". *Int. Journal of GIS*, 11(4), 1997.
- [15] Lbath A., Pinet, F. "The Development and Customization of GIS-Based Applications and Web-Based GIS Applications with the CASE Tool AIGLE". In *Proc. 8th ACM GIS*, Washington D.C, 2000.
- [16] Lisboa Filho, J.; Iochpe, C.; Beard, K. "Applying Analysis Patterns in the GIS Domain". In *Proc. 10th Annual Colloquium of the SIRC*, Dunedin, NZ, 1998,.
- [17] Lisboa Filho, J.; Iochpe, C. "Specifying analysis patterns for geographic databases on the basis of a conceptual framework". In *Proc.7th ACM GIS*, Kansas City, 1999.
- [18] Lisboa Filho, J.; Iochpe, C. "Um estudo sobre modelos conceituais de dados para projeto de bancos de dados geográficos". *Informática Pública*, 1(2), 1999.
- [19] Lisboa Filho, J.; Costa, A. C.; Iochpe, C. "Projeto de banco de dados geográficos: mapeando esquemas GeoFrame para o SIG Spring". In *Proc. GEOINFO – 1st Brazilian Workshop on Geoinformatics*, Campinas, 1999.
- [20] Lisboa Filho, J. "Projeto de Banco de Dados para Sistemas de Informação Geográfica". In: Nunes, R. C. *VIII Escola de Informática da SBC Sul*. Editora da UFSM, 2000. pp.115-146.
- [21] Lisboa Filho, J.; Iochpe, C.; Borges, K. A. "Padrões de Análise para Aplicações de Gestão Urbana em Sistemas de Informação Geográfica". *Informática Pública*, 4(1), 2002.
- [22] OpenGIS. *The OpenGIS Guide*. K. Buehler and L Mckee (eds). Open GIS Consortium Technical Committee, 1998.
- [23] Parent, C. et al. "Spatio-temporal conceptual models: data structures + space + time". In *Proc.7th ACM GIS*, Kansas City, 1999.
- [24] Pinet, F.; Lbath, A. "A Visual Modelling Language for Distributed Geographic Information Systems". In: *Proc 16th IEEE Symposium on Visual Languages (VL'2000)*, Seattle, 2000.
- [25] Visio Corporation. *Developing Visio Solutions*. 1997. (Documentação de software)