

Projetando e Executando Processos de Análise Geográfica com GISCASE

CIRANO IOCHPE¹

CLÁUDIO RUSCHEL¹

GUILLERMO N. HESS¹

LUCIANA V. DA ROCHA¹

ALÉCIO P. D. BINOTTO¹

MÁRCIA ALMEIDA¹

Palavras-chave

Processos de análise geográfica - Modelagem conceitual - UML e ferramenta CASE.

Resumo

Este trabalho apresenta o GISCASE, uma ferramenta CASE UML baseada no framework conceitual GeoFrame-A para auxílio aos usuários de SIG na modelagem de seus processos de análise geográfica. A partir de uma especificação gráfica do processo geográfico, a ferramenta gera um código XML intermediário, independente de plataforma, criando programas específicos para um sistema de SIG. Adicionalmente, o GISCASE é todo desenvolvido utilizando software e tecnologias livres, independente de plataforma de SIG e tem sua arquitetura dividida em módulos, o que significa que é simples desenvolver módulos com suporte a uma vasta gama de programas de SIG.

1. Introdução

A administração pública (tanto em nível municipal, estadual ou federal) utiliza uma quantidade significativa de informação de natureza geográfica. Alguns conjuntos de dados espaciais como divisão territorial, setores censitários, bacias hidrográficas, são altamente disponíveis no Brasil, gratuitamente ou com baixo custo. Por outro lado, muitas organizações públicas têm acesso a dados geográficos originados de cartografia, sensoramento remoto ou levantamentos de campo, geralmente obtidos através de elevados investimentos e de iniciativas próprias.

Dados dessa natureza são administrados por sistemas de informação geográficos (SIG), que apresentam recursos para armazenamento, processamento e apresentação de resultados, principalmente sob forma de mapas. Comumente, os usuários destas ferramentas são oriundos de áreas que não a computação, como geografia, geologia, planejamento urbano, entre outras. Desse modo, eles nem sempre estão familiarizados com a modelagem dos processos de análise geográfica que devem executar.

A prática na manipulação dos dados geográficos tem nos mostrado que há uma necessidade por modelos e ferramentas que auxiliem na definição e modelagem tanto de dados geográficos quanto de processos de análise geográfica, em uma linguagem tal que todos os tipos de usuários possam entender. As ferramentas utilizadas na Engenharia de Software, mais especificamente aquelas utilizadas para modelagem bancos de dados geográficos (BDG), podem ser consideradas como

¹ {ciochpe, claudior, hess, vargas, abinotto, marcia }@inf.ufrgs.br

candidatas a suprir essa necessidade. Alguns trabalhos relacionados propuseram modelos de dados e frameworks para guiar a modelagem de BDG, tais como MADS [4], GeoFrame-T [6] e GeoOOA [3]. Contudo, esses modelos e frameworks somente suportam a descrição dos componentes estáticos de um SIG, isto é, os dados. Dessa forma, há uma falta em termos de modelos e ferramentas para os componentes dinâmicos do SIG, isto é, os processos. Embora alguns softwares de SIG forneçam ferramentas para a definição de processos, estas são proprietárias e dependentes de plataforma, projetadas especificamente para o software em questão.

Adicionalmente, diversos aplicativos que não necessariamente são softwares de SIG podem se valer de alguma funcionalidade de um SIG ou ainda de processos (ou resultados de processos) por ele executado. Como as APIs dos softwares de SIG são bastante complexas, também há uma lacuna no modo como as ferramentas de SIG podem intercambiar dados (por dado entende-se dados propriamente ditos, processos, funções, etc.) entre si e com aplicações convencionais.

Deste modo, estamos propondo o GISCASE, uma ferramenta de software para guiar o projeto de processos de análise geográfica (PAG) baseada no framework conceitual chamado GeoFrame-A [8]. O objetivo é o de prover uma interface gráfica, utilizando uma semântica formal que permite aos usuários não muito familiarizados com as linguagens de SIG, a definição de seus próprios processos utilizando diferentes softwares de SIG, uma vez que possui uma arquitetura independente de plataforma.

O artigo está organizado como segue. Após iniciar com uma breve motivação para este trabalho, descrevendo a idéia por trás do GISCASE, nós apresentamos um panorama geral do desenvolvimento do framework GeoFrame-A, apresentando as origens da idéia da construção de um framework conceitual para a modelagem dos dados, que é a base do GISCASE. Na seção seguinte é diretamente apresentada a arquitetura do GISCASE, especialmente os módulos do editor gráfico, GPtoXML e XMLtoGIS, seguido pela descrição da implementação da técnica proposta, utilizando a API do TerraLib. Finalmente, os resultados do trabalho são discutidos, bem como as conclusões e as direções dos trabalhos futuros.

2. Operações de Análise Geográfica

Um Processo de Análise Geográfica (PAG) é, em geral, composto por longos (e também complexos) algoritmos, utilizando operações que transformam os dados envolvidos. Grande parte das funcionalidades utilizadas em processos de análise geográfica é implementada na maioria dos softwares de SIG, mas não utilizam os mesmos nomes nem organizam os parâmetros necessários da mesma forma, de modo que o usuário não consegue reaproveitar processos que já tenha executado.

Para definição do GeoFrame-A, realizou-se um levantamento sobre como operações que envolvem análise geográfica são classificadas por diversos autores. O resultado desse estudo definiu um conjunto básico de operações a serem empregados na modelagem de PAG. Para cada operação, determinou-se os possíveis parâmetros de entrada e tipos de resultado, conforme as representações previstas no GeoFrame-T. Dentre o conjunto básico de operações as relacionadas a seguir destacam-se:

- Seleção ou "Seleção Não-Espacial": restringe o conjunto de dados espaciais ou descritivos sobre o qual é aplicada, para aqueles que satisfaçam um predicado de seleção. Semelhante à instrução `SELECT` da linguagem SQL.
- Seleção por Região: restringe o conjunto de dados espaciais sobre o qual é aplicada para aqueles que estejam contidos em uma região definida por um polígono.
- Seleção Espacial: restringe o conjunto de dados espaciais sobre o qual é aplicada para aqueles que possuam uma relação espacial (ex: toca, disjunto, está contido) sobre um conjunto de dados espaciais de referência.

- Zona de Influência ou “Buffer”: cria um polígono a partir de uma distância específica de objetos espaciais de entrada.
- Superposição ou “Overlay”: cria um novo conjunto de dados espaciais através da superposição de dois conjuntos de dados espaciais.
- Agregação ou “Dissolve”: cria um novo conjunto de dados espaciais através da agregação de dados espaciais que possuam valores iguais em um mesmo atributo.
- Classificação ou Álgebra: atualiza valores de atributos a partir de expressões lógicas e matemáticas. Através de parâmetros específicos pode contemplar funcionalidades da álgebra de mapas.
- Interpolação: reúne diversas operações de análise espacial e de modelagem numérica de terreno.
- Distância: calcula a distância entre dois dados espaciais específicos.
- Transformação: conversão de coordenadas entre projeções cartográficas.
- Centróide: cria um ponto para cada polígono.
- Diagrama Voronoi: oposto ao centróide, gera polígonos adjacentes a partir de pontos.
- Declividade: a partir de um modelo numérico de terreno, calcula a declividade de cada unidade espacial.
- Intervisibilidade: classificação de uma região em visível e não-visível a partir de um ponto de observação.
- Difusão: utilizado em modelagem de rede ou em campos geográficos, determina rotas de menor custo.

As três últimas operações apresentadas, apesar de serem consideradas básicas são específicas de domínios de aplicação como modelagem de terreno e de redes.

3. O Framework Geoframe-A

Um framework conceitual, como o definido para o GeoFrame[11], significa um projeto genérico de um domínio específico que pode ser adaptado para servir como um padrão para a construção de aplicações. No caso do GeoFrame, aplicações de SIG. O GeoFrame é um framework conceitual orientado a objetos baseado na linguagem Unified Modeling Language (UML) [2].

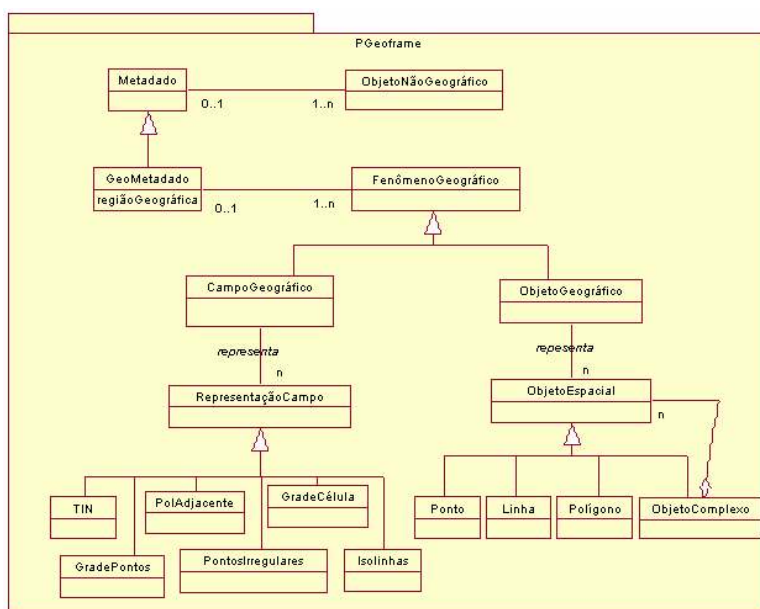


Figura 1-
O pacote PGeoFrame

O framework GeoFrame oferece um diagrama de classes, o qual está especificado no pacote PGeoFrame da figura 1 [11]. Este pacote apresenta as classes que são usadas como a base para a modelagem de uma aplicação de SIG. O esquema de dados produzido com o uso deste framework pode ser denominado como esquema UML-GeoFrame.

A figura 1 apresenta o diagrama de classes do pacote PGeoFrame, com as classes utilizadas no framework conceitual. Assim, os objetos que possuem o componente espacial são representados como instâncias da classe FenômenoGeográfico, enquanto que os objetos que não possuem representação espacial (objetos descritivos, ou não-espaciais) são instanciados a partir das classes ObjetoNãoGeográfico. Seguindo o princípio de dicotomia dos dados geográficos introduzidos por Goodchild [12], temos a visão de campo e a visão de objeto. No GeoFrame, estas visões estão representadas pelas classes CampoGeográfico e ObjetoGeográfico respectivamente, as quais possuem a sua representação espacial especificadas pelas classes RepresentaçãoCampo e ObjetoEspacial. Conforme mostra a figura 2 a representação espacial no GeoFrame é feita através de estereótipos, introduzidos como pictogramas no canto superior direito do retângulo que representa Classes no formalismo da UML.

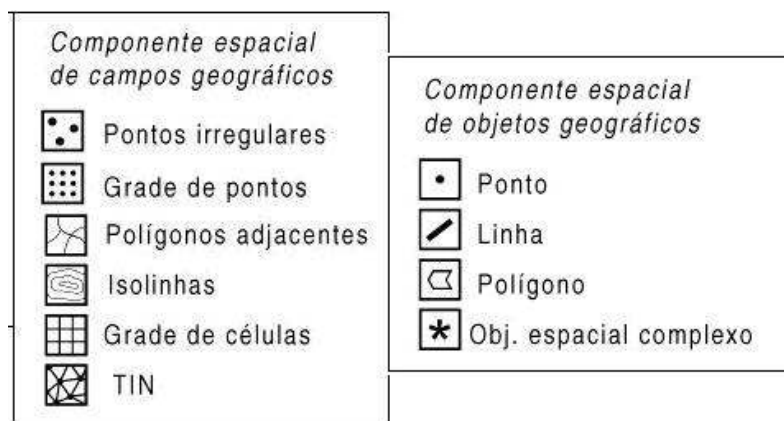


Figura 2 - Os estereótipos espaciais do GeoFrame

Uma extensão do framework GeoFrame foi proposta por Rocha [6] permitindo que a dimensão temporal fosse representada na modelagem dos dados, além da dimensão espacial proposta por Lisboa [11]. Nesta extensão é possível representar a temporalidade tanto nos aspectos espaciais dos dados quanto na parte descritiva destes, sendo que o mesmo mecanismo de estereótipos foi utilizado para representar a dimensão temporal. O exemplo apresentado a seguir do uso do framework GeoFrame-T serve para demonstrar a importância da modelagem dos dados geográficos usando temporalidade, como mostra a

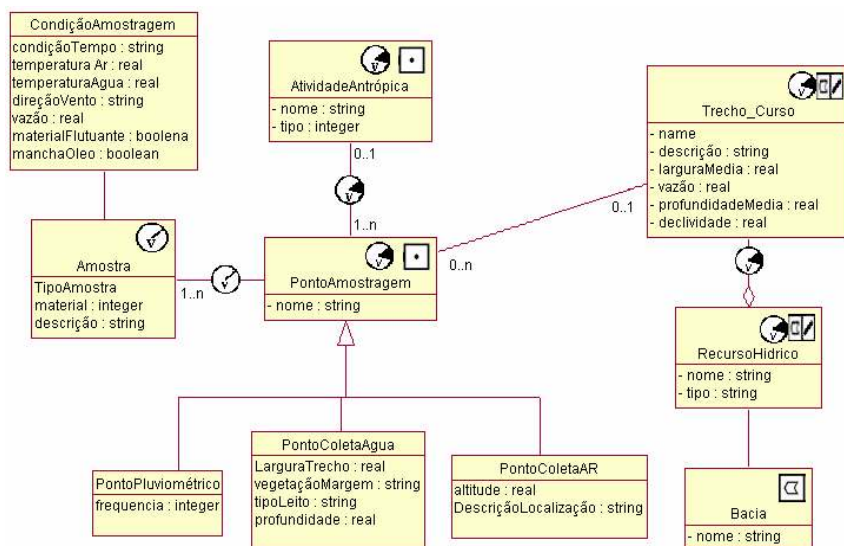


Figura 3 - Exemplo de modelagem de dados usando o GeoFrame-T

figura 3 com a modelagem de uma aplicação para controle da poluição ambiental. Neste exemplo é possível observar a representação em alto nível dos aspectos da realidade representados e seus relacionamentos.

Nesta modelagem são consideradas as atividades antrópicas (qualquer atividade produzida pelo homem) que produzem poluição atmosférica e/ou lançam resíduos na água, como por exemplo uma indústria, um curtume, etc. Assim, uma atividade antrópica tem associada a ela pelo menos um ponto para coleta de amostra, sendo que este ponto pode ser para coleta de ar, coleta de água, ou controle pluviométrico. Estes pontos registram a quantidade de poluentes lançados no meio ambiente sendo feita periodicamente amostragem destes lançamentos. São consideradas para cada amostragem, as condições em que a mesma foi adquirida.

Esta modelagem considerou desejável manter as amostras feitas em um ponto de amostragem. Isso está modelado no relacionamento temporal existente entre Amostra e PontoAmostragem. Um ponto de amostragem está associado a uma atividade antrópica ou a um trecho de recurso hídrico. Como é interesse saber todos os pontos de amostragem vinculados a uma atividade antrópica com suas respectivas amostras, o relacionamento existente entre PontoAmostragem e AtividadeAntrópica também é temporal. Dessa forma pode ser feito um controle efetivo da poluição que uma determinada atividade produziu ao longo do seu ciclo de vida.

Apesar de o GeoFrame com a extensão temporal já ser capaz de representar a contento a realidade através da representação estática dos dados, a parte dinâmica não estava expressa no modelo. Esforços contínuos foram recentemente feitos por Ruschel [7] para permitir maior poder de expressão ao GeoFrame, permitindo que além da modelagem estática dos dados, como visto no exemplo anterior, o framework pudesse expressar a parte que envolve a dinâmica da transformação dos dados, ou seja, o processo de análise geográfica usado na manipulação dos dados em um SIG. Assim, foi desenvolvido o GeoFrame-A (GeoFrame com ação), estendendo o GeoFrame com suporte a modelagem de processos de análise geográfica.

No GeoFrame-A, especificamente no diagrama de classes onde os dados espaciais e descritivos são modelados, o PAG é representado com sendo uma Classe de Atividade (Activity Class), definida na especificação da UML 2. Tirando proveito dos conceitos da UML, o comportamento de uma Atividade é caracterizado como uma seqüência de unidades subordinadas onde cada elemento individual é uma Ação. Um estereótipo semelhante a uma engrenagem identifica esta classe, enquanto outro estereótipo descreve a representação espacial.

Adicionalmente, o diagrama de atividades da UML detalha a especificação do PAG e é possível definir fluxo de controle e de dados durante o PAG, bem como as entradas necessárias para cada operação e as saídas esperadas. Mais ainda, os dados são representados como nodos de objetos e as operações como nodos de atividades.

A metodologia proposta para utilização do GeoFrame-A estimula a utilização de diagramas de caso de uso e diagramas simplificados de atividade nas fases iniciais de modelagem. Entretanto, para especificações de problemas mais simples, tais diagramas podem ser suprimidos. Recomenda-se que, ao menos, sejam desenvolvidos o diagrama de classes, com a especificação externa do problema de análise geográfica, e o diagrama de atividade, com a especificação interna do problema, envolvendo todas as operações necessárias.

Ao longo deste artigo é apresentado um exemplo de processo simples de análise geográfica modelado com o GeoFrame-A e implementado com a ferramenta GISCASE. O problema inserido no processo consiste em identificar áreas específicas em municípios que se encontram a mais de 30

quilômetros de distância de estradas pavimentadas. Cada área identificada será representada através de um ponto, para facilitar a apresentação em mapas e o processamento do resultado por outras aplicações. A resolução desse problema pode ser usada em diversas aplicações de uma administração em nível estadual ou federal, como, por exemplo, subsídio para priorização de obras em programas de pavimentação de estradas, ou identificação de áreas com potencial econômico prejudicado pela dificuldade de acesso.

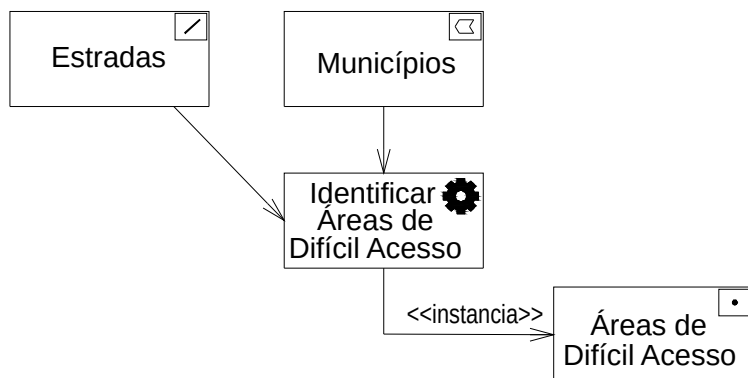


Figura 4 -
Diagrama de classes
utilizando o GeoFrame-A

A Figura 4 apresenta a definição desse processo usando um diagrama de classes com o GeoFrame-A. São utilizadas apenas duas classes de informação espacial: *Estradas*, representada por linhas e *Municípios*, representados por polígonos. O resultado do processo é um conjunto de pontos representando áreas distantes da malha rodoviária pavimentada.

Apesar de poder ser produzido através do editor gráfico disponível no GISCASE, o diagrama de classes não é utilizado diretamente pela ferramenta, servindo apenas para fins de documentação. Por outro lado, o diagrama de atividade é interpretado pela ferramenta e considerado como ponto de partida para a especificação do processo na ferramenta GISCASE.

4. A Ferramenta GISCASE

A ferramenta GISCASE foi concebida para gerar código-fonte que implemente soluções para processos de análise geográfica a partir de especificações contidas em um diagrama de atividade da UML, desenvolvido conforme os preceitos do GeoFrame-A. Isso é possível porque a semântica do diagrama de atividades em nível conceitual é suficiente para realizar uma conversão automática para sistemas computacionais, se algumas convenções forem utilizadas. Assim, foi definido que os objetos das classes espaciais devem ser representados através de registros em tabelas de um banco de dados geográfico. Considerou-se também que as operações definidas no diagrama de atividades têm implementações correspondentes disponíveis nas APIs dos softwares de SIG.

A ferramenta GISCASE foi praticamente toda desenvolvida em Java, usando o ambiente Eclipse. A opção pela linguagem Java ocorreu porque é de distribuição livre, independente de plataforma, além de oferecer uma série de recursos para interpretar arquivos em formato XML. A ferramenta consiste em três módulos principais, que devem ser executados de forma consecutiva: Editor Gráfico, GPtoXML e XMLtoGIS. A figura 5 ilustra a arquitetura que será detalhada nas seções a seguir.

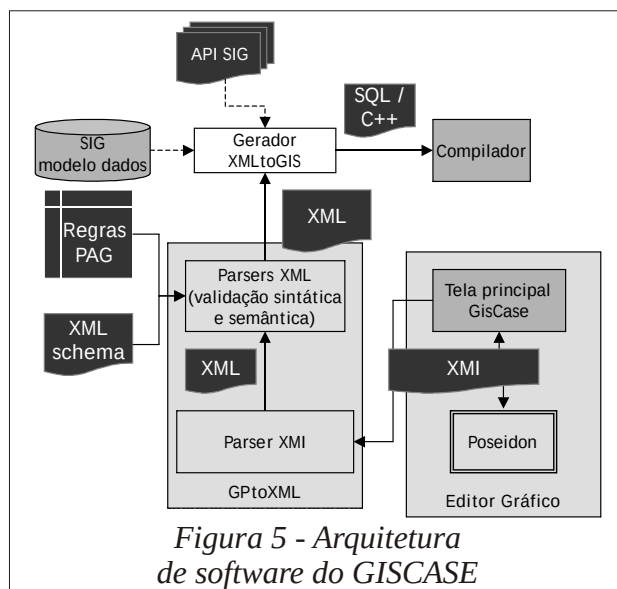


Figura 5 - Arquitetura
de software do GISCASE

Os módulos são gerenciados por uma interface principal, na qual o usuário pode visualizar (e também editar) os resultados de cada módulo. Também através desta interface o usuário pode definir qual o sistema de banco de dados e API de SIG que deve ser utilizada. Em outras palavras, o usuário define graficamente um PAG e a ferramenta gera os programas correspondentes automaticamente.

A figura 6 apresenta uma cópia de tela da interface principal do GISCASE. Ela está dividida em 3 janelas: à esquerda a árvore de navegação, à direita e acima a janela com o código do arquivo ativo e à direita e abaixo a janela de log.

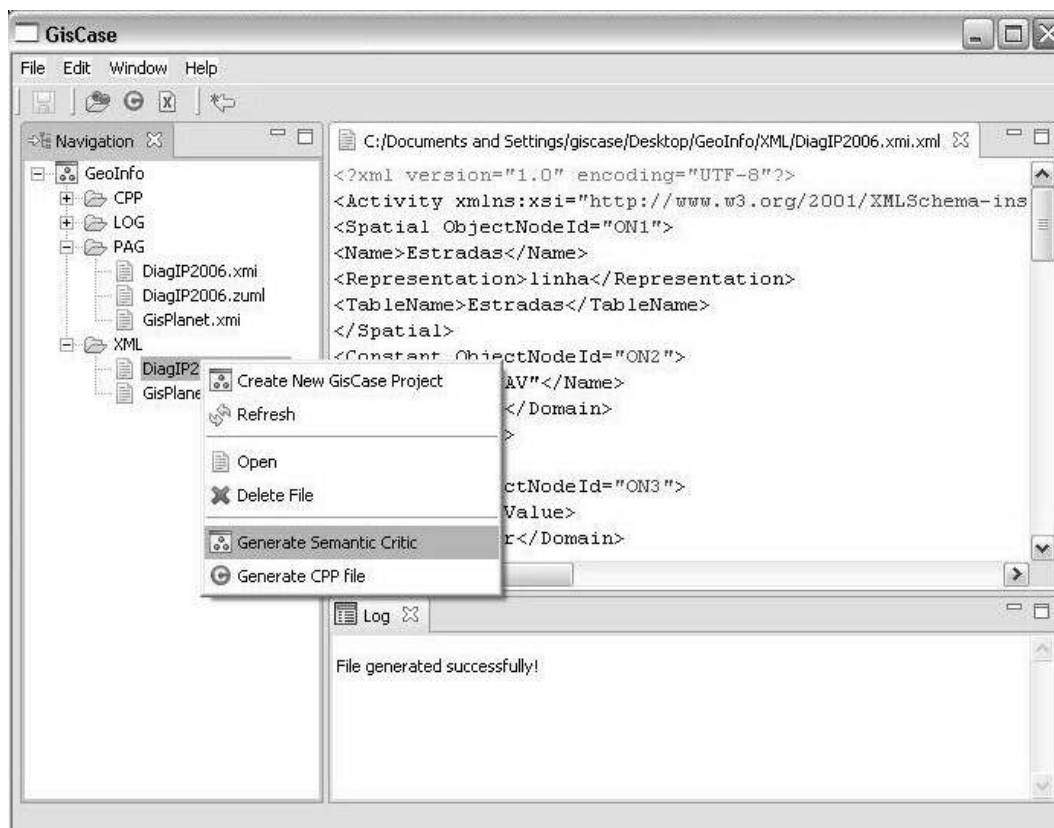


Figura 6 - Tela da interface do GISCASE

A janela de navegação permite ao usuário visualizar a estrutura do seu projeto. Para cada projeto de processo do usuário é criada uma pasta (um diretório) e abaixo deste são criadas outras pastas. A de nome CPP contém os arquivos com os códigos-fonte gerados para as APIs de SIG. O diretório PAG contém os arquivos XMI dos processos definidos pelo usuário na ferramenta case externa ao GISCASE. Na pasta XML são armazenados os arquivos XML gerados pela ferramenta a partir do PAG do usuário. Por fim, a pasta LOG apresenta os arquivos de log gerados.

A janela principal apresenta o código do arquivo ativo, podendo ser este XMI, XML ou CPP. Nesta tela o arquivo está habilitado para ser editado pelo usuário. A janela de log, por sua vez, apresenta as mensagens geradas durante o processo de transformação de um processo. As mensagens podem ser tanto de sucesso quanto de fracasso, especificando quais os erros encontrados.

Na implementação inicial, o GISCASE suporta a API TerraLib [9], biblioteca brasileira desenvolvida pelo INPE para aplicações utilizando banco de dados geográficos e linguagem C++. Esta biblioteca é baseada em um projeto ainda em andamento e classificado como de código aberto, software livre e desenvolvido segundo a orientação a objetos. Dentre as diversas funcionalidades que apresenta, possui implementação para o conjunto de operações geográficas definidas para implementação no GISCASE.

As aplicações desenvolvidas com a TerraLib acessam informações espaciais e descritivas armazenadas em diversas opções de software de banco de dados (Access, PostGres, Oracle, SQL Server). Para que esses dados sejam reconhecidos é necessário que sejam mantidos através de um esquema de dados específico para a TerraLib. Portanto, nessa versão inicial, para que o programa gerado pelo GISCASE possa ser executado com sucesso, é necessário que os dados a serem utilizados sejam previamente carregados em um banco de dados geográfico que contenha o esquema de dados da TerraLib.

O modo mais simples disponível atualmente para criar esse esquema de dados e efetuar a carga de dados é a partir do aplicativo TerraView [10], que pode ser considerado como um “Desktop GIS”. Esse aplicativo, construído com a TerraLib e de distribuição livre, permite a importação de arquivos em formato “shapefile” para um banco de dados compatível com o esquema da TerraLib.

4.1 Editor Gráfico

O editor gráfico é uma ferramenta externa que pode ser ativada pelo GISCASE e que serve para o usuário criar seu PAG graficamente, usando o diagrama de atividades.

O GISCASE foi concebido para usar um editor gráfico genérico ou ferramentas CASE que tenham, no mínimo, os elementos básicos de um diagrama de atividades. O editor deve ser capaz de diferenciar um nodo de ação (Action Node) e um nodo de objeto (Object Node) e também de representar os fluxos entre eles. Cada fluxo deve ser identificado como sendo um fluxo de dados ou de controle, através de estereótipos, assim como seus papéis. Outros elementos de controle de fluxo, tais como nodos de separação (paralelismo) e junção (sincronização) devem também estar disponíveis.

O editor gráfico deve, ainda, poder gravar o resultado como um arquivo no formato OMG XML Metadata Interchange (XMI). Essa restrição deve-se ao fato de que algumas ferramentas CASE já suportam e implementam XMI, que especifica um modelo aberto para intercâmbio de dados de programação pela internet de forma padronizada, assim tornando as aplicações criadas em ambientes colaborativos consistentes e compatíveis.

Na implementação inicial do GISCASE, optou-se em utilizar uma ferramenta CASE UML existente, denominada Poseidon [5]. A razão pela qual o Poseidon foi escolhido foi por ser um requisito do projeto que toda a ferramenta GISCASE deveria ser baseada em software livre (utilizou-se a versão 3 da Community Edition), e também pela possibilidade de exportar o diagrama de classes para XMI. Esse software é ativado a partir da interface principal do GISCASE quando o usuário deseja editar graficamente um diagrama de atividades. A Figura 7 apresenta uma cópia parcial de tela da interface do Poseidon.

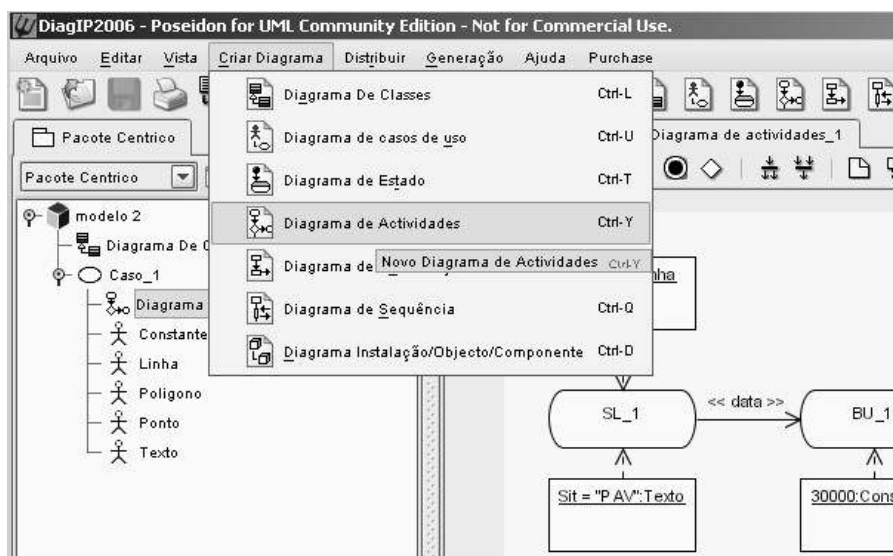


Figura 7 -
Diagrama de atividades
compatível com a ferramenta
GISCASE

O diagrama de atividades deve ser construído utilizando-se algumas convenções.

Os nodos de objetos (retângulos) são utilizados para representar os dados espaciais envolvidos (representações de ponto, linha ou polígono), assim como parâmetros necessários para a execução das operações de análise geográfica (expressões como texto, valores numéricos como constantes).

Os nodos de atividades (retângulos com cantos arredondados) identificam as operações de análise geográfica com nomenclatura reconhecida pelo GISCASE. Os nomes são reconhecidos por uma sigla de duas letras. Deve ser adicionado o símbolo de sublinhado mais um algarismo para que seja possível utilizar mais de uma vez a mesma operação no processo. O fluxo entre nodos de objetos e atividades sempre representa fluxos de dados. Já o fluxo entre nodos de atividades pode representar fluxo de dados ou de controle. Quando representa fluxo de dados, deve ser acrescentado o estereótipo <<data>>. Em algumas operações é necessário distinguir dois conjuntos de dados espaciais utilizados como dados de entrada. Dados complementares, conforme a definição de cada operação, devem receber o estereótipo <<ref>>.

A Figura 8 detalha o processo apresentado na Figura 4 através de um diagrama de atividades desenvolvido no editor gráfico adotado no GISCASE.

A operação de Seleção por Atributos (SL_1) fornece como resultado um subconjunto de estradas onde o atributo “Sit” armazena o valor “PAV” (ou seja, situação: pavimentada). Sobre as estradas selecionadas é aplicada a operação de “Zona de Influência” ou Buffer (BF_1), que cria polígonos preservando uma distância de 30.000 metros a partir da linha que representa a estrada. A operação de Superposição ou Overlay (OV_1) do tipo Diferença (Dif) gera polígonos, recortados a partir dos municípios, que não encontram-se superpostos à zona de influência de estradas. No final do processo, esses polígonos são transformados em pontos representativos através da operação Centróide (CT_1), que são armazenados em uma nova classe de dados espacial, denominada AreasDificilAcesso.

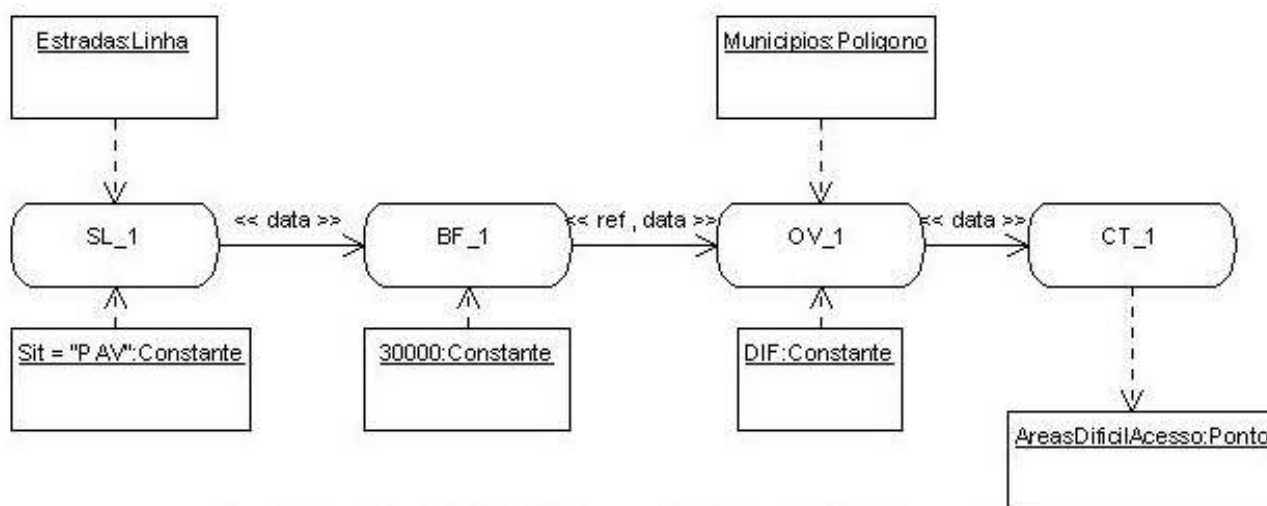


Figura 8 - Diagrama de atividades compatível com a ferramenta GISCASE

Após concluído o desenho do processo no editor gráfico, ele deve ser gravado no formato XMI (na interface do Poseidon, menu Arquivo, Exportar Projeto para XMI), de tal forma que possa ser interpretado pelos demais módulos do GISCASE.

4.2 GPtoXML Parser

O parser GPtoXML é o módulo responsável por traduzir o PAG em uma codificação na eXtensible Markup Language (XML) [1]. Primeiramente, este módulo transforma o arquivo XMI gerado no editor gráfico através de um parser XMI. Este parser foi customizado para o formato de XMI gerado pelo Poseidon. Para o caso de utilização com outro editor gráfico, o parser deverá ser adaptado.

A informação relevante (elementos) é codificada em um novo arquivo XML, o qual é baseado em um XML-Schema definido especialmente para o GISCASE. O esquema XML criado pode ser visto como a codificação do GeoFrame-A para XML [13]. Todos os componentes do modelo GeoFrame-A estão contemplados pelo esquema. Da mesma forma que os nodos e ações de um PAG são instâncias das classes do GeoFrame, o arquivo XML de um PAG é uma instância do XML-Schema que definimos.

O esquema possui dois componentes básicos, os Object Nodes (ON nodos de objeto) e os Action Nodes (AC nodos de ação). Os Object Nodes são as entradas e também as saídas para cada uma das operações da atividade descrita no PAG. Um ON pode ser um dado espacial, um campo descritivo do banco de dados, uma constante ou até mesmo uma variável.

As ações são as operações executadas durante o processo. Podem ser definidas para serem realizadas sobre a parte descritiva ou espacial do dado, e podem ser unárias ou binárias. Um AC referencia os ON que são suas entradas e sua saída. Cada ação deve possuir ao menos uma entrada e exatamente uma saída.

Conforme especificado na seção 4.1, o estereótipo <<data>> deve ser usado para especificar um fluxo de dados entre duas ações. Caso não seja especificado nenhum estereótipo no fluxo entre duas ações o parser considera que se trata de um fluxo de controle. Neste caso, o programa simplesmente ordena as duas operações. Se um fluxo de dado é encontrado, então um ON é criado como sendo a saída da primeira ação e uma das entradas da segunda ação.

Uma vez codificado em XML, o PAG é analisado para verificar sua correção semântica, isso é, é verificado se as ações foram corretamente definidas assim como suas entradas e saídas (Object Nodes). Essa verificação não diz respeito à validade do arquivo XML gerado. Ele pode ser válido como uma instância do esquema XML, mas ser inválido semanticamente. Isso ocorre quando o usuário especifica parâmetros incorretos como entradas ou como saída de uma determinada ação. Para tanto, o validador semântico verifica:

- Se o número de entradas (ON) de cada ação está correto;
- Se os tipos de dado das entradas e da saída estão todos corretos;
- Se os tipos de dado espacial especificados como entrada(s) e saída são geometricamente consistentes.

Um arquivo de log é criado para possibilitar ao usuário verificar os erros, se houver.

4.3 Gerador XMLtoGIS

O PAG descrito em XML permite o armazenamento e intercâmbio em um formato padrão. Contudo, nenhuma das soluções de SIG suporta a definição de um PAG neste formato. Desse modo, este módulo traduz o PAG em um programa específico para o SIG desejado. Para tanto, o gerador XMLtoGIS considera que todos os dados necessários na definição do processo estão armazenados em um banco de dados geográfico.

Seguindo essa idéia, para cada software de SIG é necessário o desenvolvimento de uma biblioteca específica que mapeie as operações definidas no GeoFrame-A para as operações oferecidas pela API do software de SIG. Desse modo, o módulo XMLtoGIS gera um código-fonte que faça referência e chamadas as operações desta API, passando todos os parâmetros necessários em uma ordem compatível com aquela da definição do processo.

Na implementação inicial, foi criada uma biblioteca denominada “GcGeoOperations para TerraLib”. Para cada operação que pudesse ser definida no diagrama de atividades, foi desenvolvida uma função que utiliza os recursos disponíveis da TerraLib. As funções dessa biblioteca foram escritas em linguagem C++, visto que essa é a linguagem utilizada pela TerraLib. Assim, o produto final do GISCASE é o código-fonte gerado pelo módulo XMLtoGIS, anexando o código-fonte previamente desenvolvido da biblioteca GcGeoOperations para as operações utilizadas no processo.

Foi incluído um subconjunto das operações básicas para modelagem de PAG apresentadas na seção 2, restringindo-se à utilização de objetos espaciais (ou seja, representação de ponto, linha e polígono). As operações incluídas na GcGeoOperations (com o prefixo que as identifica na definição de um processo) são: Seleção (SE), Seleção Espacial (SS), Seleção por Região (RS), Zona de Influência (BU), Superposição (OV), Agregação (DI), Distância (DT), Centróide (CT), Álgebra (AL).

Para que o código-fonte gerado possa ser executado, ele deve ser compilado e linkeditado com as bibliotecas de SIG necessárias, ou interpretado dentro do ambiente de um software de SIG. Instruções para compilação com Terralib podem ser obtidas em [9].

Quando o programa é executado, o processo lê os dados necessários no banco de dados, executa as operações geográficas tais como definidas e armazena os resultados em tabelas novas ou existentes. É importante ressaltar que os nomes reais das tabelas e campos devem ser utilizados no diagrama de atividades, de modo a que sejam reconhecidos na execução do processo. Os dados intermediários gerados na execução do processo são apagados ao final do processo.

5 Conclusões e Trabalhos Futuros

A principal contribuição deste artigo é uma nova abordagem para usuários de SIG que não estão acostumados a projetar formalmente um processo de análise geográfica (PAG), devido ao fato de que cada software possui sua própria interface e um conjunto de comandos específico. O GeoFrame-A e o GISCASE objetivam oferecer um modelo e uma ferramenta para auxiliar usuários típicos de SIG a projetar seu PAG em um alto nível de abstração. Além disso, ao permitir que um mesmo PAG definido conceitualmente seja reaproveitado e executado em diferentes softwares de SIG, apresenta aos usuários de SIG uma nova abordagem nos seus métodos de trabalho.

A arquitetura do GISCASE utiliza os princípios de software livre e desse modo possibilita que, no futuro, sejam gerados wrappers para outras bibliotecas de SIG, além da TerraLib, já suportada atualmente. Adicionalmente, ao armazenar a definição do PAG em um arquivo XML é possível compartilhar e distribuí-lo facilmente, sobretudo na Internet.

Nessa implementação inicial, o GISCASE oferece aos usuários do TerraView uma forma de gerar programas executáveis para processos executados com frequência, considerando que o TerraView é operado através de uma interface de menus e ainda não possui uma macro linguagem para execução de processos.

Como trabalhos futuros pretendemos implementar outras operações de análise geográficas além das já implementadas na biblioteca GcGeoOperations, bem como estender o GISCASE de modo a gerar código para outras bibliotecas e softwares de SIG. Já está em desenvolvimento, também, uma interface que possibilitará ao usuário definir seu PAG utilizando uma gramática formal baseada em álgebra de mapas [14], compila-la e gerar um arquivo XML compatível com o esquema XML do GISCASE [15]. Isso possibilitará ao usuário optar entre a definição do PAG utilizando um diagrama UML ou uma linguagem textual formal.

Keywords

Geographic Analysis Processes, Conceptual Modeling, UML, Case Tool.

Abstract

This work presents GISCASE, an UML CASE tool based on the conceptual framework GeoFrame-A to aid GIS users to model their geographical analysis processes. Starting from a graphical specification of the geographic process, the tool generates an XML intermediate code free of platform and then creates the specific programs for a GIS system. In addition, GISCASE is entirely developed using free software and technologies, independent of GIS platform and has a modular architecture, which means it is easy to develop modules to support a wide range of GIS Software.

6. Referências Bibliográficas

- [1] Bradley, Neil. *The XML companion*. Boston: Addison-Wesley, 2002.
- [2] Fowler, Martin. *UML Distilled: a brief guide to the standard object modeling language*. 2nd ed. Reading: Addison-Wesley, 2000.
- [3] Köesters, G.; Pagel, B.; Six, H. GeoOOA: Object Oriented Analysis for Geographic Information Systems. II International Conference on Requirements Engineering (ICRE '96). Colorado, EUA, 1996.
- [4] Parent, C. et al. Modeling Spatial Data in the MADS Conceptual Model. International Symposium on Spatial Data Handling, SDH 98. Vancouver, Canada, 1998.
- [5] Poseidon web site. Disponível em: <<http://www.gentleware.com/>>.
- [6] Rocha, L. V., Edelweiss, N., Iochpe, C. GeoFrame-T: A Temporal Conceptual Framework for Data Modeling. IX ACM GIS. Atlanta, EUA, 2001.
- [7] Ruschel, C., Iochpe, C., Rocha, L.V., Designing Geographic Analysis Processes on the Basis of the Conceptual Framework GeoFrame. ICEIS 2005. Miami, EUA, 2005.
- [8] Ruschel, C., Extensão do Framework GeoFrame para Processos de Análise Geográfica. Porto Alegre: PPGC-UFRGS, 2003. Dissertação de mestrado. Disponível em: <<http://www.inf.ufrgs.br/~ciochpe>>.
- [9] TerraLib Oficial Web Documentation. Disponível em: <<http://www.terralib.org>>.
- [10] Projeto TerraView. Disponível em: <<http://www.dpi.inpe.br/terraview>>.
- [11] Lisboa F. J. Projeto conceitual de Banco de Dados Geográficos através da Reutilização de Esquemas, utilizando Padrões de Análise e um Framework Conceitual. PhD Thesis. Porto Alegre: PGCC da UFRGS, 2000.
- [12] Goodchild, M. Geographical Data Modelling. In A. Frank, M. Goodchild, Two Perspective on Geographical Data Modelling. NCGIA, Santa Bárbara, 1990.
- [13] Iochpe, C., Hess, G. N., Ruschel, C., Binotto, A. P. D., de Almeida, M. A. S., Rocha, L. V. Designing and Performing Geographic Analysis Process with GISCASE. GEOINFO 2005. Campos do Jordão, Brasil, 2005.
- [14] Câmara, G., Freitas, U., Cordeiro, J. Towards an Algebra for Geographic Fields. VII Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens. Curitiba, Brasil, 1994.
- [15] Martins, T. E. F. Um Compilador para a Linguagem TerraMap. Porto Alegre II-UFRGS, 2005. Trabalho de conclusão de curso de graduação.

Apendice A - Código XML para o PAG da Figura 5

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Activity xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="C:\Documents and Settings\Hess\My
  Documents\UFRGS\projetos\GISCase\pag2.xsd">
- <Spatial ObjectNodeId="ON1">
  <Name>Estradas</Name>
  <Representation>linha</Representation>
  <TableName>Estradas</TableName>
</Spatial>
- <Constant ObjectNodeId="ON2">
  <Name>Sit = "PAV"</Name>
  <Domain>string</Domain>
  <Size>10</Size>
</Constant>
- <Constant ObjectNodeId="ON3">
  <Value>30000</Value>
  <Domain>integer</Domain>
</Constant>
- <Spatial ObjectNodeId="ON4">
  <Name>Municipios</Name>
  <Representation>poligono</Representation>
  <TableName>Municipios</TableName>
</Spatial>
- <Constant ObjectNodeId="ON5">
  <Name>AND</Name>
  <Domain>string</Domain>
  <Size>10</Size>
</Constant>
- <Spatial ObjectNodeId="ON6">
  <Name>AreasDificilAcesso</Name>
  <Representation>ponto</Representation>
  <TableName>AreasDificilAcesso</TableName>
</Spatial>
- <Spatial ObjectNodeId="GisCase_AC1">
  <Name>GisCase_SL_1</Name>
  <Representation>linha</Representation>
  <TableName>GisCase_SL_1</TableName>
</Spatial>
- <Spatial ObjectNodeId="GisCase_AC2">
  <Name>GisCase_BF_1</Name>
  <Representation>Poligono</Representation>
  <TableName>GisCase_BF_1</TableName>
```



```

    </Spatial>
- <Spatial ObjectNodeId="GisCase_AC3">
  <Name>GisCase_OV_1</Name>
  <Representation>poligono</Representation>
  <TableName>GisCase_OV_1</TableName>
  </Spatial>
- <Action ActionId="AC1">
  <Name>SL_1</Name>
  <Input InId="ON1" />
  <Input InId="ON2" />
  <Output OuId="GisCase_AC1" />
  </Action>
- <Action ActionId="AC2">
  <Name>BF_1</Name>
  <Input InId="GisCase_AC1" />
  <Input InId="ON3" />
  <Output OuId="GisCase_AC2" />
  </Action>
- <Action ActionId="AC3">
  <Name>OV_1</Name>
  <Input InId="GisCase_AC2" Ref="1" />
  <Input InId="ON4" />
  <Input InId="ON5" />
  <Output OuId="GisCase_AC3" />
  </Action>
- <Action ActionId="AC4">
  <Name>CT_1</Name>
  <Input InId="GisCase_AC3" />
  <Output OuId="ON6" />
  </Action>
</Activity>

```

Apêndice B - Código-fonte para TerraLib do PAG da Figura 5

```
#include <TeAdoDB.h>
#include <TeLayerFunctions.h>
#include <TeLayer.h>
#include <TeDataTypes.h>
#include <TeRepresentation.h>
#include <TeDatabaseUtils.h>
#include <TeGeoProcessingFunctions.h>
#include <TeInitQuerierStrategy.h>
#include <iostream>
#include <string.h>
#include "TeProgress.h"

int main(){
    string dbname = "C:\\Giscase\\GDB_IP.mdb"; TeAdo* db = new TeAdo();
    db->close();
    if(!db->connect("localhost","", "", dbname, 0)){
        cout << "Erro em conetar-se via ADO: " << db->errorMessage() << endl;
        cout << endl << "Press Enter";
        db->close();
        getchar();
        return 1;
    }

    cout << "Conexao estabelecida com sucesso via ADO com o banco de dados " << dbname <<
    "'!";
    TeAdoPortal* dbPortal = new TeAdoPortal(db);
    TeDatabasePortal* portal = dbPortal;
    if(!portal)
        return false;
    bool val;
    TeInitQuerierStrategies();
    val = GcSelect("Estradas", TeLINES, "", "GisCase_SL_1", TeLINES, db, portal);
    if(!val)
        cout << "Erro na função GcSelect!" << endl;
    val = GcBuffer("GisCase_SL_1", TeLINES, "GisCase_BF_1", 30000, db);
    if(!val)
        cout << "Erro na função GcBuffer!" << endl;
    val = GcOverlay("Municipios", "GisCase_BF_1", "GisCase_OV_1", "", db, portal);
    if(!val)
        cout << "Erro na função GcOverlay!" << endl;
    val = portal->freeResult();
    db->close();
    return 0;
}
```

Autores

Cirano IOCHPE

ciochpe@inf.ufrgs.br

Doutor em Ciência da Computação/ Universitat Karlsruhe, Alemanha.

Guillermo N. HESS

hess@inf.ufrgs.br

Doutorando em Ciência da Computação/UFRGS.

Cláudio RUSCHEL

claudior@inf.ufrgs.br

Engenheiro Civil, Mestre em Ciência da Computação/UFRGS.

Alecio P. D. BINOTTO

abinotto@inf.ufrgs.br

Mestre em Ciência da Computação/UFRGS. Pesquisador do CETA-Center of Excellence on Advanced Technologies

Luciana Vargas da ROCHA

vargas@inf.ufrgs.br

Mestre em Ciência da Computação/UFRGS.

Márcia ALMEIDA

marcia@inf.ufrgs.br

Mestranda em Ciência da Computação/UFRGS.