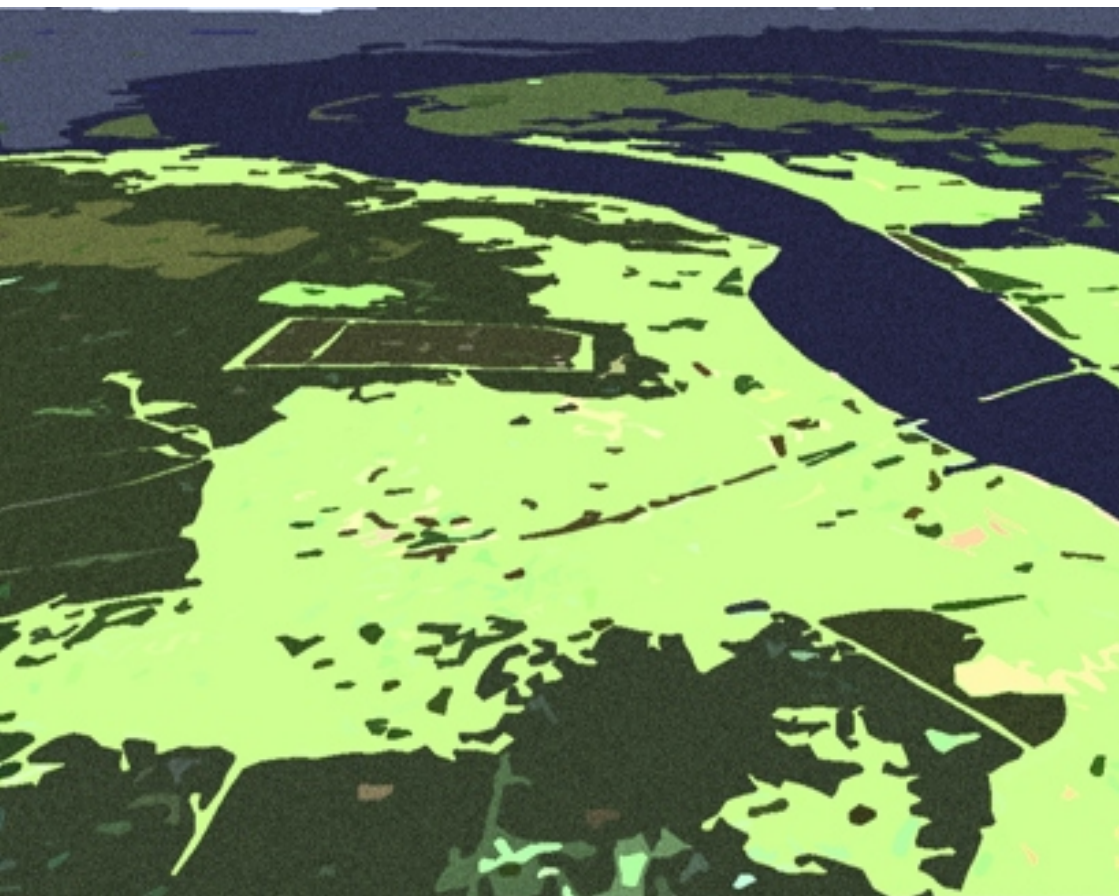


Documentos

Maio, 2006 **60**

ISSN 1677-9274

Usando o Google Earth para publicar Dados Proprietários



República Federativa do Brasil

Luiz Inácio Lula da Silva

Presidente

Ministério da Agricultura, Pecuária e Abastecimento

Roberto Rodrigues

Ministro

Empresa Brasileira de Pesquisa Agropecuária - Embrapa

Conselho de Administração

Luis Carlos Guedes Pinto

Presidente

Silvio Crestana

Vice-Presidente

Alexandre Kalil Pires

Claudia Assunção dos Santos Viegas

Ernesto Paterniani

Hélio Tollini

Membros

Diretoria-Executiva da Embrapa

Silvio Crestana

Diretor-Presidente

José Geraldo Eugênio de França

Kepler Euclides Filho

Tatiana Deane de Abreu Sá

Diretores-Executivos

Embrapa Informática Agropecuária

Eduardo Delgado Assad

Chefe-Geral

José Ruy Porto de Carvalho

Chefe-Adjunto de Administração

Kleber Xavier Sampaio de Souza

Chefe-Adjunto de Pesquisa e Desenvolvimento

Deise Rocha Martins dos Santos Oliveira

Supervisora da Área de Comunicação e Negócios



*Empresa Brasileira de Pesquisa Agropecuária
Embrapa Informática Agropecuária
Ministério da Agricultura, Pecuária e Abastecimento*

*ISSN 1677-9274
Maio, 2006*

Documentos 60

Usando o Google Earth para publicar Dados Proprietários

José Iguelmar Miranda

Campinas, SP
2006

Embrapa Informática Agropecuária
Área de Comunicação e Negócios (ACN)

Av. André Tosello, 209
Cidade Universitária "Zeferino Vaz" – Barão Geraldo
Caixa Postal 6041
13083-970 – Campinas, SP
Telefone (19) 3789-5743 – Fax (19) 3289-9594
URL: <http://www.cnptia.embrapa.br>
e-mail: sac@cnptia.embrapa.br

Comitê de Publicações

Adriana Farah Gonzalez (secretária)
Ivanilde Dispato
Kleber Xavier Sampaio de Souza (presidente)
Luciana Alvim Santos Romani
Marcia Izabel Fugisawa Souza
Stanley Robson de Medeiros Oliveira

Suplentes

José Iguelmar Miranda
Laurimar Gonçalves Vendrusculo
Maria Goretti Gurgel Praxedes
Silvio Roberto Medeiros Evangelista

Supervisor editorial: *Ivanilde Dispato*
Normalização bibliográfica: *Marcia Izabel Fugisawa Souza*
Editoração eletrônica: *Área de Comunicação e Negócios (ACN)*

1ª. edição on-line - 2006

Todos os direitos reservados.

Miranda, José Iguelmar.

Usando o Google Earth para publicar dados proprietários / José Iguelmar Miranda. —
Campinas : Embrapa Informática Agropecuária, 2006.

42 p. : il. — (Documentos / Embrapa Informática Agropecuária ; 60).

ISSN 1677-9274

1. Publicação de Mapas. 2. Mapas na Internet. 3. Google Earth. 4. Criação de Mapas.
I. Título. II. Série.

CDD — 005.720223 (21st. Ed.)

Autor

José Iguelmar Miranda

Ph.D. em Geoprocessamento, Pesquisador da
Embrapa Informática Agropecuária
Caixa Postal 6041, Barão Geraldo
13083-970 - Campinas, SP
Telefone (19) 3789-5736
e-mail: miranda@cnptia.embrapa.br

Apresentação

A disseminação de dados espaciais pela *web* ganhou um grande aliado com o lançamento do Google Earth - GE, uma iniciativa da empresa Google. Trata-se de uma aplicação extremamente rica de informações sobre todo o planeta, no que diz respeito ao uso de imagens de satélites de recursos naturais. Atualmente, a aplicação só pode ser executada em computadores com sistema operacional Windows, pelo motivo dela usar ferramentas de *software* específicas desse ambiente operacional.

A Google oferece aos interessados diferentes versões do GE, variando de US\$20,00 (vinte dólares anuais) a US\$400,00 (quatrocentos dólares), podendo esse valor ser acrescido se outras opções disponíveis forem escolhidas. No entanto, se as exigências do usuário não forem comerciais, como a manipulação de grandes bancos de dados espaciais, pode-se usar a versão cliente, de uso gratuito, para disseminar dados próprios usando essa rica ferramenta de visualização de dados espaciais.

O objetivo desse documento é mostrar como o usuário pode usar o potencial da versão cliente do GE e divulgar seus dados pela *web*.

Eduardo Delgado Assad
Chefe-Geral

Sumário

Introdução.....	9
Geography Markup Language - GML.....	10
Keyhole Markup Language - KML.....	19
Google Earth.....	21
Google Earth Client.....	23
Como inserir um mapa vetorial no Earth.....	27
Como inserir uma imagem no Earth.....	30
Outras opções da KML.....	35
<ScreenOverlay>	35
<Polygon>	37
Referências Bibliográficas.....	42

Usando o Google Earth para publicar Dados Proprietários

José Iguelmar Miranda

Introdução

Recentemente, o *site* de busca eletrônica Google lançou o Google Earth - GE. Trata-se de um sistema sofisticado para navegação em qualquer região do globo terrestre. Mosaicos de imagens de satélites foram juntados cobrindo todas as regiões da Terra. O mosaico para cobertura da Terra do GE foi construído com imagens Landsat com resolução espacial de 30m. Alguns lugares contam com recobrimento de imagens Ikonos, de 1m. Além das imagens de satélite, o GE utiliza também imagens topográficas, geradas pela missão Shuttle Radar Topography Mission - SRTM. Essa missão fez uma cobertura total da Terra, gerando imagens com informações sobre o relevo.

Usando as imagens SRTM e Landsat, o GE permite a visualização, em três dimensões, de relevos naturais de todos os pontos da Terra. Essa opção não funciona para edifícios e monumentos, a menos que a informação tenha sido inserida no banco de dados do GE. Além da navegação, o sistema permite o usuário traçar qualquer rota ligando diferentes pontos geográficos, como a trajetória entre um hotel e um centro de convenções, ou mesmo entre dois endereços residenciais. As aplicações da ferramenta ainda estão para serem exploradas, mas elas se ajustam a diversos campos do conhecimento, tanto comercial como científico.

O sistema precisa de uma plataforma Windows, pelo menos 2000, ou Mac OS x 10.3.9, para ser executado. Usuários Unix/Linux, por enquanto, ainda não podem usá-lo. Isso se deve ao fato de que grande parte de seu desenvolvimento, conforme documentação, usa controles ActiveX. Controles ActiveX pertencem à família de tecnologias Component Object Model - COM para Windows, da Microsoft. São componentes de *software* adicionais que podem ser incorporados a aplicações Windows, ou páginas *web*, permitindo maiores funcionalidades, como animações, vídeos, navegação em três dimensões etc., e escritos em uma linguagem de programação, como C++ ou Visual Basic.

A base de funcionamento do GE é uma variante da Geography Markup Language - GML, a Keyhole Markup Language - KML. A codificação das imagens e vetores vistos no aplicativo estão descritos nessa linguagem de codificação para compartilhar dados espaciais na *web*. Na realidade, a Google comprou a empresa Keyhole (www.keyhole.com), uma empresa especializada em mapeamento digital, de Mountain View, CA, e passou a adotar sua implementação da GML, chamada KML. A compra foi efetuada em 27 de outubro de 2004. Portanto, como a base do GE é a KML, vamos abordar nesse documento o estudo da GML, para entender a KML. Posteriormente, vamos explicar como usar a KML e a versão cliente grátis do GE para publicação de dados espaciais proprietários.

Geography Markup Language - GML

A linguagem Geography Markup Language - GML compartilha dado espacial (vetorial ou matricial) através da codificação do dado espacial em Extended Markup Language - XML. A GML é uma iniciativa do consórcio OpenGIS e tem por base a OpenGIS Abstract Specification - OAS (www.opengis.org). As iniciativas OpenGIS e GML oferecem um padrão, livre do vendedor, para codificar qualquer tipo de dado geográfico bem como qualquer método para mostrá-lo (Miranda, 2002). O objetivo da GML é o transporte e armazenamento da informação geográfica, incluídos a geometria e propriedades do atributo geográfico (Open Geospatial Consortium, 2005). A exemplo da XML, a GML e, conseqüentemente, a KML, **não** é uma linguagem para o desenho dos gráficos que ela codifica. Os procedimentos de visualização dos atributos geográficos fica a cargo de outras rotinas que devem ser desenvolvidas por quem implementa a GML.

A idéia é que qualquer representação digital do mundo real possa ser pensada como um conjunto de **atributos**, sendo o estado de um atributo definido por um conjunto de **propriedades**, onde cada propriedade pode ser imaginada como uma tripla {nome, tipo, valor}. A quantidade de propriedades que um atributo pode ter, juntamente com seus nomes e tipos, é determinada pelo **tipo do atributo**. Atributos geográficos são aqueles com propriedades cujos valores podem ter uma **geometria**. Um outro conceito importante é o de uma **coleção de atributos**, tendo por sua vez um tipo de atributo e conseqüentemente pode ter características próprias, em adição aos atributos que ele contém.

A GML segue o modelo geométrico definido pelas especificações acima. Assim, os elementos geométricos com dimensões 0, 1 e 2 em um sistema de referência espacial - spatial reference system - SRS, são representados por **pontos**, **linhas** e **polígonos**, respectivamente. Composições são permitidas, como **múltiplos pontos**

e **múltiplos polígonos**. Um requisito da GML é que exista um elemento no topo das descrições especificando o SRS no qual as medidas (lat, long ou graus, minutos, segundos, por exemplo) foram feitas.

Pode-se, por exemplo, querer criar uma representação digital de determinada reserva ambiental. Isso seria feito através de uma coleção de atributos onde atributos individuais representariam rios, estradas, tipos de solos, tipos de vegetação etc. A classificação dos fenômenos do mundo real determinariam os tipos de atributos necessários para serem definidos. Cada atributo teria suas especificidades. Por exemplo, "Rio" teria uma propriedade chamada 'nome', com valor do tipo 'string.' De maneira semelhante, "Estrada" poderia ter uma propriedade do tipo 'string' chamada 'classificação' e uma propriedade do tipo 'integer' chamada 'número.' Propriedades com tipos simples (integer¹, string, real, boolean) são chamadas de *propriedades simples*.

Os atributos requeridos para modelar o mundo real possuem *propriedades geométricas*, assim como propriedades simples. As propriedades geométricas precisam ser identificadas, de maneira que "Rio" poderia ter uma propriedade chamada 'centerLineOf'. Finalmente, o modelo digital para a reserva poderia ser expresso como uma coleção de atributos, que poderia ter um atributo do tipo 'ModeloReserva', interpretado como tendo uma propriedade 'string' chamada 'modelDate', fornecendo a data de seu estabelecimento e uma propriedade geométrica chamada 'boundedBy' fornecendo a extensão, ou limites geográficos, sob a qual o modelo é válido.

Os atributos simples são para descrever as entidades geográficas encontradas no mundo real. A codificação em GML não se preocupa em fazer a visualização gráfica dos atributos geográficos, como desenhar os mapas. Para desenhar um mapa com a GML é necessário transformar a GML em um formato gráfico, ou através de um desenho direto ou transformando a codificação XML em Scalable Vector Graphics - SVG, Vector Markup Language - VML ou Virtual Reality Markup Language - VRML. Essa transformação pode ser feita em qualquer ponto na cadeia de processamento entre o armazenamento do dado e o dispositivo de visualização.

A arquitetura de armazenamento e o transporte de dados da GML devem ficar a cargo de quem implementar esse padrão. Um simples arquivo GML poderia representar vários mapas. Um ponto importante é que as coordenadas dos pontos em uma estrutura GML são especificados em relação a um Spatial Reference System - SRS cuja descrição pode estar também codificada em GML.

A geometria para descrever os diversos atributos suportados pela GML deve seguir um padrão, fornecido em XML. O modelo de atributos simples definidos pelo Open Geospatial Consortium - OGC para a GML define as seguintes geometria de classes:

¹ Não se traduz primitivas usadas na GML e KML.

- Point
- LineString
- LinearRing
- Polygon
- MultiPoint
- MultiLineString
- MultiPolygon
- GeometryCollection

Adicionalmente, deve-se fornecer informações para codificar as coordenadas e um elemento Box para definir os limites ou extensão geográfica da área de estudo. Coordenadas são simplesmente uma lista de tuplas de coordenadas. Informações adicionais devem ser fornecidas indicando dois tipos de separadores: 1) entre as coordenadas (x,y) e 2) entre pares de coordenadas, (x_1,y_1) e (x_2,y_2) . No exemplo a seguir, essas informações foram definidas como: 1) as coordenadas são separadas por vírgulas e 2) pares de coordenadas são separados por espaço branco. A separação decimal para os valores das coordenadas é o ponto. Exemplo da especificação com a GML:

- Codificando um conjunto de coordenadas:

```
<coordinates decimal = "." cs = "," ts = "whitespace" >
  1.03,2.167 4.16,2.5 4.87,3.0 10.00,4.5
</coordinates >
```

coordinates indica ao interpretador GML um conjunto de coordenadas. As coordenadas devem ser lidas conforme os parâmetros: *decimal*, o separador de casas decimais. Nesse caso, o ponto; *cs* - coordinate separator, o separador do par x,y . Nesse caso, a vírgula; *ts* - tab spacing, o separador dos pares de coordenadas. Nesse caso, um espaço branco. Os valores *decimal*, *cs* e *ts* podem mudar, como vírgula, ponto-e-vírgula e espaço, respectivamente.

O elemento Point é usado para codificar instâncias da classe geométrica ponto. Cada elemento Point possui apenas um par de coordenadas. A geometria do ponto deve especificar o SRS no qual a coordenada é medida, chamado de srsName. Notar que este parâmetro torna-se opcional se o ponto estiver contido em outro elemento que já tenha o SRS definido. A seguir, exemplo da codificação do ponto.

- Codificando um ponto:

```
<point srsName = "SC:33" >
  <coordinates >
    44.6,50.8
  </coordinates >
</Point >
```

srs (spatial reference system) é o sistema de referência espacial no qual as coordenadas devem ser calculadas. No caso, elas devem estar dentro de um retângulo com coordenadas previamente definidas, tendo por nome (srsName) SC:33.

Como foi dito, todos elementos geográficos definidos devem estar contidos em uma região geográfica com especificação da sua extensão. Na GML isto é feito através do elemento Box, que contém duas tuplas de coordenadas. A primeira especifica os valores mínimos e a segunda contém os valores máximos. Esse elemento obrigatoriamente deve conter um srsName, uma vez que ele não pode estar contido em outro elemento geométrico. O exemplo a seguir mostra a definição:

- Codificando um Box:

```
<Box srsName = "EPSG:4326" >  
  <coordinates>  
    0.0,0.0 144.6,150.8  
  </coordinates>  
</Box>
```

Embora não seja objetivo fazer uma análise detalhada da GML, analisa-se os três perfis usados para definir a GML. Conforme especificação dos idealizadores da GML, ela foi feita para poder ser adotada por uma grande classe de usuários, utilizando uma variedade de tecnologias XML. Os três perfis são:

- Perfil 1: para os implementadores que desejam usar uma solução com base em Document Type Definition - DTD puro e não estão preparados para desenvolver aplicações DTD específicas. Esse perfil requer o uso de GML Feature e GML Geometry em DTD.
- Perfil 2: para os implementadores que desejam usar uma solução DTD puro e estão preparados para desenvolver aplicações DTD específicas. Esse perfil requer o usuário criar uma aplicação específica do GML Feature que usa a GML Geometry em DTD.
- Perfil 3: para aqueles preparados para fazer uso do Modelo e Sintaxe do W3C Resource Description Format - RDF e Esquema RDF. Essa abordagem requer um controle da especificação geoespacial.

No Perfil 1, existem três principais DTDs:

- DTD para GML Feature
- DTD para GML Geometry
- DTD para Spatial Reference System

Usando essas DTDs, codifica-se uma grande variedade de informações geoespaciais. Um atributo geográfico nesse perfil é representado por uma tag <Feature> que engloba nenhum ou várias propriedades geométricas simples. Uma propriedade simples é qualquer propriedade que pode ser codificada usando um conjunto de caracteres passíveis de uma análise sintática. O Perfil 1 admite apenas boolean, integer, real e string. Como exemplo desse perfil, mostra-se a codificação de uma estrada:

- Codificando uma estrada:

```
<feature typeName = "Estrada" >
  <description >
    PI
  </description >
  <property typeName = "classifica" >
    estrada_estadual
  </property >
  <property typeName = "number" type = "integer" >
    11
  </property >
  <geometricProperty typeName = "linearGeometry" >
    <lineString srsName = "SC:33" >
      <coordinates >
        0.0,10.0 10.0,30.0 25.0,30.0 8.7,50
      </coordinates >
    </LineString >
  </geometricProperty >
</Feature >
```

O Perfil 2 usa os mesmos DTD para GML Geometry e Spatial Reference System do Perfil 1. Feature DTD específicas incrementam esse perfil:

- Feature DTD específicas da aplicação
- DTD para GML Geometry
- DTD para GML Spatial Reference System

Diferente do Perfil 1 da GML, o Perfil 2 não possui uma DTD para GML Feature específica. O usuário pode construir sua própria aplicação de Feature GML seguindo os padrões adotados pela especificação GML. A DTD para GML Geometry fornece ao usuário um conjunto predefinido de propriedades geométricas para que ele possa usar e assim descrever atributos geográficos, incluindo-os em suas aplicações específicas das DTD Feature. Essas propriedades geométricas incluem propriedades comuns de entidades geográficas tais como localização e extensão. Adicionalmente, o Perfil 2 da GML fornece ao usuário alguns metadados básicos para descrever atributos geográficos, incluindo nome e descrição.

O Perfil 2 da GML introduz nomes descritivos para as primitivas geométricas, já apresentadas no Perfil 1, por exemplo:

- PointProperty: location, position, centerOf
- LineStringProperty: centerLineOf, edgeOf
- PolygonProperty: extendOf, coverage

A semântica precisa das propriedades geométricas não é ainda parte da especificação GML, mas era esperado que ela fosse definida em versões subseqüentes. Por exemplo, quando da especificação da GML não existia semântica para uma questão como "O que significa a extensão de um polígono"?

A codificação de atributos geográficos no Perfil 2 permite ao usuário construir uma aplicação com uma DTD Feature específica. Na documentação da GML (Open Geospatial Consortium, 2005) encontram-se as regras que governam a especificação da DTD. Para efeito de ilustração, descreve-se a seguir a especificação XML para o atributo "Estrada" mostrado anteriormente com o Perfil 1. Compare as duas descrições e note a diferença:

- Codificando uma estrada:

```
<estrada>
  <description>
    PI
  </description>
  <classification>
    estrada_estadual
  </classification>
  <number>
    11
  </number>
  <linearGeometry>
    <lineString srsName = "SC:33" >
      <coordinates>
        0.0,10.0 10.0,30.0 25.0,30.0 8.7,50
      </coordinates>
    </lineString>
  </linearGeometry >
</Estrada>
```

Percebe-se que esta descrição é mais direta e inteligível para o usuário. Existe uma construção que facilita a abstração dos elementos geográficos usando "Features" mais próximas do entendimento comum. De qualquer maneira, há uma

preocupação em construir os elementos do Perfil 2 usando como base os elementos do Perfil 1. Por exemplo, os elementos `lineString` e `coordinates` são exatamente os já definidos no Perfil 1.

A definição do Perfil 3 foi decorrência da necessidade de se entender o significado dos dados na informação geoespacial. Por exemplo, um plano de informação em um determinado Sistema de Informações Geográficas - SIG pode ser usado para armazenar a rede de drenagem. Em outro sistema, a rede de drenagem poderia conter apenas informações limnológicas, armazenando apenas códigos numéricos sobre a qualidade da água. Nesse caso, a migração da rede de drenagem entre os sistemas seria problemático porque o significado dos dados não é parte intrínseca dos dados, mas em termos de um conjunto de convenções. Nesse caso, a migração dos dados, se necessária, deveria estar acompanhada de um grande esforço manual. Tais fatores foram motivadores no desenvolvimento da GML.

Especificamente para esse tipo de problema, a GML foi construída usando o W3C Resource Description Format - RDF. Essa alternativa foi responsável pelo Perfil 3 para codificar informação geoespacial usando o RDF, e um conjunto formal de definições, usando o RDF Schema, para a GML. A Fig. 1 mostra a relação entre os três perfis, DTD e RDF para a GML.

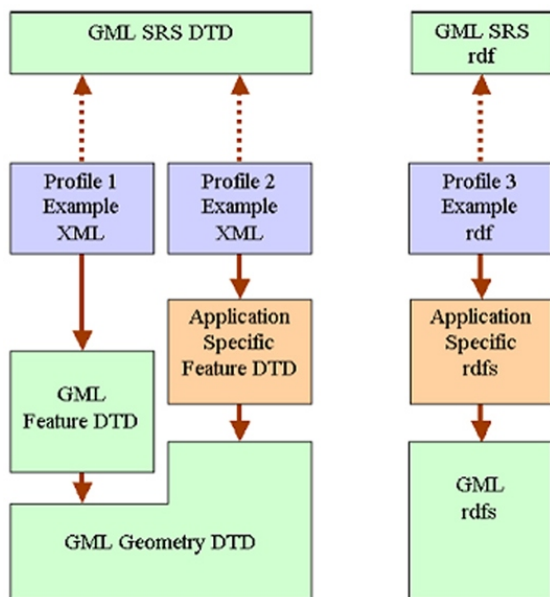


Fig. 1. Os três perfis da GML e seu relacionamento com DTD e RDF Schema.

Fonte: Open Geospatial Consortium (2005).

A codificação de classes geométricas com o RDF Schema para a GML é inteiramente consistente com a Geometry DTD da GML para os Perfis 1 e 2. Então, quando se faz uma codificação no nível do Perfil 3, apenas o RDF Schema é usado, em vez da DTD (Fig. 1). Mesmo assim, seções da DTD com a GML Geometry podem ser geradas mecanicamente a partir das definições do RDF Schema para as GML Geometry Classes. Uma Geometry Class é uma classe abstrata da qual são construídas as subclasses geométricas. Todas as geometrias têm um Spatial Reference System, identificado por um nome. O RDF Schema para a classe Geometry é:

```
<rdfs:Class rdf:ID = "Geometry" >
  <rdfs:comment >
    Geometry é uma classe abstrata raiz da hierarquia. Portanto, ela não é instanciável.
  </rdfs:comment >
</rdfs:Class >
<rdf:Property ID = "srsName" >
  <rdfs:domain rdf:resource = "#Geometry"/>
  <rdfs:range rdf:resource = "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal"/>
</rdf:Property >
```

Não se mostra aqui a definição para todos os elementos geométricos básicos usados no Perfil 3. Mas enfatiza-se que o Perfil 3 conta com a habilidade de adicionar novos atributos e tipos geográficos de uma maneira mais "legível" e mais formal do que o Perfil 1 e o Perfil 2. Para efeito de comparação mostra-se a codificação de atributos geográficos, como foi realizado para os Perfis 1 e 2, só que nesse caso considera-se um projeto com mais planos de informação que os exemplos anteriores, quando lida-se apenas com "Estrada:"

```
<?xml version = "1.0" encoding = "UTF-8"? >
<rdf:RDF xml:lang = "en"
  xmlns:cerrado = "http://www.xyz.org/cerrado/example_profile3_schema.rdf#"
  xmlns:gml = "http://www.opengis.org/gml/gml.rdf#"
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs = "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#" >
  <cerrado:ModeloTemas >
    <gml:boundedBy >
      <gml:Box srsName = "SC:33" >
        <gml:coordinates >
          0.0,0.0 187.56,158.98
        </gml:coordinates >
      </gml:Box >
    </gml:boundedBy >
    <cerrado:modeloData >
      outubro 2005
    </cerrado:modeloData >
  </cerrado:modeloMembro >
```

```

<cerrado:Rio >
  <gml:name >
    rio das Onças
  </gml:name >
  <gml:description >
    Rio que corta a área de cerrado em estudo.
  </gml:description >
  <gml:centerLineOf >
    <gml:LineString srsName = "SC = 33" >
      <gml:coordinates >
        0.0,50.0 33.8,55.7
      </gml:coordinates >
    </gml:LineString >
  </gml:centerLineOf >
</cerrado:Rio >
</cerrado:modeloMembro >
<cerrado:modeloMembro >
  <cerrado:Estrada >
    <gml:description >
      PI
    </gml:description >
  <cerrado:classificação >
    Estrada municipal
  </cerrado:classificação >
  <cerrado:numero >
    11
  </cerrado:numero >
  <cerrado:GeometriaLinear >
    <gml:LineString srsName = "SC = 33" >
      <gml:coordinates >
        10.0,35.98 77.8,49.7
      </gml:coordinates >
    </gml:LineString >
  </cerrado:GeometriaLinear >
</cerrado:Estrada >
</cerrado:modeloMembro >
</cerrado:ModeloTemas >
</rdf:RDF >

```

A especificação XML acima descreveu dois temas, Rio e Estrada, para o atributo "cerrado." Esse é um exemplo hipotético, mas na prática, é assim que a GML funciona. Observar que algumas especificações XML são antecedidas por "cerrado", e os nomes podem ser em português. No cabeçalho, existe uma URL indicando onde encontrar os "namespaces" (espécie de dicionário de variáveis) usados na descrição. Da mesma forma deve ser fornecido a URL para a gml. A tag do RDF é importante, pois sem ela e os prefixos que aparecem na descrição codificariam essa FeatureCollection (coleção de atributo) de maneira semelhante ao Perfil 2 da GML. Concluindo, o exemplo mostra quatro "namespaces:"

- rdf: descrição dos Resource Description Format do W3C
- rdfs: RDF Schema do W3C
- gml: definição do RDF Schema da Geography Markup Language do OGC
- cerrado: definição do RDF Schema específico para o atributo cerrado, a partir de uma organização hipotética xyz.org.

Existe ainda uma série de especificações da GML não coberta neste documento. Mas esse material só é de interesse para quem pretende fazer uma implementação da GML. Como este não é o caso na maioria das vezes, o material aqui apresentado serve apenas como um esclarecimento de como essa especificação XML funciona para entender a Keyhole Markup Language, essa sim, de interesse por permitir a interação com o GE. Para o leitor interessado em uma publicação mais atual sobre a GML, consultar o documento [GML_05-033r9_profile.pdf](#). Esse documento é datado de 05 de julho de 2005, gerado pelo consórcio OGC, com informações sobre o perfil específico de implementação da GML.

Keyhole Markup Language - KML

A KML é uma gramática e formato de arquivo XML para modelar e armazenar atributos geográficos, como pontos, linhas, imagens e polígonos para serem desenhados no *software* cliente do GE. Um arquivo KML é processado pelo GE numa maneira semelhante como os arquivos HTML são processados por um navegador *web*. A KML usou o padrão GML para sua implementação. Você poderá observar adiante a semelhança entre as implementações, embora existam diferenças sintáticas, a semântica é a mesma. A linguagem foi criada originalmente pela empresa Keyhole Inc. Images, situada na cidade de Mountain View, na Califórnia. No ano de 2004 a empresa foi comprada pela Google, que depois da sua aquisição, lançou o produto de maior impacto no mercado de mapas interativo pela *web*: o Google Earth.

A Fig. 2 é um esquema ilustrativo de como deve acontecer a implementação da GML. O processo não é simples. Toda a responsabilidade de análise sintática e semântica (GML Parser) fica a cargo do implementador. Observar na figura que as definições da GML, como proposto pelo OGC, é o ponto mais importante, mostrado na caixa superior da figura. O processo funciona da seguinte maneira: um cliente (navegador *web*) faz uma solicitação ao servidor de mapas *web*, com padrão Web Feature Server - WFS, também um padrão OGC. A solicitação é recebida e passada para a função GetFeature. Essa se responsabiliza por consultar as definições da GML (GML Document), que por sua vez dispara todo o processo para se certificar que a solicitação atende ao estipulado pelo OGC.

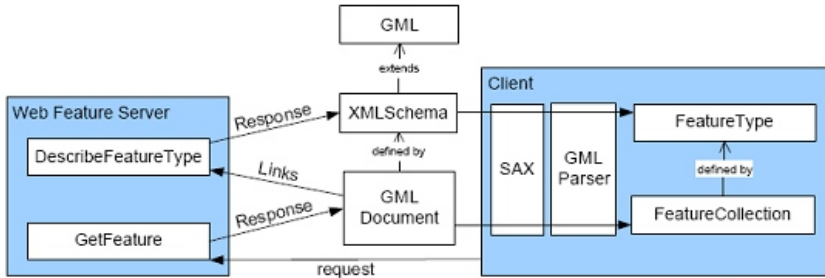


Fig. 2. Modelo de análise sintática para implementar a GML.

Fonte: GeoTools (2005).

Se todo o processo de consulta for aprovado, o servidor de mapas *web* envia uma resposta ao cliente, atendendo sua solicitação, que no caso seria o desenho de um mapa ou imagem no cliente. Em caso de erro, o cliente será notificado e a consulta se encerra. Veja que do lado cliente, descreveu-se anteriormente uma *FeatureCollection* e uma *FeatureType*. Além do GML Parser, o cliente conta com um Simple API for XML - SAX. Trata-se de uma interface padronizada, orientada por eventos, para processar a interação de aplicações com documentos XML. SAX é uma interface comum implementada em Java para diferentes analisadores (*parsers*) de XML, assim como o Java Database Connectivity - JDBC é uma interface comum implementada em Java para diferentes bancos de dados relacionais.

Portanto, os técnicos da Keyhole fizeram uma implementação da GML sob este aspecto, só que chamaram sua implementação de KML. Algumas das facilidades oferecidas pela KML para lidar com mapas na *web* incluem:

- definição de ícones e rótulos para identificar qualquer local na superfície do planeta;
- uso de câmeras virtuais que podem focalizar, em diferentes ângulos de visão, diferentes pontos na Terra;
- superposição de imagens do usuário com as imagens do globo disponível no GE;
- carga dinâmica de arquivos KML a partir de localizações remotas ou local;
- leitura de arquivos no formato KMZ (arquivos KML compactados). O benefício é que imagens e mapas podem ficar armazenadas no computador local, sem necessidade de tráfego pela rede;
- a tag <description> permite a inserção de várias facilidades do HTML, como âncoras, imagens e tabelas etc.

Mais sobre essa linguagem será apresentado adiante, quando se explica como abrir arquivos KML próprios para serem lidos no cliente GE.

Google Earth

O Google Earth é um sistema para navegação geográfica, em duas e três dimensões. Seu antecessor é o Google Maps (<http://maps.google.com>). Esse é um sistema que também permite navegação geográfica, mas a pesquisa é apresentada em mapas estáticos na *web*. O usuário digita um local na caixa de texto "Search" e o Google Maps apresenta o mapa da região requisitada. A navegação é permitida com o uso de setas de orientação que aparecem no canto superior esquerdo do mapa. Além da navegação, é permitido aumentar ou diminuir a escala, usando uma régua graduada logo abaixo da orientação.

No canto superior direito do mapa aparecem três opções: "Map", "Satellite" e "Hybrid". Os nomes são auto-explicativos. Quando usando a opção "Hybrid", os nomes dos logradouros aparecem por cima da imagem de satélite. As opções "Map" e "Hybrid" não estão disponíveis para todas as cidades do mundo. O quadro informativo do Maps é completado com a apresentação de uma escala, em milhas e quilômetros, no canto inferior esquerdo (Fig. 3). As imagens de satélite disponíveis para o GE são as mesmas que para o Maps.

Na Fig. 3, aparecem os elementos básicos do *software*. A visualização mostra a Embrapa Informática Agropecuária, ao lado da Faculdade de Engenharia Agrícola, dentro do campus da Unicamp, em Campinas, SP. Esta é uma imagem de alta resolução espacial, não disponível para todas regiões do globo. O Google não especifica o satélite, mas pode-se assumir uma imagem Ikonos multiespectral, com resolução de 4m. As datas de passagem das imagens também não são informadas, mas a empresa procurou colocar imagens mais recentes com menor cobertura de nuvens. Algumas regiões, como o nordeste brasileiro, ficam prejudicadas por uma intensa cobertura de nuvens.

A navegação no Maps é apenas em duas dimensões e é feita em grandes incrementos de deslocamento nas imagens. O usuário não tem flexibilidade de fazer sobrevôos com pequenos incrementos. A única vantagem do Maps em relação ao GE é que ele pode ser acessado de qualquer plataforma computacional, pois o cliente é sempre um navegador *web*.



Fig. 3. Tela do Google Maps.

O GE está disponível em quatro versões: Google Earth Client, Google Earth Plus, Google Earth Pro e Google Earth Enterprise. A primeira versão é grátis. Permite apenas navegação, 2D e 3D, nas imagens. A segunda requer uma subscrição anual de US\$20,00. A vantagem é a liberação de um mecanismo para inserir dados de GPS, planilhas eletrônicas com informações cadastrais, desenho de figuras geométricas, como linhas e polígonos, além de permitir a inserção de objetos em 3D. Outra vantagem é a impressão, de forma legível, de partes da imagem em uma impressora.

A versão Pro custa US\$400,00 anuais. Além das vantagens apresentadas anteriormente, essa versão permite a importação de dados geográficos em forma de mapas, a inserção de até 2.500 localizações, por endereço ou latitude e longitude, compartilhamento de dados. A Google oferece mais opcionais para essa versão, com custos adicionais. Por exemplo, para o usuário incorporar dados de um sistema geográfico de informações nos formatos .shp e .tab, é necessário pagar mais US\$200,00 pelo módulo "GIS Data Importing Module". Existem outras opções, todas custando o mesmo valor.

E por fim, a versão Enterprise, opção direcionada para empresas e sem valores especificados no *site*. Essa versão libera todo o potencial do GE para o usuário, permitindo a combinação de dados da empresa com os dados do GE ou simplesmente usando os dados só da empresa. A versão é composta de três

elementos: Google Earth Fusion, com o objetivo de integrar os dados da empresa, como imagens de satélite, modelos numéricos de terreno e dados vetoriais; Google Earth Server, com o objetivo de gerenciar os dados da empresa para *softwares* clientes, o Google Earth Client. E o próprio Google Earth Client, para visualização, impressão e gerência dos dados.

Google Earth Client

Esse documento se atém apenas a versão grátis do GE. Mesmo assim, será mostrado que é possível usar a interface do GE, com todos seus recursos, para publicar dados próprios. A única restrição é que apenas as plataformas Windows 2000 ou superior e Mac OS x 10.3.6 podem acessar os dados a serem publicados.

Para obter uma cópia do GE cliente, é suficiente acessar o endereço eletrônico do Google, escolher "more" e depois escolher a opção "Earth." Todas as orientações de instalação estão disponíveis na página, portanto, evita-se fornecer detalhes. Parte-se do pressuposto que o usuário instalou uma versão cliente do GE.

Um primeiro ponto importante para colocar dados próprios no GE é observar o sistema de projeção e o datum usados. O sistema de projeção padrão é o latitude-longitude e o datum escolhido é o World Geodetic System 84 - WGS84. O GE usa uma projeção cilíndrica simples, onde os meridianos e paralelos são eqüidistantes, dispostos em linhas retas paralelas e se cruzando em ângulos retos. Se seus dados estiverem armazenados no SIG com outra projeção e datum, eles devem ser transferidos, senão haverá um deslocamento de imagens e mapas ao se sobrepor no GE.

Todos os detalhes sobre a implementação da KML podem ser encontrados no documento "Google Earth KML Documentation", disponível em <http://www.keyhole.com/kml>. Não seria produtivo repetir todo o documento aqui. Nosso objetivo é enfatizar o uso de muitas de suas opções para disponibilizar uma aplicação no GE.

A aplicação usa a versão cliente do GE como um servidor de mapas na *web*. O procedimento padrão é criar um arquivo com a estrutura dos dados (mapas e imagens) que se quer divulgar no GE, usando a KML, e salvá-lo em um arquivo separado. A criação do arquivo pode ser feita em processador de texto simples, como o Bloco de Notas, ou em um ambiente integrado para XML. Inicia-se com uma aplicação simples. Ela apenas mostra a localização da Embrapa Informática Agropecuária no GE.

Passo 1: criar o arquivo Embrapa.kml

Todo arquivo para ser lido no GE deve ter uma extensão .kml ou .kmz. Essa última extensão significa que o arquivo .kml se encontra na forma compactada. Abrir qualquer processador de texto e inserir o texto (adiante detalha-se as opções usadas) e salvá-lo como Embrapa.kml:

```
<kml xmlns="http://earth.google.com/kml/2.0">
<Folder>
  <open>0</open>
  <name>
    <![CDATA[Embrapa Inform&acute;tica Agropecu&acute;ria]]>
  </name>
  <description>
    <![CDATA[<a href="http://www.embrapa.br/">Busca na Embrapa</a>]]>
  </description>
  <Placemark>
    <Name>Embrapa</name>
    <description>
      <![CDATA[Embrapa Inform&acute;tica Agropecu&acute;ria]]>
    </description>
    <LookAt>
      <longitude>-47.0619</longitude>
      <latitude>-22.8178</latitude>
      <range>540.68</range>
      <tilt>0</tilt>
      <heading>3</heading>
    </LookAt>
    <Point>
      <coordinates>-47.0619,-22.8178</coordinates>
    </Point>
  </Placemark>
</Folder>
</kml>
```

Passo 2: iniciar o Earth

Iniciar o GE e escolher a opção "File -> Open" ou <CTRL>O. Em seguida, o GE abre uma janela para se escolher o arquivo, apresentando a lista de opções com as extensões válidas. Localize o diretório onde você salvou o arquivo Embrapa.kml e o abra. O GE lê seu arquivo, faz a verificação sintática e abre o arquivo, criando na janela "Places" um diretório "Temporary Places". É nesse diretório que ele guarda as suas descrições KML para seus arquivos de dados espaciais. No exemplo, o GE vai se deslocar de sua opção "default" para as coordenadas que localizam a Embrapa Informática Agropecuária. Isto é tudo.

Analisando o arquivo do exemplo:

- `<kml xmlns = "http://earth.google.com/kml/2.0" >`
Essa linha de comando deve ser a primeira do arquivo e é obrigatória. Ela indica onde a KML encontra suas definições, no caso, no endereço eletrônico fornecido, no próprio Google. Quando falou-se sobre a GML, a primeira linha era uma referencia ao XML. Mas isto era apenas um indicativo de como deveria funcionar o esquema da GML. Os técnicos da Keyhole optaram por ter a especificação da KML, embora o arquivo possa ter como primeira linha uma especificação XML, não obrigatória.
- `<Folder> </Folder>`
O `<folder>` funciona como uma estrutura de organização de arquivos num diretório. Sob um `<folder>`, organiza-se `<placemark>`'s, mapas, imagens e mesmo outros `<folder>`'s. Todos os atributos debaixo de um `<folder>` formam uma coleção estruturada de atributos geográficos e permite uma visão unificada dessa coleção.
- `<open>0</open>`
Essa opção funciona como uma chave lógica que controla a disposição inicial do `<folder>`, assumindo os valores 0 (`<folder>` condensado) ou 1 (`<folder>` expandido).
- `<name> </name>`
Essa tag da KML serve para dar nome ao `<folder>`. E não somente ao `<folder>`, mas a `<placemark>`, `overlay` ou qualquer outro atributo usado no GE. Sobre o nome que aparece no `<folder>`, veja explicação a seguir.
- `<![CDATA[Embrapa Informática Agropecuária]]>`
Como se usa caracteres especiais acentuados, o elemento CDATA serve para o propósito. Ele indica ao analisador de sintaxe do XML para ignorar os caracteres especiais usados dentro dos colchetes. A forma genérica é: `<![CDATA[caractere especial]]>`.
- `<description> </description>`
Uma tag simples para fornecer informações descritivas acerca do elemento. A descrição aparece na janela "Places" do GE quando o usuário clicar no elemento, por exemplo, um `<folder>`. Se o elemento for um `<placemark>` e o usuário clicar sobre ele, aparece uma janela "pop-up" com a descrição dada. Experimente clicar sobre osímbolo do alfinete que aparece na janela de visualização de imagens e veja o que acontece.
O elemento de descrição suporta texto simples ou formato HTML. Um texto descrevendo uma URL válida é automaticamente convertida para um hiperlink. Quando se pretende usar um texto e uma URL, ou quando incluindo imagens no HTML, o elemento CDATA deve ser usado obrigatoriamente. Esse foi o caso no exemplo: `<![CDATA[Busca na Embrapa]]>`

- `<Placemark>` `</placemark>`

Essa é uma tag complexa, isto é, ela pode conter outras tags para sua definição completa. Ela é usada para descrever uma primitiva geométrica na superfície da Terra. Ela representa a descrição de um ponto na superfície do planeta através de um par de coordenadas. As coordenadas são definidas usando o padrão [longitude, latitude, altitude], sendo que: -180 < longitude < 180 e -90 < latitude < 90. A altitude é fornecida em metros tomando como referência a superfície do esferóide que representa a Terra (datum). Como já dito, o datum usado é o WGS84. Depressões são mostradas na superfície. Outras tags são usadas para a definição completa do `<placemark>`, conforme aparece no exemplo:

```
<name> Embrapa</name>
<LookAt>
  <longitude>-47.0619</longitude>
  <latitude>-22.8178</latitude>
  <range>1000.0</range>
  <tilt>0</tilt>
  <heading>3</heading>
</LookAt>
<Point>
  <coordinates>-47.0619,-22.8178</coordinates>
</Point>
```

A tag `<name>` já foi descrita. O elemento `<LookAt>` especifica a posição da câmera colocada sobre determinado ponto na superfície do planeta. Pode-se definir a posição da câmera não somente para um `<placemark>`, mas também para `<folder>`'s, overlays e outros atributos. Quando um atributo tiver uma tag `<LookAt>`, clicando duplamente sobre o item na janela "Places" determinará ao GE voar suavemente ao local especificado pela câmera. Considerar que as coordenadas da câmera não são necessariamente as coordenadas do atributo geográfico. Pode-se colocar uma câmera, por exemplo, no canto de uma área que represente um grande polígono. No caso do exemplo, coloca-se a câmera no estacionamento da Embrapa Informática Agropecuária. Geralmente, usa-se o bom senso e se posiciona a câmera em um ponto central, ou próximo do centro, do elemento geográfico de interesse.

As tags `<longitude>` e `<latitude>` não oferecem dificuldade de entendimento e foram descritas anteriormente. Elas estão indicando o ponto na superfície onde deve ser colocada para localizar a Embrapa Informática Agropecuária. A tag `<range>` é uma tag filha da `<LookAt>`. Ela determina a altitude, em metros, da superfície onde a câmera deve aparecer. No exemplo, escolheu-se uma altitude de 1.000m. `<tilt>` é uma tag filha de `<LookAt>`, e serve para indicar o ângulo de visão do ponto designado. O valor 0 indica que não existe tilt (perspectiva), portanto, a visão acontece de cima do ponto especificado. E este é o caso do exemplo.

A tag `<heading>` é simples, filha de `<LookAt>` e descreve a distância angular ao longo do horizonte ao ponto de visada de onde colocou-se a câmera. A distância é medida a partir do norte. Como se especificou um valor 3 para o exemplo, isto significa retirar texto uma distância angular de apenas 3°, na direção leste. Valores negativos indicam direção oeste. Essa foi a última tag usada para definir `<LookAt>`. A próxima tag, `<point>` se aplica ao elemento `<placemark>`.

`<point>` é também uma tag complexa. Ela define a localização das coordenadas de um ponto sobre o mapa. A tag filha é `<coordinates>`, conforme mostra o exemplo. Nesse caso, `<point>` está indicando a posição geográfica do `<placemark>`. No caso, as coordenadas do `<placemark>` são as mesmas do ponto de visão da câmera, definida com `<LookAt>`.

Finalizando a definição do arquivo KML, fechou-se as tags `<Folder>` e `<kml>`. Como todo arquivo XML, deve-se ter o cuidado de "fechar" todas as tags abertas.

Como inserir um mapa vetorial no Earth

O GE permite a inclusão de vários atributos geométricos. Listou-se esses elementos, sem se deter em cada um. Como um exemplo geral, insere-se dados vetoriais representando estradas. Os outros elementos geográficos são semelhantemente definidos. Pode-se georreferenciar pontos (`<point>`), linhas e polilinhas (`<linearRing>`, `<lineString>`), polígonos (`<polygon>`), coleção de linhas e polígonos, coordenadas (`<coordinate>`) e elementos em 3D (`<extrude>`). Esses elementos contam com estilos, permitindo a definição de cores, por exemplo.

Como exemplo de uma linha representando uma estrada, usa-se o código a seguir:

```
<Folder>
  <name>PI-210</name>
  <description>Rodovia Estadual</description>
  <visibility>0</visibility>
  <open>0</open>
  <Placemark>
    <visibility>0</visibility>
    <open>0</open>
    <Style>
      <LineStyle>
        <color>ff00ffff</color>
      <LineStyle>
    <Style>
    <LineString>
      <Coordinates>
```

```

-41.75048432,-2.93169962 -41.70845712,-2.96777836
-41.70716397,-2.96894219 -41.70677603,-2.96958876
-41.70651740,-2.97088191 -41.70690535,-2.97346820
-41.70742260,-2.97915803 -41.70418974,-2.99894315
-41.70393111,-3.00075355 -41.65880036,-3.01627129
-41.64780863,-3.02477372 -41.62912269,-3.03961255
-41.62627777,-3.04168158 -41.62388545,-3.04397692
-41.62246299,-3.04552869 -41.61961807,-3.04630458
-41.60332445,-3.06337409 -41.57177172,-3.07320199
-41.55315043,-3.08005565 -41.53814995,-3.09350436
-41.50472216,-3.09945283 -41.49153209,-3.09699585
-41.47653161,-3.10178049 -41.46140181,-3.10187747
-41.46023798,-3.10161884 -41.45881552,-3.10006707
-41.44510819,-3.08131647 -41.44329779,-3.08079921
-41.44187533,-3.07963538 -41.43954767,-3.07937675
-41.43502166,-3.07873018 -41.43088360,-3.07795429
-41.42428856,-3.07536800 -41.37973972,-3.05816918
-41.37973972,-3.05816918
</coordinates>
</LineString>
</Placemark>
</Folder>

```

O exemplo mostra a definição do arquivo KML para desenhar a estrada estadual PI-210. O estilo da linha é definido pelo segmento de código:

```

<Style>
  <LineStyle>
    <color>ff00ffff</color>
  </LineStyle>
</Style>

```

A cor da linha é definida pela tag simples `<color>` e representa uma cor com 32 bits em formato hexadecimal, usando a notação AABBGRRR, ou AA - alfa, indicando o valor de transparência; BB - azul; GG - verde e RR - vermelho. O intervalo de variação de cores vai de 00000000 a ffffffff. No caso, tem-se uma linha da cor amarela. A dupla `<LineString>` e `<coordinates>` definem o vetor que representa a estrada estadual. Cada par de coordenadas está separado por vírgulas, enquanto um espaço branco separa os pares de coordenadas. Coloca-se duas coordenadas por linha, mas a escolha é livre, desde que dois princípios básicos sejam observados: a vírgula para separar as coordenadas e um espaço branco para separar os pares de coordenadas.

Como desenhar vários segmentos de linhas, ou como desenhar estradas federais e estaduais separadamente? Usando apenas a tag `<LineString>` não resolve o problema. A KML junta o final de uma linha com outra, seja qual for a distância entre elas. A solução está no uso da tag `<MultiGeometry>`. Veja como ela foi usada no exemplo a seguir:

```
<MultiGeometry>
<LineString id = "khLineString553" >
<tessellate> 1 </tessellate>
  <coordinates>
    -41.84250585,-3.19347652 -41.84858548,-3.18549700
    -41.84904145,-3.18488904 -41.84942142,-3.18458506
    etc
    -41.77950570,-3.12318082 -41.77958170,-3.11998901
    -41.77958170,-3.10532191 -41.77958170,-3.10524592
    -41.77996168,-3.07803958 -41.75594715,-2.94671962
  </coordinates>
</LineString>
</MultiGeometry>
```

Esse exemplo mostra outra tag interessante da KML, `<tessellate>`, que permite um atributo `LineString` contornar acidente de terreno. Ela é uma tag simples, filha de `<LineString>`. Por exemplo, se a estrada passar por um acidente geográfico acentuado, como um alto relevo, a linha é desenhada com um sobressalto. Usa-se também no exemplo um padrão de linha da Keyhole, conforme especificado no comando `<LineString id = "khLineString553" >`. O "id" identifica um tipo de linha com formato predefinido. Para usá-lo, basta uma chamada.

Como colocar arquivos vetoriais, independentes de formato, no GE? Você deve ter observado nos exemplos anteriores que o arquivo vetorial foi repassado para o KML com as próprias coordenadas dos vetores a serem desenhados. Para o KML, um arquivo vetorial funciona assim, um conjunto de dados no formato ASCII, obedecendo a um formato padrão de preenchimento. Não é difícil de migrar dados vetoriais de formatos proprietários do SIG em uso para o formato ASCII. Uma vez no formato ASCII, insere-se esses dados no KML entre as tags `<coordinate>` e `</coordinate>`. E se o arquivo resultante for muito grande, gerando também um arquivo KML muito grande? Nesse caso, é possível compactar o arquivo KML com o WinZip, por exemplo, e renomear sua extensão para .kmz.

Usuários do ArcGIS, um produto da Environmental System Research Institute (ESRI), contam com uma função que gera arquivo kml diretamente do seu formato proprietário "shapefile", arquivos com a extensão .shp. Para isso, a ESRI desenvolveu recentemente a função Arc2Earth, que migra os dados do formato shapefile para o GE, kml.

Como inserir uma imagem no Earth

Os dois principais tipos de dados em qualquer SIG são vetorial e matricial. Se há uma maneira de desenhar os dois, qualquer dado de uma base de dados cartográfica pode ser mostrado. O GE, através da KML, conta com essas facilidades. No exemplo anterior, viu-se como desenhar dados vetoriais. Apresentou-se o exemplo de uma estrada, mas poderia ser qualquer informação vetorial, como polígonos representando diferentes tipos de cobertura vegetal ou tipos de solos. Apenas lembrar que os dados cartográficos precisam estar definidos no sistema de projeção lat, long e usando o datum de referência WGS84.

Para que colocar imagens no GE, visto que ele já as tem? Por vários motivos. O usuário pode querer colocar suas próprias imagens relativas a uma região de estudo, para divulgação, fazendo uso das facilidades de navegação do GE. Ele pode também ter imagens da área de estudo mais recentes do que as mostradas no GE. Ele pode querer agregar textos próprios de identificação em diferentes locais nas imagens, opção não permitida na versão cliente do GE. Ou ainda o usuário pode dispor de imagens de alta resolução espacial, Ikonos ou QuickBird, de regiões de estudo onde o GE só disponibiliza imagens Landsat. Além dessas, outras situações podem aparecer.

O trecho de código a seguir mostra um exemplo de inserção de uma imagem no GE:

```
<GroundOverlay >
  <visibility >0</visibility >
  <Icon >
    <href >C:\temp\TabCosteiros.jpg </href >
  </Icon >
  <LatLonBox >
    <rotation >0</rotation >
    <north >-2.98846867665222</north >
    <South >-3.10588622091239</south >
    <east >-41.7481566636012</east >
    <west >-41.8324697019907</west >
  </LatLonBox >
</GroundOverlay >
```

A tag que controla todo o processo de mostrar a imagem é `<groundOverlay>`, uma tag complexa. Ela contém as outras tags necessárias para definir e colocar uma imagem em sobreposição no globo do GE. Essas tags são mostradas no exemplo acima. Começando com `<visibility>`, uma tag simples que define o tipo de visibilidade inicial de um `<folder>`, `<placemark>` ou, nesse caso, `overlay` (imagem). No exemplo, ela tem valor zero, portanto, a imagem não é mostrada quando o GE lê o arquivo KML.

A tag <icon> já foi definida anteriormente.

<href> é uma tag simples. Ela define a localização de uma imagem a ser usada como overlay ou de um ícone a ser usado para a tag <placemark>. A localização tanto pode ser no disco local como num servidor *web* remoto. As tags <x>, <y>, <w> e <h> são usadas quando se pretende extrair uma parte da imagem para ser usada como ícone. O valor x, por exemplo, representa um deslocamento na imagem a partir do vértice esquerdo. As outras tags são explicadas de forma análoga.

<LatLonBox> é uma tag complexa, filha de <groundOverlay> e usada para especificar as coordenadas do próprio overlay (imagem). As suas tags filhas são <rotate>, <north>, <south>, <west> e <east>. A primeira é usada para especificar o eixo de rotação de uma imagem a partir de seu ponto central. No exemplo, não existe rotação. As outras quatro tags definem o conjunto de coordenadas do retângulo envolvendo da imagem, ou overlay. Elas definem as latitudes limites na sentido Norte-Sul e as longitudes limites no sentido Leste-Oeste.

A seguir, apresenta-se o arquivo completo KML, mostrados nos exemplos anteriores, com a colocação de imagem (overlay) e dois exemplos de arquivos vetoriais representando duas estradas:

```
<kml xmlns="http://earth.google.com/kml/2.0">
<Folder>
  <name><![CDATA[Piau&iacute;]]></name>
  <description>
    <![CDATA[Informa&ccedil;&otilde;es]]>
  </description>
  <visibility>0</visibility>
  <open>1</open>
  <Folder>
    <name>BR-343</name>
    <description>Rodovia Federal</description>
    <visibility>0</visibility>
    <open>0</open>
    <Placemark id="khPlacemark552">
      <!--
      <name>BR343</name>
      <Snippet>Sul ao norte</Snippet>
      -->
      <LookAt>
        <longitude>-41.7786</longitude>
        <Latitude>-3.0650</latitude>
        <range>50000.00</range>
        <tilt>0</tilt>
```

```

    <heading>3.8</heading>
  </LookAt>
  <visibility>0</visibility>
  <open>0</open>
  <MultiGeometry>
  <LineString id = "khLineString553" >
  <tessellate>1</tessellate>
  <coordinates>
  -41.84250585,-3.19347652 -41.84858548,-3.18549700
  -41.84904145,-3.18488904 -41.84942142,-3.18458506
  -41.85025737,-3.18420508 -41.85230925,-3.18329314
  -41.86887623,-3.17614958 -41.86864825,-3.17561761
  -41.86826827,-3.17516164 -41.86728033,-3.17462967
  -41.86614040,-3.17424969 -41.86522846,-3.17341374
  -41.86507647,-3.17280578 -41.86484848,-3.17219782
  -41.86454450,-3.17189384 -41.86401253,-3.17189384
  -41.86294860,-3.17090590 -41.86196066,-3.16984196
  -41.86120070,-3.16938599 -41.86059274,-3.16908201
  -41.86028876,-3.16893002 -41.85861686,-3.16862604
  -41.84265784,-3.16665016 -41.84220187,-3.16657416
  -41.84174589,-3.16627018 -41.84098594,-3.16611819
  -41.83103055,-3.16505426 -41.83065057,-3.16490226
  -41.82973863,-3.16482627 -41.82685081,-3.16467428
  -41.82609085,-3.16444629 -41.82540689,-3.16406632
  -41.82479893,-3.16368634 -41.82411497,-3.16323037
  -41.82365900,-3.16353435 -41.82297504,-3.16376233
  -41.82229108,-3.16376233 -41.82153113,-3.16345835
  -41.81879530,-3.16201444 -41.81750338,-3.16072252
  -41.81621146,-3.15920261 -41.81522352,-3.15745472
  -41.81453956,-3.15601081 -41.81362762,-3.15426292
  -41.81165174,-3.15251502 -41.80862442,-3.15004466
  -41.80862442,-3.15004466 -41.78140559,-3.12804452
  -41.78056964,-3.12736056 -41.78011367,-3.12675260
  -41.77980969,-3.12606864 -41.77958170,-3.12492871
  -41.77950570,-3.12318082 -41.77958170,-3.11998901
  -41.77958170,-3.10532191 -41.77958170,-3.10524592
  -41.77996168,-3.07803958 -41.75594715,-2.94671962
  </coordinates>
  </LineString>
  </MultiGeometry>
  </Placemark>
</Folder>
<Folder>
  <name>PI-210</name>
  <description>Rodovia Estadual</description>
  <visibility>0</visibility>

```

```

<open>0</open>
<Placemark>
  <visibility>0</visibility>
  <open>0</open>
  <Style>
    <LineStyle>
      <color>ff00ffff</color>
    </LineStyle>
  </Style>
  <LineString>
    <coordinates>
      -41.75048432,-2.93169962 -41.70845712,-2.96777836
      -41.70716397,-2.96894219 -41.70677603,-2.96958876
      -41.70651740,-2.97088191 -41.70690535,-2.97346820
      -41.70742260,-2.97915803 -41.70418974,-2.99894315
      -41.70393111,-3.00075355 -41.65880036,-3.01627129
      -41.64780863,-3.02477372 -41.62912269,-3.03961255
      -41.62627777,-3.04168158 -41.62388545,-3.04397692
      -41.62246299,-3.04552869 -41.61961807,-3.04630458
      -41.60332445,-3.06337409 -41.57177172,-3.07320199
      -41.55315043,-3.08005565 -41.53814995,-3.09350436
      -41.50472216,-3.09945283 -41.49153209,-3.09699585
      -41.47653161,-3.10178049 -41.46140181,-3.10187747
      -41.46023798,-3.10161884 -41.45881552,-3.10006707
      -41.44510819,-3.08131647 -41.44329779,-3.08079921
      -41.44187533,-3.07963538 -41.43954767,-3.07937675
      -41.43502166,-3.07873018 -41.43088360,-3.07795429
      -41.42428856,-3.07536800 -41.37973972,-3.05816918
      -41.37973972,-3.05816918
    </coordinates>
  </LineString>
</Placemark>
</Folder>
<Folder>
  <name><![CDATA[Tabuleiros Litor&acirc;neos]]> </name>
  <description><![CDATA[Projeto de Irriga&ccedil;&atilde;o]]> </description>
  <GroundOverlay>
    <visibility>0</visibility>
    <Icon>
      <href>C:\temp\TabCosteiros.jpg</href>
    </Icon>
  </LatLonBox>
  <north>-2.98846867665222</north>
  <south>-3.10588622091239</south>
  <east>-41.7481566636012</east>

```

```

<west>-41.8324697019907</west>
<rotation>0</rotation>
</LatLonBox>
</GroundOverlay>
</Folder>
</Folder>
</Kml>

```

Para facilitar, colocou-se dois pares de coordenadas por linhas. Esse procedimento não é necessário. Todas as coordenadas poderiam estar em uma só linha, desde que as coordenadas estejam separadas por vírgula e os pares estejam separados por um espaço branco. O único problema é a inserção da imagem TabCosteiros.jpg, referenciada localmente no diretório "c:\temp". Se o arquivo for lido pelo GE, e a opção "Tabuleiros Litorâneos" for escolhida ("check-list"), o GE mostra uma mensagem de erro numa janela e colocará uma imagem em branco na área especificada pela tag <groundOverlay>, com a inscrição "Image not Found", como era de se esperar (Fig. 4).

Para evitar esse tipo de problema, a imagem pode ser enviada para o usuário através de uma solicitação via *e-mail* para miranda@cnptia.embrapa.br.

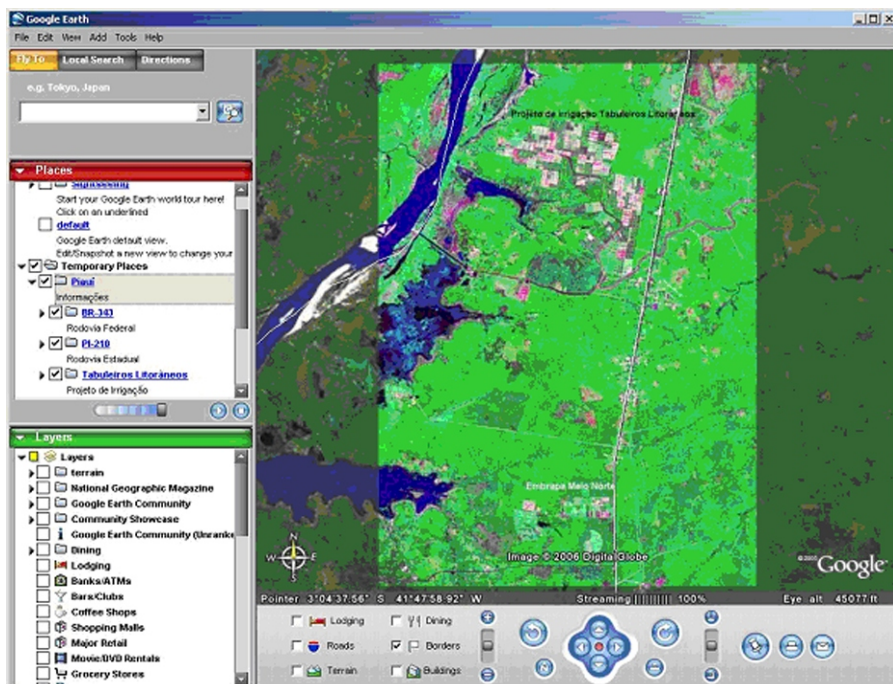


Fig. 4. O arquivo KML "Tabuleiros Litorâneos" aberto na interface do GE.

Outras opções da KML

< ScreenOverlay >

É possível ainda colocar uma marca, ou figura, de forma a identificar pessoalmente a parte visível no GE que mostra a área de trabalho. Isto é feito usando a tag complexa <screenOverlay>, usada para mostrar informação independente da posição da câmera. Desta maneira, pode-se criar cabeçalhos, aviso de direitos, informações etc. na própria área de monitor que aparece a imagem proprietária. Pode-se definir múltiplos <screenOverlay>, associando-os com as tags pais <folder> e <document>. As tags filhas são:

- <drawOrder> tag simples, usada quando imagens vão ser sobrepostas, a ordem de desenho pode ser definida a priori. Dessa maneira, imagens com bordas em cores escuras, ou pretas, podem receber um número menor, desenhando-as primeiro. Ela é filha de <groundOverlay> e <screenOverlay>. Quando não definida, a KML assume que seu valor é zero. Imagens com valores mais altos para essa tag são desenhadas por último.
- <icon> (requerida), essa tag permite a inserção de dados matriciais, ou imagens, no GE. Ela referencia um arquivo com a imagem a ser sobreposta no GE, podendo estar no disco local ou remotamente. Usuários que acessem o sistema em outro local físico, não precisam ter uma cópia da imagem se ela estiver referenciada com um endereço URL. Essa é uma tag do tipo complexa e filha de <groundOverlay>, <screenOverlay> e <iconStyle>. Por sua vez, as suas tags filhas são <href>, <x>, <y>, <w> e <h>.
- <visibility> uma tag simples que define o tipo de visibilidade inicial de um <folder>, <placemark> ou, nesse caso, overlay (imagem).
- <rotation> essa tag simples, filha de <screenOverlay>, <groundOverlay> e <latLonBox> é usada para especificar o eixo de rotação de uma imagem a partir de seu ponto central. Usa-se os valores +180 a -180 para indicar a rotação da imagem a partir do zero. O valor inicial dessa tag é zero, indicando sem rotação.
- <size>, tag simples, filha de <screenOverlay>, usada para especificar o tamanho de uma imagem. Para manter as dimensões originais da imagem, use o comando:
<size x = "0" y = "0" xunits = "fraction" yunits = "fraction" />
Para forçar a imagem aparecer no tamanho *xPixels* por *yPixels*:
<size x = "*xPixels*" y = "*yPixels*" xunits = "pixels" yunits = "pixels" />

- `<screenXY>` - define as coordenadas no monitor em que a imagem, definida em `<icon>`, será mapeada. Exemplo, se `screenXY` for (10, .15) com um `overlayXY` de (1,1) colocará o canto superior esquerdo da imagem 10 pixels a partir do vértice esquerdo do monitor e 15% abaixo do vértice superior do monitor.
- `<overlayXY>` uma tag simples, filha de `<screenOverlay>`. Ela define as coordenadas do ponto em que ocorrerá a sobreposição da imagem com as coordenadas da tela do monitor. É necessário fornecer os valores de X e Y e as unidades desses valores (pixels ou fração em porcentagem). Ela é usada em conjunção com o elemento `<screenXY>`, definido acima. Por exemplo, para colocar a imagem no centro da tela do monitor:
`<overlayXY x="0.5" y="0.5" xunits="fraction" yunits="fraction"/>`
`<screenXY x="0.5" y="0.5" xunits="fraction" yunits="fraction"/>`

A posição da tag pode ser definida em termos de coordenadas absolutas da área visível do cliente ou como uma porcentagem dessa área. Veja um exemplo no código a seguir:

```
<ScreenOverlay>
  <name>Icone</name>
  <description>
    Posicionamento absoluto:
    canto superior esquerdo.
  </description>
  <Icon>
    <href>c:\temp\google.gif</href>
  </Icon>
  <overlayXY x="0" y="1" xunits="fraction" yunits="fraction"/>
  <screenXY x="0" y="1" xunits="fraction" yunits="fraction"/>
  <size x="0" y="0" xunits="fraction" yunits="fraction"/>
</ScreenOverlay>
```

para ver o comportamento dessas linhas de código, insira-as no arquivo Embrapa.kml logo após o fechamento do elemento `<placemark>` `</placemark>`. Um elemento novo existe nas linhas de código apresentadas, a referência a uma imagem (google.gif). Se o arquivo XML for executado, o GE apresentará uma mensagem de erro "Image not Found", como era de se esperar.

Para evitar esse tipo de problema, a imagem pode ser enviada para o usuário através de uma solicitação via *e-mail* para miranda@cnptia.embrapa.br.

<Polygon>

Essa tag é utilizada para o desenho de polígonos no KML. Polígonos em duas dimensões podem ser criados mostrando seus contornos (vazio), preenchido e podem também ser posicionados em altitudes e ressaltados para formar objetos em três dimensões. <Polygon> é uma tag complexa com as seguintes tags filhas

- <outerBoundaryIs> uma tag complexa que define os limites externos de um polígono. É uma tag requerida na definição de um polígono. Ela usa a tag filha:
 - <linearRing> tag complexa que define a linha fechada do polígono, definida em conjunção com a tag pai <outerBoundaryIs>. Opcionalmente, pode conter um ou mais polígonos internos, definido pela tag <innerBoundaryIs>.
 - <linearRing> usa a tag <coordinates> para definir os valores das coordenadas dos vértices do polígono.
- <innerBoundaryIs> tag complexa que define os limites internos de uma ilha dentro de um polígono. Essa tag tem a mesma tag filha da tag anterior. As mesmas observações anteriores são válidas aqui, preservando as respectivas particularidades de cada tag.:

Polígonos em KML podem possuir estilos, definidos pela tag <style>. Com isto, pode-se definir cores de linhas e de preenchimento. Para que um polígono ganhe feições de três dimensões, é necessário usar a tag simples:

- <extrude> permite extrusão vertical de atributos de duas dimensões. Não são só polígonos que podem ser ressaltados, mas também <placemark> e <path>. A tag recebe um valor inteiro que representa a altitude de extrusão em metros.

Para se desenhar um polígono no GE, a seqüência de elementos a serem definidas no KML é a seguinte:

```
<Polygon>
  <outerBoundaryIs>
    <linearRing>
      <coordinates>
        -23.6534,-47.1643, ..... -23.6534,-47.1643
      </coordinates>
    </LinearRing>
  </outerBoundaryIs>
</Polygon>
```

Observe que as coordenadas do ponto inicial e final são iguais. Compare a declaração acima com uma feita originalmente em GML:

```

<gml:polygon>
  <gml:outerBoundaryls>
    <gml:linearRing>
      <gml:coordinates> ... </gml:coordinates>
    </gml:linearRing>
  </gml:outerBoundaryls>
</gml:polygon>

```

praticamente não existe diferença, exceto pelo prefixo "gml" em cada linha do código. Para o caso de se querer criar uma "ilha" dentro do polígono mais externo, o código seria:

```

<Polygon>
  <outerBoundaryls>
    <linearRing>
      <coordinates>
        -23.6534,-47.1643, ..... -23.6534,-47.1643
      </coordinates>
    </LinearRing>
  </outerBoundaryls>
  <innerBoundaryls>
    <linearRing>
      <coordinates>
        -56.3065,-40.2252..... -56.3065,-40.2252
      </coordinates>
    </linearRing>
  </innerBoundaryls>
</Polygon>

```

primeiro, declara-se o polígono mais externo, em seguida, o(s) polígono(s) mais interno(s), em ordem seqüencial. As próximas linhas de código em KML exemplificam o uso do atributo <polygon> para identificar no GM a localização da mata de Santa Genebra, uma área de preservação em Campinas, SP, nas proximidades do distrito de Barão Geraldo. O código foi implementado de maneira a mostrar o desenho apenas dos limites da mata - polígono não preenchido, e da mesma área preenchida, com uma tonalidade cinza transparente:

```

<?xml version = "1.0" encoding = "UTF-8"?>
<kml xmlns = "http://earth.google.com/kml/2.0">
<Folder>
  <name> Santa Genebra </name>
  <visibility>0 </visibility>
  <Placemark>
    <name> Limites </name>
    <description> mata de Sta Genebra

```



```

Lat = -22.8186
Long = -47.1121
</description>
<visibility>0</visibility>
<Style>
  <geomColor>ff00ffff</geomColor>
  <geomScale>3</geomScale>
</Style>
<LineString>
  <coordinates>
-47.11947688,-22.81019162 -47.10793420,-22.81710607
-47.10442121,-22.82218039 -47.10520188,-22.82753351
-47.10548069,-22.83528439 -47.10904944,-22.83160412
-47.11356614,-22.82764504 -47.11462561,-22.82613947
-47.11440256,-22.82100939 -47.11490442,-22.81989416
-47.11713488,-22.81933654 -47.12075940,-22.81950383
-47.12349172,-22.82034025 -47.12561066,-22.81933654
-47.12561066,-22.81933654 -47.11947688,-22.81019162
  </coordinates>
</LineString>
</Placemark>
<!-- poligono preenchido -->
<Placemark>
  <name> Cobertura </name>
  <description> mata de Sta Genebra
  Lat = -22.8186
  Long = -47.1121
  </Description>
  <Style>
    <geomColor>8800FFFF</geomColor>
    <polyMode>outlineFill</polyMode>
  </Style>
  <drawOrder> </drawOrder>
  <GeometryCollection>
    <Polygon>
      <outerBoundaryIs>
        <LinearRing>
          <coordinates>
-47.11947688,-22.81019162 -47.10793420,-22.81710607
-47.10442121,-22.82218039 -47.10520188,-22.82753351
-47.10548069,-22.83528439 -47.10904944,-22.83160412
-47.11356614,-22.82764504 -47.11462561,-22.82613947
-47.11440256,-22.82100939 -47.11490442,-22.81989416
-47.11713488,-22.81933654 -47.12075940,-22.81950383

```

```

-47.12349172,-22.82034025 -47.12561066,-22.81933654
-47.12561066,-22.81933654 -47.11947688,-22.81019162
</coordinates>
</LinearRing>
</outerBoundaryIs>
</Polygon>
</GeometryCollection>
</Placemark>
</Folder>
</kml>

```

Os dois polígonos estão definidos entre as duas tags <placemark>. O primeiro polígono, que mostra apenas o contorno da mata de Santa Genebra, não foi definido com a tag <polygon>, como pode ser observado no código, mas como a tag <lineString>. Essa tag pode cumprir o papel do <polygon>, uma vez que os valores das coordenadas inicial e final são o mesmo. A cor da linha foi definida no tag <style>:

```

<Style>
  <geomColor> ff00ffff </geomColor>
  <geomScale> 3 </geomScale>
</Style>

```

Como foi explicado, as cores são definidas em um elemento de 32 bits no formato AABBGGRR. Nesse caso, a tag <geomColor> definiu a cor a ser usada no traçado da linha da seguinte forma:

- AA = FF - valor de alfa ou da transparência. Nesse caso, a linha é desenhada de forma sólida (FF);
- BB = 00 - não há nenhuma componente de cor azul na composição final da cor da linha;
- GG = FF - a cor final da linha recebe valor máximo da componente azul; RR = FF - a cor final recebe também valor máximo para vermelho.

Dessa maneira, a cor final da linha será uma combinação das cores azul e vermelho, produzindo a cor amarela, conforme mostra o desenho do polígono de contorno da mata no GE. A tag <geomScale> é do tipo simples, filha de <style>, e é usada como valor de escala, tanto para ícones como para linhas. No primeiro caso, ela define um fator de escala do ícone em pixels e no segundo caso, a largura da linha. A tag simples <geomColor> é usada para definir a cor dos objetos definidos dentro da tag pai <style>.

O polígono preenchido está definido no segundo <placemark>, dentro da tag <GeometryCollection>, embora não fosse necessária. Notar que o nível de transparência do polígono (AA) foi definido como 88, em hexadecimal, permitindo o usuário ver a mata de Santa Genebra "por baixo" do polígono preenchido (Fig. 5).

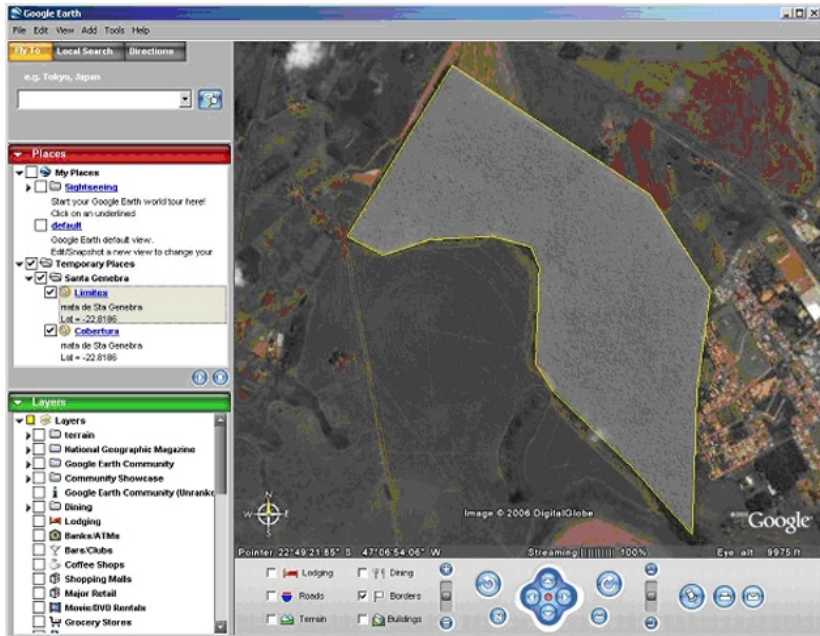


Fig. 5. O arquivo KML "Santa Genebra" aberto na interface do GE.

Existem muitas outras opções na KML que podem ser usadas. Não se discorre sobre todas elas. O objetivo foi apenas de mostrar ao usuário que é possível publicar dados proprietários usando a versão grátis do cliente do GM. Com as informações apresentadas nesse documento, temos certeza de que qualquer interessado será capaz de publicar seus dados usando uma ferramenta tão útil como é o GM.

Referências Bibliográficas

GEOTOOLS. **Geotools**: the open source Java GIS toolkit. Disponível em: <<http://www.geotools.org>>. Acesso em: 5 out. 2005.

MIRANDA, J. I. **Diretivas para disponibilizar mapas na Internet**. Campinas: Embrapa Informática Agropecuária, 2002. 29 p. (Embrapa Informática Agropecuária. Documentos, 14). Disponível em: <<http://www.cnptia.embrapa.br/modules/tinycontent3/content/2002/doc14>>. Acesso em: 04 set. 2005.

OPEN GEOSPATIAL CONSORTIUM. **Open Geospatial Consortium [home page]**. Disponível em: <<http://www.opengeospatial.org/>>. Acesso em: 6 set. 2005.



Informática Agropecuária

**Ministério da
Agricultura, Pecuária
e Abastecimento**

