MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS
(GOCNAE, 1961)

Reestruturação do pacote GML para a TerraLib

CAP-365 Paradigmas e Ferramentas de Desenvolvimento de Software

Emerson M. A. Xavier
emerson@dpi.inpe.br

São José dos Campos, 15 de dezembro de 2006.

Considerações iniciais

- Trabalho de CAP-349 Bancos de Dados Geográficos
  - Próximas etapas:

- Reestruturar as classes
  - Dividir a TeGMLEncoder em várias partes
    - Por tipo de documento (GML / XML Schema)
    - Por tipo de feição (ponto / linha / polígono / ...)
    - Encoder + Decoder
  - Possibilitar o uso das classes tanto p/ servidores como para clientes

Objetivo

- Implementar o pacote GML para a TerraLib com a utilização dos conhecimentos adquiridos ao longo do curso
  - O que antes era uma classe, agora é um pacote

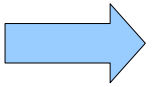
Roteiro

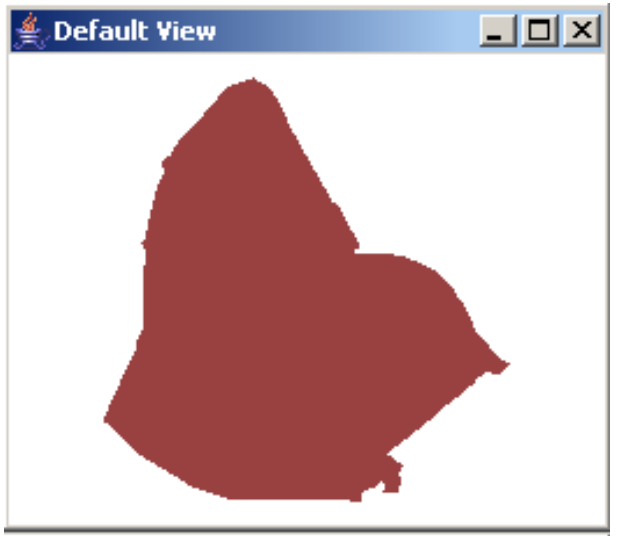
- Introdução
- Implementação
  - TeGMLEncoder
  - Forma canônica
  - Template
  - Singleton
  - Factory Method
- Próximas etapas
- Conclusões

Geography Markup Language (GML)

- Especificação do OGC para codificar informação geográfica num arquivo XML

```
<distritos>
      <TeGeometry>
        <gml:Polygon srsName="EPSG:29193">
          <gml:outerBoundaryIs>
            <gml:LinearRing>
              <gml:coordinates>
                 330221.3,7396108.7 ...
              </gml:coordinates>
            </gml:LinearRing>
          </gml:outerBoundaryIs>
        </gml:Polygon>
      </TeGeometry>
      <sprarea>3842344.0313</sprarea>
      <sprperimet>8576.6837</sprperimet>
      <sprrotulo>54</sprrotulo>
      <sprnome>54</sprnome>
      <id2>413</id2>
      <area>3852</area>
      <cod>70</cod>
      <sigla>SCE</sigla>
      <deno>SANTA CECILIA</deno>
      <object_id_7>53</object_id_7>
</distritos>
```

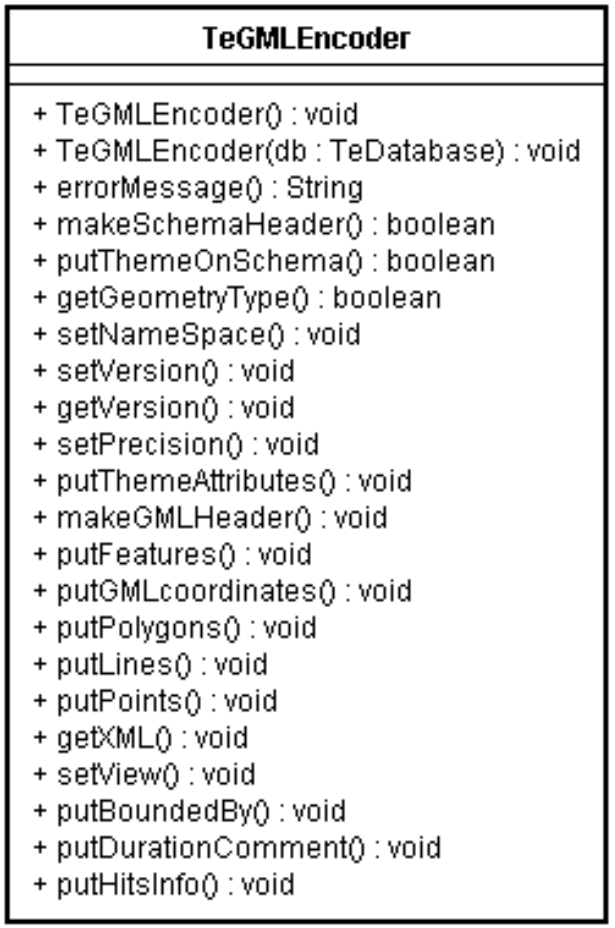




Attribute Name	distritos
area	3852
cod	70
deno	SANTA CECILIA
id2	413
object_id_7	53
sigla	SCE
sprarea	3842344.0313
sprnome	54
sprperimet	8576.6837
sprrotulo	54
TeGeometry/Polygon/...	330221.3,7396108.7 ...

Implementação

- TeGMLEncoder
  - Criar o objeto GML e o XML Schema associado
  - Spec GML 2.1.2 e 3.1.1

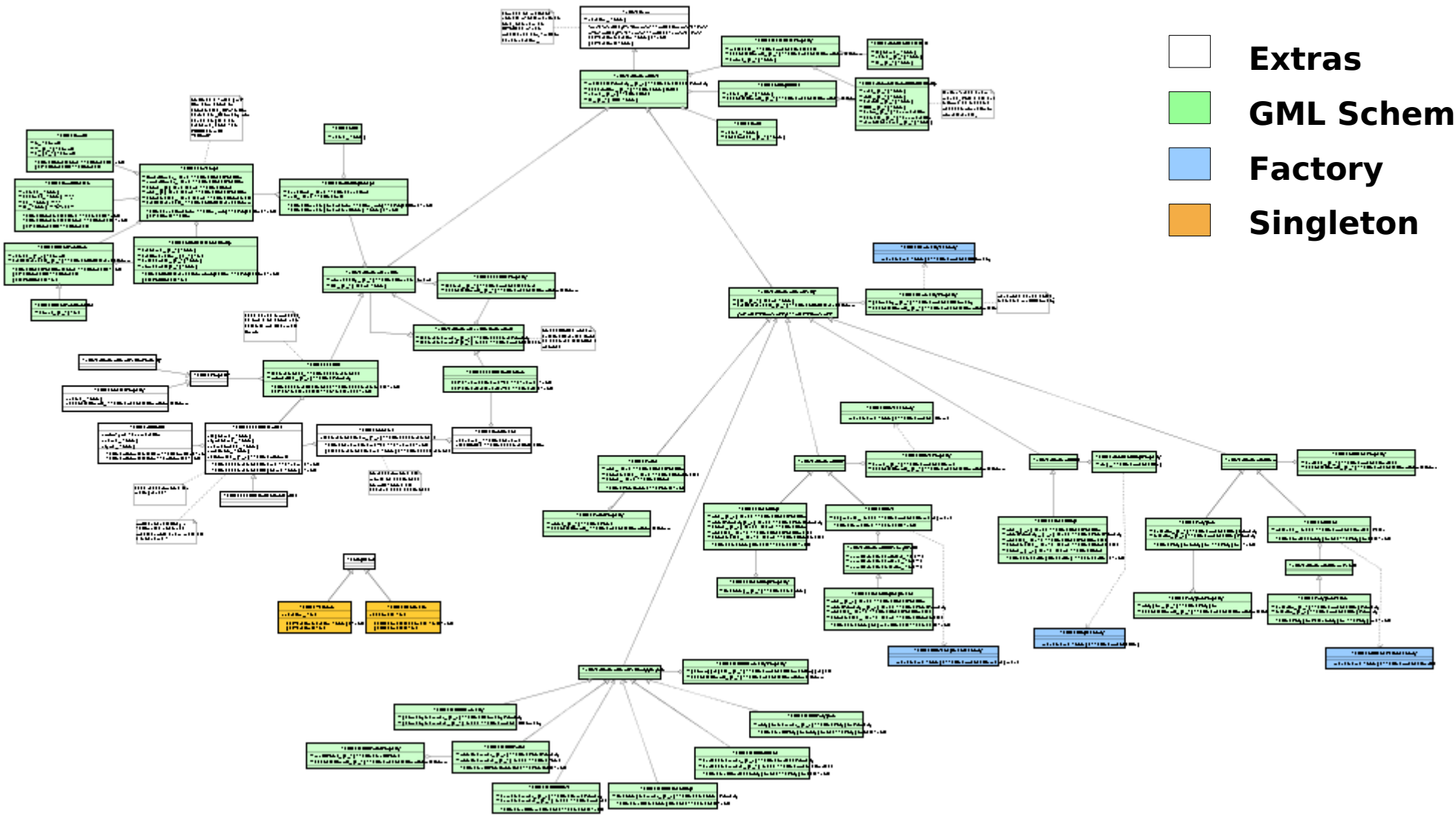


Programação estruturada “dentro” de uma programação orientada-a-objetos

Pacote TeGML

- Uma nova abordagem a partir do conhecimento adquirido dentro e fora da sala de aula
  - Foco nos Schemas XML do GML

Pacote TeGML



Forma Canônica

```
namespace TeOGC
{

/** \class TeGMLCode
  * \brief This class represents a name or code with an (optional) authority.
  *
  * If the codeSpace attribute is present, then its value should identify a dictionary,
  * thesaurus or authority for the term, such as the organisation who assigned the
  * value, or the dictionary from which it is taken. A text string with an optional
  * codeSpace attribute.
  *
  */
class TeGMLCode : public TeGMLBase
{
protected:

    std::string value_;  //!< (Mandatory)

public:

    // PUBLIC METHODS

private:

    // Unimplemented constructors and operators
    TeGMLCode( const TeGMLCode& rhs );
    TeGMLCode& operator=( const TeGMLCode& rhs );
};

} // end namespace TeOGC
```

Forma Canônica – Métodos

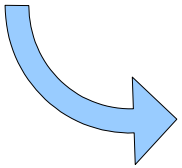
- Características
  - Don't-put-your-hand-in-the-hole-of-the-tatu style
  - Métodos `const`
  - Sem `using namespace std;`

```
/** @name Accessor methods
* Methods used to get or set object's properties.
*/
//@{

/** \brief Returns the value associated to this object.
  */
const std::string& getValue() const;

/** \brief Sets the string value.
    \param value The value.
  */
void setValue( const std::string& value );

//@}
```

Tratamento de exceções

- Mudança de estilo

```
bool getTePolygon( TePolygon& polygon )
{
// processamento ...

if( polygon.empty() )
     return false;

return true;
}
```



```
void getTePolygon( TePolygon& polygon )
{
// processamento ...

if( polygon.empty() )
     throw TeGMLException( "Empty TePolygon.", "Surface" );

}
```

O TeGMLException é uma especialização do TeException

Implementação

- Quando surgiu no código:

```
if( xmlDecoder->setCurrentElement( "gml:metaDataProperty" ) )
{
     do
     {
               TeGMLMetadataProperty* mdProp = new TeGMLMetadataProperty();
               mdProp->readFromXML( xmlDecoder );
               metadataProperty_.push_back( mdProp );
     }
     while( xmlDecoder->setNextSiblingAsCurrent( "gml:metaDataProperty" ) );

     xmlDecoder->closeElement();
}

if( xmlDecoder->setCurrentElement( "gml:name" ) )
{
     do
     {
               TeGMLCode* name = new TeGMLCode();
               name->readFromXML( xmlDecoder );
               name_.push_back( name );
     }
     while( xmlDecoder->setNextSiblingAsCurrent( "gml:name" ) );

     xmlDecoder->closeElement();
}
```

Implementação

- Quando surgiu no código:

```
if( xmlDecoder->setCurrentElement( "gml:metaDataProperty" ) )
{
     do
     {
               TeGMLMetadataProperty* mdProp = new TeGMLMetadataProperty();
               mdProp->readFromXML( xmlDecoder );
               metadataProperty_.push_back( mdProp );
     }
     while( xmlDecoder->setNextSiblingAsCurrent( "gml:metaDataProperty" ) );

     xmlDecoder->closeElement();
}

if( xmlDecoder->setCurrentElement( "gml:name" ) )
{
     do
     {
               TeGMLCode* name = new TeGMLCode();
               name->readFromXML( xmlDecoder );
               name_.push_back( name );
     }
     while( xmlDecoder->setNextSiblingAsCurrent( "gml:name" ) );

     xmlDecoder->closeElement();
}
```

O mesmo comportamento ...

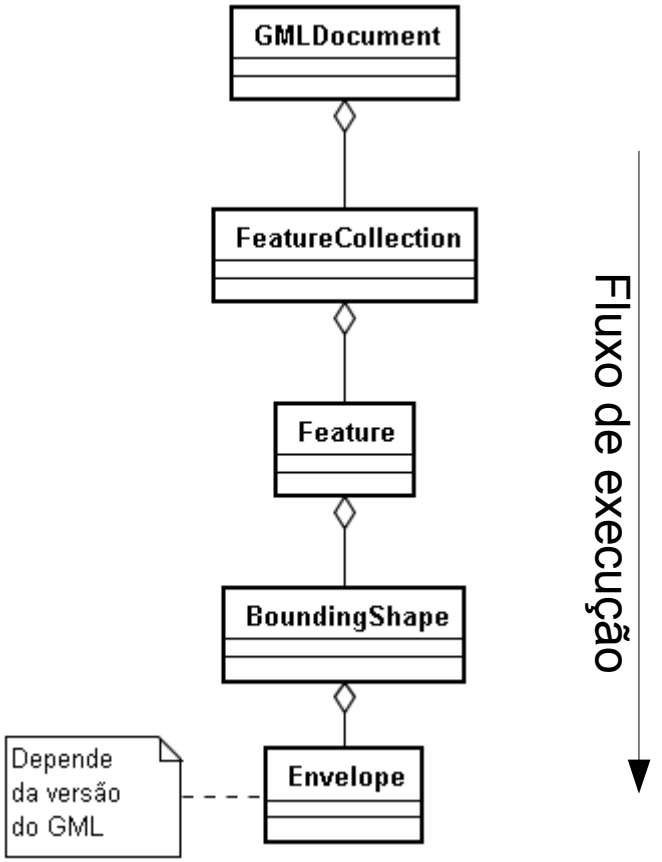
Template

- Economia:
  - Tempo, código e erros

```
template <typename T>
bool TeReadPtrVectorFromXML( std::vector< T* >& basicVector,
    const std::string& tagName, TeXMLDecoder* xmlDecoder)
{
    if( xmlDecoder->setCurrentElement( tagName ) )
    {
            do
            {
                T* ptr = new T();

                ptr->readFromXML( xmlDecoder );

                basicVector.push_back( ptr );
            }
            while( xmlDecoder->setNextSiblingAsCurrent( tagName ) );

            xmlDecoder->closeElement();

            return true;
    }
    return false;
}
```

Implementação

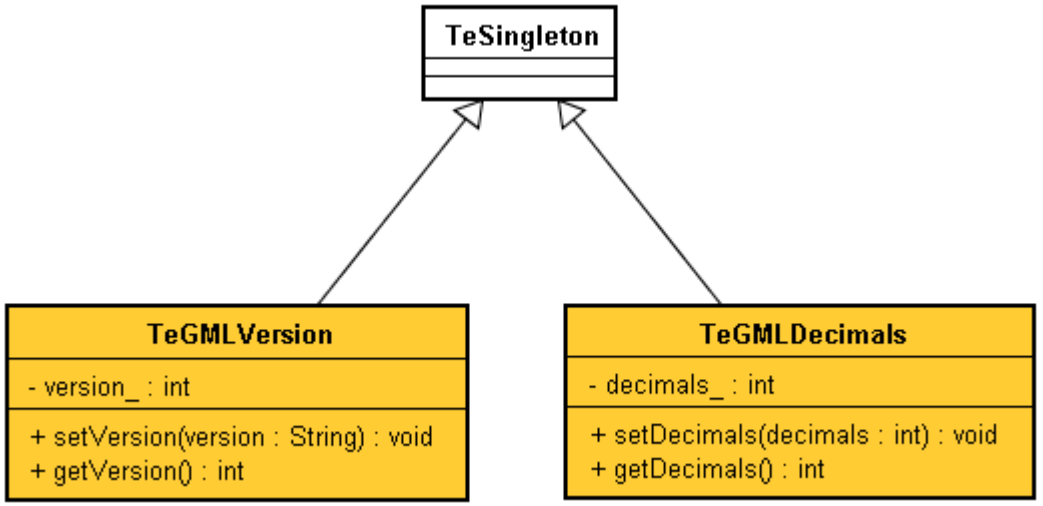
- Alguns objetos têm comportamento diferente de acordo com a versão do GML
- Solução
  - Uso de um atributo `version_`
    - Estava “poluindo” a interface



Qual a outra solução possível?

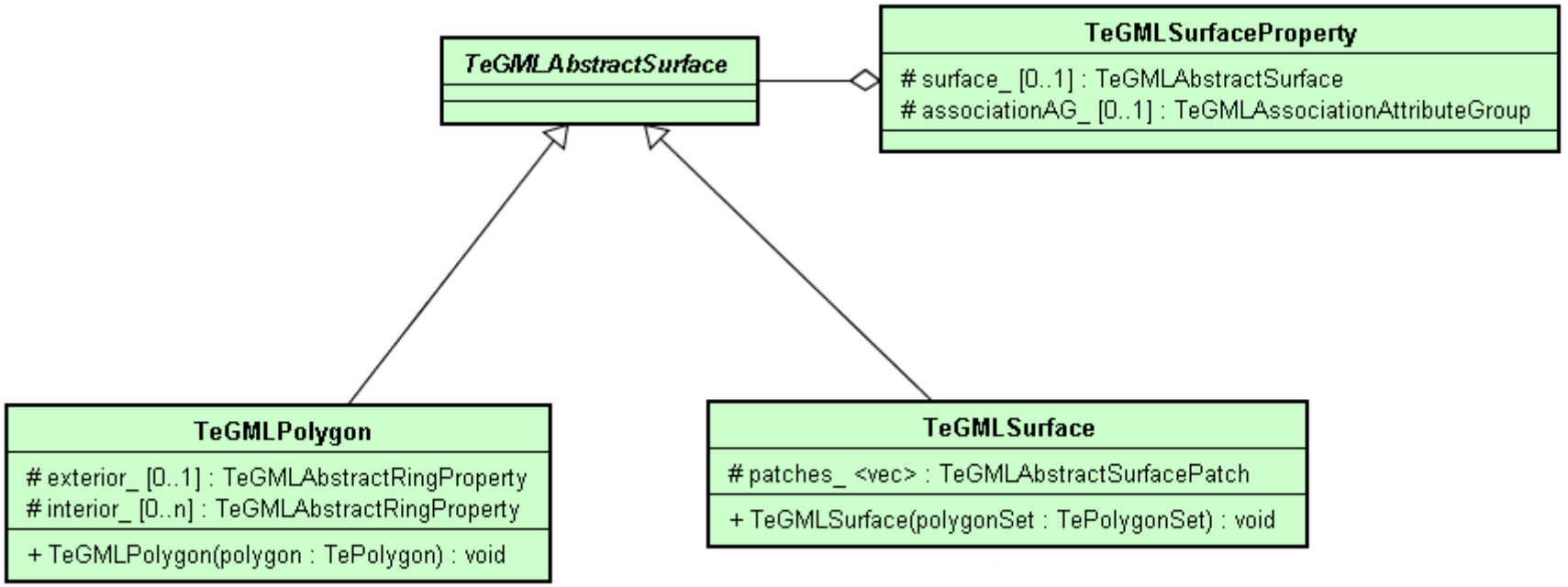
Singleton

- Caracteríticas
  - A versão tem uma única instância e um único ponto de acesso
    - Definido pelo GMLDocument
    - Acesso em qualquer nível
  - O mesmo se aplica ao Decimals
    - Definido pela projeção (SRSAttributeGroup)
    - Acesso pelos objetos que vão converter `double` em `string`



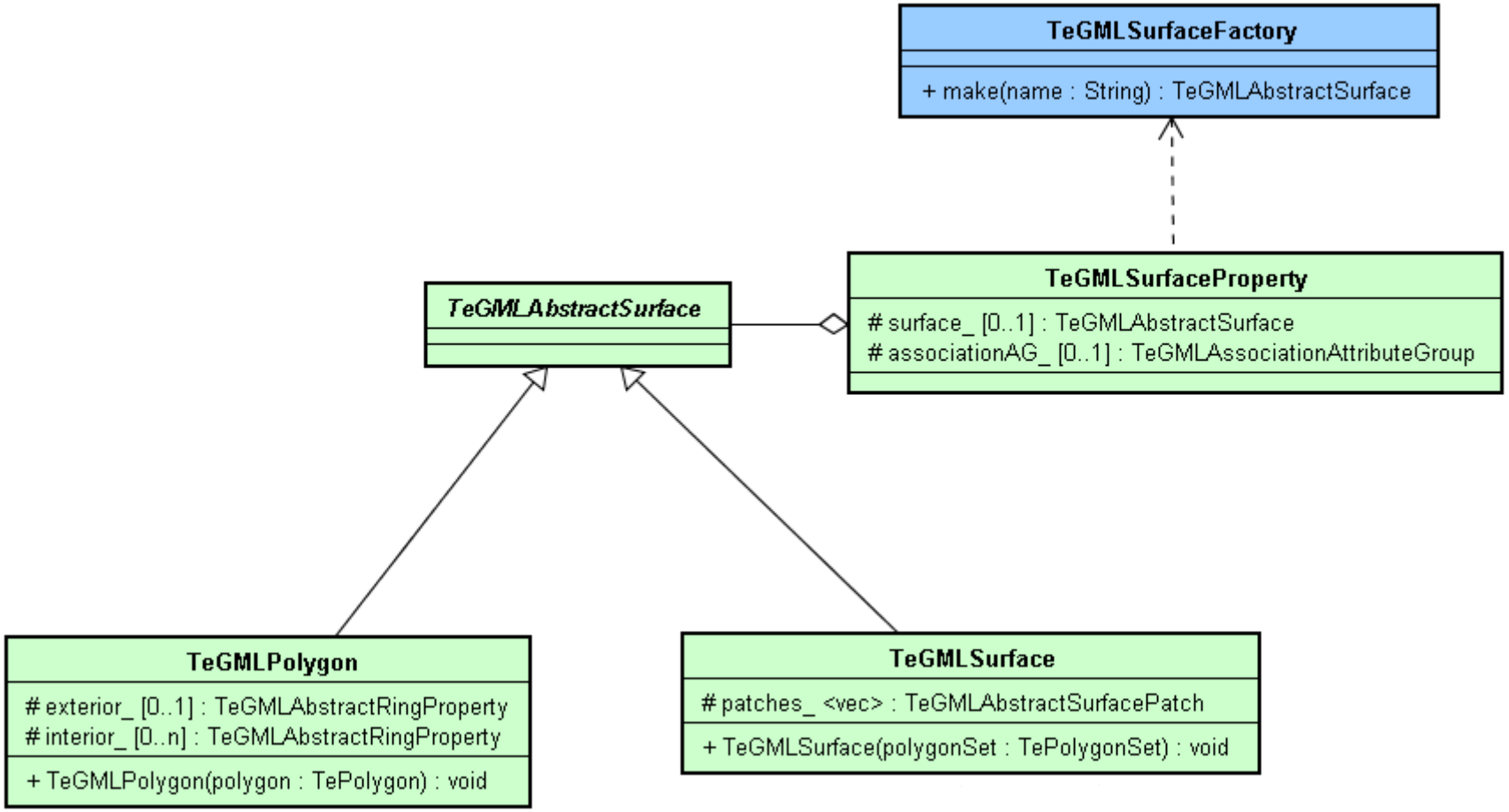
Implementação

- Queremos ler um objeto AbstractSurface



Factory Method

- Solução: fábrica



Factory Method

- Código
  - Mais simples que o Abstract Factory

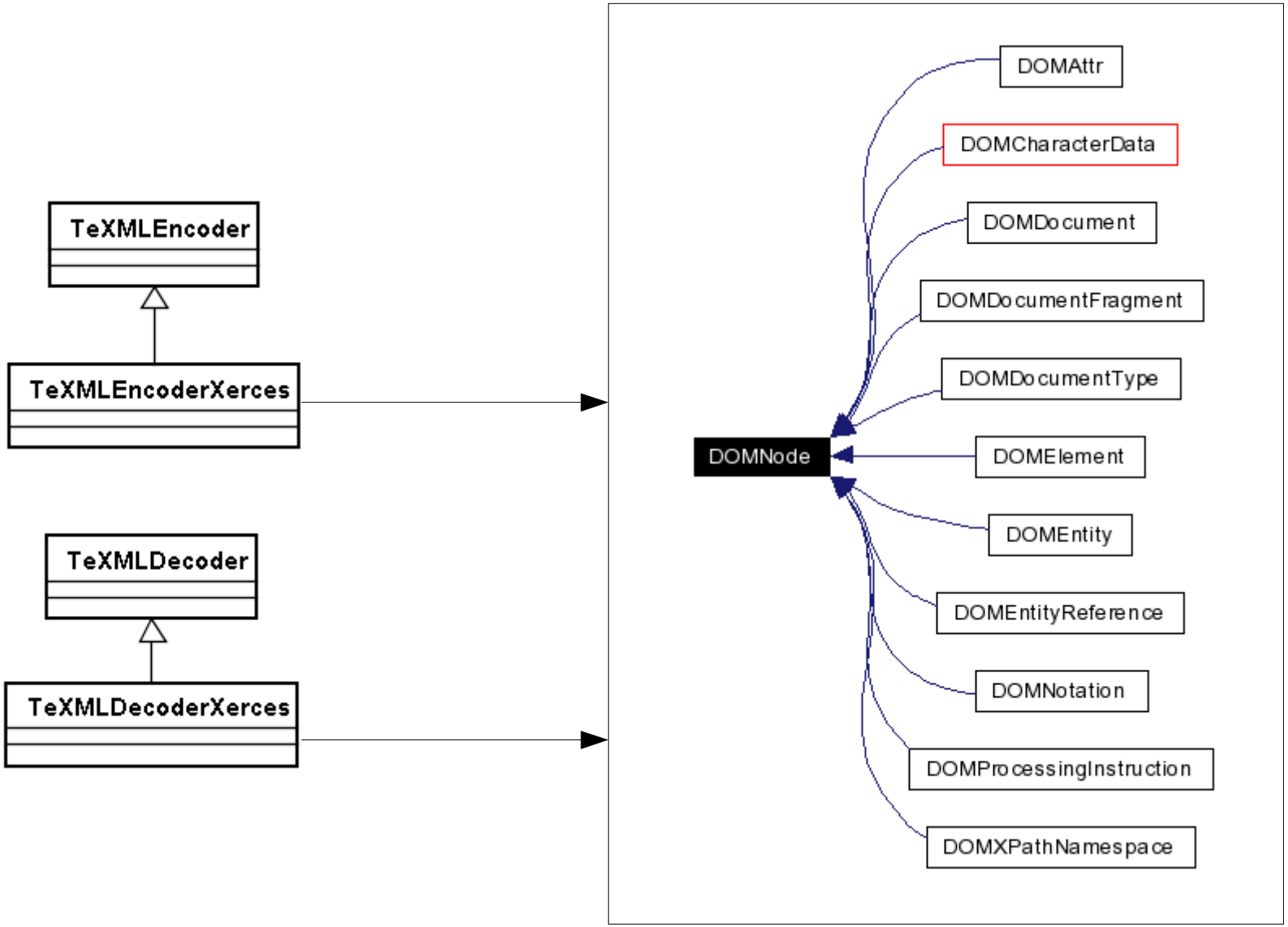
```
TeOGC::TeGMLAbstractSurface*
TeOGC::TeGMLSurfaceFactory::make( const std::string& name )
{
     if( name == "gml:Polygon" )
     {
          return new TeGMLPolygon();
     }
     else if( name == "gml:Surface" )
     {
          return new TeGMLSurface();
     }
     else
          throw TeGMLException( "Invalid object.", "Factory" );

     return 0;
}
```

Template + Factory

```
template< class T, class TFactory >
class TeGMLAbstractProperty : public TeGMLProperty
{
protected:

    T* element_; //!< A pointer to the basic element. (Mandatory)

public:

    /** \brief Method that read content from a XML document.
        \param xmlDecoder A valid pointer to a xml decoder, used to read tags.
         */
    virtual void readFromXML( TeXMLDecoder* xmlDecoder )
    {
            if( xmlDecoder->setFirstChildAsCurrent() )
            {
                element_ = TFactory::make( xmlDecoder->getElementName() );

                element_->readFromXML( xmlDecoder );

                xmlDecoder->closeElement();
            }
            else
                throw TeGMLException( "Missing element.", "AbstractProperty" );
    }

};
```

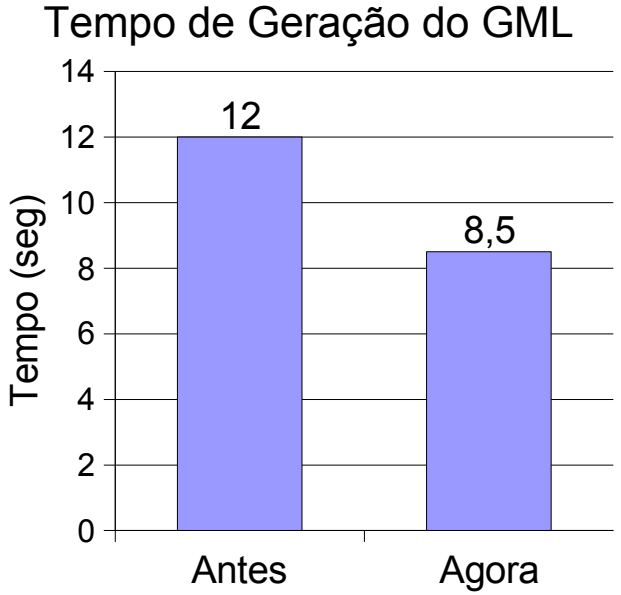
```
typedef TeGMLAbstractProperty< TeGMLAbstractSurface, TeGMLSurfaceFactory >
    TeGMLSurfaceProperty;
```

Facade


TeXMLEncoder
TeXMLEncoderXerces
TeXMLDecoder
TeXMLDecoderXerces
DOMAttr
DOMCharacterData
DOMDocument
DOMDocumentFragment
DOMDocumentType
DOMNode
DOMElement
DOMEntity
DOMEntityReference
DOMNotation
DOMProcessingInstruction
DOMXPathNamespace

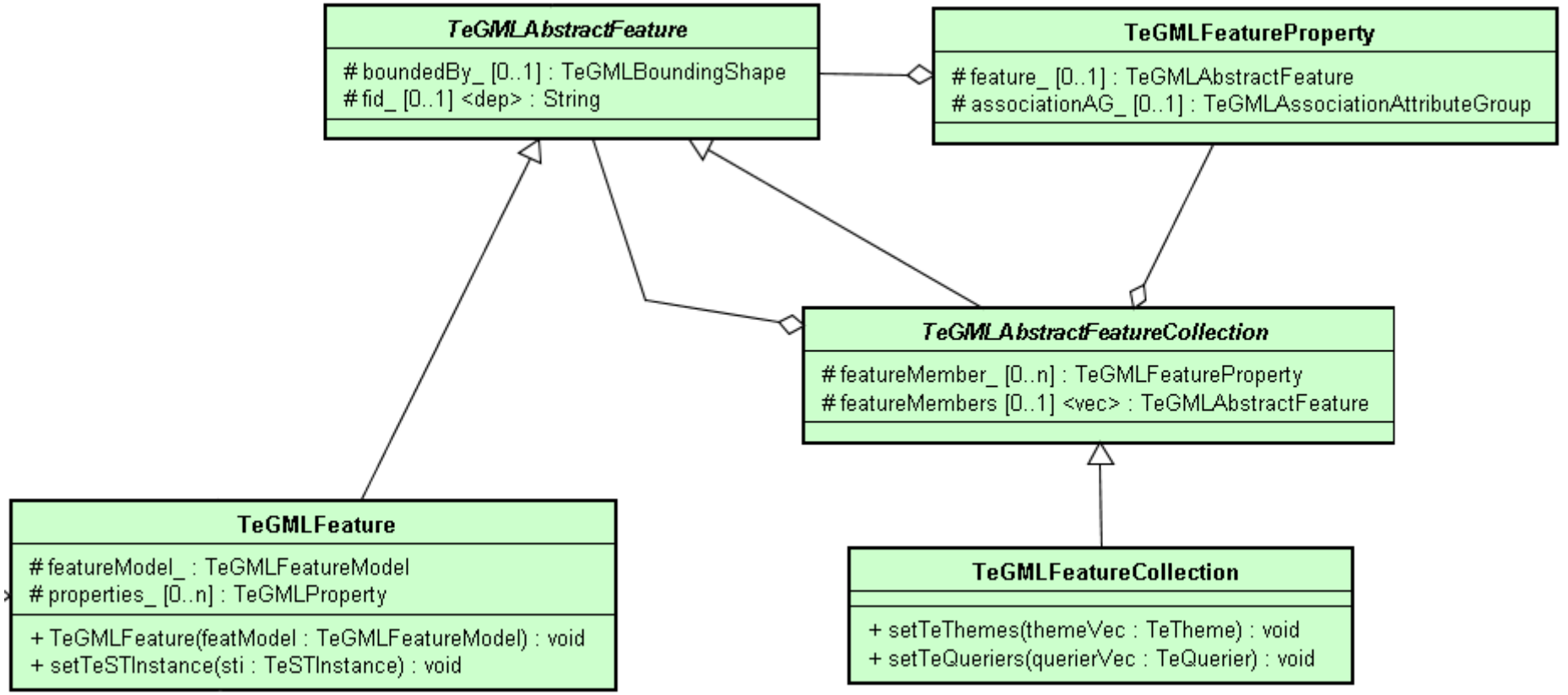
Resultados complementares

- Municípios do Brasil
  - Arquivo original: munic_2001.shp
    - Tamanho: 9.83 MB (shp + dbf + shx)
    - 5798 polígonos / 5562 feições



Próximas etapas

- Continuar a implementação do pacote seguindo a mesma doutrina



Conclusões

- A reestruturação do pacote TeGML sob a ótica dos padrões de projeto trouxe benefícios:
  - Estruturais (facilidade de manutenção)
  - Performance
- A identificação de um padrão de projeto não é necessariamente imediata
  - Um diagrama UML ajuda
- O XML é o padrão de intercâmbio de dados
  - Haskell XML Toolbox | HaXml | HXML

Agradecimentos

- Grupo TerraLib Webservices
  - Gilberto Ribeiro
  - Karla Fook
  - Vanessa Souza
  - Sérgio Cruz

Debates

- Frases sobre o GoF book
  - “Lendo esse livro você vê que seus programas são realmente desorganizados.”
  - “Posso dividir meus projetos orientados a objetos em antes e depois desse livro.”
  - “Você nunca programou realmente orientado a objetos antes de aprender as técnicas deste livro!“

Fonte: clientes www.submarino.com.br