

Geographic Markup Language

Gilles Gesquière

Institut de l'Université de Provence, site d'Arles
Laboratoire LSIS, Marseille, France

Références de ce cours

- Extrait du cours de Ian Turton : <https://courseware.education.psu.edu/courses/geog585/content/lesson06/1.html>

Gilles Gesquière - LSIS/ Université Aix- Marseille- gilles.gesquiere@lsis.org

1- Liste des choses à faire

- Lire OGC GML Simple Features Profile
- http://portal.opengeospatial.org/files/?artifact_id=15201
- Lecture du cours GML

Gilles Gesquière - LSIS/ Université Aix- Marseille- gilles.gesquiere@lsis.org

2- Introduction

- Geographic Markup Language (GML)
 - ▶ Langage basé sur le XML
 - ▶ Utilisé pour l'échange de données géographiques
 - ▶ Utilise des schémas XML pour définir la géométrie d'objets nécessaires pour encoder les éléments géographiques
 - ▶ Nécessaire de définir un schéma d'application

Gilles Gesquière - LSIS/ Université Aix- Marseille- gilles.gesquiere@lsis.org

2- Introduction

- GML est utilisé dans WFS (Web Feature Service)
- GML est aussi inclus dans d'autres standards qui permettent d'échanger des éléments géographiques.
- Un autre exemple d'utilisation de GML : Standard GeoRSS
 - ▶ <http://www.georss.org>

Gilles Gesquière - LSIS/ Université Aix- Marseille- gilles.gesquiere@lsis.org

2- Introduction

- Objectifs :
 - ▶ Définition de GML
 - ▶ Description de l'utilisation de GML dans l' "open web mapping"
 - ▶ Relations entre XML et GML
 - ▶ Définition du schéma d'application
 - ▶ Comprendre comment construire et appliquer un schéma d'application à un document GML

Gilles Gesquière - LSIS/ Université Aix- Marseille- gilles.gesquiere@lsis.org

2- Introduction à GML

- Langage basé XML
- Permet d'exprimer des caractéristiques géographiques dans l'objectif de les rendre interopérables
- Deux points
 - ▶ Le langage qui décrit le document
 - ▶ L'instance qui contient les données

Gilles Gosquière - LISIS/ Université Aix- Marseille- gilles.gosquiere@isik.org

2- Introduction à GML

- Un document général GML est décrit en utilisant un Schéma GML
- Cela permet de décrire des données géographiques génériques qui contiennent
 - ▶ Des points
 - ▶ Des lignes
 - ▶ Des polygones
- Il y a des extensions spécialisées en fonction de communautés
 - ▶ Exemple
 - ▷ cityGML
 - ▷ SensorML
 - ▶ http://en.wikipedia.org/wiki/GML_Application_Schemas

Gilles Gosquière - LISIS/ Université Aix- Marseille- gilles.gosquiere@isik.org

2- Introduction à GML

- L'utilisation principale de GML est de permettre les échanges interopérables de données géographiques
- Utilisé dans WFS (Web Feature Service) afin d'envoyer des objets géographiques entre les serveurs et les clients
- GML peut aussi servir de format commun afin d'échanger les données
- Gain de temps et peut éviter d'altérer les données

Gilles Gosquière - LISIS/ Université Aix- Marseille- gilles.gosquiere@isik.org

2- Introduction à GML

- Utilisation de GML pour le stockage de données
- Assez rare car GML est un langage basé sur du texte ce qui implique des fichiers de grandes tailles.
- Fichiers facile à lire de par leur construction à l'aide de balises.

Gilles Gosquière - LISIS/ Université Aix- Marseille- gilles.gosquiere@isik.org

3- Version de GML

- GML continue à évoluer
 - ▶ Version 1 : pas vraiment utilisée
 - ▶ Version 2 : (GML 2) obsolète techniquement
 - ▶ Version 3 : toujours utilisée dans WFS 1.0
 - ▶ Version 4 : Utilisée par défaut dans WFS 1.1
 - ▷ Pas beaucoup de serveurs en GML 3
 - ▷ Taille de la spécification 550 pages
 - ▶ En janvier 2006, la version 3.1.1 du GML est proposée.
 - ▶ La norme ISO 19136 correspondant au standard OGC GML 3.2.1, publiée en 2007
 - ▶ GML SWG travaille actuellement sur GML 4

Gilles Gosquière - LISIS/ Université Aix- Marseille- gilles.gosquiere@isik.org

4- Géométries

- GML 2
 - ▶ Point
 - ▷ Représente un simple point dans l'espace géographique
 - ▷ peut optionnellement inclure une élévation
 - ▶ Polygone
 - ▷ Représente un espace géographique fermé
 - ▷ Les coordonnées peuvent optionnellement avoir une élévation
 - ▷ Un polygone peut avoir des trous ainsi que des îles dans ces trous, mais les trous ne peuvent pas toucher l'extérieur du polygone
 - ▶ LineString
 - ▷ Représente un linéaire géographique.
 - ▷ Les coordonnées peuvent optionnellement avoir une élévation
 - ▷ Les lignes sont représentées comme une série de points jointes par des lignes.

Gilles Gosquière - LISIS/ Université Aix- Marseille- gilles.gosquiere@isik.org

4- Géométries

- GML 3 permet de définir de nouvelles géométries
 - Courbes
 - Représentation plus générale que que LineString qui inclut LineStrings, mais aussi des courbes lisses comme
 - Arcs
 - Splines Cubiques
 - Bézier
 - Surface
 - Représentation de type polygone qui inclut
 - Patches / Carreaux (polygones avec des courbes)
 - Rectangles
 - Triangles
 - Coverages (Grilles)
 - Moyen de représenter un raster en GML

Gilles Gosquière - LISIS/ Université Aix - Marseille - gilles.gosquier@isik.org

4- Géométries - Exemple GML : Un pont

```

<Bridge>
  <span>100</span>
  <height>200</height>
  <gml:centerLineOf>
    <gml:LineString>
      <gml:pos>100 200</gml:pos>
      <gml:pos>200 200</gml:pos>
    </gml:LineString>
  </gml:centerLineOf>
</Bridge>
    
```

Elément racine

Eléments spécifiques pour un pont

Ligne centrale du pont

Définition d'une ligne et des points qui la composent

Gilles Gosquière - LISIS/ Université Aix - Marseille - gilles.gosquier@isik.org

4- Géométries - Géométries simples

Il y a 3 géométries de base qui couvrent la majorité des cas
Ces géométries simples sont les seules permises si on veut travailler avec le "Simple Feature Profile"

- Points
- Lignes
- Polygones

Gilles Gosquière - LISIS/ Université Aix - Marseille - gilles.gosquier@isik.org

4- Géométries Géométries simples- Point

```

<gml:Point>
  <gml:pos>0 100</gml:pos>
</gml:Point>

<gml:Point>
  <gml:coordinates>
    0,100
  </gml:coordinates>
</gml:Point>
    
```

Deux modes de création

- <gml:pos>
- <gml:coordinates>
- Le schéma fournit aussi <gml:coord> mais juste pour la rétro compatibilité avec GML 2

Gilles Gosquière - LISIS/ Université Aix - Marseille - gilles.gosquier@isik.org

4- Géométries - Géométries simples- Points

Schéma pour le point

```

<complexType name="PointType">
  <annotation>
    <documentation>
      A Point is defined by a single coordinate tuple.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractGeometricPrimitive">
      <sequence>
        <choice>
          <element ref="gml:pos"/>
          <element ref="gml:coordinates"/>
          <element ref="gml:coord"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
    
```

Gilles Gosquière - LISIS/ Université Aix - Marseille - gilles.gosquier@isik.org

4- Géométries Géométries simples- Lignes

- Lines est représenté par LineStrings
- Série de points reliés par des lignes directes
- Deux possibilités
 - <gml:coordinates>
 - <gml:posList> (à préférer)
- X and Y sont séparés par une virgule et les coordonnées par des espaces.
- Dans posList, il y a une attribut de dimension qui permet de définir le nombre d'élément pour chaque point. (juste X et Y ou X, Y, Z);

```

<complexType name="LineStringType">
  <annotation>
    <documentation>
      A LineString is a special curve that consists of a single
      segment with linear interpolation. It is defined by two
      or more coordinate tuples, with linear interpolation
      between them. It is backwards compatible with
      the LineString of GML 2. GM_LineString of ISO
      19107 is implemented by LineStringSegment.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractCurveType">
      <sequence>
        <choice>
          <choice minOccurs="2" maxOccurs="unbounded">
            <element ref="gml:pos"/>
            <element ref="gml:pointRep"/>
            <element ref="gml:coord"/>
          </choice>
          <element ref="gml:coordinates"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
    
```

Gilles Gosquière - LISIS/ Université Aix - Marseille - gilles.gosquier@isik.org

4- Géométries Géométries simples- Polygone

- Polygons est la géométrie de base la plus complexe
- Un polygone est une aire fermée créée par une liste de segment
- Description par un élément `<gml:exterior>` du polygone qui contient un `<gml:LinearRing>` qui liste les points actuels formant la liste circulaire de points
- Comme un polygone peut contenir des trous (ou en terme de GML des listes circulaires intérieures), le polygone peut avoir 0 ou plusieurs éléments `<gml:interior>`
- Techniquement, on peut avoir un polygone sans liste circulaire extérieure, mais pour quelle utilité?

```
<gml:Polygon>
  <gml:exterior>
    <gml:LinearRing>
      <gml:coordinates>.....</gml:coordinates>
    </gml:LinearRing>
  </gml:exterior>
  <gml:interior>
    <gml:LinearRing>
      <gml:coordinates>.....</gml:coordinates>
    </gml:LinearRing>
  </gml:interior>
</gml:Polygon>
```

Gilles Gosquière - LISIS/ Université Aix- Marseille- gilles.gosquier@lisis.org

4- Géométries Géométries simples- Polygone

- Exterior/Interior schema
 - ▶ La seule différence entre le schéma pour l'intérieur et l'extérieur est le nom de l'élément.
 - ▶ Seul l'élément extérieur est incluí

```
<element name="exterior" type="gml:AbstractRingPropertyType">
  <annotation>
    <documentation>
      A boundary of a surface consists of a number of rings.
      In the normal 2D case, one of these rings is distinguished
      as being the exterior boundary. In a general manifold this
      is not always possible, in which case all boundaries shall be
      listed as interior boundaries, and the exterior will be empty.
    </documentation>
  </annotation>
</element>
```

Gilles Gosquière - LISIS/ Université Aix- Marseille- gilles.gosquier@lisis.org

4- Géométries Géométries simples- Polygone

• Polygon Schema

```
<complexType name="PolygonType">
  <annotation>
    <documentation>
      A Polygon is a special surface that is defined by a single
      surface patch. The boundary of this patch is coplanar and
      the polygon uses planar interpolation in its interior. It is
      backwards compatible with the Polygon of GML 2,
      GM_Polygon of ISO 19107 is implemented by PolygonPatch.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractSurfaceType">
      <sequence>
        <element ref="gml:exterior" minOccurs="0"/>
        <element ref="gml:interior" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Gilles Gosquière - LISIS/ Université Aix- Marseille- gilles.gosquier@lisis.org

4- Géométries Géométries simples- Polygone

LinearRing

- ▶ C'est l'élément qui contient les coordonnées
- ▶ Il y a au moins 4 points afin de former la liste circulaire, le premier et le dernier étant les mêmes.

```
<complexType name="LinearRingType">
  <annotation>
    <documentation>
      A LinearRing is defined by four or more coordinate
      tuples
      with linear interpolation between them; the first
      and last
      coordinates must be coincident.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractRingType">
      <sequence>
        <choice>
          <choice minOccurs="4"
            maxOccurs="unbounded">
            <element ref="gml:pos"/>
            <element ref="gml:pointRep"/>
          </choice>
          <element ref="gml:coordinates"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Gilles Gosquière - LISIS/ Université Aix- Marseille- gilles.gosquier@lisis.org

4- Géométries Géométries multiples

- Il est possible de grouper des géométries qui ne sont pas forcément contigues.
- Par exemple, un ensemble d'îles peuvent être considérés comme une seule entité géographique.
- Regroupement sous la forme d'une géométrie multiple (Multi Géométrie)
- Il y a 3 types de multi géométrie
 - ▶ Multipoint (composé de points)
 - ▶ MultiCurve (composé de courbes)
 - ▶ MultiSurface (composé de surfaces (polygones)).

Gilles Gosquière - LISIS/ Université Aix- Marseille- gilles.gosquier@lisis.org

4- Géométries Géométries multiples

MultiPoint Schema

```
<complexType name="MultiPointType">
  <annotation>
    <documentation>
      A MultiPoint is defined by one or more Points,
      referenced through pointMember elements.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractGeometricAggregateType">
      <sequence>
        <element ref="gml:pointMember"
          minOccurs="0"
          maxOccurs="unbounded"/>
        <element ref="gml:pointMembers" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

- Tous les schémas de géométrie multiple semblent similaires
- Nous ne montrons que le schéma pour multipoint
- Il contient simplement une liste de `<gml:pointMembers>` ou un élément `<gml:pointMembers>`.
- Un `pointMember` est juste une autre façon d'appeler un Point (Actuellement une substitution dans le schéma GML)
- Un élément `pointMembers` est juste un tableau d'éléments `pointMember` (ou points).

Gilles Gosquière - LISIS/ Université Aix- Marseille- gilles.gosquier@lisis.org

5- Features

- Un feature (ou objet géographique) correspond à la façon qu'utilise GML pour référer une chose à laquelle nous nous intéressons.
- C'est un complément à la géométrie liée.
- Dans l'exemple précédent, le point était un feature alors que la centerLineOf était juste une géométrie.
- Cette information est utile, mais ne représente pas forcément l'élément important.
- En GML, un feature n'inclut pas toujours une géométrie.
- Nous pouvons alors utiliser GML afin de stocker des informations liées à des objets qui n'ont pas de géométrie.

Gilles Gosquière - LISIS/ Université Aix-Marseille - gilles.gosquier@lisik.org

5- Features

- Un feature peut contenir des géométries multiples, pas comme les multi-géométries comme présentée précédemment, mais comme des façons différentes de dessiner le feature.
- Par exemple considérons la représentation d'une station de radio sur une carte d'une région. Nous pourrions la repérer
 - ▶ à l'aide d'un point (symbolisée comme une petite tour graphiquement)
 - ▶ comme un polygone qui montre l'étendue de la zone de transmission, ...
 - ▶ En GML, la solution est d'avoir une seule station radio avec plusieurs champs contenant de la géométrie (un point, un polygone, ou un ensemble de polygone pour représenter les immeubles, ...)
 - ▶ GML propose le concept de DefaultGeometry pour servir une géométrie par défaut pour un feature.
- Un schéma peut donc être défini pour votre propre profile d'application.
 - ▶ Il faut alors étendre <gml:AbstractFeatureType>
 - ▶ Décrit alors les propriétés requises que d'autres applications GML peuvent s'attendre à avoir.

Gilles Gosquière - LISIS/ Université Aix-Marseille - gilles.gosquier@lisik.org

5- Feature Abstract Feature Schema

```
<complexType name="AbstractFeatureType" abstract="true">
  <annotation>
    <documentation>
      An abstract feature provides a set of common properties,
      including id, metaDataProperty, name and description
      inherited from AbstractGMLType, plus boundedBy.
      A concrete feature type must derive from this type and
      specify additional properties in an application schema.
      A feature must possess an identifying attribute
      ('id' - 'fid' has been deprecated).
    </documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:boundedBy" minOccurs="0"/>
        <element ref="gml:location" minOccurs="0"/>
        <!-- additional properties must be specified
        in an application schema -->
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Gilles Gosquière - LISIS/ Université Aix-Marseille - gilles.gosquier@lisik.org

5- Feature Avoir plus d'un Feature

- Possibilité d'avoir plusieurs features
- GML fournit un élément de type FeatureCollection
 - ▶ Cet élément peut contenir d'autres featurecollection
 - ▶ Un featurecollection peut contenir un élément <gml:boundedBy> qui donne la boîte englobant la collection. Cela permet au client de décider si les features appartenant à une collection seront dans la zone écran à afficher sans avoir à la parcourir dans un premier temps.

Gilles Gosquière - LISIS/ Université Aix-Marseille - gilles.gosquier@lisik.org

5- Feature FeatureCollection Schema

- Le Schéma FeatureCollection est simplement une extension de AbstractFeatureTypeCollectionType.
- L'attribut substitutionGroup "gml:_Feature" est ce qui fait de featureCollection un feature.

```
<element name="FeatureCollection"
  type="gml:FeatureCollectionType"
  substitutionGroup="gml:_Feature"/>
<complexType name="FeatureCollectionType">
  <complexContent>
    <extension base="gml:AbstractFeatureCollectionType"/>
  </complexContent>
</complexType>
```

Gilles Gosquière - LISIS/ Université Aix-Marseille - gilles.gosquier@lisik.org

5- Feature AbstractFeatureCollectionType Schema

Classe de base de FeatureCollection.
Etend un type BoundedFeature avec une séquence d'éléments <gml:featureMember> ou un élément <gml:featureMembers>
Il y a juste une substitution pour les features (comme pointMember et Point dans MultiPoint).

```
<complexType name="AbstractFeatureCollectionType" abstract="true">
  <annotation>
    <documentation>
      A feature collection contains zero
      or more features.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="gml:BoundedFeatureType">
      <sequence>
        <element ref="gml:featureMember" minOccurs="0"
          maxOccurs="unbounded"/>
        <element ref="gml:featureMembers" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Gilles Gosquière - LISIS/ Université Aix-Marseille - gilles.gosquier@lisik.org

5- Feature Type BoundedFeature

- Bas de la pile de schémas
- BoundedFeature est une extension d'un type AbstractFeatureType.
- Il ajoute les éléments requis <gml:boundedBy> et les 4 autres éléments optionnels
- Façon qu'utilise GML pour forcer une featurecollections à avoir un élément boundedBy.

```
<complexType name="BoundedFeatureType" abstract="true">
  <annotation>
    <documentation>
      Makes boundedBy mandatory -
      used to build Feature Collections
    </documentation>
  </annotation>
  <complexContent>
    <restriction base="gml:AbstractFeatureType">
      <sequence>
        <element ref="gml:metaDataProperty" minOccurs="0"
          maxOccurs="unbounded"/>
        <element ref="gml:description" minOccurs="0"/>
        <element ref="gml:name" minOccurs="0"
          maxOccurs="unbounded"/>
        <element ref="gml:boundedBy"/>
        <element ref="gml:location" minOccurs="0"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

Gilles Gosquière - LISIS/ Université Aix - Marseille - gilles.gosquier@isik.org

6- Schéma d'application

- Schéma permettant de définir le feature "pont"
- Défini par "the international bridge building brotherhood" (www.ibbb.org) et basé sur le schéma GML défini par l'OGC.
- Le schéma définit deux éléments Ville (City) et Pont (Bridge)
- City est un FeatureCollection. Est utilisé pour contenir tous les autres éléments qui sont requis pour définir une ville
- Bridge est un feature et un type complexe qui étend le type AbstractFeature et définit 3 éléments
 - ▶ L'envergure
 - ▶ La hauteur
 - ▶ Un élément GML centreLineOf (qui est un alias pour LineString)
- Dans un schéma d'application réel, il sera aussi nécessaire de définir les autres éléments (routes, immeubles, ...)

Gilles Gosquière - LISIS/ Université Aix - Marseille - gilles.gosquier@isik.org

6- Schéma d'application

```
<xs:schema targetNamespace="http://www.ibbb.org"
  xmlns=""
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:cs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:import namespace="http://www.opengis.net/gml"
    schemaLocation="..IGML3.0.1base/feature.xsd"/>
  <xs:element name="City"
    type="gml:AbstractFeatureCollectionType"
    substitutionGroup="gml:_FeatureCollection"/>
  </xs:element>
  <xs:element name="Bridge"
    type="app:BridgeType"
    substitutionGroup="gml:_Feature"/>
  <xs:complexType name="BridgeType">
    <complexContent>
      <xs:extension base="gml:AbstractFeatureType">
        <xs:sequence>
          <xs:element name="span"
            type="xs:integer"/>
          <xs:element name="height"
            type="xs:integer"/>
          <xs:element ref="gml:centerLineOf"/>
        </xs:sequence>
      </xs:extension>
    </complexContent>
  </xs:complexType>
</xs:schema>
```

Envergure
Hauteur
Ligne centrale du pont

Gilles Gosquière - LISIS/ Université Aix - Marseille - gilles.gosquier@isik.org

7- Document d'instance

- Exemple d'instance conforme au schéma d'application construit précédemment.
- Il consiste en un élément de ville (featureCollection) qui a un élément englobant (boundedBy) qui permet de signaler l'extension d'une collection (pour un autre client par exemple)
- Il y a ensuite une liste de caractéristiques (featureMembers) chacune d'entre elles étant un pont
- Chaque pont a une envergure (span), une hauteur (height) et une ligne centrale (centerLineOf element)
- Un client peut utiliser ces informations afin de calculer sur la carte le point ou permettre de faire des analyses sur ces données

```
<City xsi:schemaLocation="http://www.ibbb.org City.xsd"
  <gml:boundedBy>
    <gml:Envelope>
      <gml:pos>100 200</gml:pos>
      <gml:pos>230 250</gml:pos>
    </gml:Envelope>
  </gml:boundedBy>
  <gml:featureMembers>
    <Bridge gml:id="bridge 1">
      <span>100</span>
      <height>200</height>
      <gml:centerLineOf>
        <gml:LineString gml:id="line 24">
          <gml:pos>100 200</gml:pos>
          <gml:pos>200 200</gml:pos>
        </gml:LineString>
      </gml:centerLineOf>
    </Bridge>
    <Bridge gml:id="bridge 2">
      <span>100</span>
      <height>200</height>
      <gml:centerLineOf>
        <gml:LineString gml:id="line 3">
          <gml:pos>130 250</gml:pos>
          <gml:pos>230 250</gml:pos>
        </gml:LineString>
      </gml:centerLineOf>
    </Bridge>
  </gml:featureMembers>
</City>
```

Gilles Gosquière - LISIS/ Université Aix - Marseille - gilles.gosquier@isik.org

7- Document d'instance

- Identifiants
 - ▶ Tous les objets GML doivent avoir un identifiant (ID).
 - ▶ Les ID doivent être uniques dans une instance de documents.
 - ▶ Si on a un feature route et un feature pont sur laquelle passe la route, il suffit juste d'avoir un objet géométrique qui sera utilisé dans les deux features.
 - ▶ Si mise à jour, il n'y a qu'une seule géométrie à mettre à jour. On est aussi ainsi sûr que la route passera toujours sur le pont.
- Cela rend utile le schéma suivant


```
<gml:centerOf xlink:href="#P1"/> This refers to a point with the ID P1
  somewhere else in the document
  <app:Tower gml:id="T1">
    <app:height>90</app:height>
    <gml:centerOf xlink:href="Transport.xml#P1"/>
  </app:Tower> Référence à un point dans un document séparé (Transport.xml) qui
  est dans le même répertoire que le document d'instance. Possibilité de faire
  référence à une géométrie dans un fichier différent sur une marche différente.
  <app:Tower gml:id="T1">
    <app:height>90</app:height>
    <gml:centerOf xlink:href="http://www.ccg.leeds.ac.uk/ian/Transport.xml#P1"/>
  </app:Tower> La géométrie (P1) est trouvée dans Transport.xml sur
  www.ccg.leeds.ac.uk.
```

Gilles Gosquière - LISIS/ Université Aix - Marseille - gilles.gosquier@isik.org

8- Exercice

- Lancer votre copie locale de GeoServer et demander le schéma topp:states et requête des (ou tous) les features de topp:states.
- Regarder le schéma et comparer le au document d'instance retourné par la requête getFeature
- Utiliser ce cours afin de comprendre le contenu de ce document.

Gilles Gosquière - LISIS/ Université Aix - Marseille - gilles.gosquier@isik.org