

Gittok Lecture Note

05 インスタンスの表現

太田守重

2014

応用スキーマとキットのインスタンスは どのようにXMLで表記するか.

応用スキーマを外部のシステムと交換するときは、XML文書の形をとる。そのための規則はどうか.

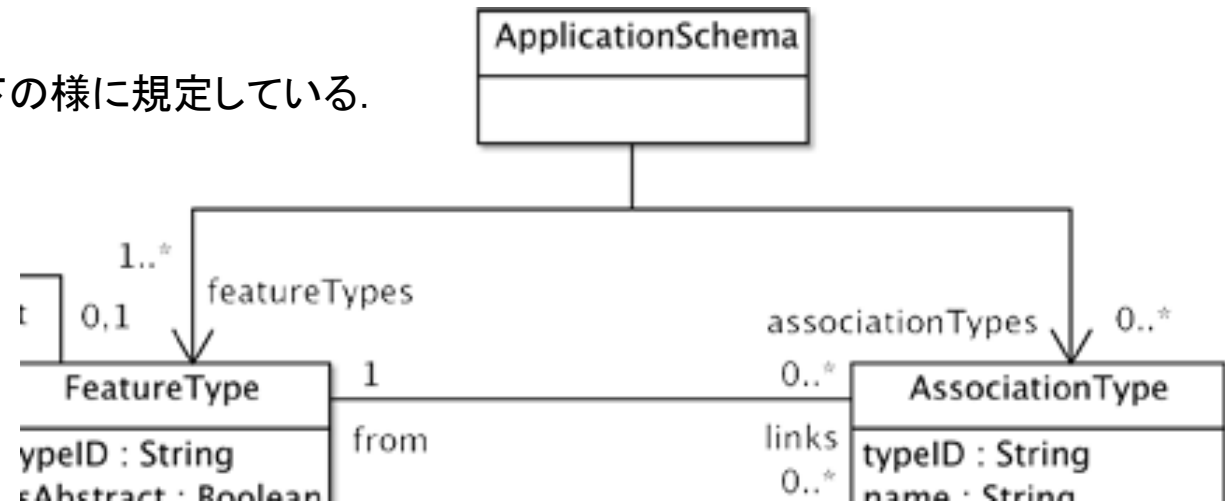
データセットを記述するための規則であるインスタンスモデルとは、どのようなものか.

まず、XML表現の基本的な方針を説明し、応用スキーマのXML表現及び応用スキーマに準拠して作られるデータセットのXML表現について学ぶ.

ただし、データセットのXML表現の学習は、空間スキーマなど、地物の属性が使用するデータ型や参照系に関する知識を学んでからでもいい.

応用スキーマのXML表記規則

UMLでは, 応用スキーマは以下の様に規定している.



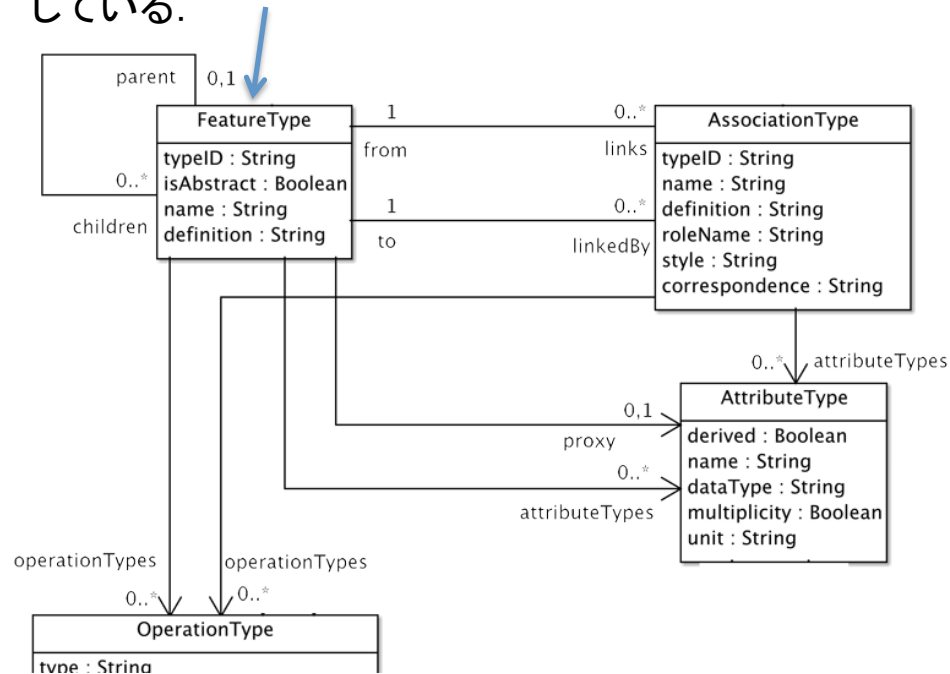
これに応じて, XML文書は, 右の様に記述する.
ApplicationSchema要素は応用スキーマを示す
ルート要素である.

featureTypes要素は地物型の要素を1つ以上
含まなければならないが, 応用スキーマの中
に関連が含まれない場合は, associationTypes
要素は, 関連型の要素をもたない.

```
<ApplicationSchema>
  <featureTypes>
    .....
  </featureTypes>
  <associationTypes>
    .....
  </associationTypes>
</ApplicationSchema>
```

地物型のXML表現規則

UMLでは、地物型は以下の様に規定している。



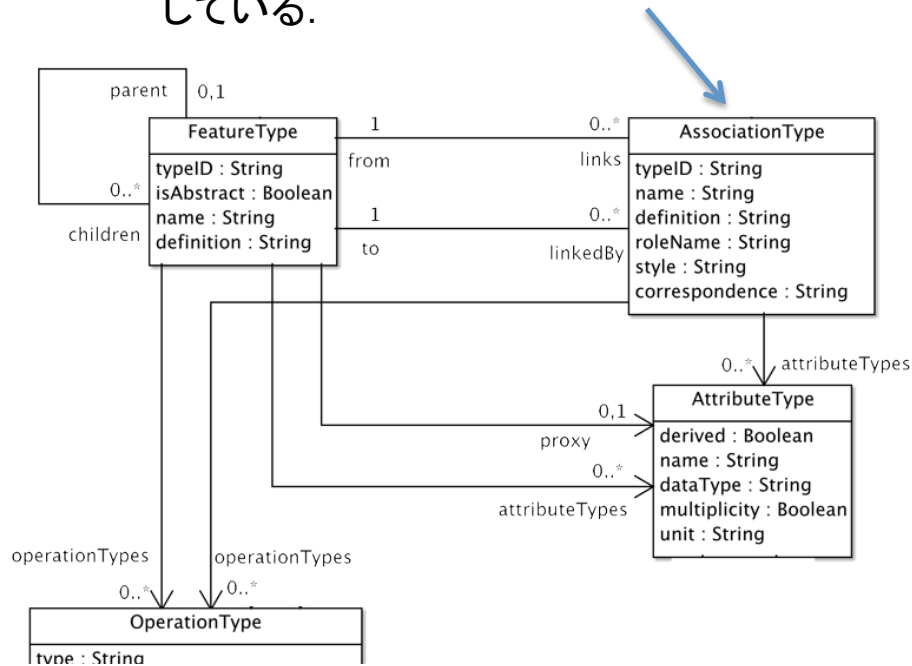
これに応じて、XML文書としては、下の様に記述する。

```

<FeatureType typeID="..." isAbstract="..." name="..."
  definition="..." >
  <proxy>...</proxy>
  <attributeTypes>...</attributeTypes>
  <operationTypes>...</operationTypes>
  <parent idref="..." />
  <children idref="..., ..., ..." />
  <links idref="..., ..., ..." />
  <linkedBy idref="..., ..., ..." />
</FeatureType>
  
```

関連型のXML表現規則

UMLでは、関連型は以下の様に規定している。

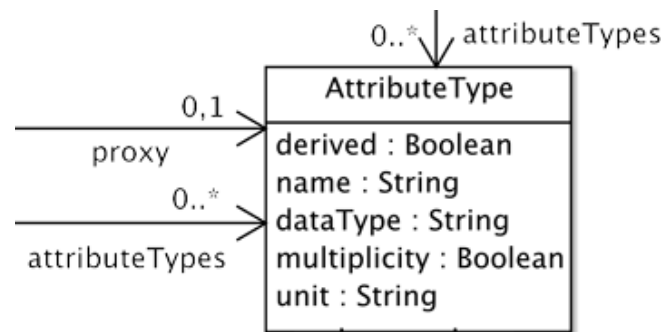


これに応じて、XML文書は、下の様に記述する.

```
<AssociationType typeId="..." name="..." definition="..."
    roleName="..." style="..." correspondence="...">
    <attributeTypes>...</attributeTypes>
    <operationTypes>...</operationTypes>
    <from idref="..." />
    <to idref="..." />
</AssociationType>
```

属性型のXML表現規則

UMLでは, 属性型は以下の様に規定している.



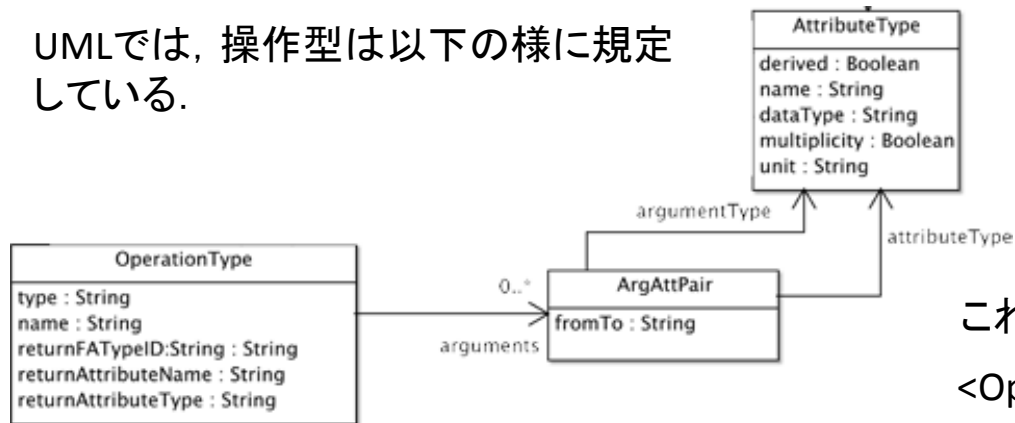
これに応じて, XML文書は, 下の様に記述する.

```
<AttributeType derived="..." name="..." dataType="..."  
multiplicity="..." unit="..." />
```

属性型のXML要素は, 子要素をもたない空要素である.

操作型のXML表現規則

UMLでは, 操作型は以下の様に規定している.



これに応じて, XML文書は, 下の様に記述する.

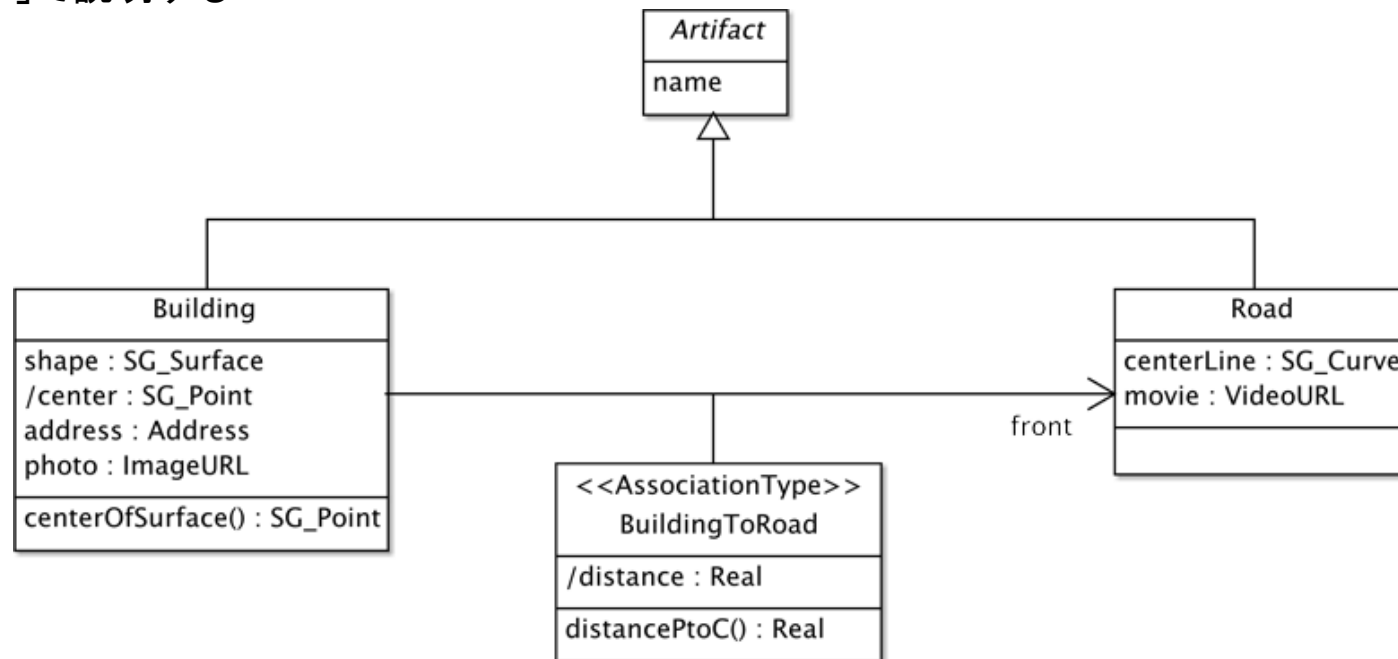
```
<OperationType type="..." name="..."
  returnFATypeID="..." returnAttributeName="..."
  returnAttributeType="...">
  <arguments>
    <ArgAttPair fromTo="...">
      <argumentType>
        <AttributeType .../>
      </argumentType>
      <attributeType>
        <AttributeType .../>
      </attributeType>
    </ArgAttPair>
    .....
  </arguments>
</OperationType>
```

応用スキーマのXML表現(例)

以下の応用スキーマのXML文書 (simpleSchema.xml) は, 以下の URL の Samples にある. これをダウンロードすれば, 内容を確認できる.

wwwc.dcn.s.ne.jp/~morishigeota/gittok/

なお, 属性のデータ型については, 「06 単純データ型と主題属性」, 「07 空間スキーマ」で説明する.

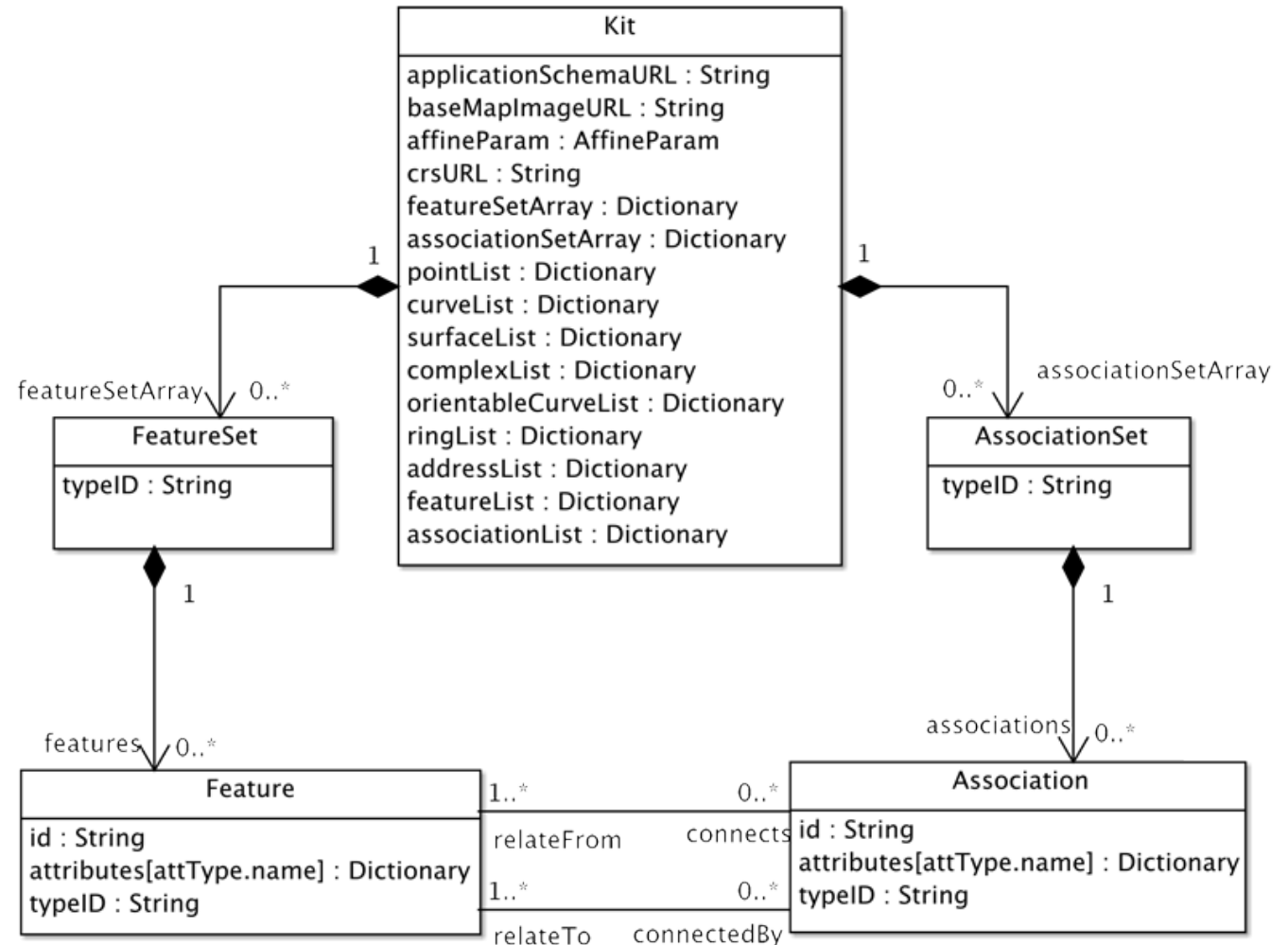


データセットの表現規則

(インスタンスモデル)

gittokでは、地物や関連のインスタンスの集りをkitと呼ぶ。キットは様々な属性をもつと共に、地物インスタンスの配列 (featureSetArray), 関連インスタンスの配列 (associationSetArray)を保存する。

地物インスタンスの配列は、同じ型をとる地物の集り (FeatureSet)である。地物インスタンスはFeatureというインスタンス型に従って実装される。関連セットの配列も同様の構造をもつ。



キットクラス

Kit
applicationSchemaURL : String
baseMapImageURL : String
affineParam : AffineParam
crsURL : String
featureSetArray : Dictionary
associationSetArray : Dictionary
pointList : Dictionary
curveList : Dictionary
surfaceList : Dictionary
complexList : Dictionary
orientableCurveList : Dictionary
ringList : Dictionary
addressList : Dictionary
featureList : Dictionary
associationList : Dictionary

キットは地物と関連のインスタンスの集り、一般的に言えば空間データセットである。キットクラスは、そのルートとして、以下に示す属性をもつ。

applicationSchemaURL: キットが準拠する応用スキーマのURL

baseMapImageURL: 幾何属性を取得するときに使用した計測基図のURL

affineParam: 幾何属性の画面座標から地上の平面座標を求めるパラメータ (「09 参照系」を参照)

crsURL: 地上の平面座標と緯度経度を相互に換算するための座標参照系のパラメータを記述するファイルのURL (「09 参照系」を参照)

featureSetArray: 地物集合の配列

associationSetArray: 関連集合の配列

pointList: 点の集り

curveList: 曲線の集り

surfaceList: 曲面の集まり

complexList: 幾何複体の集り

orientableCurveList: 有向曲線の集り

ringList: 閉曲線(輪)の集り

addressList: 住所の集まり

featureList: 地物の集り

associationList: 関連の集まり

注) 点, 曲線, 曲面, 幾何複体, 有向曲線及び閉曲線については「07 空間スキーマ」参照

キットのXML表記の骨格

```
<Kit applicationSchemaURL="..." baseMapImageUrl="..." crsURL="...">
  <affineParam>
    .....
  </affineParam>
  <featureSetArray>
    <FeatureSet typeId="...">
      <features idref="...,...,..." />
    </FeatureSet>
    .....
  </featureSetArray>
  <associationSetArray>
    <AssociationSet typeId="...">
      <associations idref="...,...,..." />
    </AssociationSet>
    .....
  </associationSetArray>
  <pointList>
    .....
  </pointList>
  <curveList>
    .....
  </curveList>
```

```
    <surfaceList>
      .....
    </surfaceList>
    <complexList>
      .....
    </complexList>
    <orientableCurveList>
      .....
    </orientableCurveList>
    <ringList>
      .....
    </ringList>
    <addressList>
      .....
    </addressList>
    <featureList>
      .....
    </featureList>
    <associationList>
      .....
    </associationList>
  </Kit>
```

地物と関連のインスタンス

Feature	1..*	0..*	Association
id : String attributes[attType.name] : Dictionary typeID : String	relateFrom	connects	id : String attributes[attType.name] : Dictionary typeID : String
	1..*	0..*	
	relateTo	connectedBy	

id: 地物インスタンスの識別子

attributes[attType.name]: 属性型の名前を
キーとする属性値の配列(連想記憶)

typeID: このインスタンスの型の識別子

connects: この地物が関連元になって繋がる
関連インスタンスへの参照. 複数になる場
合, それぞれの関連の型は異なる.

connectedBy: この地物が関連先になって繋
がる関連インスタンスへの参照. 複数になる
場合, それぞれの関連の型は異なる.

id: 関連インスタンスの識別子

attributes[attType.name]: 属性型の名前を
キーとする属性値の配列(連想記憶)

typeID: このインスタンスの型の識別子

relateFrom: 関連元になる地物インスタンス
への参照

relateTo: 関連先になる地物インスタンスへ
の参照

地物のXML表記

地物インスタンスは, idをもち, 対応する地物型のid (typeID)をもつ. 属性は単純データ型をとるものはFeature要素のXML属性とし, 複合データ型をとるものは子要素にして, iderefによる参照を行う.

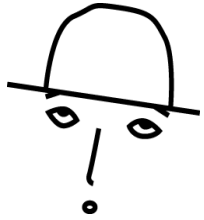
さらに, 関連インスタンスに連結する場合は, connects要素, 関連される場合は connectedBy要素を使って関連インスタンスを参照する.

```
<Feature id="..." typeID="..." 属性名1="..." 属性名2="..." .....>
  <属性名 idref="..." />
  <属性名 ideref="..." />
  ...
  <connects idref="..., ..." />
  <connectedBy idref="..., ..." />
</Feature>
```

Feature要素のXML属性になるのは属性のデータ型が以下のもののいずれかの場合である.

Integer,
Real,
Bool,
CharacterString,
URL,
ImageURL,
VideoURL,
SoundURL

属性が幾何データ型のいずれか, または住所型をとる場合は, 子要素にして, 間接参照させる.

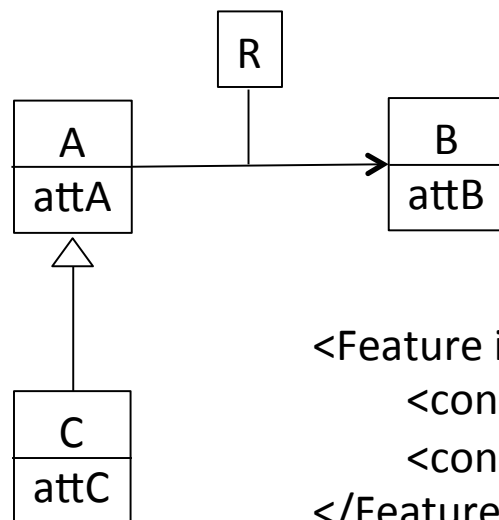


地物に親があったら、属性や操作はどうなるの？

継承する属性と操作は、
インスタンスの属性と操作に併合される。

でも、gittok では、関連は継承しません。

(例)



Aのインスタンスから継承



```
<Feature id="c01" typeID="C" attC="..." attA="..." >  
  <connects/>  
  <connectedBy/>  
</Feature>
```

関連のXML表記

関連インスタンスは, idをもち, 対応する関連型のid (typeID)をもつ. 属性は単純データ型をとるものはAssociation要素のXML属性とし, 複合データ型をとるものは子要素にして, iderefによる参照を行う.

さらに, 関連元になる地物インスタンスはrelateFrom要素, 関連先はrelateTo要素を使って地物インスタンスを参照する.

```
<Association id="..." typeID="A" 属性名1="..." 属性名2="..." .....>
  <属性名 idref="..." />
  <属性名 ideref="..." />
  ...
  <relateFrom idref="..., ..."/>
  <relateTo idref="..., ..."/>
</Feature>
```

関連の対応関係 (correspondence) が
1:1, 1:n のときは, relateFromの参照数は1
m:1 のときは, relateToの参照数は1