

Gittok Lecture Note

# 03 一般地物モデルと応用スキーマ

太田守重

2014

UMLクラス図の対象は

実世界に生起する地理的な現象に限らない.

でも私たちは地物及びその間の関係を応用スキーマとして記述する.

そのためには, 特有のルールがほしい.

このルールを一般地物モデル (GFM: General feature Model) と呼ぶ.

ではGFMとはなにか?

GFMに従って作られる応用スキーマとはなにか?

# 抽象化の枠組み

実世界の現象を抽象した概念を地物という。地物は型として、またはインスタンスとして表現される。

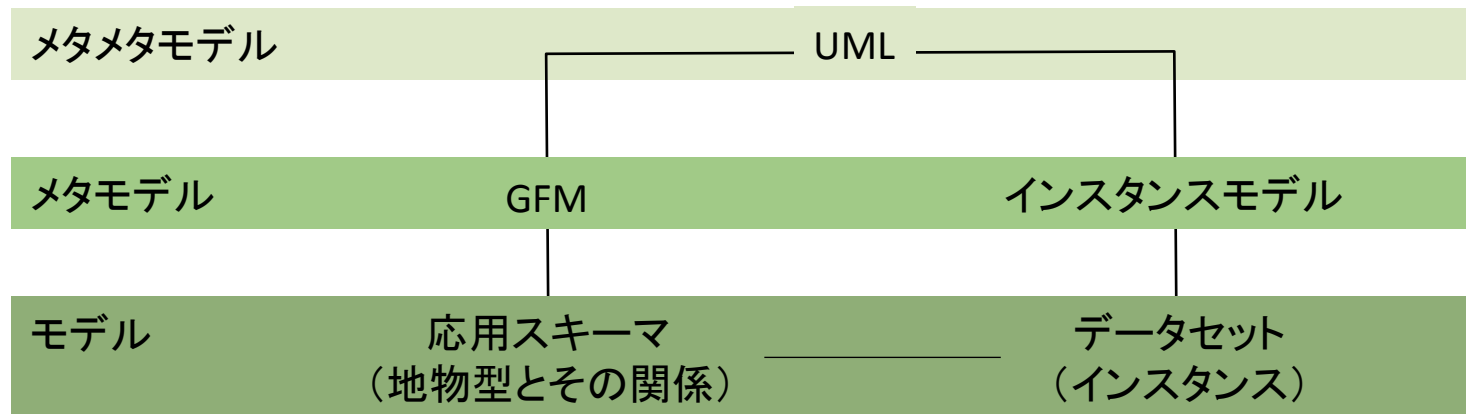
地物型はインスタンスのメタモデル。

型及びそれらの関係の記述は応用スキーマ。

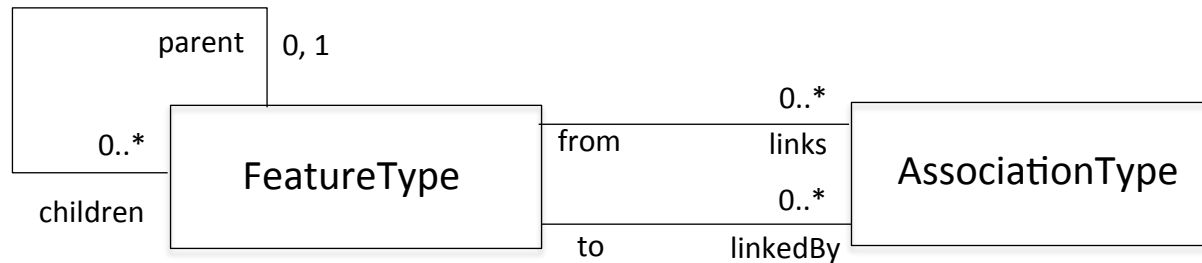
応用スキーマの記述規則(メタモデル)は一般地物モデル (GFM: General Feature Model)。

インスタンスの記述規則(メタモデル)はインスタンスモデル。

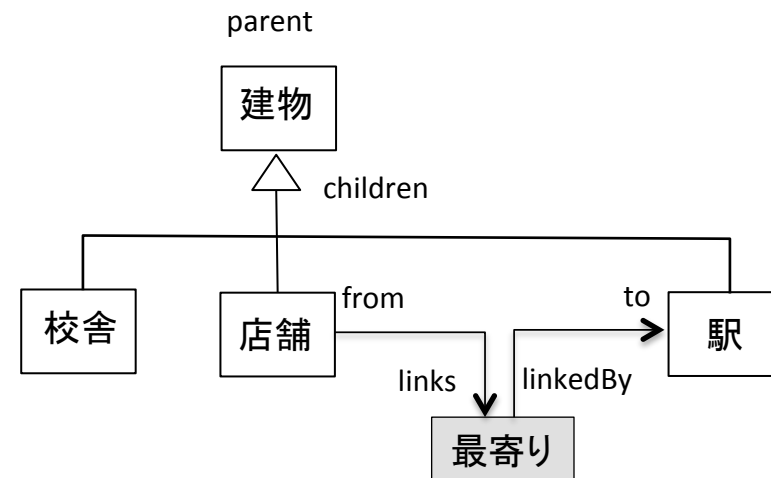
GFM及びインスタンスモデルの記述はスキーマ言語 (UML)で行う。



## 地物型と関連型(GFMの骨格)

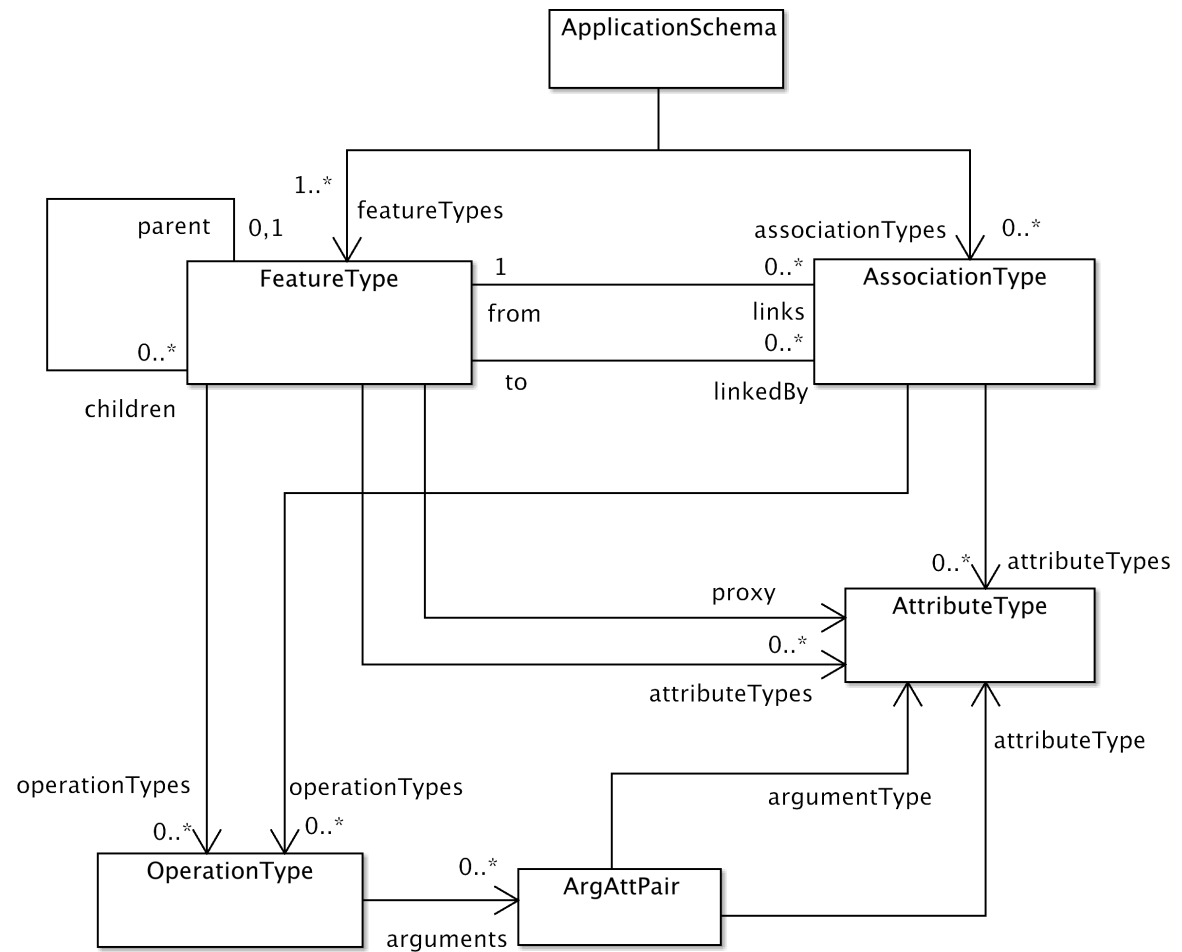


地物型は、親子の関係をもち、子は親の性質を受け継ぐ。これを継承関係という。例えば校舎は建物である、つまり建物を親とし、建物の性質をもつ。次に、地物型は他の地物型と関連することがある。例えば、レストランに行くために「最寄り」の駅を教わることがあるが、そのレストランと、駅は「最寄り」という関連をもつ。そこで店舗と駅の間に「最寄り」という関連型を定義する。



# gittok の一般地物モデル(GFM)

GFMの概略を右に示す. 応用スキーマ(ApplicationSchema) は地物型 (FeatureType) と関連型 (AssociationType) からなる. 地物型どうしには継承関係がありうる. 地物型は属性 (AttributeType) をもつことがあり, その中の幾何属性は, グラフィック表示における地物の代理 (proxy) となることがある. 地物型は操作 (OperationType) をもつことがあり, 操作は属性を演算の引数 (arguments) として使用し, 演算結果は属性になる. 操作の引数と属性の対応付けは引数と属性の対 (ArgAttPair) で行う.



Proxy is a spatial geometry used as representation of a feature on the map.

# 応用スキーマ (ApplicationSchema)

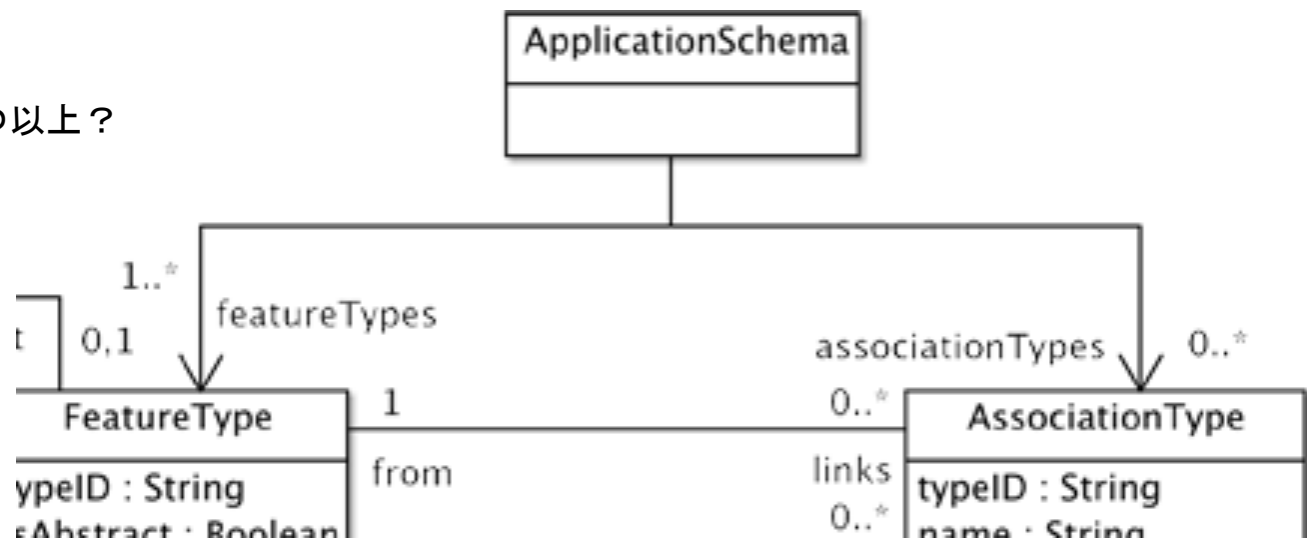
GFMのこの部分は、応用スキーマについて規定している.

応用スキーマは、特定の応用分野のために作られるスキーマで、複数の地物型 (featureType)と、地物型どうしの関連を記述する複数の関連型 (associationType)からなる.  
関連型は無い場合もある.



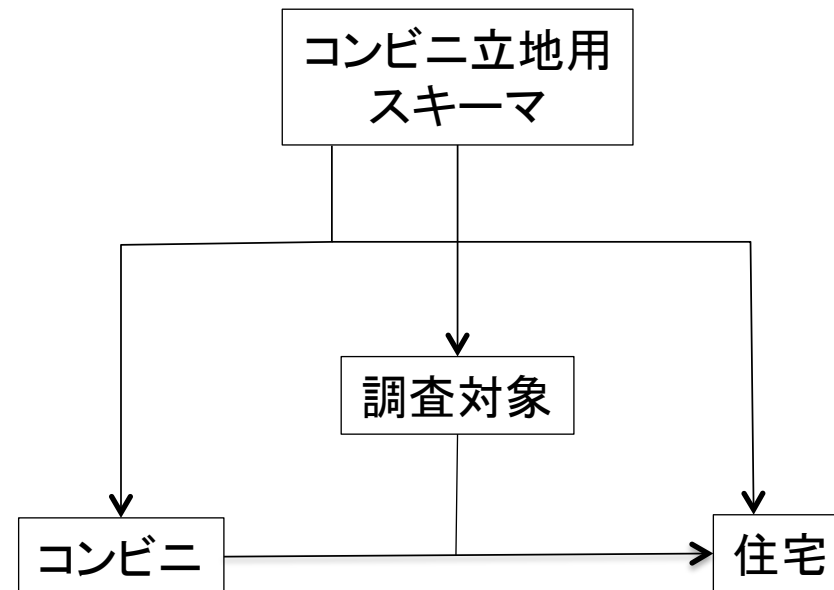
複数って2つ以上？

ここでは、複数という言葉は、一つ以上という意味で使います.

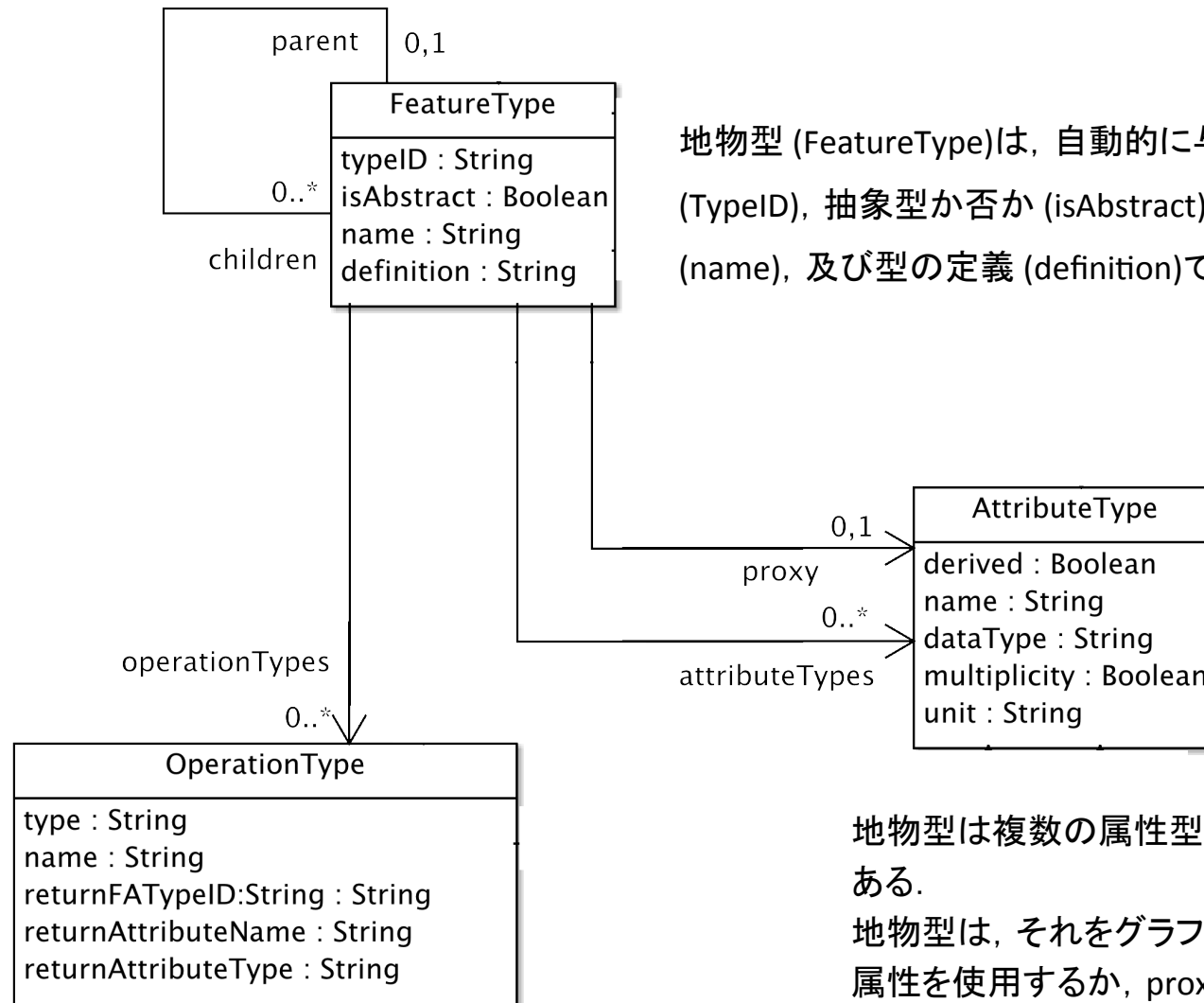


例えば、住宅地にコンビニを立地する場合、  
候補地の選定に使われる地物は対象地域内の  
住宅とコンビニ、とする。

その応用スキーマは  
地物型として  
「コンビニ」と「住宅」を含み  
関連型として  
コンビニと住宅を関連づけ  
る「調査対象」を含む。



# 地物型



地物型 (FeatureType)は、自動的に与えられる型のID (TypeID), 抽象型か否か (isAbstract), 型の名前 (name), 及び型の定義 (definition)で記述される.

もつことがある？



もたないこともありうるという意味.

地物型は複数の属性型 (attributeTypes)をもつことがある.

地物型は、それをグラフィック表示するときに、どの属性を使用するか、proxyで示す. Proxyは幾何属性でなければならない.

地物型は、複数の操作型 (operationTypes)をもつことがある.



# 属性型

地物や関連がもつ、固有の静的な特性を属性という.

AttributeType
derived : Boolean name : String dataType : String multiplicity : Boolean unit : String

属性型 (AttributeType)は、派生属性かどうか (derived), 名前 (name), その属性のデータ型 (dataType), 多重度 (multiplicity), そして属性が数値になるときは、必要に応じて計量単位 (unit) を記述する.

派生属性とは、ここでは、操作の戻り値になる属性を指す.

データ型は、空間属性型、時間属性型、場所属性型、及び主題属性型に分かれる.

空間属性とは、地物の空間的な性質.

時間属性とは、地物の時間的な性質.

場所属性は、実世界中の地物場所と仮想世界の場所.

主題属性はその他の属性、例えば名称、延長、階層、重さなどを表現する単純データ型をとる。

例えばコンビニならば、

コンビニ
/位置:SG_Point

コンビニは地物型で、  
位置という属性をもつ。  
この属性は派生属性であり、  
別の場所にある操作で値が求められて  
結果がここに入れられる。

住宅ならば

住宅
形状:SG_Surface 階数:Integer

住宅は地物型で、  
形状と階数を属性とする。  
これらは、コンビニの位置を求める  
操作で使われる。



SG\_Point

SG\_Surface

Integerってなに？

点, 面, 整数のこと。

ここでは階数は主題属性として  
整数で表すし, 位置や形状  
空間属性として点や面であら  
わすということです。

点と面については講義用スラ  
イド07, 整数については06で  
説明します。

# 空間属性

地物の空間的な性質を空間属性という。  
空間的な性質とは、地物の形状と、  
形状同士の隣接・境界関係のことである。

形状は、地球上の位置座標及びその組み合わせからなる。  
最も基本的な形状は、点、線、面である。空間属性は、それらの組み合わせでもいい。前者は幾何プリミティブ、後者は幾何複体と呼ばれる。

互いに接触する地物同士は、境界関係をもつ。面同士は線を境界として共有する。線同士は交差点を介して隣接する。

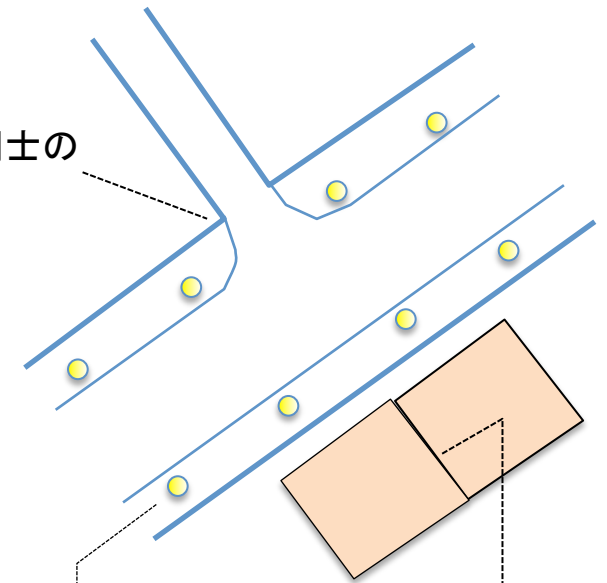
GISでは、スクリーンに表示された図形を指定して、地物インスタンスを検索する。従って、表示される図形は、地物の空間属性であるとともに、地物インスタンスの代理 (proxy) でもある。

3本の線同士の  
交差点

proxy

面の境界線

街路樹  
樹種: イチョウ  
管理者: \* \* 市  
設置年度: 1990年  
位置: p1



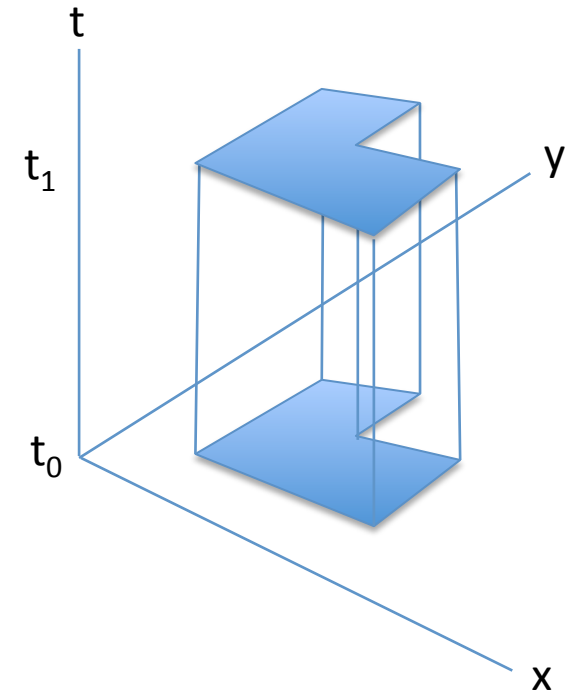
# 時間属性

地物の時間的な性質を時間属性という。  
時間的な性質とは、地物の発生から消滅までの期間、  
及び期間同士の隣接・境界関係のことである。  
なお、発生と消滅の時点が一致する場合は、瞬間という。

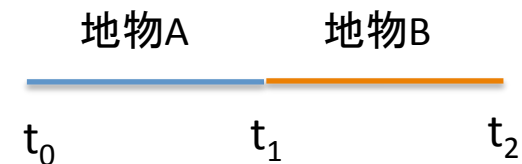
期間は、開始時点から終了時点までの間隔のことである。  
時間属性は、瞬間と期間の組み合わせでもいい。瞬間や期間は時間幾何プリミティブ、その組み合わせは時間幾何複体と呼ばれる。

終了時点と開始時点が一致する地物同士は、時間境界関係をもつ。期間同士は瞬間を介して隣接(置換)する。

GISでは、ユーザインタフェースとなるスクリーンに表示された期間や瞬間を指定して、地物インスタンスを検索することができる。従って、表示される時間幾何は、地物の時間属性であるとともに、地物インスタンスの代理(proxy)でもある。



地物の存在は期間  $(t_0, t_1)$  で示すことができる。



gittokは、現状は、時間属性の実装は行っていない。

# 場所属性

## 場所

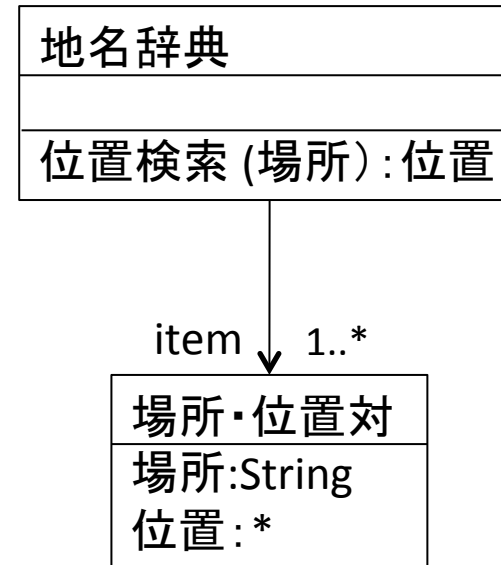
空間上の位置を間接的に表現する識別子

## 地名辞典 (Gazetteer)

場所と位置の対を使って、場所をキーにして位置を応える.

空間は地球の実空間と、ネットの中などの仮想空間がある. 実空間の場所は、住所やランドマークの名称で示され、位置は、地上の平面座標や緯度経度である.

仮想空間上の場所は、URLを位置とする語句 (My Home Page) で示される. しかし実世界のものを説明するサイトのURLを場所属性と考え、地球上の座標を位置にすることによって、仮想空間と実空間の対応関係を示す地名辞典を作ることができる.



<a href="url">My HomePage</a>

gittokは、現状は、地名辞典の実装は行っていない.

# 主題属性

地物を説明する短い語句

主題属性のデータ型の例,

**数 (Number)**

長さ、幅、面積、高さ、階数, ...

**文字列 (Character String)**

名前、表題、説明、種別, ...

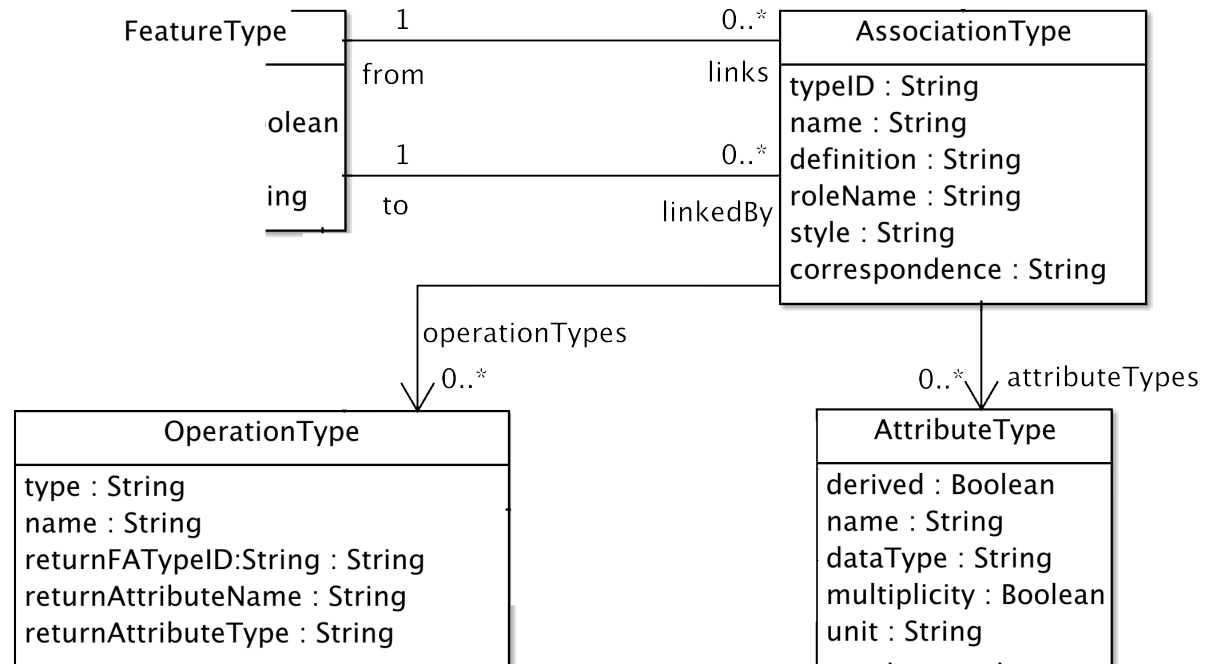
**ブール値 (Boolean)**

yes/no, on/off, ...

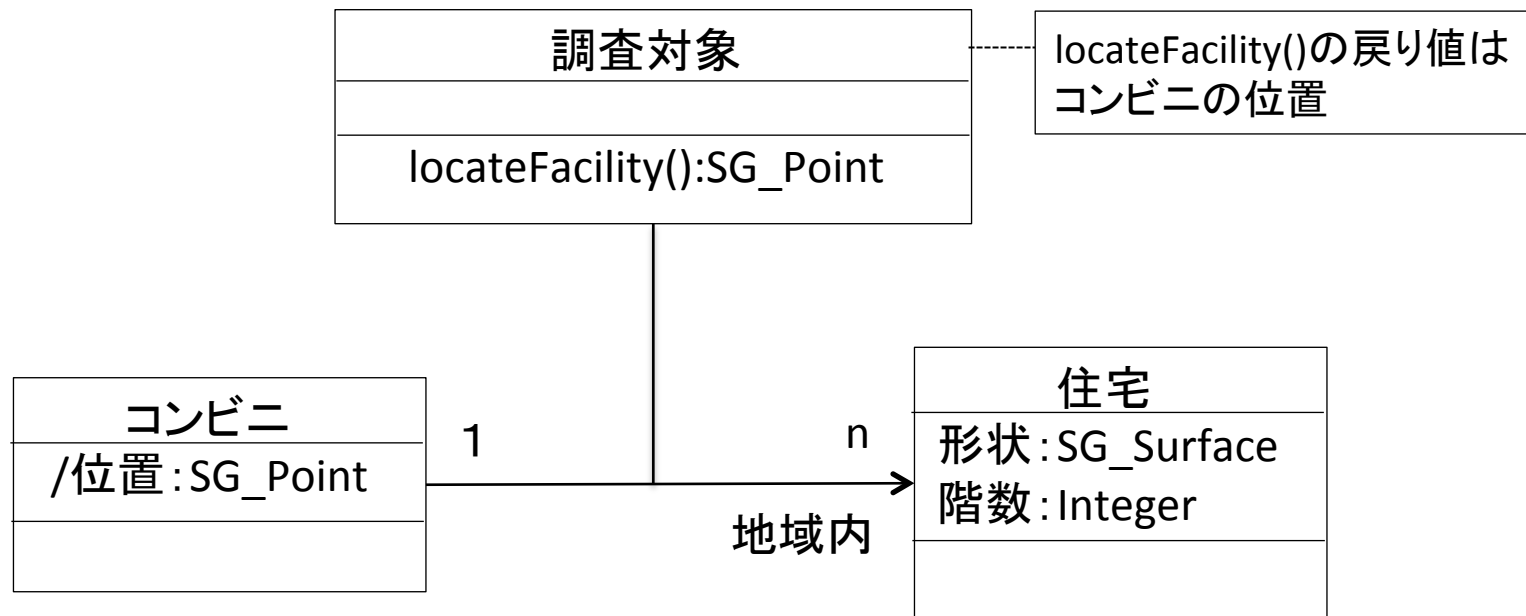
# 関連型

関連型は, 自動的に決まるID (typeID), 名前 (name), 定義 (definition), 関連先 (to)の地物がどのような役割 (roleName) をもつか, 普通の関連か合成かを示す関連のスタイル(style), そして, 関連元と関連先の地物インスタンスが1:1, 1:n, m:1, m:nのいずれかを示す対応関係 (correspondence)によって記述する.

また, 関連型は地物型と同様に, 独自の属性及び操作をもつことができる.



例えば、関連「調査対象」の場合、  
コンビニと住宅を関連させると共に、  
操作 locateFacility() を起動させて住宅の属性を使ってコンビニの位置を  
求めることができる(詳細については、操作の説明参照)。





# 属性と関連

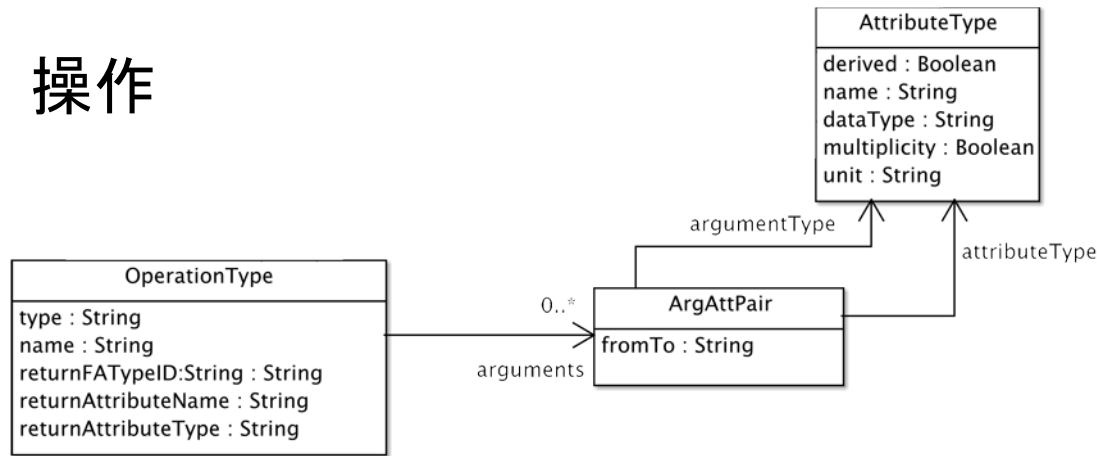
属性の型はデータ型のみで、地物型になることはない。



地物の関連先は地物型のみで、データ型になることはない。



# 操作



地物の操作を定義するためには、まず、既存の操作群の中から、①操作を選択し、その操作の戻り値が、地物ないし関連のどの属性と対応するかを指定し、②操作のどの引数と、③地物のどの属性が対応するかを指定する。

これができれば、操作型を実行して、戻り値を地物インスタンスの属性値とすることができる。

## OperationType

- ① type:Center  
name: locateFacility  
returnFATypeID: (コンビニ型のID)  
returnType: SG\_Point  
returnAttribute: 位置  
arguments
  - fromTo: to
- ② argumentType(0)
  - name: surfaces
  - dataType: SG\_Surface
  - multiplicity: true
  - unit: nullargumentType(1)
  - name: stories
  - dataType: Integer
  - multiplicity: true
  - unit: null
- ③ attributeType(0)
  - name: 形状
  - dataType: SG\_Surface
  - multiplicity: false
  - unit: nullattributeType(1)
  - name: 階数
  - dataType: Integer
  - multiplicity: false
  - unit: null

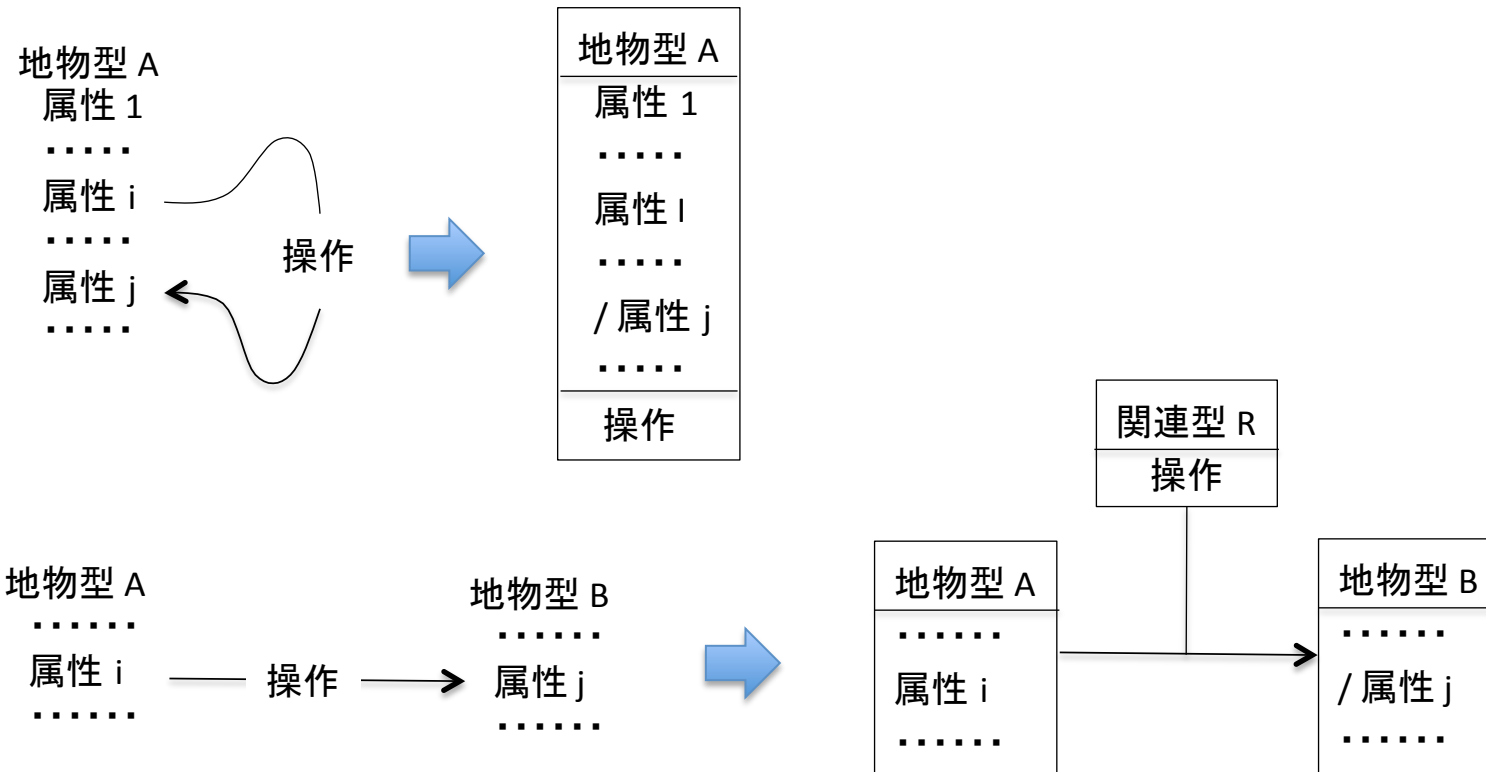
操作は、別の言葉で言えば関数である。

関数は、一つ以上の定義域の一つ以上の元を使って一つの値域の中の一つの元を求める。

応用スキーマの場合、定義域や値域が、同じ地物ないし関連の中にある属性になる場合は、

操作はその地物ないし関連がもつ。

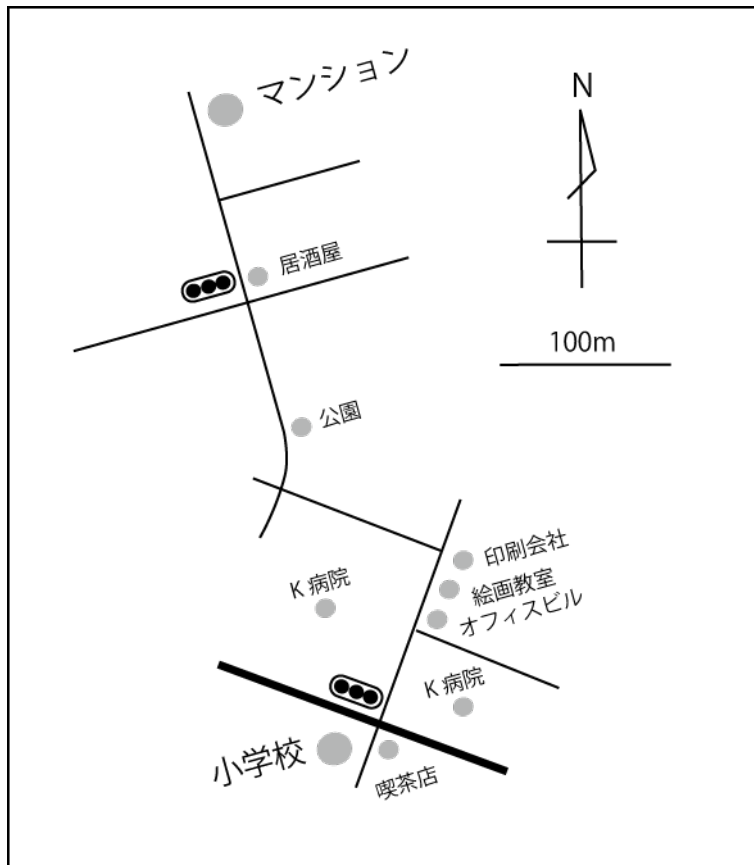
定義域と値域が2つの地物にまたがる場合は、関連に操作が置かれる。



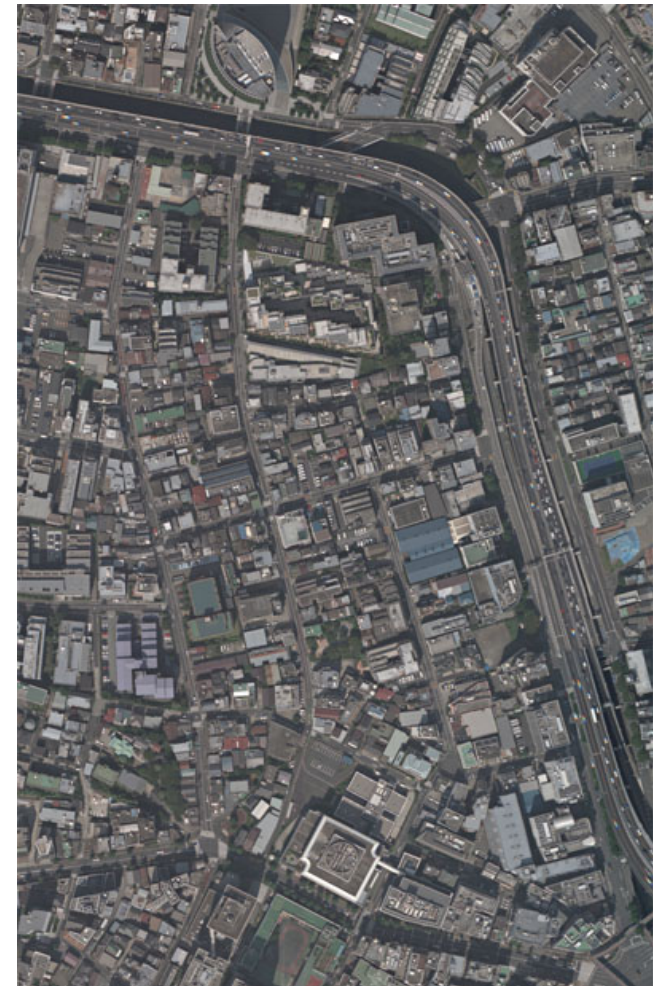
# 論議領域(Universe of discourse)

興味を引くものすべてを含む、実世界又は仮想世界の物の見え方 (view)  
その設定から、応用スキーマ作成が始まる。

例えば『自宅から小学校までの経路』

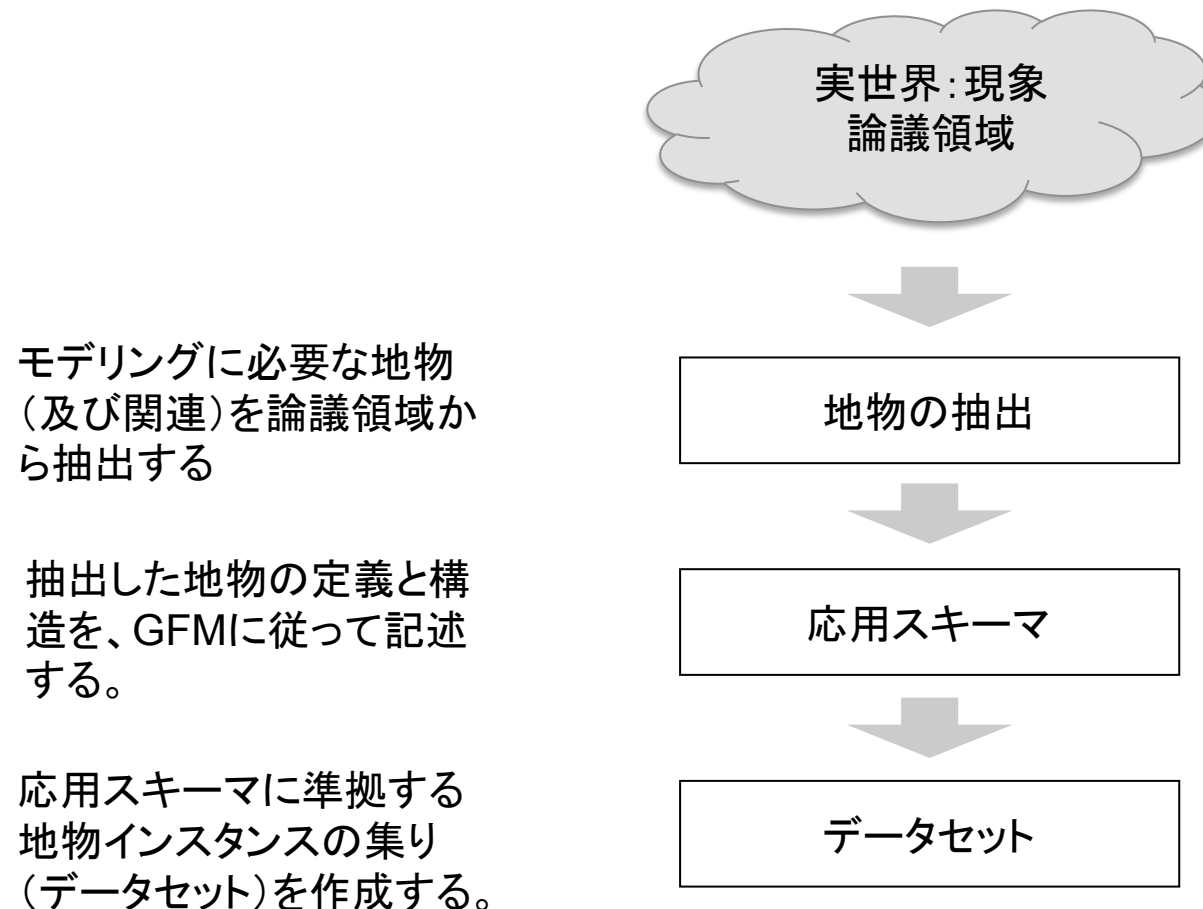


対応する  
実世界



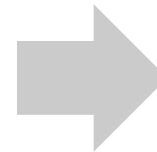
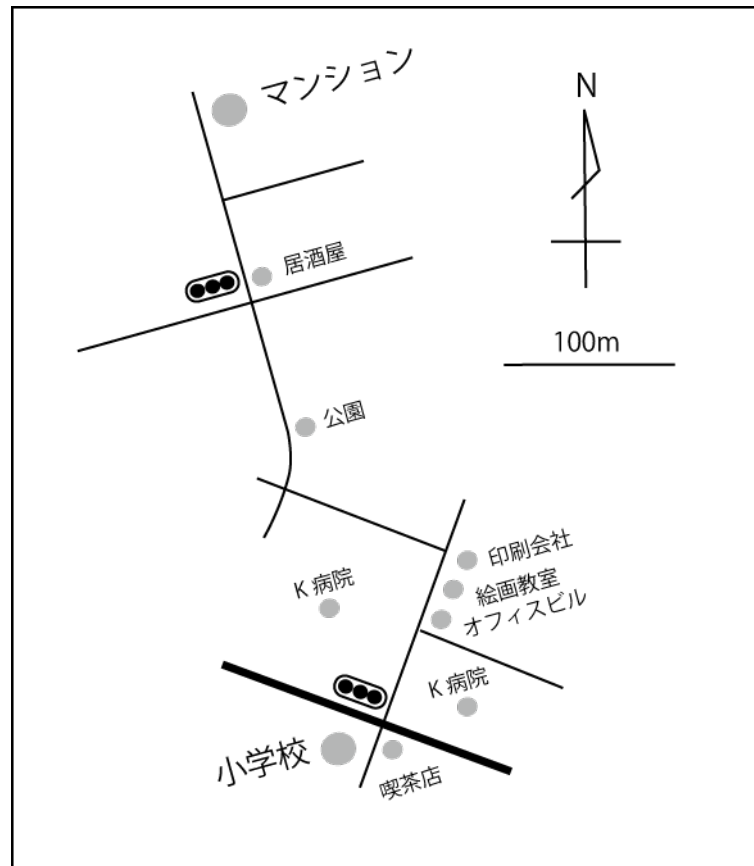
# 応用スキーマ

GFMが示すルールに従って、応用目的ごとに作られるスキーマ



## 地物の抽出

現場や資料などを観察して、どのようなモノがあるか列挙する.

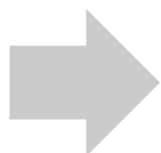


マンション  
居酒屋  
公園  
K病院  
印刷会社  
絵画教室  
オフィスビル  
喫茶店  
小学校  
道路  
信号

## 地物型と関係の検討

列挙したそれぞれのものがどのような種類のものか, 分類する.

マンション  
居酒屋  
公園  
K病院  
印刷会社  
絵画教室  
オフィスビル  
喫茶店  
小学校  
道路  
信号



### 地物

建物(種類, 位置)

公園(位置)

道路(形状)

信号(位置)

上記のものは総て,  
人工構造物である。

### 地物どうしの関連

道路は人工構造物に  
接している。

# UMLクラス図の設計

分類された概念を指す言葉は、地物型になる。そして、その間に発見された相互関係は関連や継承になって、クラス図の設計が行われる。

