

# I sistemi per la gestione di basi di dati geografiche

*Regole di corrispondenza tra GeoUML e  
il modello logico di riferimento di  
Simple Features Access*

Maggio 2007

Alberto Belussi

# Progettazione logico/fisica di una base di dati geografica

La progettazione logico/fisica di una base di dati geografica definisce le strutture di memorizzazione della componente spaziale e alfanumerica dell'informazione in essa contenuta.

La progettazione logico/fisica segue la progettazione concettuale. Essa richiede quindi che lo schema concettuale dei dati sia già stato redatto. Esso è infatti il punto di partenza della progettazione logico/fisica.

L'altro essenziale prerequisito è costituito dalla scelta di un sistema per gestione di basi di dati geografiche (GEO-DBMS). In particolare, deve essere noto il modello dei dati del sistema.

# Criteri generali della progettazione logico/fisica

- Creare una struttura normalizzata per ridurre la ridondanza dei dati e quindi le anomalie che si possono verificare in fase di aggiornamento dei dati.
- Gestire i casi di geometria condivisa in modo esplicito nella struttura dei dati (attraverso la definizione di strati topologici) quando possibile.
- Rappresentare il più possibile nella struttura fisica dei dati i vincoli di integrità spaziali significativi espressi nello schema concettuale.
- Creare una struttura che sia fruibile per l'uso applicativo; questo criterio risulta ovviamente in contrasto con l'esigenza di una struttura normalizzata introdotta dal primo criterio.

# Mapping da GeoUML verso un modello logico geo-relazionale

## **Classi Normali e Associazioni**

Per quanto riguarda la trasformazione delle classi normali senza attributi geometrici e delle associazioni tra classi valgono le regole previste per il mapping tra il modello Entità-Relazioni e il modello relazionale nelle basi di dati tradizionali.

Poiché il modello logico/fisico è geo-relazionale, è sempre possibile rappresentare relazioni (o tabelle) in una base di dati geo-relazionale. Quindi sarà possibile tradurre le classi normali e le associazioni presenti in uno schema GeoUML in relazioni di una base di dati geo-relazionale.

# Mapping da GeoUML verso un modello logico geo-relazionale

## **Attributi geometrici e vincoli strutturali**

I costrutti del modello GeoUML che contengono riferimenti alla geometria, vengono tradotti in un modello logico/fisico attraverso l'introduzione nello schema logico di **attributi geometrici** nelle tabelle e di **strati topologici (se il sistema lo consente)**.

***D:** Cosa sono esattamente gli strati topologici nel modello logico? Quanti **strati topologici** devono essere creati in una base di dati territoriale? Infine quali **attributi geometrici** vanno messi sullo stesso strato?*

*Esistono diverse possibili alternative!*

# Osservazioni sull'uso degli strati in uno schema logico/fisco

Rappresentare un insieme di valori geometrici in un'unica struttura topologica (strato topologico) all'interno di una base di dati geo-relazionale implica quanto segue:

- la gestione esplicita della geometria condivisa fra i valori geometrici contenuti nello strato;
- la possibilità di implementare i vincoli geometrici nella struttura dati dello strato;
- una maggiore facilità di aggiornamento dei dati contenuti nello strato, in particolare dove esistono vincoli di integrità spaziale da rispettare;
- la necessità di considerare l'intero strato per ogni accesso ai dati in esso contenuti.
- la possibilità di interrogare con maggiore efficienza i dati contenuti nello strato, in quanto le operazioni più potenti dei sistemi agiscono di solito intra-strato.

# Suddivisione in strati logici degli attributi geometrici di uno schema GeoUML

Tenendo presente le osservazioni riportate in precedenza si possono considerare i tre seguenti criteri:

- generare uno strato topologico nello schema logico (*strato logico*) **per ogni strato topologico dichiarato nello schema concettuale** (*strato concettuale*).
- porre sullo stesso *strato logico* tutti gli attributi geometrici legati direttamente o indirettamente a livello concettuale (attraverso vincoli strutturali) allo *strato concettuale* corrispondente.
- generare uno strato topologico nello schema logico (*strato logico aggiuntivo*) **per rappresentare gli attributi geometrici delle classi legate da vincoli strutturali (inclusi tratti e sottoaree) non legate a strati concettuali**.
- porre sullo stesso *strato logico aggiuntivo* tutti gli attributi geometrici legati direttamente o indirettamente a livello concettuale attraverso vincoli strutturali agli attributi che hanno dato vita allo strato logico aggiuntivo
- porre preferibilmente sullo stesso strato i valori degli attributi geometrici che vengono **utilizzati insieme nelle elaborazioni realizzate con maggior frequenza** dalle applicazioni.

# Le regole di mapping tra GeoUML e il modello Simple Feature Access

Classe  $C_1$  contenente gli attributi alfanumerici:

$A_1, \dots, A_n$

## Mapping degli attributi alfanumerici normali

Per la classe  $C_1$  viene creata una tabella  $C_1$  che contiene tanti attributi quanti sono gli attributi alfanumerici della classe  $C_1$ . La tabella  $C_1$  ha un attributo aggiuntivo detto anche *FID* (feature ID) di tipo integer e con nome  $idC_1$ . Tale attributo diventa chiave primaria della tabella  $C_1$ .

$C_1(\underline{idC_1}, A_1, \dots, A_n)$

I tipi degli attributi vengono scelti in base alla seguente corrispondenza:

**String:** VARCHAR(x)

**Integer:** INTEGER

**Real:** REAL o DOUBLE

# Le regole di corrispondenza tra GeoUML e il modello SFS

*Classe*  $C_1$  contenente gli attributi alfanumerici:

$A_1, \dots, A_n$  dove  $A_{i,1}, \dots, A_{i,k}$  sono chiave primaria

## Mapping delle chiavi primarie

Alla tabella che rappresenta la  $C_1$  viene aggiunto il seguente vincolo:

$C_1(\underline{\text{id}C_1}, A_1, \dots, A_n)$

**vincolo  $C_1\text{-PK}$  ( $A_{i,1}, \dots, A_{i,k}$ )**

se la chiave primaria include ruoli allora anche la chiave esportata che rappresenta il legame con l'altra classe va indicata nel vincolo.

# Le regole di corrispondenza tra GeoUML e il modello SFS

## Mapping degli attributi enumerati

Per ogni *attributo enumerato*  $A_{enum}$  con valori  $(v_1, \dots, v_n)$ , aggiungere alla tabella che rappresenta la classe  $C$  che contiene  $A_{enum}$  il seguente vincolo:

**vincolo  $C\_A_{enum}$  ( $A_{enum}$  IN  $(v_1, \dots, v_n)$ )**

(sintassi: **vincolo** <nomeVincolo> (<nome attributo> IN (<elenco valori>)) )

Per ogni *attributo enumerato*  $A_{enum}$  con *dominio enumerato*  $D$  contenente i valori  $(v_1, \dots, v_n)$ , si genera un dominio indicando il tipo e il vincolo:

**dominio  $D\_C\_A_{enum}$  tipo varchar(x)**

**vincolo (value IN  $(v_1, \dots, v_n)$ )**

(sintassi **dominio** <nome> **tipo** <tipoSQL> **vincolo** (value IN (<elenco valori>)) )

e si assegna il dominio  $D\_C\_A_{enum}$  come tipo dell'attributo  $A_{enum}$  nella tabella  $C$ .

# Le regole di corrispondenza tra GeoUML e il modello SFS

## Mapping degli attributi enumerati gerarchici

Per ogni *attributo enumerato gerarchico*  $A_{enumG}$  della classe  $C$  con dominio specificato ad esempio nel seguente modo:

$(v_1 A_1:(v_{1,1} B_1:(v_{1,1,1}, v_{1,1,2}), v_{1,2}, v_{1,3}), v_2 A_{2,1}:(v_{2,1}, v_{2,2}) A_{2,2}:(v_{2,3}, v_{2,4}), v_3, v_4)$   
aggiungere una tabella con la seguente struttura:

$C_{A_{enumG}}(\underline{idA_{enumG}}, value, A_1, B_1, A_{2,1}, A_{2,2})$

e con i vincoli seguenti:

**vincolo  $C\_value$  ( $value$  IN ( $v_1, v_2, v_3, v_4$ ))**

**vincolo  $C\_A_1$  (( $value = v_1$  AND  $A_1$  IN ( $v_{1,1}, v_{1,2}, v_{1,3}$ )) OR  
( $value \neq v_1$  AND  $A_1$  is NULL AND  $B_1$  is NULL))**

...

(sintassi vincolo <nome> (<espressione clausola WHERE di SQL>))

Inoltre aggiungere alla tabella  $C$  l'attributo  $idA_{enumG}$  chiave esportata della tabella precedente.

# Le regole di corrispondenza tra GeoUML e il modello SFS

*Classe  $C_1$  contenente un attributo geometrico  $G_1$ .*

## *Mapping degli attributi geometrici*

Si aggiunge un attributo geometrico  $G_1$  alla relazione  $C_1$  con dominio geometrico scelto secondo la corrispondenza mostrata nel lucido successivo.

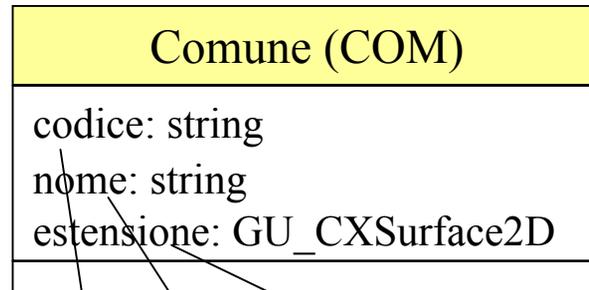
# Corrispondenza tra tipi geometrici GeoUML e tipi geometrici SFS

Tipo GeoUML	Tipo SFS
GU_Point*	Point
GU_CPCurve*	LineString
GU_CPRing*	LineString + vincolo
GU_CPSurface2D	Polygon
GU_CXPoint*	Multipoint + vincolo
GU_CXCurve*	Multilinestring + vincolo
GU_CNCCurve*	Multilinestring + vincolo
GU_CXRing*	Multilinestring + vincolo
GU_CXSurface2D	Multipolygon

Tipo GeoUML	Tipo SFS
GU_Complex*	GeometryCollection + vincolo
GU_Aggregate*	GeometryCollection
GU_MPoint*	Multipoint
GU_MCurve*	GeometryCollection + vincolo
GU_MRing*	GeometryCollection + vincolo
GU_MSurface2D	GeometryCollection + vincolo

N.B.: si noti che tutti i tipi 3D  
degenerano allo spazio 2D.

# Esempio di traduzione



FID

Comune (idComune, codice, nome, estensione: Multipolygon)

# Le regole di corrispondenza tra GeoUML e il modello SFS

*Associazione A tra una Classe  $C_1$  e Classe  $C_2$ .*

## Mapping delle associazioni

Se l'associazione è uno a molti o uno a uno si esporta il feature id della classe coinvolta con cardinalità molti nella tabella della classe coinvolta con cardinalità uno usando il nome del ruolo come nome dell'attributo.

$C_1(\underline{\text{id}C_1}, A_1, \dots, A_n, \boxed{\text{ruolo}C_2}) \longrightarrow C_2(\underline{\text{id}C_2}, B_1, \dots, B_m)$

Se l'associazione è molti a molti si genera una tabella esplicita per rappresentare l'associazione:

$C_1(\underline{\text{id}C_1}, A_1, \dots, A_n) \xrightarrow{\quad} C_2(\underline{\text{id}C_2}, B_1, \dots, B_m)$   
 $\swarrow \quad \searrow$   
 $A(\boxed{\underline{\text{ruolo}C_1}}, \boxed{\underline{\text{ruolo}C_2}}, D_1, \dots, D_k)$

dove  $D_1, \dots, D_k$  sono gli eventuali attributi dell'associazione.

# Le regole di corrispondenza tra GeoUML e il modello SFS

*Classe tratto (sottoarea)  $T_1$  contenente gli attributi alfanumerici:*

$B_1, \dots, B_n$

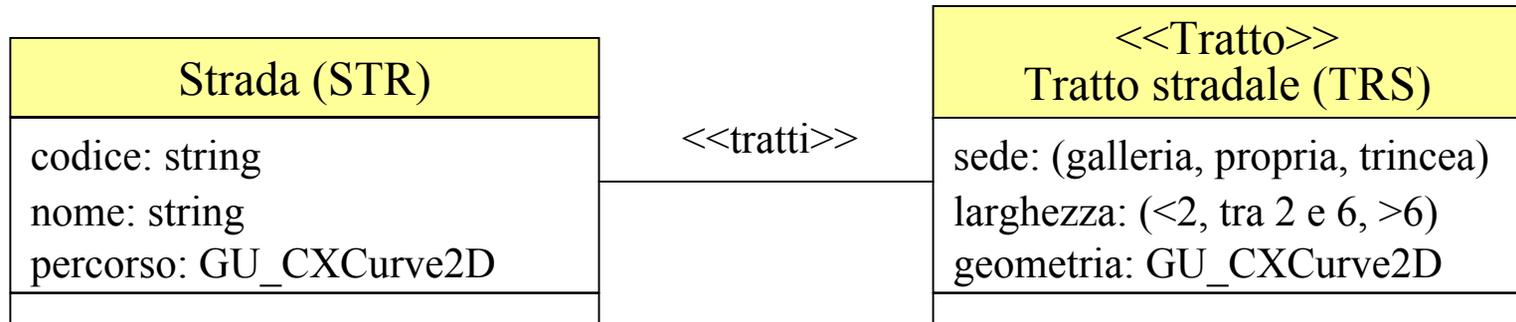
## *Mapping delle classi tratto (sottoarea)*

Per la classe tratto (sottoarea)  $T$  viene creata una relazione (tabella)  $T$  che contiene tanti attributi quanti sono gli attributi alfanumerici della classe  $T$  e un attributo geometrico detto *geometria* di tipo Multilinestring (Multipolygon).

La relazione (tabella)  $T_1$  ha un attributo aggiuntivo detto *SID* (segment or subregion ID) di tipo integer e con nome *sidT*. Tale attributo insieme al feature id della classe “padre del tratto”  $C_i$  (detto *id C<sub>i</sub>*) diventa chiave primaria della tabella  $T$ .

**T(sidT, idC<sub>i</sub>,  $B_1, \dots, B_n$ , geometria: Multilinestring)**

# Esempio di traduzione



**Strada** ( idSTR, codice, nome, percorso: Multilinestring)

**TrattoStradale** (sidTRS, idSTR, sede, larghezza,  
geometria: Multilinestring)

**vincolo** TRS\_sede (sede **IN** ('galleria', 'propria', 'trincea'))

**vincolo** TRS\_larghezza (larghezza **IN** ('<2', 'tra 2 e 6', '>6'))

# Le regole di corrispondenza tra GeoUML e il modello SFS

## Mapping degli strati topologici

Per ogni strato topologico S presente nello schema GeoUML si genera dove possibile uno strato topologico nello schema logico, facendo confluire sullo strato tutti gli attributi geometrici direttamente o indirettamente vincolati a S.

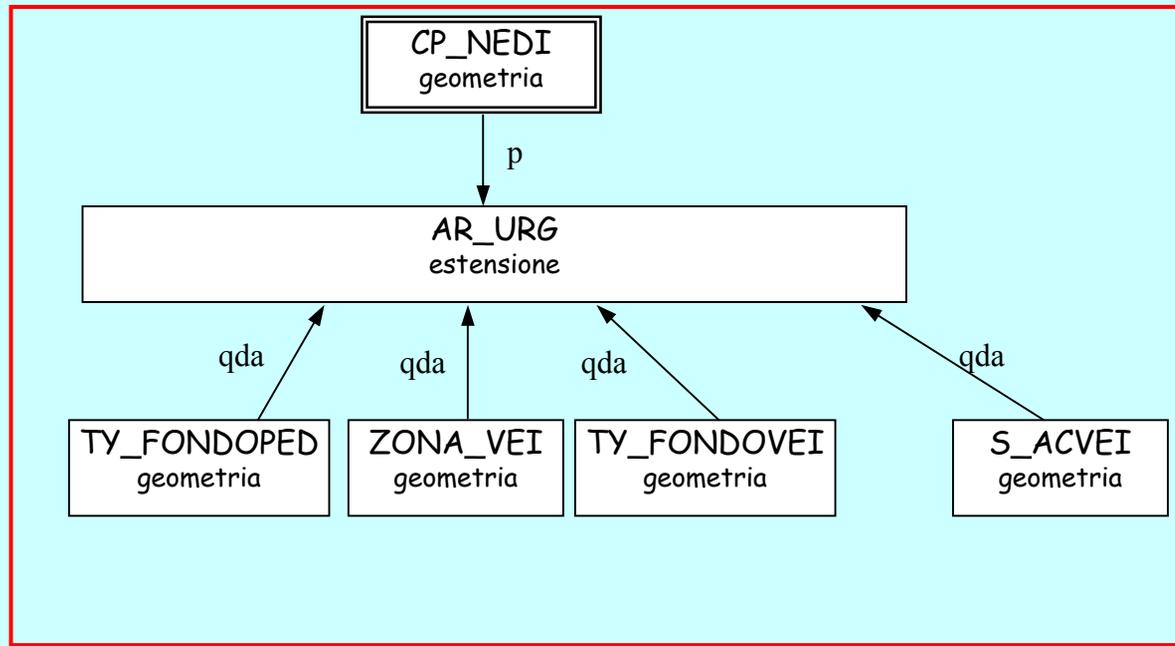
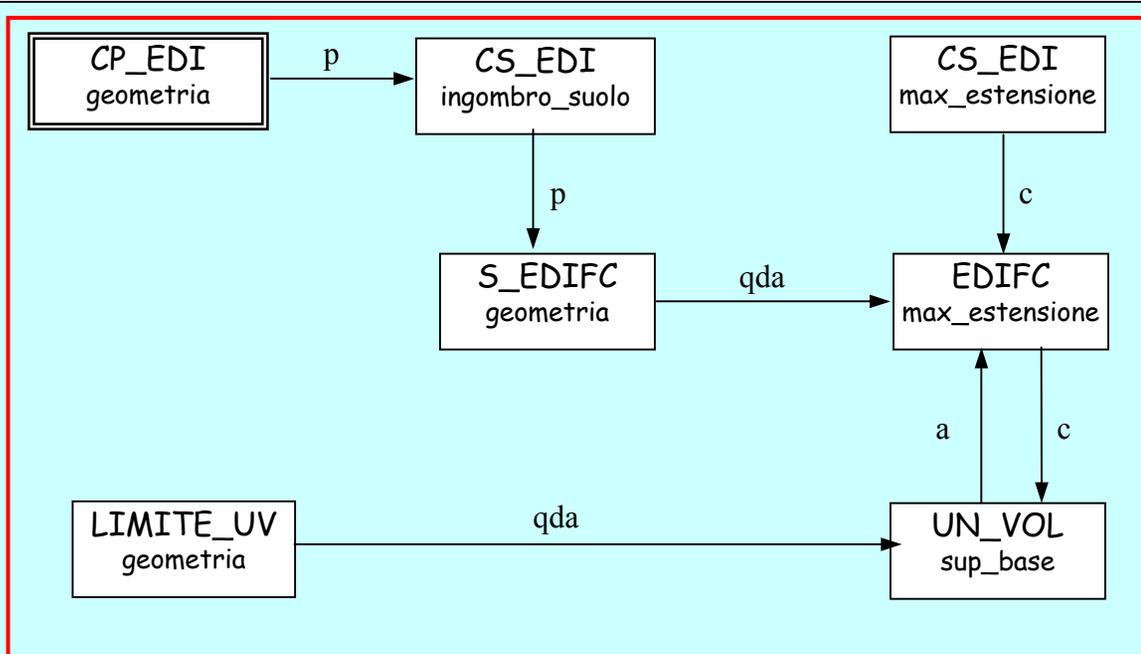
# Le regole di corrispondenza tra GeoUML e il modello SFS

## Generazione degli strati topologici

Nel caso in cui sia possibile generare strati topologici a livello logico, allora dato uno schema GeoUML è possibile applicare il seguente procedimento per la generazione degli strati:

- generare il grafo dei vincoli strutturali (**grafo strutturale**): ogni attributo geometrico, classe tratto/sottoarea o strato è un nodo e ogni vincolo strutturale è un arco non orientato)
- generare i **sottografi connessi** del grafo strutturale
- **per ogni sottografo connesso generare uno strato topologico a livello logico** e associare allo strato tutti gli attributi geometrici presenti nel sottografo, comprese le geometrie dei tratti e delle sottoaree ed esclusa la geometria dello strato topologico concettuale se è l'unico strato del sottografo.

# Esempio



## Mapping degli strati (ESEMPIO)

### Strato CP\_EDI

CS\_EDI.ingombro\_suolo → CP\_EDI

CS\_EDI.max\_estensione → CP\_EDI

S\_EDIF.geometria → CP\_EDI

EDIFC.max\_estensione → CP\_EDI

UN\_VOL.sup\_base → CP\_EDI

LIMITE\_UV.geometria → CP\_EDI

### Strato CP\_NEDI

AR\_URG.ingombro\_suolo → CP\_NEDI

TY\_FONDOPED.geometria → CP\_NEDI

TY\_FONDOVEI.geometria → CP\_NEDI

ZONA\_VEI.geometria → CP\_NEDI

S\_ACVEI.geometria → CP\_NEDI