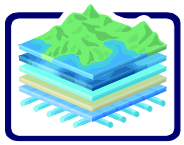
POLITECNICO
DI MILANO



SpatialDBgroup

Modello Implementativo Shape Topologico (SHAPE_TOPO)

21 aprile 2011

SpatialDBgroup

SpatialDBgroup@polimi.it

http://SpatialDBgroup.polimi.it

Introduzione

Questo Modello Implementativo si basa sull'idea di rappresentare le Componenti Spaziali delle classi in particolari strutture dette Insiemi Topologici. Un insieme topologico viene a sua volta rappresentato da un certo numero di Shape.

Il documento è strutturato nel modo seguente:

1. definizione delle proprietà di un generico Insieme Topologico
2. definizione della rappresentazione di una istanza di componente spaziale di una classe in un insieme topologico
3. definizione della rappresentazione del Mapping generale tra le classi e gli insiemi topologici
4. richiami sulla struttura degli shape in appendice

1. L'Insieme Topologico

Si intende per insieme topologico un insieme di primitive geometriche 2D e/o 3D che rispettano delle proprietà topologiche negli spazi 2D e 3D.

Siano IT1, IT2 ...ITx ... insiemi topologici.

Un insieme topologico ITx è composto da 4 **shape topologici** contenenti la rappresentazione geometrica delle **primitive** topologiche. Le primitive sono suddivise in **edge** (primitive lineari) e **node** (primitive puntiformi).

I 4 shape topologici di un insieme topologico ITx sono:

- ITx_E_3D: shape di tipo PolylineZ – 1 part (non accetta record con curve multipart) chiamato shape degli edge nello spazio 3D.

- ITx_E_2D: è uno shape di tipo Polyline – 1 part (non accetta record con curve multipart) chiamato shape di edge nello spazio 2D.

- ITx_N_3D: è uno shape di tipo PointZ chiamato shape di nodi nello spazio 3D.

- ITx_N_2D: è uno shape di tipo Point chiamato shape di nodi nello spazio 2D.

Tra le primitive dei 4 shape di uno stesso insieme topologico devono sussistere le seguenti proprietà topologiche e metriche allo scopo di garantire che la sovrapposizione di qualsiasi coppia di primitive non generi nuovi punti di intersezione e che le primitive soddisfino dei requisiti di

distanza minima che impediscano la modifica della geometria delle primitive a causa di operazioni di correzione topologica.

1. **Vincoli interni ad ognuno dei 4 shape**

   a. Ogni edge di ITx_E_2D e di ITx_E_3D deve essere semplice (no autointersezione, autotangenza)
   b. Ogni edge di ITx_E_3D non può avere segmenti verticali, ossia edge nei quali due vertici consecutivi o un estremo e il suo vertice più vicino differiscano solo nel valore della coordinata Z.
   c. Per ogni coppia di edge di ITx_E_2D (ITx_E_3D) la loro intersezione è vuota oppure deve coincidere con uno degli estremi degli edge coinvolti.
   d. Due nodi di ITx_N_2D (ITx_N_3D) non possono coincidere.

   Osservazioni:

   - non possono esistere due edge uguali o parzialmente sovrapposti (condivisione) in ITx_E_2D (ITx_E_3D).

2. **Vincoli di consistenza Edge/Nodi**

   a. Ogni nodo di ITx_N_2D (ITx_N_3D) è isolato (intersezione nulla con tutti gli edge di ITx_E_2D(ITx_E_3D)) oppure coincide con un estremo di un edge di ITx_E_2D (ITx_E_3D).

   Osservazioni:

   - non è accettata la coincidenza di un nodo con un vertice interno di un edge.

3. **Vincoli di consistenza tra spazio 2D e spazio 3D**

   a. Per ogni edge di ITx_E_3D deve esistere un edge in ITx_E_2D che coincide esattamente con la sua proiezione planare (ottenuta eliminando la Z da tutti i suoi vertici)
   b. Per ogni nodo di ITx_N_3D deve esistere un nodo in ITx_N_2D che coincide esattamente con la sua proiezione planare.

   Osservazioni:

   - questi vincoli impongono che un edge (node) definito nello spazio 3D abbia un corrispondente elemento nello spazio 2D, ma viceversa ammettono edge (node) definiti nello spazio 2D che non abbiano un corrispondente elemento nello spazio 3D.

4. **Vincoli sulla distanza minima**

Si devono definire i valori dei seguenti parametri metrici:

- La risoluzione decimale di consegna, ossia il numero di decimali utilizzati nella descrizione delle coordinate di tutte le primitive topologiche; si può sperimentare una risoluzione di 10^{-5} (5 decimali); è possibile fornire valori con un numero di decimali inferiori, ma non superiori.
- La distanza minima di indistinguibilità (Dm) tra due coordinate. Introdotta per tener conto degli errori dovuti alla rappresentazione dei numeri reali permette di stabilire una soglia sopra la quale si ritiene che punti distinti rimangano tali e sotto la quale due punti distinti possano convergere allo stesso punto. I valori di questa soglia possono essere determinati in base al sistema di memorizzazione e/o al sistema di correzione topologica dei dati effettuata dai sistemi di generazione delle strutture topologiche come Radius topology (da stabilire insieme).

Chiamati:

N2D(N3D) l'insieme dei nodi $\in$ ITx_N_2D (ITx_N_3D)

E2D (E3D) l'insieme degli estremi degli edge $\in$ ITx_E_2D (ITx_E_3D),

V2D (V3D) l'insieme di tutti i vertici interni di tutti gli edge $\in$ ITx_E_2D (ITx_3D)

D(Oi, Oj) la distanza tra due geometrie Oi e Oj

valgono i seguenti vincoli:

a. per ogni edge $\in$ ITx_E_2D ($\in$ ITx_E_3D) la distanza tra due vertici consecutivi o tra un estremo e il suo vertice più vicino deve essere sempre maggiore di Dm per evitare segmenti di edge o edge troppo corti.
b. ogni coppia <pi, pj> con pi,pj $\in$ N2D $\cup$ E2D $\Rightarrow$ D(pi,pj) >Dm
c. ogni coppia <pi, pj> con pi,pj $\in$ N3D $\cup$ E3D $\Rightarrow$ D(pi,pj) >Dm
d. per ogni coppia <pi, lj> con pi $\in$ N2D $\cup$ E2D e lj $\in$ ITx_E_2D $\Rightarrow$ D(pi,lj) >Dm
e. per ogni coppia <pi, lj> con pi $\in$ N3D $\cup$ E3D e lj $\in$ ITx_E_3D $\Rightarrow$ D(pi,lj) >Dm

f. per ogni coppia <vi,lj> con vi ∈ V2D e e lj ∈ ITx_E_2D ⇒ D(vi,lj) >Dm
g. per ogni coppia <vi,lj> con vi ∈ V3D e e lj ∈ ITx_E_3D ⇒ D(vi,lj) >Dm

Osservazioni:

- le condizioni b. e c. impediscono estremi di edge vicini a nodi, estremi di edge vicini ad altri estremi di edge, nodi vicini a nodi (node-node di radius per il 2D); si noti che se questa condizione è soddisfatta dai nodi e dagli estremi degli edge nello spazio 2D essa lo sarà anche per tutti i nodi ed estremi nello spazio 3D (poiché le primitive 2D sono la proiezione di quelle 3D) ad eccezione di nodi/estremi che differiscano solo per la coordinata Z e per i quali è necessario verificare che la differenza di quota sia >Dm. Viceversa se una coppia di punti confrontati nello spazio 2D non soddisfa la condizione è possibile che essa sia invece rispetta nello spazio 3D.
- le condizioni d. ed e. impediscono nodi o estremi di edge troppo vicini a segmenti o vertici interni di edge (node-edge radius)
- le condizioni f. e g. impediscono vertici di edge vicini a segmenti o altri vertici di edge (edge-edge radius)
- I vincoli precedenti stabiliscono anche una regola di creazione dei poligoni virtuali (pareti verticali o per bufferizzazione di linee); infatti immaginando di creare un rettangolo si deduce che la larghezza deve essere sempre > Dm sia che sia costruito con un unico edge (ring) e sia che sia costruito con più edge.

Casi particolari di Insiemi topologici

Un insieme topologico può essere associato ad un numero ridotto di shape e in tal caso rimangono valide le sole proprietà applicabili agli shapefile effettivamente presenti.

Geometrie presenti	Shape presenti nell'insieme topologico
Lineari e puntiformi in 2D	ITx_E_2D e ITx_N_2D
Lineari in 2D	ITx_E_2D
Puntiformi in 2D	ITx_N_2D
Lineari in 3D	ITx_E_3D e ITx_E_2D
Puntiformi in 3D	ITx_N_3D e ITx_N_2D
Lineari in 3D con topologia solo 3D	ITx_E_3D
Puntiformi in 3D con topologia solo 3D	ITx_N_3D
Lineari e puntiformi in 3D con topologia solo 3D	ITx_E_3D e ITx_N_3D

Una geometria 3D inserita in un insieme topologico implica, in generale, la generazione anche del corrispondente shapefile in 2D ottenuto per proiezione.

E' data quindi la possibilità di definire anche un insieme topologico solo 3D che memorizza le geometrie 3D negli shapefile ITx_E_3D e ITx_N_3D, ma che non richiede la corrispondente componente planare in 2D.

2 Rappresentazione di una Componente Spaziale di una classe in un Insieme topologico

Gli **edge** sono geometrie elementari che sono usate per descrivere le curve e i boundary delle surface GeoUML e anche gli attributi a tratti e sottoaree. I **nodi** descrivono i componenti delle geometrie di tipo Point GeoUML e gli attributi a eventi.

Nella seguente tabella 1 si indica per ogni tipo GeoUML quale sia lo shape topologico nel quale gli oggetti geometrici sono memorizzati; i tipi GU_Aggregate2D e GU_Aggregate3D e il collassamento sono trattati poi.

Tipo GeoUML	Shape topologico dei componenti
GU_Point2D	ITx_N_2D
GU_Point3D	ITx_N_3D e ITx_N_2D(*)
GU_CPCurve2D	ITx_E_2D
GU_CPCurve3D	ITx_E_3D e ITx_E_2D(*)
GU_CPSimpleCurve2D	ITx_E_2D
GU_CPSimpleCurve3D	ITx_E_3D e ITx_E_2D(*)
GU_CPRing2D	ITx_E_2D
GU_CPRing3D	ITx_E_3D e ITx_E_2D(*)
GU_CPSurface2D	ITx_E_2D
GU_CXPoint2D	ITx_N_2D
GU_CXPoint3D	ITx_N_3D e ITx_N_2D(*)
GU_CXCurve2D	ITx_E_2D
GU_CXCurve3D	ITx_E_3D e ITx_E_2D(*)
GU_CXRing2D	ITx_E_2D
GU_CXRing3D	ITx_E_3D e ITx_E_2D(*)
GU_CNCurve2D	ITx_E_2D
GU_CNCurve3D	ITx_E_3D e ITx_E_2D(*)
GU_CXSurface2D	ITx_E_2D
GU_CPSurfaceB3D	ITx_E_3D e ITx_E_2D(*)
GU_CXSurfaceB3D	ITx_E_3D e ITx_E_2D(*)

TABELLA 1

(*) lo shape è omesso se la topologia è solo 3D

3 Definizione del mapping di base concettuale/fisico

Il mapping di base definisce le strutture dati per la memorizzazione degli insiemi topologici, delle classi con relativi attributi monovalore e della corrispondenza tra classi e insiemi topologici. Inoltre il mapping definisce le regole per la definizione dei nomi da assegnare agli shapefile e ai relativi attributi. Le regole sono:

- Il codice alfanumerico assegnato ad ogni elemento della specifica concettuale (classe, attributo, ruolo,…) sarà utilizzato come nome del corrispondente elemento nello schema fisico.
- L'uso del codice alfanumerico è opzionale nella composizione dei nomi della specifica concettuale e pertanto sarà generato automaticamente quando non viene assegnato nella specifica concettuale.
- Nel caso in cui il mapping abbia generato automaticamente codici uguali oppure nel caso in cui si ereditano nomi uguali nelle trasformazioni di schema (ad es., attributi di un datatype che vengono riassorbiti nella classe dove esistono altri attributi con codice alfanumerico uguale sono risolti automaticamente dal sistema attraverso un cambiamento dei nomi (introduzione di suffissi numerici).
- Il nome di un attributo di uno shapefile non deve superare i 10 caratteri, altrimenti il sistema provvederà al troncamento e all'introduzione di un suffisso numerico per mantenere l'univocità dei nomi.

Mapping dell'insieme topologico

Durante la definizione del mapping per la data product specification vengono definiti il nome, il codice e la descrizione di ogni insieme topologico e la corrispondenza tra insiemi topologici e le componenti spaziali di ogni classe.

Un insieme topologico è quindi associato ad un insieme di componenti spaziali di una o più classi (gli attributi a tratti/sottoaree sono trattati poi). Componenti spaziali di classi diverse e di diversa dimensionalità possono confluire in uno stesso insieme topologico e una classe con più componenti spaziali le può memorizzare in insiemi topologici diversi.

Data una componente spaziale (detta anche attributo geometrico) CL.AG di una classe CL esiste un preciso shapefile di un insieme topologico (o complementare) destinato a contenerla; esso è determinato nel modo seguente:

1. in base al mapping esplicito, CL.AG→ ITx
2. in base alla tabella 1 precedente e al tipo di insieme topologico, CL.AG.type → shapefile topologico di ITx

Il nome degli shapefile di un insieme topologico è composto da un prefisso ed un suffisso costanti concatenati al codice dell'insieme topologico definito dall' utente nel seguente modo:

IT_*codice*_E_3D.shp **IT_*codice*_E_2D.shp** **IT_*codice*_N_3D.shp** **IT_*codice*_N_2D.shp**

Supponiamo che si definisca un insieme topologico con nome "Copertura del suolo" e codice alfanumerico "SUOLO" si otterranno i seguenti file: IT_SUOLO_E_3D, IT_SUOLO_E_2D, IT_SUOLO_N_3D, IT_SUOLO_N_2D.

Il nome degli shapefile sarà quindi assegnato ai corrispondenti file con estensione .shp, .dbf e .shx.

Negli shapefile (più correttamente all'interno dei file .dbf dello shape file) è aggiunto un attributo chiamato "**primid**" di tipo Number 9 (0 decimale) che contiene l'identificatore di ogni primitiva dello shapefile topologico; tale identificatore è univoco nello shapefile.

La corrispondenza tra le istanze della componente spaziale di una classe (CL.GA) e le primitive dello shapefile è descritta in un file .dbf aggiuntivo chiamato "**file di composizione**". Il nome dei file di composizione seguono la stessa convenzione stabilita per gli shapefile: si utilizza un prefisso ed un suffisso concatenato al codice dell'insieme topologico

IT_*codice*_E_3D_COMP, **IT_*codice*_E_2D_COMP,**

IT_*codice*_N_3D_COMP e **IT_*codice*_N_2D_COMP.**

Ad es., per l'insieme topologico della copertura del suolo si generano i seguenti file di composizione:IT_SUOLO_E_3D_COMP, IT_SUOLO_E_2D_COMP, IT_SUOLO_N_3D_COMP, IT_SUOLO_N_2D_COMP.

I file di composizione sono composti da due attributi <primid, geoid> entrambi di tipo Number 9 (0 decimale); esso definisce quali primitive di uno shapefile compongono ogni geometria delle componenti spaziali di classi memorizzate in tale shapefile.

L'attributo "**geoid**" rappresenta l'identificatore univoco di ogni singola geometria di una delle componenti spaziali associate all'insieme topologico ed è univoco nell'intero insieme topologico e inoltre la coppia <primid,geoid> è univoca all'interno di ogni file di composizione. Data un'istanza di geometria di una componente spaziale di una classe identificata da un valore geoidN, l'insieme delle coppie <primidX1, geoidN>, <primidX2, geoidN>,..., <primidXk, geoidN> definisce

l'insieme di primitive {primidX1, primidX2,..., primidXk} che compongono tale istanza di attributo geometrico.

Condivisione delle primitive:

se due istanze geoidA e geoidB di componenti spaziali associate allo stesso insieme topologico condividono una o più primitive primidY1,...,primidYk, nel DBF di composizione esisteranno le coppie <primidY1, geoidA>,...,<primidYk, geoidA>, <primidY1, geoidB>,...,<primidYk, geoidB>.

In particolare, se le due istanze geoidA e geoidB sono uguali, esse condividono tutte le loro primitive.

Mapping delle classi

Le regole precedenti determinano come sono rappresentate le istanze delle geometrie delle componenti spaziali delle classi. Inoltre, per ogni classe **CL(A1,...An, AG1,..., AGm)** composta da un insieme di attributi descrittivi monovalore A1,..,An (cardinalità massima 1) e dalle componenti spaziali AG1,...,AGm il mapping prevede la produzione di un ulteriore file .dbf con la struttura

codiceCL(ClassID, codiceA1, ..., codiceAn, codiceAG1, ..., codiceAG'm)

dove:

- il nome del file corrisponde al codice alfanumerico della classe stessa assegnato nella specifica concettuale; Es., Area di circolazione veicolare → AC_VEI.dbf, Area di circolazione pedonale →AC_PED.dbf, Edificio → EDIFC.dbf
- ClassID è un attributo (String(70)) atto a rappresentare l'identificatore delle istanze (feature, oggetti) della classe. ClassID è univoco in una classe e se tale classe appartiene ad una gerarchia è univoco in tutta la gerarchia; non si impone alcun algoritmo per l'assegnazione dei valori a tale attributo.
- Il nome degli attributi codiceA1, ..., codiceAn, codiceAG1, ..., codiceAGm corrisponde al codice alfanumerico assegnato nella specifica concettuale ai corrispondenti attributi;
- I tipi degli attributi descrittivi codiceA1, ..., codiceAn sono definiti in base alla seguente tabella di conversione:

tipo geouml	tipo dbf
Integer	--> Tipo N - massimo 9 cifre (0 decimali)
Real	--> Tipo N - massimo 10 cifre (3 decimali)
String (x)	--> Tipo C - lunghezza x (valore massimo di x è 254)
NumericString (x)	--> Tipo C - lunghezza x (il valore massimo di x è 254)
Date	--> Tipo D
DateTime	--> Tipo C – lunghezza 19 (formato gg/mm/aaaa hh:mm:ss)

\TimeStamp --> Tipo C – lunghezza 8 (formato hh:mm:ss)

Boolean --> Tipo N – massimo 2 cifre senza decimali (0 per falso e 1 per vero)

- I valori memorizzati negli attributi codiceAG1, ..., codiceAG rappresentano il "geoid" associato alla corrispondente componente spaziale e pertanto il tipo è Number 9 (0 decimale);
- l'attributo ScRil (scala di rilievo - vedi documentazione sul GeoUML, cap. 8) può essere aggiunto alla tabella di classe per classi a istanze monoscala.

Osservazioni su alcuni esempi di scenari d'uso

1. Componenti spaziali 3D con proprietà di adiacenza nello spazio 2D. Si prendano 2 classi di esempio C1(id, AG1:GU_CPSurfaceB3D,…) e C2(id, AG2: GU_CPSurfaceB3D) le cui componenti spaziali appartengono allo stesso insieme topologico; l'impatto della presenza di eventuali attributi a tratti/sottoaree è trattato successivamente. L'insieme topologico è composto dagli shape ITx_E_3D e ITx_E_2D: il primo contiene gli edge che descrivono il boundary 3D delle 2 superfici e il secondo contiene gli edge ottenuti dalla proiezione planare degli edge 3D. L'insieme topologico impone che gli edge 3D e i corrispondenti edge 2D non si intersechino negli spazi 3D e 2D rispettivamente; ciò impone che nel caso in cui si intersechino i soli edge dello spazio 2D sia necessario spezzare opportunamente anche gli edge 3D al fine di eliminare le intersezioni tra le corrispondenti proiezioni nello spazio 2D. Si fa anche notare che nel caso in cui degli edge 3D disgiunti generino edge 2D uguali sia necessario eliminare la duplicazione prodotta dalla proiezione, generando quindi un edge 2D condiviso (lo stesso valore di primid sarà associato a diversi geoid nel file .dbf ITx_E_2D_COMP).
2. Componente spaziale in 3D con proprietà di adiacenza nello spazio 2D con componenti spaziali 2D. Si prendano 2 classi di esempio C1(id, AG1: GU_CPSurfaceB3D,…) e C2(id, AG2: GU_CPSurface2D) le cui componenti spaziali sono associate allo stesso insieme topologico. L'insieme topologico è composto dagli shape ITx_E_3D e ITx_E_2D: il primo contiene gli edge che descrivono il boundary 3D della componente spaziale AG1 e il secondo contiene gli edge ottenuti dalla proiezione planare degli edge 3D e gli edge della componente spaziale AG2. A differenza dell'esempio precedente, nel caso di edge di ITx_E_2D intersecanti sarà necessario spezzare anche gli edge 3D se e solo se gli edge 2D coinvolti sono anche proiezione di qualche edge 3D. Nel caso in cui degli edge 3D proiettati generino degli edge 2D uguali ed eventualmente uguali anche ad alcuni edge 2D generati dalla componente spaziale AG2 sarà necessario, analogamente all'esempio precedente, la rimozione della duplicazione e la generazione della condivisione.

4. L'Insieme complementare

Si tratta dell'insieme nel quale sono inserite le geometrie di tutti gli attributi geometrici delle classi e degli strati topologici che non confluiscono in alcun insieme topologico poiché non hanno proprietà di adiacenza da rispettare. A questo insieme non si applicano pertanto i vincoli che caratterizzano l'insieme topologico.

Questo insieme è costituito dagli stessi 4 shapefile degli insiemi topologici:

- ITx_E_3D: shape di tipo PolylineZ – 1 part (non accetta record con curve multipart) chiamato shape degli edge nello spazio 3D.

- ITx_E_2D: è uno shape di tipo Polyline – 1 part (non accetta record con curve multipart) chiamato shape di edge nello spazio 2D.

- ITx_N_3D: è uno shape di tipo PointZ chiamato shape di nodi nello spazio 3D.

- ITx_N_2D: è uno shape di tipo Point chiamato shape di nodi nello spazio 2D.

e dai corrispondenti file di composizione:

ITx_E_3D_COMP, ITx_E_2D_COMP, ITx_N_3D_COMP e ITx_N_2D_COMP.

I nomi degli shapefile, dei file di composizione e dei loro attributi seguono le regole descritte nella sezione precedente.

Si noti che, per semplicità, è stato mantenuto, come negli insiemi topologici, uno shapefile lineare composto da curve connesse semplici (1 part).

L'inapplicabilità dei vincoli caratterizzanti l'insieme topologico permette di non dover identificare le porzioni di geometria condivisa e quindi è possibile descrivere con una primitiva l'intera geometria della componente spaziale di un oggetto di una classe.

Inoltre ad ogni tipo GeoUML corrisponde solo uno shapefile come mostrato in Tabella 2 (rimosso il doppio shape per quelli 3D)

Tipo GeoUML	Shape dell'insieme complementare
GU_Point2D	ITx_N_2D
GU_Point3D	ITx_N_3D
GU_CPCurve2D	ITx_E_2D
GU_CPCurve3D	ITx_E_3D
GU_CPSimpleCurve2D	ITx_E_2D
GU_CPSimpleCurve3D	ITx_E_3D
GU_CPRing2D	ITx_E_2D
GU_CPRing3D	ITx_E_3D
GU_CPSurface2D	ITx_E_2D
GU_CXPoint2D	ITx_N_2D
GU_CXPoint3D	ITx_N_3D
GU_CXCurve2D	ITx_E_2D
GU_CXCurve3D	ITx_E_3D
GU_CXRing2D	ITx_E_2D
GU_CXRing3D	ITx_E_3D
GU_CNCurve2D	ITx_E_2D
GU_CNCurve3D	ITx_E_3D
GU_CXSurface2D	ITx_E_2D
GU_CPSurfaceB3D	ITx_E_3D
GU_CXSurfaceB3D	ITx_E_3D

5. Ulteriori regole di mapping

Mapping dei domini

Ogni dominio è memorizzato in un file .dbf contenente i campi:

Nome campo	Tipologia	Descrizione
CODE	Stringa (80)	Codice del valore a cui puntano i valori dell' attributo posizione
NAME	Stringa (160)	Trascodifica del codice – Valore del dominio
DEFINITION	Stringa (254)	Descrizione del valore di dominio
ALPHACODE	Stringa (80)	Ulteriore codice alfanumerico

Se il dominio non è embedded il file avrà il nome del codice alfanumerico del dominio preceduto dal prefisso D_ se è un dominio semplice o H_ se è un dominio gerarchico.

Nel caso in cui il dominio sia embedded il nome del file sarà dato dal codice alfanumerico della classe (o della associazione) concatenato al codice alfanumerico dell' attributo sempre preceduto dal prefisso D_ se è un dominio semplice o H_ se è un dominio gerarchico.

Opzionalità dei valori

Un attributo descrittivo opzionale (cardinalità minima 0) ammette che un record possa lasciare indefinito il valore per l'attributo; la stessa proprietà si applica anche ai ruoli descritti nel file della classe (casi uno a molti e uno a uno). Il valore NULL del GeoUML viene sostituito per ognuno dei tipi dbase definiti in Sezione 3.1 da uno dei valori del tipo (ad esempio, -9999 per il tipo intero) che sarà chiamato valore nullo sostitutivo. Lo stesso approccio si applica agli attributi opzionali che hanno associato un dominio (gerarchico); in questo caso il valore nullo sostitutivo non deve corrispondere a nessuno dei valori definiti nei domini della specifica. Infine i ruoli opzionali nelle associazioni utilizzeranno la stringa vuota per indicare l'opzionalità dell'associazione. Questa regola impone che tutti gli attributi descrittivi e i ruoli siano definiti come obbligatori dal MI Shape-Flat; infatti ogni record conterrà in ogni attributo un valore applicativo o il valore nullo sostitutivo.

Se è stato dichiarato nella specifica concettuale il dominio D_NI (dominio Null Interpretation) non si definisce il valore nullo sostitutivo descritto in precedenza. Si definisce invece per ogni tipo dbase definito nella Sezione 3.1 un valore interpretativo sostitutivo per ogni valore del dominio D_NI (ad esempio, se una classe ha un attributo di tipo integer e uno di tipo stringa con il dominio D_NI contenente i valori 91 e 92 sarà necessario indicare i valori interpretativi sostitutivi di 91 e 92 per il tipo integer e il tipo stringa). Si noti che il valore nullo sostitutivo viene quindi usato negli attributi quando non è definito il dominio di interpretazioni dei valori nulli, altrimenti negli attributi si usa direttamente lo specifico valore di interpretazione del null.

Se una classe ha una componente spaziale opzionale (cardinalità minima 0) l'assenza della geometria in un'istanza comporta che non esisteranno primitive negli shape file dell'insieme topologico a cui appartiene la componente spaziale che facciano riferimento a tale istanza.

Mapping dell'attributo di un attributo geometrico

Gli attributi di attributi geometrici con cardinalità massima 1 sono riassorbiti nella classe. Come per gli attributi della classe il loro nome sarà definito utilizzando il codice alfanumerico.

Mapping degli attributi multivalore

Ogni attributo di una classe o ogni attributo di attributo geometrico con cardinalità massima maggiore di 1 richiede la generazione di un ulteriore file .dbf.

Il nome di questo file sarà composto dalla concatenazione <codice alfanumerico classe >_<codice alfanumerico attributo>.

Esempio attributo tipo della giunzione stradale: GZ_STR_GZ_STR_TY.dbf.

Nel caso in cui ci sia un attributo di attributo geometrico multivalore con il codice alfanumerico uguale ad un attributo multivalore della classe, il nome del dbf sarà composto dalla concatenazione <codice alfanumerico classe>_<codice alfanumerico attributo geometrico >_<codice alfanumerico attributo di attributo geometrico>.

Il file conterrà i seguenti campi:

- **ClassREF:** che contiene l'identificatore ClassID dell'oggetto della classe dell'attributo al quale il valore appartiene.
- l'attributo multivalore che ha come nome il codice alfanumerico dell'attributo (esempio GZ_STR_TY)

La coppia di attributi <ClassREF, attributomultivalore> sono univoci all'interno del file.

Mapping degli attributi di tipo datatype

Se l'attributo della classe di tipo datatype ha cardinalità massima uguale a 1

- gli attributi del datatype sono trasformati in attributi normali della classe; si applicano quindi le regole sul nome date per gli attributi normali monovalore.

Se l'attributo della classe di tipo datatype ha cardinalità massima maggiore di 1

- si genera un file .dbf (come nel caso dell'attributo multivalore); il nome del file è ottenuto dalla concatenazione <codice alfanumerico classe >_<codice alfanumerico attributo datatype>.
- Il file contiene tutti gli attributi componenti il datatype col nome costruito con il codice alfanumerico dell'attributo e si aggiunge l'attributo **ClassREF** che contiene l'identificatore ClassID dell'oggetto della classe dell'attributo al quale appartiene il valore del datatype.

Mapping degli strati topologici

Lo strato è trattato come una classe e quindi richiede un file dfb il cui nome è uguale al codice alfanumerico dello strato stesso e composto dai seguenti campi:

- TopoID: identificatore del record sebbene non sia necessario dato che lo strato è monoistanza; si è deciso di mantenerlo per omogeneità.

- GEOMETRY: contiene l'identificatore "geoid" del valore della componente spaziale.

Tenendo presente che la sua componente spaziale è in genere composta con le geometrie delle componenti spaziali di altri strati e/o classi se ne deriva che sia conveniente associare ad uno stesso insieme topologico lo strato e le classi/strati componenti.

Mapping degli Attributi a eventi, tratti, sottoarea

La geometria degli eventi, tratti o sottoaree deve essere memorizzata nello stesso insieme topologico della relativa componente spaziale. L'implementazione adottata è quella dei tratti, eventi e sottoaree minime, ossia si identificano le porzioni della geometria di una componente spaziale che condividono gli stessi valori di tutti gli attributi a tratti, eventi o sottoarea definiti sulla componente spaziale.

Attributi a eventi

Gli eventi sono memorizzati nello shape dei nodi ITx_N_2D (ITx_N_3D) e poiché sono definiti su una componente lineare o poligonale descritta negli shapefile ITx_E_2D (ITx_N_3D) devono quindi coincidere con gli estremi di un edge o isolati (appartenenti alla parte interna di un poligono).

Ogni insieme di eventi definiti su un attributo geometrico produrrà un file .dbf il cui nome è dato dalla concatenazione <codice alfanumerico classe>_<codice alfanumerico attributo geometrico>**_E** e che sarà composto dai seguenti campi:

- EventID: identificatore univoco della combinazione di eventi,
- ClassREF: contiene l'identificatore ClassID dell'oggetto della classe a cui fanno riferimento gli attributi a eventi
- GEOID: identificatore della geometria dell'evento
- tutti gli attributi ad eventi definiti sull'attributo geometrico; il nome di questi attributi è composto dal codice alfanumerico associato all'attributo a eventi.

Data una classe C1(AG1: GU_CPCurve3D, A1: int aEventi su AG1) il mapping produce oltre ai seguenti file per la descrizione della classe e della componente spaziale AG1:

IT_codice_ E_3D (geometry, primid)

IT_codice_E_3D_COMP (primid, geoid)

codiceC1(ClassID, codiceAG1(geoid))

anche i seguenti file per gli eventi

IT_codice_ N_3D (geometry, primid)

IT_codice_N_3D_COMP (primid, geoid)

codiceC1_codiceAG1_E(EventID, ClassREF, GEOID, codiceA1)

Attributo a tratti su CP/CXcurve*D

I tratti sono memorizzati nello stesso shape ITx_E_2D (ITx_E_3D) nel quale sono memorizzati gli edge della componente spaziale a cui si riferiscono.

Ogni insieme di attributi a tratti definiti su un attributo geometrico produrrà un file .dbf il cui nome è dato dalla concatenazione <codice alfanumerico classe>_<codice alfanumerico attributo geometrico>**_SG.** Un record del file può contenere un singolo tratto minimo che condivide gli stessi valori per tutti gli attributi a tratti o un insieme di tratti minimi; nel primo caso la geometria associata ad ogni riga è una CPCurve*D, mentre nel secondo caso sarà un CXCurve*D. La scelta tra le due opzioni avviene nella definizione della data product specification.

Il file .dbf sarà composto dai seguenti campi:

- SegmentID: identificatore univoco della combinazione di valori degli attributi a tratti,
- ClassREF: contiene l'identificatore ClassID dell'oggetto della classe a cui fanno riferimento gli attributi a tratti
- GEOID: identificatore della geometria del tratto o dei tratti minimi.
- tutti gli attributi a tratti definiti sull'attributo geometrico; il nome di questi attributi è composto dal codice alfanumerico associato all'attributo a tratti.

Si noti che, poiché l'unione dei tratti definiti sulla geometria di una componente spaziale produce la geometria della componente spaziale stessa, la componente spaziale derivabile è omessa dal corrispondente file .dbf della classe. Si ricorda che ciò implica che nel caso in cui l'attributo a tratti sia opzionale si debbano generare anche i tratti associati al valore NULL.

Data una classe C1(AG1: GU_CPCurve3D, A1: int aTratti su AG1, A2: int aTratti su AG1,…) il mapping produce i seguenti file per la descrizione della classe e della componente spaziale AG1:

IT_codice_ E_3D (geometry, primid)

IT_codice_E_3D_COMP (primid, geoid)

codiceC1(ClassID, …)

e il seguente file per i tratti

codiceC1_codiceAG1_SG (SegmentID, ClassREF, GEOID, codiceA1,codiceA2)

Attributo a sottoaree su CP/CXsurface2D

Gli archi che rappresentano i confini delle sottoaree sono memorizzati nello stesso shape ITx_E_2D (ITx_E_3D) nel quale sono memorizzati gli edge della componente spaziale a cui si riferiscono.

Ogni insieme di attributi a sottoaree definito su un attributo geometrico produrrà un file .dbf il cui nome è dato dalla concatenazione <codice alfanumerico classe>_<codice alfanumerico attributo geometrico>**_SR (**ad es., **AC_VEI_ AC_VEI_SUP**_SR.dbf).

Analogamente ai tratti un record del file può contenere la singola sottoarea minima che condivide gli stessi valori per tutti gli attributi a sottoarea o un insieme di sottoaree minime (in base alla scelta effettuata in fase di generazione del mapping); nel primo caso la geometria associata ad ogni riga è una CPSurface2D, mentre nel secondo caso sarà un CXSurface2D.

Il file .dbf sarà composto dai seguenti campi:

- SubRegID: identificatore univoco della combinazione di valori degli attributi a sottoarea,
- ClassREF: contiene l'identificatore ClassID dell'oggetto della classe a cui fanno riferimento gli attributi a sottoarea
- GEOID: identificatore della geometria della sottoarea o delle sottoaree minime.
- tutti gli attributi a sottoarea definiti sull'attributo geometrico; il nome di questi attributi è composto dal codice alfanumerico associato all'attributo a sottoarea.

Analogamente ai tratti si noti che, poiché l'unione delle sottoaree definite sulla geometria di una componente spaziale produce la geometria della componente spaziale stessa, la componente spaziale derivabile è omessa dal corrispondente file .dbf della classe. Si ricorda che ciò implica che nel caso in cui l'attributo a sottoaree sia opzionale si debbano generare anche le sottoaree associate al valore NULL.

Data una classe C1(AG1: GU_CPSurface2D, A1: int aSottoaree su AG1, A2: int aSottoaree su AG1,…) il mapping produce i seguenti file per la descrizione della classe e della componente spaziale AG1:

IT_codice_ E_2D (primid)

IT_codice_E_2D_COMP (primid, geoid)

codiceC1(ClassID, …)

e il seguente file per le sottoaree

codiceC1_codiceAG1_SR (SubRegID, ClassREF, GEOID, codiceA1,codiceA2)

Attributi a tratti sul contorno 2D di una superficie 2D

I tratti sono memorizzati nello stesso shape ITx_E_2D nel quale sono memorizzati gli edge della componente spaziale a cui si riferiscono. Il mapping è equivalente a quello dei tratti.

Si noti che, poiché l'unione dei tratti definiti sui confini della geometria di una componente spaziale produce la geometria della frontiera della componente spaziale stessa, la componente spaziale

derivabile è omessa dal corrispondente file .dbf della classe. Si ricorda che ciò implica che nel caso in cui l'attributo a tratti sia opzionale si debbano generare anche i tratti associati al valore NULL. Nel caso in cui siano definiti sia gli attributi a sottoarea che gli attributi sul contorno di una superficie, si privilegerà la scelta dei tratti per la generazione della geometria della componente della classe.

Data una classe C1(AG1: GU_CPSurface2D, A1: int aTratti sul contorno 2D di AG1, A2: int aTratti sul contorno 2D di AG1,…) il mapping produce i seguenti file per la descrizione della classe e della componente spaziale AG1:

IT_codice_ E_2D (primid)

IT_codice_E_2D_COMP (primid, geoid)

codiceC1(ClassID,…)

e il seguente file per i tratti

codiceC1_codiceAG1_SG(SegmentID, ClassREF, GEOID, codiceA1, codiceA2)

Attributi su surface B3D

La geometria B3D viene descritta attraverso la rappresentazione della componente B3D della superficie, ossia attraverso la curva 3D che descrive gli anelli 3D che delimitano la superficie.

1. Attributi a tratti sul contorno3D.

   La geometria dei tratti è memorizzata nello stesso shape ITx_E_3D della componente spaziale sulla quale è definito questo attributo. Si applicano quindi le regole di mapping e l'osservazione sulla derivazione descritte precedentemente per il mapping degli attributi a tratti.

2. Attributi a sottoarea.
   2.1 Implementazione delle sottoaree in 3D, ossia definendone gli anelli di confine delle sottoaree come curve 3D. In questo caso le sottoaree sono rappresentate da anelli e non da superfici e gli anelli sono memorizzati nello stesso shape ITx_E_3D che memorizza la componente B3D della componente spaziale su cui è definito l'attributo a sottoarea. Pertanto il mapping è equivalente a quello degli attributi a tratti. Poiché l'unione dei confini delle sottoaree (tolti gli edge interni alla superficie) definite sulla geometria della componente spaziale produce la geometria della componente spaziale stessa, la componente spaziale derivabile è omessa dal corrispondente file .dbf della classe. Si ricorda che ciò implica che nel caso in cui l'attributo a sottoaree sia opzionale si debbano generare anche le sottoaree associate al valore NULL. Si noti che nel caso in cui siano definiti sia gli attributi a sottoarea implementati in 3D che quelli a tratti sul contorno 3D si privilegerà la scelta dei tratti per la generazione della componente spaziale.
   2.2 Implementazione delle sottoaree in 2D. In questo caso gli attributi a sottoarea sono esplicitati attraverso i poligoni nel 2D e pertanto si devono avere sia lo shape ITx_E_3D

contenente gli edge 3D della componente B3D che implementa la componente spaziale e sia lo shape ITx_E_2D contenente gli edge 2D che descrivono i confini delle singole sottoaree;

Data una classe C1(AG1: GU_CPSurfaceB3D, A1: int aSottoaree su AG1, A2: int aSottoaree su AG1,…) il mapping produce oltre ai seguenti file per la descrizione della classe e della componente spaziale AG1:

IT_codice_ E_3D (primid)

IT_codice_E_3D_COMP (primid, geoid)

codiceC1(ClassID, codiceAG1(geoid),…)

anche i seguenti file per le sottoaree

IT_codice_ E_2D (primid)

IT_codice_E_2D_COMP (primid, geoid)

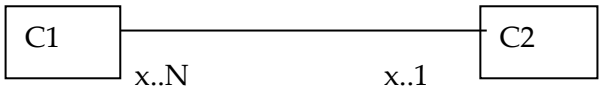
codiceC1_codiceAG1_SR(SubRegID, ClassREF, GEOID, codiceA1, codiceA2)

3. Attributi a tratti sul contorno 2D. I tratti sono memorizzati nello shape ITx_E_2D come per le sottoaree implementate nel 2D. Valgono le stesse regole degli attributi a tratti sulle curve 2D, eccetto per il nome del file .dbf dei tratti che sarà composto da: codiceclasse_codiceattributogeometrico_SG_2D.

Mapping delle associazioni

1. Associazione 1xN, 1x1 implementate con attributi di ruolo nel file .dbf di una delle due classi.

   Dato lo schema concettuale:



   Se N>1 sarà aggiunto un attributo nel file .dbf della classe C1 chiamato col codice alfanumerico del ruolo di C1 nell'associazione che conterrà per ogni oggetto della classe C1 l'identificatore ClassID dell'oggetto di C2 referenziato nell'associazione. Nel caso in cui N=1 si sceglie la classe col ruolo alfabeticamente minore per aggiungere il ruolo che implementa l'associazione.

2. Un'associazione NxN senza attributi richiede un ulteriore file .dbf . chiamato **A_** <codice alfanumerico classe 1>_<nome ruolo>_<codice alfanumerico classe 2> Il file è composto da due attributi chiamati codice alfanumerico classe 1 e codice alfanumerico classe 2 e contenenti gli identificatori ClassID degli oggetti delle classi coinvolti in un'associazione. La classe 1 è considerata quella che appare come prima in ordine alfabetico.

3. Un'associazione NxN con attributi richiede un ulteriore file .dbf chiamato **A_**codice alfanumerico associazione. Questo file contiene i due attributi di identificazione come nel caso precedente e aggiunge a questi gli attributi dell'associazione col nome chiamati con i relativi codici alfanumerici. In presenza di attributi multivalore o di tipo datatype ci si comporta come nel caso delle classi sostituendo ovviamente il codice alfanumerico della classe con quello dell'associazione preceduto dal prefisso **A_**.

Mapping degli aggregati generici (GU_Aggregate2D o **GU_Aggregate3D)**

Per ogni componente spaziale di tipo aggregato le primitive componenti saranno suddivise tra gli shapefile 2D o 3D di un insieme topologico in base alla dimensione dell'aggregato. Viene inoltre definito un ulteriore file .dbf che permette la descrizione dei componenti di un aggregato chiamato codice alfanumerico classe_codice alfanumerico attributo geometrico_AG. Si omette la componente spaziale corrispondente nel file .dbf della classe.

Data la classe C(…, AG1:GU_Aggregate2D,…) si generano i seguenti file:

IT_codice_E_2D (…, primid)

IT_codice_E_2D_COMP (primid, geoid)

IT_codice_N_2D (…, primid)

IT_codice_N_2D_COMP (primid, geoid)

codiceC(ClassID,…)

codiceC_codiceAG1_AG(ClassREF, GEOID, tipo)

dove:

- **ClassREF:** contiene l'identificatore ClassID dell'oggetto a cui si riferisce l'aggregato

- **GEOID:** identificatore assegnato al singolo componente dell'aggregato
- **TIPO** codice che indica la tipologia dell'aggregato: line, point, poly

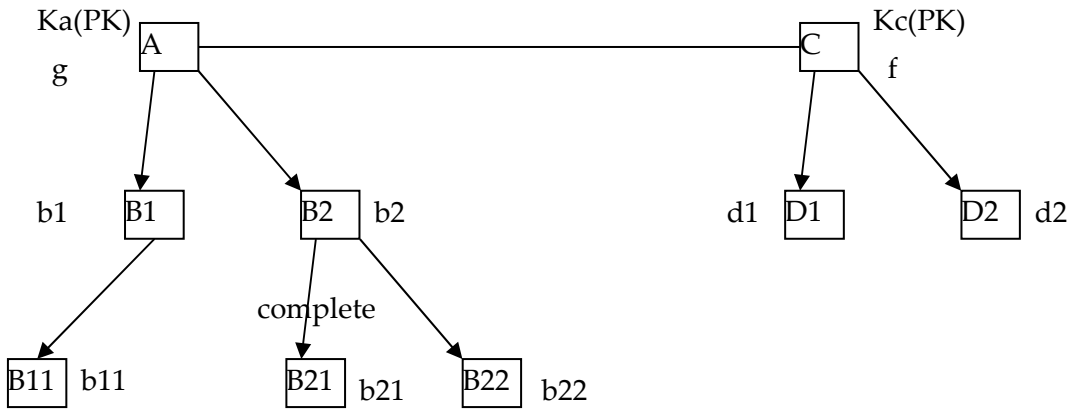
Si noti che nel caso in cui l'aggregato contenga più di un poligono semplice è necessario definire ciascun poligono semplice come un componente dell'aggregato, mentre per i punti e le curve il componente può essere sia il punto/curva semplice o un multipunto/multicurva. Inoltre si noti che l'attributo tipo serve per distinguere gli edge che compongono curve da quelli che compongono i poligoni dell'aggregato.

Mapping in presenza di collassamento

Per ogni attributo sul quale è previsto il collassamento il mapping definisce nel file .dbf della classe un attributo che riporta il geoid della geometria principale della componente spaziale collassata (poligoni per una componente spaziale di tipo surface) e si aggiungono al massimo i due attributi per contenere le componenti collassate nei quali si riporta il geoid della geometria collassata corrispondente. Il nome degli attributi generati sarà composto dal codice alfanumerico dell'attributo geometrico seguito dal suffisso _P e _L per la componente collassata a punti e curve.

Mapping delle gerarchie di classi

Data una gerarchia di classi come quelle in Figura presente in **S** si applica la seguente regola di mapping:



Le classi della gerarchia astratte (può essere la parte alta della gerarchia) e ogni superclasse che si collega ai figli con gerarchia *complete* non viene implementata (implementazione nei figli).

Le altre classi sono implementate seguendo le regole di mapping della classe normale definiranno i propri attributi specifici e includeranno tutti gli attributi e i ruoli (attributi che implementano nella classe un'associazione uno a molti o uno a uno) ereditati dagli antenati nella gerarchia con i vincoli

associati (not null, vincoli delle chiave primarie, vincoli sulle cardinalità dei ruoli,…); anche gli attributi ereditati multivalore e gli attributi ereditati di tipo DataType vengono implementati come fossero della classe, quindi con una tabella dedicata (se necessario).

Il popolamento dei file delle classi segue le seguenti regole: il file che contiene le istanza di una classe A padre di una gerarchia si devono inserire i soli oggetti istanze solo di A, ossia un oggetto di una sottoclasse di A (B1, B2) non è materializzato anche nel file di A; da ciò si deduce che il file di A avrà istanze solo se la gerarchia, di cui A è padre, viene dichiarata *incomplete*. Altrimenti tutte le istanze di A si ottengono dall'unione delle istanze di B1 con le istanze di B2.

Le associazioni molti, molti definite su una superclasse A sono ereditate dalle classi figlie. Per memorizzare le istanze di relazione si genera un unico file di associazione per tutte le classi, il quale raggruppa tutte le istanze delle associazioni tra due classi (una di una gerarchia e una dell'altra).

Mapping e popolamento

Il mapping ignora le classi non popolate a nessun livello di scala e le associazioni e/o ruoli che le coinvolgono. Se una classe è popolata ad almeno un livello di scala, ma non a tutti tutte le associazioni e/o ruoli che le coinvolgono diventano opzionali (ad es., gli edifici sono composti da almeno 1 unità volumetrica e questa è popolata ad un solo livello di scala, l'associazione di composizione diventerà opzionale prima di effettuare il mapping della specifica).

Appendice. **Richiami sulle caratteristiche degli shapefile**

Lo shapefile prevede un attributo geometrico memorizzato in un file XX.shp supportato da un file indice XX.shx e gli attributi descrittivi monovalore memorizzati in un file XX.dbf in formato dBASE. La corrispondenza tra una geometria e i propri attributi descrittivi è di tipo 1x1, basata sul record number e quindi è posizionale.

La precisione per i numeri reali è data dal FP 64bit.

Attributi geometrici
Il file della geometria ha un header nel quale il campo shape type riporta il tipo di geometria contenuta nel file.

Si riportano i tipi di nostro interesse: nullshape, per il 2D point, Polyline, Polygon, Multipoint e per il 3D pointZ, PolylineZ, PolygonZ, MultipointZ, Multipatch.

Ogni record del file riporta come informazione associata il tipo di appartenenza.

Si noti che:

- **nullshape** significa che lo shape ammette record con geometria vuota che per noi corrisponde a dire se si ammette NULL. Non si tratta quindi non di una definizione di tipo, ma di un constraint. Sfruttando la replicazione del tipo nei record, il modello shape afferma però che i record contenuti in un file dichiarato nullshape devono essere di tipo nullshape oppure di uno solo degli altri tipi.

- le curve e superfici semplici e complesse sono gestite all'interno dello stesso tipo (il tipo multipatch non considerato permette la definizione di anelli in 3D in modo simile a polygonZ).

  - **record polyline, polylineZ**: è composto da un certo numero di parti: 0 è un errore, 1 significa curva semplice e n>1 significa collezione di curve semplici senza vincoli topologici tra le curve. Una part è una curva semplice connessa con eventuali vertici duplicati (non esplicita, ma è sottinteso che sia composta da almeno 2 vertici distinti).

  - **record polygon, polygonZ**: afferma che un poligono è un insieme di anelli (curva con almeno 4 vertici, chiusa e simple) e inoltre richiede che un poligono sia "clean" ed esprime questa proprietà in modo non preciso e in termini di segmenti più che di frontiere come nel simple feature model e risulta quindi ambigua: vietata intersezione tra segmenti di due anelli (evidente se uno dei due anelli è un buco, ma se entrambi sono esterni significa che non si accetta una collezione con poligoni sovrapposti definiti dirty polygons) e vietato che anelli si tocchino se non in punti (buco con shell o tra shell). Dal punto di vista tecnico un record è un contenitore non ordinato di parti ciascuna delle quali descrive un anello. Gli anelli che sono Frontiere esterne si distinguono dalle parti che sono frontiere interne per l'orientamento della sequenza di coordinate (orientamento orario per frontiera esterna e opposto per buchi interni). Per distinguere un poligono da una collezione di poligoni è necessario verificare quanti anelli abbiano un orientamento orario.