

# L'aggiornamento cartografico dei Database Territoriali Integrati

L'evoluzione delle tecnologie SDI, dalla cartografia tradizionale ai DB-topografici con la Spatial Topology Management

di Giovanni Corcione

Il 2 dicembre 2005, il comune di Bolzano ha organizzato, presso la Libera Università di Bolzano, un seminario sul progetto di *Innovazione e posizionamento strategico del Servizio Informativo Territoriale*, finanziato dal Fondo Sociale Europeo. Al Progetto hanno collaborato: l'Ufficio-SIT del Comune Di Bolzano, il Consorzio dei Comuni della Provincia di Bolzano, il Politecnico di Milano (Dipartimenti di Elettronica ed Ingegneria Idraulica), Oracle Italia, Abaco di Mantova e Cefriel di Milano (*ICT Center of Excellence For Research, Innovation, Education and industrial Labs partnership*). Nella sessione delle demo dedicate allo *Studio dei meccanismi di aggiornamento del Database Geografico* Oracle Italia ha presentato una sperimentazione per la Procedura di Aggiornamento Cartografico di un generico DataBase Territoriale Integrato (DBTI). Al termine della sessione, il Cefriel di Milano ha presentato la sperimentazione di un applicativo per l'aggiornamento di un database topografico, implementato sulla base della Procedura Di Aggiornamento Sperimentale Oracle e della suite DbMAP ASJ della Abaco di Mantova.

Nell'articolo descriveremo come la Procedura Di Aggiornamento è stata realizzata utilizzando il DBMS Oracle 10g Rel.2.0 e l'infrastruttura del *Persistent Topology Data Model* dell'opzione Spatial. Infine, riporteremo il codice sorgente delle procedure PL/SQL e query SQL più significative adoperate dalle componenti client/server di DbMAP ASJ per l'impianto della "Procedura Sperimentale Di Aggiornamento Interattivo".

## Ambito ed Architettura di un DBTI

L'architettura di riferimento per la sperimentazione della *Procedura Di Aggiornamento* è quella del Sigmater, un progetto e-gov di cooperazione applicativa tra i più significativi in Italia. In questa architettura i dati interscambiati con le porte di dominio vengono centralizzati ed integrati proprio in un DataBase Territoriale Integrato.

I nuovi dati transitati - in particolare gli oggetti topografici - vengono depositati nelle *anticamere* del DBTI per essere bonificati; in seguito, sono sottoposti a processi di aggiornamento ed infine, integrati in maniera consistente con gli oggetti territoriali già presenti nel sistema: in questo modo i *Servizi General Purpose*, così come i *webService* degli enti cooperanti, riceveranno dati sempre aggiornati dal nostro DBTI.

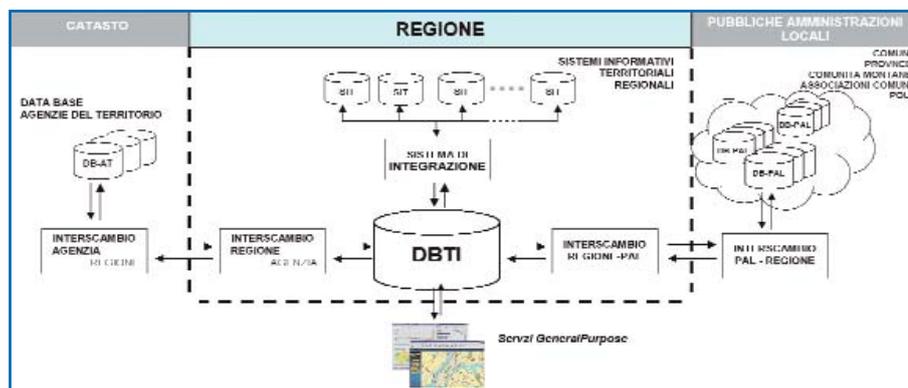


Figura 1 - Architettura generale e flusso dati del progetto SIGMATER ([www.sigmater.it](http://www.sigmater.it))

## Requisiti e Presupposti per la Procedura di Aggiornamento

Si ipotizza che:

- ✓ Il DBTI sia in grado di tenere sempre aggiornati i tre livelli di massima granularità areale della CTC: le Aree\_Verdi, le Aree\_Unità\_Volumetriche e le Aree\_Stradali, memorizzate in apposite tabelle spaziali che denomineremo *tabelleTopografiche*.
- ✓ Le tabelleTopografiche siano aggiornate da *nuovi* oggettiTopografici provenienti dalle fonti esterne cooperanti col sistema di interscambio.
- ✓ Gli oggettiPreesistenti delle tabelleTopografiche abbiano solo geometrie di tipo poligono conformi all'OpenGIS, tra loro adiacenti o disgiunti, senza alcuna sovrapposizione e nessuna intersezione.
- ✓ I nuoviOggettiTopografici - depositati nella tabella NEW\_OF - siano definiti da un identificativo record, da un poligono conforme all'OpenGIS, dal tipo di area topografica e da un identificativo di sessione di aggiornamento.
- ✓ L'aggiornamento degli oggettiTopografici consenta due distinte modalità operative: automatica e interattiva

## Impianto del DBTI

Nel DBTI vengono memorizzate le aree delle tabelle Topografiche: Aree\_Verdi, Aree\_Unità\_Volumetriche e Aree\_Stradali.

Queste aree, originariamente nel formato shape, vengono importate nelle tabelle topografiche del DB o tramite tool DbMAP ASJ o tramite il programma SHP2SDO.EXE. Questo eseguibile converte shapeFile in file di input per il programma SQLDR di caricamento massivo di dati in un DBMS.

Ciascuna delle tabelle Topografiche è stata definita da un identificativo univoco di oggetto e da una geometria SDO\_GEOMETRY.

Nella vista USER\_SDO\_GEOM\_METADATA dei metadati della Spatial sono state registrate le *tabelle Topografiche* (Aree\_Verdi, Aree\_Unità\_Volumetriche e Aree\_Stradali) per ciascuna delle quali sono stati impostati i seguenti parametri: GaussBoaga-Est come sistema di riferimento, geometrie bidimensionali in coordinate X e Y, la tolleranza di 0.01 metri sia per la dimensione X che per la Y.

Tramite le funzioni SQL di *validate* della Spatial è stata verificata la conformità OGC dei poligoni delle *tabelle Topografiche* e sempre con funzioni SQL della Spatial, in maniera massiva, sono state corrette le anomalie più ricorrenti:

```
select SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT(geom, 0.01)
from AREE_STRADALI;
```

```
update AREE_VERDI a set a.geom = SDO_UTIL.RECTIFY_GEOMETRY(geom, 0.01)
where c.id = 99999; --id di geometria invalida
```

Le stesse tabelle sono state poi sottoposte al processo di *indicizzazione spaziale*, affinché gli oggetti Topografici potessero essere selezionati con l'SQL tramite *relazioni topologiche* tra gli oggetti stessi. Per esempio, per accertare che gli oggetti Topografici costituissero una copertura areale, senza che vi fosse tra loro alcuna sovrapposizione/intersezione geometrica, è stato utilizzato proprio l'operatore topologico SDO\_OVERLAPBDYINTERSECT della Spatial ed esattamente nelle seguenti modalità:

```
select count(*) from AREE_VERDI a, AREE_STRADALI b
where SDO_OVERLAPBDYINTERSECT(a.geom,b.geom)='TRUE';
```

```
select count(*) from AREE_VERDI a, AREE_VERDI b
where SDO_OVERLAPBDYINTERSECT (a.geom,b.geom)='TRUE'
```

Dopo la validazione dei dati, si è constatato che gli oggetti Topografici, messi a disposizione dal Consorzio Comuni Di Bolzano, sono tutti risultati geometricamente validi, OGC compliant, tra loro adiacenti e senza alcuna sovrapposizione areale.

## Trasposizione Topologica degli Oggetti Topografici

Per avere un continuo areale, ovvero senza sovrapposizione/intersezione tra oggetti Topografici, si è stabilito di trasporre le geometrie SimpleFeature nel Persistent Topology Data Model della Spatial.

Tale *schema data model* consente di memorizzare in maniera *persistente* la Topologia nel DBTI. Infatti, mediante

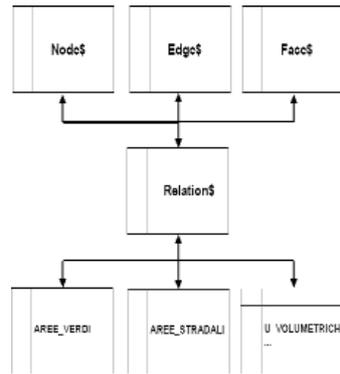


Figura 2

Oracle10g Spatial Topology Management, gli oggetti Territoriali si possono memorizzare sotto forma di primitive topologiche e sotto forma di oggetti del DB di tipo *sdo\_topo\_geometry*. Le primitive topologiche sono costituite da nodi, archi e aree, mentre le *sdo\_topo\_geometry* sono oggetti composti da queste primitive topologiche. Infine la tabella RELATION\$ *mantiene* la relazione tra ciascun oggetto *sdo\_topo\_geometry* e le primitive topologiche che lo compongono, così come schematizzato in *figura 2*.

Le primitive topologiche (nodi, archi e aree), essendo *normalizzate*, mantengono i dati consistenti in modo che non esista ridondanza di bordi tra aree. In questo modo la consistenza topologica viene risolta direttamente dallo schema dati.

Per esempio, nel modello simpleFeature, il bordo in comune di due particelle adiacenti (G1 e G2 in *figura 3*) viene memorizzato nella prima e nella seconda area. Così, quando si modifica la geometria di una, per garantire l'adiacenza di entrambe le particelle, è necessario modificare anche la geometria dell'altra. Nel caso di topologyFeature l'arco in comune di due particelle viene memorizzato una sola volta contribuendo a formare sia la prima che la seconda particella: il cambiamento dell'arco "e2" corrisponde automaticamente alla modifica di entrambe le feature G1 e G2, e al tempo stesso, al mantenimento della adiacenza topologica.

Le *StoredProcedure Topologiche*, disponibili sia come PL/SQL nel DBMS che come package di classi Java per applicazioni client, consentono l'editing e la *validazione* delle primitive topologiche. Le nuove TopoAPI della release 10.2 del DBMS Oracle, forniscono funzioni che *traspongono* direttamente geometrie simpleFeature in geometrie TopologyFeature, già *relazionate* alle primitive topologiche.

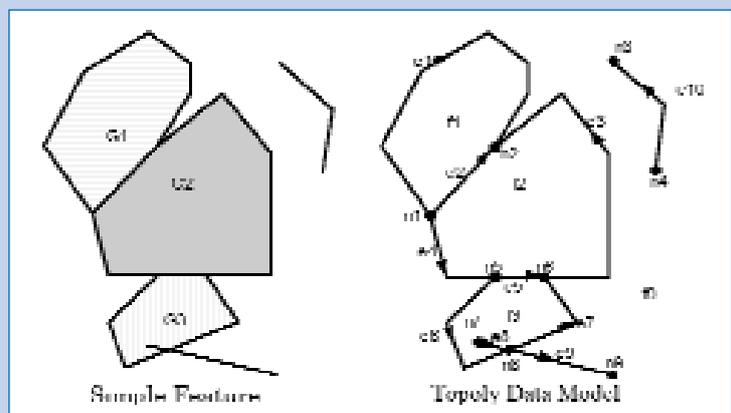


Figura 3

Trasporre le geometrie simpleFeature delle tabelle Topografiche in feature Topologiche del *Persistent Topology Data Model* è veramente facile con la nuova funzione SDO\_TOPO\_MAP.CREATE\_FEATURE:

Nell'esempio che segue, un semplice codice sorgente per la trasposizione delle simpleFeature di Aree Stradali in corrispondenti topologyFeature:

```

DECLARE
BEGIN
for stradeRec in
    (select ID, GEOM from COPERTURA where tipo = 'STRADA') LOOP

INSERT INTO AREE_STRADALI VALUES
(stradeRec.id
SDO_TOPO_MAP.CREATE_FEATURE('T_VANGA','T_AREE_STRADALI','TGEOM',
stradeRec.geom));
END LOOP;
END;
/

```

Spatial Topology Management può supportare più di uno schema topologico: questa peculiarità consente al sistema l'integrazione di topologie appartenenti a diverse rappresentazioni cartografiche ( come la CT-Comunale e la cartografia Catastale).

### La Procedura di Aggiornamento

Non appena i nuovi oggetti topografici transitano nel sistema centrale, vengono validati e depositati nella tabella NEW\_OF per essere definitivamente trasferiti nel DBTI tramite la *Procedura di Aggiornamento Cartografico*.

Le funzionalità Spatial e di Topology Management costituiscono l'infrastruttura software di base per la *Sperimentazione dei Processi Di Aggiornamento*, questi ultimi realizzati appositamente per modellare la procedura sia nella modalità automatica che interattiva.

La *modalità automatica* permette di aggiornare le tabelle topografiche della CTC sottoponendo le aree dei nuovi oggetti territoriali alla sequenza di processi riportata nel DataFlow di *figura 4*.

Si controlla prima l'*impatto topologico* delle nuove aree rispetto a quelle preesistenti; poi, se la consistenza topologica viene rispettata, si creano le feature topologiche per le nuove aree; infine, con l'approvazione dell'aggiornamento, si cancellano le aree preesistenti impattate, si rimpiazzano le nuove aree e si *ripulisce* la topologia.

La *fase di impatto*: l'impatto delle nuove feature con il DBTI, viene controllato in base alla relazione topologica che esiste tra il confine ricoperto dalle nuove aree - *Confine Nuove Aree* - e le sottostanti aree preesistenti.

L'aggiornamento viene accettato solo se il *Confine Nuove Aree* tocca e/o ricopre completamente le aree preesistenti ma senza intersecarle.

Il procedimento sperimentato, rappresentando nel grafico di *figura 5*, è quindi il seguente:

- a) Si determina il confine delle nuove aree servendosi della funzione di aggregazione della Spatial:

```

select SDO_AGGR_UNION(SDOAGGRTYPE(c.geom, 0.001)) into
ConfineNuoveAree from NEW_OF C
where C.PROCESSNUMBER= 5;

```

- b) Tramite l'operatore topologico *coveredBy*, si selezionano dalle tabelle Topografiche (Aree Stradali, Aree Verdi e Unità Volumetriche) tutte le aree Preesistenti ricoperte dal *ConfineNuoveAree*. Le aree preesistenti vengono parcheggiate nella tabella temporanea AREE\_PRESISTENTI

```

select id, geom from AREE_STRADALI
where SDO_COVEREDBY(geom, ConfineNuoveAree) = 'TRUE';

```

...eseguita per tutti i livelli topografici

- c) Analogamente al *ConfineNuoveAree*, si determina il *Confine Aree Preesistenti* aggregando le aree preesistenti impattate e precedentemente memorizzate nella tabella AREE\_PRESISTENTI

- d) Nel caso in cui il *ConfineNuoveAree* coincida con il *ConfineAreePreesistenti*, si procede col processo di creazione nuovi oggetti

```

Select SDO_GEOM.RELATE (ConfNuoveAree, 'determine', ConAreePreesistenti
, 0.01 ) from TabelaCONFINI;
'EQUAL'

```

- e) Se, invece, i due confini risultano *inconsistenti* si interrompe la procedura nella modalità automatica e si procede, eventualmente, nella modalità interattiva.

La *fase di creazione*: superata la verifica di impatto, i nuovi oggetti vengono associati ai relativi livelli topografici e le geometrie vengono trasposte in SDO\_TOPO\_GEOMETRY tramite la funzione *SDO\_TOPO\_MAP.CREATE\_FEATURE*. Durante il processo si definisce il TOPO\_MAP (una cash-memory del Topology Management) nel quale si memorizzano tutte le primitive topologiche incluse nel rettangolo di ingombro del *ConfineAreePreesistenti*. Il TOPO\_MAP offre il vantaggio di

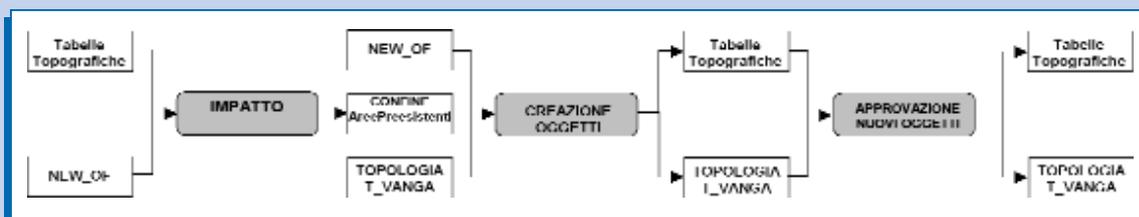


Figura 4

bloccare (lock) automaticamente le primitive topologiche, in modo che le stesse primitive siano soltanto lette e non modificate da altri processi contemporanei.

**La fase di approvazione:** questo processo cancella gli oggetti topografici preesistenti topologicamente inclusi nel *ConfineAreePreesistenti* e li rimpiazza coi nuovi oggetti topografici generati dal processo di Creazione. Infine, la topologia viene ripulita eliminando tutte le primitive topologiche appartenenti agli oggetti topografici precedentemente cancellati: questa operazione assicura un corrispondenza univoca tra le primitive topologiche *FACE* e le aree delle tabelle topografiche.

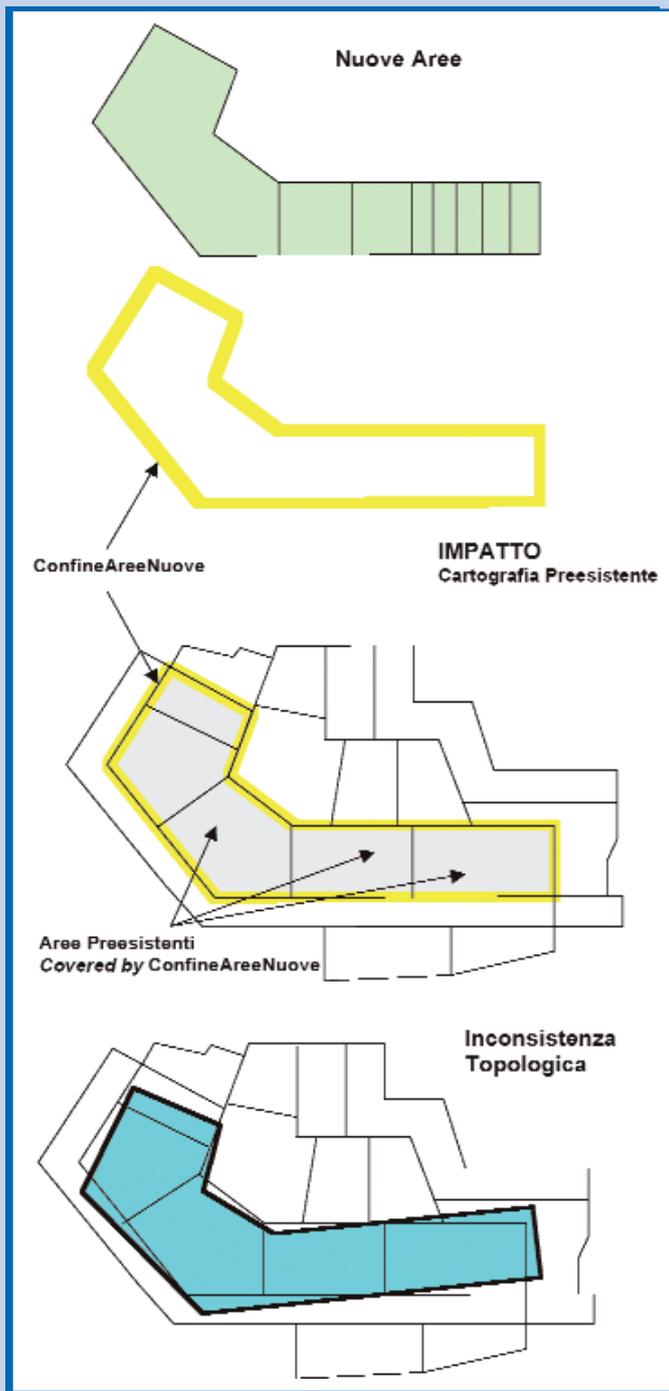


Figura 5 - Il complesso processo di aggiornamento della topologia descritto nel presente paragrafo (fasi a, b, c, d)

## La Procedura di Aggiornamento Interattivo

I processi della modalità di aggiornamento automatico possono essere attivati ed orchestrati interattivamente da un applicativo client. Infatti, come nella procedura automatica, viene gestito l'impatto topologico delle nuove aree e l'utente decide, interattivamente, se procedere o meno; le nuove aree e le preesistenti vengono frazionate in tante microaree; infine, le microaree vengono selezionate dall'applicativo client per comporre in altrettanto nuovi oggetti topografici.

La fase di *impatto interattivo*: esegue gli stessi controlli previsti dal processo di Impatto della procedura automatica, con la differenza che, nella modalità interattiva, è possibile continuare l'aggiornamento con i processi successivi anche se si verifica inconsistenza topologica. Inoltre, il processo seleziona le aree preesistenti che hanno una *qualsiasi interazione* con il confineAreeNuove:

```
select id, geom from AREE_STRADALI
where SDO_ANYINTERACT(geom, ConfineNuoveArea) =
'TRUE'; ...eseguita per tutti i livelli topografici
```

La fase di *frazionamento microaree*: tramite la *ADD\_POLYGON\_GEOMETRY*, del package *Topo-API*, il processo fraziona le aree dei nuovi oggetti e le aree preesistenti sottostanti in *MicroAree*: aree costituenti il livello di massima granularità areale (vedere figura 6). Tali aree vengono fornite sia nella versione *sdo\_topo\_geometry* (*topoFeature*) che nella versione *sdo\_geometry* (*simpleFeature*) affinché siano accedute dalle piattaforme GIS che già si interfacciano con la Saptial.

La fase di *editing oggetti*: viene invocato dall'applicativo client per editare *interattivamente* i nuovi oggetti territoriali. Dalla pseudo finestra grafica dell'applicativo client, si selezionano le microaree per consegnarle alla funzione "InsertUpdateTopology": le microaree vengono composte in un oggetto univoco (di tipo *sdo\_topo\_geometry*) che viene inserito in una delle tabelle topografiche.

InsertUpdateTopology fa parte del package di Aggiornamento sviluppato apposta per le attività di *Studio dei Meccanismi Di Aggiornamento del Database Geografico* supportate dal Comune di Bolzano.

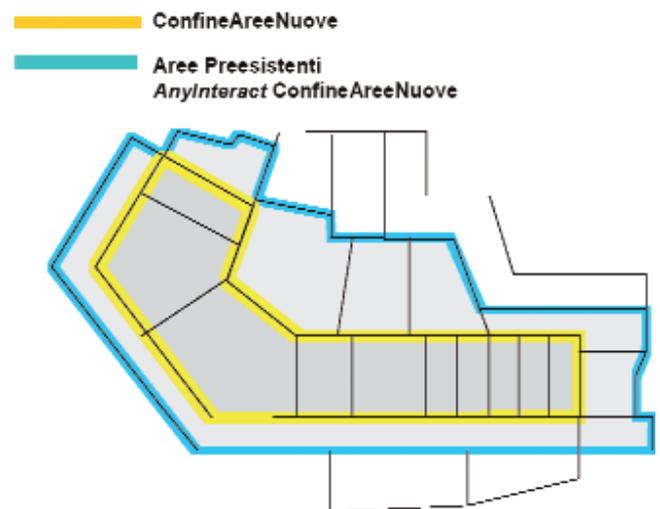


Figura 6 - Effetto del processo del frazionamento in microaree

## Applicativo Sperimentale di Aggiornamento Interattivo

Abaco e Cefriel hanno realizzato l'applicativo client interattivo basato sul package sperimentale di aggiornamento cartografico, *DbMAP ASJ* e l'application server *Oracle AS 10g*.

Questo applicativo (*web-based*) attiva, interattivamente, i processi dell'aggiornamento cartografico residenti nel DBTI sperimentale.

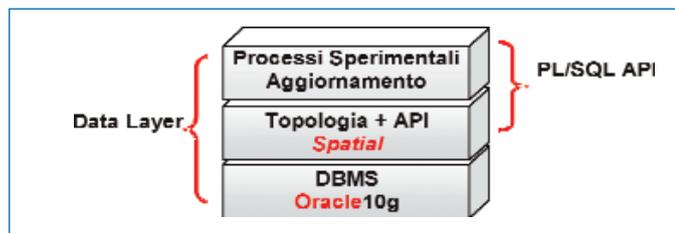


Figura 7 - Il layout delle tecnologie impiegate nel progetto

Dal browser viene scelta una delle diverse sessioni di aggiornamento presenti nella tabella *NEW\_OF* e si attiva il processo di Impatto per verificare la consistenza topologica dei nuovi oggetti; in seguito, viene presentata la pagina web principale dell'Applicativo.

Dalla pagina web, indicata in *Figura 8*, con il pushButton *frazione* si attiva il processo di frazionamento che genera le *microaree*. Le altre opzioni sono di ausilio all'editing: ovvero, consentono di selezionare le microaree del nuovo oggetto topografico e il tipo di area topografica. Al termine della selezione, viene proposta un'ulteriore frame di aggiornamento ove inserire i valori degli attributi alfanumerici dell'oggetto topografico considerato.

Per ottenere una nuova area stradale composta dall'unione delle microaree 130 e 131, l'applicativo ABACO attiva la funzione *InsertUpdateTopology* tramite il seguente comando SQL:

```
InsertUpdateTopology(1,listface(130,131), 0, 18138, 6, 999).
```

Dopo l'esecuzione del comando, nella tabella *AREE\_STRADALI*, esisterà un nuovo record definito dall'identificativo 19138, dal campo *APPARTENENZA* posto a 999 e da una *topo\_geometry* relazionata alle primitive topologiche: *face 130 e 131*.

## Conclusioni

L'aggiornamento cartografico rappresenta il punto più debole del DBTI di un sistema di interscambio. Come ha dimostrato la sperimentazione, la transazione di oggetti territoriali avviene al termine di un intero flusso di processi, più o meno complessi, operanti su una SDI. Le alte prestazioni del *DBMS Oracle 10g r.2* e la sua *Spatial Data Option*, in grado di integrare tutti i modelli geometrici del DBTI, hanno fornito la piattaforma tecnologica, *spazialmente* evoluta, per impiantare una SDI capace di gestire l'Aggiornamento Cartografico

L'*interoperabilità* della piattaforma Oracle Spatial, l'aderenza della Suite *ABACO DbMAP ASJ* agli standard applicativi (*J2EE, SQL, geometrie OGC*), hanno consentito successivamente di ampliare l'Aggiornamento Cartografico nella modalità interattiva *web-based*.

*DbMAP ASJ*, grazie alla sua facilità di interfacciamento verso i DBMS, ha potuto editare le *TopologyFeature* del DBTI invocando direttamente, via *sql*, le funzionalità della *Spatial*.

## Approfondimenti

Sul sito internet all'URL [www.geo4all.it/geomedia/geonotes](http://www.geo4all.it/geomedia/geonotes) sono disponibili informazioni di approfondimento sull'articolo, tra cui la raccolta completa del codice *PL/SQL* usato negli esempi citati nel corpo dell'articolo, e il documento integrale che distribuito durante la giornata del seminario citato in prefazione, permette ai lettori un nutrito approfondimento del tema trattato.

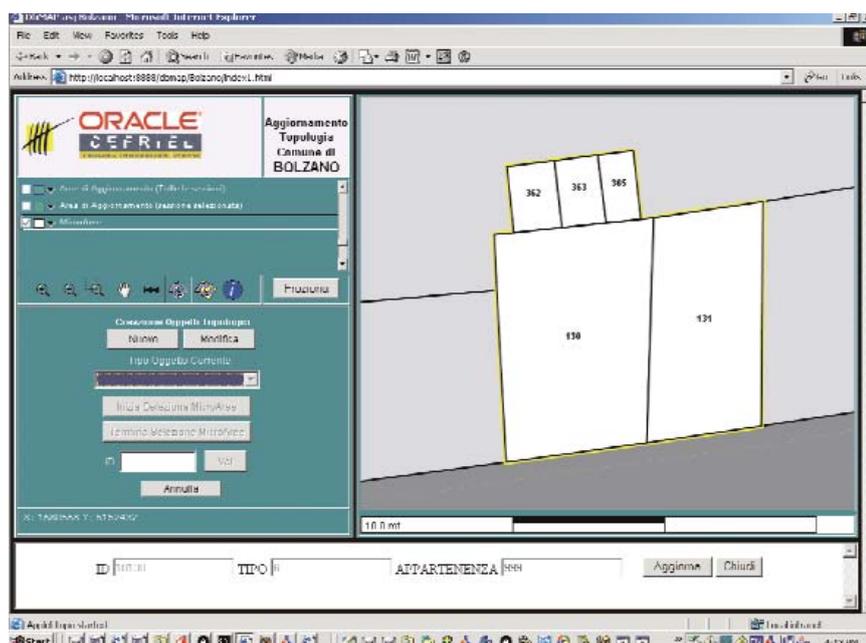


Fig. 8 - Nell'immagine una cattura di schermo del sistema

## Autore

GIOVANNI CORCIONE  
Giovanni.Corcione@oracle.com

SOLUTION TEAM  
Oracle Italia

