DAL FORMATO *shape* A UNA STRUTTURA *object oriented*: PROBLEMATICHE E VANTAGGI

Daniela Carrion(*), Daniele Marzi(**), Federica Migliaccio(*)

(*) DIIAR–Politecnico di Milano – P.zza L. da Vinci, 32 – 20133 Milano federica.migliaccio@polimi.it
(**) Comune di Vimercate – P.zza Unità d'Italia, 1 – 20059 Vimercate (MI) daniele.marzi@comune.vimercate.mi.it

Riassunto
Il crescente utilizzo della cartografia all'interno dei sistemi GIS impone delle riflessioni sulle caratteristiche di coerenza geometrica e logica che essa deve possedere al fine di consentire l'analisi territoriale. Il presente lavoro intende presentare le problematiche incontrate durante il passaggio dal formato shp, considerato uno standard di fatto per la memorizzazione del dato geografico GIS, ad una struttura *object oriented* in grado di gestire l'integrità spaziale del dato e la validazione dell'attributo associato ad esso.
Lo studio prevede la modellazione delle relazioni spaziali tra oggetti cartografici che popolano il *geodb* al fine di fornire una rappresentazione numerica del mondo più "vicina" alla realtà.
L'utilizzo del *personal geodatabase* (OODB) è stato sperimentato nell'ambito del progetto di potenziamento del SIT del Comune di Vimercate: sono state individuate e affrontate problematiche legate alla conversione delle geometrie del P.R.G. comunale vigente in formato GIS *oriented* al fine di poterle integrare con il patrimonio cartografico comunale esistente (in formato nativo ESRI) e di creare una base informativa necessaria per poter operare analisi territoriali anche complesse.

Abstract
Because of the increasing use of Cartography inside GIS it is important to carefully consider its geometric and logic coherence characteristics which make spatial analysis possible. The aim of this paper is to present the problems found in the migration from the shp format, a GIS standard, to an object oriented structure, which can manage the data spatial integrity and the assessment of its attributes.
The study concerns the modelling of spatial relationships between the geodb cartographic elements, in order to obtain a numeric representation of the world more "near to the real".
The use of the personal geodatabase (OODB) has been experimented while developing the project of the Vimercate Municipality GIS: the problems that have been highlighted and faced are related to the geometric transformation of the PRG in the GIS oriented format, to its merging with the pre-existent cartography (in ESRI format) and to the creation of the necessary database to perform complex environmental analysis.

Introduzione – le basi di dati ad oggetti

Un *geodatabase* può essere definito come un archivio integrato contenente dati cartografici e altre informazioni alfanumeriche in formato digitale, utilizzate nelle attività di lavoro di una Organizzazione o di un Ente.
La costruzione di un *geodatabase* può essere sintetizzata, dal punto di vista concettuale, come una sequenza di operazioni piuttosto semplici da definire:

- per ogni aspetto della realtà di cui si vogliono archiviare informazioni si definisce una ENTITA', caratterizzata da diversi ATTRIBUTI;
- ogni specifica entità deve essere identificata da un CODICE

- le entità (e quindi i loro dati) possono essere collegate fra loro ("messe in RELAZIONE") grazie al sistema di codifica;
- le entità georeferenziate devono riferirsi ad un unico sistema cartografico e di riferimento.

Come si vede, procedendo nel modo descritto sopra si struttura l'insieme di dati secondo il ben noto modello relazionale.
Il modello entità-relazioni fu originariamente sviluppato come mezzo per costruire modelli concettuali di basi di dati in maniera che questi poi potessero essere facilmente traducibili nei principali modelli logici. Ad esso hanno fatto seguito molti altri schemi di modelli concettuali: lo scopo era di fornire ulteriori livelli di astrazione per meglio rappresentare i fenomeni del mondo reale.
I DBMS[1] orientati ad oggetti rappresentano lo sviluppo più recente nella tecnologia delle basi di dati e hanno lo scopo di fornire strumenti per modellizzare tipi di dati più complessi, allo scopo di superare alcune restrizioni insite nei modelli relazionali.
Nel contesto della modellizzazione ad oggetti, si usa il termine oggetto (invece del termine "entità") per riferirsi a ciò che si vuole modellizzare del mondo reale. Un oggetto è formato sia dai dati che lo rappresentano sia dalle operazioni (metodi) che possono essere applicati all'oggetto.
Ogni oggetto è una istanza di una classe di oggetti, che definisce gli attributi e le operazioni per tutte le sue istanze: ogni istanza ha il suo identificatore univoco grazie al quale si può fare riferimento a quella istanza all'interno della base di dati.
Il concetto di modellizzazione ad oggetti è molto adatto per la rappresentazione di oggetti topografici, che rappresentano le entità fondamentali di una base di dati "topografica" (o geodatabase).

Principi della modellizzazione ad oggetti

Invece di fare riferimento a relazioni fra oggetti (o entità), gli schemi di modellizzazione ad oggetti (e la programmazione orientata ad oggetti) introducono metodi di astrazione che aiutano a distinguere vari tipi di relazione e che sono basati su una serie di principi. Cerchiamo di comprendere le basi di questi principi.
Un oggetto può essere definito come un'entità che possiede uno stato rappresentato da attributi (definiti come valori di variabili locali) e da un insieme di operazioni che possono essere eseguite sull'oggetto stesso. Nel processo di **classificazione**, oggetti individuali che sono dello stesso tipo vengono riconosciuti e, appunto, classificati come tali. Quindi ogni singolo oggetto appartiene a una classe che definisce il tipo di oggetto (si dice anche che un insieme di oggetti è classificato in un "tipo").
In molti casi, nel mondo reale le classificazioni sono di tipo gerarchico e ciò si riflette nella modellizzazione semantica per mezzo del processo di **generalizzazione**: i "tipi" sono riuniti insieme per formare tipi più complessi. Cioè, se più classi hanno in comune attributi, è possibile definire una "super-classe" che ha gli stessi attributi delle classi dalle quali è stata definita.
Il processo opposto alla generalizzazione è la **specializzazione**, e richiede di identificare i diversi componenti di un tipo complesso. Quindi nuove classi di oggetti possono essere derivate da una super-classe: le nuove classi **ereditano gli attributi** e i metodi della super-classe, oltre ad offrire altri attributi e operazioni; un'istanza della classe derivata è anche un'istanza della super-classe.
Strettamente connessa alla possibilità di ereditare attributi e metodi di una super-classe, c'è la possibilità, da parte delle classi derivate, di **ridefinire metodi** della loro super-classe (o delle loro super-classi). Questa proprietà (chiamata "polimorfismo") risulta utile per ottenere comportamenti specifici per una classe usando per la super-classe messaggi già disponibili.
Dato che le operazioni sono parte integrante di un oggetto e dal momento che le effettive implementazioni delle operazioni fanno parte di applicazioni software, gli oggetti possono essere utilizzati più facilmente tramite programmi applicativi. E' possibile definire una classe di oggetti

[1] DBMS = Data Base Management System (Sistema di Gestione della Base di Dati)

come descrizione astratta valida per una larga varietà di oggetti reali, e poi derivare nuove classi dalla classe "di base": qualunque applicazione in grado di riconoscere la descrizione astratta e di usare i metodi da essa forniti sarà comunque capace di manipolare oggetti delle classi derivate che ereditano tali metodi e, se necessario, ridefinirli.
Ciò significa che il codice software (programma) dell'applicazione non deve essere modificato: è questo il motivo per cui le basi di dati orientate ad oggetti possono essere utilizzate da qualsiasi applicazione più facilmente delle basi di dati relazionali, dove tutte le operazioni devono essere realizzate in programmi "esterni". Tuttavia, le basi di dati orientate ad oggetti non sono ancora così largamente usate nei prodotti commerciali, dove prevalgono le basi di dati relazionali.
In tempi recenti gli sforzi sono andati nella direzione di combinare i vantaggi delle basi di dati relazionali (largamente diffusi) e i benefici della modellizzazione ad oggetti in sistemi ibridi "ad oggetti – relazionali". E' possibile infatti, per le basi di dati descritte come modellizzate ad oggetti, essere costruite usando una combinazione di tecniche di descrizione orientate agli oggetti e di un modello concettuale relazionale per l'archiviazione dei dati. E' così possibile tradurre la struttura della gerarchia ad oggetti in tabelle di tipo relazionale: si tratta di un approccio di tipo "evolutivo", nel quale si aggiungono le caratteristiche del modello ad oggetti a una struttura esistente di tipo relazionale.

Sperimentazione di utilizzo del *geodatabase object oriented*
L'utilizzo del *personal geodadtabase object oriented* è stato sperimentato nell'ambito del progetto di potenziamento del SIT del Comune di Vimercate.
Le problematiche affrontate erano fondamentalmente legate alla conversione delle geometrie del P.R.G. comunale vigente in un formato "GIS *oriented*", allo scopo di poterle integrare con il patrimonio cartografico comunale esistente, disponibile in formato nativo ESRI, e di creare una base di dati adatta allo scopo di operare analisi territoriali anche complesse.
Il file *shape*, infatti, pur essendo considerato lo standard di fatto per la memorizzazione del dato geografico GIS, non è un formato topologico, cioè non consente di gestire l'integrità spaziale del dato e la validazione dell'attributo associato ad esso.
Invece l'archivio di informazioni costituito sotto forma di *geodatabase* rappresenta un unico insieme di dati strutturati, con le seguenti caratteristiche:

- in un unico oggetto si integrano dati cartografici (descrizione della geometria) e tematici (attributi);
- la ridondanza dei dati è ridotta al minimo;
- viene ottimizzata l'integrità dei dati;
- viene facilitata la gestione e l'interrogazione dei dati.

In particolare, la sperimentazione eseguita sui dati ha permesso di svolgere diverse verifiche:

- verifiche sulla possibilità per più insiemi di oggetti cartografici di ereditare le proprietà geografiche dal *dataset* di appartenenza (come la proiezione, il sistema di coordinate e il dominio geografico);
- verifiche sull'inserimento e gestione dei metadati (in formato XML) relativi a ciascuna componente del *database*;
- verifiche (sempre per quanto riguarda la documentazione della base di dati) sulla descrizione della qualità e del contenuto dei dati (descrizioni dei campi, definizioni dei codici e informazioni sui contatti).

Il processo di gestione del *dataset* di partenza, costituito dalle sorgenti cartografiche del PRG comunale realizzate su piattaforma Autodesk® e georeferenziati sulla base del volo aerofotogrammetrico del '99, ha visto la sua diretta importazione in ambiente ArcCatalog® nella versione ArcINFO. Tramite questa interfaccia software, la principale di ArcGIS per definire e gestire il *geodatabase*, è stato possibile utilizzare i nuovi strumenti (introdotti dalla versione 9.0)

dedicati al caricamento di tutte le *feature class* CAD individuali sotto forma di nuove tabelle e nuove *feature class*[2] *geodatabase*, specificandone per ognuna il tipo di geometria.
Nella procedura di caricamento delle sorgenti dati si è rivelato utile specificare *query* per limitare le *feature* da caricare nel *personal geodatabase* (implementato in ambiente Microsoft Access). Questo tipo di operazione ha agevolato le successive operazioni di *editing* topologico, necessarie per la correzione degli errori presenti nelle geometrie del PRG generate da sistemi CAD, che verranno in seguito analizzate.
Il primo evidente vantaggio nel caricamento dei dati CAD all'interno di un *geodb* è stato l'ottenimento di un "catalogo" strutturato di *feature class* – o meglio oggetti – caratterizzate da geometria uniforme, condividenti lo stesso riferimento spaziale del *feature dataset*[3] creato (relativo all'azzonamento di PRG), ovvero la stessa proiezione e lo stesso sistema di rappresentazione (Gauss-Boaga nel Datum Roma40).
Questo importante passaggio segna l'effettiva migrazione dai sistemi per la gestione di cartografia vettoriale verso una struttura di memorizzazione aperta dedicata alla gestione di dati spaziali autoconsistenti e memorizzati in tabelle all'interno di un DBMS relazionale.
Il secondo vantaggio emerso dalla sperimentazione del modello dati *geodatabase* consiste nella possibilità di definire regole comportamentali che aiutano a garantire l'integrità del dato come i domini, ovvero elenchi di valori validi dell'attributo per una colonna nell'intera tabella.
In particolare sono stati creati degli elenchi espliciti di codici accettabili per classificare le *feature class* sugli interventi urbanistici in corso (domini *coded value*).

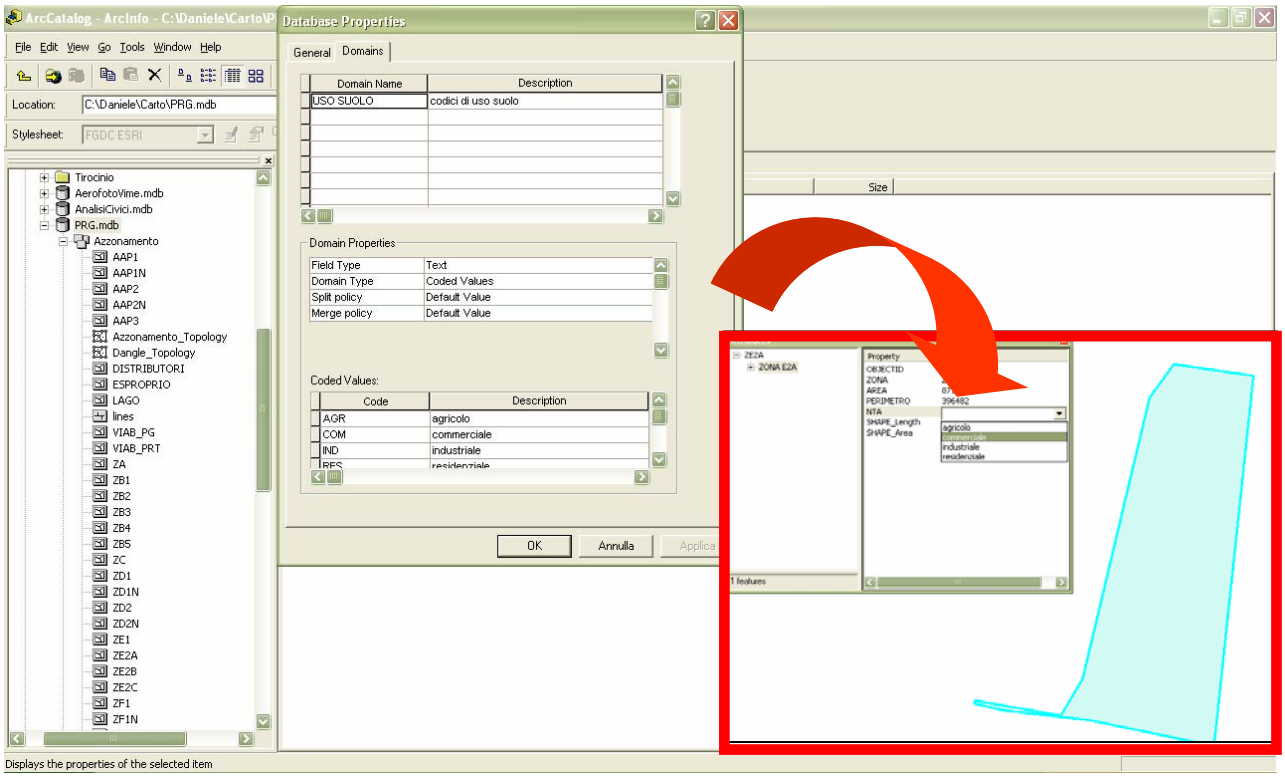

Figura 1 – Impostazione del dominio di tipo "coded value": i valori inseriti verranno visualizzati in un elenco a discesa in fase di editing cartografico.

[2] Le *feature class* all'interno del geodatabase sono un insieme di oggetti spaziali (come punti, multi punti, linee o poligoni che rappresentano un oggetto del mondo reale in un *layer* della carta) che hanno lo stesso tipo di geometria ed attributi
[3] Associazioni di *feature class* e riferimenti spaziali definiti, memorizzati in una tabella fisica come righe all'interno del RDBMS. Per questa ragione tabelle e *feature class* devono possedere un nome unico, che siano o meno in un *feature dataset*.

L’implementazione delle regole di dominio impostate hanno impedito in fase di *editing* l’assegnazione di attributi illegali.
All’interno del *feature dataset* del *geodatabase* sono state successivamente utilizzate le regole topologiche, per creare un nesso spaziale tra gli elementi delle *feature class* lineari corrispondenti alle geometrie dell’azzonamento del Piano Regolatore Generale comunale.
La definizione a priori di precise regole topologiche di intersezione ha permesso di operare un controllo di qualità sui dati vettoriali “in ingresso” nel *geodatabase*, individuando tutti quegli errori generati in fase di digitalizzazione sulle geometrie, in modo da poter generare le coperture poligonali utilizzate tipicamente nella rappresentazione delle zone urbanistiche di piano (definite omogenee in termini di indici e parametri edificatori).
Gli errori generati a valle del processo di validazione della topologia sono stati trattati in ambiente ArcMap® 9.0 che offre un set completo di strumenti di *editing* che permettono di verificare la conformità delle relazioni spaziali con i vincoli e le regole impostate.

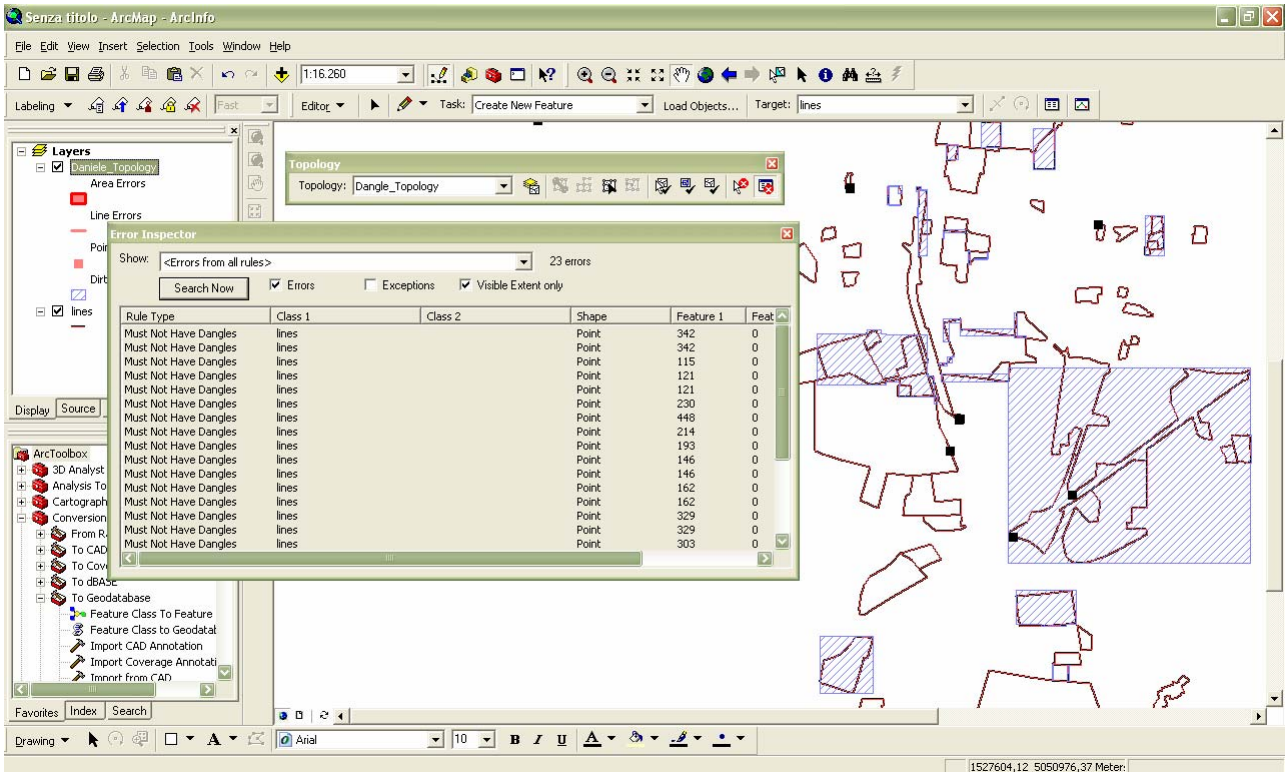

Figura 2 – Visualizzazione in ambiente ArcMap delle dirty areas generate dal processo di validazione topologico eseguito in modalità automatica: in questo caso specifico i poligoni evidenziano dove sono state apportate le correzioni degli errori geometrici riscontrati sulle feature

Impostando ulteriori regole topologiche tra diverse *feature class* di PRG è stato possibile mantenere la congruenza geometrica e logica delle *feature class* poligonali, anche in fase di modifica, attraverso gli strumenti per l’*editing* della geometria coincidente.
Il concetto di topologia delle *coverage* ArcINFO, dove le relazioni topologiche risiedevano all’interno delle stesse tabelle del *database* (poligono modellato come una serie di archi e *label point*) viene reinterpretato in modo meno restrittivo all’interno del *geodatabase* dove la coincidenza geometrica tra varie *feature* è gestita in ogni sessione di *editing*, grazie all’informazione sulla geometria contenuta in ogni oggetto (che può esistere come un’entità a sé stante).

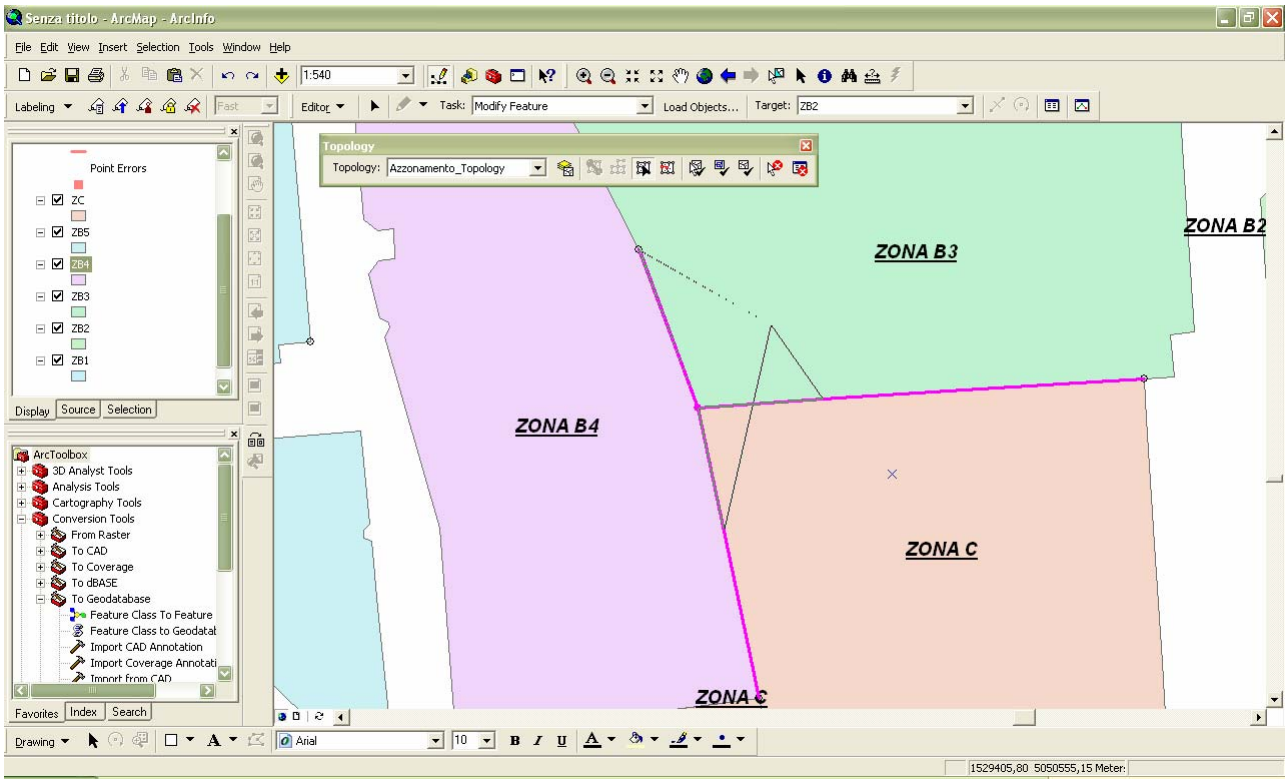

Figura 3 – Sessione di editing topologico in ambiente ArcMap: più feature class che contengono bordi coincidenti possono essere editate simultaneamente, gli attributi geometrici delle feature class coivolte dalla modifica cambiano in tempo reale.

Conclusioni

I vantaggi della sperimentazione del *geodatabase* orientato ad oggetti si traducono nella memorizzazione centralizzata per i dati spaziali e gli attributi; nella possibilità di definire regole comportamentali che aiutino a garantire l'integrità del dato come quelle implementate sul dominio di alcuni campi per prevenire l'assegnazione di attributi illegali, piuttosto che l'affidamento di regole per prevenire relazioni tematiche non possibili.

L'avvio del processo di validazione della topologia, definita con precise relazioni spaziali di adiacenza, coincidenza e connessione, ha dato modo di ottenere diverse *feature* di errore di tipo poligonale e/o lineare (come per esempio le numerose sovrapposizioni tra zone omogenee di PRG, errori geometrici ereditati dai disegni di origine in formato .dwg) accompagnate dal corrispondente sommario degli errori geometrici governabili in modo autonomo e differenziato.

Bibliografia

Bertino E., Catania B., Ferrari E., Guerrini G. (1997), *Sistemi di Basi di Dati. Concetti e architetture*, Città Studi Edizioni.

Burrough P., McDonnel A. (1998), *Principles of Geographical Information Systems*, Oxford University Press, Oxford

Environmental System Research Institute *ArcGIS 9 geodatabase workbook*, copyright 2004 ESRI

Environmental System Research Institute *Building a geodatabase*, copyright 1999-2002 ESRI