UN MODELLO UML PER L'INTEROPERABILITÀ DI GEODATABASE SPAZIO-TEMPORALI SU SISTEMI DISTRIBUITI MULTIPIATTAFORMA

Vincenzo BARRILE , Francesco COTRONEO

Università degli Studi Mediterranea di Reggio Calabria - DIMET,
barrile@ing.unirc.it, francesco.cotroneo@unirc.it

Riassunto

Come noto, la maggior parte dei sistemi GIS che vengono oggi progettati hanno la capacità di utilizzare per la gestione dei dati i Geodatabase, ovvero dei database spaziali la cui caratteristica principale è quella di modellare i dati geografici utilizzando il paradigma ad oggetti. Purtroppo tale tipo di modellizzazione, capace di mantenere la banca dati consistente ed efficiente, viene implementato nei vari software GeoDB solo nello strato di rappresentazione (e con funzionalità limitate), ovvero di interfaccia utente, mentre nello storage dei dati ci si appoggia a Server DBMS che utilizzano il paradigma relazionale.

Questa scelta non solo comporta un notevole degrado delle prestazioni del sistema, ma soprattutto rende difficile la rappresentazione all'interno di un GIS di quei contesti dotati di notevole dinamismo e che pertanto richiedono che la gestione dei dati (popolamento e querying) venga effettuata ad ogni livello attraverso interfacce operative dotate di un alto grado di astrazione. In particolar modo risulta arduo implementare e mantenere dei Geodatabase dediti allo storage di features variabili nel tempo (la variabilità non deve intaccare la consistenza tra componente spaziale ed alfanumerica).Tale tipo di implementazione deve purtroppo passare, nella maggior parte dei casi, attraverso la specializzazione del software GIS General Purpose utilizzato, con la stesura di codice opportuno. Ciò implica che i GIS così costruiti sono prodotti chiusi in quanto difficilmente portabili su piattaforme diverse da quelle sulle quali sono stati originariamente implementati.

A seguito di tali considerazioni si è deciso di definire e proporre un modello rappresentativo dei dati spazio-temporali attraverso i costrutti del linguaggio di semantica UML. Tale modello descritto in UML diviene così un insieme di regole standard, le quali possono essere interpretate da un plug-in software sviluppabile per ogni applicativo GIS.

Una tale accoppiata (UML e Plug-In) consente da una parte una gestione semplice ed allo stesso tempo efficiente dei Database spatio-temporali, dall'altra permette un'agevole operazione di comunicazione tra GIS implementati su sistemi distribuiti e multipiattaforma.

Abstract

As known, most part of existing GIS systems can use Geodatabases in order to manage data. They are spatial databases which are able to model geographical data using object programming. Unfortunately, this kind of modelling, able to maintain a consistent and efficient data set, is implemented in GeoDB software only in representation layer (with limited functionalities), that is in the user-interface point of view. On the other hand, it exploits a DBMS server, using the relational paradigm, for data storage. Therefore it is hard to implement and manage such Geodatabases used for storage of time-varying features (but variability does not involve consistency of spatial and alphanumeric components). Consequently, a model representative of space-time data has been implemented by using UML semantic. So, such UML-based model is a set of standard rules, which can be interpreted by a GIS-generalized plug-in software. Joint of UML and plug-in allows both a simple and time-efficient management of space-time databases, and as easy communication among such GIS systems implemented on distributed and multi-platform systems.

Introduzione
L'evoluzione tecnologica e metodologica nel campo dei GIS è stata fortemente condizionata dal contributo delle aziende private, diversamente da quanto successo per i sistemi informativi per il management dove l'introduzione di nuove funzionalità e, soprattutto, la definizione di nuovi paradigmi progettuali sono occorsi secondo le modalità indicate da standard internazionali alla cui specificazione hanno partecipato istituzioni accademiche, enti operanti nel settore, organizzazioni e privati. I diversi percorsi storici delle varie soluzioni implementative per i GIS hanno portato ad una situazione particolarmente complessa sul piano dell'interoperatibilità intesa sia come capacità di comunicazione tra sistemi distribuiti dediti alla soluzione di una medesima problematica, sia come possibilità di migrare gli applicativi (porting) su piattaforme hardware e software appartenenti a diversi produttori. Gli ostacoli che quotidianamente i manutentori degli applicativi GIS incontrano nell'affrontare i problemi legati all'interoperatibilità si possono classificare in tre grandi famiglie:

- Tecnico–rappresentativi
- Semantico
- Comportamentale

I primi sono di più « basso livello » e riguardono aspetti di compatibilità tra sistemi operativi, relative librerie (soprattutto per quanto concerne l'accesso ai dati: driver odbc, jet ecc...), formati per la rappresentazione dei dati, parser dei metadata, legendistica ecc.. Su questo aspetto moltissimo è stato fatto; per quanto riguarda poi lo storage fisico dei dati vi sono dei formati introdotti attraverso processi di standardizzazione spesso condotti a livello di organizzazioni internazionali con forti contributi del mondo accademico, ed anche standard de facto per i quali le aziende proprietarie hanno reso pubbliche le specifiche. L'adozione di formati standard incontra però la resistenza dei produttori di software ad adottarli per preservare le "proprie" quote di predominio; comunque occorre sottolineare come gli standard per la memorizzazione del dato siano una soluzione al problema di tipo sufficiente ma non necessaria. Partendo proprio da queste considerazioni si è definito un formato affatto adatto per la memorizzazione dei dati, ma efficace per le operazioni di interscambio, specie di strutture spazio temporali preventivamente definite; ovvero tale formato contiene tutte le informazioni utili alla rappresentazione ma in forma implicita, cosicché occorre ci sia da parte del software GIS che effettua l'import nel proprio formato proprietario, una elaborazione atta a strutturare le informazioni con riferimenti espliciti alla topologia, alla legendistica, al tipo di dati alfanumerici, metadati, documentazione e quant' altro.

UML ed interopertibilità tra GeoDatabase Spazio-Temporali
Tra le varie possibilità rappresentative della componente temporale all'interno dei GIS, quelle maggiormente diffuse utilizzano schemi entità-relazione affidandosi per le query all'algebra relazionale, dove la dimensione temporale è gestita da un attributo di Timestamp nelle tabelle di relazione. Altre metodologie, intendendo inserire una dimensione temporale "continua", integrano al modello relazionale metodologie statistiche per garantire la continuità spaziale e topologica degli elementi Geografici. La metodologia proposta in questa nota utilizza un approccio Object Oriented per la gestione di features variabili nel dominio del tempo (la variabilità è gestita in modalità discreta e non continua).
La scelta di un approccio ad oggetti pone però problemi nell'ambito dell'interopertibilità tra GIS, poiché si ha un'incompatibilità semantica nelle diverse implementazioni dei GeoDatabase (i database per la rappresentazione ad oggetti delle entità geografiche). Questi utilizzano il paradigma ad oggetti per la generazione di modelli rappresentativi della realtà, e purtroppo le proprietà di tale paradigma come la derivazione, l'overriding, il polimorfismo, le interfacce di accesso ai dati, vengono implementate parzialmente e con modalità differenti da produttore a produttore ed inoltre esse non vengono utilizzate per modellare sempre le medesime categorie rappresentative. Il risultato è l'impossibilità di costruire un software che automatizzi la conversione tra Geodatabase; si tratta infatti di incompatibilità semantica. Ancora più complessa è la situazione a riguardo dell'interoperatibilità comportamentale cioe' alla possibilità di migrare i processi operativi e

automatizzati definiti all'interno di un GIS per uno scopo specifico; ciò riguarda, ad esempio, le macro costruite a partire da query concatenate, come anche i moduli special purpose che aggiungono funzionalità non derivabili dai Software GIS. Che tali moduli non siano interoperabili tra GIS può essere anche accettato poiché ciò riguarda una problematica più generica, ossia il porting tra software generalistici. E' necessario invece trovare una modalità di interscambio per la definizione di operazioni eseguite tramite successioni di funzioni comuni a tutti i softwae GIS (query su dati alfanumerici, query spaziali, geoprocessing overlay, buffering, ecc…)
Proprio per questi problemi la definizione dello schema concettuale qui proposto è stato definito utilizzando l'UML, e di conseguenza il Plug-In implementato e integrato nei vari software GIS ne processa le particolari istanze e struttura le classi di oggetti utilizzando le implementazioni del paradigma ad oggetti del sistema ospite, ma al contempo sopperisce alle mancanze funzionali di questo, con codice proprio.
La trattazione dello schema viene esposta di qui in avanti commentando un caso tipico in ambito gestionale GIS, come la rappresentazione di edifici per i quali, nel tempo, si vuole sia possibile la tracciabilità delle variazioni architettoniche (abusivismo o aggiornamento parametri di tassazione, etc.) o degli attributi alfanumerici (numero proprietari che si sono susseguiti nel tempo, restauro per l'allineamento al piano colore, etc..). Grazie a tali tipi di informazioni si possono effettuare ad esempio facilmente studi di carattere socio-economico.
Lo schema UML del modello proposto dagli autori in relazione alla problematica su evidenziata, è riportato in fig. 1.

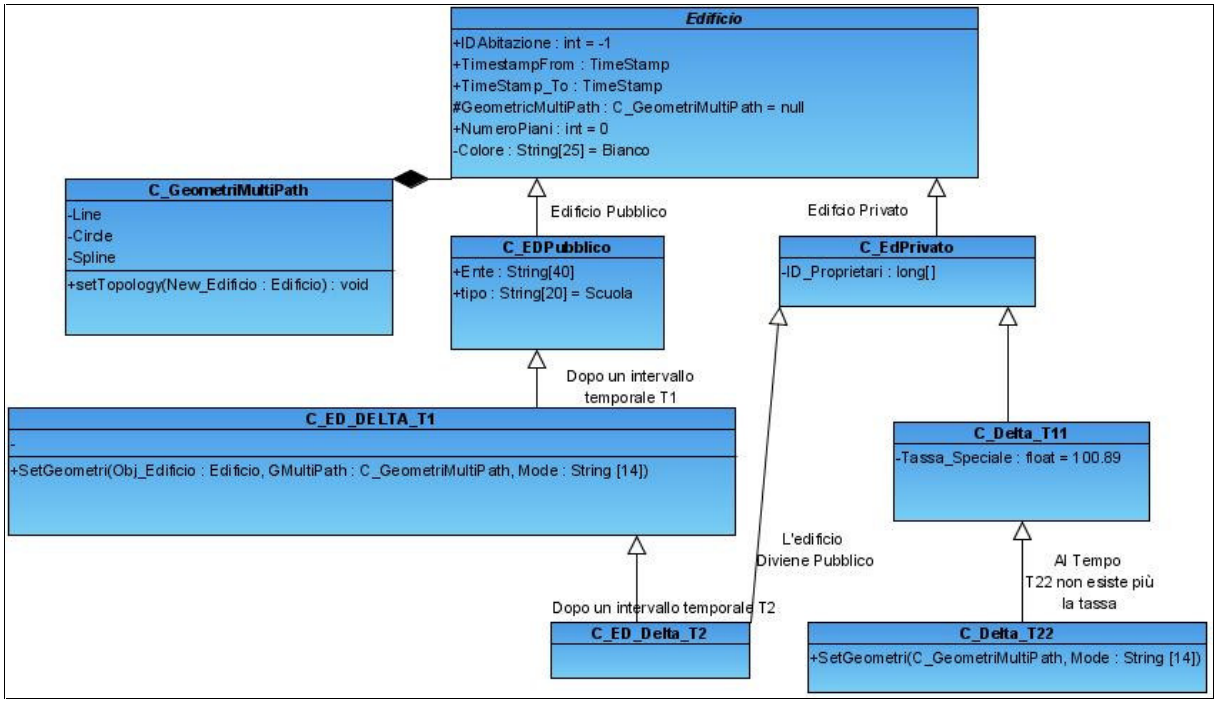

Figura 1. Schema UML Spazio-Temporale per la rappresentazione di edifici.

Senza entrare nel dettaglio, commentando il perché di tutti gli attributi e la loro specifica dichiarazione, si nota che lo schema in figura 1 presenta una Classe madre per gli edifici "generici" (si tratta di una classe astratta), comprendente due attributi necessari per l'estensione temporale: TimestampFrom e TimestampTo; grazie ad essi viene memorizzato l'intervallo temporale entro il quale un determinato edificio (forma, attributi,…) sia da considerarsi "valido" (in altre parole l'oggetto esiste solo in quel dato intervallo temporale). Questi attributi sono ereditati da tutte le classi che specializzano la classe madre. Di queste le prime che sono situate nell'albero di derivazione specializzano gli edifici pubblici e quelli di tipo privato. Se a questo punto viene modificata la struttura o il contenuto informativo anche di un solo edificio pubblico (per esempio al

tempo T_1, viene creata "a run-time" una classe di nome C_ED_delta_T1 utilizzata per istanziare l'occorrenza dell'edificio modificato. Viene settato successivamente l'attributo TimestampFrom al tempo T_1 e al puntatore GeometriMultiPah viene fatta puntare la nuova Shape della feature in esame (ciò può essere fatto direttamente, o chiamando la funzione setGeometri() è possibile ottenere la nuova geometria come modifica della geometria posseduta precedentemente). Occorre sottolineare che la dimensione temporale con tale approccio non è rappresentata per via di indici o join spaziali e/o alfanumeriche. Qui l'oggetto modificato da una parte conserva gli elementi di tracciabilità storica (è possibile risalire a come si presentava l'edificio in tutti gli stati precedenti), dall'altra è una entità a sé stante, e può godere di tutti i privilegi della modellazione ad oggetti: ad esempio i vincoli topologici e comportamentali possono essere diverse da quelli del medesimo oggetto negli stati passati, così le relazioni spaziali con altri elementi, ecc..., ecc… E cosa molto importante, tutti gli edifici (di ogni epoca e tipologia) volendo posso essere inseriti nello stesso layer, potendo utilizzare gli strumenti standard del GIS per effettuare così query Spazio-Temporali con limitate risorse computazioni. Continuando a scendere nell'albero di derivazione si osserva che la classe C_ED_Delta_T2 (generata per istanziare gli edifici già modificati al tempo T_1) si caratterizza per avere una doppia derivazione: questi edifici derivano le proprietà della classe C_Ed_Privato, ciò sta ad indicare che l'ente pubblico li ha ceduti ai privati. Infine ritornando al discorso della univocità dell'edificio in funzione dell'intervallo temporale in cui esso è validato, si sottolinea che gli edifici istanziati con la classe C_Delta_T22 non derivano l'attributo "tassa speciale", semplicemente perché la tassa pagata al tempo T_{11} era, ad esempio, localizzata in quel dato periodo (ciò non si evince dalla figura, ma l'attributo è marcato effettivamente come "private" -); in altre parole l'edificio possiede quell'attributo solo in quello specifico intervallo temporale.
Da quanto detto, sotto il profilo dell'interopertibilità occorre che i Plug-In preposti all'import di dati istanziati interpretino correttamente i dati secondo questo schema di riferimento, e adattino gli stessi alla particolare implementazione del sistema locale (cosa duale per l'export); è ovvio che essi si devono far poi carico, in maniera trasparente, di presentare all'utente del GeoDatabase una interfaccia operativa che funzioni secondo lo schema di figura 1, e allo stesso tempo di strutturare i dati avvalendosi delle risorse del Software GIS ospite: in tal senso si è davanti ad una incompatibilità sia semantica che comportamentale.

Implementazione in ArcGIS 8.3

ArcGIS 8.3 mette a disposizione, all'interno dei suoi GeoDatabase, funzionalità interessanti per la gestione e la rappresentazione dei dati geografici pensati come oggetti. Tuttavia soffre in tal senso di diverse limitazioni in relazione all'implementazione dello schema concettuale di figura 1: supporta la derivazione su un solo livello, non possono esistere più genitori per una classe, la derivazione è affidata ad un attributo di identificazione di classe, che l'utente deve impostare in maniera non trasparente. Un esempio di implementazione delle classi di figura 1 all'interno del GeoDatabase di ArcGIS (da parte del Plug – In integrato) è riportato in figura 2.

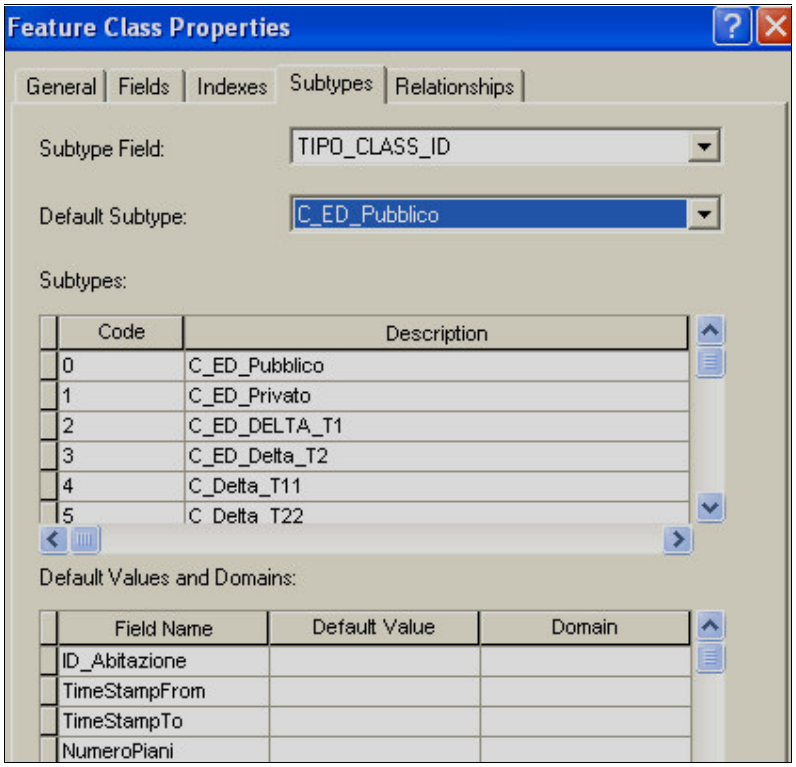

Figura 2. Classi importati in ArcGIS (Interfaccia ArcCatalog)

Come anticipato, nella figura 2, è ben evidente come tutte le classi devono essere dichiarate come figlie dirette della classe Edifici. Svantaggio più o meno derivante da questo fatto, è che non è possibile definire attributi specifici per le classi derivate.
Tale limite viene superato dal Plug-In proposto creando, in primis, delle Features Class (classi di ArcGis) di appoggio: per la classe C_EDPubblico (edifici pubblici) viene creata la F. Class C_EDPubblico_Inheritance, per la F. Class C_EdPrivato, la C_EdPrivato_Inheritance, e così via, (cfr. figura 3). Tali classi contengono gli attributi che li specializzano e che quindi non sono presenti nelle classi madre, e contengono inoltre l'identificativo ereditato (nello specifico ID_Abitazione).
A questo punto vengono create delle relazioni con mlteplicità uno a uno sul codice dell'edificio (presente in ogni classe) tra la Feature Class di appoggio e la Feature Class che è stata creata per simulare il meccanismo di ereditarietà a più livelli. Ad esempio, come da figura 3, 4 e 5. vi sono due relazioni che legano le F.Class degli edifici pubblici e privati alla F.Class Edifico, e conseguentemente la F. Class C_Delta_T11__Inheritance è in relazione uno a uno con la F. Class C_EdPrivato_Inheritance.

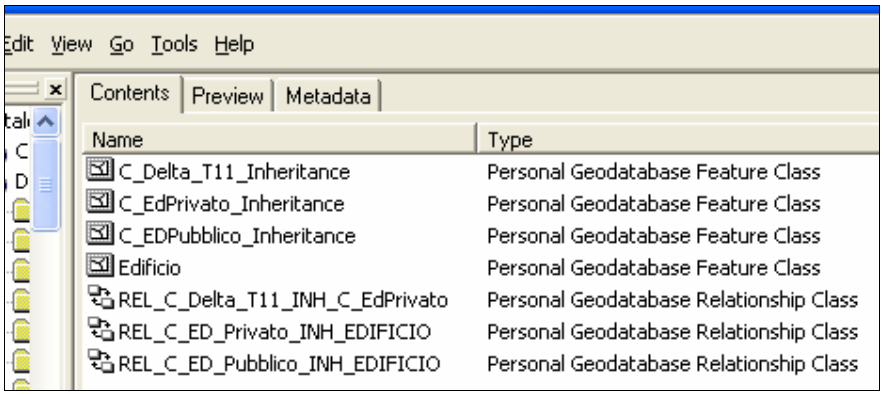

Figura 3. Feature Class e Relazioni Spaziali creati dal Plug – In in ArcGIS

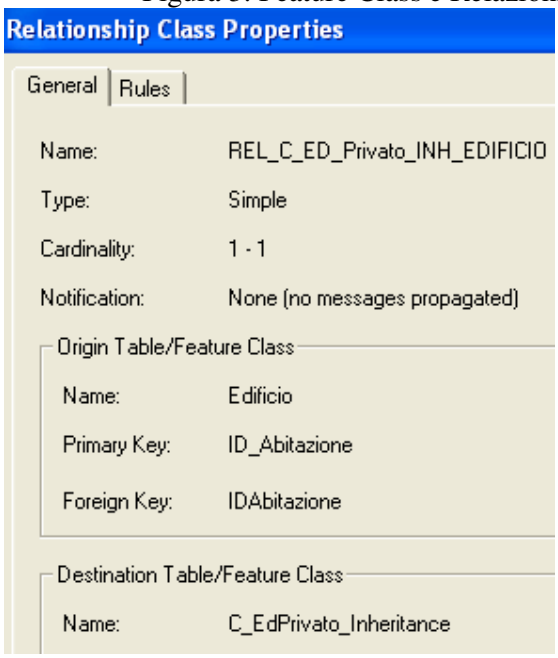

Figura 4. Relazione per Feature Class derivate da Edificio

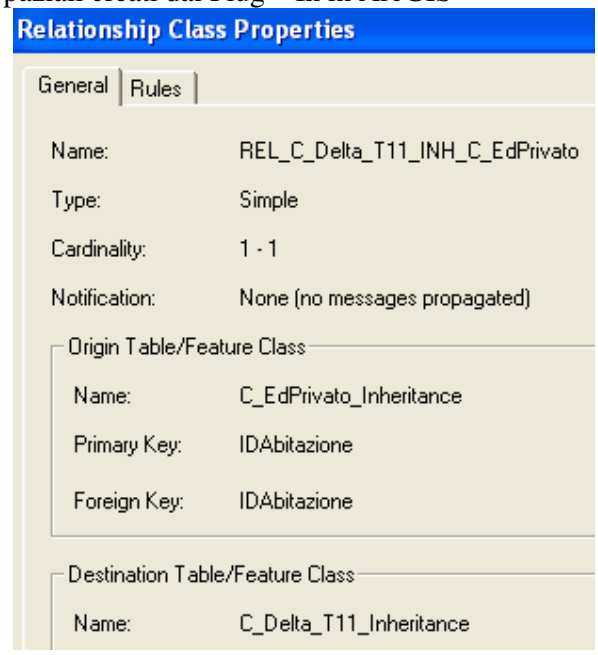

Figura 5. Relazioni per Feature Class derivate da Edifico Privato

Conclusioni

Oggi sono sempre più le applicazioni in campo GIS che necessitano di integrare la componente temporale nella gestione dei dati geografici; tra le varie soluzioni è possibile creare modelli che si basano sul paradgma ad oggetti, come quello indicato, ad esempio, in questa nota.
Purtroppo l'implementazione di tale paradigma differisce notevolmente tra i vari produttori di Software GIS causando ovvi problemi di interoperabilita'.
La soluzione di prevedere un Plug-In che interpreti i dati e le relative strutture rappresentative espresse in formato UML secondo uno schema non ambiguo, quale quello presentato, sembra risultare una valida soluzione in molti contesti applicativi dove tale schema è totalmente adattabile.

Bibliografia

Douglass B. P. (2004), "Real Time Uml", Addison-Wesley Professional, pp. 29-41.

Najm E., Nestmann U., Stevens P. (2004), "Formal Methods for Open Object-Based Distributed Systems", Springer, pp. 147-163.

Arctur D., Zeiler M.(2003), "Designing Geodatabases", ESRI Inc., pp. 21-49.

Dov D. (2002), "Object-Process Methodology", Springer, pp. 87-109.

Christakos G., Bogaert B., Serre M. (2002) "Temporal Gis", Springer, pp. 103-114.

Wegener M., Fotheringham A. S. (2000), "Spatial Models and Gis", Taylor & Francis, pp. 174-188.

Montgomery L.D. (1995), "Temporal Geographic Information Systems Technology and Requirements: Where We Are Today", The Ohio State University, Columbus, Ohio.