



Università degli Studi di Pisa  
Facoltà di Scienze Matematiche, Fisiche e Naturali  
Dipartimento di Informatica

**Verifica di congruità di dati geografici  
tramite un sistema basato su regole**

**Orazio Martorina**

Tesi di Laurea in  
Tecnologie Informatiche  
Anno Accademico 2004/2005

*Relatore:* Prof. Paolo Mogorovich

---

<b>INTRODUZIONE</b>	1
<b>CAPITOLO 1 – CONTESTO</b>	
<b>Introduzione</b>	3
<b>1.1 Costruzione di una carta numerica</b>	4
1.1.1 Inserimento manuale di una mappa esistente	5
1.1.2 Inserimento automatizzato di una mappa esistente	5
1.1.3 Processo aerofotogrammetrico	6
<b>1.2 Collaudo ed accettazione mappa</b>	7
1.2.1 Test a campione	7
<b>1.3 Analisi dei dati di una mappa: la componente descrittiva</b>	8
1.3.1 Rientro attributi numerici in un dato range di valori	8
1.3.1.1 Attributi numerici: collaudo	8
1.3.1.2 Attributi numerici: applicazione	9
1.3.2 Ricerca e Analisi per attributi di testo	9
1.3.2.1 Attributi di tipo testo: collaudo	9
1.3.2.2 Attributi di tipo testo: applicazione	10
1.3.3 Ricerca e Analisi per attributi di tipo speciale	10
1.3.3.1 Attributi di tipo data: collaudo	10
1.3.3.2 Attributi di tipo data: applicazione	11
1.3.4 Funzioni Statistiche	11
<b>1.4 Analisi dei dati di una mappa: la componente geografica</b>	12
1.4.1 Analisi componente geografica: collaudo	12
1.4.2 Analisi componente geografica: applicazione	13
<b>1.5 Operazioni di congruità</b>	13
1.5.1 Copertura di un Layer tramite punti con distanza minima assegnata	14
1.5.2 Congruità tra la misura di un punto e le curve che lo contengono	14

---

**CAPITOLO 2 – APPROCCI AL PROBLEMA**

<b>Introduzione</b>	16
<b>2.1 Specifica Utente</b>	16
2.1.1 Collaudo	17
2.1.2 Applicazioni	17
<b>2.2 Il Geodatabase ESRI</b>	18
2.2.1 La struttura	20
2.2.2 Il linguaggio	22
2.2.3 I Limiti	23
2.2.3.1 Collaudo	24
2.2.3.2 Applicazione	24
<b>2.3 Altri Approcci</b>	25
<b>2.4 Strategia di una possibile soluzione</b>	26
2.4.1 Il linguaggio	27
2.4.2 La struttura	28

**CAPITOLO 3 – PROGETTAZIONE E IMPLEMENTAZIONE**

<b>Introduzione</b>	30
<b>3.1 Architettura e Tecnologia di Base</b>	30
<b>3.2 Analisi regole</b>	31
<b>3.3 Parser</b>	34
3.3.1 Schema Automa	34
3.3.1.1 Raffinamento condizioni sugli attributi	37
3.3.2 Implementazione Automa	40
3.3.2.1 Analizza regola	41
<b>3.4 Engine</b>	42
3.4.1 Creazione nuovo shapeFile	43
3.4.2 Restrizioni sulla geometria	43
3.4.3 Condizioni spaziali	44
<b>3.5 Portabilità ed espandibilità</b>	45

<b>CAPITOLO 4 – ESEMPI</b>	
<b>Introduzione</b>	47
<b>4.1 Regole con restrizioni descrittive e spaziali</b>	47
4.1.1 Condizione descrittiva su layer con restrizione descrittiva	49
4.1.2 Condizione spaziale su layer con restrizione sulla geometria	51
4.1.3 Condizione spaziale doppia su layer con restrizione descrittiva e spaziale	53
<b>CAPITOLO 5 – CONCLUSIONI</b>	55
<b>APPENDICE A – CODICE PARSER MANAGER</b>	57
<b>APPENDICE B - CODICE ENGINE MANAGER</b>	63
<b>GLOSSARIO</b>	68
<b>BIBLIOGRAFIA</b>	70

## Introduzione

L'utilizzo di mappe geografiche in formato digitale è da decenni legato a molteplici settori applicativi ed organizzativi. Le tecniche per la digitalizzazione e modellazione di una mappa in formato digitale hanno avuto nell'ultimo decennio un notevole sviluppo tecnologico. La tecnologia utilizzata in fase di modellazione è identificata con il termine *GIS* (Geographic Information System). Questa tecnica ha la caratteristica di associare a ogni singolo oggetto presente in una mappa geografica tutte le informazioni intrinseche dell'oggetto stesso, siano esse qualitative, vedi il nome, siano esse geografiche, forma geometrica e coordinate della sua ubicazione nello spazio.

In ambiente GIS i dati vengono catalogati in *layers* (strati) tematici che possono essere tra loro relazionati tramite collegamento e sovrapposizione geografica. Ogni layer conterrà le informazioni riguardanti una precisa entità reale, per esempio strade, edifici, falde acquifere; ogni entità potrà essere specializzata aggiungendo degli attributi o suddividendola in layers, per esempio nel caso delle falde acquifere per separare i fiumi dai laghi o le acque potabili dalle acque paludose.

Dato l'elevato potenziale di tale tecnologia, si è reso necessario individuare degli standard che indichino in modo non ambiguo le direttive da seguire nella modellazione di una mappa, al fine di garantire la massima interoperabilità dei dati e degli applicativi. L'Intesa tra Stato, Regioni ed Enti Locali sui Sistemi Informativi Geografici, approvata il 26 settembre 1996, rappresenta il tentativo più organico di modificare in termini positivi la situazione dell'Informazione Geografica in Italia per creare uno stimolo verso una partecipazione più ampia delle istituzioni, delle imprese e del mondo scientifico [1]. In questi anni l'Intesa ha redatto diversi documenti contenenti le specifiche da seguire nella creazione di una mappa *georeferenziata*.

Tra gli enti pubblici sono avviati svariati progetti che riguardano lo sviluppo di applicativi GIS per la pianificazione del territorio, la localizzazione, l'allocazione di Risorse, il Map on Demand.

Scopo della tesi è, partendo dalla costruzione di una mappa georeferenziata, individuare le necessità e le operazioni dell'utente in fase di collaudo e di applicazione, analizzare le

soluzioni esistenti e fornire una strategia di soluzione che possa sopperire i limiti delle soluzioni già presenti nel mercato e soddisfare le esigenze dell'utente sia in fase di collaudo che applicazione.

# Capitolo 1

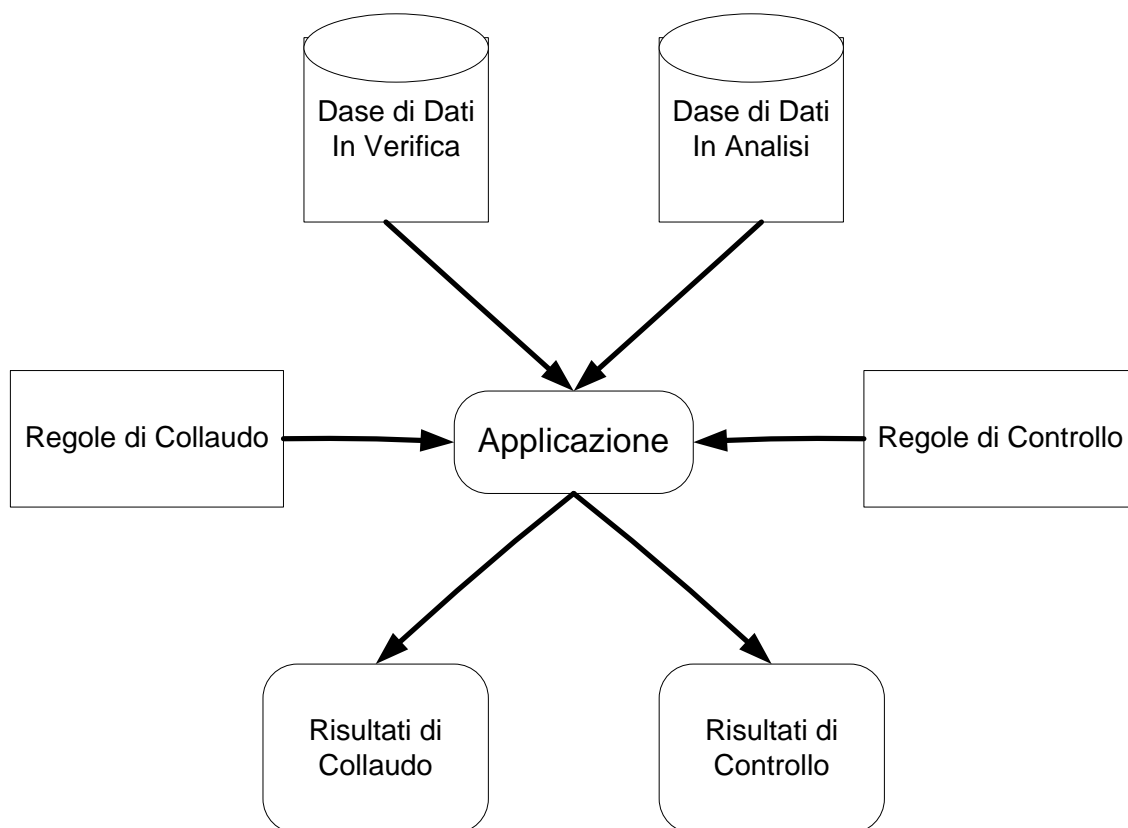
## Contesto

### Introduzione

Notevoli sono i vantaggi nell'avere la mappa digitalizzata di un territorio, per gli enti pubblici e non solo per essi, ma altrettanto e forse anche più dannoso può essere se questi dati non sono corretti. Notevoli sono gli sforzi fatti per fornire norme per la realizzazione di una mappa digitale: ai svariati documenti redatti dall'Intesa GIS si affiancano diversi standard quali ISO/TC211 [2] , CEN/TC287 [3] , INSPIRE Drafting Teams [4] , OGC [5] ed altri, ma nonostante ciò gli errori spesso non vengono individuati a causa di procedure di validazione non sistematiche e onerose che richiedono la presenza e interazione costante dell'utente.

Per questo motivo si è deciso di realizzare uno strumento in grado di applicare una serie di regole definite a priori dal generico utente senza bisogno di una interazione diretta.

Sebbene l'idea sia sorta per verificare la correttezza di una generica mappa, si è cercato di mantenere la struttura delle regole il più generica possibile permettendone l'applicazione in contesti che esulano dal collaudo e rientrano nell'utilizzo quotidiano del dato, permettendo uno snellimento del lavoro ed una maggiore accuratezza.



## 1.1 Costruzione di una carta numerica

Il problema di riportare in formato elettronico/digitale dati complessi e vasti quali quelli cartografici è un problema sempre in studio e sviluppo, e richiede una elevata accuratezza sia della strumentazione hardware e software sia dell'utente che procederà all'operazione. Vi sono diverse tecniche per procedere alla digitalizzazione, alcune delle quali ormai abbandonate a favore di nuove tecniche più precise.

Tra le varie tecniche disponibili per la digitalizzazione di una mappa ricordiamo la prima sviluppata, che si basava sulla digitalizzazione manuale di una mappa già esistente, e la più diffusa oggi, ovvero l'analisi a partire da foto *aerogrammetriche*.

Qualsiasi sia la tecnica utilizzata il processo è esposto a sviste da parte dell'operatore, errori di inquadramento, errori di attribuzione, coordinate o dimensioni poco precise.



Tutto ciò rende difficoltoso poter lavorare su tali dati e porta a volte a risultati fuorvianti e discordanti, con spreco di tempo e risorse.

Questi errori sono più evidenti quando si mettono a confronto mappe dello stesso territorio realizzate da strutture diverse, oppure quando si esaminano i diversi layer di una mappa di grosse dimensioni.

### **1.1.1 Inserimento manuale di una mappa esistente**

Una delle prime tecniche per la digitalizzazione delle mappe consisteva nel disporre una mappa esistente su un piano, fissandola bene ai lati in modo da impedirne qualsiasi spostamento, e per mezzo di una speciale penna si trasferivano tutte le geometrie ad un software specifico semplicemente ripassando i tratti presenti in essa. Questa tecnica, sebbene apparentemente meno costosa rispetto al processo aerofotogrammetrico che richiede un cospicuo investimento iniziale, introduce un rilevante numero di errori dovuto alla possibilità di un errore umano causato dall'affaticamento o distrazione/svista nel seguire manualmente la parte grafica; in più si basa su mappe precedentemente fatte e quindi anche esse con potenziali errori. La durata del processo di digitalizzazione poteva protrarsi per giorni con la possibilità di una deformazione della mappa a causa dell'umidità e quindi con aggiunta di errore.

Con la richiesta di mappe sempre più precise, la raffinatezza dei software e l'avanzamento della tecnologia, questa tecnica è andata persa a favore prima dell'analisi automatizzata di mappe precedentemente scannerizzate e poi dell'analisi di foto aerogrammetriche il cui cospicuo investimento iniziale è abbondantemente ammortizzato e recuperato nel tempo.

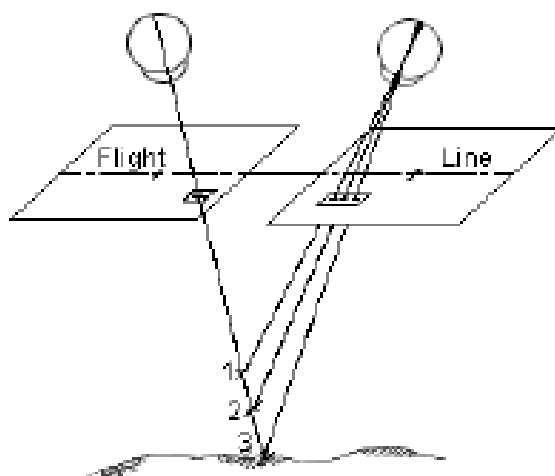
### **1.1.2 Inserimento automatizzato di una mappa esistente**

Una prima evoluzione della tecnica precedentemente vista è stata la scannerizzazione della mappa esistente per poi procedere alla sua costruzione attraverso un applicativo software. L'applicativo permette di selezionare una componente grafica per poi seguirla

con un inseguitore automatico. Questa tecnica ha permesso di ovviare agli errori introdotti dalla eventuale deformazione della mappa in fase di acquisizione ma è sempre legato ad una mappa già esistente e quindi potenzialmente “obsoleta” e/o con errori.

### 1.1.3 Processo aerofotogrammetrico

La tecnica oggi più diffusa per la costruzione di una carta di base è quella di partire da foto scattate in momenti precisi durante un piano di volo pianificato al fine di avere una visione della zona completa e omogenea dal punto di vista dell'altitudine. Queste foto successivamente vengono unite per sovrapposizione al fine di ottenere dalle zone sovrapposte un modello stereoscopico grazie al quale attraverso la triangolazione di due riprese è possibile ottenere le coordinate nei tre assi degli oggetti presenti.



Dalle immagini stereoscopiche attraverso uno *stereorestitutore* l'utente traccia le entità presenti nell'immagine fornendone forma e coordinate. Questo processo richiede un'immagine di alta qualità e una strumentazione adeguata. Il sistema DMC<sup>®</sup> (Digital Mapping Camera) della Intergraph<sup>®</sup> [8] è basato su un'accurata fotocamera digitale per riprese aeree e su un tool applicativo specifico per le attività di postprocessing, ma nonostante l'elevata risoluzione dell'immagine ed il software scritto ad hoc, l'attività di analisi richiede sempre un vigile e cospicuo intervento umano per il controllo dell'esatta

estrazione delle geometrie e loro classificazione. Ciò espone il processo a potenziali sviste ed errori, causando una non corretta digitalizzazione della mappa.

## **1.2 Collaudo ed accettazione mappa**

Come si è visto, qualsiasi sia la tecnica adottata per la costruzione di una mappa digitale, essa è soggetta ad errori che possono essere di rappresentazione, nel caso di misure o forme poco precise, di classificazione errata delle entità o di valore errato degli attributi. Ciò rende necessario una fase di controllo della mappa non solo come processo post-produttivo, quindi legato alla stessa azienda che ha effettuato la digitalizzazione ma anche da parte dell'acquirente in fase di collaudo. La società o ente acquirente pone dei vincoli di precisione della mappa, vincoli che andrà a verificare al momento della consegna prima dell'accettazione della mappa stessa. Vi possono essere diverse metodologie per verificare la bontà di una mappa, ma la maggior parte si basano su una verifica a campione affidata ad un esperto all'interno della stessa azienda acquirente o ad uno esterno, quale un docente universitario.

### **1.2.1 Test a campione**

La fase di test avviene individuando in modo casuale delle zone nella mappa per poi procedere a controllare la *topologia* degli oggetti e le informazioni degli attributi. In alcuni casi vengono effettuate delle semplici *query* relative all'intera mappa, per esempio sul valore massimo o minimo di un attributo o sul numero di entità specifiche. Questo tipo di test risulta attualmente il più diffuso e, per quanto si possa cercare di raffinarlo, oltre ai possibili errori o sviste umane ha una tipologia di errore intrinseca che è data proprio dalla tecnica a campione. In questo modo può accadere che non si vadano a testare zone in cui possono esservi delle discordanze e quindi si avrà come risultato l'accettazione di un dato non corretto.

## **1.3 Analisi dei dati di una mappa: la componente descrittiva**

Una mappa digitale è un dato strutturato; infatti oltre a contenere le informazioni geografiche, quindi forma e disposizione nello spazio dell'oggetto rappresentato, contiene anche informazioni di tipo descrittivo, quali la tipologia, il nome, la disposizione in funzione di via e numero civico nel caso di edifici, le dimensioni non solo aerali ma volumetriche e quindi l'altezza. Queste informazioni sono altrettanto utili e spesso più di quelle geografiche perché oggetto di provvedimenti e iniziative. Vediamo alcune analisi che possono essere fatte su una mappa per un normale utilizzo o quali controlli potrebbero essere fatti. Si ricorda che ci sono due ambienti diversi ad operare su una mappa, quello di collaudo e quello di analisi.

### **1.3.1 Rientro attributi numerici in un dato range di valori**

Una delle operazioni più comuni è quella di verificare che il valore numerico dell'attributo di una *feature* rientri in uno spettro ammissibile. Questi controlli si possono rendere necessari non solo in fase di collaudo ma anche nell'utilizzo quotidiano di una mappa.

#### **1.3.1.1 Attributi numerici: collaudo**

In fase di collaudo è possibile verificare che gli attributi rientrino in un range di valori ammissibili: si può controllare per esempio che non vi siano edifici con altezza minore o maggiore di un valore noto o con valore negativo. Nel caso di attributi numerici riguardanti una categoria è possibile specificare che essi siano valori interi o che appartengano ad un elenco di valori prestabiliti. Questi controlli possono essere fatti sull'intera mappa o iterati suddividendo la mappa in regioni al fine di effettuare dei controlli più mirati. Si pensi ad una mappa in cui è rappresentata una zona più sviluppata e industrializzata ed una rurale: è possibile effettuare dei controlli sugli

attributi indicando un valore diverso per le due zone al fine di assicurarsi una maggiore attendibilità della fase di collaudo e quindi della mappa.

### 1.3.1.2 **Attributi numerici: applicazione**

Le possibili applicazioni su una mappa sono innumerevoli; a titolo di esempio possiamo considerare la necessità di individuare le strutture che superano una soglia minima di altezza. Questo controllo può rendersi necessario nella pianificazione di un volo o perché, a causa di un sisma o della scoperta di sottosuolo argilloso, gli edifici più alti sono quelli che potrebbero risentirne maggiormente. Più semplicemente si potrebbe anche rendere necessario un tale controllo in visione di un quadro urbanistico che tenga conto anche dell'estetica della città vista dall'alto.

### 1.3.2 **Ricerca e Analisi per attributi di tipo testo**

Così come per gli attributi di tipo numerico, può rendersi necessario effettuare delle ricerche e operazioni su campi di tipo testo, ed anche in questo caso possono essere effettuate sia per operazioni di collaudo che applicative.

#### 1.3.2.1 **Attributi di tipo testo: collaudo**

Il controllo di correttezza di un attributo di tipo testo può essere più delicato rispetto ad uno di tipo numerico poiché gli errori che si possono commettere nel digitare una parola o una breve frase sono decisamente maggiori e più difficili da individuare. In fase di collaudo si potrebbe applicare un analizzatore di testo al fine di verificare la correttezza del contenuto dell'attributo ma in generale può essere sufficiente, così come per gli attributi di tipo numerico, verificare l'appartenenza del valore ad un dominio. Un controllo altrettanto importante è dato dalla ricerca di caratteri speciali. Un tale controllo si potrebbe rendere necessario nel momento in cui i dati non vengono inseriti manualmente ma importati da una tabella o da un generico file, poiché è possibile che

cambi il set di caratteri con la conseguenza che alcuni caratteri, quali i simboli o le lettere accentate, non vengano riconosciuti e vengano quindi sostituiti da un punto interrogativo o da un qualche simbolo.

### 1.3.2.2 **Attributi di tipo testo: applicazione**

Così come in fase di collaudo, anche a livello applicativo potrebbe rendersi necessaria la ricerca dei caratteri speciali e accentati all'interno di feature destinate alla visualizzazione su web o su supporti che potrebbero non supportare tale codifica. Un'altra operazione su attributi di tipo testo potrebbe essere l'individuazione di feature particolari classificate in funzione del valore di un attributo. Un esempio potrebbe essere dato da un layer rappresentante le falde acquifere contenente un attributo che indica se potabile o meno e se pubblica o privata.

Un'altra applicazione su attributi di tipo testo può essere data dalla necessità di individuare in un layer delle attività produttive tutte quelle riguardanti una categoria, oppure, in un layer dei lotti di un centro urbano, tutti quelli non edificati al fine di realizzare nuove aree verdi.

### 1.3.3 **Ricerca e Analisi per attributi di tipo speciale**

Tutte le considerazioni e i controlli discussi sia in fase di collaudo che a livello applicativo per gli attributi numerici e testuali sono riportabili per qualsiasi altro tipo di attributo. E' possibile ad esempio per attributi di tipo Data effettuare dei controlli sulla sua correttezza o appartenenza ad un intervallo ammissibile.

#### 1.3.3.1 **Attributi di tipo data: collaudo**

Così come si è visto per gli attributi di tipo testuale, per gli attributi di tipo data la possibilità di commettere degli errori è molto alta. Nel caso i valori non siano inseriti manualmente ma importati da una tabella o file è necessario prestare molta attenzione

poiché tali valori potrebbero essere importati in modo errato a causa delle diverse notazioni utilizzate, ovvero se indicare per prima il mese o il giorno. Tali errori possono non essere immediati da individuare poiché per date in cui il giorno sia un valore maggiore di dodici potrebbe non verificarsi alcuna anomalia per poi invece presentarsi nel momento in cui il giorno sia un valore minore o uguale a dodici e quindi vedersi scambiati il valore del mese con quello del giorno.

Altri controlli possibili sono l'appartenenza del valore ad un range di dati possibili quali ad esempio minore della data odierna e maggiore del primo gennaio millenovecentosettanta poiché la maggior parte dei sistemi in automatico setta tale giorno quando non riesce ad interpretare un valore possibile per esempio perché anziché di uno zero è stata digitata una O.

### 1.3.3.2 **Attributi di tipo data: applicazione**

Il controllo del giusto riconoscimento della data può rendersi necessario non solo in fase di collaudo ma anche in fase applicativo nel momento in cui si cambi piattaforma o applicativo per l'accesso ai dati.

A livello applicativo può essere molto utile effettuare delle ricerche dei valori appartenenti ad un dato intervallo. Si pensi alla necessità, per una valorizzazione del patrimonio storico, di una ricerca di quelle costruzioni edificate prima di una certa data o ancora la ricerca di quegli edifici costruiti prima dell'entrata in vigore della legge sull'obbligatorietà della antisismicità.

### 1.3.4 **Funzioni Statistiche**

Data una mappa, vi è una serie di notizie standard che può essere utile sapere per farsi una idea della correttezza della mappa o della diversificazione urbana.

A livello applicativo alcuni esempi potrebbero essere il numero di strutture di un certo tipo, per esempio farmacie o tabacchi, presenti nel territorio oppure la percentuale di

suolo occupata da feature appartenenti ad una specifica categoria, per esempio zone verdi.

Sebbene questo genere di informazioni trova particolare rilevanza a livello applicativo, a livello di collaudo trova anche una sua rilevanza ed utilità nell'individuare ad esempio potenziali errori di attribuzione. Ad esempio una percentuale di "Baracche" presenti nella mappa pari a zero indicherebbe la classificazione di tali strutture come generici edifici o comunque in modo non corretto.

## **1.4 Analisi dei dati di una mappa: la componente geografica**

Come detto in precedenza la componente geografica di una mappa rappresenta la forma geometrica e la posizione di ogni oggetto presente. La forma di un oggetto e la sua esatta posizione nello spazio sono informazioni di elevata rilevanza in tutti i settori di utilizzo, alcuni di essi possono essere applicazioni legate alla viabilità o alla gestione del patrimonio. Si pensi ad un immobile che, per errore di rappresentazione, configuri in un'area a diversa tassazione o di un potenziale percorso per la costruzione di una strada che poi risulta invece essere intralciato da una costruzione la cui posizione nella mappa non era stata inserita correttamente.

### **1.4.1 Analisi componente geografica: collaudo**

Uno dei comuni controlli che si possono effettuare sulle topologie è quello di correttezza fra due layers. Un esempio classico può essere dato dalle strade che devono confinare ma non intersecare le altre geometrie. Altri esempi sono dati dalle curve di livello che non possono intersecare se stesse o dalle linee di accesso che devono intersecare i bordi delle strade. Questi controlli possono essere fatti in fase di collaudo ed in alcune piattaforme è possibile specificarli come vincoli al fine che siano sempre rispettati anche in successive modifiche dei dati.



### 1.4.2 Analisi componente geografica: applicazione

Alcuni controlli legati ad una corretta gestione del territorio possono per esempio prendere spunto dalla norma che obbliga gli edifici a stare oltre un fissato raggio dalle linee di costa e quindi si potrebbe richiedere l'individuazione di tutte quelle costruzioni abusive. Altri esempi d'utilizzo possono essere dati da controlli sulla distanza l'uno dall'altro di alcune tipologie di edifici quali ad esempio i tabacchi o di abitazioni da centrali elettriche o industrie. Un altro controllo che si può effettuare sulla componente geografica è nell'ambito dei piani regolatori comunali, ovvero che questi rispecchino i piani regolatori provinciali e regionali. Ovviamente i controlli sulla componente geografica possono essere associati a controlli o restrizioni in funzione del valore degli attributi e quindi si potrebbe imporre che tutti gli edifici che ricadono in un'area protetta, o che si trovino ad una data distanza da un oggetto identificato per mezzo di un attributo, abbiano altezza minore di un valore che può essere fisso o in funzione del proprio perimetro o che siano stati sottoposti a collaudo negli ultimi dieci anni. Trova applicazione pratica l'esempio appena riportato per quelle aree vittime di eventi sismici e quindi con edifici potenzialmente esposti a crepe o per edifici circondanti zone ad alto impatto storico-culturale in cui l'arbitraria altezza sminuirebbe l'imponenza di un dato Bene. Altro esempio di associazione tra controlli sulla componente geografica e controlli descrittivi potrebbe essere dato dall'obbligo per tutti gli edifici di tipo condominiale appartenenti ad aree particolari di contenere anche dei posti auto e a edifici di tipologia non condominiale di avere un giardino interno. Tale provvedimento è già stato intrapreso da diversi comuni per incrementare la diffusione delle aree verdi nella città.

## 1.5 Operazioni di congruità

Si è visto tra le funzioni statistiche e controlli sulla componente geografica la necessità di sapere la percentuale di oggetti appartenenti ad una specifica categoria presenti nel

territorio o controllo che dati oggetti stiano ad una distanza minima l'uno dall'altro. A volte tutto ciò non è sufficiente poiché si vuole una condizione ancora più forte ovvero che tutto il territorio sia coperto da oggetti che hanno un proprio raggio d'azione. Altri controlli che si potrebbero voler fare potrebbero essere la corrispondenza tra due diverse rilevazioni. Si pensi per esempio a ponti radio per comunicazioni o a punti di pronto intervento il cui raggio d'azione è in funzione della dimensione e tipologia del territorio. Dato un layer lineare o aerea, rappresentante la potenza di segnale radio percepita in una zona, può essere utile applicare dei controlli di congruità al fine di verificare la corrispondenza tra le rilevazioni puntuali e le rilevazioni aeree o lineari.

### **1.5.1 Copertura di un Layer tramite punti con distanza minima assegnata**

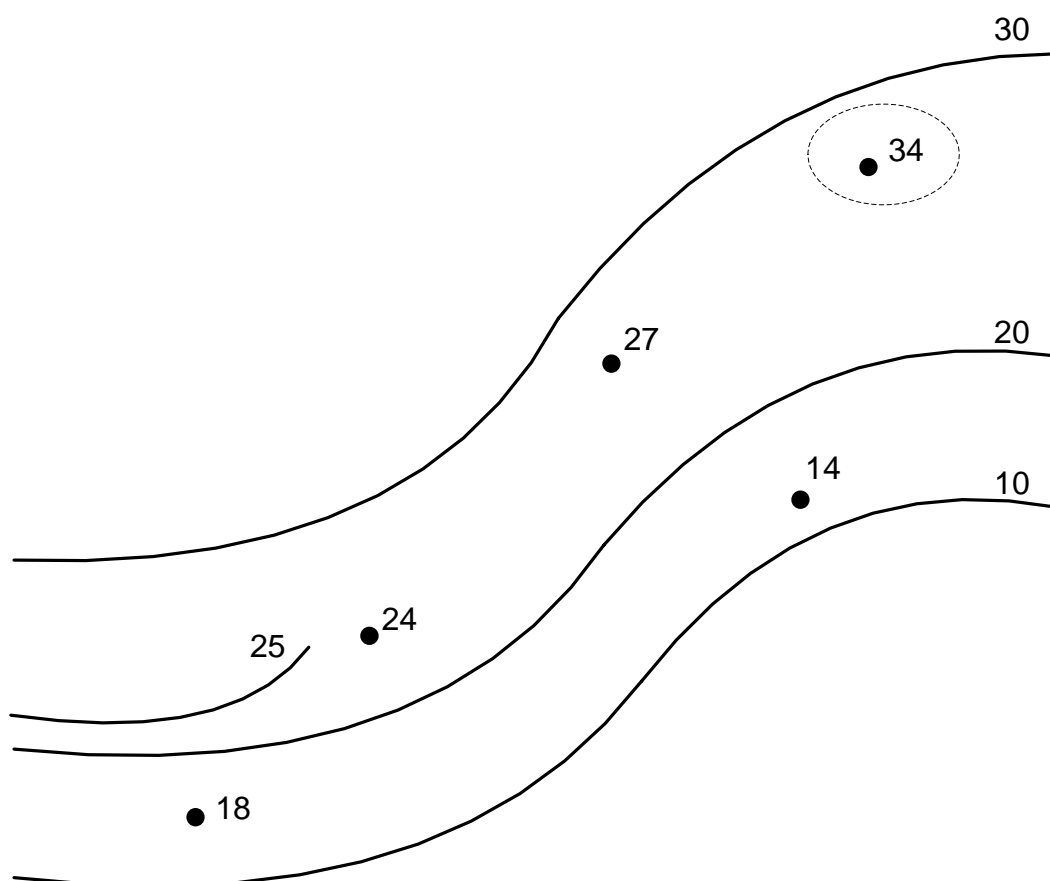
In alcune circostanze può essere necessario controllare che l'intero layer sia coperto per tutta la sua superficie da oggetti riguardanti una entità o delle rilevazioni. Il caso più ricorrente è quello dei punti quotati. Spesso si vuole assicurare che l'intera superficie sia stata sottoposta a controllo planimetrico e che quindi siano state effettuate, a distanze non superiori ad un fissato valore, le rilevazioni indicanti l'altezza rispetto al livello del mare. Ciò si rende necessario per una più accurata e corretta informazione fornita dalla mappa, utile al fine di successivi controlli anche in visione di possibili alluvioni, frane o straripamenti di fiumi.

Come già detto nella parte introduttiva un controllo di questo genere può essere utilizzato anche in altre applicazioni quali per esempio il controllo di un segnale radio o rilevazioni meteorologiche attraverso la copertura del territorio con dei rilevatori.

### **1.5.2 Congruità tra la misura di un punto e le curve che lo contengono**

Dalla necessità di assicurarsi che un territorio sia coperto da una data rilevazione, nasce la necessità di verificare che tale rilevazione non discordi dalle rilevazioni effettuate in altri layers. Un esempio potrebbe essere quello di controllare che non vi sia discordanza

tra le rilevazioni effettuate per mezzo di punti quotati e curve di livello. Per fare ciò è necessario per ogni punto quotato controllare che abbia un valore congruo con le curve di livello che lo “contengono”. Una simile operazione, concettualmente molto semplice, si traduce in una serie di controlli non altrettanto semplici a causa della natura delle geometrie e soprattutto delle possibili combinazioni dei punti all’interno delle curve.



## Capitolo 2

# Approcci al problema

### Introduzione

Nel precedente capitolo si è visto quali sono le principali tecniche per la realizzazione di una mappa digitale georeferenziata e quali sono i principali controlli per verificarne la bontà. Si è visto che, nonostante i vari controlli e tecniche raffinate, le mappe sono sempre soggette ad errori i quali possono anche avere un notevole impatto nell'utilizzo quotidiano. Si sono visti i principali errori che si possono commettere, ed alcuni di essi, sebbene siano abbastanza immediati da verificare, non vengono controllati. Alla luce di questi errori, si esamineranno le soluzioni esistenti per la risoluzione e, con l'ausilio di persone impegnate quotidianamente nell'uso e controllo di mappe, proveremo a individuare una strategia di soluzione del maggior numero di incoerenze.

### 2.1 Specifica Utente

Dopo un confronto con alcuni operatori, sia nel campo del collaudo che nel campo dell'utilizzo, è emerso che, sebbene per motivazioni diverse, questi spesso non procedono con l'accuratezza che vorrebbero perché il sistema a disposizione non li aiuta a farlo. Vediamo quali sono le esigenze che l'utente richiederebbe nel caso del collaudo di una mappa e nel caso del suo utilizzo.

### 2.1.1 Collaudo

Dopo aver individuato le zone della mappa da controllare, il collaudo si sviluppa in due fasi. Una prima fase è il controllo di corrispondenza tra la mappa e la realtà, ovvero si verifica che tutti gli oggetti presenti nella mappa esistano realmente e soprattutto che sia vero il viceversa, ovvero che siano rappresentati in mappa tutti gli oggetti presenti. Questa prima fase consiste nel recarsi fisicamente sul luogo per verificare la corrispondenza tra mappa e realtà: questa fase esula dal problema che stiamo affrontando. Nella seconda fase del collaudo l'operatore provvede a stilare una serie di possibili incongruenze che potrebbero essere presenti nella mappa da esaminare per poi procedere alla verifica manuale. E' a questo punto che l'utente incontra la maggiore difficoltà poiché i controlli dovrà effettuarli manualmente e proprio per questo motivo solo su porzioni ristrette della mappa con la possibilità quindi di non rilevare delle incongruenze presenti nelle restanti zone della mappa non esaminate. Alcune piattaforme permettono di effettuare dei controlli di tipo descrittivo ed anche dei controlli base sulla componente geografica, ma non consentono di elencare più di un controllo da far successivamente eseguire in maniera sequenziale e automatica. La richiesta dell'utente in questa fase è la possibilità di avere uno strumento che gli permetta di specificare dei vincoli o delle regole sulla componente geografica e sugli attributi e che queste siano verificate in modalità batch, ovvero sequenzialmente e senza l'intervento dell'utente, al fine di poterle applicare sull'intera mappa anziché su una ristretta regione di essa. Ovviamente tali regole dovranno produrre un output indicante l'esito e questo potrebbe essere uno shapefile contenente le features che hanno violato la regola oppure una lista di valori nel caso di funzioni statistiche, ed in ogni caso un file di log indicante la corretta interpretazione ed esecuzione della regola e l'output prodotto.

### 2.1.2 Applicazioni

Le possibili applicazioni su una mappa sono innumerevoli, ma spesso sono ostacolate dalla non immediata semplicità degli strumenti messi a disposizione dall'applicativo

utilizzato. L'utente medio, ed in genere anche l'utilizzatore più esperto, non procede ad effettuare dei controlli o applicazioni poiché richiedono una interazione con l'utente e la comprensione dello strumento messo a disposizione non sempre banale. Alcuni applicativi (ad esempio ArcMap, ArcInfo, Autodesk Map3D) mettono a disposizione uno strumento che guida passo passo l'utente nella scelta delle possibili operazioni ma questo, oltre ad avere un numero di applicazioni limitato, non permette la specifica di più condizioni da eseguire in modalità sequenziale, ed in più richiede la conoscenza della logica di funzionamento del linguaggio SQL, poiché tutte le piattaforme si basano su database relazionali, e della lingua inglese.

In conclusione l'utente richiede la necessità di poter specificare più regole o condizioni per applicarle autonomamente in modalità sequenziale e, soprattutto, la possibilità di poterle specificare in un linguaggio semplice, in pratica così come le "pensa", senza dover ricorrere a complicate riformulazioni matematiche e traduzioni.

## **2.2 Il Geodatabase ESRI**

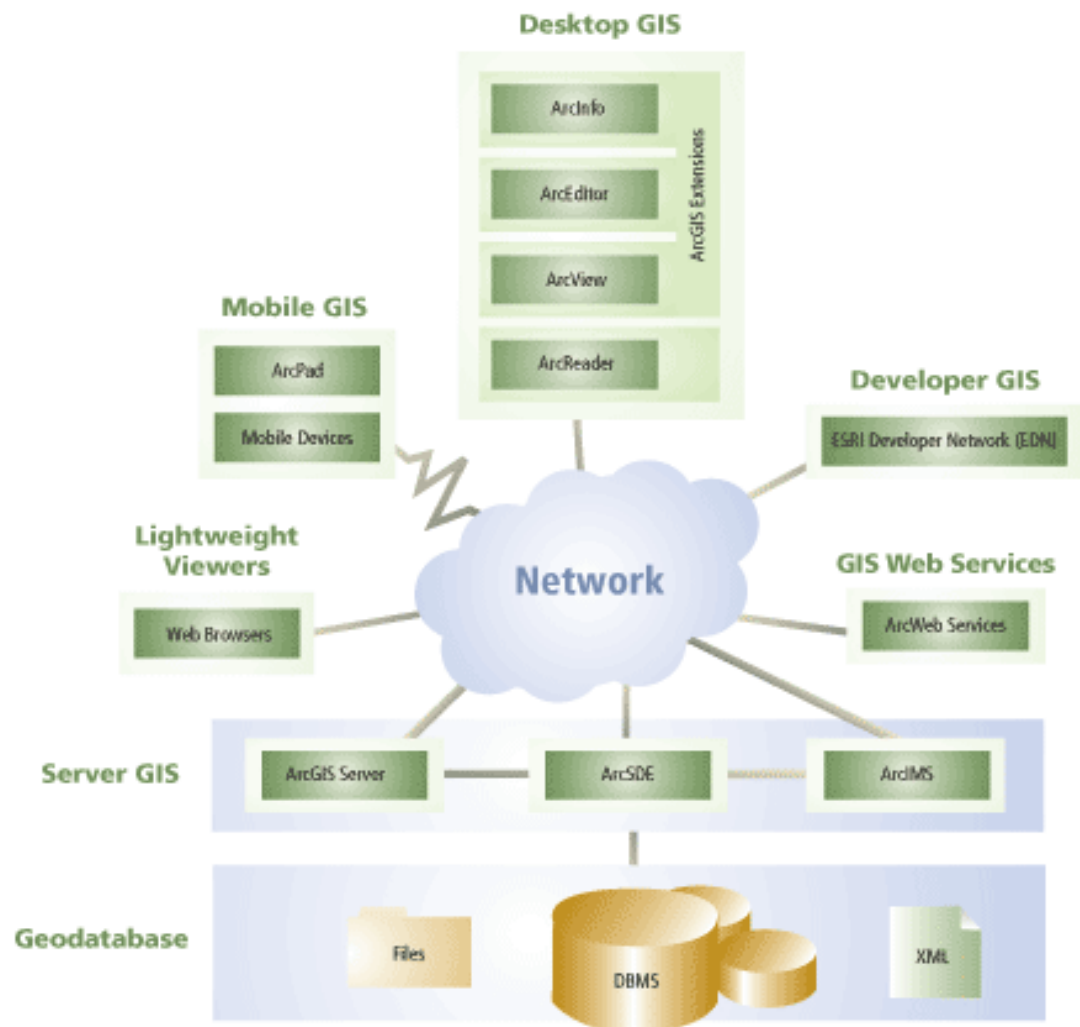
Iniziamo con l'esaminare una prima soluzione disponibile nel mercato fornita dalla ESRI®, società leader nel settore. La ESRI sfruttando la sua posizione di prima azienda inseritasi nel mondo GIS ed adesso leader nel settore, ha sviluppato una suite di applicativi denominata ArcGIS in grado di accompagnare l'utente dalla creazione di una mappa all'utilizzo quotidiano di essa. Per far questo si avvale di concetti semplici e fondamentali usati nella gestione e memorizzazione delle informazioni, e su questi applica un motore che provvede all'evasione delle richieste che arrivano da applicativi a più alto livello. Nella famiglia dei prodotti ArcGIS evidenziamo gli applicativi ArcInfo ed ArcSDE.

ArcInfo è il più completo applicativo della suite per la gestione e manipolazione di mappe. Con ArcInfo è possibile caricare e visualizzare diversi formati di mappe non solo vettoriali, sovrapporle ed assegnare livelli di trasparenza. E' possibile, oltre che visualizzare, modificare una mappa andando a modificare fisicamente la forma,

dimensione o posizione di una feature, oppure inserirne una nuova o eliminarne una esistente. Con ArcInfo è anche possibile eseguire alcune interrogazioni sulla mappa con condizioni sugli attributi e condizioni spaziali, quali l'intersezione o sovrapposizione, e la possibilità di applicare un buffer alla geometria, ovvero di considerare come se la geometria fosse di una dimensione  $x$  più grande o piccola.

ArcSDE invece fornisce un'interfaccia agli applicativi quali ArcInfo con i principali RDBMS commerciali, quali IBM DB2 Universal Database and Informix Dynamic Server, Oracle, and Microsoft SQL Server. ArcSDE supporta l'accesso multi utente ed a database relazionali distribuiti, fornisce una serie di servizi che migliorano le prestazioni nella gestione dei dati, estende i tipi di dati memorizzabili nel RDBMS ed abilita la portabilità dei dati tra gli RDBMS.

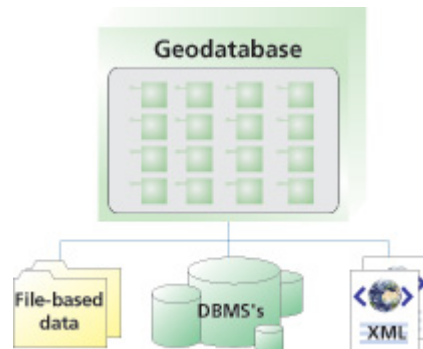
Si è parlato degli applicativi per la gestione dei dati, ma il cuore di ArcGIS, ovvero la fonte dei dati su cui opera l'intera suite, è data dal Geodatabase, abbreviazione di geographic database. Il Geodatabase è il modello implementato dalla ESRI per rappresentare informazioni geografiche usando la tecnologia standard dei database relazionali. Il geodatabase definisce tutti i tipi di dati che possono essere usati in ArcGIS (features, rasters, indirizzi, indagini statistiche) e come essi sono rappresentati, come ci si accede, si memorizzano e si gestiscono.



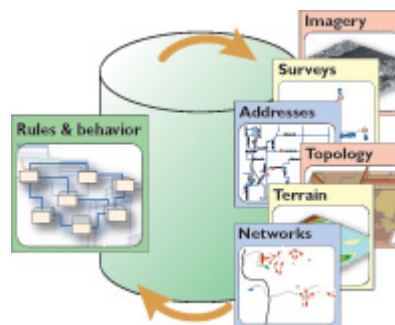
### 2.2.1 La struttura

Il geodatabase è un'architettura multi strati, applicazione e memorizzazione, dove gli aspetti legati alla scrittura e lettura fisica dei dati, definizione attributi, processamento delle query e gestione delle transazioni multi utente è delegata al RDBMS, mentre l'integrità dei dati ad alto livello e le funzioni di processamento delle informazioni sono mantenute dall'applicazione GIS, e tutti gli applicativi ArcGIS interagiscono con questo generico oggetto GIS che è il geodatabase.





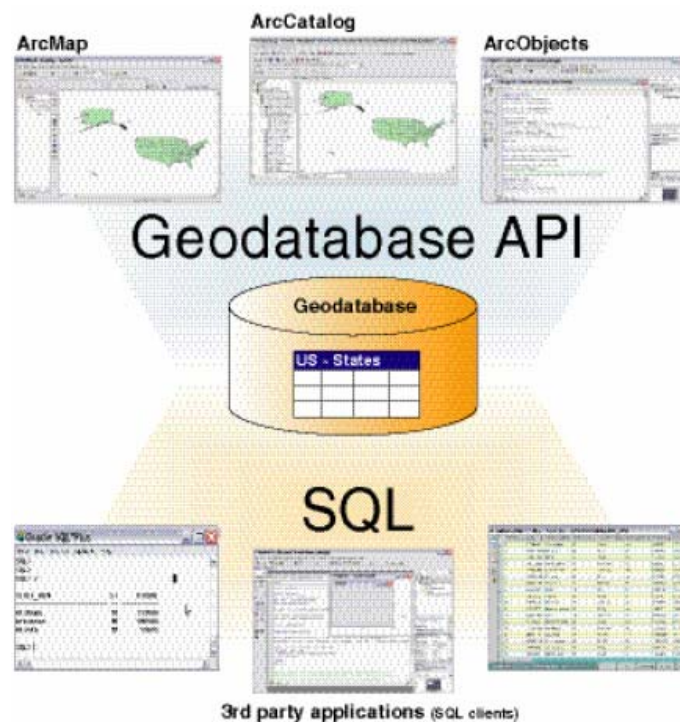
La separazione del geodatabase logico dalla memorizzazione permette il supporto di numerosi tipi di file, ad esempio il supporto XML per i metadati e a diversi RDBMS. L'applicazione GIS ha la responsabilità di mantenere e definire lo schema usato per rappresentare i dati geografici e le applicazioni logiche per provvedere al loro mantenimento e integrità. L'insieme delle regole e delle definizioni sono memorizzate in una collezione di tabelle del geodatabase memorizzate nel RDBMS.



Al fine di rendere l'architettura flessibile e più semplice, i controlli sulla topologia e le geometrie vengono memorizzati come combinazione delle tre geometrie base: punto, linea, poligono. Ogni entità del mondo reale verrà mantenuta in una struttura chiamata FeatureClass all'interno della quale ogni singolo oggetto sarà una feature: il riferimento spaziale di una FeatureClass è mantenuto in una struttura chiamata FeatureDataset, che potrà essere riferita da più FeatureClass. Al fine di impedire possibili errori ed evitare ridondanze, le features contigue della stessa FeatureClass condivideranno fisicamente la parte geometrica in comune, così da evitare buchi o interruzioni tra le geometrie sia in fase di costruzione che di modifica.

### 2.2.2 Il linguaggio

Ci siamo soffermati molto sul funzionamento e sulla struttura della soluzione ESRI sia perché è quella presa come riferimento nel nostro progetto, sia perché le altre soluzioni non si discostano molto da essa. Si è detto che tutti gli applicativi ArcGIS accederanno ai dati attraverso il geodatabase e si è anche visto come alcune geometrie vengano fisicamente condivise da due features contigue al fine di controllare la correttezza della topologia. Si è visto il geodatabase usare una base di dati relazionale per la memorizzazione, quindi è possibile imporre dei vincoli ai campi, vincoli che possono essere di valore minimo, massimo o appartenenza ad un range. La struttura del geodatabase, oltre ai vincoli visti tipici di un DBMS, permettere di specificare su una serie di FeatureClass dei vincoli legati alla topologia, alla sovrapposizione o intersezione, totale o parziale, delle features. E' possibile per esempio specificare che le features di una FeatureClass non si sovrappongano tra loro o che coprano per intero le features di un'altra FeatureClass. Ogni qualvolta si tenti di modificare una FeatureClass inserendo, rimuovendo o modificando una feature, si controllerà che l'operazione richiesta rispetti i vincoli imposti e solo in tal caso sarà effettuata, e, nel caso si utilizzi ArcSDE, verrà effettuato il versionamento dei dati, ovvero salvata copia del dato prima della modifica per permettere il roll-back in caso di necessità. Si è detto che è possibile accedere ai dati del geodatabase attraverso delle API sfruttate da tutti gli applicativi ArcGIS, ma è possibile accedervi anche direttamente attraverso istruzioni SQL. Per istruzioni che non vanno a modificare i dati o che non hanno alcun impatto sulla geometria della FeatureClass, accedervi bypassando il geodatabase potrebbe significare un incremento di prestazioni, ma si corre il rischio di violare dei vincoli presenti nel geodatabase o di ottenere un aggiornamento parziale dei dati a causa della non applicazione di regole imposte su quella FeatureClass.



Maggiore è il rischio di errori nell'accesso tramite SQL quando si va a modificare la componente geografica utilizzando estensioni per il supporto ai dati spaziali del RDBMS usato: per esempio, con lo Spatial Extender SQL per DB2, è possibile accedere direttamente ai dati con istruzioni SQL di lettura, modifica e scrittura anche dei dati spaziali e geometrici senza il coinvolgimento del geodatabase e di ArcSDE, quindi non effettuando il versionamento dei dati ed alcun controllo delle regole e vincoli sulla topologia.

### 2.2.3 I Limiti

Si è detto che la soluzione ESRI permette di poter effettuare dei controlli sui valori degli attributi, come ad esempio l'appartenenza ad un range, e anche dei controlli sulla componente geografica, come l'intersezione o il buffer, ma, oltre a non coprire l'intera casistica delle regole e condizioni che l'utente vorrebbe applicare, non è possibile una esecuzione batch delle regole.

### 2.2.3.1 Collaudo

Gli strumenti messi a disposizione, sia per vincoli da rispettare tra layers sia per vincoli all'interno del layer, possono aiutare molto in fase di collaudo ma non sempre sono sufficienti.

Si pensi alle operazioni di congruità: non è possibile tramite gli strumenti messi a disposizione verificare che l'intera mappa sia coperta da rilevazioni puntuali o che questi siano congrui alle rilevazioni lineari.

Oltre ai suddetti limiti, è da notare che i vincoli di coerenza tra le geometrie di più layers vengono meno quando si vanno a confrontare tali layers con altri fatti in momenti o da ditte diverse, o quando uno di questi per comodità viene scisso in due; per esempio, si potrebbe voler scindere un layer delle falde acquifere, per individuare falde potabili e non, oppure pubbliche e private. In tal caso non è detto che i vincoli vengano ereditati dai due layers e specificare di nuovo i vincoli o effettuare dei controlli di correttezza richiederebbe la conoscenza del linguaggio SQL e delle istruzioni messe a disposizione dalla piattaforma, operazioni non banali e non alla portata di tutti.

### 2.2.3.2 Applicazione

Per quanto riguarda l'uso dal punto di vista applicativo di una mappa, non è possibile applicare un buffer o una condizione di distanza con valore non costante, ovvero memorizzato all'interno di un attributo della feature. Si pensi al caso in cui si vuole che le abitazioni siano ad una distanza di sicurezza da fonti con emissione di onde radio; in questo caso la distanza di sicurezza dipende dall'oggetto stesso, poiché in funzione della sua potenza, e quindi è memorizzata come valore di un attributo.

Un forte limite che potrebbe non essere dello strumento, ma che comunque porta gli utenti a non utilizzarlo a pieno, è la non estrema semplicità di utilizzo e creazione delle condizioni che si vogliono applicare. L'applicativo mette a disposizione una guida per la creazione di condizioni sugli attributi ma, anche se quando si tratta di operazioni semplici quali verificare che il valore di un attributo sia minore o maggiore di un valore o di un altro attributo, il suo utilizzo è alla portata di tutti, nel momento in cui si

vogliono effettuare operazioni di somma o media non lo è più, e richiede di procedere attraverso una procedura diversa. Lo stesso dicasi per condizioni che riguardano sia restrizioni sia spaziali che sugli attributi: in tal caso bisogna procedere effettuando prima la restrizioni sugli attributi, e sul risultato applicare la restrizione spaziale seguendo un'altra procedura.

Tutto ciò porta l'utente ad applicare delle condizioni in modo non completo omettendo qualche restrizione o peggio ancora a non effettuarle.

## **2.3 Altri Approcci**

Intergraph ha puntato molto sull'acquisizione dei dati con lo sviluppo del DMC<sup>®</sup> (Digital Mapping Camera), un'accurata fotocamera per riprese aeree dotata di sofisticato sensore CCD per assicurare un'alta qualità d'immagine, legata ad un avanzato rilevatore GPS per l'accuratezza delle coordinate, e di una tecnologia proprietaria per la compensazione della qualità dell'immagine in movimento. Tutto questo, insieme all'ampia capienza di memorizzazione e al software per il postprocessing, costituisce uno dei punti di forza di Intergraph. Questa software house non si è interessata solo alla acquisizione e creazione della mappa, ma anche alla gestione con il pacchetto GeoMedia. Per attirare maggiormente il consenso del mercato ha sviluppato una serie di applicativi specifici per diversi settori, dal trasporto aereo al trasporto su rotaie o su strada, al rilevamento e gestione truppe (per il dipartimento della difesa) o ancora al controllo e gestione della rete gas ed elettrica. Anche Intergraph permette di effettuare dei controlli sia sui valori degli attributi sia sulla componente geografica, in particolare attraverso la soluzione Geomedia Curator [10], componente di una suite denominata Geo-Intelligence Production System (GIPS), ma, dalla documentazione disponibile, sembra non aggiungere nulla di nuovo alla soluzione ESRI.

Anche Autodesk ha diversificato il proprio pacchetto software per soddisfare le diverse esigenze nei diversi comparti. Sfruttando l'esperienza maturata e la posizione

dominante in ambito CAD, grazie all'applicativo Autocad, ha puntato molto sulla precisione e l'integrazione dei dati vettoriali GIS con quelli di tipo CAD. Autodesk punta molto sulla capacità di memorizzazione dei dati in doppia precisione e sull'integrazione dei dati GIS con CAD [13] ma punta molto anche sulla visualizzazione e gestione di essi via web [14]. Autodesk, così come ArcGIS, offre il supporto per RDBMS quali Oracle e applica dei controlli di correttezza della topologia in fase di editing di una mappa, ma non sembra, dalla documentazione disponibile [11], offrire degli strumenti per un controllo di correttezza o analisi post-processing, se non legate alle specializzazioni del proprio applicativo, per le diverse soluzioni quali fornitura energia elettrica, costruzioni, trasporto, etc.

In conclusione sia Intergraph che Autodesk utilizzano un database relazionale per la memorizzazione dei dati, così come ArcGIS, e puntano molto sulla specializzazione dell'applicativo, soprattutto Intergraph, ma entrambe non introducono grosse novità nell'ambito di nostro interesse.

## **2.4 Strategia di una possibile soluzione**

Si è visto quali sono le esigenze dell'utente e quali sono le opportunità che offrono i vari applicativi presenti sul mercato, ognuno con i propri approcci e pregi, ma nessuno completamente corrispondente alle esigenze dell'utente in questione. Partendo dalle specifiche utente sopra riportate, si proverà a individuare una possibile strategia che possa il più possibile andare incontro alle esigenze dell'utente pur mantenendo una sua generalità. Si vuole quanto più possibile fornire una soluzione che non sia specifica al settore dell'utente in questione ma sia poi riusabile in tutti gli ambiti GIS, e che sia indipendente per lo meno a questo livello concettuale dalla piattaforma su cui poi sarà realizzata.

### 2.4.1 Il linguaggio

Il target di persone che si vuole raggiungere con questa soluzione è abbastanza variegato, ma sicuramente si vuole prediligere quegli utenti che non hanno nozioni informatiche ed a volte nemmeno di lingua straniera, e quindi incontrano, sebbene molto preparati nel loro settore, grosse difficoltà ad utilizzare gli strumenti offerti dall'applicativo in uso per effettuare dei controlli. Spesso il linguaggio utilizzato in tali applicativi è SQL arricchito di funzioni spaziali quali buffer e intersect ed a volte viene fornito uno wizard per la creazione passo passo della regola con le possibili scelte dei comandi. Nonostante lo wizard e la semplicità del linguaggio SQL, l'utente non sempre riesce a sfruttare a pieno l'applicativo, poiché in ogni caso il linguaggio richiede un minimo di conoscenza della sintassi e della logica matematica, e comunque li porta a dover riscrivere ed adattare la regola ad ogni controllo; soprattutto, nel caso dell'esecuzione di più regole, spesso non è possibile specificarle ed eseguirle consecutivamente in modo autonomo ed automatico ma è necessaria la scrittura ed esecuzione di una nuova regola solo al termine della precedente. Inoltre, sebbene gli applicativi esistenti permettano di effettuare delle analisi sia descrittive che topologiche, non è possibile, per lo meno agevolmente ovvero senza ricorrere ad artifici creandosi ogni volta delle mappe di appoggio ad hoc, effettuare delle operazioni complesse di congruità come quelle discusse nel capitolo precedente.

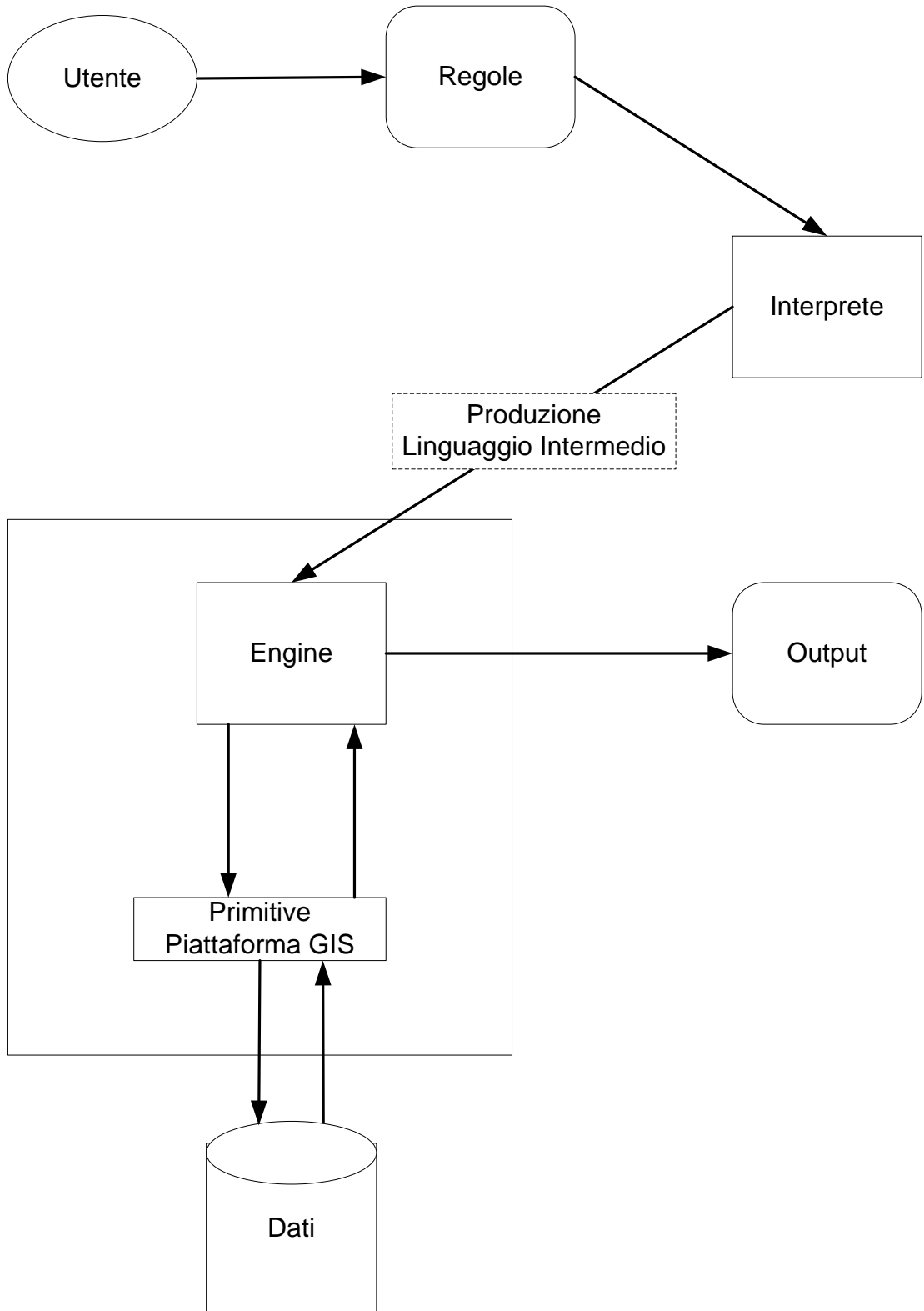
In conclusione si vuole permettere di esprimere delle regole in un linguaggio quanto più naturale possibile, e quindi legato anche alla lingua italiana, con assenza di ambiguità e aumentare il potere espressivo del linguaggio SQL con annesse funzioni spaziali. Tali regole si vuole poterle elencare in una struttura, tipicamente un file di testo, alla quale poi accederà l'applicativo per eseguirle in modalità batch, fornendo per ogni regola un risultato che potrà essere dato da un nuovo file o FeatureClass oppure da una annotazione che riporti se sono o non sono state trovate violazioni alla regola.

## 2.4.2 La struttura

La soluzione dovrà avere una struttura modulare ed a livelli, al fine di permettere la massima portabilità e manutenibilità. Il modulo che si occuperà di interpretare la regola dovrà essere totalmente indipendente dal modulo che andrà ad eseguirla, e quest'ultimo dovrà essere strutturato in modo flessibile al fine di adattarsi all'architettura sottostante. Il Parser della regola disporrà di un proprio set di parole chiave in funzione delle quali effettuerà l'operazione richiesta. Questo set sarà suddiviso per tipologie e quindi facilmente adattabile alla lingua usata e sarà totalmente indipendente dall'infrastruttura sottostante. All'utente sarà richiesta solo la conoscenza del nome dei layer e degli attributi da inserire come condizione in una regola.

Il modulo che si occuperà dell'esecuzione metterà a disposizione del Parser un'interfaccia per la specifica dei dati e delle operazioni da effettuare. Questo modulo, che chiameremo Engine, si servirà delle funzioni dell'architettura sottostante per eseguire fisicamente le condizioni espresse dal linguaggio. Il modulo dovrà produrre in output per ogni regola, una FeatureClass contenente gli oggetti che la violano, e dovrà produrre un file di log in cui riporterà per ogni regola il tempo di esecuzione, la sintassi riconosciuta, le operazioni applicate e l'esito finale con conseguente nome del file prodotto.





## Capitolo 3

# Progettazione e Implementazione

### Introduzione

Dopo la panoramica sulle esigenze dell'utente e su quali siano le opportunità che offrono i vari applicativi presenti sul mercato, si è posto il problema di decidere su quale piattaforma realizzare la strategia di soluzione sopra discussa al fine di testarne con mano la bontà e l'utilità. A tal fine, senza esprimere un giudizio su quale sia l'applicativo più completo o che vada più incontro alle esigenze dell'utente, si è scelta la piattaforma ESRI, sia per la maggiore diffusione sia perché è quella di cui si dispone e si ha una maggiore conoscenza ed accesso a funzionalità e supporto.

### 3.1 Architettura e Tecnologia di Base

La suite ArcGIS mette a disposizione un ambiente di sviluppo che permette di accedere ai dati manipolandoli e modificandoli da codice così come è possibile farlo da applicativo. ArcView, ArcEditor e ArcInfo facenti parte della suite ArcGIS offrono accesso ai dati sia attraverso codice che attraverso interfaccia grafica ed ognuno di questi aggiunge funzionalità rispetto al precedente. Utilizzare o disporre di un applicativo della linea ArcGIS non è l'unico modo per accedere al geodatabase, la ESRI ha messo a disposizione le stesse funzioni attraverso le librerie MapObjects per permettere l'accesso al geodatabase in un linguaggio che non sia necessariamente quello

usato dagli applicativi della suite ArcGIS, ovvero Visual Basic ma anche Java, C++ ed un qualsiasi linguaggio del .Net Framework. Le librerie MapObjects sono disponibili con licenza in prova per trenta giorni direttamente dal sito ESRI ma disponendo di ArcInfo si è preferito sviluppare la soluzione attraverso tale applicativo e quindi in Visual Basic.

### **3.2 Analisi regole**

Per progettare quello che poi sarebbe stato il nostro interprete della regola si è proceduto stilando una serie di possibili regole emerse dalle discussioni con alcuni utenti:

1. Le aree di PRG con TIPO = 6,7,11 devono essere contenute nelle aree di PTC con DEST <> “Parco”
2. Le linee di CDL non devono intersecare se stesse
3. Le aree di PRG con SUPERFICIE > 48 mq devono avere TIPO = 60
4. Gli EDIFICI non devono intersecare i PARCHI
5. Gli EDIFICI devono avere ALTEZZA < 48
6. Le RECINZIONI non devono intersecare le SCARPATE
7. Le linee di accesso devono intersecare i bordi delle STRADE
8. Il bordo degli EDIFICI deve stare oltre 100 m dalle coste
9. Gli EDIFICI di TIPOLOGIA = “Tabacchi” devono stare oltre 300 m da se stessi
10. Gli EDIFICI con Altezza > 50 devono avere numero elementi minore di 10
  
11. Gli EDIFICI entro 200 m da EDIFICI ID = “0054” devono avere ALTEZZA < 40
12. Gli Edifici contenuti in PRG devono stare oltre RaggioSicurezza da CentraliElettriche
13. Gli EDIFICI di TIPO=4 contenuti in PRG TIPO=6,7,11 devono avere ALTEZZA < MAXALTEZZA

14. Gli EDIFICI di TIPO=4 contenuti in PRG TIPO=6,7,11 devono avere ALTEZZA media minore di avgALTEZZA
15. Gli EDIFICI di TIPO=4 contenuti in PRG TIPO=6,7,11 devono contenere le aree di GIARDINI
16. Gli EDIFICI di TIPO=4 contenuti in PRG TIPO=6,7,11 devono contenere le aree di GIARDINI e contenere POSTO\_AUTO
17. Gli EDIFICI contenuti in PRG TIPO=6 devono avere somma delle superfici minore di maxSuperficie
18. Gli EDIFICI contenuti in PRG TIPO=6 devono avere *superficie* minore di maxSuperficie
19. Gli EDIFICI contenuti in PRG TIPO=6 devono avere AREA cumulativa minore di maxSuperficie
20. Le aree di PRG tipo=6 che contengono EDIFICI devono avere area > 200.000

Dalle regole si è passato ad una loro generalizzazione rimuovendo gli articoli, le preposizioni e quanto di “non rilevante” lasciando quindi solo i dati strettamente necessari. I dati sono stati poi classificati in funzione di quello che rappresentano al fine di trovare una notazione standard che potesse rappresentarli:

1. Layer cond.attributi severità cond.Spaziale layer cond.attributi
2. Layer severità cond.Spaziale layer
3. Layer cond.attributi severità cond.possesso cond.attributi
4. Layer severità cond.Spaziale layer
5. Layer severità cond.possesso cond.attributi
6. Layer severità cond.Spaziale layer
7. Layer severità cond.Spaziale Feature-calculate layer
8. Feature-calculate layer severità cond.Spaziale layer
9. Layer cond.attributi severità cond.Spaziale layer cond.attributi
10. Layer cond.attributi severità cond.attributi
11. Layer cond.Spaziale layer cond.attributi severità cond.possesso cond.attributi

12. Layer cond.Spaziale layer severità cond.Spaziale layer
13. Layer cond.attributi cond.Spaziale layer cond.attributi severità cond.possesso cond.attributi
14. Layer cond.attributi cond.Spaziale layer cond.attributi severità cond.possesso cond.attributi
15. Layer cond.attributi cond.Spaziale layer cond.attributi severità cond.Spaziale layer
16. Layer cond.attributi cond.Spaziale layer cond.attributi severità cond.Spaziale layer cond.Spaziale layer
17. Layer cond.Spaziale layer cond.attributi severità cond.possesso cond.attributi
18. Layer cond.Spaziale layer cond.attributi severità cond.possesso cond.attributi
19. Layer cond.Spaziale layer cond.attributi severità cond.possesso cond.attributi
20. Layer cond.attributi cond.Spaziale layer severità cond.possesso cond.attributi

La generalizzazione delle regole nella forma appena mostrata ne evidenzia subito due tipologie, quelle che hanno lo scopo di verificare il valore di un attributo e quelle che invece verificano che una data condizione sulla componente geografica sia rispettata. A queste due tipologie si aggiungono le regole, come quelle dalla undici alla venti, in cui il layer di origine è frutto di una condizione spaziale con un altro layer. Da questa analisi e generalizzazione è emersa una loro possibile scrittura come combinazione dei seguenti simboli dove il carattere | ha il significato di oppure:

- Pre-Condizione:** Layer | Layer cond.Attributi | Feature-calculate Layer |  
 Feature-calculate Layer cond.Attributi |
- Condizione:** **Pre-Condizione** cond.Spaziale **Pre-Condizione** severità |  
**Pre-Condizione** severità
- Post-CondizioneD:** **Condizione** cond.Descrittiva cond.Attributi |
- Post-CondizioneS:** **Condizione** cond.Spaziale **Pre-Condizione** |  
**Post-CondizioneS** cond.Spaziale **Pre-Condizione** |

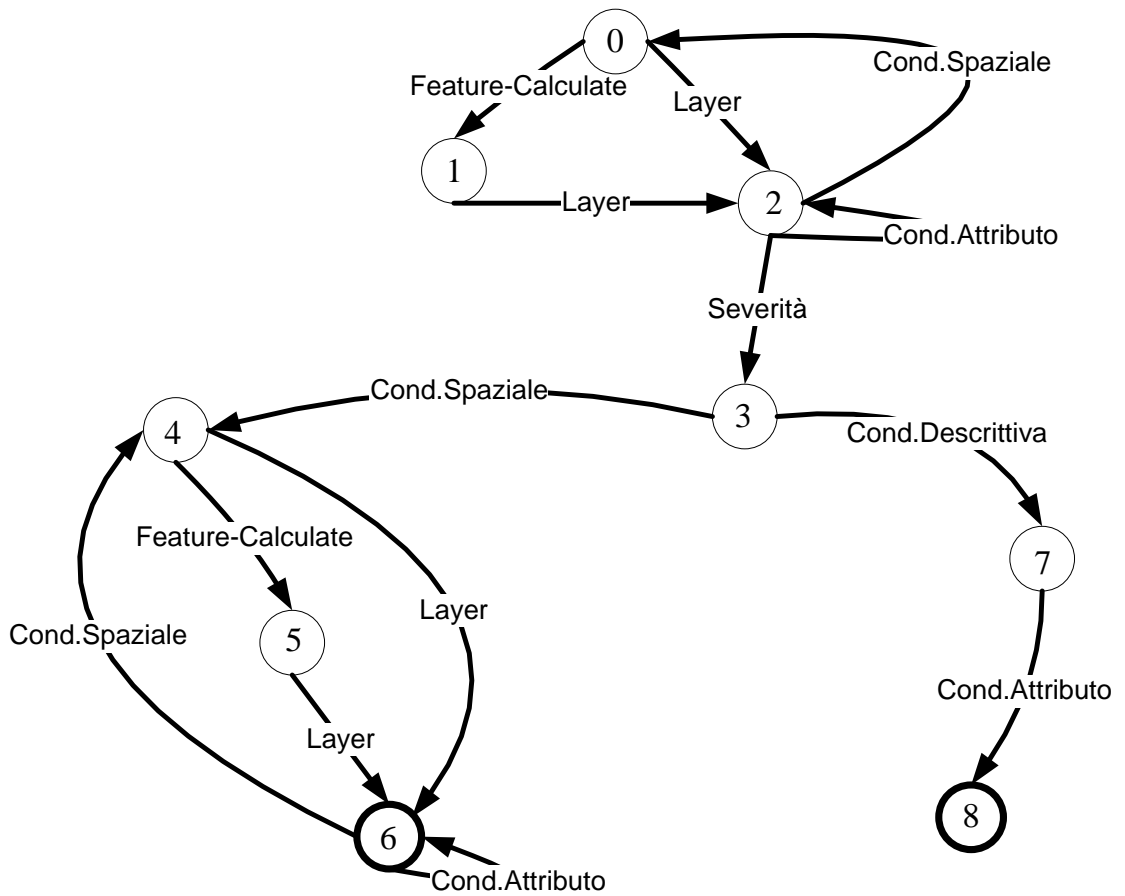
Questa suddivisione ha dato origine all'automa che guiderà il Parser nell'interpretazione della regola.

### 3.3 Parser

Il Parser è strutturato a moduli e parte dal concetto di automa a stati finiti in cui ad ogni stato si ha un numero finito di possibili ingressi ed in funzione di essi avviene la transizione da uno stato all'altro. Ogni stato rappresenta un punto ben preciso dell'esecuzione della regola e si distinguono in stati di transizione, stato iniziale e stati finali, nei quali si dovrebbe arrivare alla conclusione della regola.

#### 3.3.1 Schema Automa

L'automa, al fine di soddisfare le regole di analisi discusse nel primo capitolo ed il grado di semplicità richiesto dall'utente, procede in due fasi: l'individuazione del layer che si vuole esaminare e il controllo che si vuole effettuare. Il concetto di controllo è da estendere poiché si può richiedere di effettuare un controllo di tipo descrittivo e quindi che il valore di uno o più attributi rispetti certi vincoli, ma anche un controllo spaziale, ovvero che il layer intersechi o contenga la topologia di un altro layer. E' possibile applicare una condizione spaziale o descrittiva anche ad un layer risultante da una precedente applicazione di una condizione spaziale. Si pensi ad esempio alla necessità di controllare che tutti gli edifici in una zona a rischio sismico siano stati costruiti dopo una certa data oppure rispettando già le norme di sicurezza introdotte successivamente come obbligatorie per legge. Per questo motivo, ed in vista di potenziali occorrenze future, si è permesso di effettuare qualsiasi controllo sia spaziale che descrittivo tra un numero dinamico di layer. Per chiarire meglio il concetto si propone uno schema semplificato dell'automa del Parser dove lo stato zero è lo stato iniziale e gli stati sei ed otto sono quelli finali. L'etichetta di ogni arco rappresenta la famiglia dei possibili ingressi che permettono di transitare allo stato successivo. Per gli ingressi non riconosciuti non avverrà alcuna transizione di stato e si passerà all'input successivo, questo al fine di permettere una certa elasticità nell'uso degli articoli e preposizioni varie.



Come è possibile vedere a primo impatto lo schema si può suddividere in tre fasi: la prima è data dai nodi zero, uno, due, la seconda dai nodi quattro, cinque, sei, la terza dai nodi sette ed otto. Dalle tre fasi è stato escluso il nodo tre che ha la funzione di giunzione, ma spiegheremo meglio il concetto dopo aver analizzato le tre fasi.

Le fasi uno e due sono simili, per non dire identiche, fatta eccezione che la prima contiene lo stato iniziale mentre la seconda uno dei due stati finali. Entrambe permettono di riconoscere una regola che produca come risultato finale un layer su cui sono state applicate delle restrizioni spaziali e/o descrittive. Al layer risultato della prima fase si potranno applicare le volute condizioni descrittive risultanti dalla terza fase, oppure una o più condizioni spaziali iterando nella seconda fase. La terza fase ha lo scopo di riconoscere le condizioni descrittive da applicare al risultato della prima fase e quindi si pone come alternativa alla seconda fase. Il nodo tre, come detto precedentemente, funge da giunzione tra la fase uno e la possibilità di scegliere se

applicare una condizione sulla geometria (spaziale o topologica – fase due) o se applicare una condizione di qualità sugli attributi (descrittiva – fase tre). Oltre a fungere da connettore e divisore tra le due possibili condizioni, il nodo tre ha anche lo scopo di specificare la severità della condizione, ovvero se si vuole che le features risultanti vengano eliminate automaticamente o solo evidenziate, per esempio inserendole in un nuovo shapefile in modo da essere successivamente visualizzate come warning.

Si è detto che le etichette degli archi rappresentano una famiglia di possibili parole chiave che determinano la transizione da uno stato all'altro. Alcune delle parole chiave sono rappresentate dalla tabella sottostante in cui il simbolo | ha il significato di oppure ed è utilizzato per dividere le parole riconosciute.

Layer:	<i>Tutti i nomi dei layers che è possibile incontrare nelle regole</i>
Attributi:	<i>Tutti i nomi degli attributi dei layers incontrati, a runtime, nell'interpretazione della regola</i>
Valore:	<u>Attributi</u>   Numero   Stringa
Feature-calculate:	bordi   estremi   punto_centrale   baricentro   centroide
Severità:	deve   non deve   dovrebbe   non dovrebbe
Cond.Spaziale:	interseca   contiene   contenuto   coincide   stare entro <u>Valore</u> da   stare oltre <u>Valore</u> da   entro <u>Valore</u> da
Cond.Descrittiva:	avere   non avere
Operatori:	<   >   =   <>   minore   maggiore   uguale   diverso   contiene   inizia   finisce
Funzioni:	<u>Attributi</u> cumulativa   <u>Attributi</u> media   superficie_reale   somma_delle_superfici   numero_elementi
Cond.Attributi:	<u>Attributi</u> <u>Operatori</u> <u>Valore</u>   <u>Funzioni</u> <u>Operatori</u> <u>Valore</u>   <u>Cond.Attributi</u> e/o <u>Cond.Attributi</u>

I valori delle etichette Layer e Attributi non sono noti a compile-time poiché dipendono dai layers che si vogliono esaminare. A tempo di esecuzione l'etichetta Layer conterrà i nomi dei possibili layers che si vogliono esaminare. Questi valori potranno essere o i layers caricati nell'applicativo (come nel nostro caso quelli presenti in ArcMap) o quelli contenuti in una o più cartelle selezionate dall'utente. L'etichetta Attributi invece conterrà il nome di tutti e soli gli attributi dei layers incontrati nella regola in esame. Nel caso ci sia un conflitto, ovvero due layers all'interno della stessa regola che hanno un attributo con lo stesso nome presente nella regola, si utilizzerà la notazione layer.nomeAttributo per differenziarli. All'etichetta Valore può essere assegnato il nome di un attributo oppure una costante di tipo alfa-numerico. L'etichetta Feature-calculate specifica se nel calcolo della regola bisogna tenere conto della geometria



naturale della feature o se solo dei bordi nel caso di area o estremi nel caso di linea oppure del centroide o baricentro.

E' possibile che vi siano dei conflitti tra i nomi dei layers o attributi e le parole chiave. Nel caso degli attributi, il cui nome è uguale ad una delle parole chiave del simbolo Funzioni che non hanno come primo termine il nome di un attributo, si è usata la convenzione che il nome dell'attributo in questione sia specificato nella regola tra doppie virgolette ("nomeAttributo"). Nel caso dei layers, se un layer ha come nome una parola chiave dell'etichetta Feature-Calculate, si è seguita la convenzione degli attributi, ovvero che il nome del layer in questione nella regola sia specificato tra doppie virgolette ("nomeLayer").

Nel caso il nome di un layer o di un attributo sia formato da due termini separati da spazio, si è scelto di far rappresentare tale termine nella regola sostituendo lo spazio con il simbolo underscore (Strade\_Principali).

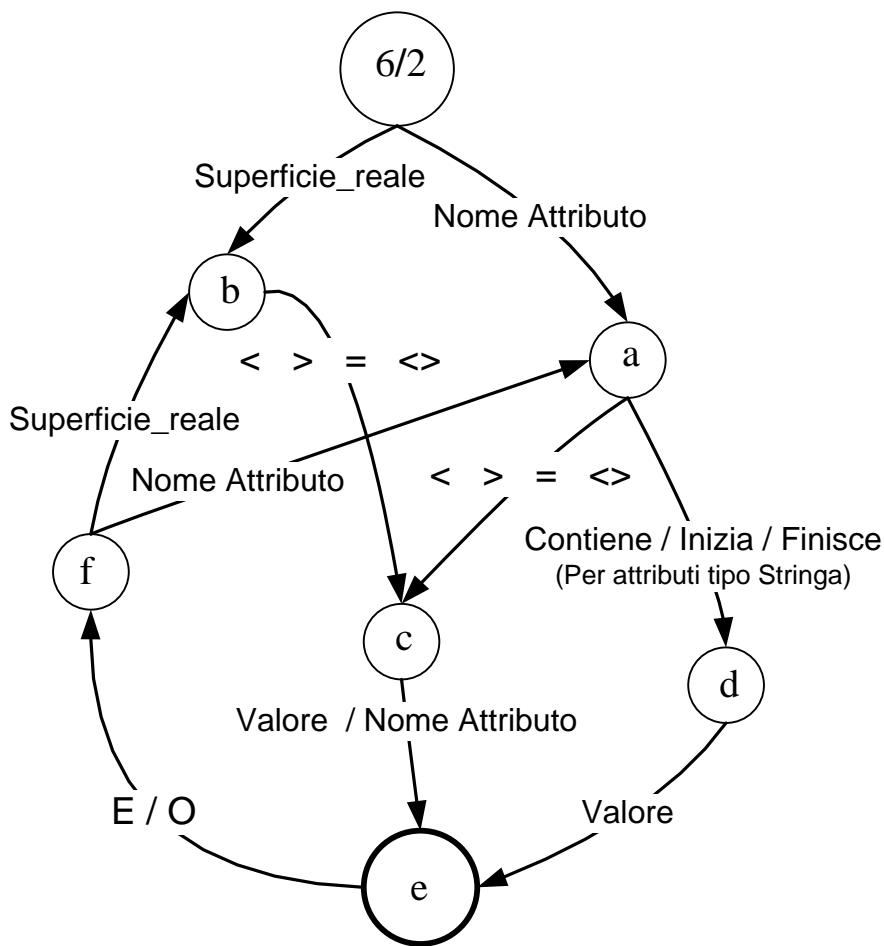
La Cond.Spaziale specifica il tipo di operazione spaziale da effettuare discriminando i termini in funzione delle coniugazioni, per esempio trattando in modo diverso il termine contiene o contenere dal termine contenuto e quindi applicando una operazione spaziale diversa. Così come l'etichetta Cond.Spaziale in realtà racchiude in sé un significato e quindi un potenziale che va oltre le generiche parole chiave sopra elencate, lo stesso vale per la maggior parte delle etichette presenti. Prendiamo ad esempio l'etichetta Cond.Attributi che tra tutte è sicuramente la più articolata.

### 3.3.1.1 **Raffinamento condizioni sugli attributi**

Dato un layer, è possibile specificare innumerevoli condizioni sugli attributi. Per evitare errori a tempo di esecuzione è necessario verificare la correttezza dei tipi dei parametri in ingresso alle funzioni che esprimono le condizioni. Si è visto che è possibile specificare come condizione che il valore di un attributo inizi per una particolare stringa, e ciò non avrebbe senso in un campo di tipo numerico o data, inoltre si è visto che è possibile effettuare delle operazioni matematiche sui dati quali somma o media dei valori di un attributo, e queste non avrebbero senso in attributi di tipo non numerico.

Partendo dallo schema generale raffigurato sopra, distinguiamo due possibili schemi per le condizioni sugli attributi: quello in cui la condizione rappresenta una restrizione sulle features di un layer, come nello stato due e sei, e quello in cui la condizione è parte fondamentale della regola, stato otto.

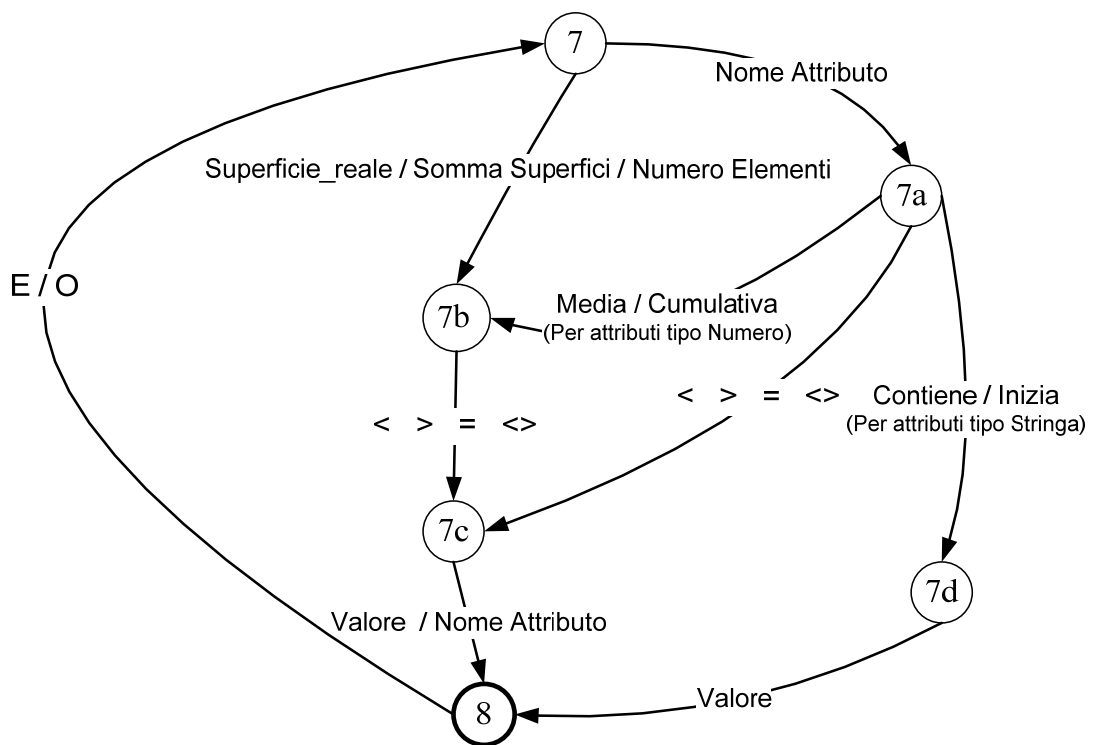
Iniziamo con il trattare il caso in cui si effettuano delle semplici operazioni di restrizione o condizioni del tipo un “attributo deve avere valore minore (maggiore o altro) rispetto ad una costante” oppure che “inizi o contenga una stringa fissata”. Lo schema seguente è una piccola esemplificazione di quanto detto.



Questo schema si pone come sostituto o generalizzazione degli stati due e sei, ovvero tutti gli archi entranti e uscenti a tali stati, ad eccezione dell’arco etichettato Cond.Attributi, è come se confluissero allo stato 6/2 e dallo stato “e” partiranno tutti gli

archi uscenti dai due suddetti stati. Nel caso dello stato sei, che è finale, gli stati 6/2 ed “e” saranno anche essi finali.

Consideriamo adesso il caso dello stato otto in cui, oltre alle condizioni di restrizione sugli attributi, si avranno anche delle condizioni descrittive che si vuole vengano rispettate dalle features. Esempio di tali condizioni può essere volere che in una determinata zona la superficie adibita ad attività commerciali sia minore di un valore prefissato, o che la media di un dato valore rientri in un dato range, o ancora che vi siano un numero di elementi, per esempio tabacchi o farmacie, maggiore o minore di un limite fissato.



Nello schema sopra riportato si possono vedere come alle condizioni di restrizione prima discusse si sono aggiunte delle condizioni matematiche, quali somma e media, e delle operazioni che vanno ad interagire con la componente geografica. Si è detto che la geometria di una feature viene memorizzata in un database relazione, e di solito insieme ad essa viene aggiunto un campo numerico che ne indica la dimensione. Raramente, in particolare nelle “vecchie” mappe, tale valore viene inserito dall’utente, anziché

calcolato ed inserito in automatico dal sistema, con conseguente possibile errore d'inserimento. A tal fine, oltre alla funzione che conta il numero di elementi che rispettano una condizione, si è data la possibilità di calcolare la superficie di una feature direttamente dalla componente geografica, e questo ovviamente lo si può fare sia per calcolare la superficie di una singola feature che per trovare la somma delle superficie di più features.

In tutti i casi, ovvero sia che la condizione sugli attributi sia di tipo restrittiva o descrittiva, è possibile annidare più condizioni con gli operatori "E" e "O".

### **3.3.2 Implementazione automa**

Il nostro Parser allo stato zero per poter iniziare la sua opera di interpretazione della regola necessiterebbe solo della regola stessa e, visto che si è detto voler creare un nuovo shapefile contenente le features che violano tale regola, servirà anche sapere la destinazione dove si vuole che successivamente l'Engine memorizzi tale file e dove il Parser creerà il suo file di log, contenente per ogni regola la sua interpretazione e risultato. Il Parser adesso sarebbe pronto per iniziare l'elaborazione e per crearsi le strutture di appoggio di cui necessita ma, al fine di renderlo quanto più possibile indipendente dalla piattaforma, si è previsto che prenda in input anche la connessione ad un database che userà come appoggio. Per la connessione al database è stato scelto il driver ADODB. Tale scelta perché probabilmente è tra i driver openSource più veloci e perché supporta innumerevoli database, commerciali e non. Al fine di ottenere tutti i parametri necessari, il Parser, verrà avviato da una funzione che provvederà a creare una connessione ADODB con un database utilizzando il driver specifico per quel database, nel nostro caso abbiamo utilizzato un file mdb e quindi driver Microsoft Access, e leggendo le regole da un file o input generico, provvederà a invocare il Parser passando di volta in volta la singola regola.

### 3.3.2.1 **Analizza regola**

L'esecuzione del nostro Parser ha inizio con la chiamata della funzione `Analize` passandole la connessione `ADODB` al database di appoggio e una stringa con la regola ed il percorso da usare per output e file temporanei. `Analize` procederà all'interpretazione della regola avvalendosi di altre funzioni ma, per semplificare la leggibilità e manutenzione del codice, si è lasciato alla funzione `Analize` il compito di invocare tutte le funzioni dell'Engine che procederanno all'esecuzione delle operazioni richieste. La funzione `Analize` procederà con lo scomporre la regola in token ovvero in parole e procederà subito, attraverso la funzione `findLayer`, alla ricerca della prima parola chiave possibile ovvero il nome di un Layer, eventualmente preceduto dalla specifica che deve essere applicata una operazione di restrizione sulla geometria, come ad esempio, nel caso di geometria poligonale, di considerare solo i bordi dell'area. Per procedere a tale operazione `findLayer` si appoggerà ad altre funzioni quali `isLayer` e `isShapeCalculate` che prenderanno in input solo la parola attualmente in esame. Dopo aver individuato il layer in oggetto `findLayer`, prima di ripassare il comando alla funzione `Analize`, vedrà se sono state specificate delle restrizioni sugli attributi in quel layer attraverso la funzione `isAttribute` ed in tal caso procederà alla loro interpretazione avvalendosi dell'ausilio della funzione `getAttribute`.

Individuato il primo layer, `Analize` procederà a verificare se applicare condizioni spaziali su esso con la funzione `isSpatialCondition` ed in tal caso procederà a interpretarla con la funzione `getSpatialCondition` per tenerne traccia in una struttura dati creata ad hoc e richiamare `findLayer` per procedere all'individuazione del secondo layer con il quale subito dopo applicare la condizione spaziale. Questa operazione verrà iterata fino a quando `Analize` non incontrerà la condizione di severità. Il Parser, ed in particolare la funzione `Analize`, è stato ottimizzato applicando le restrizioni sia descrittive che geometriche congiuntamente alla prima esecuzione di una condizione spaziale o descrittiva al fine di ottimizzare l'elaborazione. Per fare ciò utilizza delle strutture dati quali `myLayer` in cui terrà traccia del layer in oggetto, delle eventuali restrizioni geometriche da applicarvi e della condizione spaziale da applicare con il prossimo layer individuato. Raggiunta la condizione di severità si dispone del layer

sorgente, ovvero quello su cui applicare le condizioni spaziali o descrittive, quindi Analize procederà nell'attesa di individuare una condizione o l'altra attraverso le funzioni `isDescriptionCondition` or `isSpatialCondition`. Nel caso di condizione descrittiva demanderà il compito di interpretare l'intera condizione a `getAttributeCondition` per poi procedere all'invocazione della appropriata funzione che andrà ad eseguirla.

Nel caso in cui dopo la condizione di severità vi sia una condizione spaziale, Analize procederà come visto all'inizio, ovvero con l'interpretazione della condizione spaziale seguita dall'individuazione del secondo layer, con eventuali relative restrizioni descrittive e/o geometriche, per poi procedere all'esecuzione della condizione spaziale ed iterare l'operazione nel caso incontrasse un'altra condizione spaziale, ad esempio se si vuole verificare che tutti gli edifici non appartenenti al centro storico contengano sia un posto auto che un'area adibita a verde.

### **3.4 Engine**

L'Engine è quella parte della soluzione a cui è demandato il compito di fungere da intermediario fornendo al Parser una interfaccia per accedere a tutte le funzioni di cui necessita, e di provvedere alla loro esecuzione nella piattaforma sottostante. L'Engine per natura sarà legato all'infrastruttura sottostante, quindi varierà in funzione della piattaforma per le diverse funzioni messe a disposizione e non solo nel nome e uso. Al fine di facilitare l'adattabilità dell'Engine alle diverse piattaforme si è cercato anche qui di suddividere le diverse operazioni in moduli indipendenti. La modularità dell'Engine si può suddividere in tre gruppi, un modulo per la creazione dello `shapeFile`, un modulo per le restrizioni da applicare sulla geometria ed il cuore dell'Engine che è la funzione `fillFeatCls` a cui è demandato il compito di applicare tutte le condizioni spaziali.

### 3.4.1 Creazione nuovo shapeFile

La creazione di un nuovo shapeFile non viene chiamata in causa solo al momento della creazione del file di output ma anche durante l'esecuzione della regola per la creazione di file temporanei. La funzione `createFeatCls`, incaricata di tale compito, prende in ingresso una stringa contenente il nome del file che andrà a creare e una `FeatureClass` per ereditare coordinate spaziali e attributi. La funzione permette come ingresso altri due input opzionali, il primo è una stringa contenente il percorso dove si vuole che venga creato il file (nel caso in cui non fosse specificato lo si creerà nella stessa locazione della `FeatureClass` passata come parametro), il secondo sarà una stringa indicante il tipo di geometria che dovrà contenere (anche in questo caso se non specificato si presupporrà sia la stessa geometria della `FeatureClass` passata).

La funzione procede con la creazione della lista degli attributi del nuovo file, modificando se specificato il campo indicante la geometria. Successivamente essa procede alla creazione fisica del file, sollevando un errore nel caso non dovesse essere in grado di completare l'operazione. Generalmente la causa della non riuscita di tale operazione è il non aver specificato una locazione di destinazione o il non disporre dei diritti sulla locazione contenente la `FeatureClass` sorgente.

Completata l'operazione, e quindi creato lo shapeFile, la funzione restituirà la `FeatureClass` associata ad esso.

### 3.4.2 Restrizioni sulla geometria

Alcune tra le operazioni più delicate e meno immediate da poter essere eseguite con gli strumenti standard messi a disposizione dalla piattaforma sottostante sono sicuramente quelle riguardanti la componente geografica; non solo le operazioni che richiedono di effettuare una restrizione sulla geometria, ad esempio il considerare solo il perimetro di un'area o il punto centrale, ma anche le operazioni necessarie per l'esecuzione di condizioni spaziali. Si pensi ad una regola che consideri come campo di applicazione non la dimensione dell'oggetto stesso ma di un raggio ampio a scelta. Tale raggio verrà chiamato Buffer e si pensi al caso in cui si voglia individuare quelle abitazioni che

ricadono nel raggio di  $x$  metri da una centrale elettrica o autostrada. Tale operazione in alcuni casi richiederà una rimodulazione delle features creandone una di dimensioni pari al Buffer per poi procedere alla normale esecuzione della regola.

La funzione a cui è demandato il compito di applicare le dovute restrizioni alla geometria di una FeatureClass, con all'occorrenza anche le eventuali restrizioni descrittive, è la `applyGeometryRestriction`, che prende come input una struttura dati contenente il layer di origine e l'eventuale restrizione geometrica da applicare; in più prevede la specifica opzionale della locazione dove creare lo shapeFile temporaneo e quella di scegliere se applicare un Buffer, ed in tal caso la sua ampiezza. La funzione `applyGeometryRestriction` procede con la creazione di un nuovo shapeFile in cui andrà ad inserire tutte le features del layer passato come parametro che rispettano le eventuali condizioni descrittive, ed a queste applicherà la dovuta restrizione geometrica avvalendosi della funzione `getGeometry`. Al termine dell'esecuzione, la funzione `applyGeometryRestriction`, restituirà il riferimento alla nuova FeatureClass.

La funzione `getGeometry` prima citata ha il compito, passatogli una geometria ed una condizione di restrizione, di restituire la geometria risultante. Tale funzione viene invocata oltre che dalla `applyGeometryRestriction` anche da altre funzioni per fornire alla `createFeatCls` la giusta geometria del nuovo shapeFile.

### 3.4.3 Condizioni spaziali

Le regole interpretate dal Parser possono avere come oggetto uno o più layers. Nel caso di un solo layer, la funzione `applyGeometryRestriction` assolve pienamente al compito di esecuzione di tale regola con eventuali restrizioni sulla geometria e condizioni o restrizioni descrittive, ma la maggior parte delle regole coinvolge almeno due layers e sono queste quelle più interessanti. Nel caso di più layers il compito di eseguire la dovuta condizione è affidato alla funzione `fillFeatCls`. Essa prende in ingresso due strutture dati contenenti le due FeatureClass in oggetto con relative restrizioni e condizioni, una stringa contenente la locazione dove memorizzare il layer risultante e un attributo booleano per indicare se la condizione geometrica da applicare si vuole che sia rispettata o meno, vedi `contenga` o `non contenga`.



La funzione procede inizialmente con il controllare se vi sono restrizioni geometriche da applicare al secondo layer, ed in tal caso invoca la `applyGeometryRestriction` per procedere alla loro applicazione; dopodiché procede all'applicazione delle eventuali restrizioni descrittive del primo layer, ovvero quello di cui alla fine si vuole avere gli oggetti che violano la regola. Avendo filtrato gli oggetti del primo layer ottenendo solo quelli che rispettano le restrizioni descrittive, si procede per ognuno di essi con l'applicazione dell'eventuale restrizione geometrica, per poi controllare se verifica la condizione spaziale imposta con il secondo layer in accordo con le restrizioni descrittive di quest'ultimo. Al termine dell'esecuzione la funzione restituirà, come nel caso della `applyGeometryRestriction`, il riferimento alla nuova `FeatureClass`.

### **3.5 Portabilità ed espandibilità**

Da un qualsiasi applicativo software non ci si aspetta solo che adempia al proprio scopo, sebbene questa sia la richiesta principale, ma anche che sia affidabile e quindi solido e che sia progettato e scritto in modo da facilitare futuri interventi atti a modificarne il comportamento o ampliarne le funzionalità. Queste caratteristiche, un tempo lasciate in secondo piano, sono divenute fondamentali nello sviluppo di un progetto poiché ci si è resi conto che una buona progettazione, oltre a facilitare lo sviluppo del progetto stesso, permette a lungo andare un ritorno notevole, non solo per eventuali estensioni ma anche per il riuso del codice in altri progetti. L'applicativo sviluppato è stato fatto proprio in questa ottica, dalla struttura totalmente modulare alla indipendenza tra il Parser e l'Engine.

Tutte le funzionalità del Parser sono suddivise in moduli indipendenti a cui è affidato un preciso compito, dall'individuazione delle restrizioni descrittive alle restrizioni geometriche e condizioni spaziali. La funzione `findLayer`, a cui è demandato il compito di individuare i layers presenti in una regola, utilizza una funzione, la `getFeatureClassByName`, la quale ha il compito di restituire il riferimento corretto, in funzione della piattaforma, alla classe rappresentante il layer per poi memorizzarlo nella

struttura dati myLayer da passare all'Engine. Tutta la gestione del database di appoggio, creazione tabelle ed inserimento valori, è affidata a funzioni indipendenti, così come la gestione e rimozione dei file temporanei nell'Engine e il browse per la specifica dei layers potenzialmente coinvolti nelle regole. Tutto ciò al fine di poter subito restringere una anomalia al sistema ma soprattutto per permettere una rapida migrazione di piattaforma permettendo di operare in parallelo nella ricodifica delle funzioni senza richiedere una reingegnerizzazione e limitando al minimo i rischi di malfunzionamento. La struttura modulare adottata si presta bene anche per future espansioni del sistema, al fine di permettere il riconoscimento di regole con nuove restrizioni e condizioni. Basterà agire nel Parser sulle singole specifiche funzioni quali isSpatialCondition e getSpatialCondition o isAttributeCondition e getAttributeCondition o nel caso di una nuova restrizione sulla geometria sulla funzione isShapeCalculate. Tali funzioni in linea di massima possono vedersi come una concatenazione di condizioni dove, in funzione della parola chiave individuata, si applicano una sequenza di operazioni; questo permette una immediata espansione delle funzionalità senza per questo dover modificare il codice già esistente.

Per quanto concerne l'Engine la cosa non cambia, nel caso di una nuova restrizione sulla geometria basterà introdurre nella getGeometry il supporto a tale nuova restrizione senza dover modificare il codice già presente poiché anche qui le possibili restrizioni sono identificate per mezzo di concatenazione di condizioni (istruzione CASE).

Nel caso invece di una nuova condizione spaziale, dopo la sua interpretazione con la funzione getSpatialCondition, se necessario sarà minimo l'intervento da dover apportare sulla funzione fillFeatureCls e nel caso di condizione descrittiva basterà l'interpretazione in getAttributeCondition e solo raramente sarà necessario l'apporto di modifiche in applyGeometryRestriction.

In conclusione si evince che l'apporto di nuove funzionalità al sistema è pienamente supportato poiché previsto già in fase di progettazione e non richiede una rimodulazione del codice o sostanziali modifiche.

## Capitolo 4

### Esempi

#### Introduzione

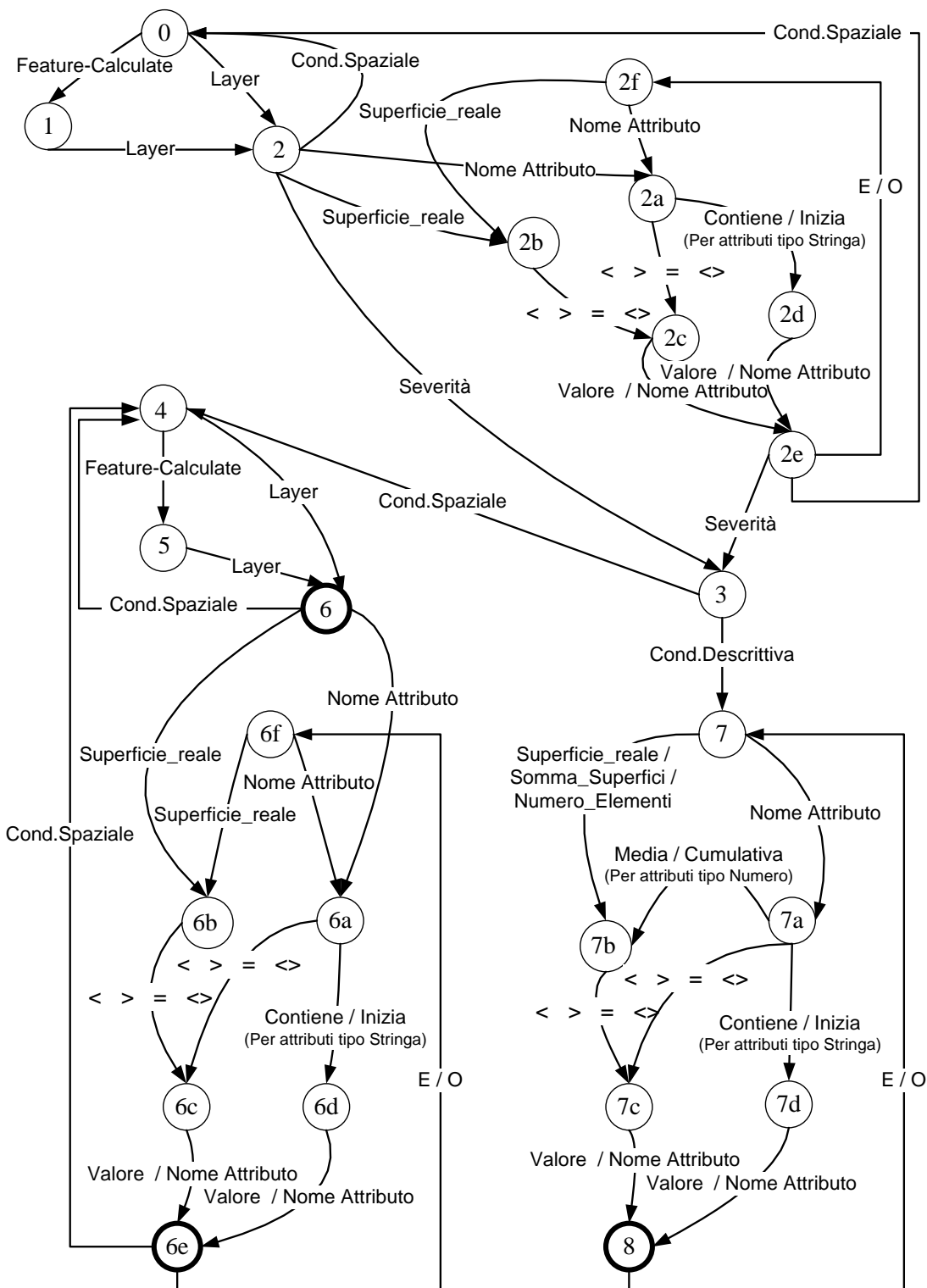
La soluzione presentata ha avuto una implementazione “parziale” dovuta esclusivamente a motivi di tempo. In particolare non è stato curato l’output del file di log contenente l’interpretazione reale della regola (attualmente restituisce una interpretazione non proprio user-friendly), ed alcune funzioni sugli attributi, ad esempio `numero_elementi` e `somme_delle_superfici`, non sono state implementate.

In questo capitolo vedremo, data una regola, la sua interpretazione nell’automa e la sua esecuzione attraverso il Parser e l’Engine.

#### 4.1 Regole con restrizioni descrittive e spaziali

Provvederemo di seguito a mostrare l’iterazione dell’applicativo proposto su alcune regole che coprono le principali casistiche sopra individuate. Le regole comporteranno l’applicazione di condizioni descrittive, quindi che i valori dell’attributo, o attributi, di un layer rispettino un fissato vincolo, e condizioni spaziali, ovvero che gli oggetti di un layer si trovino in una precisa posizione in funzione degli oggetti di altri layer. Sul layer le cui feature dovranno rispettare le condizioni imposte, che chiameremo di origine, saranno prima applicate le restrizioni descrittive e subito dopo le spaziali, al fine di limitare il controllo ad un sottoinsieme delle features.

Di seguito riportiamo lo schema espanso dell’automa al fine di aiutare la successiva interpretazione delle regole presentate.

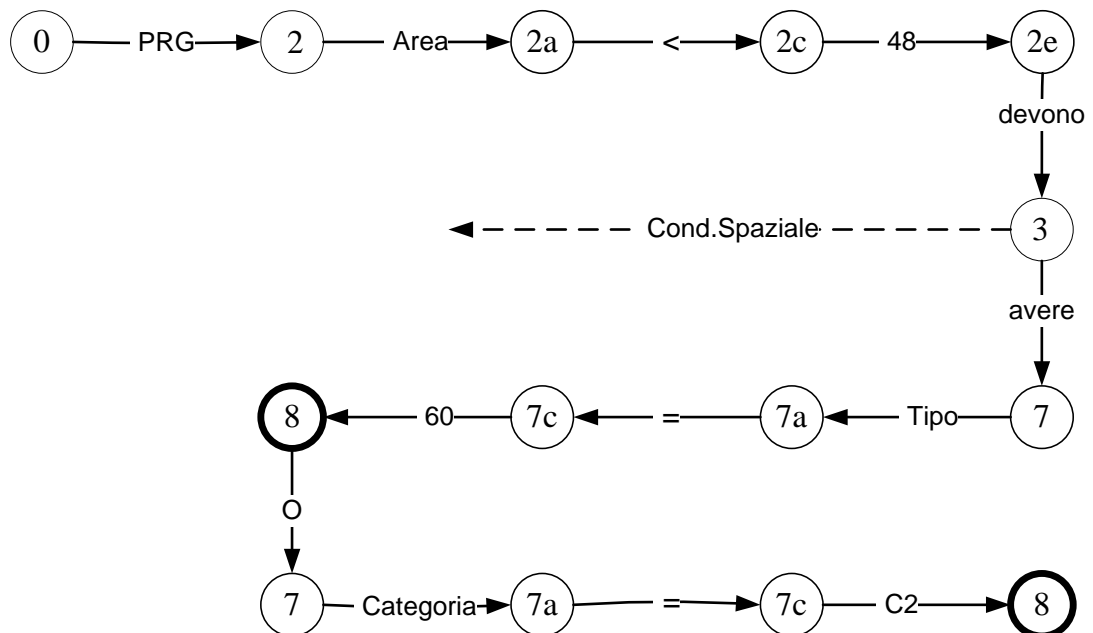


#### 4.1.1 Condizione descrittiva su layer con restrizione descrittiva

Proviamo a vedere come si comporta il nostro automa e di riflesso il Parser nella interpretazione della seguente regola:

Le aree di PRG con AREA > 48 mq devono avere TIPO = 60 o CATEGORIA = C2

L'interpretazione della regola nell'automa avviene tramite la transizione attraverso i seguenti stati con le relative parole chiavi individuate



Il Parser come prima operazione procede a suddividere la regola passatagli come stringa; per far ciò la regola viene passata ad una classe, chiamata Tokenizer, che provvede alla restituzione delle singole parole, che chiameremo Token, in funzione del carattere di delimitazione che si specifica; di default il carattere di delimitazione sarà lo spazio. A questo punto il Parser procede con la ricerca del nome di un layer o di una operazione da effettuare sulla geometria così come da automa allo stato zero. La ricerca viene effettuata dalla funzione findLayer, che scorre i vari Token fino a quando non incontra il nome di un layer. Per riconoscere il nome del layer, findLayer procederà alla

ricerca del Token in un database di appoggio in cui sono stati precedentemente memorizzati i nomi di tutti i layer disponibili, o per lo meno quelli a cui ci si riferirà, e l'etichetta, ovvero il nome con il quale si farà loro riferimento nelle regole. L'etichetta di un layer viene inserita nel database dal sistema, nel nostro caso in modo automatico, sostituendo l'eventuale spazio nei nomi dei layer con \_ (underscore). Durante la ricerca del layer la funzione findLayer controllerà la presenza di parole chiave indicanti un'operazione da effettuare sulla geometria (FeatureCalculate), ed eventualmente ne terrà traccia in una struttura creata ad hoc, che conterrà anche il riferimento al layer ed eventuale condizione spaziale. Individuato il layer di riferimento, procederà al suo caricamento e memorizzazione del riferimento alla classe nella struttura sopra citata ed all'inserimento dei nomi degli attributi nel database di appoggio con il loro tipo, ovvero se numero o testo. A questo punto, prima di tornare il controllo alla funzione che l'ha invocata, nel nostro caso Analize, findLayer invoca la funzione getAttributeCondition per individuare la presenza di restrizioni descrittive.

La funzione getAttributeCondition procede con la scansione dei Token fino a quando incontra una condizione spaziale o una condizione di severità, e solo allora termina per restituire il controllo alla funzione padre. GetAttributeCondition controlla per ogni Token se questo è presente nel database di appoggio e quindi se è il nome di un attributo di un layer presente nella regola; in tal caso procede all'interpretazione della condizione, controllando che sia in accordo con il tipo dell'attributo già specificato nella tabella di appoggio, e alla sua memorizzazione nella classe attributeCondition. La funzione getAttributeCondition riconoscerà l'operatore di maggiore e il valore a seguire, e quindi, non essendo seguiti da nessun operatore di giunzione "e" od "o" ed incontrando la condizione di severità "devono", inserirà l'intera condizione nella struttura e restituirà il controllo alla funzione Analize.

La classe attributeCondition fornisce un'interfaccia per memorizzare le restrizioni descrittive legate ad un layer e per ottenere all'occorrenza tutte le restrizioni presenti per un dato layer.

Avendo riconosciuto la prima parte della regola ovvero "Le aree di PRG con AREA > 48 mq devono" Analize procede con l'analisi del Token successivo per controllare se si tratta di una condizione spaziale o descrittiva. Incontra "avere" quindi procede nel ramo

cond.Descrittiva ed invoca nuovamente la condizione `getAttributeCondition` per procedere all'estrapolazione della condizione descrittiva. Si vuole far notare che `Analize`, nel caso la regola avesse previsto "... non devono avere ...", avrebbe riconosciuto il termine "non" e lo avrebbe passato come parametro alla funzione `getAttributeCondition` al fine di tenerne conto nella gestione dell'operatore; nel nostro caso infatti la condizione "Tipo = 60" diventerà nell'applicativo "Tipo <> 60", poiché si cercano quelle feature che non rispettano tale condizione; nel caso ci fosse stato il non invece la condizione non avrebbe subito mutamenti.

Dopo aver individuato le due condizioni descrittive, essendo giunto alla fine della regola, `getAttributeCondition` terminerà restituendo il controllo ad `Analize`, che provvederà a chiamare la funzione dell'Engine `applyGeometryRestriction` passandogli la struttura dati contenente il riferimento al layer.

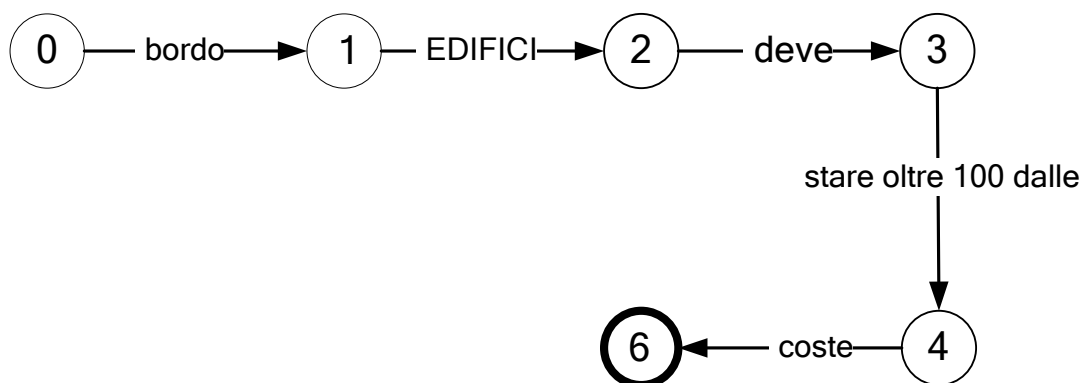
La funzione `applyGeometryRestriction` inizia con l'inserimento nel file di log del nome del layer, le eventuali restrizioni sulla geometria, e le restrizioni e condizioni descrittive. Fatto ciò procede con la creazione del nuovo shapefile e con la ricerca di tutte le feature che soddisfano le condizioni descrittive specificate. `ApplyGeometryRestriction` copierà le feature trovate nel nuovo shapefile applicando, se specificato, la restrizione sulla geometria invocando `getGeometry`. Al termine `applyGeometryRestriction` restituirà la struttura contenente il riferimento alla classe del nuovo shapefile, che nel nostro caso sarà aggiunto dalla funzione `Analize` alla lista dei layer caricati in `ArcInfo` per permettere all'utente una rapida consultazione.

#### 4.1.2 Condizione spaziale su layer con restrizione sulla geometria

Dopo aver visto il comportamento dell'automa su una regola con condizione descrittiva, vediamo il suo comportamento su una regola con condizione spaziale e restrizione alla geometria del layer di origine:

Il bordo degli EDIFICI deve stare oltre 100 m dalle coste

L'interpretazione della regola nell'automa avviene tramite la transizione attraverso i seguenti stati con relative parole chiavi individuate



Come nel caso precedente, il Parser procede con la suddivisione in Token della regola e con l'invocazione della funzione `findLayer`; questa fino a quando non incontra il nome di un layer, controllerà se il Token in esame indica una restrizione da applicare alla geometria invocando la funzione `isShapeCalculate`, che restituisce una stringa vuota o un stringa indicante la restrizione da applicare e di cui ne verrà tenuta traccia nella struttura dati `firstLayer` di tipo `myLayer` contenente anche il riferimento al layer.

Il controllo, individuato il layer e non essendoci restrizioni descrittive, torna alla funzione `Analize` che procede alla ricerca di una condizione descrittiva o spaziale.

Individuata la condizione spaziale, `Analize` invoca la funzione `getSpatialCondition` che individua la parola chiave "stare oltre", procede alla ricerca di un valore o di un nome d'attributo contenente la dimensione del buffer da applicare, per poi concludere con l'individuazione della parola chiave "da". Questa condizione spaziale implica una condizione di non intersezione tra due layer, dove alla geometria del primo si dovrà applicare un buffer. La condizione spaziale di intersezione verrà memorizzata nella struttura `firstLayer` mentre la condizione di buffer in un'altra struttura chiamata `bufferCondition` di tipo `myBuffer`.

Completata la condizione spaziale il controllo torna alla funzione `Analize`, la quale procede con l'individuazione del secondo layer, sempre tramite la funzione `findLayer`.

Individuato il secondo Layer `Analize`, non essendo state trovate restrizioni sulla geometria da `findLayer`, legge la restrizione riguardante il buffer memorizzata nella



---

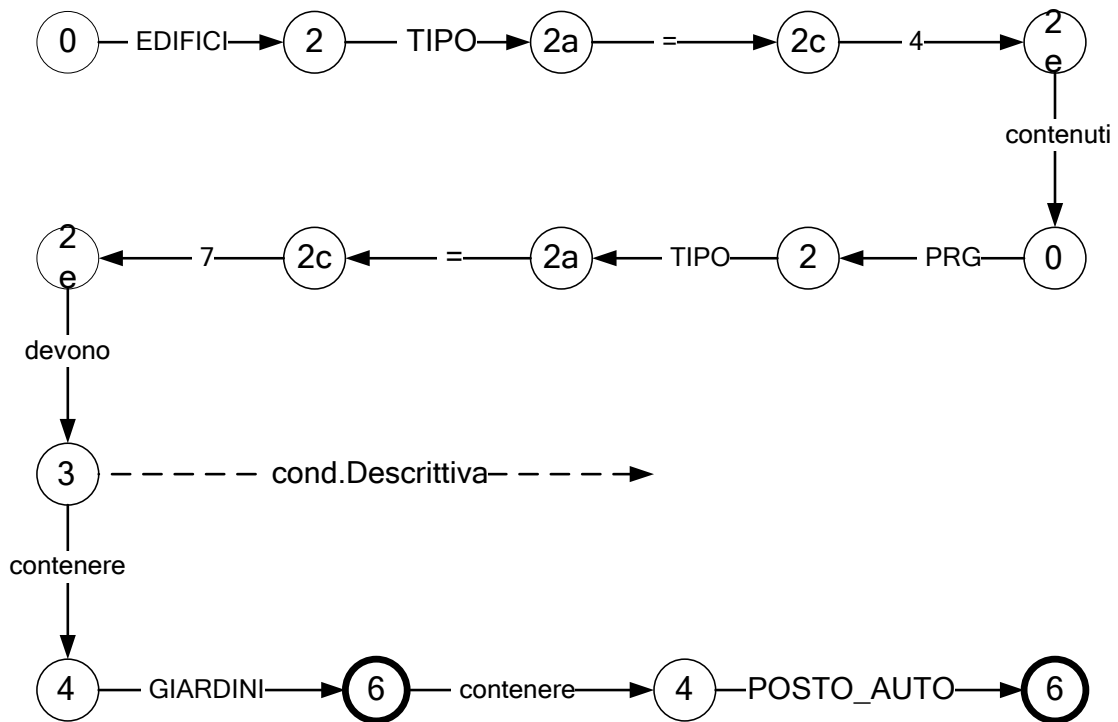
struttura `bufferCondition` e la memorizza nella struttura `secondLayer` di tipo `myLayer` contenente il riferimento al secondo layer. L'esecuzione procede con l'invocazione da parte di `Analyze` della funzione `fillFeatCls`.

`FillFeatCls` procede subito con l'applicazione, se presente, della restrizione geometrica sul secondo layer, che da ora in avanti chiameremo di restrizione e con delle restrizioni descrittive al primo layer che chiameremo di origine. Fatto ciò `fillFeatCls` continua con la scrittura in un file di log delle operazioni che si appresta ad effettuare ed alla creazione dello shapefile che conterrà le feature trovate. L'esecuzione di `fillFeatCls` termina con l'applicazione della condizione spaziale di intersezione tra i due layer inserendo le feature individuate nel nuovo shapefile e restituendo il riferimento al nuovo shapefile alla funzione `Analyze`, che nel nostro caso procederà al suo inserimento tra i layer caricati in `ArcInfo`.

#### **4.1.3 Condizione spaziale doppia su layer con restrizione descrittiva e spaziale**

Per concludere la casistica delle possibili combinazioni procediamo ad una rapida analisi della seguente regola:

Gli EDIFICI di TIPO=4 contenuti in PRG TIPO=7 devono contenere le aree di GIARDINI e contenere POSTO\_AUTO



Il Parser, dopo aver individuato il primo layer con relativa restrizione descrittiva, procede alla ricerca della condizione di severità o, come nel nostro caso, spaziale; avendo individuato una condizione spaziale, procederà, come visto nell'esempio precedente, al riconoscimento della condizione attraverso la funzione `getSpatialCondition` per poi procedere all'individuazione del secondo layer con relativa restrizione descrittiva e all'applicazione della condizione spaziale attraverso la funzione `fillFeatCls`. Il layer restituito dalla funzione `fillFeatCls` sarà il nuovo layer di origine su cui si andranno ad applicare le successive condizioni descrittive e spaziali. L'esecuzione del Parser prosegue con l'interpretazione del termine "devono" per poi incontrare la condizione spaziale. Come nel caso precedente Analize procede con l'interpretazione della condizione, l'individuazione del layer e con l'esecuzione della condizione spaziale. Non essendo giunti al termine della regola, Analize procede con l'interpretazione ed individua una condizione spaziale che sarà applicata sul layer appena restituito da `fillFeatCls` ed al layer successivo presente nella regola. Al termine dell'esecuzione di `fillFeatCls`, essendo giunto alla fine della regola, Analize procederà con l'inserimento in ArcInfo del layer restituito da `fillFeatCls` nell'ultima esecuzione.

## **Capitolo 5**

### **Conclusioni**

Il lavoro si colloca nell'ambito degli strumenti di analisi con l'obiettivo di estendere le funzionalità fornite da piattaforma GIS e di semplificarne l'utilizzo.

Il sistema si propone di soddisfare le esigenze dell'operatore sia in fase di analisi di una mappa sia in fase di collaudo attraverso l'interpretazione e la verifica di regole. Si è fornito quindi uno strumento che permette di specificare delle regole in modo semplice e di applicarle in sequenza senza alcun intervento da parte dell'operatore.

L'implementazione è stata effettuata momentaneamente solo per piattaforma ESRI e permette di creare regole con condizioni e restrizioni sia spaziali che sul contenuto degli attributi.

Il sistema si suddivide in due componenti, il Parser, scritto in Visual Basic, e l'Engine, a cui è demandato il compito di invocare le primitive della piattaforma al fine di eseguire le operazioni necessarie al soddisfacimento della regola. I due moduli comunicano attraverso l'invocazione da parte del Parser delle funzioni appropriate e fornendo i riferimenti ai dati necessari con la specifica attraverso parole chiave dell'operazione da effettuare.

Il sistema è stato testato su mappe con un numero di oggetti da tremila a tredicimila, applicando dieci regole scritte in modo da coprire la possibile casistica, ovvero con restrizioni descrittive sia su attributi di tipo numero che di tipo testo, restrizioni sulle geometria e tutte le possibili condizioni descrittive e spaziali.

L'esecuzione delle regole ha impiegato un tempo variabile tra il minuto ed i quattro minuti, ed ha sempre restituito sempre il valore aspettato.

## Appendice A

### Codice Parser Manager

```
Public Function Analize(ByVal Regola As String, ByVal dbConnection As  
ADODB.Connection, tempLocation As String) As Boolean
```

```
    Dim primoLayer As myLayer
```

```
    Dim secondoLayer As myLayer
```

```
    Dim negazione As Boolean
```

```
    Dim token As Tokenizer
```

```
    Dim freeAttributeCondition As attributeCondition
```

*'----- freeAttributeCondition conterrà tutte le restrizioni e condizioni descrittive  
presenti per poi assegnarle al corretto layer. Questa struttura permette di poter  
accedere rapidamente alle restrizioni di un dato layer con esattezza nel momento in cui  
si voglia-----*

```
    Set freeAttributeCondition = New attributeCondition
```

```
    freeAttributeCondition.initialize
```

```
    Set token = New Tokenizer
```

```
    token.stringa = Normalize(Regola)
```

```
    token.nextToken
```

*‘----- si provvede tramite la funzione findLayer a caricare la classe del layer specificato nella regola con annesse restrizioni sulla geometria e sugli attributi nella struttura primoLayer di tipo myLayer -----’*

```
primoLayer = findLayer(token, dbConnection, freeAttributeCondition)
```

```
Dim bufferCondition As myBuffer
```

```
If (token.Parola = "non") Then
```

```
    negazione = True
```

```
    token.nextToken
```

```
Else
```

```
    negazione = False
```

```
End If
```

*‘----- findLayer è terminata perché ha incontrato una condizione spaziale o una condizione di severità. Si procede con la ricerca di eventuali condizioni spaziali da applicare per ottenere il layer di origine -----’*

```
While (token.moreToken And isSpatialCondition(token.Parola) <> "")
```

```
    getSpatialCondition token, primoLayer, negazione, bufferCondition, dbConnection
```

```
    secondoLayer = findLayer(token, dbConnection, freeAttributeCondition)
```

*‘----- nel caso si dovesse applicare al secondo layer sia una restrizione geometrica che un buffer, si procede subito alla loro esecuzione -----’*

```
If (secondoLayer.geometryRestriction <> "" And  
bufferCondition.geometryRestriction <> "") Then
```

*‘----- si rimuovono dal database di appoggio gli attributi del secondo layer per poi inserirli nuovamente con riferimento al nuovo layer. Non si fa un aggiornamento poiché il nuovo layer potrebbe contenere più attributi di quello di origine -----*

```

RemoveProperty dbConnection, secondoLayer.FeatCls
secondoLayer = applyGeometryRestriction(secondoLayer,
bufferCondition.geometryRestriction, bufferCondition.valueGeometryRestriction,
freeAttributeCondition)
InsertLayer dbConnection, secondoLayer.FeatCls
InsertProperty dbConnection, secondoLayer.FeatCls

```

*‘----- nel caso non sia presente una restrizione sulla geometria e si debba applicare un buffer si procede alla sua memorizzazione nella struttura myLayer ovvero tra le restrizioni sulla geometria -----*

```

ElseIf (secondoLayer.geometryRestriction = "" And
bufferCondition.geometryRestriction <> "") Then
secondoLayer.geometryRestriction = bufferCondition.geometryRestriction
secondoLayer.valueGeometryRestriction =
bufferCondition.valueGeometryRestriction
End If

```

*‘----- individuato il layer di origine ed il layer di restrizione si procede all’esecuzione della condizione spaziale con l’invocazione della funzione fillFeatCls ----  
-----*

```

RemoveProperty dbConnection, primoLayer.FeatCls
primoLayer = fillFeatCls(primoLayer, secondoLayer, negazione, tempLocation,
freeAttributeCondition)
InsertLayer dbConnection, primoLayer.FeatCls
InsertProperty dbConnection, primoLayer.FeatCls

```

```

negazione = False
If (token.Parola = "non") Then
    negazione = True
    token.nextToken
End If

```

*‘----- l’ultima funzione eseguita sul token stata findLayer quindi ci troviamo ad una parola chiave che sarà una condizione di severità o condizione spaziale e questa deciderà se rimanere in ciclo o uscire -----*

```

Wend

```

```

While (token.moreToken And isSeverityCondition(token.Parola) = "")
    token.nextToken
Wend
token.nextToken
If (isDescriptionCondition(token.Parola) <> "") Then
    token.nextToken

```

*‘----- si procede con l’interpretazione delle condizioni descrittive da applicare al layer -----*

```

    getAttributeCondition(token, dbConnection, Not negazione,
freeAttributeCondition)
    RemoveProperty dbConnection, primoLayer.FeatCls
    primoLayer = applyGeometryRestriction(primoLayer, tempLocation,
freeAttributeCondition)
    InsertLayer dbConnection, primoLayer.FeatCls
    InsertProperty dbConnection, primoLayer.FeatCls
    negazione = False
Else

```



*'----- in questo caso significa che non si vuole applicare una condizione descrittiva bensì spaziale si procede come nel primo caso iterando fino a quando non si è giunti al termine della regola -----'*

```

While (token.moreToken)
  If (token.Parola = "non") Then
    negazione = True
    token.nextToken
  End If
  getSpatialCondition token, primoLayer, negazione, bufferCondition,
dbConnection
  secondoLayer = findLayer(token, dbConnection, freeAttributeCondition)
  If (secondoLayer.geometryRestriction <> "" And
bufferCondition.geometryRestriction <> "") Then
    secondoLayer = applyGeometryRestriction(secondoLayer,
bufferCondition.geometryRestriction, bufferCondition.valueGeometryRestriction,
freeAttributeCondition)
    InsertLayer dbConnection, secondoLayer.FeatCls
    InsertProperty dbConnection, secondoLayer.FeatCls
  ElseIf (secondoLayer.geometryRestriction = "" And
bufferCondition.geometryRestriction <> "") Then
    secondoLayer.geometryRestriction = bufferCondition.geometryRestriction
    secondoLayer.valueGeometryRestriction =
bufferCondition.valueGeometryRestriction
  End If
  RemoveProperty dbConnection, primoLayer.FeatCls
  primoLayer = fillFeatCls(primoLayer, secondoLayer, negazione, tempLocation,
freeAttributeCondition)
  InsertLayer dbConnection, primoLayer.FeatCls
  InsertProperty dbConnection, primoLayer.FeatCls
  negazione = False

```

Wend

End If

*‘----- si inserisce lo shapefile risultante tra i layer caricati in ArcMap -----’*

Dim pDoc As IMxDocument

Dim Display As IFeatureLayer

Dim pMap As IMap

Set pDoc = Application.Document

If TypeOf pDoc.ActiveView Is IMap Then

    Set pMap = pDoc.FocusMap

    Set Display = New FeatureLayer

    Set Display.featureClass = primoLayer.FeatCls

    Display.name = primoLayer.FileName

    pMap.AddLayer Display

End If

End Function

## Appendice B

### Codice Engine Manager

```
Public Function fillFeatCls(ByRef myLayerSource As myLayer, ByRef  
myLayerRestrict As myLayer, ByVal neg As Boolean, tempLocation As String) As  
myLayer
```

```
    Dim newFeatcur As IFeatureCursor  
    Dim newFeatBuff As IFeatureBuffer  
    Dim sourceFeatCur As IFeatureCursor  
    Dim sourceFeat As IFeature  
    Dim restrictFeatCur As IFeatureCursor  
    Dim restrictFeat As IFeature  
    Dim newGeometry As IGeometry  
    Dim tempQueryFilter As IQueryFilter  
    Dim selectFeaturesRestrict As ISelection
```

*‘----- si controlla se si devono applicare delle restrizioni sulla geometria del  
secondo layer ed in tal caso si applicano -----*

```
    If (myLayerRestrict.geometryRestriction <> "") Then  
        myLayerRestrict = applyGeometryRestriction(myLayerRestrict, tempLocation)  
    End If
```

*‘----- Applico le restrizioni descrittive al primo layer -----’*

```

Set tempQueryFilter = New queryFilter
tempQueryFilter.WhereClause =
Main.freeAttributeCondition.getCondition(myLayerSource.FeatCls.AliasName)
Set sourceFeatCur = myLayerSource.FeatCls.Search(tempQueryFilter, False)
Set sourceFeat = sourceFeatCur.NextFeature

```

*‘----- Inserisco nel file di log le operazioni che ho effettuato -----’*

```

With Main.TextStream
  .Write (" " & myLayerSource.geometryRestriction)
  .Write (" " & myLayerSource.FeatCls.AliasName)
  .Write (" " & sourceAttributeFilter.WhereClause)
  .Write (" " & neg)
  .Write (" " & myLayerSource.spFilter.SpatialRel)
  .Write (" " & myLayerSource.spFilter.SpatialRelDescription)
  .Write (" " & myLayerRestrict.geometryRestriction)
  .Write (" " & myLayerRestrict.FeatCls.AliasName)
  .Write (" " & myLayerSource.spFilter.WhereClause)
End With

```

*‘----- creo il nuovo shapefile passandogli il nome, composto dai nomi dei due layers, la posizione dove crearlo e la geometria -----’*

```

fillFeatCls.FileName = myLayerSource.FileName & myLayerRestrict.FileName
Set fillFeatCls.FeatCls = createFeatCls(myLayerSource.FeatCls,
fillFeatCls.FileName, tempLocation, getGeometry(sourceFeat.Shape,
myLayerSource.geometryRestriction).GeometryType)
Set fillFeatCls.spFilter = New spatialFilter

```

*'----- preparo la struttura per l'inserimento delle feature nella nuova featureClass --  
-----*

```
Set newFeatcur = fillFeatCls.FeatCls.Insert(True)
Set newFeatBuff = fillFeatCls.FeatCls.CreateFeatureBuffer
```

*'----- Applico le restrizioni descrittive al secondo layer al fine di ridurre il numero di  
feature su cui applicare la condizione spaziale -----*

```
tempQueryFilter.WhereClause =
Main.freeAttributeCondition.getCondition(myLayerRestrict.FeatCls.AliasName)
Set selectFeaturesRestrict = myLayerRestrict.FeatCls.Select(tempQueryFilter,
esriSelectionTypeSnapshot, esriSelectionOptionNormal,
myLayerRestrict.FeatCls.FeatureDataset.Workspace)
```

```
While Not sourceFeat Is Nothing
```

*'----- Estraggo dal primo layer la giusta geometria da applicare ed effettuo la  
ricerca, uso spatial filter del primo layer per ricercare in layerRestrict poiché qui vi è  
memorizzata la condizione spaziale -----*

```
Set newGeometry = getGeometry(sourceFeat.Shape,
myLayerSource.geometryRestriction)
Set myLayerSource.spFilter.Geometry = newGeometry
Set restrictFeatCur = selectFeaturesRestrict.Search(myLayerSource.spFilter, False)
Set restrictFeat = restrictFeatCur.NextFeature
```

```
On Error GoTo Log0
```

*'----- inserisco la feature nella nuova featureClass. Es: non devono intersecare, inserisco la feature sorgente se trovo una feature intersecata -----'*

```

If Not restrictFeat Is Nothing And neg = True Then
    Set newFeatBuff.Shape = newGeometry
    For count = 0 To sourceFeat.Fields.FieldCount - 1
        If UCase(sourceFeat.Fields.Field(count).name) <> "SHAPE" And
newFeatBuff.Fields.Field(count).Editable Then
            newFeatBuff.Value(count) = sourceFeat.Value(count)
        End If
    Next count
    newFeatcur.InsertFeature newFeatBuff

```

*'----- inserisco la feature nella nuova featureClass. Es: devono intersecare, inserisco la feature sorgente se non trovo una feature intersecata -----'*

```

ElseIf restrictFeat Is Nothing And neg = False Then
    Set newFeatBuff.Shape = newGeometry
    For count = 0 To sourceFeat.Fields.FieldCount - 1
        If UCase(sourceFeat.Fields.Field(count).name) <> "SHAPE" And
newFeatBuff.Fields.Field(count).Editable Then
            newFeatBuff.Value(count) = sourceFeat.Value(count)
        End If
    Next count
    newFeatcur.InsertFeature newFeatBuff
End If

```

```

Set sourceFeat = sourceFeatCur.NextFeature
Wend

```

```

newFeatcur.Flush

```

*'----- rimuovo i file temporanei -----'*

RemoveFile tempLocation, myLayerSource.FileName

RemoveFile tempLocation, myLayerRestrict.FileName

Log0:

With Main.TextStream

.Write (" " & restrictFeat.OID)

End With

End Function

## Glossario

**GIS:** Sistema Informativo Geografico (o Territoriale), è un sistema per la gestione, l'analisi e la visualizzazione di informazioni con contenuto geografico/spaziale.

**Layer:** Classe di oggetti in un sistema GIS, la cui struttura dati è organizzata secondo layer o livelli tematici. Un layer può essere più o meno generale a seconda del modello adottato. Ad esempio l'entità delle Strade può essere strutturato in vari layer quali: rete autostradale, rete viaria principale e secondaria, ponti, tunnel, etc, al fine di permettere una migliore gestione e più chiara visualizzazione, ma può anche essere strutturata in un unico layer.

**Mappa georeferenziata:** La mappa è una rappresentazione grafica bidimensionale e in scala di un'area geografica. Può essere il risultato di un processo di fotointerpretazione o il risultato di un'elaborazione GIS.

**Stereorestitutore:** strumento per la costruzione di una mappa in formato cartaceo o numerico. Lo strumento utilizza una strumentazione ottico meccanica di precisione e permette, tramite il posizionamento di una coppia di foto aeree e un meccanismo di triangolazione, di conoscere le coordinate degli oggetti rappresentati; permette inoltre la ricostruzione del modello stereoscopico per facilitare il riconoscimento degli oggetti e valutarne la quota. L'uscita dello strumento è una carta o un archivio numerico.





**Aerofotogrammetria:** La Aerofotogrammetria è una tecnica di rilevazione della posizione di punti mediante l'utilizzo di immagini fotografiche stereoscopiche del terreno riprese da aeromobile.

Mediante lo stereorestitutore è possibile effettuare il rilievo di dettaglio a partire dal modello opportunamente orientato, essendo possibile, grazie ai punti di inquadramento, calcolare la scala e l'orientamento, rispetto al riferimento cartografico scelto, del modello stesso.

**Topologia:** branca della matematica, utilizzata nel mondo GIS, che studia le relazioni spaziali tra oggetti che non dipendono dalla forma degli oggetti.

**Query:** : interrogazione di un database in modo da ottenere un sottoinsieme dei dati contenuti nello stesso DB o una loro aggregazione. Solitamente una query viene passata al DBMS in linguaggio SQL. Nel mondo GIS una query svolge le stesse funzioni svolte nei DBMS non geografici, con l'aggiunta di condizioni geografiche al criterio di selezione.

**Feature:** istanza di una classe (o layer) GIS. Una Feature rappresenta uno specifico oggetto di un layer. Un layer è formato da più feature. Esempio: Layer "Strade"; Feature "Via E. Zerboglio", "Via Randaccio", "Via dei Larici", ...

## Bibliografia

- 1 Intesa GIS  
[http://www.intesagis.it/info\\_intesagis.asp](http://www.intesagis.it/info_intesagis.asp)
- 2 Standard ISO/TC211  
<http://www.isotc211.org/>
- 3 Standard CEN/TC287  
<http://www2.nen.nl/nen/servlet/dispatcher.Dispatcher?id=204514>
- 4 Standard INSPIRE Drafting Teams  
<http://inspire.jrc.it/>
- 5 Standard OGC  
<http://www.opengeospatial.org/>
- 6 Gruppo di lavoro sugli standard del settore dell'Informazione Geografica:  
Automated Mapping Facilities Management  
<http://www.amfm.it/attivita/gdl/standard/>
- 7 Cartografia: appunti di Maurizio Trevisani, membro del Sistema Informativo  
Regionale dell'Ambiente
- 8 Intergraph DMC<sup>®</sup> (Digital Mapping Camera)  
[http://www.intergraph.com/resource\\_files/literature/IMGS208/IMGS208\\_screen.pdf](http://www.intergraph.com/resource_files/literature/IMGS208/IMGS208_screen.pdf)
- 9 ESRI GeoDatabase  
<http://www.esri.com/software/arcgis/geodatabase/index.html>
- 10 Intergraph GeoMedia Curator  
[http://www.intergraph.com/resource\\_files/literature/MI001A0/MI001A0\\_print.pdf](http://www.intergraph.com/resource_files/literature/MI001A0/MI001A0_print.pdf)
- 11 Autodesk Civil 3D e Land Desktop  
[http://images.autodesk.com/gost/files/Whitepaper\\_C3D\\_2005\\_vs\\_LDT\\_2005FINAL.pdf](http://images.autodesk.com/gost/files/Whitepaper_C3D_2005_vs_LDT_2005FINAL.pdf)
- 12 Autodesk Design Server  
[http://images.autodesk.com/adsk/files/2203272\\_GISDesSvr\\_WP\\_Oracle8i.pdf](http://images.autodesk.com/adsk/files/2203272_GISDesSvr_WP_Oracle8i.pdf)
- 13 Autodesk, removing obstacles between CAD and GIS  
[http://images.autodesk.com/adsk/files/3364052\\_CriticalTools0.pdf](http://images.autodesk.com/adsk/files/3364052_CriticalTools0.pdf)

14 Autodesk, MapGuide 6 e ArcIMS 4

[http://images.autodesk.com/adsk/files/4349822\\_3429771\\_Autodesk\\_MapGuide\\_6\\_and\\_ESRI\\_s\\_ArcIMS\\_4.pdf](http://images.autodesk.com/adsk/files/4349822_3429771_Autodesk_MapGuide_6_and_ESRI_s_ArcIMS_4.pdf)

15 Specifica classi piattaforma ESRI

<http://edndoc.esri.com/arcobjects/9.0/default.asp>