Consultas espaciales en una arquitectura de componentes GIS

Tesis profesional presentada por Olivia Loranca Mateos

Maestría en Ciencias con Especialidad en Ingeniería en Sistemas Computacionales. Departamento de Ingeniería en Sistemas Computacionales. Escuela de Ingeniería, Universidad de las Américas Puebla.

Jurado Calificador

Presidente: Dr. Gerardo Ayala San Martín
Vocal y Director: Dr. David Ricardo Sol Martínez
Secretario: Dr. J. Alfredo Sánchez Huitrón

Cholula, Puebla, México a 30 de octubre de 2000.

Introducción

En materia computacional, cuando hablamos de consultas, pensamos en la clásica consulta a la base de datos, haciendo uso de select en SQL. Por ejemplo, selecciona todas las ciudades que fueron fundadas en 1531. En una base de datos con las principales ciudades de la República Mexicana y los años en las que fueron fundadas, esta selección nos reportaría a la ciudad de Puebla.

Pero, ¿qué sucede cuando la consulta que esperamos realizar es sobre la relación geográfica que existe entre los datos?. Por ejemplo queremos seleccionar los ríos que cruzan una comunidad.

En este caso, ¿Cómo se realiza la consulta?, ¿Cuál sería una forma sencilla para capturar el requerimiento del usuario?. Además, considerando que es común este tipo de preguntas, ¿Es necesario contar con un sistema especializado en este tipo de datos?, ya que implica contar con equipos de gran capacidad de almacenamiento y velocidad, o ¿Podemos tener acceso remoto a ellos?.

El objetivo primordial del Componente de Consultas Espaciales del sistema UDLASIG, es el de proporcionar una herramienta que facilite, en forma remota, la captura, cálculo y recuperación de los resultados de consultas espaciales, para su adecuada representación gráfica. Perteneciendo este componente a una Arquitectura de Componentes GIS, los cuales forman el sistema.

3.1 Introducción

En éste capítulo describimos el formato OpenGis *[Beddoe 99],* la importancia que tiene es porque se utiliza para comunicar a los componentes. Este formato es un modelo, que espera estandarizar las geometrías utilizadas en SIG′s, por lo que se explica cada una de las características de sus clases.

También se plantea en que consiste el Sistema de Referencia Espacial, el cual esta directamente relacionado con la jerarquía de clases del Modelo de Gemetrías de OpenGis.

Un estándar no menos importante, que utilizamos para la implementación de este prototipo, es el Lenguaje de Consultas Estructurado (SQL92), por lo que se especifica la forma como se almacenan los datos. Existen otros estándares como SQL/MM, el cual planea manejar en una forma más apropiada o natural los objetos geográficos.

3.2 Modelo de Objetos Geométricos

El Modelo de Objetos Geométricos es propuesto por el Consorcio OpenGIS, en el cual se usa una notación OMT. En este modelo la clase Geometría tiene como subclases a las clases Punto, Curva, Superficie y Colección Geométrica. Asocia a cada objeto geométrico un Sistema de Referencia Espacial, que describe la coordenada espacial en el que se define el objeto geométrico. Las Clases nombradas MultiPunto, MultiPolilínea y MultiPolígono corresponden a colecciones de Puntos, Polilíneas y Polígonos respectivamente. MultiCurva y MultiSuperficie son introducidas como superclases abstractas, las cuales generalizan la colección que contienen Curvas y Superficies *[Beddoe 99]*. El Apéndice C, describe los principales métodos utilizados en cada una de estas clases. La figura 3.1 muestra el esquema general de la especificación OpenGis sobre la jerarquía de Clases Geomérticas.

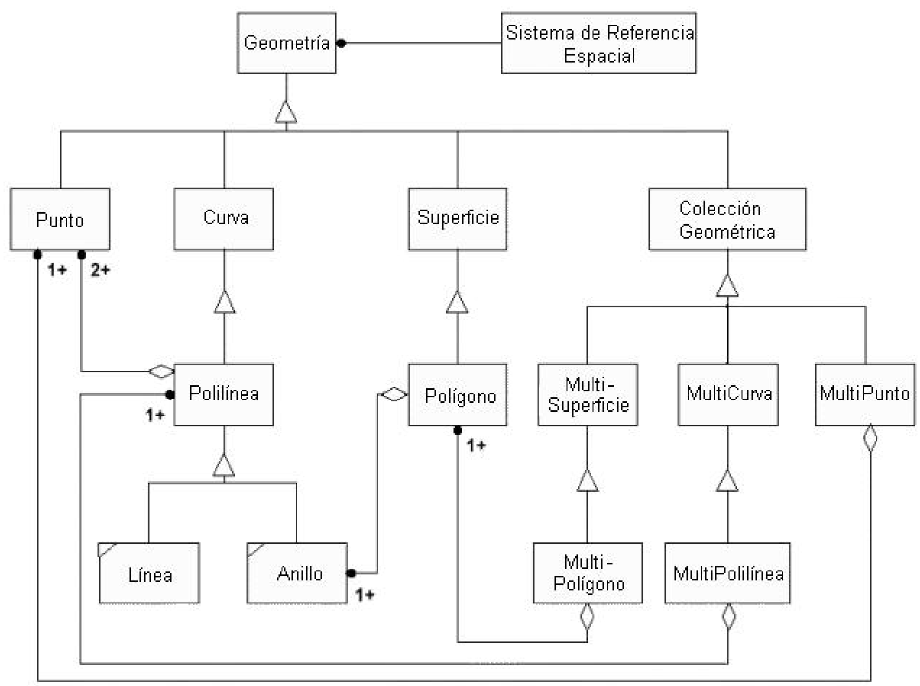

Figura 3.1. Jerarquía de Clases Geométricas

Geometría

Geometría es la dase de mayor jerarquía, y no se pueden crear instancias de esta clase, ya que es una clase abstracta.

Las subclases instanciables de Geometría se restringen a objetos geométricos de 0, 1 y 2 dimensiones. Los cuales son representados en coordenadas espaciales de dos dimensiones ($\Re^2$).

Punto

Un punto es una geometría de dimensión 0 y es representada por una localización simple en coordenadas de espacio. Un Punto tiene un valor para la coordenada "x" y un valor para la coordenada "y", ver figura 3.2.

MultiPunto

Un MultiPunto es una colección geométrica. Los elementos de un MultiPunto son restringidos a Puntos, los cuales no están conectados ni ordenados. Un MultiPunto es simple si no existen Puntos con valores idénticos, es decir no existen Puntos iguales. Un ejemplo de MultiPunto se muestra en la figura 3.2. El límite (boundary) de un MultiPunto es el conjunto vacío.

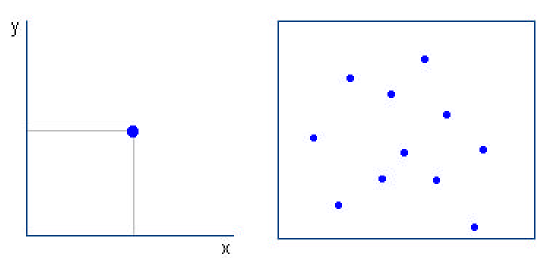

Figura 3.2. Ejemplo de Puntos y Multipuntos

Curva

Una Curva es un objeto geométrico de una dimensión, usualmente es una sucesión de puntos, con el subtipo curva se especifica la interpolación entre puntos.

Una Curva es simple si no pasa por el mismo punto dos veces. El ejemplo 1 de la figura 3.3 muestra una curva simple. El ejemplo 2 representa una curva no simple. Se dice que es una curva cerrada, si el punto inicial es igual al punto final, ver ejemplo 3 de la misma figura. El ejemplo 4 muestra una curva cerrada no simple.

El límite de una curva cerrada es el conjunto vacío. Una Curva que es simple y cerrada es un Anillo (Ring). El límite de una Curva abierta consiste en dos puntos, el inicial y el final.

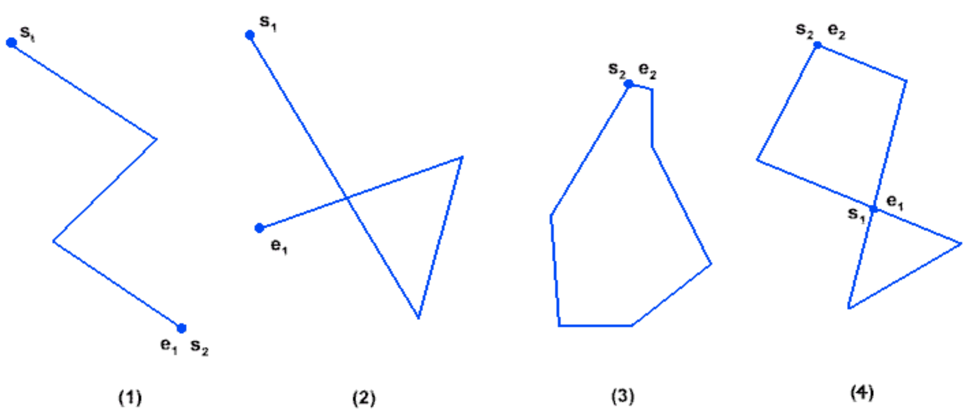

Figura 3.3. Ejemplo de Curvas

Polilínea

Una Polilínea es una Curva, que es subclase de Curva, la cual es una interpolación lineal entre puntos. Cada par consecutivo de puntos definen un segmento de Línea. Un Anillo es una Polilínea cerrada y simple. Una línea es una Polilínea de 2 puntos exactamente.

MultiCurva

MultiCurva una clase no instanciable, es una Colección Geométrica de dimensión uno y sus elementos son Curvas. Es simple si todos sus elementos son Simples y solo puede haber intersección entre los límites de sus elementos.

La MultiCurva es cerrada, si todos sus elementos son cerrados. El límite de una MultiCurva Cerrada es siempre el conjunto vacío.

MultiPolilínea

Una MultiPolilínea es una MultiCurva y sus elementos son Polilíneas. La figura 3.4 muestra 3 ejemplos de MultiPolilínea. La primera esta formada por dos Polilíneas que se tocan en uno de sus puntos iniciales. La segunda esta formada por una Polilínea no simple y una cerrada. La tercera MultiPolilínea, está formada por dos Polilíneas cerradas.

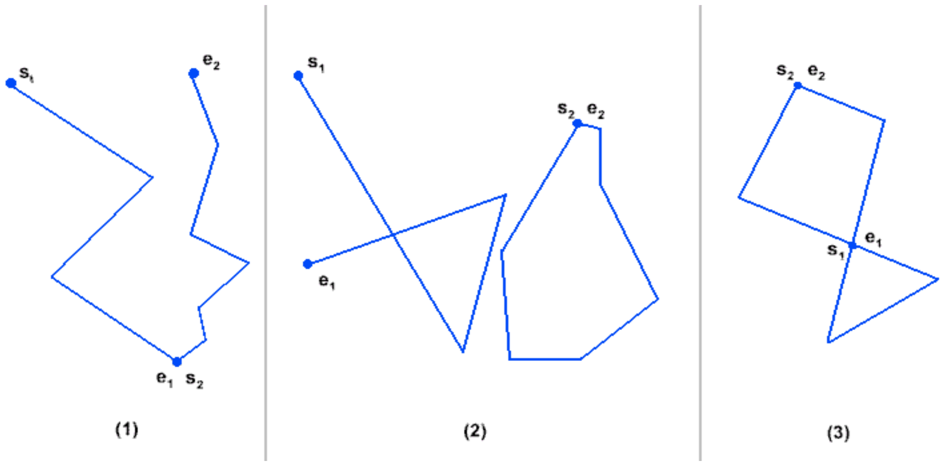

Figura 3.4 Ejemplo de MultiPolilínea

Superficie

OpenGis define a una Superficie Simple como un límite exterior con 0 o más límites interiores. El límite de una Superficie simple es el conjunto de curvas cerradas correspondientes a los límites exterior e interior.

Polígono

Un Polígono es una Superficie Simple Plana. Una Superficie es un objeto geométrico de dos dimensiones.

Un Polígono es topológicamente cerrado. El límite de un Polígono consiste de un conjunto de Líneas que forman el interior y el exterior. Un Polígono no puede tener líneas que cortan, púas o perforaciones.

El interior de un Polígono está formado por un conjunto de puntos conectados. El exterior de un Polígono es el conjunto de puntos no conectados.

Un Polígono es definido por un límite exterior y cero o más límites interiores. La figura 3.5 muestra tres ejemplos de Polígonos, formados por un límite exterior.

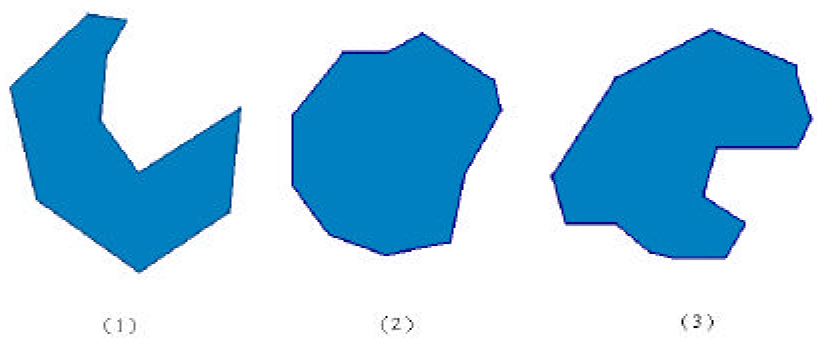

Figura 3.5. Ejemplo de Polígonos

MultiSuperficie

Una MultiSuperficie es una Colección Geométrica de dimensión dos cuyos elementos son Superficies.

MultiSuperficie no es una clase instanciable. La subclase instanciable es MultiPoligono la cual corresponde a una Colección de Polígonos.

MultiPolígono

MultiPolígono es una MultiSuperficie cuyos elementos son Polígonos, en la figura 3.6 se incluyen algunos ejemplos. En un MultiPolígono los interiores de dos polígonos no pueden intersectar. Los límites de dos Polígonos no pueden cruzar, sin embargo si pueden tocarse por un numero finito de puntos. Un MultiPoligono es topológicamente cerrado y no puede tener líneas que lo corten, púas ni perforaciones.

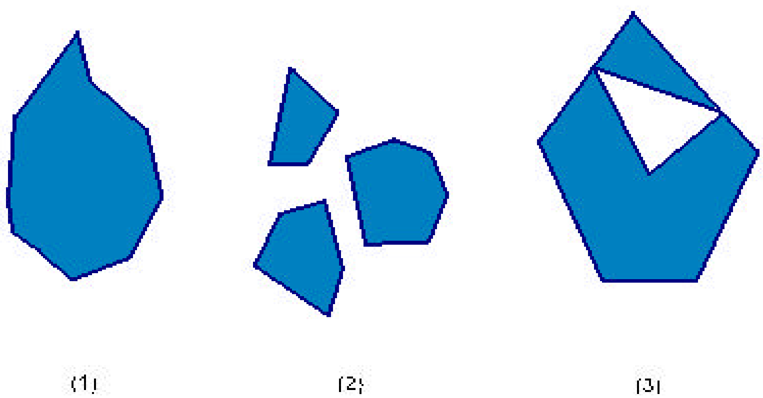

Figura 3.6. Ejemplos de MultiPolígonos

El límite de un MultiPolígono es el conjunto de curvas cerradas (Polilíneas) correspondientes a los límites de los Polígonos. Cada una de estas Curvas corresponde al límite de un Polígono exactamente.

3.3 Modelo para Sistemas de Referencia Espacial

El propósito de la especificación de lo abstracto es crear un modelo conceptual que permita la implementación de especificaciones. De la especificación de lo abstracto derivan dos modelos. El primer modelo es el Esencial, cuyo propósito es establecer la unión conceptual del software o diseño del sistema con el mundo real. Es una descripción de cómo trabaja el mundo (o debe trabajar). El segundo modelo es el Abstracto, el cual es una descripción que plantea cómo debe trabajar el software.

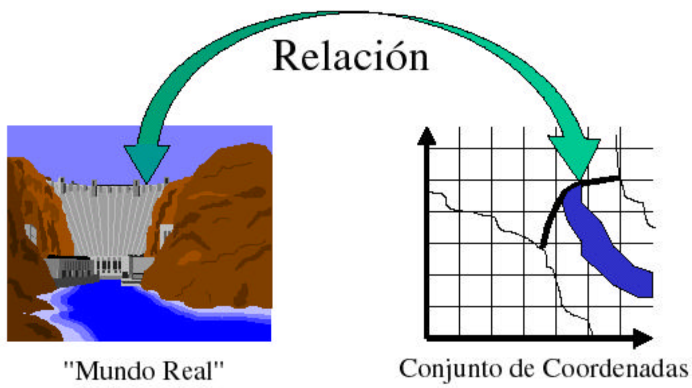

Figura 3.7. Relación entre Localizaciones y Coordenadas

Un mapeo entre estos dos modelos se muestra en la figura 3.7. En donde la localización física de una presa, que pertenece al mundo real (Modelo Esencial), es representada como un conjunto de coordenadas en nuestro sistema (Modelo Abstracto).

Las coordenadas son usadas para designar o "nombrar" las posiciones de un espacio, en un sistema de referencia particular. Las coordenadas espaciales son abstracciones de localizaciones en el espacio, son nombres que les damos a las localizaciones, las cuales nos permiten realizar algunas operaciones como calcular la distancia entre dos puntos expresados con coordenadas, o encontrar la intersección entre dos áreas, ya que sus límites están expresados en coordenadas.

Las coordenadas son tuplas que toman valores numéricos en cada posición o ordenadas. En donde las coordenadas espaciales de dos dimensiones corresponde al conjunto {(x,y) | ”x” y “y” son números reales}, y las coordenadas espaciales de tres dimensiones corresponde al conjunto {(x,y,z) | ”x”, “y” y “z” son números reales positivos} o un subconjunto de estos.

Las coordenadas espaciales son coordenadas con semántica espacial. Esta semántica puede ser compleja, siendo el objetivo del Modelo Esencial clarificar esta semántica. La semánticas de las coordenadas espaciales se refieren a la relación entre coordenadas y el lugar.

3.4 SQL

OpenGis propone utilizar el esquema del estándar SQL para el almacenamiento, recuperación, consulta y actualización de colecciones de características geoespaciales simples, en aplicaciones ODBC.

Una característica simple es definida como una especificación abstracta que tiene ambos atributos espaciales y no espaciales. Los atributos espaciales son valores geométricos y se basan en geometrías 2D con interpolación lineal entre vértices.

Una colección de características geoespaciales simples conceptualmente se almacenan como tablas en columnas con valores geométricos en un DBMS Relacional (RDBMS), cada característica se almacena como un renglón de la tabla.

OpenGIS propone utilizar una de las siguientes 3 alternativas de SQL con ODBC:

1. SQL numérico para el almacenamiento de la geometría y acceso ODBC.

2. SQL binario para el almacenamiento de la geometría y acceso ODBC.

3. SQL92 con Tipos Geométricos implementando la tabla de características soportando ambos accesos geométricos a ODBC textual y binario.

En un ambiente SQL92, una columna con valores geométricos, es implementada con una referencia de llave foránea en una tabla geométrica. Es decir un valor geométrico es almacenado usando uno o más renglones en una tabla geométrica. Por ejemplo, la presa de la figura 3.7, puede ser definida en el sistema, por un conjunto de coordenadas. Estas coordenadas son almacenadas en una tabla, utilizando renglones diferentes para cada par “x” y “y”, utilizando el identificador del objeto geométrico (ID), en cada renglón almacenado:

ID geométrico	x	y
01	25.74	34.78
01	37.43	38.89
01	42.29	40.23

El término SQL92 con tipos Geométricos, es usado para referirse al ambiente SQL92 que se ha extendido con un conjunto de tipos Geométricos. En este ambiente las columnas con valores geométricos son implementados como una sola columna. En este caso el conjunto de coordenadas pueden ser almacenadas en el mismo renglón, junto con su identificador, para poder recuperar el objeto geométrico, quedando la tabla de la siguiente manera:

ID	Localización					
01	(25.74	34.78	37.43	38.89	42.29	40.23)

En estos ambientes, cada columna geométrica es asociada con un Sistema de Referencia Espacial, el cual identifica el Sistema de coordenadas para todas las geometrías almacenadas en la columna y proporciona un significado a los valores de las coordenadas para cualquier instancia almacenada en la columna.

El estándar que un futuro traerá grandes ventajas en el manejo de objetos geométricos, es el estándar internacional ISO/IEC, el cual es un proyecto para el desarrollo de un una Librería SQL para aplicaciones de multimedia aprobada en 1999. Esta nueva estandarización nombrada SQL Multimedia (SQL/MM), especificará paquetes de SQL usando definiciones de tipos de datos abstractos (ADT).

El esquema para la base de datos, necesario para soportar OpenGis, se muestra en la figura 3.8. El cual, fué implementado en el Servidor OpenGis de nuestro sistema [Escobar 00].

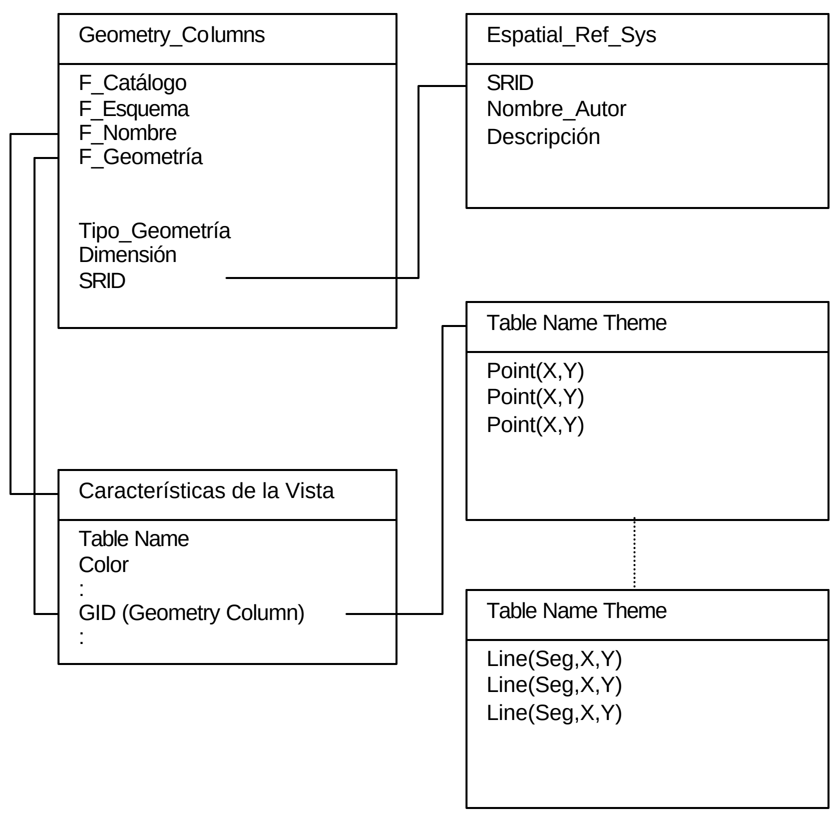

Figura 3.8. Esquema de características de las tablas utilizando SQL92

SQL/MM intenta estandarizar las librerías de clases para ciencia e ingeniería, texto y procesamiento de documentos, métodos para el manejo de objetos multimedia, objetos como imagen, sonido, animación, música, y video.

Otros estándares en el área de Información Geográfica y Espacial, que se espera incluya SQL/MM son: [ISO/TC 211 99]

1. ANSI X3L1 - Sistemas de la Información Geográfica.

2. ISO TC 211- Información Geográfica / Geomatics

3.5 Conclusiones

Diferentes modelados de datos podemos emplear en un SIG, sin embargo es necesario buscar el intercambio de información de nuestra aplicación con otros sistemas. Por lo que decidimos utilizar el Modelo de Objetos Geométricos propuesto por OpenGis, ya que es el que más se acerca a lo que será un estándar.

Este Modelo de objetos Geométricos, está estrechamente relacionado con el Sistemas de Referencia Espacial permitiendo obtener consultas sobre los objetos geográficos. Es decir todos los objetos para poder desplegarlos y analizarlos, deben estar bajo un mismo Sistema de Coordenadas.

El Sistema de Coordenadas que se empleó en nuestra aplicación es el sistema de Coordenadas Cartesianas.

El Componente de consultas espaciales, proporciona al usuario la información necesaria para la consulta, contemplando que los datos sean recuperados y almacenados utilizando el estándar SQL92.

Otro estándar, que seguramente traerá mayor beneficio es el SQL99 (SQL/MM), que considera tipos de datos abstractos, objetos como imágenes, sonidos, música y video, así como también el estándar ISO/TC 211 99 que permitirá el manejo de Objetos Geográficos.

4.1 Introducción

Para poder realizar un análisis espacial, es necesario contar con un modelo de interacción visual, el cual nos ayude a encontrar las relaciones que existen entre los objetos. Existen diferentes modelos, de las cuales explicaremos el modelo de relaciones Topológicas Binarias *[Egenhofer 93]* y el modelo de 9 intersecciones *[Beddoe 99]*.

También es importante contar con un lenguaje, que nos permita capturar el tipo de consulta que aplicaremos sobre los datos. En este capítulo se describe un lenguaje visual, para Consultas en SIG´s.

4.2 Modelo de Relaciones Topológicas Binarias

Este modelo se define en términos de invariantes topológicas de cuatro intersecciones entre objetos espaciales homogéneos. Las relaciones son definidas de acuerdo a los resultados obtenidos, identificando ocho relaciones topológicas entre dos objetos espaciales *[Egenhofer 93]*.

Para determinar la relación entre dos objetos se debe buscar la existencia de las siguientes cuatro intersecciones, las cuales retornan “0” cuando no existe la intersección y “¬0” cuando existe.

1. Existen límites comunes entre los dos objetos ($\partial \cap \partial$).

2. Existe interiores en común ($^\circ \cap {}^\circ$).

3. Parte del límite de un objeto que coincide con parte del interior del otro objeto ($\partial \cap {}^\circ$).

4. Parte del interior de un objeto coincide con el límite del otro objeto ($^\circ \cap \partial$).

$$R_r\ (A\,.\,B) \quad \Leftrightarrow \quad I\,(A\,.\,B) \quad = \quad \begin{pmatrix} \partial \cap \partial & \partial \cap {}^\circ \\ {}^\circ \cap \partial & {}^\circ \cap {}^\circ. \end{pmatrix}$$

Las ocho relaciones que se pueden detectar, a partir de los resultados obtenidos de estas cuatro intersecciones son las siguientes:

No toca a (Disjoint)

Si las cuatro intersecciones entre todas las caras no existe, entonces las dos áreas están separadas, esta relación es representada en la figura 4.1.

$$R_{disjoint}\ (A\,.\,B) \ \Leftrightarrow\ I\,(A\,.\,B) \quad = \quad \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

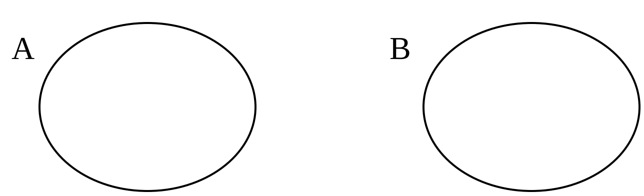

Figura 4.1 Relación “No toca a”

Toca a (Meet)

Si la intersección entre los límites de los objetos existe y las otras tres intersecciones no se cumplen entonces estos objetos se tocan, ver figura 4.2.

$$R_{meet}\ (A\ .\ B) \Leftrightarrow I\,(A\ .\ B) = \begin{pmatrix} \neg 0 & 0 \\ 0 & 0 \end{pmatrix}$$

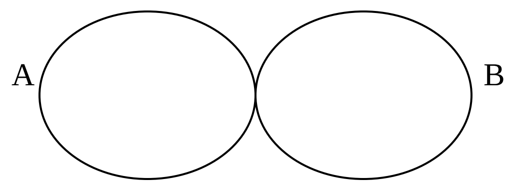

Figura 4.2 Relación "Toca a"

Igual a (Equal)

Dos regiones son iguales, ver figura 4.3, si existen ambas intersecciones de limites e interiores.

$$R_{equalst}\ (A\ .\ B) \Leftrightarrow I\,(A\ .\ B) = \begin{pmatrix} \neg 0 & 0 \\ 0 & \neg 0 \end{pmatrix}$$

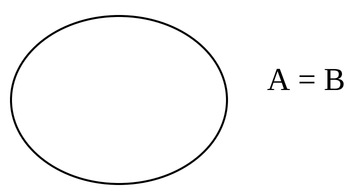

Figura 4.3 Relación "Igual a"

Dentro de (Inside)

Una región A esta dentro de otra región B si: 1) A y B comparten el mismo interior pero no los límites. 2) Si el límite de A es un subconjunto del interior de B. 3) y el límite de B no intersecta con ninguna parte del interior de A, ver figura 4.4.

$$R_{inside}\ (A\ .\ B) \iff I\,(A\ .\ B) = \begin{pmatrix} 0 & \neg 0 \\ 0 & \neg 0 \end{pmatrix}$$

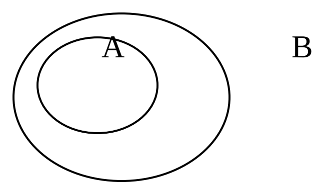

Figura 4.4 Relación "Dentro de"

Contiene a (Contain)

Una región A contiene a otra región B, ver figura 4.5. Si 1) A y B comparten el mismo interior, pero no tienen límites en común. 2) Si el límite de B es un subconjunto del interior de A, 3) el límite de A no intersecta con el interior de B

$$R_{contain}\ (A\ .\ B) \iff I\,(A\ .\ B) = \begin{pmatrix} 0 & 0 \\ \neg 0 & \neg 0 \end{pmatrix}$$

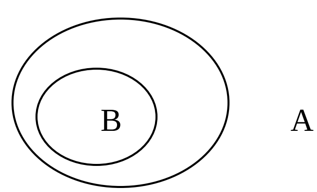

Figura 4.5 Relación "Contiene a"

Es cubierto por (Covered by)

Una región es cubierta por otra región B si ambas regiones tienen partes en común en sus límites e interiores. Si parte del interior de A intersectan con parte de los límites de B y si el interior de B intersecta con parte del límite de A. Esta relación es representada por la figura 4.6.

$$R_{\text{covered by}}\ (A\,.\,B) \Leftrightarrow I\,(A\,.\,B) = \begin{pmatrix} \neg 0 & \neg 0 \\ 0 & \neg 0 \end{pmatrix}$$

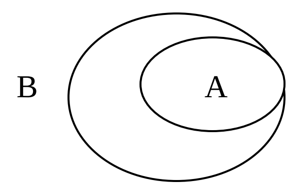

Figura 4.6 Relación "Es cubierto por"

Cubre a (Covers)

Una región A cubre a otra región B, si ambas regiones comparten el mismo límite e interior; Si el interior de B intersecta con el límite de A; y si el interior de A es parte del límite de B, ver figura 4.7.

$$R_{\text{covers}}\ (A\,.\,B) \Leftrightarrow I\,(A\,.\,B) = \begin{pmatrix} \neg 0 & 0 \\ \neg 0 & \neg 0 \end{pmatrix}$$

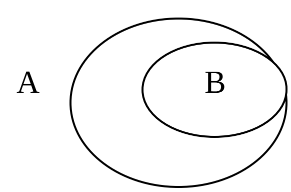

Figura 4.7 Relación "Cubre a"

Cubre parte de (Overlap)

Una región cubre parte de otra, ver figura 4.8, si tienen límites e interiores en común y ambos límites intersectan con el interior del otro objeto.

$$R_{overlap}\ (A\ .\ B) \Leftrightarrow I\ (A\ .\ B) = \begin{pmatrix} \neg 0 & \neg 0 \\ \neg 0 & \neg 0 \end{pmatrix}$$

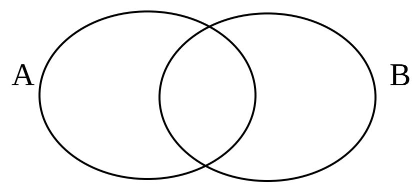

Figura 4.8 Relación "Cubre parte de"

Este enfoque obtiene la relación que existe entre dos objetos, sin embargo, solo es empleado para obtener relaciones entre superficies. Un modelo más completo es el Modelo de 9 Intersecciones, este modelo obtiene las relaciones entre puntos, polilíneas y áreas, el cual describimos a continuación.

4.3 Modelo de Nueve-Intersecciones Dimensionalmente Extendido DE-9IM

En este modelo, cada objeto p que se representa en espacio R_2, es considerado como un conjunto de puntos, formando un interior, un límite y un exterior. La relación topológica entre dos objetos p y q, es descrita por las nueve intersecciones de interior, límite y exterior de p, con el interior, límite y exterior de q *[Papadias 95].*

Dada una Geometría, podemos obtener su Interior, límite y Exterior representados por *I(a), B(a), E(a)* respectivamente. *[Beddoe 99]*

1. El Interior de un objeto se refiere al conjunto de Puntos conectados.

2. El Exterior se refiere al conjunto de puntos no conectados.

3. El límite de un objeto se refiere al conjunto de geometrías de una dimensión menor a la del objeto.

De acuerdo a lo anterior podemos decir que el límite de un punto es el conjunto vacío, y su interior es el mismo punto. El límite de una polilínea son dos puntos, el punto inicial y el punto final de la polilínea y su interior es el resto de los puntos conectados del objeto. Sin embargo, el límite para una polilínea cerrada es el conjunto vacío, ya que el punto inicial es igual al punto final (para el límite de las polilíneas se aplica la regla mod 2). El límite de un polígono es una polilínea cerrada y su interior esta formado por el conjunto de puntos conectados, ver Figura 4.9 y definiciones de estas geometrías en el capítulo anterior.

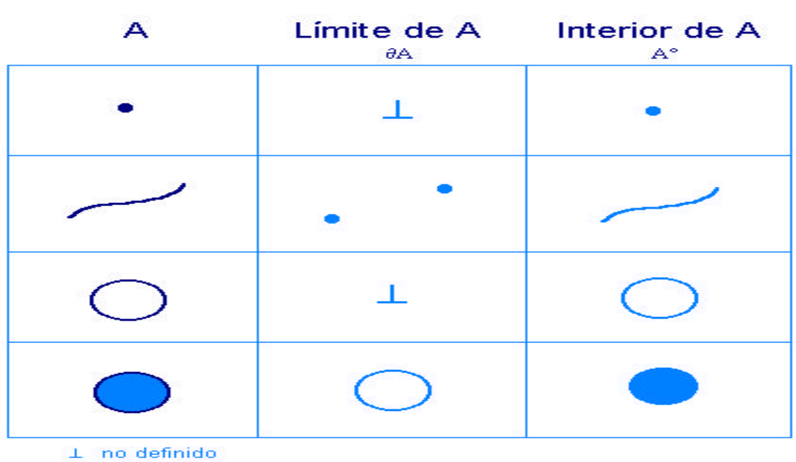

Figura 4.9 Ejemplo de Límite e Interior

La intersección de $I(a)$, $B(a)$, $E(a)$ puede dar por resultado un juego de geometrías, x, de dimensión mixta. Por lo que $dim(x)$ retorna la máxima dimensión (-1, 0, 1, 2), donde –1 representa al conjunto vacío ($\emptyset$).

	Interior	Boundary	Exterior
Interior	$dim(I(a) \cap I(b))$	$dim(I(a) \cap B(b))$	$dim(I(a) \cap E(b))$
Boundary	$dim(B(a) \cap I(b))$	$dim(B(a) \cap B(b))$	$dim(B(a) \cap E(b))$
Exterior	$dim(E(a) \cap I(b))$	$dim(E(a) \cap B(b))$	$dim(E(a) \cap E(b))$

Tabla 4.1. Matriz de Intersecciones del Modelo DE-9IM

La matriz 4.1 consiste en un conjunto de 9 valores, uno para cada celda de la matriz. Cuyos valores posibles son { T, F, *, 0, 1, 2 } y sus significado en cualquier celda son los siguientes:

P = T $\Rightarrow$ *dim(x)* $\in \{0, 1, 2\}$, $x \neq \emptyset$

P = F $\Rightarrow$ *dim(x)* = -1, $x = \emptyset$

P = * $\Rightarrow$ *dim(x)* $\in \{-1, 0, 1, 2\}$, *x* = a cualquier valor.

P = 0 $\Rightarrow$ *dim(x)* = 0

P = 1 $\Rightarrow$ *dim(x)* = 1

P = 2 $\Rightarrow$ *dim(x)* = 2

Las relaciones que se pueden obtener a partir de estas intersecciones se definen a continuación. Utilizando P para referirse a geometrías de dimensión 0 (Punto y MultiPunto), L para geometrías de dimensión 1 (Polilínea y MultiPolilínea) y para geometrías de dimensión 2 (Polígono y MultiPolígono).

No toca a (Disjoint)

Dados dos objetos A y B, A no toca a B (ver figura 4.10), Si.

$$a.Disjoint(b) \Leftrightarrow a \cap b = \emptyset$$

Expresado en términos de DE-9IM:

$$a.Disjoint(b) \Leftrightarrow (I(a) \cap I(b) = \emptyset) \wedge (I(a) \cap B(b) = \emptyset) \wedge (B(a) \cap I(b) = \emptyset) \wedge (B(a) \cap B(b) = \emptyset)$$

*Û a.Relate(b, ‘FF*FF****’)*

a \ b	**Interior**	**Límite**	**Exterior**
Interior	***F***	***F***	*
Límite	***F***	***F***	*
Exterior	*	*	*

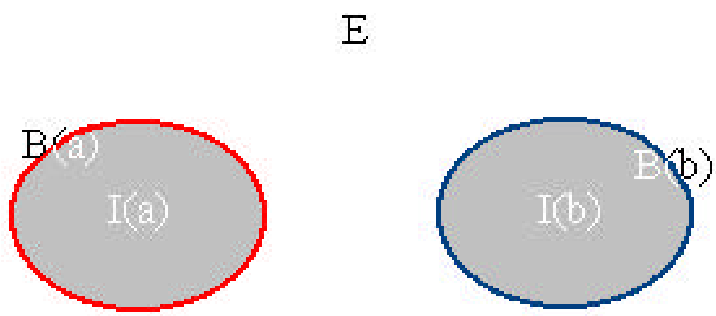

Figura 4.10. Ejemplo de la relación “No Toca a”

Intersecta a (Intersects)

$$a.Intersects(b) \Leftrightarrow !\ a.Disjoint(b)$$

Toca a (Touches)

La relación "Toca a" entre dos geometrías a y b es aplicado a grupos de A/A, L/L, L/A, P/A y P/L pero no a relaciones entre grupos P/P, ver figura 4.11. Este es definido como:

$$a.Touches(b) \Leftrightarrow (I(a) \cap I(b) = \varnothing) \wedge (a \cap b) \neq \varnothing$$

Expresado en términos de DE-9IM:

$$a.Touches(b) \Leftrightarrow (I(a) \cap I(b) = \varnothing) \wedge ((B(a) \cap I(b) \neq \varnothing) \vee (I(a) \cap B(b) \neq \varnothing) \vee (B(a) \cap B(b) \neq \varnothing))$$

*Û a.Relate(b, 'FT*******') Ú a.Relate(b, 'F**T*****') Ú a.Relate(b, 'F***T****')*

a \ b	Interior	Límite	Exterior
Interior	*F*	*T*	*
Límite	*	*	*
Exterior	*	*	*

a \ b	Interior	Límite	Exterior
Interior	*F*	*	*
Límite	*T*	*	*
Exterior	*	*	*

a \ b	Interior	Límite	Exterior
Interior	*F*	*	*
Límite	*	*T*	*
Exterior	*	*	*

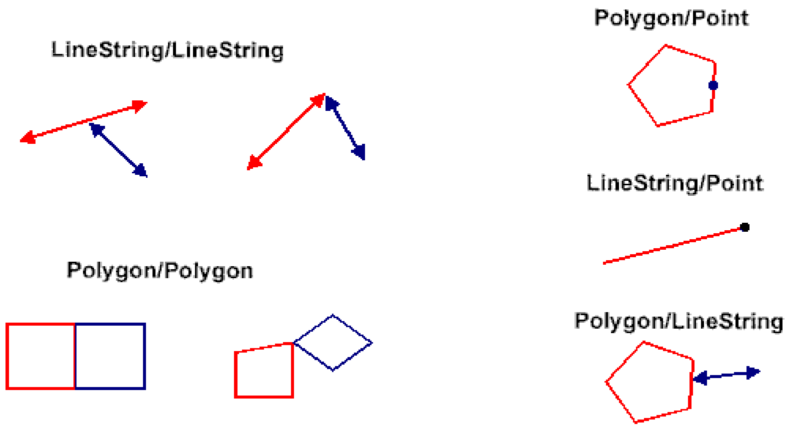

Figura 4.11 Ejemplos de la relación "Toca a"

Cruza a (Crosses)

La relación "Cruza a" es aplicada a situaciones como L/L y L/A. Ver figura 4.12. Esta realación es definida como:

$$a.Crosses(b) \Leftrightarrow (dim(I(a) \cap I(b)) < max(dim(I(a)), dim(I(b)))) \wedge (a \cap b \neq a) \wedge (a \cap b \neq b)$$

Expresado en términos de DE-9IM:

Si $a \in L$, $b \in A$:

$$a.Crosses(b) \Leftrightarrow (I(a) \cap I(b) \neq \varnothing) \wedge (I(a) \cap E(b) \neq \varnothing)$$

$$\Leftrightarrow a.Relate(b, 'T*T******')$$

a \ b	Interior	Límite	Exterior
Interior	*T*	*	*T*
Límite	*	*	*
Exterior	*	*	*

Si a $\in$ L, b $\in$ L:

$$a.Crosses(b) \Leftrightarrow dim(I(a) \cap I(b)) = 0$$

$$\Leftrightarrow a.Relate(b, \text{'0********'})$$

a \ b	Interior	Límite	Exterior
Interior	*0*	*	*
Límite	*	*	*
Exterior	*	*	*

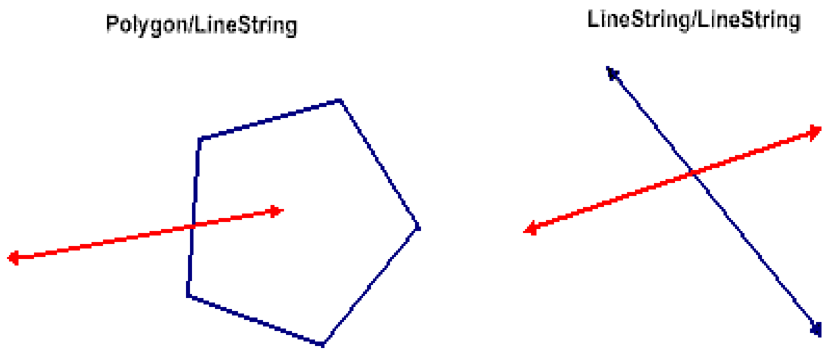

Figura 4.12. Ejemplos de la relación "Cruza a"

Dentro de (Within)

La relación "dentro de" es definida como:

$$a.Within(b) \Leftrightarrow (a \cap b = a) \wedge (I(a) \cap I(b) \neq \emptyset)$$

Expresado en términos de DE-9IM:

$$a.Within(b) \Leftrightarrow (I(a) \cap I(b) \neq \emptyset) \wedge (I(a) \cap E(b) = \emptyset) \wedge (B(a) \cap E(b) = \emptyset)\ )$$

$$\Leftrightarrow a.Relate(b, \text{'T*F**F***'})$$

a \ b	Interior	Límite	Exterior
Interior	*T*	*	*F*
Límite	*	*	*F*
Exterior	*	*	*

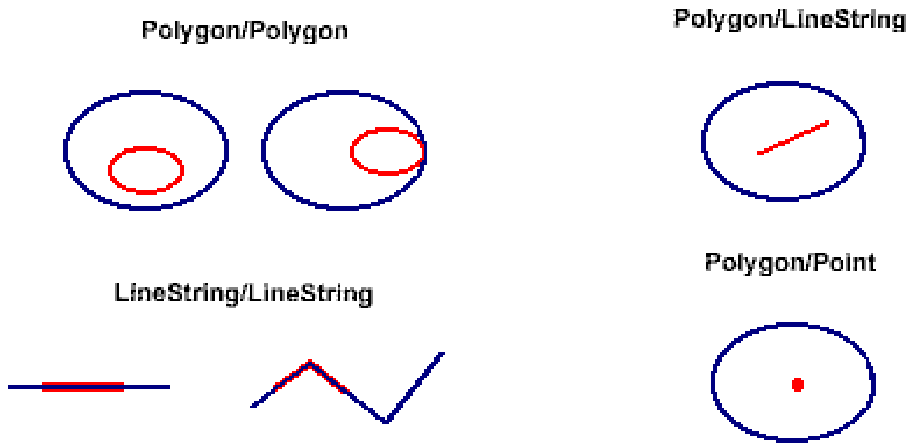

Figura 4.13 Ejemplos de la relación "Contiene a"

Contiene a (Contains)

a.Contains(b) $\Leftrightarrow$ b.Within(a), ver figura 4.13.

Cubre parte de (Overlaps)

La relación "Cubre parte de" para situaciones como A/A, L/L. Esta relación es representada en la figura 4.14 y definida como:

$$a.\mathrm{Overlaps}(b) \Leftrightarrow (\dim(I(a)) = \dim(I(b)) = \dim(I(a) \cap I(b))) \wedge (a \cap b \neq a) \wedge (a \cap b \neq b)$$

Expresado en términos de DE-9IM:

Si $a \in A, b \in A$:

$$a.\text{Overlaps}(b) \Leftrightarrow (I(a) \cap I(b) \neq \emptyset) \wedge (I(a) \cap E(b) \neq \emptyset) \wedge (E(a) \cap I(b) \neq \emptyset)$$

$$\Leftrightarrow a.\text{Relate}(b, \text{'T*T***T**'})$$

a \ b	**Interior**	**Límite**	**Exterior**
Interior	*T*	*	*T*
Límite	*	*	*
Exterior	*T*	*	*

Si $a \in L, b \in L$:

$$a.\text{Overlaps}(b) \Leftrightarrow (\dim(I(a) \cap I(b) = 1) \wedge (I(a) \cap E(b) \neq \emptyset) \wedge (E(a) \cap I(b) \neq \emptyset)$$

$$\Leftrightarrow a.\text{Relate}(b, \text{'1*T***T**'})$$

a \ b	**Interior**	**Límite**	**Exterior**
Interior	*1*	*	*T*
Límite	*	*	*
Exterior	T	*	*

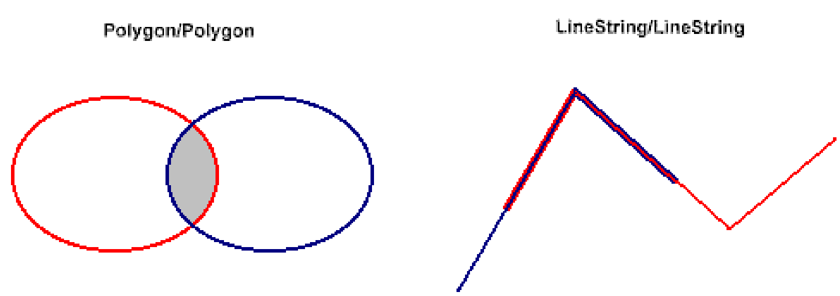

Figura 4.14 Ejemplo de la relación "Cubre parte de"