

Manipulación de Campos Continuos y su aplicación a los Sistemas de Información Geográfica

Arturo Zambrano, Luis Polasek

Director Silvia Gordillo (PhD)

Trabajo de Grado de la Facultad de Informática de la UNLP.
Diciembre 2001

Agradecimientos

Agradecemos a toda la gente del grupo GIS, con quienes trabajamos durante los últimos 3 años. Ellos son:

- Silvia Gordillo, directora del proyecto.
- Diego Cano, topologías.
- Javier Bazzocco y Gisella Trilla, apéndice.
- Alejandra Lliteras, sistema de referencia.

En especial queremos agradecer al LIFIA y a todos sus miembros, de quienes recibimos la formación profesional y humana necesaria para dar nuestros primeros pasos en investigación. Por último queremos agradecer a la Universidad Nacional de La Plata y expresar nuestro apoyo a la universidad pública y gratuita.

Índice General

Agradecimientos	iii
1 Introducción	1
1.1 Modelos Ambientales	2
1.2 La Continuidad en los Modelos Ambientales	3
2 Medición y Representación de Campos	5
2.1 Medición y Representación de Valores y Ubicaciones	5
2.1.1 “Las escalas de medición”	5
2.1.2 Ubicación de la Medición	7
2.2 Representando Campos	8
2.2.1 Discretización del Espacio	9
2.3 Modelos de Datos Espaciales para los Campos	10
2.3.1 Grillas de Celdas	10
2.3.2 Poligonos	10
2.3.3 TINs (Triangulated Irregular Networks, Redes Trianguladas Irregulares)	11
2.3.4 Grillas de Puntos	11
2.3.5 Grillas Irregulares de Puntos	11
2.3.6 Modelos de Contornos	12
2.3.7 Características de los modelos de datos espaciales	12
2.3.8 Modelos de datos espaciales como una representación de la realidad	15
2.3.9 Estructuras de datos para campos que modelan datos espaciales	16
2.4 Modelando con variables continuas	17
2.4.1 Una estrategia para tratar la continuidad espacial	17
2.5 Variables de Campo	18
2.5.1 Tipos de datos campo y variables de campo	18
2.5.2 Igualdad espacial y anidamiento	18
3 Operaciones sobre variables de campo	21
3.1 Operaciones GIS	21
3.1.1 Álgebra de mapas (Tomlin)	21
3.1.2 Las categorías de Raper y Maguire	22
3.1.3 Clasificación de Burrough	22
3.2 Matemática sobre campos	23
3.2.1 Asignaciones:	24

3.3	Convirtiendo modelo de datos de campos	25
3.3.1	Interpolación de modelos de campos numéricos	26
3.3.2	Muestreando para campos numéricos	28
3.3.3	Remuestreo del modelo de campos numéricos	29
3.3.4	Interpolando campos de modelos categóricos	33
3.3.5	Muestreado de datos categóricos	34
3.3.6	Remuestreo de modelo de campos categóricos	34
3.3.7	Problemas con los campos convertidos	35
3.3.8	Las matrices de conversión	35
3.3.9	La estructura computacional	35
3.4	Operadores Aritméticos	38
3.4.1	Operadores aritméticos binarios	38
3.4.2	Operadores Unarios	38
3.4.3	Operaciones aritméticas vectoriales	39
3.4.4	Trigonometría	39
3.4.5	Aritmética booleana	39
3.5	Operaciones que involucran diferentes modelos espaciales	39
3.5.1	Escalas incompatibles	42
3.6	Estadísticas descriptivas para campos y variables de campo	44
3.6.1	Media	45
3.6.2	Suma (Integración)	46
3.6.3	Máximo y Mínimo	47
3.6.4	Desviación estándar	48
3.7	Estadísticas Inferenciales	48
3.7.1	Geoestadística y la teoría de variables regionalizadas	49
3.7.2	Construcción de variogramas	50
3.7.3	Kriging	50
3.7.4	Usando Geoestadística	51
3.7.5	Geoestadística para modelos de datos espaciales	52
4	Investigaciones Relacionadas	53
4.0.6	Map Algebra	53
4.0.7	Interfaces modelo-datos	54
4.0.8	Soluciones orientadas a las BD	55
4.0.9	Lenguajes formales y álgebras	56
4.0.10	Virtual Data Sets	57
4.0.11	Open GIS	58
4.1	Modelos Matemáticos	59
4.1.1	Orientación a objetos	61

5	Breve Descripción del Framework para Aplicaciones GIS	63
5.1	Sistema de Referencia	64
5.2	Modelo Topológico	66
5.3	Visualización de la información	67
5.4	Campos Continuos	68
5.4.1	Representaciones para campos continuos	69
5.4.2	Métodos de estimación	70
6	Una Arquitectura OO para Operaciones con Campos Continuos	73
6.1	Clasificación de Operaciones para Campos Continuos	73
6.1.1	Operaciones Unarias para Campos Continuos	74
6.1.2	Operaciones N-arias para Campos Continuos	75
6.2	Arquitectura OO para operaciones con campos continuos	79
6.2.1	Estructura estática	79
6.2.2	Operaciones compuestas	83
6.3	Estructura dinámica	84
6.4	Extensiones a la Arquitectura	86
6.4.1	Hot Spots	86
6.4.2	Hot Spots en la arquitectura de operaciones para campos continuos	87
7	Conclusiones y Trabajo Futuro	95
7.1	Conclusiones	95
7.2	Trabajo Futuro	96
	Bibliografía	99
A	Referencia Diagramas UML	103
A.1	Diagramas Estáticos del Modelo	103
A.1.1	Comentarios.	103
A.1.2	Design Patterns	104
A.1.3	Paquetes	105
A.1.4	Clases	106
A.1.5	Asociaciones	107
A.1.6	Herencia de Implementación (Generalización/Especialización)	109
A.1.7	Herencia de Interfase (Especialización/Refinamiento)	110
A.1.8	Dependencias	110
A.1.9	Relación de Agregación	111
A.1.10	Relación de Composición	111
A.1.11	Navegabilidad	112
A.1.12	Restricción	112
A.1.13	Restricciones Complejas	113
A.1.14	Asociación Calificada	113
A.1.15	Asociación de Clases	114

A.2	Diagramas Dinámicos del Modelo	115
A.2.1	Objetos y Mensajes	115
A.2.2	Creación de Objetos	116
A.2.3	Condiciones, Ciclos	117
A.2.4	Mensajes Asíncronos	118
A.2.5	Callbacks, Recursión	119
A.2.6	Tipos de flechas para los distintos mensajes	119
B	Patrones de Diseño	121
B.1	¿Qué es un Patrón de Diseño ?	121
B.2	Bridge	122
B.2.1	Intención	122
B.2.2	También Conocido Como	122
B.2.3	Estructura	123
B.2.4	Participantes	123
B.2.5	Colaboraciones	123
B.3	Command	124
B.3.1	Intención	124
B.3.2	También Conocido Como	124
B.3.3	Estructura	124
B.3.4	Participantes	124
B.3.5	Colaboraciones	125
B.4	Composite	125
B.4.1	Intención	125
B.4.2	Estructura	126
B.4.3	Participantes	126
B.4.4	Colaboraciones	127
B.5	Strategy	127
B.5.1	Intención	127
B.5.2	También Conocido Como	127
B.5.3	Estructura	127
B.5.4	Participantes	127
B.5.5	Colaboraciones	128

Índice de Tablas

3.1	El resumen de datos espaciales modela las conversiones para los datos numéricos .	36
3.2	El resumen de datos espaciales modela las conversiones para los datos categóricos .	37
3.3	Combinación de modelos de datos espaciales	41

Índice de Figuras

2.1	Modelos de Datos Espaciales para Campos Continuos	13
2.2	Modelos Constantes	14
2.3	Modelos de Superficie	14
2.4	Modelos de Puntos	15
3.1	Asignación de variables de campo.	24
3.2	Relación entre valores vecinos.	27
3.3	Interpolación en un modelo de contorno.	28
3.4	Conversión piecewise a piecewise.	30
3.5	Distintos tipos de curva.	31
3.6	Conversión de TIN a piecewise.	31
3.7	Conversión de TIN a piecewise.	32
3.8	Conversión del modelo de datos.	43
3.9	Distintas formas de resolver la conversión del modelo de datos.	44
3.10	Conversión del modelo de contornos a polígonos.	45
3.11	Distintos tipos de curva.	51
5.1	Objetos geográficos	64
5.2	Diagrama de clases de la subarquitectura del sistema de referencias	65
5.3	Diagrama de clases de la subarquitectura para topologías	67
5.4	Diagrama de clases de la subarquitectura de Appearances	71
5.5	Diagrama de clases: Campo continuo y muestra	71
5.6	Diagrama de clases: Campo continuo y representaciones	72
5.7	Diagrama de clases: Campo continuo, representaciones y métodos de estimación	72
6.1	Ejemplo de una selección	74
6.2	Esquema de una operación puntual	76
6.3	Esquema de una unión de dominios	78
6.4	La clase Evaluator	80
6.5	Instanciación de patrón Strategy. Construcción de muestras	81
6.6	Instanciación de patrón Strategy. Construcción de dominios	82
6.7	Instanciación de patrón Strategy. Operaciones puntuales	82
6.8	Reglas de conversión entre diferentes representaciones	83
6.9	Diagrama de clases de la arquitectura completa	88
6.10	El patrón Composite aplicado a la jerarquía de Requirements	89

6.11 Diagrama de instancias de campos continuos.	89
6.12 Diagrama de instancias de un <i>OperationEvaluator</i>	90
6.13 Diagrama de instancias de un <i>ComplexRequirement</i>	90
6.14 Diagrama de interacción para resolver un requerimiento.	91
6.15 Una hot-spot card	92
6.16 Una hot-spot card para generar la muestra	92
6.17 Una hot-spot card para creación de dominios	93
6.18 Una hot-spot card para operaciones puntuales	93

Capítulo 1

Introducción

Desde la perspectiva de los GIS, el problema consiste en que las tareas de análisis y modelamiento se vuelven importantes una vez que GIS se ha convertido en una tecnología establecida. El análisis de datos espaciales es un ítem en la agenda de investigación para GIS de segunda generación. Aparentemente se ha alcanzado una nueva era en los GIS, en la cual el foco de atención para la investigación necesita moverse desde los GIS **manejando** información a los GIS **usando** dicha información, con el consiguiente incremento en el énfasis en crear herramientas capaces de modelar y analizar esa información. Lamentablemente esas nuevas necesidades no han sido aun reflejadas en grandes iniciativas de investigación.

Los GISs han creado un gran número de BD espaciales que necesitan ser analizadas. Hay una oportunidad muy importante para los geógrafos, al empezar una nueva revolución relacionada al análisis y uso de **información geográfica**.

Los temas ambientales estan entre los más importantes a la hora de la toma de decisiones. La dinámica de los sistemas hidrológicos y atmosféricos de la tierra implica que todos los sistemas estén altamente interrelacionados, dinámica y espacialmente. El impacto de un evento en una posición geográfica generalmente afecta a otras, vecinas o no. Los sistemas para manejar datos espaciales y las técnicas analíticas para convertir esos datos en información son hoy en día herramientas vitales para lograr un entorno natural saludable.

Se está haciendo un considerable progreso en la integración de los sistemas de información espacial, GIS y los modelos matemáticos que rigen en medioambiente [GPS93]. Para la mayoría de los proyectos de modelos ambientales, los GISs son vistos como bases de datos apropiadas y bien estructuradas para el manejo de grandes cantidades de datos espaciales. La herramientas GIS tradicionales, como los *overlays* y *buffering* son también importantes para el desarrollo de juegos de datos derivados (derivative datasets) que sirven para generar vistas sobre variables. Muchos expertos esperan que pronto se incorporen mejores métodos de análisis espacial a los GIS actuales, la tecnología GIS se convertirá en una herramienta importante en todos los aspectos del modelamiento, incluyendo construcción de modelos, validación y operación. Sin embargo, existen grandes incompatibilidades que obstruyen la verdadera integración. Los GIS manejan datos estáticos y discretos mientras los modelos ambientales tratan fenómenos dinámicos y continuos. Las BD GIS manipulan datos en términos de *ubicación*, distribución y relaciones espaciales, mientras que los modelos ambientales trabajan con la transferencia masa y energía. Para lograr una integración total de ambos necesitamos agregar el dinamismo y la continuidad a nuestro concepto

de datos espaciales; e interacción espacial y funcionalidad a los modelos ambientales.

Este trabajo de investigación apunta a la primera de estas necesidades, considerando lo que implica trabajar con fenómenos continuos directamente en el contexto de GIS. Se tendrá en cuenta cómo los fenómenos espacialmente distribuidos se caracterizan en los modelos ambientales, una estrategia para trabajar con dichos fenómenos de manera que puedan ser eficientes y convincentemente incorporados en aplicaciones GIS.

Ya que la mayoría de los fenómenos ambientales tratan con fenómenos que son continuos en el espacio, es necesario proveer formas en las que los modeladores puedan trabajar directamente con los fenómenos continuos, en vez de verse forzados a trabajar con datos espaciales discretos. Debemos reconocer que los datos espaciales son *representaciones* de la realidad, no la realidad en sí, por lo que deberíamos concentrarnos en lograr un análisis que no dependa de la representación, sino de la realidad en sí (*frame free analysis*).

1.1 Modelos Ambientales

Los modelos matemáticos existen desde el desarrollo de la matemática por parte de los chinos, griegos e hindúes, pero los modelos han progresado rápidamente desde los 50's como resultado del desarrollo de las computadoras y los lenguajes de programación. Mientras que los modelos por computadora se hicieron populares inicialmente con los científicos dedicados a la física, quienes tenían teorías definidas y desarrolladas formalmente para trabajar; el uso de tales modelos se ha trasladado gradualmente a la biología, medicina, recursos naturales, desarrollo urbano, ciencias sociales y ciencias ambientales.

Se pide a la gente que trabaja en la toma de decisiones ambientales que justifique estas decisiones, y los modelos físicos son particularmente atractivos para lograr este propósito. A diferencia de los antiguos modelos de caja negra, estos modelos intentan describir procesos físicos reales y las interacciones que ocurren en el ambiente. La utilización de modelos físicos se irá incrementando en el futuro, ya que son “mejores” modelos en el sentido de que tienen bases teóricas más rigurosas.

Mientras los modelos son, por definición, simplificaciones de la realidad, a menudo se asume en *management* que son precisos y que incluyen todos los factores relevantes. Es difícil contradecir la conclusiones (resultados) del modelo sin tener un conocimiento profundo de éste. Los encargados de la toma de decisiones necesitan saber qué suposiciones implica el modelo y cómo éste es afectado por los cambios de las variables. Esto ayuda en la determinación de qué elementos son importantes y cuales pueden ser ignorados. También es necesaria información acerca de la validez del modelo y la información acerca de qué datos fueron utilizados, cuál es su origen y cuándo fueron generados, que variables son medidas y cuales son estimadas.

A pesar de su creciente popularidad, existen muchas dificultades al desarrollar modelos físicos espacialmente distribuidos. Además de problemas como la creación de soluciones aceptables a ecuaciones continuas no lineales y la necesidad de crear discretizaciones del espacio para procesos iterativos, la adquisición de datos exactos y significativos para estos modelos es de fundamental importancia para asegurar la validez de los resultados. Estos modelos requieren cantidades masivas de datos físico-espaciales. Pero desafortunadamente la cantidad no necesariamente asegura que sean provistos los datos que necesitan los modelos ambientales. A medida que los modelos se

vuelven más sofisticados y sensibles, los modeladores piden más y “mejores” datos. Entonces los modeladores se ven forzados a inventar formas creativas para transformar los datos que ya tienen en los que necesitan. Ya que el modelamiento es una actividad creativa, es casi imposible anticipar que datos necesitarán los modeladores.

Cada nueva fuente de datos (data source) lleva a un incremento en el esfuerzo para modelar, a una demanda mayor en la recolección de datos y, posiblemente, a un mal uso de los datos disponibles. En los primeros tiempos de desarrollo de los modelos matemáticos digitales, los científicos tenían que tratar directamente con los datos; recolectarlos, verificarlos e interpolarlos. Hoy en día, existen enormes cantidades de datos disponibles. Mientras que la posibilidad de errores de tipeo es pequeña (comparadas con la cantidad de datos manipulados), se incrementa la posibilidad de mal uso o uso incorrecto de la calidad de los datos. Las interfaces entre los modelos y las bases de datos espaciales generalmente tienden a usar métodos de transformación de fuerza bruta, en los cuales los datos de origen son convertidos sin pensar en las consecuencias que puede traer esa manipulación. Por lo tanto se necesita el desarrollo de bases teóricas sensatas para el uso de datos espaciales en los modelos ambientales.

Afortunadamente, la base de este trabajo ya ha comenzado a realizarse. Décadas de trabajo de los geógrafos nos han provisto de muchos conceptos y herramientas para la manipulación de datos espaciales a través de operaciones numéricas. Los geógrafos ya han estudiado las características espaciales y las propiedades abstractas relacionadas con la topología y la geometría. El *framework* topológico-geométrico de la realidad se relaciona directamente con los modelos de datos actuales disponibles en los GIS. La sensatez de esos principios puede ser comprobada mediante el frecuente redescubrimiento por parte de numerosos equipos dedicados al modelamiento ambiental, que construyen interfaces entre la realidad y los datos espaciales para sus proyectos específicos.

1.2 La Continuidad en los Modelos Ambientales

Dado que los modelos ambientales basados en la física dependen de principios físicos, la matemática detrás de ellos a menudo tiene la forma de ecuaciones diferenciales. Estas ecuaciones implícitamente suponen la continuidad del espacio y el cambio constante de los valores de variables independientes. El desafío para los científicos y los analistas de GIS que trabajan con modelos ambientales es transformar esa continuidad para que tenga cabida en el mundo discreto de las computadoras.

Existen representaciones discretas para las ecuaciones continuas y la continuidad del espacio, y son ampliamente utilizadas. Los métodos numéricos de diferencias finitas sirven para resolver ecuaciones diferenciales, discretizando el tiempo y el espacio en pequeñas unidades. Estas soluciones algebraicas para tratar las ecuaciones diferenciales son calculadas para unidad de espacio-tiempo, y el resultado final se obtiene integrando los resultados parciales (generalmente a través de una adición) en toda el área de estudio. El método de elementos finitos divide el área de estudio en unidades que son homogéneas de manera que permite simplificar algunos términos de la ecuación dominante. Las soluciones analíticas pueden ser determinadas para cada elemento y la solución total es calculada a través de la solución simultánea de un conjunto de ecuaciones. Como alternativa algunos de los modelos globales del clima discretizan realizando análisis espectral. En este caso, en lugar de discretizar el espacio, el espectro es disectado en un conjunto de ecuaciones ordinarias elementales para las cuales se puede encontrar solución.

Tal como las ecuaciones en los modelos matemáticos son continuas, la mayoría de los fenómenos que describen también lo son. Por ejemplo, la temperatura del aire y la radiación solar son campos físicos continuos. Ya que no podemos medir los fenómenos continuos en todos los lugares, es necesario desarrollar técnicas para obtener información acerca de los campos recolectando datos en un número finito de puntos. De la misma manera, necesitamos técnicas que nos permitan representar continuidad con esos conjuntos de datos finitos. Pocas variables ambientales, particularmente aquellas en las ciencias biológicas que se refieren a individuos, como árboles o animales, no son continuas en un sentido estricto. Conceptualmente, tales fenómenos pueden ser convertidos en campos tomando el límite del valor del fenómeno (en este caso la cantidad o frecuencia) dividido por el área que tiende a cero. Este proceso produce como resultado una superficie de densidad continua. Los modelos que usan tales variables calculan el promedio de cambio de la densidad. En el ambiente urbano, se utilizan conceptualizaciones continuas de fenómenos discretos, como modelos de sistemas de transporte. Ciertos fenómenos discretos de los que se mide la densidad deben ser tratados con cuidado. Por ejemplo si estamos modelando la cantidad de individuos en cierta área, esto se puede hacer con valores binarios asociados en cada punto (indicando la presencia o ausencia de un individuo). Pero la densidad no puede ser medida de esta manera, pues también depende de la unidad sobre la que es medida (por ejemplo, individuos por km^2), por lo tanto todas las medidas de densidad llevan una escala implícita. Ya que el valor de tales campos cambia de acuerdo al área de observación, los campos de densidad no son verdaderos campos continuos físicos y no pueden ser modelados de la misma manera.

La continuidad, por supuesto, también existe respecto al tiempo. Como el espacio, el tiempo es difícil de discretizar. Es bastante común en muchas áreas de la matemática discretizar el tiempo como una sucesión de “imágenes” o “fotos” del estado del fenómeno. Ya que la continuidad en el tiempo puede ser tan importante como la continuidad en el espacio, muchos investigadores están trabajando en nuevos modelos de datos temporales. Sin embargo, para el propósito de este trabajo de investigación, se asume que la continuidad del tiempo es adecuadamente representada por alguna de las discretizaciones temporales tradicionales.

Finalmente, la continuidad también surge en las medidas utilizadas. Muchos fenómenos ambientales son medidos usando escalas continuas. Temperatura, radiación solar, precipitaciones, pueden ser medidos con tantos dígitos decimales como lo permitan los instrumentos. La continuidad es una característica fundamental de todos los sistemas naturales y debe ser cuidadosa y explícitamente tratada siempre que un fenómeno natural sea transformado en una representación digital.

Capítulo 2

Medición y Representación de Campos

2.1 Medición y Representación de Valores y Ubicaciones

2.1.1 “Las escalas de medición”

Dado que no podemos enumerar la infinita variedad de la naturaleza como un conjunto infinito de “estados” [Cas89], nos vemos forzados a concentrarnos solamente en un pequeño subconjunto de estos estados cuando modelamos un sistema de este tipo. Los estados que elegimos incluir dependen del fin, los intereses y, mayormente, de las herramientas de medición que tengamos disponibles. La medición, se define como la asignación de números a objetos o eventos de acuerdo a ciertas reglas. Las herramientas de medición permiten establecer estas reglas para asignar números a estados abstractos (“observables”). Estos números luego pueden ser usados como un modelo para representar aspectos del mundo empírico. En cualquier situación hay una variedad de maneras en las que podemos medir estos estados abstractos y, como resultado, tenemos diferentes sistemas de medición. La manera en que midamos la realidad afectará a la forma de modelarla. A continuación se mostraran los distintos tipos de sistemas de medición disponibles para modelos ambientales. Cada conjunto de datos espaciales que se incorpora al modelo debe incluir alguno de estos sistemas.

Las “escalas de medición” [Ste46] dividen a las formas en que podemos medir un fenómeno en cuatro sistemas. Las dos primeras formas son categóricas, ya que a cada observación se le asigna una categoría o clase de un conjunto finito, generalmente pequeño, de estas categorías o clases. Estos valores sirven como identificación que se usa para asignar al fenómeno un nombre o clase. Aunque se pueden utilizar símbolos matemáticos para identificar estas clases (ej.: 1 = rocoso, 2 = arcilloso), por definición, estos valores de clases no se pueden utilizar en expresiones matemáticas. Es decir, no tienen un valor matemático explícito. Si los valores no tienen un orden inherente, si los nombres solo sirven simplemente como una forma de distinguir una entidad o sus características de otra, la medición es *nominal*. Las mediciones binarias son un caso especial de las nominales en donde el número de casos es igual 2. Los sistemas binarios se usan generalmente para indicar condiciones opuestas como alto/bajo, existe/no existe, dentro/fuera de un área de interés. Los valores en un sistema *ordinal* identifican a un conjunto ordenado de clases (ej.: 1 = bueno, 2 = medio, 3 = pobre). Algunos ejemplos de fenómenos medidos en sistemas nominales y ordinales son:

- nominal: tipo de suelo, textura de suelo, tipo de roca (binario: existencia de capas duras, alta concentración de sales)
- ordinal: clase de drenaje, erosión potencial

Como se indicó anteriormente, si bien los valores basados en sistemas de medición categóricos no pueden ser usados en ecuaciones matemáticas, pueden ser usados en programas de computación en alguna de las siguientes maneras:

- para seleccionar un determinado proceso,
- para seleccionar un sustituto real en una tabla de búsqueda, o
- (raramente) como un parámetro (solamente en conjuntos ordenados).

Las observaciones en donde se usan sistemas de medición basados en números reales son fundamentales para los modelos matemáticos. Existen dos tipos de sistemas de numeración de números reales: sistemas de intervalos que tienen un rango de posibles valores en entre $[-\infty, \infty]$ y sistemas proporcionales con un rango mas limitado de $[0, \infty]$. Frecuentemente, los sistemas proporcionales son vistos simplemente como un subconjunto de los intervalos. Sin embargo, hay algunas diferencias críticas entre estos sistemas de medición cuando se deben realizar operaciones aritméticas. Por ejemplo, mientras que es posible sustraer 15 m. de una elevación de 10 m. para obtener -5 m. (sistema de intervalo), no es posible sustraer 15 m. de 10 m. de agua (sistema proporcional). Algunos ejemplos de fenómenos espaciales que son medidos usando números reales son:

- intervalo: temperatura, elevación, humedad
- proporcionales: lluvias, velocidad de vientos, grado de filtración, pH, NDVI

También hay un número de sistemas de medición con números reales que tienen restricciones adicionales en su uso en ecuaciones matemáticas. Dos de estos sistemas que son particularmente importantes en los modelos ambientales son los sistemas radiales y los vectoriales.

Los sistemas de mediciones radiales producen valores especificados en grados o en radianes y son la unidad de medición utilizada en las coordenadas geográficas basadas en latitud y longitud como así también en mediciones de orientación y ángulos de giro. Dado que estos sistemas son circulares (o semicirculares en el caso de latitudes) su rango esta usualmente limitado y pueden volver a valores iniciales. Por ejemplo, la orientación y la dirección de los vientos son medidas de 0° a 360° siendo 360° igual a 0° , mientras que la longitud varia entre -180° y 180° . Es importante destacar que dentro del conjunto de sistemas radiales, los valores pueden ser o bien intervalos (latitud, ángulo de giro) o proporcionales (pendiente, dirección de un compás).

Los sistemas de medición de vectores producen valores con dos componentes los cuales determinan dirección y magnitud. Hay dos formas de expresar estos elementos:

- dirección (en grados) y magnitud, y
- los componentes x e y de un vector geométrico basado en coordenadas cartesianas.

Cada uno de estos puede ser determinado geoméricamente del otro. Se utiliza generalmente la forma (x, y) para la manipulación matemática. Algunos ejemplos de fenómenos que pueden expresarse como vectores son el flujo horizontal de agua, vientos (observados como dirección y magnitud) y la pendiente del suelo (observado como orientación e inclinación).

Mientras que hay diferencias importantes en como se pueden manipular cada uno de estos sistemas de medición, para nuestro propósito es conveniente clasificar estos sistemas de medición en *categoricos* y *numéricos*. Mientras que los números pueden ser usados libremente en expresiones aritméticas y por lo tanto pueden ser incorporados libremente en modelos matemáticos, los valores categoricos están restringidos a los usos indicados anteriormente. En ciertas situaciones las operaciones entre números de intervalos crean valores inválidos. Sin embargo, hablando prácticamente, los valores de intervalos proporcionales son frecuentemente combinados en ecuaciones matemáticas basadas en principios físicos. En este caso, se incluyen uno o más valores basados en parámetros empíricos, los cuales se asumen que convierten el valor de intervalo a un valor proporcional en las unidades apropiadas.

Los datos (numéricos) radiales también tienen restricciones complejas en las operaciones aritméticas. Los resultados deben conformar al rango de las características circulares de la variable resultante. Por ejemplo, si a y c miden la declinación del norte (rango posible $[0,360]$) y b mide el ángulo de giro (rango posible $(-\infty, \infty)$) entonces si $a=330^\circ$ y $b=85^\circ$,

$$c = a + b = (330 + 85 - 360)^\circ = 55^\circ$$

$$c = b * 10 = (850 - 360)^\circ = (490 - 360)^\circ = 130^\circ$$

O, si a y c miden la longitud (rango posible $[-180,180]$) y b mide movimiento en la dirección este-oeste (rango posible $(-\infty, \infty)$), y valores negativos indican ángulos medidos en la dirección oeste, entonces si $a=-150^\circ$ y $b=70^\circ$

$$c = a + b = -150 + 70 = -80^\circ$$

$$c = ab = -150 - 70 = (-220 + 360) = 140^\circ$$

Por lo tanto dado que solamente se distinguen las mediciones categoricas y numericas, durante la construcción de las ecuaciones matemáticas se asumen las restricciones en el uso de sistemas de medición diferentes, también como las consideraciones normales para las unidades de medición apropiadas.

2.1.2 Ubicación de la Medición

Dado que el espacio es continuo, es posible expresar una ubicación en la superficie de la tierra a cualquier grado de precisión requerido. Así los valores usados para expresar la ubicación son elementos del conjunto de números reales. La ubicación se expresa generalmente en términos de un sistema de coordenadas rectangulares o esféricas puesto sobre la superficie que se examina y se utiliza a menudo (x,y) como método genérico de expresar la ubicación de un punto.

Por supuesto, cuando un sistema de coordenadas rectangular se pone sobre la superficie curva de la tierra, aparecen distorsiones. Las proyecciones de los mapas son las herramientas geométricas que

se utilizan para relacionar las ubicaciones en la superficie de la tierra expresadas como un conjunto de latitud y de longitud (sistema de coordenadas esféricas) con la posición en, generalmente, una grilla rectangular, expresada por un par (x, y) . Afortunadamente para la gran mayoría de los esfuerzos de modelar en los cuales la ubicación es una parte integral, hay disponibles fórmulas que se pueden utilizar para convertir entre los sistemas de coordenadas rectangulares y esféricas. Snyder [Sny87] ha proporcionado a un resumen excelente de las transformaciones de sistemas de coordenadas más importantes.

2.2 Representando Campos

En la representación específica de fenómenos espaciales continuos, es útil establecer el concepto de *campo*. Un campo físico se define tradicionalmente como una entidad que se distribuye sobre el espacio y cuyas características son funciones de las coordenadas espaciales y , en el caso de campos dinámicos, del tiempo:

$$z = f(x, y) \text{ o } z = f(x, y, t)$$

Los campos escalares son caracterizados por una función de la posición y , posiblemente, del tiempo, donde el valor en cada punto es un escalar, mientras que el valor en cualquier ubicación en un campo vectorial es un vector (es decir campos representando vientos donde el valor en una ubicación tiene magnitud y dirección). Goodchild [Goo92] ha sugerido que el elemento fundamental de la información geográfica es la tupla

$$T = \langle x, y, z_1, z_2, \dots, z_n \rangle$$

la cual describe el valor de n variables espaciales en la ubicación (x, y) . Dado que x e y son continuos, se concluye que el número de tuplas es infinito. Así el conjunto infinito de las tuplas $\langle x, y, z \rangle$ describe un campo que contiene los valores de una sola variable espacial sobre el espacio.

Puesto que son continuos, los campos físicos se distinguen por su extremadamente alto grado de autocorrelación espacial. Así, mientras que no podemos medir el valor de un fenómeno continuo en todas partes, sabemos que las ubicaciones cerca de las que podemos medir tendrán valores muy similares. El conocimiento de la autocorrelación espacial, sin embargo, nos da poca información sobre cuan rápidamente e irregularmente cambian los valores entre las ubicaciones en las cuales sabemos el valor. Para representar y manipular campos con modelos matemáticos, debemos tener alguna manera de conectar la variación continua del campo que se observa en naturaleza a los números o a las letras individuales guardados en la computadora como representaciones del valor del campo en ciertas ubicaciones. En algunos casos especiales, los valores y la variación en el espacio se pueden representar mediante una ecuación, por ejemplo:

$$z = x^2 + xy + y^2$$

donde x, y son coordenadas cartesianas horizontales y z es el valor del fenómenos en cualquier ubicación (x, y) . Sin embargo, puesto que las superficies en la realidad raramente son suaves, el acoplamiento entre la realidad continua y su representación en el ordenador se obtiene:

1. dividiendo el espacio continuo en ubicaciones discretas donde los valores discretos pueden ser

medidos y registrados, y

2. estableciendo alguna regla para interpolar valores desconocidos en estas ubicaciones.

El primero de estos pasos se conoce como discretización. El segundo paso se logra con el uso de modelos de datos espaciales. En las secciones siguientes se consideraran estos dos temas en mayor detalle.

2.2.1 Discretización del Espacio

Los geógrafos han intentado por siglos dar estructura a la complejidad espacial de la naturaleza. Incluso cuando la geografía se considera sobre todo como cualitativa, es necesario sin embargo partir el espacio para poder describir regiones y elementos en una manera analítica. Durante la revolución cuantitativa de la geografía, el foco ha dado vuelta sobre el descubrimiento de estructuras geométricas y las técnicas matemáticas que se podrían utilizar para explicar distribuciones espaciales.

Los mapas son una herramienta importante para entender las estructuras espaciales, son reconocidos por muchos geógrafos como un lenguaje especial para la información espacial [Har69]. Los mapas, las líneas y las sombras creados por los cartógrafos se pueden reducir a las primitivas básicas espaciales: punto, línea y área. Tobler llama esto el paradigma cartográfico de los fenómenos geográficos [Tob90]. Dentro del modelo de la realidad geográfica de los mapas, a los fenómenos continuos se les dá la estructura del modelo de punto/ línea/ área con el uso de las líneas de contorno y de otras isolíneas.

Con el desarrollo de la detección remota basada en satélites y de mapas digitales basados en grillas, se hizo popular un segundo modelo formal del espacio, la trama (raster). Se ha generado mucha discusión sobre la representatividad del modelo de trama contra el modelo de punto/ línea/ área o de vector. Muchas compañías de GIS indican que sus productos “son integrados”, es decir capaces de visualizar tramas y datos vectoriales y en algunos casos se realizan traducciones entre ellos.

Desafortunadamente, se ha olvidado el punto más importante. El problema es más fundamental que simplemente el desarrollo de algoritmos para convertir imágenes de trama a sus representaciones de vector. El centro de esta discusión debería ser cuan bien estos modelos representan la realidad que piensan retratar.

De hecho, los simples conceptos de trama y vector son incompletos para trabajar con las variadas representaciones de los fenómenos continuos. Hay más para representar de la realidad que apenas partirla en pedazos. Es útil considerar las distintas etapas de aumento de abstracción implicadas en la construcción e implementación de un modelo matemático a partir de un proceso natural.

Esta investigación se centra en la conexión entre un modelo geográfico específico, el campo, y varios modelos de datos espaciales que se pueden utilizar para representarlo. Ahora volvemos a considerar estos modelos de datos.

2.3 Modelos de Datos Espaciales para los Campos

En un sentido literal, lo mismo que un modelo hidrológico representa hidrológica y un modelo de crecimiento de cultivos representa el crecimiento de cultivos, el termino “modelo de datos” sugiere la idea de un modelo formal para los *datos*, no de la realidad. Es importante reconocer que el proceso de desarrollar modelos de datos espaciales de una realidad específica, llamado *modelado de datos*, involucra la discretización de la variación espacial de esa realidad. Desafortunadamente el modelado de datos se confunde frecuentemente con temas de estructuras de datos [Goo92] y se complica en cuestiones de como se representarían los puntos, líneas y áreas. De hecho, esta confusión de términos puede ser parcialmente por la falta de un entendimiento sobre las diferentes formas en que estos modelos de datos representan la realidad. Cada uno abarca uno o más importantes suposiciones sobre la forma de la realidad representada. Estas suposiciones afectan críticamente la forma en que cada modelo de datos puede ser manipulado matemáticamente.

Hay seis modelos de datos diferentes disponibles para la representación de los campos [Goo92]: grillas de celdas, polígonos, TINs, modelos de contornos, grillas de puntos y puntos irregulares. En las siguientes secciones, definiremos y discutiremos cada uno de estos modelos y describiremos como cada uno modela la realidad. Para esta y futuras discusiones, es útil que definamos el concepto de *elemento espacial*. Estos son los componentes geométricos básicos de los modelos espaciales (ej.: punto, celda (pixel), línea) y que son entidades individuales las cuales son referenciadas y manipuladas por la computadora. Cada uno de estos elementos esta ubicado en el espacio y se le asignan uno o mas valores específicos. El conjunto completo de elementos espaciales en un modelo de datos espacial usado para representar una instancia específica de un fenómeno es un *conjunto de datos*.

2.3.1 Grillas de Celdas

Una grilla de celdas particiona toda el área de estudio en rectángulos, los cuales están uniformemente alineados en dos direcciones perpendiculares. Las celdas que no son regulares (ej.: celdas hexagonales) son considerados modelos de polígonos. El ejemplo mas común de una grilla de celdas es una imagen obtenida remotamente compuesta de pixels. El valor del fenómeno sobre el área cubierta por cada celda (el elemento espacial) es representada por un único valor aun cuando haya una variación considerable dentro de la celda. Como resultado, los valores cambian abruptamente en los bordes de la celda. La geografía de una grilla de celdas puede ser descripta completamente especificando el alto y ancho de la celda, el origen de la celda, la orientación con respecto a un compás de las filas y las columnas y la proyección utilizada.

2.3.2 Poligonos

Los polígonos particionan el toda área de estudio en regiones contiguas de forma irregular. Como en las grillas de celdas, el valor del fenómeno dentro de un polígono particular se define como una constante y cambia abruptamente en los bordes del polígono. Los limites de un conjunto de polígonos se puede definir por el fenómeno (ej: zonas de vegetación) o pueden ser independientes del fenómeno (ej.: bloques de corte, líneas divisorias de aguas cuando se utilizan para particionar las características del suelo). En las bases de datos ambientales, los datos estructurados como polígonos

son por lo general categóricos (ej.: suelos, tipos de vegetación, líneas divisorias de aguas). Para que sean útiles en modelos matemáticos, estas bases de datos categóricas son usualmente vinculadas a tablas relacionales las cuales describen varias propiedades numéricas y de otro tipo para cada clase.

2.3.3 TINs (Triangulated Irregular Networks, Redes Trianguladas Irregulares)

Los TINs particionan toda el área de estudio en regiones triangulares. El valor del fenómeno se especifica solo en los nodos de los triángulos. Sin embargo, dado que se asume que la superficie de cada triángulo es una función de las coordenadas rectangulares, se pueden calcular los valores en cualquier parte dentro de la cara de un triángulo directamente a partir de los valores en los nodos. (Se pueden asumir varias formas diferentes para la variación de esas caras triangulares, pero dado que las caras planares son las más fáciles de trabajar y la mayoría de las implementaciones de TINs comerciales solo permiten caras planares, solo consideraremos esta forma aquí.) Dado que no hay un cambio abrupto en los valores en los bordes planares, hay si un cambio abrupto en la pendiente. La ubicación de los límites se define por la ubicación de los nodos. Así, la correspondencia entre la superficie real y la que es representada por la superficie de los triángulos se determina por el conjunto de puntos (nodos) seleccionados para definir los puntos críticos de la superficie. Dado que los TINs definen superficies continuamente variantes, los modelos de TIN nunca pueden ser usados para estructurar datos categóricos (no numéricos).

2.3.4 Grillas de Puntos

Las grillas de puntos almacenan el valor del fenómeno en cada intersección de una grilla regular. Estos valores representan el valor actual del fenómeno en esa ubicación. La ubicación de cada punto de la muestra lo determina la grilla, independientemente del fenómeno. Si un conjunto de datos con estructura de grilla de puntos se ha derivado de una fuente de datos primarios (ej.: elevaciones de pares estéreos), no se hacen suposiciones correspondientes a la representatividad de cada valor dentro de la vecindad de los puntos tomados. Sin embargo, si el conjunto de datos ha sido derivado de algún otro conjunto de datos espaciales, los puntos pueden ser realmente representativos de la vecindad. Por ejemplo, los valores en una grilla de puntos derivados de una grilla de celdas son representativos de la vecindad de la celda mas que de los valores de los puntos. La geografía de una grilla de puntos debe ser descripta especificando el espaciado x e y , el origen y la orientación de la grilla y la proyección usada.

2.3.5 Grillas Irregulares de Puntos

Los modelos de grillas irregulares de puntos almacenan los valores del fenómeno en ubicaciones puntuales esparcidas irregularmente. La ubicación de los puntos puede ser determinada por el fenómeno. En este caso, los valores se asumen representativos de las ubicaciones vecinas (ej.: ubicaciones representativas cuidadosamente seleccionadas para una colección de datos sobre lluvias).

Sin embargo, los datos de una grilla irregular de puntos pueden ser tomados en ubicaciones determinados por factores distintos al del fenómeno estudiado (ej.: estaciones meteorológicas ubicadas en aeropuertos). En este caso, el valor en cada punto es menos representativo de las condiciones circundantes.

2.3.6 Modelos de Contornos

Los modelos de contornos son únicos entre estos modelos de datos espaciales usados para fenómenos continuos. A diferencia de los otros modelos espaciales, los modelos de contornos se construyen manteniendo el valor del fenómeno constante y determinando su ubicación. Se construyen líneas para conectar ubicaciones adyacentes cuyos valores igualen el valor de la línea de contorno deseada. Este modelo identifica explícitamente todos los lugares que exhiben un valor expresado por una de las líneas de contorno. Sin embargo, el valor de la superficie se define solamente sobre las líneas de contorno. Las ubicaciones de las líneas de contorno se determinan tanto por el fenómeno como por los valores seleccionados sobre los cuales se dibujan las líneas de contornos.

Una característica única de este modelo es que los elementos espaciales son líneas, en vez de puntos o áreas como en los otros cinco modelos. Estas líneas son ordenadas por valor de forma que las líneas de contorno vecinas son o bien de valores iguales o diferentes solamente por un intervalo de contorno. Como los TINs, los modelos de contornos particionan el espacio en regiones sobre las cuales varía el valor del fenómeno. A diferencia de los TINs, la variación entre líneas de contorno no es lineal o claramente definida. La única suposición que se puede realizar sobre la variación entre las líneas es que los valores del fenómeno se mantienen dentro del rango definido por los valores de las líneas de contorno que lo encierran y que las ubicaciones cercanas a las líneas de contorno tienen un valor cercano al de las líneas. Finalmente, como en los TINs, dado que las líneas de contorno deben ser medidas en un sistema de medición continuo, los modelos de contorno nunca pueden ser utilizados para representar datos categóricos.

2.3.7 Características de los modelos de datos espaciales

Habiendo definido y descrito brevemente los seis modelos de datos usados para representar campos, ahora es posible considerar sus diferentes características en una manera global. Goodchild ha sugerido que estos seis modelos representan dos formas distintas de explotar la autocorrelación espacial de los campos [Goo92]. Los modelos *piecewise*¹ hacen uso de la suposición que las ubicaciones cercanas son similares mientras los modelos de *muestreo* explotan el hecho que si sabemos el valor en una ubicación podemos estimar el valor en las ubicaciones cercanas.

Los modelos *piecewise* seccionan la superficie en regiones contiguas. Un valor se define en toda ubicación de la superficie. La variación continua del valor del fenómeno dentro de cada región se describe en función de las coordenadas. En dos modelos, grilla de celdas y polígonos, esta función es una constante mientras que en el modelo TIN la función es lineal. Así si el valor del fenómeno representado es dibujado como una tercera dimensión, la grilla de celdas y los modelos de polígonos producen una superficie escalonada de regiones horizontales, mientras que las regiones del modelo TIN son planos inclinados con los bordes de cada región coincidentes con los de sus vecinos. La

¹Por partes.

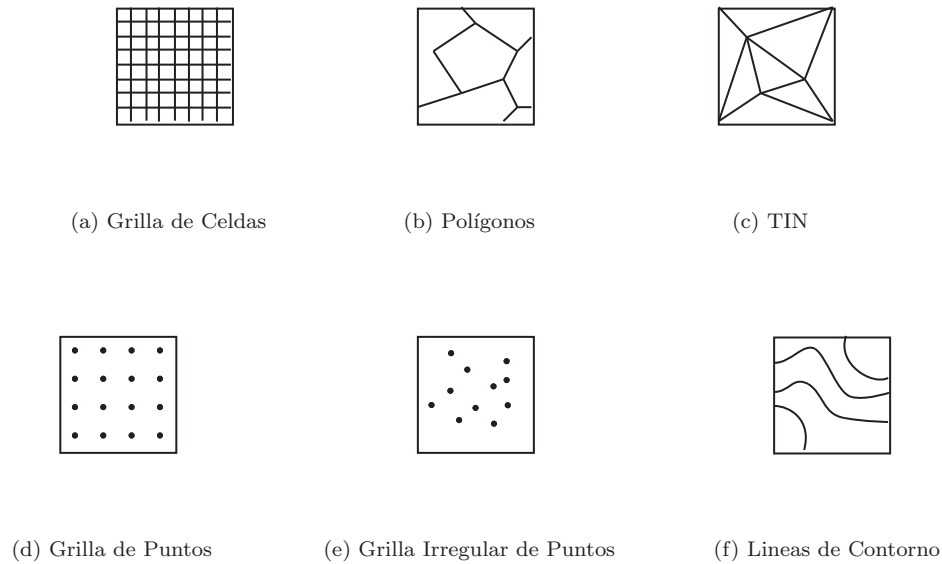


Figura 2.1: Modelos de Datos Espaciales para Campos Continuos

suposición crucial en todos los modelos piecewise es que el valor o función asignado a cada región es representativo del valor promedio o de la tendencia general de la superficie en la región. Mientras que cada punto individual podría no ser representado precisamente, se asume que la integral de los valores sobre esta superficie produciría el valor de la función lineal asignada.

Los modelos de muestreo usan una aproximación completamente diferente. En estos modelos, el fenómeno es precisamente muestreado en un número específico de puntos diferentes. El muestreo se realiza en puntos, como en los modelos de grillas de puntos y de grillas irregulares de puntos, o a través de líneas como en los modelos de contorno. Ningún valor es asignado a ubicaciones que no hayan sido muestreadas y, excepto en el limitado caso de los modelos de contornos, no se proporciona ninguna información sobre la variación en el valor del fenómeno entre los sitios de las muestras. Para representar la superficie continua entre estas ubicaciones de las muestras se debe suponer que la variación entre estos puntos puede ser descrita por una función matemática. Sin embargo, a diferencia de los modelos piecewise, esta función no siempre está claramente definida. Frecuentemente se utilizan funciones lineales, también son comunes otros tipos (ej.: funciones de alto orden que ajustan la superficie exactamente a puntos de una grilla en una ventana de 3×3). La función de interpolación elegida puede variar para el mismo conjunto de datos en diferentes aplicaciones. También, la precisión de una predicción del valor de un punto dado en la superficie varía dependiendo de la distancia a un sitio muestreado, en general, el valor de un punto muy cercano a un sitio muestreado puede ser predicho con mayor precisión que el valor de una ubicación a una mayor distancia.

Los modelos de contorno muestran características combinadas de los modelos piecewise y de muestreo. Mientras que son realmente muestreados en el sentido descrito anteriormente, son muestras completas de todas las ubicaciones con los valores de las líneas de contorno seleccionadas. Esto provee información adicional sobre la variación entre las líneas. Así, desde una perspectiva

piecewise, esto implica la existencia de alguna información sobre la variación dentro de cada región limitada por las líneas. Y, como en los TINs, las superficies contorneadas dibujadas en tres dimensiones muestran superficies que varían suavemente entre las líneas.

Así tenemos dos grupos de modelos con amplias diferencias en sus suposiciones básicas. Mientras que los modelos piecewise proveen una representación generalizada del fenómeno continuo, los modelos de muestreo proveen datos precisos en un número limitado de ubicaciones. Los esquemas de muestreo pueden ser imparciales (como en las grillas de puntos) o parciales (como en los modelos de contornos y de algunas grillas irregulares de puntos). En términos de representación de superficies, es útil considerar a los 6 modelos en tres grupos distintos. Los modelos piecewise constantes describen una superficie horizontal escalonada con cortes verticales en los bordes de las celdas o los polígonos (fig. 2.2). Los modelos de superficie, TIN y de contorno, describen una superficie continua con valores variando dentro de regiones y con continuidad en los bordes (bordes de triángulos o líneas de contorno) (fig. 2.3). Los modelos de punto no describen una superficie continua; se debe utilizar la interpolación para construir estas superficies (fig. 2.4).

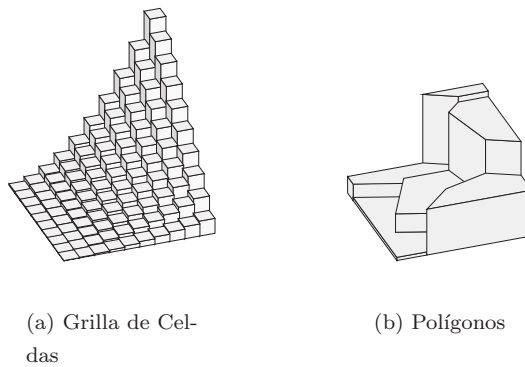


Figura 2.2: Modelos Constantes

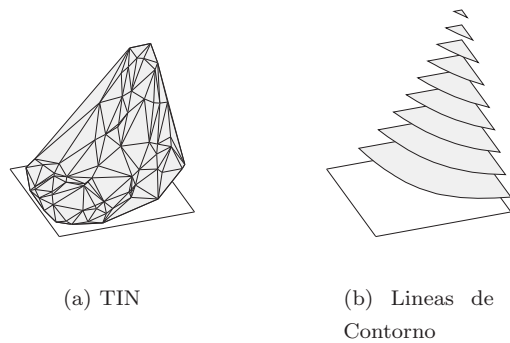


Figura 2.3: Modelos de Superficie

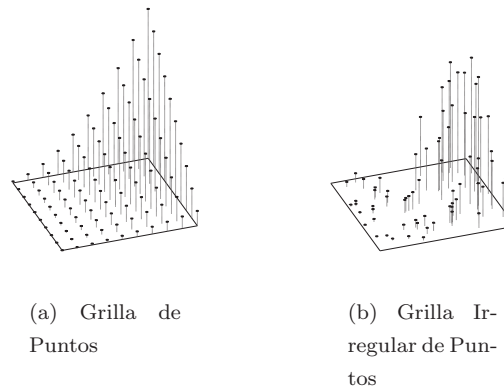


Figura 2.4: Modelos de Puntos

2.3.8 Modelos de datos espaciales como una representación de la realidad

El hecho de cuan bien un modelo de datos espacial representa la realidad es un tema complejo. El error aparece inicialmente en las mediciones del fenómeno y puede incluir malas mediciones o almacenamiento de los valores, las ubicaciones, o ambos.

Si asumimos que la superficie real ha sido muestreada en forma precisa y almacenada, los modelos de datos espaciales son solamente una representación de la realidad. ¿Es posible estimar la precisión con la cual un modelo de datos espacial se ajusta realmente? ¿Como podemos evaluar como encaja una representación discreta de una superficie continua? La respuesta directa a estas preguntas es simplemente, con dificultad. Dado que es continua, la superficie real no puede ser descrita completamente. Por lo tanto el modelo no puede compararse directamente con ella. También el alto grado de autocorrelación espacial entre los valores en los puntos de un campo invalida muchas de las herramientas estadísticas que pueden ser usadas para medir el grado de correspondencia entre el modelo y la realidad.

Otro tema importante relacionado con la variabilidad del fenómeno tiene que ver con la suavidad del modelo de datos espacial usado para representar el campo físico. Mientras que muchos fenómenos pueden exhibir una gran variabilidad de frecuencia, los conjuntos de datos digitales que representan estos campos generalmente solo producen modelos de baja variación de frecuencia. Los modelos constantes piecewise reemplazan esta variación local con el promedio local, mientras que los modelos de superficie describen variabilidad expresando la razón de cambio del valor del fenómeno a través de mediciones de inclinación (es decir la inclinación de los triángulos de un TIN o la cercanía de las líneas de contorno). El como se usan e interpretan estos modelos de la realidad debe ser determinado por la aplicación en la cual serán usados. Por ejemplo, si las pequeñas variaciones en el valor del fenómeno son importantes, es esencial que el modelo de datos espacial sea capaz de representar estas variaciones. Esto se puede realizar a través del uso de elementos espaciales pequeños o el desarrollo de mediciones de variabilidad como las provistas por la aplicación de técnicas geostatísticas. En otros casos, donde las pequeñas variaciones en el valor del fenómeno llevan como resultado modelos inconclusos, se requerirá un modelo de datos que suavice estas altas variaciones de frecuencia mientras que mantenga una buena representación

de las bajas variaciones de frecuencias. Si un conjunto de datos no puede describir un nivel de variabilidad apropiado, entonces se deben buscar algunos medios para introducir esa información. Esto es responsabilidad del modelador, no del conjunto de datos.

Un interés más amplio se da sobre las anomalías que aparecen en la relación entre la realidad y el modelo de algunos conjuntos de datos. Un excelente ejemplo de estas anomalías son aquellos que pueden observarse en modelos de TIN que han sido derivados de modelos de contorno. Debido a los algoritmos usados para crear los TINs a partir de los contornos, un modelo derivado puede mostrar estas anomalías como triángulos llanos (que ocurren cuando se obtienen los tres nodos de un triángulo de una misma línea de contorno) y represas y vallas (que ocurren cuando los triángulos cruzan a través de cerros o valles, los cuales no son capturados por las líneas de contorno) [Kum92]. Si estas anomalías crean problemas depende de la aplicación en donde será utilizado el TIN. Si el TIN debe ser usado para la creación de un relieve sombreado de la superficie, estas características pueden ser extremadamente engañosas. Similarmente pueden tener un gran impacto en los resultados de un modelo hidrológico dado que causaran que se modifiquen las direcciones de flujo. Por otro lado, si el modelo espacial será usado en un modelo matemático donde solamente es importante el valor escalar del punto, los valores interpolados de los triángulos del TIN deben ser similares a aquellos producidos por el modelo de contorno. Así la semejanza entre la realidad y el modelo es en algunos casos un tema de elección subjetiva - ¿se asemeja este modelo a la versión de la realidad del modelador. Y finalmente, en la ausencia de un conocimiento experto sobre el fenómeno representado, debemos caer en las suposiciones básicas de los modelos de datos espaciales que han sido elegidos para modelar la realidad y esperar que esta selección fue encaminada por el deseo de representar la realidad de la forma más precisa posible.

2.3.9 Estructuras de datos para campos que modelan datos espaciales

Como indico Goodchild, las estructuras de datos frecuentemente se confunden con los modelos de datos [Goo92]. La razón de esto es simple - hay una compleja proyección entre los modelos de datos y las estructuras de datos. Si consideramos solo dos grandes categorías de estructuras de datos - raster y vector - la proyección entre modelo de datos y estructura de datos se vería como sigue:

Modelo de datos	→	Estructura de datos
Grilla de Celdas	→	raster
Poligonos	→	vector
TINs	→	vector
Contornos	→	vector
Grillas de Puntos	→	vector o raster
Grillas Irregulares de Puntos	→	vector

Debemos destacar que, aunque existen autores que consideran como modelo de datos a los rasters y vectores, y como estructuras de datos a las grillas de celdas, polígonos, etc. creemos que la categorización realizada por Goodchild es la mas acertada.

Así, el conjunto de datos debe ser almacenado en formato vector, pero puede representar uno de varios modelos de datos diferentes. Para poder utilizar apropiadamente el conjunto de datos espacial, es necesario conocer que modelo de datos espacial se utilizó durante la etapa de modelado de datos en el desarrollo de la base de datos.

2.4 Modelando con variables continuas

Para los modeladores ambientales, diseñar y codificar un modelo matemático es una tarea completamente diferente a la de acceder y manipular datos en un GIS. Por un lado los modeladores pueden utilizar lenguajes algebraicos y de programación bien conocidos y estructurados, siguiendo las reglas validadas y extensamente probadas para la substitución y la solución. Por otra parte, cuando se manipulan datos espaciales para el uso en los modelos, los modeladores solamente tienen para trabajar el lenguaje de un GIS específico. Los procedimientos que deben seguir para conseguir los datos espaciales no se codifican en ningún lenguaje común. No hay reglas comunes aceptadas ni valores por defecto para guiar cómo se utilizarán los datos espaciales en los modelos ambientales. Así, mientras que los modeladores pueden utilizar un lenguaje simbólico común para expresar el desarrollo matemático y para probar así la validez de su acercamiento, no hay una manera simple de expresar las transformaciones y las manipulaciones que se necesitan incorporar a los datos espaciales del modelo. La consecuencia de esto es que es muy difícil evaluar la validez de los datos incorporados en los modelos que se han basado en datos espaciales y, consecuentemente, es difícil evaluar la validez de los resultados del modelo.

2.4.1 Una estrategia para tratar la continuidad espacial

Son necesarias estrategias y técnicas comunes para manejar datos espaciales en todas sus formas sobre fenómenos continuos. Esto proporciona un marco en el cual puedan ser tratados muchos temas relacionados con la representación de fenómenos continuos. Se puede proporcionar un conocimiento de las suposiciones básicas que se incorporan a cada modelo de datos del campo y los medios para expresar las anomalías de estas suposiciones ([Kem97a], [Kem97b]). Esta estrategia debe específicamente:

- permitir la expresión y la manipulación de variables y datos sobre fenómenos continuos en lenguajes simbólicos comunes. Es decir la estrategia debe ser capaz de ser incorporada en implementaciones de lenguajes de programación de modelos ambientales. Esto contrasta directamente con la estructura del lenguaje tipo natural del álgebra de Tomlin [Tom90] y es más favorable al ambiente científico.
- eliminar la necesidad de considerar la forma de la discretización espacial (el modelo de datos) siempre que sea posible. Mientras que creemos que es deseable y posible alcanzar este objetivo para la mayoría de las operaciones, es necesario que se prevea el ingreso de información adicional para algunas operaciones.
- proporcionar una sintaxis para incorporar las operaciones primitivas apropiadas para modelos ambientales con campos pero que no estén todavía disponibles en GIS o lenguajes de

programación comunes. Éstos incluyen operaciones para realizar versiones discretas de “diferenciación” y de “integración” en las variables que representan campos y la incorporación del concepto de campos vectoriales.

- dirigir y permitir el rápido desarrollo de enlaces directos entre los modelos ambientales y cualquier GIS.

En el resto de este capítulo se establecerán los fundamentos de esta estrategia propuesta para manejar datos espaciales continuos en proyectos de modelos ambientales.

2.5 Variables de Campo

2.5.1 Tipos de datos campo y variables de campo

Para poder manipular datos sobre fenómenos continuos espacialmente, comenzamos definiendo el *tipo de datos campo* que se utilizará además de los tipos de datos tradicionales (ej.: float, integer, character, etcétera). Las variables declaradas como tipo de dato campo son *variables de campo*. Las variables de campo son la representación lógica o funcional del concepto de campo. Estas variables son espacialmente continuas y representan los valores del campo durante un solo instante de tiempo. Como otros tipos de variables, los campos se representan con símbolos. Utilizaremos las letras mayúsculas para denotar a las variables de campo. Por ejemplo, el campo de la temperatura se puede representar por el símbolo T o por TEMP.

Para cualquier variable de campo, debe ser posible determinar un valor en cualquier ubicación y estos valores pueden ser diferentes de ubicación a ubicación dentro de la misma variable de campo. Usando un sistema de coordenadas cartesianas, nos podemos referir a la temperatura en una ubicación específica del campo usando la notación $T(x, y)$. Mientras que las coordenadas cartesianas son el valor por defecto, es posible denotar el valor de la variable en un punto usando cualquier sistema de coordenadas. La notación $T(x, y)$ refuerza la noción que el valor en cualquier punto de un campo es una función de su ubicación.

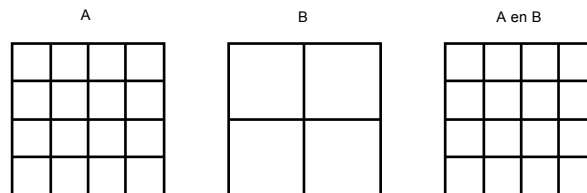
2.5.2 Igualdad espacial y anidamiento

Es útil definir los conceptos de igualdad y de anidamiento espacial pues son esenciales para hacer operaciones matemáticas sobre variables de campo. Estos conceptos se utilizan para comparar las discretizaciones espaciales específicas de diversas variables de campo. En variables de campo espacialmente equivalentes, la geografía de todos los elementos espaciales corresponde exactamente y totalmente. Tal condición se encuentra en las grillas de celdas co-registradas de iguales dimensiones, es decir si A y B son grillas de celdas espacialmente equivalentes (“A como B”), tienen las mismas dimensiones de celda, origen, orientación y proyección. Los polígonos espacialmente equivalentes son menos comunes en conjuntos de datos ambientales puesto que las ubicaciones del límite son determinadas generalmente por los fenómenos observados. Sin embargo, cuando una variable de campo es derivada de otra por la substitución de un conjunto de las clases a un conjunto de valores numéricos, la variable que resulta será espacialmente equivalente a la original.

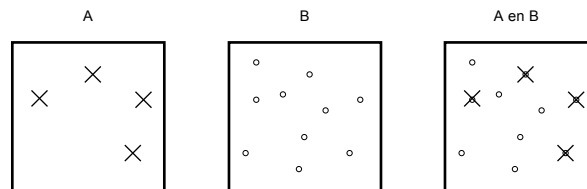
La equivalencia espacial es esencial para la mayoría de las operaciones matemáticas en variables de campo.

El anidamiento espacial indica que una variable espacial anida dentro de otra variable espacial. La definición varía levemente en modelos piecewise y muestreados. Para los modelos espaciales piecewise, si A anida espacialmente dentro de B (“A en B”):

- cada elemento en A reside totalmente dentro de un elemento en B, y
- el conjunto de las líneas que forman los límites de B es un subconjunto del conjunto de las líneas que forman los límites de A.



Para modelos muestreados, el anidamiento espacial simplemente significa que los elementos espaciales de A son un subconjunto de los elementos espaciales de B, nuevamente A esta anidada en B:



El anidamiento se presenta con más frecuencia al trabajar con grillas de celdas. Por ejemplo, consideramos dos grillas de celdas, A y B, con el mismo origen, orientación y proyección pero distintas dimensiones en las celdas. Si A tiene un ancho de celda de 10 segundos y B tiene un ancho de celda de 1 grado, A se anida en B tal que hay 36 celdas de A en cada celda de B.

En este capítulo hemos introducido el concepto de campo y considerado cómo los campos están individualizados y representados en la computadora. Puesto que la manera en la cual se representan los campos es fundamental en la determinación de cómo pueden ser realizadas las operaciones matemáticas, las características asociadas a variables de campo describen el modelo de datos usado y otras características críticas relacionadas con la densidad de la información, los aspectos temporales y el sistema de medición. Estas características son críticas en la determinación de cómo puede ser manipulada matemáticamente dentro de la computadora.

Capítulo 3

Operaciones sobre variables de campo

Hay una amplia gama de operaciones que se pueden realizar usando variables de campo en modelos ambientales. Este capítulo intenta dar cierta organización a esta variedad. Comenzamos con una revisión rápida de algunos de los diversos esquemas de clasificación que se han ideado para ordenar la amplia gama de operaciones sobre datos espaciales que se pueden realizar en los GIS. Esto proporciona un fondo y un contraste útiles para las secciones restantes de este capítulo en las cuales examinamos detalladamente las operaciones que se pueden realizar en variables de campo dentro de modelos matemáticos.

3.1 Operaciones GIS

La investigación en GIS se concentra en las características especiales y los problemas de manipular datos espaciales. Es útil, por lo tanto, comenzar con la perspectiva de los GIS. Consideraremos tres acercamientos contrastantes para ordenar la amplia gama de operaciones que se pueden realizar sobre datos espaciales y que pueden estar disponibles en los GIS que aquí se consideran.

3.1.1 Álgebra de mapas (Tomlin)

Mientras que el acercamiento del álgebra de mapas de Tomlin [Tom91] fue diseñado para manipular solamente datos espaciales en modelo de datos de grilla, su esquema de organización fundamental ha encontrado una amplia aceptación. Tomlin ordenó en cuatro clases generales las varias operaciones que se pudieron aplicar a diversas capas de datos (que pueden ser diversos atributos o diversos instantes de tiempo) - locales, focales, incrementales y zonales [Tom90].

- Las operaciones locales funcionan con valores de una sola ubicación (una celda) a través de de diversas capas.
- Las operaciones focales utilizan los valores de una vecindad alrededor de una sola ubicación.

- Las operaciones incrementales permiten la extensión de la operación a las celdas vecinas que exhiben un atributo conectado, tal como el sentido o posición del flujo a lo largo de una característica lineal. Esto prevee la consideración de vecindades anisotrópicas más grandes.
- Las operaciones zonales funcionan en todas las ubicaciones dentro de la misma zona (clase).

Este esquema de clasificación acentúa la diferencia entre las operaciones que se realizan en los datos para una sola ubicación y las que relacionan a la vecindad definida por proximidad, distancia y dirección.

3.1.2 Las categorías de Raper y Maguire

Raper y Maguire [RM91] han identificado cinco categorías funcionales importantes para las operaciones de GIS. Las cinco categorías son:

- recolección de datos, transferencia, validación y edición
- estructuración de los datos
- manipulación de datos - reestructuración, generalización y transformación
- análisis y consulta
- presentación

Este acercamiento ordena las operaciones funcionales contenidas dentro de los GIS típicos, clasificándolos según la secuencia de los pasos que se pudieron realizar durante la implementación de un proyecto de GIS en particular.

3.1.3 Clasificación de Burrough

El acercamiento de Burrough de es mucho más teórico y tiene una aplicación más amplia [Bur92]. Ha construido 9 clases de operaciones GIS:

- Las operaciones de la clase 1 derivan nuevos valores de los valores exactos de objetos discretos e incluyen operaciones aritméticas y booleanas y métodos de clasificación (taxonomía numérica).
- Las operaciones de la clase 2 producen valores no-exactos de valores exactos de objetos discretos e incluyen métodos estadísticos y de regresión de estimación de valores.
- Las operaciones de la clase 3 derivan nuevos valores del objeto de los valores de ubicaciones dentro de la vecindad de objetos discretos. Están incluidas las operaciones de adyacencia, de conectividad y de proximidad.

- La clase 4 son también operaciones de vecindad, pero se refieren a operaciones de vecindad en superficies continuas e incluyen filtrado, índices de variación espacial, cómputo de derivados (pendiente, aspecto, redes de drenaje), interpolación y adaptación de superficies.
- Las operaciones de la clase 5 son lo contrario de la 3 y 4 puesto que asignan a las localizaciones en su vecindad los valores basados en el valor en la ubicación original usando para ello *buffer zones* y de las operaciones de punto-en-polígono.
- Las operaciones de la clase 6 crean nuevos objetos espaciales usando superposición, buffering, cálculo del centroide y suavizado.
- Las operaciones de la clase 7 derivan valores basados en atributos geométricos de los objetos bajo estudio e incluyen la medición del tamaño, de la forma y de la topología
- Las operaciones de la clase 8 producen informes sumarios incluyendo histogramas, conteo de ocurrencias y *cross-sections*.
- Las operaciones de la clase 9 son para gerencia de datos e incluyen la rectificación, cambio de proyección y funciones de unión.

Este sistema de clasificación hace una cierta distinción entre las operaciones realizadas en datos que representan campos continuos y en los datos que representan objetos discretos. Sin embargo, solamente la clase 4 se ocupa explícitamente de operaciones en superficies continuas. El resto de las operaciones, si van a ser realizadas en datos que representan fenómenos continuos, requieren datos discretos como entrada de información. Las operaciones continuas deben ser aproximadas.

3.2 Matemática sobre campos

Mientras que la clasificación de Burrough es la más cercana al describir las operaciones matemáticas que se pueden realizar en los datos espaciales, no trata el tema desde la perspectiva de la manipulación matemática tradicional. Ahora volvemos a considerar las operaciones que se pueden realizar en datos espaciales desde la perspectiva matemática, dejando atrás el acercamiento tradicional de los GIS.

Hemos observado anteriormente que la computadora es incapaz sumar dos campos continuos para producir un tercero. Todos los campos se deben reducir a números finitos antes que pueda proceder la manipulación matemática. Esta es la función de los modelos de datos espaciales de fenómenos continuos. Sin embargo, hay una complicación adicional. Para manipular dos campos simultáneamente (como en la suma o multiplicación), las ubicaciones de los números finitos simples que representan el valor del campo deben corresponder. Para sumar el campo A al campo B, uno debe sumar el valor de A al valor de B en la misma ubicación. Diversos modelos de datos espaciales expresan la ubicación de maneras que son generalmente incompatibles. Esto implica que para realizar operaciones matemáticas en datos en varios modelos de datos espaciales, debemos primero convertir todos los modelos espaciales a uno equivalente, o por lo menos extraer las estimaciones de los valores para las localizaciones en una variable de campo para la cual tengamos datos en la otra variable de campo. Esta condición se puede expresar los más directamente posible en “=” o la operación de asignación del álgebra tradicional. Comenzamos, por lo tanto, con una consideración detallada de la asignación y después examinamos otras operaciones matemáticas importantes en la construcción de modelos matemáticos.

3.2.1 Asignaciones:

Esta operación es la más importante de todas las operaciones matemáticas. Por definición, todas las ecuaciones matemáticas requieren de la asignación. En los lenguajes de programación estándares, las declaraciones de asignación como por ejemplo $(A=B)$ o $(A:=B)$ o $(A\leftarrow B)$ substituyen el valor de la variable del lado izquierdo por el valor de la variable del lado derecho. Si el tipo de las dos variables no es igual, se realiza una conversión para colocar el valor de la variable del lado derecho en el tipo de los datos requerido por la variable del lado izquierdo. Una convención similar se debe mantener aquí.

Como con variables escalares simples, la versión conceptual de la operación de asignación para los campos es simple. Si B es el campo de la temperatura y $A = B$, entonces A es una copia del campo de la temperatura. Cada ubicación tiene el mismo valor en A como en B . Pero en la computadora hay un numero de diferentes maneras de representar campos. Si A se declara con un modelo de datos distinto al de B , entonces puede ser posible que un valor que fue especificado en una ubicación determinada en A no este precisamente especificado en la misma ubicación en B (ver Figura 3.1). Así la asignación, la operación matemática simple y fundamental, se convierte en una operación espacial compleja cuando están implicados los campos. Requiere la conversión de un modelo espacial de datos a otro. Puesto que cada modelo proporciona una representación de la realidad distinta, es importante enfrentar estas diferencias directamente durante la operación. Sin embargo, nuestro argumento es que es posible codificar estas diferencias de una manera tal que las decisiones con respecto cómo convertir un modelo a otro se puedan manejar automáticamente, sin la entrada de información del modelador. En la sección siguiente discutimos los temas que determinan cómo deben ser hechas estas conversiones y presentar un esquema para ordenar y seleccionar los procedimientos apropiados.

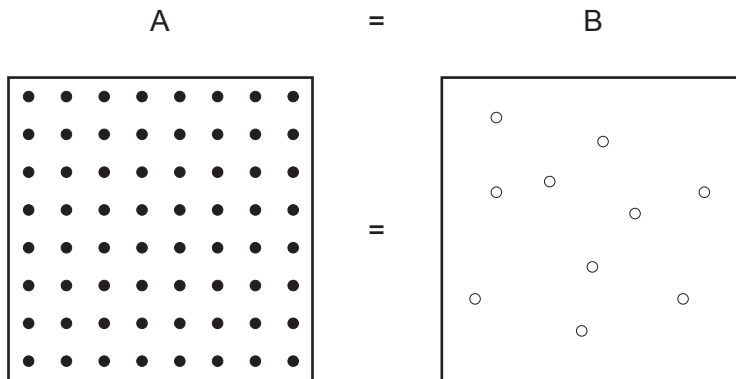


Figura 3.1: Asignación de variables de campo.

Se debe observar que se está utilizando el término *conversión* en vez del término (posiblemente más común) *transformación* para describir este proceso de la asignación modelo-a-modelo. El término transformación es a menudo sinónimo de la función y se utiliza comúnmente referir a

cambios en sistemas coordinados. Tales transformaciones son generalmente invertibles; es decir una transformación se puede invertir para que los datos originales vuelvan sin ningún cambio o degradación en los datos o la pérdida de información [Tob79b]. En el caso de modelos de datos espaciales, las conversiones o las transformaciones se realizan que son raramente inversibles puesto que conducen generalmente a una pérdida de información. Por ejemplo, aunque se pueden extraer de un TIN los valores exactos para un conjunto de puntos dado, no es posible volver al conjunto de puntos del TIN original si los nodos del TIN no son parte de la estructura de puntos. (Una excepción es cualquier proceso de conversión que hace mas densa la estructura sin cambiar el modelo de datos espacial.) Por lo tanto, preferimos el término conversión para acentuar que el modelo también como el contenido de información, en la mayoría de los casos, se cambia permanentemente por una de estas operaciones.

3.3 Convirtiendo modelo de datos de campos

La selección de técnicas apropiadas para convertir modelos de datos de campo a otros modelos de datos de campo requiere que consideremos varios temas. La más importante es la consideración de cómo cada modelo representa la realidad. En un capítulo anterior, fueron repasados seis modelos y fueron descriptas las maneras de modelar la continuidad aprovechando la auto-correlación espacial. Los modelos difieren en las suposiciones que se deben hacer para derivar la superficie continua de la representación discreta pero cada uno provee una cierta conexión con la realidad. Para convertir modelos, debemos explotar la conexión de cada modelo con la realidad para extraer los datos de un modelo y ponerlos en otro. Este proceso se puede realizar en dos etapas. Primero debemos derivar una superficie continua del modelo original de datos espaciales discreto, después debemos utilizar una técnica apropiada para muestrear la superficie continua para producir el modelo deseado.

La derivación de una superficie continua de una representación discreta implica la *interpolación espacial*. Goodchild ha definido a la interpolación espacial como la tarea de computar una superficie continua a partir de un conjunto de puntos de muestra [Goo92], aunque Tobler sugiere que también se incluya al cómputo usando cualquier otro modelo espacial de datos usado para representar los fenómenos continuos [Tob88]. Aquí definimos a la interpolación espacial como *el conjunto de reglas para obtener un campo completo a partir de un modelo de datos espacial*. La interpolación espacial tiene una larga historia. Siempre ha sido una importante herramienta para los geólogos que deseaban interpretar las limitadas pistas que recogían de la superficie de la tierra. Una de las primeras cosas que los geógrafos han aprendido es cómo interpolar elevaciones de mapas de contorno. Con la rápida extensión del uso de las computadoras para la investigación científica y el desarrollo de las matrices de elevación y de los mapas de contorno digitales a fines de los años 60, la interpolación se convirtió en un procedimiento de cómputo importante y extensamente examinado. Existen muchos acercamientos y algoritmos para la interpolación y la conversión entre diversos modelos de datos espaciales. Se pueden encontrar excelentes revisiones de los métodos de interpolación espaciales en [Sch76], [Lam83] y más recientemente en [Bur86].

El muestreo puede ser definido similarmente como *un conjunto de reglas para obtener datos espaciales a partir de un campo*. Estos dos procesos juntos, interpolación espacial y muestreo, se consideran como resampling [Tob88]. Partiendo al resampling en estas dos etapas, nos aseguramos que es mantenida la conexión con realidad y que la representación final es la más cercana posible porque tenemos una mejor suposición de la realidad.

Al realizar la conversión espacial del modelo de datos es necesario considerar el tipo de dato implicado. Los datos numéricos se miden en una escala continua y permiten así la derivación de los valores que caen entre éstos en el conjunto de datos original. Puesto que los valores en los conjuntos de datos categóricos son conjuntos pequeños, de elementos finitos de clases discretas, los valores derivados deben también ser miembros de estos conjuntos y no se puede crear ningún valor nuevo en una operación de conversión. Esta diferencia fundamental sugiere diversos acercamientos para que la conversión de los modelos de datos espaciales sean apropiados a estos diversos tipos de datos. También, las operaciones que implican datos categóricos sobre fenómenos continuos se restringen a un subconjunto de los seis modelos de datos espaciales. Los datos categóricos no se pueden almacenar en modelos de superficie - TINs y los modelos de contorno. Los modelos de puntos se pueden utilizar para almacenar datos categóricos, aunque cualquier uso práctico de estos datos requiere la construcción inicial de modelos *piecewise* como representación de la superficie continua.

La siguiente sección discute cómo cada modelo de datos espacial se puede convertir para poder realizar operaciones matemáticas sobre los datos almacenados en diversas representaciones. Esto demuestra un principio fundamental de este trabajo - no se asume ningún modelo común para la representación de campos. Cada campo se puede representar diversas maneras, la opción con la cual se modela debe ser dependiente de muchas cosas, incluyendo el fenómeno que se representa. Así damos un paso importante fuera de los confines de las grillas de celdas del álgebra de mapas.

A continuación se presentarán las ideas que permiten realizar las conversiones entre distintos modelos, pero no se describirán las técnicas concretas, ya que en capítulos posteriores se mostrará cómo implementarlas.

Comenzamos considerando cómo se puede utilizar cada modelo para obtener una superficie continua. Esto es seguido por la consideración de cómo se puede muestrear cada modelo de campo. La conversión de modelos categóricos de datos se trata por separado de la conversión de modelos de datos numéricos. Definimos el modelo de datos *fuentes* como el original o el lado derecho de la declaración de asignación, y el modelo de datos *destino* como el lado izquierdo de la asignación. El término *región* se utiliza para referir a las celdas o a los polígonos. Se debe predefinir la estructura de los modelos de los datos del destino, puesto que, dentro del contexto de realizar operaciones matemáticas en los campos representados por datos espaciales, los modelos serán convertidos solamente si hay una necesidad que sean espacialmente equivalentes a otros modelos. Así mientras que la generalización o el mejoramiento de una TIN o de un modelo de contorno puede ser deseable para propósitos de almacenamiento o visualización, estas conversiones no se consideran en este contexto.

3.3.1 Interpolación de modelos de campos numéricos

La interpolación a partir de un modelo de datos discreto se puede pensar como el proceso de encontrar el valor del campo en un número infinito de ubicaciones. Por lo tanto para determinar el procedimiento apropiado de interpolación para un modelo de datos espacial determinado necesitamos encontrar una técnica que permita que encontremos el valor en cualquier ubicación. Los TINs proporcionan una descripción completa de una superficie, así que solo se requiere una interpolación para construir una superficie continua. Las grillas de celdas y los polígonos también proporcionan una representación continua en el sentido espacial pero los valores cambian abruptamente en los límites de la región. Por lo tanto, no es necesario idear una técnica por la cual pueda

ser encontrado el valor en cualquier localización. Aunque está claro que esta superficie escalonada no es idéntica a la superficie original, podemos utilizar la suposición que el valor dentro de cada región es representante de todas las ubicaciones cubiertas por esa región para sugerir la suposición que el mejor valor en cualquier localización es la de la celda o polígono que lo incluye.

Por otra parte, cerca de los límites de las regiones origen, puede ser más apropiado asignar un valor intermedio entre el valor de regiones adyacentes. Puesto que los modelos *piecewise* constantes representan las superficies continuas en las cuales los valores de los puntos cerca del límite de la región tenderán hacia el valor de regiones vecinas, este acercamiento puede proporcionar estimaciones del valor más exactas en la superficie verdadera. Así, en el caso de la interpolación de grillas de celdas, puede ser razonable calcular los valores del punto como un promedio pesado de la distancia al centro de la celda. Es ciertamente necesario una forma de promedio para derivar los valores de los puntos en la superficie que caen exactamente sobre las esquinas o los límites de las regiones de la fuente:

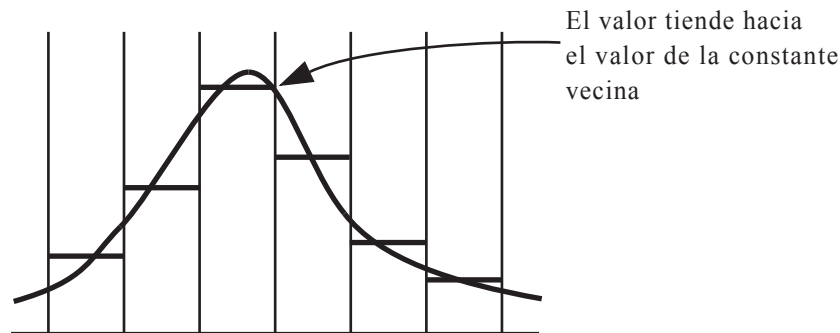


Figura 3.2: Relación entre valores vecinos.

Para los otros modelos de datos, la interpolación de campos implica el encontrar los valores en un número de puntos. Aunque los modelos de contorno describen una superficie que varía suavemente, los valores solamente son sabidos en las líneas del contorno. Hay disponibles muchos algoritmos para encontrar valores entre las líneas de contorno [Sch76], [WH91].

Conceptualmente, el valor en cualquier ubicación en un modelo de contorno puede ser determinado calculando la distancia de un punto a la(s) línea(s) de contorno adyacente(s) y usando la interpolación lineal a lo largo de estas líneas para determinar el valor intermedio que se debe asignar al punto. El valor en el punto será

$$z = \left(\frac{a}{a+b} \right) (y - z) + x = \frac{ay+bx}{a+b}$$

La interpolación de una superficie a partir de puntos (grillas de puntos o grillas irregulares de puntos) tiene una larga historia y hay tantas técnicas para la interpolación como tantas aplicaciones (e.g. vea [Lam83] y [McC88]):

“Se han desarrollado en el pasado numerosos algoritmos para la interpolación de

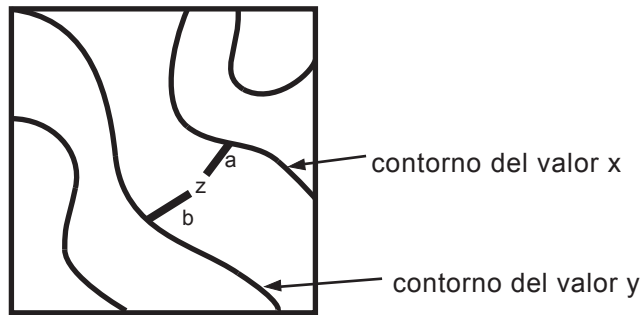


Figura 3.3: Interpolación en un modelo de contorno.

puntos. Pero ninguno de ellos es superior a los otros para todas las aplicaciones, y la selección de un modelo apropiado de interpolación depende en gran parte de los tipos de datos, del grado de exactitud deseado, y de la cantidad de esfuerzo de cómputo producida.” ([Lam83], p. 130)

Es deseable que los aspectos referentes a la interpolación sean configurables de manera que pueda seleccionarse el más adecuado al problema que se desea resolver. La selección también dependerá del fenómeno que se modela, del modelo de datos, del software disponible y del conocimiento y de la experiencia del modelador. Las técnicas incluyen las distancias pesadas, Kriging, *splines*, polinomios de interpolación, series de Fourier y mínimos cuadrados. Por lo tanto, el término genérico, interpolación de puntos, se utiliza aquí para referir al espectro de las técnicas disponibles. La selección de una técnica específica será hecha por 1) la persona que compila los datos y encapsula una técnica apropiada con el conjunto de datos, 2) el modelador que hará una selección basado en el conocimiento del fenómeno, los datos o la aplicación, o 3) el software con limitaciones en las técnicas disponibles.

3.3.2 Muestreando para campos numéricos

Según lo descrito anteriormente, cada uno de los modelos de datos espaciales proporciona una representación de la variación espacial de un fenómeno continuo. Para los modelos *piecewise* constantes, el valor registrado para cada celda o polígono representa el promedio de todos los valores dentro de esa región en el campo. Tal promedio es realizado automáticamente por los sensores remotos con resoluciones espaciales dadas por el tamaño de pixel (celda). Sin embargo, para los campos que deben ser medidos tomando muestras en la tierra, los valores no pueden recogerse en todas partes. Para determinar los valores medios para las regiones en un modelo *piecewise* constante, es necesario determinar el valor medio de una muestra de los valores que caen dentro de la región. Así al generar un modelo *piecewise* constante, necesitamos comenzar con un conjunto denso de valores de puntos en la superficie.

El muestreo de un campo para la representación como TIN requiere la selección de los puntos críticos en la superficie que a través de la triangulación produce planos que se aproximan a la superficie verdadera. Cuando se han medido muchos puntos, la selección de puntos críticos puede

ser hecha automáticamente [Kum92]. La construcción de un modelo de contorno requiere una densa red de valores de puntos para poder trazar las líneas seleccionadas del contorno entre los puntos tan exactamente como sea posible [Yoe84]. Finalmente, crear modelos de campos de puntos es el más directo, puesto que la superficie se muestrea simplemente en cualquier ubicación para el cual el modelo requiera un valor.

3.3.3 Remuestreo del modelo de campos numéricos

Habiendo considerado las dos etapas independientemente, es importante ahora considerar la combinación de la interpolación y del muestreo a partir de un modelo específico a otro puesto que hay ciertos aspectos que requieren la modificación a medida que procede la conversión a través de la representación intermedia de la continuidad. Alguno de los procedimientos descritos anteriormente se pueden o deben modificar, combinar o simplificar para que la conversión proceda lo más directamente posible con la menor pérdida de información. En la siguiente sección, se asume que cada modelo de datos fuente incorpora toda la información disponible sobre el campo verdadero. Así, si remuestreamos un modelo de contorno a una grilla de celdas, por ejemplo, la suposición de la variación constante entre las líneas del contorno nos permite que asignemos valores interpolados entre las líneas a las celdas del destino. Esto puede, de hecho, ser usado para crear datos puesto que es probable que la grilla de celdas tenga varios elementos espaciales entre cada línea de contorno. Sin embargo, si aceptamos las suposiciones inherentes del modelo de contorno, la grilla de celdas destino se convierte simplemente en una representación alterna del mismo campo. Más bien que agregando datos, hemos perdido la información implícita en la localización exacta de las líneas de contorno. Si no se pueden asegurar las suposiciones básicas de un modelo de datos determinado (por ejemplo, un modelo de contorno construido de datos muy escasos), entonces se debe utilizar el conocimiento adicional para determinar el procedimiento apropiado de conversión.

De modelos *piecewise* constantes

De modelos piecewise constantes a modelos piecewise constantes. Dado que las grillas de celdas y los polígonos representan una superficie continua, la creación de nuevos modelos a partir de estos modelos fuente requiere que se cambien los límites de la celda o del polígono (ver Figura 3.4). Se requiere un método para estimar un valor a partir de los valores de un conjunto completo o parcial de regiones en la superficie (escalonada) original. Esto implica el uso de *interpolación de área*. En cuanto a la interpolación de puntos, hay diversas técnicas para la interpolación de área [GL80], [Lam83], [FG92]. Sin embargo, muchos de éstos, incluyendo técnicas que preservan el volumen tales como la interpolación [pyncnophylatic] [Tob79a] o métodos basados en el algoritmo EM [FG92], se han ideado para datos categóricos y datos de conteo y son innecesarios aquí.

De hecho, el *área pesada* simple es suficientemente exacta y conceptualmente consistente para los campos. Puede ser expresado como:

$$z_t = \sum_{a_t} z_{st} a_{st}$$

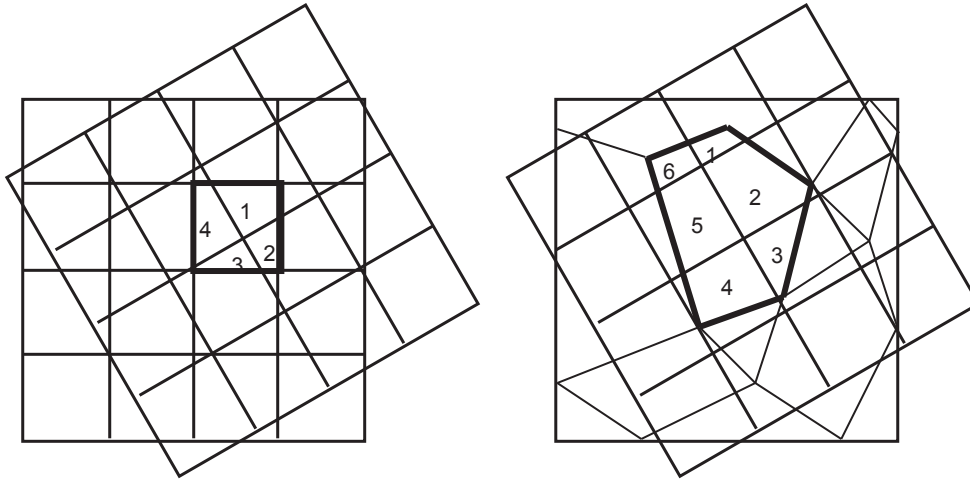


Figura 3.4: Conversión piecewise a piecewise.

donde t es la zona destino, s es una zona fuente, st es la intersección del destino y una zona de la fuente, z es el valor en la zona indicada, a es el área de la zona y

$$z_{st} = z_s$$

De modelos piecewise constantes a modelo de puntos. Con el modelo de grilla de celdas o de polígono como representación de la superficie continua, los valores para los puntos se pueden muestrear directamente de la superficie. Se debe tener presente, sin embargo, que cuando se crea a partir de uno de los modelos *piecewise* constantes, los valores de los puntos no serán indicativos del valor en la ubicación, sino que por el contrario será el promedio en la vecindad. Consecuentemente, los modelos basados en puntos derivados no son conceptualmente iguales a los modelos basados en puntos originales.

Desafortunadamente, hay varias complicaciones con el remuestreo de regiones a puntos. Se discutió anteriormente la disponibilidad de valores pesados por distancia cerca de los límites de las regiones. También, para los modelos en los cuales los polígonos o las celdas son grandes en comparación al espaciado de los puntos, el muestreo producirá grandes correcciones en los valores de los puntos que no ayudan a reflejar la verdadera naturaleza de la superficie subyacente. El uso selectivo de un filtro espacial apropiado puede proporcionar mas suavidad a través de los límites de la región anteriores y producir una representación más realista de la verdadera superficie. La decisión si utilizar un filtro o el acercamiento de distancias pesadas debe depender del conocimiento de la suavidad de la variación en el fenómeno. Por ejemplo (ver Figura 3.5), si la superficie es muy suave, como en caso el A, el filtrado o distancias pesadas pueden ser apropiados. En el caso B, sin embargo, la irregularidad de la superficie invalida las suposiciones subyacentes de la suavidad inherentes en el filtrado y las técnicas de pesado, es claro que son inadecuadas para este tipo de campo. Para este último caso, deben ser utilizadas técnicas alternativas. Se debe utilizar el conocimiento de la forma superficial para determinar el procedimiento(s) apropiado(s) de conversión. Afortunadamente, como será demostrado más adelante, es posible encapsular este

conocimiento en el modelo de datos del origen.

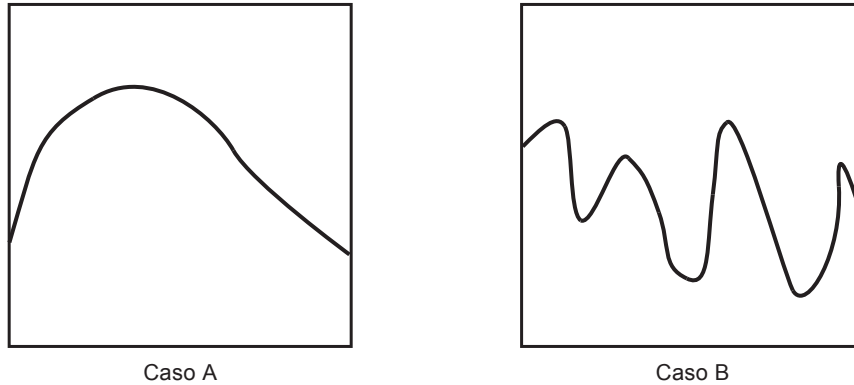


Figura 3.5: Distintos tipos de curva.

De TINs

De TINs a modelos piecewise constantes. Los TINs proporcionan una descripción completa de una superficie continua. Por lo tanto, para muestrear un modelo *piecewise* constante es necesario realizar un promedio del área cubierta por cada región destino (véase la Figura 3.6). Conceptualmente, el promedio del área se haría encontrando el valor medio dentro de cada triángulo en cada región del destino y después realizando el pesado del área. Para encontrar el valor medio dentro de cada triángulo es necesario dividir los triángulos en piezas más pequeñas para los cuales se pueden encontrar el centro y su valor relacionado. Los valores y las áreas de estos triángulos más pequeños se pueden utilizar como la base para el pesado del área.

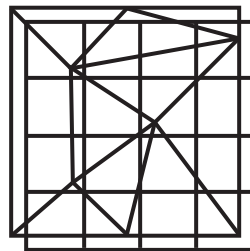


Figura 3.6: Conversión de TIN a piecewise.

De TINs a modelo de puntos. De TINs modelos de puntos permite un procedimiento de interpolación exacto de TIN-a-punto que calcule la ecuación lineal en cada triángulo a partir de las coordenadas rectangulares de los nodos y de sus valores. La ecuación que resulta se puede utilizar para encontrar el valor en cualquier punto que caiga en ese triángulo.

De modelos de contorno

De modelos de contorno a modelos piecewise constantes. Puesto que la superficie representada por un modelo de contorno no se puede describir simplemente por una relación matemática, es necesario recurrir aquí a un modelo de datos intermedio, una grilla de puntos, aunque a través de la cual se realiza esta conversión. Usando un procedimiento de interpolación de contorno-a-puntos, se crea una grilla densa de puntos. Esto proporciona la información sobre la variación del valor del campo dentro de cada región del destino que se puede utilizar para estimar el valor representativo requerido. Esta estimación a partir de una grilla de puntos requiere el uso de un *promedio de los puntos*.

La validez de hacer un promedio de puntos depende del número de puntos que caen dentro de cada región del destino sin embargo no queda claro el mínimo número crítico de puntos que se deben utilizar. En muchos casos, el número de puntos de un conjunto de puntos interpolados que caen dentro de cada región del destino variará dependiendo de la configuración de los puntos interpolados y del modelo de los datos del destino:

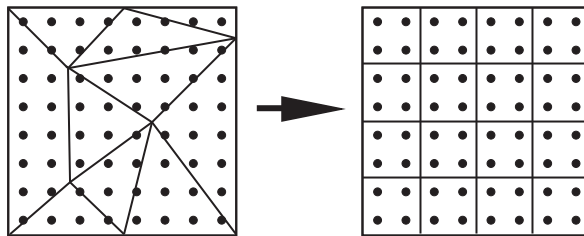


Figura 3.7: Conversión de TIN a piecewise.

Es importante reconocer que el promedio de un número de puntos diferentes para cada región del destino puede afectar la validez estadística del modelo del destino.

Dado un número mínimo específico de puntos que se interpolarán para cada región del destino, se puede determinar el espaciado de la grilla de puntos interpolados directamente de las dimensiones de las celdas o del tamaño de los polígonos. Por ejemplo, asumamos que es suficiente como mínimo 4 puntos. Así, para una grilla de celdas destino, si Δx y Δy son el ancho de las celdas de la grilla en las dos direcciones perpendiculares, la grilla interpolada debe tener espaciamientos de $\Delta x/3$ y de $\Delta y/3$. Para un destino con modelo de polígono, los espaciamientos de la grilla en ambas direcciones deben ser iguales a:

$$\sqrt{\frac{a_p}{4}}$$

donde a_p = área del polígono más pequeño. Para los polígonos que son aproximadamente rectangulares, esto debe poner por lo menos 4 intersecciones de grilla en todos los polígonos. Sin embargo, puesto que es probable que los polígonos no sean rectangulares, será necesario controlar que caigan que por lo menos 4 puntos en cada polígono. Para cualquier polígono sin el número mínimo de puntos, se puede calcular una nueva grilla, más densa para el área cubierta por ese polígono. Mientras que este procedimiento de control de punto-en-polígono es muy intensivo de

computar, existen algoritmos para su solución [Bur86]. Por lo tanto no sería necesario obtener información adicional del usuario.

De modelos de contorno a modelo de puntos. Las conversiones de contorno a puntos utilizan los procedimientos de interpolación de contorno-a-punto descritos anteriormente para combinar la interpolación y el muestreo de la superficie continua descrita por el modelo de contorno.

De modelos de puntos

De modelos de puntos a modelos piecewise constantes. Si el número de puntos en un modelo de puntos es mucho mayor que el número celdas o de polígonos, se puede proceder al remuestreo de los modelos de puntos directamente calculando la media de cada conjunto de puntos que caen en cada región. Sin embargo, si éste no es el caso, será necesario invocar un procedimiento de interpolación que se pueda utilizar para determinar una superficie continua representada por una red densa de valores de puntos que puedan ser promediados dentro de cada región. Según lo descrito arriba, se puede determinar automáticamente la relación entre la densidad del modelo de punto y el tamaño de las celdas o los polígonos sin la entrada de información adicional.

De modelos de puntos a modelos de puntos. En este caso, la interpolación y el muestreo serán hechos simultáneamente interpolando valores de los puntos en el modelo origen para los puntos destino en el modelo del resultado. Se puede determinar el procedimiento apropiado de interpolación por las características del modelo origen así como las densidades relativas del origen y destino.

3.3.4 Interpolando campos de modelos categóricos

Ahora tomamos en consideración la conversión de los campos expresados usando datos categóricos. Mucho de lo que se ha discutido anteriormente se aplica aquí aunque se presentan varias diferencias. Recordando que no todos los modelos espaciales de datos son apropiados para los datos categóricos, esta discusión se restringe a considerar los modelos *piecewise* constantes y los modelos de puntos.

Ya sean numérico o categórico, las grillas de celdas y los polígonos presentan una superficie continua. La naturaleza de bloques de los modelos *piecewise* constantes es la única representación posible de los campos medidos en escalas categóricas puesto que los datos son discretos y no pueden ser continuos entre los límites. Así el campo representado por estos modelos es la aproximación más cercana a la realidad que se puede proporcionar. Cerca de los límites de la zona, puede ser apropiado asumir que ocurre una transición con la clase en cualquier punto siendo determinada por una función binaria de probabilidad. Sin embargo, puesto que la suposición de que las regiones en un modelo *piecewise* representan a la clase más común dentro de la zona, es igualmente apropiado asumir que la clase más probable en cualquier ubicación es la indicada por la clase de la zona en la cual recae.

No se pueden generar campos continuos de datos categóricos puntuales usando cualquiera de las técnicas matemáticas de interpolación puesto que los valores no se pueden “interpolarse” entre las clases. Se deben utilizar técnicas geométricas. Estas técnicas implican el mapeo de proximidad a través de la construcción de polígonos de Thiessen (también conocidos como los gráficos de Voronoi

y de Dirichlet, diagramas o tessellations). En los polígonos de Thiessen se asocia a cada punto de la región del plano al punto más cercano. Asignando el valor del punto a todo polígono que lo incluye, se supone que el punto más cercano conocido proporciona a la mejor información sobre cualquier punto desconocido [Bur86], [Boo86]. Es importante notar que esta suposición puede ser inválida para los campos que se saben que son altamente heterogéneos. Existen muchas implementaciones de los algoritmos para la construcción de los polígonos de Thiessen [AHA92]. Una vez convertido a polígonos de Thiessen, las características de estas representaciones de campos son las de los modelos *piecewise* constantes y las conversiones proceden de acuerdo a ellas.

3.3.5 Muestreado de datos categóricos

Una vez que existen los campos continuos categóricos, se debe realizar el muestreo para las grillas de celdas y los polígonos particionando las zonas del origen a las zonas del destino, el primer paso es el pesado de área. Sin embargo, para estos datos categóricos, no es posible utilizar completamente los procedimientos de pesado de área para determinar un valor representativo para las regiones del destino puesto que los datos categóricos no pueden ser manipulados matemáticamente. Así, para determinar el valor de una región en el modelo del destino es necesario idear un conjunto de reglas que se puedan utilizar para determinar qué clase debe tomar precedencia en la nueva región. Hay varios tipos de reglas que pueden ser invocadas, incluyendo:

- la clase del origen que cubre la porción más grande de la región del destino se convierte en el nuevo valor.
- reglas de prioridad que determinan qué clase toma precedencia si aparece dentro de una región del destino. Este acercamiento puede ser tomado cuando las clases indican el nivel de impacto previsto; si cualquier parte de una región del destino contiene una porción de cualquier región del origen clasificada como de alto impacto, la región entera del destino se clasifica similarmente.
- se conservan las proporciones. Esto requiere una estructura de datos diferente para el modelo del destino que pueda conservar varios pares de valores donde cada par indica la clase y la proporción de la región cubierta por esa clase.

El muestreo de campos categóricos para los modelos de puntos es simplemente una cuestión de determinar la clase del área en la cual cae el punto. No es posible una manipulación matemática.

3.3.6 Remuestreo de modelo de campos categóricos

Las opciones disponibles para el remuestreo de modelos categóricos son limitadas por la restricción en los modelos de datos convenientes. Dado que los modelos *piecewise* constantes dan la mejor representación de la superficie, sólo los modelos de puntos necesitan ser interpolados a través de la construcción de regiones próximas para crear representaciones completas como modelos *piecewise*. El muestreo se realiza mediante el particionado y las reglas descriptas anteriormente o el muestreo de puntos.

3.3.7 Problemas con los campos convertidos

Es importante enfatizar un punto respecto a la veracidad de las conversiones espaciales del modelo de datos. Cada conversión crea un nuevo modelo de la realidad representada. Mientras que un proceso de conversión causa generalmente una cierta pérdida de información, el número de elementos espaciales realmente puede aumentar. Esto puede causar un cambio evidente en la suavidad de la variación y de una adición evidente de datos. Hemos discutido que un aumento en el número de elementos espaciales no aumenta los grados de libertad en el conjunto de datos y que puede ser útil asociar la información relevante sobre el carácter del modelo de datos de la fuente al modelo del destino. Sin embargo, también hemos observado que estos cambios son consecuencias del proceso, y en ausencia del conocimiento experto sobre el campo, no cambian substancialmente qué se sabe sobre la relación entre el modelo y la realidad. Es importante el efecto de tales transformaciones en la representación, pero sus efectos son manejables si se reconocen las características relevantes de los conjuntos de datos.

3.3.8 Las matrices de conversión

De acuerdo con la discusión precedente, podemos resumir estos varios procedimientos en un par de *matrices de conversión*, mostradas abajo en las Tablas 3.1 y 3.2. Debido a que están ordenadas con el fin de una claridad conceptual, los procedimientos descritos en estas matrices se podían implementar como conjunto específico de reglas y de operaciones de decisión dentro de cualquier lenguaje de programación. Observe que en la Tabla 3.1, los TINs y los contornos no se pueden utilizar para los modelos predefinidos del destino puesto que la estructura esta determinada por el fenómeno representado. Las conversiones para TINs y los modelos de contorno presentados aquí se pueden utilizar para reducir el volumen de datos o para propósitos de visualización. Se incluyen aquí por completitud pero se somborean para indicar que no serán utilizadas para manipulaciones matemáticas.

3.3.9 La estructura computacional

Como se dijo anteriormente, las matemáticas continuas requieren una cierta forma de discretización para que el cálculo pueda ser realizado en una computadora. Las ecuaciones diferenciales frecuentemente se solucionan en la computadora con el uso de grillas de diferencia finitas. Las dimensiones de estas grillas de cómputo se definen con respecto a la escala de los procesos bajo estudio. Así los modelos globales de flujo con grillas de 10° de latitud y longitud se refieren a procesos que funcionan en la escala de cientos de kilómetros de mientras que los modelos de erosión del suelo basados en grillas de ancho de celda en el orden de los 100m se refieren a los procesos que funcionan cerca de la escala de la hectárea.

Virtualmente todos los modelos ambientales serán diseñados para la implementación en un modelo de datos espacial específico con dimensiones predefinidas. Ésta es la *estructura de cómputo* del modelo matemático. Puesto que la definición de la estructura de cómputo depende de los procesos que son estudiados y no de los datos espaciales disponibles para el modelador, no hay ninguna expectativa que exista una relación fija entre la estructura de cómputo y las características espaciales de las variables de campo que utiliza el modelo. Como veremos, puesto que ésta es la

De	Grilla de Celdas	Polígonos	TIN	Contorno	Grilla de Puntos	Grilla Irreg. de Puntos
Grilla de Celdas	pesado de area	pesado de area	interp. de punto y triang.	interp. de punto y triang.	muestreo de puntos	muestreo de puntos
Polígonos	pesado de area	pesado de area	interp. de punto y triang.	interp. de punto y triang.	muestreo de puntos	muestreo de puntos
TIN	pesado de area	pesado de area	agregar o quitar puntos	linea de contorno	interp. de puntos	interp. de puntos
Contorno	interp. de puntos y prom.	interp. de puntos y prom.	selec. nodos y triang.	agregar o quitar contornos	interp. de puntos	interp. de puntos
Grilla de Puntos	interp. de puntos y prom.	interp. de puntos y prom.	selec. nodos y triang.	contornos	interp. de puntos	interp. de puntos
Grilla Irreg. de Puntos	interp. de puntos y prom.	interp. de puntos y prom.	selec. nodos y triang.	contornos	interp. de puntos	interp. de puntos

Tabla 3.1: El resumen de datos espaciales modela las conversiones para los datos numéricos

De	Grilla de Celdas	Poligonos	Grilla de Puntos	Grilla Irreg. de Puntos
A				
Grilla de Celdas	partición y apl. de reglas	partición y apl. de reglas	muestreo de puntos	muestreo de puntos
Polígonos	partición y apl. de reglas	partición y apl. de reglas	muestreo de puntos	muestreo de puntos
Grilla de Puntos	partición Thiessen y apl. de reglas	partición Thiessen y apl. de reglas	partición Thiessen y apl. de reglas	partición Thiessen y apl. de reglas
Grilla Irreg. de Puntos	partición Thiessen y apl. de reglas	partición Thiessen y apl. de reglas	partición Thiessen y apl. de reglas	partición Thiessen y apl. de reglas

Tabla 3.2: El resumen de datos espaciales modela las conversiones para los datos categóricos

estructura en la cual serán realizados todos los cálculos matemáticos del modelo, se convierte en la estructura del valor por defecto, hacia la cual tenderán todas las otras durante las operaciones matemáticas y la construcción de operaciones intermedias.

3.4 Operadores Aritméticos

Habiendo examinando la operación de asignación y sus conversiones relacionadas, ahora repasaremos brevemente otras operaciones y funciones que se pueden utilizar en ecuaciones matemáticas. Los campos categóricos se pueden utilizar solamente en aritmética booleana, así pues, a excepción de la sección de las operaciones booleanas, lo siguiente se aplica solamente a los campos numéricos.

3.4.1 Operadores aritméticos binarios

Los operadores aritméticos binarios combinan dos números a través de operaciones simples como la suma, resta, multiplicación y división. Si una variable es un escalar y otra es un campo, el resultado de la operación es aumentar o disminuir los valores del campo uniformemente según la operación especificada. La suma y resta de 0 y la multiplicación y división por 1 crean campos idénticos. El multiplicar por 0 crea un *campo nulo*, uno en el cual el valor en todas las ubicaciones es 0. La división por 0 no se puede realizar. Si ambas variables son campos, estas operaciones aritméticas se pueden visualizar como combinar los valores de las variables para cada ubicación en el espacio:

```
WATER_DEPTH = IMPERV_LAYER_DEPTH - WATER_TABLE_DEPTH
```

En matemáticas abstractas, se pueden realizar operaciones aritméticas binarias sin la restricción de números reales o enteros (la aritmética vectorial tiene diversas convenciones y se trataran por separado), con dos excepciones: la división por 0 que no se permite y la división entera que puede crear un número real.

3.4.2 Operadores Unarios

Los operadores unarios operan sobre un solo valor para crear un valor derivado. Éstos incluyen la negación, el valor absoluto, logaritmo, raíces y el exponenciación. La exponenciación se puede considerar también como un caso especial de la multiplicación que funciona con un solo valor. Con un campo, la negación crea una reflexión de la superficie a través del plano 0, mientras que el valor absoluto refleja solamente los valores que estén encima de 0.

3.4.3 Operaciones aritméticas vectoriales

Se pueden realizar también operaciones aritméticas sobre datos vectoriales. Algunas de estas operaciones se definen como sigue. Si $a = \langle a_1, a_2 \rangle$, $b = \langle b_1, b_2 \rangle$ y c es un escalar,

adición:	$a + b = \langle a_1 + b_1, a_2 + b_2 \rangle$
substracción:	$a - b = \langle a_1 - b_1, a_2 - b_2 \rangle$
multiplicación por un escalar:	$ca = \langle ca_1, ca_2 \rangle$
producto punto o interno:	$a \bullet b = a_1b_1 + a_2b_2$ (a escalar)
producto cruz:	no se puede realizar en 2 dimensiones

Como con las operaciones aritméticas binarias sobre datos escalares, estas operaciones asumen la combinación de los valores correspondientes del vector para cada ubicación en el campo cuando se realizan en datos del campo.

3.4.4 Trigonometría

Mientras que no son estrictamente operaciones aritméticas, igual que las operaciones aritméticas, las operaciones trigonométricas requieren de valores escalares. Los únicos argumentos permitidos en operaciones trigonométricas son los radianes y los grados, que se pueden transformar exactamente de uno a otro. No hay restricciones en la ejecución de operaciones trigonométricas en las variables de campo que se dan en radianes o en grados. Las operaciones trigonométricas son comunes en los modelos que consideran la intensidad de la radiación solar (que usa una variable de campo del aspecto) o régimen de flujo (que usa una variable de la pendiente).

3.4.5 Aritmética booleana

Los campos categóricos y numéricos se pueden manipular con aritmética booleana. Los operadores de la aritmética booleana proveen las herramientas para trabajar con datos categóricos y son útiles para determinar el flujo de las operaciones que se realizarán en un modelo matemático.

3.5 Operaciones que involucran diferentes modelos espaciales

Aun las operaciones simples de la aritmética binaria en variables del campo no son necesariamente simples. Sin embargo, si A y C son variables de campo espacialmente equivalentes y b es un escalar, las ecuaciones:

$$C = A + b$$

$$C = A - b$$

$$C = A/b$$

$$C = A * b$$

realizan la operación especificada en cada valor del conjunto de datos de la variable derecha y pone el resultado en el conjunto de datos de la variable del lado izquierdo. Si las tres variables son variables de campo y espacialmente equivalentes, entonces la aritmética se realiza directamente en los valores de cada elemento espacial y el resultado se pone en el elemento espacial correspondiente de la variable del lado izquierdo. Recordemos que un elemento espacial, según lo definido en el Capítulo 2, es el componente geométrico básico de un modelo de datos espacial (e.j. punto, celda, polígono).

Las dificultades se presentan cuando las variables de campo no son espacialmente equivalentes. En este caso, se debe realizar una conversión para 1) poder realizar la operación en el lado derecho y 2) la respuesta se pueda poner en la variable del lado izquierdo. La pregunta ahora es qué conversión se debe realizar primero. Consideramos el siguiente caso. La estructura de cómputo del modelo es una grilla de celdas. Deseamos desarrollar una variable intermedia para la precipitación mensual. Los únicos datos disponibles son un mapa de contorno de la precipitación anual total y registros de estaciones meteorológicas dispersas detallando los porcentajes de la precipitación anual total de cada mes. Por lo tanto tenemos un modelo de contorno que se debe multiplicar por un modelo de puntos irregulares para crear un grilla de celdas. ¿Convertimos el modelo de contorno al de puntos irregulares, multiplicamos y después convertimos el resultado a la grilla de celdas, o convertimos el modelo de contorno y el de puntos irregulares a la estructura de cómputo antes de la multiplicación? Es claramente necesario un conjunto de reglas de prioridad.

Intuitivamente, es posible desarrollar un conjunto de reglas para la conversión. Puesto que la estructura más conveniente para la mayoría de las operaciones matemáticas y espaciales es la grilla, una regla simple puede ser que todas las variables sean convertidas a grillas antes de que se realice el cálculo. Sin embargo, esto puede conducir a una pérdida innecesaria de información, dado si la variable destino no es una grilla. La Tabla 3.3 muestra una amplia gama de tales combinaciones de operaciones e indica el modelo al cual se deben convertir las variables antes que se realice la operación aritmética. De tal análisis, se puede idear un conjunto de reglas. Las decisiones sobre las cuales se basan muchas de estas reglas dependen del tamaño o del espaciamiento relativo de los elementos espaciales. Este concepto se expresa aquí como la *densidad* que se define como *el número de elementos espaciales por unidad de área*. Lo que sigue es un ejemplo de tal conjunto de reglas, mencionado en orden de prioridad:

1. Si ambas fuentes son espacialmente equivalente, utilice esa estructura.
2. Si cada fuente es un TIN o un modelo del contorno, utilice la estructura del destino.
3. Si todas las variables son grillas anidadas, utilice la grilla más densa.
4. Si el destino es espacialmente idéntico a uno de los orígenes, utilice la estructura del destino.
5. Si uno de los orígenes es un TIN o un contorno, y el otro es una grilla, utilice la grilla.
6. Si todos tienen aproximadamente la misma densidad, utilice el destino.

7. Si solamente uno es un modelo de puntos, utilice los puntos, a menos que los puntos sean muy escasos.
8. Utilice la estructura más densa. Si hay un empate, utilice el destino.

Mientras que hay oportunidad para la experimentación al idear el conjunto de reglas de la prioridad perfecto, está claro que la implementación de cualquier conjunto de reglas se puede hacer sin información del modelador.

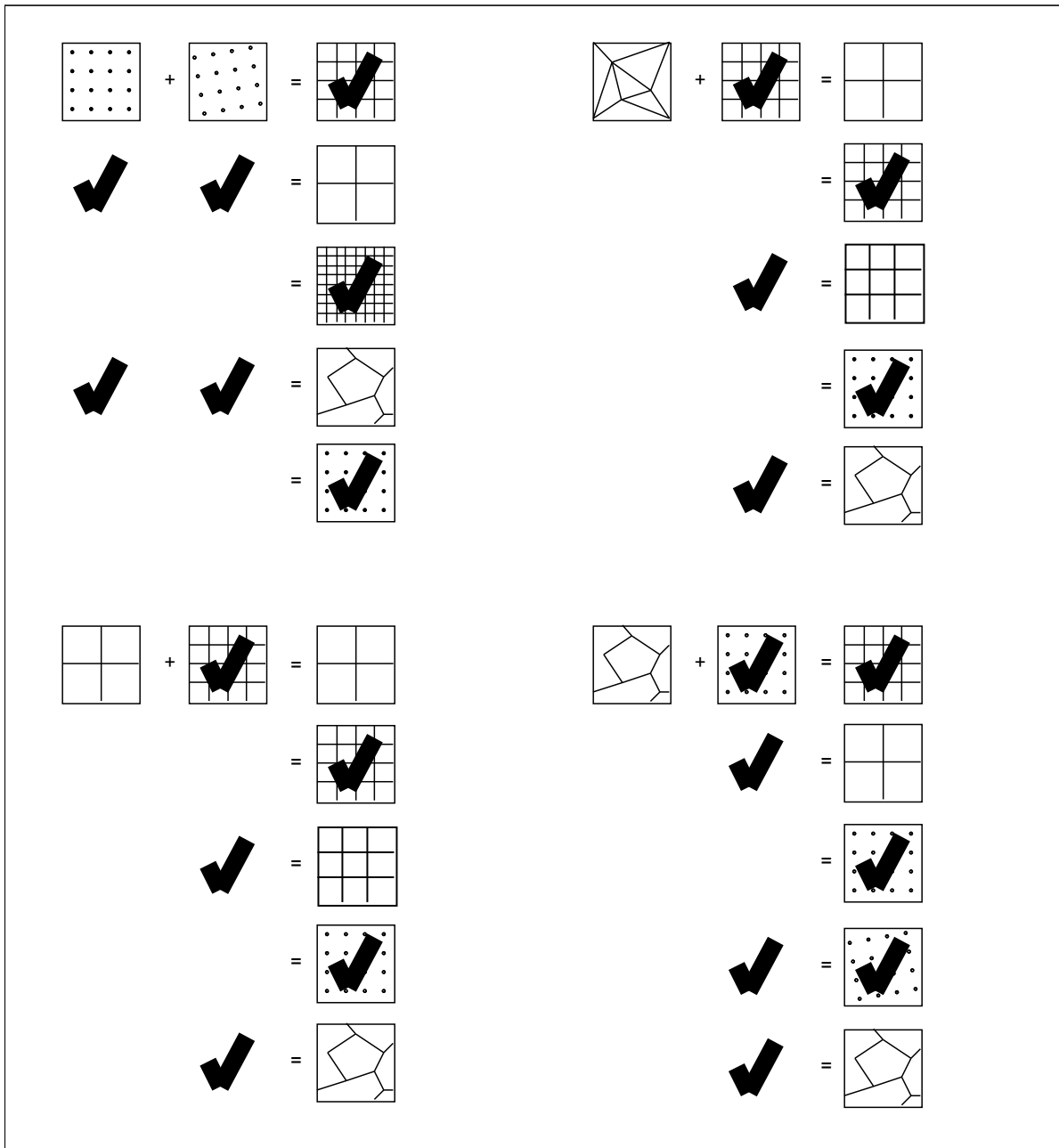
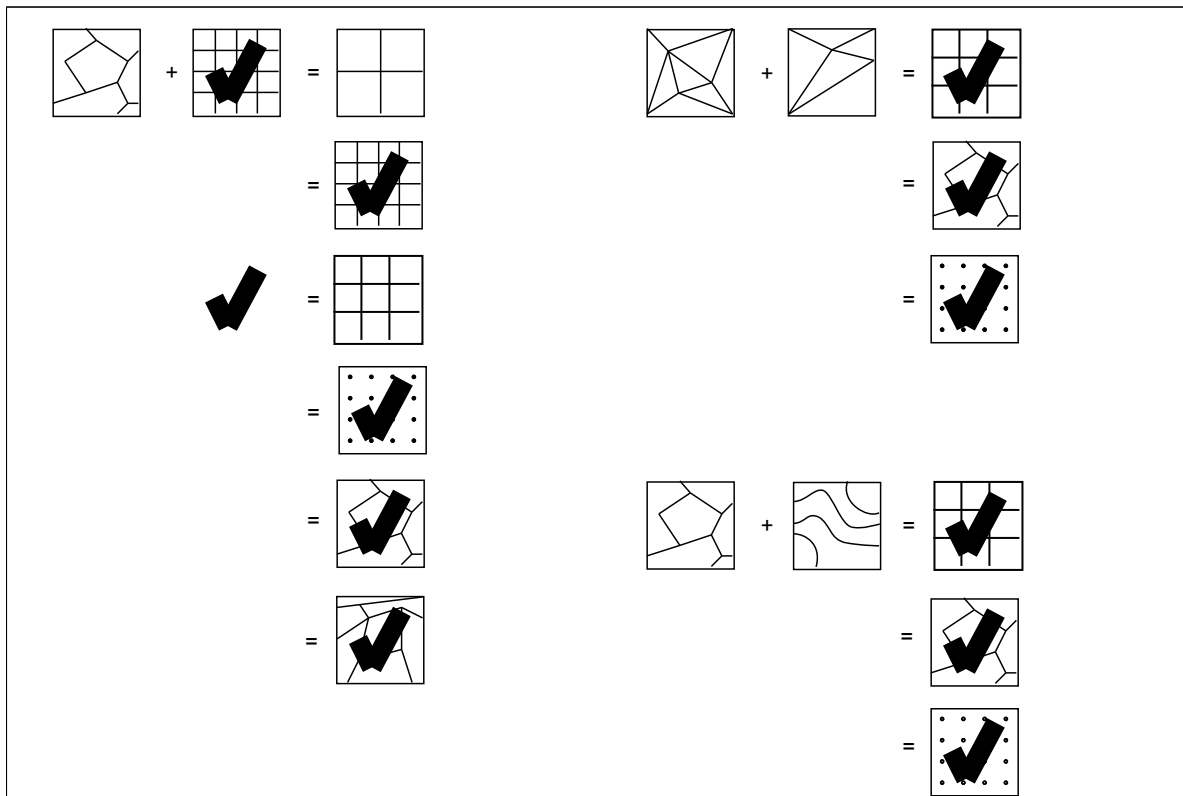


Tabla 3.3: Combinación de modelos de datos espaciales



Este esquema ilustra algunas de las diversas combinaciones de modelos de datos espaciales que se pueden requerir en una operación matemática. Los iconos representan diversos modelos y estructuras de datos espaciales. Los iconos marcados indican los modelos de cómputo preferidos para cada operación. Donde hayan dos marcas para una sola operación, se puede utilizar cualquier modelo.

Tabla 3.3: Combinación de modelos de datos espaciales (Continuación)

3.5.1 Escalas incompatibles

La anterior discusión ha ignorado el tema fundamental de las escalas incompatibles, un asunto multifacético y complejo. Puesto que hay procesos que funcionan a diversas escalas en cualquier sistema natural, la frecuencia de muestreo y la densidad de un conjunto de datos espacial determinará qué procesos son representados y pueden ser modelados. La mezcla de conjuntos de datos basados en frecuencias de muestreo y densidades espaciales mezclados pueden conducir a resultados confusos o engañosos. Este aspecto de escalas incompatibles es un tema que se repite en todas las ciencias que realizan modelos pero, puesto que la naturaleza de esta resolución es dependiente de los dominios y los procesos de modelos específicos, esto está más allá del alcance de este trabajo.

Si se puede asumir que los conjuntos de datos que deben ser utilizados describen juntos el nivel apropiado de variabilidad de los campos que son representados, hay algunos temas genéricos de incompatibilidad de escala que pueden ser tratados. Por ejemplo, suponemos que se está

desarrollando un modelo que utilizará una grilla de diferencia finita de 10' latitud y longitud y una de las variables de entrada necesarias está disponible solamente con un espaciado de grilla de 1° de latitud y longitud. Para propósitos estadísticos, se puede sugerir que es inadecuado aumentar la densidad de la resolución de este conjunto de datos de modo de poder utilizarlo con otro conjunto de datos. Esto es porque el número de grados de libertad, determinados normalmente por el número de elementos de datos, sigue siendo el del conjunto de datos original, aunque el número de elementos aumenta perceptiblemente con el aumento de densidad. Dado que se han creado nuevos puntos o celdas de los datos originales, ellos no representan las observaciones independientes separadas (es decir grados de libertad). Así, los grados de libertad disponibles no son determinados por el número de elementos espaciales de la variable con mayor densidad sino por el número de elementos en la variable original. Esto implica que cada variable que resulta de una operación en la cual una o más de las variables originales han aumentado su densidad debe llevar una indicación de los verdaderos grados de libertad. Mientras que esto no se proporciona para la sintaxis descrita más adelante, es una deseable adición para versiones futuras.

Sin embargo, para las conversiones de modelos de datos espaciales, a menudo es el modelo más denso el que determina cómo deben ser realizadas las operaciones y cómo pueden ser determinados exactamente los resultados. Como lo demuestran las figuras 3.8 y 3.9, al realizar las conversiones del modelo de datos, si se restringen las operaciones los modelos más dispersos solamente esto puede conducir a una pérdida innecesaria de información.

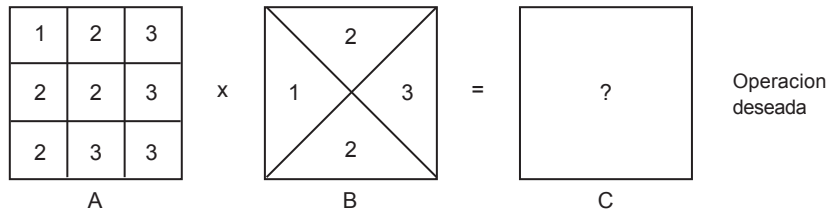


Figura 3.8: Conversión del modelo de datos.

Aquí, la grilla de celdas A es multiplicada por el modelo de polígonos B para producir el polígono (o grilla de celdas) disperso C. Hay varias maneras en que se pueden integrar estos diversos modelos de datos para este cálculo. El caso 1 da el resultado verdadero puesto que los dos modelos de datos origen son superpuestos para producir un modelo intermedio denso que incluye toda la variación registrada. Un promedio pesado por área de estos valores produce el resultado final. Los otros tres casos demuestran cada uno de las alternativas para la selección de un modelo de cómputo desde las estructuras del modelo origen a la del destino. En el caso 2, el modelo B se convierte al modelo de grilla de celdas del modelo A antes de hacer la multiplicación y el promedio. El caso 3 utiliza el modelo B como la estructura de cómputo. En el caso 4, ambos modelos de origen son convertidos, por un promedio pesado por área al modelo del destino. Está claro que usando el modelo más disperso, el destino (caso 4), la estructura de cómputo produce la estimación menos exacta del producto. Así, mientras que la densificación no se debe utilizar para simular un aumento de la información disponible, es apropiada durante las conversiones de modelo.

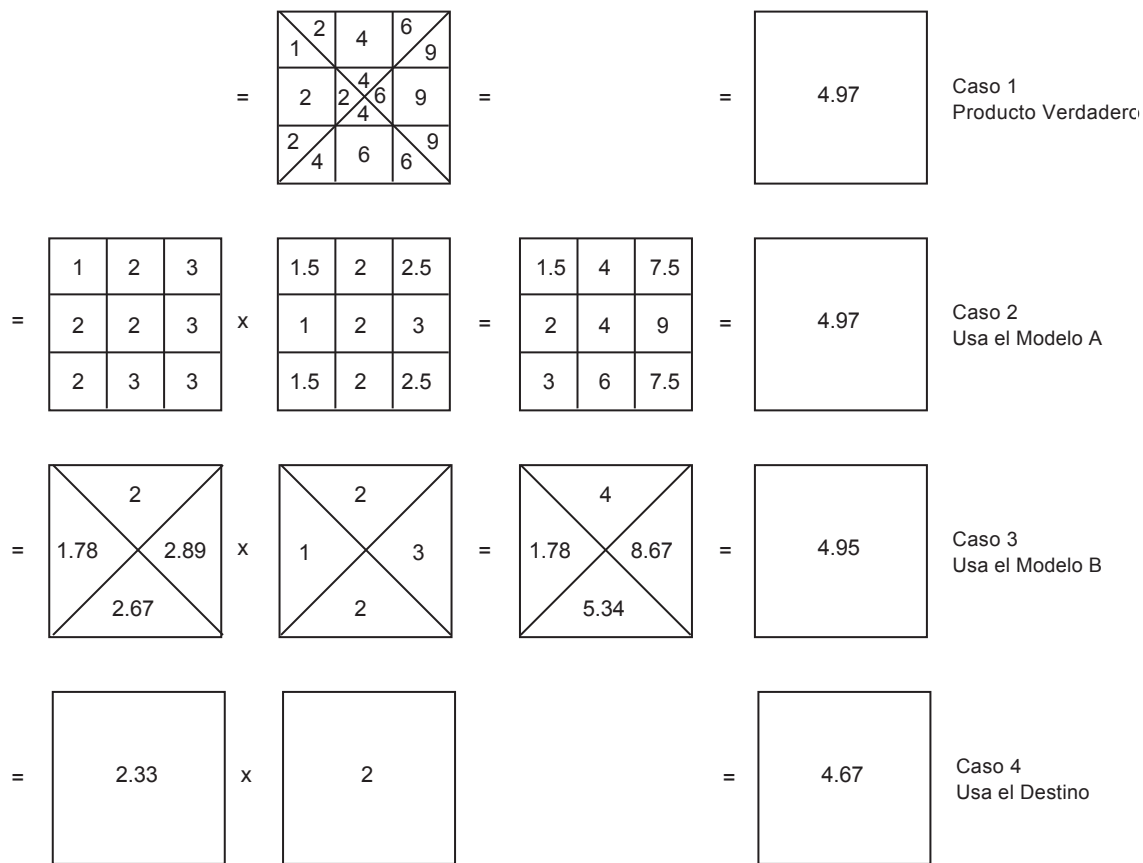


Figura 3.9: Distintas formas de resolver la conversión del modelo de datos.

3.6 Estadísticas descriptivas para campos y variables de campo

Hay varias medidas estadísticas que se utilizan para resumir o para describir conjuntos de observaciones. Estas medidas incluyen la tendencia central que dan una indicación de los valores típicos observados y la dispersión que sugieren el grado de variabilidad en el conjunto [Sil79]. Para conjuntos de observaciones, el valor medio es la suma de todos los valores dividida por número de valores, los valores máximo y mínimo son simplemente el valor más grande y más pequeño del conjunto de datos, y la desviación estándar es la raíz cuadrada de la suma de los cuadrados de la diferencia entre cada valor y el promedio divididos por el número observaciones.

Mientras que estas medidas se definen simplemente para un conjunto de valores discretos, es posible expresar alguna de estas estadísticas en términos de derivadas e integrales de funciones continuas. Por ejemplo, un valor máximo local de una función ocurre en los puntos donde la primer derivada es igual a 0 y la segunda derivada es negativa. El valor medio de una función

$f(x, y)$ de dos dimensiones en los intervalos $x = [a, b]$ e $y = [c, d]$ se define como

$$\bar{f} = \frac{1}{(b-a)} \frac{1}{(d-c)} \int_a^b \int_c^d f(x, y) dx dy$$

Sin embargo, cuando los campos son campos físicos que no se pueden describir por simples funciones, la definición y la determinación de tales estadísticas descriptivas son extremadamente difíciles. Afortunadamente, los conjuntos de datos discretos proporcionados por los modelos de datos espaciales nos proveen de medios de idear estadísticas representativas para los campos. En esta sección repasamos los métodos por los cuales se puede idear la estadística descriptiva simple para los campos representados por los varios modelos de datos espaciales. Se debe recordar que existen muchas medidas estadísticas sofisticadas para analizar datos espaciales [Hai90] y [Ans89], incluyendo medidas de autocorrelación e índices espaciales del modelo espacial. Mientras que éstos pueden ser a menudo mejores descriptores de los conjuntos de datos, la estadística espacial, todavía tiene que incorporarse extensamente en los modelos matemáticos. Por lo tanto, confinamos la siguiente discusión a esas medidas estadísticas que se puedan utilizar en los modelos matemáticos de procesos ambientales. Estas medidas se pueden utilizar solamente con los campos numéricos.

3.6.1 Media

Mientras que calcular la media para un campo continuo que no sea descrito por una función simple con una integral definida es difícil o imposible, es posible calcular la media para los campos representados por variables de campo discretizadas. Dado que los modelos *piecewise* constantes son representativos de la superficie continua y que el valor para cada región es representativo de todos los valores de la región, el valor de la región se puede considerar la media local. La media del conjunto de datos completo en un modelo *piecewise* constante se calcula como la suma de cada valor de la región multiplicado por su área dividida por el área total del área en estudio. Semejantemente, los campos representados por TIN también pueden ser promediados encontrando la media de cada triángulo (el valor en el centro de figura) y calculando el promedio pesado por área sobre el campo.

Si asumimos que la variación entre las líneas de contorno es suave, la media para los campos representados por los modelos de contorno se pueden calcular construyendo polígonos entre las líneas de contorno adyacentes. Los polígonos limitados por los contornos se definen como áreas contiguas limitadas pero no intersectadas por las líneas de contorno y, posiblemente, la frontera del área bajo estudio:

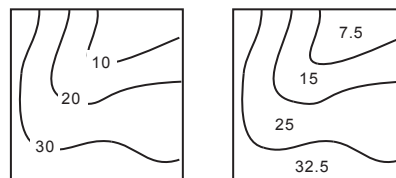


Figura 3.10: Conversión del modelo de contornos a polígonos.

A cada polígono se le asigna un valor basado en los valores de las líneas de contorno que lo limitan. La media de la superficie completa se calcula usando el promedio pesado por área. Dado que las grillas de puntos son conjuntos de puntos muestreados en una grilla regular independiente del fenómeno muestreado, una media aritmética simple del conjunto de datos en la grilla de puntos proporciona una estimación razonable de la verdadera media del campo de la misma manera que la media de la muestra se utiliza para estimar la media de una población. Este acercamiento requiere que se presuma que no hay variación espacial sistemática con una periodicidad igual al espaciado de la grilla. Para los puntos irregulares, la media de los valores del conjunto de puntos también puede proporcionar una estimación de la media de la superficie si los valores de los puntos de la muestra se asumen escogidos al azar. Sin embargo, la autocorrelación espacial puede hacer que este acercamiento sea inválido si los puntos no se distribuyen uniformemente sobre la superficie puesto que los puntos muestreados en ubicaciones cercanas harán más pesado el valor de la media al valor en esa área. Así, si los puntos irregulares están agrupados, es más apropiado construir los polígonos de Thiessen alrededor de cada punto y continuar como si el conjunto de datos fuera un modelo de polígono (es decir utilizar el promedio pesado por área) o triangular la superficie y continuar como con los TINs. La elección de cual proceso utilizar depende, si las puntos se pueden considerar representativos de sus vecinos (en este caso son apropiados los polígonos de Thiessen) o puntos críticos de la superficie (en este caso se preferirá la triangulación). La información que se utiliza para tomar esta decisión se puede encapsular en el conjunto de datos o, en el caso de los problemas debido a la autocorrelación espacial, se puede invocar una prueba para verificar la aleatoriedad espacial.

Desde luego que es muy posible que la media sobre la superficie no fuera estacionaria. En este caso puede ser necesario una aproximación mas sofisticada para calcular la media, incluyendo la tendencia de la superficie y análisis de Fourier.

3.6.2 Suma (Integración)

Al trabajar con conjuntos de datos discretos que cuentan cosas tales como casos de cáncer por provincias o votos por distrito electoral, si deseamos saber “¿cuánto?” es representado por el conjunto de datos en su totalidad, simplemente sumamos todos los elementos del conjunto. Al trabajar con campos tales como cantidad de precipitación o contenido de nitrógeno, saber “¿cuánto?” requiere la integración de la función que representa la superficie incluida bajo el área de estudio. En dos dimensiones, la suma de una función definida $f(x, y)$ incluida en los intervalos $x = [a, b]$ e $y = [c, d]$ es

$$\text{suma}(f) = \int_a^b \int_c^d f(x, y) dx dy$$

El resultado de la operación de integración es el área bajo la curva. En dos dimensiones el resultado es el volumen.

Cuando se trabaja con campos que no son descriptos por funciones, se pueden encontrar las integrales dividiendo el área del estudio en regiones para las cuales se puede calcular un valor medio. La media de cada región es multiplicada por el área de la región para dar el volumen de la región y la integral sobre el área de estudio es el total de todos los volúmenes de la región. Por lo tanto, para calcular el integral de cualquier modelo de datos espacial simplemente necesitamos

encontrar el conjunto de las regiones para las cuales podemos determinar valores medios. Para las grillas de celdas y los polígonos, estas regiones son definidas por el modelo. La media de una región definida por un triángulo en un TIN es el valor en el centro del triángulo. Se pueden derivar polígonos valuados por superficie a partir de modelos de contorno según lo descrito arriba. Los modelos de puntos requieren inicialmente la construcción de regiones. Esto se logra a través de la construcción de los polígonos de Thiessen o con una triangulación basando esta opción de acuerdo a si los puntos son representativos o críticos según lo descrito arriba. Una vez que se hayan construido las regiones a partir de los modelos de puntos, se procede con la integración según lo especificado para los modelos de polígono o TIN.

La salida final de muchos modelos matemáticos es un solo valor escalar final que se utiliza para representar la integración de contribuciones individuales a través de toda el área de estudio. Puesto que la mayoría de los lenguajes de programación incluyen la operación explícita claramente definida de `suma` que realiza suma de todos los valores en un conjunto especificado, proponemos una nueva operación, `integración` que substituye a la `suma` al trabajar con variables de campo. Aquí se abrevia a la forma

$$a = \text{integ}(B)$$

donde `a` es un escalar y un `B` es un campo numérico. También, a menudo deseamos saber las integrales individuales dentro de regiones específicas tales como provincias o tipos de suelo. Por lo tanto, es útil agregar un segundo campo categórico opcional a la función `integ` que define las regiones dentro las cuales debe ser calculada la integración.

$$a = \text{integ}(B, C)$$

donde `C` es un campo categórico.

3.6.3 Máximo y Mínimo

El máximo y el mínimo de un campo descrito por una variable espacial se pueden considerar generalmente que son las observaciones máximas y mínimas en el conjunto de datos. Dado que la superficie se describe totalmente en modelos *piecewise* constantes, estas medidas descriptivas son simplemente los valores extremos en el conjunto de valores de los elemento espaciales. Para los TINs, los extremos se extraen del conjunto de valores de los nodos. Para los modelos de puntos, se podrán utilizar procedimientos sofisticados de interpolación tales como *splines* o los ajustes polinómicos ara definir la superficie de la cual serán extraídos los extremos. Sin embargo, dado que muchas de estas técnicas utilizan los puntos de la muestra como extremos, y puede ser difícil determinar los valores extremos de algunas superficies, es también razonable extraer los valores máximo y mínimo directamente del conjunto de valores de los puntos.

Los modelos de contorno no dan valores extremos exactos puesto que el valor del campo varía suavemente a través de líneas de contorno. Los valores máximos y mínimos se deben extrapolar del valor de las líneas más altas y más bajas de contorno contenidas dentro del área de estudio. Mientras que es cierto que estos valores extremos caen dentro de un intervalo de contorno de las

líneas de contorno con valores extremos, no hay información adicional disponible dentro del modelo de datos espacial para producir estimaciones exactas.

3.6.4 Desviación estándar

La desviación estándar de una función, $f(x, y)$ se puede definir como:

$$\sigma = \sqrt{\frac{\int_a^b \int_c^d (f(x, y) - \bar{f})^2 dx dy}{\int_a^b \int_c^d dx dy}}$$

donde \bar{f} es la media de la función f . Puesto que esto es difícil determinar para un campo no definido por una función simple, es necesario recurrir otra vez al uso de las representaciones discretas. Para calcular la desviación de estándar para un modelo *piecewise* constante, cada cuadrado de la diferencia se debe pesar por el área de cada región:

$$\sigma = \sqrt{\sum_i \left((x_i - \bar{x})^2 \times \left(\frac{area_i}{area_{total}} \right) \right)}$$

donde i el índice de cada región. Las desviaciones estándar de los modelos TIN y de contorno se pueden encontrar de una manera similar una vez que el valor medio para cada región se haya determinado según lo descrito arriba. Debe ser observado que las desviaciones estándar encontradas de esta manera serán menores que el valor verdadero puesto que los medios locales de cada región han quitado ya algo de la verdadera variación.

Las grillas de puntos y los puntos irregulares son los únicos modelos que pueden producir un valor estadístico correcto de la desviación estándar. En estos modelos, los valores representan el valor verdadero en un punto y el conjunto en sí mismo se puede tratar simplemente como un conjunto de observaciones. Según lo observado arriba, pueden presentarse problemas con la autocorrelación espacial para los modelos de puntos irregulares así que puede ser preferible construir polígonos de Thiessen o triangular y proceder de acuerdo a polígonos o a TINs.

3.7 Estadísticas Inferenciales

Finalmente consideramos abreviadamente el área de la estadística deductiva como operación en modelos ambientales. En sentido estricto, la estadística deductiva no forma parte de los modelos por sí mismos, sino que son importantes en el resto de los aspectos de los modelos matemático incluyendo el ajuste del modelo, la prueba y evaluación, construcción de intermediarios, visualización de resultados e importantemente en la interpolación y el muestreo.

Debido a la inherente característica de la autocorrelación espacial y de la carencia de independencia que resulta en los conjuntos de datos espaciales, los investigadores que se ocupan de datos espaciales a menudo encuentran necesario evitar acercamientos estrictamente paramétricos a la estadística deductiva en favor de procedimientos no paramétricos, cualitativos y robustos. Los acercamientos paramétricos requieren suposiciones de la independencia y de la normalidad, que es poco realista para muchos conjuntos de datos espaciales. Sin embargo, los resultados de técnicas robustas y no paramétricas son limitadas por su generalidad (es decir los resultados no se pueden mantener para la población de donde fue adquirida la muestra) y tienden a ser muy conservadoras [Ans89].

También, muchos estadísticos discuten que un conjunto de datos que constituye una variable de campo no sea el conjunto de muestras escogidas al azar requeridas por la estadística deductiva tradicional sino que sea un solo caso de una variable estocástica. De hecho, se ha sugerido que un conjunto de datos espacial forma la población completa y consecuentemente, no puede ser utilizada ninguna estadística deductiva [Sum83], [Ans89], aunque Anselin ha discutido que esta condición a menudo no se considera debido “a la naturaleza imperfecta de la medición, y al error inherente (o ruido)” ([Ans89], p. 8). Si éste es el caso, el conjunto de datos espacial se puede ver como el resultado de un conjunto de procesos estocásticos que han medido la señal subyacente junto con el error estocástico de la medición [Hai90].

3.7.1 Geoestadística y la teoría de variables regionalizadas

Afortunadamente, recientemente ha ganado renombre un nuevo acercamiento a la estadística deductiva que proporciona a algunas herramientas nuevas para ocuparse de los campos. Se han desarrollado la geoestadística y la teoría de variables regionalizadas en un número de diversas disciplinas, originalmente en topología y climatología, pero la fuerza impulsora común detrás de este desarrollo ha sido la necesidad de desarrollar técnicas estadísticas que trabajen con fenómenos espacialmente continuos correlacionados. Los campos se describen tradicionalmente como funciones de la ubicación, [OG89]:

$$z(x, y) = f(x, y) + e$$

donde $z(x, y)$ es el valor de una variable z en la ubicación (x, y) y f denota una función determinística y e es un término de error al azar. Tomando este concepto, el uso de ajuste polinómico para interpolar superficies en un número limitado de sitios de muestra implica que la mayoría de la variación observada es determinística. Una manera alternativa de ver la variación observada es considerarla como esencialmente azarosa pero espacialmente dependiente.

La geoestadística es la aplicación de la teoría de variables regionalizadas a los datos espacialmente distribuidos [Jou86]. Aunque Journel sugiere

Desde un punto de vista teórico, poco es nuevo en esta teoría, la cual toma prestado la mayoría de sus modelos y herramientas del concepto de funciones al azar estacionarias y de técnicas de análisis de la variación y de la predicción generalizada de mínimos cuadrados. ([Jou86], p. 120)

Esta teoría proporciona una “teoría concisa, coherente y útil” ([OG89], p. 268) brindando un nuevo conjunto de herramientas probabilísticas que se pueden utilizar para estimar y para describir la variación espacial de fenómenos en la superficie de la tierra.

Todo la geoestadística depende del desarrollo de un modelo de la dependencia espacial observada en la variable regionalizada. Este modelo es el *variograma* que relaciona la variación del valor de la variable entre dos sitios cualquiera con la distancia que lo separa. Así, la etapa inicial en cualquier análisis geoestadístico es el desarrollo del variograma.

3.7.2 Construcción de variogramas

El variograma es construido a partir de datos de la muestra calculando la variación entre los valores de la muestra para un rango de diversas distancias de separación. Si la variación espacial es isotrópica entonces la estimación de la semivarianza, Υ^* , se calcula como [OW91]:

$$\Upsilon^*(h) = \frac{1}{2m(h)} \sum_{i=1}^{m(h)} \{z(x_i) - z(x_i + h)\}^2$$

donde $z(x_i)$ y $z(x_i + h)$ son valores observados en x_i y $x_i + h$, h es la distancia, $\Upsilon^*(h)$ es la estimación de $\Upsilon(h)$ y $m(h)$ es el número de comparaciones emparejadas en esa distancia. Una vez que se haya calculado la variación para diversos h , se construye una *muestra* o un variograma *experimental* trazando $\Upsilon^*(h)$ contra h . Si el campo es anisotrópico, este cálculo se hace individualmente para diversas direcciones con cada conjunto de puntos tratados por separado en manipulaciones posteriores.

El variograma de la muestra resultante consiste en un conjunto de puntos discretos (o conjuntos de puntos si se considera anisotrópico). Para poder utilizar este gráfico para estimar la semivarianza en cualquier distancia dada, será necesario ajustar un modelo suave a esos puntos. Usando la aproximación de los mínimos cuadrados, los puntos del variograma de la muestra serán ajustados a uno de diversos modelos, incluyendo el esférico, linear, exponencial, gaussiano, De Wijsian y las funciones de Bessel [Bur91]. Davis observa que el proceso de ajustar modelos es “hasta cierto punto un arte, requiriendo experiencia, paciencia y a veces suerte. El proceso no es satisfactorio en su conjunto, porque las conclusiones no son únicas” ([J.C86], p. 245).

El variograma resultante tiene varias características interesantes que se pueden interpretar con respecto al fenómeno en estudio (ver Figura 3.11). El *rango* es la distancia sobre la cual los puntos ejercen influencia sobre sus vecinos; más allá del rango, marcado por el comienzo del *umbral*, no se observa ninguna influencia. El *nugget* representa la variación, no espacial, completamente al azar [OW91]. Sin embargo, muchos ejemplos de variogramas y sus modelos de ajuste no se corresponderán con esta forma clásica.

3.7.3 Kriging

Kriging es “un nombre genérico para una variedad de algoritmos generalizados de valoración de mínimos cuadrados” [Jou86] basados en el variograma y sus derivados. En su caso más simple,

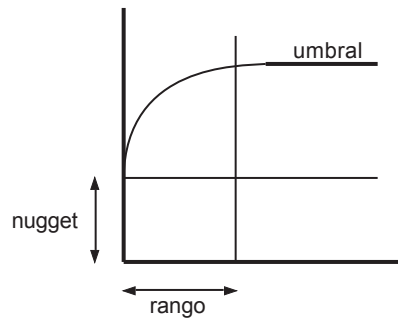


Figura 3.11: Distintos tipos de curva.

una estimación de puntos hecha usando Kriging es una suma pesada linealmente de los datos de la muestra [OG89]:

$$z^*(x_0) = \sum_{i=1}^n \lambda_i z(x_i) \text{ con } \sum_{i=1}^n \lambda_i = 1$$

donde x_i son los n puntos usados en la estimación, $i=1$ a n , y $z^*(x_0)$ es la estimación del valor en el punto x_0 . Los pesos son elegidos solucionando un conjunto de ecuaciones simultáneas:

$$\sum_{i=1}^n \lambda_i \cdot \Upsilon(x_i, x_j) + \Psi = \Upsilon^*(x_i, x_0) \text{ para todo } j$$

donde $\Upsilon(x_i, x_j)$ es la semivarianza entre x_i y x_j , $\Upsilon(x_i, x_0)$ es la semivarianza entre x_i y el punto que se estima (x_0) e Ψ es el multiplicador de Lagrange agregado para alcanzar la minimización. Las semivarianzas se determinan a partir del variograma. La variación de la estimación es

$$\sigma^2(x_0) = \sum_{i=1}^n \Upsilon^*(x_i, x_0) + \Psi$$

Las estimaciones que usan Kriging son imparciales y óptimas [J.C86].

3.7.4 Usando Geoestadística

La geoestadística está saliendo rápidamente del dominio de los investigadores estadísticos y de las manos de médicos hacia diversos campos [OW91]. Su aplicación más importante está en la interpolación puesto que se ha encontrado que las estimaciones usando kriging son tan buenas o mejores que las realizadas con otras técnicas [Bur86], [Jou86] aunque este conflicto ha sido discutido por otros [PW86]. La existencia de una medida del error en la variación de la estimación es inusual entre los procedimientos de interpolación. Las técnicas para idear estimaciones de “bloque” de muestras de puntos eran uno de los primeros progresos en el conjunto de herramientas disponibles. Las estimaciones de bloque son particularmente importantes en la geología de minas donde se deben utilizar de base muestras dispersas para determinar el grado del mineral incluido en un gran

bloque de explotación minera [J.C86], aunque la aplicación de esta técnica para la derivación de estimaciones para el valor de los fenómenos modelados usando modelos *piecewise* continuos son obvias. La geoestadística es también útil para planear diseños óptimos y de hojas de operación de muestreo eficientes para los fenómenos continuos y para realizar la clasificación espacial [OW91]. Finalmente, el análisis del variograma en sí mismo proporciona a una cierta penetración en la forma de la variación espacial del fenómeno bajo estudio.

Sin embargo, hay muchos problemas con la geoestadística y el Kriging. Algunos estadísticos cuestionan las bases estadísticas fundamentales de la teoría de variables regionalizadas y de sus derivados en geoestadísticas (Philip y Watson, presentan una dura crítica). En un nivel menos fundamental, los problemas importantes de la geoestadística se presentan debido a las demandas de cómputo extremadamente altas del método. Cada estimación de punto requiere la solución de un conjunto de ecuaciones simultáneas - 17 o más se utilizan en algunos paquetes de contornos [J.C86]. También, Lam ha observado que “el método no es fiable a menos que estén disponibles una gran cantidad de valores de muestra” ([Lam83], p. 134); [WO92] han mostrado que deben ser utilizadas por lo menos 100 y preferiblemente 200 puntos.

Finalmente, como Oliver, et al, han observado, “el método es de ninguna manera universalmente aplicable y los investigadores necesitan estar absolutamente seguros que es apropiado a sus circunstancias cuando lo usan” ([OG89], p. 262-3). De hecho, mientras que “el proceso kriging es una solución elegante a un problema difícil, requiere una comprensión clara de su acercamiento y datos algo más elegantes que los a menudo disponibles para alcanzar un resultado confiable” ([McC88], p. 753).

3.7.5 Geoestadística para modelos de datos espaciales

¿Cómo se puede utilizar la geoestadística para modelos ambientales con variables de campo? Claramente kriging proporciona una excelente alternativa de interpolación para muchas de las conversiones. Sin embargo, si va a ser utilizado bajo demanda dentro de modelos matemáticos, sería absolutamente esencial que el variograma esté encapsulado en el conjunto de datos pues la derivación de éste es una tarea que desperdicia tiempo y necesita un conocimiento experto. Tal encapsulamiento parece apropiado mientras que se espere que el proveedor de los datos esté lo suficientemente familiarizado con el fenómeno y los datos que lo representan para realizar esta tarea de modelo geoestadístico.

Puesto que la geoestadística se está promoviendo más extensamente (observe su inclusión en textos estadísticos estándares como *Análisis estadístico y de datos en geología* de Davis), ahora está apareciendo software para asistir al desarrollo de variogramas y estimaciones kriged. Kriging se incluye ya en muchos GISs comerciales, incluyendo Arc/Info (de ESRI, Inc.), GeoEAS (del EPA, Las Vegas) y Surfer (de Golden Software).

Capítulo 4

Investigaciones Relacionadas

Este capítulo presenta algunos trabajos de investigación relacionados, llevados a cabo por investigadores que trabajan en la búsqueda de soluciones para lograr la integración de los GIS y los modelos ambientales.

4.0.6 Map Algebra

Map Algebra [Tom90] es quizás el primer intento significativo para lograr una separación entre las operaciones de la forma de discretización espacial. Esto se logra permitiendo solo una discretización simple. Empezando con el requerimiento de que todos los datos espaciales que serán manipulados son almacenados en grillas, map algebra provee un *framework* organizativo y un vocabulario para expresar muchas operaciones GIS estándar.

Fundado en el tradicional *overlay* de mapas, map algebra provee un lenguaje que automatiza muchos de los conceptos de los modelos cartográficos. Tomlin sostiene que: “el modelo cartográfico es una metodología general para el análisis y la síntesis de datos geográficos” [Tom90]. En el modelo cartográfico cada tema en una región es expresado como un mapa o *layer* individual. En un primer momento, el modelamiento cartográfico se realizaba manualmente, superponiendo mapas transparentes (referentes a distintos temas) y observando la combinación de características presente en la región de estudio. El análisis de mapas provee una manera de implementar y extender las técnicas de superposición manual a través de una computadora. Chan y Tomlin dicen que: “en el corazón de los modelos cartográficos están las ideas de descomponer los *datasets* y las especificaciones que controlan el procesamiento de datos en componentes elementales que pueden ser recombinados con facilidad y flexibilidad” t-giscm-90 (p 351). Tomlin reconoce que la aproximación a los modelos cartográficos está “orientada a la función”, ya que el lenguaje hace más hincapié en las funciones que se aplican a los datos que en la representación de los mismos. La forma en que se almacenan los datos y las funciones especificadas no intentan representar la realidad, sino facilitar la manipulación de los datos digitales.

Esta reducción a componentes elementales se logra requiriendo que cada elemento a ser examinado con Map Algebra debe estar discretizado en forma de un *layer* de celdas rectangulares. Esto facilita la implementación de las funciones de análisis ya que solo existe una posible discretización del espacio. Sin embargo, como resultado, todas las operaciones son conceptualizadas en una grilla

en vez de en los campos en sí. Por ejemplo, la distancia desde un punto debe pensarse en términos de una grilla de $N \times N$, en lugar de utilizar la distancia lineal. Por lo tanto el modelador debe trabajar en un universo de grillas en vez de en un universo continuo descrito por la leyes físicas.

¿Todo lo que necesita un modelador de fenómenos ambientales es un universo de grillas? ¿Es posible convertir cualquier conjunto de datos en una grilla? ¿Es ésta la mejor solución?, ciertamente no, si consideramos la realidad que esta siendo representada. Ciertos fenómenos pueden ser representados de otras maneras. Si la matemática no requiere una estructura de grilla, la conversión innecesaria de una representación a otra lleva a una inaceptable pérdida de información. Todo movimiento debe ser discretizado a 4 u 8 direcciones, por lo tanto, los caminos entre dos puntos se distorsionan innecesariamente. Si se reducen todos los campos a un simple modelo datos de grilla, en el cual los valores son constantes dentro de las celdas y cambian abruptamente en los bordes de las celdas, es difícil conceptualizar y trabajar con funciones continuas tales como diferenciación e integración.

Map Algebra requiere que forcemos la realidad para que “encaje” en una estructura, en vez de permitir elegir una estructura más apropiada para la realidad que se quiere analizar. El modelador es forzado a trabajar en una discretización simple y estar constantemente consciente las limitaciones que eso acarrea. A menos que el tamaño de la grilla pueda reducirse tendiendo a cero, es imposible pensar en los procesos naturales continuos que se desarrollan en espacios también continuos, en un mundo de grillas. Más que una verdadera álgebra para modelar fenómenos, Map Algebra es una nomenclatura unificada para definir formalmente operaciones GIS en datos modelados como grillas. Esta debilidad en el nexo entre la realidad y el modelo es crítica.

A pesar de sus limitaciones, map algebra ha conseguido una notable popularidad. Ha sido elegido como el framework para el lenguaje de modelado incluido en un producto *raster* de ESRI: GRID [ESR91] y es la base para el software GIS MapBox (Decision Images, sin fecha). Una vez que los datos son ingresados en modelo (adaptados a éste), es posible expresar muchas operaciones matemáticas entre distintos conjuntos de datos representando variables ambientales, en una sintaxis estructurada. Por ejemplo, si queremos sumar dos temas (campos), llamémoslos FIRST y SECOND, la sintaxis según Tomlim sería:

```
RESULT = LocalSUM of FIRST and SECOND
```

la implementación de Map Algebra en el software GRID permite una sintaxis más directa:

```
RESULT = FIRST + SECOND
```

4.0.7 Interfaces modelo-datos

Otra solución completamente diferente al problema de enlazar datos digitales acerca de fenómenos espacialmente distribuidos con los modelos matemáticos de los procesos ambientales, consiste en desarrollar interfaces genéricas entre los sistemas de información espacial y los programas de aplicación. En general, estas interfaces permiten que los datos sean almacenados en diferentes archivos y formatos en sus sistemas de origen. Cuando se va a utilizar el modelo matemático, los datos necesarios son requeridos a la interface. La interface accede a las BDs y extrae los conjuntos de

datos pedidos, y los convierte automáticamente a un formato apropiado para el modelo. Luego la interface envía los datos al modelo, el cual calcula el resultado. La salida del modelo se maneja inversamente.

Breuning y Perkhoff describen el desarrollo de un sistema de interfaces diseñado para mediar entre varias bases de datos heterogéneas y programas de aplicación “orientados a la tarea”. En este caso, “el objetivo de la integración de datos es el uso transparente en diferentes sistemas de BD autónomos” [BA92]. Usando una estructura de cliente-servidor, el DIS (Data Integration System) se sitúa entre los programas de aplicación y las BDs. Los usuarios definen y usan vistas de las BDs dependientes de la aplicación de tal manera que “la estructura y la localización de los datos, y los diferentes lenguajes para su manipulación son irrelevantes para el usuario”.

En su implementación actual DIS provee un modelo FGD (Functional Global Data) que permite el esquema de integración entre INGRESS y DASDBS Geokernal, y provee un alto nivel de abstracción de datos. El FGD es un lenguaje funcional fuertemente tipado, que soporta variables de tipo y constructores para definir nuevos tipos, también provee un conjunto de funciones primitivas. Una ventaja particular de este sistema es que los algoritmos que son específicos para ciertas representaciones de la realidad, tales como distancia u orientación, pueden ser mantenidos en la BD, en vez de escribirlos para cada aplicación específica.

Estos sistemas de integración de datos son diseñados para satisfacer las necesidades de intercambio de datos de aplicaciones específicas. En vez de permitir manipulación de datos espaciales independiente del sistema, proveen herramientas para la conversión de datos de un formato a otro. No proveen una manera en la que los modeladores puedan manipular las representaciones de fenómenos espacialmente distribuidos sin tener en cuenta las restricciones de un software específico.

4.0.8 Soluciones orientadas a las BD

Otras soluciones tratan con los problemas que se desprenden de representar la complejidad espacial de la realidad, desde una perspectiva de los datos, dejando de lado la formalización de operaciones (como si lo hace Map Algebra) y el desarrollo de interfaces entre la BD y los modelos. Estas soluciones tratan de encontrar la manera de organizar la diferentes conceptualizaciones y representaciones que pueden tomar los datos espaciales. Entonces, los datos se convierten en el foco del trabajo, en lugar de las operaciones que se realizarán con ellos.

En los 70s IBM desarrolló un prototipo de un sistema de BD-Aplicación, que fue diseñado para manipular datos standard y geográficos, dentro del mismo framework.

“El soporte para datos geográficos permite al sistema manejar información extraída de mapas, de la misma manera que maneja datos más convencionales, como strings y datos numéricos. Esto se logra porque el sistema provee un tipo de dato geográfico, y un conjunto apropiado de operadores y funciones geográficos”

Usando un diseño de BD relacional y representado los datos geográficos como puntos, líneas y áreas, este prototipo fue pensado para proveer un poderoso sistema de planeamiento urbano. El prototipo fue descrito en varios documentos internos de IBM, pero nunca fue completamente implementado. Aunque quedó incompleto, este proyecto intentó atacar el problema de si los datos geográficos, como tipo fundamental de datos, podían ser manejados concurrentemente con tipos

de datos más tradicionales.

Más recientemente, los conceptos de orientación a objetos sirvieron para definir clases que representan objetos espaciales complejos, con operaciones especializadas heredadas y encapsuladas, que pueden incorporadas en modelos computacionales. Smith presentó los elementos de un modelo basado en lógica y un lenguaje de BD, que tiene por objetivo a largo plazo la unificación de los modelos computacionales y los sistemas de BD para muchas áreas de la investigación científica [Smi92]. Desarrollando un lenguaje formal para definir y manipular entidades espacio-temporales complejas en todos los niveles de abstracción, “desde simples juegos de datos a grandes modelos matemáticos”, Smith espera poder derivar herramientas que permitan a los investigadores “enfocarse en los aspectos científicos y no en los computacionales” [Smi92]. En el centro de este trabajo de investigación esta el “Term Definition Language”, un metalenguaje que permitirá definir y manipular entidades espacio-temporales complejas, sus propiedades y relaciones, independientemente de la implementación específica de la BD. Los datos pueden ser pensados y estructurados de acuerdo a las necesidades del científico o del modelo.

Cuando el trabajo de Smith esté terminado y los científicos tengan acceso a una multitud de grandes bases de datos espaciales, vía esta base de datos basada en la lógica y el lenguaje de modelado, se podrán obtener los beneficios de la orientación a objetos dentro del modelo matemático.

4.0.9 Lenguajes formales y álgebras

Los lenguajes formales y álgebras son herramientas importantes para desarrollar y expresar conceptos abstractos tales como los campos continuos. Los lenguajes formales consisten de un conjunto de símbolos, operaciones y reglas acerca de cómo combinar las operaciones y los símbolos. Las álgebras son un subconjunto de los lenguajes formales en los cuales son importantes las propiedades de las operaciones, tales como simetría, reflexibilidad y transitividad. Tales propiedades permiten la identificación de conjuntos mínimos de operaciones y permiten la simplificación de tareas complejas.

Han existido esfuerzos para desarrollar lenguajes formales para trabajar con datos geográficos. A fines de los 60s y principios de los 70s Dacey bosquejó algunos conceptos para el desarrollo de un lenguaje formal ([Dac67], [Dac70] y [Dac71]). Aunque solo una pequeña parte de ese trabajo fundamental es accesible, la mayoría sólo publicada en reportes técnicos, muchos de sus contemporáneos reconocieron la importancia de su trabajo. Más recientemente, Molenaar ha desarrollado un trabajo acerca de un lenguaje formal. Él describió su visión de una Teoría de la Geoinformación cuyo propósito es “estructurar todo el campo de los sistemas de geoinformación de manera que aspectos comunes y diferencias de los sistemas GIS existentes puedan clarificarse.... y hacerse más transparentes” [Mol91]. Él notó que la definición de entidades dentro de una BD geográfica es muy dependiente de la razón por la cuál fueron recolectados los datos. Por ejemplo, la definición de una unidad de suelo depende de si el dato será usado para planeamiento del uso del suelo, estudios de erosión o un modelo hidrológico. En este trabajo se propone que la teoría debería proveer una gramática para especificar el contexto de la información manejada y métodos para transformar datos desde un contexto a otro.

Trabajando por más de dos décadas, Serra y otros investigadores en el Centre de Morphologie Mathématique en Francia, ha desarrollado un álgebra muy sofisticada para el análisis de

imágenes digitales, a través de la morfología matemática. ([Ser82], [Ser88], [SB88]). La morfología matemática usa un álgebra de transformación para convertir conjuntos acotados de puntos en números significativos. La técnica usa “elementos estructurantes” que “interactúan con el objeto de estudio (la imagen), modificando su forma y reduciéndolo a una especie de caricatura que es más expresiva” que la imagen original [Ser82]. Esta solución es particularmente útil para las imágenes médicas, donde los objetos de estudio (glóbulos en la sangre, tejido conectivo) se muestran de una manera no orientada en una dirección útil que permita entender las características del objeto. Las aplicaciones de la morfología matemática van desde las imágenes médicas a las satelitales.

Mientras que sus modelos de datos se limitaron a grillas hexagonales regulares, y su aplicación a extraer medidas de formas, este esfuerzo demostró como se puede desarrollar formalmente un álgebra transformacional para datos distribuidos espacialmente.

Empezando con un número reducido que limita los tipos de fenómenos que pueden ser considerados en su álgebra, gradualmente se construyen un conjunto de “herramientas” y principios que pueden ser usados para extraer la estructura espacial a partir de los datos basados en puntos. Sus más recientes esfuerzos proveen la base algorítmica para el análisis automático de imágenes médicas. Es importante entender que la investigación de Serra y su grupo está orientada a descubrir las relaciones entre los patrones en los datos y las características morfológicas de los fenómenos reales, en lugar de considerar cómo capturar características relevantes de la realidad en un dataset.

Harvey muestra una extensa revisión del rol de los lenguajes artificiales (por ejemplo, la matemática) en el desarrollo de teorías. Los lenguajes artificiales proveen “un sistema abstracto de signos y relaciones que no tienen un contenido empírico o un significado substancial” [Har69]. El argumenta que ya que los símbolos abstractos y las relaciones pueden ser definidos precisamente dentro del lenguaje, se pueden expresar la claridad y la falta de ambigüedad que son imposibles en el lenguaje natural. Una vez que los conceptos son definidos en lenguajes abstractos y teorías, pueden ser conectados con el mundo real a través de definiciones empíricas de los conceptos abstractos. Por ejemplo, un modelo de gravedad de la interacción entre ciudades se conecta con la realidad diciendo que las ciudades son “cuerpos” en la fórmula y sus respectivas poblaciones representan la masa de dichos cuerpos.

Los fundamentos básicos de las álgebras están bien desarrollados en cualquier libro de álgebra elemental. Se proveen definiciones simples de conjuntos, mapeos, relaciones y funciones que ofrecen una estructura estable para el desarrollo de un álgebra. Selby y Sweet sostienen que es posible “construir muchos sistemas matemáticos distintos, dependiendo en la elección de diferentes conjuntos de elementos, relaciones, operaciones y predicados” [SL69]. Mientras que se desarrollan muchas álgebras para propósitos específicos, cada una tiene sus propios méritos y rango de interpretaciones. Ellos agregan que: “ los sistemas matemáticos pueden ser vistos como juegos que involucran ciertos objetos (elementos) y que son jugados de acuerdo a ciertas reglas (predicados o postulados).

4.0.10 Virtual Data Sets

Andrej Vckovski en su tesis *Interoperable and Distributed Processing in GIS* [Vck] ataca el tema de integración en geoprocusamiento.

Se define la interoperabilidad como la “habilidad de intercambiar todo tipo de información espacial referida a la Tierra y los objetos y fenómenos sobre ella. Además debe ser posible correr software de red capaz de manipular esa información.”

Los impedimentos para la interoperabilidad tienen que ver con:

Diversidad Sintáctica: Las diferencias en la codificación son todavía un área conflictiva importante en el intercambio de datos. La resolución de estos problemas normalmente desemboca en tediosas conversiones manuales o automáticas, que incluye muchos pasos y herramientas software. La utilización de standards es de gran ayuda para simplificar el proceso de conversión.

Diversidad Semántica de la información: Los conflictos semánticos a menudo no pueden ser detectados debido a la ausencia de especificaciones suficientes. Cada concepto puede representar una idea diferente en el marco de diferentes modelos. Hacer un mapeo entre los conceptos de dos modelos diferentes a menudo no será suficiente, ya que no se puede prever como serán usados éstos.

Para solucionar estos problemas Vckovski propone la utilización de *Virtual Data Sets (VDS)*. La idea central es no proponer un standard para la estructuras de datos usadas (por ejemplo la estructura física de archivos), sino describir un conjunto de interfaces bien definidas. Dichas interfaces proveen métodos de acceso a los datos. Esto permite a un *data set* incluir métodos *inteligentes* para proveer los datos requeridos y ayuda a evitar los problemas asociados con la diversidad de los datos geográficos. Una aplicación que usa VDS no *lee* los datos desde un archivo, sino que invoca los métodos correspondientes definidos en el VDS, quien retornará los datos pedidos. Dependiendo del dominio de la aplicación, un VDS puede devolver datos pre-almacenados o calcularlos. Vckovski también propone la utilización VDS para información continua sobre la superficie terrestre.

4.0.11 Open GIS

En 1994 el *Open GIS Consortium* fue fundado por varias organizaciones GIS para solucionar el problema de standards incompatibles dentro de la tecnología de información geográfica. La ausencia de standards hace difícil la integración entre los sistemas de información geográfica de distintas organizaciones.

El standard OpenGIS propone una arquitectura genérica de muy alto nivel para las aplicaciones GIS. Dicho standard (*OpenGIS Abstract Specification*) se está dividido en tópicos, entre los que se destacan:

- Feature Geometry
- Spatial Reference Systems
- Locational Geometry Structures
- Stored Functions and Interpolation
- Coverage

Cada uno de estos items define la funcionalidad que deberían brindar los componentes de una aplicación GIS.

En su intento por imponer este standard, OpenGIS ha lanzado recientemente un conjunto de recomendaciones de implementación (*OpenGIS Implementation Specifications*), que implementan parte del *OpenGIS Abstract Specification* para ciertas plataformas distribuidas. Esta especificación abarca tecnologías:

- Corba
- OLE/COM
- SQL
- Interfaces para WebServers
- OpenGIS Geography Markup Language (GML)

4.1 Modelos Matemáticos

En esta sección se presenta una breve introducción acerca del arte y ciencia de los modelos matemáticos.

Definiciones

Un modelo es una representación de la realidad. Algunos autores sugieren que un modelo es una versión de situaciones del mundo real. Sin embargo, muchos factores influyen para que la representación sea una simplificación de la realidad, ciertos elementos son omitidos y se reduce la complejidad; entre esos factores tenemos:

- La complejidad del mundo real.
- Inexactitud en las medidas.
- Hipótesis imperfectas.
- Aproximaciones
- Discretizaciones de espacio y del tiempo.

Muchas definiciones de modelos tienen formas matemática. [CA85] definen los modelos como descripciones matemáticas acerca de procesos físicos. [Cas89] define modelo como:

“un encapsulamiento de una parte del mundo real dentro de los confines de las relaciones constituyendo un sistema matemático formal... Un modelo es una representación matemática de la realidad del modelador, una manera de capturar algunos

aspectos de una realidad dada dentro de un marco de elementos matemáticos, que nos proveen mecanismos para inspeccionar las propiedades de la realidad reflejada en el modelo.”

Más formalmente, un modelo puede ser definido como una representación formal de las relaciones entre cantidades y calidades definidas. Dichas cantidades y calidades describen objetos del mundo real. Esto contrasta con las entidades abstractas típicos de la matemática. En este sentido *modeling* siempre se aplica y define dentro del contexto de un sistema natural. Las expresiones formales proveen herramientas que el modelador puede usar para confirmar que el modelo es válido con respecto a la realidad, que puede ser testeado contra la realidad y que pueden realizarse predicciones acerca de la realidad.

Propósito de los modelos

Los modelos se pueden usar para:

- Evaluar las consecuencias de ciertas decisiones.
- Asistir en procesos de planeamiento.
- Desarrollar sistemas y tecnologías de control optimal.
- Proyectar fenómenos y variables ambientales futuras.
- Asistencia en la interpretación y análisis de monitoreo de datos ambientales.

Desde el punto de vista científico los modelos se construyen para mejorar el entendimiento de los sistemas naturales.

En muchos casos el objetivo de los modelos es lograr una predicción. La similaridad con el mundo real es importante, pero no determinante respecto de la calidad del modelo, es estos casos los modelos deberían ser juzgados por su utilidad como algoritmos para correlacionar observaciones y hacer predicciones.

Tipos de Modelos

Los modelos matemáticos se pueden clasificar de muchas maneras; las dimensiones por las cuales pueden ser clasificados incluyen: comportamiento relativo al tiempo o al espacio; tipos de datos, parámetros y expresiones usadas; estructura del modelo y tipo de matemática usada. El comportamiento relativo al tiempo determina si los resultados dependen de condiciones existentes en el momento del cálculo o en condiciones previas, típico en los modelos recursivos. Otro aspecto atañe a si las reglas de interacción y los parámetros varían en el tiempo. El comportamiento relativo al espacio los cálculos realizados para una posición en particular son afectados por otras posiciones. Dinámicos y estáticos son los extremos opuestos de un espectro en términos de comportamiento relativo al tiempo y al espacio. Los sistemas dinámicos son aquellos que cambian en el tiempo y posiblemente en el espacio, pueden presentar estados iniciales, cambiantes y constantes. Los

estados cambiantes solo pueden ser modelados usando ecuaciones diferenciales, mientras que, los estados constantes, por definición, corresponden a condiciones donde las derivadas son cero y por lo tanto pueden ser modelados con ecuaciones algebraicas, las oscilaciones alrededor de un estado constante pueden ser modelados con formulas matemáticas continuas. Los modelos estáticos asumen comúnmente variables y parámetros independientes del tiempo y el espacio. Los cambios en el espacio y el tiempo pueden ser continuos o discretos. Los modelos geográficos incorporan dependencia entre diferentes posiciones de manera que el valor en una posición depende de otras.

El rol de los datos en el modelo

Zeigler [Zei76] identifica tres niveles diferentes de validez de un modelo: el grado en que la salida del modelo coincide con la salida del sistema natural; la correspondencia entre las predicciones generadas por el modelo y las observaciones; y el grado en que la estructura del modelo coincide con la estructura del sistema natural. Todas esas condiciones requieren datos experimentales u observaciones acerca de la realidad que esta siendo modelada. Está claro que los datos juegan un rol crítico en desarrollo del modelo y su operación. Es importante ver que los efectos de los métodos de recolección de datos tienen un impacto innegable en el diseño y validación de los modelos.

4.1.1 Orientación a objetos

El estilo dominante de la programación ha sido el estructurado, la estructura del programa se organiza alrededor de funciones que deben ser realizadas, normalmente en una estructura de llamadas a procedimientos *top-down*. Un lenguaje orientado a objetos (OO), es aquel donde los programas son organizados basándose en los objetos que lo componen, normalmente en varias jerarquías de objetos que pueden compartir (heredar) comportamiento (métodos). OO se ha convertido en uno de los paradigmas predominantes para construir grandes sistemas.

La eficiencia en el desarrollo de sistemas dependen de cuan facilmente ellos pueden ser modificados y mejorados. Los cambios de un sistema que evoluciona tienen que ver tanto con la funciones como con la estructura de los datos. Los lenguajes procedurales sirven para localizar esos cambios a nivel de función, a través de procedimientos, funciones y librerías de rutinas; pero los cambios en la estructura de los datos, a menudo desatan cascadas de efectos laterales, en todos aquellos lugares donde las estructuras eran referenciadas. La programación OO permite localizar y acotar el impacto de esos cambios.

Un objeto comprende dos aspectos fundamentales, su estado interno (manifetado a través de sus variables de instancia), que ninguna otra parte del sistema puede acceder directamente, y un conjunto de procedimientos (llamdos métodos) que describen su comportamiento. Todo acerca de un objeto está encapsulado dentro de él y la única manera de extraer datos o cambiarlos, es a través del envío de mensajes. Muy frecuentemente, objetos de diferentes clases pueden ser similares, presentando leves modificaciones en el comportamiento, en otras palabras varían las versiones de algunos métodos. En esos casos una clase pued ser definida como subclase de otra, de manera que pueda heredar el comportamiento en común. La orientación a objetos que incluye los conceptos de encapsulamiento y herencia, brinda los siguientes beneficios:

- Los módulos del sistema se comunican mediante interfaces bien definidas, lo cual simplifica

la tarea de encontrar *bugs*

- Facilita la adaptación del software, reduciendo los costos de mantenimiento.
- Permite hacer que el sistema evolucione desde un prototipo hasta el producto final.
- OO es muy conveniente para la adaptación de datos altamente estructurados.

Capítulo 5

Breve Descripción del Framework para Aplicaciones GIS

Este trabajo se ha realizado en el marco del desarrollo de un framework orientado a objetos que permite representar los aspectos inherentes a las aplicaciones GIS. En este capítulo presentaremos algunas generalidades respecto al framework, haciendo hincapié en los diferentes subsistemas que lo componen, ellos son:

- Sistema de Referencia
- Topologías
- Representación de la información (*Appearances*)

La implementación de este trabajo se basa en dichos componentes. El sistema de referencia es utilizado para posicionar los puntos en el espacio, las topologías permiten describir la forma de los objetos, y las *appearances* para visualizar esos objetos en una interface amigable para el usuario.

En el marco de este framework todos los objetos geográficos poseen una posición geográfica y una forma. En términos más concretos todas las instancias de alguna subclase de la clase *GeoObject* conocen a una instancia de la clase *Location* (ver Figura 5.1).

Todo objeto georreferenciable deberá ser instancia de una subclase de *GeoObject*, ya sea un objeto discreto o uno continuo (campo continuo). Asociada a la *location* de un objeto geográfico se encuentra el sistema de referencia que permite interpretarla. El núcleo de cualquier GIS es su sistema de georreferenciamiento, conocido como sistema de referencia. Los aspectos más relevantes de un sistema de referencia son:

- La representación de la Tierra.
- La orientación y el desplazamiento de sus ejes.
- El sistema de coordenadas.

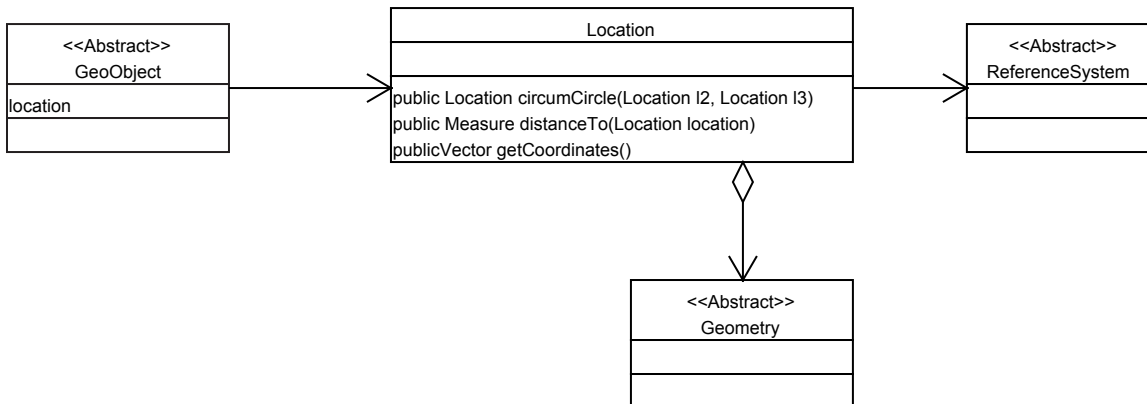


Figura 5.1: Objetos geográficos

Además de la necesidad de posicionar un objeto sobre la Tierra, es necesario poder determinar su *forma*. Dicha forma se conoce como la *topología* de un objeto geográfico. La topología explica a través de puntos, líneas y polígonos se relacionan entre sí la forma de los objetos, y es la base para la organización espacial de los objetos geográficos en un GIS. La topología provee el lenguaje básico para describir las características espaciales de los objetos. La topología de los objetos geográficos también se asocia a la *location*.

Toda esta información debería, además, ser presentada al usuario de forma que pueda ser fácilmente comprendida, para agilizar la toma de decisiones.

La arquitectura del framework fue dividida en tres partes fundamentales que detallaremos en las siguientes secciones.

5.1 Sistema de Referencia

Georreferenciar un objeto implica conocer la posición exacta de un objeto sobre la Tierra. Posicionar objetos involucra consideraciones respecto a:

1. Métricas para las medidas. Por ejemplo: *latitud-longitud, xyz, etc.*
2. Tipos de sistemas de referencia. Por ejemplo: cartesianos o polares.
3. Naturaleza del origen: una elipse, una esfera, un punto o un geoide.
4. El carácter geométrico del sistema de referencia. Por ejemplo: tamaño del origen.

Debido a la forma irregular de la superficie terrestre hay muchas maneras de representarla. De hecho, cada país elige el sistema de referencia que se adecua mejor a su situación, de acuerdo a la forma y localización sobre la Tierra. El valor de la posición de una entidad geográfica depende del sistema de referencia que se esté utilizando, es decir, el sistema de referencia permite interpretar

una posición geográfica. La mayoría de los GIS solo permiten trabajar con un sistema de referencia a la vez. La arquitectura orientada a objetos que mostramos en la figura 5.2 brinda la posibilidad de trabajar al mismo tiempo con posiciones definidas en sistemas de referencia heterogéneos.

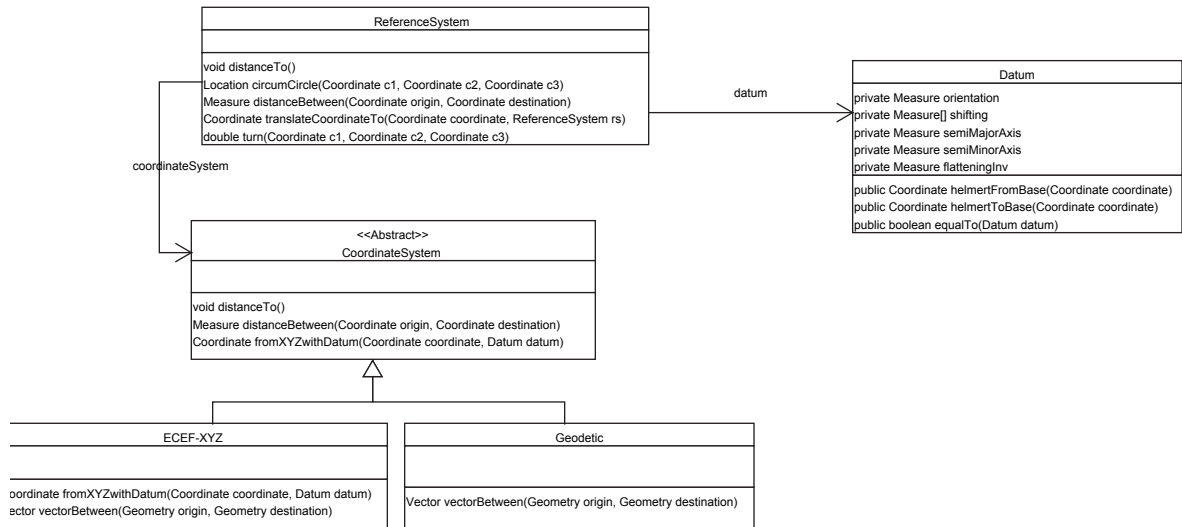


Figura 5.2: Diagrama de clases de la subarquitectura del sistema de referencias

A continuación daremos una breve descripción de las responsabilidades de cada clase en la arquitectura del sistema de referencias:

- *ReferenceSystem*: esta clase genera un marco espacial para una entidad geográfica, brindando la información y el comportamiento necesario para manipular los diferentes sistemas de coordenadas y unidades involucradas.
- *Datum*: define el tamaño y la figura de la Tierra, también define el origen y orientación del sistema de coordenadas. Los atributos *semiMajorAxis* y *flattening* permiten configurar las dimensiones del elipsoide que mapea a la Tierra, y con los atributos *orientation* y *shifting* se pueden orientar y desplazar el origen de los sistemas de coordenadas.
- *CoordinateSystem*: es una clase abstracta que constituye el marco lógico y matemático para las coordenadas, brindándole una unidad y permitiendo realizar cálculos entre objetos geográficos.

Esta clase debe contemplar dos tipos de operaciones:

1. el cambio de sistema de coordenadas
2. cálculos matemáticos en diferentes sistemas de coordenadas

- *ECEF-XYZ*: clase concreta que representa al sistema de coordenadas ECEF-XYZ. Entre sus responsabilidades se encuentran la de realizar el cálculo de distancias bajo éste sistema de coordenadas, y también poder transformar coordenadas de éste sistema a los otros sistemas de coordenadas.
- *Geodetic*: clase concreta que representa al sistema de coordenadas Geodésicas. Entre sus responsabilidades se encuentran la de realizar el cálculo de distancias bajo éste sistema de

coordenadas, y también poder transformar coordenadas de éste sistema a los otros sistemas de coordenadas.

5.2 Modelo Topológico

Una entidad geográfica posee, además de su posición, una *forma*. Existen tres elementos básicos que permiten la descripción de la forma de un objeto geográfico: puntos, arcos y polígonos. El polígono se define como un conjunto de arcos adyacentes, un arco es una línea entre dos puntos (ver Figura 5.3).

El modelo topológico se utiliza para estudiar propiedades espaciales entre los objetos, como adyacencia, conectividad, “contiene a”, etc. Una ciudad puede ser pensada como un polígono, sus calles como arcos, las construcciones como polígonos más pequeños, contenidos en la ciudad y adyacentes a las calles.

Existen entidades que no pueden ser representadas por un elemento topológico básico, sino que requieren polígonos disjuntos o polígonos con agujeros o alguna otra combinación de elementos básicos o complejos. Por ejemplo la región continental de un país puede representarse por un polígono, si el país además esta formado por islas, la forma que toma el país es la de un *polígono compuesto*, formado por el polígono mayor (parte continental) y los más pequeños (las islas). A su vez, si estamos modelando un continente, su forma estará dada por la unión de las topologías de sus países. Por lo tanto vemos que hay una relación “contiene a” que es recursiva. El modelo topológico de este framework permite describir cualquier *forma*, brindando la posibilidad de “componer” polígonos y arcos. Es decir, definir polígonos formados por otros polígonos básicos o compuestos, dicha relación composición no tiene un límite teórico, sino que se puede extender tanto como sea necesario para describir adecuadamente las entidades geográficas en cuestión.

A continuación daremos una breve descripción de las responsabilidades de cada clase en la arquitectura de las topologías:

- *TopoGeometry*: es una clase abstracta que modela el comportamiento general de la topología de una entidad. Para esto tiene información sobre su forma y un protocolo general para las operaciones sobre ella misma o con respecto a otras (e.j.: inclusión, conectividad, distancia a otros objetos, bounding box).
- *Polygon*: es una clase abstracta que modela el comportamiento de polígonos en general. Se consideran distintos tipos de polígonos: simples, disjuntos, con agujeros y algunas combinaciones entre ellos. Estos tipos, salvo el simple, se consiguen componiendo los demás, para esto se ha utilizado el patrón de diseño *Composite*, en el que esta clase (*Polygon*) representa la clase *Component* de la estructura del pattern. Los polígonos simples están representados mediante una cadena cerrada de arcos, representados por la clase *Arc*. Algunas de las operaciones que pueden citarse como ejemplo pueden ser: unión, intersección, cálculo de centroides, cercos convexos, entre otras.
- *Arc*: modela los *arcos* de la topología. Todo arco comienza en un nodo y termina en otro, por lo que la clase *arc* conoce sus nodos de inicio y fin. En caso de formar parte de polígonos, también conoce a éstos para poder tener información sobre posiciones relativas.

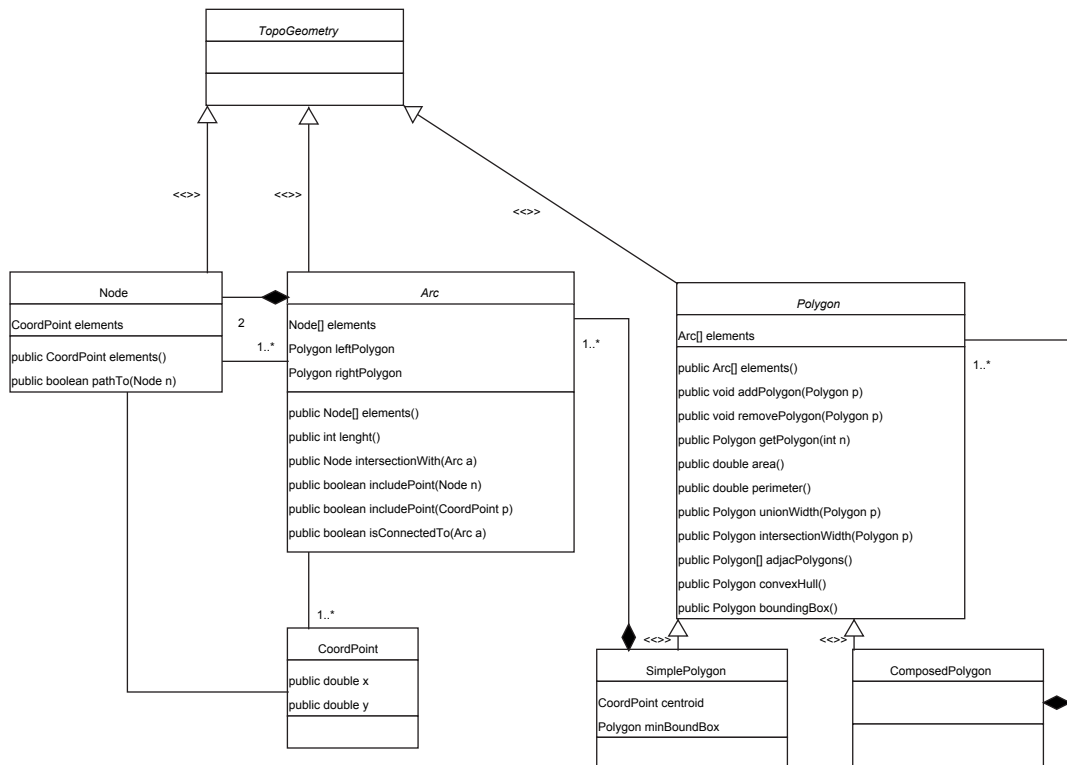


Figura 5.3: Diagrama de clases de la subarquitectura para topologías

- *Node*: representa a los nodos. Toda entidad cuya posición sea puntual están representadas de esta manera. *Node* conoce una *Coordinate* que representa su posición, esta *coordinate* representa la posición en un sistema de referencias determinado. Por lo tanto las instancia de *Node* se abstraen de esta representación.
- *SimplePolygon*: clase concreta que representa un polígono simple. Conoce una cadena cerrada de arcos que representa su perímetro.
- *CompositePolygon*: clase concreta que representa a los polígonos compuestos, es decir los disjuntos, los polígonos con agujeros y todas las combinaciones posibles. Para lograr estas composiciones conocen instancias de *Polygon*, que pueden ser simples o compuestas, de esta manera es posible definir cualquier polígono que sea necesario

5.3 Visualización de la información

Una de las características más importantes de los GIS es su habilidad para mostrar las entidades geográficas, con sus características (posición y forma), de manera que la información sea fácilmente interpretada por el usuario. Dichas entidades deben ser presentadas teniendo en cuenta aspectos fundamentales, como su posición georreferenciada y su topología.

En los SIG hay situaciones en las que las entidades geográficas tiene que ser mostradas de diferentes maneras, posiblemente en forma simultánea. Por ejemplo, consideremos un sistema que permite tener varias vistas a la vez del mismo mapa a escalas diferentes, por lo tanto el mismo objeto puede aparecer en las distintas vistas mostrando diferentes características. De hecho, ciertos objetos pueden no visualizarse en determinadas vistas.

Además, hoy en día existen varios motores gráficos poderosos, como las implementaciones de OpenGL, DirectX, Java 3D, entre otras. Por lo tanto, la forma en la que se visualiza un objeto también debería proveer un mecanismo que permita mostrarla por medio de cualquiera de dichos motores gráficos.

De lo dicho anteriormente se desprende que la “apariencia” de un objeto geográfico en una vista debería estar separada del objeto en sí mismo.

En la figura 5.4 presentamos el diagrama de clases de la subarquitectura de *Appearances*, que permite definir atributos visuales a los objetos geográficos.

A continuación daremos una breve descripción de las responsabilidades de cada clase en la arquitectura de *Apperance*:

- *Appearance*: es la responsable de los detalles relacionados a la representación visual de los objetos geográficos.
- *AppProperties*: esta clase desacopla de la apariencia la responsabilidad de administrar las propiedades que se aplicarán a las figuras (*shapes*). Un instancia de *Appearance* conoce a un objeto *AppProperties*. Por ejemplo, si se va a mostrar una apariencia compuesta por polígonos, los polígonos que no necesitan mostrarse con algún rasgo distintivo, pueden compartir la misma instancia de la clase *AppProperties*. De otra manera, si algún polígono necesita distinguirse de alguna forma, es suficiente con asignar a su apariencia las características deseadas.
- *Shape*: La forma de una apariencia puede ser un “literal” (texto), “arbitrario” (e.j.: alguna imagen seleccionada por el usuario), o “derivada” (de su posición).
- *GraphicsEngine*: es la responsable de dibujar a las apariencias. Cada subclase de la clase *GraphicsEngine* provee una implementación para un motorgráfica en particular. Por ejemplo, si se requiere una representación en 3 dimensiones, se instanciará la clase *OpenGLGraphicsEngine*. Una ventaja de esta jerarquía es la de hacer a las apariencias absolutamente independientes de las capacidades gráficas del sistema. Otra ventaja es la de permitir vistas múltiples y simultáneas de la misma apariencia.

5.4 Campos Continuos

La representación y manipulación de campos continuos es una de las actividades más desafiantes al desarrollar una aplicación GIS. Ya hemos descrito los distintos aspectos que debemos tener en cuenta cuando se trabaja con información continua en un medio computacional, el *spatial data model*, métodos de estimación, entre otros. En esta sección describiremos como se modelan dichos aspectos en el framework.

Hasta ahora hemos tratado distintos aspectos relativos al manejo de información continua, como ser: medición, unidades, precisión y representación discreta de esos datos. También hemos introducido el contexto en el cual se desarrolla este trabajo, es decir, un framework que permitirá representar los aspectos esenciales de las aplicaciones GIS.

En esta sección mostraremos como se representan los campos continuos en el marco del framework orientado a objetos descrito en este capítulo. Dicha arquitectura, presentada en [GB98] contempla los aspectos presentados hasta ahora: muestreo de datos, representación de los datos y métodos de estimación.

La primera abstracción que surge en el modelo es la de campo continuo, la cual será representada por la clase *ContinuousField*.

Definiremos primero la interface de esta clase, lo cual permitirá abstraernos de su representación. Como se desprende de lo enunciado hasta ahora, un campo continuo debería proveer alguna forma de conocer el valor del fenómeno representado en una posición determinada. Definimos entonces el mensaje: *valueAt(Location l)*. Dado que el campo es un modelo discreto de una realidad continua este punto puede ser alguno de los muestreados o no, de todas formas la manera en que se obtenga el valor tiene que ver más con su representación y funcionamiento interno, y nos ocuparemos de ello más adelante. Por lo pronto también sabemos que los campos representan un fenómeno y que están definidos por una discretización, es decir una muestra, así que es lógico que provean un protocolo para asignar y recuperar esa información (ver Figura 5.5).

5.4.1 Representaciones para campos continuos

A continuación analizaremos un aspecto de vital importancia al definir un campo continuo, su modelo de datos espacial (*spatial data model*) o representación. Como vimos en el Capítulo 2, al discretizar la información continua nos vemos en la necesidad de organizarla de manera que mejor simule la realidad (fenómeno) que queremos modelar. Dentro de los modelos de datos espaciales más utilizados en la actualidad podemos mencionar: Grillas de Puntos Regulares, Grilla de Puntos Irregulares, Triangulated Irregular Networks (TIN), entre otros. Especialmente los TINs son ampliamente aceptados debido a su relativamente sencilla implementación y buen desempeño (poca introducción de error en las estimaciones) en tareas como el modelado de terreno. Vemos entonces que un campo puede tener diferentes representaciones y estas podrían ir cambiando con el tiempo, de acuerdo a las necesidades del usuario. La primera alternativa de diseño consiste en subclasificar, implementando en cada subclase un modelo de datos espaciales, lo cual implicaría que al momento de cambiar de representación sea necesario instanciar un nuevo campo y migrar los datos de uno a otro, además notificar a todos los objetos que lo conocían que deben cambiar de colaborador. Otra alternativa consiste en definir una jerarquía de modelos de datos espaciales, que actuarán como colaboradores de un campo continuo. Entonces existirá una sola clase de campo continuo, que podrá estar configurada con diferentes representaciones, aislando a los clientes de los cambios de representación del campo continuo (ver Figura 5.6).

Esta micro arquitectura es una instanciación del patrón de diseño *Bridge* [GRJV95] que permite desacoplar una abstracción (el campo continuo) de su implementación de manera que puedan variar independientemente ¹.

¹Para más información acerca del patrón *Bridge*, puede referirse al Apéndice B.2 (página 122).

5.4.2 Métodos de estimación

El siguiente aspecto que define en cierta medida el comportamiento de un campo continuo es el método de estimación. Un método de estimación representa un algoritmo que permite interpolar el valor que toma un fenómeno en un punto a partir de la muestra. Los métodos de estimación están estrechamente relacionados con el *spatial data model* que representa la estructura del campo. Por ejemplo, si un campo continuo tiene una representación de TIN, el método de estimación por excelencia es el de coordenadas baricéntricas, pero podría usarse algún otro que aproveche la triangulación de los datos para interpolar. Otro campo que esté usando una representación de grilla irregular podría tener como método de estimación asociado un algoritmo de *Weighted Neighbors*, o un algoritmo polinomial como *Spline* o *B-Spline (Bicubic Spline)*. Podemos ver al método de estimación como una función de la representación (que contiene la muestra) y un punto. Sea F el método de estimación, r la representación y el punto p .

$F(r, p)$ =valor del fenómeno estimado en p .

Es evidente que de la misma forma en que las representaciones pueden variar, también lo pueden hacer los métodos de estimación, por lo que resulta conveniente separar esta nueva abstracción en un jerarquía de clases independiente, de forma tal que podamos agregar nuevos métodos de estimación en el futuro, y esta modificación no afecte el resto de la arquitectura. Al separar esta jerarquía, lo que tenemos es una aplicación de un patrón muy utilizado, conocido como *Strategy* [GRJV95] cuyo objetivo es: “definir un conjunto de algoritmos, encapsularlos y hacerlos intercambiables, permitiendo que la implementación de los algoritmos varíe de forma independiente de los clientes que las usan”. La abstracción *método de estimación* estará representada en un jerarquía cuya raíz es la clase abstracta *EstimationMethod* que define un protocolo muy simple, cuyas subclases deben respetar, y que permite interpolar dados una representación y un punto (*location*), el valor del que toma el fenómeno en ese punto (figura 5.7).

Métodos de estimación por defecto

Como dijimos anteriormente los métodos de estimación están ligados a ciertas representaciones. Por lo general existe un método de estimación preferido para cada representación. Podemos expresar esa preferencia usando el patrón de diseño *Abstract Factory*, cuyo objetivo es “proveer una interface para crear familias de objetos relacionados o dependientes, sin especificar sus clases concretas. De esta manera a cada representación se le puede solicitar (mediante el envío de un mensaje) el método de estimación por defecto, la respuesta variará de acuerdo a que representación recibió el mensaje.

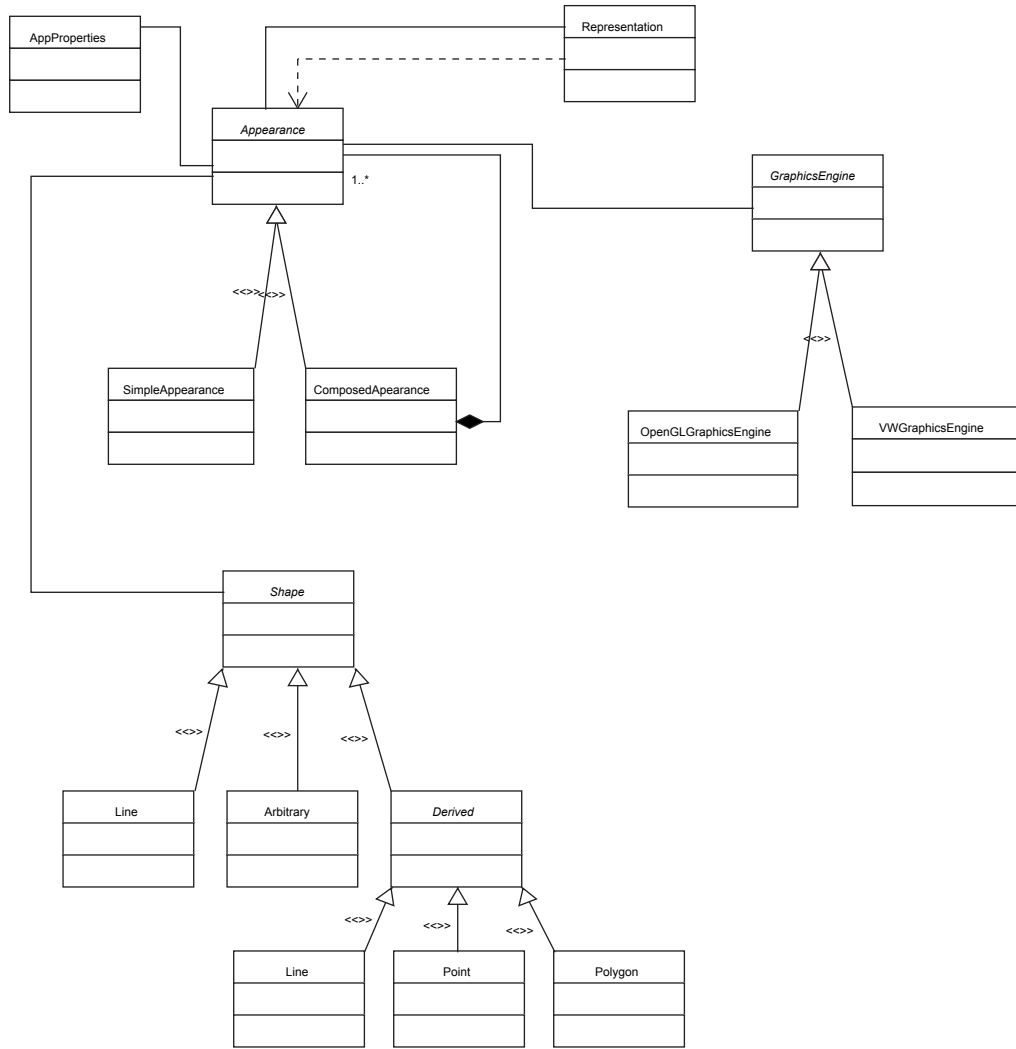


Figura 5.4: Diagrama de clases de la subarquitectura de Appearances

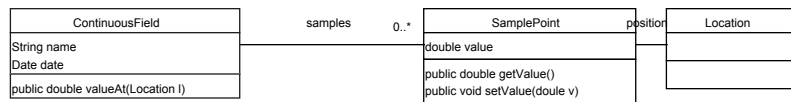


Figura 5.5: Diagrama de clases: Campo continuo y muestra

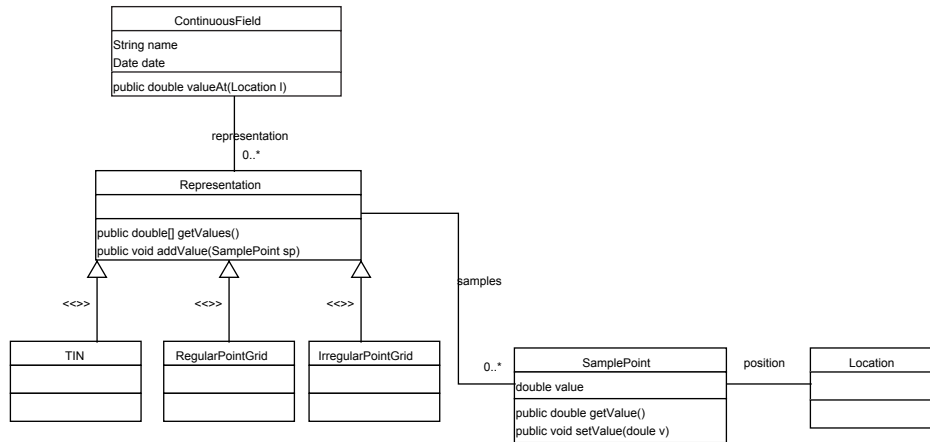


Figura 5.6: Diagrama de clases: Campo continuo y representaciones

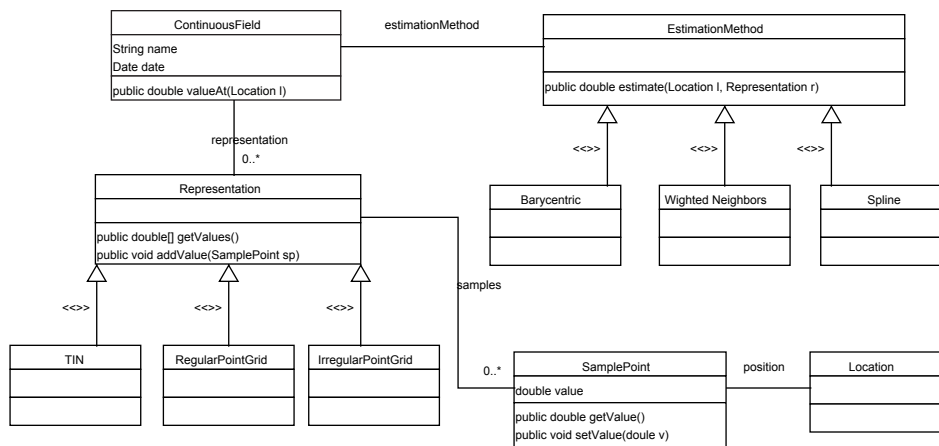


Figura 5.7: Diagrama de clases: Campo continuo, representaciones y métodos de estimación

Capítulo 6

Una Arquitectura OO para Operaciones con Campos Continuos

Este capítulo presenta el núcleo de nuestro trabajo. Aquí se describe un conjunto de operaciones y se detallan los problemas de compatibilidad espacial que se pueden presentar. Luego se describe una arquitectura orientada a objetos que permite implementar dichas operaciones, resolviendo los problemas de compatibilización espacial. Finalmente presentamos la forma en la que puede extenderse la arquitectura de manera que soporte futuras operaciones.

6.1 Clasificación de Operaciones para Campos Continuos

En esta sección se presenta un conjunto de operaciones sobre campos continuos, teniendo en cuenta la variedad y la complejidad de las mismas.

La clasificación propuesta se basa en la complejidad de los factores que deben tenerse en cuenta a la hora de resolver una determinada operación. Esta complejidad, en la mayoría de los casos esta dada por la cantidad de campos continuos que intervienen en la operación, dado que esto define una serie de consideraciones a solucionar en el momento de resolverla. Así, la idea es dividir a las operaciones en unarias y n-arias.

Tenemos entonces dos grandes tipos de operaciones:

1. Unarias: donde interviene un campo continuo y posiblemente algún otro valor de distinto tipo.
2. N-arias: donde intervienen dos o más campos continuos.

6.1.1 Operaciones Unarias para Campos Continuos

Consideraremos unarias a aquellas operaciones en las que interviene un campo continuo, incluyendo aquellas en las que sea necesario especificar algún otro parámetro adicional. A continuación describiremos cada una de ellas.

- *Selección por Condición:* Esta operación de selección toma un campo continuo y un parámetro, conocido como criterio de selección. El resultado de aplicar esta operación es un nuevo campo continuo que cumple con el criterio de selección.

Por ejemplo: se tiene un campo continuo que representa la cantidad de agua precipitada en un sector del planeta y queremos saber donde conviene plantar un vegetal que necesita al menos 300 mm de lluvia, esta consulta se puede realizar mediante una selección de los sectores del campo continuo que superan los 300 mm de precipitaciones (Figura 6.1).

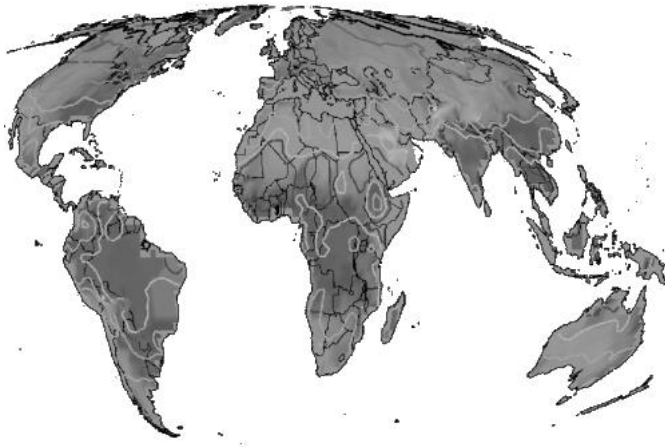


Figura 6.1: Ejemplo de una selección

- *Selección por Zona:* La selección por zona toma un campo continuo y una zona, definida mediante una topología de polígono y retorna un nuevo campo continuo que tiene por dominio dicha zona. Por ejemplo: se dispone de un campo continuo que representa el fenómeno temperatura en todo el país. Se desea realizar un cálculo teniendo en cuenta la temperatura de la provincia de Salta. Para ello se debe aplicar al campo la operación de selección por zona utilizando como parámetro una zona definida por la topología de la provincia de Salta.
- *Gradiente:* Esta operación calcula el vector gradiente en cada punto de la muestra del campo continuo al que se aplica ésta operación. El resultado es un campo continuo vectorial, con los correspondientes vectores gradientes asociados a cada uno de los puntos de la muestra.
- *Integral:* Esta operación se aplica a campos escalares y retorna el volumen debajo de la superficie del fenómeno, limitado por el dominio del campo. Esta operación se resuelve mediante el método de diferencias finitas.
- *Máximo, Mínimo y Promedio:* Estas operaciones retornan los valores extremos y el promedio del campo, sólo pueden ser aplicadas a campos escalares.

6.1.2 Operaciones N-arias para Campos Continuos

Las operaciones aplicadas a varios campos continuos presentan un complejidad singular, originada especialmente en la variedad de representaciones (spatial data models, [Kem97a] y [Kem97b]) y métodos de interpolación que se ven involucrados en su resolución. Además, cabe destacar que al operar con múltiples campos continuos es común encontrarse con que ellos no han sido definidos en el mismo dominio y/o que las posiciones geográficas que conforman la muestra tampoco coinciden.

Estos conflictos tienen su origen en la naturaleza del fenómeno que modelan estos campos. Por lo general, ciertos tipos de modelos de datos espaciales representan con mayor fidelidad ciertos fenómenos. La naturaleza del fenómeno también incide en la elección de los puntos que conformarán la muestra y en la definición del dominio.

Tomemos como ejemplo la operación *Average* que toma n campos continuos y retorna como resultado el campo *promedio*. Si aplicamos la operación *Average* a dos campos F_1 y F_2 , representados por:

$$F_1=(d_1,s_1) \text{ y } F_2=(d_2,s_2)$$

Donde d_1 y d_2 son los dominios de los campos F_1 y F_2 respectivamente y s_1 y s_2 sus muestras.

Al operar con campos debemos establecer como se formaran el dominio y la muestra del campo continuo resultante. En este caso consideraremos únicamente las operaciones de *unión* e *intersección*, tanto para dominios como para muestras. El campo resultante

$$F_3 = Average(F_1,F_2)$$

puede estar formado por alguna de las siguientes combinaciones de muestras y dominios:

$$F_3=(d_1 \cup d_2 ,s_1 \cup s_2)$$

$$F_3=(d_1 \cap d_2 ,s_1 \cup s_2)$$

$$F_3=(d_1 \cup d_2 ,s_1 \cap s_2)$$

$$F_3=(d_1 \cap d_2 ,s_1 \cap s_2)$$

La lista de combinaciones puede incrementarse arbitrariamente, de acuerdo a la cantidad de estrategias que se definan para generar los dominios y las muestras del campo continuo resultante.

Además de esto, debe considerarse que los campos F_1 y F_2 pueden estar definidos utilizando diferentes representaciones. Entonces es necesario definir cual será la representación del campo continuo de salida. Para ello deben aplicarse los criterios presentados en el Capítulo 2 de este trabajo.

Como podemos observar, la complejidad que implican este tipo de operaciones hace necesario un cuidadoso estudio, de manera tal que sea posible obtener una arquitectura de software capaz de soportar todas las formas de representación, métodos de estimación, etc.

Quizás el reto más importante al que nos enfrentamos los desarrolladores e investigadores de software es el de obtener componentes efectivamente reusables y adaptables a futuros requerimientos. Sin duda, esta premisa es la que ha motivado el desarrollo muchas de las tecnologías actuales en el mundo del software, como la orientación a objetos y componentes, entre otras.

Este caso no escapa a la regla general, dado que los GISs representan una gama de aplicaciones que van desde el catastro hasta las aplicaciones ambientales, el rango de requerimientos para operaciones con campos continuos es amplio y variado. Además de presentar un conjunto de operaciones para la manipulación de campos continuos, en este trabajo nos hemos ocupado de diseñar una arquitectura sumamente extensible y adaptable, con el fin de satisfacer futuros requerimientos.

Operaciones puntuales

Las operaciones puntuales se aplican sobre un conjunto de campos continuos. Dichas operaciones *atraviesan* dos o más campos (la cardinalidad depende de cada operación) aplicando una determinada operación aritmética sobre el mismo punto (posición geográfica) en todos los campos, y luego asignando el resultado a la misma posición en el campo resultante.

En la Figura 6.2 puede observarse una representación esquemática de una operación puntual.

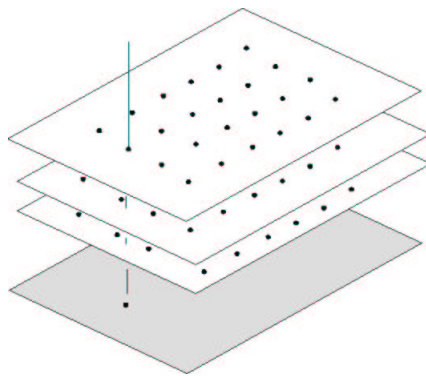


Figura 6.2: Esquema de una operación puntual

Las operaciones puntuales más utilizadas sobre campos escalares son: Suma, Resta, Producto, División y Promedio. Además en el caso de los campos vectoriales también se utilizan el producto escalar y vectorial.

Como vimos anteriormente es importante considerar dos aspectos fundamentales en la definición del campo continuo resultante, ellos son: el dominio y la muestra.

Construcción de muestras

Las muestras son conjuntos de pares $(location, value)$ que asocian un valor muestreado de un fenómeno a una posición georreferenciada. En primer lugar nos ocuparemos de la generación de las posiciones geográficas (*locations*). Cuando operamos con campos continuos y debemos definir cómo generar la muestra resultante, podemos aplicar las operaciones estándar para conjuntos, es decir, *unión*, *intersección*, *diferencia* y *diferencia simétrica*. Además es posible que el usuario desee definir explícitamente qué posiciones geográficas formarán parte de la muestra. Cada una de estas posibles estrategias tiene algunas consecuencias con respecto a la densidad de la nueva muestra y al error que puede introducir en el campo continuo resultante. A continuación enunciaremos cada una de ellas explicando sus consecuencias.

1. Muestra Arbitraria: Es posible que en ciertas ocasiones se desee generar la muestra del campo resultado en ciertos puntos elegidos por conveniencia, en este caso la densidad y la regularidad de la muestra dependerá de los puntos elegidos, así como la precisión en los datos dependerá de cuántos de esos puntos elegidos coincidan con las muestras de los campos de entrada.
2. Muestra Diferencia: Los puntos en el campo continuo resultante serán aquellos que pertenezcan a uno de los campos continuos, pero no al otro.
3. Muestra Unión: En este caso la muestra del campo resultado estará compuesta por todos los puntos que pertenecen las muestras de todos los campos. Es decir se realiza la unión de los conjuntos que forman las muestras de los campos de entrada. Debería ser utilizada cuando se desea obtener un campo resultado con una muestra bastante densa, esta condición de densidad puede ser beneficiosa a la hora de interpolar datos, pues los métodos de estimación trabajan mejor con datos cercanos al punto a interpolar, pero tiene como desventaja que muchos de los valores deberán ser interpolados en los campos continuos de entrada con la lógica pérdida de precisión en los valores asociados a los puntos que forman la muestra.
4. Muestra Intersección: En este caso se toma como puntos para la muestra del campo resultado aquellos que se encuentran en las muestras de todos los campos de entrada, se realiza la intersección de los conjuntos que representan las posiciones de las muestras en los campos de entrada. Esta opción proveerá una muestra para el campo resultado cuya densidad es menor o igual a la de los campos de entrada. Como principal ventaja se puede observar que para generar los valores asociados al campo resultado no será necesario realizar interpolaciones, con lo cual no se pierde calidad en la muestra. Como contraparte, las estimaciones generadas a partir del campo continuo resultante podrían llegar a ser menos precisas que en el caso anterior, donde se tienen más puntos en la muestra para realizar la interpolación.

Construcción de dominios

El dominio de un campo continuo es una zona geográfica, definida por una *location* con topología de polígono. La generación del dominio para el campo resultante se parece bastante a la de las muestras, dado que es posible aplicar a los polígonos las operaciones definidas para conjuntos (*unión*, *intersección* y *diferencia*). Elegir una u otra estrategia afecta la zona geográfica en la que podrán realizarse consultas en el campo resultante. El dominio actúa como una máscara que permite conocer el valor del fenómeno en ciertas zonas.

Un campo continuo puede tener puntos muestreados *fuera* del dominio. El objetivo de esta práctica es lograr una mayor precisión en las interpolaciones que se realizan cerca de los límites del mismo. Por lo tanto el dominio sirve para limitar la zona en la que se puede interpolar, pero no implica que no existan puntos de la muestra fuera de él.

También en este caso es posible que el usuario desee definir arbitrariamente el dominio del campo, por lo tanto la arquitectura deberá brindar la posibilidad de expresar esta situación.

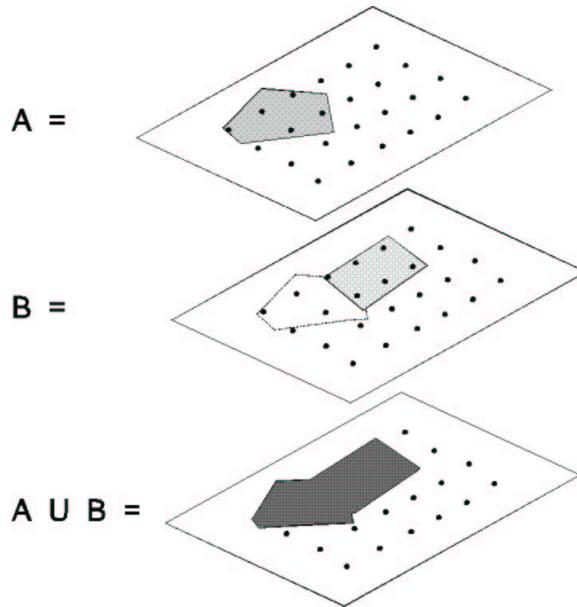


Figura 6.3: Esquema de una unión de dominios

Obtención de los valores asociados a la muestra

Como dijimos anteriormente las operaciones puntuales se caracterizan por la aplicación de una función a los valores asociados a una posición geográfica en todos los campos. El valor obtenido se asocia luego a la misma posición en el campo continuo resultante, pasando a formar parte de su muestra. Dichas posiciones geográficas se obtienen mediante alguna de las estrategias definidas anteriormente. Dado que en la aplicación de la función intervienen valores provenientes de todos los campos, estos pueden ser o no interpolados. Cuanto mayor sea la proporción de valores interpolados menor será la precisión del campo resultante. De hecho cada interpolación introduce un error propio de cada estimación. Al operar con varios valores interpolados sus errores crecen exponencialmente. Hasta la fecha no existen estudios acerca del grado de confiabilidad de los distintos modelos utilizados para representar fenómenos continuos. Dichos estudios deben ser realizados para cada fenómeno en particular, teniendo en cuenta características propias, contrastando los valores obtenidos a través de la modelización con mediciones exactas de la realidad. Pero, dado que la calidad del muestreo también influye en la performance del modelo, los estudios arrojarían resultados no extrapolables a nuevas situaciones.

Definición de la representación y método de estimación

Queda por resolver cómo definir la representación y método de estimación asociados al campo continuo resultante. En el capítulo 2 hemos presentado las diferentes representaciones (*spatial data models*) y las reglas que rigen las conversiones cuando es necesario combinarlas. Cada representación tiene un método de estimación asociado por defecto. De hecho, existe una relación muy estrecha entre ambos, ya que los métodos de estimación son específicos para cada representación. Debido a que existen varios métodos de estimación para cada representación es posible que el usuario desee especificar uno de ellos (uno distinto al método de estimación por defecto).

Por lo tanto sólo es necesario proveer una manera de aplicar las reglas presentadas anteriormente, que rigen los cambios de representación. Este asunto se trata con más detalle en la Sección 6.2.1.

En las siguientes secciones presentaremos la arquitectura orientada a objetos que permite resolver las operaciones hasta ahora enunciadas.

6.2 Arquitectura OO para operaciones con campos continuos

En lo que resta de este capítulo presentaremos una arquitectura orientada a objetos que permite manejar campos continuos a través de las operaciones que hemos presentado. Dicha arquitectura tiene en cuenta todos los aspectos relacionados con la definición de dominios, muestras, representaciones y métodos de estimación. Además presenta la posibilidad de agregar fácilmente nuevas operaciones.

La documentación de la arquitectura ha sido realizada siguiendo el estándar UML, *Unified Modelling Language*.¹

Presentaremos la estructura estática mediante diagramas de clase. La estructura dinámica será mostrada a través de diagramas de instancia. La colaboración entre las instancias se representará mediante ejemplos concretos, utilizando diagramas de interacción y colaboración. Por último se presentarán las Hot Spots [Pre94] que permiten especificar los puntos donde la arquitectura admite personalización y el agregado de comportamiento nuevo.

6.2.1 Estructura estática

Las operaciones unarias han sido incluidas en como métodos en los campos continuos. Por el contrario, debido a la complejidad que presentan las operaciones n-arias, han sido modeladas como objetos.

¹Para más información acerca de UML referirse al Apéndice A.

Podemos decir que para todas las operaciones n-arias debe seguirse la misma secuencia genérica de pasos:

1. Determinación de las posiciones de la muestra del resultado
2. Construcción del dominio del resultado
3. Generación de los valores asociados a la muestra
4. Determinación de la representación y método de estimación

Por lo tanto existirá un objeto encargado de dirigir la ejecución de estos pasos, a este objeto lo llamaremos *evaluador*. Dichos pasos se modelarán a su vez con otros objetos, permitiendo parametrizar al evaluador.

El hecho de modelar las operaciones como objetos proviene de la aplicación del patrón *Command*, descrito en [GRJV95], cuyo objetivo es: "encapsular un operación como un objeto, permitiendo parametrizar a los clientes con diferentes operaciones...". Para más información acerca del patrón *Command*, puede referirse al Apéndice B.3 (página 124).

El patrón *Command* permite modelar operaciones como objetos. En el caso del evaluador este representa la aplicación de una operación sobre un conjunto de campos. Esta aplicación de una operación modelada como un objeto debe ser parametrizada con las estrategias necesarias para definir el dominio, muestra, representación y método de estimación del campo continuo resultante. Estos colaboradores son referenciados por variables de instancia. El mensaje más importante que se le puede enviar a un evaluador es *evaluate(Collection continuousFields)*, que aplica una operación a la colección de campos continuos y retorna un campo continuo como resultado.

El objetivo de la clase *Evaluator* (Figura 6.4) y su comportamiento quedarán más claros hacia el final del capítulo, pero debemos presentarla ahora ya que juega un rol clave en la arquitectura.

Evaluator
OutcomeSampleStrategy sampleStrategy
OutcomeDomainStrategy domainStrategy
SpatialConversionRule conversionRule
PunctualOperation punctualOperation
public ContinuousField evaluate(Collection continuousFields)

Figura 6.4: La clase Evaluator

Construcción de Muestras

Las distintas opciones para la generación de muestras, presentadas en 6.1.2, han sido modeladas utilizando el patrón de diseño *Strategy* [GRJV95] cuya intención es: "definir una familia de algoritmos, encapsularlos y hacerlos intercambiables. Strategy permite al algoritmo variar independientemente del cliente que lo usa". Puede referirse al Apéndice B.5 (página 127) para obtener mayor información acerca de este patrón.

En el caso de nuestra arquitectura, el algoritmo encapsulado define la forma en que se genera la muestra del campo continuo resultante a partir de las muestras de los campos continuos de entrada. El método *buildOutcomeSample(Collection: samples)* toma un conjunto de muestras y genera una nueva. El evaluador de operaciones cumple el rol de *client* del patrón, delegando la construcción de la muestra en una instancia de alguna subclase de *OutcomeSampleStrategy*.

De esta manera la arquitectura brinda al usuario la posibilidad de elegir de qué manera se genera la muestra y permite la adición de futuras estrategias mediante la subclasificación de *OutcomeSampleStrategy* y la redefinición del método *buildOutcomeSample(Collection: samples)*.

La instanciación del patrón Strategy puede observarse en la Figura 6.5.

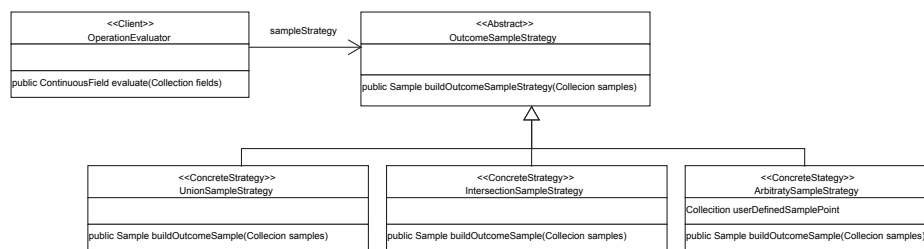


Figura 6.5: Instanciación de patrón Strategy. Construcción de muestras

Construcción de Dominios

Las estrategias presentadas en la sección 6.1.2 también fueron modeladas a través del patrón Strategy. Nuevamente el rol de cliente es representado por una instancia de la clase *Evaluator*. La jerarquía de estrategias para la generación de dominios tiene como raíz a la clase *OutcomeDomainStrategy*. Esta clase abstracta define como interfase el mensaje *buildOutcomeDomain(Collection: domains)* que debe ser redefinido por todas sus subclases concretas. Este mensaje recibe como parámetro una colección de dominios, y retorna un dominio, es decir una zona geográfica representada por una *location* con topología de polígono. Nuevas estrategias para la generación de dominios puede ser agregadas subclasificando a partir de *OutcomeDomainStrategy* y redefiniendo el mensaje *buildOutcomeDomain(Collection: domains)*.

La instanciación del patrón Strategy puede observarse en la Figura 6.6.

Generación de valores asociados a la muestra

Una vez establecidos el dominio y las posiciones que conformarán la muestra, es necesario establecer el valor que se asociará a cada una de esas posiciones.

Como explicamos anteriormente las operaciones puntuales son funciones que se aplican a los valores asociados (exactos o interpolados) a una posición en todos los campos continuos (ver Figura 6.2)

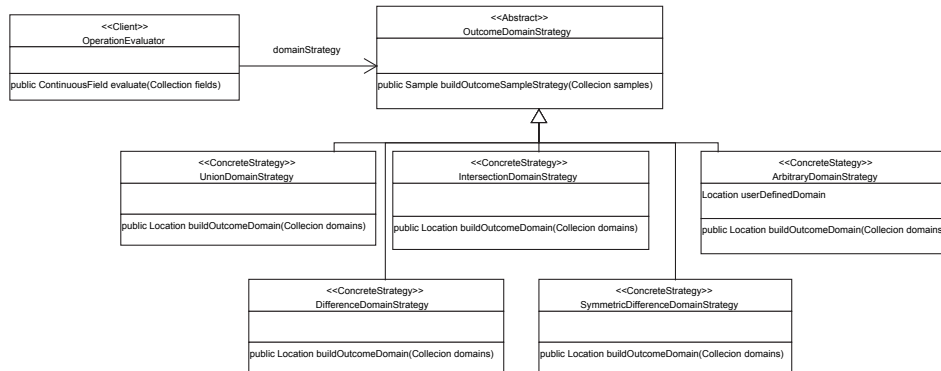


Figura 6.6: Instanciación de patrón Strategy. Construcción de dominios

Las diversas operaciones puntuales han sido organizadas en una jerarquía polimórfica, ver Figura 6.7.

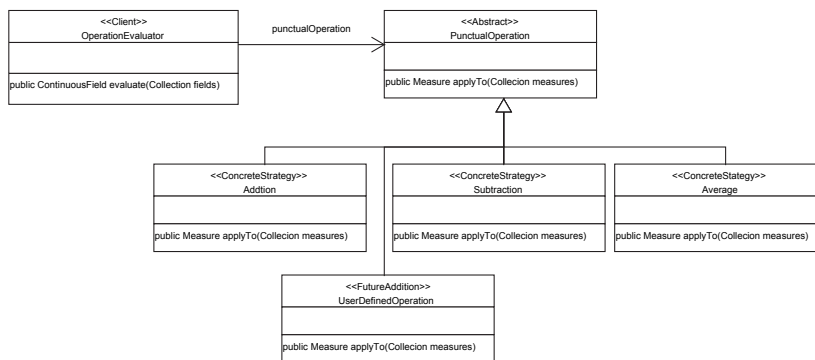


Figura 6.7: Instanciación de patrón Strategy. Operaciones puntuales

Para cada posición en la muestra del campo continuo resultante se obtienen los valores asociados a ella en los campos continuos de entrada. Los valores obtenidos pueden ser exactos (si se trataba de uno de los puntos de la muestra) o interpolados. A dicho conjunto de valores se le aplica la función correspondiente. El valor obtenido se asocia al punto de la muestra en el campo continuo resultante.

Determinación de la representación

La interacción de varios campos continuos en una operación involucra la necesidad de definir tácticas que permitan generar una nueva representación a partir de la combinación de las existentes. Hemos presentado las reglas que rigen las conversiones entre diferentes representaciones a fin de compatibilizarlas de manera que permitan la operación entre los campos. Desde el punto de vista del diseño hemos decidido encapsular dichas reglas. Existirá entonces una clase cuyas instancias serán capaces de determinar la representación que tendrá el campo resultante a partir de las representaciones de los campos continuos de entrada.

La principal ventaja de tener por separado estas reglas, consiste en que permite variarlas a gusto sin que el resto de la arquitectura deba ser modificado por ello.

Vemos en la Figura 6.8 la relación de conocimiento entre el evaluador de operaciones y una instancia de la clase *SpatialConversionRule*.

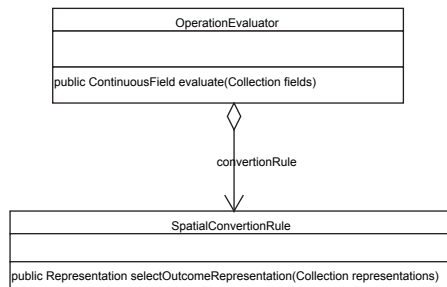


Figura 6.8: Reglas de conversión entre diferentes representaciones

En el caso del método de estimación, una vez definida la representación se asocia automáticamente el método de estimación por defecto. Pero existe la posibilidad que el usuario lo haga explícitamente, eligiendo alguno de los métodos de estimación soportados por la representación.

Integración de los elementos de la arquitectura

Ahora presentamos todos los componentes de la arquitectura.

Como puede observarse en la Figura 6.9, la pieza central, que permite personalizar la forma en la que se realizará la operación es el evaluador. Las instancias de evaluador son configuradas con una estrategia para definir el dominio, otra para definir las posiciones en la muestra y una operación puntual que sirve para calcular los valores del campo resultante. Aunque no es imprescindible, se puede determinar explícitamente la representación que tendrá el campo resultante, si esto no fuera así, se aplicarán automáticamente las reglas explicadas anteriormente para elegir la representación de salida.

6.2.2 Operaciones compuestas

Por lo general las operaciones sobre campos continuos no se aplican en forma aislada, en la mayoría de los casos, el resultado de aplicar una operación es utilizado como entrada para otra. Esto nos lleva a expresiones de la forma:

$$Op_1 (F_1, Op_2(F_2, F_3, Op_3(F_1, F_4)))$$

Donde Op_n es una operación y F_n un campo continuo.

Vemos entonces que las operaciones pueden anidarse una y otra vez de forma indefinida. Por lo tanto es imprescindible proveer una estructura que permita dicha composición sin límites. Además es importante que tanto una operación y como un campo sean tratados de la misma forma, en el sentido en que ambos pueden servir como entrada a una nueva operación.

Sin dudas esto se asemeja mucho al objetivo del patrón *Composite*, que propone: "Componer objetos en estructura tipo árbol para representar jerarquías parte-todo. *Composite* permite a los clientes tratar uniformemente a objetos y compuestos." Mayor información acerca del patrón *Composite* puede ser encontrada en el Apéndice B.4 (página 125).

En la Figura 6.10 presentamos la aplicación en nuestra arquitectura del patrón *Composite*.

Hemos decidido introducir a la clase *Requirement* que modela una operación sobre uno o varios campos, y define el mensaje *evaluate*, cuya semántica consiste en retornar el campo resultante de la aplicación de la operación. La subclase *SingleRequirement* simplemente retorna el campo continuo asociado en respuesta a ese mensaje, mientras que la subclase *ComplexRequirement* reenvía el mensaje a todos sus componentes, cuyos resultados sirven de entrada para la operación que debe aplicar el evaluador que tiene asociado.

6.3 Estructura dinámica

En esta sección presentaremos una instanciación de operaciones y campos continuos, mediante un ejemplo concreto. Para ello presentamos la siguiente situación:

Una empresa dedicada a obtener mediciones de contaminación ambiental realiza todos los meses recolección de muestras en una ciudad. Para ello utiliza equipos móviles que se sitúan en determinados puntos de la ciudad para obtener datos sobre los siguientes contaminantes del aire:

- Ozono (O^3)
- Partículas en suspensión (PS)
- Monóxido de Carbono (CO)
- Dióxido de Nitrógeno (NO_x)
- Dióxido de Azufre (SO_2)
- Plomo (Pb)

Para así con ellos poder calcular el Índice de Calidad del Aire. Debemos destacar que este muestreo se realiza una vez por mes en varios puntos de la ciudad, pero estos puntos varían de mes a mes.

En la aplicación utilizada por la empresa para registrar y almacenar los valores de contaminación del aire, utilizan un campo continuo con los valores correspondientes a un mes determinado. En la figura 6.11 vemos un diagrama mostrando las instancias de dos campos continuos con las mediciones realizadas en los meses de Enero y Diciembre de 2000.

Luego de un año de mediciones, se desea conocer si la contaminación del aire ha aumentado o disminuido. Marcando las zonas en donde el índice de contaminación ha disminuido y en aquellos en donde ha aumentado. Esto se realizara mediante la aplicación de la operación de sustracción entre los campos correspondiente a las mediciones de Diciembre/2000 y Enero/2000. Luego se aplicará una selección de aquellos valores por debajo de cero, es decir donde la contaminación ha disminuido; y una selección de aquellos valores por encima de cero, donde la contaminación ha aumentado.

Para ello primero se creará una instancia de *OperationEvaluator* cuya operación puntual (*punctualOperation*) será la sustracción (*Substraction*), la estrategia para la obtención de la muestra resultante (*sampleStratgy*) será la unión de las posiciones de las muestras de entrada (*Union-SampleStratgy*); esto es así debido a que las muestras son tomadas en distintas ubicaciones y si hiciéramos una intersección de las posiciones, tal vez tendríamos como resultado un campo sin ninguna muestra. Luego configuramos la estrategia para la obtención del dominio del campo resultante (*domainStrategy*), la cual será la unión de los dominios de los campos de entrada (*UnionDomainStrategy*). La configuración de la operación puede verse en la figura 6.12.

Una vez construida la operación y con los dos campos de entrada, podremos ahora configurar el requerimiento, el cual constará de dos *SingleRequirement* y un *ComplexRequirement*, esto puede verse en la figura 6.13.

A continuación mostraremos la interacción entre todos los objetos anteriormente construidos al realizar la operación enunciada anteriormente (Figura 6.14).

La secuencia comienza cuando el objeto *aComplexRequirement* recibe el mensaje *evaluate()*. Lo primero que realiza *aComplexRequirement* es evaluar a todos sus *subRequirements*. Los cuales retornan los respectivos campos *medDiciembre* y *medEnero*. Una vez que *aComplexRequirement* ha evaluado todos sus *subRequirements*, lo siguiente que realiza es evaluar la operación que tiene asociada (*op*), mediante el mensaje *evaluate()* y usando como parámetros los objetos *medDiciembre* y *medEnero*.

A partir de este punto es responsabilidad del objeto *op* construir el campo continuo resultado (*aResulingField*). Para ello lo primero que realiza es la construcción del dominio de *aResulingField*, esto lo realiza solicitando a los campos *medDiciembre* y *medEnero* sus respectivos dominios y luego enviando el mensaje *buildOuctomeDomain* al objeto que contiene la estrategia de construcción del dominio (*aUnionDomainStrategy*). Una vez obtenido el dominio para el campo resultante, lo que resta hacer en este paso es establecerlo al campo resultante mediante el mensaje *setDomain*.

Luego, el objeto *op* debe establecer cual será la representación que utilizará el *aResulingField*, aquí nuevamente *op* solicita a los campos *medDiciembre* y *medEnero* sus respectivas representaciones (*getRepresentation*) y con ellas invoca al mensaje *buildOutcomeRepresentation* del objeto *aSpatialConversionRule*, dicho mensaje retorna cual será la representación que utilizara *aResulingField*. A continuación se deben establecer que las posiciones de las muestras que formarán parte de la representación que se acaba de construir, para realizar esto el objeto *op* utiliza el conjunto de muestras de cada una de las representaciones de los campos *medDiciembre* y *medEnero*, para luego con estas invocar el mensaje *UnionSampleStrategy*, el cual retornara un conjunto de muestras con la unión de la ubicación de las muestras de cada uno de los conjuntos de muestras pasadas como parámetro. Una vez obtenidas las ubicaciones que formarán parte de la muestra de campo resultante se debe calcular el valor correspondiente a cada posición, para ello *op* itera sobre cada una de las muestras de la representación resultante y con cada una de ellas obtiene el valor de

dicha posición en cada uno de los campos *medDiciembre* y *medEnero* (*valueAt*). Una vez obtenidos estos valores debe realizar la operación puntual correspondiente (*Substraction*) mediante el mensaje *applyTo*. En este punto ya es posible agregar la muestra (que consta de ubicación y valor) al conjunto de muestras resultante.

Lo que falta realizar en este momento es establecer el conjunto de muestras en *aResulingField* mediante el mensaje *setSampleSet*.

En este punto ya tenemos construido el campo *aResulingField*, el cual es la diferencia entre los campos *medDiciembre* y *medEnero*. Sólo falta realizar una selección de aquellos valores que están por encima o por debajo de cero. Esto se realiza enviando en mensaje *select* al objeto *aResulingField*.

Hemos mostrado en esta sección como interactúan los objetos de la arquitectura, tanto para resolver operaciones n-arias, en este caso una sustracción. Como así también hemos mostrado como pueden incluirse operaciones unarias en una consulta.

6.4 Extensiones a la Arquitectura

Hemos mencionado al amplio espectro de aplicaciones GIS en las que es necesario manipular información continua. Debido a esa gran variedad de escenarios es imposible prever *todas* las operaciones que pudieran ser necesarias en todas esas aplicaciones. Sin embargo, es posible facilitar la introducción de nuevas operaciones incluyendo en la arquitectura puntos de *customization* donde sea posible variar el comportamiento de la arquitectura. Dichos puntos de variabilidad se conocen comúnmente como HotSpot [Pre94].

6.4.1 Hot Spots

La identificación de Hotspots en fases tempranas debería convertirse en una actividad explícita durante el proceso de desarrollo. Existe una razón fundamental para ello: los patrones de diseño presentan micro arquitecturas capaces de proveer flexibilidad en ciertos puntos de un framework, sin embargo es necesario utilizarlos en los sitios apropiados, las hotspots proveen dicha información ya que permiten combinar principios de construcción de software con la semántica del dominio para el cual el framework está siendo desarrollado [app92].

Una hot-spot card (ver Figura 6.15) es una forma de documentación que explica la semántica de un punto de variabilidad dentro de un framework y cuales son sus comportamientos alternativos. Una hot-spot card provee en primer lugar el nombre, un término conciso describiendo la funcionalidad que debería ser mantenida flexible, además especifica el grado deseado de flexibilidad. Es necesario tomar en cuenta que el grado de flexibilidad no es *gratis*, cuando se demande el mayor grado de flexibilidad, debe quedar claro se pagará un precio por ello, ya sea en performance, tiempo de desarrollo u otro factor. La siguiente sección debería expresar la funcionalidad abstrayéndose de los detalles. Finalmente, la última sección debe presentar dos situaciones concretas (por lo menos) en las que se aprecie claramente las diferencias de comportamiento.

6.4.2 Hot Spots en la arquitectura de operaciones para campos continuos

Las *hotspot cards* presentadas en las Figuras 6.16, 6.17 y 6.18 muestran los puntos de variabilidad de nuestra arquitectura. Si bien se ha propuesto que en todos los casos se provea el máximo nivel de flexibilidad, esto es permitiendo la adaptación por parte del usuario sin necesidad de reiniciar una aplicación basada en la arquitectura, sólo es posible cuando todas las variables de comportamiento esperables se encuentran preconstruidas en la arquitectura. En este tipo de casos el usuario, selecciona con ayuda de alguna herramienta el comportamiento que desea para una hotspot.

En el caso en que dicho comportamiento no se encuentre preconstruido el framework, este debe ser agregado mediante la extensión de la arquitectura por subclasificación y redefinición de métodos hook [Pre94]. En este caso la adaptación no puede ser realizada por el usuario final, sino por un desarrollador del framework.

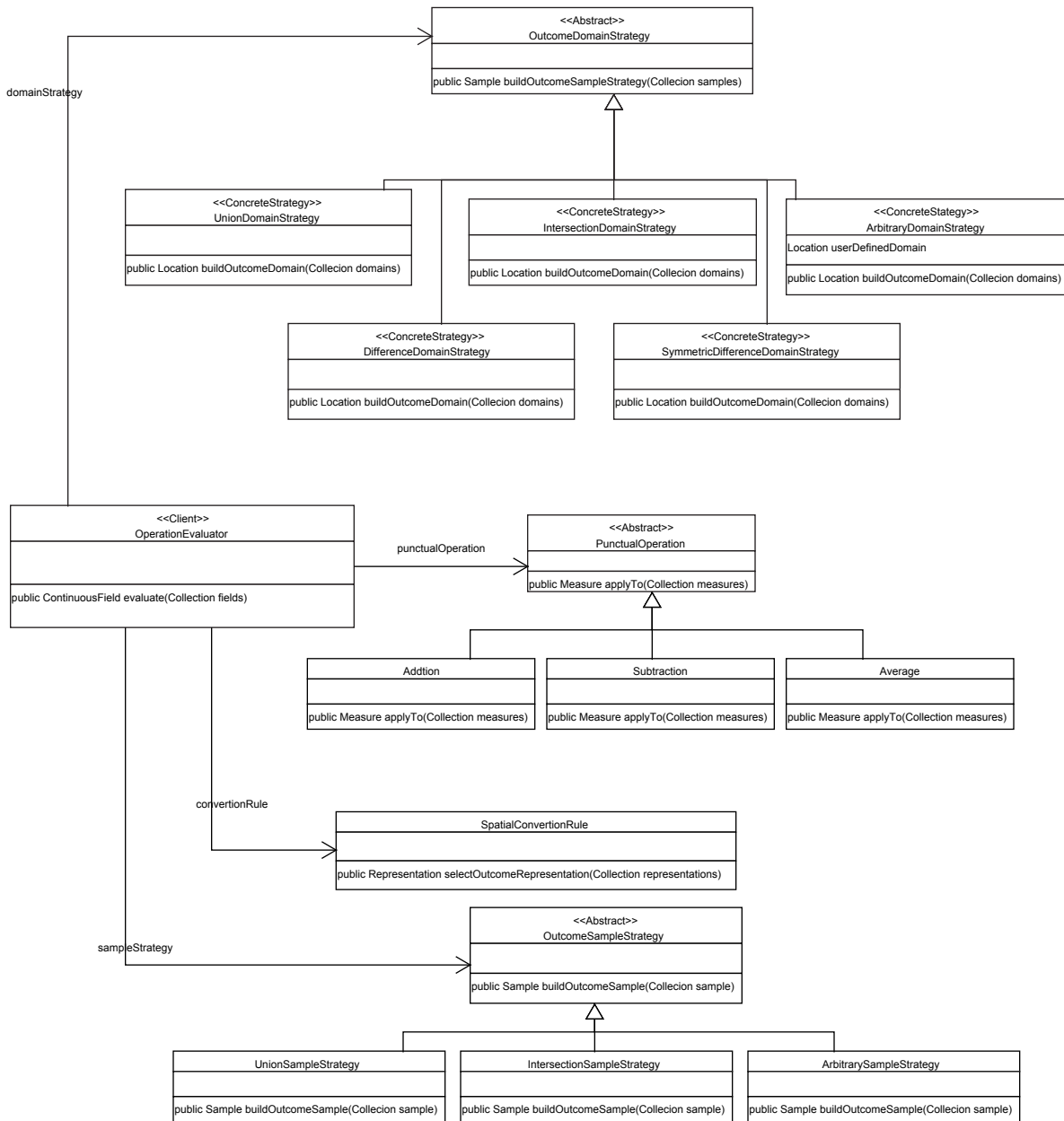


Figura 6.9: Diagrama de clases de la arquitectura completa

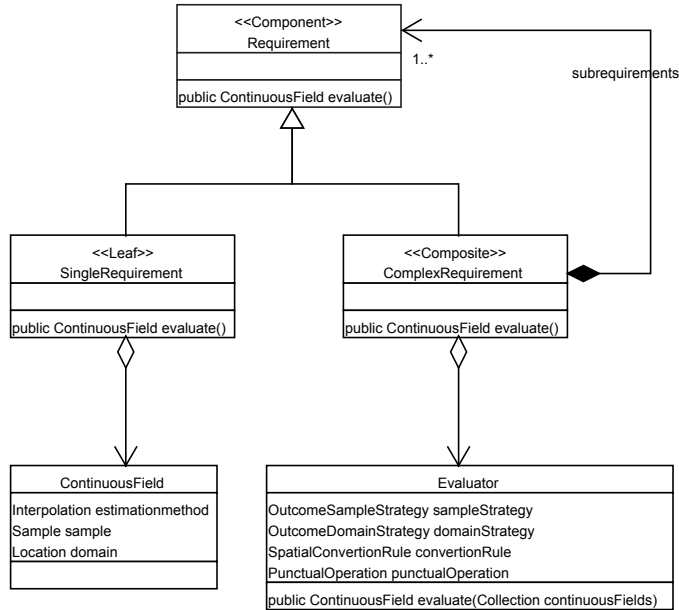


Figura 6.10: El patrón Composite aplicado a la jerarquía de Requirements

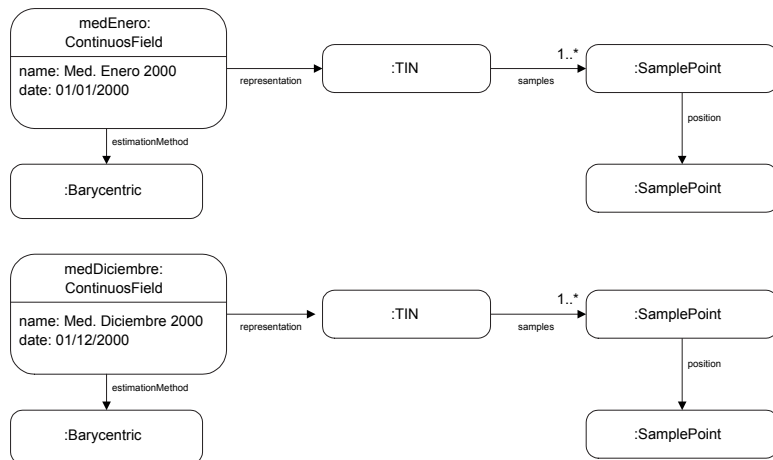


Figura 6.11: Diagrama de instancias de campos continuos.

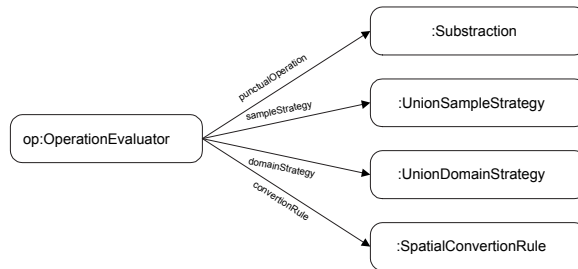


Figura 6.12: Diagrama de instancias de un *OperationEvaluator*.

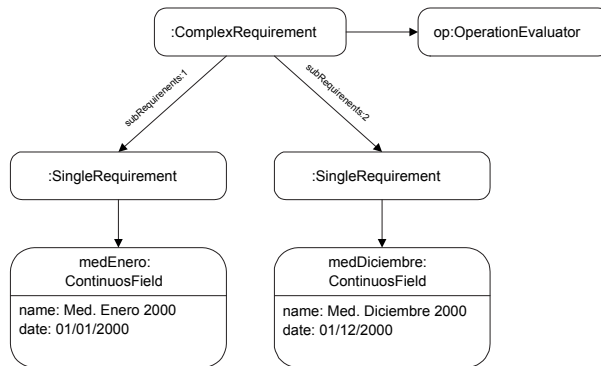
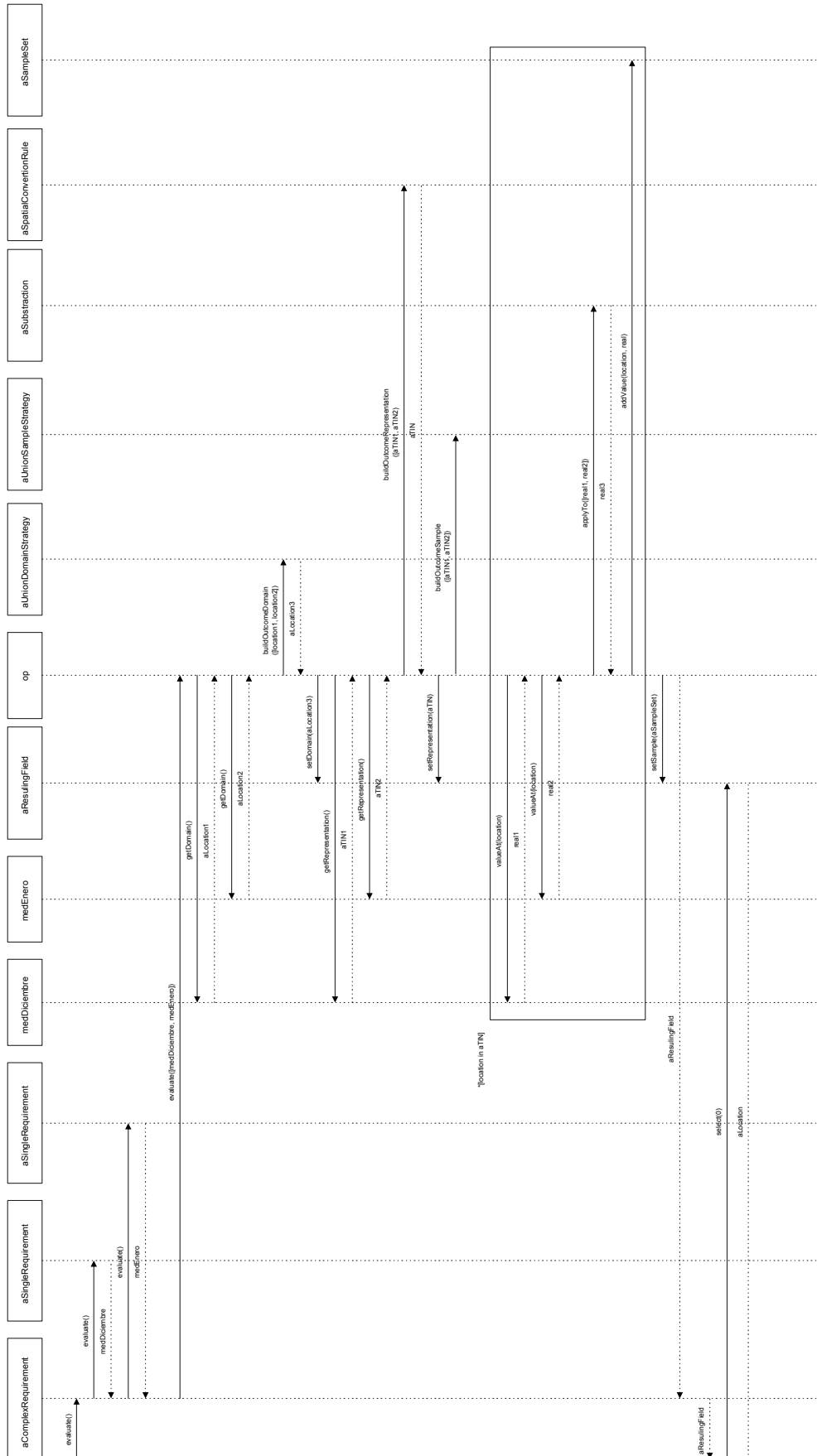


Figura 6.13: Diagrama de instancias de un *ComplexRequirement*.



<p>HotSpot Name</p> <p>Specify degree of flexibility</p> <p><input type="checkbox"/> adaptation without restart</p> <p><input type="checkbox"/> adaptation by end user</p>
<p>general description of semantics</p>
<p>sketch hot-spot behavior in at least two situations</p>

Figura 6.15: Una hot-spot card

<p>Sample Strategy</p> <p>Specify degree of flexibility</p> <p><input checked="" type="checkbox"/> adaptation without restart</p> <p><input checked="" type="checkbox"/> adaptation by end user</p>
<p>Esta hot spot encapsula variabilidad respecto a las diferentes formas en las que se pueden generar las posiciones geográficas que pertenecen a la muestra.</p>
<p>Dado un conjunto de muestras el resultante se puede generar mediante:</p> <ul style="list-style-type: none"> * Union * Intersección * Diferencia * Diferencia Simétrica * Definición por parte del usuario

Figura 6.16: Una hot-spot card para generar la muestra

Domain Strategy
Specify degree of flexibility <input checked="" type="checkbox"/> adaptation without restart <input checked="" type="checkbox"/> adaptation by end user
Esta hot spot encapsula variabilidad respecto a las diferentes formas en las que se pueden generar un dominio para un campo continuo, basándose en un conjunto de dominios de entrada.
Dado un conjunto de dominios el resultante se puede generar mediante: * Union * Intersección * Diferencia * Diferencia Simétrica * Definición por el usuario

Figura 6.17: Una hot-spot card para creación de dominios

Punctual Operation
Specify degree of flexibility <input checked="" type="checkbox"/> adaptation without restart <input checked="" type="checkbox"/> adaptation by end user
Esta hot spot encapsula variabilidad respecto a las diferentes operaciones que se pueden realizar sobre un conjunto de valores. Dichas operaciones producen como resultado un nuevo valor que puede ser asociado a un punto de una muestra.
Dado un conjunto de valores de entrada el resultante se puede generar mediante: * Adición y Diferencia * Producto y División * Promedio * Definición por el usuario

Figura 6.18: Una hot-spot card para operaciones puntuales

Capítulo 7

Conclusiones y Trabajo Futuro

7.1 Conclusiones

En este trabajo hemos presentado los aspectos relacionados a la manipulación de información continua en GIS. También hemos establecido un conjunto de operaciones para campos continuos y hemos descrito una arquitectura de objetos que permite implementar las operaciones mencionadas y que provee soporte para futuras operaciones. Se pueden mencionar algunos tópicos interesantes que se desprenden de este trabajo:

- *Interacción entre el modelo vectorial y el de campos:* Los GIS clásicos siguen uno de los modelos principales: modelo vectorial o modelo de campos [LT92]. Si bien ambos modelos han probado su efectividad en determinadas áreas, siempre ha existido un conflicto de interoperabilidad entre ambos. Por lo general los GIS permiten trabajar con uno solo de estos modelos a la vez, y esto se debe a que conceptualmente el software que los manipula ha sido construido específicamente para trabajar en un framework en particular, y los aspectos relacionados a la interoperabilidad se resuelve en etapas posteriores, tomando la forma de importadores/exportadores de datos desde y hacia el otro modelo de datos.

La arquitectura presentada en este trabajo se ha integrado a un framework para el desarrollo de aplicaciones GIS, donde ambos modelos han sido incluidos. En dicho modelo cualquier objeto geográfico posee características vectoriales (su topología por ejemplo) y puede tener atributos que sean campos continuos. De hecho un campo continuo también posee características vectoriales (su dominio). Por lo tanto ambos modelos han sido efectivamente integrados y conviven, de forma tal que es posible beneficiarse con las ventajas de cada uno.

- *Extensibilidad del modelo:* Como hemos expresado, seguramente existen otras operaciones sobre campos continuos que no han sido presentadas en este trabajo. De hecho, dado el amplio espectro de aplicaciones geográficas (ambientales, catastrales, control de cultivos, control de migraciones de animales, etc.) es imposible cubrir la totalidad de operaciones que podrían necesitarse. Sin embargo, gracias a la flexibilidad conseguida en la arquitectura de objetos que brinda soporte a las operaciones, es posible definir fácilmente nuevas operaciones.

Dichas extensiones a la arquitectura se realizan de la misma manera en que se extiende un framework white-box, es decir, mediante subclasificación y redefinición de hook-methods.

Los sectores de la arquitectura que pueden ser extendidos han sido documentados usando las hotspot-cards.

La flexibilidad del modelo se debe sin duda a la aplicación de los patrones de diseño [GRJV95], que brinda soluciones de probada eficacia a problemas de diseño recurrentes. Es importante destacar que la aplicación de los patrones de diseño debe realizarse de forma cuidadosa, ya que su uso, si bien produce arquitecturas muy flexibles, tan bien es cierto que dichas arquitecturas suelen ser sumamente complejas y difíciles de mantener. Por eso es importante lograr un equilibrio entre la flexibilidad que se desea y la complejidad que ésta introduce.

- *Aspectos implementativos:*

Esta arquitectura ha sido completamente implementada en Java, la aplicación resultante [ZPB⁺00], ha sido presentada en OOPSLA2000. La implementación no presentó ninguna dificultad ya que el diseño original de la arquitectura siempre fue representado en forma abstracta mediante documentación UML, por lo tanto nunca fue pensada para estar atada a un lenguaje OO en particular. Prototipos de esta arquitectura también fueron implementados anteriormente en Smalltalk. De ambas implementaciones podemos concluir que:

1. La construcción de prototipos funcionales en Smalltalk es más sencilla y veloz
2. Java permite expresar en forma concreta conceptos muy importantes para la teoría de OO, como ser: *interfaces, clases y métodos abstractos*. Este poder expresivo nos lleva a obtener un software de mayor calidad, aunque con un tiempo de desarrollo mayor.
3. La implementación en Java es considerablemente más veloz para las tareas relacionadas con el cálculo, por ejemplo: construcción de triangulaciones para los TIN.

7.2 Trabajo Futuro

Respecto la manipulación de información continua existe un tópico por investigar cuidadosamente, que es de vital importancia en un buen porcentaje de aplicaciones GIS. Este tema se refiere a las características temporales de los objetos. Hasta este momento hemos trabajado con objetos que presentan características puramente geográficas (o espaciales), pero en muchos dominios es necesario manipular u operar con objetos que poseen características espacio-temporales.

Podemos citar como ejemplo a las aplicaciones ambientales orientadas al seguimiento de fenómenos naturales. Dichas aplicaciones se basan en mediciones pasadas y presentes para predecir el comportamiento de fenómenos como los tornados, huracanes, incendios forestales, entre otros.

Por lo general la información referente a un fenómeno en particular puede ser representada por campos continuos, dichos campos continuos son utilizados (mediante el conjunto de operaciones presentadas) para realizar las predicciones necesarias. Pero los mecanismos de predicción necesitan a menudo, información histórica referente a la evolución del fenómeno. Esta información que por lo general se guarda en forma de tuplas (Objeto, Timestamp), debe ser consultada para satisfacer las necesidades de los mecanismos de predicción.

Si bien parece un tema menor asociar un *Timestamp* a un objeto geográfico, es necesario mencionar una serie de aspectos que tienen que ver con el modelado del tiempo. Tales aspectos, como la continuidad, determinación, orden temporal, historias, etc., hacen necesario proveer un framework

para la manipulación de objetos temporales. Además de esto, la inclusión de características temporales a los objetos geográficos, y en particular a los campos continuos, nos pone frente a nuevos problemas, por ejemplo: ¿Qué ocurre cuando es necesario obtener un campo continuo asociado a un momento en el cual no se han hecho mediciones? ¿Es posible obtenerlo interpolando temporalmente a partir de otros momentos medidos? ¿Qué tan confiables (en términos de introducción de errores) son esos métodos de interpolación temporal? Estas y muchas otras preguntas deben ser respondidas antes de incluir características temporales a un modelo con campos continuos.

Bibliografia

- [AHA92] F. Aurenhammer, F. Hoffmann, and F. Aronov. Least-squares partitioning. In *Proc. 8th European Workshop on Computational Geometry CG '92*, pages 55–57, Utrecht, the Netherlands, 1992.
- [Ans89] L. Anselin. What is special about spatial data? alternative perspectives on spatial data analysis. In University of California Santa Barbara, editor, *NCGIA Technical Paper 89-4, National Center for Geographic Information and Analysis*, Santa Barbara, CA, 1989.
- [app92] *Building App Frameworks*. 1992.
- [BA92] M. Breunig and Perkhoff A. Data and system integration for geoscientific data. In *Proceedings of the 5th International Symposium on Spatial Data Handling Charleston, SC, International Geographical Union*, pages 272–281, Columbia, SC, 1992. Commission on GIS, Humanities and Social Sciences Computing Lab, University of South Florida.
- [Boo86] B.N. Boots. *Voronoi (Thiessen) polygons, Concepts and techniques in modern geography series no. 45*. Geo Books, Norwich, UK, 1986.
- [Bur86] P.A. Burrough. *Principles of Geographical Information Systems for Land Resources Assessment*. Oxford University Press, New York, 1986.
- [Bur91] P.A. Burrough. Soil information systems. In M.F. Goodchild D.J. Maguire and D.W. Rhind, editors, *Geographical Information Systems: Principles and Applications*, pages 153–169, London, 1991. Longman.
- [Bur92] P.A. Burrough. *Are GIS data structures too simple minded?* *Computers & Geosciences* 18(4), 1992.
- [CA85] M. Cross and Moscardini A.O. *Learning the Art of Mathematical Modeling*. Halsted Press, New York, 1985.
- [Cas89] J.L. Casti. *Alternate realities: Mathematical models of nature and man*. John Wiley & Sons, New York, 1989.
- [Dac67] M.F. Dacey. *Description of line segments*. *Northwestern Studies in Geography* 13, 1967.
- [Dac70] M.F. Dacey. *The syntax of a triangle and some other figures*. *Pattern Recognition* 2, 1970.

- [Dac71] M.F. Dacey. *Poly: a two dimensional language for a class of polygons*. Pattern Recognition 3, 1971.
- [ESR91] ESRI. *Cell-based Modeling with GRID*. Environmental Systems Research Institute Inc., Redlands, CA., 1991.
- [FG92] R. Flowerdew and M. Green. *Developments in areal interpolation methods and GIS*. Annals of Regional Science 26, 1992.
- [GB98] S. Gordillo and F. Balaguer. Refining an object-oriented GIS design model: Topologies and field data. In Robert Laurini, Kia Makki, and Niki Pissinou, editors, *Proceedings of the 6th International Symposium on Advances in Geographic Information Systems (GIS-98)*, pages 76–81, New York, November 6–7 1998. ACM Press.
- [GL80] M.F. Goodchild and N.S-N. Lam. *Areal interpolation: A variant of the traditional spatial problem*. Geo-Processing 1, 1980.
- [Goo92] M.F. Goodchild. *Geographical data modeling, Computers and Geosciences*. 1992.
- [GPS93] M.F. Goodchild, B.O. Parks, and L.T. Steyaert. *Environmental Modeling with GIS*. Oxford University Press, New York, 1993.
- [GRJV95] E. Gamma, Helm R., R. Johnson, and J. Vlissides. *Design Patterns, Elements of Reusable Object-Oriented Software*. Addison Wesley, 1995.
- [Hai90] R. Haining. *Spatial Data Analysis in the Social and Environmental Sciences*. Cambridge University Press, Cambridge, 1990.
- [Har69] D. Harvey. *Explanation in Geography*. Edward Arnold, London, 1969.
- [J.C86] Davis J.C. *Statistics and Data Analysis in Geology, 2nd edition*. John Wiley and Sons, 1986.
- [Jou86] A.G. Journel. *Geostatistics: Models and tools for the earth sciences*. Mathematical Geology 18(1), 1986.
- [Kem97a] K. Kemp. *Fields as a Framework for integrating GIS and environmental process models: Representing spatial continuity*. Transactions in GIS, 1997, Vol. 1, N^o 3, 1997.
- [Kem97b] K. Kemp. *Fields as a Framework for integrating GIS and environmental process models: Specifying field variables*. Transactions in GIS, 1997, Vol. 1, N^o 3, 1997.
- [Kum92] M.P. Kumler. *An Intensive Comparison of TINs and DEMs*. Department of Geography, University of California, Santa Barbara, CA, 1992.
- [Lam83] N.S-N. Lam. *Spatial interpolation methods: a review*. The American Cartographer 10(2), 1983.
- [LT92] R. Laurini and D. Thompson. *Fundamentals of Spatial Information Systems*. Number 37 in The A.P.I.C. Series. Academic Press, New York, 1992.

- [McC88] McCullagh. *Terrain and surface modelling systems: theory and practice*. Photogrammetric Record 12(72), 1988.
- [Mol91] M. Molenaar. *Status and problems of geographical information systems: the necessity of a geoinformation theory*. ISPRS Journal of Photogrammetry and Remote Sensing 46, 1991.
- [OG89] M.A. R. Webster Oliver and J. Gerrard. *Geostatistics in physical geography. Part I: theory*. Transactions of the Institute of British Geographers 14, 1989.
- [OW91] M.A. Oliver and R. Webster. *How geostatistics can help you*. Soil Use and Management 7(4), 1991.
- [Pre94] W. Pree. Meta patterns - A means for capturing the essentials of reusable object-oriented design. In M. Tokoro and R. Pareschi, editors, *Proceedings ECOOP'94*, pages 150–162, Bologna, Italy, 1994. Springer-Verlag.
- [PW86] G.M. Philip and D.F. Watson. *Matheronian geostatistics - quo vadis?* Mathematical Geology 18(1), 1986.
- [RM91] Jonathan F. Raper and David J. Maguire. *GIS design models*. Computers-and-Geosciences. 18, Birkbeck Coll., London, 1991.
- [SB88] J. Serra and Lay B. *Algorithms in Mathematical Morphology*. Academic Press, London, 1988.
- [Sch76] C.H. Schut. *Review of interpolatin methods for digital terrain models*. The Canadian Surveyor 30(5), 1976.
- [Ser82] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, London, 1982.
- [Ser88] J. Serra. *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*. Academic Press, London, 1988.
- [Sil79] J. Silk. *Statistical Concepts in Geography*. George Allen and Unwin, London, 1979.
- [SL69] S. Selby and Sweet L. *Sets-Relations-Functions*. McGraw Hill, 1969.
- [Smi92] T.R. Smith. Towards a logic-based language for modeling and database support in spatio-temporal domains. In *Proceedings of the 5th International Symposium on Spatial Data Handling, Charleston, SC, International Geographical Union*, pages 592–601, Columbia, SC, 1992. Commission on GIS, Humanities and Social Sciences Computing Lab, University of South Florida.
- [Sny87] J.P. Snyder. *Map projections: A working manual, U.S. Geological Survey Professional Paper No. 1395*. General Printing Office, Washington, DC, 1987.
- [Ste46] S.S. Stevens. *On the theory of scales of measurement*. Science 103(2684), 1946.

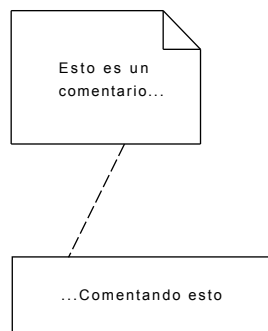
- [Sum83] M. Summerfield. *Populations, samples and statistical inference in geography*. Professional Geographer 35, 1983.
- [Tob79a] W.R. Tobler. *Smooth pycnophylactic interpolation for geographic regions*. Journal of the American Statistical Association 74(367), 1979.
- [Tob79b] W.R. Tobler. *Transformational view of geography*. American Cartographer 6(2), 1979.
- [Tob88] W.R. Tobler. Resolution, resampling, and all that. In Mounsey and R. Tomlinson, editors, *Building Databases for Global Science*, pages 129–137. Taylor and Francis, 1988.
- [Tob90] W.R. Tobler. *GIS transformations*. 1990.
- [Tom90] C.D. Tomlin. *Cartographic Information Systems and Cartographic Modeling*. Prentice-Hall, Englewood Cliffs, New Jersey, 1990.
- [Tom91] C.D. Tomlin. Cartographic modelling. In M.F. Goodchild D.J. Maguire and D.W. Rhind, editors, *Geographical Information Systems: Principles and Applications*, pages 361–374. Longman, London, 1991.
- [WH91] R. Weibel and M. Heller. Digital terrain modelling. In M.F. Goodchild D.J. Maguire and D.W. Rhind, editors, *Geographical Information Systems: Principles and Applications*, pages 269–297, London, 1991. Longman.
- [WO92] R. Webster and M.A. Oliver. *Sample adequately to estimate variograms of soil properties*. Journal of Soil Science 43(1), 1992.
- [Yoe84] P. Yoeli. *Cartographic contouring with computer and plotter*. American Cartographer 11, 1984.
- [Zei76] Bernard P. Zeigler. *Theory of Modelling and Simulation*. John Wiley & Sons, New York, 1976.
- [ZPB⁺00] A. Zambrano, L. Polasek, J. Bazzocco, G. Trilla, D. Cano, and A. Lliteras. Tilcara: an oo perspective to handle continuous fields in gis. In *Addendum to the 2000 proceedings of the conference on Object-oriented programming, systems, languages, and applications*, pages 185–186, New York, NY, USA, 2000. ACM Press.

Apéndice A

Referencia Diagramas UML

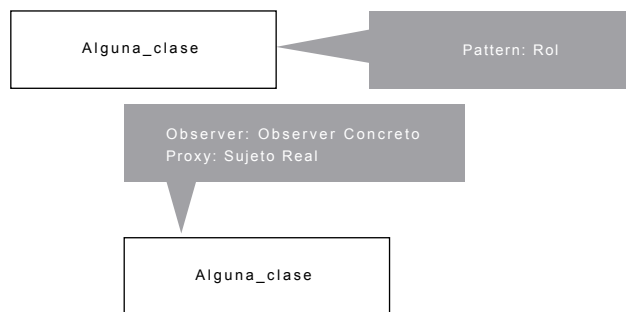
A.1 Diagramas Estáticos del Modelo

A.1.1 Comentarios.



Cualquier tipo de información que no se representa fácilmente en UML, incluyendo comentarios, código a nivel de implementación, etc. También se usa para restricciones largas.

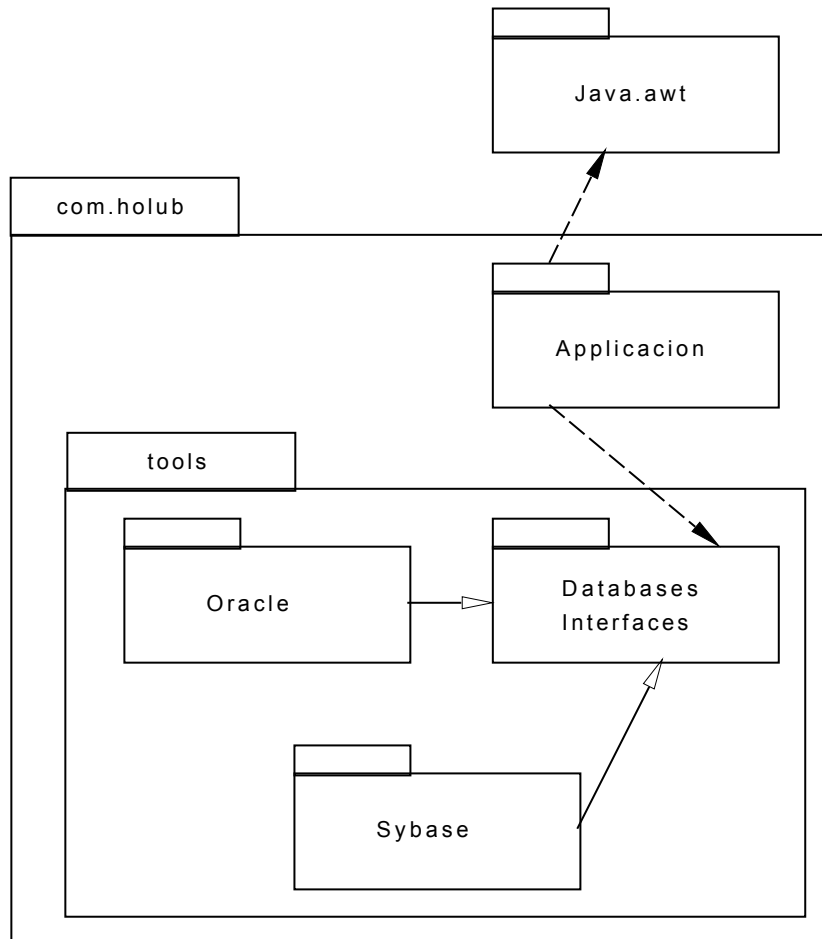
A.1.2 Design Patterns



- Notación para los patrones de diseño de Erich Gamma, como es presentada en el libro *Pattern Hatching* de John Vlissides (Addison Wesley, 1998).

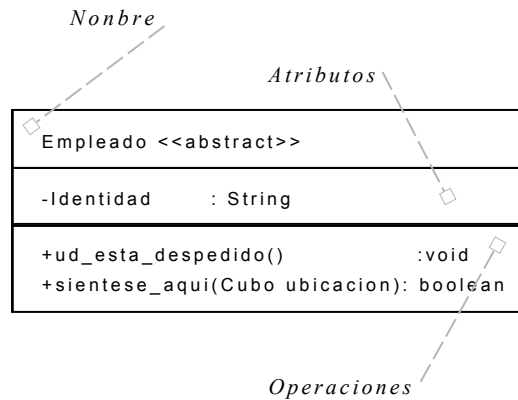
- El nombre del Pattern va a la izquierda, el rol de la clase dentro del pattern va a la derecha. Una clase puede tener diferentes roles con respecto a varios patterns. En el ejemplo de abajo, la clase sirve tanto como "Observer Concreto" en el pattern "Observer" y también como "Objeto Real" en el pattern "Proxy".

A.1.3 Paquetes



- **Espacio de nombre C++.**
- Agrupa clases con funcionalidad similar.
- Las clases derivadas no necesitan estar en el mismo paquete.
- Los paquetes se pueden anidar. Los paquetes externos algunas veces se llaman **dominios**. (“Tools”, a la izquierda, es un paquete externo, no un dominio).
- El nombre del paquete es parte del nombre de la clase. (e.j. dada la clase *pedro* en el paquete *picapiedras*, el **nombre completo de clase** es *picapiedras.pedro*).
- Generalmente son necesarios cuando el diagrama estático completo no cabe en la hoja.

A.1.4 Clases



Contiene tres *compartimientos*:

1. El *compartimiento del nombre* (requerido) contiene el nombre de la clase y otra información relacionada con la documentación: E.j.:

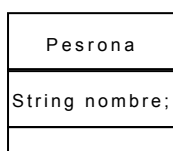
```

Alguna_clase <<abstract>>
{
    autor:      Juan Perez
    modificada: 10/6/2999
    verificada: si
}
  
```

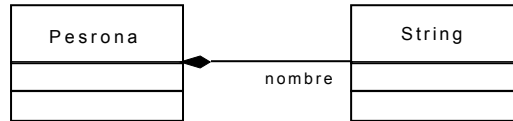
- Las comillas identifican a los *estereotipos*. E.j.: `<<utility>>`, `<<abstract>>` `<<JavaBean>>`. Se pueden usar gráficos en vez de palabras.
- Los privilegios de acceso (ver abajo) pueden preceder al nombre.
- Las clases internas (anidadas) identifican a las clases externas como un prefijo de su nombre de clase: (**Externa.Interna** o **Externa::Interna**)

2. El *compartimiento de los atributos* (opcional):

- *Durante el Análisis*: identifica los atributos (es decir las características que lo definen) del objeto.
- *Durante el Diseño*: identifica la relación con una clase de la librería de clases del lenguaje:
Esta:



es una versión mas compacta (y menos informativa) de:



Aquí, todo, es privado. Siempre.

3. El *compartimiento de las operaciones* (opcional) contiene definiciones de métodos:

nombre_de_mensaje(argumentos): tipo_retornado

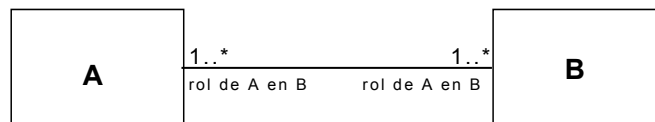
o utiliza la sintaxis de algún lenguaje de programación, excepto para los siguientes **privilegios de acceso**:

+	public
#	protected
?	private

- Las operaciones abstractas (C++ virtual, Java no-final) se indican en *itálica* (o subrayadas).
- Los nombres de las operaciones son mas fáciles de leer si están en **negrita**.

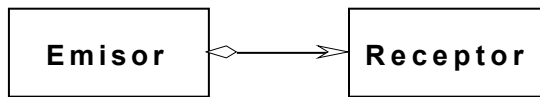
Si se omiten los atributos y operaciones, se asume que existe una definición más completa en otra hoja.

A.1.5 Asociaciones



Relaciones entre clases.

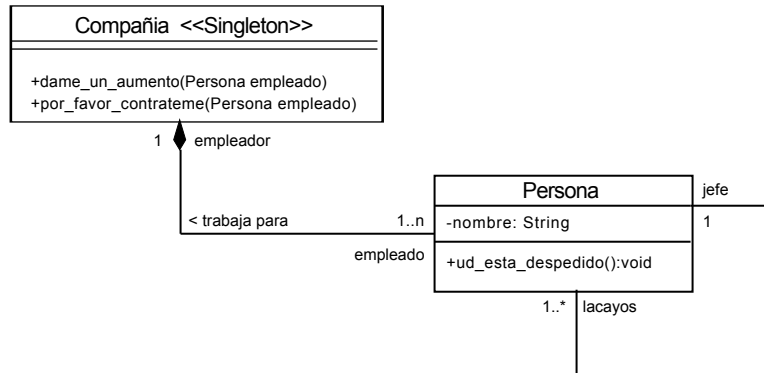
- Las clases asociadas se conectan con líneas.
- La **relación** se identifica, si es necesario con un < > para indicar la dirección (o usando flechas sólidas).
- El rol que juega la clase en la relación se indica al lado de la clase bajo la línea.
- Los estereotipos son apropiados (como <<friend>>).
- El flujo unidireccional de mensajes puede ser indicado por una flecha (pero es implícito en situaciones donde hay un sólo rol):



- Cardinalidad:

1 (usualmente omitida si la relación es 1:1)
 n (desconocida en tiempo de compilación, pero acotada)
 0..1 (1..21..n)
 1..* (1 o más)
 * (0 o más)

- Ejemplo:



```

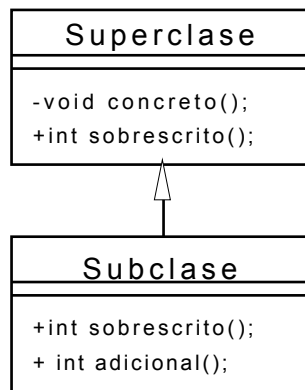
class Compania
{
    private Persona[] empleado = new Persona[n];
    public void dame_un_aumento( Persona empleado )
    { /*...*/
    }
    public void por_favor_contratame( Persona prospecto )
    { /*...*/
    }
}

class Persona
{
    private String nombre;
    private Compania empleador;
    private Persona jefe;
    private Vector subordinados = new Vector();
    public void ud_esta_despedido(Persona empleado){/*...*/}
}
  
```

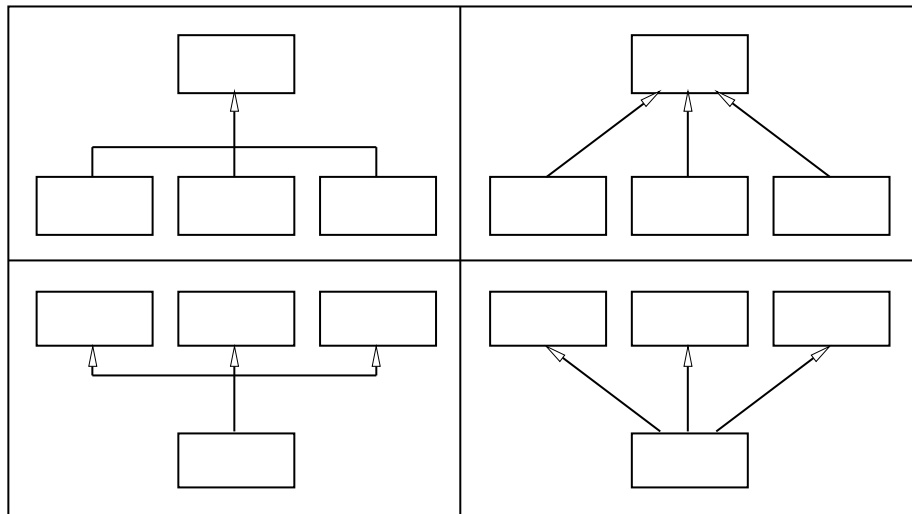
}

(Un Vector Java es un array de longitud variable. En este caso contendrá objetos de la clase Persona.)

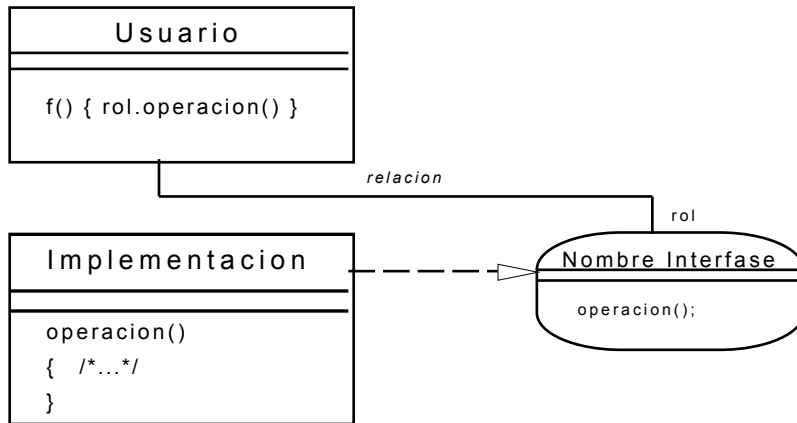
A.1.6 Herencia de Implementación (Generalización/Especialización)



←— identifica derivación. La clase derivada es la clase base, pero con propiedades adicionales (o modificadas). La (sub) clase derivada es una especialización de la (super) clase base. Las variantes incluyen:



A.1.7 Herencia de Interfase (Especialización/Refinamiento)

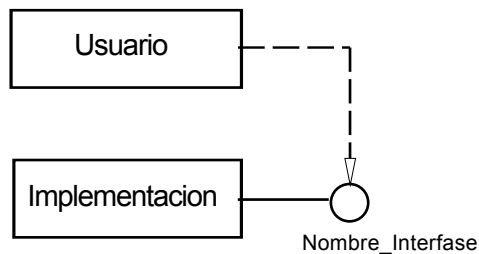


Es un contrato que especifica un conjunto de métodos que deben ser implementados por la clase derivada. En C++, una interfase es una clase conteniendo nada mas que métodos puramente virtuales. Java soporta la definición de interfaces en forma directa (“clase abstracta”, que puede contener definiciones de métodos y atributos además de las declaraciones abstractas.)

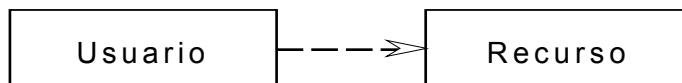
Las interfaces no contienen atributos, por lo tanto el compartimiento de los “atributos” está siempre vacío.

Cuando la clase base es una interfase, la línea de relación de “herencia” es de puntos.

UML también permite utilizar el estilo de notación “pin” de Microsoft, pero este estilo es inconsistente con el resto de UML y su uso no es recomendado.

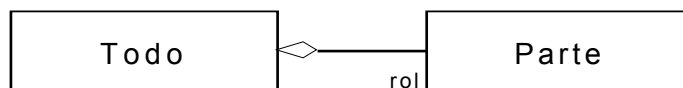


A.1.8 Dependencias



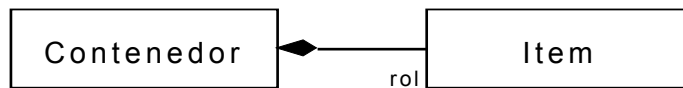
El *Usuario* usa *Recurso*, pero *Recurso* no es un miembro de la clase *Usuario*. Si *Recurso* es modificado, algunos métodos de la clase *Usuario* deben ser invocados. *Recurso* es típicamente una variable local o el argumento de algún método de la clase *Usuario*.

A.1.9 Relación de Agregación



Indica que la destrucción del “todo” no implica la destrucción de las partes.

A.1.10 Relación de Composición

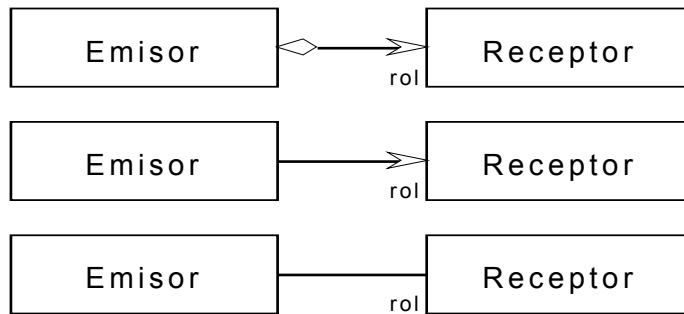


Las partes son destruidas conjuntamente con el “todo”. Realmente no existe en Java. En C++:

```

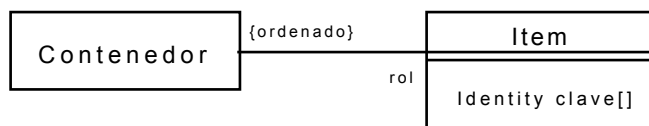
class Contenedor
{
    Item item1; // ambas variables de instancia son
    Item *item2; // parte de una relacion de ‘composicion’
public:
    Contenedor() { item2 = new Item; }
    ~Contenedor(){ delete item2; }
}
  
```

A.1.11 Navegabilidad



Los mensajes (solamente) fluyen en la dirección de la flecha. Es implícito cuando el rol no está presente: si un objeto no tiene asignado un rol en alguna relación, entonces no hay forma de enviarle un mensaje.

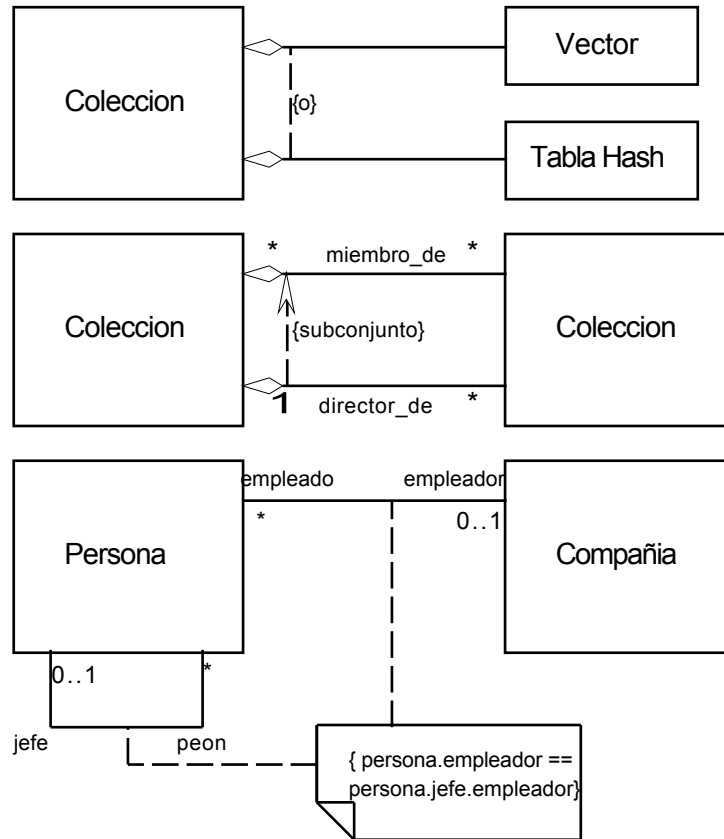
A.1.12 Restricción



Una relación restringida requiere que se aplique alguna regla. (e.j. [ordenado])

Por lo general se combina con la agregación, composición, etc.

A.1.13 Restricciones Complejas



Comentarios

- En el caso del **or**, sólo existirá una de las relaciones indexadas en algún momento dado (una **union C++**).
- En UML oficial, coloca una restricción arbitraria que afecta a mas de una relación en una caja de **“comentarios”**.

A.1.14 Asociación Calificada

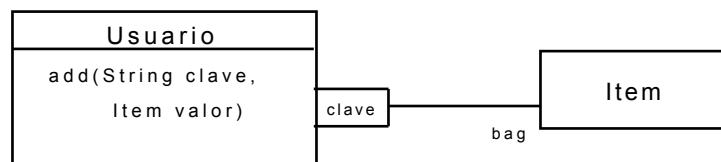


Tabla de hash, array asociativo, “diccionario”.

```

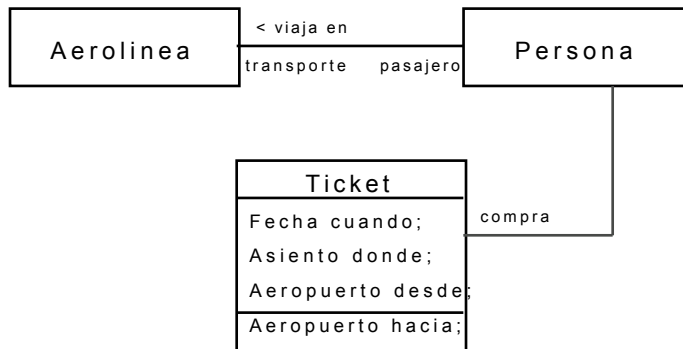
class Contenedor
class Usuario
{ // Una Hashtable es un array asociativo, indexado
  // por alguna clave y conteniendo alg\'un valor.

  private Hashtable tablaHash = new Hashtable();

  private void agregar(String clave, Item valor)
  { tablaHash.put( clave, valor );
  }
}

```

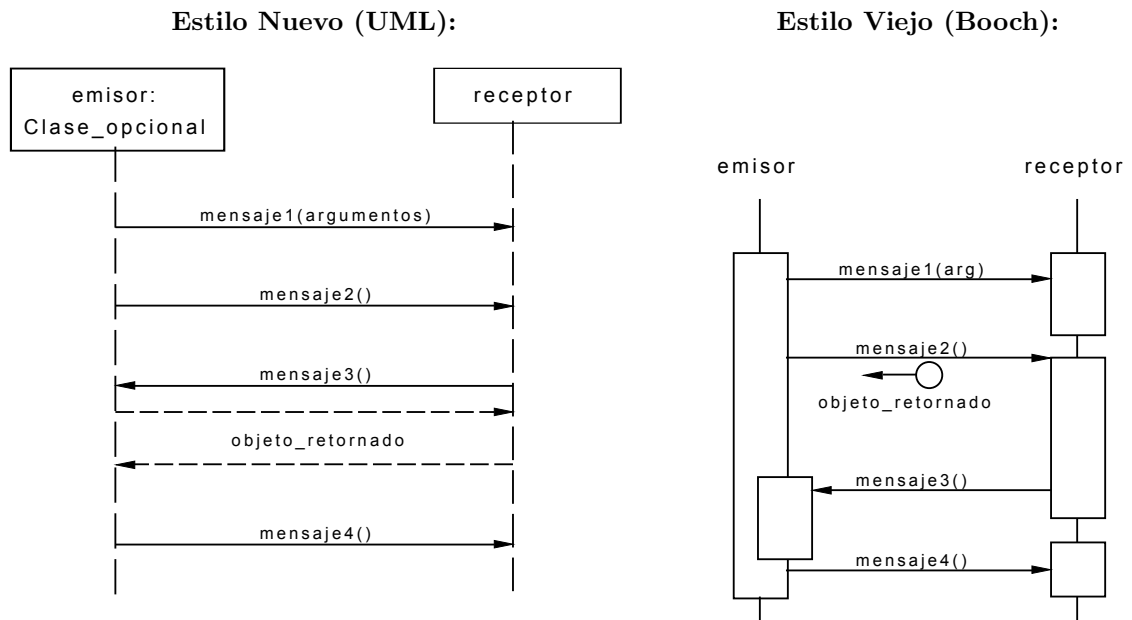
A.1.15 Asociación de Clases



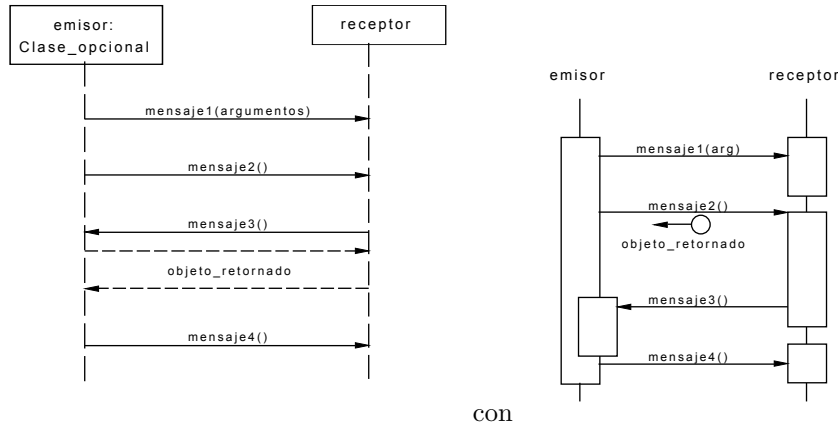
- Se utiliza cuando se requiere de una clase para definir una relación.
- Algunas veces, se requiere de una relación adicional para mostrar pertenencia. (El uno entre Persona y Ticket en el ejemplo.)

A.2 Diagramas Dinámicos del Modelo

A.2.1 Objetos y Mensajes



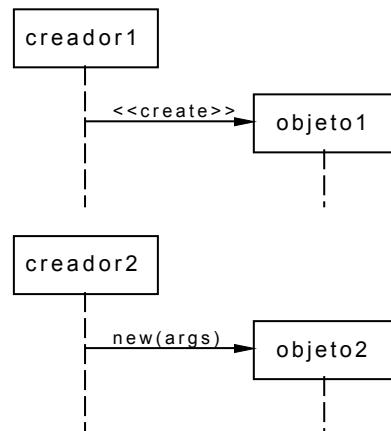
- Las líneas verticales representan objetos, no clases. Se puede agregar opcionalmente “:class” a los cuadros para hacer el diagrama mas explícito.
- \longrightarrow representa mensajes sincrónicos. (el control no retorna hasta que el mensaje invocado no haya terminado).
- \longleftarrow - representa retorno. (Las flechas se etiquetan con el nombre/tipo del objeto retornado.)
- El objeto emisor debe tener:
 1. Algún tipo de asociación con la clase del objeto receptor.
 2. El “rol” de la clase del lado receptor debe tener el mismo nombre que el del objeto receptor.
- El estilo de Booch es idéntico a UML en cuanto a funcionalidad, pero mucho mas fácil de leer. Compare:



(El “retorno” en el estilo de Booch es implícito en la parte de abajo del cuadro, no hay necesidad de \leftarrow para métodos que no retornan ningún valor. $\leftarrow\bigcirc$ identifica el objeto retornado.)

- Las flechas de retorno son esenciales en el estilo UML. de otra manera el control de flujo sería ambiguo. En diagrama mostrado arriba, no queda claro cuando se omiten las etiquetas de retorno si el objeto_retornado es la respuesta del mensaje2 o del mensaje4.

A.2.2 Creación de Objetos



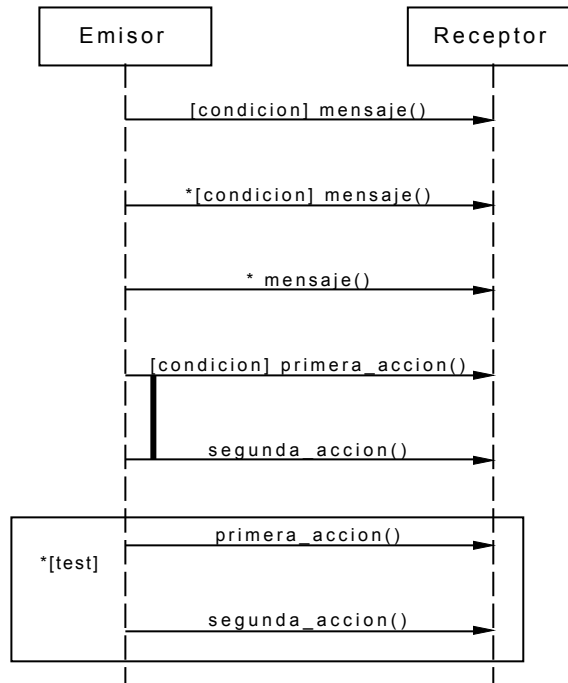
- Las cajas con los nombres aparecen en el punto de creación.
- Se utiliza la forma $\ll\text{creates}\gg$ para una creación automática, e.j., en c++:

```
class Creador1
{
  Objeto1 el_objeto; // el objeto actual, no una referencia.
}
```

(No existe una operación equivalente en Java)

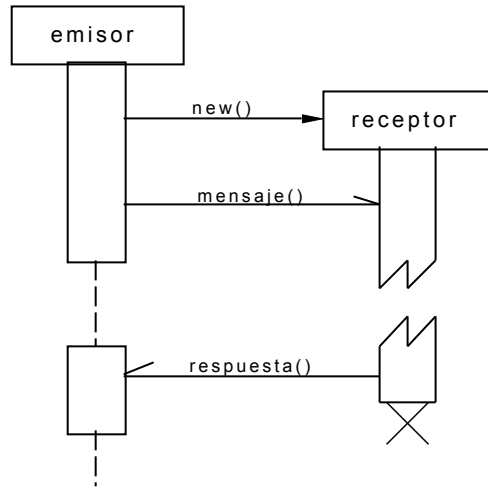
- Si se muestra un mensaje en vez de «creates», entonces el objeto se creará durante la ejecución del mensaje.

A.2.3 Condiciones, Ciclos



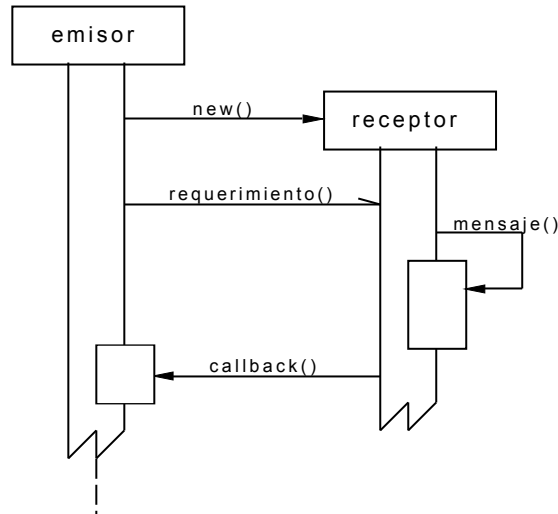
- El mensaje es enviado sólo si la condición entre corchetes es verdadera,
- Un asterisco al lado de la condición indica iteración.
- Un asterisco sin un nombre de condición significa “cada,” cuando el receptor es una agregación.

A.2.4 Mensajes Asíncronos



- Una media flecha significa “asíncrono”. (ver la tabla, abajo).
- El ensanchamiento de la línea significa “activado” (el thread esta corriendo). (En el diagrama, el thread emisor envía un mensaje() y luego se suspende a sí mismo esperando por la respuesta().)
- Un corte en la caja (“receptor” en el diagrama) significa otra actividad ocurriendo en el thread que no es relevante a escenario corriente. (E.j. El emisor puede hacer un wait() después de enviar el mensaje. el mensaje reply() ejecuta notify().)
- La X significa que se elimina el objeto (en este caso, se autoelimina). Un kill externo se representa como: $\xrightarrow{\llcorner\text{destroy}\gg} \times$

A.2.5 Callbacks, Recursión



- El mensaje callback() es ejecutado en el thread del objeto “receptor”.
- El callback() es, sin embargo un miembro de la clase del objeto emisor, por lo tanto tiene acceso a todos los campos del objeto.
- Dado que el thread del “emisor” original también esta ejecutándose, cualquier acceso a los campos por medio de callbacks debe realizarse en forma “sincronizada”.

A.2.6 Tipos de flechas para los distintos mensajes

Símbolo	Tipo de Mensaje	Descripción
	Simple	No interesa como es manejado. Usualmente es asincrónico.
	Sincrónico	El emisor se bloquea hasta que el método finaliza. Esto sería una llamada a función en una aplicación con un sólo thread.
	Asincrónico	El control retorna inmediatamente, pero el trabajo se realiza en segundo plano y el emisor puede realizar otras tareas mientras se realiza el procesamiento.
	Rechazado	El objeto receptor se puede rehusar a aceptar el pedido de ejecución del mensaje. Esto puede pasar ,por ejemplo, si se llena la cola de mensajes “activos”.
	Timeout	Usualmente se bloquea el control de un mensaje, pero puede retornar después de una determinada cantidad de tiempo, aún si no se ha completado el pedido.

Apéndice B

Patrones de Diseño

B.1 ¿Qué es un Patrón de Diseño ?

Un patrón describe un problema que ocurre repetidamente una y otra vez en nuestro ambiente, y luego describe la base de la solución a ese problema, de una manera tal que esta solución pueda utilizarse varias veces mas, sin tener que hacer lo mismo dos veces. Las soluciones se expresan en términos de objetos y de interfases.

En general, un patrón tiene cuatro elementos esenciales:

1. El nombre del patrón es la forma que utilizamos para describir con una o dos palabras un problema de diseño, sus soluciones y consecuencias. El nombre de un patrón aumenta inmediatamente nuestro vocabulario de diseño. Nos permite diseñar a un nivel de abstracción más alto. Tener un vocabulario para los patrones nos permite hablar de ellos con nuestros colegas, nombrarlos en la documentación. Hace más fácil el pensar y comunicar los diseños.
2. El problema describe cuando aplicar el modelo. Explica el problema y su contexto. Puede ser que describa problemas específicos de diseño tales como cómo representar algoritmos como objetos. Puede ser que describa las estructuras de la clase o del objeto que son sintomáticas de un diseño inflexible. El problema incluirá a veces una lista de las condiciones que deben ser resueltas antes de que tenga sentido aplicar el modelo.
3. La solución describe los elementos que forman parte del diseño, sus relaciones, responsabilidades, y colaboraciones. La solución no describe un diseño o una implementación concreta, porque un patrón es como un molde que puede aplicarse en diversas situaciones. En cambio, el patrón proporciona una descripción abstracta de un problema de diseño y cómo lo soluciona un conjunto general de elementos (clases y objetos en nuestro caso).
4. Las consecuencias son los resultados y los trade-off de aplicar el modelo. Aunque las consecuencias a menudo no son expresadas cuando se describen decisiones de diseño, son críticas para evaluar las alternativas de diseño y para entender los costos y las ventajas de aplicar el patrón. Las consecuencias para el software a menudo se refieren al trade off entre espacio y tiempo. Pueden tratar también temas como el lenguaje e implementación. Puesto que la reutilización es a menudo un factor en el diseño orientado al objeto, las consecuencias

de un patrón incluyen su impacto en la flexibilidad, la extensibilidad, o la portabilidad de un sistema. Enumerar explícitamente estas consecuencias ayuda a entender y evaluar estas consecuencias.

El punto de vista afecta en la interpretación de que es y no es un patrón. El patrón de una persona puede ser un bloque primitivo de otra persona. Los patrones de diseño son diseñados sobre por ejemplo listas encadenadas y tablas hash que se puedan codificar en clases y ser reutilizados así. No son complejos, ni diseños específicos de un dominio para una aplicación completa o un subsistema. Los patrones de diseño son descripciones de objetos y clases que se comunican, que se adaptan para cumplir requisitos particulares en la solución de un problema general de diseño en un contexto determinado.

Los nombres de un patrón de diseño identifican los aspectos claves de una estructura común de diseño que es útil para crear un diseño orientado a objetos reutilizable. El patrón de diseño identifica las clases e instancias que participan, sus roles y colaboraciones, y la distribución de responsabilidades. Cada patrón de diseño se centra en un problema o tema determinado del diseño orientado a objetos. Describe cuando se aplica, si puede ser aplicado en vista de otras restricciones del diseño, y las consecuencias y los trade off de su uso.

Aunque los patrones de diseño describen diseños orientados a objetos, se basan en soluciones prácticas que se han implementado en los lenguajes de programación orientados a objetos como Smalltalk y C++ más que en los lenguajes procedurales (PASCAL, C, Ada) o lenguajes orientados a objetos más dinámicos (CLOS, Dylan, Self).

B.2 Bridge

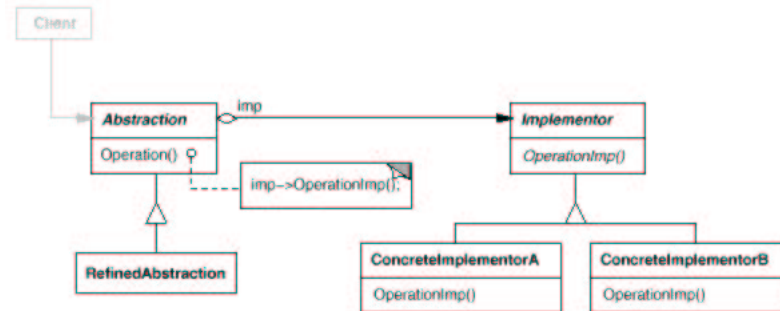
B.2.1 Intención

Desacoplar una abstracción de su implementación, de manera que ambas puedan variar independientemente.

B.2.2 También Conocido Como

Handle/Body

B.2.3 Estructura



B.2.4 Participantes

- **Abstraction**
 - define la interfase de la abstracción.
 - mantiene una referencia a un objeto de tipo Implementor.
- **RefinedAbstraction**
 - extiende la interfase definida por Abstraction.
- **Implementor**
 - define la interfase para las clases de implementación. Esta interfase no necesita corresponderse exactamente con la interfase de Abstraction; de hecho las dos interfases puede ser bastante diferentes. Típicamente la interfase de Implementor solamente provee operaciones primitivas, y Abstraction define las operaciones de alto nivel basado en estas primitivas.
- **ConcreteImplementor**
 - implementa la interfase de Implementor y define su implementación concreta.

B.2.5 Colaboraciones

- Abstraction reenvía los pedidos del cliente a su objeto Implementor.

B.3 Command

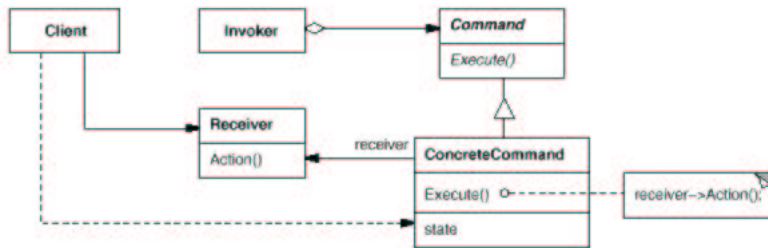
B.3.1 Intención

Encapsular una petición como objeto, permitiendo parametrizar clientes con diversas peticiones, encolarlos o registrar peticiones, y permitir deshacer peticiones.

B.3.2 También Conocido Como

Action, Transaction

B.3.3 Estructura



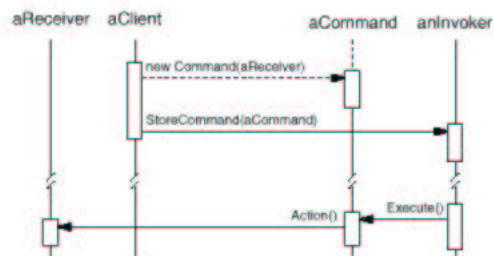
B.3.4 Participantes

- **Command**
 - declara una interfase para la ejecución de una operación.
- **ConcreteCommand**
 - define una unión entre un objeto receptor (Receiver) y una acción.
 - implementa Execute invocando la correspondiente operación(es) en el receptor (Receiver).
- **Client**
 - crea un objeto ConcreteCommand y establece su receptor.
- **Invoker**
 - solicita a Command que realice la acción.
- **Receiver**
 - sabe como realizar la operación asociada a la solicitud de una acción. Cualquier clase puede actuar como Receiver.

B.3.5 Colaboraciones

- El cliente crea un objeto ConcreteCommand y especifica su receptor.
- Un objeto Invoker contiene un objeto ConcreteCommand.
- El objeto invocante realiza un pedido invocando Execute en el command. Cuando el comando se puede deshacer, ConcreteCommand almacena información para deshacer el comando antes de invocar Execute.
- El objeto ConcreteCommand invoca la operación en su receptor para realizar el pedido.

El siguiente diagrama muestra la interacción entre estos objetos. Ilustra como Command desacopla el invocante del receptor (y el pedido que éste realiza).

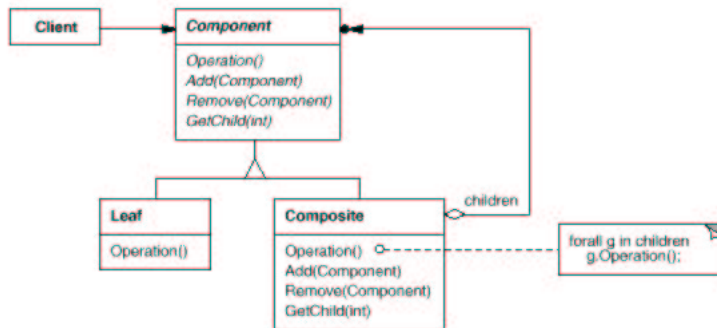


B.4 Composite

B.4.1 Intención

Componer objetos en una estructura de árbol para representar jerarquías parte-todo Composite permite a los clientes tratar en forma uniforme tanto objetos individuales como composiciones de objetos.

B.4.2 Estructura



Una estructura típica de un objeto Composite puede verse como:



B.4.3 Participantes

- **Component**
 - declara la interfase para los objetos en la composición.
 - implementa en comportamiento default de la interfase común a todas las clases.
 - declara una interfase para acceder y administrar sus componentes hijos.
 - (opcional) define una interfase para acceder a los padres de los componentes en la estructura recursiva, y las implementa si esto es apropiado.
- **Leaf**
 - representa los objetos hoja en la composición. Una hoja no tiene hijos.
 - define el comportamiento para los objetos primitivos en la composición.
- **Composite**
 - define el comportamiento para los componentes que tengan hijos.
 - almacena los componentes hijo.
 - implementa las operaciones relacionadas con los hijos en la interfase Component.
- **Client**
 - manipula los objetos en la composición a través de la interfase de Component.

B.4.4 Colaboraciones

- Los Clients usan la interfase de la clase Component para interactuar con los objetos en la estructura compuesta. Si el receptor es una hoja, entonces el pedido es manejado directamente. Si el receptor es un Composite, entonces generalmente se reenvía el pedido a los componentes hijos, posiblemente realizando operaciones adicionales antes y/o después del reenvío.

B.5 Strategy

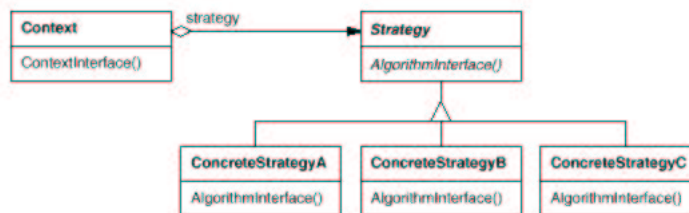
B.5.1 Intención

Definir una familia de algoritmos, encapsular cada uno, y hacerlos intercambiables. Strategy permite variar el algoritmo independientemente de los clientes que lo usan.

B.5.2 También Conocido Como

Policy

B.5.3 Estructura



B.5.4 Participantes

- **Strategy**
 - declara una interfase común a todos los algoritmos soportados. Context usa esta interfase para invocar el algoritmo definido por ConcreteStrategy.
- **ConcreteStrategy**
 - implementa el algoritmo usando la interfase de Strategy.
- **Context**

- esta configurado con un objeto ConcreteStrategy.
- mantiene una referencia a un objeto Strategy.
- puede definir una interfase que permita a Strategy acceder a sus datos.

B.5.5 Colaboraciones

- Strategy y Context interactúan para implementar el algoritmo seleccionado. Un contexto puede pasar todos los datos necesarios por el algoritmo para la estrategia cuando el algoritmo es ejecutado. Alternativamente, el contexto puede pasarse a sí mismo como argumento de la operación de Strategy. Esto permite a la estrategia realizar *call backs* cuando sea necesario.
- El contexto reenvía los pedidos de sus clientes a su estrategia. Los Clients usualmente crean y pasan un objeto ConcreteStrategy al contexto; luego, los clientes interactúan exclusivamente con el contexto. Siempre hay una familia de clases ConcreteStrategy de los cuales el cliente puede elegir.