# Patterns for GIS Applications Design

F. Balaguer, S. Gordillo*, F. Das Neves
LIFIA-Departamento de Informática, Facultad de Ciencias Exactas, UNLP
CC 11 (1900) La Plata, Buenos Aires, Argentina
[fede,gordillo,babel17]@sol.info.unlp.edu.ar
Tel/Fax: (54) (21) 228252
*also CIC-Pcia de Buenos Aires

## 1.-Introduction

Geographical Information Systems are used to collect, analyze, and present information describing the physical and logical properties of the geographic world [Shekhar97]. The complexity of the underlying domain of GIS applications, the variety of data types including spatial data and sophisticated relationships and the strong need for performance and accuracy in the final product, usually lead the GIS design task as a process closer to the implementation than to a software engineering process.

The problem of reusing design in GIS applications has become also a need because in the last times, more and more users are building GIS applications based on open systems instead of using a particular GIS product (like ARCInfo, GENAMAP, etc). Examples of solutions to extend traditional applications with spatial features are proposed in [Postemesil97], where applications which deal with very large databases containing geographical information (terrain elevation, satellite and aerial images, detailed street maps, etc.) have to be built based on conventional applications.

Web applications providing access to geographic data captured from legacy systems are another example of this kind of applications [ESRI97]. Developers have to write custom applications to visualize the required information.

Understanding and representing spatial features is usually, the first problem solved by every GIS model. Relationships and operations over geographical objets are commonly more complex than those defined in conventional applications. GIS require computing, for example, areas that were influenced by some phenomenon, or entities holding a particular spatial property.

[Tryfona95] has recently proposed using objects as the basis for the design of GIS applications. We claim, however, that object-oriented design methods and class libraries are only a part of the solution to the problem of GIS design. We think that the GIS community must record its design expertise in terms of Design Patterns as shown in this paper.

In the following section we explain how to extend a conceptual model (probably taken from a legacy system) to a geographic model by using an existing Design Pattern (Decorator). The use of Decorator makes it possible to obtain a geographic application minimizing redesign. This short explanation about the use of Decorators is made for a better understanding of the proposed Design Patterns described in section 3: the Reference System and Roles Design Patterns.

## 2.-Decorators and Geographic Objects

The application of Decorator pattern in the context of Geo Domain is presented below even when it is not a new one, because it could help to a better understanding of this section.

*Problem:*

A designer must face hybrid applications dealing with conventional transaction-based systems that must be upgraded to include spatial features, that are not built in the underlying software.

*Forces:*

- The development has to be based on a legacy system, which imposes “legacy architectures” and influences over all design decisions.
- Two sets of information and behaviors, that are applied to different concerns, must be supported: the conceptual and the geographic behavior. They are related to the model that support the legacy system and also with the geographic one.

- Enhancing the behavior and knowledge of the existing classes produce a "dirty" solution, since it mixes responsibilities, produces large protocols that are difficult to understand, and could have impact on the class hierarchies.

*Solution:*

Identify two underlying models: the conceptual model and the geographic one. It makes possible to decouple the problem into two different stages, thus limiting the concerns they have to deal with simultaneously.

To extend the conceptual model to a geographic one, the Decorator design pattern has to be applied. In this way it is possible to add spatial behavior to conceptual objects. Since Decorators provide a flexible alternative to subclassing for extending functionality [Gamma95], the obtained solution is a design where both, conceptual and geographic objects can be manipulated.
The relevance in the use of decorators in this way is that they can be applied in any GIS application, using it as a design tool besides particular situations. Moreover, this approach produces a more flexible design since we can modify and reuse conceptual or geographic objects.

Also, there may be objets that do not have an associated conceptual object (because they only have spatial features) and therefore, they only belong to the geographical model. We define an abstract class, which groups the common behavior of those objects belonging to the GIS application model, plus those that have been wrapped from the conceptual model. Figure 1 shows this hierarchy.

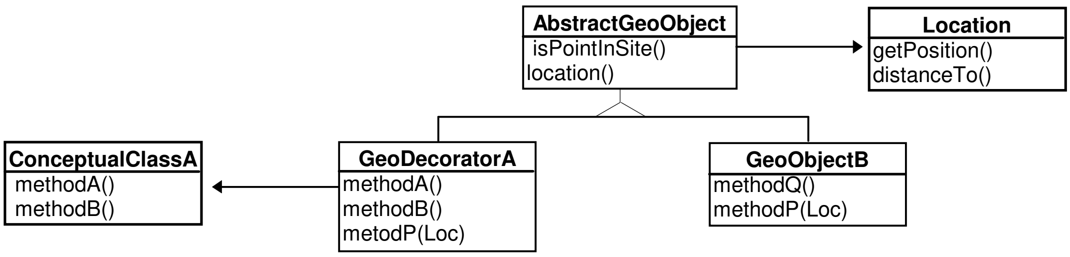


**Figure 1: A Geo-Object hierarchy. including both decorators of existing classes, and geographic objects with no relation to conceptual objects.**

The abstract class AbstractGeoObject defines the protocol to manipulate geo-referenced features, like behavior of general geographical functions (a point belonging to an area, perimeter, etc.). Also, a geographical object always knows a location, this definition is crucial because it makes the difference between a geographical object and a conceptual one. Location has been defined in [OGIS96] and it contains behavior about position and temporal data. Location also has a reference system and the necessary transformations to manipulate different kinds of geo-references (the Reference System design pattern is explained in section 3). Finally, geometry defines the representation that a geographical object has in a system, basically there are three different ways to represent an object in the geographical model: as a polygon, as a line or as a point. Figure 2 shows the relationships among Conceptual classes, GeoClasses, their locations, the geometry and the reference system.

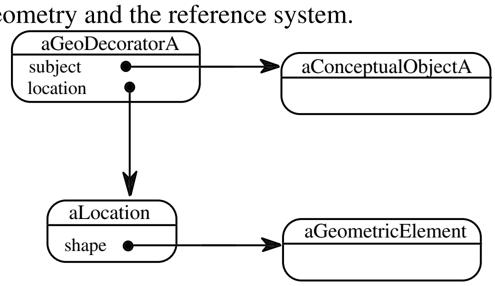


**Figure 2: Relationship among Conceptual Classes, Geo-Decorators and Locations.**

Known uses:

The process of dividing models in two mayor parts is shown in [Shekhar97], where there are some entities (Feature, SpatialObject, etc.) which are used to separate abstractions without geographical features from the geographic stuff.

## 3.-Geographical Patterns

In this section we present some patterns that have been discovered in the domain of geographic systems.

### 3.1-Reference System

*Intent:*

Associate a value with a reference system. The reference system provides a knowledge background related to the value. Without a reference system, the measure acts like an isolated value, thus missing testing, comparing and translating capabilities. Besides, the reference system provides a set of legal operations; this set conforms to the reference system's arithmetic.

*Motivation:*

Building a simple application to manage materials and products workflow inside and outside a factory, we must represent two different kinds of positions. First, a position referred to sections inside the factory, which is used to represent the workflow inside de plant. Second, a position defined within the latitude/longitude abstraction. This "outer position" is used to check out the delivery state.

In a more general sense, we can see there are different ways to represent the position of an entity on the earth, each of them based on one reference system. A reference system makes possible to explain measures and to describe operations (such as: distance, area, volume, etc.). While we are used to coordinate systems where the module of a vector is zero only if the vector is a point lying in the origin (another point), there are other reference systems with different definitions of module and distance, For instance, in ellipsoid coordinates; three magnitudes are required to use this system, two angular measures and one scalar. While angles represent latitude and longitude respectively, the scalar one represents the altitude from the surface of the earth. Thus, it is possible to model the coordinates' center like an ellipsoid. Figure 3 shows a simple representation of the ellipsoid reference system's center. Angle λ represents the longitude while φ is the latitude. The value of the scalar measure is zero on the ellipsoid's surface. Thus, we can see that multiple reference systems may be used to support different abstractions of the Earth.

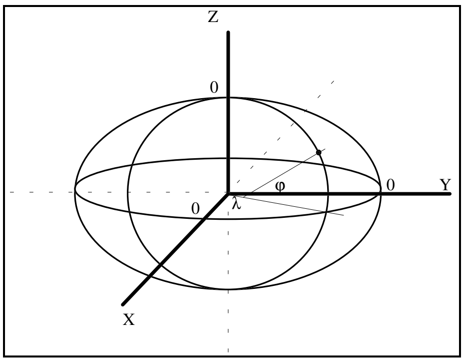


**Figure 3: Ellipsoid Reference System Center**

These reference systems were originally designed to produce the latitude/longitude abstraction. This abstraction is the main idea in which other systems are based on. But there is another kind of reference system, which is based on well-known georeferenced points (called tics). All these tics constitute a Local Reference System for any position measured with respect to them. This kind of reference system is widely used in building projects such as road construction, highways, canals, backwaters, etc. To complicate things further, the whole local system can be translated to a global one (based on latitude/longitude). In the example, we define a local reference system based in plant sections and a global one based on latitude/longitude.

*Forces:*

- ❑ A measure has to be explained by the knowledge given by a specific domain. In the GIS domain, a value of a position such as latitude/longitude is measured in grades, but there are many different ways to understand this measure (polar latitude/longitude or planar latitude/longitude)

- ❑ Building a hierarchy representing different latitude/longitude abstractions, such as ellipsoid latitude/longitude, planar latitude/longitude, etc., produces a static structure, making difficult work with a new reference system.
- ❑ It has to be possible to operate with measures defined within different reference systems.
- ❑ A measure defined within a reference system could be translated to another reference system

*Solution:*

Define an object which represents the relationship between a measure and a reference system; using it, the measure can be described. This object is responsible to encapsulate the relationship between a measure and a reference system.

This pattern is useful for supporting multiple ways to represent geo-referenced entities of the real world based on different earth abstractions. Furthermore, it makes possible to change or translate from one system to another one. This change does not affect the referenced object. The value may be expressed like single values, coordinates or could be measured following the approach proposed in [Fowler97].

Each reference system has to implement its legal operations such as: computing distances between points, comparing elements, calculating areas, etc.; also it has to specify translation operations to other systems.

Finally, a reference system models the context in which the measure has a specific meaning. Distances, areas, and volume calculations are expressed by collaborations between measures and a reference system.

*Structure:*

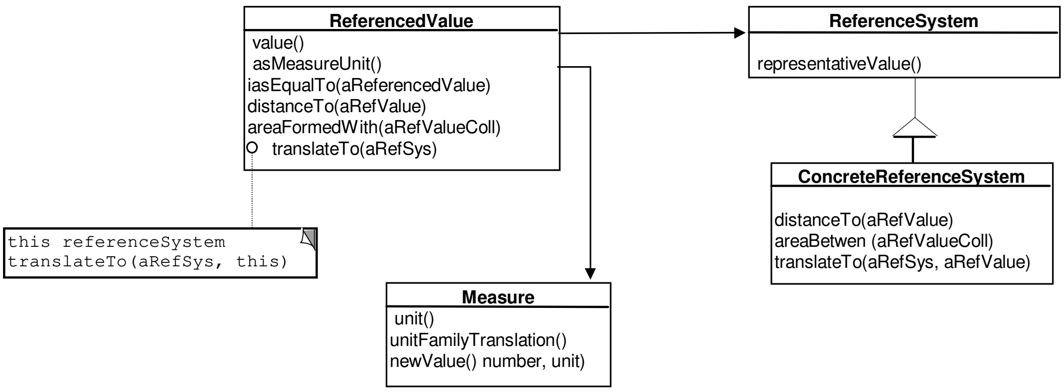


**Figure 4. Structure of the Reference System Pattern**

*Participants:*

ReferencedValue: implements the basic behavior to support the representation of the value within a reference system. ReferencedValue provides to a georeferenced object a standard protocol to implement operations that are legal in a specific ReferenceSystem. Also it implements the abstract behavior in order to accomplish translation operations to the unit used by the system.

ReferenceSystem: defines an abstract protocol, which is used to describe the context where a ReferencedValue is defined. It also defines the set of legal operations. This set has to be defined by the ReferenceSystem. Each ReferenceSystem describes the way to understand measures defined by a ReferencedValue.

Measure: is the value that has to be described into a ReferenceSystem. It is known by ReferencedValue. It collaborate with the reference system to accomplish geometric operations. It implements the Basic translation operations among units.

*Collaborations:*

- ❑ ReferencedValue delegates in the Measure object, all behavior related to translation operation between units.
- ❑ ReferenceSystem hierarchy implements geo-operations which are those related to positioning, area and distance calculation
- ❑ ConcreteReferenceSystem collaborates with ReferencedValue in order to implements translations to other ReferenceSystem.

*Consequences:*

- It associates measures (such as positions) with the reference system that is being used.
- It provides a way to change the implementation of operations without changing the Reference System, or the Location.
- If only one reference system is used or if it is 1-dimensional, the pattern produces an overhead because it works within fine granularity abstraction.

*Know uses:*

The model proposed in [OGIS96] defines a Location object which deal with geographical data (place and time). Location is used within a specific Spatial/Temporal Reference System. All entities and phenomena have location attached as part of their description.

*Sample Implementation:*

Figure 5, shows the sample structure of the Reference System pattern. It is worth to note that in this scheme there are classes that do not appear in the list of participants. The reason is that the details regarding how a value is related to a reference system may vary.

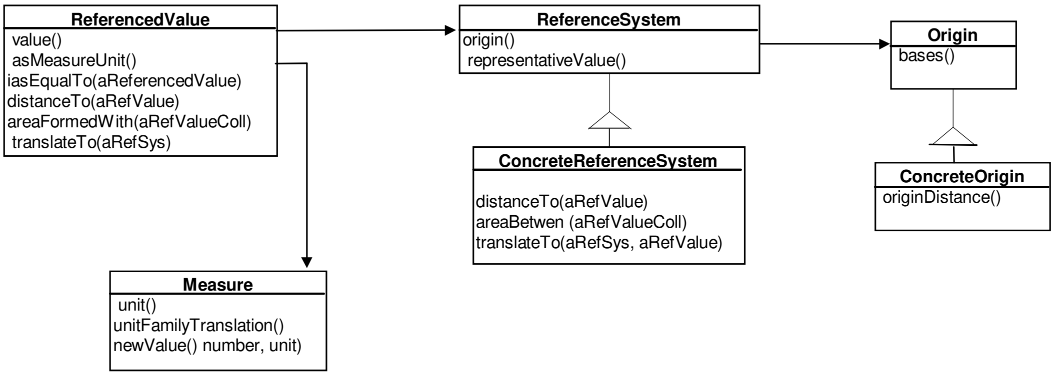


**Figure 5. Sample Structure of the Reference System pattern.**

In the sample implementation, the reference system Origin is defined. In order to calculate the distance between two points, the reference system generates a new point with the result of the difference. The distance is calculated taking the module of the vector that goes from the origin of the Reference System (not necessarily a point) to the result point.

In the same way, it is possible to model other reference systems based on spheroids or planes. The Origin describes the bases of a ReferenceSystem, it implements raw operations.

*Related Patterns:*

Bridge: The RefereceSystem hierarchy implements operations which are defined like an interface by the ReferencedValue. ReferencedValue acts the *Abstraction* role, while RefereceSystem the *Implementor* one. It is possible to change the ReferenceSystem (using its translation operation) thus the final implementation change too. Afterward it is possible to add new ReferenceSystem subclasses which implement in different ways the protocol defined by ReferencedValue.

Observations and Measurements: The value may be expressed like single values, coordinates or could be measured following the approach proposed in [Fowler97]. This pattern system describes a solution in order to encapsulate unit families; which implement mathematical and translation operations.

Instances of the ReferenceSystem hierarchy can be implemented following the Singleton pattern [Gamma95] in order to minimize the amount of instances.

### 3.2.-The Role design pattern

*Intent:*

Represent different geographic subjects of a geo-object. Roles let you implement subgroups of related geographic aspects of geo-objects (the roles). Roles lets you vary the granularity of each geographic aspect, and allow them to evolve independently from the geo-objects.

*Motivation:*

One of the characteristics of geographical applications is the need to represent many different spatial aspects, which have independent features and behavior. For example, consider that we are modeling a country in a geographical application. There are different aspects about the country that we might be interested about such as: soils, demographic areas, climates, ozone distribution, topographic characteristics, etc. Each one of these roles has its own behavior and describes information about the country. The soil aspect , for example, can identify each soil type in each region of the country, while demographic areas calculate the population in a specific area or in the whole country, and so on. However, the granularity and structure of each aspect can be different; some of the aspects, like soil type, are independent from other aspects, while others like demographic information are closely related to the political structure of the country in terms of states, cities and so one.

*Forces:*

- ❑ A geo-object needs to support different geographic features that will be responsible for many groups of related information with different granularity.
- ❑ Adding behavior to a class in order to model all geographic features for a particular application leads to monolithic classes with mixed responsibilities.
- ❑ Generating subclasses based on the feature's differences produces multiple classes in different hierarchies that stand for the same subgroup of information. This partition does not model the recurrent essence of the problem.

*Solution:*

Decouple the responsibility of managing different aspects from the corresponding geo-object, by defining a hierarchy of roles associated to a geographical object. Roles add information to the geo-object, corresponding to a particular geographic aspect. Roles are always related to a geo-object, like decorators, but they know how to gather information from other role object associated to composite objects.

*Participants:*

GeoObject: An object with spatial characteristics belonging to the GIS model, that may be considered playing different roles.
ConcreteGeoObject: Implements a concrete spatial object, and optionally creates a set of roles to
Role: defines the abstract behavior of all roles. A role implements a subset of the behavior needed for geo-objects (an aspect).
CompositeGeoObject: a GeoObject that hosts other GeoObjects, following the structure defined in the Composite pattern [Gamma95].
ConcreteRole: a concrete implementation of a subset of behavior needed for a geographic object to accomplish a set of responsibilities in a specific context.

*Collaborations:*

Geographic Object delegates the responsibility of manipulating information and behavior subsets to the corresponding role objects.

Role may forwards requests to its geographic object. It may optionally perform additional operations before and after forwarding the request.

*Structure:*

Roles can be derived from other roles. Suppose that we are interested in the population of a country and its states. Then we define a role of Demographic Areas. The role Demographic Areas in a country is a derived role which takes information for the role Demographic Area assigned to states, as shown in figures 6 and 7. The country has a Derived Demographic role, which computes the total population from the population of the parts of its owner. States have an atomic demographic role, which actually keeps the number of people living in the state.

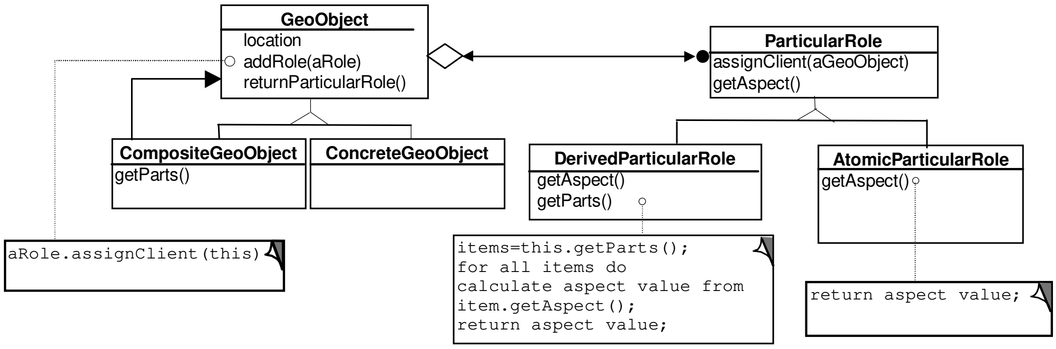


**Figure 6: Relationship between a GeoObject and its roles.**

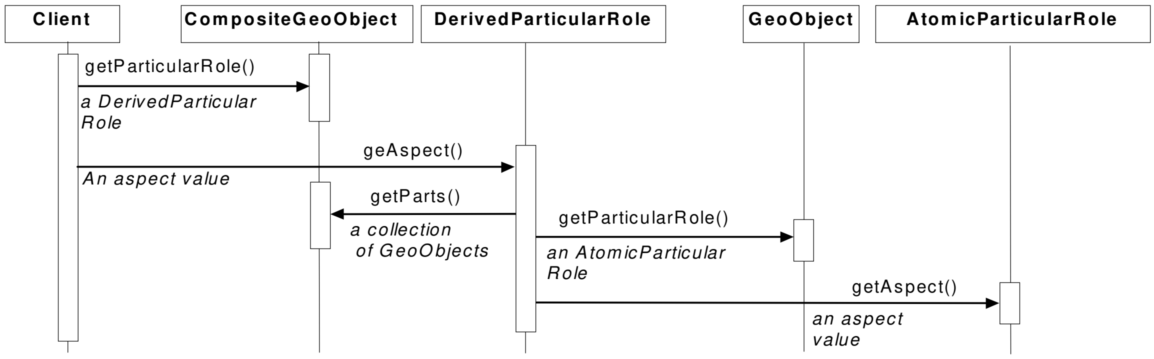


**Figure 7: Interaction diagram showing how a role (DerivedParticularRole) attached to an object within a composition hierarchy perform s getAspect()) by way of roles attached to other objects downward the hierarchy. Method calls are written with ending parenthesis, and returned objects are indicated in Italics.**

Also, note that both derived and atomic demographic roles are not tied to Countries and States, and work with any other geo-objects that need to keep demographic information or compute this information from other (polymorphic) objects. Different kind of roles can be defined and dynamically attached to geo-objects.

*Consequences:*

- ❑ It decouples behavior related with a particular role from the basic operations of a class.
- ❑ Roles provide the basic information to construct layers, a partial thematic view of a group of geo-objects. Layers provide a conceptual framework to reason about different aspects of the same geographical object, or to combine information from different ones. A layer is a set of roles defined by a query predicate applied over a set of geo-objects, and could be constructed with information about only one role or combining information from many of them. They could also be constructed from different roles of different geographical and conceptual objects. To build a layer we define a builder [Gamma95] which combines the required information of one or more roles belonging to the same or different geographic object. The builder is able to access geographic objects and roles, and generates a layer with the required information.
- ❑ Allow to mimic a composition hierarchy without duplicating it, and dynamically extend the object responsibilities and information granularity without changing the protocol of geo-objects . Roles can be implemented so that they can be attached to geo-objects within a composition hierarchy, and they use the protocol of geo-objects in order to access roles attached to other members of that hierarchy. As roles can be defined independently from geo-objects, the way they follow the structure of a hierarchy can be dynamically altered to reflect changes in the detail of the information, without changing the definition of geo-objects.
- ❑ The fact that roles are independent from the geo-object and from other roles allows more than one implementation for the same role type, that can gather data from different information sources. Separating the spatial from non-spatial data applying roles eases the implementation in traditional GIS systems running on relational databases.

Related Patterns:

Bridge [Gamma95]: Roles is related to Bridge in intent and applicability. GeoObjects and Roles forms two class hierarchies with a mutual interface/implementation contact between a geo-object and their roles. Roles are implementations of specific subsets of information about geo-objects, and in turn roles that mimic a composition hierarchy rely on the geo-objects to implement the protocol to traverse the hierarchy. However, both geo-objects and roles are separate hierarchies, and so they can evolve independently.
Also Roles, like Bridge, are applicable to divide the responsibilities among roles and geo-objects to defer the selection of a particular implementation of a role to run-time, and to avoid the proliferation of subclasses implementing different sets of responsibilities for the same kind of geo-object.
Adapter [Gamma95]: Roles can be thought as adapters of the GeoObjects, dynamically adding new responsibilities to them. The main difference is that in the case of the Adapter pattern, clients always knows the adapter and only optionally the adaptee, while in the case of Role, clients always knows the GeoObject (the adaptee), and request it for specific roles (the adapters).

## 6.-References


[ESRI97] Live MapObjects Internet Map Server Demostrations. Available at "http://maps.esri.com/ESRI/mapobjects/demos.htm"

[Fowler97] M.Fowler: "Patterns: Reusable Object Model". Addison Wesley, 1997

[Gamma95] E. Gamma, R. Helm, R. Johnson, J. Vlissides: *"Design Patterns. Elements of reusable Object-Oriented Software"*. Addison Wesley, 1995.

[OGIS96] Open GIS Consortium (OGC) (1996b), The Open GIS Guide -A Guide to Interoperable Geoprocessing, Available at http://ogis.org/guide/guide1.html

[Postmesil97] M.Postmesil: "Maps Alive: Viewing Geospatial Information on the WWW". Proceedings of the Six International World Wide Web Conference, 1997. Available at http:// www6.nttlabs.com/Hypernews/get/PAPER130.htm

[Shekhar97] S. Shekhar, M. Coyle, B. Goyal, D. Liu, and, S. Sarkar: "Data Models in Geographic Information Systems" Comm. of the ACM, April 1997, pp 103-111

[Tryfona95] N. Tryfona and T. Hadzilacos: *"Geographic Applications Development: Models and Tools for the Conceptual Level"*. Proceedings of ACM-GIS'95. Baltimore, Maryland, USA, pp 19-28.