

# **Feature Representation in Space, Theme, and Time**

Jinmu Choi (jmchoi@uga.edu)

Department of Geography

The University of Georgia

## **Abstract**

With the emergence of geographic information systems (GIS), spatial data have been represented by a traditional layer-based data model that contains the spatial geometry and thematic attributes. In the real world, geographic phenomena also have temporal attributes and relationships. Since late 1980, a feature concept has been developed that can be a unit of aggregation and analysis of spatial phenomena instead of the traditional theme or layer. A feature approach in developing GIS has the potential for more accurate representation of geographic reality because the features correspond to the basic level of human cognitive observation.

The purpose of this study is to derive a feature data model and to develop a feature-oriented GIS. A feature data model was derived basically from the feature concept, represented by three dimensions: space, theme, and time. Classes in the feature data model were derived using abstraction in spatial data transfer standard (SDTS) and key concepts of object-orientation.

A prototype system was then developed using the feature data model. This system consists of an import/export module, a feature database, and a display module. Existing spatial data, such as shape files, can be directly imported into the feature database. Specifically, the display module can be used to effectively visualize the hierarchical structure of objects in the feature data model. The developed system was applied for the visualization of multi-temporal representations of coastal features in Camp Lejeune, North Carolina. This feature-oriented system shows the possibility of better representation of geographic features in GIS.

## 1. Introduction

With the emergence of geographic information systems (GIS), the data model for capturing geographic phenomena has been focused on the spatial geometry and thematic attributes of the phenomena have been linked to the geometry. The basic geometries in the geometric model are point, line, polygon, and grid. Different geometric types need different types of data structure with their thematic attributes, so multiple geometric types cannot be handled by one theme or layer.

The layer-based model emerged to effectively manage the geometric based data. As it allows only one type of geometry to be entered into a theme or layer, multiple geometric types of geographic phenomena must be represented by multiple layers. In the real world, geographic phenomena have not only spatial geometry and thematic attribute, but also temporal situation and relationships among phenomena. The layer-based data model, however, cannot contain the relationships among phenomena. Also, one feature in multi-temporal situations cannot be represented by a layer and more than two layers must be used for multi-temporal situations. The same phenomenon in different layers may not be the same any more because it may have different IDs in different layers.

Since late 1980, there have been significant advances in both the data model of GIS and the programming paradigm of computer science. A feature concept has been developed to represent geographic phenomena in GIS (Guptill et al., 1990; USGS, 1997). A feature is a defined geographical entity and its object representation in the digital environment, so that it is the unit of aggregation and analysis instead of a theme or layer (Usery, 1993). A feature is also represented by multiple dimensions using space, theme, and time, with their relationships among features. Therefore, the feature approach for a geographic data model may be the best solution for the multiple representations of real world geographic phenomena. In computer science, the object-oriented paradigm has been applied to various fields such as programming and data modeling (Booch, 1994;

Graham, 2001). Since 1990, several researchers have attempted to apply object-oriented concepts to GIS.

Some of researchers have used object representations directly for geographic phenomena without feature concepts. Worboys et al. (1990) tried to build object-oriented data models for spatial databases. For a database design, the means of representation is provided by the data model. Using only relations in a relational model, a common difficulty in computer-aided design (CAD) and GIS is the gap between the richness of the knowledge structures in the application domain and the relative simplicity of the data model in which these knowledge structures are expressed and manipulated (Codd, 1970; Chen, 1976; Worboys et al., 1990).

Other researchers have used feature-based representations of geographic phenomena in applying object-oriented concept to GIS. A feature-based approach in building GIS has the potential to better support geographical models and analytical procedures with accurate representation of geographic reality because the features correspond to the basic level of human cognitive observation (Usery, 1993 and 1994b). When people are questioned about what is viewed in a geographic scene, the answers are not likely to include geometric elements such as points, lines, areas, and grid cells, but rather geographic entities such as rivers, lakes, buildings, and roads.

These studies have shown that the object-oriented paradigm is better for GIS than conventional data models and programming methods. The object-oriented paradigm is closer to real world situations, features and their relations, which are very complex. GISs deal with very complex spatial data that are captured from the real world.

The purposes of this paper are: 1) to derive a new geographic data model named as a feature data model for a feature representation using spatial, thematic, and temporal dimension; 2) to develop a prototype system as the implementation of the new feature data model with the object-oriented programming; and 3) to design a feature database from the new feature data model for use with the

developed prototype GIS. The developed system has been applied for geographic data representation for multi-temporal situations of a beach area along the North Carolina coast.

## **2. Feature Data Model with Object Orientation**

A feature data model can be constructed basically from the feature conceptual framework, the abstraction methods defined in spatial data transfer standard (SDTS), and key concepts of object-orientation. With the feature conceptual framework, features are represented by three dimensional aspects: space, theme, and time. Classes in the feature data model can be derived from the abstraction methods of spatial data in SDTS. System design and implementation are based on the object-orientation.

### **2.1 Conceptual Framework of a Feature**

Generic lists of features useful in GIS operations have been compiled in the past, but the lists tend to vary from one organization or individual to another according to their applications (Guptill et al., 1990; USGS, 1997). The Topologically Integrated Geographic Encoding and Referencing (TIGER) system of the U.S. Census Bureau, Digital Line Graph-Enhanced (DLG-E) of the U.S. Geological Survey (USGS), and the Feature and Attribute Coding Catalogue (FACC) of the National Geospatial Intelligence Agency (NGA) are examples of feature lists that are useful for each organization (Trainor, 1990; Guptill et al., 1990; Defense Mapping Agency, 1987; Digital Geographic Information Working Group, 2000). Therefore, Usery (1993) tried to build a conceptual framework of features, which should be effective for various applications and resolutions using region theory, cognitive category theory, cartographic principles of abstraction, and set theory.

Region theory in geography can be used as a basis for the geographic feature concept because a region is an intellectual concept and an entity for the purpose of thought (James and Jones, 1954). A feature is also an intellectual concept with attributes and relationships for a particular problem with

relevant characteristics. Cognitive category theory can be used to aggregate features based on similarity for a particular application and resolution (scale) (Rosch, 1978; Lakoff, 1987). The cartographic principles of abstraction and generalization of geographic phenomena are well-established for particular map scales (resolutions) and purposes (applications) (Robinson et al., 1984; McMaster, 1989). Set theory is used to structure thematic groupings of geographic phenomena which have no ambiguity and are established by a collection of rules (Gatrell, 1983 and 1991; Usery, 1996a).

With the conceptual framework for structuring features in GIS, a prototype feature-based GIS (FBGIS) has been developed for both vector (Tang et al., 1996; Usery et al, 2002) and raster data representation (Usery, 1994a and 1994b). The framework includes spatial, thematic, and temporal dimensions with attributes and relations for each domain to build FBGIS (Figure 1). Although FBGIS has been implemented using conventional programming techniques, the logical implementation strategy for the feature-based conceptual model was the object-oriented approach in which objects are constructed based on either sets or category theory concepts of prototypes.

	Space	Theme	Time
Attributes	point, line, area, volume, pixel,...	color, size, shape, ...	date, duration, period, ...
Relationships	topology, direction, distance, ...	is_a, kind_of, part_of, ...	is_a, was_a, will_be, ...

Figure 1. Conceptual Framework of a Geographic Feature (Usery, 1996b)

## 2.2 Feature Abstraction Methods in SDTS Data Model

With the concepts of a feature, each dimension consists of two more abstraction levels: data models and data structures (Egenhofer and Herring, 1991). For a feature data model, object-oriented

models have the capability to express more readily the knowledge structure of real world features than relational models. Features are captured by an abstraction using generalization, specialization, aggregation and association. Abstraction arises from recognition of similarities between certain objects, situations, or processes in the real world and denotes the essential characteristics of an object that distinguish it from all other kinds of objects and thus provide crisp boundaries of feature classes (Booch, 1994).

The conceptual model of the SDTS is a good example of abstraction for geographic features (USGS, 1997). The SDTS conceptual model has three parts: a model of spatial phenomena, a model of the spatial objects used to represent phenomena, and a model of spatial features that explains how spatial phenomena and spatial objects are related. Four basic terms of object-oriented model have been defined and explained in SDTS for spatial data so that each term can be directly used for a feature data model (Table 1). Classification can be used for defining a feature class. Generalization and aggregation can be used for grouping features into a feature type class. Association can be used for describing attributes and relationships of a feature.

Table 1. Terms for Spatial Data Abstraction in SDTS

Term	Definition	Example
Classification	The assignment of similar phenomena to a common class.	Route 10, Lake Herring, and Chesterfield County are all classified phenomena.
Generalization	A process in which classes are assigned to other classes. The general class includes all the instances of the constituent classes.	Sewers are included in the more general class of utilities.
Aggregation	The operation of constructing more complex phenomena out of component phenomena.	A lock is an aggregation of walls, gates, and a reservoir.
Association	The assignment of phenomena to sets, using criteria different from those used for classification.	Concrete roads may be associated with concrete sewers, concrete locks, and other phenomena constructed of concrete.

## 2.3 Key concepts in Object-Orientation for Constructing Feature Data Model

Even though object-oriented terms are ideal methods for a feature data model, the object-oriented model cannot replace the feature data model because an object can represent not only a feature but also a feature type, relationship, or any kind of geometric types. All things are objects in the object-oriented paradigm.

Three key concepts of the object-oriented paradigm are inheritance, encapsulation, and polymorphism. In an object-orientation with a feature, the encapsulation can hide information (data and method) in a class definition over hierarchical inheritance so that a class can be designed to represent characteristics of a feature that are relevant to a particular application. For example, schools and houses inherit attributes and methods of buildings that are basically area features. If schools have to be represented as area features and houses have to be represented as point features according to the resolution of an application, the houses need not inherit the area size attribute or the area size calculation method. A building class can encapsulate specific data and methods by constraints (Borgida, 1988).

The inheritance from super class to sub class makes abstraction of geographical features possible. For example, highway and cart track objects can inherit instance variables and methods from a road object. Sub class objects (highway and cart tracks) may have additional instance variables and methods unique to the sub class that the super class object (road) does not have. Polymorphism allows the same message with different instance references to invoke different methods for the appropriate classes according to the instance reference so that spatial operations on different but similar objects are made easier. For example, a method for distance measurement between two features can be applied to compute the distance between two buildings and between a building and a road, although the internal algorithm of computation is different.

## 2.4 Feature Data Model for Geographic Phenomena

In this research, a feature data model was derived from the feature conceptual framework with the abstraction methods defined in SDTS and key concepts of object-orientation. With the feature conceptual framework, features are represented by three dimensional aspects: space, theme, and time (Figure 2). Classes in the feature data model were derived from the abstraction methods of spatial data in SDTS. A feature class was derived by classification. Feature, time, and attribute type classes were derived by aggregation and generalization. Finally, three attribute classes and various relationship classes were derived by association.

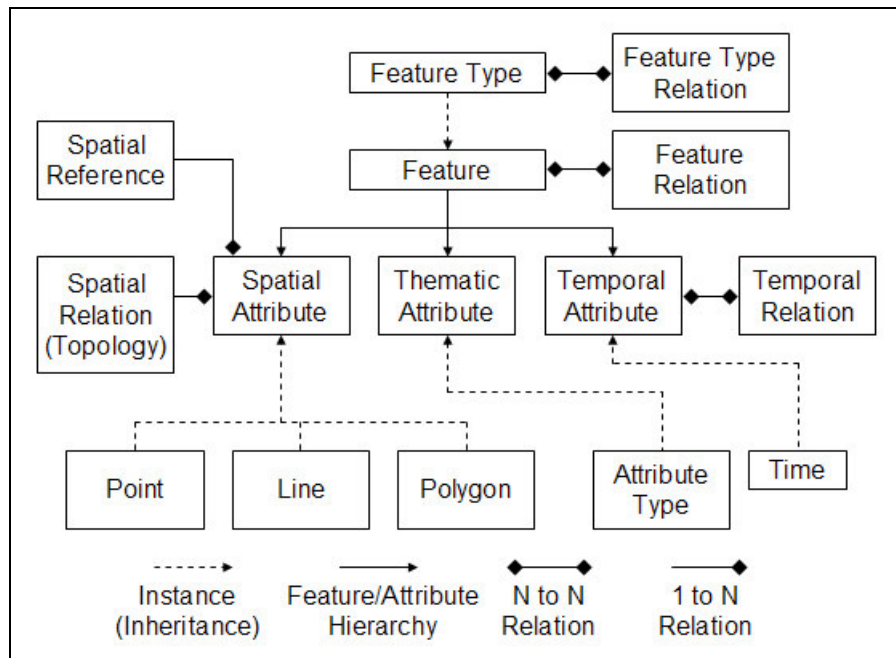


Figure 2. Feature Data Model using Object-Oriented Paradigm for a Vector Data

Each dimensional attribute is connected to its own relationships. Spatial attributes are connected to the spatial relationship (or topology) with a many-to-one relation. Spatial attributes of each feature are parts of a spatial relationship. A spatial reference class can represent a geographic coordinate or a specific projection so that it has a one-to-many relation with spatial attributes. Temporal attributes are connected to temporal relationships with a many-to-many relation. Two

temporal attributes are used for a temporal relationship and more than two relationships can reference the same temporal attribute. Especially, the thematic relationships have been modeled by feature relationships because the thematic relationships in Figure 1 represent do not the relationships between thematic attribute themselves, but the relationships between features. Feature relationships also have a many-to-many relation with features. One feature relationship composed of two features and more than two feature relationships can refer to the same feature.

A feature type class has been devised to represent a group of features based on their similarity. For all land parcels in general, a 'Land Parcel' as a feature type to which all land parcel features belong can be created. Both feature types and features are related to each other in the real world. These relationships can be implemented in a feature data model as a feature type relationship class and a feature relationship class that are used for binding two sets of feature types and features together, respectively. Relationship classes denote some form of direction such as 'flow into', 'belong to', 'part of', 'next to', and so on. Figure 3 represents feature type instances and their feature instances with their relationships. The 'Belong\_To' relationship is an instance of feature type relationship. 'Next\_To' and 'Brother\_Of' relationships are the instances of feature relationship.

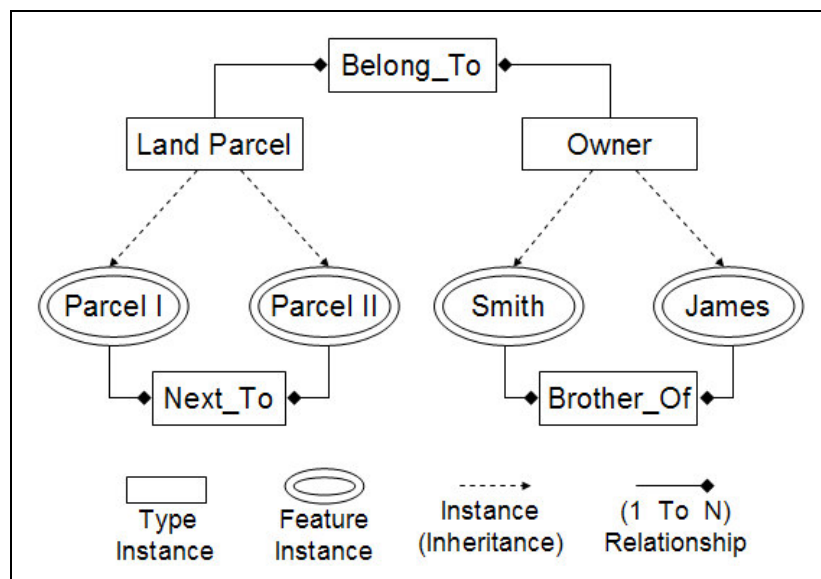


Figure 3. Feature Type and Feature Instances with their Own Relationships

In the object-oriented paradigm, an object in a different time can be saved as the same object with different attributes or situations because an object can be instantiated as two instances that inherit basic attributes and different situations through the time. If there is a road named ‘Broad Street’ and that road has been changed from September 1, 1970 to October 1, 2000 by constructing a new shortcut between two locations on the road which was originally a bypass, in an object-oriented design, a feature instance called ‘Broad Street’ has two sub-trees of attributes, one for attributes bound to September 1, 1970 and the other to October 1, 2000 (Figure 4). Each sub-tree contains a line describing a geometric attribute and a thematic attribute such as a cover type and a lane. Therefore, ‘Broad Street’ can have two instances through time and be saved as one feature without duplicating the other roads which have not changed through time.

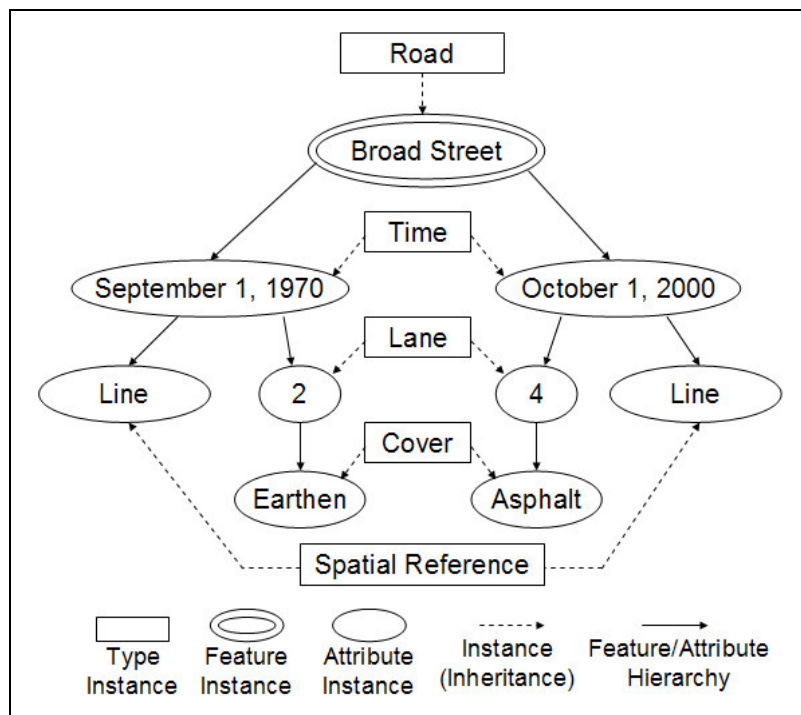


Figure 4. Feature Representation with Two Temporal Attributes

Attributes can be derived from Attribute Type objects to give a description in that ‘4’ by itself has no meaning but ‘4’ derived from ‘Lane’ Type means 4 lanes (Figure 4). Therefore, Type in a feature

data model is a convenient mechanism for making general descriptions for features and for imbuing attributes with more meaning.

Spatial attributes are used for spatially grounding these features to reality. In a feature data model, only three simple geometries among several geometric objects in the SDTS conceptual model are used to represent spatial attributes: point, line, and polygon (see Figure 2). A point is for a geometric point, which represents a coordinate  $(x, y, z)$ . A line is merely a list of points. A polygon is just a line with the implication that it forms a loop, which is an essentially a list of points that describe a simple, closed polygon.

### **3. System Development for Feature Data Model**

With the feature data model, a prototype system was developed. Classes in the data model were designed and used to build the system. The developed system has three main parts: import/export module, display module, and database for feature storage. A tree viewer has been designed in the display module to effectively visualize the hierarchical data structure of features. For the storage of features, the feature database has been designed and implemented using table structure. Using the import/export module, existing spatial data such as a shape file can be easily imported into a feature database.

#### **3.1 Class Design**

In the feature data model represented in Figure 2, fifteen classes have been identified for representing geographic features: Feature Type, Feature Type Relation, Feature, Feature Relation, Spatial Attribute, Spatial Reference, Spatial Relation, Point, Line, Polygon, Thematic Attribute, Attribute Type, Temporal Attribute, Time, and Temporal Relation. With the exception of two classes (Feature and Spatial Reference), these classes can be grouped into four groups according to the similarity of the usage of the class: type, attribute, and relationship, and geometry classes (Table

2). Feature class is storage of feature instances. The spatial Reference class, derived for a visualization purpose as a total view of geographic features, has coordinate and projection information.

Table 2. Groups of Classes in Feature Data Model

Groups	Classes	Usage
Type	Feature Type, Attribute Type, Time	Storage of descriptive information for referring classes
Attribute	Spatial Attribute, Thematic Attribute, Temporal Attribute	Storage of detail value for a feature
Relationship	Feature Type Relation, Feature Relation, Spatial Relation, Temporal Relation	Storage of detail relationship for a feature
Geometry	Point, Line, Polygon	Primary form of spatial attributes

Type classes include Feature Type, Attribute Type, and Time class. All these type classes are used to define a type so that they have the type information and its description as a data member. Time class may have only a time value as a data member. A feature only needs feature name as a data member because detailed feature characteristics are represented by the attribute classes. A feature class is inherited from Feature Type class so that the feature class also has feature type information as a data member.

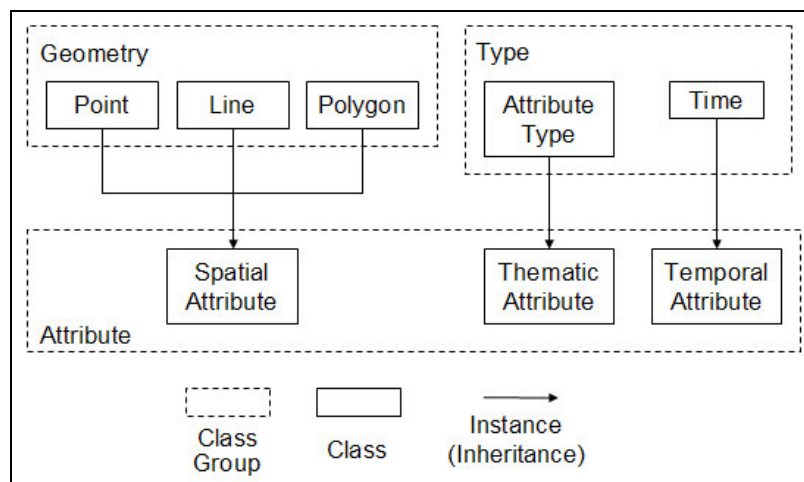


Figure 5. Attribute Classes inherited from Geometry and Type Group

Attribute classes includes Spatial, Thematic, and Temporal Attribute. Attribute classes inherits information from other group classes: Spatial Attribute from geometry classes (Point, Line, and Polygon), Thematic Attribute from Attribute Type, and Temporal Attribute from Time class (Figure 5). Therefore, attribute classes can have the information and methods of related type or geometry classes by inheritance. Specifically, the attribute instances can only have meaning when they are instantiated by the related type classes. The number '4' has no meaning by itself but it means 4-lane when it is instantiated from the attribute type instance 'Lane'. The text 'October 2, 1999' has no meaning by itself but it can be considered as time when it is instantiated from time class.

Relationship classes include Feature Type Relation, Feature Relation, Temporal Relation, and Spatial Relation. The other relationship classes, except the Spatial Relation, have the relation information and related objects as data members. Spatial Relation class will have only one instance of geometric relationship of topology from all geometric attributes. Even though topology has more information than only spatial relationship, topologic structure with geometry is already known and is no different under the feature data model than it is under the layer-based model. The topology construction should be performed every time new feature instances are added in the database. The developed system was focused on handling the feature thematic and temporal relationships, therefore the Spatial Relation class was not implemented in this prototype system.

Geometry classes are directly derived from MapObjects classes (ESRI, 1999). These geometry classes contain not only geometry data but also methods (Table 3). All classes except geometry classes are designed simply for the data storage of objects in the prototype system, so they do not have methods except get and set methods for the encapsulation of their data members. If any action is needed for a class, it can be easily implemented later.

Table 3. Data and Methods for Basic Geometry Classes

Class	Data (Attribute)	Methods (Functions)
Point	X, Y, Z, Measure shapeType	Buffer, Difference, DistanceTo, DistanceToSegment, GetCrossings,Intersect, Union, Xor
Line	Extent, shapeType IsFullyMeasured, Length, Parts	Buffer, Difference, DistanceTo, etCrossings, Intersect, MultiplyMeasures, Offset, OffsetMeasures, ReturnLineEvent, ReturnMeasure, ReturnPointEvent, SetMeasures, SetMeasuresAsLength, UpdateMeasures, Union, Xor
Polygon	Area, Centroid, Extent, Perimeter, Parts, shapeType	Buffer, Difference, DistanceTo, GetCrossings, Intersect, IsPointIn, Offset, Union, Xor

### 3.2 System Design and Implementation

Usery (1996b) used an object-oriented approach to derive a feature as an abstraction of real world phenomena, but the FBGIS was implemented using conventional functional programming methods. Vckovski (1998) used the object-oriented approach for interoperable and distributed processing in GIS. High interoperability means that the complexity should be reduced while keeping high flexibility and expressivity. The complexity reduction is accomplished by encapsulation (information hiding) of properties (data) and operations (functions or methods), which is one of the basic concepts in object-orientation.

The previous studies have shown that the object-oriented approach is a good method for the development of an information system. It leads to quality software which is extendible and reusable (Meyer, 1988). The reuse of software tends to improve the quality because reuse implies testing and debugging.

The conventional functional approach starts with the first question: “what does the system do?” This first question is adequate if the written procedure solves a fixed problem once and for all. The object-oriented design avoids the first question as long as possible, and starts with the decomposition of the categories of objects in the problem domain (Meyer, 1987). The object-

oriented approach may also result in good user interfaces, including both visual and non-visual aspects because if the objects are well chosen, the user can easily form a mental model of the system (Oosterom and Bos, 1989).

The system developed in this research for a feature data model consists of file import/export module, database, and display module (Figure 6). The file import/export module can import a shape file into a feature database and export parts of a feature database into a shape file. The import module is mainly used for populating a feature database. For a database, a feature database has been designed in the next section. A feature database is used for the storage of the data members of the classes in the feature data model (see Figure 2). The display module is used for data visualization, not only for the geometry data, but also for the hierarchy of the feature data structure, thematic attributes, and background images or other file types of vector data as different layers.

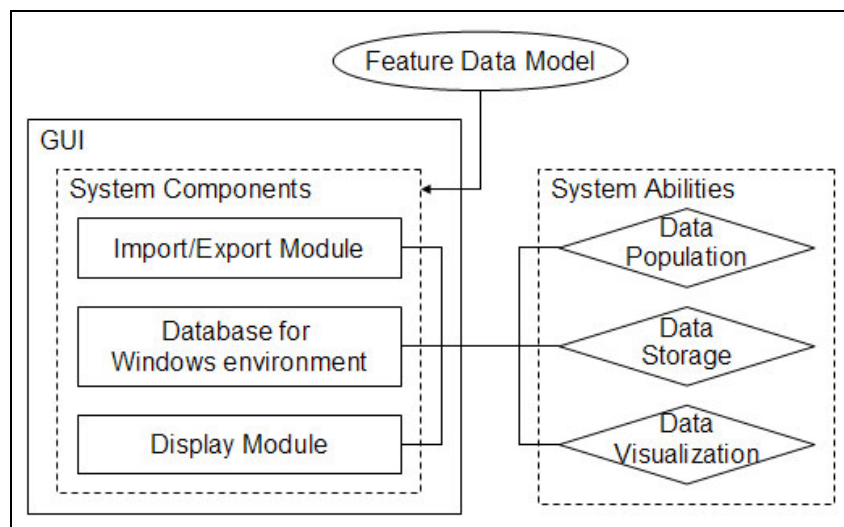


Figure 6. System design from feature data model

The system is implemented using Visual Basic 6.0 and some components in the display module have been customized from MapObjects2 (Ryu, 2000; Stephens, 2000). The main graphical user interface (GUI) has two buttons that are for the import/export and display modules (Figure 7). The import/export module has two components: main import/export interface and sub-option menu

interface (Figure 8). Once the import option is selected in the main import/export window, the sub-option interface will be displayed for the schema construction of the feature database.

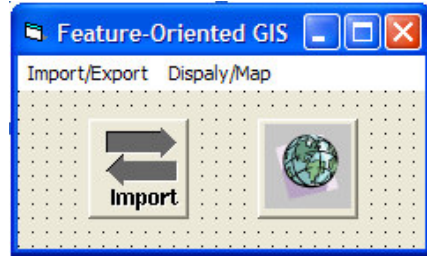


Figure 7. Main GUI of the System for Feature Data Model

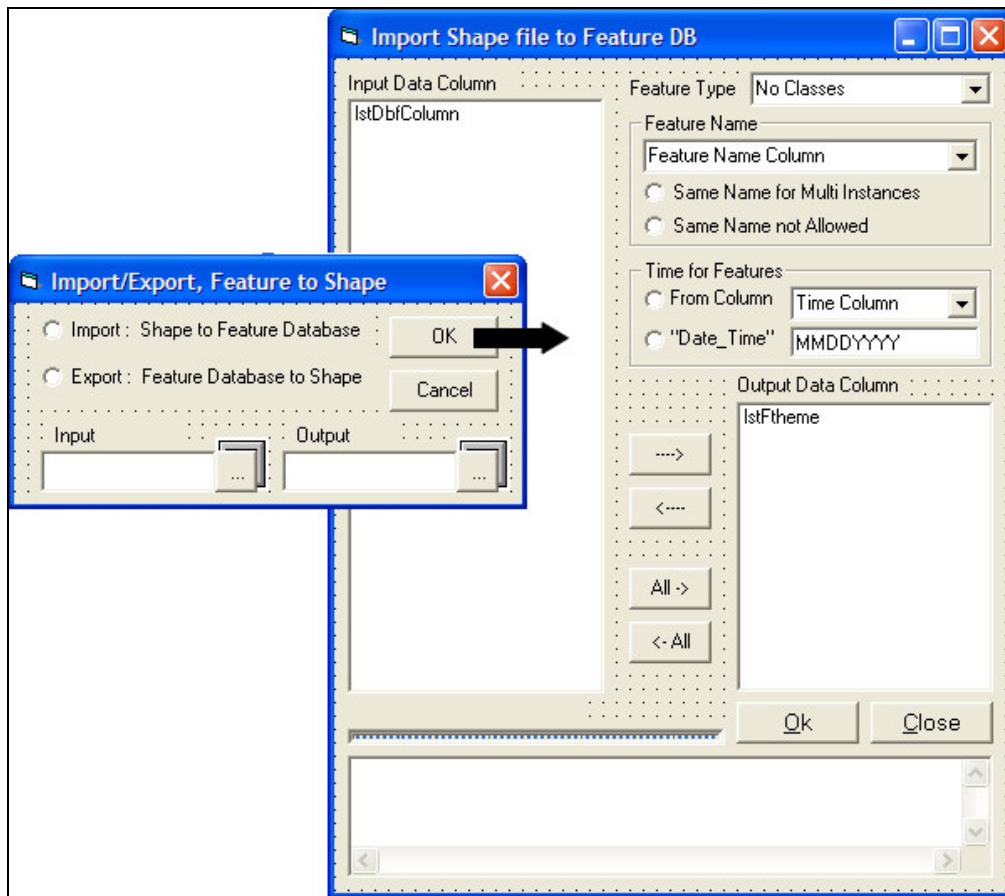


Figure 8. GUI for the Import/export Module

For a schema construction of a feature database in the sub-option window, a feature type can be named from the input filename. The feature name, time, and thematic attribute in the feature data

model can be populated from the selected column of the data file of the input shape file. For a feature name, the schema for a feature database allows the same name with different numbers to be used for multiple instances of a feature. For example, a feature 'Broad Street' can be instantiated as three different instances as 'Broad Street\_1', 'Broad Street\_2', and 'Broad Street\_3'. With this naming scheme, three instances of 'Broad Street' can be handled as either one feature for searching features with a single name or three different features containing different situations such as a certain block for road construction or changing lane numbers through the same road.

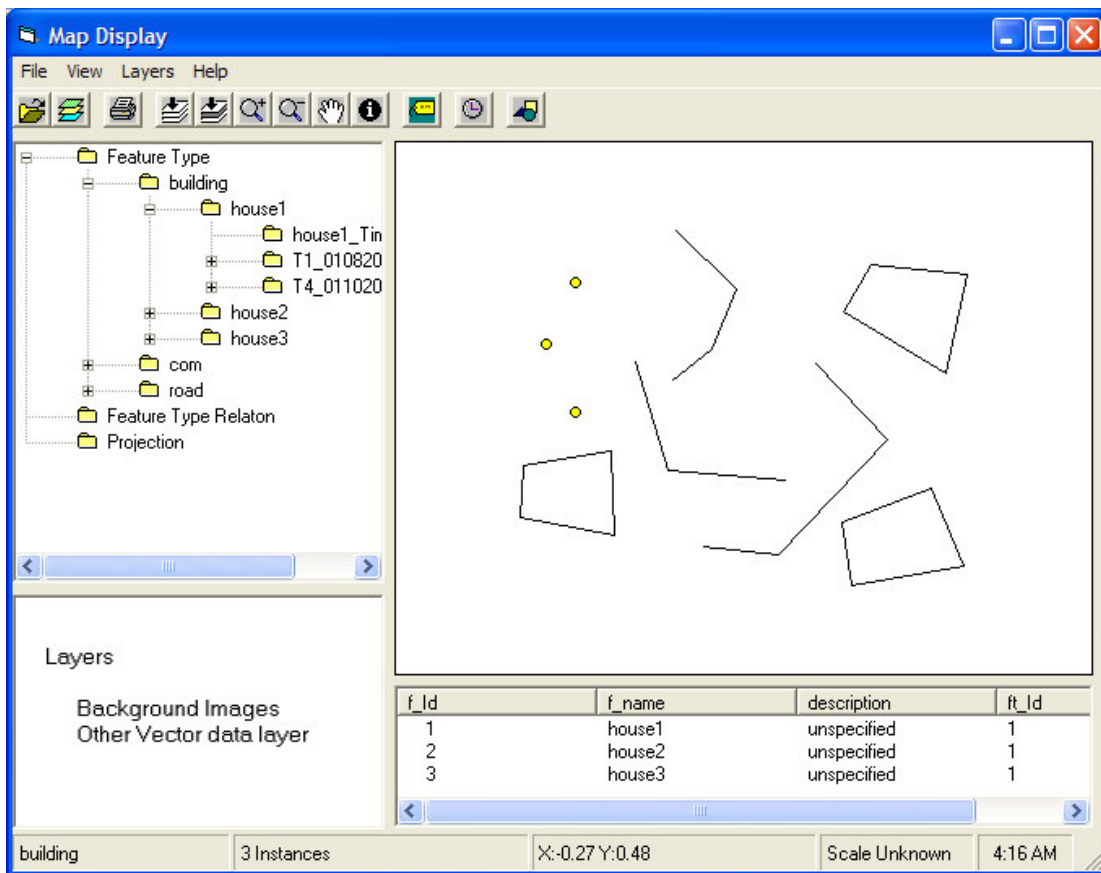


Figure 9 GUI for the Display Module

The display module is composed of four parts: a hierarchical viewer for a feature database, a map area for viewing geometry, an attribute viewer, and a layer manager (Figure 9). The hierarchical viewer uses a tree structure for the representation of the hierarchical structure of the feature data

model. The map area is a canvas where the geometry is displayed. The map area is directly connected with the hierarchical viewer. If an instance of any class is chosen in the hierarchical viewer, then all objects under the chosen object in the hierarchy will be selected and displayed using different colors from those that are not selected.

The Attribute viewer is used to visualize the data members for each object. Like the map area, the attribute viewer is also directly connected with the hierarchical viewer. Once any object is selected, the data member of the object will be shown in the attribute viewer. Finally, the layer manager is used to load background images or vector data layers that are not in a feature database, but are used for the comparison or overlay analysis with the features in the feature database. For background images, various kinds of image formats can be supported such as ERDAS Imagine files (\*.img), windows metafile (\*.wmf), standard image data such as bitmap (\*.bmp), gif (\*.gif), jpeg (\*.jpg), tiff (\*.tif), and others.

### **3.3 Feature Database**

For data storage, an object-oriented database may have better performance for applications in using an object-oriented programming language than a relational database (Graham, 2001). If a programming language is based on object-oriented programming environments, the data model is harmonious with the object representation in an object-oriented database. On the other hand, considerable data transformation, construction, and decomposition may occur in the data transfer process into a relational database (Milne et al., 1993).

Though object-oriented databases have a number of advantages over relational databases, there are still unsolved problems concerning non-procedural query languages and persistent objects. There are also no explicit standard query methods such as SQL for object-oriented database. In object-orientation, objects themselves are the storage for the data members and the data members of an object may be different from one application to another. The objects scheme for an object-

oriented database should be modified from one application to another in order to fully contain the data members of the objects. To avoid these problems, it is possible to implement an object-oriented paradigm into a relational database name as an object-relational database such as Oracle8.x (Oracle Corporation, 1998). In this way, the object-oriented paradigm can be used for system design and implementation and then the objects can be entered into tabular structures of the relational database as a physical storage of objects. Also, entered objects in tables can be easily retrieved by SQL in the relational database.

In this study, Microsoft Access database has been used for the physical storage of the objects of the feature data model (Figure 10). The data members of each class in Table 3 will be entered into a table. The tables in a feature database are related to each other for populating the data members of the classes when the feature database is uploaded into the system based on feature data model. The entity-relation between tables in a feature database is only used for populating objects.

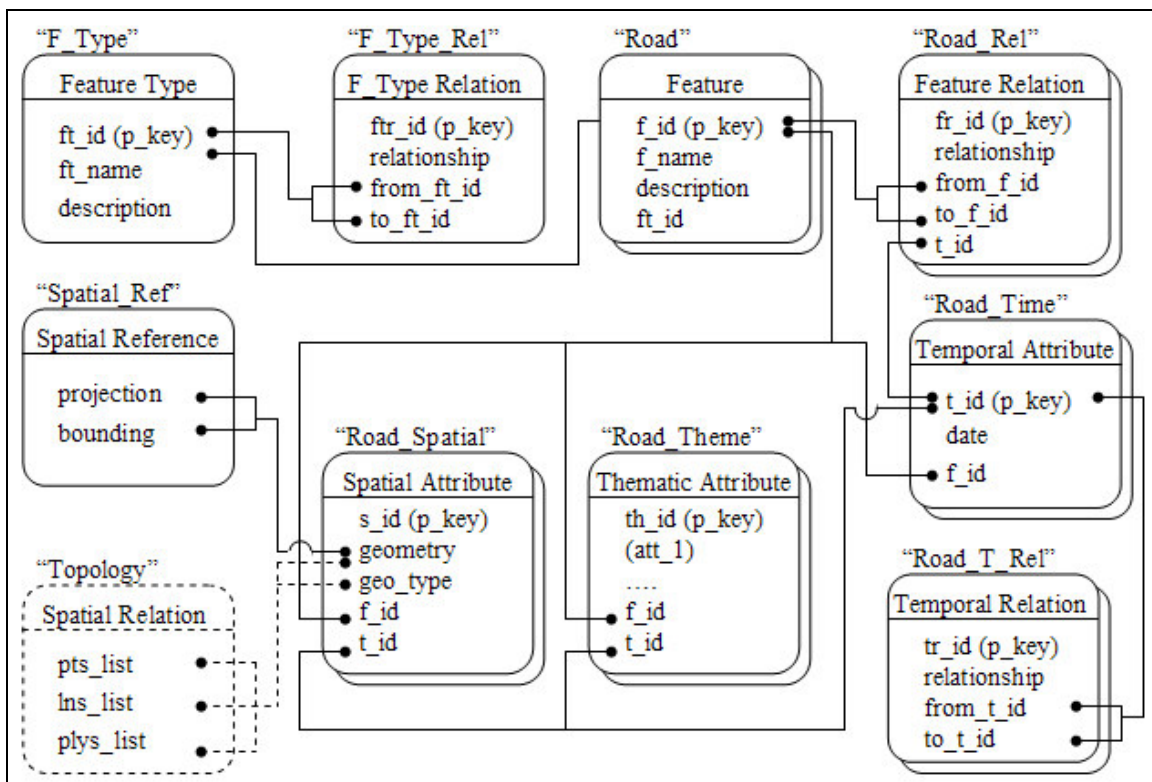


Figure 10. Table Structure for Classes in Feature Database

Once a shape file is imported into a new feature database, Feature Type, Feature Type Relation, and Spatial Reference classes will have an instance so that the corresponding table for each class will be created and populated. A feature table for Feature class will be named after the feature type instance such as 'Road' and 'Building'. All attribute and relationship classes, except Feature Type Relation class, will also be named after feature type name ('Road') with their own identifier ("Theme") such as 'Road\_Theme' for thematic attribute of the 'Road' type features (Figure 10). The instances of Attribute Type class will be used automatically for populating the column names of the thematic attribute table. The instances of Time class will be used for populating instances of the 'date' column in the temporal attribute table.

If a shape file is imported into an existing feature database, the feature type and feature type relationship tables are populated with a new instance and other tables will be created. The imported shape file should be under the same spatial reference information of the spatial reference table in the existing feature database.

#### 4. Multi-Temporal Representation of a Coastal Area

The study site is a part of the U.S. marine base, Camp Lejeune, which is located along the North Carolina coast. The study site is very small (about 350m X 250m) to visualize individual buildings (Figure 11).

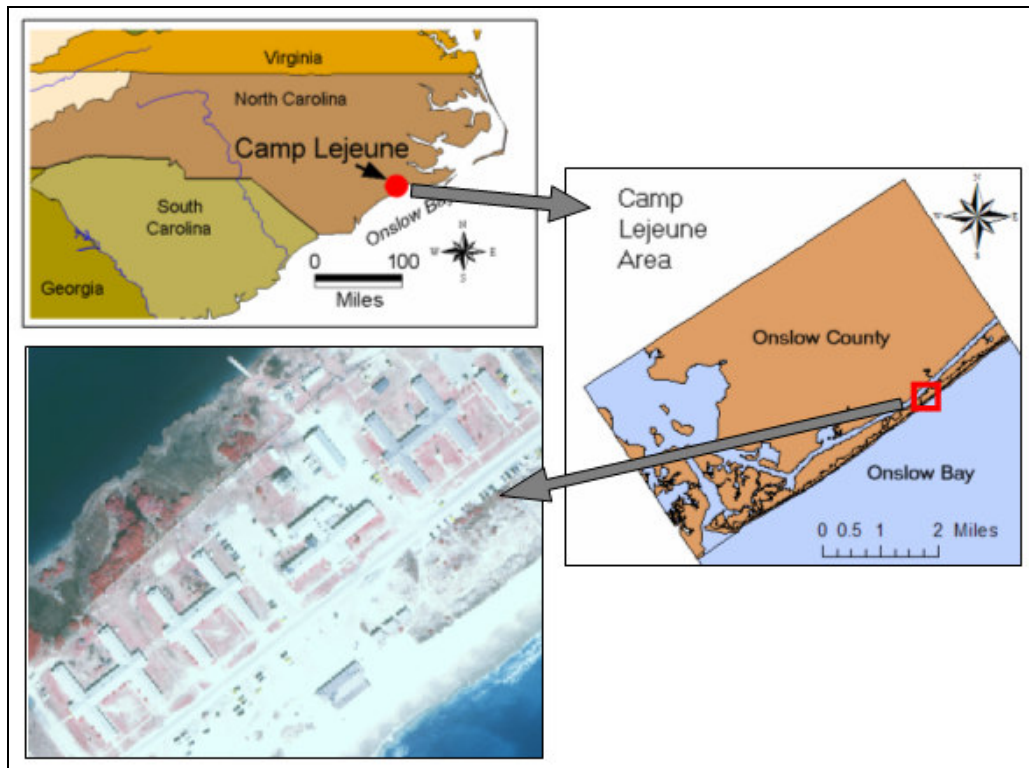


Figure 11. Study Area

On the study site, two different remotely sensed images were acquired less than one year apart: color infrared airphoto (CIR) (09/1999) and Ikonos data (05/04/2004). In this application, buildings are mapped and visualized, so very high resolution images were required. The CIR airphoto image was scanned at 1.2 m ground resolution. The Ikonos image has 1 m ground resolution in the panchromatic band and 4 m in the multispectral bands. For enhancing visual interpretation ability, the Ikonos multispectral image has been pan-sharpened so that the resulting pan-sharpened image has a 1 m ground resolution.



Figure 12. Floyd Hurricane Paths on the Study Area on Sept. 16 in 1999

Hurricane Floyd occurred over the study area on September 16, 1999 (Figure 12). When Floyd struck the study area, it was a level two hurricane with the wind speeds of 105 mph. A level two hurricane may cause some damage to roofing material, doors, and windows, and considerable damage to mobile homes and small craft in unprotected anchorages (NOAA, 1999).

In the study area, data extracted from the CIR image in 1999 and the Ikonos image in 2000 shows the removal of some features through time by Hurricane Floyd. Also, hypothetical data for 1995 has been created by duplicating 1999 data and removing one building from the duplicated data in order to visualize the addition of feature. With these three input data, one building was built between 1995 and 1999 and eight buildings was destroyed between 1999 and 2000 (Figure 13).

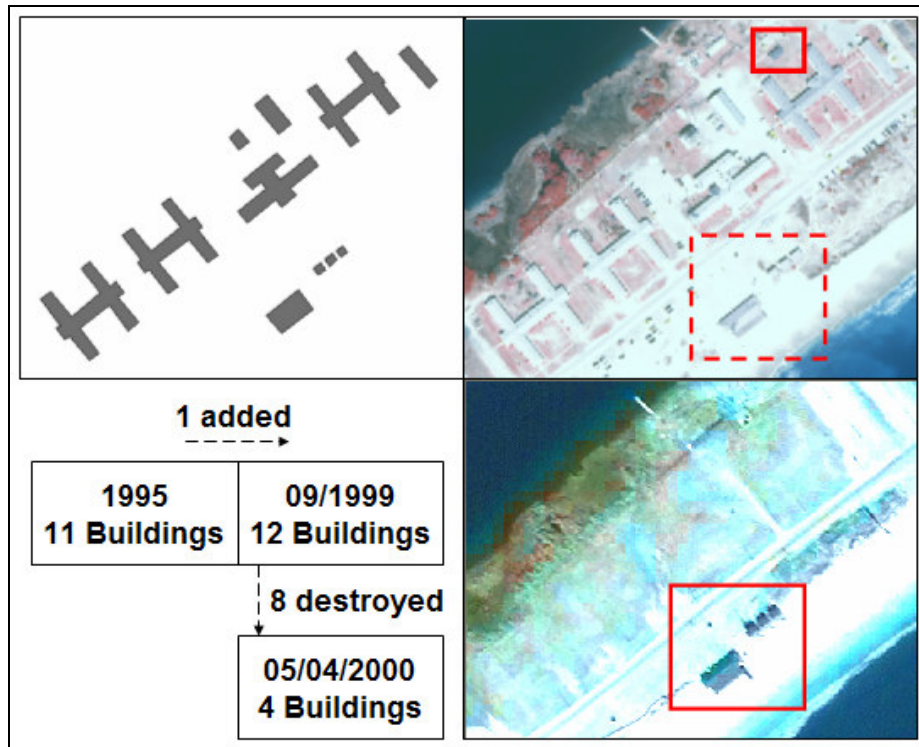


Figure 13. Source Data for Building Features with Different Time

During the population of features from different times, if a feature exists already in the feature database, the attributes and relationships of the feature refer to the existing feature based on the different times. A temporal relationship of the feature with different time can be explained by the ‘Was\_a’ relationship: ‘A feature in 1999 was a feature in 1995’ If the feature doesn’t exist previously, then the temporal relationship of the feature will have the ‘built’ relationship by itself without any reference to another feature: ‘A feature has been built in 1995’. Figure 14 shows how building features are populated in the feature database from three different temporal datasets.

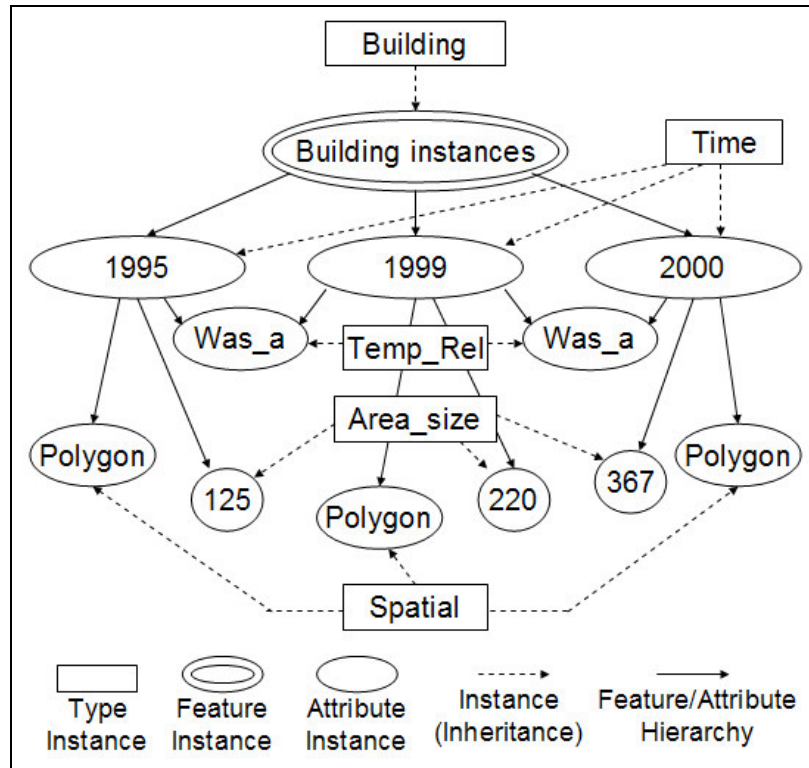


Figure 14. Populating Features on Three Different Sources

For the visualization of building features, the feature display module in the developed feature system has been used (Figure 15). The tree viewer in the display module shows the hierarchical structure of building features. Each feature has its time stamps and each time stamp has the feature relationship, spatial attributes, and thematic attributes. When a node in the tree structure is selected, the data members of the node will be shown in the attribute viewer (Right bottom section in Figure 15).

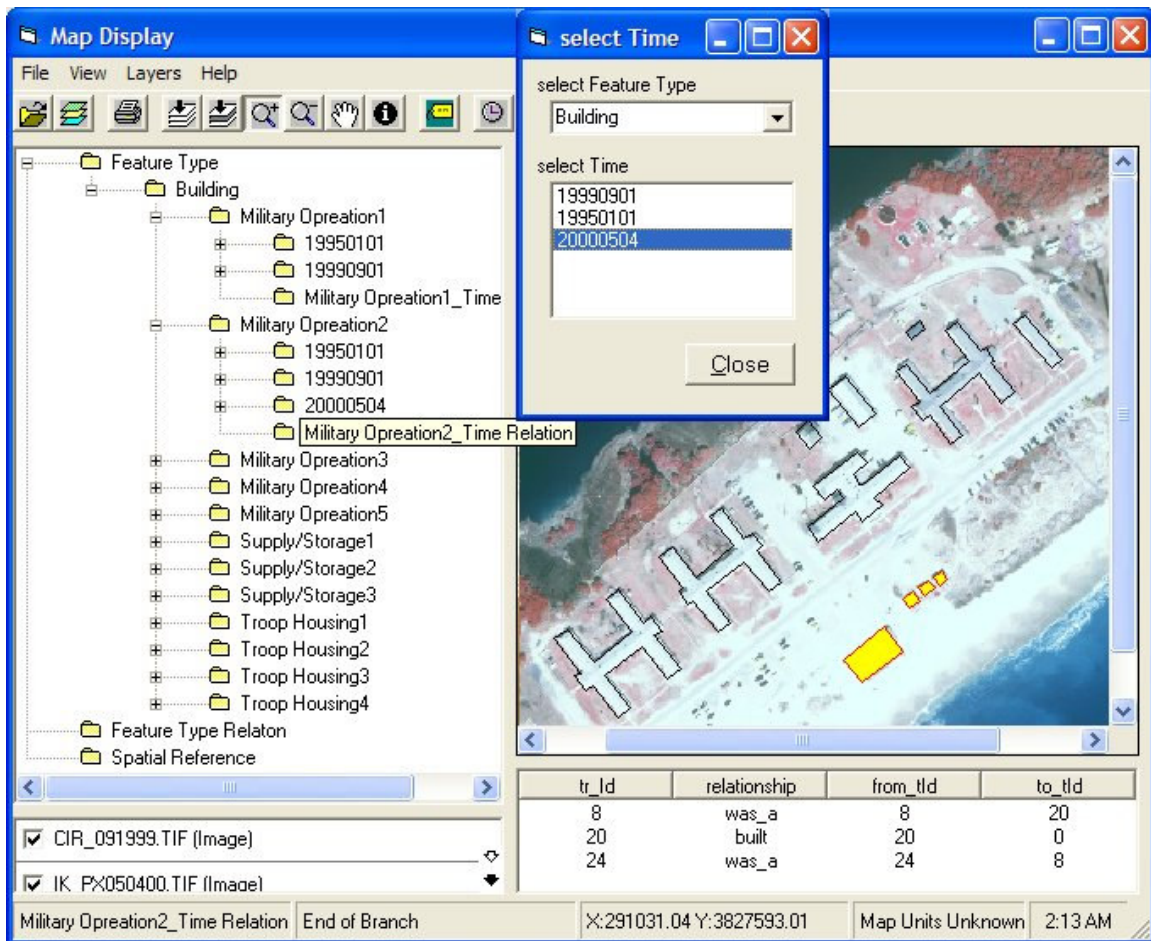


Figure 15. Feature Query based on the Time

The temporal differences of building features can be easily retrieved using the time selection tool ('select Time' window in Figure 15) that is loaded by selecting the clock icon on the menu bar. The geometry viewer in Figure 15 shows some building features selected when the specific time was chosen in the time selection tool. If another time is chosen in the time selection tool, then the features with the selected time will be shown. Therefore, features at the specific time spot can be easily and effectively queried and differentiated from the features at other time spots. Also, the time stamps of each feature can be shown easily under the feature instance node in the tree viewer. In this study, there were 12 building features in 1999 but only 4 buildings were left in 2000 after the hurricane that occurred during the time gap between the two datasets (Figure 16).

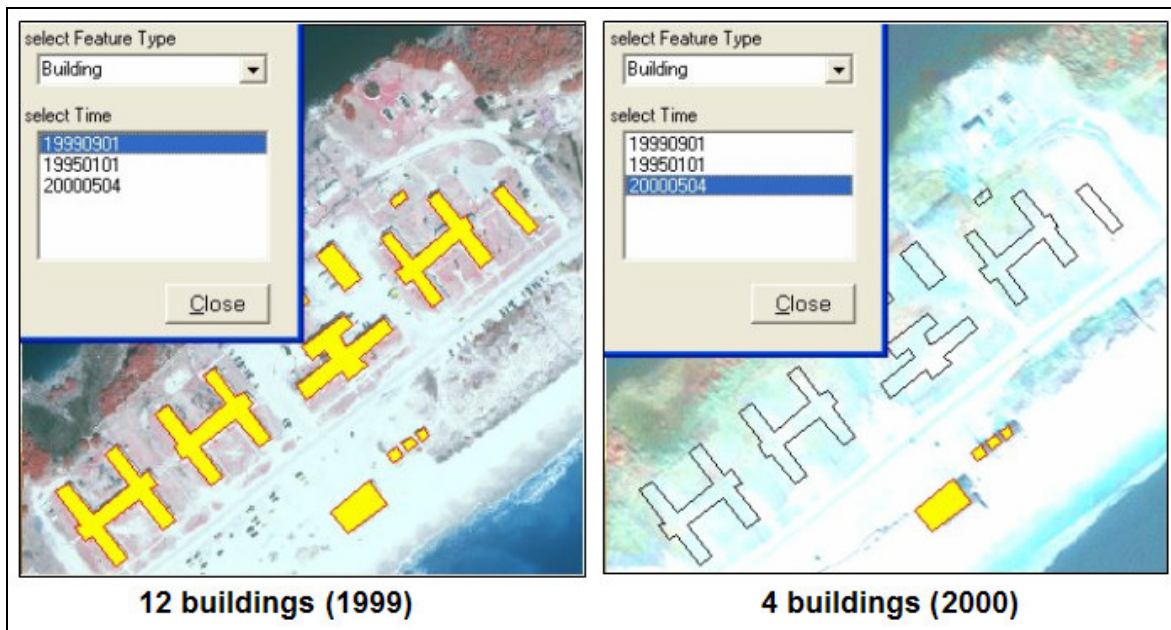


Figure 16. Temporal Differences of Buildings between 1998 and 1999

## 5. Discussion

The new feature data model derived in this research is modified from the existing feature-based data model and different from the existing layer-based geographic model in the following ways. First, the layer-based data model can only contain one geometry type such as point, line, or polygon in each layer. This mechanism has a limitation for the map representation. The feature representation on a map is always based on the scale. Therefore, the same feature types for different feature instances can be different geometric types. For building feature types, very small buildings will be mapped using point geometry rather than polygon geometry because of the difficulty for mapping it as polygon from the small scale source image data such as airphotos and satellite images. On the other hand, some large buildings should be mapped as polygon geometry rather than point geometry because of the size of the buildings. These two different sizes of buildings make it difficult to be contained in the same layer, even though they are in the same feature type. However, the feature data model has no limitation for the geometry type for each feature database. Also, for the linear

feature, the highway can be saved as polygon and the central line for the highway can be saved as line under the same feature type in the feature data model.

Second, the layer-based data model can represent implicitly the thematic relationship among features in the same layer, but cannot describe the relationships among features in different layers. This makes disjunctive thematic relations. The disjunctive thematic relations can be solved by the hierarchical representation. The feature data model can represent any kind of thematic relationship among features because the feature data model uses basically the hierarchical representation from feature type to detail each feature's thematic attributes.

Third, in the layer-based data model, a feature with multi-temporal situations should be represented by several layers because one feature can only have one time attribute in one layer. Once a feature has been mapped into a different layer, the feature in each layer is not conceptually the same feature. However, in the feature data model, the feature can have multi-temporal situations. The temporal situations are from the same feature instance node so they reference exactly the same feature. Also, the feature can have different situation information according to the reference time because the spatial and thematic attribute and relations of the feature reference the feature instance through the specific time node.

Even though the feature data model can effectively represent various relationships, there may be some disadvantages of the feature data model. First, it may be difficult to identify all relationships of a feature. Second, the identification of relationships of features requires manual effort because there is currently no automatic method for it. Finally, there is no standard term in the relationship representation of features so the same relationship may be represented by different terms. For example, 'Is A' relationship may be represented by 'Is A', 'Is-A', 'Is\_A', and so on. This inconsistency makes the implementation of query procedures using the relationships difficult.

## 6. Conclusion

In this study, a feature data model was derived from the feature concept, abstraction method of SDTS, and the key concepts of object-orientation. Geographic features can be represented effectively using three aspects; spatial, thematic, and temporal dimension. For each dimension, attributes and relationships are included in feature data model.

With the feature data model, a prototype system was developed. Classes in the data model were designed and used to building the system. The developed system has three main parts: import/export module, database for feature storage, and display module. Specifically, the tree viewer in the display module can effectively visualize the data structure of features because the feature data model uses hierarchical structures. For the storage of features, the feature database was schematically designed and implemented using table structure in the relational database. Using import/export module, the existing spatial data such as shape file can be easily imported into the feature database.

The developed system was applied for the visualization of geographic features with the multi-temporal situations. The changes of building on the beach area were effectively represented with one hypothetical situation and two real situations. Background image data were used to show the real situation on the study area.

Conclusively, the introduced feature data model has advantages over the traditional layer-based model in GIS even though it may be difficult to identify all relationships of a feature in a feature data model. For example, multiple types of geometry can be used for the same feature class. Various types of feature relationships including spatial, thematic, and temporal dimension can be represented. Multi-temporal situations of a phenomenon can be saved in one feature.

## Reference

- Booch, G., 1994, *OBJECT-ORIENTED ANALYSIS AND DESIGN with Applications*, 2<sup>nd</sup> Edition, The Benjamin/Cummings Publishing Company, Inc., RedWood, California, 589 p.
- Borgida, A., 1988, "Modeling class hierarchies with contradictions", *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data In SIGMOD Record*, Vol. 17, No. 2, pp.434-443.
- Chen, P.P.S., 1976, "The entity-relationship model-toward a unified view of data", *Association for Computing Machinery Transactions on Database Systems*, Vol. 1, No. 1, pp.9-36.
- Codd, E.F., 1970, "A relational model of data for large shared data banks", *Communications of the Association for Computing Machinery*, Vol. 13, No. 6, pp.377-387.
- Defense Mapping Agency, 1987, *Feature Attribute Coding Standard (FACS)*, internal report, Reston, Virginia: U.S. Defense Mapping Agency.
- Digital Geographic Information Working Group, 2000, "Part 4. Feature and Attribute Coding Catalogue (FACC)", *The Digital Geographic Information Exchange Standard (DIGEST)*, Edition 2.1, [http://www.digest.org/html/DIGEST\\_2-1\\_Part4.pdf](http://www.digest.org/html/DIGEST_2-1_Part4.pdf), 28 p.
- Egenhofer, M. J. and J.R. Herring, 1991, "High-level spatial data structures for GIS", In Maguire, D.J., M.F. Goodchild, and D.W. Rhind, (Eds.), *Geographical Information Systems: Principles and Applications, Volume 1*, Longman Scientific Publications, London, pp.227-237.
- Environmental Systems Research Institute (ESRI), 1999. *ESRI MapObjects™ Version 2.0*, Environmental Systems Research Institute, Inc, Redlands, California, 26 p.
- Gatrell, A.C., 1983, *Distance and Space: A Geographical Perspective*, Clarendon Press, Oxford, 185 p.
- Gatrell, A.C., 1991, "Concepts of Space and Geographical Data", In Maguire, D.J., M.F. Goodchild, and D.W. Rhind, (Eds.), *Geographical Information Systems: Principles and Applications, Volume 1*, Longman Scientific Publications, London, pp.119-134.
- Graham, I., 2001, *OBJECT-ORIENTED METHODS: Principles & Practice*, 3<sup>rd</sup> Edition, Pearson Education, London, 864 p.
- Guptill, S.C., K.J. Boyko, M.A. Domaratz, R.G. Fegeas, H.J. Rossmeissl, and E.L. Usery, 1990, *An Enhanced Digital Line Graph Design*, USGS Circular 1048, Reston, Virginia: USGS, 20 p.
- James, P.E., and C.F. Jones, 1954, *American Geography: Inventory and Prospect*, Syracuse University Press, New York, 598 p.
- Lakoff, G., 1987, *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*, University of Chicago Press, Chicago, 614 p.

- McMaster, R.B., 1989, "Numerical Generalization in Cartography", *Cartographica*, Vol. 26, No. 1, pp.1-6.
- Meyer, B., 1987, "Reusability: The Case for Object-Oriented Design", *IEEE Software*, Vol. 4, No. 2, pp.50-64.
- Meyer, B., 1988, *Object-Oriented Software Construction*, Englewood Cliffs, NJ, 534 p.
- Milne, P., S. Milton, and J.L. Smith, 1993, "Geographical object-oriented databases-a case study", *International Journal of Geographic Information Systems*, Vol. 7, No. 1, pp.39-55.
- National Oceanic and Atmospheric Administration (NOAA), 1999, "Hurricane Basics", <http://hurricane.noaa.gov/pdf/hurricanebook.pdf>, 19 p.
- Oosterom, P.V., and J.V.D. Bos, 1989, "An object-oriented approach to the design of geographic information systems", *Comput & Graphics*, Vol. 13, No. 4, pp.409-418.
- Oracle Corporation, 1998, *Oracle8: Database Administration*, Vol I, II, and III, Redwood Shores, CA.
- Robinson, A.H., R.D. Sale, J.L. Morrison, and P.H. Muehrcke, 1984, *Elements of Cartography*, 5<sup>th</sup> Ed., John Wiley & Sons, New York, 544 p.
- Rosch, E., 1978, "Principles of Categorization", In Rosch, E. and B.B. Lloyd, (Eds.), *Cognition and Categorization*, Halstead Press, New York, pp.27-48.
- Ryu, J., 1999. *Korean Visual Basic 6.0 using Bible*, Youngin.com Publishing Co., Seoul, Republic of Korea, 722 p.
- Stephens, R., 2000. *Visual Basic Graphics Programming*, 2<sup>nd</sup> Ed, John Wiley & Sons, Inc., New York, 712 p.
- Tang, A.Y., T.M. Adams, E.L. Usery, 1996, "A spatial data model design for feature-based geographical information systems", *International Journal of Geographic Information Systems*, Vol. 10, No. 5, pp.643-659.
- Trainor, T.F., 1990, "Fully automated cartography: a major transition at the Census Bureau", *Cartography and Geographic Information Systems*, Vol. 17, No. 1, pp.27-38.
- Usery, E.L., 1993, "Category Theory and the Structure of Features in Geographic Information Systems", *Cartography and Geographic Information Systems*, Vol. 20, No. 1, pp.5-12.
- Usery, E.L., 1994a, "Implementation Constructs for Raster Features", *Proceedings of ASPRS/ACSM Annual Convention*, Reno, Nevada, pp.661-670.
- Usery, E.L., 1994b, "Display of geographic features from multiple image and map databases", *Proceedings, ISPRS Commission IV Symposium on Mapping and Geographic Information Systems, International Archives of Photogrammetry and Remote Sensing*, Volume 30, Part 4, Athens, Georgia, pp.2-9.

- Usery, E.L., 1996a, "A Conceptual Framework and Fuzzy Set Implementation for Geographic Features", In Burrough, P.A. and A.U. Frank, (Eds.), *Geographic Objects with Indeterminate Boundaries*, Taylor and Francis, London, pp.71-85.
- Usery, E.L., 1996b, "A Feature-Based Geographic Information System Model", *Photogrammetric Engineering & Remote Sensing*, Vol. 62, No. 7, pp.833-838.
- Usery, E.L., G. Timson and M. Coletti. 2002. "A Multidimensional Geographic Feature System", *GIScience 2002: The Second International Conference on Geographic Information Science*, National Atmospheric and Oceanic Administration. pp.279-282.
- U.S. Geological Survey, 1997, *Spatial Data Transfer Standard (SDTS) – Part 1, Logical Specifications*, American National Standards Institute, New York.
- Vckovski, A., 1998, *Interoperable and Distributed Processing in GIS*, Taylor & Francis, London, 230 p.
- Worboys, M., H.M. Hearnshaw, and D.J. Maguire, 1990, "Object-oriented data modeling for spatial databases", *International Journal of Geographic Information Systems*, Vol. 4, No. 4, pp.369-383.