Towards 4-dimensional Geographic Information Systems

Seong-Ho Lee and Jong-Hyun Park
LBS Research Team, Electronics and Telecommunications Research Institute
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea
{sholee, jhp}@etri.re.kr

Abstract: To overcome the limitation that traditional GISs lose much information for the real world, 4-dimensional GIS has the additional reference systems including object's height and temporal dimension. This paper describes the 4-dimensional geometric object model and components. The prototype for 4-dimensional GIS consists of the data provider, manager, and renderer components. We show the virtual city that its database contains topographic maps, buildings, roads and temporal history data. This provides spatial, temporal operations and analysis functions.

Keywords: 4-dimensional GIS, spatio-temporal data.

1. Introduction

According as the past thirty years have seen a great deal of research devoted to spatial[1] and temporal database[2], the users have required to know much information. This motivates that it combines spatiotemporal database with computer graphics, wireless internet and high-speed network technologies. For example, the users using mobile internet equipments want to display the scene that they get a view of downtown in the screen. However, conventional GISs providing 2-dimensional spatial information meet to the limitation of information service. The upcoming GISs develop into the level serving much information, such as the real-time 3-dimensional location and route search of moving objects.

There are the needs of an extended geometry object model, diverse functions as indexing, operations, and analyses for spatio-temporal data, and visualization in order to implement 4-dimensional GISs. In this paper we show a 4-dimensional GIS that have spatiotemporal operations and analysis functions making decisions on new road construction and record management of buildings. In section 2, we present a new geometry class model and WKB streams. Section 3 describes components composed of the prototype, and we give the results of experiment such as relational operations and analyses in section 4. After that, section 4 represents conclusions.

2. Background

OGC shows the geometry class hierarchy[3]. Geometry is the root class of the hierarchy, which the root has subclasses as Point, Curve, Surface, and GeometryCollection. And zero, one, and two-dimensional collection classes corresponding to above classes are MultiPoint, MultiLineString, and MultiPolygon respectively. In other words, the subclasses of the Geometry defined by OGC are limited in geometry classes being on 2-dimensional spatial reference system.

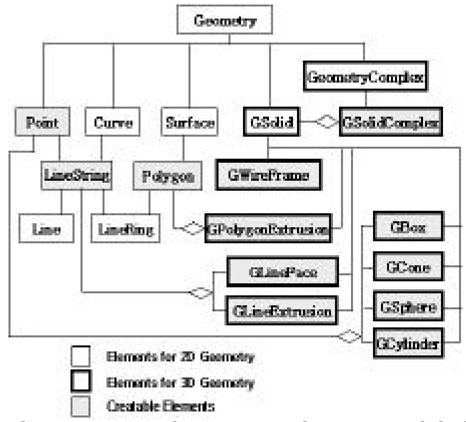

Fig. 1. The Geometry class supporting 3D spatial objects.

As shown in Fig.1, the geometry elements including Box, Cone, Sphere, Cylinder, and WireFrame were added to the object model in order to handle the complicated 3-dimensional geometry objects. These object collections supporting complex features are the subclass of Geometry(GSolid). The geometry class architecture is modified in [4] to deal with 3-dimensional spatial objects.

In [3], the Well-Known Binary Representing for Geometry(WKBGeometry) provides a portable representation of a Geometry value as a contiguous stream. It permits Geometry values to be exchanged between an ODBC client and an SQL database in binary form. We show a representation of the WKB stream for 3D features including WKBPolyExtrusion, WKBLineFace, WKBLineExt, and WKBDEM structures in Fig.2. They are the geometry types for buildings, roads, underground pipelines, and DEM data in 3-dimensional spatio-temporal databases respectively.

```
struct   WKBPolyExtrusion {
      byte          byteOrder;
      uint32        wkbType;
      float         upHgt;
      uint32        upStory;
      float         dnHgt;
      uint32        dnStory;
      float         hgtOfOneStory;
      uint32        story;
      bool          isIfFloor;
      float         base;          //DEM height
      uint32        numPoints;
      WKSPoint*     points;
};
struct   WKBLineFace {
      byte          byteOrder;
      uint32        wkbType;
      uint32        numLane;       // number of lane
      float         widthRoad;
      bool          isItOneWay;
      uint32        numPoints;
      WKSPoint*     points;
};
struct   WKBLineExt {
      byte          byteOrder;
      uint32        wkbType;
      float         radius;
      float         bDepth;        // burial depth
      uint32        numPoints;
      WKSPoint*     points;
};
```

```
struct   WKBDEM {
      byte          byteOrder;
      uint32        wkbType;
      uint32        numX;          // number of X
      uint32        numY;          // number of Y
      WKSPoint      upperLeftCoord;
      WKSPoint      lowerRightCoord;
      uint32        numPoints;
      WKSPoint*     points;
}

struct  WKBGeometry {
     int         id;
     char        name[128];
     long        time;
     int         wkbType;
     union {
        WKBPoint                  point;
        WKBLineString             linestring;
        WKBPolygon                polygon;
        WKBGeometryCollection     collection;
        WKBMultiPoint             mpoint;
        WKBMultiLineString        mlinestring;
        WKBMultiPolygon           mpolygon;
        WKBPolyExt                polyext;
        WKBLineFace               lineface;
        WKBLineExt                lineext;
        WKBDEM                    dem;
     } geometry;
}
```

Fig. 2. A representation of the WKB stream for 3D features.

3. Components of 4D GIS

We designed the architecture providing 4-dimensional spatiotemporal data as shown in Fig.3. This consists of the data provider, access object, data manager, data renderer components.

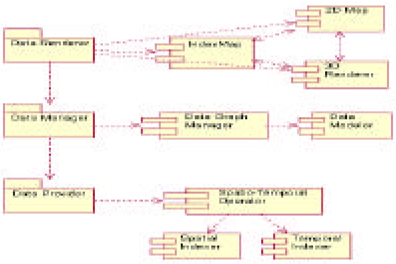

Fig. 3. The component architecture of 4D GIS.

1) Data Provider and Access Component

The data provider complies with OGC's proposal of simple feature specifications. And we extended the specifications so that the provider could deal with and provide spatial and temporal data to consumers. It retrieves 3-dimensional and temporal data through OLE DB interfaces. Moreover, this executes the relational operations such as equals, touches, contains, and so on presented in [3]. These are extended to apply to 3- or 4-dimensional data.

The data access component generates indexes on spatial and temporal data, and composes diverse query statements. The data delivered from the above data provider is processed and compiled in the forms of that manager component needs.

Fig.4 shows relationship between data provider and access component, and relative interfaces. The data access component consists of several COM objects. A Connection object provides the access to the data source of database systems through OLE DB data provider. A featureset and featuresets store the schema information of feature tables and manage those sets respectively. A recordset object manages record instances and information of columns included in database tables. The retrieval function is actually implemented in this object. And the recordset object makes the result into records to return and creates columns organizing the records. A field and fields offer information of every column and save attributes such as the name, type, length, and precision.

2) Data Manager Component

The data manager receives 2-, 3-, or 4-dimensional spatial data(WKB; Well-Know Binary) from a data provider, then keeps WKB in graph structure(DataGraph), runs modeling works and spatial analyses. The DataGraph is a linked-list structure and it handles spatial information by features.

And the geographic information saved in the DataGraph has the minimum data in order to promote management performance.

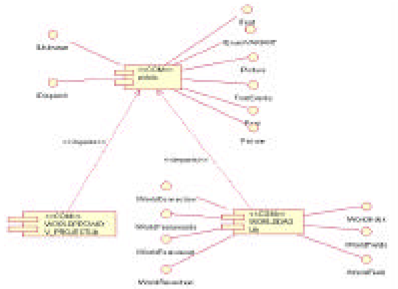

Fig. 4 Data provider and access component.

We use two modeling technique: static modeling and dynamic modeling. The former creates the geometry feature model by modeling rule and the latter is the process that builds the scene graph to use the features created by static modeling according to LOD(Level-Of-Detail). It needs to classify features of a class into various geographic elements to do dynamic modeling.

The classification assumes the form of a model library. This model library has appearance information as well as geographic shape of objects. Therefore, users can apply diverse rules to do concrete modeling by representing the rules in a rule parameter descriptor. As shown in Fig.5, a model library and rule parameter descriptor can be extended and modified by other users later on by encoding XML documents.

3) Data Renderer Components

Data renderer components to service visualization of spatial data consist of three components. First, the index map component shows the index map for overall spatial data. In general, index maps provide summarized that rather than detailed information of all spatial objects. In this component the geographic primitive is mainly defined by a polygon.

Fig. 5. Model library and rule parameter descriptor.

Second, the 2D map component displays geographic information selected on the index map by user input. The users can utilize diverse interactive functions as zoom-in/out, panning, picking, tool-tip, undo, initial view, navigation path, and so on. In the 2D map, the available primitives are point, line, lineface, rectangle, and polygon. Moreover, the 2D map component is linked with the index map and 3D map component mutually. Finally, the renderer component visualizes geographic and 3D additional information in a database, modeling results received by a modeler, topographical data using DEM, texture mapping of satellite imagery, results of spatio-temporal analyses, and navigation.

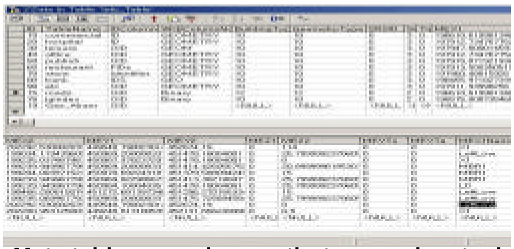

Fig. 6. Meta table managing spatio-temporal meta data.

4. Experiment on the Prototype of 4D GIS

1) Spatio-Temporal Database

In this paper, we designed the schema of meta table that manages metadata including object identifier, map extent, name, and geometry. Fig.6 shows the schema table. This table saves meta data for all feature tables such as table names, column names of identifiers and WKBs, MBRs, and many other things.

2) Composition of GUI

As described in chapter 3, the prototype has three renderer components. In Fig.7, the index map displays summarized spatial map with 2D scene, the 2D map shows the outlines of buildings and roads. And the 3D frame shows terrain covered by satellite imagery, the modeled 3D builds with rule parameter descriptor, and roads.

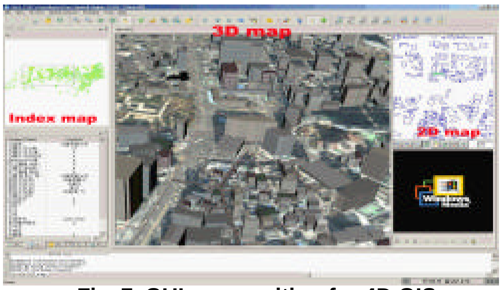

Fig. 7. GUI composition for 4D GIS.

3) Relational Operations of Spatio-Temporal Data

We implemented the diverse relational operations: equals, contains, touches, within, disjoint, crosses, overlaps, and intersects. The followings are the results of some operations. First, user fixes the window to query by mouse drag on 2D map, and key in a height, and then chooses an operator.

condition 1. operators : contains, overlaps, intersects

condition 2. query window is

(198178.0, 450952.0) ~(199051.0, 451598.0)

Fig.8 displays the results of contain, overlap, and intersect operations respectively and Fig.9 shows those of the temporal and spatio-temporal operation, that is, the history management of fire.

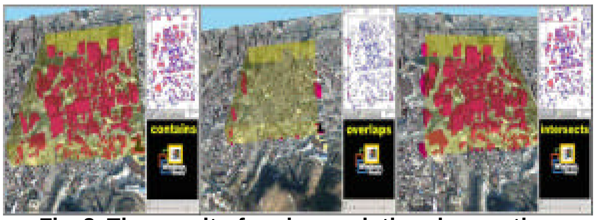

Fig. 8. The result of various relational operations.

The following query is 'Find out all buildings that have been fired from January 1^{st}, 2002 to November 17^{th}, 2002.

condition 3. operators : intersects

condition 4. spatial query window coordination is

(197717.0, 451118.0) ~ (198466.0, 452050.0)

condition 5. temporal lines

start time : (2002-01-01), end time : (2002-11-17)

time granularity : 1 month

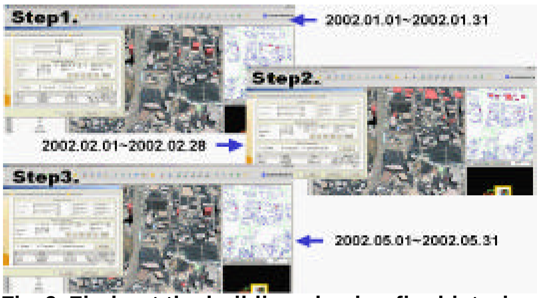

Fig. 9. Find out the buildings having fire histories.

5. Conclusions

In this paper we designed and implemented a 4-dimensional GIS. It is composed of the data provider component having spatial and temporal index module, the data manager component that runs analyses and modeling works with rule parameter descriptor, and the renderer component that displays spatio-temporal data in index, 2D, and 3D view. This prototype provides various functions, which are relational operations, 3D buffering, history management, and viewing a virtual city. This will be utilized in the fields such as a road construction and city planning.

References

[1] O. Guenther, and A. Buchmann, "Research Issues in Spatial Databases", ACM SIGMOD Record, Vol.19, No.4, Dec., 1990.

[2] V. Tsotras, and A. Kumar, "Temporal Database Bibliography Update", ACM SIGMOD Record, Vol.25, No.1, Mar., 1996.

[3] Open GIS Consortium, Inc., OpenGIS Simple Features Specification for SQL, 1999.

[4] K.H. Kim, "Design and Implementation of 3D GIS Components", Proceedings of ASPRS2001, 2001.