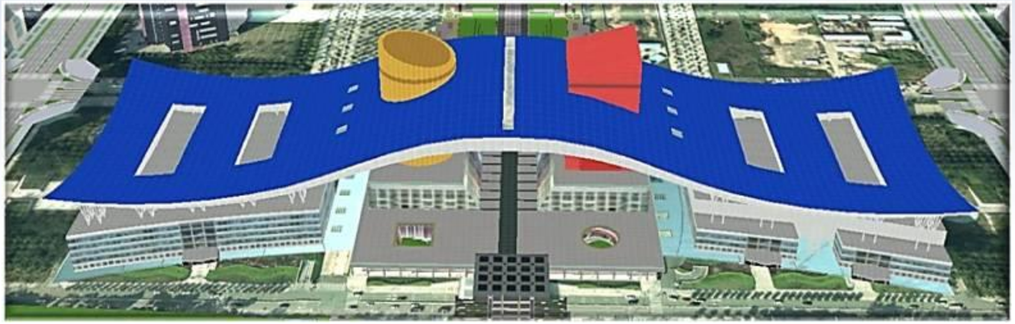


Logical Design and Implementation of the Data Model for 3D Cadastre in China

Renzhong GUO, **Changbin YU,** Biao HE,
Zhigang ZHAO, Lin LI, Shen Ying

2012.10.25

Outlines

1 Analysis of 3D Spatial Data Model and Entities

2 Design of Cadastre-oriented 3D Data Model in China

3 Design of the Detailed Logical Model

4 Case Study

Outlines

‘Property Law of the People’s Republic of China’

- **‘Property Law of the People’s Republic of China’** becomes effective in 2007 which provides legal support for establishment of 3D cadastre in China.

- **‘Property Law’ article 136**: The right to use land for construction may be established separately on the surface of, or above, or under the land. The newly-established right to use land for construction shall not damage the usufractuary right that has already been established.

Current 3D Spatial Data Models

- **Schemes for Solids Modelling**

  -CSG (Construction Solid Geometry)

  -B-rep (Boundary Representation)

  -...

- **Current 3D Spatial Data Models**

  -3D FDS (3D Formal Data Structure)

  -TEN (Tetrahedron)

  -SSM (Simplified Spatial Model)

  -UDM (Urban Data Model)

  -OO3D (Object-oriented 3D Model)

  -...

Both advantages and disadvantages

- 3D FDS
  - a formal description based on 3D vector map
  - combination of geometries and thematic elements
  - strict definitions of topological relationships lacked
- TEN
  - based on a simplicial complex
  - a typical CSG approach
- SSM
  - a simplified model of 3D FDS
  - the geometric primitive Arc omitted
  - good at visual inquiry in web-oriented applications
  - have to be convex faces (although a concave polygon can be decomposed into several convex polygons)
- UDM
  - a special version of SSM
  - faces limited to be triangles
- OO3D
  - every volume is composed of triangles in essence (i.e. every volume is composed of faces, and every face is composed of triangles)

Definitions of 2D Polygons

(1) in CGAL

Simple Polygons: edges don't intersect, except consecutive edges intersect in their common vertex, i.e. vertex's degree equals two

Relatively Simple Polygons: allows vertice whose degree bigger than two, but all of their edges must be disjoint in their interior.

(2) In ISO 19107 'Spatial Schema'

a GM_Polygon is a surface patch that is defined by a set of boundary curves and an underlying surface to which these curves adhere.

(3) in GML3

The ISO definition of a polygon is at the abstract level, and it is implemented by the OpenGIS Simple Feature Specification (SFS)

(4) in CityGML

The definition of polygons in GML3 (i.e. gml:Polygon) is just used

(5) in Oracle

Strictly adheres to the definition in GML3

(6) A Sort of User-defined Valid Polygon(van Oosterom, 2004)

Rings are not allowed to cross, but it is allowed that rings can touch, or even partially overlap themselves or each other in some cases where the tolerance plays an important role.

Definitions of 3D Solids

(1) in CGAL

Winged-edge data structure: suitable for orientable 2-manifold

Half-edge data structure: suitable for orientable 2-manifold

Quad-edge data structure: suitable for both orientable and non-orientable 2-manifold.

In CGAL, the surface of the Polyhedron is 2-manifold and organized by Half-edge data structure (also regarded as DCEL)

(2) in ISO 19107 'Spatial Schema'

a geometric object (GM_Solid) or a topological object (TP_Solid)

…

GM_Solid is a subclass of GM_Primitive, and it is the basis for 3-dimensional geometry.

The extent of GM_Solid is defined by the boundary surfaces, i.e. the boundary defines a sequence set of GM_Surface that limits the extent of this GM_Solid.

The ‘top’ of each GM_Surface as defined by its orientation shall face away from the interior of the solid

(3) in GML3

Gml:Solid implements GM_Solid in ISO 19107.

Other 3D Objects in OGC Specifications include cones, spheres. And some 3D objects are not regarded as volumetric solids, but they still appear in 3D space, e.g. free-form curves and free-form surfaces.

(4) in CityGML

Gml:Solid is just used.

(5) in Oracle

Follows OGC/ISO GML3 Specifications, i.e. simple solids, composite solids.

A simple solid: defined as a ‘Single Volume’ bounded on the exterior by one exterior surface and on the interior by zero or more interior composite surfaces.

A composite solid: a combination of N simple solids.

Topological Relationships between 3D Solids

(1) in ISO 19107 'Spaital Schema'

Aggregate package, complex package (including composites) are used

(2) in GML3

Geometry aggregates: arbitrary aggregations of geometry elements. They are not assumed to have any additional internal structures.

Geometric complexes: closed collections of geometric primitives, i.e. they will contain their boundaries.

A Geometric composite: represents a geometric complex with an underlying core geometry that is isomorphic to a primitive i.e. it can be viewed as a primitive and as a complex...

In 3D, a gml:CompositeSolid is represented by a set of orientable surfaces. Essentially, a composite solid is a collection of solids that join in pairs on common boundary surfaces and which form a single solid when considered as a whole.

(3) in CityGML

It based on the standard ISO 19107 'Spaital Schema', and it is represented by objects of GML3's geometry model. It uses only a subset of the GML3 geometry package, defining a profile of GML3.

Combined geometries in CityGML can be aggregates, complexes or composites…

An aggregate: the spatial relationship between components is not restricted. They may be disjoint, overlapping, touching, or disconnected. GML3 provides a special aggregate for each dimension, i.e., a Multi-point, a Multi-curve, a Multi-surface, and a Multi-solid.

A complex: topologically structured, i.e. its parts must be disjoint, must not overlap, and are allowed to touch, at most, at their boundaries or share parts of their boundaries.

A composite: a special complex provided by GML3. It can only contain elements of the same dimension. Its elements must be disjoint as well, but they must be topologically connected along their boundaries. In 3D, solids must be topologically connected by common boundary surfaces.

Outlines

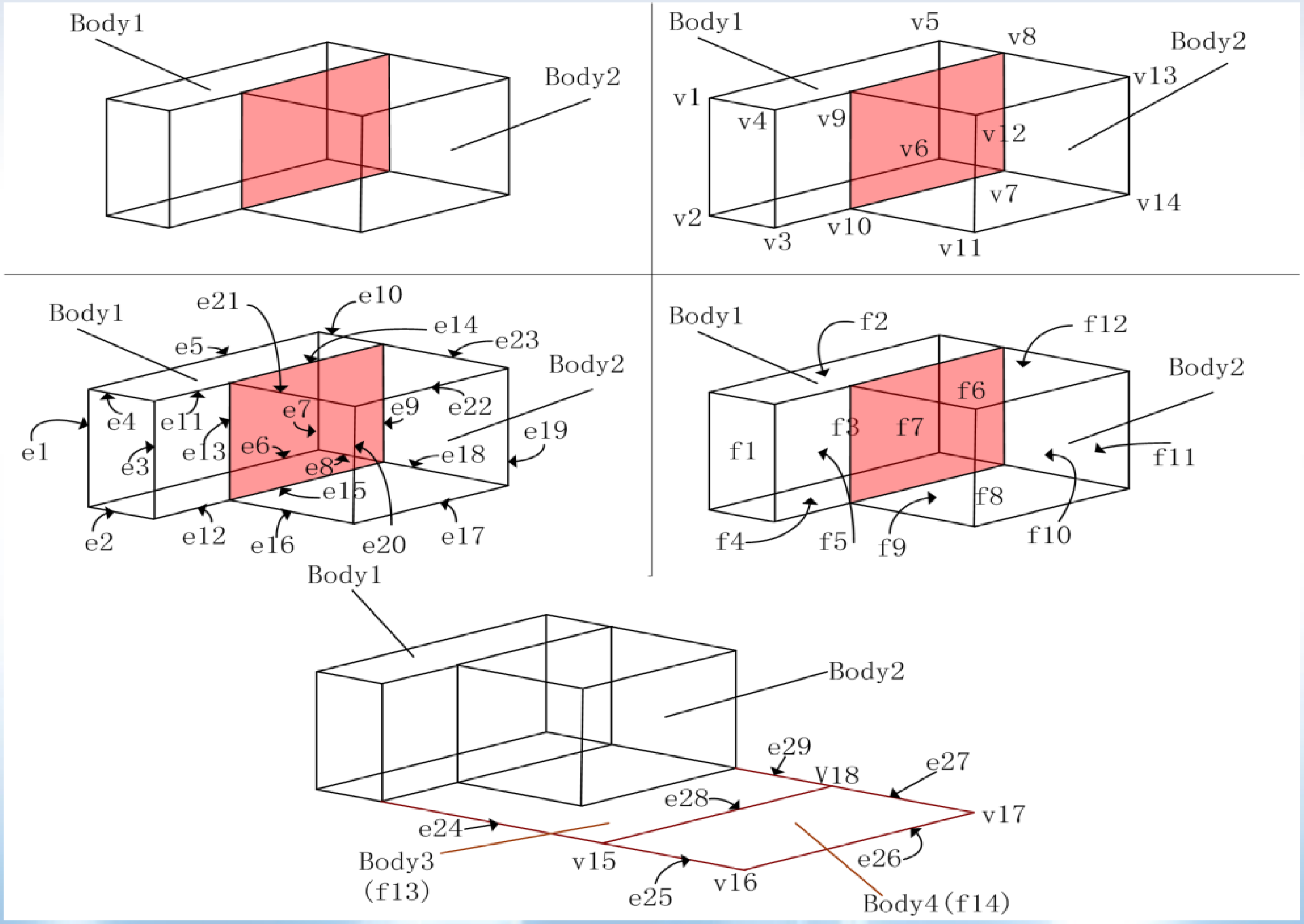
1 Analysis of 3D Spatial Data Model and Entities

2 Design of Cadastre-oriented 3D Data Model in China

3 Design of the Detailed Logical Model

4 Case Study

The Corresponding Conceptual Model



History of Designing the Conceptual Model

- In 2006, the 3D conceptual model integrating land properties and house properties was proposed, e.g. 'the joint register mechanism' adopted in title registration
- In 2007, more detailed definition of the property object given
- In 2010,2011, more detailed explanations about this data model from the perspective of technology is presented where the compatibility between PSLG(Planar Straight Line Graph) and PLC(Piecewise Linear Complex) in B-rep is great referenced.
- In 2011, detailed explanations about how to obtain body information in this data model (i.e. construction of the smallest bodies based on discrete faces) is given.
- In 2011, this data model is applied in cases of Shenzhen, China.

However, systematic analysis of the corresponding logical model of this data model has not been given yet which is greatly desired for deeper understanding of this data model.

Outlines

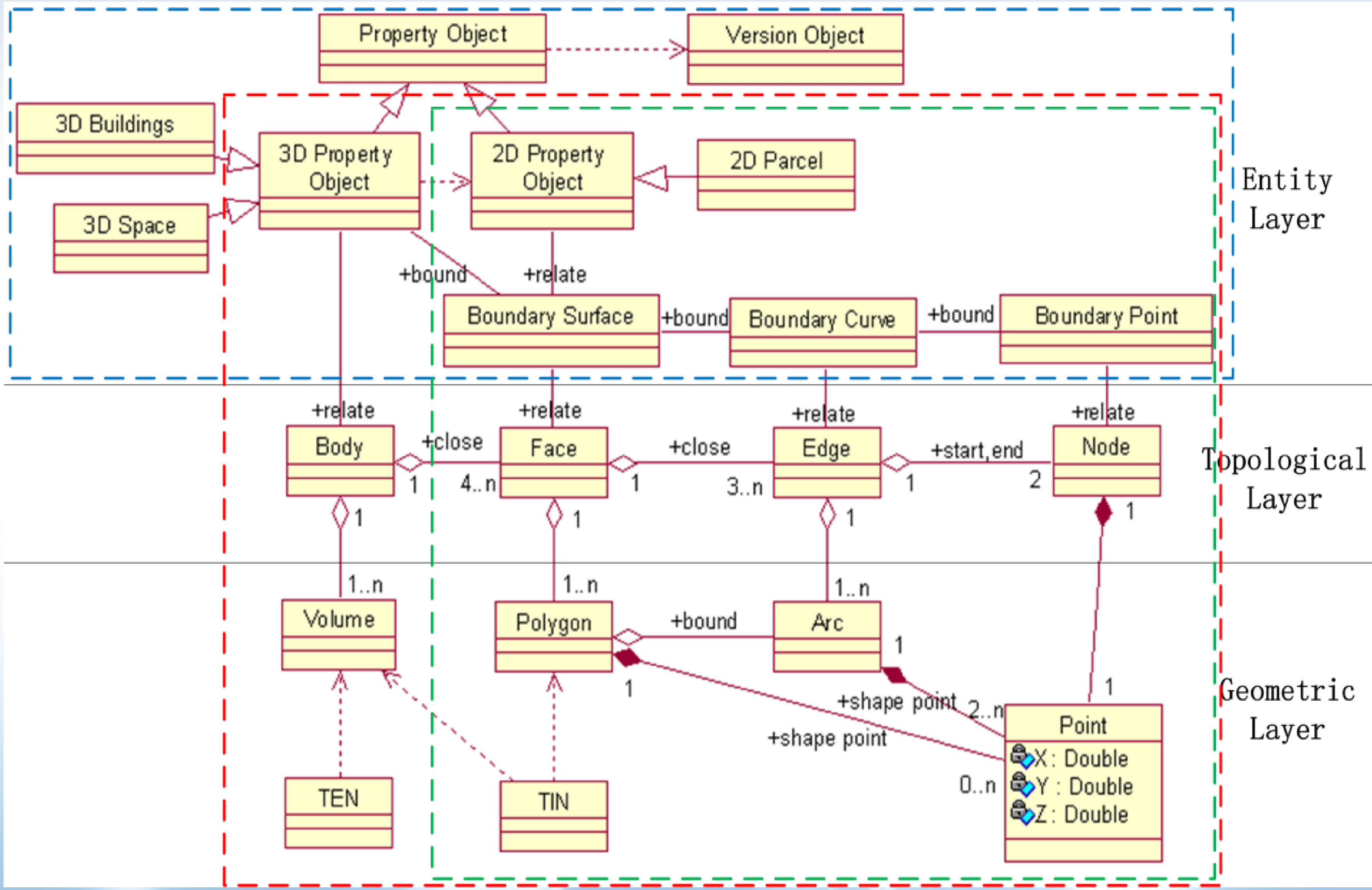
1 Analysis of 3D Spatial Data Model and Entities

2 Design of Cadastre-oriented 3D Data Model in China

3 Design of the Detailed Logical Model

4 Case Study

Detailed Design of this 3D Spatial Data Model



- There exists three layers, i.e. the geometric layer, the topological layer, the entity layer.
- In the geometric layer, geometric primitives (i.e. Point, Arc, Polygon, Volume, TIN, TEN) are basic primitives for constructing 3D solids.
- In the topological layer, topological primitives (i.e. Node, Edge, Face, Body) composed of geometric primitives are immediate primitives for constructing 3D solids.
- In the entity layer, cadastral entities (i.e. boundary surfaces, boundary curves, boundary points) which form property objects could also be regarded as topological primitives combined with semantic information

The Primitives

(1) Point

Every point is a point in 3D, and it is recorded in (X,Y,Z).

A node is a point feature consisting of a point.

(2) Arc

An arc is a directed line segment in 3D, and it is bounded by two points, i.e. the beginning point, the ending point.

An edge is a curve existing in 3D which is composed of one or more arcs.

(3) Polygon

A polygon is a closed region bounded by at least three arcs, and these arcs located in the same plane are arranged one by one to form a ring.

A polygon is a kind of relatively simple polygon, and it could be convex, concave, or may have holes.

A face is a surface in 3D comprising of one or more polygons. In 2D property objects, faces have no directions. In 3D property objects, faces are used to enclose a body, so each face has a direction.

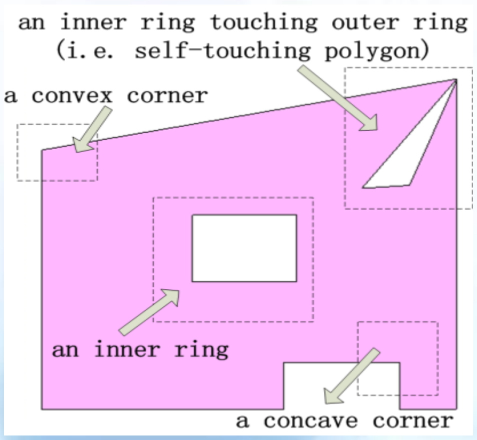
(4) Body

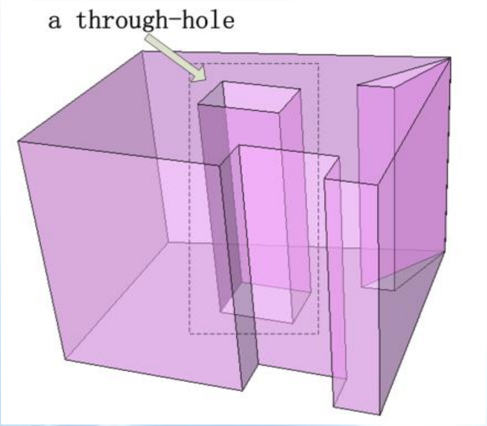
A body is a closed volume bounded by at least four faces. The face directed towards the interior of the body is regarded as the back face, and the face directed towards the exterior is regarded as the front face.

The direction of the face is determined by the normal which can be calculated from the orientation of edges bounding the face, and the right-hand rule is employed, i.e. the thumb points to the normal of the face, and the orientation of the other four fingers is the orientation of edges.

When edges bounding the face have been determined, **the orientation of edges** will be compared with the orientation of each edge itself. If they match, the edge will be tagged ‘+’, otherwise ‘-’.

A body can be convex, concave, or may have through-holes (the same definition as a simple solid in Oracle)





The Topological Relationships

The relationships must obey the following rules:

(1) between Point and Arc

the same point can be incident to different arcs

(2) Properties of Arc

An arc is directed, i.e. from the starting point to the end point, an arc can be shared by several polygons

(3) between Arc and Arc

The arcs are broken at the intersecting point if they intersect, and new arcs are formed by original arcs and the intersecting points

(4) between Arc and Polygon

Arcs are arranged one by one to form a polygon.

No isolated arcs, hanging/dangling arcs and duplicated arcs

(5) between Polygon and Polygon

Intersection between polygons is not allowed while disjunction and connection are allowed (connection at a common node or edge)

(6)between Face and Body

The body is a closed volume, and it is composed of a series of adjacent faces. No isolated faces, hanging/dangling faces and duplicated faces.

(7) between Body and Body

Intersection between bodies is not allowed while disjunction and connection are allowed (connection at a common node, edge, or face)

The ideas of **‘non-intersection’** and **‘shareness’** reflected in design of the logical model provide the basis for construction of the smallest bodies.

It is a kind of **a linear topological structure** which is suitable in most situations where surfaces of the body are planar. Meanwhile, a curved surface can be represented by a collection of polygons(e.g. triangles), and a curve can be represented by a collection of straight-line segments(i.e. arcs)

The Encoding Rule for Primitives in Practice

- A node is composed of one point
- An edge is composed of one arc
- A face is composed of one polygon

- **ID of the body** starts from 900,0000 while the remaining space after 3D partition is labeled as 0
- **ID of the face** starts from 800,0000
- **ID of the edge** starts from 500,0000
- **ID of the node** starts from 100,0000

Detailed Design of Tables

Table 1 GEOM_NODE

Node_ID	*X_Coord*	*Y_Coord*	*Z_Coord*
1013866	115319.6	18094.5	-15.1

Table 2 GEOM_BODY(optional)

Body_ID	*Volume*	*XMin*	*YMin*	*ZMin*	*XMax*	*YMax*	*ZMax*
9000312	0	115256.9	18094.5	-15.1	115319.6	18253.5	3

Table 3 TOPO_FACE_BODY

Face_ID	*OutLoop_Edge*	*Face_FrontBody*	*Face_BackBody*
8005027	5018122	0	9000312

Table 4 TOPO_FACE_EDGE

Face_ID	*Face_Edge*	*Next_Edge*
8005021	5018118	5018119

Table 5 TOPO_EDGE_NODE

Edge_ID	*Begin_Node*	*End_Node*
5018118	1013866	1013867

Analysis of the Normal Form(NF)

- A relation is in good form if the relation preferably should be either 3NF or BCNF(Boyce-Codd Normal Form).
- This relation proposed satisfies BCNF, i.e. it satisfied the following NFs:

(1) 1NF, i.e. no repeating groups

(2) 2NF, i.e. no non-key attribute is dependent on part of a key, i.e. no partial dependencies

(3) 3NF, i.e. every attribute transitively dependent on a key is a key attribute, i.e. no transitive dependencies

(4) BCNF, i.e. every determinist is a candidate key

The Implicit Mode for Storing Body Information

Step 1: Search for all faces bounding the body

$Select\ {*}\ from\ table3\ where\ Face\_FrontBody = k\ or\ Face\_BackBody = k\ as\ A$

$Foreach\ A\ in\ Step1$

Step 2: Search for the beginning edge of the only outer ring bounding the face

$Select\ {*}\ from\ table4\ where\ Face\_ID = A.Face\_ID\ and\ Face\_Edge = A.OutLoop\_Edge\ as\ B$

Step 3: Search for all the edges of the only outer ring bounding the face

$Do(Select\ {*}\ from\ table4\ where\ Face\_ID = A.Face\_ID\ and\ Face\_Edge = B.Next\_Edge\ as\ C)$

$Util(C.Next\_Edge == A.OutLoop\_Edge)$

$Foreach\ B\ in\ Step2,\ Foreach\ C\ in\ Step3$

Step 4: Search for the beginning node and the ending node bounding the edge

$Select\ {*}\ from\ table5\ where\ Edge\_ID\ in\ (B.Face\_Edge, C.Face\_Edge)\ as\ D$

$Foreach\ D\ in\ Step4$

Step 5: Search for the geometrical data of the node

$Select\ {*}\ from\ table1\ where\ Node\_ID\ in\ (D.Begin\_Node, D.End\_Node)\ as\ E$

$Foreach\ A\ in\ Step1$

Step 6: Search for the beginning edge of a certain inner ring bounding the face

$Select\ top\ 1\ {*}\ from\ table4\ where\ Face\_ID = A.Face\_ID\ and\ Face\_Edge\ not\ in\ (B.Face\_Edge, C.Face\_Edge)\ as\ F$

Step 7: Search for all the edges of a certain inner ring bounding the face

$Do(Select\ {*}\ from\ table4\ where\ Face\_ID = A.Face\_ID\ and\ Face\_Edge = F.Next\_Edge\ as\ G)$

$Until(G.Next\_Edge == F.Face\_Edge)$

$Foreach\ A\ in\ Step1$

Step 8: Search for the beginning edge of the next inner ring bounding the face

$Select\ top\ 1\ {*}\ from\ table4\ where\ Face\_ID = A.Face\_ID\ and\ Face\_Edge\ not\ in\ (B.Face\_Edge, C.Face\_Edge, F.Face\_Edge, G.Face\_Edge)$

Step 9: Search for all the edges of the next inner ring bounding the face

....

The Explicit Mode for Storing Body Information

- Detailed body information is stored explicitly in CLOB(Character Large Object) field as one record

$$
\begin{aligned}
&Body(Body\_ID,\\
&Node(id, x, y, z), Node(id, x, y, z), Node(id, x, y, z), Node(id, x, y, z), \ldots,\\
&Edge(id, bn, en), Edge(id, bn, en), Edge(id, bn, en), Edge(id, bn, en), \ldots,\\
&Face(id, be, \ldots, ee; \ldots; be, \ldots, ee), Face(id, be, \ldots, ee; be, \ldots, ee), \ldots)
\end{aligned}
$$

Outlines

1 Analysis of 3D Spatial Data Model and Entities

2 Design of cadastre-oriented 3D Model in China

3 Design of the Detailed Logical Model

4 Case Study

Case Study

Tab.6 Analysis of Right Space Objects of Several Typical Buildings in Shenzhen, China

Name	*Node Number*	*Edge Number*	*Face Number*	*Body Number*	*Timer in C/S* / ms	*Time in B/S* / ms
FengShenTing(bazaar)	516	799	350	36	>4000	<100
Excellence Century Center	84	138	64	6	>4000	<100
Convention & Exhibition Center	108	186	82	5	>4000	<100
Underground Parking Lot	18	27	11	1	>1000	<100
Luohu Railway Station	48	72	30	3	>4000	<100
Metro from Laojie to Guomao	582	891	328	14	>4000	<100
Power Supply Bureau	16	28	16	3	>3000	<100
WanXiangCheng(i.e. The MixC)	50	75	31	3	>3000	<100
ZTE Building	81	130	58	6	>4000	<100
Hongkong-Shenzhen Western Corridor	515	807	289	7	>4000	<100

One Case

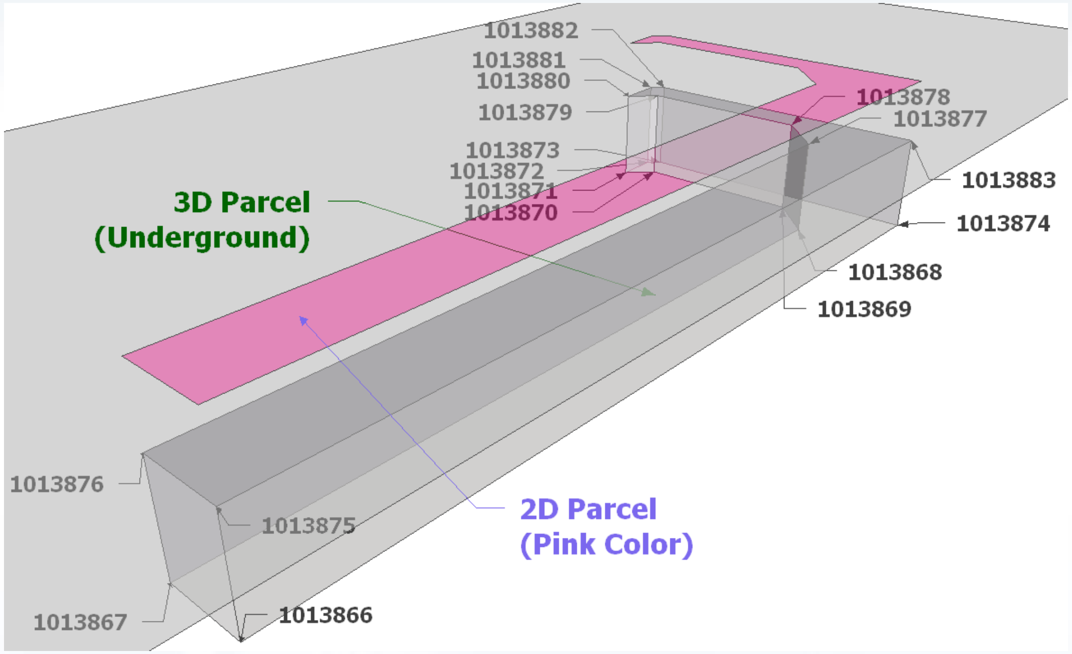


Fig. The underground parking lot of the energy building which is already leased

Two Geometrical Tables

- 9*2=18 records (Table GEOM_NODE)

Table 1 GEOM_NODE

Node_ID	*X_Coord*	*Y_Coord*	*Z_Coord*
1013866	115319.6	18094.5	-15.1
1013867	115305.5	18094.5	-15.1
1013868	115305.5	18237.8	-15.1
1013869	115296.5	18246.8	-15.1
1013870	115262.3	18246.8	-15.1
1013871	115257.6	18241.9	-15.1
1013872	115256.9	18251.4	-15.1
1013873	115259.2	18253.5	-15.1
1013874	115319.6	18253.5	-15.1
1013875	115319.6	18094.5	3
1013876	115305.5	18094.5	3
1013877	115305.5	18237.8	3
1013878	115296.5	18246.8	3
1013879	115262.3	18246.8	3
1013880	115257.6	18241.9	3
1013881	115256.9	18251.4	3
1013882	115259.2	18253.5	3
1013883	115319.6	18253.5	3

Table 2 GEOM_BODY(optinal)

Body_ID	*Volume*	*XMin*	*YMin*	*ZMin*	*XMax*	*YMax*	*ZMax*
9000312	0	115256.9	18094.5	-15.1	115319.6	18253.5	3

- 1 record (Table GEOM_BODY)

Three Topological Tables

- 9 +2 = 11 records (Table TOPO_FACE_BODY)

Table 3 TOPO_FACE_BODY

Face_ID	*OutLoop_Edge*	*Face_FrontBody*	*Face_BackBody*
8005027	5018122	0	9000312
8005028	5018123	0	9000312
8005029	5018124	0	9000312
8005030	5018125	0	9000312
8005031	5018126	0	9000312
8005021	5018118	0	9000312
8005022	5018127	0	9000312
8005023	5018118	0	9000312
8005024	5018119	0	9000312
8005025	5018120	0	9000312
8005026	5018121	0	9000312

- 9*3=27 records (Table TOPO_EDGE_NODE)

Table 5 TOPO_EDGE_NODE

Edge_ID	*Begin_Node*	*End_Node*
5018118	1013866	1013867
5018119	1013867	1013868
5018120	1013868	1013869
5018121	1013869	1013870
5018122	1013870	1013871
5018123	1013871	1013872
5018124	1013872	1013873
5018125	1013873	1013874
5018126	1013874	1013866
5018127	1013875	1013876
5018128	1013876	1013877
5018129	1013877	1013878
5018130	1013878	1013879
5018131	1013879	1013880
5018132	1013880	1013881
5018133	1013881	1013882
5018134	1013882	1013883
5018135	1013883	1013875
5018136	1013867	1013876
5018137	1013875	1013866
5018138	1013868	1013877
5018139	1013869	1013878
5018140	1013870	1013879
5018141	1013871	1013880
5018142	1013872	1013881
5018143	1013873	1013882
5018144	1013874	1013883

- 9*2 + 4*9 = 18 + 36 = 54 records (Table TOPO_FACE_EDGE)

Table 4 TOPO_FACE_EDGE

Face_ID	*Face_Edge*	*Next_Edge*
8005021	5018118	5018119
8005021	5018119	5018120
8005021	5018120	5018121
8005021	5018121	5018122
8005021	5018122	5018123
8005021	5018123	5018124
8005021	5018124	5018125
8005021	5018125	5018126
8005021	5018126	5018118
8005022	5018127	5018135
8005022	5018135	5018134
8005022	5018134	5018133
8005022	5018133	5018132
8005022	5018132	5018131
8005022	5018131	5018130
8005022	5018130	5018129
8005022	5018129	5018128
8005022	5018128	5018127
8005023	5018118	5018137
8005023	5018137	5018127
8005023	5018127	5018136
8005023	5018136	5018118
8005024	5018119	5018136
8005024	5018136	5018128

The Implicit Mode for Storing the Body (i.e. SQL of Construction)

Step 1:

```
select "SZ3D"."TOPO_FACE_BODY"."FACE_ID",
    "SZ3D"." TOPO_FACE_BODY "."OUT_LOOP_EDGE",
    "SZ3D"." TOPO_FACE_BODY "."FACE_FRONT_BODY",
    "SZ3D"." TOPO_FACE_BODY "."FACE_BACK_BODY"
   from "SZ3D"." TOPO_FACE_BODY "  where FACE_FRONT_BODY=9000312 OR FACE_BACK_BODY=9000312
```

Resluts: 11 records

Step 2, Step 3:

```
select "SZ3D"."TOPO_FACE_EDGE"."FACE_ID",
   "SZ3D"."TOPO_FACE_EDGE"."FACE_EDGE ",
   "SZ3D"."TOPO_FACE_EDGE"."NEXT_EDGE "
      from "SZ3D"."TOPO_FACE_EDGE"  WHERE FACE_ID in
      (8005027,8005028,8005029,8005030,8005031,
      8005021,8005022,8005023,8005024,8005025,8005026)
```

Results: 54 records

Step 4:

```
select "SZ3D"."TOPO_EDGE_NODE"."EDGE_ID",
   "SZ3D"."TOPO_EDGE_NODE"."BEGIN_NODE",
   "SZ3D"."TOPO_EDGE_NODE"."END_NODE"
      from "SZ3D"."TOPO_EDGE_NODE" where EDGE_ID in (
      5018118, 5018119, 5018120, 5018121, 5018122, 5018123, 5018124, 5018125, 5018126,
      5018127, 5018128, 5018129, 5018130, 5018131, 5018132, 5018133, 5018134, 5018135,
      5018136, 5018137, 5018138, 5018139, 5018140, 5018141, 5018142, 5018143, 5018144)
```

Results: 27 records

Step 5:

```
select "SZ3D"."GEOM_NODE"."NODE_ID",
      "SZ3D"."GEOM_NODE"."X_COORD",
      "SZ3D"."GEOM_NODE"."Y_COORD",
      "SZ3D"."GEOM_NODE"."Z_COORD"
   from "SZ3D"."GEOM_NODE" where NODE_ID in (
   1013866, 1013867, 1013868, 1013869, 1013870, 1013871, 1013872, 1013873, 1013874,
      1013875, 1013876, 1013877, 1013878, 1013879, 1013880, 1013881, 1013882, 1013883)
```

Results: 18 records

The Explicit Mode for Storing the Body (i.e. Stored in CLOB)

Body(9000312,

Node(1013866,115319.6,18094.5,-15.1),Node(1013867,115305.5,18094.5,-15.1),
Node(1013868,115305.5,18237.8,-15.1),Node(1013869,115296.5,18246.8,-15.1),
Node(1013870,115262.3,18246.8,-15.1),Node(1013871,115257.6,18241.9,-15.1),
Node(1013872,115256.9,18251.4,-15.1),Node(1013873,115259.2,18253.5,-15.1),
Node(1013874,115319.6,18253.5,-15.1),Node(1013875,115319.6,18094.5,3),
Node(1013876,115305.5,18094.5,3),Node(1013877,115305.5,18237.8,3),
Node(1013878,115296.5,18246.8,3),Node(1013879,115262.3,18246.8,3),
Node(1013880,115257.6,18241.9,3),Node(1013881,115256.9,18251.4,3),
Node(1013882,115259.2,18253.5,3),Node(1013883,115319.6,18253.5,3),

Edge(5018118,1013866,1013867),Edge(5018119,1013867,1013868),Edge(5018120,1013868,1013869),
Edge(5018121,1013869,1013870),Edge(5018122,1013870,1013871),Edge(5018123,1013871,1013872),
Edge(5018124,1013872,1013873),Edge(5018125,1013873,1013874),Edge(5018126,1013874,1013866),
Edge(5018127,1013875,1013876),Edge(5018128,1013876,1013877),Edge(5018129,1013877,1013878),
Edge(5018130,1013878,1013879),Edge(5018131,1013879,1013880),Edge(5018132,1013880,1013881),
Edge(5018133,1013881,1013882),Edge(5018134,1013882,1013882),Edge(5018135,1013883,1013875),
Edge(5018136,1013867,1013876),Edge(5018137,1013875,1013866),Edge(5018138,1013868,1013877),
Edge(5018139,1013869,1013878),Edge(5018140,1013870,1013879),Edge(5018141,1013871,1013880),
Edge(5018142,1013872,1013881),Edge(5018143,1013873,1013882),Edge(5018144,1013874,1013883),

Face(8005027,5018122,5018140,5018131,5018141),Face(8005028,5018123,5018141,5018132,5018142),
Face(8005029,5018124,5018142,5018133,5018143),Face(8005030,5018125,5018143,5018134,5018144),
Face(8005031,5018126,5018144,5018135,5018137),Face(8005026,5018121,5018139,5018130,5018140)
Face(8005023,5018118,5018137,5018127,5018136),Face(8005024,5018119,5018136,5018128,5018138),
Face(8005025,5018120,5018138,5018129,5018139),
Face(8005021,5018118,5018119,5018120,5018121,5018122,5018123,5018124,5018125,5018126),
Face(8005022,5018127,5018135,5018134,5018133,5018132,5018131,5018130,5018129,5018128))

- In this paper, more analysis about this data model is given, including retrospection of the history of designing the corresponding conceptual model, detailed analysis of the corresponding logical model, explanations about how to design tables based on the relational database, analysis about the normal form(NF), etc.
- And in a long period form now on, there will be a focus on how to design a useful 3D spatial data model in 3D cadastre.

Thank you