BOUNDARY, COBOUNDARY, SIX GROUPS TOPOLOGY AND 'ONE FACE WITH THREE LAYERS' IN 3D GIS

Qingyuan LI [a,b], Yanqing CHANG[a], Daiyong CAO[b]

[a]Chinese Academy of Surveying&Mapping, Beijing Beitaiping Road 16, 100039
LiQY@casm.ac.cn
[b]The Key Laboratory of Coal Resource, Ministry of Education, China, Institute Road Ding 11,Beijing ,100083

Commission II, WG II/3

KEY WORDS: 3DGIS, Boundary, Coboundary**,** Topology, One face with three layers

ABSTRACT:

On recognition to boundary and coboundary in ISO TC211 Geographic Information "Spatial schema" International Standard. The author modify 'five-group relations', which the authors have proposed formerly, to 'six-group relations'. A new concept of 'One face with three layers' in 3D GIS topological relation is given in the paper. It is a supplement and development for relation of 'Face-ExteriorRing-InteriorRing-PositiveSolid-NegativeSolid' in the 'Five group relations'. 'One face with three layers' supposes that one face has three layers, i.e. positive layer, middle layer and negative layer. Positive layer and negative layer are divided separately, middle layer overlap division of positive layer and negative layer. In this way, the integrality of a nature face is retained, division of positive side and negative side are isolated, but have some connection through middle layer. Data structure of 'One face with three layers' is designed in the paper. If construction 3D topologic relation in this way, inquire of one polyhedron's adjacent polyhedrons is very easy. Last, the author expands the concept of "One face with three layers" to 2DGIS, to deal with the line object like freeway, railway and river, which may be nature boundary of districts.

1. INTRODUCE

In spatial, the real world is three dimensions, but most current GIS can not perfectly express multi-layer information in vertical, so it is called 2D or 2.5D GIS. The domains of geology, mining, environment and other applications need 3DGIS platform to support processing real 3D information. Not all the applications need 3DGIS. In fact, for most applications, 2D or 2.5D is enough. The users who real need 3DGIS, may only 5% of GIS users, but these applications have very important position in the industry, it may produce 10% market gain. With the development of 3D visualization tools, 3DGIS again become a study hot point in GIS. In 3D GIS, there are many theory and technique problems, which have not been solved perfectly. One of them is 3DGIS's data model and topological relationship. Many scholars have studied this problem from different standpoints. Martien Molenaar (1990) and Dieter Fritsch (1996) studied formal data structure for 3D vector model. Han Guojian, (1992), Guo Dazhi (Guo, et al., 1993) study linear octatree express 3D underground object. Chen Jun & Guo Wei (Guo and Chen, 1997, 1998) have studied in 3D geometry elements topologic from 9 intersection model. Pilouk Morakot (1994), Chen Xiaoyong (1994a, 1994b) study tetrahedron voxel fill model. Li Qingyuan & Cao Daiyong(Li, 1996; Li, et al.,1996) from mine & geology application, have studied in (1) 'Five group relation' to express 3D object topologic relationship, (2) 'Introducing surface - dividing body' as a way to dynamic build and maintain the 3D topology, (3) 'One face with three layers' to deal with contradiction of integrality of face with difference of both side (positive, negative) adjoin polyhedron, (4) 3D volume function interpolation in solid inner.

In ISO Geographic Information Spatial Schema (ISO/DIS 19107), a geographic spatial model is given. Boundary and coboundary of spatial object is emphasized in topological package. In this paper, on understanding to boundary and coboundary, the authors review the five group relationships that the author gave before. Lurked problem of 'Face - ExteriorRing - InteriorRings - PositiveSolid - NegativeSolid' is discussed. A new concept of 'One face with three layers' is given in the paper, and extending it in 2D as 'One line with three layers'.

2. BOUNDARY AND COBOUNDARY OF SPATIAL ENTITY

Boundary and coboundary of geographic object are emphasized in the ISO geographic information international standard 'Spatial Schema' (ISO/DIS 19107).

Boundary is a set that represents the limit of an object, i.e. a transition between an object and the rest of its domain of discourse. Dimension of an object boundary is less one than the entity. For example, the boundary of a line (1-dimension) is a set of two end points (0-dimension). Boundary of a face (2-dimension) is a set of lines (1-dimension) which are grouped to rings that one is exterior ring and some are interior rings (if the face has holes). Boundary of a solid (3-dimension) is a set of faces, which are grouped to one exterior shell and some are interior shells (if the solid has holes). **Coboundary** is a set of topological primitives of higher topological dimension associated with a particular topological object, such that this topological object is in each of their boundaries. If a geometry object A is in the boundary of a geometry object B, then the B is coboundary of A. For example, If a node is on the boundary of an edge, that edge is on the coboundary of that node. Any orientation parameter associated to one of these relations would also be associated to the other. If an edge is on boundary of a face, the face is coboundary of the edge. If a face is on boundary of a solid, the solid is coboundary of the face.

Spatial object can associate to the object which are less one dimension and on its boundary by getting boundary operation, also can associate to the object which is higher one dimension than it and as the central object in a complex. In one dimension linear case, coboundary of one node has at most two edges. In the full topology case, there are precisely 2 edges. In the 2-dimensional planar case, the coboundary of an edge has at most 2 faces. In the full topology case, there are precisely 2 faces, one face lying to the left of the edge, and another lying to the right of the edge. In the 3-dimensional solid case, the coboundary of face has at most 2 solids. In the full topology case, there are precisely 2, one solid lying to the positive side of the face, and another lying to the negative side of the face.

3. SIX GROUPS TOPOLOGICAL RELATION

Although TC211 Spatial Schema has defined self-contained vector spatial model, it is not oriented to program realization. On the view of boundary and coboundary, the author think that the five group of relation, which the author gave formerly, are mainly considering of the spatial primitive and its boundary and coboundary. If adding shell, the next six group relations could be as a base of 3D GIS.

(1) Node-BeginEdges-EndEdges: Node, an array of edges, which begin from the node, and an array of edges, which end at the node. The BeginEdges and EndEdges, all are coboundary of the node.

(2) Edge-BeginNode-EndNode-Rings: Edge, begin and end node of the edge, and array of rings which through the edge. Two of the BeginNode and the EndNode are boundary of the edge, interior adjacent faces of the rings are coboundary of the edge.

(3) Ring-Edges-InteriorFace: Ring, an array of edges, which composed the ring, and interior face of the ring. The edges are composed component of the ring, not the boundary of the ring. The ring is cycle, so the boundary is empty. The ring has direction, interior adjoin face is on the left side of the ring. The ring is a bridge between of face and the edges that are the boundary composed component.

(4)Face-ExteriorRing-InteriorRings-PositiveSolid-NegativeSolid: Face, ExteriorRing that composed exterior boundary, InteriorRing that boundary interior holes (if they exist), positive adjacent solid and negative adjacent solid. Because ExteriorRing is most important and use frequently, so deal with ExteriorRing and InteriorRing apart. It is also accepted that combine the ExteriorRing and InteriorRing, and assume that first ring ExteriorRing, afterward are InteriorRing. When the number of ring is 1, without InteriorRing. Boundary of the face is a set of edge that are on the rings (ExteriorRing + InteriorRing). Coboundary of the face is a set of PositiveSolid and NegativeSolid.

(5) Shell-Faces-InteriorSolid: Shell, an array of the faces that composed the shell and the interior solid of the shell. A shell is composed of many oriented face and negative references of faces. A shell is a cycle, so its boundary is empty. A shell is a bridge between Solid and the faces. A shell has direction, interior solid is in interior of a shell. The direction of a shell points to away from the solid.

(6) Solid-ExteriorShell-InteriorShells: Solid, ExteriorShell and an array of InteriorShell. The boundary of solid is a set of faces or negative agents of faces, which are grouped to one ExteriorShell and some InteriorShell (if Solid have interior hole). Coboundary of a Solid is empty.

These six group relations can be expressed by C language as Appendex A.

Though in ISO/DIS 19107 said that a solid may has interior boundary shells only, no exterior boundary shell, or a face has interior boundary ring only, no exterior boundary ring. The author thinks that in general GIS application, these special cases could not be considered.

4. 'ONE FACE WITH THREE LAYERS'

4.1 Puzzle of Sub-Face Spliting

3D GIS is an expanding of 2D GIS in 3D space. Rechecking 2D GIS's topological relationship, which we are familiar with, has help to we study 3D GIS, for we often can expand the think way form 2D to 3D. Now, from a figure 1, let us check relation 'Edge – BeginNode – EndNode – LeftPolygon – RightPolygon' in 2D GIS.

Figure 1. In 2D GIS, an edge only has one left polygon and on right polygon

If the number of left or the right polygons of an edge are more than one, the edge should be divided into several edges, to ensure every edge only has one left polygon and one right polygon. Recalling in 2D full topology ,coboundary of an edge has just 2 polygons, one lying to the left and another lying to the right of the edge. In figure1, the edge AD should be divided into AB, BC, CD. In the same way, the relation of 'Face-ExteriorRing-InteriorRings-PositiveSolid-NegativeSolid' in 3D has the same case. If positive solid or negative solid of a face are more than one, the face should be divided into several sub-face, to make every sub-face only has one positive solid and one negative solid. But the sub-dividing will bring other problems.

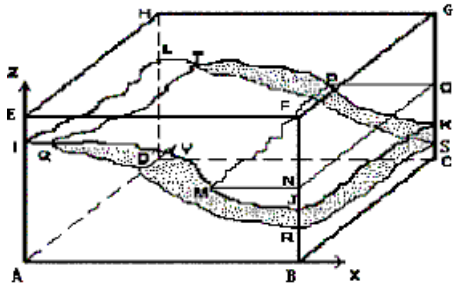

Figure 2. The ground face should be divided into three sub-faces

In figure 2, a simple geology model. Above the ground face, east is water and west is land. Under the ground face, east is modern deposit and west is base rock. The ground face seem to be split into IQTLI、QMPTQ、MJKPM three sub-faces. Another example is a fault face, in figure 3, seems to be divided into five sub-faces.

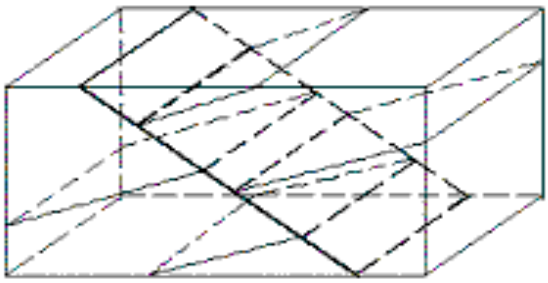

Figure 3. The fault face should be divided into five sub-faces

An original nature face (ground face, fault face) has to be divided into several sub-faces for building topologic relation. Is it in reasonable? On the other hand, change of the adjacent solid in positive side or negative side of a face will affect all division, but actually, division in positive side and negative side has not relation. We hope that integrality of a nature face should be maintained, positive and negative side division should be isolated each other, but should have some connection. So, it leads to next concept of 'One face with three layers'.

4.2 Concept of 'One Face with Three Layers'

The concept of 'one face with three layers' is supposed that one face has three layers, i.e. positive layer, negative layer and middle layer. Positive layer is split by positive side's solid. Negative layer split by negative side's solid. Middle layer overlap positive and negative layer's division. So sub-face of middle layer connects sub-face of positive layer and negative layer. A simple example is show in figure 4.

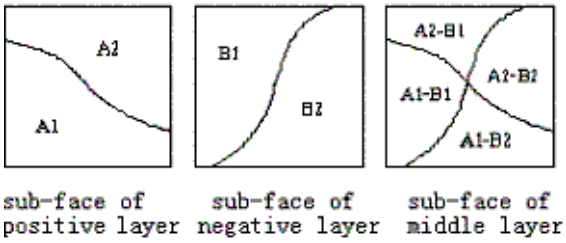

Figure 4. One face with three layer

The positive is divided into two sub-faces of A1 and A2. The negative layer is divided into two sub-faces of B1 and B2. The middle layer is divided into four sub-faces of A2-B1, A2-B2, A1-B1 and A1-B2.Now, we consider the ground-face (IJKLI) in the model of figure 2. The positive layer is divided into two sub-faces of land area (IMPLI) and water area (MJKPM) by shore line (intersection line of water top face with the ground face). It is show in figure 5.

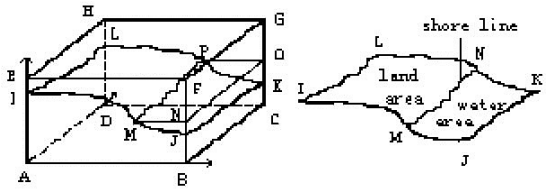

Figure 5. Shore line divide positive layer into land area and water area

The negative layer (IJKLI) is divided into two areas of base rock area (IQTLI) and loose deposit area (QJKTQ) by the base rock outcrop line QT (intersection line of unconformity face and the ground face). It is show in figure 6.

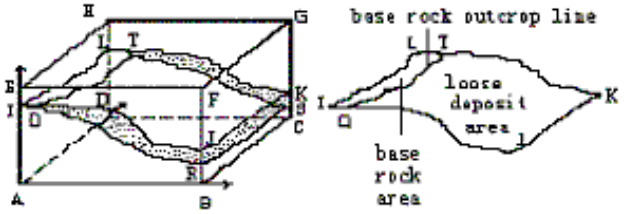

Figure 6. Base rock outcrop line divide negative line into base rock area and loose deposit area

In the middle layer, the shore line MP and base rock outcrop line QT overlap, divide the middle layer into three area of IQTLQ, QMPTQ and MJKPM. Seeing Figure.7.

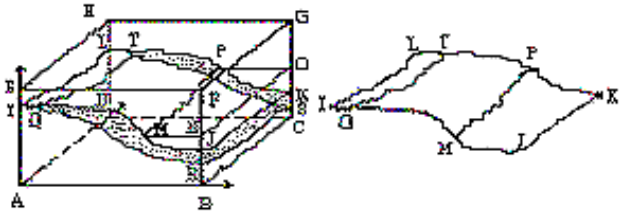

Figure 7. Middle layer of the ground surface is divided three areas

4.3 Potential Application of 'One Face with Three Layers'

Topology and data structure set by the data model of 'one face with three layers' make solid adjacent inquire very easy, such as above, below, left and right neighbor of a household in a building. In mine hydrology-geology, inquire connection of a containing water rock with mine laneway and exploitation area. Every split line in positive layer and negative layer is connected with a face. One face can be inquired connected with which face by the splitting line. It needs to be noted that in 3D GIS, not all face need to construction the topological relation of 'one face with three layers'. Only the faces such as ground face, fault or floor face, which have the effect of compartmentation, need to built 'one face with three layer'. If abuse 'one face with three layer', it may be not worth the candle.

4.4 Extension 'One Face with Three Layers' in 2D ----'One Line with Three Layers'

'One face with 3 layers' is a concept author proposed in study 3D topological relation by dint of 2D topological relation. Dose

exist similar concept in 2D GIS? The author think it dose exist too. Now let us see figure 1 again. If AD is one freeway or railway or large river, areas of 1,2,3,4 are district. We know that freeway, railway and large river often make nature boundary of district. In this case, concept of 'one face with three layers' can be used. In 2D, we can call it as 'one line with three layers'. So, for linear object like freeway, railway and river, can construct topological relation on 'one line with three layer', i.e. assume the linear object has left layer, right layer and middle layer. Left layer is split on left adjacent polygons, right layer is split on right adjacent polygons. Middle layer overlap the left layer split and right layer split, i.e. middle layer construct a relation of left side's split with right side's split. In figure1, to construct relation of 'one line with three layers' to line AD, the left layer should be split to AC, CD two sections, right layer should be split to AB, BD two sections, middle layer should be split to AB, BC, CD three sections. It can be seen that middle layer's split scheme, is split scheme of traditional 2D GIS. The topological relationship on the concept of 'one line with three layer' let the integrality of a line object (like freeway, railway, river) can be reserved, split of line left side and line right side are isolated, through middle layer, left side and right side can be setup linkage.

The structure of 'one line with three layers' is designed in Appendex B.

Traditional GIS data model is from part (arc) to entirety (linear object), but 'one line with three layers' is from entirety (linear object) to part (arc). In traditional model of point, line, polygon, a linear object is divided a lot of arcs. If user wants to inquire a river, it is needed to give the all arcs the same identify code. If user wants to inquire a region in a river left side adjacent to which region in river right side, the operation is very complicated. If using structure of 'one line with three layers', it can directly fetch left layer arc (virtual arc) left adjacent polygon, and can easy through middle layer fetch river right layer's polygon, the inquire is simpler than traditional data structure. The structure of 'One line with three layers' organises isolated arcs into a linear object with topological relation which considering the adjacent of left side and right side, getting a more convenient inquire operation and fast inquire speed by a very small index cost.

5. CONCLUSION

The authors from the view of boundary and coboundary in ISO TC211 spatial schema, check five group relations formerly gave, and analyze possible problem in construct 'Face-ExteriorRing-InteriorRings-PositiveSolid-NegativeSolid', then propose the concept of 'One face with three layers' (the author called 'One piece with three layers' previously). The corresponding data structure being designed. Then author extend the concept of 'one face with three layers' to 2D, propose the concept of ' one line with three layers' to deal with freeway, railway and river which as nature boundary of district, hoping it can help linear object adjacent inquire.

Actually, 'One face with three layer' is added an index on positive side sub-face and negative side sub-face, 'One line with three layer' is added an index on left and right of a line. They all are not obligatory relationship. If abusing it, it is possible to get more kicks than halfpence. But for some dividing face or dividing line, setup these adjunctive relationships, will convenient to spatial topological inquire.

ACKNOWLEDGEMENTS

The work is supported by Natural Science Foundation of China (SES 49771059) and by Visitor Scholar Foundation of Education Ministry Higher Education Key Laboratory.

REFERENCES

CHEN, X. and IKEDA K., 1994a. Raster dimensionl Modeling of GIS Based on Delaunay Tetraheadal Tessellations, In: *International Archives of Photogrammetry and Remote Sensing*, Munich, Germany, 30,pp. 132-139.

CHEN, X. and IKEDA K., 1994b. Raster Algorithms for Generating Delaunay Tetraheadal Tessellations. In: *International Archives of Photogrammetry and Remote Sensing*, Munich, Germany, 30, pp. 124-131.

Fritsch, D., 1996. Three-Dimentional Geographic Information System: Status and Prospects. In: Karl Kraus, Peter Waldhausl, ed. *International Archives of Photogrammetry and Remote Sensing*, Vienna, Austria: 1996, XXXI(Part B3) .pp.215~221.

GONG, J. and XIA Zhongguo, 1997. An Integrated Data Model in Three Dimensional GIS. *Journal of Wuhan Technical University of Surveying and Mapping*, 22(1), pp. 7~15.

GUO, D., YANG Weiping, HAN Guojian, 1993. A Spatial and Temporal 4-D Data Model for Mine GIS. *ACTA GEODAETICA et CARTOGRAPHICA SINICA*, 1993, 22 (2), pp. 33~40

GUO, W. and CHEN Jun, 1997. The Formal Description of Topological Spatial Relationship in 3D Based on Point Set Topology. *ACTA GEODAETICA et CARTOGRAPHICA SINICA*, 26(2), pp. 122~127.

GUO W. and CHEN Jun, 1998. A 3D Topological ER Model Based on Space Partitioning. *ACTA GEODAETICA et CARTOGRAPHICA SINICA*, 27(4), pp. 308~317.

HAN, G., GUO dazhi, JIN Xuelin, 1992. Octree Store and Index Technique of Ore Body Information. *ACTA GEODAETICA et CARTOGRAPHICA SINICA*, 21(1), pp.13~17.

ISO TC211W2, 2001, ISO /DIS19017. Geographic Information, Spatial schema.

LI, D. and LI Qingquan,1997. Study on a Hybrid Data Structure in 3D GIS. *ACTA GEODAETICA et CARTOGRAPHICA SINICA*, 26(2), pp. 128~133.

LI, R.X., 1994. Data Structure and Application Issues in 3-D Geographic Information. *GEOMATIC*, 48(3), pp. 209~224.

LI, Q.Q. and LI Deren, 1998. Research on Conceptual Frame of the Integration of 3D Spatial Data Model. *ACTA GEODAETICA et CARTOGRAPHICA SINICA* , 27(4), pp. 325~329.

LI, Q.Y., 1996. *3D Vector Structure Topological Relationship Research*: [Doctor Paper]. Beijing: C.U.M.T.

LI, Q.Y., CAO Daiyong, GAO Wentai, 1996. 3D Vector Structure Topological Relationship Based on Body Partition. In: Xu Guanhua, Li Denren, Liu Xianlin, ed. Remote Sensing in China.——*Thesis Collection of Commemorate China National*

Remote Sensing Center 15 Anniversary. Beijing: The Surveying and Mapping Press, pp.348~353.

LI, Q.Y., CAO Daiyong ZHU Xiaodi, 1999. 3D Topology Model on Body Divided. *ISPRS*, Vol. 32, Part 4 W12. pp.91-98.

LI, Q.Y. and CAO Daiyong, 1997. Dynamic Setup and Maintain of 3D GIS Topologic Relationship. *ACTA GEODAETICA et CARTOGRAPHICA SINICA*. 26(3), pp.235~240.

LI, Q.Y. and CAO Daiyong, 1998a. 3D Topologic Relation of 3D GIS. In: Lai Pohchin, Leung Yee, Shi Wenzhong, ed. *Proceedings of International Conference on Modeling Geigrophical and Environmental Systems with Geographical Information Systems*. Hong kong: Department of Grography The Chinese University of Hong Kong. pp. 832~837.

LI, Q.Y. and CAO Daiyong, 1998b. 3D Body Interpolation and Application in 3D GIS. In: Lai Pohchin, Leung Yee, Shi Wenzhong, ed. *Proceedings of International Conference on Modeling Geigrophical and Environmental Systems with Geographical Information Systems*. Hong kong: Department of Grography The Chinese University of Hong Kong. pp.826~831.

Molenaar, M., 1990. A Formal Data Structure for Three Dimensional Vector Maps. In *Proceedings of 5th Int.Symposium on Spatial Data Handling* . Zurich, pp.830~843.

Morakot, P., Klaus Tempfli, Martien Molenaa, 1994. A Tetrahedron-Based 3D Vector Data Model for Geoinformation. In: Martien Molenaa, ed. *Advanced geographic Data Modelling, －Spatial Data Modelling and Language for 2D and 3D Application*. Sylvia De Hoop, pp.129~140.

APPENDIX A. STRUCTURE OF 'ONE FACE WITH THREE LAYERS'

To realize concept of 'One face with three layers', the data structure of C is designed as follows. Here only the data members in the class are listed, all methods in the class are omitted.

```
class FACE {   // face class
   ContourARC   contourArcs[ ]; // array of contour lines
   D3RING *ExteriorRing;          // pointer of Exterior ring
   D3RING *InteriorRing[ ];       // array of interior rings
   D2EDGE *PosiSideSplitEdges[ ];// pointer array of positive
                                  // layer split edges
   D2EDGE *NegaSideSplitEdges[]; // pointer array of negative
                                  //   layer split edges
   PosiChildFace *PosiSideChildFaces[ ]; //pointer array of
                                  // positive layer child-faces
   NegaChildFace *NegaSideChildFaces[ ]; // pointer array of
                                  // negative layer child-faces
   //==== next 3 items can be omitted =======
   MidChildFace     *midChildFaces[ ]; // pointer array of
                          //middle layer child-faces
   SOLID *PositiveSolids[ ];     // pointer array of positive
                                  // side adjacent polyhedrons
   SOLID *NegativeSolids[ ] ; // pointer array of negative side
                                  // adjacent polyhedrons
};
```

In FACE class, we can also not store midChildFaces, PositiveSolids and NegativeSolids, but access them by PosiSideChildFaces and NegaSideChildFaces.

The other classes defined as follows:

```
PosiChildFace: public ChildFace {// Child face of positive layer
   SOLID *PosiSideSolid;    // pointer of positive
                            // adjacent polyhedron
   MidChildFace *MidChildFaces[ ]; // pointer array of middle
                                   //layer child-faces
};

NegaChildFace: public ChildFace{//Child face of negative layer
   SOLID *AdjacentSolid; // pointer of negative adjacent solid
   MidChildFace *MidChildFaces[ ]; // pointer array of middle
                                   // layer child-faces
};
```

We also can define a SideChildFace to replace PosiChildFace and NegaChildFace.

```
SideChildFace : public ChildFace{//Child face of negative layer
   SOLID          *AdaNegaSideSolid; // pointer of negative
                                     // adjacent polyhedron
   MidChildFace *MidChildFaces[ ]; // pointer array of middle
                                     // layer child-faces
};
```

Child face of positive layer (PosiChildFace) and negative layer (NegaChildFace) access other side solid by corresponding middle layer child face. The class of middle layer child face is defined as follow:

```
MidChildFace : public ChildFace{    //middle layer child face
  SOLID *PosiSideSolid;//pointer of positive side adjacent solid
  SOLID*NegaSideSolid;//pointer of negative side adjacent
solid
} ;
```

Positive layer, negative layer and middle layer access base face, Exterior ring of the base face and begin position, end position of contour arc in the base face by a common base class----abstract child face class.

```
class ChildFace {                    // abstract child Face
     FACE     *BaseFace;     // pointer of base Face pointer
     D3RING *ExteriorRing; // pointer of Exterior ring
     ContourFromTo *ContourSegments[]; // pointer array
                                        // of contour arc
} ;
```

ContourFromTo class is designed for searching start position and end position in the arc of the base face. It can prevent store contour arc in child face again, but can quickly get pattern contour arc from its base face.

```
class ConrourFromTo {    // contour arc
   ContourARC *BaseContour; // the contour arc responding
                            // base contour arc in base face
   D2VPOINT *From; // pointer of the contour begin point in
                            // base contour arc
   D2VPOINT *To;    // pointer of the contour end point in
                            // base contour arc
};
```

APPENDIX B. STRUCTURE OF 'ONE LINE WITH THREE LAYERS'

```
class LINE{                          //linear object
   NODE        *Begin;              // begin node pointer
   NODE        *End;                // end node pointer
   VirtualARC *LeftARCs[ ];       // array of left arc pointer
   VirtualARC *RightARCs[ ];      // array of right arc pointer
```

```
    ARC             MiddleARCs[ ];    // array of middle arc
};

class VirtualARC {                  // virtual arc
    NODE         *Begin;            // begin node pointer
    NODE         *End;              // end node pointer
    ARC          *Arcs[ ];          // array of middle arc pointer
    POLYGON *AdjacentPolygon; // adjacent polygon pointer,
                              // left arc is left adjacent polygon,
                              // right arc is right adjacent polygon
};

class ARC {                         // real arc
    NODE       *Begin;              // begin node pointer
    NODE       *End;                // end node pointer
    POLYGON *LeftPolygon;    // left adjacent polygon
    POLYGON *RightPolygon; // right adjacent polygon
    POINT         Points[ ];        // pattern point of arc
};
```