

# ATL/COM 을 이용한 3 차원+Temporal 객체 컴포넌트 구현

이현아\*, 임헌기\*, 남광우\*, 류근호\*, 이종훈\*\*

\*충북대학교 데이터베이스연구실

\*\*한국전자통신연구원

e-mail : \*{halee, hklim, kwnam, khryu}@dbalb.chungbuk.ac.kr, \*\*jong@etri.re.kr

## Implementation of 3D+Temporal Object Components Using ATL/COM

Hyun Ah Lee\*, Hun Ki Lim\*, Kwang Woo Nam\*, Keun Ho Ryu\*, Jong Hun Lee\*\*

\*Dept. of Computer Science, Chungbuk. National University

\*\*ETRI

### 요 약

GIS 기술 그룹과 데이터베이스 그룹간의 상호 연관된 연구의 필요성이 인식되면서, 시공간의 개념에 대한 동시 지원이 요구되고 있다. 또한 각 시스템간의 데이터 모델이 매우 다르기 때문에 지금까지 분산되어 있던 방대한 양의 공간 데이터에 대한 상호 운용성의 필요성이 증가함에도 불구하고 시스템의 통합 및 표준화의 진행이 어렵다. 이 논문에서는 분산된 공간 데이터의 상호 운용성을 제공하고 공간과 시간의 개념을 동시에 지원하기 위해 3 차원 시공간 컴포넌트를 설계하고 이를 ATL/COM 을 사용하여 구현함으로써 COM 컴포넌트의 장점인 확장성과 재사용성을 그대로 수용하였다. 상호 운용성을 위한 통합 데이터 모델로서 OGC 의 2 차원 표준 모델을 3 차원과 시간 영역으로 확장 시키고, 이를 기반으로 COM 컴포넌트 인터페이스를 설계하였다. 또한 공간 데이터를 WKB 구조로 정의하여 저장함으로써 시스템간의 이식성을 보장한다.

### 1. 서론

최근에는 GIS 기술 그룹과 데이터베이스의 그룹간의 상호 연관된 연구의 필요성이 인식되고 있다. 또한 지리 정보 시스템, 통신 선로 시스템, 국토 관리 시스템, 디지털 지도 제작, 과학 데이터베이스, 의료 관리 시스템 등과 같은 여러 분야의 응용 업무들이 시간과 공간 개념을 포함하고 있으며, 시간 및 공간 개념의 동시 지원이 요구되고 있다.

그러나 지금까지 시간 데이터와 공간 데이터를 지원하는 데이터베이스를 각각 분리하여 연구해 왔으므로 데이터베이스의 설계, 질의 언어, 저장구조, 질의 처리에 관련된 연구 결과를 시공간 데이터베이스에 직접 적용할 수 없다. 또한 데이터 저장 매체의 사용에서 시간과 공간 데이터를 포함하는 최근의 많은 응용들은 전형적인 기업 응용에 근거한 현재의 데이터베이스 기법이 시공간 데이터를 저장하고 접근하기에는 불충분하다는 점을 명백하게 보여주고 있다. 따라

서, GIS 소프트웨어에서 시간과 공간 개념을 갖는 데이터를 효율적으로 저장하고 조작하기 위해 시공간 데이터베이스의 개발이 필요하다[Salt00, 김동호 98].

지금까지의 지리 정보 시스템의 기술 연구는 주로 지리 정보 서비스 업체 또는 연구소를 중심으로 특정 응용만을 목적으로 개별적이며 부분적인 연구개발이 진행된 것이 현실이다. 각각의 연구 목적에 따라 아주 상이하여 다른 목적에 의한 시스템의 통합 또는 표준화를 수행할 수 없는 문제점을 표출하고 있다. 따라서 현재 분산된 공간 데이터에 상호 운용성을 제공하기 위한 기술의 필요성이 증가하고 있다.

이렇게 분산되어 있는 공간 객체 데이터를 통합하여 표현하기 위한 데이터 모델로서 OpenGIS 명세서에 제안하는 표준 2 차원 공간 모델을 기반으로 3 차원 시공간 모델을 이 동 객체와 이력 객체[G01i98]로 표현할 수 있는 3 차원 시공간 객체의 클래스 모델[이현아 00]을 제안하였다.

이 논문의 목적은 현실 세계의 시간과 공간 개념이 동시에 요구되는 데이터들을 효율적으로 관리할 수 있는 4D 시공간 데이터 제공자 컴포넌트를 개발하는데 있다. 앞서 제안한 3 차원 시공간 모델을 바탕으로 시공간 객체의 컴포넌

이 연구는 한국전자통신 연구원의 "4D 시공간 데이터 제공자 컴포넌트 개발" 위탁과제의 연구비 지원에 의해 수행되었음.

트의 인터페이스를 설계하고, COM 컴포넌트를 사용하여 데이터 제공자에서 사용할 수 있는 시공간 객체 컴포넌트를 구현한다.

2 장에서는 3 차원 시공간 데이터 모델 및 통합 데이터 모델에 관한 관련 연구를 기술하고, 3 장에서는 3 차원 시공간 클래스 모델에 대해 기술한다. 4 장에서는 클래스 모델을 바탕으로 컴포넌트의 인터페이스를 설계하고, 시공간 객체 데이터의 저장 구조에 대한 설명을 한 다음, 5 장에서 이에 대한 구현 예를 보여준다. 마지막으로 6 장에서 위의 내용을 요약하고 현재 진행중인 연구를 기술한다.

2. 관련 연구

2.1 시공간 객체 지향 데이터 모델

Worboys 는 객체 지향 개념을 시공간 모델링에 최초로 도입하였다. 그리고 공간 데이터 모델을 시공간 데이터 모델로 일반화 시키는 방법을 사용하였다[Worb94]. 여기서, 공간 객체들은 두 종류의 시간적 측면과 관련이 있고, 또한 질의를 위한 연산들의 집합이 제공된다. 그는 시공간 객체(ST-object)의 개념을 도입하였다. 여기서 시공간 객체는 공간과 이원시간 영역을 모두 가지고 있는 유일한 객체로서 정의한다. 이밖에 Simplex 라고 불리는 공간 객체인 point, finite straight line segment 또는 triangular area 가 있고, 이 Simplex 가 순서화 된 쌍인 bitemporal element 와 함께 ST-simplex 를 이룬다. ST-simplex 들의 유한 집합을 ST-complex 라고 하고, 이 ST-complex 상에서 질의 대수가 제안되었다. Union, intersection, difference 그리고 temporal projection 들도 이 ST-complex 상에서 제공된다. 그러나 제한된 연산 집합을 갖는 단일 타입을 사용함으로써 다양한 객체의 표현에 제한을 갖게 되고, 이동 객체에 대해 고려하지 않았다.

2.2 이동 객체를 위한 ADT 기반 데이터 모델

이 접근 방식은 시공간 객체 추상형 데이터 타입(ADT)을 정의하고 공간 객체 타입을 ADT 형태로 기존 데이터베이스를 확장한 공간 데이터베이스 응용에서 기인한다.

[For00]에는 연속적으로 움직이는 객체에 대한 표현을 위하여 한 값의 시간에 따른 변화를 단편이라는 조각들로 재구성하여 이 단편 안에 값의 변화를 단순한 형태의 함수로써 표현하는 단편화 표현(slice representation) 개념을 설명한다. 이 단편들은 UNIT 타입과 MAPPING 타입에 의해 표현된다. 이 밖에도 시간에 따라 위치가 변하는 이동 라인 타입과 위치와 모양 또는 영역이 변하는 이동 다각형 타입에 대하여 설명하고 있다. 이 접근 방식을 이용하여 불연속 객체에 대한 타입 정의도 가능하지만 기존의 방법과 비교하여 모델의 표현력 및 효율의 측면에 대해 언급하지 않았다. 또한 이 모델은 데이터 타입 접근 방법으로 객체지향 모델링과는 접근 방법에서 차이가 존재한다.

2.3 시스템간 상호 운용성을 위한 통합 데이터 모델

공간 데이터에 대한 방대한 양의 데이터는 각 시스템에 따라 다른 포맷과 접근 방식을 갖기 때문에 분산되어 있는 GIS 자원들을 다른 시스템에서 이용하기 위해서는 데이터를 해당 포맷에 맞춰 변환하는 과정이 필요하다. 최근에는 OLE/COM, CORBA 등의 분산 객체 기술로 사용자에게 데이터스에 대한 투명성을 제공해주는 연구가 진행되고 있다. 이를 위한 첫 단계는 표준 데이터 접근 모델의 정의이다. 대표적인 예로 OGC 의 OpenGIS 명세서가 있다. 현재 OLE/COM, CORBA, SQL 을 기반으로 하는 세 종류의 명세서가 공개되어 있으며, 데이터 모델 뿐만 아니라 기술 별

관련 인터페이스들의 구현 명세를 제시하고 있다. OLE/COM, CORBA 기반의 OpenGIS 명세서에 제시된 데이터 제공자 컴포넌트는 구현된 바가 있다([Choi00], [안경환 00]). 그러나 OpenGIS 명세서에서 제공하는 데이터 모델은 2 차원 공간으로 한정되어 있고, 이동 객체를 위한 연산은 고려되어 있지 않다.

이와 같이 지금까지의 연구는 객체의 표현에 대한 다양성의 문제나 공간적 제약을 가지고 있으므로, 이 논문에서는 3 차원 좌표로 표현되는 다양한 형태의 객체의 타입을 정의하고 더 나아가 이를 시간의 변화에 따라 처리할 수 있는 컴포넌트를 개발함으로써 객체의 다양한 시공간적 표현 뿐만 아니라 데이터의 이식성 및 상호 운용에 대한 문제를 해결한다.

3. 3 차원 시공간 객체 클래스 모델

지금까지 OpenGIS 에서 제안된 공간 데이터 모델은 2 차원으로 제한되어 있고, 3 차원 데이터 모델의 표준은 아직 존재하지 않는다. 여기서는 OLE/COM 을 위한 OpenGIS 명세서([OGIS99])의 2 차원 공간 모델을 확장하여 3 차원 공간을 표현하는 방식을 선택했고, 이에 시간 클래스를 삽입하여 3 차원 공간 객체 모델을 다시 3 차원 시공간 객체 모델로 확장하였다[이현아 00]. 그림 1 은 각 추상 클래스들의 상속 관계와 GBox 에 대한 3 차원 시공간 데이터 모델을 보여 주고 있다.

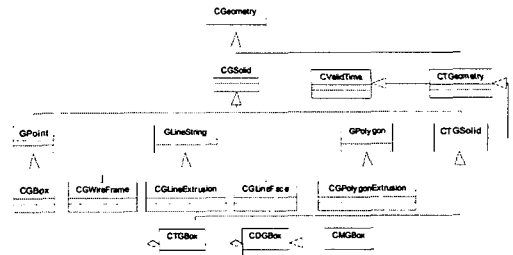


그림 1. GBox 의 3 차원 시공간 객체 모델

GSolid 는 3 차원 지리 객체 추상 클래스이며, Geometry 를 상속 받는 하위 클래스이다. 3 차원 좌표로 구성되는 GPoint, GLineString, GPolygon 의 하위 클래스를 갖는다. GBox, GCone, GCylinder, 그리고 GSphere 은 GPoint 의 집합으로 표현된다. GWireFrame 은 피쳐의 투시도를 나타낼 경우 그 외곽선을 나타내기 위한 선 객체이고, GLineExtrusion 은 GLineString 상에 존재하는 점들을 중심으로 반지름을 갖는 튜브의 형태이며, GLineFace 는 GLineString 상에 존재하는 점들을 중심으로 일정한 범위의 버퍼를 갖는다. GPolygonExtrusion 은 내부링을 갖지 않는 Polygon 의 경계상의 점들로부터 일정한 높이를 갖는 3 차원 공간 객체이다.

3 차원 시공간 객체를 구성하기 위해 3 차원 공간 클래스에 ValidTime 클래스를 삽입하여 이를 상속 받는 TGeometry 와 TGSolid 를 정의한다. 특히 TGSolid 클래스는 TGeometry 와 GSolid 클래스를 이중 상속 받는 시공간 객체에 대한 추상 클래스이다. TGBox 는 유효시간을 상속 받는 GBox 객체이고, TGBox 의 순열로써 DGBBox 가 구성되어 시간의 변화에 따른 GBox 의 변화를 나타낸다. 이외의 GCone, GCylinder, GSphere, GWireFrame, GLineExtrusion, GLineFace, 그리고 GPolygonExtrusion 역시 GBox 와 마찬가지로 각각 관계되는 하나의 타임 스탬프를 갖는 시공간 객체들의 순차적 집합으로 각 객체의 시간상의 변화를 표현할 수 있다. 3 차원 시공

간 객체 모델에서의 이동 객체의 종류는 MGBox, MGCone, MGSphere, MGCylinder 등이 있고, 이동 객체의 연속적인 3차원적 이동을 표현한다. 현재 제안한 모델에서 3차원 시공간 객체의 연속적인 형태나 모양의 변화는 고려하지 않았지만, 확장을 통해 표현이 가능하다.

#### 4. 3차원 시공간 객체 컴포넌트

##### 4.1 인터페이스 정의

COM 컴포넌트의 내부는 은폐되어 있고, 시스템은 오직 노출된 인터페이스를 통해서만 컴포넌트에 접근하게 된다. 따라서 컴포넌트를 설계하는 것은 곧 인터페이스를 설계하는 것이라고 말할 수 있다. 추상 클래스인 Geometry, GSolid, ValidTime, TGeometry 그리고 TGSolid 컴포넌트는 인터페이스 상속을 통해 각 하위 컴포넌트에 공통적인 메소드들을 제공하기 때문에, 3차원 시공간 객체의 컴포넌트를 구성하기 위해 추상 클래스들에 대한 인터페이스들을 먼저 정의해야 한다. 클래스 모델에서의 상속 관계는 인터페이스 간의 상속으로 표현되고, 다른 인터페이스를 상속 받은 하위 인터페이스는 상위 인터페이스의 메소드를 사용할 수 있다. COM 컴포넌트는 상속 받은 메소드들을 모두 구현해야 하고, 이때 실행 코드는 하위 인터페이스를 노출시키는 컴포넌트에서 해당 컴포넌트의 특성에 맞추어 작성된다.

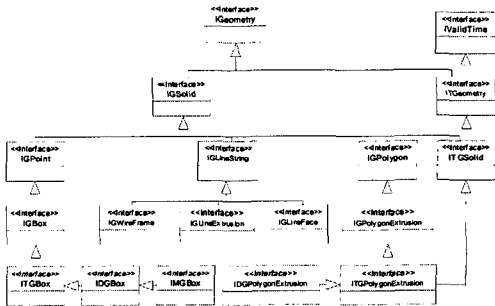


그림 2. 3차원 시공간 객체 컴포넌트의 인터페이스

그림 2는 각 클래스를 구현한 컴포넌트들 간의 인터페이스 상속 관계를 보여준다. 추상 클래스의 인터페이스 상속 관계는 클래스 모델에서의 상속 관계와 유사하다. IGeometry는 IGSolid와 ITGeometry의 두 인터페이스에 상속되고 각 하위 인터페이스는 IGeometry의 메소드들을 상속 받는다. IGSolid는 3차원 좌표계에 대한 공간 연산자들의 메소드를 갖는다. 특히 유효 시간을 표현하거나 시간에 관련된 모든 공통 연산은 IValidTime의 메소드로 정의되고, ITGeometry는 IValidTime 인터페이스를 이중 상속 받음으로써 시간에 관련된 메소드들을 자신의 하위 인터페이스인 ITGSolid로 넘겨준다. ITGSolid는 IGSolid를 이중 상속 받아 시공간 객체 컴포넌트의 인터페이스에 공간 속성을 나타내는 메소드들과 시간 속성을 나타내는 메소드들을 넘겨주게 된다.

IGSolid는 IGPoint, IGLineString, IGPolygon에 상속되고, IGPoint는 IGBox, IGCone, IGCylinder, IGSphere를, IGLineString은 IGWireFrame, IGLineString, IGLineFace를, IGPolygon은 IGPolygonExtrusion에 상속된다. 각 하위 인터페이스는 각 컴포넌트에 해당하는 객체를 구성하기 위한 메소드들을 갖는다. 또한 3차원 객체를 나타내는 인터페이스들과 ITGSolid를 이중 상속함으로써 ITGBox, ITGCone, ITGCylinder, ITGSphere, ITGWireFrame, ITGLineString, ITGLineFace, 그리고 ITGPolygonExtrusion이 구성된다. 각 컴포넌트

는 하나의 타임 스탬프와 하나의 객체의 쌍으로써 객체를 생성하기 위한 메소드들을 포함한다. IDGBox는 ITGBox의 인터페이스를 상속함으로써 공간에 관련된 메소드와 시간에 관련된 메소드들을 모두 사용할 수 있다. 또한 TGBox의 순열과 그에 대한 인덱스를 구성하기 위한 메소드를 가지고 있으며 IMGBox의 인터페이스에 상속된다. IDGCone, IDGCylinder, IDGSphere, IDGWireFrame, IDGLineExtrusion, IDGLineFace, 그리고 IDGPolygonExtrusion도 IDGBox와 같은 구성의 상속 관계를 갖는다. 단 이동 객체가 존재하지 않는 IDGWireFrame, IDGLineExtrusion, IDGLineFace와 IDGPolygonExtrusion은 하위 인터페이스가 없다.

그림 3은 CDGPolygonExtrusion 컴포넌트의 인터페이스 설계를 보여주고 있다. DGPolygonExtrusion은 TGPolygonExtrusion의 순열로 표현되고, TGPolygonExtrusion은 하나의 타임 스탬프와 하나의 공간 데이터로 구성된다. 따라서 ITGPolygonExtrusion은 ITGSolid로부터 상속 받은 메소드 get\_from(), get\_to()를 사용하여 시간 값을 설정하는 시간 관련 메소드와 IGPolygonExtrusion으로부터 상속 받은 메소드로부터 GPolygonExtrusion 객체를 구성하는 공간 관련 메소드를 갖는다. GPolygonExtrusion 객체는 GPolygon과 높이 값을 설정하는 메소드들을 통해 표현될 수 있고 이를 위해 IGPolygonExtrusion를 통해 관련 메소드를 상속 받는다. IDGPolygonExtrusion은 ITGPolygonExtrusion를 통해 상속 받은 연산자를 사용하여 TGPolygonExtrusion을 구성하는 메소드와 인덱스를 구성하는 메소드를 갖는다. 그 외에 3차원 시공간 데이터에 대한 기하 연산자의 메소드를 지원하는 ISTOperation4D 인터페이스와 위상 연산자를 지원하는 ISTRelation4D 그리고 저장 공간에 있는 데이터의 처리와 관련된 메소드를 갖는 IWks 인터페이스를 제공한다.

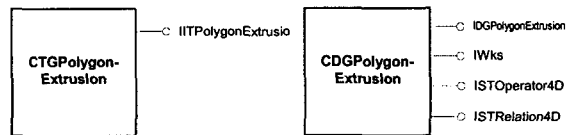


그림 3. CDGPolygonExtrusion 컴포넌트

그 외에 3차원 공간 데이터를 위한 기하 연산자를 지원하는 ISpatialOperation3D, 위상 연산자를 지원하는 ISpatialRelation3D가 3차원 공간 객체의 인스턴스 컴포넌트에서 제공된다.

##### 4.2 지리 객체의 저장 방식

각 지리 객체들은 WKB(Well-known Binary) 형식으로 저장 공간에 저장된다[OGIS99]. WKB는 연속적인 바이트의 스트림으로써 시공간 데이터 값을 표현하는 것으로 이것은 다른 시스템간의 이식성을 보장해 준다. 예를 들어 SQL 데이터 베이스에 저장된 시공간 데이터를 ODBC 클라이언트에서 사용할 수 있는 것이다. WKB에서는 Unsigned Integer와 Double의 두 가지 데이터 타입이 사용되고, XDR이나 NDR의 두 종류의 타입으로 저장될 수 있다.

WKB의 구성은 OpenGIS 명세서에 제안된 2차원 공간 데이터의 구성을 3차원 시공간 클래스 모델에 제안된 각 객체의 구조로 확장하는 방식을 사용하였다. 이에 대한 구조는 WKB의 헤더 파일에 정의되어 있으며, 그림 4는 각 객체 중 DGPolygonExtrusion의 WKB 구조의 정의 부분을 보여주고 있다.

WKB의 해석 및 구성은 IWks 인터페이스를 통해서 이루어진다. 즉 사용자가 요청한 시공간 데이터에 대한 관련 WKB는 IWks의 메소드를 통해 해석되어 객체로 구성된 후

사용자에게 제공되고, 입력되는 시공간 데이터는 IWks의 메소드를 통해 WKB로 구성되어 저장 공간으로 넘겨진다. 따라서 각 인스턴스 컴포넌트의 IWks는 각 객체의 정의된 형식에 맞춰서 컴포넌트 별로 실행 코드가 구현된다.

```

--
struct wks3DPolygon {
    uint32          numRings;
    LinearRing3D*  rings;
};
struct wks3DGPolygonExtrusion {
    uint32          height;
    wks3DPolygon   3DPolygon;
};
struct wks4DTGPolygonExtrusion{
    VTime          vtime;
    wks3DGPolygonExtrusion
    3DGPolygonExtrusion;
};
--
struct WKB3DGPolygonExtrusion {
    byte           byteOrder;
    uint32         wkbType;
    uint32         height;
    wks3DPolygon   3DPolygon;
};
struct WKB4DTGPolygonExtrusion{
    byte           byteOrder;
    uint32         wkbType;
    VTime          vtime;
    uint32         num_wkbTGPolygonExtrusion;
    wks4DTGPolygonExtrusion*
    4DTGPolygonExtrusions;
};
    
```

그림 4. DGPolygonExtrusion의 WKB 정의

5. 구현 예

다음은 3차원 시공간 객체 컴포넌트를 사용하여 3차원 시공간 객체의 WKB를 해석하여 출력하는 간단한 테스트 프로그램이다. 테스트 환경의 운영체제는 MS 윈도우 2000 서버이고, 프로그래밍 툴은 Visual C++ 6.0을 사용하였다. 데이터 제공자는 OLEDB provider를 사용하고, 데이터베이스는 SQL 서버 7.0을 사용하였다. 테스트에 사용된 데이터는 GPolygonExtrusion 타입인 AptF, BankF, BuildingF, RestaurantF 테이블과 DGPolygonExtrusion 타입인 PublicOfficeF 테이블로 구성되고, WKB 데이터는 ODBC 기반의 프로그램을 통해 생성하였다.

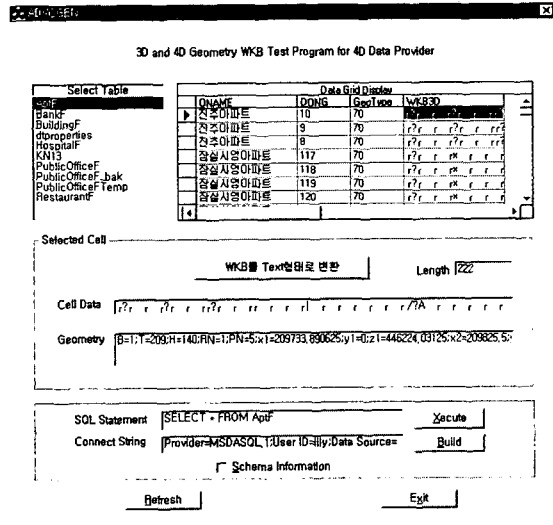


그림 5. 4D COM 컴포넌트를 사용한 OLEDB Consumer

테스트 프로그램의 동작은 다음과 같다. 일단 데이터베이스에 접속하면, 선택된 카탈로그에 존재하는 테이블 리스트가 생성되고, 테이블 이름을 선택하면 테이블에 존재하는 모든 데이터가 출력된다. 해석하고자 하는 시공간 속성의 데이터를 선택한 후 변환을 수행하면, WKB의 첫 부분에 위치하는 스트림의 저장 방식과 객체의 타입을 판별한 다음 해당 타입 컴포넌트의 IWks 인터페이스를 통해 WKB의 크기를 가져오기 위한 get\_WKBSize()와 WKB를 숫자형 데이터로 변환하기 위한 ImportFromWKB() 메소드가 호출된다. 변환된 데이터는 Geometry 공간에 출력된다.

6. 결론

현재까지 축적된 방대한 양의 공간 데이터를 효율적으로 사용하기 위해, 지금까지 시스템에 따라 개별적으로 사용되어 왔던 데이터의 포맷이나 처리 방법에 대한 상호 운용성을 요구하게 되었다. 이 논문의 목적은 현실 세계의 시간과 공간 개념이 동시에 요구되는 데이터들을 효율적으로 표현하고 관리하기 위한 통일된 인터페이스를 제공하는 3차원 시공간 객체 컴포넌트를 개발하는 데 있다. 이를 위해 먼저 데이터의 통합을 위한 3차원 시공간 데이터 모델을 제안하였고, 이를 기반으로 하여 시공간 객체 컴포넌트의 인터페이스 설계가 이루어 졌다. 2차원 표준 모델과의 호환성을 유지하기 위해 OGC에서 제안한 OpenGIS 명세서를 확장하는 방법을 사용했고, 공간 저장 방식인 WKB의 구조를 3차원 시공간으로 확장하여 재정의 하였다.

현재 3차원 시공간 객체 컴포넌트에 관련된 시공간 연산자의 정의가 진행 중이며, 완료 후 이를 시공간 객체 컴포넌트에 추가할 것이다. 그 다음 완성된 컴포넌트를 기반으로 하여 4D 시공간 데이터 제공자를 개발할 예정이다. 이를 위해 시공간 객체 컴포넌트와 시공간 연산자의 개발 이외에 시공간 인덱스와 질의 처리기가 연구 중이다.

참고문헌

[Choi00] Haeock Choi, Kwangsoo Kim, Jonghun Lee, "Design and Implementation of Open GIS Component Software", Geoscience and Remote Sensing Symposium 2000: Proceeding of IGARSS 2000, 2000

[Forl00] L. Forlizzi, B.H. Güting, E. Nardelli, and M. Schneider, "A Data Model and Data Structures for Moving Object Databases", SIGMOD Conference 2000: 319-330, 2000.

[Güting98] R. H. Güting, M.H. Bohlen, M. Erwig, C.S. Jensen, et al, "A Foundation for Representing and Querying Moving Object", Chorochronos TR, CH-98-3, 1998.

[OGIS99] Open GIS Consortium, Inc. OpenGIS, "Simple Features Specification For OLE/COM Revision 1.1", OpenGIS Project Document 99-050, 1999.

[Salt00] S. Saltinis, C.S. Jensen, and S.T. Leutenegger, "Indexing the Positions of Continuously Moving Object", SIGMOD Conference 2000: 331-342, 2000.

[Wor94] M.F. Worboys, "A Unified Model for Spatial and Temporal Information", The Computer Journal, Vol. 37, No. 1, pp.25-34, 1994.

[김동호 98] 김동호, 류근호, "관계형 시공간 데이터베이스 질의언어와 연산", 한국 정보처리학회 논문지 제 5 권 제 10 호, 1998.

[안경환 00] 안경환, 조대수, 홍봉희, "CORBA를 이용한 OpenGIS 기반 미들웨어 구현", 개방형 GIS 연구회 논문지, 제 1 권 제 1 호 pp.19-28, 1999

[이현아 00] 이현아, 임헌기, 김영일, 남광우, 류근호, "3D+Temporal 시공간 객체 모델링", 한국 정보처리학회 논문지 제 7 권 제 2 호, 2000.