

3차원+Temporal 객체를 표현하기 위한 컴포넌트 설계 및 구현

이현아^{1*} · 김진석¹ · 류근호²

Design and Implementation of Components to Represent the 3-Dimensional+Temporal Objects

Hyun Ah LEE^{1*} · Jin Suk KIM¹ · Keun Ho RYU²

요 약

지리 객체는 3차원 공간으로 구성되기 때문에 정확한 표현을 위하여 3차원 좌표계를 사용하여야 한다. 이 논문에서는 3차원+temporal 객체 모델을 정의하기 위해 2차원 공간으로 제한되어 있는 개방형 지리 데이터 모델을 3차원 공간과 유효 시간으로 확장하였다. 이러한 모델 설계 방법은 개방형 모델을 사용하는 타 시스템과의 호환성을 보장한다. 또한 이력 객체와 이동 객체를 함께 고려하여 두 종류의 시공간 객체에 대한 관리를 가능하게 하였다. 여기에서 제안된 3차원+temporal 객체 모델은 OLE/COM 기법을 사용한 3차원+temporal 객체 컴포넌트로 구현하였다. 이를 위해 제안한 객체 모델을 기반으로 컴포넌트의 인터페이스 구조를 정의하였다. 이 컴포넌트의 사용 예를 보이기 위해 3차원 시공간 데이터를 대상으로 하는 데이터 소비자를 구현하였다. 이와 같은 컴포넌트는 기존의 시스템에 제안된 모델을 적용하기 위한 비용과 노력을 최소화하는 해결책으로 사용될 수 있다.

주요어: 3차원 시공간 객체, 데이터 모델, 컴포넌트, GIS

ABSTRACT

As geographic object is consisted of 3-dimensional, it must be appropriated use the 3-dimensional coordinate system to express for exact representation. To define 3-dimensional+temporal object model

2004년 11월 18일 접수 Received on November 18, 2004 / 2004년 12월 24일 심사완료 Accepted on December 24, 2004

¹ 한국전자통신연구원 우정기술연구소 u-Post연구팀 u-Post Research Team, Postal Technology Research Center, Electronics and Telecommunications Research Institute(ETRI)

² 충북대학교 전기전자컴퓨터 공학부 School of Electronical & Computer Engineering, Chungbuk National University

* 연락처 E-mail : halee@etri.re.kr

we extended 3-dimensional space and valid time from object model of OpenGIS consortium that is limited to 2-dimensional space. This methodology guarantees compatibility with other systems to construct with the open model. Also, it permitted administration for two kinds of object by considering both objects whose position and shape changes discretely over time and objects whose position changes continuously. 3-dimensional+temporal object model was implemented by 3-dimensional+temporal object component using OLE/COM techniques. The interfaces of the component defined based on 3-dimensional+temporal object model. To use this component, we implement the data consumer for 3-dimensional+temporal data.

KEYWORD : 3-Dimensional + Temporal Objects, Data Model, Component, GIS

서 론

실세계의 다양한 개체를 대상으로 하는 시스템들은 점차 대량화되어 가는 데이터를 효율적으로 관리하기 위해 데이터베이스와 연계하여 개발되고 있다. 기존의 관계형 데이터베이스는 지금까지 텍스트 기반의 데이터 관리에 대한 요구 사항들을 충족시켜주었으나, 공간 혹은 시간과 같은 복잡한 구조의 데이터를 표현하기에는 시스템에서 지원되는 데이터 타입과 질의어들과 연산자에 한계가 있었다. 따라서 최근에는 GIS 기술 그룹과 데이터베이스 그룹간의 상호 연관된 연구의 필요성을 인식하고 이를 통해 사용 목적에 최적화된 시스템을 구축하고 있다.

그러나 여러 분야의 응용 업무들이 시간과 공간의 개념을 동시에 갖는 데이터를 대상으로 하고 있기 때문에 공간과 시간으로 분리되어 연구되어왔던 데이터베이스의 개념을 시공간 데이터에 그대로 적용하기는 어렵다. 그러므로 GIS 소프트웨어에서 시공간 데이터를 효율적으로 저장하고 조회하기 위해 시공간 데이터베이스의 개발이 필요하며(Saltenis 등, 2000), 이를 위해 적절한 시공간 데이터 모델링이 선행되어야 한다.

따라서, 국제 표준화 기구로써 조직된 개방형 GIS 컨소시엄에서는 개방형 지리 자료 모델, OGIS(OpenGIS) 서비스 모델, 정보통신 모델을

제안하여 시스템 및 제도적 상호 운용성을 제공하고 있다. 그러나 개방형 GIS 컨소시엄에서 제시한 지리 자료 모델은 2차원 공간 모델이며, 3차원 공간 및 시간으로의 확장은 아직 언급되어 있지 않았다.

이 논문 안에서 시공간 모델이 제안되더라도 기존의 시스템에 제안된 모델을 적용하려면 시스템의 전반적인 재구축이 불가피하며, 이러한 과정은 비용과 노력의 낭비를 가져온다. 시스템 재구축 시 각 요소들의 기능을 활용하고 비용과 노력을 최소화하기 위한 해결책으로 컴포넌트 기법이 사용된다.

따라서 이 논문에서는 이 컴포넌트 기법을 이용하여 시간과 공간을 함께 지원하는 3차원+temporal 객체 모델을 설계하고 구현한다.

제안된 모델은 GIS 시스템에서 최근 이슈가 되고 있는 데이터의 상호 운용성을 보장하기 위해 개방형 GIS 컨소시엄의 2차원 공간 객체 모델을 기반으로 각각 3차원과 유효 시간으로 확장한다. 그리고, 시간에 따라 공간 속성에 대한 변화의 연속성을 기준으로 시공간 객체를 이력 객체와 이동 객체로 분류(Tryfona 등, 1998)하여 설계한다.

3차원+temporal 객체 모델을 설계할 때 이력 객체와 이동 객체를 나누어 정의하는 이유는 토지에 관한 데이터와 차량에 대한 데이터를 예로 들어 설명할 수 있다.토지의 경우 다소 빈번한 갱신을 통해 쉽게 객체의 위치나 영역의 변화를

데이터베이스에서 유지하고 다룰 수 있다. 그러나 자동차의 경우 데이터베이스 갱신을 통해 관리할 경우 갱신이 너무 빈번하게 발생하거나, 기록이 시간에 뒤지거나, 갱신되는 데이터가 부정확하게 되는 등의 문제가 발생할 수 있다 (Saltenis 등, 2000). 이러한 문제점을 해결하기 위하여 이력 객체와 이동 객체를 함께 효율적으로 처리하기 위한 적절한 방법을 이 논문에서 제시하고 있다.

개방형 모델

이 논문에서 제안하고 있는 3차원+temporal 객체 모델은 국제 표준화 기구인 개방형 GIS 컨소시엄의 2차원 공간 모델(OpenGIS, 1999)을 기반으로 한다. 이는 개방형 모델을 따르는 다른 시스템들과의 호환성을 제공해준다.

개방형 GIS 컨소시엄에서 제안하고 있는 2차원 공간 모델의 특성은 다음과 같다.

- 공간 객체는 점들의 집합으로 구성된다.
- 지리 객체를 위한 추상 클래스인 Geometry를 정의하고 Geometry에서 모든 공간 연산자를 지원한다.
- Geometry의 하위 객체는 point, curve, surface이며, 이 중 curve와 surface는 각각 1차원 객체와 2차원 객체를 위한 추상 클래스이다.
- 인스턴스화 될 수 있는 객체는 point, LineString, polygon이며 각 객체는 내부, 경계, 외부로 구성된다.
- 점은 경계가 없고 선은 양 끝점을 경계로 하며, 면은 시작점과 끝점이 동일한 선을 경계로 한다.
- 모델의 구성은 잘 알려진 타입들, 즉 int, double과 같은 타입의 리스트로 이루어진다.

최상위 계층인 Geometry는 차원적으로 확장

된 9 교집합 모델(DE-9IM)을 기반으로 정의된 공간 관계 분석 연산과 함께 위상 연산, 기타 공간 관련 연산을 정의하고 상속을 통해 공간 연산들을 하위 클래스에 제공한다.

그러나 이 모델은 2차원 공간으로 한정되어 있으므로 3차원+temporal 객체를 설계하기 위해서는 먼저 2차원 모델로부터 3차원 공간 모델로의 확장된 정의가 필요하다. 따라서 이 논문에서는 3차원 공간 모델을 제안하고 있으며 이는 다음절에 자세히 기술하고 있다.

3차원+temporal 객체 컴포넌트 모델 설계

1. 설계 원칙

3차원+temporal 컴포넌트를 설계하기에 앞서 기반이 되는 시공간 모델의 정의가 먼저 이루어져야 한다. 이 논문에서는 원칙에 따라 3차원+temporal 객체 모델을 정의하였다.

- 1) 3차원 공간에 유효 시간을 포함한 3차원 시공간 객체를 대상으로 한다.

객체는 3차원 유클리안 공간에서 유효시간과 함께 표현된다. 여기서 객체는 지리 객체뿐만 아니라 지리에 관련된 모든 범위의 데이터를 의미한다.

- 2) 객체 지향 개념을 도입한다.

객체 지향 데이터 모델은 객체, 클래스, 캡슐화, 상속 그리고 다형성 등의 객체 지향 패러다임에 기반 한다. 이 개념은 동일한 객체의 이력들을 하나의 단일 엔티티에 내장시키는 것을 가능하게 하고, 시공간 시스템에 대한 대부분의 객체지향 접근들은 시간이 하나의 차원인 3차원 공간 또는 3차원 시공간에서 추상구조를 작성하는데 기초가 된다.

3) 공간 모델로부터 시공간으로 확장한다.

시공간 모델을 정의하는 방법은 시간 모델로부터 시공간을 정의하는 방법과 공간 모델로부터 시공간을 정의하는 방법이 있다. 3차원+temporal 객체 모델은 공간 모델로부터 시공간으로 확장하는 방법을 사용하고 있다.

4) 이력 객체와 이동 객체를 동시에 지원한다.

3차원+temporal 객체 모델은 이력 객체와 이동 객체를 함께 고려한다. 이력 객체는 버전간의 데이터가 변하지 않고 이동 객체는 버전간의 데이터가 시간에 따라 계속 변한다는 특성을 갖는다. 따라서 이력 객체를 관리하는 방법으로 이동 객체를 관리할 경우 저장 비용, 연산 비용, 데이터의 부정확성 문제 등이 발생할 수 있으며, 반대로 이동 객체를 관리하는 방법으로 이력 객체를 관리할 경우 불필요한 저장 비용 및 연산 비용의 증가를 초래할 수 있다. 이 모델에서는 이력 객체를 정의하고 이력 객체를 상속받는 이동 객체를 정의한다. 이 때 이동 객체에 대한 버전간의 관계를 나타내기 위한 함수를 설계하고 이력 객체와 이동 객체를 위한 연산자의 구현은 각 인스턴스 객체 내에서 각각 이루어진다.

5) 데이터의 호환성과 이식성을 고려한다.

3차원+temporal 객체 모델은 표준 모델을 기반으로 사용함으로써 각 시스템간 데이터의 확장성과 이식성을 보장한다. 그리고 컴포넌트 형태로 구현함으로써 시스템의 유지 보수 및 구축 비용의 절감의 이점을 얻을 수 있다. 데이터의 논리적인 이식성뿐만 아니라 물리적인 이식성도 함께 고려되어야 한다. 데이터의 저장 형태는 WKB(well-known binary) 구조를 사용한다. WKB의 구조는 개방형 모델에서 제안한 WKB를 3차원+temporal 객체 모델의 인스턴스 객체에 맞도록 확장하여 설계된다. WKB는 일반적인 데이터 타입의 리스트로 구성되므로 시스템간의 데이터 이동 시 타입 호환성 및 인식을 보장한다(OpenGIS, 1999).

2. 3차원 객체 모델

3차원+temporal 객체 모델은 데이터의 호환성 및 이식성을 고려하기 위해 개방형 모델을 기반으로 확장하는 방식을 택하고 있다. 그러나 국제 표준화 기구인 개방형 GIS 컨소시엄에서 제시한 공간 모델은 2차원이고, 아직 3차원 공간에 대한 개방형 모델은 제시되어 있지 않다(OpenGIS, 2001). 따라서 이 절에서는 2차원 공간 모델로부터 3차원 공간 모델을 정의한다.

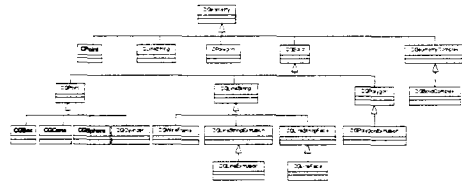


FIGURE 1. Class Architecture of 3-dimensional object

그림 1은 확장된 3차원 공간 모델(이현아 등, 2000)이다. 일반적인 공간 속성과 메소드를 가지고 있는 Geometry와 3차원 공간 속성과 관련 메소드를 가지고 있는 GSolid가 추상 클래스로 정의된다. GSolid의 하위 클래스는 GPoint, GLineString, GPolygon이 있다. 이 세 클래스는 (OpenGIS, 1999)의 2차원의 모델에서 제안한 구조와 동일한 관계를 갖지만 3차원 좌표로 표현된다. 이 세 가지 클래스의 하위에는 각각 GBox, GCone, GSphere, GCylinder와 GWireFrame, GLineStringExtrusion, GLineStringFace 그리고 GPolygonExtrusion이 존재하며 이 8개의 클래스가 인스턴스화 될 수 있는 3차원 공간 객체가 된다.

3차원 공간 객체의 경계는 다각형들의 집합으로 구성된다. 그러나 하나의 3차원 공간 객체를 저장할 때 경계를 구성하는 모든 점들을 저장하는 것은 저장 비용 및 데이터의 단순화 측면에서 비효율적이다. 따라서 각 공간 객체에 대한 기본 공간 요소들만 저장하게 된다.

1) GPoint 계열 객체 타입

GPoint의 하위 객체들은 하나의 GPoint를 갖고, 그 점을 기점으로 하는 공간 요소로 객체를 구성하게 된다.

- GBox : 직육면체, 중심점, 중심점을 기점으로 하는 가로, 세로, 높이 정보로 구성
- GCone : 원뿔형, 중심점과 반지름, 높이 정보로 구성
- GSphere : 구, 중심점과 반지름 정보로 구성
- GCylinder : 원통형, 중심점과 반지름, 높이 정보로 구성

2) GLineString 계열의 객체 타입

GLineString의 하위 객체들은 중심선이 되는 GLineString과, 선을 기점으로 하는 공간 요소로 객체를 구성한다.

- GLineStringExtrusion : 파이프형, 중심선과 중심선을 구성하는 점들이 중심이 되는 원의 반지름 정보로 구성. 중심선이 정확히 두 점만으로 이루어질 경우 GLineExtrusion으로 정의한다.
- GLineStringFace : 평면형, 중심선을 구성하는 점들로부터 일정한 넓이의 폭의 정보로 구성. 중심점이 정확히 두 점으로만 이루어질 경우 GLineFace으로 정의한다.
- GWireFrame: 실선형, 실세계 개체의 투시도를 그리기 위한 객체. 다른 인스턴스 객체들은 중심 객체와 기타 중요 공간 요소만으로 구성되지만, 상세한 건물의 투시도는 대상 개체의 주요 점 데이터를 모두 가지고 있어야 한다.

3) GPolygon 계열의 객체 타입

GPolygon의 하위 객체는 GPolygonExtrusion가 존재한다.

- GPolygonExtrusion: 형태는 밀면으로 하나의 GPolygon이고, 밀면을 구성하는 각 점들로부터 일정한 높이 값을 갖는 객체.

이 때 밀면인 GPolygon은 구멍이 없다고 가정한다.

그림 2는 위에서 설명한 3차원 공간 모델에서의 인스턴스 객체의 형태를 보여준다.

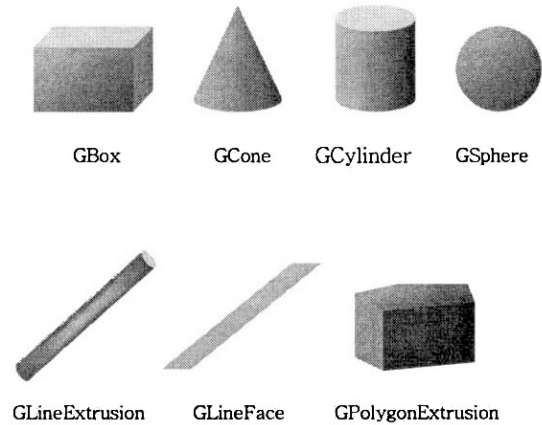


FIGURE 2. 3-dimensional object types

3. 3차원+temporal 객체 모델

3차원+temporal 객체 모델은 앞서 제안한 3차원 공간 모델에 Validtime이라는 유효시간에 대한 추상 클래스를 정의하여 공간 모델로부터 시간 차원으로 확장하여 설계된다(이현아 등, 2000).

하나의 공간 정보와 하나의 유효 시간을 갖고 있는 시공간 객체를 버전이라고 한다. 이러한 버전의 집합으로 시간에 따라 공간 속성이 변하는 공간 객체가 표현된다. 이 논문에서는 해당 객체의 버전간의 관계를 정의함으로써 이력 객체와 이동 객체를 각각 정의한다. 이 때 공간의 어떤 레벨이 시간 참조와 연관이 있는가를 명확히 결정해야 한다. 이에 대하여 (Worboys 등, 1990)에서는 공간과 시간 참조 병합에 있어서 다음과 같은 세 가지 접근 방법을 제시하였다.

- 1) 전체 지리 객체에 대한 타임스탬프 저장
- 2) 기본적인 공간 객체, 즉 point, string, polygon에 대한 타임스탬프 저장
- 3) point 수준에서의 타임스탬프 저장

첫번째 방법은 저장 구조 측면에서 가장 적은 비용이 들지만 다른 방법들에 비해 객체의 시간 특성에 대한 표현이 제한적이다. 두 번째 방법은 시간과 공간에 대한 병합을 기본적인 수준의 공간 객체에 할당하는 것으로 비싼 비용의 저장 구조를 요구하지만 객체에 대한 시간적 측면의 표현력은 비교적 풍부하다. 마지막 방법은 시간과 공간의 병합을 point 수준에서 하고 이것을 시공간 클래스들의 새로운 집합에 전파하는 방법으로 이 접근 방법은 자세하게 개발되어지고 있지 않다.

이 논문에서는 세 가지 시간과 공간의 병합 방법 중 첫번째 접근 방법을 선택한다. 3차원 공간 객체는 많은 점들의 집합으로 구성되므로 두 번째나 세 번째 접근 방법을 사용할 경우 저장 구조가 지나치게 복잡해지고 너무 비싼 저장 비용을 소비하게 된다. 복잡한 데이터 구조는 시공간 연산 비용을 증가시키는 원인이 된다.

그림 3은 지금까지 설명한 접근 방법들, 즉 3차원 공간 모델에 대한 객체 수준의 유효시간의 참조, 이력 객체와 이동 객체 등을 모두 고려하여 설계한 3차원+temporal 객체 모델의 클래스 계층도 이다.

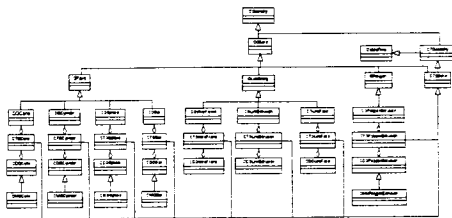


FIGURE 3. Class Architecture of 3-dimensional +temporal object

- 1) 추상 클래스
3차원+temporal 객체 모델은 인스턴스 객체

를 설계하기 전에 시간 및 공간 그리고 시공간에 대한 일반화된 개념을 제공하는 추상 클래스를 정의한다. 여기에서 정의된 추상 클래스는 가장 일반적인 공간에 대한 추상 클래스인 Geometry, 시간에 대한 추상 클래스인 Validtime, 3차원 공간 객체에 대한 추상 클래스인 GSolid, 시공간 객체에 대한 가장 일반화된 추상 클래스인 TGeometry, 마지막으로 3차원 시공간 객체에 대한 추상 클래스인 TGSolid이다.

이 다섯 가지의 추상 클래스 중 Geometry에 대한 정의와 관련 메소드들은 (OpenGIS, 1999)에 자세하게 설명되어 있으므로 여기에서는 언급하지 않는다.

가. Validtime 클래스

Validtime 클래스는 객체의 생성 시간과 소멸 시간을 나타내기 위한 from과 to의 두 가지 속성과 아래의 두 가지 기본 메소드들을 정의한다. 다음은 Validtime 클래스의 기본 메소드이며, 이때 v는 Validtime의 객체이다.

- v.IsPeriod() : v의 유효시간이 시점인지 기간인지를 판단한다. a.From < a.To일 경우 참을 반환한다.
- v.getInterval(version index As long) : v에 대해 주어진 버전에 대한 유효 시간을 가져온다.

나. TGeometry 클래스

각 하위 클래스에 공간과 시간의 통합 개념을 제공하기 위한 추상 클래스는 TGeometry이다. TGeometry는 Geometry와 Validtime 클래스로부터 각각 공간과 시간의 속성을 상속받아 일반적인 시공간 객체에 대한 개념을 정의하고 있으며 하위 클래스는 TGSolid가 있다.

TGeometry 클래스의 기본 속성은 시공간 차원을 나타내는 tDimension, 시공간적으로 비어있는 상태를 나타내는 tIsEmpty 그리고 시공간적으로 단척있는 상태를 나타내는 tIsSimple 이 있고 아래와 같은 기본 메소드를 갖는다. 여

기서 g 는 TGeometry 객체이다.

- (1) $g.tClone()$: g 의 복사본을 TGeometry 형으로 반환한다.
- (2) $g.tEnvelope()$: g 의 최소 시공간 경계 상자를 TGeometry 형으로 반환한다.
- (3) $g.tExtent3D()$: g 의 최소 시공간 경계 상자의 좌표 값을 반환한다.
- (4) $g.tsetEmpty()$: g 를 빈 TGeometry로 설정한다.

다. GSolid 클래스

3차원 공간 객체의 일반화된 개념을 제공하는 추상 클래스는 GSolid이다. 3차원 객체의 경계는 구멍을 갖지 않는 다각형들의 집합이다. 공간 객체의 경계는 항상 닫혀 있고 구멍을 갖을 수 있는 polygon과는 달리 내부 경계와 외부 경계로 구분하지 않는다. GSolid는 3차원 공간 객체를 위한 두 가지 기본 메소드를 갖고 있으며 아래에서 s 는 GSolid 객체이다.

- (1) $s.Area()$: s 의 경계를 구성하는 다각형들의 면적의 합을 반환한다.
- (2) $s.Volume()$: s 의 부피를 반환한다.

라. TGSolid 클래스

GSolid 클래스와 TGeometry 클래스를 이중 상속하여 3차원+temporal 객체를 위한 추상 클래스인 TGSolid 클래스를 정의한다. TGSolid 클래스를 통해 각 인스턴스 객체들이 시간의 속성과 공간의 속성을 모두 갖게 된다. 그림 3에서 인스턴스 객체들은 3차원 공간 클래스와 TGSolid 클래스를 이중 상속받는다. 따라서 상위에 존재하는 추상 클래스의 상속 경로에 대해 모호성이 발생하게 된다. 이 때 상속 경로를 명확히 하기 위해 추상 클래스에서 지원하는 모든 기본 메소드들은 TGSolid를 통해 정의되고, 각 공간 클래스에서는 공간 객체의 기본 메소드와 속성만을 상속한다. 다음은 TGSolid의 기본 메소드들이며, S 는 TGSolid 객체이다.

- (1) $S.tArea(version\ index\ As\ long)$: S 에

대해 주어진 버전에 대한 공간 속성의 경계 면적을 반환한다.

- (2) $S.tVolume(version\ index\ As\ long)$: S 에 대해 주어진 버전에 대한 공간 속성의 부피를 반환한다.

2) 이력 객체 클래스

이력 객체는 DGBox, DGCone, DGCylinder, DGSphere, DGWireFrame, DGLine-Extrusion, DGLineFace, DGPolygonExtrusion 이다. 이들을 DG 계열의 객체이며, 시간에 따른 공간 데이터 버전들의 집합으로 구성된다. 여기에서 D 는 불연속(discrete)을 의미한다. 각 시공간 객체의 버전을 표현하는 객체는 TGBox, TGCone, TGCylinder, TGSphere, TGWireFrame, TGLineExtrusion, TGLineFace, TGPolygonExtrusion이고 이들을 TG 계열의 객체라고 한다. TG 계열의 객체는 각 공간 객체와 CTSolid를 이중 상속받음으로써 하나의 공간과 하나의 유효시간으로 구성된다. DG 계열의 객체들은 이러한 TG 계열 객체들의 순열로 각 공간 객체에 대한 유효 시간상에서의 변화를 나타낸다.

3) 이동 객체 클래스

이동 객체는 공간 데이터가 시간에 따라 연속적으로 변화하고 있으며, 앞서 제시된 객체들은 시간에 따라 위치 정보가 불연속적으로 변하기 때문에 이동 객체를 표현하기에는 부적합하다(이현아 등, 20001). 따라서 이러한 이동 객체를 표현하기 위해 다른 제안이 필요하다. 연속적으로 이동하는 객체는 데이터베이스에 기록된 위치와 위치 사이에서도 계속 데이터가 변한다고 가정하며 이 점이 이력 객체와의 차이점이 된다.

3차원 이동 객체의 종류는 MGBox, MGCone, MGSphere, MGCylinder 등이 있으며, 이 때 이동 객체는 형태의 변화보다는 위치의 연속적인 변화를 표현하기 위하여 사용된다. 각 이동 객체는 이력 객체의 클래스를 상속 받아 정의되므

로, 이동 객체의 자료구조는 이력 객체와 동일하다. 즉 이동 객체는 해당 객체에 대한 버전들의 집합으로 저장된다. 따라서 버전간의 관계를 표현할 수 있는 함수를 정의하여 저장되지 않는 위치 정보를 추출할 수 있다. 이동 객체의 공간 속성은 하나의 중심점과 그 중심점으로부터의 각 공간 속성 값으로 구성되고, 이동 객체의 움직임을 표현하기 위해 실제로 이동하는 것은 중심점이다. 여기서는 이동 객체의 위치를 추출하기 위해 스플라인 함수를 이용하여 이동 객체의 연속적인 버전들에서 두 중심점 상의 구간을 보간하는 방법을 사용한다.

그림 4와 같이 3차원 시공간 상에서 자동차가 이동한다고 가정하고 이 자동차의 타입을 MGBox라고 하자. 시간에 따라 변하는 자동차의 위치를 구하기 위하여 x-t, y-t, z-t 평면에 대한 위치 추출함수 $p(t)$ 를 계산한다. 이 때 MGBox를 구성하는 각 버전들의 중심점은 $p(t)$ 함수상에 존재한다.

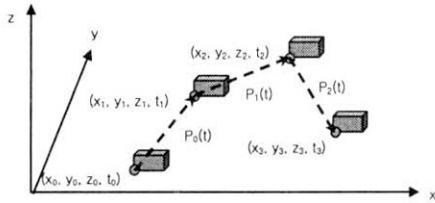


FIGURE 4. Movement of MGBox on 3-dimensional+temporal space

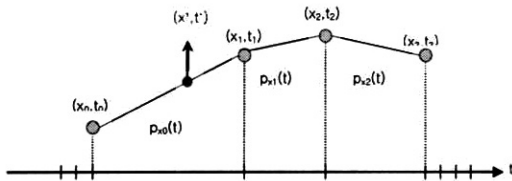


FIGURE 5. Component Structure of spatio-temporal object of GBox type

그림 5는 x-t 평면상에서 그림 4의 MGBox

의 중심점 이동을 보여준다. 이 때 점의 이동에 대한 1차 스플라인 함수는 [식 1]과 같다.

$$p_x(t) = \begin{cases} p_{x0}(t) = x_0 + d_0(t - t_0), & t \in [t_0, t_1] \\ p_{x1}(t) = x_1 + d_1(t - t_1), & t \in [t_1, t_2] \\ p_{x2}(t) = x_2 + d_2(t - t_2), & t \in [t_2, t_3] \end{cases} \quad (1)$$

단, $d_k = (x_{k+1} - x_k) / (t_{k+1} - t_k)$ 이고, $k = 0, 1, 2$ 이다.

$t_0 < t_1 < t_N$ 의 순서대로 나열되어 있다고 가정하고, 임의의 시간 t 가 주어지면, $t - t_{k+1} < 0$ 을 만족하는 최소 정수 $k+1$ 을 찾는다. 이 때 t 의 구간은 $[t_k, t_{k+1}]$ 이 되고, $t_k \leq t_1 \leq t_{k+1}$ 이 성립하는 k 를 알게 된다. 이로부터 위치 추출함수 $p_x(t)$ 는 [식 2]와 같이 표현된다.

$$p(t) = p_k(t) = (p_{xk}(t), p_{yk}(t), p_{zk}(t)), \quad t_k \leq t_1 \leq t_{k+1} \quad (2)$$

이 방법을 고차다항식으로 확장하면 더 정확한 위치를 추출할 수 있다. 그러나, 2차 스플라인의 경우 짝수번째 절점에서 버전간의 원하지 않는 굴곡이나 찌그러짐이 발생할 수 있고, 3차 스플라인의 경우 1차 도함수와 2차 도함수가 모두 연속이어야 한다. 이 논문에서는 버전들의 집합이 반드시 고차다항식의 조건을 만족한다고 가정하고 있지 않으므로 비교적 간단한 형태의 1차 스플라인 함수를 사용한다.

3차원+temporal 객체 컴포넌트 구현

1. 컴포넌트 구현 환경

3차원+temporal 객체 모델을 구현하기 위해

이 논문에서 사용한 기법은 OLE/COM이다. OLE/COM은 여러 가지 정보소스에 저장되어 있는 데이터들을 공통한 방법으로 접근할 수 있는 인터페이스를 제공하는 일련의 OLE 인터페이스의 집합이다. 컴포넌트의 내부는 은폐되어 있고, 시스템은 노출되어 있는 인터페이스만을 통해서 컴포넌트에 접근할 수 있다. 따라서 컴포넌트 설계는 곧 인터페이스의 설계이며 지원되는 인터페이스에 대한 보편성을 지원할 수 있도록 하는 것이 중요하다.

컴포넌트를 구현하기 위해 윈도우 2000 프로페셔널 운영체제에서 Visual C++ 6.0에서 ATL(activeX template library)로 컴포넌트를 설계하고, 인터페이스는 IDL(interface definition language)을 사용하여 정의하였다.

2. 컴포넌트 구조 설계

시공간 객체에 대한 인터페이스는 앞서 기술한 3차원+temporal 객체 모델의 클래스 계층도를 기반으로 한다(이현아 등, 2001). ISpatialOperator, ISpatialRelation, ITemporalRelation, ISpatioTemporalRelation, ISpatioTemporalOperator, 그리고 IWks 인터페이스는 추가적인 인터페이스로써 자신의 클래스를 갖지 않고 오직 IDL을 이용한 인터페이스 정의와 메소드 형태만을 정의한다. 이 인터페이스들은 추상 클래스를 통해 인스턴스 객체로 상속되며, 이 때 각 메소드는 각 객체의 정의에 따라 구현되며, 다음과 같은 인터페이스에 의해 제공된다.

- 1) IWks:는 물리적 데이터의 재구성에 필요한 연산 제공
- 2) IGeometry : 공간 객체에 대한 접근 및 분석, 위상 연산 제공
- 3) IGSolid : 3차원 공간 객체에 대한 일반적인 접근 연산 제공
- 4) Validtime : 시간 관계 연산 및 일반 접근 연산 제공
- 5) ITGeometry : 시공간 객체에 대한 분석

및 위상 연산 제공

- 6) ITGSolid : 3차원+temporal 객체에 대한 일반 접근 연산 제공

또한, 인스턴스화 할 수 있는 기하 객체는 DGBox, DGCone, DGCylinder, DGSphere, DGWireFrame, DGLineExtrusion, DGLineFace, DGPolygonExtrusion 등이 있다. 이 기하 객체들에 대한 클래스는 기본적으로 제공되어야 할 필수 인터페이스를 제공해야 한다.

그림 6은 GBox 형의 이력 객체와 이동 객체 컴포넌트를 구성하기 위해 5개의 추상 클래스의 인터페이스로부터 IDGBox 및 IMGBox 인터페이스가 상속되는 모습을 보여주고 있다. 각 추상 클래스의 인터페이스는 연산을 제공하기 위한 연산 관련 인터페이스들을 부가적으로 제공하고 있다.

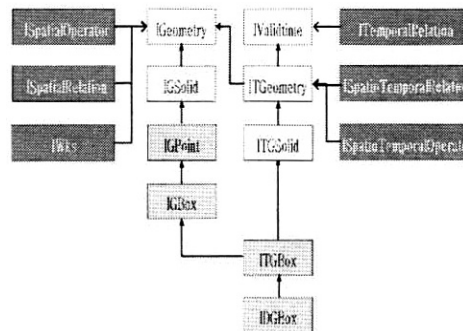


FIGURE 6. Component Structure of spatio-temporal object of GBox type

컴포넌트 인터페이스의 설계는 상속을 이용한다. COM에서 인터페이스 상속은 상위 클래스에서 정의된 모든 메소드를 사용할 수 있다. 각 추상 클래스로부터 제공되는 연산자 인터페이스들은 모두 선택적으로 상속될 수 있으므로 인스턴스 객체 단계에서만 선택적으로 상속이 가능하다.

3. 데이터 저장구조

각 지리 객체들의 저장 형식은 OGC의 OLE/COM을 위한 피쳐 명세서에서 제안한 WKB 구조를 따른다. (OpenGIS, 1999)에서 WKB는 연속적인 바이트의 스트림 구조로 시공간 데이터를 저장하며 이것은 서로 상이한 시스템 간의 물리적인 데이터 이식성을 보장해준다. WKB는 unsigned integer와 double의 두 가지 데이터 타입의 배열로 구성되고, XDR이나 NDR의 두 종류의 타입으로 저장된다.

3차원+temporal 객체 모델의 시공간 데이터를 물리적으로 저장하기 위한 방법으로 (OpenGIS, 1999)의 WKB 구조를 DG 계열로 정의된 인스턴스 객체의 구조로 확장하였다. WKB는 데이터 저장 타입, 객체의 타입, 객체의 전체 유효시간, 버전의 개수, 그리고 각 버전의 저장 타입 및 객체 타입, 유효 시간과 공간 데이터 순서로 하나의 스트림에 저장한다.

표 1은 이러한 WKB 구조를 설명하기 위해 1990년 6월에 신축된 하늘 빌딩에 대한 시공간 데이터의 일부를 보여주고 있다. 하늘 빌딩은 1996년 6월과 2000년 12월의 두 차례에 걸쳐 증축되었으므로 데이터는 세 투플로 구성된다. 여기서 각 투플은 하늘 빌딩이라는 하나의 시공간 객체에 대한 버전이 되고, 이 때 빌딩의 타입을 DGPolygonExtrusion으로 정의한다면 버전의 타입은 TGPolygonExtrusion이 된다.

그림 7은 TABLE 1의 데이터를 WKB 구조의 스트림으로 나열한 하나의 DGPolygonExtrusion 객체이다. 이 스트림을 통해 데이터가 XDR 형식으로 저장되어 있고, 저장된 객체의

타입의 식별자가 317이며, 시공간 객체의 전체 유효시간과 현재 3개의 버전으로 이루어져 있음을 알 수 있다. 각 버전은 309의 객체 타입 식별자를 가지고 있으며, 해당 버전에 대한 유효 시간과 공간 속성으로 이루어져 있다. 각 버전에 포함되어 있는 공간 속성은 객체 타입의 식별자와 PolygonExtrusion을 구성하는 기본 공간 속성의 조합으로 스트림을 구성하고 있다. PolygonExtrusion을 구성하는 기본 공간 속성들은 그림 7에서 확인할 수 있다.

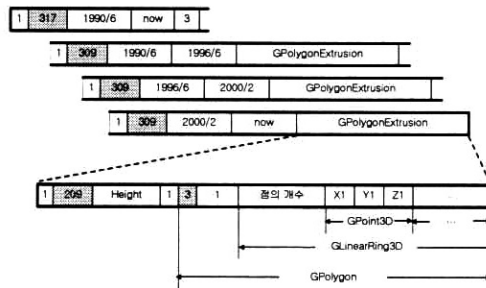


FIGURE 7. WKB structure of DGPolygonExtrusion

WKB의 해석 및 구성은 IWks 인터페이스로부터 상속된 메소드들을 통하여 이루어진다. 즉 사용자가 요청한 시공간 데이터에 대한 관련 WKB는 IWks의 ImportFromWKB() 메소드를 통해 일반적인 데이터 형식으로 해석되어 객체로 구성된 후 연산 및 질의를 통해 사용자에게 제공되고, 새로 입력되는 시공간 데이터는 ExportToWKB() 메소드를 통해 WKB로 구성되어 저장 공간으로 넘겨진다. 따라서 각 인스턴스 컴포넌트의 IWks는 객체의 정의된 형식에

TABLE 1. Static link vs. dynamic link

Start time	End time	Height	Number of Point	X1	Y1	Z1	...
1990/06	1996/06	21m	5	200415.29	451748.67	0	...
1996/06	2000/12	21m	7	200459.38	451767.23	0	...
2000/12	now	21m	7	200459.38	451767.23	0	...

맞춰서 컴포넌트 별로 실행 코드가 구현된다. DGPolygonExtrusion 객체를 해석하기 위한 ImportFromWKB() 메소드는 DGPolygonExtrusion 객체의 타입, 유효시간 그리고 버전의 개수에 대한 데이터를 해석하고, 하위 객체에 대한 WKB 분석은 TGPolygonExtrusion 클래스의 ImportFromWKB() 메소드를 호출하여 수행한다.

적용 및 평가

1. 3차원+temporal 데이터 소비자

WorldCons는 3차원+Temporal 객체 모델을 적용하여 4차원 데이터 제공자에 대한 데이터 소비자이다. WorldCons를 테스트하기 위해 SQL 서버를 사용하여 서울특별시 중구의 실제 건물에 대한 데이터베이스를 구축하였다. 데이터베이스는 commercial, education, emergency, entertain, etc, house, public, innerbuilding, lodge, religion, road, welfare 등 12개의 테이블로 분류된 건물의 데이터와 Info_Table로 구성된다. Info_Table은 각 테이블들의 3차원 지도의 출력을 위해 필요한 기타 공간 정보들, 즉 전체 공간에 대한 좌표, 전체 시간에 대한 범위, 테이블들의 공간 객체 타입 등 시공간 데이터가 저장되어 있는 테이블에 대한 스키마 정보가 저장되어 있다. 이 때 각 건물들의 공간 타입은 DGPolygonExtrusion으로 정의된다.

그림 8은 WorldCons의 시스템 구조도이다. 데이터베이스는 SQL 서버를 사용하고 있으며 WorldCons에서는 질의를 통해 원하는 객체를 호출하여 WKB 분석 및 그래픽으로의 출력이 가능하며, WKB 분석이 정확히 수행되는지 확인하기 위해 리스트 박스를 통해 좌표 값을 출력해 준다.

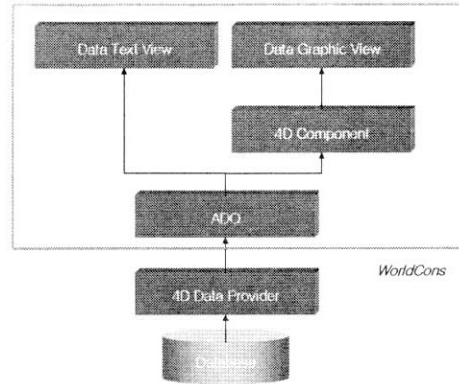


FIGURE 8. Structure of WorldCons

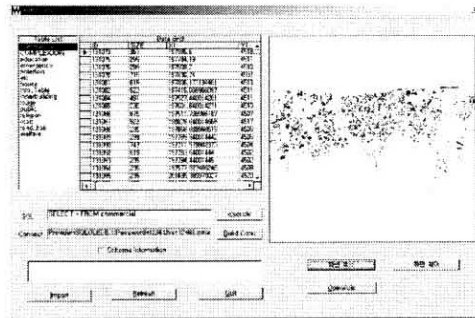


FIGURE 9. Screen shot of WorldCons - commercial table view

그림 10은 예제 질의문에 대한 수행 결과로 선택된 객체의 출력 결과이다.

Example) `SELECT GEOMETRY4D FROM EMERGENCY WHERE ID=262152`

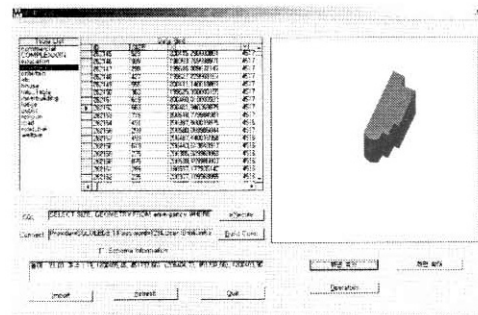


FIGURE 10. Screen shot of search result

2. 4차원 데이터 제공자

(류근호 등, 2001)에서 개발한 4차원 데이터 제공자 컴포넌트는 3차원+temporal 객체 컴포넌트를 이용하여 시스템을 개발한 예이다. 4차원 데이터 제공자에서 사용된 OLEDB는 마이크로소프트사에서 제공하는 인터페이스를 이용하여 데이터를 제공하는 컴포넌트이다. OLEDB의 장점은 데이터베이스의 종류에 관계없이 데이터의 접근을 가능하게 하며 데이터 소스에 대한 투명성을 제공한다는 것이다.

그림 11은 OLEDB를 이용한 4차원 데이터 제공자의 시스템 구조이다. 4차원 데이터 제공자는 효율적인 시공간 연산 수행을 위한 OLEDB 데이터 제공자 부분과 데이터 저장 관리 및 시공간 인덱스를 제공하는 시공간 엔진 부분으로 구성된다. 3차원+temporal 객체 컴포넌트는 OLEDB 데이터 제공자에 3차원+temporal 객체들간의 관계를 식별해주고 객체들간의 시공간 연산 기능을 제공하기 위해 사용되었다. 또한 데이터베이스로부터 획득한 WKB 데이터를 숫자 데이터로 분석하고, 새로운 데이터를 데이터베이스에 저장하기 위해 WKB 데이터로 생성하는 역할을 한다. 따라서 OLEDB 데이터 제공자는 3차원+temporal 객체를 관리하기 위한 별도의 기능을 구축할 필요가 없다.

OLEDB 데이터 제공자는 3차원+temporal 객체 컴포넌트 이외에도 질의 처리기, 피쳐 스키마, 질의 최적화기로 구성된다. 피쳐 스키마는 사용자 질의에서 시공간 질의를 식별한다. 이러한 각 요소들은 모두 컴포넌트 형태로 제공되며, OLEDB 데이터 제공자는 컴포넌트들을 모두 포함하고 있는 하나의 컴포넌트로 구현되었다.

이러한 데이터 제공자에서뿐만 아니라 다양한 GIS 응용 시스템에 3차원+temporal 객체 컴포넌트를 제공함으로써 3차원+temporal 객체에 대한 관리 기능을 쉽게 제공해 줄 수 있다.

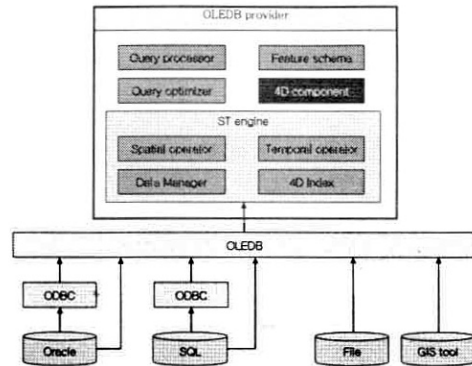


FIGURE 11. Screen shot of search result

3. 기존 연구와의 비교 및 평가

앞 절에서는 3차원+temporal 객체 컴포넌트를 사용하여 각각 데이터 제공자와 소비자를 구축한 예를 보였다. 다음은 컴포넌트에 사용된 3차원+temporal 객체 모델을 기존의 시공간 모델과 비교하기 위해 몇 가지의 기준을 제시하고 있다.

- 기준 1. 2차원 공간 객체를 관리할 수 있는가?
- 기준 2. 3차원 공간 객체를 관리할 수 있는가?
- 기준 3. 유효 시간을 관리할 수 있는가?
- 기준 4. 거래 시간을 관리할 수 있는가?
- 기준 5. 이력 객체를 관리할 수 있는가?
- 기준 6. 위치가 변하는 이동 객체를 관리할 수 있는가?
- 기준 7. 위치와 영역이 변하는 이동 객체를 관리할 수 있는가?
- 기준 8. 이력 객체의 시공간 연산을 지원하 는가?
- 기준 9. 위치가 변하는 이동 객체의 시공간 연산을 지원하는가?
- 기준 10. 위치와 영역이 변하는 이동 객체의 시공간 연산을 지원하는가?
- 기준 11. 개방형 모델과 호환되는가?

기존에 이미 다양한 시공간 모델들이 제안되었으며, 특히 (Worboys, 1994), (Yeh 등, 1994), (Forlizzi 등, 2000), (OpenGIS, 1999)은 이 논문

TABLE 2. Existing Models vs. 3-dimensional+temporal model

기준	(Worboys, 1994)	(Yeh 등, 1994)	(Forlizzi 등, 2000)	3차원+temporal 객체 모델
기준 1	○	○	○	○
기준 2	x	x	x	○
기준 3	○	○	○	○
기준 4	○	x	x	x
기준 5	○	○	△	○
기준 6	x	○	○	○
기준 7	x	○	○	x
기준 8	○	x	x	○
기준 9	x	△	△	△
기준 10	x	△	△	x
기준 11	x	△	△	○

에서 제시하는 시공간 모델을 설계하기 위한 기반 모델이다. TABLE 2는 위와 같은 11가지의 기준에 대해 (Worboys, 1994), (Yeh 등, 1994), (Forlizzi 등, 2000)의 시공간 모델과 3차원+temporal 객체 모델을 비교하였다. (OpenGIS, 1999)은 2차원 공간 모델이므로 비교대상으로 포함시키지 않았다.

여기에서 주어진 기준을 만족할 때 ○, 기준에 대한 내용이 제한적으로 언급되어 있는 경우는 △, 기준을 만족하지 못할 때는 ×로 표시하였다. (Worboys, 1994)와 (Yeh 등, 1994)는 11가지 중 5가지의 기준을 만족하였고, (Forlizzi 등, 2000)은 4가지의 기준을 만족하였다. 따라서 3차원+temporal 객체 모델은 7가지 기준을 만족 시킴으로써 다른 3가지 시공간 모델에 대해 비교적 향상된 표현력을 가지고 있음을 알 수 있다. 또한 이 논문에서 제시한 모델을 기반으로 구현된 컴포넌트는 완전히 독립된 기능을 제공하는 소프트웨어로서 시스템에 탑재함으로써 새로운 GIS 시스템 개발에 대한 노력과 비용을 줄이는 장점을 제공한다. 컴포넌트의 장점은 확장성, 이식성, 재활용성이고, 특히 COM aggregation이나 COM containment 기법을 사용하여 기존의 컴포넌트에서 새로운 기능을 추가할 수 있으며

로 기능 확장이 용이하다.

결론 및 향후 연구

지리 데이터는 시간에 따라 변하는 공간으로 이루어져 있으며 다양한 시스템에 분산되어 있다. 이러한 시공간 데이터를 효율적으로 다루기 위한 연구들은 주로 2차원 공간으로 제한되어 왔으나 최근에는 3차원 공간에 대한 요구가 증가하고 있다. 또한 시간도 하나의 차원으로 고려하여 3차원 이상의 데이터에 대한 관리에 대한 필요성도 증가하고 있다.

이를 위하여 이 논문에서는 3차원 시공간 데이터에 대한 효율적으로 관리할 수 있는 3차원+temporal 객체 모델을 설계하고 컴포넌트로 구현하였다. 3차원+temporal 객체 모델 설계는 개방형 모델과의 호환성, 3차원 공간과 유효시간으로 구성된 3차원 시공간 객체의 관리, 이력 객체와 이동 객체에 대한 효율적인 표현에 초점을 두고 있다. 이를 위하여 개방형 GIS 컨소시엄에서 제안한 2차원 공간 모델을 기반으로 하여 3차원 공간과 유효시간으로 확장하는 방법을 사용하였다. 제안한 모델에서의 인스턴스 객체는 버전들의 집합으로 구성되고 있으므로, 버전

들의 관계를 함수로 정의함으로써 이력 객체와 이동 객체를 동시에 표현하였다.

이 논문에서 제안한 3차원+temporal 객체 모델을 사용함으로써 다른 시스템간의 데이터의 상호 운용성을 보장하고 다양한 시공간 객체에 대한 관리 기능을 제공할 수 있으며, 이러한 모델은 차원적으로 확장된 개방형 모델링의 방향을 제시하였다. 이를 기반으로 구현된 3차원+temporal 객체 컴포넌트는 다른 추가적인 작업 없이 컴포넌트만을 삽입함으로써 3차원+temporal 객체를 관리할 수 있는 새로운 GIS 어플리케이션을 쉽게 구축할 수 있게 해준다.

향후에는 이동 객체의 위치 추출 함수를 고차원 스플라인 함수로 확장함으로써 더 정확한 위치의 추출을 가능하게 할 수 있다. 그리고 위치뿐만 아니라 형태도 시간에 따라 동시에 이동하는 이동 객체에 대해 정의하고 이러한 이동 객체를 위한 관계 분석 연산자의 연구가 필요하다. 이러한 이동 객체에 대한 설계는 3차원+temporal 객체 모델에 대해 증가된 표현력을 제공한다. 아울러 단일 객체만이 아닌 복합 객체를 대상으로 설계를 확장할 수 있다. **KAGIS**

참고문헌

- 류근호 외. 2001. 4D 시공간 데이터 제공자 컴포넌트 개발, 한국전자통신연구원 중간보고서.
- 이현아, 임헌기, 김영일, 남광우, 류근호, 2000. 3D+Temporal 시공간 객체 모델링. 2000 한국정보처리학회 추계학술발표논문집 7(2): 89-92쪽.
- 이현아, 임헌기, 남광우, 류근호, 이종훈. 2001. ATL/COM을 이용한 3D+Temporal 객체 컴포넌트 구현. 2001 한국정보처리학회 춘계학술발표논문집 8(1): 69-72쪽.
- Worboys, M. F., H. M. Hearshshow and D. J. Maguire. 1990. Object-Oriented Modeling for Spatial Database. International Journal of GIS 4(4).
- Worboys M. F., 1994. A Unified Model for Spatial and Temporal Information. The Computer Journal 37(1): 26-34.
- Tryfona N. and C. S. Jensen. 1998. Conceptual data modeling for spatiotemporal applications. Chorochronos TR, CH-98-08.
- Forlizzi L., B. H. Guting, E. Nardelli and M. Schneider. 2000. A Data Model and Data Structures for Moving Object Databases. SIGMOD Conference 2000: 319-330.
- Open GIS Consortium, Inc. 1996. The OpenGIS Guide: Introduction to Interoperable Geoprocessing. OpenGIS TC Document. 96-001.
- Open GIS Consortium, Inc. 1999. Simple Features Specification For OLE/COM Revision 1.1. OpenGIS Project Document. 99-050.
- Open GIS Consortium, Inc. 2001. The OpenGIS Abstract Specification Topic 1: Feature Geometry(ISO 19107 Spatial Schema), version 5. OpenGIS Project Document 01-101.
- Snodgrass, R. I, Ahn. 1985. A taxonomy of time in databases. Proceeding of ACM SIGMOD International Conference on Management of Data pp. 236-246.
- Saltenis, S., C. S. Jensen and S. T. Leutenegger. 2000. Indexing the Positions of Continuously Moving Object. SIGMOD Conference 2000: 331-342.
- Yeh, T-S. and B. de Cambray. 1994. Modeling Highly Variable Spatio-Temporal Data. PRISM TR, 50. **KAGIS**