

4차원 시공간 데이터를 위한 OpenGIS 모델의 확장

김 상 호[†] · 지 정 희^{††} · 류 근 호^{†††}

요 약

실세계의 많은 객체들은 시간적 정보와 공간적 정보를 함께 가지고 있고, 이러한 정보를 다루기 위한 시공간 데이터 모델링의 목적 중에 하나는 시공간 객체를 위한 응용프로그램을 지원할 수 있게 하는 것이다. 객체지향 시공간 모델은 객체단위로 시공간 데이터를 관리 할 수 있기 때문에 효율적으로 시공간 데이터를 관리할 수 있어야 한다. 따라서, 이 논문에서는 시공간 데이터 관리를 객체지향 방식으로 지원할 수 있도록 개방형 지리정보 시스템에서 제안한 2차원 공간모델을 실세계 정보에 맞게 차원적으로 확장하고, 시간 정보를 포함하도록 확장한 모델을 제안한다. 그리고, 제안된 모델의 타당성을 검증하기 위해, 모델을 위한 시공간 데이터 제공자 컴포넌트인 data source 컴포넌트, session 컴포넌트, command 컴포넌트와 rowset 컴포넌트를 구현하고 수행 예를 보인다. 제안된 시공간 데이터 모델과 데이터 제공자는 서로 다르게 저장된 시공간 데이터를 객체지향 모델링을 사용하여 저장, 검색 및 관리할 수 있게 한다.

키워드 : 시공간 데이터 모델, 데이터 제공자, OpenGIS

Extension of OpenGIS Model for Four Dimensional Spatiotemporal Data

Sang Ho Kim[†] · Jeong Hee Chi^{††} · Keun Ho Ryu^{†††}

ABSTRACT

Most of data objects in the real world have spatial and temporal information simultaneously. One of the objectives of spatiotemporal databases is to provide a very expressive framework for spatiotemporal database applications. The object oriented spatiotemporal model can manage spatiotemporal data efficiently because the model is able to store the complete history of any objects. Therefore, in order to handle the spatiotemporal data in object oriented fashion, we proposed a four dimensional spatiotemporal model in the object oriented fashion that extended dimensionally two dimensional spatial model of OpenGIS to fit into the real world information and to include temporal information. In order to proof the proposed model, we design a data provider component which consists of the data source, the session, the command, and the rowset component, and implement it. The proposed four dimensional spatiotemporal data model and provider in the object oriented fashion can be used to store, retrieve and manage the object oriented spatiotemporal data.

Key Words : Spatiotemporal Data Model, Data Provider, OpenGIS

1. 서 론

시공간 데이터 베이스에 관련되어 지난 30년간 시간 데이터베이스 및 공간 데이터베이스 연구가 활발하게 수행되었다. 실세계의 객체들은 실제로 공간적인 정보뿐만 아니라 시간적 정보와 밀접한 관계를 유지한다. 따라서 실세계 객체들은 시간과 공간 속성의 변화를 모두 표현해야 객체를 정확하게 표현할 수 있다. 그러나 과거의 연구는 공간 데이터베이스와 시간 데이터베이스의 두 분야로 서로 독립적으로 연구되어 왔다. 시간 데이터베이스에서는 실세계의 과거

상태 그리고 데이터베이스의 과거 상태라는 두 가지 관점에서 데이터베이스의 정보를 확장하려고 하였고, 공간 데이터베이스에서는 일반적으로 공간 객체를 point, line, region의 추상화를 통한 데이터베이스의 공간 확장을 하였다. 그러나, 최근의 많은 응용분야는 시간에 따라 변하는 공간객체 즉, 시간과 공간을 통합한 시공간 객체를 다룰 수 있는 데이터베이스를 요구하게 되었다. 일반적 의미에서 이러한 시공간 객체를 이동 객체(moving object)라고 하며 시공간 데이터베이스(spatiotemporal database)란 이동 객체를 다루는 데이터베이스라고 한다[1].

시공간 데이터베이스에서 시공간 데이터를 다루기 위한 모델링 기법은 시간분할 스냅 모델[2], 공간-시간 복합 모델[3], 3-영역 데이터모델[4]과 객체지향 모델링기법[5]이 있다. 이러한 기법 중에서 객체지향 모델링 기법은 공간 객체 대한 자체의 구조와 동작을 표현하는 메소드를 가질 수 있고,

※ 이 연구는 유비쿼터스 바이오 정보기술 연구센터의 연구비 지원으로 수행하였음

† 준 회원 : 충북대학교 전기전자컴퓨터공학부 초빙교수

†† 준 회원 : 충북대학교 대학원 전자계산학과

††† 종신회원 : 충북대학교 전기전자컴퓨터공학부 교수

논문접수 : 2003년 5월 6일, 심사완료 : 2005년 3월 23일

객체의 상태에서 특정 시간 속성값으로 정의할 수 있어서 시공간 객체를 모델링 하는데 이용된다. 객체 지향 시공간 모델은 유효 시간과 거래 시간을 지원 함으로 완전한 이력을 제공해 줄 수 있고, 속성, 지리와 시간 데이터 요소를 일반적인 데이터베이스를 다루듯이 사용할 수 있다[6]. 그러나 모델링 단계를 수행이 되었으나, 현재의 지리정보 시스템 도구들은 이 모델을 지원하지 못했다. 지리 정보 시스템이 시공간 모델을 지원하지 못했던 가장 큰 이유는 공간과 일반 속성, 또는 시간과 일반속성만으로 구성되어 있는 데이터베이스를 하나로 통합하는데 데이터의 이질성과 고유의 특성에 맞는 구조 때문에 많은 어려움이 있었다.

따라서, 이 논문에서는 시공간 객체의 완전한 이력 제공을 가능하게 하고, 속성, 지리와 시간 데이터 요소를 일반 데이터베이스 요소처럼 사용할 수 있도록 하는 객체 지향 시공간 모델을 제안한다. 제안되는 시공간 모델은 개방형 지리정보시스템에서 제안된 2차원 공간객체를 공간 차원적으로 확장하여 3차원 정보를 유지하게 하고, 유효시간을 지원하는 클래스를 제안하여 이 클래스로부터 상속을 받아 시간 정보를 유지하는 실세계 시공간 객체를 표현하도록 한다. 이렇게 제안된 객체지향 시공간 모델을 검증하기 위해 시공간 데이터제공자 컴포넌트를 설계, 구현한다. 구현된 데이터 제공자는 데이터 통합작업 없이 이질적인 데이터들을 사용할 수 있게 한다. 이때, 검증에 사용된 시공간 데이터는 서울시 중구 지역의 가스 이력 정보이며, 시공간 질의를 수행하여 타당성을 보였다.

논문의 효율적인 구성을 위해서 2장에서는 관련 연구로 기존의 시공간 데이터 모델과 시공간 데이터를 관리 하는 기법에 대해 살펴본다. 3장에서는 실세계 삼차원 공간 객체 정보와 유효시간을 지원하는 객체지향 4차원 시공간 모델을 제안한다. 3장에서 제안된 모델을 검증하기 위해, 4장에서는 객체 지향 데이터 제공자 컴포넌트를 설계하고, 5장에서 이 설계를 기반으로 시공간 데이터 제공자를 구현하고 6장에서 기존의 시공간 모델과 비교한다. 마지막으로 7장에서 결론 및 향후 연구방향에 대해 기술한다.

2. 관련 연구

이 장에서는 기존 시공간 데이터 모델과 각 모델의 문제점, 시공간 데이터를 효율적으로 관리하는 기법에 대해 기술한다.

2.1 기존 시공간 데이터베이스 모델

시공간 정보를 저장하고 다루는 방법에는 시간분할 스냅 샷 모델[2], 시공간 복합 모델[3], 3-영역 데이터모델[4]과 객체지향 모델링기법[5]등이 있다. 시간분할 스냅 샷 모델은 시간의 흐름 상에서 공간 객체들에 발생한 변화를 스냅 단위로 관리한다. 하지만 개별적인 스냅 샷은 해당 시점의 정보만을 가지며 이전 스냅과의 변경은 직접적으로 표현하지

못하기 때문에 스냅간의 변화는 두 스냅을 모두 비교하여야 하는 것과 사건이 발생할 때마다 스냅 전체를 저장하기 때문에 중복저장의 문제점을 가지고 있다. 시공간 복합 모델은 지리데이터와 시간 데이터를 구분하여 복잡성을 감소시켜서 표현하는 장점이 있지만, 시간상의 두 점 사이에서 일어나는 변화를 알아내기 힘들다. 3-영역 데이터 모델은 비공간 속성, 공간 속성, 시간 속성을 서로 분리하여 표현하고, 지리 현상을 표현하기 위해 이들 사이에 연결을 제공한다. 그러나 이 기법은 위상 관계를 저장하지 못하고, 데이터 중복이 많다는 단점을 가지고 있다.

시간 단위로 객체의 이력을 관리하기에 효과적인 방법은 객체지향 모델링 기법이다[7]. 객체 지향 모델의 핵심 개념은 객체이다. 이때 객체는 실세계 개체 또는 피처의 추상적 표현이며, 각 객체는 자체의 구조와 그들의 동작을 표현하는 메소드들을 표현함으로 속성을 가진다. 객체의 상태는 특정 시간에 속성값으로 정의되고, 객체의 구조와 메소드의 캡슐화로 다른 객체에 의한 내부 간섭을 피하게 된다. 객체에 접근할 수 있는 방법은 메시지를 통해서이며, 동일한 구조를 가진 객체는 class로 그룹화된다.

Worboy[8]에 따르면 객체지향 연구는 관계형 모델이 효과적으로 해결할 수 없는 문제에 사용되어왔다. 그러나, 객체지향 접근을 수행하는 많은 연구는 수행되지 않았다. Snodgrass[5]의 객체 지향 모델은 객체 단위로 시간정보를 저장할 수 있도록 설계 되었으나, 실질적으로 이 모델을 실제로 구축한 지리정보 시스템은 아직 없으며, 실제로 시공간 데이터 베이스를 적용하는 방법에 대한 고려를 수행하지 않아 실제로 분산된 여러 시공간 데이터를 단일 시스템으로 통합하여 다루는 조건을 전제로 하고 있다. 따라서, 이러한 모델은 실제로 구현되기 어렵다. 이러한 이유로 객체 지향 시공간 모델을 구현할 때, 방대한 시공간 데이터를 통합 작업 없이, 저장된 형태로의 관리를 가능하게 하는 방법을 선택해야 시공간 데이터를 효율적으로 관리할 수 있다.

2.2 시공간 데이터의 관리

현실의 데이터 중심적인 응용 프로그램들은 데이터 관리에 다음과 같은 요구사항을 가진다. 먼저, 데이터 스스로의 접근을 가능하도록 하여야 한다. 시공간 데이터는 서로 다르게 구축되어 있는 정보에 따라, 통합 없이 공간 데이터와 시간 데이터, 시공간 데이터에 접근 할 수 있도록 지원하여야 효율적인 시공간 데이터베이스 시스템을 만들 수 있다. 둘째, 특정 데이터를 관리할 수 있어야 한다. 이질적 데이터 소스를 SDTS[11, 12]처럼 하나의 표준 형태로 정의한다면, 이러한 표준 데이터 소스는 데이터 모델의 질의언어, 트랜잭션 그리고 인덱싱 영역에서 장점을 가질 수 있지만, 그러한 변환 작업으로 인해 많은 비용소모와 데이터 손실을 가져올 수 있다. 현재의 응용프로그램은 원본 형태의 데이터 소스를 직접 접근하길 원하고, 그들의 저장된 원본을 활용하기를 원한다. 마지막으로 시공간 데이터베이스의 다양한

응용 분야에 따른 상이한 데이터 접근 모델을 허용하여야 한다. 응용 프로그램이 운영되는 환경에 따라 서로 다른 요구사항을 필요로 한다. 이러한 요구사항들은 하나의 데이터베이스 관리 시스템을 사용하여 해결할 수도 있다. 그러나 하나의 데이터베이스 관리 시스템으로 사용하는 방법의 장점은 질의와 트랜잭션 유틸리티 같은 데이터베이스 시스템 서비스가 일정한 방법으로 모든 이질적 데이터를 관리하는 게 가능하다는 것이다. 단점으로는 원본 소스로부터 데이터를 옮기는 작업이 매우 번거롭고 많은 비용도 소요된다. 그리고 더 큰 문제는 그들이 서로 다른 틀일 때 일어난다. 서로 다른 틀들은 데이터베이스 형태가 아닌 그들 자체의 형태를 가질 수 있기 때문이다.

따라서, 시공간 데이터베이스에서 지원하는 새로운 데이터 타입과 방대한 양의 자료를 모두 처리하기 위한 방법으로 개방형 구조를 가진 데이터 제공자를 이용하여 기존의 데이터 베이스에서 데이터를 접근할 수 있고, 변형되지 않는 원본 자체를 직접 활용하며, 응용 프로그램에 영향 받지 않는 데이터베이스를 구현할 수 있다.

3장에서는 이러한 데이터 제공자에 사용될 객체지향 사차원 시공간 모델을 제안한다.

3. 객체 지향 시공간 데이터 모델

이 장에서는 객체 단위로 시공간 데이터 모델을 정의하기 위해 시간단위로 객체를 관리할 수 있는 ValidTime 클래스와 공간 정보를 나타내는 Geometry 클래스로부터 상속받는 이력 객체 모델 TGeometry 클래스와 클래스 메소드를 설계하고, 이러한 설계를 바탕으로 3차원 시공간 데이터 모델을 설계한다. 기존의 개방형 지리정보시스템에서 제안되는 모델[13]은 2차원으로 한정되어 있기 때문에, 이 논문에서는 삼차원 실 세계 정보의 왜곡 없는 저장과 관리를 위해 공간 차원으로 확장하고, 시간차원을 관리할 수 있도록 클래스를 제시한다.

3.1 시간 모델

시간 데이터베이스에서는 유효시간(valid time), 거래시간(transaction time), 그리고 사용자 정의 시간(user defined time) 등을 사용할 수 있다. 이러한 시간들은 각각의 고유한 기능을 갖는다. 유효시간(valid time)은 현실세계에 존재하는 객체에 대한 논리적인 시간을 의미하며 객체가 발생하였거나 객체가 소멸된 시간을 가리킨다. 유효시간은 일반적으로 사용자에게 의해서 결정되는데 사용자가 위치한 시간시스템에 의해서 상대적인 값으로 표현된다. 이 논문에서는 실세계 객체의 변화를 표현할 수 있는 유효시간만을 지원하는 것으로 한정한다. 따라서 이 절에서는 유효시간 지원을 위한 클래스 ValidTime를 제안한다. (그림 1)는 시간을 지원하기 위한 클래스이다. ValidTime 클래스는 실세계에서 존재하는 객체의 논리적인 시간을 나타낸다.

ValidTime	
from : time	
to : time	
+isPeriod() : boolean	
+getInterval() : Time	
+before(t:ValidTime) : boolean	
+equals(t:ValidTime) : boolean	
+meets(t:ValidTime) : boolean	
+overlaps(t:ValidTime) : boolean	
+during(t:ValidTime) : boolean	
+starts(t:ValidTime) : boolean	
+finishes(t:ValidTime) : boolean	
+intervalToMin(t:Time) : double	
+intervalToSec(t:Time) : double	
+updateFrom(newtime : Time) : boolean	
+updateTo(newtime : Time) : boolean	
+intersectionZone(t:ValidTime) : ValidTime	

(그림 1) ValidTime 클래스

<표 1> ValidTime 클래스의 메소드들

메소드 이름	설 명
IsPeriod	ValidTime이 시간 구간을 나타낼경우 Period가 되고 특정 시간을 나타낼경우 시간 instance가 된다.
getInterval	Period의 시간 구간을 값을 반환한다.
IntervalToMin	시간 구간 값을 분으로 환산하여 반환한다.
IntervalToSec	시간 구간 값을 초로 환산하여 반환한다.
before	A before B: A가 B보다 시간적으로 앞서 있을 경우 true반환
equals	A equals B: A와 B의 시간 또는 시간 구간이 같을 경우 true반환
meets	A meets B: A의 To값과 B의 From값이 같을 경우 true반환
overlaps	A overlaps B: A의 구간과 B의 구간이 겹칠 경우 true 반환
during	A during B: A가 B에 포함될 경우 true반환
starts	A starts B: A.From = B.From 그리고 A.To < B.To 일 경우 true반환
finishes	A finishes B: A.From > B.From 그리고 A.To = B.To 일 경우 true 반환

유효시간에 대한 모델링은 Time 타입의 from과 to의 두 가지 속성과 4가지 기본 메소드인 IsPeriod(), getInterval(), IntervalToMin(), IntervalToSec()과 객체간의 시간 관계를 위한 6가지 메소드, before(), equals(), meets(), overlaps(), during(), starts(), finishes()를 가지고 있는 두 클래스를 정의한다. 다음 <표 1>은 각각의 메소드에 대한 정의를 보여 준다. 여기서 A, B는 각각 ValidTime 클래스의 객체를 나타낸다.

3.2 이력 객체 모델

공간 객체 클래스인 Point, LineString, Polygon에 시간 차원을 확장하여 공간 객체에 대한 이력을 표현하기 위한 2차원 시공간 객체 추상 클래스인 TGeometry를 모델링 한다. TGeometry는 Geometry클래스, ValidTime 클래스로부터 상속 받은 시공간 객체에 대한 추상클래스이다.

TGeometry클래스의 객체는 시간 객체이며 공간 객체이

TGeometry	
tDimension	: long
tIsEmpty	: boolean
tIsSimple	: boolean
+tBoundary()	: TGeometry
+tEqual(tG: TGeometry)	: boolean
+tDisjoint(tG: TGeometry)	: boolean
+tIntersect(tG: TGeometry)	: boolean
+tTouch(tG: TGeometry)	: boolean
+tCross(tG: TGeometry)	: boolean
+tWithin(tG: TGeometry)	: boolean
+tContain(tG: TGeometry)	: boolean
+tOverlap(tG: TGeometry)	: boolean
+tDistance(tG: TGeometry, vt: ValidTime)	: double
+tBuffer(distance: double, vt: ValidTime)	: Geometry[]
+tIntersection(tG: TGeometry, vtimepoint: ValidTime)	: Geometry
+instanceAt(vt: ValidTime)	: TGeometry
+lifetime()	: ValidTime
+instancePeriod(vt: ValidTime)	: TGeometry
+location()	: Geometry[]

(그림 2) TGeometry 클래스

다. 그리고 하위 클래스로써 TPoint, TLine, DPoint, DLine String, DPolygon를 가지게 된다. 기본적인 속성으로써 tDimension, tIsEmpty, tIsSimple 이 있고 TGeometry사이의 시공간 위상관계를 위한 tContain(), tCross(), tDisjoint(), tEqual(), tIntersect(), tOverlap(), tTouch(), tWithin() 메소드들이 있으며 시공간 분석을 위한 메소드으로써 tBoundary(), tBuffer(), tDistance(), tIntersection(), lifetime(), InstanceAt(), InstancePeriod() 그리고 location()이 있다. (그림 2)는 TGeometry 클래스를 보여준다.

다음 <표 2>는 각 메소드에 대한 설명을 보여준다.

<표 2> TGeometry 클래스의 메소드들

메소드 이름	설 명
tBoundary	TGeometry형태로 시공간 최소 경계를 반환
tEqual	다른 TGeometry와 시간적으로 동시에 공간적으로 동등한 경우 TRUE
tDisjoint	다른 TGeometry와 시간적 그리고 공간적으로 서로 관련이 없을 경우 TRUE
tIntersect	다른 TGeometry와 시간적 그리고 공간적으로 교차되어 있을 경우 TRUE
tTouch	다른 TGeometry와 시간적 그리고 공간적으로 접촉(touch)되어 있을 경우 TRUE
tCross	다른 TGeometry를 시간적으로 교차하고 공간적으로 가로지를 경우 TRUE
tWithin	다른 TGeometry에 시간적으로 공간적으로 포함 될 경우 TRUE
tContain	다른 TGeometry를 시간적으로 공간적으로 포함 할 경우 TRUE
tOverlap	다른 TGeometry를 시간적으로 공간적으로 중첩 할 경우 TRUE
tDistance	다른 TGeometry의 한 Point와 시간적으로 동등(equals)할때 공간적으로 최소 거리를 반환.
tBuffer	주어진 시간에서 주어진 거리에 속하는 Geometry객체들을 반환
tIntersection	주어진 시간에서 다른 TGeometry객체와 겹친부분을 Geometry형식으로 반환
instanceAt	주어진 시간점(time point)에서의 Geometry반환
Lifetime	TGeometry의 ValidTime반환
instancePeriod	주어진 시간구간(time period)에서의 TGeometry반환
Location	시간에 따른 Geometry의 변화를 집합 형식으로 반환

설계된 TGeometry 클래스는 다음 예와 같이 사용된다.

```

...
TGeometry A,B
/* TGeometry로 A,B를 선언, 이때 A는 [2004.3-2004. 12],
B는 [2003.12- 2004.5]의 유효시간을 가졌다고 가정*/

A.tWithin(B)
/* TGeometry A,B가 시간적으로 동시에 존재하며(여기서는
[2004.3-2004.5]), 공간적으로 Within한 관계가 있으면 이를
선택*/
A.tTouch(B)
/* TGeometry A,B가 시간적으로 동시에 존재하며,
공간적으로 Touch한 관계가 있으면 이를 선택*/
(중략)
    
```

위와 같은 예는 다음 절에서 제시될 시공간 데이터 모델 클래스에서도 동일한 방식으로 적용된다.

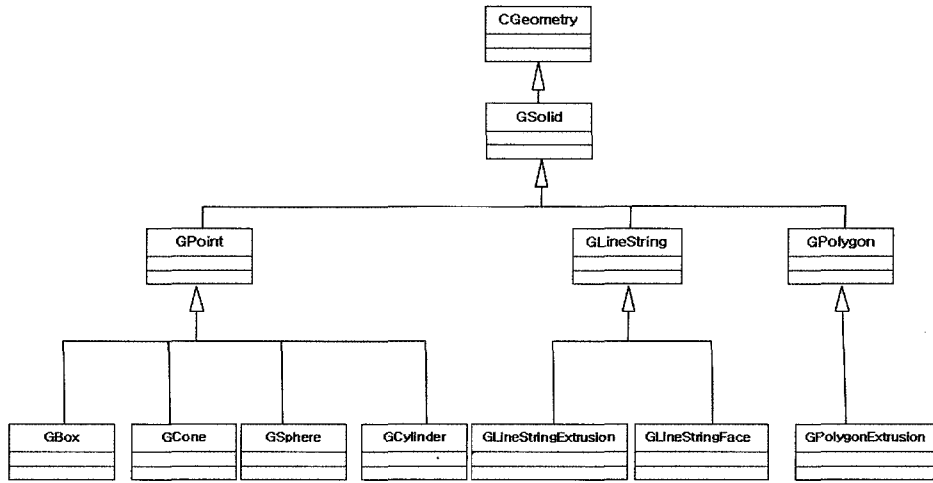
3.3 시공간 데이터 모델

지금까지 정의한 TGeometry 클래스와 ValidTime클래스를 이용하고, 2차원 공간 객체의 차원적 확장을 수행하여 4차원 시공간 데이터 모델을 제시한다.

(그림 3)은 국제 표준으로 인정되고 있는 개방형 지리정보시스템의 2차원 공간모델을 3차원 객체들로 확장한 3차원 공간 모델을 보여준다.

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.

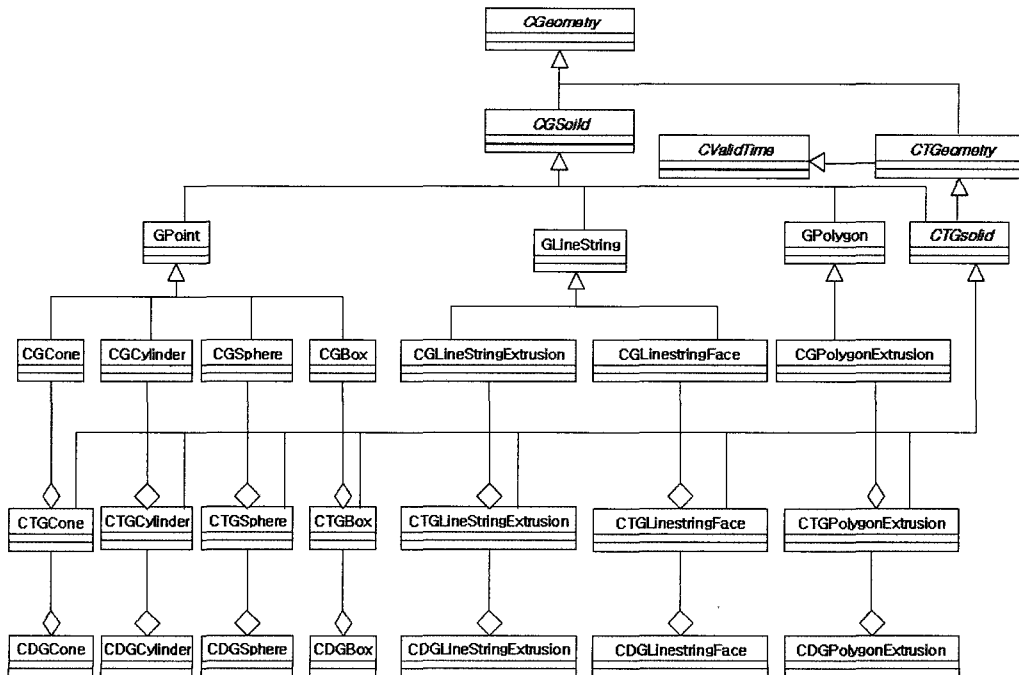
GPoint의 하위 객체들은 하나의 GPoint를 갖고, 그 점을 기점으로 하는 공간 요소로 객체를 구성하게 된다. GBox 형태는 직육면체이고, 중심점과 함께 저장되는 공간 정보는 중심점을 기점으로 하는 가로, 세로, 높이이다. GCone은 원뿔형이며, 중심점과 반지름, 높이를 갖는다. GSphere는 구의 형태로 중심점과 반지름을, GCylinder는 중심점과 반지름, 높이를 갖는 원통형 객체이다. GLineString의 하위 객체들은 중심선이 되는 GLineString과, 선을 기점으로 하는 공간 요소로 객체를 구성한다. GLineStringExtrusion은 중심선과 중심선을 구성하는 점들이 중심이 되는 원의 반지름 값으로 그려지는 관의 형태라고 정의한다. GLineStringFace는 중심선을 구성하는 점들로부터 일정한 넓이의 폭을 갖는 형태이다. GPolygon의 하위 객체는 GPolygonExtrusion 하나뿐이다. GPolygonExtrusion의 형태는 밀면으로 하나의 GPolygon이고, 밀면을 구성하는 각 점들로부터 일정한 높이 값을 갖는 객체이다. 이 때 밀면인 GPolygon은 구멍이 없다고 가정한다. 3차원 객체에 대한 분류는 <표 3>에서 정리하였다.



(그림 3) 3차원 공간 모델

<표 3> 3차원 공간객체의 분류

객체 명	설명
GSolid	3차원 지리 객체 추상 클래스
GPoint	x, y, z 좌표 값을 갖고 있는 3차원 Point
GBox	한 Point을 기준으로 가로, 세로, 높이 값을 갖는 직육면체
GCone	한 Point를 기준으로 반지름, 높이 값을 갖는 원뿔
GSphere	한 Point를 기준으로 반지름 값을 갖는 구
GCylinder	한 Point를 기준으로 반지름, 높이 값을 갖는 원통
GLineStringExtrusion	두 점을 갖는 GLineString과 GLineString 상의 점을 중심으로 하는 원의 반지름 값을 갖는 GLineExtrusion의 집합
GLineStringFace	두 점을 갖는 GLineString과 GLineString 상을 중심으로 하는 폭의 값을 갖는 GLineFace의 집합
GPolygonExtrusion	GPolygon과 GPolygon의 경계에 위치하는 점들로부터 일정한 높이 값을 갖는 객체



(그림 4) 시공간 데이터모델

<표 4> 시공간 객체

시공간 객체	설 명
DGBox	TGBox 객체의 순열로 표현되는 객체. 단 TGBox은 유효 시간 객체를 상속 받은 GBox 객체
DGCone	TGCone 객체의 순열로 표현되는 객체. 단 TGCone은 유효 시간 객체를 상속 받은 GCone 객체
DGSphere	TGSphere 객체의 순열로 표현되는 객체. 단 TGSphere은 유효 시간 객체를 상속 받은 GSphere 객체
DGCylinder	TGCylinder 객체의 순열로 표현되는 객체. 단 TGCylinder은 유효 시간 객체를 상속 받은 GCylinder 객체
DGPolygonExtrusion	TGPolygonExtrusion 객체의 순열로 표현되는 객체. 단 TGPolygonExtrusion은 유효 시간 객체를 상속 받은 GPolygon 객체
DGLineStringFace	TGLineStringFace 객체의 순열로 표현되는 객체. 단 TGLineStringFace은 유효 시간 객체를 상속 받은 GLineStringFace 객체
DGLineStringExtrusion	TGLineStringExtrusion 객체의 순열로 표현되는 객체. 단 TGLineStringExtrusion은 유효 시간 객체를 상속 받은 GLineStringExtrusion 객체

(그림 4)은 3차원 공간 모델과 1차원 시간 모델을 적용한 4차원 시공간 데이터모델을 보여준다. 시간 클래스인 CValidTime, CGeometry클래스, CValidTime클래스로부터 상속을 받은 시공간 클래스인 CTGeometry를 이용하여 시공간 데이터 모델을 구성한다.

CTGSolid는 TGeometry클래스와 CGSolid클래스로부터 상속을 받는 시공간 추상 클래스이고, CGPolygonExtrusion을 내부의 한 속성으로 가지고 있으며 시간 클래스의 시간 속성값을 상속 받는다. 즉, CTGPolygonExtrusion은 CGPolygonExtrusion과 이 공간속성의시간을 가지고 있는 시공간 클래스이다. 그러므로 실제 시공간 객체에 대하여 독립적인 시공간 속성으로 사용될 수 없다. 즉 한 객체의 한 버전정보로 CDGPolygonExtrusion을 이루는 한 요소로써 사용된다. CDGPolygonExtrusion의 배열을 시공간 속성으로 가지고 있으며, 시간 클래스로부터 상속 받은 시간 속성은 모든 버전 즉, CTGPolygonExtrusion들의 전체 유효시간을 나타낸다.

이력정보를 관리할 수 있는 시공간 객체를 <표 4>에서 정리하였다. 제안된 시공간 모델의 검증을 위해, 4장에서는 시공간 데이터 제공자를 설계하고 5장에서는 설계된 시공간 데이터 제공자를 구현한다.

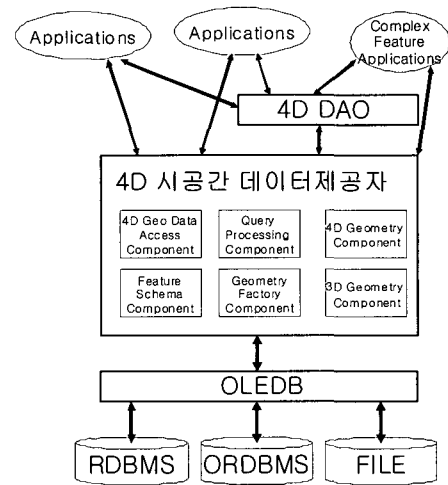
4. 객체 지향 시공간 데이터 제공자

4.1 데이터 제공자 컴포넌트의 설계

이 절에서는 시공간 데이터 제공자의 구성요소를 설계한다. (그림 5)는 시공간 데이터 제공자의 위치와 각각의 데이터 저장장치에 따른 OLE/DB 컴포넌트의 관계를 설명한다. 각각의 저장장치에 따른 OLE/DB 제공자를 이용하여 엔진 레벨에서 제공자를 이용하여 데이터에 접근하고, 접근된 데이터를 제공자 레벨에서 응용 프로그램에 제공한다.

이때 제공되는 데이터의 형태는 ORACLE, SQL Server 파일이나 기타 Tool에서 제공되는 데이터에 상관없이 이용 가능하다.

시공간 데이터 제공자는 기본적으로 Data Source, Session, Command, Rowset등 네개의 계층적 컴포넌트로 이루어진다. 각각의 컴포넌트를 구현하기 위해 각 컴포넌트의 공통적 프로세스를 담당하는 IUnknown 인터페이스 클래스 부분과 인터페이스를 구현하는 클래스들로 구성된다.



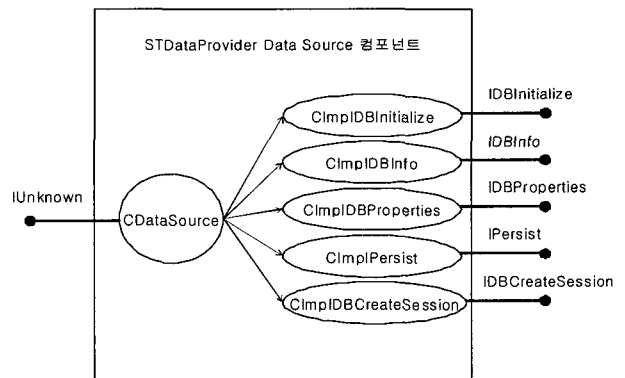
(그림 5) 시공간 데이터 제공자 구성도

4.2 데이터 제공자 컴포넌트 설계

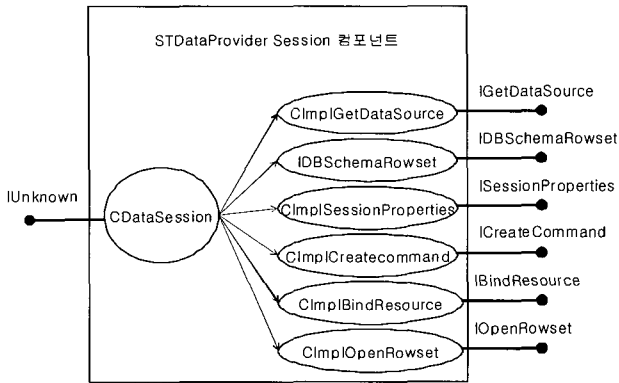
4.2.1 DataSource 컴포넌트

DataSource 컴포넌트는 데이터 제공자의 유형, 데이터 소스, 사용자 아이디, 패스워드 정보를 가지고 데이터 제공자를 초기화 하고, Session 객체를 생성하는 인터페이스를 제공한다. 또한 DataSource에 대한 세부적인 속성을 지정할 수 있다

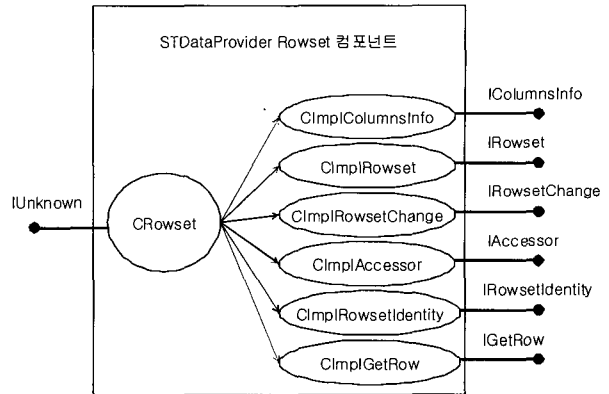
(그림 6)은 Data Source컴포넌트를 구성하고 있는 클래스와 인터페이스를 나타낸다.



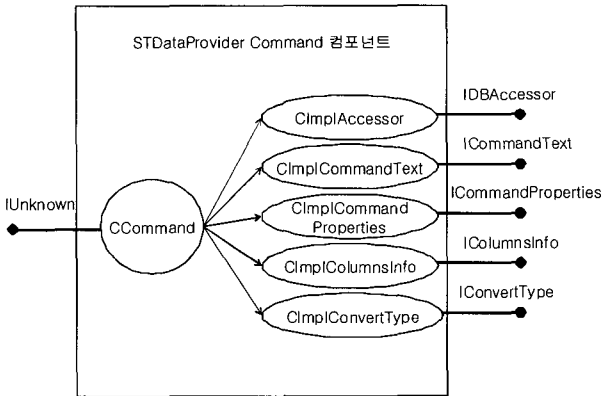
(그림 6) Data Source 컴포넌트



(그림 7) Session 컴포넌트



(그림 9) Rowset 컴포넌트



(그림 8) Command 컴포넌트

4.2.2 Session 컴포넌트

Session 컴포넌트는 실제 데이터 소스에 대한 연결을 담당하게 된다. 데이터 제공자가 지원하는 스키마 정보들과 데이터 소스에 대한 정보, 그리고 Session에 대한 정보들을 제공하며, 데이터 베이스의 테이블 이름을 이용하여 Rowset 결과를 얻을 수 있다.

Data Source 컴포넌트와 비슷하게 CDBSession 클래스는 전체 Session 컴포넌트의 공통적 처리를 담당한다. (그림 7)는 Session 컴포넌트의 기본구조를 보이고 있다.

4.2.3 Command 컴포넌트

Command 객체는 사용자의 질의를 이용하여 Rowset 객체를 받는 역할을 한다. 그리고, 사용자의 질의와 관련된 부분으로서, 특정한 정보를 가지고 질의를 호출할 수 있게 하는 역할도 한다.

(그림 8)은 Command 컴포넌트의 기본 구현 클래스 모델을 보이고 있다.

4.2.4 Rowset 컴포넌트

Command 객체의 결과나 Session 객체의 결과로서 반환되는 Rowset 객체이다. 실제 데이터 베이스 테이블에 대한 접근을 제공하며, 데이터 읽기 뿐만 아니라 수정, 삭제 또한

제공한다. IColumnInfo 인터페이스는 특별히 데이터 베이스에 대한 필드 정보를 가지고 있다.

(그림 9)는 이러한 Rowset 컴포넌트의 구조를 보이고 있다.

5. 시공간 데이터 제공자 컴포넌트의 구현

이 논문에서 제안한 4차원 객체 지향 시공간 모델을 검증하기 위한 시공간 데이터 제공자의 구현 환경은 마이크로소프트 윈도우 2000 서버와 데이터베이스 관리시스템은 마이크로소프트 SQL 서버 2000를 사용하였고, 펜티엄3 800 Mhz 와 RAM 320MB 머신에서 구현하였다. 사용한 프로그래밍 툴은 마이크로 소프트 Visual C++ 6.0과 언어로는 C++과 ATL COM을 사용하였다.

서로 다른 시공간 데이터를 관리하기 위해, 우선 해당 데이터베이스나 파일에서 구축된 테이블 정보를 필요로 한다. (그림 10)은 테이블의 DBSCHEMA_TABLES 정보를 Rowset 형태로 반환된 결과이다.

(그림 11)은 실제 시공간 데이터를 읽어온 모습이다. WKB 필드가 시공간 데이터를 가지고 있고, 그림에서와 같이 이진 데이터의 형태로 보인다.

Row Handle	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE
0x00000012	4D GIS	dbo	sysusers	SYSTEM TABLE
0x00000013	4D GIS	dbo	dtproperties	TABLE
0x00000014	4D GIS	dbo	FTest	TABLE
0x00000015	4D GIS	dbo	jobs	TABLE
0x00000016	4D GIS	dbo	sales	TABLE
0x00000017	4D GIS	dbo	stores	TABLE
0x00000018	4D GIS	dbo	Table1	TABLE
0x00000019	4D GIS	dbo	titleauthor	TABLE
0x0000001A	4D GIS	dbo	titles	TABLE
0x0000001B	4D GIS	kang	ApIF	TABLE
0x0000001C	4D GIS	kang	BankF	TABLE
0x0000001D	4D GIS	kang	BuildingF	TABLE
0x0000001E	4D GIS	kang	FTest2	TABLE
0x0000001F	4D GIS	kang	HospitalF	TABLE
0x00000020	4D GIS	kang	KN13	TABLE
0x00000021	4D GIS	kang	PublicOfficeF	TABLE
0x00000022	4D GIS	kang	RestaurantF	TABLE
0x00000023	4D GIS	dbo	sysalternates	VIEW
0x00000024	4D GIS	dbo	sysconstraints	VIEW

(그림 10) 테이블 스키마 정보

Row Handle	ID	SIZE	WKB
0x00000001	2	1000	...
0x00000002	3	1000	...
0x00000003	4	1000	...
0x00000004	5	1000	...
0x00000005	6	1000	...
0x00000006	7	1000	...
0x00000007	8	1000	...
0x00000008	9	1000	...
0x00000009	1	1000	...
0x0000000A	1	1000	...
0x0000000B	1	1000	...
0x0000000C	1	1000	...
0x0000000D	1	1000	...
0x0000000E	1	1000	...

(그림 11) WKB 데이터

이때 구성된 WKB 필드는 Unsigned Integer, Double 등의 연속 기술된 숫자타입으로써 기하 인스턴스를 연속적으로 기술한다. 또한, WKB 표현은 빅엔디안(Big Endian)과 리틀엔디안(Little Endian) 두개의 표현 방식중에 하나를 숫자타입의 연속 기술로 얻어진다.

위와 같이 구성된 데이터에서 제공되는 정보를 이용, 시간, 공간, 시공간 질의를 지원할 수 있는 시스템을 구현하였다. 이때 사용된 공간 데이터 셋은 서울 특별시 중구 지역의 건물로 commercial, education, emergency, entertain, house, lodge, public, religion, welfare, etc 10개의 테이블로 분류된 건물의 데이터와 각 건물 테이블 전체공간에 대한 좌표, 스키마 정보를 가진 info_table 테이블로 구성된다. 이 건물 Geometry의 레코드 수는 총 10484개이다. 각 테이블의 구조 모두 같다. 건물에 대한 Geometry는 OGC에서 제시된 WKB 포맷을 확장하여 이용한다. 그리고, 이력 데이터 셋은 시나리오에 의해 각 건물 데이터에 대한 가스 경보 이력을 관리하며, 그 이력 데이터는 'Gas_Alarms' 테이블에 저장되며, 각 건물의 OID와 과거 가스 경보 이력에 대한 Valid Time이 저장된다.

데이터에 대한 시간질의 “2002년 6월15일에 가스 경보가 울린 건물들을 검색하라.”의 결과 화면이 다음 (그림 12)이다. “지정된 영역과 교차하는 2002년 1월 1일부터 2002년 5월 31일까지 가스 경보가 울린 건물들을 검색하십시오.”와 같은 공간과 시간 인터벌에 대한 질의 결과화면은 (그림 13)와 같다. 이때의 공간 질의로 사용되는 지정된 영역은 사용자가 그린 윈도우 영역이 된다.

6. 평 가

Narciso[7]는 그의 논문에서 기존의 시공간 모델간의 문제점들을 비교 하였고, 이 논문에서는 그 평가된 논문들과 제안된 모델간의 문제점을 <표 5>과 같이 구분한다.

기존의 시공간 모델은 스냅샷 모델[2], 시공간 복합모델[3], 3영역 데이터 모델[4], 객체지형 모델[5], 시공간 큐브모델[14], 3D/4D 시공간 모델[15]과 관계형 모델[16]로 구분 가능하다. 각각의 장단점을 비교해 보면 스냅 샷모델은 다른 모델보다 많은 데이터를 가져야 표현 가능하고, 위상 관계를 구할 때 복잡한 알고리즘이 필요하며, 시간 질의를 할 경우 복잡하다는 단점과, 데이터 중복, 데이터 불일치와 변화를 구별하기 힘들다는 단점을 가지고 있다. 시공간 복합 모델은 많은 양의 데이터, 데이터 중복, 변화를 구별하기 힘들다는 점과 복잡한 모델이라는 단점을 가지고 있다. 시공간 모델에서 주로 이용되고 있는 관계형 모델은 데이터 중복과 지리데이터를 표현하는데 난이한 문제를 가지고 있다. 이러한 기존의 문제점들을 해결하기 위한 방법으로는 객체 단위로 시공간 객체를 표현하는 방법이 있으나, 실제로 구현되지 못했다는 문제를 가지고 있었다. 이 논문에서는 구현될 수 있는 객체지향 시공간 모델을 개방형 시스템에서 제시한 모델을 3차원 실세계 정보로 확장하고, 유효시간 클래스를 상속받도록 하여 객체 단위의 시공간 데이터 관리를 가능하게 하고, 이를 구현하여 기존 객체지향 모델의 문제점을 해결하였다.

ID	SIZE	X1	Y1	Z1	X2	Y2	Z2	UPTIME	UPTISTORY	COUNT
1
2
3
4
5
6
7
8
9
10

(그림 12) 타임 포인트 질의 수행 결과

ID	SIZE	X1	Y1	Z1	X2	Y2	Z2	UPTIME	UPTISTORY	COUNT
1
2
3
4
5
6
7
8
9
10

(그림 13) 시공간 질의에 대한 수행 결과(intersect & time Interval)

〈표 5〉 기존의 시공간 모델과 제안된 모델의 문제점 비교

문 제	시공간 큐브	스냅샷 모델	시공간 복합 모델	3영역 모델	3D/4D 모델	관계형모델	객체지향 모델	제안된 모델
개별적인 지리 피쳐	√							
위상 관계 미포함	√			√				
많은 양의 데이터	√	√	√	√				
위상 관계를 구할 때 복잡한 알고리즘	√	√						
복잡한 시간 질의	√	√			√			
데이터 중복	√	√	√	√		√		
데이터 불일치		√						
변화를 구별하기 어려움		√	√					
빈약한 시간 resolution		√	√					
복잡한 모델			√	√	√			
지리 데이터 표현의 난이성						√		
시간이 포함되지 않은 모델		√						
구현되지 않음	√		√	√	√		√	

7. 결 론

실세계 객체를 정확하게 표현하기 위해서는 다차원의 공간 정보와 실세계에서 일어난 시간 정보를 유지하고 관리하여야 한다. 지금까지의 연구에서는 이러한 시공간 데이터를 표현하기 위한 모델로 시간분할 스냅 모델, 공간-시간 복합 모델, 3-영역 모델과 객체 지향 모델링 기법등이 제안 되었다. 이 중에서 시간에 따라 변하는 공간 객체를 가장 잘 표현하기 위한 방법은 객체 지향 모델링 기법이다. 그러나, 객체 지향 시공간 모델을 구현 시에 공간과 일반 속성, 또는 시간과 일반속성만으로 구성되어 있는 데이터베이스를 하나로 통합하는데 있어서 데이터의 이질성과 고유의 특성에 맞는 구조 때문에 실제로 구현되지 못했다. 이렇게 서로 다르게 구축된 데이터를 제공하기 위하여 개방형 지리정보 시스템에서 제공하는 2차원 공간 모델이 있으나, 이 모델은 2차원 공간으로만 한정되어 있어 3차원 공간 객체를 정확하게 표현할 수 없다.

따라서, 이 논문에서는 서로 다르게 구축된 시공간 데이터를 제공할 수 있는 제공자를 제안 한다. 이를 위해서 시간 모델을 표현하기 위한 ValidTime 클래스와 TGeometry 클래스를 설계했다. ValidTime 클래스가 나타내는 유효시간은 현실세계에 존재하는 객체에 대한 논리적인 시간을 의미하며, 객체가 발생하였거나 객체가 소멸된 시간을 가리킨다. TGeometry 클래스의 객체는 Geometry 클래스와 Valid Time 클래스로부터 상속 받은 시공간 객체에 대한 추상 클래스 이다. 두 클래스 설계를 바탕으로 시공간 모델을 제안한다.

제안된 시공간 모델은 시간과 3차원 공간 속성을 갖는 TGPoint, TGLineString, TGPolygonExtrusion과 TGSolid등의 시공간 클래스를 바탕으로 DGPoint, DGLineString과

DGSolid등의 시공간 이력객체들을 제안했다.

제안된 시공간 객체를 검증하기 위한 데이터 제공자 컴포넌트는 Oracle 데이터, Sql Server 데이터, 파일이나 기타 Tool에서 제공되는 데이터에 상관없이 이용가능 하도록 개방형 지리정보 시스템 인터페이스를 확장하여 설계 되었다. 따라서, 기존 시공간 모델의 문제점인 데이터 간의 통합을 통하지 않으면 실제 구현될 수 없는 문제점을 해결할 수 있었다.

검증에 사용된 시공간 데이터는 서울시 중구 지역의 가스 이력 정보이며, 시공간 질의를 수행하여 타당성을 보였다.

이 논문에서 제안된 시공간 데이터 모델은 객체 단위의 시공간 정보의 관리를 가능하게 하며, 실제로 구현되지 못했던 객체 지향 시공간 모델을 구현하였다. 구현에 사용된 데이터 제공자는 이질적으로 저장되어 있는 시공간 데이터를 저장, 갱신과 검색을 가능하게 하였다.

향후에는 제시된 모델과 구현된 프로토타입을 기반으로 시공간 데이터 관리 시스템을 구현하여 기존 시공간 시스템과의 비교 분석 및 성능향상 방안에 대한 연구를 수행할 것이다.

참 고 문 헌

- [1] M. Erwig, R. H. Guting, M. Schneider and M. Vazirgiannis, "Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases", Chorochronos Technical Report CH-97-8, 1997.
- [2] G. Langran, "A Framework for Temporal Geographic Information System", Cartographica, Vol.25, No.3, 1988.
- [3] G. Langran, Time in Geographic Information Systems, Taylor&Francis, 1992.

[4] C. Claramunt and M. Thuriault, "Managing Time in Gis: An Event-Oriented Approach", Recent Advances on Temporal Database, Clifford J. and Tuzhilin A. eds., Springer-Verlag, Zurich, pp.23-42, 1995.

[5] M.F. Worboys, H.M. Hearshshow, and D.J. Maguire, "Object-Oriented Modelling for Spatial Database", Int. Journal of GIS, Vol.4, No.4, 1990.

[6] Montgomery, L. "Temporal Geographic Information Systems Technology and Requirements : Where we are Today", Unpublished Master's Thesis, The Ohio State University, Columbus, Ohio. <http://thoth.sbs.ohio-state.edu/research/theses/montgomery.txt>. 1995.

[7] Flor Eugenia Narciso, "A spatiotemporal data model for incorporating time in geographic information systems (GEN-STGIS)", Ph.D's Thesis, University of South Florida, USA. 1999.

[8] Worboys, "Object-oriented approaches to geo-referenced information", International Journal of Geographical Information Systems. Vol.8, No.4, pp.385-399, 1994.

[9] Worboys, "A Unified Model for Spatial and Temporal Information", The Computing Journal, Vol.37, pp.26-34, 1994.

[10] Worboys, A Generic Model for Spatio-Bitemporal Geographic Information. pp.25-39, New York: Oxford University Press, 1998.

[11] 김은형, NGIS 1996년 최종보고서, <http://bora.dacom.co.kr/~eun1955/report/report.html>, 1996.

[12] U. S. Geological Survey, Spatial Data Transfer Standard, 1993.

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

[14] Hagerstrand, T. "What about People in Regional Science?", Reginal Science Association Papers, Vol. 24, pp.7-21, 1970.

[15] Hazelton, N., Bennett, L., and Masel, J. "Topological Structures for 4-dimensional Geographic Information Systems", Computers, Envrioment and Urban Systems, Vol.16, No.3, pp.227-237, 1992.

[16] Raafat, H., Yang, Z., and Gauthier, D. "Relational spatial topologies for historical geographical information", International Journal of Geographical Information Systems, Vol.8, No.2, pp.163-173, 1994.



김 상 호

e-mail : kimsh@dblab.chungbuk.ac.kr

1997년 충북대학교 컴퓨터과학과 졸업

1999년 충북대학교 대학원 전자계산학 (석사)

2004년 충북대학교 대학원 전자계산학 (박사)

2004년~2005년 충북대학교 연구원

2005년~현재 충북대학교 전기전자컴퓨터공학부 초빙교수

관심분야 : 시공간 데이터베이스, Web Visualization,

Component, GIS, 유비쿼터스, 이동객체 관리 기법



지 정 희

e-mail : jhchi@dblab.chungbuk.ac.kr

1999년 충주대학교 전자계산학과 졸업

2001년 충주대학교 대학원 전자계산학 (석사)

2003년 충북대학교 대학원 전자계산학 박사수료

관심분야 : 시공간 데이터베이스, Temporal GIS, 시공간 질의 최적화, 시공간 색인기법, 이동객체 관리 기법



류 근 호

e-mail : khryu@dblab.chungbuk.ac.kr

1976년 숭실대학교 전자계산학과 졸업

1980년 연세대학교 공학대학원 전자계산학 (석사)

1988년 연세대학교 대학원 전자계산학 (박사)

1976년~1986년 육군군수지원사전산실(ROTC 장교), 한국전자통신연구소(연구원), 한국방송통신대, 전산학과(조교수) 근무

1989년~1991년 Univ. of Arizona 연구원(TempIS Project)

1986년~현재 충북대학교 전기전자컴퓨터공학부 교수

관심분야 : 시간 데이터베이스, 시공간 데이터베이스, Temporal GIS, 객체 및 지식베이스 시스템, 지식기반 정보검색 시스템, 데이터 마이닝, 데이터베이스 보안 및 Bio-Informatics