이동객체 컴포넌트 설계 및 구현

Design and Implementation of Moving Object Components

임복자*, 김경숙*, 남광우**, 이기준*

Bog-Ja Lim, Kyoung-Sook Kim, Kwang-Wu Nam, and Ki-Joune Li

*부산대학교 전자계산학과 시공간 데이터베이스 연구실
{asnael,kskim}@stem.cs.pusan.ac.kr, lik@pnu.edu

**한국 전자 통신 연구원 4S통합기술 연구팀 선임연구원
kwnam@etri.re.kr

요약 위치기반서비스와 같은 응용분야에서는 이동객체를 저장·관리하고 처리할 수 있는 데이터베이스 관리시스템이 중요하다. 본 논문에서는 이동객체 데이터베이스 관리시스템 중 질의를 처리하기 위하여 기존의 연구들을 바탕으로, 기본적으로 제공되어야 하는 이동객체 데이터 형과 연산자를 정의한다. 또 정의된 데이터모델을 이용하여 이동객체 컴포넌트를 개발한다.

1. 서론

이동객체는 시간에 따라 위치나 면적 등의 공간적 속성이 연속적으로 변하는 객체를 말한다. 최근 들어와서 GPS, 이동단말기 등의 이동객체의 위치추적장치와 무선통신의 발달 그리고 지리정보시스템의 발달로 이동객체에 대한 서비스가 기술적으로 점차 가능하게 되었다. 이동객체의 대표적인 응용분야는 위치기반서비스이다. 위치기반서비스는 사용자의 위치정보를 이용하여 친구찾기, 경로검색, 주변정보검색, 응급서비스 등을 제공한다. 그러나, 이동객체에 대한 서비스는 아직 초기단계에 있고 많은 부분에 대한 연구가 필요하다. 특히 다양한 서비스를 제공하기 위해서는 이동객체의 위치정보를 데이터베이스화하고, 관련된 질의를 처리해 주는 이동객체를 위한 데이터베이스 시스템 구축이 중요하다. 이동객체 데이터베이스 시스템은 크게 이동객체를 저장하는 저장 컴포넌트, 이동객체의 현재 및 과거 위치에 대한 색인 컴포넌트, 질의처리 컴포넌트로 구성된다.

본 논문에서는 질의처리 컴포넌트개발을 위한 프리미티브 컴포넌트로써, 이동객체의 특성을 잘 표현할 수 있는 데이터 형과 질의를 처리하기 위한 기본연산자를 제공하는 이동객체 컴포넌트를 설계하고 구현하였다.

본 논문의 구성은 다음과 같다. 2장에서는 이동객체 데이터모델링을 위한 관련 연구를 소개한다. 3장에서는 본 논문에서 제공하는 이동객체 데이터모델을 제시하고, 4장에서는 이동객체에 대한 기본연산자 구현에 대해 살펴본다. 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

이동객체는 공간과 시간의 요소를 모두 가지고 있음으로 객체의 특성을 잘 표현할 수 있는 데이터 모델이 필요하다. 따라서,

본 논문에서는 기존의 공간데이터 모델과 시간데이터 모델, 그리고 현재까지 연구되었던 이동객체의 데이터 모델을 바탕으로 이동객체 컴포넌트의 데이터모델을 정의한다.

2.1 공간객체 데이터 모델

공간 데이터 모델은 이미 많은 연구가 되었고 표준화 작업도 이루어졌다. 대표적으로 OGC의 개방형 단순기하모델(이하 SFG:OpenGIS Simple Feature Geometry), ISO/TC 211, VPF, SDTS 등이 있는데, 현재 많은 응용분야에서 OGC의 SFG모델을 표준으로 삼고 있고, 개발된 컴포넌트도 존재한다. 따라서, 본 논문에서는 기존에 개발된 [1]의 구현 사양을 따른 OpenGIS Simple Feature Geometry 컴포넌트(이하 OGIS 컴포넌트)[6,7]와 연계하여 공간데이터에 대한 데이터모델과 공간연산자를 사용하도록 한다. 이동객체에 대한 데이터모델도 SFG의 모델을 기반으로 한다.

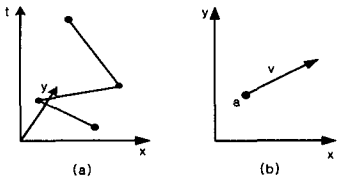
2.2 시간객체 데이터 모델

시간 데이터에 대한 모델은 [2,3]에서 살펴볼수 있다. [2]에서는 시간 데이터형으로 Instant와 Period를 정의하고, 13가지의 관계연산자를 정의하고 있다. [3]에서는 Interval의 시간데이터 형과 관계연산자를 정의하였다.

2.3 이동객체 데이터 모델

이동객체에 대한 데이터 모델에 관한 연구는 크게 두가지 방향으로 연구되었다. 첫째는 이동객체가 과거에서 현재까지 시간이 변함에 따른 위치 변화를 나타내기 위해서 <그림 1>의 (a)와 같이 (X, Y, T)의 시공간상의 다각선(polyline)으로 표현하는 궤적기반의 데이터 모델이다[5]. 두번째는 MOST (Moving Object Spatio-Temporal) 모델과 같이 이동객체의 현재위치를 유지관리하기 위한 함수 기반의 데이터 모델이다[4]. 이동객체는 시간에 따라 위치가 변하므로 현재위치는 계속적인 갱신이 발생하게 된다. 따라서, 갱신비용을 줄이기 위하여 이동객체의 현재위치를 시간에 따른 함수 형태로 나타내게 된다. 그리고, 함수를 이용하여 어느정도의 이동객체의 미래 위치도 예측할 수 있다는 장점이 있다.

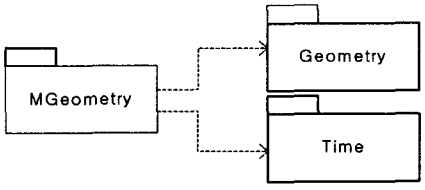
본 논문에서는 이동객체의 변하는 위치를 저장하고 관리하기 위하여 궤적기반의 모델을 바탕으로 이동객체 데이터 모델을 정의한다.



<그림 1> 이동객체 데이터 모델

3. 이동객체 컴포넌트 설계

이동객체 컴포넌트는 크게 공간데이터형과 연산자를 제공하는 Geometry 네임스페이스, 시간데이터형과 연산자를 제공하는 Time 네임스페이스, 이동객체의 데이터형과 연산자를 제공하는 MGeometry 네임스페이스로 구성된다.

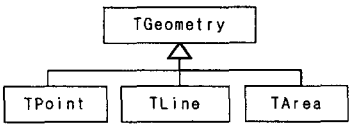


<그림 2> 이동객체 컴포넌트 네임스페이스

3.1 Geometry 네임스페이스(공간객체)

공간객체에 대한 기본적인 데이터 형이나 연산자는 OGC의 SFG모델를 바탕으로 설계한다. 그러나, OGC의 SFG모델은 3차원 공간객체에 대한 부분은 정의되지 않았다. 따라서, 3차원 공간객체에 대한 모델을 추

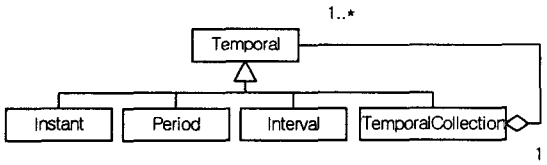
가해야 한다. 그러나, 이동객체는 단순한 3차원 공간객체가 아니라 2차원 공간과 시간이 합쳐진 (X,Y,T)차원의 시공간객체이다. 본 논문에서는 시공간객체로 시공간 점객체, 시공간 선분객체, 시공간 면객체로 한정하여 정의한다. 시공간객체에 대한 모델은 다음과 같다.



<그림 3> 시공간 데이터 모델

3.2 Time 네임스페이스 (시간객체)

Time 네임스페이스는 이동객체의 시간속성을 표현하기 위하여 기본적으로 순간시간을 나타내는 Instant, 시간구간을 나타내는 Period, '일주일', '한달'과 같은 일정기간을 표현하는 Interval을 정의하고 각 관계연산자와 집합연산자를 제공한다. 시간관계연산은 [2]에서처럼 13가지가 존재하고, 집합관계연산을 위하여 복합시간객체를 추가정의한다. <그림 4>는 시간객체에 대한 모델을 도식화한 것이다.

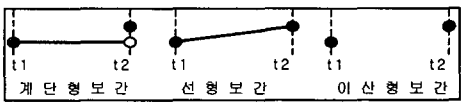


<그림 4> 시간객체 데이터 모델

3.3 MGeometry 네임스페이스 (이동객체)

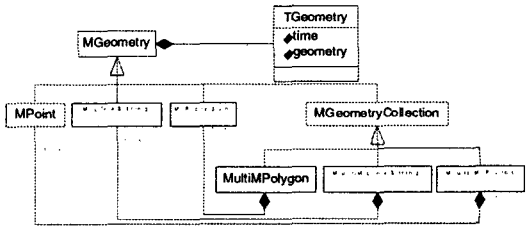
TGeometry는 순간시간(Instant)에 대한 공간객체를 나타내는 데이터 형으로써, 이동객체는 TGeometry의 시간적 순서를 따른 복합연관관계를 가지거나 이동객체의 집합연관관계를 가지는 집합체로 표현된다.

이동객체는 이론적으로 시간에 따라 연속적으로 객체의 속성이 변하지만 자료획득과정에서 이산적인 자료형태를 가진다. 따라서, 표현되지 않은 데이터는 서로 이웃한 데이터를 이용하여 유추하여 계산한다. 유추방법으로는 그림 5의 세가지 보간법을 기본적으로 사용하고 사용자가 추가할 수 있다.



<그림 5> 보간방법

MGeometry 네임스페이스는 이동객체에 대한 데이터 형으로써, 시간에 따라 위치가 연속적으로 변하는 공간객체를 나타내는 이동점객체(MPoint), 이동선객체(MLineString), 이동면객체(MPolygon)의 단순이동객체와 이동객체가 독립적으로 하나이상 모인 이동복합객체(MGeometry-Collection)를 정의하고, 각 데이터형을 처리할 수 있는 기본연산자 및 이동객체 관계연산자, 궤적 관계연산자, 기하연산자 등을 제공한다. 자세한 연산자 구현 설명은 4장에서 살펴보겠다. 다음은 이동객체의 데이터 모델에 대한 개략적인 다이어그램이다.

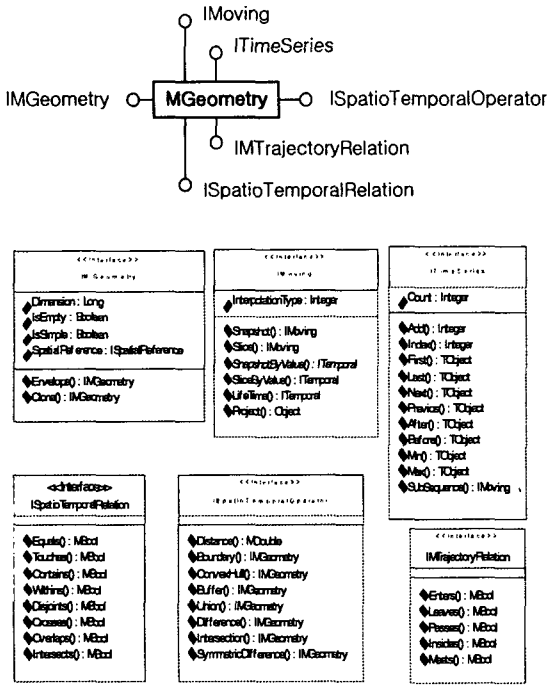


<그림 6> 이동객체 데이터 모델

4. 이동객체 컴포넌트 구현

본 논문에서 구현한 이동객체 컴포넌트는 기존에 개발된 OGIS 컴포넌트를 사용하는 구조를 가지고 있다. 따라서, OGIS 컴포넌트에서 제공하고 있는 인터페이스를 이용하여 이동객체 컴포넌트를 구현하였다. 그러나, 이동객체 연산자는 이동객체의 특성을 반영하여 다시 구현하였다. 예를 들면, 거리계산부분은 공간객체 사이의 거리계산

과 이동객체 사이의 거리계산과는 다르게 계산된다. <그림 7>은 이동객체 컴포넌트에서 구현된 기본연산자들의 인터페이스이다.

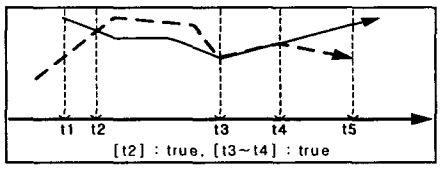


<그림 7> 이동객체 인터페이스

이동객체에 대한 기본적인 연산자는 IMoving 인터페이스에서, TGeometry의 시간순서에 대한 처리는 ITimeSeries 인터페이스에서 구현하였다. 그리고, 이동객체의 위상을 처리하기 위한 연산자는 ISpatioTemporalRelation 인터페이스에서, 이동객체의 움직인 궤적에 대한 연산자는 ITrajectoryRelation 인터페이스에서 구현되었다. 그러나, 본 논문에서는 연산자의 구현상 어려움으로 인하여 연산자의 계산부분에 이동선객체와 이동면객체가 위치가 변경되지 않는다는 제한을 두었다.

4.1 ISpatioTemporalRelation

이동객체의 위상관계를 연산하는 방법은 3차원 객체로 위상관계를 연산하는 방법과 공간객체의 위상관계의 시간적 변화로 처리하는 방법 두 가지로 처리할 수 있다. 본 논문에서는 후자의 방법으로 이동객체 사이의 위상관계가 시간에 따른 공간 위상관계를 만족하는지를 조사한다. 공간 위상 관계는 일치(Equal), 포함(Contain), 교차(Intersect), 분리(Disjoint) 등의 8가지로 정의한다. 따라서, 연산 후 얻어지는 결과값은 시간에 따라 변화되는 논리형 값을 가지게 된다. <그림 8>은 두개의 이동점객체의 교차관계를 나타낸다.



<그림 8> 이동점객체와 이동점객체의 교차관계

알고리즘 1 Intersect

```
Input
  g1, g2 : MGeometry
Output
  relation : MBool
Begin
  if ExistCommonLifeTime(g1,g2) then
    s1 ← Slice(g1);  s2 ← Slice(g2);
    intersection ←GetIntersection(s1, s2);
    MakeMBoolean(relation, intersection);
  end if
end
```

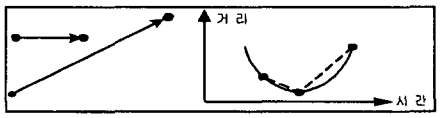
알고리즘 1은 이동객체 교차관계의 연산방법을 나타내고 있다. 다른 위상관계도 이와 비슷한 과정으로 처리된다.

4.2 ISpatioTemporalOperator

이동객체에 대하여 버퍼연산(Buffer), 거리계산(Distance) 등과 같은 기하연산자와 합집합, 교집합, 차집합 등의 집합연산자도 앞의 경우와 마찬가지로 시간에 따른 공간 기하연산자를 확장하여 구현한다. 본 논문

에서는 이동객체의 거리계산과 교집합 연산에 대해서만 설명하겠다.

이동객체는 시간에 따라 위치가 변하게 된다. 따라서, 이동객체 사이의 거리도 시간에 따라 변하게 된다. 이동객체 사이의 거리계산은 서로 같은 시간에 존재하는 공간객체 사이의 거리함수를 이용하여 최소거리를 측정함으로써 이루어 진다.

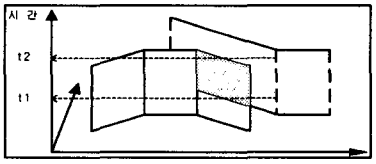


<그림 9> 이동점객체와 이동점객체의 거리

알고리즘 2 Distance

```
Input
  g1, g2 : MGeometry
Output
  value : MDouble
Begin
  if ExistCommonLifeTime(g1,g2) then
    s1 ← Slice(g1);  s2 ← Slice(g2);
    while
    notEmptyTimeInterval(s1.Seg,s2.Seg)
    then
        distanceF←GetDistF(s1.Seg,s2.Seg);
        GetInflection(distanceF, time, dist);
        MakeMDouble(value, time, dist );
    end while
  end if
end
```

이동객체의 집합연산을 수행하기 위해서는 우선 두 이동객체의 공통시간을 고려한다. 그리고 공통시간에서의 교차하는 영역이 존재하느냐 하지않는냐를 검사해 결과를 구한다. 예를 들어, 차집합의 경우 공통 시간이 존재하지 않으면 원래의 이동객체가 결과값이 되고, 공통 시간이 존재하면 공통되지 않는 시간에서는 원래의 값을 유지하면서 공통 시간에서는 상대방과의 차가, 차집합연산의 결과인 이동객체가 되는 것이다. 나머지 집합 연산도 이와 같이 연산을 수행할 수 있다. <그림 10>은 이동선객체 사이의 교집합 연산의 결과를 나타낸 것이다. 진하게 표현된 부분이 결과로서, 공통시간에서의 공통영역(교차부분)이다.



<그림 10> 이동선객체와 이동선객체의 교차부분

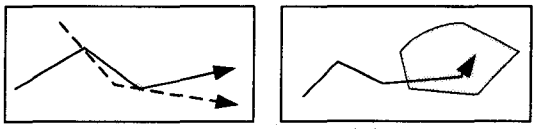
4.3 IMTrajectoryRelation

이동객체의 궤적 위상관계는 이동객체가 움직인 궤적 사이의 위상관계를 말한다. 궤적위상관계는 표1과 같이 5가지를 정의할 수 있다. 이외의 위상관계도 필요에 따라 확장 정의할 수 있다.

<표1> 궤적위상관계

궤적위상	위상 시퀀스
진입	외부 ≫ 내부
퇴거	내부 ≫ 외부
통과	외부 ≫ 내부 ≫ 외부
접촉	외부 ≫ 경계 ≫ 외부
내재	내부

본 논문에서는 이동점객체 사이의 궤적 위상관계는 시간에 따라 움직인 궤적을 나타내는 다각선(Polyline)을 이용하여 처리한다. 그러나, 이동선객체와 이동면객체 제약으로 인하여 이 경우는 이동객체의 정사영 공간객체를 이용하여 연산하게 되고, 이동선객체, 이동면객체 사이의 궤적 연산은 본 논문에서 정의하지 않았다.



(a) meet (b) enter

<그림 11>이동객체 사이의 궤적 위상관계

예를들어, <그림11>은 이동점객체에 대한 궤적위상관계를 나타낸다. (a)의 경우는 "1시에서 3시사이에 서로 만나는 사람을 찾아라"와 같은 질의를 처리하기 위한 접촉(Meet)연산자를, (b)의 경우는 "특정 지역에 들어간 자동차들을 찾아라"와 같은 질의를 위한 진입(Enter)연산자를 도식화한 것이다.

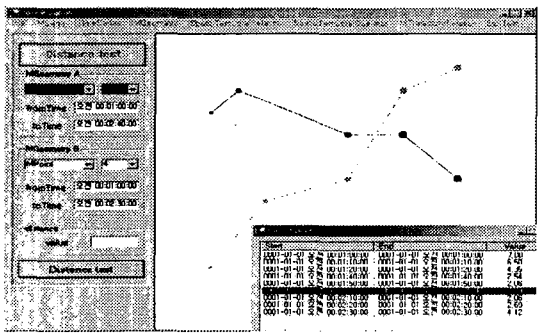
다음은 이동객체 궤적 연산 중 진입연산을 위한 알고리즘이다.

알고리즘 3 Enter

```
Input
  g1, g2 : MGeometry
Output
  relation : Bool
Begin
  if ExistCommonLifeTime(g1,g2) then
    s1 ← Slice(g1);  s2 ← Slice(g2);
    sequence ←GetStateSequence(s1, s2);
    if ExistEnterSequence(sequence ) then
      MakeBoolean(relation, true );
    end if
  end if
end
```

4.4 응용프로그램 예

다음의 그림들은 동일한 데이터에 대하여 두 이동점객체의 거리 연산을 수행한 결과를 보여준다.



<그림 12> 두 이동점객체의 거리 연산 예

5. 결론

본 논문에서는 이동객체의 질의를 처리하기 위한 기본 컴포넌트로, 이동객체에 대한 데이터 모델과 연산자를 제공하는 컴포넌트를 설계하고 구현하였다. 개발된 컴포넌트는 기존에 개발된 OGIS 컴포넌트를 기반으로 하였으며 컴포넌트 형태로써 다른 응용프로그램에 재사용할 수 있다. 하지만 이동객체에 대한 연산자 구현에서 몇 가지 제약사항을 두어 개발된 단점이 있다. 앞으로 다양한 이동객체를 표현하고 처리할 수 있는, 모델과 알고리즘의 개선이 필요하다.

참고문헌

[1] OpenGIS consortium. OpenGIS Simple Features Specification for OLE/COM Revision 1.1

[2] ISO. ISO/TC 211 Geographic information/Geomatics

[3] James F. Allen and George Ferguson. Actions and Events in Interval Temporal Logic. Journal of Logic and Computation 4(5): 531-579, 1994

[4] A. P. Sistla, O. Wolfson, S. Chamberlain and S. Dao. Modeling And Querying Moving Objects. ICDE 422-432, 1997

[5] R.H. Güting, M.H. Böhlen, M. Erwig, C.S. and M. Vazirgiannis. A Foundation for Representing and Querying Moving Objects. ACM TODS 25:1 1-42, 2000

[6] 황정래, 강혜경, 강인수, 이기준, OpenGIS의 Simple Feature Geometry 컴포넌트 구현, 개방형 지리정보시스템 학술회의 논문집, Vol.3.No.2, pp.139-156, 2000

[7] 김도현, 개방형 지리 정보 서비스 컴

포넌트 설계 및 구현, 제28회 한국정
보과학회 추계학술발표 논문집, 2001