

GML 기반의 이동객체 데이터모델 및 질의어 명세

정 원 일[†] · 배 해 영^{††}

요 약

지리정보의 상호 운용성을 제공하기 위해 제안된 GML은 무선 환경에서의 응용을 위해 다양한 동적 객체 스키마를 명세하고 있다. 이러한 GML 동적 객체 스키마는 시간의 변화에 따라 그 위치와 모양이 연속적으로 변화하는 이동객체의 표현을 지원하여, 이동객체의 위치 정보를 기반으로 다양한 응용 서비스를 제공하는 위치기반서비스에서 GML 활용도가 증대되고 있다. 특히, GML 질의어는 지리정보의 통합, 공간 연산자의 지원 등을 중심으로 많은 연구가 수행되었다. 그러나, 기존의 GML 질의어는 이동객체에 대한 질의를 고려하지 않고 있다. 따라서 이 논문에서는 GML 문서에 대한 이동객체 질의를 지원하기 위해 표준 XML 질의어인 XQuery를 확장한 이동객체 질의어를 제안한다. 또한, 제안 질의를 위한 데이터 모델 및 대수 연산 그리고 다양한 이동객체 연산자를 설계하고, 제안 질의어를 돌풍 관계 시스템에 적용하여 제안 질의어가 다양한 위치기반서비스에 효과적으로 사용될 수 있음을 보인다.

A Specification of a Moving Objects Data Model and Query Language over GML

Warnill Chung[†] · Hae-Young Bae^{††}

ABSTRACT

GML proposed to offer interoperability of geography information is specifying various dynamic object schema for application in the wireless environment. These GML dynamic object schema supports the expression of moving objects that the position and shape change consecutively over time, so practical use of GML is increased in location-based service that provides various application services based on location information of moving objects. In particular, the conventional query languages about GML have been studied with priority given to the integration of geographic information, the support of spatial operations, and etc. However, it lacks the support for moving objects queries. Therefore, in this paper, we propose the moving objects query language that supports moving objects queries for GML documents by extension of XQuery that is standard XML query language. Also, data model, algebra operation and various moving objects operators for the proposed query language are designed, and we apply the proposed query language to tornado government organization system to show that the proposed query language can be used effectively in various location-based service.

키워드 : GML, 이동객체 데이터모델(Moving Objects Data Model), 이동객체 질의어(Moving Objects Query Language)

1. 서 론

XML(eXtensible Markup Language)[1]은 차세대 웹 문서의 표준으로 정착되어 다양한 분야의 문서들이 XML을 기반으로 작성되고 있다. 특히, 지리정보의 교환 표준으로 XML을 이용하는 연구가 활발히 진행되고 있다[2]. 이러한 연구의 결과로 OGC(OpenGIS Consortium)에서는 XML을 기반으로 지리 정보의 저장 및 전송을 위한 인코딩 표준으로 GML(Geography Markup Language)[3-5]을 제안하였다. GML은 동적 객체스키마 등을 제공하여 무선 환경에서의 동적 객체에 대한 다양한 응용이 가능하게 하고 있다[5]. 이에, 최근 이동단말과 무선 컴퓨팅 기술의 발달로 향후 무선 인터넷의 킬러 어플리케이션으로 자리잡고 있는 위치기반서비스[7]에서 시간의 흐름에 따라 그 위치와 모양

이 변하는 이동객체[16, 17]를 표현하는 데 GML의 활용도가 크게 증가하고 있다. 특히, 효과적인 위치기반서비스를 제공하기 위해서는 이동객체에 대한 다양한 질의와 질의 처리에 필요한 연산을 수행할 수 있어야 한다.

GML 문서에 대한 질의어 연구는 GML[3, 4]의 명세에 따라 공간 정보 및 비공간 정보를 검색하기 위한 질의에 관한 연구[12-15, 21]는 널리 수행되어 왔으나, GML[5]의 명세에 따라 동적 객체 표현을 이용하여 GML 문서내 이동 객체 정보를 검색할 수 있는 질의에 대한 연구는 아직까지 고려되지 않고 있다. 기존의 GML 문서에 대한 질의어 연구는 웹 상에서 클라이언트가 URL 형식의 질의를 던지고 이를 이미지나 문서 형식으로 반환해주는 방식인 웹 맵 서버 인터페이스 구현 명세에 관한 연구[12, 20]와 웹 지리정보시스템에서 상호운용성을 제공하기 위해 GML 문서의 통합과 클라이언트에서의 시각화에 관한 연구[13]가 있다. 그러나 이러한 연구는 단순히 GML 문서를 통합하는 명세와 클라이언트측에서의 시각화에만 초점을 맞추고 있다. Cor-

※ 본 연구는 정보통신부의 대학 S/W 연구센터 지원사업의 연구 결과임.

† 준 회원 : 인하대학교 대학원 전자계산공학과

†† 종신회원 : 인하대학교 전자계산공학과 교수

논문접수 : 2003년 9월 8일, 심사완료 : 2003년 10월 29일

coles[14]에서는 SQL 구문을 확장하여 GML에 대한 공간 질의가 가능한 명세를 제안하고 있다. 제안된 기법에서의 SQL을 확장하는 방법이 GML 문서를 처리하기 위한 하나의 대안이 될 수 있으며, NCITS(InterNational Committee for Information Technology Standards)와 ISO(International Organization for Standardization) 데이터베이스 분과에서 SQL/XML[15] 표준에 대한 연구를 수행하고 있다. 그러나, 이런 접근 방법은 제안 질의어가 현재 XML 질의 표준에 부합되지 않으므로 XML 저장소(repository)에 직접적인 질의나 갱신에 적합하지 않고, GML 문서를 대상으로 확장된 SQL을 처리하기 위한 시스템을 새로이 개발해야 한다는 문제점이 있다. Vatsavai[21]에서는 GML 문서에 대한 공간 질의를 지원하기 위해 기존의 XML 질의어를 확장한 공간 질의어 명세를 제안하고 있다. 이 연구에서는 [22]에서 제시하고 있는 GML 공간 데이터와 공간 연산을 기존의 XML 질의어에 공간 함수의 형태로 확장하고 있으나, 이동객체에 대한 질의는 언급하고 있지 않다.

따라서 본 논문에서는 위치기반서비스에서 효과적으로 관리되어야 할 이동객체 정보를 포함하는 GML 문서를 대상으로 다양한 질의 표현 및 이동객체 연산을 지원하기 위해 XML 표준 언어인 XQuery를 확장한 이동객체 질의어를 제안한다. XQuery[11]는 기존에 제안된 Lorel[8], XML-QL[9], XQL[10] 등 XML 문서를 검색하는 질의 언어들 중 다양한 형태의 XML 데이터 소스에 폭넓게 적용될 수 있도록 설계되었기 때문에 차세대 XML 질의어의 표준으로 자리잡고 있어, 본 연구에서는 XQuery를 대상으로 질의 확장을 수행한다. 본 논문에서는 먼저 이동객체 질의어에서 적용할 데이터 모델과 제안 데이터 모델을 기반으로 수행될 대수 연산을 정의하고, 이동객체를 위한 다양한 연산자들을 설계하여 보다 풍부한 질의 표현이 가능하도록 한다. 또한 제안 이동객체 질의어를 돌풍관계시스템에 적용하여 이동객체 질의에 제안 이동객체 질의어의 효용성을 검증함으로써, 제안 이동객체 질의어가 다양한 위치기반 응용서비스에 효과적으로 적용될 수 있음을 보인다.

본 논문의 구성은 다음과 같다. 먼저 제 2장에서는 이동객체 데이터베이스와 기존의 GML 질의어와 XML 질의어에 대해 검토한다. 제 3장에서는 이동객체 질의어 명세를 위한 이동객체 데이터 모델과 이동객체 대수에 대해 기술한다. 제 4장에서는 XQuery를 기반으로 확장된 질의 언어에 대해 명세하고, 제 5장에서는 제안된 GML 기반의 이동객체 질의어에 대한 실험을 수행한다. 마지막으로 제 6장에서는 결론 및 향후 연구에 대해 기술한다.

2. 관련 연구

이 장에서는 본 논문에서 제안하는 GML 기반 이동객체 질의어 명세와 관련된 GML, GML 질의어, XML 질의어 그리고 이동객체에 대해 알아본다.

2.1 GML

OGC에서는 공간 정보와 비공간 정보를 포함한 지리정보 데이터를 저장하고 전송하기 위한 XML 기반 인코딩 표준으로 GML[3-5]을 발표하였다. OGC는 개방형 지리정보시스템 환경을 위해 지리정보 데이터와 응용프로그램간 표준 인터페이스를 제시하는 것을 목표로 2002년까지 GML v2.x를 발표한 이후, 2003년 2월 GML v3.0을 제안하였다. GML v2.x에서는 XML DTD(Document Type Definition)를 기반으로 하는 프로파일을 제공하는 GML v1.x를 포함하여, XML 스키마를 기반으로 복잡한 피처와 그들간의 관계를 인코딩할 수 있는 스키마를 제공한다[3, 4]. GML v3.0은 기존의 GML과의 호환성을 제공하며, 모듈화, 복합 지오메트리(geometry), 시공간 참조시스템, 위상, 메타데이터, 그리드 데이터, 측정 단위 등을 추가하였다[5]. 특히, GML v3.0에서는 동적 개체 스키마를 통해 위치, 시간, 그리고 위상 개념들을 포함으로써 위치기반서비스에서 효과적으로 관리되어야 하는 이동객체의 정보를 표현하는 데 활용될 수 있다.

2.2 GML 질의어

OGC에서는 지리정보 데이터의 웹 환경에서 상호운용성을 제공하고, 서로 다른 타입의 지리정보의 접근성을 제공하기 위해 웹 맵 서버 인터페이스 구현 명세를 제안하였다[12, 20]. OGC는 웹 기반 공간 데이터의 접근 표준을 정의하기 위해 웹 매핑(Web Mapping)이라는 분야를 제시하고 활용되고 있다. 즉, HTTP상에서 URL 형태의 질의를 입력받아 그 결과로 GIF, JPEG, GML 등의 웹 표준 데이터를 얻도록 한다. 웹 맵 서버 인터페이스 구현 명세에 관한 연구는 웹 환경에서 상호운용성을 제공하는 지리정보 서비스 개발을 위한 표준으로 제안되었지만[20], 입력의 형태가 URL로 제한됨으로써 데이터 소스가 GML 문서일 경우 이를 GML에 대한 질의로 변환해야 하고, 또 기존의 SQL을 지원하는 데이터 소스일 경우에도 적절한 변환 작업이 수행되어야 하는 문제점이 있다. 또한, URL의 질의 형태에 대한 변환을 수행하여도 GML이 가지는 구조적인 특성을 잃어버리게 되는 결과를 초래한다.

Shekhar[13]에서는 WMS(Web Map Server)와 GML을 기반으로 상호운용적인 웹 매핑 시스템을 제안하고 있다. 제안 시스템은 웹 환경에서 GML 클라이언트(client)에 의해 파싱(parsing)되고 렌더링(rendering)될 GML 출력물을 직접 생성한다. 또한 GML 클라이언트는 GML 문서에 대해 DOM 또는 SAX 파서(parser)를 이용하여 클라이언트측에서 점이나 영역질의 그리고 조인을 수행할 수 있도록 하고 있다. 그러나 이 기법 또한 GML에 대한 질의어에 대한 연구는 [12]와 같이 URL 형식을 취하고 따르고 있다.

Corcoles[14]에서는 SQL 구문을 확장하여 GML에 대한 공간 질의어 명세를 제안하고 있다. 제안 기법에서는 기존의 XML 질의어에 공간 정보를 표현하기 위한 데이터 모델과 대수 표현을 기술하고 있으며, SQL에 기반을 두고 *select-*

from-where 구문과 매우 유사한 질의어를 명세하고 있으며, 공간 연산자(disjoin, touches 등)와 비공간 정보를 위한 전통적인 연산자(=, >, <, ...) 를 제공하고 있다. 이 연구에서는 SQL 구문을 확장하여 GML 또는 XML 문서에 대해 공간 질의 및 비공간 질의를 수행할 수 있는 질의어를 제안하고 있으나, 이러한 접근 방법에서 GML 문서에 대한 질의 수행을 위해 제안 질의어를 XML 표준 질의어에 맞게 변형해야 하는 문제가 발생하므로 기존의 XML 저장소(repository)에 대한 직접적인 질의나 갱신이 불가능하다.

Vatsavai[21]에서는 GML 문서에 대한 공간 질의를 지원하기 위해 기존의 XML 질의어를 확장한 공간 질의어 명세를 제안하고 있다. 이 연구에서 [22]에서 제시하고 있는 GML 표준 공간 데이터 타입과 공간 연산을 XQuery의 FOR, LET, WHERE절에 공간 함수의 형태로 확장하였다. 제안 기법은 공간 정보를 포함하는 XML 문서에 대한 직접적인 질의를 수행할 수 있는 질의어를 제안하고 있으나, 제안 질의어에 대한 데이터 모델 및 질의 대수가 정의되지 않고 있으며, 더욱이 이동객체에 대한 질의는 언급하고 있지 않다.

2.3 XML 질의어

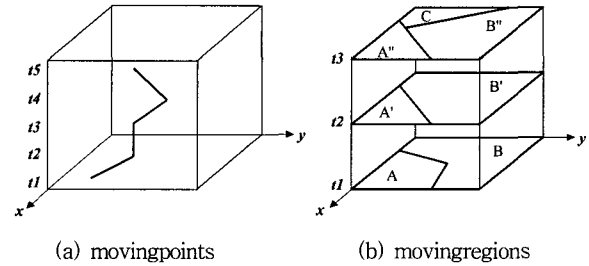
XML 질의어는 데이터베이스와 같이 XML 문서나 XML 문서의 컬렉션에 접근할 수 있다. W3C XML Query 워킹 그룹에서는 웹에서 XML 문서로부터 데이터를 추출하기 위한 유연한 질의 기능을 제공하기 위한 목적으로 XQuery 명세를 제공한다. XQuery는 XML-QL[9], XPath[12], XQL [10] 등의 XML 질의어 개념과 SQL의 개념을 추가하여 확장한 질의 언어로, 다양한 형태의 XML 데이터 소스에 포괄적으로 사용될 수 있도록 설계되어 있어 XML 질의 언어의 표준으로 자리잡고 있다[11].

따라서 본 연구에서는 질의 확장의 대상으로 XQuery를 이용한다. XQuery에서 경로 표현은 XPath의 축약형 구문을 사용하여 효과적으로 지역 경로 내에서 사용하는 축의 수를 줄일 수 있도록 한다. 경로 표현은 질의로서 홀로 사용될 수 있으며, 사용자들이 쉽게 질의를 읽고 쓸 수 있어야 한다는 요구 사항을 만족하며 URL상에 쉽게 삽입될 수 있는 장점을 가지고 있다. XQuery에서 FLWR 표현은 경로 표현의 취약점을 보완하여 노드 컬렉션의 반복, 변환, 임의의 결과 구성, IF-THEN-ELSE의 조건적인 처리, Inner/Outer 조인, 질의문 내에 함수, 사용자 정의 함수를 통한 확장 등을 제공한다. 또한 수학적, 논리적인 기본 연산자와 비교 연산자, UNION/INTERSECT/EXCEPT 연산자 등을 제공한다.

2.4 이동객체

시간의 흐름에 따라 객체의 위치 및 모양이 연속적으로 변화하는 특성을 지닌 데이터를 이동객체라 한다. 이러한 이동객체는 사람, 동물, 위성, 자동차, 배 등과 같이 시간에 따라 객체의 위치가 계속 변화하는 이동점과 영토, 숲, 태

풍, 기압권, 암 등과 같이 시간에 따라 위치와 모양이 변화하는 이동영역으로 분류된다[16, 17]. (그림 1)은 이동점과 이동영역을 설명하기 위한 그림이다.



(그림 1) 이동객체의 예

(그림 1)(a)에서 이동점은 시간의 변화(t1, t2, t3, t4, t5)에 따라 그 위치(x, y좌표)가 연속적으로 변화하고 있다. (그림 1)(b)는 t1에서의 이동영역(A, B)이 t2에서 그 이동영역(A', B')이 변경되었으며, 또한 t2에서의 이동영역(A', B')이 t3에서 (A'', B'', C)로 변경되는 과정을 나타내고 있다.

이동점에 대한 질의는 이동객체의 특정 시점에 대한 위치를 기준으로 수행되며, 이러한 이동점과 관련된 질의는 이동객체의 경로, 방향 및 거리등에 대한 질의가 존재한다. 이동영역에 대한 질의는 이동객체의 특정 시점에 대한 이동영역이 존재하는 위치와 모양을 표현하는 영역을 기준으로 수행되는 질의이다.

3. GML 기반 이동객체 데이터모델

최근 W3C에서 XML에 대한 데이터 모델과 대수에 대한 연구[18, 19]를 수행하고 있으나, 아직까지 WD(Working Draft) 상태에 머무르고 있다. [14]에서는 GML 문서의 구성 요소인 정점집합과 간선집합과 그들간의 상호관계를 방향성이 있는 그래프로 표현하고 있으며, 이 모델은 강력하고 단순한 형태를 취하고 있어 모델의 확장이 용이하다는 장점을 지니고 있어 본 논문에서는 [14]에서 제안한 데이터 모델을 이동객체 지원이 가능한 형태로 확장한다.

XML 문서에 대한 질의를 수행하기 위해서는 이동점 객체, 이동영역 객체 등과 같이 높은 추상화 단계의 객체를 표현할 필요는 없고, 정수나 문자열 등과 같이 단순 타입에 대한 표현이 필요하다. 그러나, GML에 표현된 이동객체 타입에 대한 질의 수행을 위해서는 이동객체 연산의 대상인 이동점, 이동영역과 같은 고수준의 객체에 대한 확인 및 문자와 숫자로 구성된 속성에 대한 질의를 위해서는 저수준 객체에 해당하는 단순 타입에 대한 확인이 필요하다.

본 논문에서는 [14]에서 제안한 데이터 모델에 새로운 정점(Vertex) 타입과 속성을 확장하며, 추가된 새로운 정점을 이동객체 정점이라 하고 V_{MO} 로 표기한다. GML 문서의 이동객체 타입은 본 논문에서 제안하는 데이터 모델의 V_{MO} 를 통해 표현된다. 그리고, 확장된 데이터 모델에서는 이동객

체 속성을 갖지 않는 XML 문서에 대해 이동객체 정점이 표현되지 않으므로, XML 문서에 대한 질의 수행시 원래 데이터 모델의 장점을 그대로 유지할 수 있다. 제안 데이터 모델은 논리적인 데이터 모델로 구성요소의 저장 방법에 대한 설명은 생략되며, [14]에서 이동객체 정보 확장을 위해 필요한 부분에 대해서만 기술한다.

GML 문서는 제안 데이터모델에서 방향성이 있는 그래프(directed graph)로 표현되며, $G = (V, E, A, R, O)$ 로 표기한다. V 는 그래프에서 엘리먼트와 값의 정점들로 구성된 집합으로, V 는 $V = (V_{element} \cup V_{geometry} \cup V_{MO} \cup V_{int} \cup V_{string} \dots)$ 로 형식화된다. $V_{element}$ 는 엘리먼트를 나타내는 정점이고, $V_{geometry}$ 는 지오메트리 타입을 표현하기 위한 정점이며, V_{MO} 는 이동객체를 표현하는 정점이다. V_{int} 와 V_{string} 은 데이터 값을 포함하는 정점들이다. E 는 XML 문서에서 엘리먼트간의 관계를 나타내는 엘리먼트 간선집합이고, A 는 XML 속성에 의해 표현된 엘리먼트와 값들 사이에 존재하는 관계를 나타내는 속성 간선집합이다. R 은 IDREF와 IDREFS 속성들을 통해 다른 엘리먼트로부터 참조되어지는 엘리먼트들간의 관계를 표현하는 참조 간선집합이고, O 는 부모 엘리먼트를 자식 엘리먼트로 연결하는 E , A 또는 R 에서 간선들 사이의 지역적 순서를 나타낸다.

3.1 정점 속성

정점은 데이터 값을 나타내는 정점인 $V_{type(v)}$, 지오메트리 엘리먼트를 표현하는 정점인 $V_{geometry}$ 과 이동객체 엘리먼트를 표현하는 정점인 V_{MO} , 그리고 나머지 엘리먼트들을 나타내는 정점인 $V_{element}$ 들이다.

각 XML 엘리먼트(element)는 엘리먼트 정점 $v \in V_{element}$ 로 표현되며, 엘리먼트 정점의 타입은 엘리먼트이다. v 는 유일하고(unique), 논리적인(logical) 그리고 불변의(immutable) 추상 식별자(abstract identifier)를 갖는다. 즉, v 는 고유의 역할을 규정짓는 키로써 유일해야 하고, 물리적인 저장으로부터 독립적인 논리적인 특징을 가지며 식별자를 통해 표현된 관계를 보존하기 위해 변하지 않는 특성을 갖는다. 또한 추상 식별자는 질의에 의해 직접 접근되지 않는다.

각 데이터 값은 값 정점 $v \in V_{type(v)}$ 로 표현되고, 값 정점은 유일한 식별자를 갖지 않고, 문자나 숫자 등에 해당하는 값과 타입 속성을 갖는다.

지오메트리 엘리먼트 정점은 $v \in V_{geometry}$ 로 표현되고, 지오메트리 엘리먼트 정점의 타입은 좌표(coordinate) 엘리먼트, 기본(primitive) 지오메트리 엘리먼트, 그리고 집단(aggregate) 지오메트리 엘리먼트 등이 있다.

이동객체는 이동점과 이동영역으로 분류됨을 관련연구에서 살펴보고, 또한 각 이동점과 이동영역에 대한 집단화된 데이터 타입이 존재한다. 이에 V_{MO} 에서는 이동점 엘리먼트(MOPoint), 이동영역 엘리먼트(MORegion), 다중 이동점 엘리먼트(MOMultiPoint), 다중 이동영역 엘리먼트(MOMul

tiRegion) 등 이동객체 엘리먼트에 대한 4개의 엘리먼트 타입을 정의한다. 그리고 정의된 각 타입에서는 GML 스키마에서 제공하는 엘리먼트들을 포함하게 되는 데, 공간 정보를 표현하기 위한 점(Point), 영역(Polygon), 다중점(MultiPoint), 다중영역(MultiPolygon) 등의 지오메트리 엘리먼트들과 시간 정보를 나타내기 위한 타임스탬프(timeStamp) 엘리먼트들을 갖는다. 기타 GML에서 제공된 스키마에서 참조되는 엘리먼트에 대한 설명은 생략한다. 아래는 4개의 이동객체 엘리먼트에 대한 타입 정의이며, [부록 1] 이동객체 스키마에서는 각 엘리먼트 타입을 정의한 스키마 명세이다.

[정의 1] 이동점 엘리먼트(MOPoint)은 시간의 변화에 대한 타임스탬프(gml : timeStamp) 값에 따라 그 점의 위치(gml : Point)가 변화하는 엘리먼트 타입으로 1개 이상의(gml : Point, gml : timeStamp)로 구성된다.

[정의 2] 이동영역 엘리먼트(MORegion)는 시간 변화 값인 타임스탬프에 따라 그 영역(gml : Polygon)의 위치와 모양이 변화하는 엘리먼트 타입으로 1개 이상의(gml : Polygon, gml : timeStamp)로 구성된다.

[정의 3] 다중 이동점 엘리먼트(MOMultiPoint)는 시간 속성인 타임스탬프의 변화에 따라 점들의 집단(aggregation)인 다중 이동점(gml : MultiPoint)을 구성하는 점들의 위치가 변화하는 엘리먼트 타입으로 1개 이상의(gml : MultiPoint, gml : timeStamp)로 구성된다.

[정의 4] 다중 이동영역 엘리먼트(MOMultiRegion)는 시간 속성인 타임스탬프의 변화에 따라 영역의 집단인 다중 이동영역(gml : MultiPolygon)을 구성하는 영역들의 위치와 모양이 변화하는 엘리먼트 타입으로 1개 이상의(gml : MultiPolygon, gml : timeStamp)로 구성된다.

XML 문서로부터 데이터 모델을 생성하기 위해, 즉 XML 엘리먼트나 명확한 값을 데이터 모델의 정점으로 변환하기 위해 함수 $vertex(x)$ 를 사용한다. 이 함수는 이동객체 정점을 지원하기 위해 [14]에 명시된 함수를 확장한다. $vertex(x)$ 의 정의는 아래와 같다.

[정의 5] 타입이 엘리먼트인 XML 엘리먼트 x 에 대해, $vertex(x)$ 는 엘리먼트의 추상 식별자와 타입으로 엘리먼트를 반환하며, $vertex(x) \in V \equiv (get_oid(x), element) \in V_{element}$ 로 표기된다.

[정의 6] x 의 타입이 지오메트리 타입일 경우, $vertex(x)$ 는 $value(x)$ 와 그 타입을 반환하며, $vertex(x) \in V \equiv (value(x), type(x)) \in V_{geometry}$ 로 표기된다. x 의 타입은 GML 스키마로부터 구한다.

[정의 7] x 의 타입이 이동객체 타입일 경우, $vertex(x)$ 는 $mo_value(x)$ 와 그 타입을 반환하며, $vertex(x) \in$

$V \equiv (mo_value(x), type(x)) \in V_{MO}$ 로 표기된다. x 의 타입은 [부록 1]에서 명세한 이동객체 스키마로부터 구한다.

[정의 8] 값 정점에 대해 $vertex(x)$ 는 값과 함께 구체적인 데이터 타입인 문자열이나 숫자 등을 반환하며, $vertex(x) \in V \equiv (x, type(x)) \in V_{type(v)}$ 로 표기된다.

3.2 간선 속성

제안 모델에서의 간선 구조는 [6, 14]로부터 확장하여 정의한다. 엘리먼트 간선(Element Edges : E)은 부모 엘리먼트와 자식 엘리먼트의 연관성을 나타내는 방향성 간선으로, 자식 엘리먼트는 데이터 값을 나타내는 데이터 정점이나 하위 엘리먼트 정점이 될 수 있다. 각 간선 $e \in E$ 는 ($name \in T_{name}, parent \in (V_{element}, V_{geometry}, V_{MO}), child \in V$)로 표기된다. name은 엘리먼트 간선의 이름을 나타내고, parent는 간선의 부모 정점을 나타낸다. $\sim data$, $\sim comment$, $\sim PI$ 간선의 경우는 child가 데이터 정점이 되고, 그렇지 않으면 경우에 child는 간선의 하위 엘리먼트 정점이 된다. 엘리먼트 간선의 이름 중 $\sim data$ 는 데이터를 포함하는 데이터 정점과 엘리먼트 정점을 연결하는 간선이고, $\sim comment$, PPI 는 XML 문서에 대한 코멘트와 처리 명령들을 포함한 정점을 포함하는 간선이다.

속성 간선(Attribute Edge : A)은 엘리먼트와 속성의 관계를 나타내는 방향성이 있는 간선으로, 속성 간선은 엘리먼트 간선과 같이 parent, value, name, type 같은 속성을 갖는다. name은 속성 이름을 나타내고 parent는 속성을 갖는 엘리먼트이고 value는 속성 값을 나타낸다. 속성 간선은 ($name \in T_{name}, parent \in (V_{element}, V_{geometry}, V_{MO}), child \in V_{type(v)}$)로 표시한다.

참조 간선(Reference Edge : R)은 엘리먼트 사이의 참조 관계를 나타내는 방향성 간선으로, 속성 중 IDREF, IDREFS, foreign key, Xlink, ULI와 같은 참조 타입으로 식별되는 경우 그래프 모델에서 참조간선집합 R에 속한다. 참조 간선은 기본적인 속성으로 parent, child, type을 가지며, refedges는 참조 정보를 제공하는 속성 또는 엘리먼트 간선에 대한 정보를 나타낸다. 그리고, 타입 속성은 간선이 참조 간선임을 나타내고 Xlink 또는 URI와 같은 참조 종류를 나타낸다. 참조 간선은 ($parent \in V_{element}, refedged \in P(E \cup A), child \in V$)로 표기된다.

3.3 순서 속성

제안 모델에서는 [6]에서 설계된 순서를 적용한다. O는 모든 간선들이 같은 간선 집합으로 분류되고, 동일한 부모를 공유하는 경우에 한해서 간선들 사이의 순서를 정의한다. O는 아래와 같이 형식화될 수 있고, succ는 순서상 e의 후행자를 의미한다.

$$O \equiv \{(e \in E \cup A \cup R, succ \in E \cup A \cup R) \mid parent(e) = parent(succ) \wedge (e \in E \wedge succ \in E \vee e \in A \wedge succ \in A \vee e \in R \wedge succ \in R)\}$$

참조 간선들의 경우 다중치 참조(multi-valued references)들의 각 참조들간의 순서는 참조 규칙에 의해 정의되어지고, refedges간의 순서는 다른 참조들간의 순서를 결정한다. 즉, 동일한 타입의 refedges를 갖는 참조들간의 총 순서는 오직 하나이고, 모든 참조 간선들간의 순서는 부분적임을 의미한다. 순서에서 선행 간선은 $pred(x) \in E \cup A \cup R \equiv e(o) \mid \exists o \in O: succ(o) = x$ 형식에 의해 결정된다.

3.4 데이터 모델의 적용

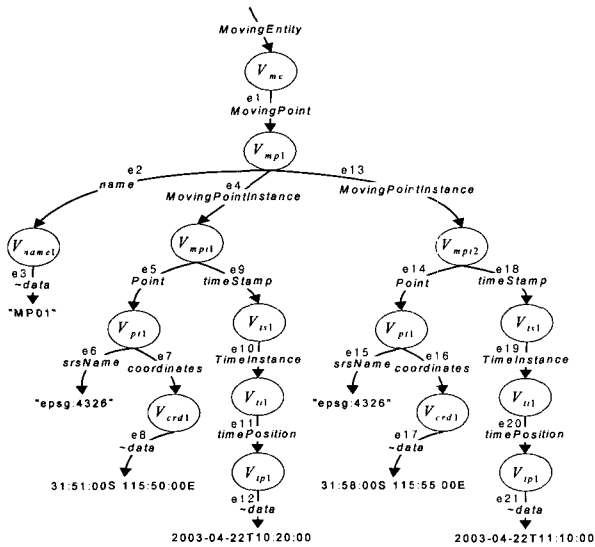
본 절에서는 지금까지 논의한 데이터 모델을 기반으로 작성된 예제 GML 문서와 함께 GML 문서를 제안 데이터 모델로 사상시킨 그래프를 제시한다.

```
<MovingEntity>
  <mo : MovingPoint>
    <name> MP01 </name>
    <mo : MovingPointInstance>
      <gml : Point srsName = "epsg : 4326">
        <gml : coordinates> 31 : 51 : 00S 115 : 50 : 00E
      </gml : coordinates>
    </gml : Point>
    <gml : timeStamp> <gml : TimeInstant>
      <gml : timePosition> 2003-04-22T10 : 20 : 00
    </gml : timePosition>
    </gml : TimeInstant> </gml : timeStamp>
  </mo : MovingPointInstance>
  <mo : MovingPointInstance>
    <gml : Point srsName = "epsg : 4326">
      <gml : coordinates> 31 : 58 : 00S 115 : 55 : 00E
    </gml : coordinates>
    </gml : Point>
    <gml : timeStamp> <gml : TimeInstant>
      <gml : timePosition> 2003-04-22T11 : 10 : 00
    </gml : timePosition>
    </gml : TimeInstant> </gml : timeStamp>
  </mo : MovingPointInstance>
</mo : MovingPoint>
</MovingEntity>
```

(그림 2) 이동객체 정보를 표현한 GML 문서

(그림 2)에서는 이동객체 정보를 포함하는 GML 문서의 예로, 이동점 객체가 시간에 따라 그 위치가 변화하는 정보를 나타내고 있다.

(그림 3)에서는 (그림 2)의 GML 문서에 대해 제안 데이터 모델에 적용한 데이터 그래프를 나타내고 있다. (그림 3)의 그래프에서 V_{pt} , V_{cds} , V_{ls} , V_{ti} , V_{tp} 는 각각 GML 문서 스키마에서 제공하는 Point 타입, coordinates 타입, timeStamp 타입, TimeInstant 타입, 그리고 timePosition 타입을 나타내는 정점이고 V_{me} , V_{mp} , V_{mpi} 는 (그림 2)의 MovingEntity 엘리먼트, MovingPoint 엘리먼트, MovingPointInstance 엘리먼트를 나타내는 정점들이다.



(그림 3) GML 문서의 데이터 모델 사상

3.5 대수 연산

XML은 반 구조적인 데이터에 속하며, 구조가 약하고 구조의 변경이 빠르므로 예외사항이 발생할 수 있다. 그러므로 XML 대수는 그래프 구조, 타입과 그래프 구조의 이질성, 참조 관계, 순서와 같은 XML이 갖는 특성에 대한 연산을 처리할 수 있어야 한다. 따라서 본 장에서는 제안 데이터 모델에 대한 대수 연산을 정의한다.

제안된 데이터 모델에 대한 대수 연산은 [6]에서 정의하고 있는 항해(navigation : ϕ), 구조순환(Kleene Star : *), 맵(map), 선택(selection), 조인(Joins), 중복제거(distinct), 정렬(sort), 무순서(unorder) 등의 연산들과 결과 구조체를 생성하기 위한 연산들을 사용하며, 제안 대수 연산은 완전성(Completeness)을 제공하기 위해 이동객체 정점들에 적용될 이동객체 연산자를 정의한다.

이동객체 연산자는 이동점, 이동영역, 다중 이동점, 다중 이동영역 등 이동 객체의 공간 속성과 유효시간 속성을 기반으로 제공되며, <표 1>에서 이동객체 연산자들을 소개하고 있다. 연산자 표기를 위해 사용한 기호 중에서 MP는 이동점과 다중 이동점 객체를 표기하기 위해 사용되고, MR은 이동영역과 다중 이동영역 객체를 표시한다. P는 이동점과 다중 이동점 객체, R은 이동영역과 다중 이동영역에 대해 각 이동객체의 특정 시간과 그 특정 시간에 해당하는 지오메트리 객체의 정보를 표현한다. 또 질의시간 구간을 Tp, 질의 시점을 Tc, 객체의 좌표(coordinates)를 location이라 한다.

<표 1>에서 MOFirst는 이동객체의 처음 값부터 i만큼의 떨어진 객체 정보를 반환하는 연산자이고, MOCCount는 이동객체의 위치 정보가 변경된 수를 연산하며, MOBefore는 특정 질의시간을 기준으로 가장 가까운 이전 객체 정보를 반환하는 연산자이다. MODistance는 특정 질의 시점에서 두 이동객체간의 거리를 연산하고, MOCenter는 질의 시점

<표 1> 이동객체 연산자

연산자	입력	출력	도메인
MOFirst	a, i	b	$a \in MP \vee MR, i \in \text{int}, b \in P \vee R$
MOLast	a, I	b	$a \in MP \vee MR, i \in \text{int}, b \in P \vee R$
MOCCount	A	i	$a \in MP \vee MR, i \in \text{int}$
MOBefore	a, Tc	b	$a \in MP \vee MR, b \in P \vee R$
MOAfter	a, Tc	b	$a \in MP \vee MR, b \in P \vee R$
MODistance	a, b, Tc	r	$a \in MP \vee MR, b \in MP \vee MR, r \in \text{real}$
MOCenter	a, Tc	b	$a \in MP \vee MR, b \in \text{location}$
MOMin	a, Tp	b	$a \in MP \vee MR, b \in PR$
MOMax	a, Tp	b	$a \in MP \vee MR, b \in PR$
MOProjectTime	a	Tc	$a \in P \vee R$
MOProjectValue	a	b	$a \in P \vee R, b \in \text{location}$
MOTime	a, b	Tc	$a \in MP \vee MR, b \in P \vee R$
MOValue	a, Tc	b	$a \in MP \vee MR, b \in P \vee R$
MOMeetTime	a, b, Tp	Tc	$a \in MP \vee MR, b \in MP \vee MR$
MOPeriod	a, Tp	b	$a \in MP \vee MR, b \in MP \vee MR$
MOContains	a, b, Tc	c	$a \in MP, b \in MR, c \in P$
MOTrajectory	a, Tp	b	$a \in MP, b \in \{\text{location}\}$
MOLength	a, Tp	r	$a \in MP, r \in \text{real}$
MONearest	a, b, Tc	c, r	$a \in MP, b \in \{MP \vee MR\}, c \in P \vee R, r \in \text{real}$
MOFarthest	a, b, Tc	c, r	$a \in MP, b \in \{MP \vee MR\}, c \in P \vee R, r \in \text{real}$
MOSpeed	a, Tc	r	$a \in MP \vee MR, r \in \text{real}$
MODirection	a, Tc	r	$a \in MP \vee MR, r \in \text{real}$
MOTraverse	a, Tp	b	$a \in MR, b \in R$
MOArea	a, Tc	r	$a \in MR, r \in \text{real}$
MOPerimeter	a, Tc	r	$a \in MR, r \in \text{real}$

에서 이동객체의 중앙점을 구하는 연산이다. MOMin는 질의시간 구간동안 이동객체의 최소값을 갖는 위치정보를 반환하고, MOProjectTime과 MOProjectValue는 이동객체 인스턴스에서 각각 시간 값과 위치정보 값을 반환하는 연산자이다. MOTime은 이동객체가 특정 위치를 지난 시간을 반환하고, MOValue는 질의 시점에서 이동객체의 정보를 반환한다. MOMeetTime은 질의시간 구간동안 이동하는 두 객체가 만나는 시간을 연산하고, MOPeriod는 질의시간 구간동안 이동객체 인스턴스들의 집합을 반환한다. MOContains는 질의 시점에 이동점이 이동영역에 존재했던 위치를 반환하는 연산자이다. MOTrajectory는 질의시간 구간동안 이동한 객체의 궤적 정보를 반환하고, MOLength는 질의시간 구간동안 이동점 또는 다중이동점이 이동한 길이를 반환하며, MONearest는 질의 시점에서 이동점과 가장 가까운 이동객체 정보와 거리를 반환한다. MOSpeed는 질의시점에서 이동점 또는 다중이동점의 속도를 반환하고, MODirection은 질의 시점에서 이동점 또는 다중이동점의 이동방향을 반환한다. MOTraverse는 이동영역 또는 다중이동영역이 질의시간 구간동안 이동한 영역을 반환하고, MOArea는 질의 시점에서 이동영역 또는 다중이동영역의 면적을 연산하며, MOPerimeter는 질의 시점에서 이동영역 또는 다중이동영역의 둘레길이를 반환한다.

다음 (질의 1)은 시간에 따라 그 위치가 이동하는 이동점 정보를 포함한 (그림 2)의 GML 문서를 대상으로 이동점 질의를 표현한 예이다.

(질의 1) “가장 최근 입력된 이동점 MP01의 위치정보를 구하라.”는 질의는 아래의 대수로 표현될 수 있다.

```
A := ϕ[E, MovingPoint] (*child( ϕ[E, #](x))(x : root)) ;
B := σ [last(value( ϕchild([E, ~data](child( ϕ[E, name](e))))))
      = "MP01", 0)](e : child(A)) ;
C := child( ϕ[E, Point](B)) ;
```

위 예는 루트정점으로부터 MovingPoint 간선을 향해한 결과 {e4, e13}이 변수 A에 결합된다. 변수 B는 변수 A에 결합된 간선집합에 대해 하위간선 name의 데이터 정점 값이 “MP01”인 선택 조건을 만족하는 정점집합 {V_{mpi1}, V_{mpi2}} 중에서 last(⋯, 0) 연산을 통해 가장 최근에 입력된 이동점인 V_{mpi2}이 변수 B에 결합된다. 변수 C에는 변수 B에 결합된 정점 집합의 하위 간선 Point에 대한 정점인 V_{crd1}가 결합된다. 따라서 이 질의의 결과는 V_{crd1} 정점이 포함하는 엘리먼트에 대한 정보를 다음과 같이 반환한다.

```
<gml : coordinates> 31 : 58 : 00S 115 : 55 : 00E
</gml : coordinates>
```

4. 이동객체 질의어

3.5절에서 설명한 대수 연산은 질의를 표현하기에 매우 복잡하고, 처리 과정을 표현하기에는 비효율적인 면을 가진다. 따라서 본 장에서는 제안하는 이동객체 질의어는 XQuery[11]를 확장하여 사용자에게 좀 더 익숙한 인터페이스를 제공하고, XQuery를 지원하는 기존의 시스템에 쉽게 적용 가능하도록 한다.

제안 질의어는 XQuery의 FLWR 표현식에서 FOR, LET, WHERE, RETURN절에 <표 1>의 이동객체 연산자들을 사용한다. 다음은 <표 1>의 이동객체 연산자들을 사용자 질의에 적용시킨 예제에 대해 설명한다. 예제 질의는 돌풍 지역, 이동중인 사람, 구급차를 대상으로 하며, 사용되는 스키마 정보는 (그림 4)와 같다.

```
<schema>
<include schemaLocation = "MovingObject.xsd"/>
<element name = "Tornado" type = "mo : MORegionType"/>
<element name = "Ambulance" type = "mo : MOPointType"/>
<element name = "Person" type = "mo : MOPointType"/>
</schema>
```

(그림 4) 예제 질의를 위한 스키마

(그림 4)에서 Tornado는 이동영역 객체로 [부록 1]의 MovingObject 스키마 파일의 MORegionType으로 정의되고, Ambulance와 Person은 이동점 객체로 MOPointType으로 정의된다.

(질의 2) “홍길동의 가장 최근 위치를 구하라.”

```
FOR $p IN document("Person.xml") // Person[name = "홍길동"]
RETURN <PersonLocation>
      <location> MOLast($p/mo : MOPointType, 0)
      </location>
</PersonLocation>
```

(질의 2)에서는 사람들의 이동점 정보를 가지는 Person.xml에서 name이 홍길동인 Person 정점들에 대해 \$p/mo : MOPointType과 0을 입력으로 MOLast 연산을 적용하여 시간적으로 가장 최근의 위치정보를 검색하게 된다.

(질의 3) “현재 홍길동의 위치에서 가장 가까이 있는 구급차를 검색하라.”

```
FOR $p IN document("Person.xml") // Person[name = "홍길동"],
  $a := document("Ambulance.xml") // Ambulance
LET $c := $a
WHERE MONearest($p/mo : MOPointType, $a/mo : MOPointType, now)
RETURN <NearestAmbulance>
      <ambulance> {$c} </ambulance>
</NearestAmbulance>
```

(질의 3)에서는 Person.xml에서 name이 홍길동인 Person 정점과 Ambulance.xml에서 Ambulance 정점에 대해 \$p/mo : MOPointType과 \$a/mo : MOPointType 그리고 now를 입력으로 MONearest 연산을 적용하여 현재 시점을 기준으로 홍길동의 위치 정보로부터 가장 가까운 구급차의 위치 정보를 검색한다. 여기서 now는 현재 시간을 의미하는 시간 상수이다.

(질의 4) “2003년 4월 23일 0시 0분 0초부터 2003년 4월 23일 12시 0분 0초까지 유효시간동안 구급차 A가 이동한 궤적을 구하라.”

```
FOR $a IN document("Ambulance.xml") // Ambulance[name = "A"]
RETURN <AmbulanceTrajectory>
      <trajectory>
        MOTrajectory($a/mo : MOPointType,
          MOPeriod("2003-04-23T00 : 00 : 00",
            "2003-04-23T12 : 00 : 00"))
      </trajectory>
</AmbulanceTrajectory>
```

(질의 4)에서는 Ambulance.xml에서 name이 A인 Ambulance 정점에 대해 \$a/mo : MOPointType과 두 질의 시점을 입력받아 질의시간 구간을 연산하는 MOPeriod 연산의 결과를 입력으로 MOTrajectory 연산을 적용하여 2003년 4월 23일 0시 0분 0초부터 2003년 4월 23일 12시 0분 0초까지 구급차 A가 이동한 궤적을 검색한다.

(질의 5) “현재까지 Tornado T의 피해 면적을 검색하라.”

```
FOR $t document("Tornado.xml") // Tornado[name = "T"]
RETURN <TornadoStatus>
      <area> MOTraverse($t/mo : MOPolygonType,
        MOPeriod(first, now)) </area>
</TornadoStatus>
```

(질의 5)의 표현에서는 Tornado.xml에서 name이 T인 정점에 대해 피해 면적을 구하기 위해 Tornado의 영역 정보인 \$t/mo : MOPolygonType과 MOPeriod 연산의 결과를 입력으로 MOTraverse 연산을 수행한다. 여기에서 MOPeriod는 시작 시간 상수 first와 현재 시간 상수 now를 입력으로 질의시간 구간을 연산한다.

(질의 6) "Tornado T의 현재 진행 방향과 이동 속도는?"

```
FOR $t document("Tornado.xml") // Tornado[name = "T"]
RETURN <TornadoDirSpeed>
  <direction> MODirection($t/mo : MOPolygonType, now)
  </direction>
  <speed> MOSpeed($t/mo : MOPolygonType, now)
  </speed>
</TornadoDirSpeed>
```

(질의 6)에서는 Tornado.xml에서 name이 T인 정점에 대해 현재 시간을 기준으로 Tornado T의 진행 방향을 연산하기 위해 Tornado의 영역 정보인 \$t/mo : MOPolygonType과 현재 시간 상수 now를 입력받아 MODirection 연산을 수행한다. 또한 \$t/mo : MOPolygonType과 now를 입력으로 MOSpeed 연산을 수행하면 Tornado T의 이동 속도를 얻을 수 있다.

(질의 7) "현재 구급차 A가 이동중인 Tornado T와 만날 위치와 이동해야 할 거리를 구하라."

```
FOR $a document("Ambulance.xml") // Ambulance[name = "A"],
  $t document("Tornado.xml") // Tornado[name = "T"]
RETURN <ATDistance>
  <location>
    MOValue($a/mo : MOPointType,
      MOMeetTime($a/mo : MOPointType,$t/mo :
        MOPolygonType))
  </location>
  <distance>
    MODistance(MOValue($a/mo : MOPointType,
      now),
      MOValue($a/mo : MOPointType,
        MOMeetTime($a/mo : MOPointType,
          $t/mo : MOPolygonType)))
  </distance>
</ATDistance>
```

(질의 7)에서는 Ambulance.xml에서 name이 A인 Ambulance 정점과 Tornado.xml에서 name이 T인 Tornado 정점에 대해 두 이동객체가 만나게 되는 지점에 대한 연산은 Ambulance 위치 정보인 \$a/mo : PointType와 Tornado 영역 정보인 \$t/mo : PolygonType 를 입력으로 각 이동객체의 이동방향과 속도를 연산하고 이로부터 미래에 두 이동객체가 만나게 되는 시점을 MOMeetTime에서 연산하고, Ambulance A의 위치 정보와 MOMeetTime의 시간 정보를 입력으로 MOValue 연산을 수행하여 Ambulance A와 Tornado T가 만날 지점을 얻게된다. 또한 Ambulance A가 Tornado T와 만날 지점까지의 거리는 A의 위치 정보와 현재 시간 상수를 입력으로 MOValue을 수행하여 얻은 현재 A의 위치정보와 앞서 설명한 두 이동객체가 만나게 될 지점을 입력으로 MODistance 연산을 수행함으로써 결과를 유도할 수 있다.

(질의 8) "현재 Tornado T의 영향권내에 속하는 사람을 검색하시오."

```
FOR $t document("Tornado.xml") // Tornado[name = "T"],
  $p document("Person.xml") // Person
WHERE MOContains($p/mo : MOPointType,
  $t/mo : MOPolygonType, now) == 1
RETURN <PersonInTornado> <person> $p/name </person>
</PersonInTornado>
```

(질의 8)은 FOR 절에서 Tornado.xml에서 name이 T인 Tornado 정점과 Person.xml에서 Person 정점에 검색하고, WHERE 절에서 T의 현재 영역 정보와 Person의 위치를 입력으로 MOContains 연산을 수행한다. MOContains 연산으로부터 얻은 결과는 T의 영향권 영역내에 존재하는 사람들에 대한 정보로, RETURN절에서는 검색된 사람의 name에 대한 출력 결과를 형식화한다.

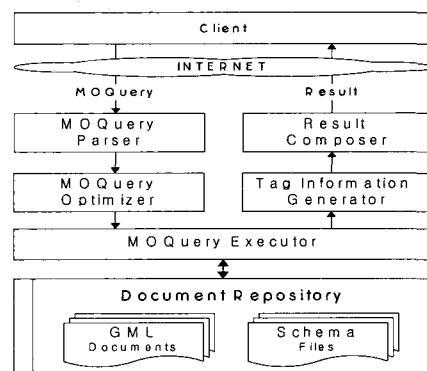
5. 실험 및 평가

이 장에서는 제안 이동객체 질의어를 처리하기 위한 시스템을 설계하고, 이를 돌풍 관제 시스템에 적용하여 제안한 이동객체 질의가 위치를 기반으로 하는 다양한 형태의 서비스에 응용될 수 있음을 보인다. 그리고 이를 통하여 기존 질의어와 제안 질의어를 비교 평가한다.

실험에 적용한 돌풍 관제 시스템에 대한 응용은 MS Windows 2000에서 Visual C++ 6.0을 이용하여 구현하였다. 실험에 이용된 이동객체 데이터는 2003년 4월 23일 0시부터 2003년 4월 23일 18시를 유효시간 구간으로 설정하고, 이 유효시간동안 사람과 구급차에 대해 2분, 돌풍에 대해서는 10분의 폴링 타임을 주기로 임의 획득된 데이터이다. 이동객체 중 사람들에 대한 이동 정보는 Person.xml, 구급차에 대한 이동정보는 Ambulance.xml, 그리고 돌풍에 대한 이동정보는 Tornado.xml에 유지된다. 3.4절에서 이동점 객체에 대한 정보를 포함하는 GML 문서를 (그림 2)에서 보여주고 있다.

5.1 이동객체 질의처리

본 절에서는 4장에서 제안한 이동객체 질의어 처리에 대해 설명한다.



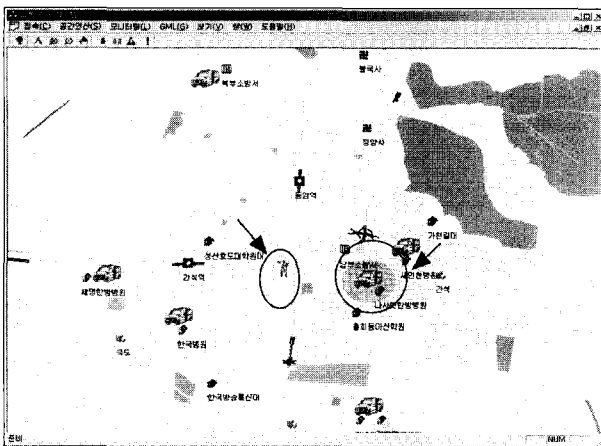
(그림 5) 이동객체 질의 처리 시스템의 구조도

(그림 5)는 이동객체 질의 처리 시스템의 구조도로써 이동객체 질의인 MOQuery는 MOQuery Parser에 입력된다. MOQuery Parser는 Document Repository 정보를 바탕으로 입력된 이동객체 질의를 파싱한다. 이 결과는 트리의 형태로 MOQuery Optimizer에게 전달된다. MOQuery Optimizer에서는 간단한 최적화 작업을 거쳐 생성된 플랜을 MOQuery Executor에게 전달한다. MOQuery Executor는 질의 수행을 통해 결과를 반환하는 역할을 수행하는 데, 여기에서 Document Repository에 접근을 위해 MSXML(Microsoft XML Parser) 4.0을 이용하였다. Document Repository는 GML 문서들과 이 문서들에 대한 각종 구조 파일을 유지하고 있다. MOQuery Executor를 통해 산출된 결과는 Tag Information Generator에서 MOQuery의 RETURN문에 작성한 질의 결과의 구조에 따라 엘리먼트를 생성 및 질의 결과의 형식화 작업을 수행한다. Result Composer는 MOQuery Executor를 통해 유도된 질의 처리 결과와 Tag Information Generator에서 질의의 최종 결과로 형식화된 문서를 완성한다.

5.2 이동객체 질의수행

이 절에서는 5.1절의 이동객체 질의처리 시스템을 이용한 돌풍관제시스템에서 4장의 이동객체 질의어를 처리한 결과를 보인다.

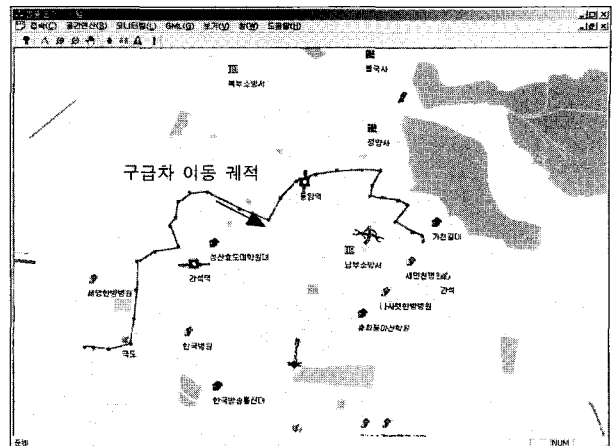
(질의 3)의 “현재 홍길동의 위치에서 가장 가까이 있는 구급차를 검색하라.”는 이동객체 질의에 대한 처리 결과는 다음과 같다.



(그림 6) MONEarst 연산 수행 결과

이 질의는 홍길동의 현재 위치로부터 구급차들간의 거리를 구하고, 이 중에서 최소값을 검색한다. 이러한 과정은 현재 홍길동의 위치에서 가장 가까운 구급차를 구하는 연산으로 (그림 6)은 질의 수행 결과 화면이다.

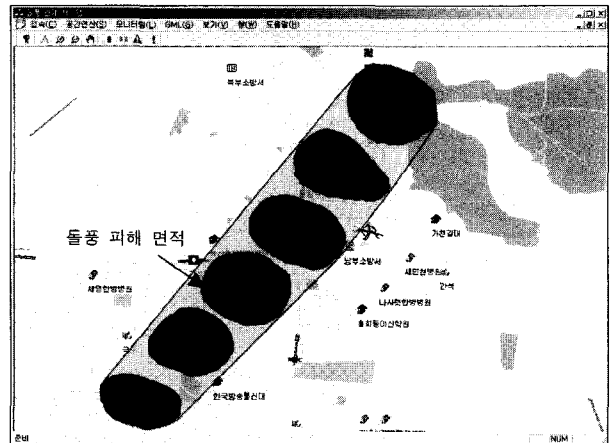
(질의 4)의 “2003년 4월 23일 12시부터 2003년 4월 23일 13시까지 유효시간동안 구급차 A가 이동한 궤적을 구하라.”는 이동객체 질의처리 결과는 다음과 같다.



(그림 7) MOTrajectory 연산 수행 결과

구급차 ‘A’가 질의시간 구간동안 이동한 튜플들을 구하고, 이를 시간 순서대로 정렬을 수행한 다음 ‘A’의 공간 속성을 프로젝션함으로써 질의시간 구간동안 구급차 ‘A’가 이동한 궤적을 구하게 된다. (그림 7)은 (질의 4)를 수행한 결과로, 구급차 ‘A’가 질의시간 구간 동안 이동한 궤적이 붉은 선으로 표시되고 있다.

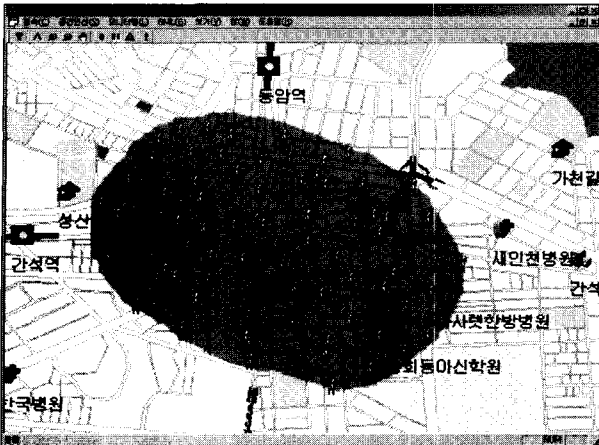
(질의 5)의 “현재까지 돌풍 T의 피해 면적을 검색하라.”는 이동객체 질의에 대한 처리 결과는 다음과 같다.



(그림 8) MOTraverse 연산 수행 결과

돌풍 ‘T’가 최초 폴링된 시점부터 현재까지 이동한 영역들을 검색하여 각 폴링된 수만개의 이동영역들을 얻는다. 이러한 이동영역들을 입력으로 볼록외곽(convex hull) 연산을 수행하면 이동영역들의 최외곽 볼록점으로만 구성되는 볼록외곽(convex hull) 객체가 생성된다. 이 볼록외곽(convex hull)의 면적을 구함으로써 실제 돌풍이 지나온 영역을 검색할 수 있고, (그림 8)에서 돌풍의 피해 면적을 붉은 색으로 나타내고 있다.

(질의 8)의 “현재 Tornado T의 영향권내에 속하는 사람을 검색하시오.”라는 이동객체 질의에 대한 처리 결과는 아래와 같다.



(그림 9) MOContains 연산 수행 결과

현재 돌풍 'T'의 영역과 사람들의 위치를 입력으로 contains 연산을 수행하면 'T' 영역내의 있는 사람들을 검색할 수 있고 이로부터 사용자의 이름을 검색하게 된다. (그림 9)는 질의 수행 결과로, 현재 돌풍지역은 붉은 영역으로 표시되고 이 영역안에 존재하는 사람들은 MOContains 연산을 통해 찾을 수 있다.

5.3 평가

지금까지 GML 문서에 대한 질의 처리를 위해 다양한 관련 연구들이 제시되었다. 기존의 GML 기반 질의어를 GML 저장소에 대한 직접적인 질의 처리를 위한 질의 변환이 필요 여부와 이동객체 질의 지원 여부 등은 <표 2>와 같다.

<표 2> GML 질의어 비교

질의어	질의 변환	이동객체 지원
Shekhar[13]	필요	무
Corcoles[14]	필요	무
Vatsavai[21]	불필요	무
제안 질의어	불필요	유

<표 2>에서처럼 Shekhar[13]은 URL 형식의 질의어 형태로 웹 매핑 시스템을 제안하고 있으며, 클라이언트측에서 점, 영역, 조인 질의 등을 수행한다. 이 방법에서는 GML 저장소에 대한 질의 처리를 위해 URL 형식의 질의어를 GML 저장소에 대한 표준 질의어로 변환을 해야하며, 이동객체에 대한 질의를 지원하지 않는 단점이 있다. Corcoles[14]에서는 SQL 구문을 확장하여 GML 문서에 대한 질의어 명세를 제안하고 있다. 이 방법은 SQL을 기반으로 질의어를 확장함으로써 사용자에게 보다 친숙하다는 장점이 있으나, GML 저장소에 대한 질의 처리를 위해 XML 표준 질의어로 질의 변환이 필요하고 또한 이동객체에 대한 질의 명세는 포함하고 있지 않다. Vatsavai[21]에서는 GML 문서에 대한 공간 질의를 지원하기 위해 기존의 XML 질의어를 확장하였다. 이 방법은 기존의 XML 질의어를 기반을 두고

있어 GML 저장소에 대한 공간 질의가 가능한 장점이 있으나, 이동객체를 위한 질의 명세는 포함하고 있지 않다.

이 논문에서 제시하고 있는 GML 기반의 이동객체 질의어는 기존 질의어들의 문제점을 해결하기 위하여 GML 저장소에 대한 공간 및 비공간 정보에 대한 직접적인 질의가 가능하도록 표준 XML 질의어인 XQuery를 확장하였으며, 또한 이동점과 이동영역으로 분류되는 이동객체 데이터에 대한 다양한 질의를 지원함으로써 위치기반서비스에 보다 폭넓게 활용될 수 있도록 하였다.

6. 결론

이동객체란 위치기반서비스에서 시간의 흐름에 따라 그 위치와 모양이 연속적으로 변화하는 차량, 휴대용 전화기, 노트북, PDA 등을 의미하며, 위치기반서비스에서는 이러한 이동객체의 정보를 효과적으로 관리하는 것이 필요하다. 또한 OGC에서는 기존의 지리정보 뿐 아니라 이동객체의 정보의 저장 및 전송을 위한 목적으로 XML 기반 인코딩 표준인 GML을 제안하였다. GML은 차세대 웹 문서의 표준인 XML을 기반으로 작성되어, 향후 이동객체 정보의 상호운용성을 제공하는 방법으로 그 활용도가 높을 것으로 예상되며, 이에 관한 연구가 널리 진행되고 있다. 특히, 수행중인 GML 연구들 중에서 질의 표현은 GML 문서에 대한 효과적인 검색 방법을 제공하기 위한 필수로 인식되고 있다.

그러나, 기존의 GML 질의어에 대한 연구는 질의 표현에 많은 제약이 따르는 URL(Uniform Resource Locator)을 이용하는 방법, GML 문서에 대한 직접적인 질의가 불가능하여 GML에 대한 질의로 변환을 해야하는 문제가 있는 SQL 구문을 확장하여 GML에 대한 공간 질의를 수행하는 방법, 그리고 데이터 모델과 대수에 대한 언급없이 OGC에서 명세한 공간 함수를 기존의 XML 질의어를 확장하여 GML 문서에 대한 공간 질의가 가능하게 하는 방법 등으로, GML 문서에 표현된 이동객체 정보를 변환없이 직접 적용할 수 있는 이동객체 질의어에 관한 연구는 아직까지 수행되지 않고 있다.

따라서 본 논문에서는 위치기반서비스에서 필수적인 이동객체 정보를 포함하는 GML 문서를 대상으로 이동객체 질의를 처리하기 위해 XQuery를 확장한 이동객체 질의어 명세를 제안하였다. 이 논문에서는 제안 이동객체 질의어를 위해 먼저 이동객체에 대한 정점, 간선, 순서 속성에 대한 정의를 통해 데이터 모델을 제안 및 제안 데이터 모델을 적용한 예를 보였고, 이동객체를 기반으로 다양한 연산자의 명세 및 이동객체 대수 연산에 대해 설명하였다. 제안 이동객체 질의어에 대한 설명은 제안한 데이터 모델 및 대수 연산을 기반으로 다양한 이동객체 연산을 이용하여 확장된 질의의 예를 기술하였다. 아울러 실험으로는 제안 이동객체 질의어를 돌풍 관제 시스템에 적용시켰으며, 여기서는 이동객체 질의어를 공간 데이터베이스의 SQL로 변환하여 처리한 결과를 보였다. 제안 질의어는 XQuery를 기반으로 작성

되어 기존의 XML에 대한 질의를 지원할 뿐 아니라 GML 문서의 이동객체 정보에 대한 효과적인 질의가 가능하게 함으로써, 응급재난, 위치기반상거래, 텔레메틱스 등 다양한 위치기반 응용 서비스에 널리 적용될 수 있다.

향후 연구로는 유효시간 구간동안 취득된 이동객체의 정보로부터 특정 시점에 대한 이동객체의 정보를 검색하기 위한 연산자들에 대한 연구가 필요하다. 그리고 제안한 이동객체 질의어를 실제 응용에서 적용할 수 있는 다양한 평가와 함께 기존의 SQL을 지원하는 시스템의 재활용을 위해 제안 이동객체 질의어를 SQL로 변환하는 방법에 대한 연구가 수행되어야 한다.

참 고 문 헌

- [1] T. Bray and et al., "Extensible Markup Language (XML) 1.0 (Second Edition)," W3C Recommendation, <http://www.w3.org/TR/REC-xml>, 2000.
- [2] M. Arikawa and K. Kubota, "A Standard XML Based Protocol for Spatial Data Exchange-Its Capabilities and Real Applications," International Workshop on Emerging Technologies for geo-based applications, May, pp.37-45, 2000.
- [3] OGC, "Geography Markup Language (GML) Implementation Specification v1.0," Document Number : 00-029, <http://www.opengis.net/gml/00-029/GML.html>, 2000.
- [4] OGC, "Geography Markup Language (GML) Implementation Specification v2.1.1," Document Number : 02-009, <http://www.opengis.net/gml/02-009/GML2-11.html>, 2002.
- [5] OGC, "Geography Markup Language (GML) Implementation Specification v3.0," Reference number : OGC 02023r4, 2003.
- [6] D. Beech and et al., "A Formal Data Model and Algebra for XML," <http://www-db.stanford.edu/dbseminar/Archive/FallY99/malhotraslides/malhotra.pdf>, 1999.
- [7] K. H. Ryu, Y. A. Ahn, J. W. Lee and Y. J. Lee, "Moving Object Database and Their Applying to Location-Based Services," SIGDB Journal, Vol.17, No.3, pp.57-74, 2001.
- [8] S. Abiteboul and et al., "The Lorel Query Language for Semistructured Data," International Journal on Digital Libraries, 1, 1, pp.68-88, 1997.
- [9] A. Deutsch and et al., "XML-QL : A Query Language for XML," Technical Report NOTE-xml-ql-19980819, <http://www.w3.org/TR/1998/NOTE-xml-ql-19980819.html>, 1998.
- [10] J. Robie, "The design of XQL," <http://www.w3.org/style/XSL/Group/1998/09/XQL-design.html>, 1998.
- [11] S. Boag and et al., "XQuery 1.0 : An XML query language," W3C Working Draft, <http://www.w3c.org/TR/xquery>, 2003.
- [12] OGC, "Web Map Server Interface Specification," Revision 1.0, 2000.
- [13] S. Shekhar and et al., "WMS and GML based Interoperable Web Mapping System," In Proceedings of the Ninth ACM International Symposium on Advances in Geographic Information Systems, ACMGIS. ACM Press, 2001.
- [14] J. E. Corcoles and P. Gonzalez, "A Specification of a Spatial Query Language over GML," In Proceedings of the Ninth ACM International Symposium on Advances in Geographic Information Systems, ACMGIS, ACM Press, 2001.
- [15] Jim Melon, "XML-Related Specifications(sql/xml)," ftp://sqlstandards.org/SC32/WG3/Progression_Documents/Informal_working_drafts/wd-xml-2001-06.pdf, 2001.
- [16] M. Erwig and et al., "Spatio-Temporal Data Types : An Approach to Modeling and Querying Moving Object in Databases," GeoInformatica, Vol.3, No.3, pp.269-296, 1999.
- [17] R. H. Gutting and et al., "A Foundation for Representing and Querying Moving Objects," ACM Transactions on Database Systems, Vol.25, No.1, pp.1-42, 2000.
- [18] M. Fernandez and et al., "XQuery 1.0 and Xpath 2.0 Data Model," W3C Working Draft, <http://www.w3.org/TR/query-datamodel>, 2003.
- [19] D. Draper and et al., "XQuery 1.0 and Xpath 2.0 Formal Semantics," W3C Working Draft, <http://www.w3.org/TR/query-algebra>, 2003.
- [20] OGC, "Web Map Server Draft Candidate Implementation Specification," Version 1.0.7, 2001.
- [21] R. R. Vatsavai, "GML-QL : A Spatial Query Language Specification for GML," <http://www.cobblestoneconcepts.com/ucgis2summer2002/vatsavai/vatsavai.htm>, 2002.
- [22] OGC, "OpenGIS simple features specification for sql," <http://www.opengis.org/techno/specs/99-049.pdf>, 1999.

[부 록 1] 이동객체 스키마

```

MovingObject.xsd
<?xml version = "1.0" encoding = "UTF-8"?>
<schema targetNamespace = "http://.../app"
  xmlns : mo = "http://app"
  xmlns : gml = "http://www.opengis.net/gml"
  xmlns = "http://www.w3.org/2000/10/XMLSchema"
  elementFormDefault = "qualified" version = "3.00">
  <annotation>
    <appinfo> Moving Object Schema </appinfo>
    <documentation xml : lang = "en">
      This Schema provides constructs for handling moving objects.
    </documentation>
  </annotation>
  <!------->
  <include schemaLocation = "gml.xsd"/>
  <include schemaLocation = "feature.xsd" />
  <!------->
  <element name = "MovingObject" type = "mo :
    MovingObjectType" />
  <!------->
  <complexType name = "MovingObjectType">
    <choice>
      <element ref = "mo : MOPointType" minOccurs = "1"
        maxOccurs = "unbounded"/>
      <element ref = "mo : MORegionType" minOccurs = "1"
        maxOccurs = "unbounded"/>
      <element ref = "mo : MOMultiPointType" minOccurs = "1"
        maxOccurs = "unbounded"/>
    </choice>
  </complexType>

```

```

    <element ref = "mo : MOMultiRegionType" minOccurs = "1"
      maxOccurs = "unbounded"/>
  </choice>
</complexType>
<!------->
<element name = "MOPoint" type = "mo : MOPointType" />
<!------->
<complexType name = "MOPointType">
  <attribute name = "name" type = "string" use = "required"/>
  <element ref = "mo : MOPointInstanceType" minOccurs = "1"
    maxOccurs = "unbounded"/>
</complexType>
<!------->
<element name = "MOPointInstance"
  type = "mo : MOPointInstanceType" />
<!------->
<complexType name = "MOPointInstanceType">
  <sequence>
    <element ref = "gml : Point" minOccurs = "1"
      maxOccurs = "1" />
    <element ref = "gml : timeStamp" minOccurs = "1"
      maxOccurs = "1" />
  </sequence>
</complexType>
<!------->
<element name = "MORegion" type = "mo : MORegionType" />
<!------->
<complexType name = "MORegionType">
  <attribute name = "name" type = "string" use = "required"/>
  <element ref = "mo : MORegionInstanceType" minOccurs = "1"
    maxOccurs = "unbounded"/>
</complexType>
<!------->
<element name = "MORegionInstance"
  type = "mo : MORegionInstanceType" />
<!------->
<complexType name = "MORegionInstanceType">
  <sequence>
    <element ref = "gml : Polygon" minOccurs = "1"
      maxOccurs = "1" />
    <element ref = "gml : timeStamp" minOccurs = "1"
      maxOccurs = "1" />
  </sequence>
</complexType>
<!------->
<element name = "MOMultiPoint"
  type = "mo : MOMultiPointType" />
<!------->
<complexType name = "MOMultiPointType">
  <attribute name = "name" type = "string" use = "required"/>
  <element ref = "mo : MOMultiPointInstanceType"
    minOccurs = "1" maxOccurs = "unbounded"/>
</complexType>
<!------->
<element name = "MOMultiPointInstance"
  type = "mo : MOMultiPointInstanceType" />
<!------->
<complexType name = "MOMultiPointInstanceType">
  <sequence>
    <element ref = "gml : MultiPoint" minOccurs = "1"
      maxOccurs = "1" />
  </sequence>

```

```

    <element ref = "gml : timeStamp" minOccurs = "1"
      maxOccurs = "1" />
  </sequence>
</complexType>
<!------->
<element name = "MOMultiRegion"
  type = "mo : MOMultiRegionType" />
<!------->
<complexType name = "MOMultiRegionType">
  <attribute name = "name" type = "string" use = "required"/>
  <element ref = "mo : MOMultiPolygonInstanceType"
    minOccurs = "1" maxOccurs = "unbounded"/>
</complexType>
<!------->
<element name = "MOMultiRegionInstance"
  type = "mo : MOMultiRegionInstanceType" />
<!------->
<complexType name = "MOMultiRegionInstanceType">
  <sequence>
    <element ref = "gml : MultiPolygon" minOccurs = "1"
      maxOccurs = "1" />
    <element ref = "gml : timeStamp" minOccurs = "1"
      maxOccurs = "1" />
  </sequence>
</complexType>
<!------->
</schema>

```



정 원 일

e-mail : wncung@dblab.inha.ac.kr
 1998년 인하대학교 전자계산공학과(공학사)
 1998년~현재 인하대학교 대학원
 전자계산공학과 박사과정
 관심분야 : 공간데이터베이스 클러스터,
 이동체 데이터베이스, GML,
 LBS 등



배 해 영

e-mail : hybae@inha.ac.kr
 1974년 인하대학교 응용물리학과(공학사)
 1978년 연세대학교 대학원 전자계산학과
 (공학석사)
 1989년 숭실대학교 대학원 전자계산학과
 (공학박사)
 1985년 Univ. of Houston 객원 교수
 1992년~1994년 인하대학교 전자계산소 소장
 1982년~현재 인하대학교 전자계산공학과 교수
 1999년~현재 지능형 GIS 연구센터 소장
 2000년~현재 중국 중경우전대학교 대학원 명예 교수
 관심분야 : 공간데이터베이스, 지리정보시스템, 분산데이터베이스,
 데이터베이스 클러스터 멀티미디어 데이터베이스 등