

GML기반 개방형 이동체 프리젠테이션 시스템 구현

김 동 호^{*} · 김 진 석^{**}

요 약

인터넷 기반의 광역화 물류환경인 e-Logistics에서 요구하는 가시성을 충족하기 위해서는 지역별 전자지도 및 차량 위치정보에 대한 공유와 통합이 필수적이다. 기존의 대부분의 차량관제시스템들은 독자적인 체계를 중심으로 구축되었기 때문에 정보의 공유와 통합을 위해서는 막대한 비용이 필요하다. 이를 효과적으로 해결하기 위해서 본 논문에서는 XML 엔코딩 표준인 GML을 이용하여 이동체 위치정보와 지도정보의 프리젠테이션을 위한 컴포넌트 기반의 개방형 시스템을 설계하고 구현하였다. 아울러 e-Logistics 환경의 배송 시나리오를 통한 실험을 통해 확장성 및 상호 운용성에 대한 우수성을 확인하였다.

Implementation of Open Moving Object Presentation System Based on GML

Dong-Ho Kim^{*} · Jin-Suk Kim^{**}

ABSTRACT

The integration and union of regional digital map and vehicle location data in e-Logistics which means virtual service architecture based on the Internet among the logistics companies is indispensable clause in order to satisfy the visibility characteristics. Most conventional vehicle monitoring and controlling systems were constructed with individual and mutual exclusive architecture so that there are required lots of cost for sharing and integrating the information among them. In this paper, we design and implement an component-based open system which represent moving object location information as well as digital map using the GML(Geography Markup Language - international standard encoding based on XML). We also test proposed system on the delivery vehicle scenario in e-Logistics environment, and validate its superior in terms of extensibility and interoperability.

키워드 : 이동체(Moving Object), GML, XML, e-Logistics, 컴포넌트 기반(Component-based), 차량관제(Vehicle Monitoring and Control)

1. 서 론

인터넷의 활성화로 인해 물류의 대상 지역이 무제한으로 광역화되었으며, 정보처리 기술을 기반으로 한 높은 수준의 물류체계 유연성 및 효율성을 요구하게 되었다. 뿐만 아니라, 물류에서의 전자상거래 도입은 B2B 다자간 물류정보의 실시간 통합 환경과 고객 서비스의 지식정보화를 요구하게 되었다.

이러한 인터넷을 기반으로 하여 새로이 형성되고 있는 가상의 물류 기업 활동 및 서비스 체계를 e-Logistics로 정의할 수 있으며, 이를 위한 프레임워크는 다양한 유형의 물류정보시스템 간의 B2B 통합, 물류 정보 및 차량의 흐름에 대한 실시간 모니터링과 예외상황 발생에 따른 지능화된 경고/조치를 위한 가시화(Visibility), 최적화된 물류계획 수립 및 물류계획의 동적/지능적 재조정을 위한 물류 최적화 시스템

등으로 구성된다. e-Logistics 지능화 시스템은 e-Logistics 환경에서 획득/유통/관리되는 물류정보를 활용하여 효과적이고 효율적인 e-Logistics 체계의 설계, 운용계획, 실행통제 활동을 실현하기 위한 지식 집약적인 정보기술로서, 세부적으로는 e-Logistics 체계의 설계 및 운용계획 지능화를 위한 최적화 기술(Optimization)과 e-Logistics 환경에서 실물의 가시성 확보를 바탕으로 지능적이고 민첩한 실행통제 활동을 실현하기 위한 이동체(Moving Object) 기술로 구성된다[7].

이동체 기술은 물류 정보시스템에서 개별적인 물류 사업자가 보유하고 있는 운송 차량에 대한 실시간 정보를 보다 효과적으로 관리할 수 있도록 해준다. 물류 사업자의 운송 차량은 대표적인 이동체로서 시간과 공간적인 요소들을 동시에 보유하고 있으며, 위치와 형태등과 같은 공간적인 특성들은 시간에 따라서 변화하는 특성을 가진다. 이동체 기술은 기존의 차량관제 시스템들이 가진 기술적인 한계점을 해결하기 위하여 다양한 유형의 차량 위치데이터에 대한 통합, 대용량 이동체 위치정보 관리, 물류정보지식관리, 표준화 이

^{*} 종신회원 : ETRI 우정기술연구센터 선임연구원

^{**} 정회원 : ETRI 우정기술연구센터 책임연구원

논문접수 : 2004년 7월 19일, 심사완료 : 2004년 8월 12일

동체 정보 공유 및 프리젠테이션 등의 핵심 기능으로 구성된 이동체 관리 시스템으로 구체화 된다.

이동체 관리 시스템의 대표적인 적용 분야인 차량 관제 시스템에 소요되는 전자지도의 구축은 고비용이므로 전자지도의 공유 및 활용은 현재 주요 관심사가 되고 있으며, 초기부터 주행안내, 교통정보, 도로망 정보(중앙선 정보 및 교차점 정보 포함)를 참조할 수 있는 스키마로 설계하여 구축하는 경우도 있으나 비용적인 문제로 인하여 현실화의 어려움을 겪고 있다. 이러한 문제의 해결책으로 기 구축된 전자지도를 다른 데이터와 연계하는 방안을 사용하는 경우가 많다. 하지만 기존의 기법에서는 차량 위치에 관련된 고급정보의 공유성 및 상호표현성이 미비하다. 이러한 이유로 기존의 차량관제시스템들이 공통적으로 갖는 한계점인 서비스의 광역화와 다형성의 제한성을 내포하고 있다. 즉, 지역별 차량관제시스템은 타 차량관제시스템의 정보를 활용하기 어려우며, 이를 표현하기 위해서는 EDI, EAI등과 같은 기술을 기반으로 많은 노력을 지속적으로 들여야 하는 문제점을 가진다.

이를 해결하기 위해서는 표준명세 기반의 개방형 인터페이스로 차량 위치정보의 상호 공유성 및 표현성을 증대하는 방안이 필요하다. 즉, 표준에 기반을 둔 개방형 이동체 프리젠테이션 기술을 개발함으로써 비용과 성능의 관점에서 물류 업무의 특성을 만족하는 보다 효율적인 차량관제가 되도록 해야 한다.

또한 컴포넌트 기반의 구축기법을 적용하여 개발된 기능을 타 시스템에서 활용이 용이하게 시스템 설계 및 구현이 필요하다. 따라서 본 논문에서는 GML 표준 명세를 지원하는 개방형 이동체 프리젠테이션 시스템을 컴포넌트 개발 기법을 적용한 시스템을 제안한다.

보다 효과적인 설명을 위해 세부적으로 다음과 같이 논문을 구성하였다. 먼저 2장에서는 본 연구와 관련되어 이동체 기술과 GML 인터페이스의 이전 연구들을 정리한다. 3장에서 5장까지는 컴포넌트 기반의 개방형 이동체 프리젠테이션 인터페이스 모델, 설계, 구현 및 실험에 관련 사항을 설명한다. 6장에서는 제안한 시스템의 요약과 향후 연구방향을 기술한다.

2. 관련 연구

2.1 이동체 관리 기술

이동체는 시간에 따라 객체의 공간 정보가 연속적으로 변경되는 시공간 데이터로 이동점과 이동영역으로 구분된다. 이동점(Moving point)은 시간에 따라 객체의 위치가 변하는 것으로 이동 점 객체에는 사람, 동물, 자동차, 비행기, 배 등이 있다. 이동영역(Moving region)은 시간에 따라 객체의 위치뿐만 아니라 모양까지 변하는 것으로 한 국가의 행정 구역이나 폭풍의 영향권, 압세포의 상태 등을 예로 들 수 있다. 이와 같은 이동체를 저장 및 관리하는 이동체 데이터베이스는 위치 기반 서비스, 차량 위치 추적, 물류 관리 등을 위한

기본 소프트웨어 인프라로 인식되고 있다[3].

이동체 관리를 위한 대표적인 응용 시스템 연구에는 DOMINO, CHOROCHRONOS, DEDALE, Battlefield Analysis가 있다. DOMINO[9]는 이동 객체의 수송에 대한 웹 기반 실시간 궤도 응용의 개발을 촉진하는 이동 객체 소프트웨어 도구로서, DBMS 기술을 활용하여 만든 실시간 위치 추적 시스템 프로토타입이다. 그러나 DOMINO 프로토타입은 이동 객체의 현재 위치, 속도, 방향정보를 이용하여 미래의 이동 위치를 예측하는 방법에 주로 초점을 맞추고 있다. CHOROCHRONOS는 시공간 데이터베이스의 특수한 형태인 이동 객체에 관한 연구가 집중적으로 수행되었고, 이동 객체의 데이터 모델링 및 인덱싱에 관한 연구 결과를 가장 많이 발표하였다. 특히 이 연구는 GPS 기반의 수송 관리 시스템과 멀티미디어 시스템에 적용한 응용 시나리오를 제시하였다. 하지만 이들 연구들은 이동체 정보의 상호 운용성과 호환성을 고려하지 않았기 때문에 이에 대한 취약한 문제점이 있다.

2.2 GML/SVG

과거에는 포맷의 표준화 필요성으로 인하여 많은 연구가 진행되었으나 각 포맷들이 지니고 있는 변환비용, 표준 포맷간의 비통일성 등의 문제점으로 인하여 그 활용범위를 넓히지 못하였다. 1990년대 후반 이후 데이터 공유 및 교환을 위하여 XML에 관한 관심이 증가함에 따라 XML 기반의 지리 정보 인코딩 표준화 방안으로 GML(Geography Markup Language)[8], G-XML[4]에 관한 연구가 시도되었다. 이후 ISO, OGC(OpenGIS Consortium) 등의 국제기구 중심으로 상업화와 개방형 환경으로의 전환을 위하여 지속적인 연구를 진행하고 있다. 최근 OpenGIS Consortium는 GIS 데이터 전송의 표준으로 XML(eXtensible Markup Language)을 발전시킨 GML(Geography Markup Language) 3.0 Spec. 을 발표하였다. 또한 2001년 9월 4일 W3C에서 SVG(Scalable Vector Graphics)[10]가 발표되었다. SVG는 2차원의 벡터 그래픽을 위해 만들어 졌는데 정량적인 통계, 그래프, 지도 그리고 기술적인 다이어그램을 포함한 정보의 다양한 형식의 표현을 위해 유용하다. 그러나 표준 명세에 기반을 둔 지리정보에 대한 부가 연산의 통합 연계가 미흡하여 개별 응용분야에서 실용화를 위한 연산 및 시스템을 구축해야 한다. [14]에서는 공간정보에 대한 XML명세인 XQuery를 확장한 이동체 모델과 질의어 정의를 통하여 클라이언트와 서버간에 XML문서를 통한 정보 교환을 제시하였다.

2.3 CBD

컴포넌트기반의 개발(Component Based Development : CBD)은 재사용 가능한 소프트웨어 모듈 컴포넌트를 생성 및 조립 생산, 선택, 평가 및 통합으로 구성하여 더 큰 컴포넌트를 생성하거나 완성된 어플리케이션 소프트웨어를 구축하는 개발 기법이다. 여기서 컴포넌트는 또 다른 컴포넌트를 만들거나 어플리케이션을 만들기 위한 소프트웨어 블록

이며 독립적으로 개발 및 배포가 가능한 소프트웨어 패키지를 의미한다. 소프트웨어 컴포넌트를 조립해 새로운 어플리케이션을 만들 수가 있으므로 개발기간을 단축할 수 있고, 기존의 컴포넌트를 재사용함으로써 생산성과 경제성을 높일 수 있다. 비즈니스 로직이 틀릴 경우에는 컴포넌트를 재조합하거나 새로운 컴포넌트를 끼워 넣기만 하면 된다. 최근 몇 년간 몇몇 선도적인 조직과 전문가들에 의해 컴포넌트를 기반으로 한 주목할 만한 소프트웨어 개발 접근법들이 소개되었다. 현재까지 소개된 다양한 방법론들 중 최근 산업계나 학계에서 많이 사용되고 있는 CBD 방법론들은 다음과 같이 정리할 수 있다.

RUP[6]는 생명주기 전체를 지원하는 소프트웨어 개발을 위한 프로세스 프레임워크로, Jacobson의 OOSE 방법론의 확장이라고 볼 수 있으며 컴포넌트 설계에 있어 UML 표기법에 상당한 비중을 두고 있다. 반복적인 개발 방법을 제안하고 있으며, 각각의 반복은 요구사항 분석, 분석·설계, 구현·테스트 및 평가 과정을 포함하고 있어 자체로서도 하나의 개발 주기를 이룬다. Catalysis[2]는 주로 학계에서 연구대상으로 채택되는 방법론으로, 이 방법론이 포함하고 있는 다양한 기초 개념과 기법은 기타 다른 방법론들의 형성에 영향을 주었다. Object와 Action을 핵심개념으로 사용하고 있으며, CBD의 바이블로 통하지만 실제화하기 위해서는 프로세스의 정리가 필요하다. 마르미-III[12]는 한국전자통신연구원에서 만든 컴포넌트 기반 소프트웨어 개발 방법론이다. 마르미-III는 컴포넌트의 개발 및 컴포넌트 기반의 시스템 개발에 필요한 작업과 작업 수행에 필요한 기법 및 작업별 산출물을 정의하고, 작업에 따른 상세한 개발 절차와 지침을 제공한다. UML Components 방법론은 소프트웨어 컴포넌트의 핵심 개념 중에 하나인 컴포넌트 명세 작성을 위한 가장 간단하고 실용적인 프로세스를 제공해주고 있다. 이 방법론은 앞서 언급한 Catalysis와 Advisor 방법론에 큰 영향을 받았으며, RUP를 부분적으로 수용하고 있다. UML Components는 컴포넌트를 위한 많은 개념들과 공정 및 지원 활동들이 생략되어 있지만, CBD 도입에 부담을 갖기 쉬운 초보자라도 쉽게 습득할 수 있다는 장점을 가지고 있다.

3. 시스템 모델

이 장에서는 차량 위치 추적을 위한 이동체 관리 시스템에서 이동 차량의 위치에 대한 데이터 표현과 연산자를 포함하는 데이터베이스 모델을 정리한다[1].

3.1 이동체 데이터베이스

이동체는 형태에 따라서 이동 점과 이동 영역으로 세분화된다. 이것은 이동체에 대한 데이터 표현 복잡도를 의미하는데, 이동 차량의 경우는 이동 점에 해당하며 다음과 같이 데이터 형태를 정의한다.

[정의 1] (이동체) 시간의 변화에 따라 객체의 위치 값만 변화되는 이동체를 말한다. 이동체 MP 는 시간 속성, 공간 속성, 일반 속성을 가지며, $MP = \langle T_A, S_A, G_A \rangle$ 가 된다. □

[정의 2] (시간 속성) MP 의 시간 속성 $T_A = \langle vt_s, vt_e \rangle$ 로 구성되며 vt_s 는 시작 시간, vt_e 는 종료 시간을 나타낸다. 이때 vt_s 와 vt_e 는 유효시간(Valid time)의 집합 S_{VT} 의 원소가 된다. 유효시간은 실세계에서 발생된 시간을 나타내며, $S_{VT} = \{t_0, t_1, t_2, \dots, t_k, \dots, t_{now}\}$ 이고, 각 원소들은 $t_0 < t_1 < t_2 < \dots < t_k < \dots < t_{now}$ 의 순서를 가진다. $t_k = t_{k-1} + 1$, $t_k = t_0 + k$, $k \geq 0$ 인 정수로 정의된다. t_{now} 는 현재 시간을 의미하는 시간 상수이다. □

[정의 3] (공간 속성) MP 의 공간 속성 $S_A = \langle x, y \rangle$, $x, y \in R$, R 은 실수이다. □

[정의 4] (이동체 데이터베이스) 이동체 데이터베이스를 구성하는 이동체들의 집합은 $S_{MP} = \{MP_0, MP_1, \dots, MP_n\}$ 이다. S_{MP} 로 구성된 이동체 데이터베이스의 이력 집합은 $H_{MP} = \{H_{MP_0}, H_{MP_1}, \dots, H_{MP_n}\}$ 이다. S_{MP} 에 속하는 각각의 MP_i 에 대한 모든 이력 집합은 $H_{MP_i} = \{mp_{i_0}, mp_{i_1}, \dots, mp_{i_k}\}$ 이고, $mp_{i_k} = \langle T_A(mp_{i_k}), S_A(mp_{i_k}), G_A(mp_{i_k}) \rangle$ 이다. □

[정의 2]에서 유효시간의 도메인은 선형 시간(Linear time), 이산 시간(Discrete time), 절대 시간(Absolute time)이며, 하나의 이동체 데이터베이스는 동일한 유효시간의 주기(Granularity)를 가진다. 그리고 [정의 4]에서 mp_{i_k} 는 MP_i 의 k 번째 이력 정보를 의미하고, $T_A(mp_{i_k})$ 는 MP_i 의 k 번째 시간 속성, $S_A(mp_{i_k})$ 는 k 번째 공간 속성, $G_A(mp_{i_k})$ 는 k 번째 일반 속성을 의미한다.

〈표 1〉 이동체 연산자

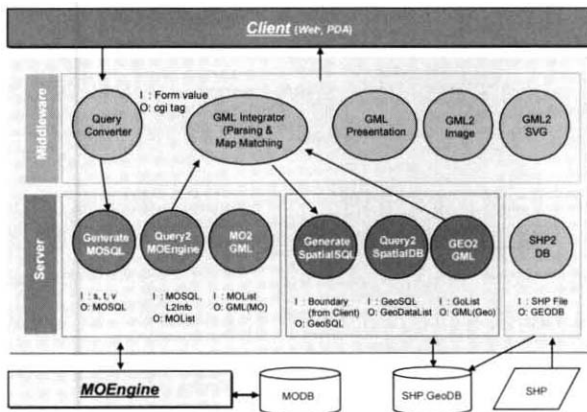
종류	입력 값	출력 값	기능
m_distance	$G_A \times G_A \times T_A$	Real	특정 유효시간 동안 두 이동 객체간의 거리 계산
trajectory	$G_A \times T_A$	$\{S_A\}$	이동 객체의 특정 유효시간 동안의 이동 경로 추출
length	$\{S_A\}$	Real	이동 객체의 특정 유효시간 동안의 이동 거리 계산
position_at	$G_A \times T_A$	S_A	임의의 한 시점에 대한 이동 객체의 위치 검색
m_nearest	$G_A \times T_A$	S_A	특정 유효시간 동안 가장 가까이 위치하는 객체 검색
m_farthest	$G_A \times T_A$	S_A	특정 유효시간 동안 가장 멀리 존재하는 객체 검색

이동체의 위치정보 관련 연산자에는 〈표 1〉에서 보여진 바와 같이 $m_distance$, $trajectory$, $length$, $position_at$, $m_nearest$, $m_farthest$ 가 있다.

nearest, m_farthest 등이 있다. 입력 및 출력 값에서 G_A 는 차량 소유주, 운전자, 차종 등과 같은 이동 객체의 일반 속성을 의미한다. S_A 는 공간 속성을 나타내고, T_A 는 시간 속성인 유효시간을 의미한다. *Real*은 실수 값을 나타내며, {}로 묶인 속성들은 집합을 의미한다.

3.2 GML 인터페이스

모바일 단말기에서 전송되는 다양한 사용자 입력 질의들에 대해 MODB나 SpatialDB를 이용해 이를 처리하여 디지털 지도상에 GML, 이미지, SVG 형태로 이동체 위치정보를 서비스하며, 차량의 위치 관련 정보 및 공간 데이터를 GML 형식으로 변환하는 변환기 기능과 GML 파일을 화면에 출력하는 기능, 이동체 엔진에서 전달된 GML 형태의 차량 정보와 지도 정보를 통합하고 디지털 지도상의 정확한 위치에 차량을 출력하기 위한 맵 매칭 기능을 수행한다. 이동체 위치정보 Web/Mobile 인터페이스 시스템은 (그림 1)에서 보인바와 같은 Client Layer와 MiddleWare Layer, Server Layer로 구성된다.



(그림 1) GML 인터페이스 구성도

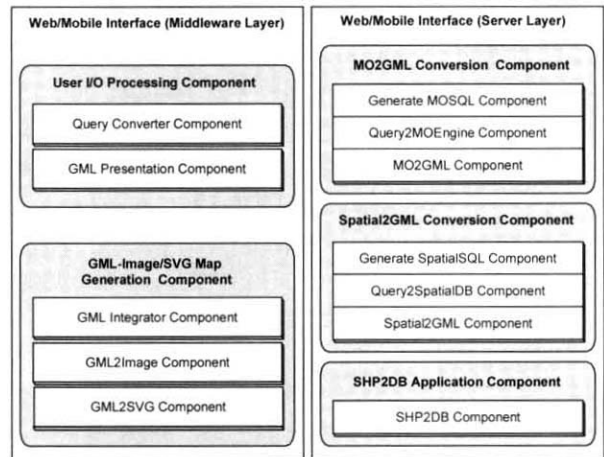
Client Layer 부분은 사용자에게 입출력 기능을 제공하는 Web/Mobile 사용자 인터페이스로서 사용자가 이용하는 각종 장비로부터 다양한 질의를 입력 받을 수 있고, 해당 질의 결과를 GML 이나 이미지, SVG 형태로 디스플레이하는 사용자 브라우저이다. MiddleWare Layer 부분은 사용자 브라우저로부터 입력된 질의를 CGI Tag 형태로 변환하여 Server Layer에 전달하고, Server Layer로부터 전송된 GML 형태의 질의 결과를 통합한 후 맵 매칭 기능을 수행하여 사용자 브라우저에 GML 형태로 디스플레이하거나 혹은 이미지나 SVG 형태로 변환하여 디스플레이하는 컴포넌트 어플리케이션이다. Server Layer 부분은 CGI Tag 형태로 변환된 질의를 이용하여 이동체 데이터베이스를 검색한 후 해당 이동체 정보를 GML 형태로 변환하고, 이동체 위치에 해당하는 지도 영역 데이터를 공간 데이터베이스를 검색하여 GML 형태로 변환하는 컴포넌트 어플리케이션이다. 또한, 이외에

도 다양한 지도 정보를 가진 SHAPE 파일을 읽어 공간 데이터베이스에 저장하는 컴포넌트를 포함한다.

4. 설 계

4.1 구 조

개방형 이동체 프리젠테이션 인터페이스 컴포넌트는 (그림 2)에서 보인 바와 같이 두 계층으로 구성된다. 즉, MiddleWare Layer와 Server Layer에 포함된 각각의 기능성 모듈들을 독립적으로 개발되고 배포되는 단위의 소프트웨어로 정의한다. 이러한 컴포넌트화는 이동성 관리 응용 도구의 확장 및 유지 보수, 새로운 응용의 개발과 같은 영역에서 각 기능성 모듈들을 보다 쉽게 재사용함으로써 효율적인 응용 개발 및 유지 보수성을 제공할 수 있다.



(그림 2) 이동체 위치정보 컴포넌트 구조도

개방형 이동체 프리젠테이션 인터페이스에서 MiddleWare Layer와 Server Layer를 구성하는 각 컴포넌트들의 세부 기능은 다음과 같다.

사용자 I/O 처리 컴포넌트 그룹은 Query 생성 Component와 GML Presentation Component로 구성된다. Query Converter Component는 사용자 입력 정보를 분석하고 입력 정보를 Server Layer에 전달하기 위해 CGI Tag 형태의 질의 데이터를 생성하며, GML Presentation Component는 GML에 대한 JDOM Parsing을 통해 이동체와 이동체의 위치가 포함된 전자 지도를 생성한다.

GML-Image/SVG 지도 생성 컴포넌트 그룹은 GML Integrator Component, GML2Image Component, GML2SVG Component 등 세 가지로 구성된다. GML Integrator Component는 이동체에 대한 지도 영역을 매칭시키고 MO GML 과 Spatial GML을 통합하여 하나의 GML 파일을 생성하며, GML2Image Component는 GML에 대한 JDOM Parsing을 통해 GML의 각 속성 데이터를 추출하여 GML에 대응하는 이미지를 생성한다. 그리고 GML2SVG Component는 GML

에 대한 JDOM Parsing을 통해 GML을 SVG로 변환한다.

MO2GML 변환 컴포넌트 그룹은 Generate MOSQL Component, Query2MOEngine Component, MO2GML Component 으로 구성된다. Generate MOSQL Component는 MiddleWare Layer로부터 전달 받은 질의 데이터를 MOEngine에 전송하기 위해 패킷 형태로 변환하거나 MODB를 검색하기 위한 SQL문을 생성하며, Query2MOEngine Component는 MOEngine을 통해 패킷 형태의 질의문이나 SQL문을 이용하여 MODB를 검색한다. 그리고 MO2GML Component는 MODB로부터 검색된 이동체 정보를 이용하여 GML 파일을 생성한다.

Spatial2GML 변환 컴포넌트 그룹은 Generate SpatialSQL Component, Query2SpatialDB Component, Spatial2GML Component 등으로 구성된다. Generate SpatialSQL Component는 MiddleWare Layer로부터 전달 받은 질의 데이터나 영역 데이터를 통해 SQL문을 생성하며, Query2SpatialDB Component는 SQL문을 이용하여 SpatialDB로부터 새로운 지도 영역을 검색하며, Spatial2GML Component는 SpatialDB로부터 검색된 새로운 지도 영역 정보를 이용하여 GML 파일을 생성한다.

SHP2DB 응용 컴포넌트는 ESRI사의 Shape포맷으로 저장된 전자지도 파일을 읽어 각 지도정보를 SpatialDB에 저장한다.

〈표 2〉 MiddleWare Layer 컴포넌트 인터페이스

컴포넌트 그룹	컴포넌트별 IDL 정의
User I/O Processing Component	<pre> module QueryConverter { interface IQueryConverter { boolean setUserInput (); }; }; module GMLPresentation { interface IGMLPresentation { boolean setGMLPresentation (); }; }; </pre>
GML-Image/SVG Map Generation Component	<pre> module GMLIntegrator { interface IGMLIntegrator { boolean setMOGML (); Rectangle getBoundary (); boolean setSpatialGML (); octet getIntegratedGML (); }; }; module GML2Image { interface IGML2Image { boolean setGML2Image (); octet getImage (); }; }; module GML2SVG { interface IGML2SVG { boolean setGML2SVG (); octet getSVG (); }; }; </pre>

〈표 3〉 Server Layer 컴포넌트 인터페이스

컴포넌트 그룹	컴포넌트별 IDL 정의
MO2GML Conversion Component	<pre> module Query2MOEngine { interface IQuery2MOEngine { boolean setQuery2MOEngine (); MOInfo_List getMOInfo (); }; }; module MO2GML { interface IMO2GML { boolean setMO2GML (); octet getMOGML (); }; }; </pre>
Spatial2GML Conversion Component	<pre> module GenerateSpatialSQL { interface IGenerateSpatialSQL { boolean setGenerateSpatialSQL (); SpatialSQL getGeneratedSpatialSQL (); }; }; module Spatial2GML { interface ISpatial2GML { boolean setSpatial2GML (); octet getSpatialGML (); }; }; </pre>
SHP2DB Application Component	<pre> module SHP2DB { interface ISHP2DB { boolean setMapPropertyInfo (); }; interface ITOPO2DB { boolean setTopoPropertyInfo (); boolean setTurnInfo (); }; }; </pre>

본 논문에서는 컴포넌트들의 인터페이스 설계에서 OMG CORBA에서 표준으로 정해놓은 IDL(Interface Definition Language)을 적용하였다. 이를 통해 구현을 위해 사용될 프로그래밍 언어에 독립적인 인터페이스를 정의할 수 있도록 지원해줌으로써 객체 생성에 사용된 구현 언어들 간의 상호 운용성을 지원할 수 있다.

MiddleWare Layer의 컴포넌트들은 각기 하나의 인터페이스를 가지며, 각 인터페이스는 컴포넌트에서 독립적인 작업 수행 시 필요한 외부 메시지를 입력 받고 입력 받은 메시지를 통해 처리한 작업 결과를 출력하기 위한 추상 메소드들로 구성되었다.

Server Layer의 컴포넌트들도 MiddleWare Layer의 컴포넌트와 마찬가지로 각기 하나의 인터페이스를 가지고 있으며, 각 인터페이스들도 역시 메시지의 입출력을 위한 추상 메소드들로 구성하였다. 단, SHP2DB 컴포넌트의 경우 컴포넌트 내에서 실현되는 기능이 두 개로 구분되어지기 때문에 두 개의 인터페이스로 구성하였다.

MO2GML 변환 컴포넌트에서는 (그림 3)에서 보인 바와 같은 스키마를 GML문서의 헤더로서 포함하여 사용자 질의에 대응하여 생성된 이동체 위치정보를 인코딩한다.

```

<?xml version = "1.0" encoding = "UTF-8"?>
<schema targetNamespace = "http://moms.etri.re.kr/MO" xmlns = "http://www.w3.org/2001/XMLSchema" xmlns:gml = "http://www.opengis.net/gml" xmlns:xlink = "http://www.w3.org/1999/xlink" xmlns:mo = "http://moms.etri.re.kr/MO" xmlns:xls = "http://www.opengis.net/xls" elementFormDefault = "qualified" version = "0.1">
  <annotation>
    <appinfo>MOSchema.xsd v0.1 2003-10</appinfo>
    <documentation xml:lang = "kr">GML schema for the MO Data</documentation>
  </annotation>
  <!-- import constructs from the GML Feature and Geometry schemas -->
  <import namespace = "http://www.opengis.net/gml" schemaLocation = "feature.xsd"/>
  <import namespace = "http://www.opengis.net/xls" schemaLocation = "ADT-Nav.xsd"/>
  <!-- === global element declarations === -->
  <element name = "MO" type = "mo:MOType" substitutionGroup = "gml:_FeatureCollection"/>
  <element name = "carnum" type = "string"/>
  <element name = "zone" type = "mo:ZoneType"/>
  <element name = "QueryType" type = "mo:QueryCategory"/>
  <element name = "Trajectory" type = "mo:TrajectoryQueryType"/>
  <element name = "Location" type = "mo:LocationQueryType"/>
  <element name = "Time" type = "mo:TimeQueryType"/>
  <element name = "SQL" type = "mo:SQLQueryType"/>
  <element name = "Prob" type = "string"/>
  ...
  <complexType name = "MOType">
    <complexContent>
      <restriction base = "gml:AbstractFeatureCollectionType">
        <all>
          <element ref = "xls:ReferenceSystem" minOccurs = "0"/>
          <element ref = "xls:BoundingBox" minOccurs = "0"/>
          <element ref = "mo:QueryType" minOccurs = "0"/>
          <element ref = "mo:Trajectory" minOccurs = "0"/>
          <element ref = "mo:Location" minOccurs = "0"/>
          <element ref = "mo:Time" minOccurs = "0"/>
          <element ref = "mo:SQL" minOccurs = "0"/>
        </all>
      </restriction>
    </complexContent>
  </complexType>
</schema>

```

(그림 3) MO2GML 스키마

5. 구현

5.1 시스템 환경

본 논문에서 제안한 개방형 이동체 프리젠테이션 인터페이스는 Java언어로 작성되었으며, Intel Pentium 4 Xeon Dual Processing 워크스테이션급PC에서 MS Windows XP 운영체제를 기반으로 웹 서버(Resin v2.x)와 JDK v1.3.x을 토대로 서비스된다.

GML 데이터의 생성 및 분석을 위해 JDOM API를 이용하였다. JDOM을 통해 GML 데이터를 생성할 수 있으며, 또한 GML 데이터로부터 공간 객체를 분석해 낼 수 있다. J2EE를 바탕으로 개발하였기 때문에 현재의 윈도우 운영체제가 아닌 기타 유닉스 및 리눅스 환경에서도 서버를 구동할 수 있도록 하였다. J2EE 환경은 WAS를 기반으로 구축되었으며, 본 과제에서는 Resin 2.1.9 Server를 사용하여 Servlet과 JSP를 구동하였다.

전자지도는 ESRI사의 Shape 형식의 청주지역 파일을 데이터베이스로 변환구축하고, 사용자 질의입력에 대하여 위저드 방식(Wizard method)의 인터페이스 및 SQL입력 방식을 지원한다.

부 문	구성 및 기능	
Server	구 성	<ul style="list-style-type: none"> OS : Windows XP 프로페셔널 JavaVM : JRE1.4 이상 WAS : Resin21.9
	기 능	<ul style="list-style-type: none"> 스크립트 인코딩/디코딩 이동체 위치정보와 전자지도 매칭 SHP2GML(SHP2DB, DB2GML) SHP2SVG, SHP2SVG 변환 HTML 문서 또는 이미지 처리 Java 소켓통신 처리 Web Client 용 애플릿 배포 담당 GMI2Image
Clint(Web)	구 성	<ul style="list-style-type: none"> OS : Windows 98/NT/2000/XP Internet Explorer 5.0 이상
	기 능	<ul style="list-style-type: none"> 질의 입력(위저드 및 MOQL) 소켓통신 요청 GML 처리용 DOM Interface 전자지도 및 질의결과 출력

(그림 4) 인터페이스 환경 및 기능

질의형식 위치조회에 해당하는 질의는 (그림 5)의 순서와 같이 1) 차량번호입력, 2) 검색 날짜/시간 입력의 순으로 진행된다. 입력을 완료한 상태에서 SQL 보기 버튼을 누르면 실제로 이동체 엔진에 전달되는 MOQL을 확인할 수 있다.



(그림 5) 위저드 방식의 사용자 질의 입력기

검색조건을 다 입력하고 검색 버튼을 누르면 (그림 6)과 같이 전자지도상에 해당 이동체의 위치가 중앙에 표시되고 검색결과 창에는 이동체의 차량번호와 검색시간, 위치(X,Y 좌표)가 나타난다. 지도를 다른 곳으로 이동시켰더라도 결과창에서 위치좌표 부분을 마우스로 클릭하면 해당 위치로 지도가 이동된다.

지도 창에 지도가 나타나면 지도 조작 아이콘을 통해서 지도를 확대/축소/이동할 수 있으며 각 Level 버튼을 통해 해당 레벨의 지도를 확인할 수 있다.



(그림 6) 사용자 질의 실행결과 및 GML보기 화면

5.2 실험

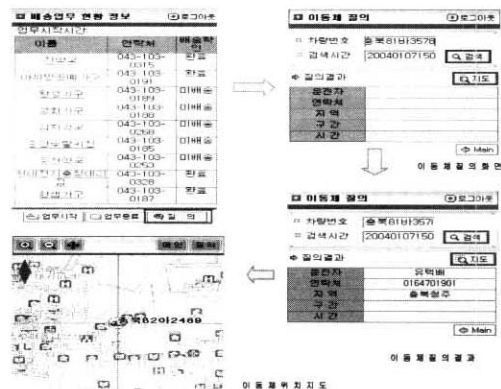
H전자충청대리점은 2004년 1월 6일에 전자물류거래를 통해 6개 물품을 S물류로 발송을 의뢰하였다. 2004년 1월 7일 e-Logistics 운송차량관제센터의 관리자는 회원사인 S물류

의 운송내역정보를 토대로 모의주행경로를 생성하며, S물류의 운전자는 PDA로그인후 자신의 배달내역을 센터로부터 수신하여 운송/배달 업무를 시작한다. S물류 운전자는 배달 중 자신의 위치를 전송하고 고객위치로 전자지도도를 통해 확인하며 배달상태를 입력 및 전송한다. 한편 H전자충청대리점에서는 의뢰물품의 배달상황 및 차량정보를 수시로 확인하며, S물류관리자는 소속 차량의 운행상태를 상시 확인하고 운전자가 업무를 종료하면 미배송 상태 일괄 입력(전송)한다. 그리고 S물류관리자는 상시 차량별 캐적 검색/분석, 이벤트&추론 통계정보 분석(최적경로 생성 자료로 활용)한다.



(그림 7) H전자 충청대리점의 배송조회 화면

운전자가 배송업무현황을 확인하고 업무시작 버튼을 누르면 업무시작을 확인하는 경고창이 나타나고 사용자가 확인을 누르면 상단에 업무시작시간이 나타나고 리스트에서 배송 확인을 확인할 수 있는 CheckBox가 생성된다. 리스트에 나타난 배송확인 미배송을 Check하면 완료된 형태로 서버에 통보되고 잘못 눌러서 완료 시킨 경우는 다시 완료 CheckBox를 클릭하여 취소할 수 있다. 운전자가 업무를 종료하고자 할 경우 업무종료 버튼을 누르면 업무종료를 확인하는 경고창이 나타나고 다시 확인을 누르면 업무가 종료되었다는 메시지와 함께 현재의 배송업무 현황이 보인다.



(그림 8) PDA에서 운전자의 배송업무 실행

또한 운전자가 질의버튼을 클릭하면 차량의 위치를 조회할 수 있는 창이 나타나고 차량번호와 검색시간을 입력후 검색 버튼을 누르면 질의결과가 Text형태로 나타난다. 지도버튼을

누르면 지도항에서 조회한 차량의 위치를 확인 할 수 있다. 모바일용 전자지도는 확대/축소/이동의 기능을 제공한다.

5.3 평 가

기존의 대부분의 차량관제시스템들은 시스템 간 폐쇄적인 특성으로 시스템 내 지리정보/위치정보 통합/표현이 어려우며, 시스템 수가 증가할수록 난이도가 증가하고 막대한 구축비용이 요구되는 문제점을 가진다.

본 논문에서 제안한 개방형 이동체 프리젠테이션 인터페이스는 GML 기반의 개방형 컴포넌트 구조를 통해 지리정보와 이동체 정보의 효율적인 통합뿐만 아니라 기타 시스템 간의 정보 교환/통합도 용이한 특성을 가진다.

또한 e-Logistics 환경의 물류회사를 대상으로 실시간 차량관리 지원을 위한 핵심기술로서 내/외부 고객만족도 개선을 기대할 수 있다. 즉, 기존의 차량추적시스템들의 정보통합을 위한 표준 모델로 활용할 수 있을 것으로 기대하며, 표준 명세 및 개방형 컴포넌트 구조로 광역화지역을 대상으로 실시간 서비스가 요구되는 환경에서 소요비용의 절감과 효율적인 차량운행 분석 및 제어를 기대할 수 있다.

6. 결 론

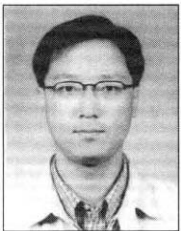
인터넷을 기반으로 하는 관련 기술과 비즈니스 영역이 확장 및 발전함에 따라 이전에 개발된 시스템에 기능 및 성능을 추가하고 보완하는 필요성이 제기되어 왔으며, 이를 해결하기 위한 연구들이 진행되어 왔다.

본 논문에서는 GML을 기반으로 컴포넌트 기법을 적용한 이동체 프리젠테이션 인터페이스를 제안하였다. 본 논문에서 구현된 인터페이스를 통하여 표준화된 방식을 적용한 지리정보 및 차량 위치데이터의 공유 및 전송을 가능하며, 이를 통한 향상된 상호운용성을 확보할 수 있다. 이동체 프리젠테이션 인터페이스를 기반으로 차세대 물류차량 관제 시스템을 구축 및 운용비용과 시간 절감의 기대효과를 확신한다. 본 논문에서 연구된 이동체 기술은 최근 각광을 받고 있는 유비쿼터스 기술 분야의 정보응용기술로서 확장하기 위한 연구를 진행 중에 있다.

참 고 문 헌

[1] Y. A. Ahn, D. H. Kim and K. H. Ryu, "Design of A Moving Object Management System for Tracking Vehicle Location," The KIPS Transactions : Part D, Vol.9-D, No.5, pp. 827-836, Oct., 2002.
 [2] D. D'souze and A. Wills, "Objects, Components, and Frameworks with UML : The Catalysis Approach," Addison-Wesley, 1998.
 [3] R. H. Guting et al., "A Foundation for Representing and Querying Moving Objects," ACM Transactions on Database Systems, Vol.25, No.1, pp.1-42, 2000.

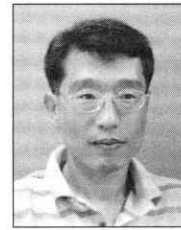
[4] G-XML, <http://www.dpc.or.jp/>.
 [5] ISO/TC 211, <http://www.isotc211.org/>.
 [6] I. Jacobson, G. Booch and J. Rumbaugh, "The Unified Software Development Process," Addison-Wesley, 1999.
 [7] D. H. Kim, H. A. Lee, J. S. Kim, Y. A. Ahn and K. H. Ryu, "Moving Objects Relational Model and Design for e-Logistics Applications," ICITA-2002, Nov., 2002.
 [8] OpenGIS Consortium, Inc., Geography Markup Language (GML) 3, 2003.
 [9] O. Wolfson, P. Sistla, B. Xu, J. Zhou, S. Chamberlain, N. Rishe and Y. Yesha, "Tracking Moving Objects Using Database Technology in DOMINO," Proc. of NGITS '99, The 4th Workshop on Next Generation Information Technologies and Systems, Zikhron-Yaakov, Israel, pp.112-119, 1999.
 [10] World Wide Web Consortium, Inc. Scalable Vector Graphics (SVG), <http://www.w3.org/TR/SVG11/>, 2003.
 [11] 김동호, 이현아, 이혜진, 김진석, "통합 이동체 위치데이터 인터페이스 모델", 한국정보처리학회 추계학술대회논문집, pp. 631-634, Nov., 2003.
 [12] 마르미, <http://www.component.or.kr>.
 [13] 이현아, 이혜진, 김동호, 김진석, "이동체 관리 시스템을 위한 이동체 질의어 설계", 한국정보과학회 추계학술대회논문집, pp.148-150, Oct., 2003.
 [14] 정일원, 배혜영, "GML 기반의 이동객체 데이터모델 및 질의어 명세", 정보처리학회논문지D, 제11-D권 제1호, pp.39-50, Feb., 2004.
 [15] 한국전자통신연구원, "50대 품목 기술/시장 보고서 : GPS", 한국전자통신연구원, 2001.



김 동 호

e-mail : kdh@etri.re.kr
 1993년 충북대학교 전자계산학과(학사)
 1995년 충북대학교 대학원 전자계산학과 (이학석사)
 1999년 충북대학교 대학원 전자계산학과 (이학박사)

1999년~현재 ETRI 우정기술연구센터 선임연구원
 관심분야 : 시공간 데이터베이스, 이동체 기술, 지리정보시스템, 위치기반서비스, 물류응용정보시스템 등



김 진 석

e-mail : kimjs@etri.re.kr
 1982년 울산대학교 전자계산학과(학사)
 1988년 동국대학교 대학원 전자계산학과 (공학석사)
 2004년 충북대학교 대학원 전자계산학과 (이학박사)

1983년~현재 ETRI 우정기술연구센터 u-Post연구팀장/책임연구원
 관심분야 : CSCW, 소프트웨어공학, 위치기반서비스, 물류정보시스템 등