

인터넷에서의 버스 노선 안내 서비스 시스템의 구현

유승원 차상균

{mokmon, chask}@kdb.snu.ac.kr

서울대학교 전기공학부

Implementation of a Busline Guidance System on Internet

Seongwon Yoo Sang K. Cha

School of Electrical Engineering, Seoul National University

요약

최근 자동차 수의 급속한 증가와 대중 교통의 발달로 인하여 ITS(Intelligent Transportation System)과 같은 지리/교통 안내 시스템의 연구 개발이 활발히 이루어지고 있다. 이러한 시스템의 중요한 기능 중의 하나로 대중 교통의 관리와 안내 서비스가 있다. 특히 우리나라의 경우 아직 많은 사람들이 대중 교통을 이용하고 있고, 대중 교통이 대도시 교통 체증의 중요 해결책으로 인식되므로 이러한 기능이 더욱 필요하다. 본 논문에서는 인터넷을 통하여 접근할 수 있는 버스 노선 안내 서비스 시스템을 개발한 경험을 소개한다. 버스 노선 안내 서비스 시스템은 사용자가 접근하기 용이해야 하며 실용적인 서비스 결과를 제공할 수 있어야 한다. 또한 추후 개발 되는 여러 ITS 구성 요소들과 유기적으로 결합할 수 있는 상호 운용성이 필요하다. 본 논문에서는 분산 객체 미들웨어의 표준인 CORBA를 이용하여 인터넷을 통한 접근이 용이하고 상호 운용성이 뛰어난 시스템 구조를 제안하였다. 또한 버스 노선 안내를 효과적으로 하기 위한 모델링 방법과 스키마 구조를 제안하였다. 이 시스템의 구조와 스키마는 추후 다른 대중 교통 안내 서비스 시스템과의 연계 혹은 다른 ITS 구성 요소와의 연계에 쉽게 대응할 수 있을 것으로 기대된다.

1 서론

최근 자동차의 수가 급격히 증가하고 대중 교통이 다양하게 발달함에 따라 이들의 효율적인 관리와 사용자들의 다양한 요구를 해결하기 위하여 ITS(Intelligent Transportation System), IVHS(Intelligent Vehicle and Highway System)등의 연구가 활발히 이루어지고 있다.

대중 교통 관리와 여행자 안내 시스템은 ITS의 중요한 기능의 하나로 일반인들이 가장 쉽게 접할 수 있는 시스템이다. 유럽의 경우 이미 EUROBUS 프로젝트를 통하여 전체 유럽의 대중 교통 시스템의 통합을 시도하고 있다[1]. 우리나라의 경우 서울시민의 승용차/승합차의 통행은 29.3%인데 반하여 버스/지하철에 의한 통행은 53.5%를 차지할 정도로 대중 교통의 이용량이 많으며[2], 특히 대도시의 교통난을 해소하기 위하여 대중교통은 전략적으로 매우 중요하다.

그러나 이러한 대중교통의 중요성에 비하여 대중 교통 안내 서비스 시스템에 대한 연구는 그다지 많이 이루어지지 않고 있다. 현재 구현된 시스템은 버스나 지하철의 노선을 보여 주거나 역이름과 같은 비공간 데이터를 이용한 검색 등의 간단한 서비스만을 제공하는 것이 대부분이며[3], 특히 다른 교통 정보 관리 시스템과의 연계 등을 고려하지 못하고 있다. 한편 [4]에서는 웹 기반의 버스 노선 안내 시스템을 구현하며 공간 데이터의 사용 및 다른 교통 정보 관리 시스템과의 연계 가능성을 모색하였다. 이 시스템은 서버/클라이언트 구조로 자바 애플릿으로 구현된 클라이언트가 사용자로부터 질의를 받고 CGI를 통하여 서버로 넘겨준 후 다시 결과를 받아 사용자에게 보여준다.

본 논문에서는 CORBA와 지리 정보 데이터베이스를 기반으로 버스 노선 안내 시스템을 구현한다. ITS는 여러 구성 요소로 이루어진 시스템으로 각 요소의 설계 및 보수의 편의와 각 요소들간의 상호 운용성을 위하여 컴포넌트화되는 것이 필요하다. 이러한 컴포넌트 사이의 호환성을 제공하는 산업 표준으로는 CORBA[5], OLE/COM[6]이 존재한다. OLE/COM이 아직 윈도우즈 운영체제를 중심으로 지원되는데 비해 CORBA는 PC, 유닉스 서버, 메인 프레임 등의 다양한 환경에 대해 지원되므로 거시적 통합에 유리하다. 논문에서 제시하는 시스템은 기존의 데이터베이스에 CORBA를 사용하는 wrapper를 두어 서비스를 제공하는 컴포넌트를 구성하고 이러한 서버 컴포넌트들로부터 데이터를 제공받아 사용자에게 서비스를 제공하는 클라이언트 컴포넌트로 구성된다. 이러한 구조는 각 컴포넌트의 통합 및 상호 운용이 용이하고, 비교적 쉽게 이룰 수 있는 장점이 있다.

또 논문에서는 버스 정류장, 노선 정보를 노선 안내에 적합하게 모델링 하는 방법에 대해 기술하였다. 이 방법은 복잡한 연계 구조를 가진 대중 교통 네트워크를 최단 경로 탐색을 위한 그래프로 대응시키는 것으로 다른 대중 교통 안내 서비스에 적용하거나 여러 대중 교통의 연계 서비스에도 적용할 수 있을 것으로 기대된다.

본 논문은 다음과 같이 구성되어 있다. 2절에서 논문에서 제시하는 시스템 구조를 설명하고 3절에서 구현된 시스템의 구현 환경과 버스 정류장과 노선 정보의 모델링 방법, 이를 이용하는 버스 노선 안내 알고리즘에 대해 기술한다. 4절에서는 결론과 향후 계획을 기술하도록 하겠다.

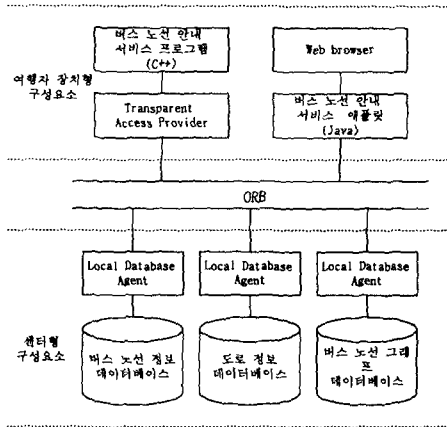


그림 1: 버스 노선 안내 서비스 시스템의 구조

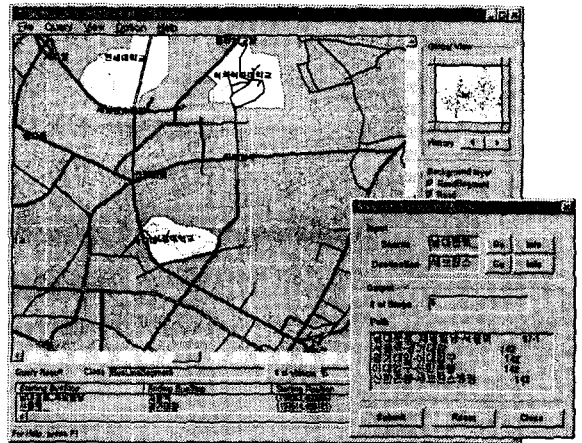


그림 2: 버스 노선 안내 서비스 시스템 화면

2 시스템 구조

논문에서 제안하는 시스템의 구조는 그림 1과 같다. 버스, 지하철과 같은 대중 교통 시스템은 하나의 통합된 데이터베이스 또는 여러 곳의 기관에서 관리되는 경우가 많다. 예를 들어 우리나라 지하철은 도시철도공사, 지하철 공사, 철도청의 세 곳에서 운영하고 있다. 또한 교통 정보와 지리 정보 등은 공간 속성을 가지므로 공간적으로 멀리 떨어진 데이터는 서로 다른 데이터베이스에 존재하는 경우가 많다. 특히 ITS에서는 사용자가 휴대용 단말기나 노선 안내 시스템과 같은 여행자 장치형 구성 요소에 접근하여 교통 관제 센터 등과 같은 센터형 구성 요소들에서 교통/지리 정보를 받아오게 되는데 이러한 센터형 구성 요소는 도로 장치나 차량 장치로부터 정보를 분석/처리하는 곳으로 그림 1과 같이 분산된 데이터베이스의 형태라고 생각할 수 있다.

그림 1에서는 버스 정보와 도로 정보가 각각 서로 다른 데이터베이스에 구축되어 있고 이들로부터 버스 노선 안내 서비스를 하기 위하여 버스 노선 그래프 데이터베이스가 구축되어 있다. 버스 정보 데이터베이스는 버스 회사, 버스 노선, 정류장 등의 정보가 있을 수 있으며 도로 정보 데이터베이스는 각 도로의 종류, 특성, 공간 정보, 교통량 등의 정보가 있을 수 있다. 사용자는 이들 데이터베이스에 동시에 접근하여 도로와 버스 노선에 대한 정보를 얻거나 노선 안내 서비스를 받게 된다. 각 데이터베이스는 서로 다른 기관에서 운영할 수도 있고 서로 다른 형태를 가질 수도 있으므로 여행자 장치형 구성 요소들에 일관된 인터페이스를 제공하기 위하여 wrapper로서 LDA(Local Database Agent)[7]를 두고 있다. LDA는 데이터베이스에 종속적인 인터페이스를 감추고 여행자 장치형 구성 요소들에 일관된 인터페이스를 제공하는다.

여행자 장치형 구성 요소는 각 데이터베이스의 LDA를 통하여 일관된 인터페이스로 원하는 데이터를 전송 받을 수 있게 된다. 여행자 장치형 구성 요소는 그림 1 애플릿 형태와 같이 직접 CORBA 인터페이스를 사용할 수도 있다. 그러나 캐쉬 등의 추가적인 기능과, 네트워크 투명한 인터페이스를 제공하기 위하여 TAP(Transparent Access Provider)[7]과 같은 계층을 추가할 수도 있다. 그림 1에서 버스 노선 안내 서비스 애플릿은 CORBA의 차바 바인딩을 이용하여 직접 LDA에 접근하는 것이며 사용자는 웹을 통하여 버스 노선 정보 및 안내 서비스를 받을 수 있게 된다.

다. 그러나 버스 노선 안내 서비스 프로그램은 TAP interface를 통하여 LDA에 접근하는 C++ 응용 프로그램의 형태가 된다.

이와 같은 시스템은 데이터베이스와 LDA가 하나의 컴포넌트를 이룰 수 있어 데이터베이스의 세부 사항이 변화한다고 하더라도 여행자 장치형 구성 요소들이 영향을 받지 않을 수 있으며 새로운 데이터베이스가 추가되더라도 일관적인 인터페이스를 제공하는 LDA를 추가함으로써 기존의 여행자 장치형 구성 요소들을 그대로 사용할 수가 있는 장점이 있다.

3 시스템 구현

3.1 구현 환경

그림 1의 데이터베이스는 상용 ODBMS인 Objectivity/DB를 사용하였고, LDA는 Solaris 2.5.1에서 SPARC C++ 컴파일러로 구현되었다. ORB는 IONA의 Orbix 2.3을 사용하였으며 여행자 장치형 구성 요소 부분은 MS-Windows상에서 Visual C++ 5.0과 JDK로 구현하였다.

도로 정보 데이터베이스에 저장된 도로 데이터는 약 7만개의 교차로와 10만개의 도로로 이루어진 서울시 도로 데이터를 사용하였으며 이 도로 데이터 위에 버스 노선 및 정류장 정보를 입력할 수 있는 그래픽 입력 툴을 제작하여 부분적으로 버스 노선 데이터를 입력하였다.

논문에서 기술한 버스 노선 안내 서비스 프로그램을 이용하여 남대문로에서 신촌 세브란스 병원까지의 버스 노선을 찾아주는 간단한 예가 그림 2에 나타나 있다.

3.2 데이터 모델링

버스 노선 정보와 도로 정보는 그림 1과 같이 서로 다른 데이터베이스에 나누어 보관되어 있으며 이들 정보는 효율적인 노선 검색을 위하여 통합된 materialized view를 갖는 것이 좋다. 버스 노선 안내 시스템에서 중요한 것은 이렇게 생성된 버스 노선 그래프 데이터베이스이므로 버스 노선 그래프의 스키마에 대해서 설명하기로 한다.

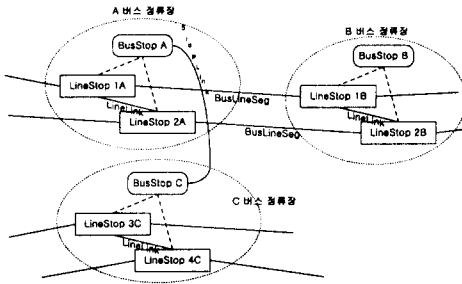


그림 3: 버스 정류장 및 버스 노선의 모델링

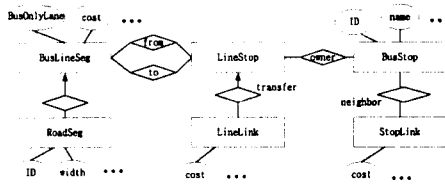


그림 4: ER diagram

그림 3과 같이 버스 정류장은 BusStop 객체에 대응된다. 그러나 하나의 버스 정류장에도 여러 버스가 정차하므로 개념적으로 각 버스는 LineStop이라는 객체에 정차하는 것으로 하고 LineStop은 자기가 존재하는 BusStop과 관계를 가진다. 각각의 LineStop은 BusLineSeg 객체에 의해 이어져서 하나의 버스 노선을 만들게 되며 BusLineSeg는 도로를 나타내는 RoadSeg 객체를 저장한다. 하나의 BusStop에서 LineStop 끼리는 LineLink로 연결되어 버스를 갈아타는 것을 표현하도록 한다. 즉 하나의 LineStop에서 다른 BusStop의 LineStop을 가리키면 같은 버스를 타고 다음 정거장으로 가는 것이고, LineLink를 통하여 같은 BusStop의 다른 LineStop을 가리키면 버스를 갈아타는 것이 된다. 그런데 실제로 주변의 버스 정류장을 보면 기다리는 승객을 분산시키기 위하여 같은 곳에 정류장을 여러 개 만드는 경우도 있고, 혹은 정류장은 다르지만, 서로 인접하여 있어서 도보로 이동하여 쉽게 갈아탈 수 있는 정류장들이 존재한다. 이와 같은 경우는 다른 BusStop으로 이동하는 것이지만 실제로는 버스를 타는 것이 아니고 걸어서 갈아타는 것으로 처리해 주어야 한다. 이런 경우를 처리하기 위하여 인접 버스 정류장끼리 StopLink를 통하여 연결한다.

지금까지 설명한 스키마를 ER diagram으로 표시하면 그림 4와 같다. 이와 같은 모델링은 버스 노선 뿐만이 아니라 노선과 연계가 존재하는 대부분의 대중 교통 시스템에 큰 수정없이 반영할 수 있으며 서로 다른 대중 교통 시스템과의 연계에도 사용될 수 있을 것으로 생각한다.

3.3 버스 노선 안내 알고리즘

앞의 절에서 설명한 스키마는 복잡한 버스 노선과 노선 간의 연계 구조를 노드와 링크로 이루어진 그래프 구조로 대응시킬 수 있다. 즉 BusStop, LineStop이 node가 되며 BusLineSeg, LineLink, StopLink가 링크가 된다. 각 링크의 cost를 계산하면 버스 노선 안내 문제는 weighted graph의 최단 경로 탐색 문제로

대치되게 된다. 각 링크의 cost는 다음과 같이 계산한다.

- BusLineSeg: BusLineSeg를 이루는 각 RoadSeg의 길이의 합 * 평균 버스 속도 (버스 전용 차선의 경우 평균 버스 속도에 가중치를 둔다.)
- LineLink: 갈아타는 버스의 평균 배차 간격 / 2 + 가중치(갈아타는 버스의 요금)
- StopLink: BusStop 간의 거리 / 평균 보행 속도 + 교차로 횡수 * 평균 신호 대기 시간 + 가중치(갈아타는 버스의 요금)

이와 같이 생성된 그래프는 dijkstra 혹은 A*와 같은 일반적인 경로 탐색 알고리즘을 큰 수정없이 적용할 수 있다.

4 결론 및 향후 계획

본 논문에서는 CORBA와 지리 정보 데이터베이스를 기반으로 하여 인터넷에서 버스 노선 안내 서비스를 제공하는 시스템의 구현 경험을 기술하였다. 기존의 OO 패러다임에 더하여 CORBA와 같은 분산 객체 패러다임의 영향으로 근래의 소프트웨어는 컴포넌트화되고 있으며 GIS 관련 응용 프로그램이나 ITS의 각 모듈들도 컴포넌트화 되는 경향이 있다. 그러므로 논문에서 제시하는 것과 같이 데이터베이스와 LDA에 의한 서버 컴포넌트와 CORBA를 통하여 이들에 접근하는 클라이언트 컴포넌트로 이루어진 버스 노선 안내 시스템은 사용자의 접근의 용이하며 ITS의 다른 구성 요소들과의 결합이 비교적 쉬운 상호운용성을 제공한다. 또한 논문에서 제시한 버스 노선 및 정류장의 모델링과 노선 안내 알고리즘은 현재 비교적 연구가 이루어지지 않는 대중 교통 안내 시스템의 실용적인 구현 사례로서 의미가 있으며 다른 대중 교통 시스템에 적용이 용이하다는 장점을 가진다. 향후 계획으로는 노선 안내 알고리즘의 휴리스틱 적용과 실험, 버스와 지하철 간과 같이 다른 대중 교통 시스템과의 연계가 가능한 시스템의 설계와 알고리즘의 개발 등이 이루어질 것이다.

참고문헌

- [1] B. de Saint-Laurent, C. Chauvet, and F. Khodja, "Advanced public transport passenger information: Findings of the eu-robus project," in *Proceedings of the the Vehicle Navigation and Information System conference*, 1993.
- [2] 서울시정개발연구원, 서울시 교통센서스 결과에 따른 정책 방향 및 활용방안, 1998.
- [3] 서울시버스노선안내, "http://www.metro.seoul.kr/kor/living/frame_bus.html."
- [4] 배수강 이승룡 최대순 정태충 승현우, "웹기반 대중교통 안내시스템 설계 및 구현," *정보과학회논문지(C) 제5권 제4호*, pp. 426-439, 1999.
- [5] Object Management Group, Framingham, Massachusetts, *The Common Object Request Broker: Architecture and Specification*, 1998.
- [6] D. Box, *Essential COM*. Addison Wesley, 1998.
- [7] S. K. Cha, K. Kim, C. Song, J. Kim, Y. Kwon, and J. Jun, "A middleware for transparent access to multiple spatial object databases," in *Proceedings of the Interop*, 1997.