

컴포넌트에 기반한 여행자정보고급화 시스템의 설계 †

(A Design of Advanced Traveler Information System based on Component)

김진환*, 장재영**, 이봉규***

Jin-hwan Kim, Jea-Young Chang, Bong-Gyou Lee

요약 지능형 교통 시스템(ITS)은 여행자에게 출발 전 또는 출발 후 도움이 되는 서비스와 유용한 데이터를 제공하는 여행자정보고급화 시스템을 포함하고 있다. 이 시스템은 출발전교통정보 안내 서비스시스템, 운전중 교통정보제공 서비스시스템, 동적주행안내 서비스시스템 들로 구성되어 있다. 여행자정보고급화 시스템은 표준화 작업을 총괄하는 기준 프레임워크인 국가 아키텍처에 따라 설계될 필요가 있다. 최근 소프트웨어 기술이 급속히 개선되고 안정화됨에 따라 미리 개발된 강력한 ITS 기술도 재사용할 필요성이 있다. 컴포넌트와 개방형 인터페이스에 기반한 ITS 표준화는 현재 ITS 기술의 재사용성 문제를 해결하는 방법이 된다. 본 논문은 UML을 이용하여 컴포넌트에 기반한 여행자정보고급화 시스템을 설계한 결과를 제시하고 있으며 이 UML 방법은 새로 개발되는 ITS 컴포넌트를 위한 표준화된 모델을 제공할 것으로 기대된다.

ABSTRACT ITS includes an Advanced Traveler Information System(ATIS) that provides travelers with service and facility data for the purpose of assisting prior to embarking on a trip or after the traveler is underway. ATIS consists of three major subsystems, which are a Pre-trip Traffic Information subsystem(PTIS), an En-route Traffic Information Subsystem(ETIS), and a Dynamic Route Guide Subsystem(DGIS). ATIS needs to be designed and implemented in accordance with the National ITS Architecture, a reference framework that spans all of standards activities. Recently, as software technology is rapidly improved and stabilized, there are some needs to reuse pre-developed and powerful ITS technology. ITS standardization based on components and open interfaces becomes a way to solve these reusability of current ITS technology. This paper focuses on how could we design and implement ATIS based on the component with the aid of UML(Unified Modeling Language). The UML methodology is expected to provide a standardized model for newly developed ITS components.

키워드 : 정보통신, 공간 정보, GIS, SHS, GNSS, ITS

1. 서론

지능형 교통 시스템(ITS: Intelligent Transport System) 서비스 분야는 신호교차로 제어에서부터 요금 자동징수, 각종 매체를 통한 정보의 제공, 차량의

정보 및 위치 추적 그리고 고속도로의 교통류 관리까지 매우 다양하다[1]. 현재 국가 ITS 아키텍처의 7개 서비스 분야중 하나인 여행자정보고급화 시스템(ATIS: Advanced Traveler Information System)은 기본적으로 첨단교통정보관리 시스템(ATMS: Advanced Traffic

† 본 연구는 2001년도 정보통신부의 "국가사회 정보화 정책연구"로부터 지원 받았으며 이에 감사드립니다.

* 한성대학교 부교수

** 한성대학교 전임강사

*** 한성대학교 부교수

Management System)을 통해 수집된 정보를 수집 및 재가공하여 사용자에게 전달하는 시스템이다[2]. 여행 자정보고급화 시스템은 정보제공을 위해 기존의 전광판과 인터넷, 유무선 ARS, Kiosk, 차내항법장치(CNS: Car Navigation System) 등을 사용하고 있다. 현재는 문자와 음성 위주의 정보전달이 주종을 이루고 있으나 점차 그래픽과 영상으로 변경되고 있다. ATIS의 가장 중요한 역할은 교통정보센터와 차량장치 간의 데이터 교환이며 이를 위한 표준화 작업과 함께 여러 기관에서 수집한 교통정보를 공유할 수 있는 틀이 마련될 필요가 있다.

유럽[3], 미국[4], 일본[5] 등은 물론 국내에서도 ITS 국가기본계획 설정 단계에서부터 이러한 표준화의 필요성이 대두되었으며 현재까지 표준화에 관련된 연구가 유관 기관들을 중심으로 활발하게 진행되고 있다[6]. 그러나 ITS 표준화 대상분야가 아키텍처 구조, 즉 아키텍처 인터페이스들과 그 사이의 데이터 흐름들을 분석하는 것에 중점을 두고 있기 때문에 향후 개별 시스템을 실제로 연계하여 통합 관리 운영 체계로 확장하는 데에는 현실적인 어려움이 예상된다. 즉 차세대 교통 체계인 지능형 교통 시스템의 다양한 서비스들을 제공하는 시스템들은 일반적으로 광범위한 영역 내에서 개별적 혹은 통합된 형태로 구성되어 작동하는 복잡 다단성이 내포되며 ITS 제반 기능과 서비스 범위, 체계의 구성과 환경, 지리적인 전개의 복잡성 등까지 고려할 때 시스템간 또는 서비스간의 비호환성 및 비상호운영성(non-interoperability) 문제를 해결하는 방법이 필요하다.

정보화의 급진전으로 각종 전문 영역이 정보기술과의 결합을 통하여 적극적인 발전을 도모하는 상황에서 ITS 분야도 결코 예외가 아니며 최근 정보기술 분야의 새로운 소프트웨어 패러다임(paradigm)인 분산 컴퓨팅 기법의 컴포넌트(component) 기술과 개방화(openness) 기술이 도입될 필요가 있다. 현재 개방화 기술은 분산처리 시스템의 핵심인 상호 운영성을 지원하며 이미 개발된 분산 처리 시스템들의 재사용성(reusability)을 가능하게 하는 것을 목적으로 하고 있다[7]. 컴포넌트 기술은 프로그램 개발에 소요되는 시간을 상당히 단축시킬 수 있으며 인터넷/인트라넷 환경에 적합한 분산 환경 컴퓨팅을 쉽게 구축할 수 있도록 설계되어 있다. 이러한 소프트웨어 기술의 변화는 지리정보시스템(GIS: Geographic Information Systems) 분야에도 이미 커다란 영향을 주었으며 ITS 분야의 기술적, 정책적 측면에서도 컴포넌트 기술이 매우 필요할 것으로 사료된다.

컴포넌트 기술은 소프트웨어 접근 면에서 하드웨어 플랫폼에 무관하며, 구성 면에서 플러그-앤-플레이 식의 소프트웨어 조립을 가능하게 한다. 또한 관리면에서도 소프트웨어 동적 재사용을 가능하게 하며 설계 면에서 컴포넌트 자체가 독립적인 성격을 가지는 기술이다. 이러한 컴포넌트 기술은 정의된 인터페이스를 통하여 접근할 수 있는 바이너리 수준의 객체로서 분산 컴퓨팅 환경에서 원격호출을 기반으로 하고 있다[8, 9]. 향후 인터넷과 분산/네트워크 컴퓨터에 대한 요구가 급증함에 따라 컴포넌트 기술의 필요성이 더욱 대두될 것으로 예상된다.

본 논문에서는 ITS 국가 아키텍처를 기반으로 여행자정보고급화 시스템(ATIS: Advanced Traveller Information System) 서비스[10]를 위한 컴포넌트들을 도출한 결과와 실제 UML 기법을 이용한 설계 결과를 기술한다. 본 논문의 구성은 다음과 같다. 우선 2절에서는 컴포넌트 기술 동향을 소개하며 3절에서는 ATIS 서비스 구성에 필요한 컴포넌트들을 기술한다. 4절에서는 ATIS 서비스를 위한 인터페이스 설계 결과를 제시한다. 마지막으로 5절에서는 결론을 기술한다.

2. 관련 연구

본 절에서는 컴포넌트의 개념과 개발 방법론에 대해서 개괄적으로 소개하고 이 방법이 성공적으로 적용된 GIS 분야에서의 컴포넌트 개발 현황을 기술한다.

2.1 컴포넌트 개발 방법론

객체지향 방법론(object-oriented methodology)은 기존의 구조적 방법론(structural methodology)이 제공하지 못했던 소프트웨어의 재사용성이나 개발과 유지보수의 편리성 등의 문제들을 해결하기 위한 새로운 소프트웨어 개발 방법론으로 각광을 받았다[12]. 그러나 실제 환경에서 순수한 객체지향 방법론에서 장점으로 내세운 소프트웨어 재사용 문제에 있어서 큰 효과를 발휘하지 못했다. 그 이유는 우선 객체지향 방법론 자체가 재사용의 목적으로 개발된 것이 아니며 확장성이나 추상화의 개념을 확실하게 보장하지 못했기 때문이다. 컴포넌트란 개념은 객체지향 방법론을 발전시켜 소프트웨어 재사용에 초점을 맞추어 제안된 새로운 개념이다. 우선 컴포넌트는 외부에서 이 컴포넌트의 서비스를 받을 수 있도록 메소드(method)나 속성 형태로 인터페이스를 정의한다. 그리고 그 인터페이스를 구현하기 위한 내부의 데이터나

프로그램 구조는 외부와 철저히 차단된다. 다시 말해서 컴포넌트의 외부적 형태는 인터페이스로 그 기능을 대변하게 된다. 좀더 구체적으로 컴포넌트는 다음과 같은 원칙을 따르는 소프트웨어의 단위라고 정의할 수 있다. [11]

○ 데이터와 프로그램의 통합 관리

컴포넌트는 데이터와 그 데이터를 처리하기 위한 프로그램으로 구성된다. 이 개념은 객체지향 기법의 클래스라는 개념과 유사하지만 하나의 컴포넌트를 구현하는데는 내부적으로 하나의 클래스나 그 이상의 클래스를 이용하여 구현된다.

○ 캡슐화(encapsulation)

컴포넌트를 사용하는 다른 소프트웨어 모듈은 이 컴포넌트의 내부 구현에서 완전히 차단되고 오직 인터페이스를 통해서만 서비스를 받을 수 있다.

○ 식별성(identification)

컴포넌트들은 현재의 상태와 관련 없이 유일한 식별자를 갖으며 이 식별자를 통해서 컴포넌트의 유일성을 판별한다.

이와 같은 특성을 갖는 컴포넌트의 개념은 기존의 구조적 방법론이나 순수한 객체지향 방법론과 비교하여 많은 장점을 갖는다. 우선 컴포넌트는 재사용성이 뛰어나다. 컴포넌트는 기능 위주로 설계되며 그 기능은 인터페이스를 통해서 다른 컴포넌트나 소프트웨어에게 서비스를 제공한다. 따라서 원하는 기능을 갖는 컴포넌트가 존재하면 다른 소프트웨어에서 쉽게 활용될 수 있다. 또한 컴포넌트의 상품화가 가능하다. 기존의 소프트웨어 상품들은 주로 패키지 형태로 상품화가 이루어졌으나 컴포넌트는 독립적으로 완전한 소프트웨어가 되지 못하더라도 상품가치가 있다. 컴포넌트의 개념을 활용하면 소프트웨어는 필요한 기능들을 갖춘 컴포넌트들을 조립하여 구성할 수 있다. 따라서 기존에 개발된 컴포넌트들을 적절히 활용한다면 소프트웨어의 설계 및 개발이 용이해지고 또한 유지보수 과정도 단순해진다. 마지막으로 컴포넌트의 장점은 표준화가 쉽다는 점이다. 컴포넌트는 인터페이스로 정의되는데 특정 서비스를 제공하는 컴포넌트를 정의하고 그 인터페이스를 표준으로 정의한다면, 그 인터페이스를 준수하여 개발된 모든 컴포넌트들은 어떠한 제품으로도 활용이 가능하게 된다. 실제로 지금까지 여러 그룹에서 컴포넌트 모델을 위한 사양이 개발되었다. 본 논문에서는 그 중에서 대표적인 컴포넌트 모델인 CORBA와 COM/DCOM에 대해서 간략히 소개한다.

COM/DCOM

COM/DCOM은 마이크로소프트사에서 개발한 것으로 클라이언트와 객체기반의 서버 프로그램간의 상호통신 방식을 정의한 모델이다[11]. 특히 서로 다른 언어나 개발 환경에서도 컴포넌트의 개발을 가능하게 해주는 프로그래밍 모델이다. COM에서 각각의 서비스는 하나의 COM 객체에 의해서 제공되며 이러한 COM 객체가 하나의 컴포넌트가 된다. COM 객체는 DLL, ActiveX 그리고 EXE 파일의 세 가지 형태로 제공되는데 DLL과 ActiveX는 동일한 주소 공간에서 동작하고, EXE는 실행파일 형태이므로 다른 주소공간에서 동작한다. DCOM은 COM을 네트워크 상으로 확장한 것이다. COM이 단일 시스템 내에서 혹은 단일 플랫폼 내에서 일어나는 것이라면 DCOM은 컴포넌트들이 네트워크 상에서 분산되어 있을 때 상호 통신하는 모델이다.

CORBA

CORBA는 분산 객체기술과 객체지향 기술을 통합한 표준 아키텍처로서 OMG(Object Management Group)에서 제정하였다[13]. COM 모델과는 달리 제정 초기부터 분산 객체에 초점을 맞추었으며 DCOM과 유사한 개념이라고 볼 수 있다. CORBA의 설계 목표는 운영체제나 개발 언어에 의존적이지 않도록 하는 것이었다. CORBA에서는 ORB(Object Request Broker)가 중요한 역할을 하는데 ORB는 객체간에 관계를 설정하는 일종의 미들웨어(middleware)로 클라이언트가 서버 객체가 제공하는 기능들을 투명하게 활용할 수 있는 기반을 제공한다. CORBA에서는 서비스 객체가 일종의 컴포넌트가 된다. 따라서 서비스 객체에 대한 정의는 인터페이스를 통해서 이루어지는데 IDL(Interface Definition Language)이라는 특별한 언어를 이용하여 정의한다. 따라서 컴포넌트, 즉 서비스 객체를 생성하기 위해서는 IDL로 인터페이스를 정의하고 IDL의 컴파일의 결과로 클라이언트에서는 어떠한 메소드로 서비스 객체를 활용할 것인가가 명시되고, 서비스 객체는 실제로 구현할 프로그램의 구조가 생성된다. 따라서 컴포넌트 개발자는 이를 이용하여 서비스 객체를 구현하게 되며 이 객체를 사용하는 다른 객체들은 이 서비스 객체의 내부 구조와는 무관하게 그 기능들을 활용할 수 있게 된다.

현재 CORBA는 주로 엔터프라이즈 환경에서 주로 이용되며 COM/DCOM은 윈도우 기반의 데스크탑 환경에서 주로 이용되고 있다.

2.2 GIS에서의 컴포넌트 개발 현황

컴포넌트 기술이 가장 성공적으로 응용되고 있는 분야 중 대표적인 것이 GIS이다. 1994년 OGC(Open GIS Consortium)에서는 이미 컴포넌트 기반의 개방형 GIS 시스템 구축을 위한 OpenGIS를 개발하였다[15]. OpenGIS는 이질적인 환경에서도 상호 운용될 수 있는 GIS를 구축하기 위해 개발한 표준 사양으로, 공간 데이터와 공간 연산에 대한 분산 접근을 위한 소프트웨어 기본 사양이 포괄적으로 정의되어 있다. OpenGIS는 추상 명세와 구현 명세로 나누어지는데 추상 명세는 GIS를 구축하기 위한 개념적인 모델을 제시한 것이고, 구현 명세는 추상 명세를 기반으로 실제 시스템에서 개발할 수 있도록 실제 컴포넌트 모델에 적용한 설계 사양이다.

현재 국내에서도 개방형 GIS 컴포넌트에 대한 소프트웨어 개발이 상당히 진행된 상태이다. 특히 ETRI에서 주도하고 있는 개방형 GIS 컴포넌트 소프트웨어 개발 연구에서는 OGC에서 제정한 컴포넌트 표준을 기반으로 데이터 제공 컴포넌트, 핵심 공통 컴포넌트, 응용 컴포넌트로 구성된 기본 아키텍처들이 개발되고 있다[14].

우선 데이터 제공 컴포넌트는 다양한 GIS 응용 시스템들이 기본적으로 요구하는 기능들을 표준화된 인터페이스를 통해 지원하는 컴포넌트로서 공간 데이터 관리 기능, 공간 검색 기능 그리고 공간 연산자 처리 등을 제공하는데, 이와 같은 기능은 대부분 공간 데이터베이스에서 제공된다. 따라서 데이터 제공 컴포넌트는 다양한 공간 데이터베이스에 제공되는 기능들을 래퍼(wrapper)형식으로 공통된 인터페이스를 통해서 접근할 수 있다. 데이터 제공 컴포넌트에 대한 구체적인 사양은 OGC의 OpenGIS를 준수하여 개발되고 있다. 핵심 공통 컴포넌트는 GIS의 고유 기능을 수행하는데 필요한 기능 위주로 구성된다. 구체적으로 공간 분석, 래스터 분석, 매핑 등의 기능들을 수행하는 컴포넌트들로 구성된다. 마지막으로 응용 컴포넌트는 GIS 기능과 MIS 기능을 통합하여 상수도, 토지, 도로와 같은 특정 응용 분야를 위한 컴포넌트로 구성된다. 이러한 응용 컴포넌트는 그 용도에 따라 다양하게 구축될 수 있고 추가될 수도 있다.

이와 같이 GIS 분야에서는 국내외적으로 상당히 많은 연구가 진행된 상태이며 지금도 OpenGIS를 중심으로 발전해나가고 있는 추세이다.

2.3 ATIS 구성

여행자정보보고급화 시스템은 기능에 따라 출발전 교

통정보안내 서브시스템(PTIS; Pre-trip Traffic Information Subsystem), 운전중 교통정보제공 서브시스템(ETIS; En-route Traffic Information Subsystem), 동적주행안내 서브시스템(DRGS; Dynamic Route Guide Subsystem) 등으로 구성된다[11, 20].

출발전 교통정보안내 서브시스템(PTIS)

VTIS(Value-added Traffic Information System) 응용영역에 속하는 서브시스템으로서 개인 단말장치나 공중단말장치를 사용하는 사용자에게 도로 교통정보는 물론 교통관련 편의시설, 여행관련 편의시설, 업무관련 편의시설 정보 등을 제공하고 사용자가 원하는 경우에 여행 기종점 정보를 입력받아서 여행경로를 도출하여 제공하는 서브시스템이며 대형 온라인 대화형 데이터베이스 시스템이 필요하다.

운전중 교통정보제공 서브시스템(ETIS)

교통정보를 수신할 수 있는 차량장치를 부착한 차량의 운전자에게 권역내 주요도로에 대한 교통정보, 돌발상황정보, 기상정보, 램프/차선제어정보 등을 제공하는 체계이다. 센터는 광역통신장치를 통해서 정보를 단방향으로 방송하는 것을 원칙으로 하지만, 여러 가지 장애물로 통신사각지대가 발생할 우려가 큰 지역에서는 노변통신장치를 통해서 협역방송을 할 수도 있으며, 대개의 경우 두 가지 방식을 상호 보완적으로 활용하는 것이 효과적이다. 제공되는 서비스는 유료이기 때문에, 과금방식이 중요하며 이에 따라 아키텍처 구상이 달라질 수도 있다.

동적주행안내 서브시스템(DRGS)

차량단말기를 부착한 차량에 동적교통정보를 제공하면 차량단말기는 수신된 동적교통정보를 토대로 운전자의 여행 기종점에 적합한 최적 여행경로를 도출하고, GPS 위성을 통해서 차량의 현재위치를 파악하면서 주행안내 서비스를 제공하는 서브시스템이다. 즉 센터가 무선통신망을 통해서 대량의 데이터를 방송하고 차량장치는 이를 받아 필요정보를 추출하여 동적최적경로를 도출해 내며, 차량의 위치에 따라 지속적으로 운전자의 주행을 안내하는 체계가 된다.

3. ATIS 컴포넌트 구성

3.1 컴포넌트 구성

각 서브시스템은 센터, 여행자 장치, 차량장치, 도로 장치 등의 네가지 물리적 구성요소로 구축되며 정

보수집, 정보 처리/관리, 정보 표현/제공 등 3 단계의 컴포넌트가 구성될 필요가 있다. 정보 수집 단계 컴포넌트는 차량장치와 도로장치에 구성되며 정보 처리/관리 단계 컴포넌트는 대부분 센터형 구성요소에 적용된다. 그리고 정보 표현/제공 단계 컴포넌트는 차량장치, 도로장치 그리고 여행자 장치에 구성될 수 있다. 물리적 특성과 기능에 따라 차량장치와 도로장치는 정보수집 단계 컴포넌트와 정보 표현/제공 단계 컴포넌트가 일부 또는 전부가 구성될 수 있다. 다양한 기능과 특성에 따라 상이한 컴포넌트가 필요하므로 차량장치와 도로장치 그리고 여행자 장치에 필요한 정보 수집 단계와 정보 표현/제공 단계의 컴포넌트 구성에 대한 구체적인 기술은 본 논문에서 생략한다.

정보 처리/관리 단계 컴포넌트는 사실상 각 서비스 시스템의 센터에 구성된다. 현재 ITS 아키텍처에는 60여개의 서비스시스템이 구성되어 있으며 ITS 서비스를 제공하기 위한 중심 서버 시스템인 센터들이 40여개로 정의되어 있다[11, 16]. 즉 실시간 혹은 준실시간 정보를 수집할 수 있는 각종 정보 수집원과 교통정보 제공 또는 교통류 제어 시스템과 연결하여 정보를 수집 및 가공, 분석하는 과정을 거쳐 교통상황에 적합한 전략을 수립하는 중앙운영체제가 서버 내에 구성된다. 그리고 센터는 ITS의 여러 서비스시스템에서 도출되는 각종 정보를 연계하여 정보의 효율적 이용 및 관리를 도모하는 역할과 함께 센터의 고유 기능을 수행하기 위한 시스템 관리 및 시설물 관리 기능을 수행하게 된다[11, 17].

출발전교통정보안내 서비스시스템의 여행자정보센터 그리고 운전중 교통정보제공 서비스시스템과 동적주행안 내 서비스시스템의 부가정보센터가 서버로서의 중심 기능을 수행하게 된다. 그리고 두 센터는 권역교통정보센터와 연계하여 도로, 교통, 돌발상황, 기상, 차로제어, 램프제어, 대체교통수단, 대중교통, 중차량관리, 요금징수, 교통단속 등의 정보를 수신하게 된다. ITS 서비스를 위한 센터형 구성요소의 기능은 공통적으로 정보수집/관리, 전략도출, 정보제공, 시설물관리, 정보연계, 연계제어 등 6개의 구축단위로 구분하여 정의할 수 있으며 이 단위를 센터내의 컴포넌트로 구성할 수 있다. 본 논문에서는 전략도출 컴포넌트를 여행자정보센터의 기능과 특성을 고려하여 정보도출 컴포넌트로 표현하며 별도로 구성될 수 있는 정보연계와 연계제어 컴포넌트를 하나의 컴포넌트로 구성하며 이를 정보연계 컴포넌트로 기술한다. 여행자정보센터와 부가정보센터의 특성상 교통정보를 다른 센터와 연계하여 교통 시설물을 제어하는 기능보다는 교통 및 관련 정보를

다른 장치나 센터에 전달하는 기능이 더욱 중요하기 때문에 연계제어 컴포넌트를 별도로 구성하는 것을 배제한다.

3.2 여행자정보센터 컴포넌트

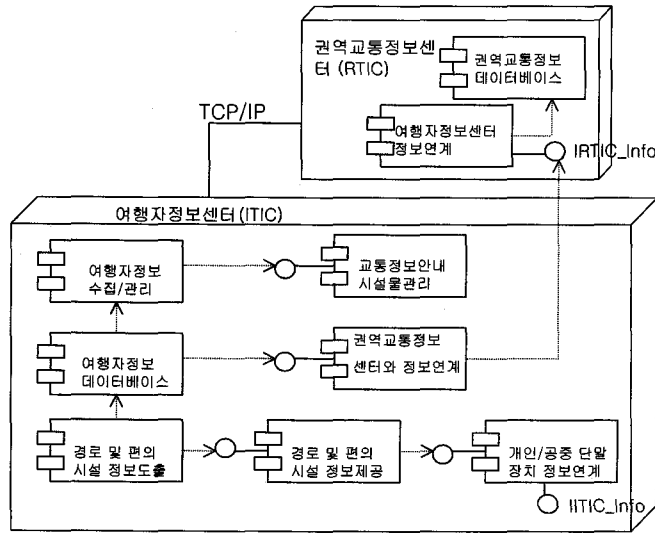
UML 기법중 배치(deployment) 다이어그램 방식으로 작성된 여행자정보센터(ITIC)의 컴포넌트 구성은 그림 1과 같다. 여행자정보센터는 권역교통정보센터(RTIC)와 물리적으로 다른 장소에 위치하기 때문에 노드로 표현되며 두 노드는 TCP/IP 통신 프로토콜로 연결되는 상황을 가정하였다. 그리고 여행자정보센터의 정보연계 컴포넌트와 권역교통정보센터의 정보연계 컴포넌트는 IRTIC_Info 인터페이스로 연결된다. 출발전교통정보제공 안내 서비스시스템의 물리적 구성요소인 개인단말장치와 공동단말장치는 여행자정보센터와 TCP/IP 방식으로 연결되며 정보연계 컴포넌트 간의 인터페이스는 IITIC_Info로 표현된다(그림 2 참조). 그림 1과 2에서 노드내의 컴포넌트들은 종속(dependency) 관계로 표현되며 특히 그림 1의 권역교통정보센터는 여행자정보고급화 서비스시스템과 직접 관련이 있는 컴포넌트들만 기술되었다.

여행자정보 수집/관리 컴포넌트

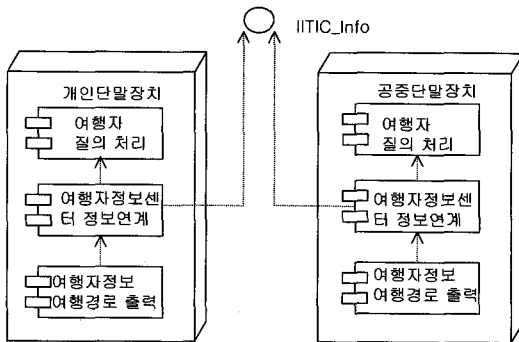
이 컴포넌트는 기타 정보제공자로부터 정보를 수집하는 컴포넌트로 세부적으로 교통, 여행, 업무 관련 편의시설에 대한 정보를 수집/관리하는 클래스들로 구성된다(그림 3 참조). Traffic_Facility 클래스, Travel_Facility, Business_Facility 클래스는 각각 교통, 여행, 업무 관련 편의 시설에 대한 정보를 삽입, 갱신하는 연산(operation)들로 구성되며 데이터베이스를 구축하는 DBConnect 클래스와는 포함 관계로 표현된다. 그리고 해당 서비스시스템의 시설물 관리 컴포넌트와 내부 인터페이스를 통하여 정보를 수집하며 수집된 정보는 여행자정보센터내의 데이터베이스에 구축된다.

시설물관리 컴포넌트

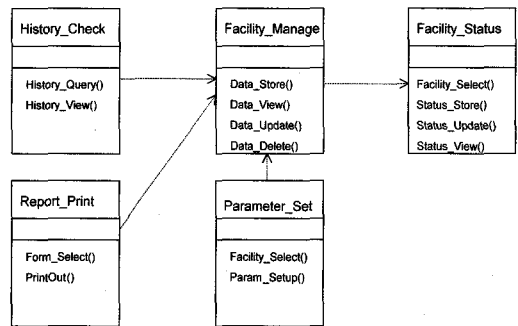
그림 4에서 시설물관리 컴포넌트는 시설물 상태 조회 및 갱신 기능을 수행하는 Facility_Status 클래스와 정보 변경사항을 수정하는 Facility_Manage 클래스 그리고 시설물 설치이력을 조회하는 History_Check 클래스를 포함한다. History_Check 클래스는 Facility_Manage 클래스에 종속되며 Facility_Manage 클래스는 다시 Facility_Status 클래스에 종속되는 관계가 유지된다.



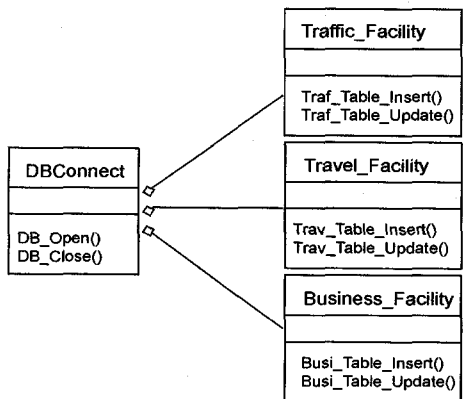
〈그림 1〉 여행자정보센터 컴포넌트



〈그림 2〉 개인단말장치와 공중단말장치 컴포넌트 구성



〈그림 4〉 시설물관리 컴포넌트의 클래스 구성



〈그림 3〉 여행자 정보 수집/관리 컴포넌트의 클래스 구성

이외에 선택된 형식에 따라 보고서를 출력하는 Report_Print 클래스와 제어변수를 설정하는 Parameter_Set 클래스가 구성되며 이들은 Facility_Manage 클래스와 종속 관계가 유지된다.

여행경로 및 편의시설 정보도출 컴포넌트

여행경로를 도출하는 Travel_Path 클래스와 교통, 여행, 업무 편의시설 등에 대한 정보를 도출하는 Info_Manage 클래스가 구성되며 두 클래스 간에 정보도출 관계인 Info_Derive 연관 관계가 설정된다(그림 5 참조).

여행경로 및 편의시설 정보제공 컴포넌트

그림 5의 정보도출 컴포넌트에서 생성된 여행경로 정보와 편의시설 정보를 공중단말장치와 개인단말장치

에 각각 제공하기 위해서 Pub_Term_Information 클래스와 Per_Term_Information 클래스가 구성된다 (그림 6 참조). 그리고 이 두 클래스 사이에는 정보제공을 위한 Info_Provide 연관관계가 설정된다.

개인단말장치 및 공중단말장치와 정보연계 컴포넌트

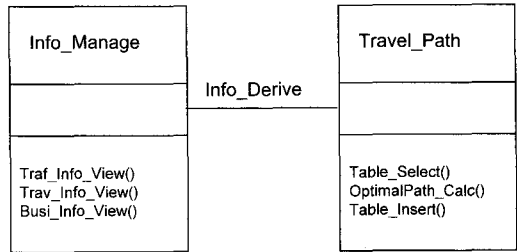
여행자정보센터에서 가장 중요한 역할을 하는 컴포넌트이며 공중단말장치와 개인단말장치와 정보 연계를 위한 Pub_Term_Manage 클래스와 Per_Term_Manage 클래스가 구성된다(그림 7 참조). 두 클래스는 Info_Manage 연관관계가 유지된다. 실제로 공중단말장치와 개인단말장치는 여행자정보센터의 정보연계 컴포넌트가 제공하는 IITIC_Info 외부 인터페이스를 통하여 편의시설 정보와 여행경로 정보를 제공받게 된다.

권역교통정보센터와 정보연계 컴포넌트

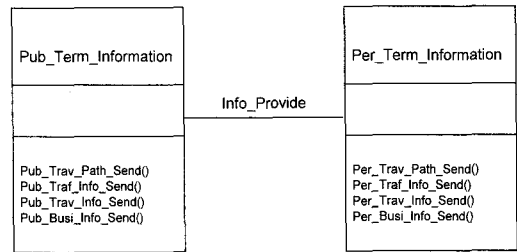
여행자정보센터는 권역교통정보센터로부터 IRTIC_Info 외부 인터페이스를 통하여 도로, 교통, 돌발상황, 기상, 차로제어, 램프제어, 대체교통수단, 대중교통, 주차량관리, 요금징수, 교통단속 등의 정보를 제공받으며 이 인터페이스에 대한 구체적 내용은 다음 절에서 기술된다. 여행자정보센터내에 구성되는 '권역교통정보센터와 정보연계를 위한 컴포넌트'와 권역교통정보센터에 구성되는 '여행자정보센터와 정보연계를 위한 컴포넌트'는 동일한 기능을 수행한다. 단지 차이점은 권역교통정보센터에서는 각종 교통 정보를 생성하는 반면 여행자정보센터에서는 생성된 정보를 수집만 하며 다른 센터로 보내기 위하여 별도의 정보를 생성하는 기능은 없는 것으로 국가 ITS 아키텍처에 정의되어 있다. 또한 그림 8의 정보연계 컴포넌트는 부가정보센터에서도 동일하게 사용된다. 그림 8에서 Info_Create 클래스는 정보를 생성하거나 수집한 정보를 저장하는 기능이 수행되며 Info_Transfer 클래스는 실제 다른 센터와 정보를 송수신하는 기능이 수행된다. 이 두 클래스는 종속 관계가 설정된다. Comm_Setup 클래스는 센터 간의 통신 환경을 설정하며 Interface_Control 클래스는 정보 교환을 위하여 센터별로 인터페이스를 제공하는 기능을 수행한다.

여행자정보센터의 '권역교통 정보센터와 정보 연계' 컴포넌트와 '시설물관리' 컴포넌트는 실제로 부가정보센터와 권역교통정보센터에서도 공통으로 구성되는 컴포넌트들이다. 본 논문에서는 여행자정보센터를 구성하는 컴포넌트들 간의 내부 인터페이스에 대한 기술과 부가정보센터를 구성하는 컴포넌트들에 대한 구체적 기술을 논문 지면상 생략한다. 또한 권역교통정보센터의 컴포넌트들은 여행자정보고급화 시스템과 직접 연

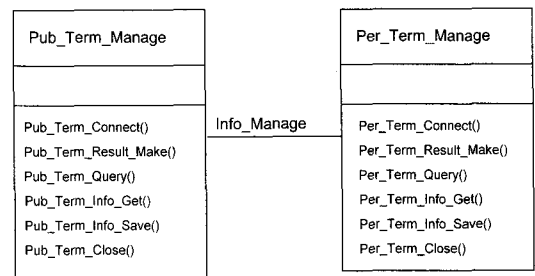
관된 정보연계 컴포넌트만이 기술되었다.



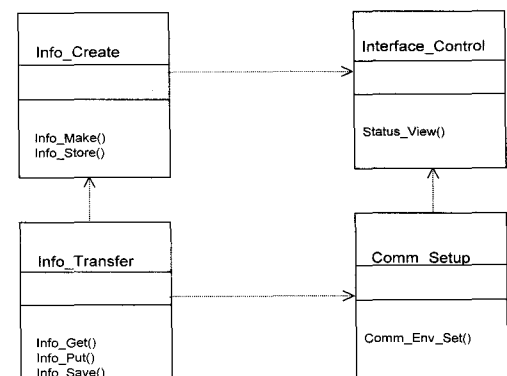
〈그림 5〉 여행경로 및 편의시설 정보도출 컴포넌트의 클래스 구성



〈그림 6〉 여행경로 및 편의시설 정보제공 컴포넌트의 클래스 구성



〈그림 7〉 개인/공중단말장치와 정보연계 컴포넌트의 클래스 구성



〈그림 8〉 권역교통정보센터와 정보연계 컴포넌트의 클래스 구성

〈표 15〉 권역정보센터에서 제공하는 정보흐름과 관련 메시지 형식

정보흐름명	참조 메시지	세부 컴포넌트명
도로	교통정보 교환을 위한 정보형식	NetworkEvent
교통정보	교통정보 교환을 위한 정보형식	TrafficInformation
돌발상황정보	돌발상황 관리를 위한 정보 형식	IncidentInformation
기상정보	여행정보제공을 위한 정보형식	WeatherInformation
환경정보	여행정보제공을 위한 정보형식	EnvironmentInformatron
대체교통수단정보	정의안됨	없음
대중교통정보	정의안됨	없음
차로	돌발상황 관리를 위한 정보 형식	StreetStatus
램프제어정보	교통제어를 위한 정보 형식	DeviceStatus
교통단속정보	자동 교통 단속을 위한 정보 형식	TrafficEnforcement
중차량관리정보	정의안됨	없음
요금징수원칙정보	자동 요금 징수를 위한 정보형식	TollPolicy

4. 센터간 연계를 위한 컴포넌트 인터페이스의 상세 설계

2절에서 언급한 바와 같이 컴포넌트가 제공하는 서비스는 인터페이스로 표현된다. 따라서 3절에서 도출한 각 컴포넌트에 대한 인터페이스의 설계는 컴포넌트에 기반한 ITS 아키텍처의 설계에 가장 중요한 역할을 한다. 또한 인터페이스를 정의하는 방법에 따라 각 센터의 컴포넌트에 대한 재사용 가능성 여부가 결정된다. ITS 국가 아키텍처에서는 각 센터간에 교환되는 정보의 종류 및 특성을 명시하였고 국토연구원에서 이루어진 후속 연구에서 ASN.1 형식으로 각 정보의 구체적인 메시지 형식이 정의되었다(6, 10). 따라서 센터간의 연계를 위한 컴포넌트의 인터페이스는 이 메시지들을 활용하여 정의할 수 있다. 지금까지 정의된 메시지 형식으로는 여행정보제공을 위한 정보형식, 교통정보 교환을 위한 정보형식, 교통제어를 위한 정보 형식, 돌발상황 관리를 위한 정보 형식, 자동 교통 단속을 위한 정보 형식, 자동 요금 징수를 위한 정보형식 등이다. 그리고 현재 동적 주행 안내를 위한 정보 형식이 개발 중에 있다(6, 10).

본 절에서는 그림 1에서 권역정보센터의 컴포넌트 중에서 여행자 정보센터와의 연계 컴포넌트에 대한 인터페이스의 설계 결과를 제시한다. 우선 ITS 국가 아키텍처에서의 정의를 보면 권역정보센터에서 여행자정

보센터로 제공하는 정보는 표 1과 같다.

ITS 국가 아키텍처에서는 이들 대해서 세부적인 메시지 형식을 정의하고 있으며 각각은 독립적으로 운용된다. 또한 이들 메시지들은 ATIS에 관련되지 않은 기타 센터에서도 수정 없이 공통적으로 사용될 수 있도록 정의되었다. 따라서 본 연구에서는 컴포넌트의 재사용을 극대화하기 위해서 각각을 독립적인 세부 컴포넌트로 정의하였다. 단, 이 중에서 ITS 국가 아키텍처에서 메시지 정보형식에서 정의되어 있지 않은 대체교통수단정보, 대중교통정보, 중차량관리정보는 설계 대상에서 제외되었다.

본 논문에서는 표 1에서 정의된 컴포넌트 중에서 돌발상황정보의 설계 과정과 결과를 예로 제시한다. 우선 돌발정보를 표현하기 위한 메시지 형식은 그림 9와 같다.

이 그림에서 보는바와 같이 메시지 형식은 크게 IncidentIdentity, IncidentLocation, IncidentDescription, IncidentTimeline로 구성되며 각각은 다음과 같은 정보를 담고있다.

- IncidentIdentity: 특정한 교통 네트워크에서 현재 진행중인 돌발상황을 식별하는 데이터 항목들의 집합
- IncidentLocation: 현재까지 진행중인 모든 돌발상황의 위치
- IncidentDescription: 현재 진행중인 모든 돌발상황에 대한 설명

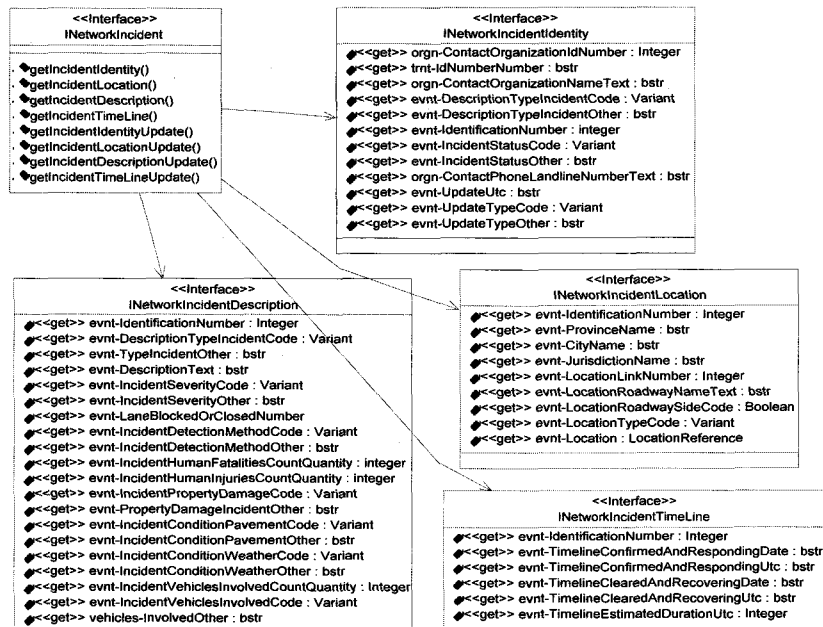
<그림 9> ASN.1 형식의 돌발정보 메시지 형식 (6)

<pre> INCIDENTIDENTITY DEFINITIONS AUTOMATIC TAGS ::= BEGIN IncidentIdentity ::= SEQUENCE { orgn-ContactOrganizationIdNumber Integer (SIZE(0..4294967295)), tmt-IdNumberNumber UTF8String (SIZE(1..256)), orgn-ContactOrganizationNameText UTF8String (SIZE(1..128)), evt-DescriptionTypeIncidentCode Enumerated evt-DescriptionTypeIncidentOther UTF8String (SIZE(1..256)) evt-IdentificationNumber Integer evt-IncidentStatusCode ENUMERATED evt-IncidentStatusOther UTF8String (SIZE(1..256)) orgn-ContactPhoneLandlineNumberText UTF8String(SIZE(1..32)), evt-UpdateUtc UTF8String (HHMMSS.ssss), evt-UpdateTypeCode ENUMERATED evt-UpdateTypeOther UTF8String (SIZE(1..256)) } END INCIDENTLOCATION DEFINITIONS AUTOMATIC TAGS ::= BEGIN IMPORTS LocationReference FROM GLOBALS; IncidentLocation ::= SEQUENCE { evt-IdentificationNumber Integer (0..4294967295) , evt-ProvinceName UTF8String (SIZE(1..128)), evt-CityName UTF8String (SIZE(1..128)), evt-JurisdictionName UTF8String (SIZE(1..128)), evt-LocationLinkNumber Integer (0..4294967295), evt-LocationRoadwayNameText UTF8String (SIZE(1..64)), evt-LocationRoadwaySideCode Boolean evt-LocationTypeCode ENUMERATED evt-Location LocationReference } END INCIDENTDESCRIPTION DEFINITIONS AUTOMATIC TAGS ::= BEGIN </pre>	<pre> IncidentDescription ::= SEQUENCE { evt-IdentificationNumber Integer (0..4294967295), evt-DescriptionTypeIncidentCode Enumerated evt-TypeIncidentOther UTF8String (SIZE(1..256)) evt-DescriptionText UTF8String (SIZE(1..256)), evt-IncidentSeverityCode ENUMERATED evt-IncidentSeverityOther UTF8String (SIZE(1..256)) evt-LaneBlockedOrClosedNumberInteger (0..255), vnt-IncidentDetectionMethodCode BIT STRING evt-IncidentDetectionMethodOther UTF8String (SIZE(1..256)) evt-IncidentHumanFatalitiesCountQuantity INTEGER (0..255), evt-IncidentHumanInjuriesCountQuantity INTEGER (0..255), evt-IncidentPropertyDamageCode BIT STRING evt-PropertyDamageIncidentOther UTF8String evt-IncidentConditionPavementCode BIT STRING evt-IncidentConditionPavementOther UTF8String (SIZE(1..256)) evt-IncidentConditionWeatherCode BIT STRING evt-IncidentConditionWeatherOther UTF8String evt-IncidentVehiclesInvolvedCountQuantity INTEGER (0..255), evt-IncidentVehiclesInvolvedCode BIT STRING vehicles-InvolvedOther UTF8String (SIZE(1..256)) } END INCIDENTTIMELINE DEFINITIONS AUTOMATIC TAGS ::= BEGIN IncidentTimeline ::= SEQUENCE { evt-IdentificationNumber Integer (0..4294967295), evt-TimelineConfirmedAndRespondingDate UTF8String evt-TimelineConfirmedAndRespondingUtc UTF8String evt-TimelineClearedAndRecoveringDate UTF8String evt-TimelineClearedAndRecoveringUtc UTF8String evt-TimelineEstimatedDurationUtc INTEGER } END </pre>
---	---

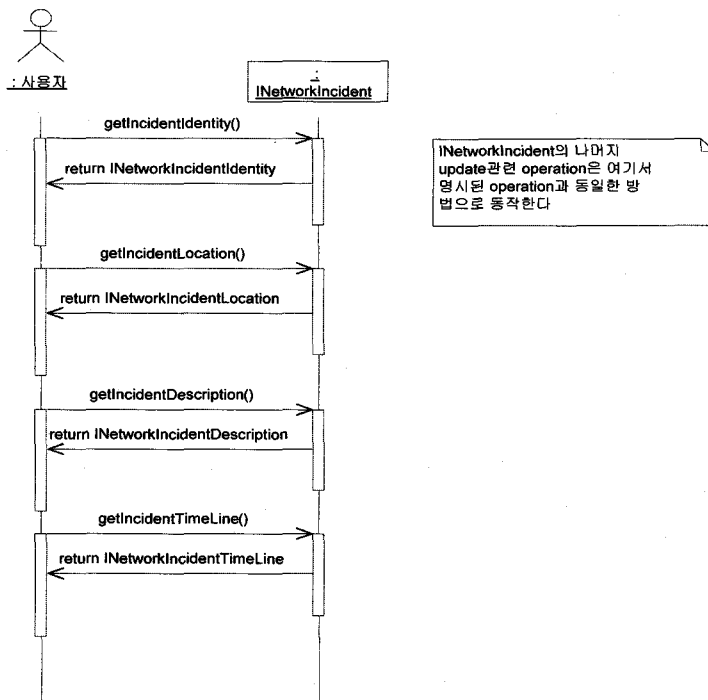
- IncidentTimeline: 현재 진행중인 모든 돌발상황들의 시간

이 중에서 IncidentIdentity는 돌발상황의 발생 여부와 상황정보 그리고 특정 상황을 나타내는 식별자를 포함하고 있다. 따라서 나머지 메시지들은 IncidentIdentity에서 제공되는 식별자를 이용해서 각 돌발상황들을 구별한다. 이러한 메시지 정의로부터 정의한 컴포넌트인 IncidentInformation의 구체적인 인터페이스는 그림 10과 같다. 이 그림에서 INetworkIncident 객체를 제외한 나머지 객체는 위의 돌발상황에 대한 4가지 정보와 일대일로 매핑되며 별도의 메소드를 갖고 있지 않다. 그리고 INetworkEvent 객체는 각 객체의 생성을 요청하

는 메소드들로 구성되어 있다. 이 객체에서 갱신에 관련된 메소드들은 돌발상황에 변경이 이루어졌을 때 그 상황을 요청하는 역할을 한다. 이 컴포넌트의 인터페이스에 대한 순차 다이어그램(sequence diagram)은 그림 11과 같다. 이 그림에서 보는 바와 같이 사용자는 우선 INetworkIncident 객체를 생성한 후 이 객체에서 제공하는 getIncidentIdentity 메소드를 이용하여 특정 돌발상황의 식별에 필요한 정보를 얻는다. 이 후에는 getIncidentLocation, getIncidentTimeLine, getIncidentDescription 등의 메소드를 이용하여 돌발상황에 대한 자세한 정보를 담은 객체를 얻게 된다.



〈그림 10〉 돌발상황에 대한 정보를 제공하는 IncidentInformation 컴포넌트의 인터페이스 구성



〈그림 11〉 IncidentInformation 컴포넌트의 활용방법을 표현한 순차 다이어그램

5. 결론

본 논문에서는 ITS 서비스 중 여행자 정보 고급화 서비스를 컴포넌트로 설계하는 방안이 제시되었다. ITS 국가 아키텍처에서 제안된 각 센터의 구성요소를 분석하여 여행자 정보 고급화 서비스에 필요한 컴포넌트 구성요소를 도출하는 방안과 함께 실제로 컴포넌트를 도출한 결과가 본 논문에서 기술되었다. 컴포넌트의 도출은 재사용에 중점을 맞추어 각 센터들이 이를 이용하여 쉽게 조립하여 구축할 수 있는 방향으로 설계되었고, 이를 위해 도출된 컴포넌트의 개방형 인터페이스가 제안되었다. 비록 본 논문에서 제안된 컴포넌트 설계 결과는 ATIS 분야에만 국한된 것이지만 국가 ITS에서 제안한 모든 구성요소들로 확장될 수 있다.

향후에는 이 결과를 토대로 ATMS, CVO 등 ITS의 다른 분야들에 대해서도 컴포넌트에 기반한 아키텍처를 설계할 예정이며 향후 국가 ITS 구축 과정시 컴포넌트 개발에 대한 기대효과를 다음과 같이 기술한다. 첫째, ITS 구축비용 절감으로 합리적인 아키텍처 정립과 컴포넌트 구성은 실제 시스템 구축시 사업 주체와 지역간의 중복적인 설치요소를 지양하며 표준화된 기능을 구현할 수 있는 컴포넌트 설치로 구축비용을 절감할 수 있다. 둘째, 시스템 개발과 구축기간을 단축할 수 있기 때문에 적용 가능한 서비스와 기능을 중심으로 단계적인 개발이 가능하다. 셋째, 점진적 서비스 구현 또는 시스템 구축이 가능하다. 즉, 정의된 서브시스템의 서비스 및 기능은 현재의 교통여건과 기술수준 등을 고려한 것으로서 이들 중 우선적으로 개발되어야 할 시스템을 선정할 수 있다. 따라서, 실제 적용 가능한 기능을 중심으로 점진적인 서비스 구현 및 시스템 구축이 가능하다. 넷째, 정보/자료의 효율적 공유 또는 교환으로서 각 ITS 관련 주체가 개별적으로 수집하거나 보유하고 있는 정보 및 자료가 컴포넌트를 통하여 적재적소에 연결되며 효과적으로 교환될 수 있다. 다섯째, 시스템 확장에 대비할 수 있다. 관련 기술이 발달함에 따라 제공기능은 보다 다양해질 것이며, 설치 범위도 확장될 수 있다.

참고문헌

- [1] 정성원, "지능형 교통시스템(ITS) 표준화 연구", 한국정보과학회 정보과학회지, 제 16권 제6호, 1998년 6월, pp. 43-50.
- [2] 최종욱, "첨단교통정보 시스템(ATIS: Advanced

- Transportation Information System)", 한국정보과학회 정보과학회지, 제 16권 제6호, 1998년 6월, pp. 14-22.
- [3] Shibata, J. and French, R., "A comparison of intelligent transportation systems: progress around the world through 1996," ITS America, June 1997.
- [4] Euler, G., "National ITS Program Plan," vol. 1, 1st Ed., March 1995.
- [5] Vertis, "Vertis: ITS services," <http://www.vertis.or.jp>.
- [6] 국토연구원, '99국가 ITS 기술표준화 사업(2단계), 1999.
- [7] Guy Eddon and Henry Eddon, Inside COM, Microsoft Press, 1998.
- [8] Jacobson, I., Griss, M., and Jonsson, P., Software Reuse, Addison Wesley, 1997.
- [9] Rumbaugh, J. et al., Object-oriented Modeling and Design, Prentice-Hall, 1991.
- [10] 국토연구원, 국가 ITS 아키텍처 확립을 위한 연구(2단계), 1999.
- [11] 김경주, 조남규 역, UML Components, 인터뷰비전, 2001.
- [12] C. Szyperski, Component Software, Addison Wesley, 1998.
- [13] OMG, The Common Object Request Broker: Architecture and Specification, 1998.
- [14] 김민수, 김광수, 오병우, 이기원, "응용 시스템 구축을 위한 OLE/COM 기반의 GIS 데이터 제공자 컴포넌트 시스템에 관한 연구", The Journal of GIS Association of Korea, Vol. 7, No. 2, pp. 191-207, Oct 1999.
- [15] OpenGIS Consortium, Inc., The OpenGIS Guide, 1998.
- [16] S. Kubota, et al., "A study of data distributed methods in ITS service center", In Proc. of 8th World Congress on Intelligent Transport Systems, Sydney, October 2001.
- [17] S. W. Dellenback, "Deploying a flexible and extensible ITS architecture", In Proc. of 8th World Congress on Intelligent Transport Systems, Sydney, October 2001.

- [18] G. T. Heineman(Ed.), Component-Based Software Engineering, Addison-Wesley Pub Co (Sd), June, 2001.
- [19] Brown, Large-Scale Component-Based Development, Prentice-Hall, May, 2000.
- [20] 건설교통부 자료실, http://www.moct.go.kr/DH/mct_hpg/mctthpg_no/ITS/itsmain.html



김진환

1986년 서울대학교 컴퓨터공학과 졸업(학사)
1988년 서울대학교 대학원 컴퓨터공학과(석사)
1994년 서울대학교 대학원 컴퓨터공학과(박사)

1994년 - 1996년 서울대학교 컴퓨터신기술공동연구소 특별연구원

1995년 - 현재 한성대학교 정보전산학부 부교수

관심분야: 분산실시간 시스템, 멀티미디어 시스템, ITS 등



장재영

1992년 서울대학교 계산통계학과 졸업(학사)
1994년 서울대학교 계산통계학과 대학원 졸업(이학석사)
1999년 서울대학교 계산통계학과 대학원 졸업(이학박사)

2000년 3월 ~ 현재 한성대학교 정보전산학부 전임강사

관심분야: 질의처리, 데이터웨어하우스, 논리 데이터베이스



이봉규

1988년 연세대학교 졸업(학사)
1992년 Cornell University 졸업(석사)
1994년 Cornell University 졸업(박사)

1993년~1997년 Cornell University 조교수

1997년~현재 한성대학교 정보전산학부 부교수
관심분야: GIS, ITS, GPS, XML, GML 등