

객체지향 기반의 ITS 시스템 아키텍처 구축방안

ITS System Architecture based upon Object-oriented Methodology

Byoungjo Yoon^{a,*}

^a *Incheon National University, Incheon 406-772, Republic of Korea*

A B S T R A C T

The purpose of this study is to present in-vehicle dynamic route guidance systems based upon object-oriented ITS LSA. We expect to apply efficiently for connection and expansion of systems what is more to present systems based upon object-oriented of the other ITS LSA

본 연구는 객체지향 기반의 ITS 아키텍처를 구축하는데 있어서 객체를 생성하는 새로운 방법을 제시하고 있다. 우리는 객체 지향적 접근방법에서 나타나는 문제점을 찾고 이를 해결하기 위하여 작업의뢰기술도 라는 새로운 객체 생성방법을 제시하였다. 이를 토대로 차량 내 동적경로안내 시스템에 본 연구에서 제시된 방법을 적용하여 객체지향적 기반의 시스템 아키텍처를 제시하였다.

KEYWORDS

ITS System,
Architecture
Object-oriented
Responsibility
Assignment
Diagram
UML

ITS시스템
객체지향적 방법
동적경로안내시스템
아키텍처
작업의뢰기술도

© 2016 Korea Society of disaster Information All rights reserved

* Corresponding author. Tel. 82-32-835-8778. Email. bjyoon63@inu.ac.kr

ARTICLE HISTORY

Recieved Mar. 17, 2016
Revised Mar. 23, 2016
Accepted Mar. , 2016

1. 서론

1.1 연구의 배경 및 목적

우리나라에서 ITS라는 분야를 도입한지 10여년이 흘렀다. 그동안 많은 발전을 통해 많은 곳에서 ITS 서비스가 적용되고 있는 이 시점에서 과연 ITS의 진정한 의미를 가진 시스템이 구축되어 있는지를 보면 그렇게 긍정적인 면을 찾아보기 쉽지 않다. 그 이유는 현재 우리나라에서 구축되어 있는 ITS 시스템을 보면 단일 시스템 중심으로 이루어져 있다. 첨단 모델도시의 도시간 연계 시스템 미구축, 버스정보시스템과 같은 독립적인 정보서비스 이러한 예들은 ITS라는 시스템의 특징을 대변하지 못하는 결과라고 할 수 있다. ITS는 교통이라는 분야에 정보기술을 적용하는 것이다. 이 말은 교통이라는 한 장소에서 다른 장소로 옮겨지는 것이라는 단순한 정의에 입각하여 볼 때 교통이라는 정보를 공간적 제약 없이 이용할 수 있느냐는 것이다. ITS는 시스템을 기반으로 이루어져 있기 때문에 시스템의 특성을 반영할 수 있는 것이 필요하다. 이것의 기반이 되는 것이 시스템 아키텍처이다. ITS시스템 아키텍처는 수많은 시스템들로 구성되어 있다. 각각의 시스템들은 시스템간에서 서로 얽혀있는 동시에 다양한 구성요소들의 집합으로 이루어져 있기 때문에 이러한 ITS시스템의 특성을 하나로 통합하는 것이 필요하다.

ITS시스템 아키텍처를 개발하는데 있어서 시스템의 특성인 사용자의 요구와 시스템의 운영환경의 변화에 대처할 수 있는 시기적 적절성, 과도한 비용의 추가 없이 시스템의 확장, 수정 그리고 적용이 가능한 시스템의 유지성을 제공하는 아키텍처가 되어야한다. 왜냐하면 ITS시스템 아키텍처는 시스템의 주요 기능을 현실화하는 각각의 하위 시스템으로 시스템이 분할되어 있고, 시스템

전체의 기능을 지원하는 각각의 하위시스템으로 세분화된 특수한 기능이 있고, 그리고 시스템의 기능을 어떻게 하위 시스템이 지원하고 전달하는지에 대한 내용을 가지고 있기 때문이다. 그러면 ITS시스템 아키텍처는 아키텍처의 특성을 어떻게 이용하여 개발할 것인가에 대한 방법을 제시해야 할 필요가 있다.

ITS시스템 아키텍처는 알고리즘하나만을 가지고 개발될 수 없다. 과거의 시스템은 알고리즘 중심의 개발이었고 계산 속도를 줄이는데 초점을 맞추었다. 하지만 시스템 아키텍처에서 알고리즘은 시스템의 하부에 들어가는 작은 구성요소 밖에 되지 않는다. 따라서 ITS시스템 아키텍처를 개발하는데 있어서 새로운 방법이 필요하다.

ISO 산하 TC204 WG에서는 ITS시스템 아키텍처의 개발 기반은 객체지향 접근방법이 되어야 한다고 하였다. 하지만, ITS 시스템 접근방법에 있어서 명확한 방법론을 제시하지 못하고 있는 상황에서 새로운 패러다임을 제시하는 것이 중요하다고 생각한다.

따라서, 본 연구는 차량내 동적경로안내시스템을 대상으로 객체지향적 방법을 적용한 ITS 시스템 아키텍처를 구축하는 방안을 제시하는데 있다. 본 연구를 통하여 제시된 시스템 아키텍처 구축 방안으로 다른 시스템의 아키텍처 구축에 기반을 마련할 뿐만 아니라, 대상 시스템의 연계 및 확장에 효율적으로 이용할 수 있을 것이라 기대한다.

1.2 연구의 내용 및 방법

본 연구에서는 객체지향적 방법을 적용하여 차량내 동적경로안내시스템을 위한 아키텍처를 구축하는 방안을 제시하였다. 본 연구의 배경을 살펴보기 위하여 현재 ITS시스템 아키텍처를 구축하고 있는 나라들의 ITS시스템 아키텍처 구축 현황을 통해 ITS시스템 아키텍처의 진행사항 및 특성을 파악하였다. ITS시스템 아키텍처를 객체지향적인 기반으로 구축하기 위하여 객체지향이 가지고 있는 특성별로 연구를 진행하였다.

우선 ITS시스템 아키텍처 중 차량내 동적경로안내시스템의 특징을 분석하였다. 이러한 분석을 토대로, 연구는 차량내 동적경로안내시스템의 사용자요구조건 및 시스템요구조건을 기술하였다. 사용자요구조건 및 시스템요구조건은 모델로 표현하기 위하여 4가지의 모델을 적용하였다. ITS시스템 아키텍처를 구축하기 위하여 객체지향적 언어인 UML(Unified Modeling Language)[Gradym Booch and Jacobson, 1999]⁴⁾을 이용하여 차량내 동적경로안내시스템의

4) UML은 Object Management Group(OMG)에 의해서 객체지향모델링을 위한 표준이 되었다.

논리아키텍처를 표현하였다.

본 연구의 과정은 다음과 같다. 첫째, 시스템의 상위수준을 분석하였으며 상위수준은 그 아래 단계인 하위수준으로 세분화하여 시스템을 구축하였다. 각각의 상위수준과 하위수준은 객체지향적 방법의 특성인 클래스와 객체를 만들기 위한 과정과 연관되어 있다. 본 연구는 시스템을 객체지향적으로 구축하기 위하여 객체계층모형을 구축하였다. 객체계층모형에서 만들어진 클래스와 객체는 동적상태 모델과 정적상태의 모델로 표현하였으며, 이를 통하여 객체지향적 ITS시스템 아키텍처를 구축하였다. 마지막으로 구축된 아키텍처를 분석·검증하는 과정으로 Chidamber와 Kemerer가 제시한 평가방법⁵⁾을 통하여 검증결과를 제시하였다. 마지막으로 제 5장은 결론을 통해 본 연구의 내용을 정리하고 향후 연구과제를 제시함으로써 본 연구를 마무리하였다.

2. 문헌고찰

2.1 ITS 시스템 아키텍처

건축물의 설계도를 제작하는 건축계획과 유사한 개념으로 ITS아키텍처는 통합된 지능형교통시스템의 청사진을 제공한다. 지능형교통시스템이라는 “정보건축물”을 세우기 위해 건축물의 입지에서부터 시설규모, 건축방법 등 전체적인 측면과 건축물을 구성하는 각종 부속시설들의 기능, 위치, 접속방법 등을 제시한다. 그러나 설계도가 포함된 건축계획과 달리 아키텍처는 시스템 설계 이전의 단계라 할 수 있다. 건축물의 상세설계도면이 제시되는 것이 아니라, 건축물이 목표로 하는 시스템기능을 달성하기 위해서 개념설계에 해당되는 건축물의 전체 골격을 구성하는 것이다.

시스템 아키텍처는 건축계획과 달리 구축대상이 개별 시설물 또는 사업이 아닌 시스템전체가 대상이 된다. 정보인프라를 다루는 시스템의 아키텍처는 개별 건축물들간의 연관관계와 달리 필연적으로 각 시스템의 시스템과 구성요소간의 상호연관관계가 복잡하게 얽혀 있어, 개별사업 차원이 아닌 국가 전체적인 시스템차원에서 수립되는 것이 바람직하다. 일반적으로 시스템 아키텍처란 전자·정보·통신분야의 대규모 시스템 개발과정에서 도출된 개념으로써, 시스템의 목적과 이용자 요구사항을 기초로 하는 전체적인 시스템 골격(framework)을 의미한다. 각종 대규모 시스템들은 시스템을 구성하는 다양한 기술이 여러 가지 형태로 결합되어 성립하게 된다. 다양한 기술로 구성되는 대규모 시스템을 구축할 경우에는 먼저 시스템전체의 구조를 사전에 정리하고, 그것에 준거하는 형태로 개별적으로 시스템을 개발하는 것이 일반적이다. 즉, 아키텍처란 시스템이 반드시 갖추어야 할 기능적, 비기능적 사항들을 정의하는 수단으로서, 시스템의 시스템 및 기능을 정의하고, 시스템/기능영역의 경계와 참여주체를 정의 표현한다.

따라서, ITS 시스템 아키텍처는 ITS에서 목표로 하는 각종 사용자시스템을 통합적으로 구현하기 위한 틀로서 ITS구축사업의 효과를 극대화하고, ITS의 구축방향을 미래지향적으로 제시하는 작업이다. 다양한 사용자시스템 구현을 위해서는 세부적인 구축시스템이 필요하며, 아키텍처에서는 이 시스템의 기능적/비기능적 사항과 물리적 구성장치, 정보흐름 등을 정의하게 된다.

시스템 아키텍처는 1) 참조아키텍처, 2) 논리아키텍처, 3) 물리아키텍처로 구성된다. 이들은 독립적인 특성을 가지고 있는 반면에 3가지의 구성이 연계되어 하나의 시스템을 형성하는 복합적 구성요소를 가지고 있다. 하나의 시스템을 이루는데 있어서는 이들 3가지가 체계적으로 구축되어 있어야 한다.

5) Chidamber and C. F. Kemerer, " A metrics suite for object-oriented design," IEEE Trans. on Software Engineering, Vol. 20, No. 6

2.2 연구사례

Koutarou Katou⁶⁾는 과거의 절차적 방법의 단점인 시스템의 확장성 및 재사용성의 단점을 보완할 수 있는 객체지향적 방법에 대하여 연구하였다.

Yukihiro Matsunaga와 Tomoji Kishi⁷⁾는 다양한 기술과 분야로 이루어진 ITS는 시스템아키텍처를 개발하는데 있어서 장기적인 관점으로 계획되어야 한다고 하였다. 이러한 관점에서, ITS 시스템 아키텍처는 다양한 정보를 허용하는데 있어서 융통성이 있어야 하기 때문에 ITS 시스템 아키텍처를 개발하려는 방법으로 객체지향적 접근 방법을 제시하였다.

3. 차량내 동적경로안내시스템 논리아키텍처

3.1 객체기반의 논리아키텍처 구축 필요성

ITS시스템 아키텍처를 구축하는데 있어서는 다음의 2가지의 특징에서 접근하였다. 첫째는 사회의 요구 및 기술개발의 변화에 대처할 수 있는 재사용성이고 두 번째는 정보 및 통신과의 상호연결성 및 유지성이다.

객체지향적 방법을 적용하기 위한 ITS시스템 아키텍처는 시스템 아키텍처를 확장하거나 재사용하는데 용이하다. 시스템 아키텍처를 구축하는 방법들 중 가장 보편적으로 쓰였던 방법은 절차적 방법이었다. 이 방법은 크기가 큰 시스템을 분석하는데 조금 효과적이었다. 그리고 설계 기간이 짧은 경우에 장점을 가지고 있었다. 그러나 절차적 방법은 시스템에서 처리된 정보를 체계적으로 조직화하는데 어려움이 있으며, 새로운 기능을 조직화하고 시스템이 변화하거나 확장되어 질 때 많은 노력이 요구된다.

그러나 객체지향적 방법은 기능과 정보를 하나로 구성하고 있다. 그리고 기능과 정보는 “객체”를 이용하여 시스템을 나타낸다. 여기서 객체는 시스템내에 있는 정보와 기능을 단일화된 모델로 생성하는 것이다. 이 방법은 몇몇 서비스가 증가되거나 변화하는 경우에 강력하게 대처할 수 있다. 본 연구는 ITS시스템 아키텍처를 구축하는데 있어서 객체지향적 방법을 채택하였으며, 이것은 사회적인 요구와 기술의 개발에 따른 변화에 더 쉽게 대처할 수 있는 ITS시스템 아키텍처를 구축하는데 적합하기 때문이다. 또한 본 연구에서는 객체지향적 분석방법을 위하여 UML(Unified Modeling Language)를 이용하였다.

3.2 객체기반의 논리아키텍처 구축 절차 및 범위

프로세스의 분석 및 설계 절차는 첫째, 차량내 동적경로안내시스템의 사용자 시스템 요구 항목을 정의한다. 둘째, 정의된 시스템의 요구 항목에서 각 시스템의 정보의 속성 및 기능을 추출한다. 셋째, 추출된 정보의 속성 및 기능으로 모델을 생성한다. 본 연구에서 개발하는 모델은 크게 동적모델과 정적모델로 구분된다. 동적모델은 유즈케이스(UseCase Model)⁸⁾, 클래스책임추대이어그램(Responsibility Delegation Diagram(RDD))⁹⁾, 시퀀스 다이어그램(Sequence Diagram)¹⁰⁾으로 표현하였고, 정적모델은 클래스 다이어그램(ClassDiagram)¹¹⁾으로 표현하였다.

Fig 1은 차량내 동적 경로안내 시스템을 위한 논리아키텍처 구축 절차 및 범위를 나타낸 것이다.

6) Koutarou Katou(1998), ITS System Architecture Development, 5th ITS World Congress Conference

7) Youkihiro Matsunaga, and Tomoji Kishi(1998), Applying the Object-Oriented Method for ITS System Architecture Development, 5th ITS World Congress Conference.

8) 사용자의 요구를 실행가능 하게 하는 모델이다.

9) RDD는 use case 와 sequence diagram 사이에 상호작용을 나타내는 것이다.

10) Sequence Diagram은 객체들 사이에 메시지의 흐름을 묘사하는 것을 나타내는 것이다.

11) Class Diagram은 클래스 사이에 정적인 관계를 묘사하기 위해 사용하는 것이다.

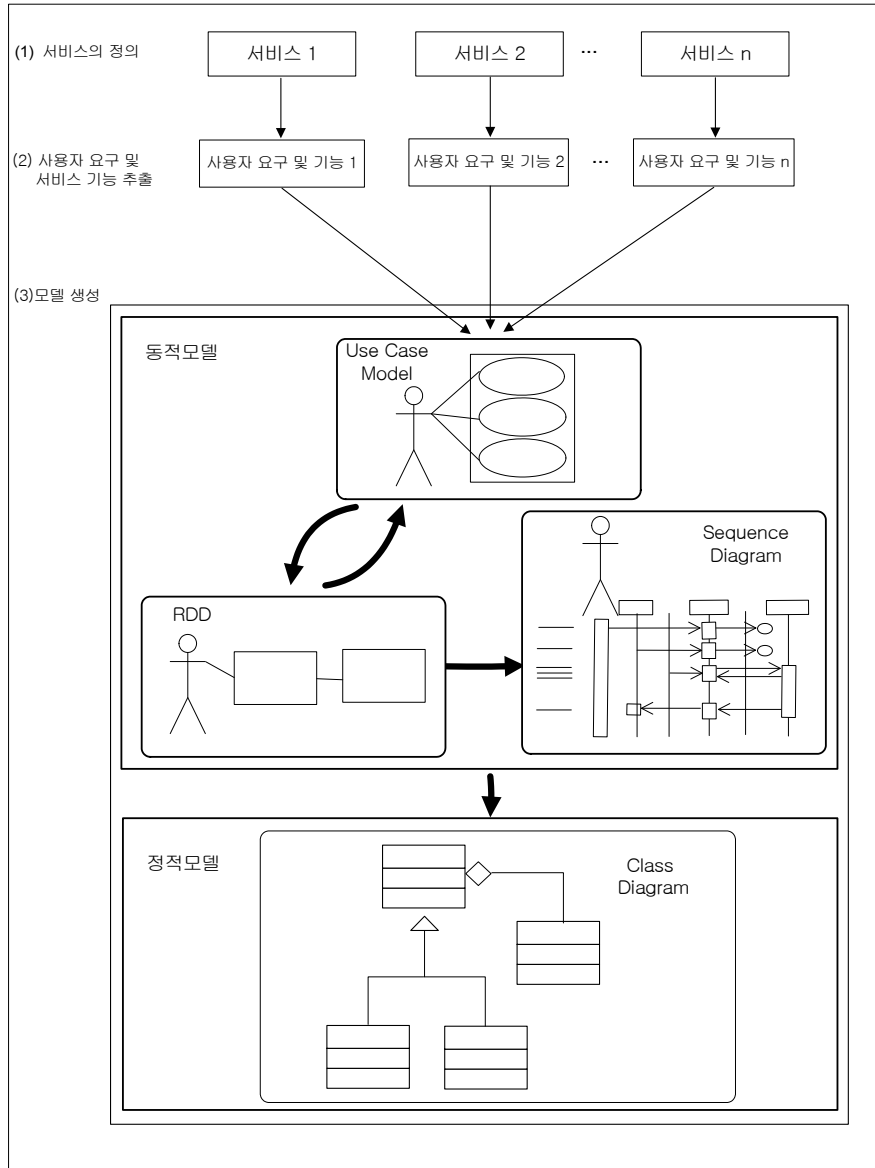


Fig 1. Logical architecture building procedures and scope

3.3 객체기반의 논리아키텍처 구축

본 연구에서 객체지향 기반의 논리 아키텍처 개발 방안을 제시하였다. 차량내 동적경로안내시스템은 차량단말기를 부착한 차량에 동적교통정보를 제공하면 차량단말기는 수신된 동적 교통정보를 토대로 운전자의 여행 기중점에 적합한 최단경로를 도출하고, GPS위성이나 노변통신장치를 통해서 차량의 현재위치를 파악하면서 주행안내서비스를 제공하는 서브시스템이다.

따라서, 본 연구에서는 차량내 동적경로안내시스템에 대한 사용자 요구사항 및 시스템의 기능을 정리하였다. Table 1 은 차량내 동적경로안내시스템을 위한 사용자 요구 및 시스템의 기능을 나타낸 것이다.

표를 보면, 사용자요구 및 시스템 기능은 크게 시스템의 운영상태를 점검하는 시스템운영상태점검, 차량의 상태를 점검하는 차량상태점검, 날씨, 주차정보 등을 제공하는 부가정보제공 그리고 차량의 경로를 제공하는 경로안내제공으로 구분하여 사용자의 요구 및 시스템의 기능을 분석하였다. 하지만, 본 연구에서는 시스템운영상태점검, 차량상태점검, 그리고 경로안내 제공에 대한 내용만 연구에 포함하였다.

Table 1. User requirements and system functionality requirements

요구조건	
시스템 운영상태점검	시스템은 차량의 운영시스템 상태를 점검하여 운전자에게 제공한다.
차량상태점검	시스템은 차량의 연료소모량 및 기능적인 요소와 같은 차량의 상태를 점검하여 운전자에게 제공한다.
	만약, 차량의 상태에 문제가 있다면, 시스템은 운전자에게 차량의 문제점을 제공하며 문제를 해결하기 위한 정보를 제공한다.
부가정보제공	시스템은 날씨정보, 타 수단의 요금정보와 같은 부가정보를 운전자에게 제공한다.
	부가정보는 주유소, 음식점, 차량수리센터 등의 정보제공 기능을 포함하고 있다.
경로안내제공	시스템은 운전자에게 선택된 목적지까지의 경로를 제공한다. 이러한 정보는 정보제공자로부터 정보를 제공받는다.
	경로정보는 노변정보시스템과 위치정보로부터 현재 차량의 위치와 정보제공자로부터 받은 경로를 제공한다.
	동시에 지도갱신제공자로부터 차량의 위치를 표시할 지도를 갱신받고 노변으로부터 받은 도로상태를 운전자에게 제공한다.
	경로정보는 경로를 나타낼 디스플레이를 제공하며 차량의 위치표시, 교통지도표시 내용을 운전자에게 제공한다.

1) 차량내 동적경로안내시스템 유즈케이스(Use Case) 모델

차량내 동적경로 안내시스템을 위한 유즈케이스 모델은 크게 상위 수준의 유즈케이스 모델과 하위수준의 유즈케이스 모델로 구분하였다. 유즈케이스 모델은 시스템을 사용하는 사용자가 시스템에 요구하는 정보를 시스템이 사용자에게 제공하는 것을 표현하는 것이다.

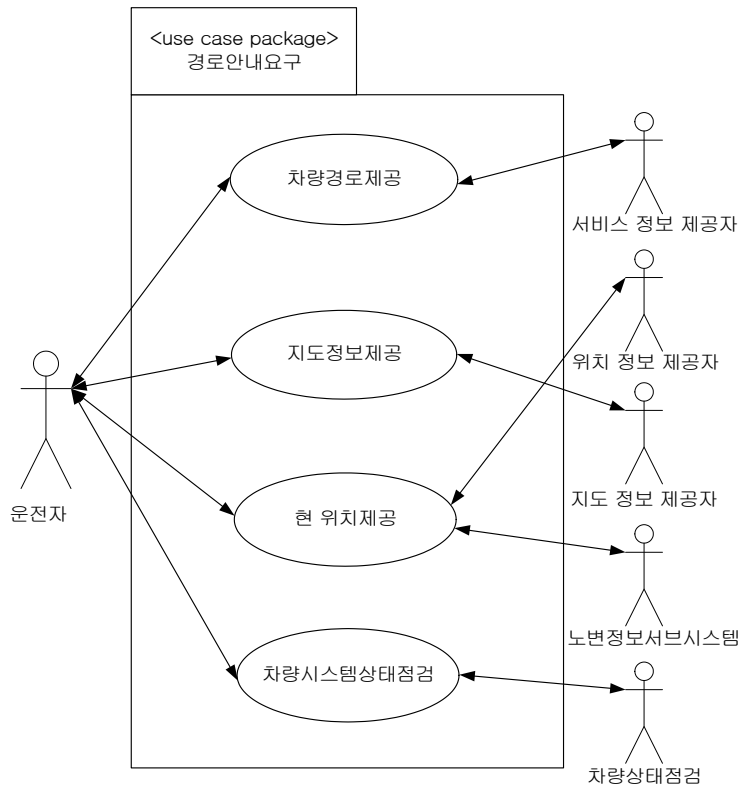


Fig 2. High-level use case model

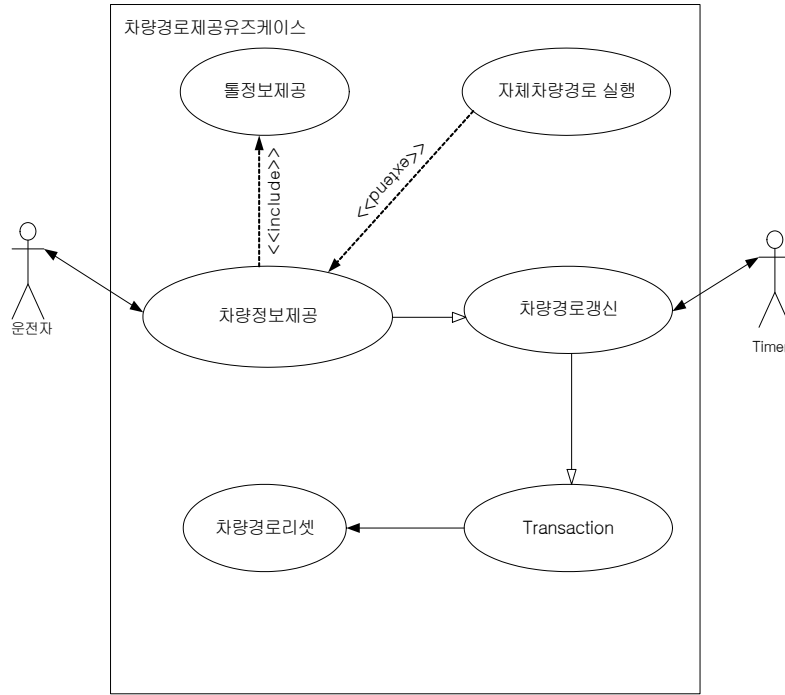


Fig 3. Lower-level use case model

2) 차량내 동적경로안내시스템 작업의뢰기술도 모델

차량내 동적경로 안내시스템의 작업의뢰기술도는 유즈케이스와 클래스 다이어그램, 시퀀스 다이어그램과의 관계에서 클래스나 객체를 어떻게 생성하는지에 대한 명확한 방법을 제시해주며 적합한 클래스를 선정하기 위한 방법을 제공해 준다. 작업의뢰 다이어그램은 자신이 하는 일에 대한 책임 아래 자기 자신이 할 수 있는 일과 그렇지 못한 경우를 다이어그램으로 표현하였다. Fig 4는 작업의뢰 다이어그램의 개념을 표현한 것이다.

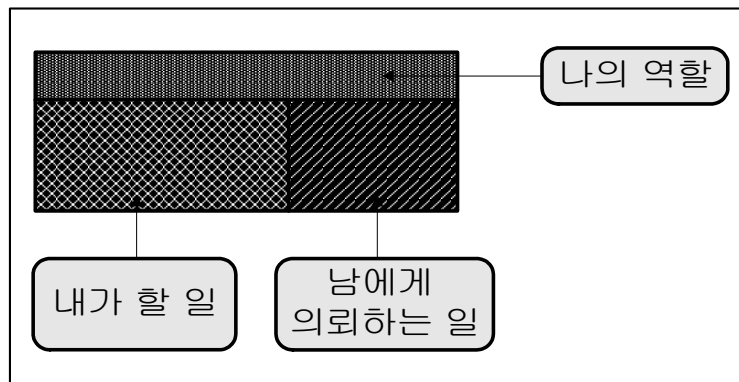


Fig 4. Conceptual diagrams of the work commissioned

Fig 5는 시스템운영상태에 대한 작업의뢰기술도를 나타낸 것이다. 그림을 보면, 차량시스템상태점검이라는 유즈케이스에서 시스템과 유즈케이스간의 관계를 작업의뢰 기술도를 통해 표현한다. 이를 통해 객체는 어떻게 생성되고 어떠한 관계를 가지는 지를 알 수 있으며 생성된 객체는 객체가 동적인 특징, 정적인 특징 그리고 두가지 특징을 가지는지를 구분 할 수 있다.

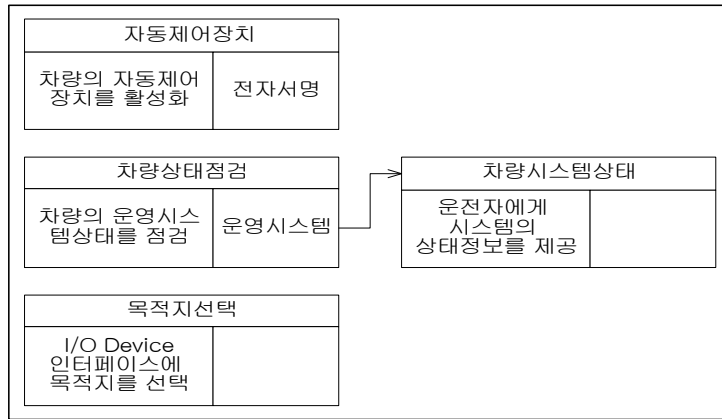


Fig 5. System operating status

3) 클래스 (객체) 생성

본 절은 작업의뢰기술평도를 통해 클래스 및 객체를 생성하였다. 객체는 자신이 무엇을 해야 하고 또 다른 클래스에게 자신이 할 수 없는 일을 요청해야 하는데 이러한 특성을 통하여 제시하였다. 클래스는 인터페이스인 I/O Device와 차량운영시스템, 지도정보제공자클래스, 차량의 위치를 제공하는 위치클래스, 경로를 안내하는 경로안내클래스, 통행을 생성하는 통행클래스, 경로를 계산하는 경로계산 클래스, O-D, 통행시간과 위치를 제공하는 Probe클래스, O-D경로 클래스, 통행시간클래스, 위치클래스, 노변정보를 제공하는 노변정보시스템클래스, 도로상태를 점검하는 차량ITS컴포넌트클래스로 클래스를 생성하였다.

4) 차량내 동적안내시스템의 시퀀스 다이어그램

시퀀스 다이어그램은 경로안내제공 유즈케이스를 처리하는데 메시지를 생성하여 클래스 (객체)간 관계를 표현한다. 우선 시퀀스 다이어그램을 만들기 위해서는 앞 절에서 제시된 작업의뢰기술평도의 흐름에 새롭게 만든 클래스 (객체)를 이용한다. 클래스와 클래스간의 관계표현은 메시지를 생성하여 표현할 수 있다. Fig 6은 차량내 동적경로안내시스템의 경로안내를 위한 시퀀스 다이어그램을 나타낸 것이다.

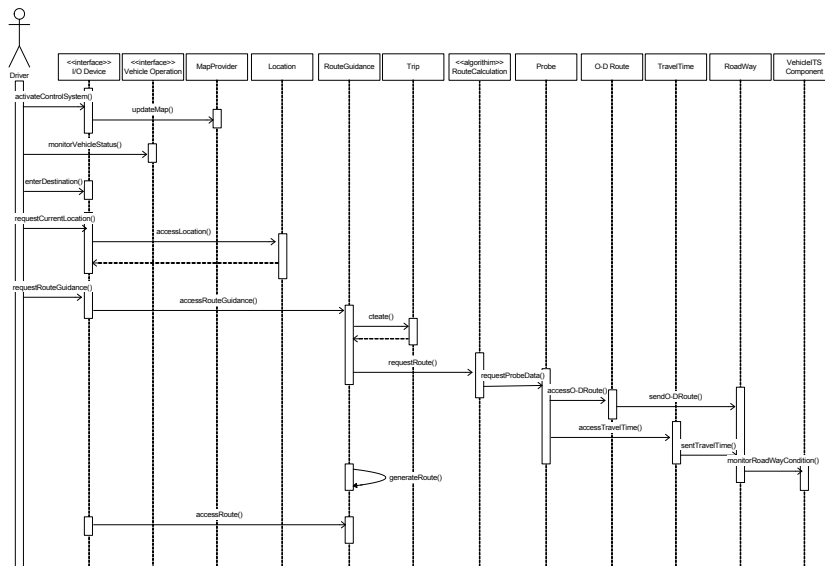


Fig 6. Vehicle Dynamic Information System Sequence Diagram

5) 차량내 동적안내시스템의 클래스다이아그램

본 연구에서 경로안내를 처리하기 위한 과정을 클래스다이아그램으로 표현하였다. 경로안내시스템의 논리아키텍처를 표현하기 위하여 클래스간의 관계를 이용하여 제어관계를 표현하였다. Fig 7은 차량내 동적경로안내시스템의 경로안내를 위한 클래스 모델이다.

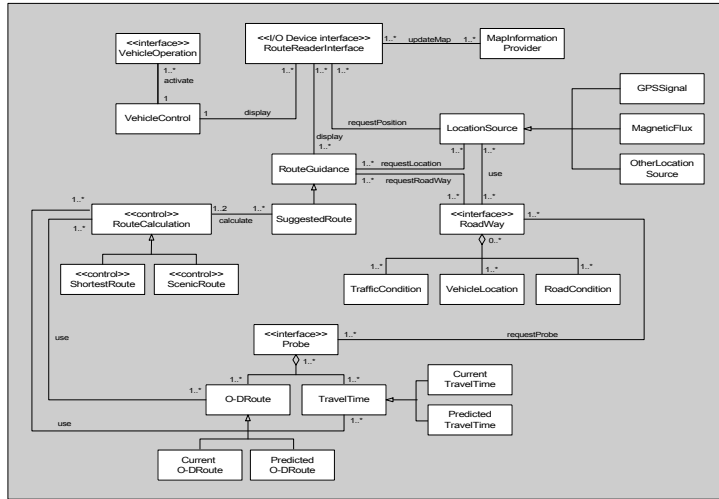


Fig 7. Class dynamic model of the vehicle guidance system

3. 논리아키텍처 구축 검증

본 연구에서는 논리아키텍처가 객체지향적으로 구축되었는지 알아보기 위하여 Chidamber와 Kemerer,(1994)에 의해 제시된 방법을 이용하여 분석하였다.

3.1 상속의 깊이 정도

Fig 8은 객체와 객체간에 얼마나 같은 속성과 메소드가 존재하는가를 의미를 나타내는 상속의 깊이정도를 나타낸 것이다.

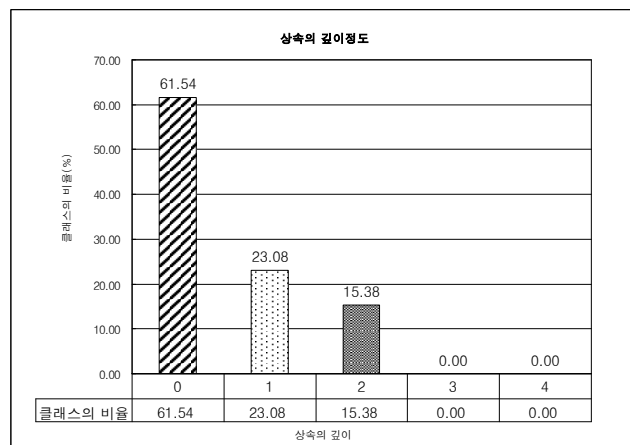


Fig 8. About the depth of inheritance

본 연구에서 구축한 아키텍처의 경우 클래스의 61.54%가 상속이 없는 것으로 나타났고 상속의 정도가 1개인 것은 23.00%로 나타났으며 상속이 2개인 것은 15.38%로 나타났다. 이러한 결과는 상속의 깊이정도가 보통 2보다 많은 경우는 복잡한 시스템아키텍처를 의미하는 것이기 때문에 본 연구에서 만들어진 아키텍처는 상속의 깊이 측면에서 잘 구축되었다고 평가할 수 있다.

3.2 클래스의 객체결합관계

클래스의 객체결합관계는 독립적인 클래스가 얼마나 많이 있는가를 의미한다. Fig 9는 클래스의 객체결합관계를 표현한 것이다. 그림에서 보는 바와 같이, 독립적인 클래스는 36.36%로 나타났고, 클래스의 객체결합이 1개인 경우는 36.36%, 2개인 경우는 27.27%로 나타났다. 클래스의 객체 결합 관계는 얼마나 시스템아키텍처가 재사용성에 적합한지를 나타내는 척도이다. 본 연구에서 제시한 시스템의 논리아키텍처는 재사용성 측면에서 적합한 것으로 분석되었다.

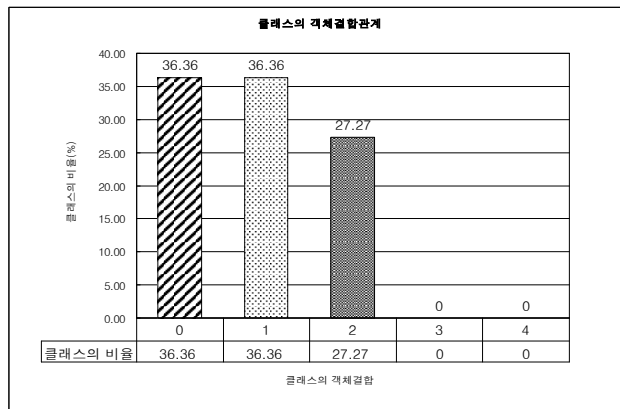


Fig 9. Combined class of object relations

3.3 클래스의 하위클래스 정도

클래스의 하위클래스정도는 상속과 관련되어 있다. 상속이란 하나의 클래스가 또 다른 클래스에게 자신이 가지고 있는 속성을 똑 같게 부여하는 것으로 시스템을 효율적으로 운영하는데 적합한 방법이다. 하지만 상속이 많아지면 시스템의 운영 속도 측면에 좋지 않은 결과를 초래하게 된다. Fig 10은 하위클래스의 정도를 나타낸 것이다. 그림에서 하위클래스가 없는 경우는 4개, 1개의 하위클래스를 가지는 경우는 4개로 나타났고, 2개의 하위클래스를 가지는 경우는 3개, 3개 이상은 없는 것으로 나타났다.

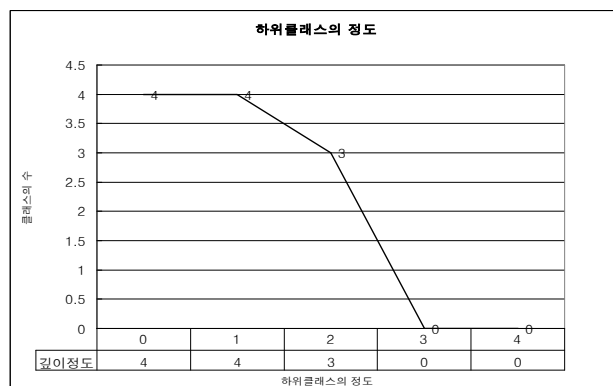


Fig 10. Combined class of object relations

4. 결론 및 향후 연구과제

본 연구에서는 차량내 동적경로안내시스템을 대상으로 객체지향적 방법을 적용한 논리아키텍처를 구축 방안을 제시하고자 하였다. ITS 시스템을 구축하는데 있어야 중점으로 두어야 할 것이 시스템의 적용성 및 재사용성이다. 정보기술의 발전 및 이용자들의 요구사항의 다양성은 시스템이 자주 변화하고 대응해야 한다는 증거이다. 시스템을 구축하는데 있어서 기존의 절차적 방법으로는 환경의 변화에 대응하지 못하는 단점을 가지고 있다. 국제변화에 민첩하게 대응하기 위해서는 새로운 패러다임 찾아야 하는데 그게 바로 객체기술을 이용한 시스템 구축이다. ITS분야도 마찬가지이다. 본 연구에서 차량내 동적경로안내시스템을 이용하여 ITS시스템 아키텍처를 구축방안을 제시하였다. 본 연구를 통하여 제시된 시스템 아키텍처 구축 방안으로 다른 시스템의 아키텍처 구축에 기반을 마련할 뿐만 아니라, 대상 시스템의 연계 및 확장에 효율적으로 이용할 수 있을 것이라 기대한다.

References

- MOCT, National Intelligent Transportation Systems (ITS) Architecture II , 1999
- ISO TC 204, Transport Information and Control Systems-Reference Model Architecture for the TICS Sector-Part 1 : TICS Fundamental Services, ISO/TR 14813-1, 1999.
- ISO TC 204, Transport Information and Control Systems-Reference Model Architecture for the TICS Sector-Part 2 : Core TICS Reference Architecture, ISO/TR 14813-2, 1999.
- ISO TC 204, Transport Information and Control Systems-Reference Model Architecture for the TICS Sector-Part 3 : Example Elaboration, ISO/TR 14813-3, 1999
- ISO TC 204, Transport Information and Control Systems-Reference Model Architecture for the TICS Sector-Part 4 : Reference Model Tutorial, ISO/TR 14813-4, 1999
- Doug Rosenberg and Kendal Scott, Applied Use Case Driven Object Modeling, Addison-Wesley, 2001.
- Perdita Stenvens and Rob Pooley, Using UML Software engineering with object and components, Updated Edition, Addison-Wesley, 2000
- Seo, Jonggook(2015) A Hierarchical Analysis on the Commuting Behaviors and Urban Spatial Characteristics. Journal of the Korean Society of Disaster Information. Vol.11 No.4, pp. 506-514