IN-VEHICLE DYNAMIC ROUTE GUIDANCE SYSTEM BASED UPON OBJECT-ORIENTED ITS LOGICAL SYSTEM ARCHITECTURE

객체지향 기반의 ITS 시스템 논리아키텍쳐 구축방안

정종모 (청석엔지니어링, 연구원) 김형진 (연세대학교 사환건, 교수) 홍재영 (Integral Signal Processing, 사장) 박범진 (건설기술연구원, 연구원)

Key Words : Object-Oriented, In-Vehicle Dynamic Route Guidance System, ITS, Architecture

목 차

I. Introduction

Modern society is becoming increasingly information-oriented at the global level, and the road traffic is no exception. Also in South Korea, the use of information technologies on roads, traffic and vehicles are being promoted in order to solve such problems as traffic accident, congestion and the environmental deterioration, as well as to meet the market-expansion needs of the automobile industry, the information and communication industry and other industries. To promote the use of information technologies on roads, traffic and vehicles, the Ministry of Construction & Transportation (MOCT) initiated the National Intelligent Transportation Systems (ITS) Architecture project in 1999(1), which is based on the 64 user services. However, The Prime Question is whether to cope with user needs or, rather, to meet change of ITS information technologies in short-term and long-term future transportation conditions as the basis for ITS services. Therefore ITS architecture should be developed in its method to maintain of consistency, reuse, and expansion, especially when the system is large and/or complex. There have been studies that developed ITS Architecture. Koutarou(1999), Youkihiro Matsunaga & Tomoji

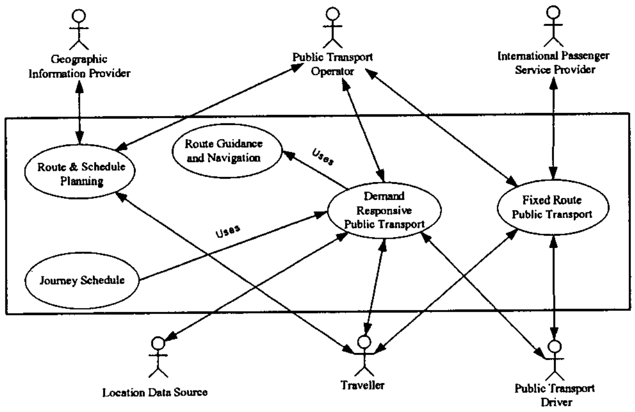
Kishi(1999), Yuji Hasegawa & Hiroshi Miki(1999), and Jan W. Tierolf(2000) presented that object-oriented method is suitable for the maintenance of consistency, reuse, and expansion, Nevertheless, they didn't presented how to apply object-oriented method for ITS Architecture. In our view, for ITS Architecture, such as in-vehicle dynamic guidance systems, we focused on how to apply object-oriented for in-vehicle dynamic guidance systems, one of ITS Architecture.

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

1. The Current Condition of International ITS LSA

As we know, the International ITS architecture defined the system architecture as a high level abstraction. The system architecture is subdivided into the following categories: Reference architecture, Logical architecture, and Physical architecture (1999). ISO 14813(2), (3), (4), (5) is a multi-part Technical Report detailing the Reference Architecture and its associated documentation for the Transport Information and Control System (TICS) (1999). Its core TICS reference architecture, one of the Reference Model Architecture for the TICS sector, is completely redesigned by applying object-oriented method (1999). This

TICS Model have a total of 32 services in TICS fundamental services and 9 conceptual packages and their abstract classes in Core TICS Reference architecture; that is Roadway, Transport, Vehicles, Events, Payments, Roadside Periperals, Operating Interfaces, Travel Terminals and Vehicle Interfaces.

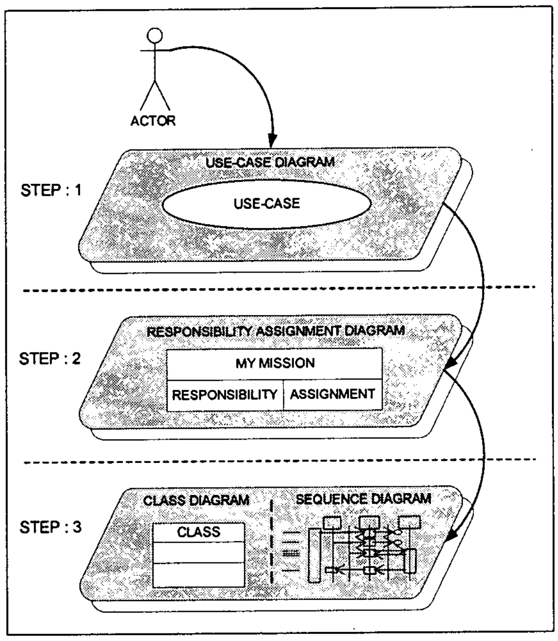

Geographic Information Provider
Public Transport Operator
International Passenger Service Provider
Route & Schedule Planning
Route Guidance and Navigation
Uses
Demand Responsive Public Transport
Fixed Route Public Transport
Uses
Journey Schedule
Location Data Source
Traveller
Public Transport Driver

<Figure 1> Use Case Model in TICS

National ITS architecture in Korea also defined reference architecture, logical architecture, and physical architecture like the International Standard. Logical architecture is divided into 62 user services under 7 major service categories. However, it appears that up until now, the logical architecture was designed based upon the so-called process-oriented which looks the system in terms of function and data structure. Recently, The Korea Research Institute for Human Settlements (KRIHS), one of the research agencies of MOCT, is under consideration ITS architecture that based upon object-oriented system that is to be applied by international ITS architecture. Besides, a number of Koreas research institutes actively participate in international standardization actively joined the international stream as one of ISO (the International Organization for Standardization) TC(Technical Committee) 204s WG (WorkingGroup).

2. The Basic Activities of a Business Model and Information System in ITS LSA

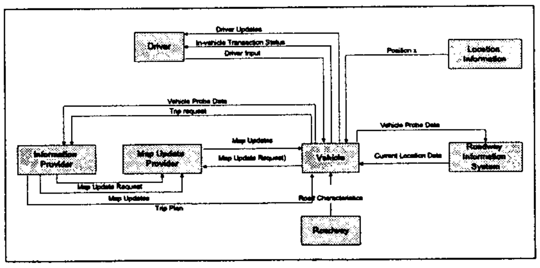
As a means to design architecture guideline, we adopted the object-oriented method, which is used in ISO/TC204. More specifically, we chose an in-vehicle dynamic route guidance system a subsystem of ATIS, for adopting the Unified Modeling Language (UML) notation (6), (7). The reason for this is that at phase of ITS LSA. It offers us how to analyze and design ITS LSA. Since the UML requires the object-oriented paradigm, we designed framework that develops the ITS LSA for in-vehicle dynamic guidance systems. Figure1 shows the basic activities of a business model and information system in ITS LSA. The outline of this condition is shown as following.



<Figure 2> The Basic Activities of a Business Model and Information System in ITS LSA

1) Requirements Capturing

In order to construct a more effective and stable ITS LSA, it is necessary to capture user needs, system functional requirement and define the services required to fulfill these needs. The purpose of requirements capturing is to capture a good picture of what requirement are imposed on the information system, built on which users want and what user oneself can accomplish. From the viewpoint applying the object-oriented for ITS LSA to in-vehicle dynamic guidance systems, we grouped subject users and system into system operation status, vehicle operation status, vehicle guidance; surveyed user needs and systems functional requirement. To capture user needs and system functional requirement to in-vehicle dynamic guidance systems, we, as shown schematically shown in Figure 2, defined the limit of application about systems that based upon the object-oriented method.



<Figure 3> Example for Information Flow in In-vehicle Guidance System

Ⅱ. An illustrative Example

1. The Use Case Model in ITS LSA

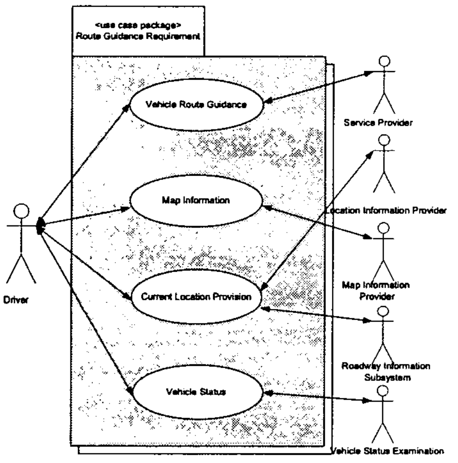
To analyze and design ITS LSA more faithfully, our approach to ITS LSA using the use case model is to capture the part of the business in which we are interested by using the use case model. We will describe the business and its environment.

The business is made up of all the relevant business processes, for example the processes that are appropriate for system application. The external environment is made up of, for example, user, partners and suppliers that take part in the process.

In our research, we represent that these processes are modelled with use cases, while the environment is modelled by using what we call actors.

In this way, we can propose use-case package of route guidance requirement. Besides, we can adopt the proposed use-case package that belongs to route guidance under Traveler Information in TICS reference architecture. In this top-level, there are four use-cases under use-case package that so-called Route Guidance Requirement: *vehicle route, map information, current location, and vehicle system status,* and six actors: *Driver, Service Provider, Location Information Provider, Map Information Provider, Roadway Information Subsystem, and Vehicle Status Examination*

The top-level use case diagram is shown in Fig. 4. It notes that proposed actors and use-cases in this diagram are in very abstract terms.



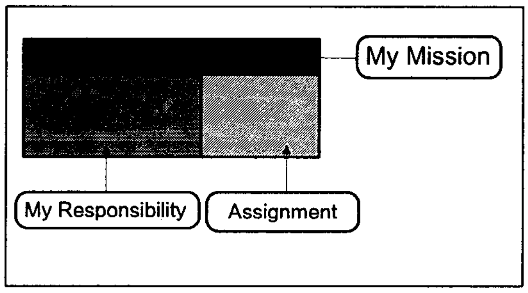
<use case package>
Route Guidance Requirement
Vehicle Route Guidance
Map Information
Current Location Provision
Vehicle Status
Driver
Service Provider
Location Information Provider
Map Information Provider
Roadway Information Subsystem
Vehicle Status Examination

<Figure 4> A use case model for the in-vehicle dynamic route guidance system

Fig. 4 described the use case model for the in-vehicle dynamic route guidance system. In fig. 4, we proposed 4 use cases: vehicle route guidance, map information, current location provision and vehicle status. It is more logical that our system belongs to the in-vehicle system that connects outer systems since our users primarily want to know the current travel route. In Fig 4, we are trying to identify all the relevant use-cases about route guidance requirement from actors point of view.

2. RAD Model in ITS LSA

In this Section, we should be very careful about existing object-oriented method. We know that there is a big conceptual analysis/design gap in how to transform the captured requirements into the appropriate attributes and methods of the suitable class/object. To reduce analysis/design gap, we suggested *Responsibility Assignment Diagram (RAD)* that will express how to create class/object easier than robustness analysis (2000). Even though it does not belong to UML. The RAD defines interaction to process one of use-case. It organized into three factors: My Mission, My Responsibility, and Assignment. Fig. 5 expressed composition for RAD. In Fig. 5, RAD represents what my mission should be able to do with Information System

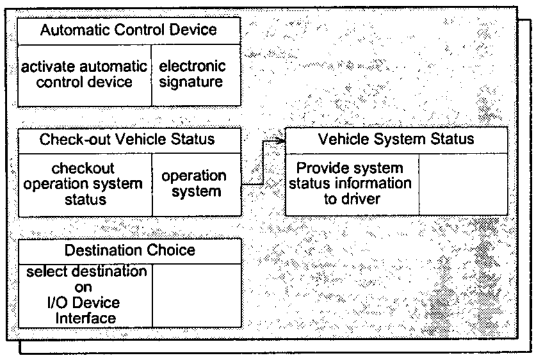


<Figure 5> Example for Responsibility Assignment Diagram

In our approach, we present that one view is drawn a flow of event for information-system use case, and for each view we present only those objects that participate through RAD.

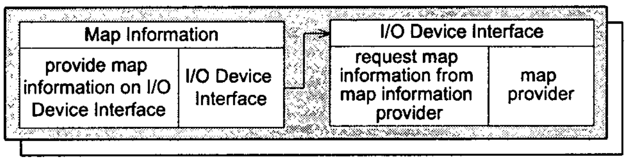
RAD will govern work with all of the other diagrams. It notes that each RAD can represent a flow of events for several use case. We should show RAD, which processes each use-case.

Fig. 6 describes the system operation by adopting RAD. This includes three missions: *Automatic Control Device, Check-out Vehicle Status, and Destination Choice.* The process is explained by system operation that expresses a system status. Though the system operation only shows the relations between vehicle system and driver. If there is no error in system status, vehicle system status will display system status information to driver and the driver will select his/her destination on I/O Device Interface.


Automatic Control Device
activate automatic control device
electronic signature
Check-out Vehicle Status
checkout operation system status
operation system
Vehicle System Status
Provide system status information to driver
Destination Choice
select destination on I/O Device Interface

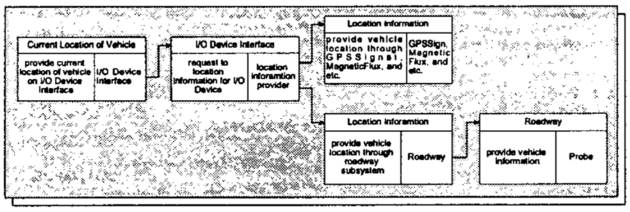
<Figure 6> Example for System Operation RAD

Fig. 7 shows a flow of event map information by using the RAD. In fig.11, map information is required through the I/O Device Interface and the I/O Device Interface is connected with map provider.


Map Information
provide map information on I/O Device Interface
I/O Device Interface
I/O Device Interface
request map information from map information provider
map provider

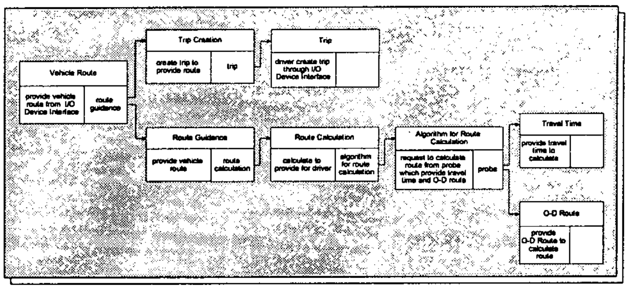
<Figure 7> Example for Map Information RAD

We are trying to identify all current location of vehicle from two points of view. There are two methods: a long-range from GPS Sign, Magnetic Flux, and etc and a dedicated short-range communications (DSRC) system. Current Location of Vehicle RAD, as schematically shown in Fig. 8, presents multiple connections between the I/O Device Interface and Location Information. The multiple connections are to capture the location information through variable methods that are provided via the long-range and DSRC.


Current Location of Vehicle
provide current location of vehicle on I/O Device Interface
I/O Device Interface
I/O Device Interface
request to location information for I/O Device
location inforamtion provider
Location Information
provide vehicle location through GPSSignal, MagneticFlux, and etc.
GPSSign, Magnetic Flux, and etc.
Location Inforamtion
provide vehicle location through roadway subsystem
Roadway
Roadway
provide vehicle information
Probe

<Figure 8> Example for Location Reference of Vehicle RAD

Fig. 9 describes the process by introducing the concept of RAD Pattern Vehicle Route, Trip Creation, Route Guidance, Trip, Route Calculation, Algorithm for Route Calculation, Travel Time, and O-D Route to provide and guide vehicles route. The vehicle route is subdivided into trip creation and route guidance. Trip Creation is created by trip that is provided to driver through I/O Device Interface We also depicted Route Guidance that provides vehicle route, which is calculated by using algorithm for route calculation. Algorithm for Route Calculation requests to calculate route from probe that provides travel time and O-D route.


Vehicle Route
provide vehicle route from I/O Device Interface
route guidance
Trip Creation
create trip to provide route
trip
Trip
driver create trip through I/O Device Interface
Route Guidance
provide vehicle route
route calculation
Route Calculation
calculate to provide for driver
algorithm for route calculation
Algorithm for Route Calculation
request to calculate route from probe which provide travel time and O-D route
probe
Travel Time
provide travel time to calculate
O-D Route
provide O-D Route to calculate route

<Figure 9> Example for Vehicle Route RAD

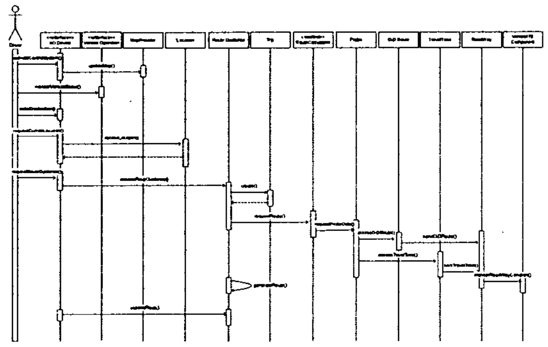
Based upon the proposed RAD, we can further identify which use-case defined in use-case diagram can be processed by RAD more faithfully. In our research, we can then carefully select the potential usages of each class in terms of entity, control, and interface class groups. We found that the appropriate classes can be selected easier with our proposed methodology. The proposed classes included I/O Device Interface, Vehicle Operation, Map Provider, Location, Route Guidance, Trip, Route Calculation, O-D Route, Probe, Travel Time, Roadway, Vehicle ITS-Component.

3. Sequence Diagram in ITS LSA

In our research work, we applied the proposed classes to sequence diagram that processes the in-vehicle dynamic guidance system by using messages between classes. To represent messages to sequence diagram, We makes sequencing framework for in-vehicle dynamic guidance system to systematically:

- Driver activates system through transmission messages, which is ControlSystem(), in I/O Device Interface
- The I/O Device updates map information that sends updateMap() to MapProvider
- Driver check-out operation status of vehicle to VehicleOperation, which is a interface, by sending monitorVehicleStatus().
- If Vehicle Operation System have not error, Driver selects the destination on I/O Device Interface using enterDestination().
- Driver requests vehicle location to acquire the current vehicle location information by sending requestCurrentLocation().

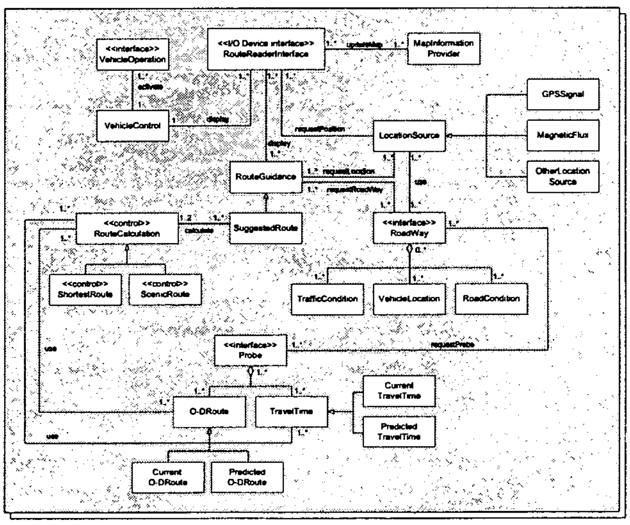
- To acquires the vehicle location, It requests and receives current location information to location class using accessLocation().
- Driver requests route to the I/O Device Interface by sending requestRouteGuidance().
- The I/O Device Interface accesses RouteGuidance class that receives message, which is accessRouteGuidance().
- The RouteGuidance creates Trip and send message which is create() rightly.
- The RouteGuidance receives Trip again.
- The RouteGuidance requests route by sending requestRoute() to the RouteCalculation which is the algorithm interface
- To provides the route, The RouteCalculation requests O-D Route and TravelTime to calculate route to Probe by using requestProbeData().
- The Probe accesses to the O-D Route and TravelTime by sending accessO-DRoute()/accessTravelTime().
- O-DRoute sends sendO-DRoute() to Roadway.
- TravelTime sends sendTravelTime() to Roadway.
- The Roadway is inspected traffic condition by VehicleITSComponent. Then, generates route
- The RouteGuidance generates route to generateRoute()
- The I/O Device Interface sends to driver vehicle route which is generated by the RouteGuidance.



<Figure 10> Example-Sequence diagram for In-vehicle dynamic route guidance system

Fig. 10 expressed sequence diagram using the messages that represent relation classes. In Fig 10, the class relationship that presents sequence diagram is illustrated. In our research work, we also represented the proposed classes to other method that is class diagram that process the in-vehicle dynamic guidance system by using messages between classes.

In Fig 11, we are trying to identify all relevant classes. VehicleOperation activates VehicleControl. Then, VehicleControl send vehicle operation status under RouteReaderInterface, which is I/O Device Interface, showing the vehicle operation status. RouteReaderInterface also updates map information from MapInformationProvider. RouteReaderInterface requests current location to LocationSource. Then, LocationSource requests vehicle location from GPSSignal, MagneticFlux, and OtherLocationSource, which are subclasses. At the same time, LocationSource uses RoadWay, which is the interface. The Roadway has a subclass that inherits TrafficCondition, VehicleLocation, and RoadCondition. It requests O-D Route and TravelTime to Probe which is the interface. We also represented generalization relationship between O-D Route and current O-D Route and Predicted O-D Route. TravelTime under probe indicates generalization relationship between Current TravelTime and Predicted TravelTime and TravelTime. This indicates that the TravelTime and O-D Route are an abstract class.
RouteReaderInterface illustrats an association between RouteGuidance. This shows route through communication between two classes. To provide route, SuggestedRoute, which is subclass, requests to provide route to RouteCalculation which is route calculation algorithm. This indicates a generalization relationship between RouteGuidance and SuggestedRoute. To calculate route, we represented an association relationship between SuggestedRoute and RouteCalculation which is control. This calculates route to the SuggestedRoute. RouteCalculation is a generalization relationship RouteCalculation, which is control, and ShortestRoute and ScenicRoute. Because the ShortestRoute and ScenicRoute classes inherit from the RouteCalculation, either these class can be substituted in this relationship.



<Figure 11> Example-Class diagram for In-vehicle dynamic route guidance system

III. Conclusion

The purpose of this study is to present in-vehicle dynamic route guidance systems based upon object-oriented ITS logical system architecture (ITS LSA). We should be expected to apply efficiently for the connection and expansion of systems as well as to present systems based upon object-oriented of the other ITS LSA. We represented more detailed ITS systems that have reusability, flexibility and maintenability. To analyze/design ITS LSA more faithfully, our approaches to ITS LSA using the object-oriented method are: (1) to create four diagrams: use-case, responsibility assignment, sequence, and class; (2) to apply in-vehicle dynamic route guidance system for the quality of object-oriented design in our conceptual model: vehicle route guidance, map information, current location provision, and vehicle status. Based on this analysis, we found that our design works appropriately within object-oriented method. In our research work, we have shown that our proposed LSA approach method plays a vital role to develop ITS LSA based upon the object-oriented through an in-vehicle dynamic route guidance system while confirming the upcoming international ITS LSA.

Reference

1. MOCT, National Intelligent Transportation Systems (ITS) ArchitectureII , 1999
2. 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.
3. 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.
4 ISO TC 204, Transport Information and Control Systems-Reference Model Architecture for the TICS Sector-Part 3 : *Example Elaboration*, ISO/TR 14813-3, 1999
5 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
6 Doug Rosenberg and Kendal Scott, *Applied Use Case Driven Object Modeling*, Addison-Wesley, 2001.
7 Perdita Stenvens and Rob Pooley, *Using UML Software engineering with object and components*, Updated Edition, Addison-Wesley, 2000