A Novel Design of Intelligent Transportation System Architecture with Object Oriented Process

Autthasith Arrayangkool and Asst. Prof. Apinetr Unakul
KMITL, Bangkok, Thailand
art958@hotmail.com

Abstract- *Nowadays, some ITS architectures are wildly used and they describe main structural and functional process of ITS system in a format of data flow diagrams. These diagrams have intensive rigidity, non-reusable property, large size and complexity. Disadvantages of data flow diagrams can be dissipated by using object oriented process and UML. UML can represent ITS systems in diverse views and have many types of diagram. However, UML diagrams are easy to understand. The object oriented methodology has flexibility and reusable characteristic. Therefore, by using object oriented process and UML, designers or developers can understand, design, develop and deploy ITS systems easily. This paper describes details and benefits of object oriented ITS architecture and comparisons of object oriented design with another design methodology.*

Keywords: ITS architecture, object oriented process.

1. Introduction

The Intelligent Transportation System (ITS) is a system using various technologies to improve convenience, safety and efficiency of transportation. Nowadays, the ITS systems are difficult to design, develop and deploy because of limited area for new infrastructure, complexity and diversity of systems and a problem of non-interaction between novel and existing systems. Consequently, the ITS architecture plays a role in facility related to structural and functional plans and characteristic of interaction between two or more systems. Thus, the efficiency of ITS systems depends on ITS architecture. An existing ITS architecture, Keystone Architecture Required for European Networks (KAREN) project is particularly interest to European ITS developers while the National ITS Architecture is being promoted by the U.S. Department of Transportation. Both of these architectures have different physical and functional system views. However they describe structure and functional process by data flow diagrams. Although, data flow diagrams can explain sending and receiving information of components, these diagrams have intensive rigidity, non-reusable property, large size and complexity. Some characteristics of data flow diagrams may cause troubles of uncomfortable development and modification. These problems can be solved by using object oriented process and UML. Unified Modeling Language (UML) has many types of diagram such as use case diagram, class diagram, collaboration diagram, component diagram and deployment diagram. However, all UML diagrams are easy to understand. The object oriented approach has flexibility and reusable feature and can represent ITS systems in various views. Designers and developers can gain benefit from advantages of object oriented process and UML to design, to develop and to deploy ITS system comfortably.

This paper presents characteristics of object oriented design, UML, views and their diagrams in section 2. Structural architecture features and components details are in section 3. Design of ITS system is in section 4. Section 5 describes assessment and comparison of object oriented with another methodology and section 6 concludes this paper.

2. Object Oriented Design and UML Review

Object-oriented design is the discipline of defining the objects and their interactions to solve a problem. It has features such as encapsulation, abstraction, polymorphism, and inheritance. Encapsulation or Information hiding is an ability used to protect some components of the object from external entities. Object oriented approach has the inheritance property that sub-class can inherit characteristics from super-class. Polymorphism means that you can have multiple classes that can be used interchangeably. Thus, object oriented methodology has flexibility and reusable property. Moreover, object oriented design is absolutely represented by UML. UML is an important tool for system development process. We can use UML to generate models and diagrams and to communicate with participants. UML can explain systems in diverse views by various diagrams as follows:

Use case view / User view presents commitments of systems by consideration about users and their use cases. This view consists of use case diagrams. They can display user requirements or system services.

Class view shows structural designs and functional sequences of systems. This view is formed of class diagrams explaining system structures, sequence diagrams, collaboration diagrams, state diagrams and activity diagrams describing working steps of systems and cooperation between two or more systems.

Component view represents software implementation by component diagrams composed of software modules and relationships between two or more modules.

Deployment view explains physical systems actually deployed in defined area. In this view, there are deployment diagrams showing physical structures,

communications between entities and software structures deployed within each entity.

3. Object Oriented ITS Architecture

Design and development process of this architecture began with studying and analyzing well-known existent ITS architectures. We can understand many good prominent points of these architectures and use them to develop an object oriented ITS architecture. Meanwhile, we can find some their disadvantages which can be reduced by using object oriented process and UML. The object oriented ITS architecture is consisted of five main packages. Sausage diagram package, organization package and product and technology package work together in a format of various systems called system package. The last package is data model leading to cooperation between system compositions or between systems. Characteristics, sources, importance, design and development of each package can be described as follows:

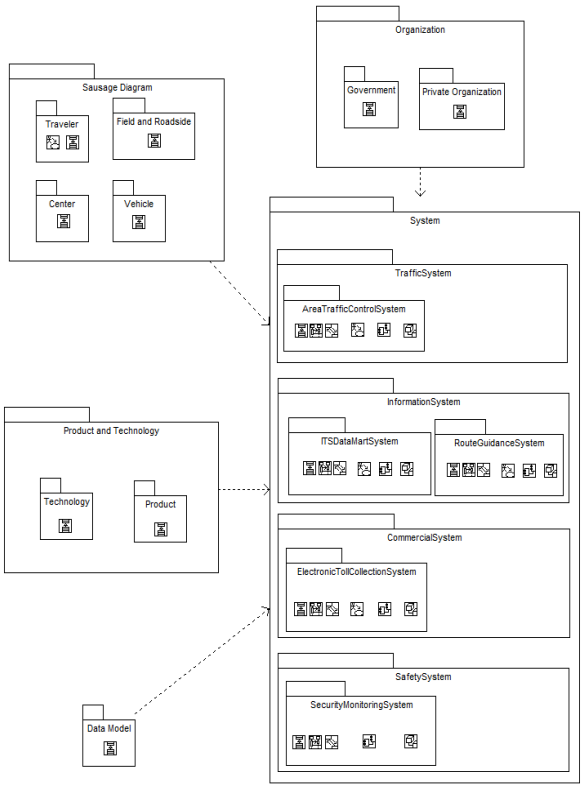

Figure 1. The object oriented ITS architecture

A) Sausage diagram package is a significant part of each traffic system. This package is originated from National ITS Architecture's sausage diagram adjustment and traffic systems analysis. We can classify compositions and their relationships into four sub-packages as follow:

Traveler package collects characteristics of travelers, system users and their system usages. This package has two views which are use case view and class view. The use case view shows various types and diverse use cases of travelers.

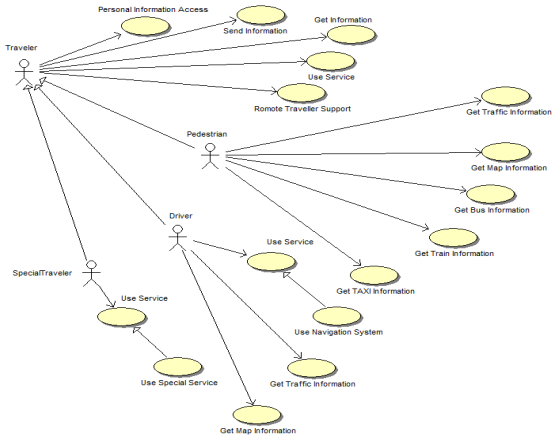

Figure 2. Traveler use case diagram

The Class view has a class diagram containing the Traveler base-class and sub-classes of various type of travelers such as Pedestrian and Driver.

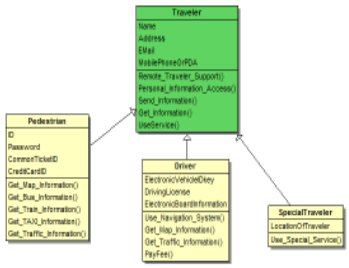

Figure 3. Traveler class diagram

Center package represents centers of traffic systems. It has a class diagram of class view indicating the Center base-class, sub-classes of four main types of centers, sub-classes of specific type centers and their relationships. One of four main types of systems is the TrafficCenter class which controls and manages traffic systems. The CommercialCenter class looks after financial activities and ITS systems specially related to business. The SaftyCenter class works as a security and safety observer including repairs unexpected situation. The TravelInformationCenter class collects information or data related to traveling and services users or travelers.

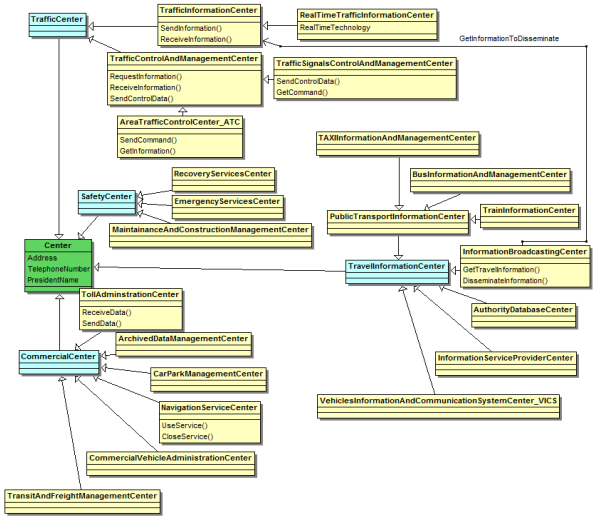

Figure 4. Center class diagram

Field and Roadside package is a container of various infrastructures. In its class view, there is a class diagram which presents the FieldAndRoadside base-class and sub-classes of diverse types of Infrastructures such as motorways, roads, bus stops and traffic signals.

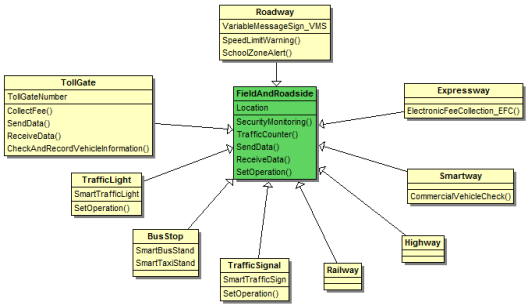

Figure 5. Field and Roadside class diagram

Vehicle package is a representative of all vehicles in systems. Its class view has a class diagram explaining the Vehicle base-class and sub-class of diverse types of vehicles such as PersonalVehicle and TransitVehicle.

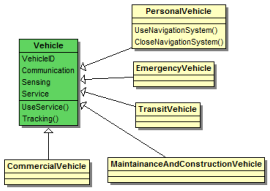

Figure 6. Vehicle class diagram

B) Organization package represent organizations or institutes associated with ITS system. This package can be separated into the government package and the private organization package following Figure 7.

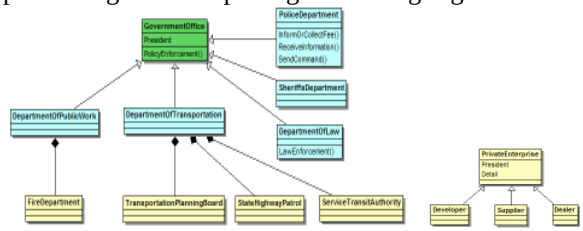

Figure 7. Government and private organization class diagram

C) Product and Technology package plays a role in ITS system improvement. Designers and developers can choose to use many types of diverse products or technologies to develop ITS system following Quality of Service (QoS) requirements. Some products or technologies may be created by organizations. Products can be divided into display, sensor, communication, OS and software type. Meanwhile the Technology base-class has SensorTechnology, DisplayTechnology, ProcessingTechnology and CommunicationTechnology sub-class. One of the technology attributes is standard.

D) System package originates from cooperation of sausage diagram package, organization package and product and technology package. System package can be separated into four sub-packages by characteristics and main purposes of systems as follows:

Traffic system package has objectives related to control or management traffic. For example, Area Traffic Control (ATC) system controlling traffic congestion level and traffic flow is a sub-package of traffic system package.

Commercial system package has purposes associated with financial activities or business such as Electronic Toll Collection (ETC) system.

Safety system package aims at preserving safety and security of traffic systems and reducing accident. An Example of safety systems is security monitoring system.

Information system package is a container of systems having purposes related to data collection and information service such as route guidance system and ITS data mart system.

The design of these systems is presented in section 4.

E) Data model package explains the format of information used to communicate within system or between systems effectively. This data model is composed of three global layers (Geographical data layer, Transportation network layer, Physical layer) following iTransIT framework paper.

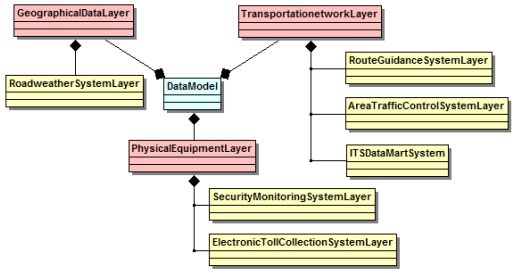

Figure 8. Data model class diagram example

4. ITS system design

The ITS system design with object oriented approach starts from consideration related to defined area requirements or problems. A next step is seeking for system types which can solve problems or support requirements. After that, all systems will be designed in various views. Lastly, data model of all systems will be created. All views of ITS system design can be described by an example of Area Traffic Control (ATC) system for Phuket city (in Thailand) as follows:

A) Use case view is a user view relevant to various types of users and their system usages. In this view, the Quality of Service (QoS) is taken into account in order that systems are in conformity with defined area. This example has two Traveler types are Driver and Pedestrian. Use cases of Traveler are Watching Traffic Signals, Watching Traffic Lights and Abiding by Traffic Information.

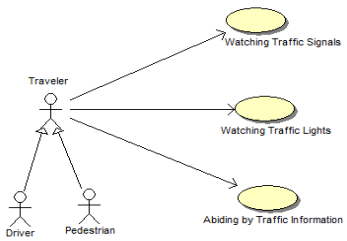

Figure 9. Area Traffic Control system use case diagram

B) Class view is a system aspect describing a main structure and functional process in this system. Designers or developers can use classes in sausage diagram package, product and technology package and organization package to design this system. Furthermore, they may create new classes in these packages for special component representation. In this example, there are various classes representing main structure of this system.

All classes and their relationships can be described by a class diagram as follows:

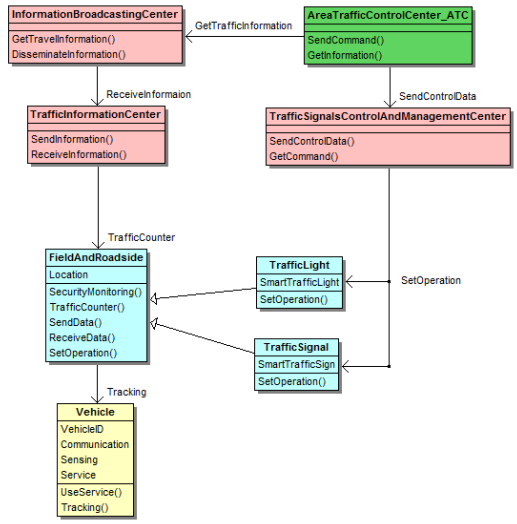

Figure 10. Area Traffic Control system class diagram

After that, we can analyze working process or functional sequence of this system step by step. This analysis should begin with the most important or authoritative class which is AreaTrafficControlCenter class in this case. This working process can be described by sequence diagram and collaboration diagram as follows:

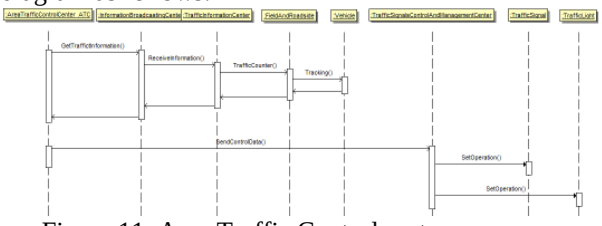

Figure 11. Area Traffic Control system sequence diagram

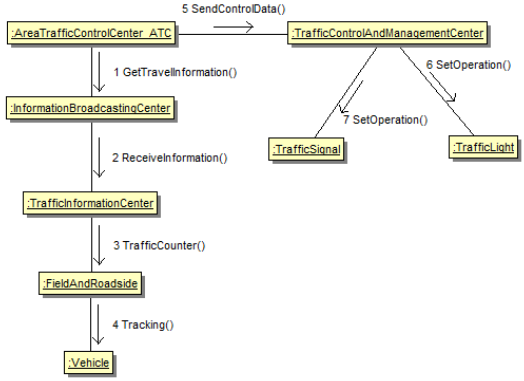

Figure 12. Area Traffic Control system collaboration diagram

C) Component view is a system aspect relevant to analysis and design of software within each structural composition. This design is a model of software modules, their relationships and interfaces with various types of technologies or products. Examples of software module are Task Management, Information Transceiver, Protocol Stack, Processing Library, Information Storage and Retrieval module, etc.

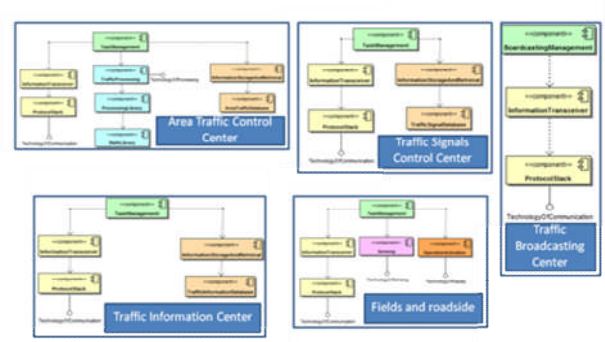

Figure 13. Area Traffic Control system component diagram

D) Deployment view is a system aspect of system installation and physical characteristics. This view is described by deployment diagram. It is composed of deployment nodes, networks and software models within each deployment node. The deployment node is specific name physical entity of system such as Phuket ATC center. The network made from Technology or Product package is a specific type of communication, connection or link between nodes such as Fiber Optic.

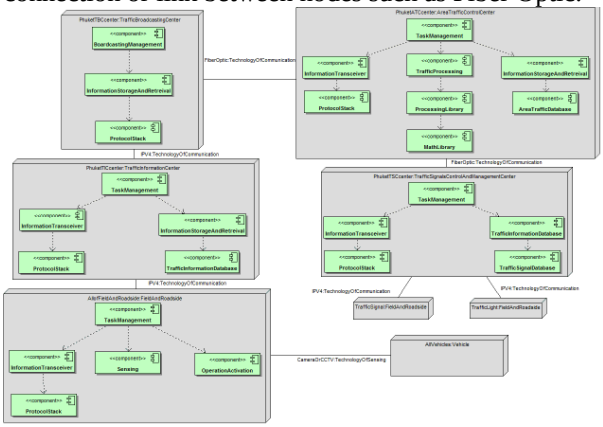

Figure 14. Area Traffic Control system deployment diagram

All views mentioned before belong to Area Traffic Control system. We can design other systems by the same token. If some systems have parts designed in an existing system, we can reduce unnecessary design by reuse (refer to an existing part). For instance, if we are designing ITS Data Mart system, we can refer to Traffic Information Center in Area Traffic Control system for getting traffic information without redesign of information source. After all systems are designed completely, we can design data model for these systems.

5. Assessment

The structure and design of object oriented ITS architecture have main benefits related to solving some existing ITS architecture problems as follows:

1. Flexibility – Object oriented approach has the inheritance property that sub-class can inherit characteristics from super-class. Therefore, we can use polymorphism, inheritance in the form of generalization/specification relationship to support new elements or modifications in the future.
2. Views – This architecture represents systems in various views. They help reduce complexity and size of diagrams. Therefore, designer or developer can understand them and use this architecture and to communicate with participants easily. For example, we can use deployment diagram (in deployment view) to discuss with project executive and to talk to system user with use case diagram. Moreover, component diagram is used for programmer.
3. Reuse – We can use some characteristics of existing super-class to create sub-class without redefine by inheritance.
4. Information hiding – we can use this feature to understand information access of each component by using keyword public or private.
5. Tools – This architecture can be described by various tools such as Bouml (which is used in this paper), Rational Rose, etc.
6. Comprehensive description

Another methodology such as National ITS Architecture can be describe by Turbo architecture tool. This tool has 3 main diagrams that are Subsystem, Interconnect and Flow diagram.

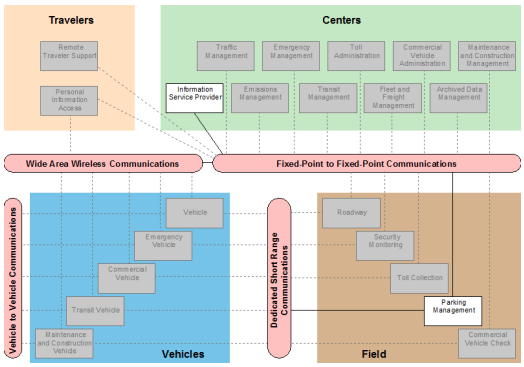

Figure 15. Subsystem diagram

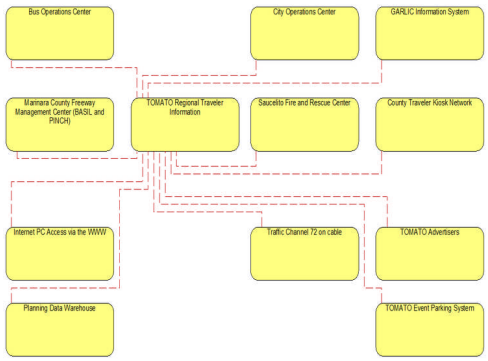

Figure 16. Interconnection diagram

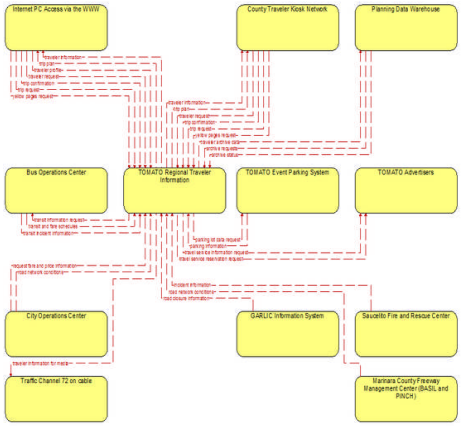

Figure 17. Flow diagram

Subsystem diagram cannot support new main structure in the future. Because, it cannot be add new component. However, class diagram can do by inheritance and polymorphism. Interconnect diagram has not various types of connection but class diagram has. Although Flow diagram can describe flow of data, it cannot represent working process. However, collaboration and sequence diagram are able to show functional sequence. National ITS architecture cannot describe system at the level of software component. Nevertheless, Component and deployment diagram can show software components of each part.

6. Conclusion

This paper describes another approach for design and development of ITS architecture and ITS system by using object oriented process and UML. This process has benefits relevant to reducing some problems of data flow ITS architecture problems. Moreover, this object oriented ITS architecture can help designers or developers to see an overview of all systems, to understand various objectives and diverse composition of system and to know works which are done in implementation. Moreover, we can use this ITS architecture to create software which can generate concrete diagrams and can give information and advice about ITS systems.

Finally, we anticipate that this object oriented ITS architecture can participate in establishment of new real effective ITS systems or projects extremely.

7. Reference

[1] European Commission, "The KAREN European ITS Framework Architecture," http://www.frame-online.net/, 2004.

[2] U.S. Department of Transportation, "The National ITS Architecture Version 5.0," http://itsarch.iteris.com/itsarch/index.htm, 2004.

[3] René Meier, Anthony Harrington and Vinny Cahill, "A Framework for Integrating Existing and Novel Intelligent Transportation Systems," September, 2005.

[4] Charles R. Berger, "Intelligent Transportation Systems Provide Operational Benefits for New York Metropolitan Area Roadways: A Systems Engineering Approach," September. 2007.