SATELLITIC AUTOMATED HIGHWAY SYSTEM: VISUAL MODELLING OF THE REQUIREMENTS USING RATIONAL ROSE AND UML

by

Qoutaiba Hammed Al-Qaysi

A Dissertation Submitted to the Faculty of Graduate Studies
in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major Subject: Civil Engineering

at

DALHOUSIE UNIVERSITY

Halifax, Nova Scotia, July, 2002

0-612-77600-X

Dalhousie University
Faculty of Engineering

Department of Civil Engineering / Ph.D.

The undersigned hereby certify that they have examined and recommend to the faculty of graduate studies for acceptance, the dissertation entitled "Satellitic Automated Highway System: Visual Modelling Of The Requirements Using Rational Rose And UML" by Qoutaiba Hammed Al-Qaysi in partial fulfilment of the requirements for the degree of Doctor Of Philosophy

Dated: July 30, 2002

Supervisor: [redacted]
Dr. Nouman Ali

Co-Supervisor: [redacted]
Dr. Peter Hitchcock

External Examiner: [redacted]
Dr. Amer Shalaby
University of Toronto

Examiners: [redacted]
Dr. Said Easa

Examiners: [redacted]
Dr. Denis Riordan

Dalhousie University
Faculty of Engineering

DATE: 20/09/2002

AUTHOR:	Qoutaiba Hammed Al-Qaysi
TITLE:	Satellitic Automated Highway System: Visual Modelling Of The Requirements Using Rational Rose And UML
MAJOR SUBJECT:	Civil Engineering
DEGREE:	Doctor Of Philosophy
CONVOCATION:	October, 2002

To Nada

TABLE OF CONTENT

CHAPTER7

OBJECT ORIENTED ITSC ARCHITECTURE IN A COMMUNICATION PROSPECTIVE WITHIN THE SAHS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Professor Nouman Ali, for his timely guidance and encouragement. His support made the completion of this dissertation possible. I would like to thank my co-supervisor Professor Peter Hitchcock, Professor Said Easa, and Professor Denis Riordan for their guidance.

The author would also like to thank the faculty and staff in the Department of Civil Engineering, Dalhousie University for their support.

ABSTRACT

Today, advances in the fields of data communication and automated controls make the implementation of automated highway systems more possible. Automated highway systems (AHS) will likely improve the safety, efficiency, and convenience of land travel (Ease, 1999). However, the design of such an automated highway system is still a matter of much debate (Bement et al, 1998). Currently, substantial research effort is being spent on the design of intelligent transportation system (ITS) architectures in the United States, Japan, Europe and Australia.

The programs under research in the regions mentioned above, however, have major drawbacks inherent in the design methodology used. Firstly, most developers use a process-oriented methodology for the design of ITS architectures, which reduces the stability and the flexibility of the architecture to changes in the future. Secondly, none address the need for a backup communication system. This dissertation addresses the need to provide the Intelligent Transportation System of Canada (ITSC) a Stable and flexible architecture based on an object-oriented (OO) methodology and to integrate the backup communication concept with the new ITSC architecture, which will enhance reliability and dependability of systems that will be developed based on the architecture. Partial developments of ITS models are performed according to the requirements of the Intelligent Transportation System of Canada (ITSC) architecture to a level necessary to visualize the hidden communication problems within the architecture. To highlight communication problems, the dissertation utilizes the Unified Modelling Language and Rational Rose as a Case Tool.

Results show that the requirements of the proposed Canadian Intelligent Transportation System architecture will not be fulfilled in the case of communication malfunction, creating an urgent need for a backup communication system. The implementation of such a backup communication system will be achieved by using an Intelligent Digital Modem along with other components such a Stratospheric Satellite. A three-type analysis (scientific, documental, and visual) is performed using an object-oriented methodology in order to prove the need of a backup communication system, our solution for backup communication is adequate, and object-oriented (OO) based architecture provide stability and flexibility. Thus, the new Intelligent Transportation System of Canada (ITSC) is provided a stable and flexible architecture based on an object-oriented (OO) methodology and a backup communication concept with the new ITSC architecture integrated, which will enhance reliability and dependability of systems that will be developed based on the architecture.

Chapter 1

INTRODUCTION AND OVERVIEW

1.1 Motivations of the Research

In pursuing the Intelligent Transportation Systems of Canada (ITSC) Plan (Transport Canada, 1999), Canadian Intelligent Transportation Architecture is developed to present a common framework for developers of the system (ITSC, 2000). The introduction of advanced technology into our vehicle highway system may present a possible approach to achieve our goals in obtaining and maintaining a safe, smart, strategic, sustainable transportation system. The effectiveness of our ITSC architecture as a tool guiding the development of our future Canadian transportation system depends on its stability and flexibility against the changes in user needs, designer needs, environment, and technology. These two quality attributes within the architecture cannot be maintained unless the ITSC architecture is developed based on object-oriented methodology, and provided with a communication backup concept as an integral part of the architecture. The main motivation of this dissertation is to prove that flexibility and stability within the ITSC architecture can be enhanced by adapting an object-oriented methodology and integrating a back-up communication concept in the visualization and documentation of the architecture.

1.2 An Intelligent Transportation Systems Plan for Canada

In a document published in November 1999, Transport Canada touched on the topic of its vision of implementing Intelligent Transportation Systems into our vehicle highway system (Transport Canada, 1999). The document identified the goals and means for such implementation. In order to give Canadians a safe, smart, strategic, and sustainable transportation system suitable for the twenty-first century, a common framework should be developed. The main incentive for this framework is to promote transportation safety, to support trade and tourism, to improve our quality of life, and to sustain strategic investment in transportation. The plan is based on five essential foundations: Partnerships for Knowledge, Developing Canada's ITS Architecture, a Multimodal ITS Research and Development (R&D) Plan, Deployment and Integration of ITS across Canada, and Strengthening Canada's ITS Industry. The plan explains how technology may improve and change our transportation system. It recognizes four key components of the system: the vehicle, the user, the infrastructure, and the communication system:

1.2.1 The Vehicle

ITS may enable us to locate, identify, assess and control vehicles within the system. New technologies may be used to execute various functions within the ITS such as fleet management, navigation and routing advice, toll application, border crossing, and automating the driving activities by automating the control functions on vehicles, which can help improve the safety and efficiency of the transportation system.

1.2.2 The User

ITS will be able to provide the system user traveler information and monitoring capability. Traveler information provides motorists with information and advisories on traffic and infrastructure conditions as well as available services: route guidance in response to changing traffic conditions, monitoring of driver performance and conditions.

1.2.3 The Infrastructure

ITS will enable us to monitor, detect, respond, control, and administer functions related to applications needs, weather and environmental conditions, as well as to traffic conditions data. For example, such data may be used to make intelligent control and activation decisions.

1.2.4 The Communication System

ITS will enable us to integrate communication with our conventional transportation systems. The integration of communication systems and modes is what makes ITS work. The ability to exchange information between the above three functions in the system provides the necessary linkages to allow for the gathering of data that can be processed into intelligence, which can then be used to determine and activate appropriate command and control actions.

1.3 THE CANADIAN INTELLIGENT TRANSPORTATION SYSTEM ARCHITECTURE

The development of the ITSC architecture is the second milestone in the plan of intelligent transportation for Canadians as we mentioned earlier. A recognition of needs by the transportation community in Canada is to ensure the subsystems that are comprising the intelligent transportation not only communicate in a common way effectively, but also that each subsystem has a set of assumptions which are not in conflict with each other (ITSC, 2000). ITSC architecture, responded to 1999 Transport Canada report, published in September 2000. The architecture is developed using process-oriented, and visualized no backup communication. The ITSC architecture was modeled by:

- 35 user services
- 90 sub user services
- 32 subsystem, 71 terminators, and 405 Architecture flow
- 79 market packages

It benefited from the experiences of international architectural efforts of the U.S, Japan, Europe and Australia.

This dissertation is designed to provide a fruitful solution that may help ITSC to maintain the focus on the goals that were set for such development. This effort will present a road map for our research and development of the futuristic transportation system for Canada and the planning efforts in the research and development for innovation required to increase the effectiveness of our transportation system. We may not witness that future, but we will hope to provide a safer and healthier transportation system and mobility to our future generations.

The developers of the ITSC architecture tried to cover as much as possible the main quality attributes areas such as "emergent properties", "usability", "manageability" but was inadequate in improving other areas of quality attributes such as "command and control", reliability, availability, degraded operation, safety, and stability. In order to understand the general direction of this dissertation and its contribution to the knowledge in this particular area we would like to explore these important facts in more detail:

1.3.1 Command and Control

It is not clear from the ITSC architecture what controls what and what triggers the function or the process. There appears to be disarray. The initial analysis of its visualization indicates that the presentation of the Canadian architecture failed in giving a full picture of what should be the sequence of the process or the functionality. We think that this is due one factor, which is the methodological framework that was used in this architecture. This factor is the use of process-oriented development methodology, which was chosen by the developers to create a level of compatibility with the United States National Intelligent Transportation architecture. This is understandable due to the continuity of both national networks but there were inherent problems with the United States National Architecture. This undermined any technical or financial benefit of following the process-oriented concept. The solution to this problem is to have a parallel architecture developed using object-oriented techniques in order to have a better understanding of the system itself by understanding the flow of events and the actor that will trigger the events.

1.3.2 Reliability

The ITSC Architecture explores and visualizes the various designated functions needed in order to satisfy the user need. These functions need an integrated communication between the subsystems in order to execute these functions. In the case of the malfunctioning of this communication, the architecture does not show any backup Communication reliability to cover these set events or these functionality types. This should be the core of any system that will be developed according to the ITSC architecture.

1.3.3 Availability

The ITSC architecture touches on the various functionalities that are required to satisfy the user requirements and needs. These functions should be available when required by the user. In the case of Communication malfunctioning, ITS will not work as the "Intelligent Transportation Plan for Canadians" (Transport Canada, 1999). This will undermine the great effort by the ITSC community and the developers to have this quality attribute affected negatively.

1.3.4 Degraded operation

The graceful degradation operation should be a part of any ITS architecture in order to include the predefined procedures in every design as part of the requirements of every subsystem within the architecture. ITS is a large system, and very few large systems with many separate sub-systems rarely operate at a hundred percent functionality. There are usually errors. In order to reduce the effects of any faulty modes of operation, safety valves must be incorporated within the architecture. The moment these systems are developed and implemented based on a process-oriented architecture, without back-up communication systems, if any malfunction occurs in the system, then most likely, the faults may be traced back to the architecture since the back-up communication system was not mentioned, or it may lead one to the understanding that the back-up communication was not required. The request is for the architecture to be modified.

1.3.5 Safety

Most of the functions and processes that are within the ITSC architecture have a safety impact on the operators or the end users of the system. The malfunction of the subsystem or the communication network will have safety impact. The ITSC architecture does not cover this case. The implementation of a backup communication system within the architecture will reduce the safety risk.

1.3.6 Security

The Integrity of the system itself and the data it uses should be explicitly indicated within the documentation and visualization of the ITSC architecture.

1.3.7 Stability

The ITSC architecture is the base of all future ITS development. This base should be stable by meeting the present and the future needs of the user and the developer.

Due to the time and the resource limitations, this dissertation will focus on finding a solution for two of the problems that the ITSC architecture possesses. The first problem is within the development methodology and the second problem is within the communication structure.

1.4 Research Objectives

The research has two main objectives:

1. To provide the Intelligent Transportation System of Canada (ITSC) a Stable and flexible architecture based on object-oriented (OO) methodology.

2. To integrate the backup communication concept with the new ITSC architecture, which will enhance reliability and dependability of systems that will be developed based on the architecture.

1.5 Methodological Framework

In order to satisfy the objectives laid out in the previous section, the following steps and tasks were performed:

1. The development of a partial object-oriented Canadian Automated Highway System (CAHS) conceptual project development in order to prove the need for backup communication.
2. The development of a partial object-oriented Satellitic Automated Highway System (SAHS) conceptual project development in order to prove, that our solution for backup communication is adequate and object-oriented (OO) based architecture provide stability & flexibility.

1.6 Contribution of this Dissertation

The contributions of this dissertation is summarized by the following:

- Identification of a backup communication problem within ITSC architecture.

- Application of object-oriented methodology to ITSC architecture enhances the stability and the flexibility.
- Analytical, documental and visualization need for a backup communication system.
- Integration of a backup communication concept into ITSC architecture by using object-oriented methodology.
- Analytical, documental and visualization demonstrations for a backup communication solution.

1.7 Scopes and Structure of Dissertation

This dissertation consists of eight chapters:

1. Chapter 1 is an introductory chapter.
2. Chapter 2 presents a review of the literature available on ITS architectures presently under development, worldwide. It also provides historical perspectives and a comparison of methodologies used.
3. Chapter 3 provides an insight to the dilemma of architecture development methodologies and notation. An overview of the process-oriented and object oriented development concepts and their historical background.
4. Chapter 4 covers the methodological framework that is adapted throughout the dissertation.
5. Chapter 5 covers the scientific analysis; documental analysis; and visual analysis of the Canadian Automated Highway System (CAHS) conceptual project in order to prove the need for a backup communication system need.
6. Chapter 6 covers the scientific analysis; documental analysis; and visual analysis of the Satellitic Automated Highway System (SAHS) conceptual project in order to provide proves, that our solution for backup communication is adequate with an object-oriented (OO) based architecture providing stability and flexibility.
7. Chapter 7 provides a discussion about an object-oriented ITSC communication architecture prospective within SAHS. An overview of the role of an Intelligent Digital Modem (IDM) in the SAHS communication system.

8. Chapter 8 includes the conclusions drawn from this dissertation and recommendations for future research and development.

Chapter 2

REVIEW OF ITS ARCHITECTURE WORLD EXPERIENCE: METHODOLOGY AND VISUALIZATION PERSPECTIVES

2.1 Overview of ITS Architecture

The Intelligent Transportation Architecture for every nation that has a ITS program is the basis for developing and implementing an ITS technology. ITS architecture is the basis for finding areas where it can standardize its specification that is in contact with ITS technology. In order to achieve that standard and specification, the ITS Architecture must be stable enough for developers to perform present and future developments. Stable architecture without deficiency means business opportunities for the nation that developed the architecture. Other nations may like to develop their own systems based on the stable architecture, its standards and specifications. ITS programs in the world are competing for this opportunity. Although, there is a working relationship between these nations, there is national competition in the international market. At this point, we will discuss briefly the current developments of the ITS programs worldwide. We will focus on both the strengths and weaknesses of the ITS programs of the leading nations.

2.2 ITS America Architecture

The United State National ITS Architecture (ITSA, 2002) was the first ITS program in the world, it led to the establishment of standardization and specification activities in the international arenas in relation to ITS technology. Their process-oriented methodology was widely adopted by other nations. Over five years of effort has produced the functions that are required for ITS programs, the physical entities or subsystems where these functions reside, and the information flows that connect these functions and physical subsystems together into an integrated system. The illustration given in figure 2.1 visualizes such views.

The United States National ITS architecture supports seven-user bundles:

A. Travel And Traffic Management
B. Public Transportation Management
C. Electronic Payment
D. Commercial Vehicle Operations
E. Emergency Management
F. Advanced Vehicle Safety Systems
G. Information Management

In the national United States architecture, user services play the role of basic requirements for system development and they are what architecture may provide to the user. The details of the user services are given as follows:

A. Travel and Traffic Management (User Services)

1. Pre-trip Travel Information
2. En-route Driver Information
3. Route Guidance
4. Ride Matching And Reservation
5. Traveler Services Information
6. Traffic Control
7. Incident Management
8. Travel Demand Management
9. Emissions Testing and Mitigation
10. Highway-rail Intersection

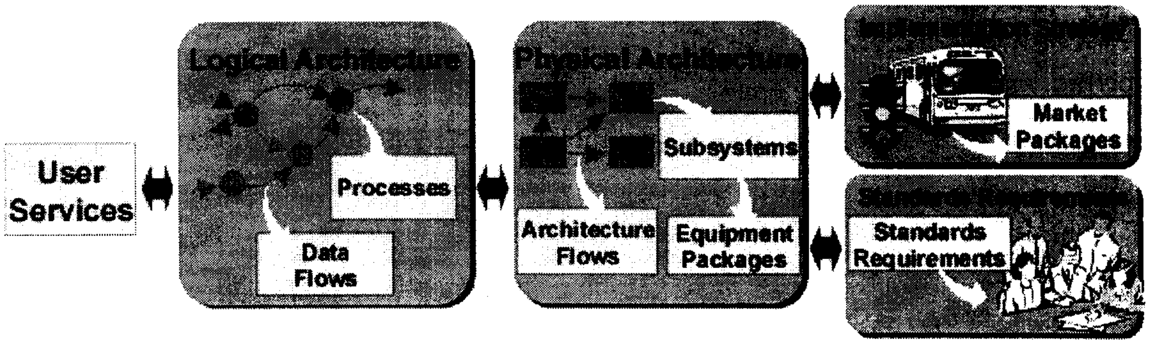

Figure 2.1 United States National ITS
Architecture Views (ITSA, 2002)

B. Public Transportation Management (User Services)

1. Public Transportation Management
2. En route Transit Information
3. Personalized Public Transit
4. Public Travel Security

C. Electronic Payment (User Services)

1. Electronic Payment Services

D. Commercial Vehicle Operations (User Services)

1. Commercial Vehicle Electronic Clearance
2. Automated Roadside Safety Inspection
3. On-board Safety Monitoring
4. Commercial Vehicle Administrative Processes
5. Hazardous Material Incident Response
6. Commercial Fleet Management

E. Emergency Management (User Services)

1. Emergency Notification And Personal Security
2. Emergency Vehicle Management

E. Advanced Vehicle Safety Systems (User Services)

1. Longitudinal Collision Avoidance
2. Lateral Collision Avoidance
3. Intersection Collision Avoidance
4. Vision Enhancement For Crash Avoidance
5. Safety Readiness
6. Pre-crash Restraint Deployment

7. Automated Vehicle Operation

F. Information Management (User Service Bundle)

1. Archived Data Function

An example of the United States ITS Logical Architecture visualization of a process using process oriented development methodology is shown in Figure 2.2.

The logical Architecture defines the processes and the data required to fulfill the requirements of the user. The main functions that are supported by the United States National Logical Architecture are as listed below.

1. Manage Traffic
2. Manage Commercial Vehicles
3. Provide Vehicle Monitoring and Control
4. Manage Transit
5. Manage Emergency Services
6. Provide Driver and Traveler Services
7. Provide Electronic Payment Services
8. Manage Archived Data

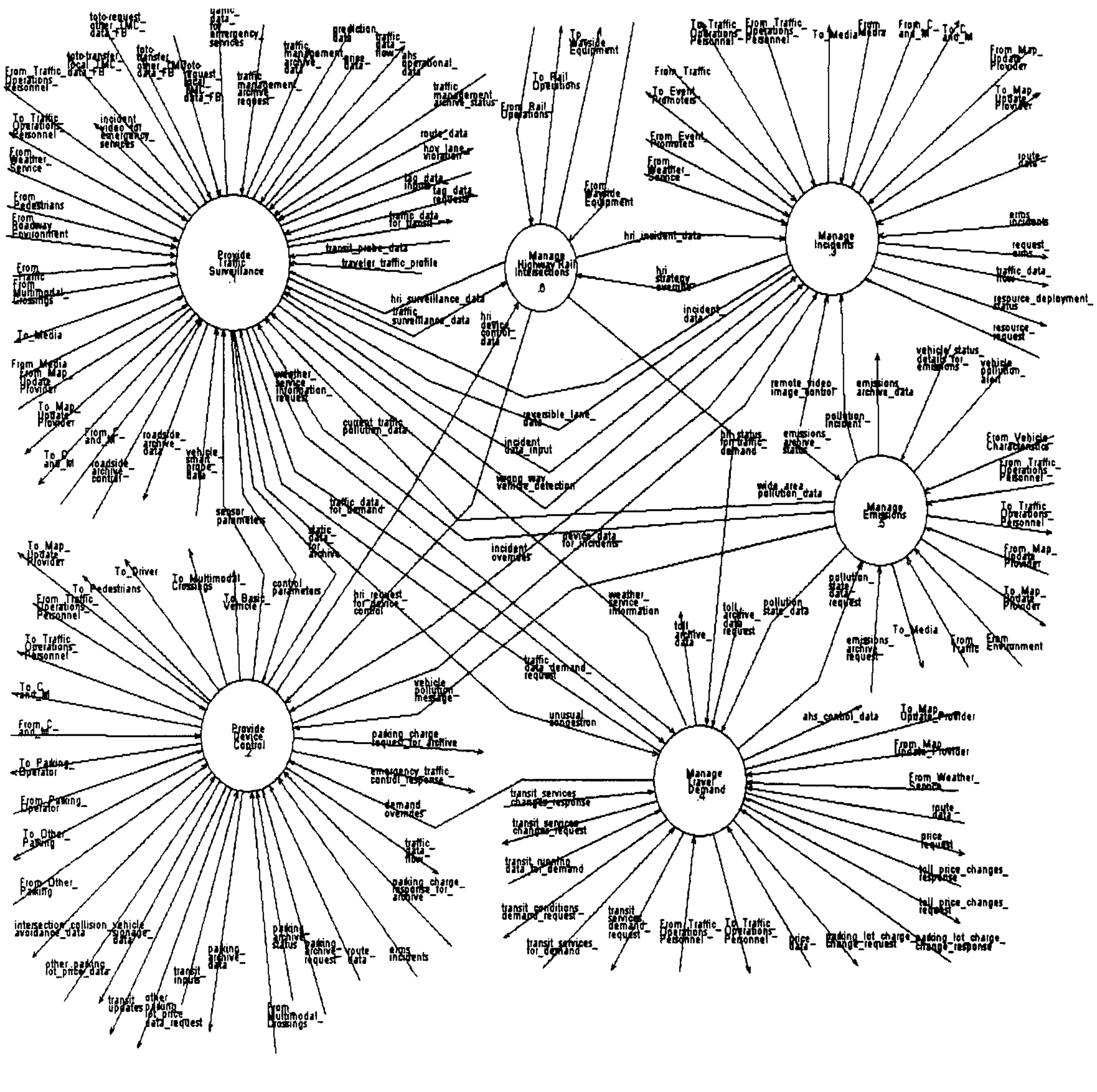

Figure 2.2 United States National ITS Logical Architecture visualization of Manage Traffic (ITSA, 2002)

The United States Nation Physical Architecture support the following subsystems:

A. Center

1. Commercial Vehicle Administration (cvas)
2. Emergency Management (em)
3. Emissions Management (emms)
4. Fleet and Freight Management (fms)
5. Information Service Provider (isp)
6. Toll Administration (tas)
7. Traffic Management (tms)
8. Transit Management (trms)

B. Roadside

1. Commercial Vehicle Check (cvcs)
2. Parking Management (pms)
3. Roadway Subsystem (rs)
4. Toll Collection (tcs)

C. Traveler

1. Personal Information Access (pias)
2. Remote Traveler Support (rts)

D. Vehicle

1. Commercial Vehicle Subsystem (cvs)
2. Emergency Vehicle Subsystem (evs)
3. Transit Vehicle Subsystem (trvs)
4. Vehicle (vs)

See Figure 2.3 for the visualizations and the physical view of the sub systems of the US National ITS Architecture.

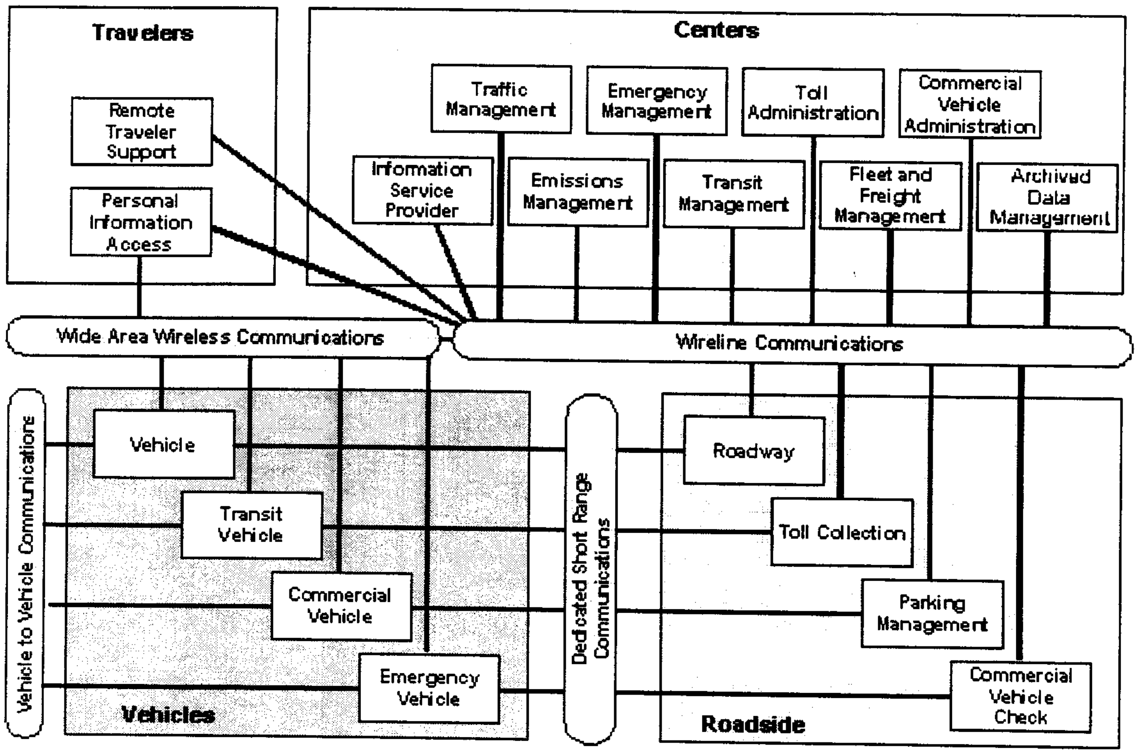

Figure 2.3 United States ITS National Physical Architecture visualization using a process-oriented development methodology (ITSA, 2002)

The United States National Intelligent Transportation Architecture has two weaknesses. It uses a process-oriented development and diagrammatic (visual) explanations and documentation. Such diagrams do not clearly delineate the beginnings, the middle and the endings of the processes. The second weakness is that there is a lack of a backup communication system lending to an unstable, and inflexible architecture. Since there is no back-up communication system, it is unreliable for users.

2.3 JAPAN NATIONAL ITS ARCHITECTURE

In 1996, the Japanese government agencies have published a document that describes the long-term vision for achieving Japan's ITS goals. Based on this document, the Japanese ITS architecture was developed. The Japanese where the first to develop an ITS architecture, using an object-oriented development methodology as a way to provide flexibility toward the changing needs of users and developers (ITSJ, 2001).

This development methodology made Japan's ITS one of the most stable architectures in the world. Japan's architecture identifies nine development areas, twenty-one user services, fifty-six specific user services and one hundred and seventy-two specific user sub services. The lists of their user services are as follows:

1. Provision of route guidance traffic information
2. Provision of destination-related information
3. Electronic toll collection
4. Provision of driving and road conditions information
5. Danger warning
6. Assistance for driving
7. Automated highway systems
8. Optimization of traffic flow
9. Provision of traffic restriction information in case of incident
10. Improvement of maintenance operations

11. Management of specially permitted commercial vehicles
12. Provision of roadway hazard information
13. Provision of public transport information
14. Assistance for public transport operations and operations management
15. Assistance for commercial vehicle operations management
16. Automated platooning of commercial vehicles
17. Pedestrian route guidance
18. Vehicle-pedestrian accident avoidance
19. Automated emergency notification
20. Route guidance for emergency vehicles and support for relief activities
21. Utilization of information in the advanced information and telecommunications society

2.3.1 Logical Architecture

Japans logical architecture consists of two main parts: the Informational Model and the Control Model:

3.3.1.1 Information Model

An information model is a diagrammatic representation of the relationships between the functions and the information to be handled by those functions. An example of the Information Model is given in figure 2.4. The Model represents a class diagram that gives the relation between the data and the function that operated on that data. The seven types of attributes are as follows: spot, routes, roadway, moving body, schedule, operational body, and external institute.

2.3.1.2 Control Model

The control model is the result of modeling the relationships among the procedures and functions. The objectives of determining the control model are to clarify the functions in the subsystems such as Centers, Roads and Vehicles in constructing physical class diagram. An Example for the Control Model is given in Figure 2.5.

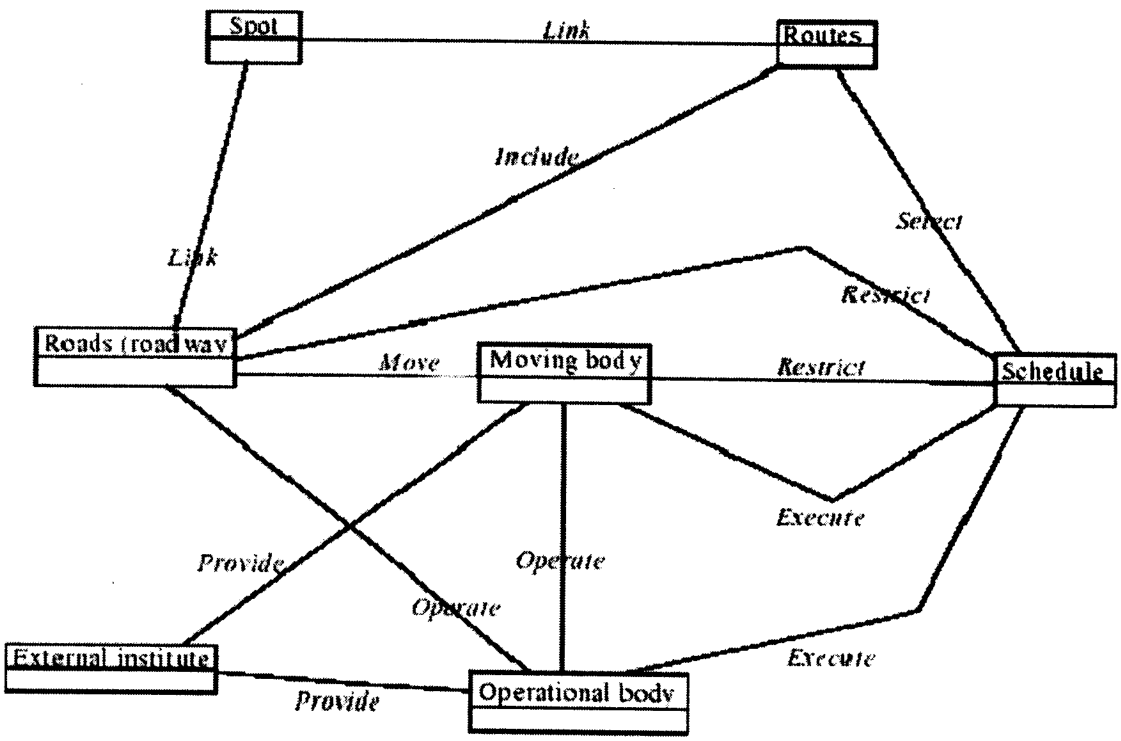

Figure 2.4 Japan National ITS Logical Information Model Architecture (ITSJ, 2001)

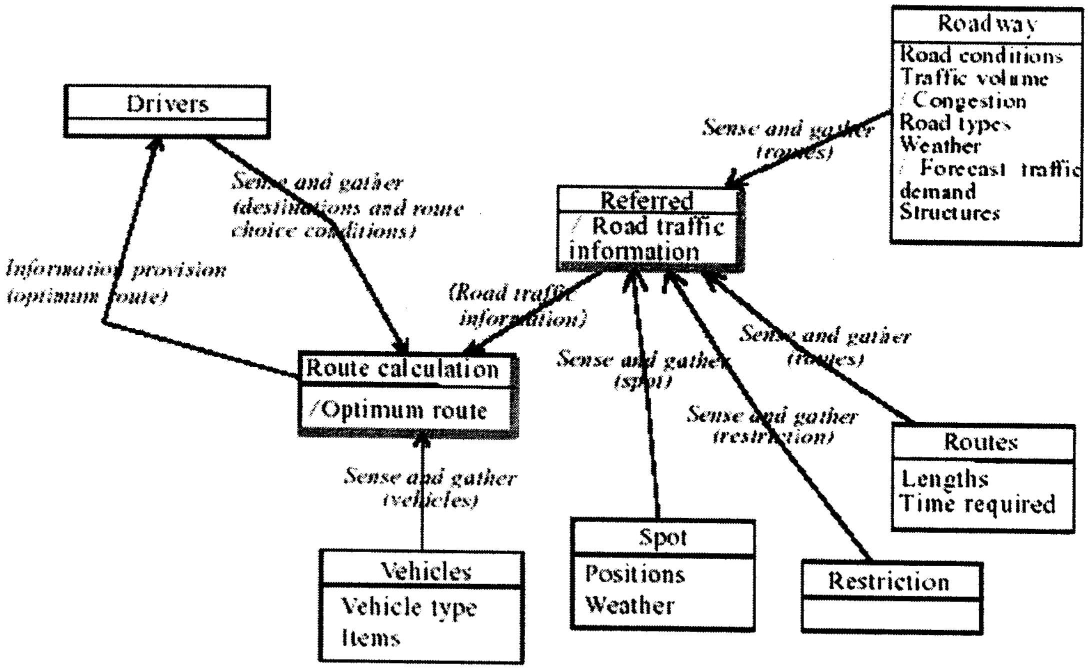

Figure 2.5 Japan National ITS Logical Control Model Architecture (ITSJ, 2001)

2.3.2 Physical Architecture

The Japanese ITS Physical Architecture is a model of the overall image of the system to visualize the systems and sub-systems of the architecture. It arranges the vehicles, roads, centers; It integrates the physical entities of the entire system with regard to the combination of the functions determined by the Logical Architecture and the information handled by those functions. The Japanese physical architecture is given in Figure2.6. Although, Japan's ITS architecture is one of the most stable and flexible architectures in the world, it has flaws as all other ITS programs do. Japan's ITS program was one of the first programs that realized the object-oriented process is the only way to meet the ever-changing needs of both the users and developers. This allows the ITS Japanese architectural users and developers to change the information, functions, and data to meet the specific needs of time and technology. This advantage gives the Japanese users and developers the upper hand in the competitive race of standardization and specification of very stable and flexible architecture. However, The Japanese ITS program has failed to realize one of the most important safety concerns which all the ITS program should have. A backup communication system is what worldwide ITS programs need for enhanced reliability.

2.4 EUROPEAN NATIONAL ITS ARCHITECTURE

The Keystone Architecture Required for European Networks (KAREN) started on April 1, 1998, and ended in the spring, 2000. It was the European ITS system architecture effort, requested by the High Level Group on road transport telematics, approved by the European Council of Ministers, and funded as part of the fourth Framework Program (ITSE, 2000). The KAREN Functional Architecture describes the European ITS structure in terms of the functions that can implement the KAREN list of User Needs, and how these functions link to the outside world (the terminators), in particular to the users of the ITS system. It also shows the data that are used in the ITS architecture. At its highest level, the KAREN Functional Architecture consists of a number of Functional Areas:

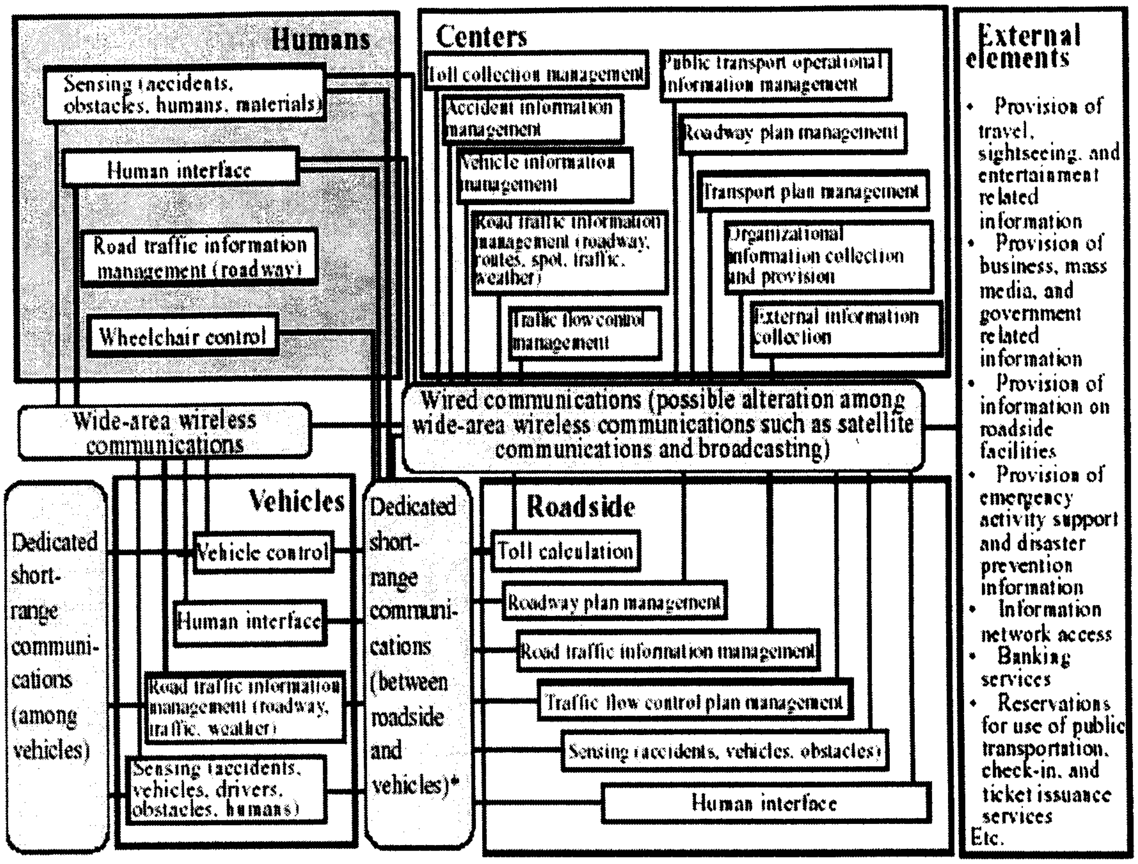

Figure 2.6 Japan National ITS Physical Architecture (ITSJ, 2001)

A1-provide electronic payment facilities

A2-provide safety and emergency facilities

A3-manage traffic

A4-manage public transport operations

A5-provide advanced driving facilities

A6-provide traveler journey assistance

A7-provide support for law enforcement

A8-manage freight and fleet operations.

Figures 2.7 illustrate the eight functional types supported by KAREN. The physical architecture given by KAREN is set of examples utilizing the functional decomposition. Figure 2.8 demonstrates an example for part of the physical architecture. The European National ITS architecture program does not offer stable architecture for its users and developers because the methodology used to develop the ITS architecture is process-oriented. It is very difficult if not impossible to change a small part of the architecture without changing the whole structure of the architecture. Furthermore, the European ITS architecture has as many deficits as many others ITS programs in the world do, with regard to lack of back-up communications, thus adversely affecting safety. In other words, the

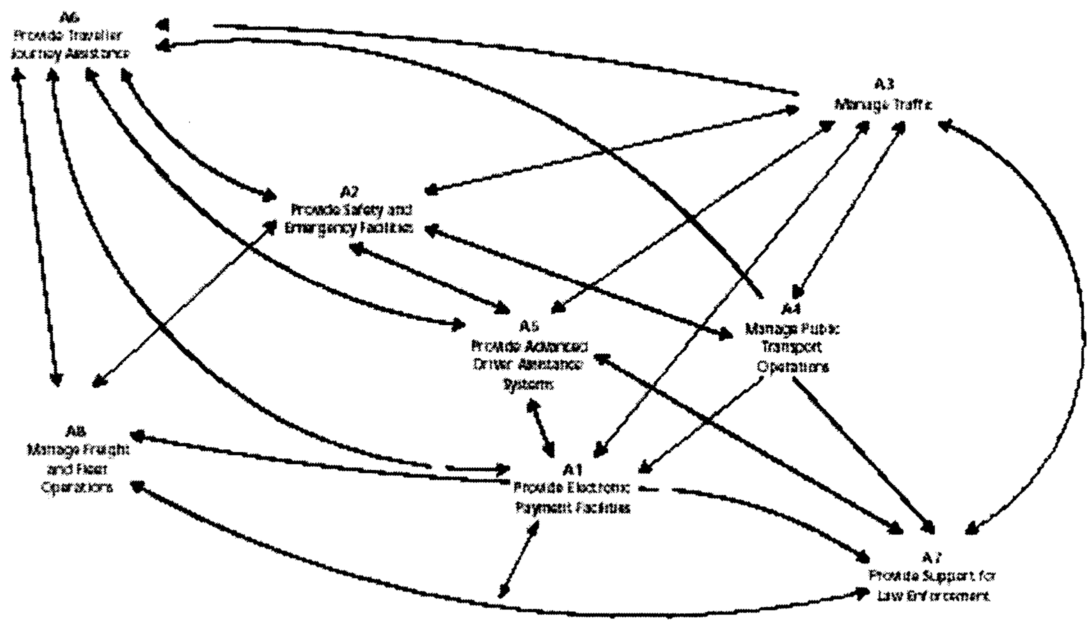

Figure 2.7 Europe National ITS Function that are
supported by KAREN (ITSE, 2000)

The physical architecture given by KAREN is set of examples utilizing the functional decomposition. Figure 2.8 demonstrates an example for part of the physical architecture. The European National ITS architecture program does not offer a stable architecture for its users and developers because the methodology used to develop the ITS architecture is process-oriented. It is very difficult if not impossible to change a small part of the architecture without changing the whole structure of the architecture. Furthermore, the European ITS architecture has as many deficits as many others ITS programs in the world do, with regard to lack of back-up communications, thus adversely affecting safety. In other words, the European ITS architecture does not offer backup communication functions in case of a malfunction to the backbone communication functions.

European ITS architecture does not offer backup communication functions in case of a malfunction to the backbone communication functions.

2.5 AUSTRALIA NATIONAL ITS ARCHITECTURE

In 1998 The Intelligent Transportation Systems Australia (ITSAU) published a report titled "Intelligent Transport Solutions for Australia: Summary Report" recommending: "further effort on standards and agreement on a national architecture would facilitate the deployment of ITS applications". In response to this request, effort started to develop a multi-modal National ITS Reference Architecture (ITSAu, 1999), summarizing ITS uses, systems, communication linkages and actors that will improve the future development and deployment of ITS services within Australia by providing a framework for the development of standards, acting as an educational tool, promoting the integration of systems and increasing the competitiveness of the ITS industry. The Australians identified several stages in the architecture development process, these include development of a:

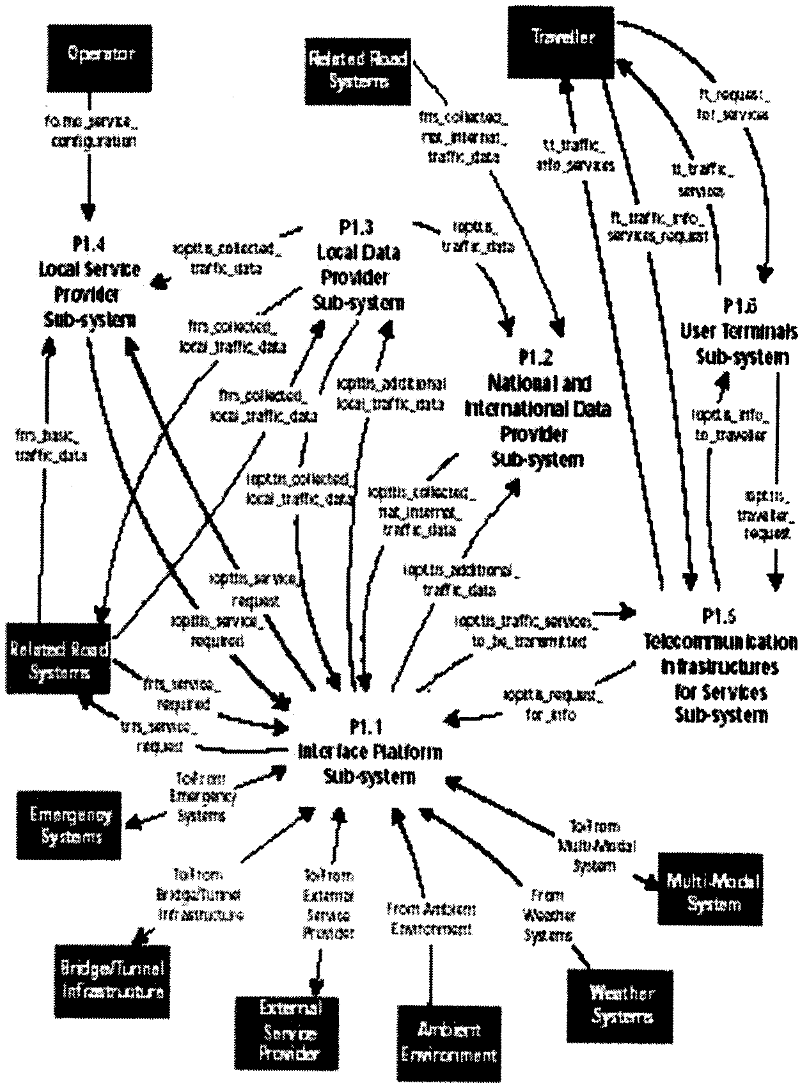

Figure 2.8 European ITS Physical Architecture as given by KAREN (ITSE, 2000)

1. Reference architecture
2. Logical architecture
3. Physical architecture

The first stage was finalized in 1999.

The development methodology was object-oriented to develop the Unified Modeling Language (UML) views of the system. This methodology involves:

1. Use of Case Diagrams
2. Class Diagrams
3. Package Diagrams
4. Sequence (or Interaction) Diagrams.

The Australian National ITS Architecture supports the following user services:

A. Traveler Information

1. Pre-trip Information
2. On-trip Driver Information
3. On-trip Public Transport Information
4. Personal Information Services
5. Route Guidance and Navigation
6. Transportation Planning Support

B. Traffic Management

1. Traffic Control

2. Incident Management
3. Demand Management
4. Policing/Enforcing Traffic Regulations
5. Infrastructure Maintenance Management

C. Vehicle

1. Vision Enhancement
2. Automated Vehicle Operation
3. Longitudinal Collision Avoidance
4. Lateral Collision Avoidance
5. Safety Readiness
6. Pre-crash Restraint Deployment

D. Freight and Fleet

1. Commercial Vehicle Pre-clearance
2. Commercial Vehicle Administrative Processes
3. Automated Roadside Safety Inspection
4. Commercial Vehicle On-board Safety Monitoring
5. Commercial Vehicle Fleet Management

E. Public Transport

1. Public Transport Management
2. Demand Responsive Transport Management
3. Shared Transport Management

F. Emergency

1. Emergency Notification and Personal Security
2. Emergency Vehicle Management
3. Hazardous Materials and Incident Notification

A. Electronic Payment

1. Electronic Financial Transactions

B. Safety

1. Public Travel Security
2. Safety Enhancement for Vulnerable Road Users
3. Intelligent Junctions and Links

Examples of Australian package diagrams are: use case diagram, sequence diagram and class operation diagrams are shown in Figures 2.9 - 2.12. The Australian ITS program is in the preliminary stage. However, the Australians are in the right direction with regard to developing a very stable and flexible architecture by using an object-oriented methodology, while learning from the experience of the other nations that have developed their ITS architecture with regard to a process-oriented methodology.

2.6 AUTOMATED HIGHWAY SYSTEM CONCEPTS WITHIN THE WORLDS ARCHITECTURE

Automated Highway System refers to development concepts for a system that will automate " vehicles (cars, buses, and trucks) operating on special lanes [networks], where drivers totally relinquish vehicle control to on-board and infrastructure computers. Once the computers take over, drivers can take both hands off the steering wheel, both feet off the pedals, and both eyes off the road. The primary goal of AHS is to improve the capacity and safety of transportation systems in both urban and rural area." The Automated highway System is believed to produce the most noticeable improvement in efficiency and safety between the other ITS bundle (Easa, 1999). The concept of automated highway systems was first introduced in the US by General Motors in General Motors Futurama, displayed at the 1939-1940 New York Worlds Fair. The US Intermodal Surface Transportation Efficiency Act (ISTEA), which was passed in 1991, called for the development of an AHS prototype. In 1994 the National Automated Highway System Consortium (NAHSC) was formed. In California, the highway automation was successfully demonstrated in 1997 (Bement, 1998). The two partial development examples in this study to evaluate the ITSC architecture were based on this concept to promote the establishment of such research program in Canada.

A Fully Automated Highway System has pros and cons that were the basis of debates for some time, in spite of the proven pros concerning its ability to cause a positive impact on: highway capacity; travel time; safety; adverse driving behavior; predictable traffic

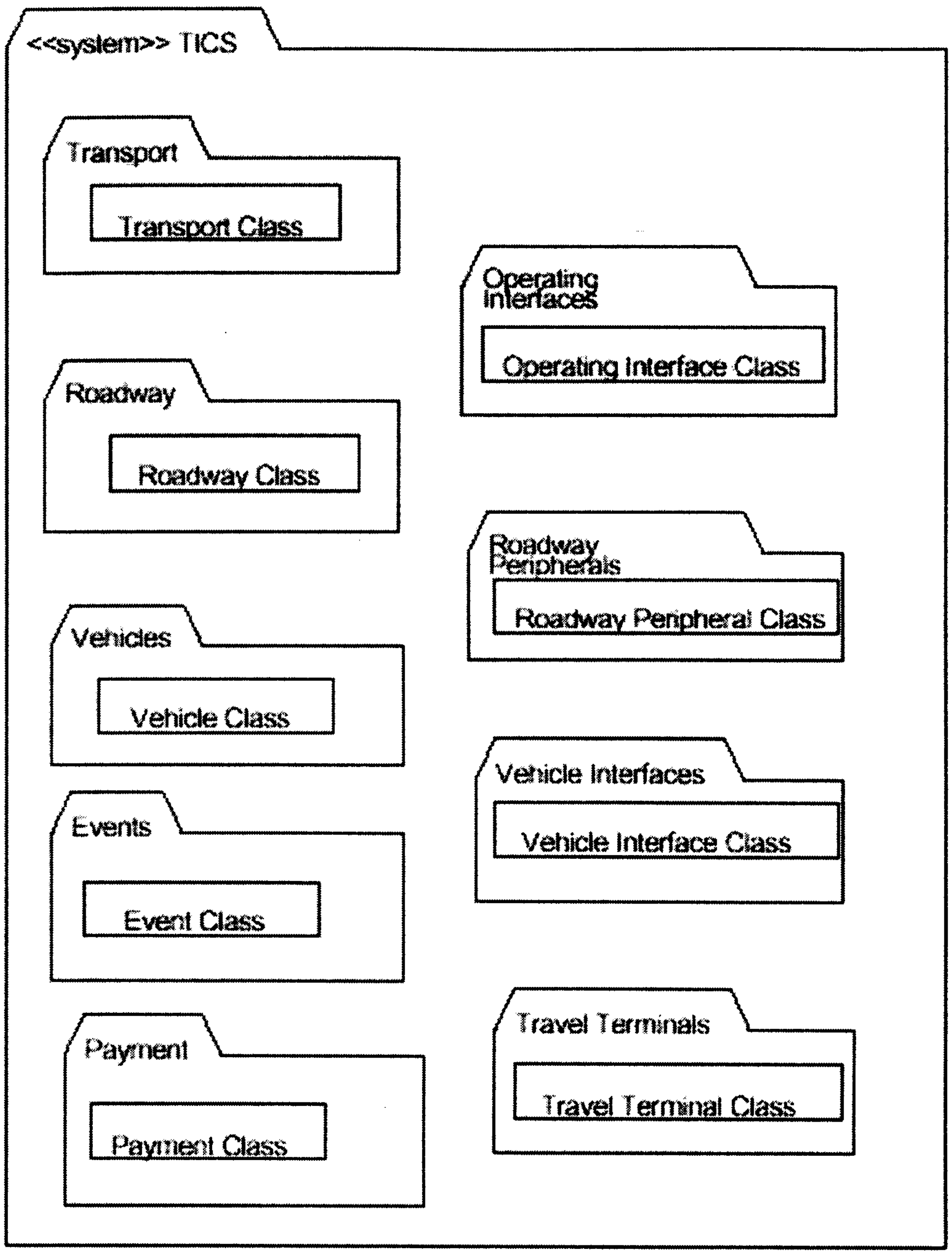

Figure 2.9 Australia National ITS Packages
Model and tier related classes (ITSAu, 1999)

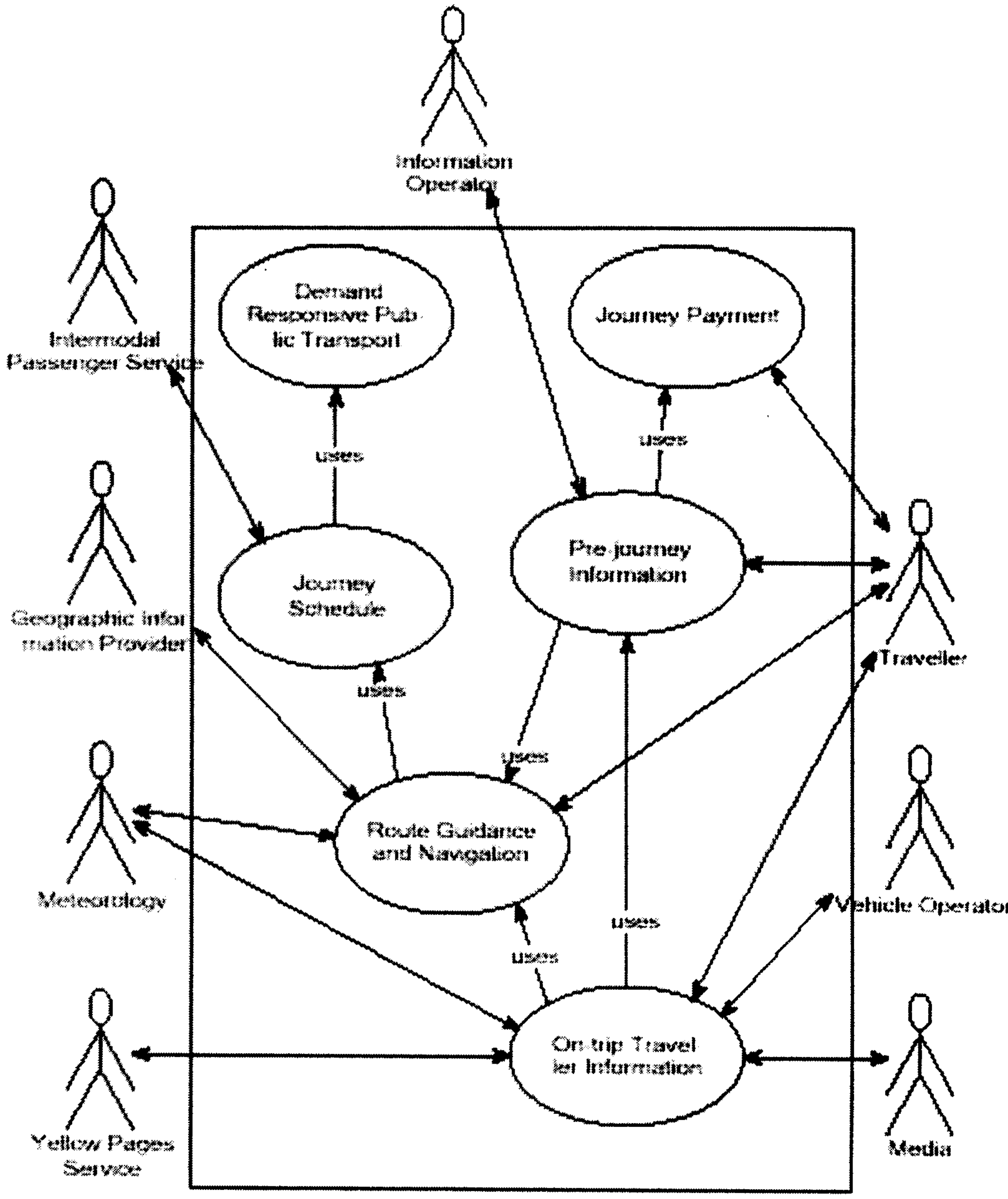

Figure 2.10 Australia National ITS Visualization
of Use Case Diagram (ITSAu, 1999)

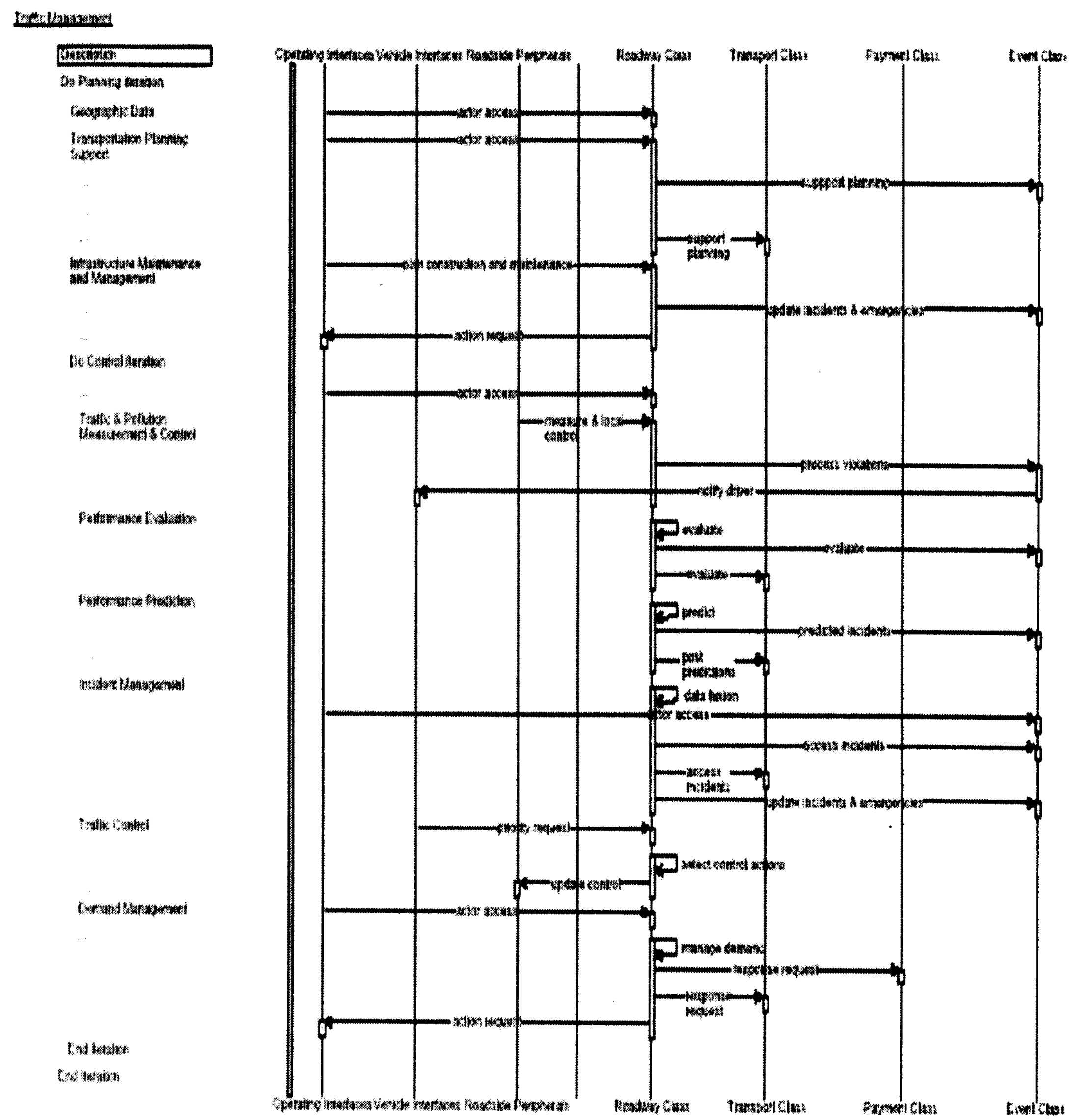

Figure 2.11 Australia National ITS visualization of a Sequence Diagram (ITSAu, 1999)

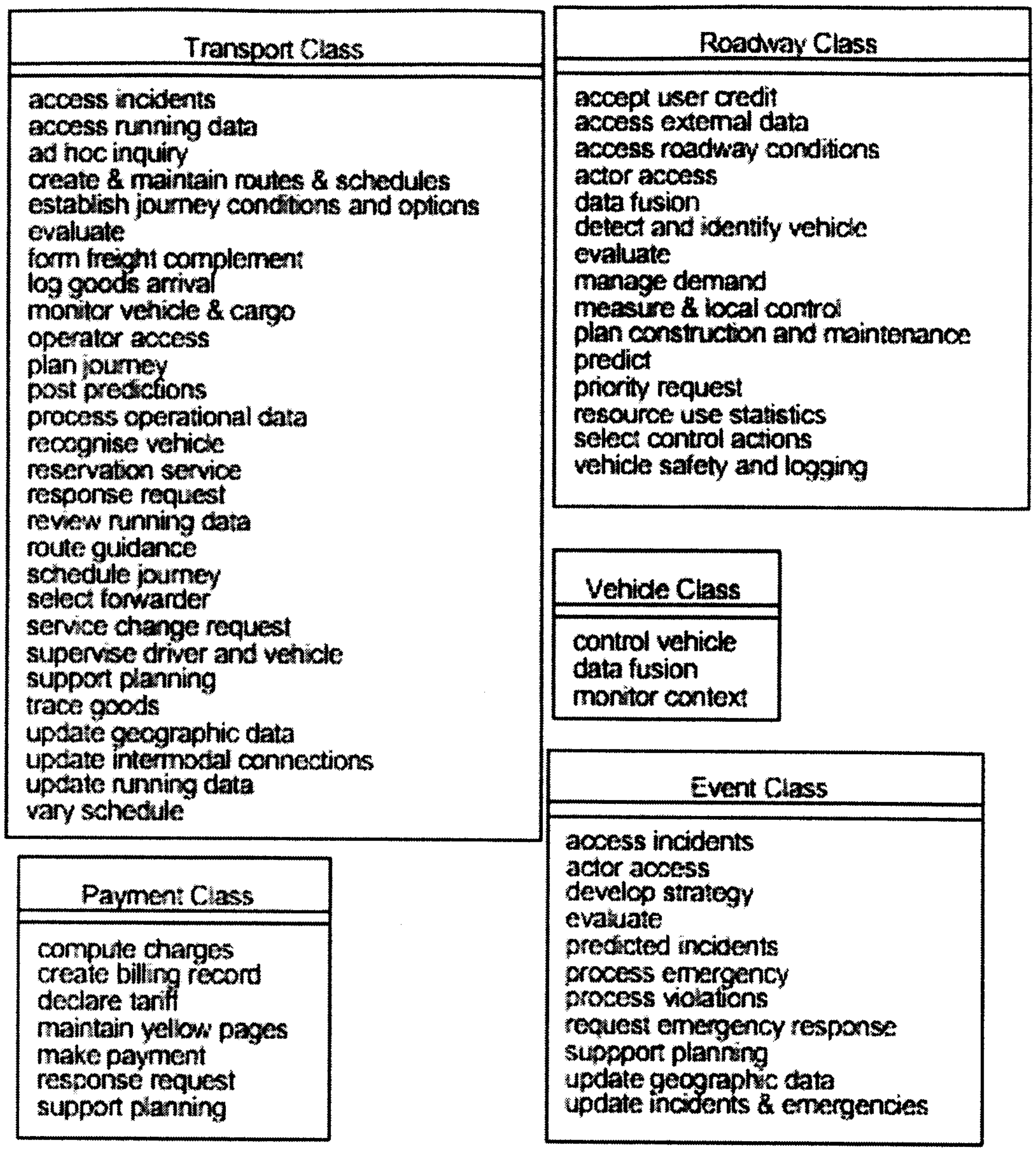

Figure 2.12 Australia National ITS Architecture
Classes and their functions (ITSAu, 1999)

condition and travel time; exhaust emission and energy consumptions. The cons of having fully automated highway system are in the area of stretch of concept, long-term aspect, vehicle costs, reliability and safety, and social equity (Shladover, 1998). The Transportation Research Board special report 253 by the Committee for a Review of the National Automated Highway System Consortium Research reported the need for research in the area of liability and communication (Bement, 1998).

In the precursor system analyses of Automated Highway System final report an elaboration on the two main concepts of integrating automated highway into our freeway system, the Shared and Dedicated space concepts. In the first concept, both automated and manual vehicles use common right-of-way and common on and off ramps. The second concepts are based on the need to build new-dedicated lanes for automated vehicles served by exclusive on and off ramps. Examples of the two concepts are illustrated in Figure 2.13 & 2.14 (US DOT, 1997).

2.7 Summary

This chapter is an overview of attempts by leading nations in the field of ITS and ITS architectural development. These nations' experience and findings were the basis for the ITSC architecture. This fact leads to the inheritance of their strengths and weaknesses, especially in the case of the United States national ITS architecture. Their development methodologies and their adaptation of a backup communication system within their ITS architecture documentation and visualization are summarized in Table 2.1.

Table 2.1 Summary of development methodology and backup communication implementation in leading nations of ITS and ITS architecture.

Nation	Development Methodology	Backup Communication
US	Process-Oriented	No
Japan	Object-Oriented	No
Europe	Process-Oriented	No
Australia	Object-Oriented	Unclear

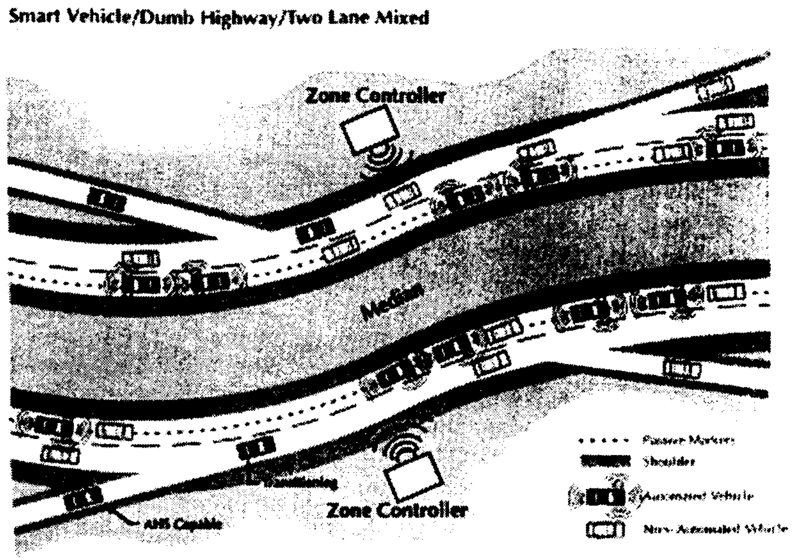

Figure 2.13 shared space concept to integrate AHS with our freeway system (US DOT, 1997)

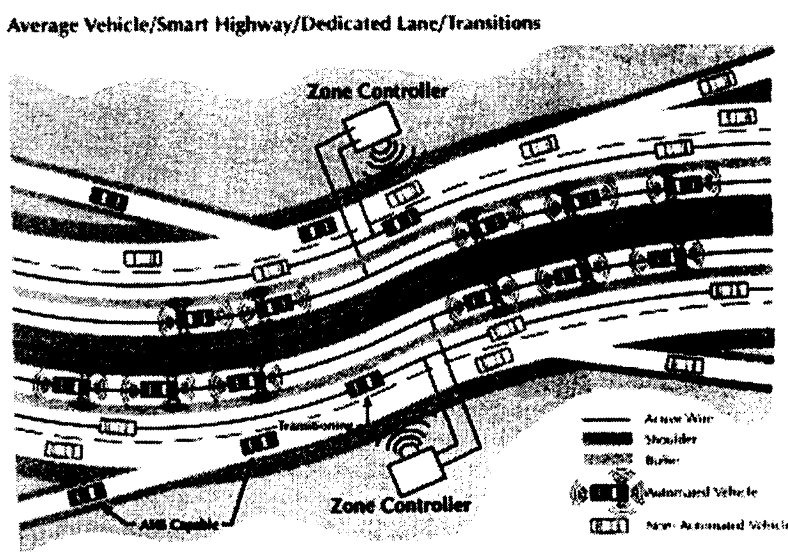

Figure 2.14 dedicated space concept for integrating AHS with our freeway system (US DOT, 1997)

Chapter 3

PROCESS-ORIENTED OR OBJECT-ORIENTED ARCHITECTURE THE SYSTEM MODELING DILEMMA

The main objective of this chapter is to introduce the two basic methodologies that are available to develop system architecture. An overview of the backgrounds and concepts, notation, and offering is provided. In order to provide ITSC a stable and flexible architecture that the current architecture lacks, object-oriented methodology was selected for reasons that will be explored in this chapter. Object-oriented is the means that enable us to integrate the backup communication concept with the new ITSC architecture, which will enhance reliability and dependability of systems that will be developed based on the architecture.

3.1 PROCESS-ORIENTED APPROACH

The process-oriented methodology was chosen to provide the foundation for the structural development concept behind the ITSC architecture. The technique in perception is that the developers define what the system should do before deciding how it should do it. The new system architecture specification evolved from a series of data flow diagrams DFD. The concept itself is not new, beginning in the early 1960's, although its acceptability was after the publication of a paper by Stevens and others in IBM System Journal (Stevens et al., 1974).

The process-oriented methodology starts with the users need and system expectations. The documentation of this stage covers the unsatisfied desires (users need), the proposed solution (System concept) and the main feature of the proposed solution. Then the documentation of the users need is organized in hierarchical manner. The users need forms the basis of the user requirements, which are analysed to extract the process (functional) requirements. The terminator that is documented in the physical architecture and the context of the architecture are developed in this stage.

The development of the logical architecture for ITSC starts with process decomposition. The primary objective of this type of decomposition is to create hierarchical layers of processes that can be broken to a "primitive " level; construct a levelled set of Data Flow Diagrams using concepts based on Gane & Sarson and Yourdon/DeMarco ((Gane and Sarson, 1979); (De Marco, 1982); (Yourdon, 1991)); define specifications for low–level processes definitions; and decomposition of diagrams to show the hierarchical structure of the ITSC logical levelled set of diagrams. In a study conducted by The Software Engineering Institute of Carnegie Mellon University, Bchmann and others reported that the logical view should take into account requirements such as performance and system availability. It should address concurrency, distribution, and fault tolerance (Bchmann et al., 2000). The ITSC architecture did not clearly address these aspects. The physical architecture is last stop in the process-oriented architecture development methodology. It represents the grouping of processes into physical units, locations, and the communication path (Hately and Pirbhai, 1987). The groups then reorganized to form marketable units. Issues such as simplicity in system development and maintenance, localization of change, maximum opportunity for re-use of components, good management of variability, protection of data from illegal access, and better management of complexity can address better by having an object-oriented alternative (Jesty et al., 1998).

In main object of the ITSC architecture is to provide a vehicle for communicating the ITS to interested stakeholder and developers in Canada. In order to achieve this target, the quality attributes of system architecture should be built within the documentation and visualization of the architecture (Barbacci et al., 1995). The study is designed to address two quality attributes concerns, the flexibility and stability of the architecture, plus the solution to these concerns by adapting an object-oriented methodology along with a backup communication system clearly mentioned in the documentation and visualization of the ITSC architecture.

3.2 ORIGINS OF THE OBJECT-ORIENTED PARADIGM

The origins of Object-oriented concepts can be rooted the development of object and frame concepts within the emerging software engineering communities in the sixties and seventies. The work of Ole-Johan Dahl and Kristen Nygaard of Norway created the influential work on an object-oriented concept with their introduction of Simula67 in 1967 (Dahl And Nygaard, 1966). As the name implies, Simula67 was generally used for simulation modeling and proved to be a significant influence on later object-oriented languages and concepts. Smalltalk, developed at XEROX in Palo Alto in the 1970s The software engineering community in the artificial intelligent side developed a twin concept to object called frames, which appears in the pioneer work of Marvin Minsky around 1975 (Minsky, 1975).

Since then, many methodologists participate in developing the concept we will cover some of their contributions. Grady Booch gave a good analogy to the essence of the object-oriented approach. His work covers the notation of the method, discusses analysis and design strategies (Booch, 1994). Peter Coad& Edward Yourdon in their book titled "*Object-Oriented Analysis*" introduced object-oriented analysis in a clear manner and introduced an object-oriented analysis (OOA) methodology consisting of five steps: identifying classes and objects, identifying structures, identifying subjects, defining attributes, and defining services. All of these items are combined into an "object diagram," which resembles a dataflow diagram combined with an entity-relationship diagram(Coad and Yourdon, 1991). Jacobson et al provided in depth introduction to the object-oriented technique. It presents object-oriented software engineering (OOSE) as a new methodology that emphasizes the interaction of the user with the system and emphasizes the problem domain (Jacobson et al , 1992). In an attempt to provide a smooth transition to object-oriented techniques from procedure-oriented technique Rumbaugh et al proposed a complete methodology called the object modeling technique (OMT), which covers analysis, design, and implementation. The authors contrast their OMT with structured analysis and design and with Jackson's structured development method (Rumbaugh et al, 1991). The object-oriented approached were defined by Wirfs-Brock et al and provided a complete coverage of object-oriented principles. They emphasize a responsibility-driven viewpoint of analysis and design that emphasizes clients and servers. They also suggest that quality of design can be measured by counts of the number of classes, the number of subsystems, the number of contracts per class, and the number of abstract classes (Wirfs-Brock et al, 1991). In an attempt to incorporate the methods of Booch, Rumbaugh, and others to provide a direct route from requirements to implementation Coleman et al introduce fusion method. This method, fusion, is a systematic method for use by medium- to large-size teams in the industrial environment and was developed at Hewlett-Packard Labs, Bristol, UK (Coleman et al, 1993). For most of the history of object-oriented, as seen from the previous examples there have been many methods competing to be the singular way to model objects. As a result, the object-oriented community has been fractured into various factions, each supporting its favorite modeling language. Now with the introduction of the Unified Modeling Language (UML)(OMG, 2001), object-oriented practitioners and purveyors of OO tools and technology have been united behind a single language, if not a single methodology. What is accomplished by the object-oriented paradigm is that the *data* and *how* to manipulate the data are combined (encapsulated) into a single representation (an Object or Class). Additionally, UML defines diagrams that deal with the distribution and timing of the system. By combining multiple, related, diagram types, UML do address all aspects of system architecture.

3.3 Elements of the Object-oriented Approached

The basic element in an object-oriented system architecture modeling is an object. An object is an encapsulation of both data and functionality with the added support of message passing and inheritance. We refer to the data in an object as the attributes of the object, while the functionality is provided by the methods. These two entities—attributes and functionality—form a single logical entity, an object. This contrasts with the more traditional structured programming, which considers data separately from the procedures that act on them. Such a logical grouping of data, along with the procedures that will affect them, gives a conceptual as well as a physical basis for an object. For example, the same logical grouping mimics some nice properties that humans take for granted in the way we conceptualise everyday objects. The old joke about whether the price of the car includes a steering wheel indicates to us that we have predetermined that the object has the functionality of steering embedded in the concept of the car (object). Objects themselves are usually created through an instantiation process that uses a general template called a class. The template contains the characteristics of the class, without containing the specific data that needs to be inserted into the template to form the object.

Classes are the basis for most design work in objects. Although the purpose may be to instantiate objects, that particular task can be delayed as the higher level of abstraction is created. Classes are either superclasses (root classes), created with a set of basic attributes and methods, or subclasses, meaning they inherit the characteristics of the parent's class and may add (or remove) functionality as desired. A superclass may be created with general characteristics found in all classes of a particular relationship. This type of relationship forms what we call a class hierarchy lattice. Another type of relationship, interaction between classes, may take two different forms: classes may share data through aggregate or component grouping, or classes may share objects. Aggregate classes interact through messages, which are directed requests for services from one class (called a client) to another class (called a server). Notice that the class that makes the request depends upon the collaborating server class; the client is said to be coupled to the server. The serving class may have no dependence on the class using the requested material, so clearly this relationship is not commutative. The relationship in which two or more different classes form a component, thus developing a PART_OF, is also called a HAS_A relationship. If one object uses another object through another class, the dependency is now upon the attributes and methods of the used object. Because of this additional complexity, we choose to consider the uses relationship separately from simply the passing of attributes. This dissertation will used techniques from Booch's Object-oriented Analysis and Design (OOAD) development method to address the research objectives. The following common stapes that was taken to partially develop the object oriented architecture:

1. Gather Information about the proposed Satellitic Automated Highway Systems
2. Beginning a Project.
3. Creating use cases.
4. Finding Classes
5. Discovering Object Interaction
6. Specifying Relationships.
7. Adding Behavior and Structure.
8. Discovering Inheritance
9. Checking the Model

The building elements of system architecture should simulate the environment that these systems will interact. The environment is modeled with objects from the world that the system exists within. These objects represent instances of a more general object, which is called class. These classes combined the data and the operation that are going to operate on that data in the same building component when it receives a certain message. This has a significant value when we want to develop software that uses the same class. This process is called reuse, which has a big impact on the software industry in general and on traffic information and control systems. The object-oriented approach offers the same features such as polymorphism and inheritance that later helped greatly in the reuse Processes and where one of powerful factors that led the information, communication, and computing revolution. The object-oriented development started with efforts to simulate the real world such as cars and traffic lights and road systems. It is natural to continue being the logical method of preference in development of a Satellitic Automated Highway Systems that this dissertation is all about as a target for the Intelligent Transportation movement. The object–oriented approach should be in a position to respond to the special need of such systems that are characterized as a communication critical and safety critical systems. Moreover, the specification that guide use in the development processes should take the same direction.

3.4 UML VISUALIZATION OF THE OBJECT-ORIENTED APPROACHED

The robust notation that is represented by UML plays important roles in modeling the system architecture. The UML is an evolution from Booch, OMT, OOSE, other object-oriented methods, and many other sources. In Figure 3.1, some methodologies that had been participating in the evolution of UML are illustrated. UML acts as means for decision communication, an important responsibility. It provides the semantics that help in capturing all important and strategic tactical decision. The structure of UML provides a sold base for human to reason and tool to manipulate. The Canadian Automated Highway System (CAHS) and the Satellitic automated Highway System (SAHS) are very complex system, therefore it is essential to visualize and model such system architectures to for fill our objectives. The UML is the industry standard modeling language after the standardizing processes conducted and adapted by MOG in 1997. UML was the language of choice throughout this dissertation.

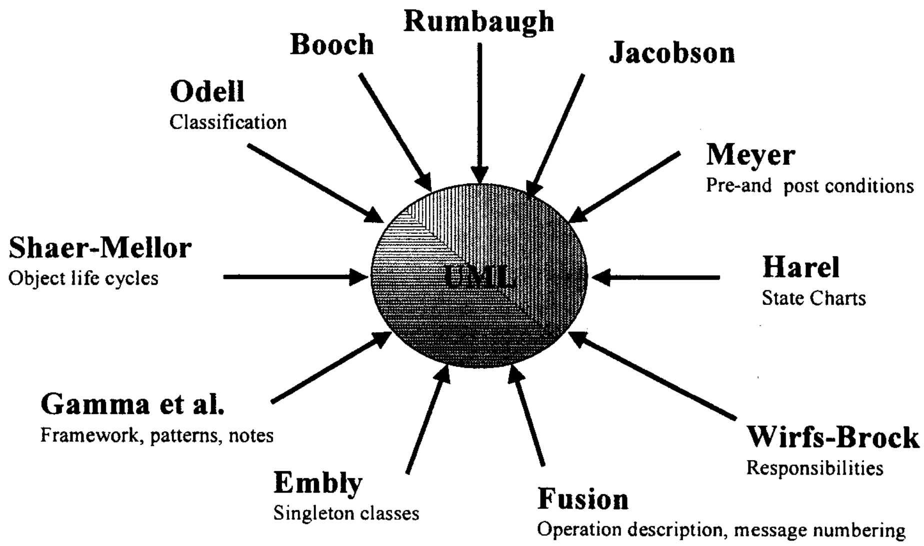

Figure 3.1: UML unifies Booch, OMT, Objectory modeling, and other modeling languages

Therefore, it is essential to visualize and model such system architectures to fulfill our objectives. The UML is the industry standard modeling language after the standardizing processes conducted and adapted by MOG in 1997. UML was the language of choice throughout this dissertation.

The two prospective of defining the artifact that can be modeled and produces are the definition and the development artifacts. The first view, the definition prospective is detailed in the specification of the language itself. The first view is in areas where UML semantics, notation guide, and UML standard profile are explained. The second prospective is the development project artifacts. The type of the model to be visualized during the development, effects how we see the intensive software system compared to the real world in the eyes of the developer. The focusing on the necessary details while ignoring others or what is called abstraction determines the development artifacts that are going to be created. To capture the characteristic of a computer intensive system like A Satellitic Automated Highway system, we divide the system into small set of nearly independent views, no one view is sufficient to present the totality of the system. These models can be visualized in different levels of detail. The evaluation of the different development models that were chosen in this dissertation was based on two things: the adhesiveness to the real word, and the ability to capered the functional requirements.
The UML supports in different views the following visualizing diagrams:

- Use Case Diagram
- Class Diagram
- Behaviour Diagrams:
    - State Chart Diagram
    - Activity Diagram
    - Interaction Diagram
    - Sequence Diagram
    - Collaboration Diagram
- Implementation Diagrams:

- Component Diagrams
- Deployment Diagrams

In modeling the requirement of A Satellitic Automated Highway Systems, we chose the actors, use case diagram, class diagram, and the sequence diagram to visualize the communication problems within the architecture and to visualize the solution by partially developing the Satellitic Automated Highway System. The Use case Model is one of the first stages in software development as they model the functional requirements and scope. The two major components of a use case are actor and a use case. An actor is a role player interacting externally with the system. A possible one physical user of the system with several roles will be visualized with several actors. The external interaction of an actor may be input information to the system, receive information for the software system or input and receive information to and from a system. A use case can be defined as series of communications performed by a system that yields a quantifiable outcomes for a meticulous actor, In other words, a presentation of the functionality of the system, the capabilities that will be given to the user of the system. With the actor outside the boundary of the system that we hold the authority to change communicating externally though messages. The figure that associates the use case and the actor and the communication that is going between them is called the use case diagram. Another important part of the intensive software system model is a Class, which captures the common, behavior of a set of objects. UML provide a graphical way to present it. These classes when exist in the real world they are in a state of object. These objects are instance of the class. The graphical representation of the class in UML as a 3-part box, with the class name in the top part, a list of attributes, with optional types and values in the middle part, and a list of operations, with optional argument lists and return types in the bottom part. The attribute and operation sections of the class box can be suppressed to reduce detail in an overview. Suppressing a section makes no statement about the absence of attributes or operations, but drawing an empty section explicitly states that there are no elements in that part. The classes of the software intensive system are joined to gather forming a Class diagram. A **class** diagram is a picture for describing general descriptions of possible systems. Class diagrams and object diagrams are alternate representations of object models. Class diagrams contain classes and object diagrams contain objects, but it is possible to mix classes and objects when dealing with various kinds of metadata, so the separation is not rigid. Class diagrams are more prevalent than object diagrams. Normally you will build class diagrams plus occasional object diagrams illustrating complicated data structures or message-passing structures. Class diagrams contain icons representing classes, interfaces, and their relationships. You can create one or more class diagrams to depict the classes at the top level of the current model; such class diagrams are themselves contained by the top level of the current model. During the Analysis stage the class diagram serve as a presentation to of common roles and responsibilities of the entities that provide the system's behavior. In the Design stage the class diagram can help in capturing the structure of the classes that form the intensive software system's design. Another important part of the model is a Package. Packages serve to partition the logical model of a system. They are clusters of highly related classes that are themselves cohesive, but are loosely coupled relative to other such clusters. You can use packages to group classes, interfaces, and other packages.

You can draw logical packages only in class diagrams. While many object-oriented programming languages do not yet support this concept, using packages in a class diagram allows us to express and preserve important architectural elements of the intensive software system design. You can capture a high-level design of the system by creating a class diagram that consists only of packages. The relationships that capture certain characteristics of the software intensive system form an important part of the UML notation. Some of the relationships that UML allowed us to reflect in our model Association Relationship, Aggregate Relationship, Generalize/Inherits Relationship, Instantiates Relationship, and Meta Relationship. In order to comprehend the modeling power that is within UML the description of these relations:

- Association Relationship, a semantic related to connection type between two classes, or between a class and an interface. The Associations work in both directions it is the weakest among the relations.

- Aggregate Relationship; a relation between two classes that have a whole and part relationship.
- Generalize/Inherits Relationship, A generalize relationship between classes shows that the subclass shares the structure or behavior defined in one or more super classes.
- Instantiates Relationship, One of the relations that UML can model is the instantiates relationship, which in intensive software systems represents the act of substituting actual values for the parameters of a parameterized class, or parameterized class utility to create a specialized version of the more general item.
- Meta Relationship, the UML provide us with notation to present the relation between the metaclass and its class. The Meta Relationship is a gray line with an arrowhead pointing toward the metaclass.

Sequence Diagrams, OMG UML supports sequence diagram by providing a graphical representation of an interaction of the intensive software system. A sequence diagram traces the execution of an interaction in time. Message sequence numbers should be mentioned on the message's vertical order. The initiating of a message or the destination address should be from or to an object, which, is represented by a rectangular box with a vertical line below it. The vertical line allows us to create messages between objects.

Collaboration Diagrams, OMG UML support the Collaboration Diagram and provide a graphical representation of that. A collaboration diagram is an interaction diagram that shows the sequence of messages that implement an operation or a transaction. Collaboration diagrams show objects, their links, and their messages. They can also contain simple class instances and class utility instances.

OMG UML is rich with providing graphical icons and presentation of all modelling aspect of a system architecture of a complex system such as a Satellitic Automated Highway Systems. It does not suggest a development process methodology or any programming langue. The use of OMG UML in this dissertation is to the extent that is enough to fulfil our objectives.

3.5 The need for an Object-oriented based system architecture

Object oriented development methodology provides ITSC a Stable and flexible architecture and help to integrate the backup communication concept with the new ITSC architecture, which will enhance reliability and dependability of systems that will be developed based on the architecture, because object-oriented offers:

1. Greater simplicity in system development and maintenance.
2. Localization of change.
3. Maximum opportunity for the re-use of components.
4. Good management of variability.
5. Protection of data from illegal access.
6. Better management of complexity.

3.6 The Object-oriented mechanism to provide stable and flexible architecture:

Object-oriented development methodology provide us with the means to attach two important system qualities the current ITSC architecture lack i.e. Stability and flexibility by:

1. Alternates/Expands some parts of the System Architecture.
2. Unifies functions and information.
3. Describes a target system with "objects" making it possible to create a unified model of information and functions.
4. Organizes and structures information based on its similarities.
5. Detects the function and its related information to be added and/or changed.
6. Finds the parts that needed to be corrected quickly as well as minimizing corrections.
7. Meets future changes in social needs and development in technology.

Chapter 4

METHODOLOGICAL FRAMWORK

4.1 Overview of Methodological Framework

The methodological framework that was applied throughout the various phases of the study depends mainly on the techniques associated with the OOAD (Booch, 1997) method. These techniques may provide the tools that can allow us to visualize the problem clearly. This environment should lead us to the solution that will provide a balanced proportion of flexibility and stability to our ITSC architecture. The implementation of the above-mentioned points is in addition to the use of the UML (OMG, 2002) notation. A methodology is required to be developed. This chapter offers some insight into such methodology.

There is no standard method of analysis for determining the influence of quality architecture attributes on system. The lack of such method hinders the development of a clear relationship between architectural stability, represented by a backup communication system; and the system usability and functionality, represented by user and functional requirements. Therefore, meaningful results and physical insight into the influence of architectural attributes are better obtained through modeling or the use of diagrams. The use of OOAD in modeling can ease changes in the architectural design when a backup communication system is introduced.

4.2 Steps of Analysis

The following steps were taken in order to prove that stability and flexibility couldn't be maintained in the ITSC architecture without adopting an object-oriented methodology and a backup communication system concept integrated within the ITSC architecture:

4.2.1 CAHS Conceptual Project:

To prove the need of a backup communication system clearly visualized and document within the ITSC architecture, a partial object-oriented development is required. CAHS possess all ITSC architecture requirements. All communication modes that are available in ITSC architecture are presented in CAHS. A Three parts analysis were conducted:

4.2.1.1 CAHS Scientific Analysis

1. List the user and functional requirements in the CAHS requirement shell. The requirement used in the CAHS was used as the requirements for the ITSC architecture in the CAHS requirement shell.
2. Formulate the relationship between user and functional requirements according to the two attributes of class responsibility and use cases in the CAHS requirement shell.
3. List the relationship between the user and functional requirements in the CAHS requirement shell.
4. Determine the relationship between the use case numbers and the requirements numbers in the CAHS requirement shell.
5. List the classes in relation to the user and function requirements in a manner that corresponds to the CAHS requirement numbering system, in order to establish the class responsibility.
6. Analyze the extent to which the backbone communication class participates in fulfilling most requirements.

4.2.1.2 CAHS Documental Analysis

1. For each use case of CAHS document Characteristic Information
2. For each use case of CAHS document Main Success Scenario
3. For each use case of CAHS document Scenario Extensions
4. For each use case of CAHS document Related Information

5. For each use case of CAHS document Open Issues
6. Analyze the extent to which the backbone communication class participates in fulfilling all use cases in order to have a successful scenario

4.2.1.3 CAHS Visual analysis

1. Utilize the OOAD technique to develop use case diagrams. Use case diagrams define the architecture boundary, the external actors, and the services provided.
2. Utilize the OOAD technique to develop sequence (interaction) diagrams describing how the implied objects of the system co-operate to provide the services defined in the use case.
3. Utilize the OOAD technique to develop class diagrams. These diagrams define the abstract elements that comprise the architecture of the system.
4. Analyze the extent to which the backbone communication class participates in fulfilling most sequence diagrams in order to have a successful scenario.

Since CAHS possess all ITSC architecture requirements, then the result of the analysis can be directly generalized to the ITSC architecture.

4.2.2 SAHS Conceptual Project

In order to prove, that our solution for backup communication is adequate and an object-oriented based architecture provides stability and flexibility to the ITSC architecture a partial Object-oriented development is required. CAHS possesses all ITSC architecture requirements plus any that covers the backup communication system requirements. SAHS communication system is an integration of backbone communication system with backup communication system. A three parts solution analysis was conducted:

4.2.2.1 SAHS Scientific Analysis

1. List the user and functional requirements in the SAHS requirement shell. The requirement used in the SAHS was used as the requirements for the ITSC architecture in the SAHS requirement shell.
2. Formulate the relationship between user and functional requirements according to the two attributes of class responsibility and use cases in the SAHS requirement shell.
3. List the relationship between the user and functional requirements in the SAHS requirement shell.
4. Determine the relationship between the use case numbers and the requirements numbers in the SAHS requirement shell.
5. List the classes in relation to the user and function requirements in a manner that corresponds to the SAHS requirement numbering system, in order to establish the class responsibility.
6. Analyze the extent to which the backbone communication class and a backup communication class are participates in fulfilling most SAHS requirements and examine the object-oriented localization of change offering regarding the stability and flexibility of SAHS architecture.

4.2.2.2 SAHS Documental Analysis

1. For each use case of SAHS document Characteristic Information
2. For each use case of SAHS document Main Success Scenario
3. For each use case of SAHS document Scenario Extensions
4. For each use case of SAHS document Related Information
5. For each use case of SAHS document Open Issues

6. Analyze the extent to which the backbone communication class and a backup communication class are participates in fulfilling most SAHS use cases and examine the object-oriented localization of change offering regarding the stability and flexibility of SAHS architecture within the documentation.

4.2.2.3 SAHS Visual Analysis

1. Utilize the OOAD technique to develop use case diagrams. Use case diagrams define the architecture boundary, the external actors, and the services provided.
2. Utilize the OOAD technique to develop sequence (interaction) diagrams describing how the implied objects of the system co-operate to provide the services defined in the use case.
3. Utilize the OOAD technique to develop class diagrams. These diagrams define the abstract elements that comprise the architecture of the system.
4. Analyze the extent to which the backbone communication class and a backup communication class participate in fulfilling most sequence diagrams of SAHS and examine the object-oriented localization of change offering regarding the stability and flexibility of SAHS architecture within the visualization.

Since SAHS possess all ITSC architecture requirements plus (Backup Communication system) architecture requirements, then the analysis result can be directly generalize to ITSC architecture.

4.3 RATIONAL ROSE, THE CASE TOOL

Rational Rose is a graphical software-modeling tool that uses Unified Modeling Language (UML) as a primary notation. A case tool is used within our effort to deal with complexity that is reflected from the details of the Satellitic Automated Highway Systems development. The use of patterns and heuristics extracts the system information that is required for a certain detailing. Rational Rose creates a working environment in implementing abstraction. Abstraction in the use of a subset of classes and models element, to present the development model in a easy way to comprehend and to modify. It gives a chance to experiment with different configuration and system arrangements in order to achieve the research goals.

With this tool, we will be able to:

1. Identify and design related objects, and then map them to architectural components.
2. Partition services across Rose model.

Our partial object-oriented development of Canadian Automated Highway System (CAHS) and Satellitic Automated Highway System (SAHS) depends essentially on two elements of modern system architecture development and engineering: component based development and controlled interactive development. Rational Rose supports both independent concept and implies them in the construction of our object-oriented development (Rational Software Corp., 1998).

4.3.1 Notations

As we mentioned in a previous section that notation plays an important role in the development of an intensive software system; a Satellitic Automated Highway Systems is not different. The UML is standardize notation By Object Modeling Group (OMG) and are supported by Rational Rose. The Unified modeling language enables us to communicate decisions that are not obvious or cannot be abstracted from the code it self.

UML provides us with the technique to visualize important strategic and tactical characteristics of the proposed system architecture.

4.3.2 Features of Rational Rose

Many of Rational Rose features will be used in analysis, design, and iterative construction. It is a logical decision to utilize in our development for:

1. Use -Case Analysis to capture the functionality and the component of the system.
2. Object-Oriented Modeling.
3. User Configurable Support for UML.
4. Semantic Checking.
5. Support for Controlled Interactive Development.
6. Round -Trip Engineering.
7. DDL Generation and Integration with Data Modeling Tools.
8. Documentation Generation.
9. Rose Scripting for Integration and Extensibility.
10. OLE Linking
11. OLE Automation.

4.3.3 Rational Rose User Interface

Rational Rose provides us with a Windows platform for our development with the following areas:

1. Model Browser.
2. In-Place Editing.
3. Reduced levels of Dialog Boxes.
4. Wizards for Frequent Operations.

The Rose model is a picture of system architecture. The use of UML will help us in describing how the system will work and to generate documentation to visualize the requirement. The Rose model will be used to get high-level view of the system, break down the system to manageable pieces, describe the system functional logic, and trace the requirement to the various system classes. Canadian Automated Highway System (CAHS) and Satellitic Automated Highway System (SAHS) are complex systems. The management of the information and logic within the system requires the use of such tools. Extended information about this powerful case tool can be found in Rational web page at http://www.rational.com/.

Chapter 5

A CANADIAN AUTOMATED HIGHWAY SYSTEM (CAHS): ANALYSIS OF ITSC REQUIREMENTS RELATED TO COMMUNICATION PROBLEM

The main objective of this chapter is to prove the need of a backup communication system. In order to reach the goal, we set up a conceptual system called the Canadian Automated Highway System (CAHS) is partially developed. CAHS is an object-oriented development and possess all ITSC architecture requirements. All communication modes that are available in ITSC architecture are presented in CAHS. The evaluation of the dependency of ITSC requirements within the architecture on communication in a analytical manner, "Communication is what makes ITS work" (Transport Canada, 1999). It should be clear that we are not redefining the entire system due to limited time and resources, but rather the communication between actors and the system by adopting an object-oriented methodology, using the Unified Modeling Language (UML) as notation, and Rational Rose as a case tool for some parts of the analysis. However, we are scratching the surface of the architectural design only sufficiently to the extent whereby the weakness of a lack of backup communication may manifest itself. A three parts analysis was conducted:

1. Scientific Analysis
2. Documental Analysis
3. Visual analysis

5.1 Scientific Analysis

The scientific study will contain both the functional requirements and the user requirements of the ITSC. In order to do the study we created both use case and class system. The study is about tracing each requirement to the class or classes that are responsible for the functionality of the requirement CAHS, the proposed partial architectural development effort for a system that will demonstrate clearly the need for a backup communication. The following were performed:

1. The description of the user and functional requirements was derived from the ITSC architecture and called the Requirement Description (RDescription).

2. The functional requirement number was derived from the ITSC architecture and called the Functional Requirement Number (FunctionalR#

3. The user requirement number was derived from the ITSC architecture and called the User Requirement Number (UserR #).

4. The functional requirement number was derived from the ITSC architecture and assigned to the Use case number and called Use Case number (Use Case #).

5. The description of the created class number in order to be able to trace the requirements to the responsible class was called Class responsibility number (ClassR#).

6. The automated number was created to keep truck of the study. We called it the Canadian Automated Highway System number (CAHS #)

The results were documented in CAHS requirements class responsibility shell. The shell is a database that help us to track the various related relations. The scientific analysis of CAHS requirement led us to trace requirements to its classes. Since CAHS posses all ITSC architecture requirements any conclusions fro the scientific analysis may directly generalize to the ITSC architecture.

The analysis shows that CAHS physical classes can be modeled by thirty classes; their names and numbers are given in Table 5.1. We created a class called Backbone communication class to present the four-communication channels that are available in ITSC architecture. We will keep close attention to this class in order to monitor its participation in the fulfillment of the requirement functionality.

The study covers 689 requirements back to their responsible classes. The analysis report details are given in Appendix A. A snap shot of that report is shown below.

CAHS #	RDescription	UserR#	FunctionalR#	UseC#	ClassR#
6	The CAHS shall provide travellers with instructions on turns and other manoeuvres to reach selected destinations.	1.2	1.1.1.3 1.1.3 6.2.1.1 6.2.1.2 6.2.1.3 6.2.1.4 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.6.2.1 6.7.2.1.1 6.7.2.2	1.1.1.3 1.1.3 6.2.1.1 6.2.1.2 6.2.1.3 6.2.1.4 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.6.2.1 6.7.2.1.1 6.7.2.2	6 16 13 31 30
7	The CAHS shall Enable route planning and detailed route guidance based on static, stored information collected through in-vehicle sensory, location determination,	1.2.1	6.2.2 6.2.3 6.6.2.1 6.7.2.1.1 6.7.2.2	6.2.2 6.2.3 6.6.2.1 6.7.2.1.1 6.7.2.2	6 31 30 13

Table 5.1 CAHS classes and its related number

CAHS Class Name	**Class Number**
Automated Highway System	1
Centres	2
Wayside	3
Travellers	4
Vehicles	5
Backbone Communication	6
Archived Data Management	8
Commercial Vehicle Administration	9
Emissions Management	10
Fleet and Freight Management	12
Information Service Provider	13
Maintenance Management	14
Toll Administration	15
Traffic Management	16
Transit Management	17
Emergency Management	18
Commercial Vehicle Check	19
Intermodal Terminal	20
Parking Management	21
Roadway	22
Toll Collection	23
Personal Information Access	24
Remote Traveler Support Subsystem	25
Commercial Vehicle	26
Emergency Vehicle	27
Intermodal Container	28
Maintenance Vehicle	29
Transit Vehicle	30
Vehicle	31

Notice class no.6, which is the backbone communication class. It means that the functionality of the requirement cannot be fulfilled without communication. This is true for most of the 689 requirements that were included in this part of the study. **What will happen if a malfunction occurs in the backbone communication class?**

Findings:

1. We were able to trace most of the requirements to the responsible classes.
2. The study shows that in order for the almost all CAHS architecture requirements to fulfill its functionality, it must use the communication system to perform its duty. Therefore, the communication subsystem is the backbone of the system. One of the most important findings of this study is the need for a backup communication system to increase the reliability and functionality of the system.
3. The scientific analysis enabled us to model and to document in an OOAD methodology, the findings of the study. We were able to illustrate successfully the need of a backup communication system.

Since CAHS posses, all ITSC architecture requirements we can directly generalize the findings of the scientific analysis back to the ITSC architecture.

5.2 Documental Analysis

In the documental analysis, the flow of events is captured in this partial development of CAHS in text. The numbering for this documentation is related to the number of the process in ITSC architecture. The documentation captured the communication from the actors to the various classes that are needed to perform the desired functionality. For each third level uses case, we focused only on one-way communication in the documental analysis only the actors are communicating with the system and subsystems. The system and subsystems will meet most requirements of the actors, but will not directly give specific answers to the actors. This documental analysis is based on the requirements that are provided by the ITSC architecture. We analyze and documented the following:

- For each use case of CAHS document Characteristic Information
- For each use case of CAHS document Main Success Scenario
- For each use case of CAHS document Scenario Extensions
- For each use case of CAHS document Related Information
- For each use case of CAHS document Open Issues
- Analyze the extent to which the backbone communication class participates in fulfilling all use cases in order to have a successful scenario

The documentation analysis for CAHS third level use cases are given Appendix B. For an example of this document analysis, the following five use cases were chosen:

- 4A.4: CAHS Use Case Name: Support Security and Coordination
- 6A.2: CAHS Use Case Name: Provide Information Services
- 6A.8: CAHS Use Case Name: Provide Traveller Personal Services
- 7A.2: CAHS Use Case Name: Provide Electronic Parking Payment
- 7A.4:Use CAHS Case Name: Carry-out Centralized Payments Processing

The documentations are as follows:

4A.4: CAHS Use Case Name: Support Security and Coordination

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the use case.

Goal In Context:	Support Security and Coordination
Scope:	CAHS
Level:	Strategic
Pre-Condition:	System is ON and the Backbone Communication System is Functioning
Success End Condition:	Security is Coordinated and Supported
Failed End Condition:	Security is not Coordinated and Supported
Primary Actor:	32, 33, and 34

Trigger Event: System Clock

Main Success Scenario

This scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1.	17	4.4.1.1 Manage Transit Security	17
2.	33	4.4.1.2 Manage Transit Emergencies	6
3.	6	4.4.1.2 Manage Transit Emergencies	30
4.	33	4.4.1.3 Provide Transit System Operator Security Interface	6
5.	6	4.4.1.3 Provide Transit System Operator Security Interface	17
6.	34	4.4.1.4 Provide Transit External Interface for Emergencies	6
7.	6	4.4.1.4 Provide Transit External Interface for Emergencies	17
8.	33	4.4.1.5 Provide Transit Driver Interface for Emergencies	6
9.	6	4.4.1.5 Provide Transit Driver Interface for Emergencies	30
10.	17	4.4.1.6 Collect Transit Vehicle Emergency Information	17
11.	32	4.4.1.7 Monitor Secure Area	6
12.	6	4.4.1.7 Monitor Secure Area	25
13.	33	4.4.1.8 Report Traveller Emergencies	6
14.	6	4.4.1.8 Report Traveller Emergencies	25
15.	33	4.4.2 Coordinate Multiple Agency Responses to Incidents	6
16.	6	4.4.2 Coordinate Multiple Agency Responses to Incidents	17
17.	33	4.4.3 Generate Responses for Incidents	6
18.	6	4.4.3 Generate Responses for Incidents	17
19.	34	4.4.4 Coordinate Transit Disaster Response	6
20.	6	4.4.4 Coordinate Transit Disaster Response	17

Scenario Extensions

This is a listing of how each step in the main success scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the main success scenario is rejoined or the failed end condition is met. The step refers to the failed step in the main success scenario and has a letter associated with it; i.e., if step 3 fails the extension step is 3a.

Step	Condition	Action Description
2a	Backbone Communication System Failure	Switch to Conventional Vehicle Highway System

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	Variable	Possible Variations
2a	Backbone Communication System Failure	Switch to Conventional Vehicle Highway System

Related Information

The following table gives the information that is related to the use case.

Schedule:	ASAP
Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Manage Transit
Sub Use Case(s):	N/A
Channel To Primary Actor:	Backbone Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	Backbone Communication System Failure

6A.2: CAHS Use Case Name: Provide Information Services

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the use case.

Goal In Context:	Provide Information Services
Scope:	CAHS
Level:	Strategic
Pre-Condition:	System is On and Backbone Communication System is functioning
Success End Condition:	Information Services is provided
Failed End Condition:	Information Services is not provided
Primary Actor:	33, 34 and 35
Trigger Event:	Request from 34

Main Success Scenario

This scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1.	34	6.2.1.1 Collect Traffic Data for Advisory Messages	6
2.	6	6.2.1.1 Collect Traffic Data for Advisory Messages	13
3.	35	6.2.1.1 Collect Traffic Data for Advisory Messages	6
4.	6	6.2.1.1 Collect Traffic Data for Advisory Messages	13
5.	13	6.2.1.2 Provide Traffic and Transit Advisory Messages	13
6.	35	6.2.1.3 Collect Transit Data for Advisory Messages	6
7.	6	6.2.1.3 Collect Transit Data for	13

		Advisory Messages	
8.	13	6.2.1.4 Provide Traffic and Transit Broadcast Messages	13
9.	33	6.2.1.5 Provide ISP Operator Broadcast Parameters Interface	6
10.	6	6.2.1.5 Provide ISP Operator Broadcast Parameters Interface	13
11.	33	6.2.1.6 Provide Transit Advisory Data On Vehicle	6
12.	6	6.2.1.6 Provide Transit Advisory Data On Vehicle	30
13.	31	6.2.2 Prepare and Output In-vehicle Displays	31
14.	33	6.2.3 Provide Transit User Advisory Interface	6
15.	6	6.2.3 Provide Transit User Advisory Interface	30
16.	13	6.2.4 Collect Yellow Pages Data	13
17.	33	6.2.5 Provide Driver Interface	6
18.	6	6.2.5 Provide Driver Interface	31
19.	13	6.2.6 Provide Yellow Pages Data and Reservations	13

Scenario Extensions

This is a listing of how each step in the main success scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the main success scenario is rejoined or the failed end condition is met. The step refers to the failed step in the main success scenario and has a letter associated with it; i.e., if step 3 fails the extension step is 3a.

Step	Condition	Action Description
1a	Backbone Communication System Failure	Switch to Conventional Vehicle Highway System

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	Variable	Possible Variations
1a	Backbone Communication System Failure	Switch to Conventional Vehicle Highway System

Related Information

The following table gives the information that is related to the use case.

Schedule:	ASAP
Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Provide Driver And Traveller Services
Sub Use Case(s):	N/A
Channel To Primary Actor:	Backbone Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	Backbone Communication System Failure

6A.8: CAHS USE CASE NAME: PROVIDE TRAVELLER PERSONAL SERVICES

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the use case.

Goal In Context:	Provide Traveller Personal Services
Scope:	CAHS
Level:	Strategic
Pre-Condition:	System Must be On and the Backbone Communication is functioning
Success End Condition:	Traveller Personal Services is provided
Failed End Condition:	Traveller Personal Services is not provided

Primary Actor: 32 and 34

Trigger Event: System Clock

Main Success Scenario

This scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1.	24	6.8.1.1.1 Determine Personal Portable Device Guidance Method	24
2.	24	6.8.1.1.2 Provide Personal Portable Device Dynamic Guidance	24
3.	24	6.8.1.1.3 Provide Personal Portable Device Autonomous Guidance	24
4.	32	6.8.1.2 Provide Personal Portable Device Guidance Interface	6
5.	6	6.8.1.2 Provide Personal Portable Device Guidance Interface	24
6.	24	6.8.1.3 Process Personal Portable Device Location Data	24
7.	34	6.8.1.4 Update Traveller Navigable Map Database	6
8.	6	6.8.1.4 Update Traveller Navigable Map Database	24
9.	24	6.8.1.5 Provide Traveller Emergency Message Interface	24
10.	32	6.8.2.1 Build Traveller Personal Security Message	6
11.	6	6.8.2.1 Build Traveller Personal Security Message	24
12.	24	6.8.2.2 Provide Traveller Emergency Communications Function	24
13.	24	6.8.3.1 Get Traveller Personal Request	24
14.	24	6.8.3.2 Provide Traveller with Personal Travel Information	24
15.	32	6.8.3.3 Provide Traveller Personal Interface	6
16.	6	6.8.3.3 Provide Traveller Personal Interface	24
17.	34	6.8.3.4 Update Traveller Personal Display Map Data	6
18.	6	6.8.3.4 Update Traveller Personal Display Map Data	24

Scenario Extensions

This is a listing of how each step in the main success scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the main success scenario is rejoined or the failed end condition is met. The step refers to the failed step in the main success scenario and has a letter associated with it; i.e., if step 3 fails the extension step is 3a.

Step	Condition	Action Description
4a	Backbone Communication System Failure	Switch to Conventional Vehicle Highway System

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	Variable	Possible Variations
4a	Backbone Communication System Failure	Switch to Conventional Vehicle Highway System

Related Information

The following table gives the information that is related to the use case.

Schedule:	ASAP
Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Provide Driver And Traveller Services
Sub Use Case(s):	N/A
Channel To Primary Actor:	Backbone Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	Backbone Communication System Failure

7A.2: CAHS Use Case Name: Provide Electronic Parking Payment

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the use case.

Goal In Context:	Provide Electronic Parking Payment
Scope:	CAHS
Level:	Strategic
Pre-Condition:	The system is ON and the Backbone Communication System is functioning
Success End Condition:	Electronic Parking Payment is Provided
Failed End Condition:	Electronic Parking Payment is not complete
Primary Actor:	34 and 33
Trigger Event:	System Clock

Main Success Scenario

This scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1	21	7.2.1.1 Read Parking Lot Tag Data	21
2	21	7.2.1.10 Determine Advanced Charges	21
3	21	7.2.1.2 Calculate Vehicle Parking Lot Charges	21
4	34	7.2.1.3 Collect Bad Charge Payment Data	6
5	6	7.2.1.3 Collect Bad Charge Payment Data	21
6	21	7.2.1.4 Check for Advanced Parking Lot Payment	21
7	21	7.2.1.5 Bill Driver for Parking Lot Charges	21
8	34	7.2.1.6 Manage Parking Lot Financial Processing	6
9	6	7.2.1.6 Manage Parking Lot Financial Processing	21
10	33	7.2.1.7 Update Parking Lot Data	6
11	6	7.2.1.7 Update Parking Lot Data	21
12	33	7.2.1.8 Register for Advanced Parking Lot Payment	6
13	6	7.2.1.8 Register for Advanced Parking Lot Payment	21
14	33	7.2.1.9 Manage Parking Lot Reservations	6

15	6	7.2.1.9 Manage Parking Lot Reservations	21
16	21	7.2.2 Produce Parking Lot Displays	21
17	21	7.2.3 Obtain Parking Lot Violator Image	21
18	34	7.2.4 Provide Driver Parking Lot Payment Interface	6
19	6	7.2.4 Provide Driver Parking Lot Payment Interface	31
20	33	7.2.4 Provide Driver Parking Lot Payment Interface	6
21	6	7.2.4 Provide Driver Parking Lot Payment Interface	31
22	21	7.2.5 Detect Vehicle for Parking Lot Payment	21
23	13	7.2.6 Distribute Advanced Tolls and Fares	13
24	34	7.2.7 Provide Payment Instrument Interface for Parking	6
25	6	7.2.7 Provide Payment Instrument Interface for Parking	31

Scenario Extensions

This is a listing of how each step in the main success scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the main success scenario is rejoined or the failed end condition is met. The step refers to the failed step in the main success scenario and has a letter associated with it; i.e., if step 3 fails the extension step is 3a.

Step	Condition	Action Description
4a	Backbone Communication System	CAHS does not have options to handle this situation but to switch to conventional vehicle highway system

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	Variable	Possible Variations
4a	Backbone Communication System	CAHS does not have options to handle this situation but to switch to conventional vehicle highway system

Related Information

The following table gives the information that is related to the use case.

Schedule:	ASAP
Priority:	Must
Performance Target:	N/A
Frequency:	N/A

Super Use Case:	Provide Electronic Payment Services
Sub Use Case(s):	N/A
Channel To Primary Actor:	Backbone Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	Failure of the Backbone Communication System

7A.4:Use CAHS Case Name: Carry-out Centralized Payments Processing

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the use case.

Goal In Context:	Carry-out Centralized Payments Processing
Scope:	CAHS
Level:	Strategic
Pre-Condition:	The System is On and Backbone communication is functioning
Success End Condition:	Centralized Payments Processing is carried-out
Failed End Condition:	Centralized Payments Processing is not carried-out
Primary Actor:	34
Trigger Event:	Request from 34

Main Success Scenario

This scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1	34	7.4.1.1 Process Commercial Vehicle Payments	6
2	6	7.4.1.1 Process Commercial Vehicle Payments	26
3	34	7.4.1.2 Process Yellow Pages Services Provider Payments	6
4	6	7.4.1.2 Process Yellow Pages Services Provider Payments	13
5	34	7.4.1.3 Process Driver Map Update Payments	6
6	6	7.4.1.3 Process Driver Map Update Payments	13
7	34	7.4.1.4 Process Traveller Map Update Payments	6
8	6	7.4.1.4 Process Traveller Map Update Payments	13
9	34	7.4.1.5 Process Transit User Other Services Payments	6
10	6	7.4.1.5 Process Transit User Other Services Payments	13
11	34	7.4.1.6 Process Traveller Trip and Other Services Payments	6
12	6	7.4.1.6 Process Traveller Trip and Other Services Payments	13
13	13	7.4.1.7 Collect Payment Transaction Records	13
14	34	7.4.1.8 Process Traveller Rideshare Payments	6
15	6	7.4.1.8 Process Traveller Rideshare Payments	13
16	13	7.4.2 Collect Price Data for CCAHS Use	13

Scenario Extensions

This is a listing of how each step in the main success scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the main success scenario is rejoined or the failed end condition is met. The step refers to the failed step in the main success scenario and has a letter associated with it; i.e., if step 3 fails the extension step is 3a.

Step	Condition	Action Description
1a	Backbone Communication System failure	Switch to Conventional Vehicle Highway System

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	Variable	Possible Variations
1a	Backbone Communication System failure	Switch to Conventional Vehicle Highway System

Related Information

The following table gives the information that is related to the use case.

Schedule:	ASAP
Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Provide Electronic Payment Services
Sub Use Case(s):	N/A
Channel To Primary Actor:	Backbone Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	Backbone Communication Failure

5.2.1 Summary of the Documental Analysis Findings

In the documental analysis, the extent to which the backbone communication class participates in fulfilling a use case successful scenario is monitored. All the examples use cases documentations shows that backbone communication class is an essential participant in having a successful scenario. This is true for almost all the documentation use cases of CAHS. **What will happen if a malfunction occurs in the backbone communication class?**

5.3 VISUAL ANALYSIS

Due to the limited time and resources, the object-oriented visualization of CAHS will focus only on one-way communication visualization, where only the actors are communicating with the system and subsystems. The system and subsystems will meet most requirements of the actors, but will not directly give specific answers to the actors. The steps that were taken for the partial development of an object- oriented modeling of conceptual CAHS are as follows:

5.3.1 Creating Use Case

5.3.1.1 Identified Actors of the System

The ITSC architecture provides us with the required information about the CAHS's actors. Actors are entities that interact with the proposed system asking for service or participating in providing some of the information required to fulfill the system capability.

The general-purpose actor illustrated in Figure 5.1, which captures characteristics common to all users with its subclasses, Human, Environment, Other Systems, and Systems.

The ITSC architecture elaborated in identifying the roles of the special actors that may interact with the proposed system. This detailed description was presented only to analyze the extent to which the system boundary may extend.

1 Human:

In Figure 5.5 Human Actors Categories are illustrated.

1a. Archived Data Administrator:

This actor represents the human operator who provides overall data management, administration, and monitoring duties for the CAHS data archive. This actor is responsible for the initiation of commands, requests, and queries that support the administration and management of the CAHS data archive.

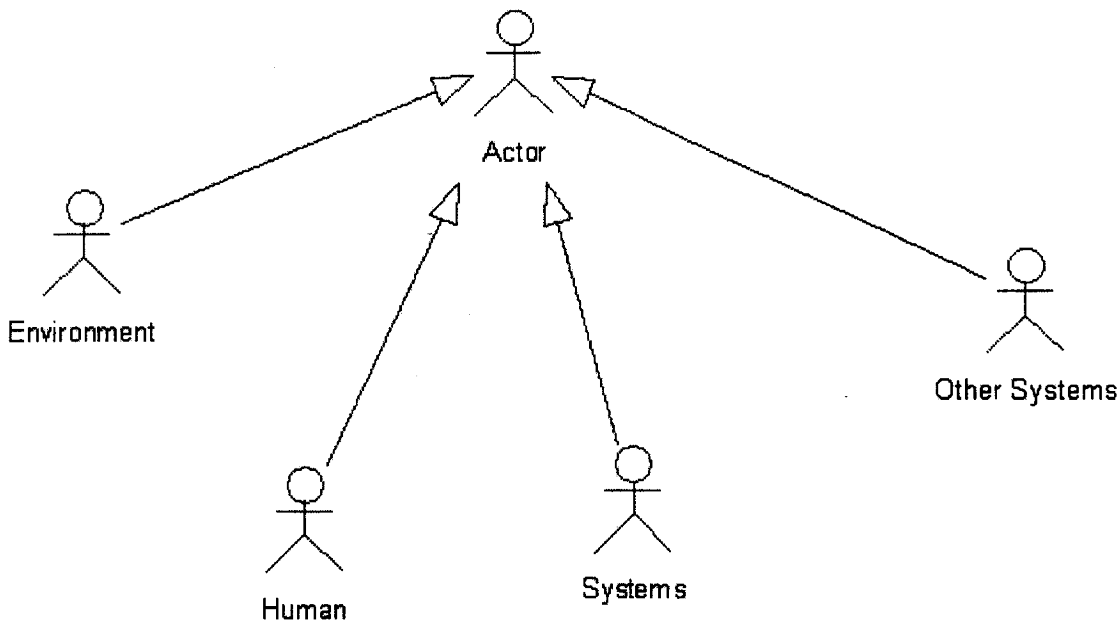

Figure 5.1 Object-oriented visualization of actor's categories in CAHS using UML

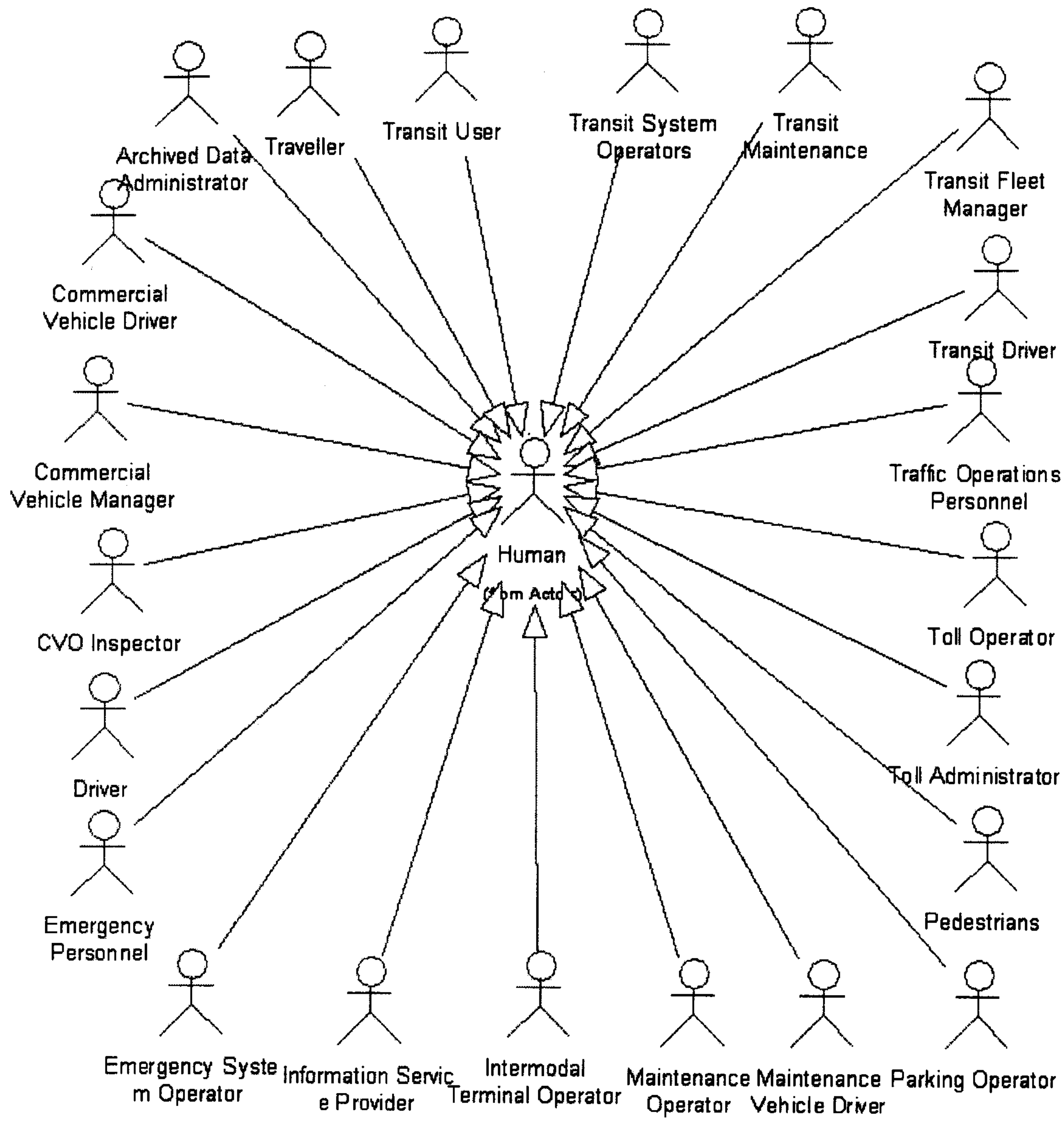

Figure 5.2 Object-oriented visualization of related Human actors for CAHS in UML

1b. Commercial Vehicle Driver:

This actor represents the human entity that operates vehicles transporting goods. This actor is responsible for the initiations of Commercial Vehicle driver and Vehicle information and requests to the Commercial Vehicle Managing System.

1c. Commercial Vehicle Manager:

This actor represents the human entities that are responsible for the dispatching and management of Commercial Vehicle fleets. This actor is responsible for the initiations of inquiries from fleet manager requesting data from Commercial Vehicle Management System.

1d. Commercial Vehicle Operation Inspector

This actor represents the human entities that perform regulatory inspection of Commercial Vehicles in the field. This actor is responsible for the initiations of manual override by the Commercial Vehicle Roadside Facility Inspector of automated pass/pull-in signage information and requests from the commercial vehicle inspector to operate the commercial vehicle inspection station.

1e. Driver:

This actor represents the human entity that operates a vehicle on the roadway. This actor is responsible for the initiations of driver commands to the Satellitic Vehicle and a traveller service request initiated by a driver or traveler. The request may result in a financial transaction, summoning of an emergency response, or initiation of another service at the behest of the driver.

1f. Emergency Personnel:

This actor represents personnel that are responsible for police, fire, emergency medical services, towing, and other special response team activities at an incident site. This actor is responsible for the initiation of current incident status information and requests from emergency personnel in the field for information and/or resources.

1g. Emergency System Operator:

This actor represents the human entity that monitors all CAHS emergency requests, and sets up pre-defined responses to be executed by an emergency management system. This actor is responsible for the initiations of emergency operator inputs supporting call taking, dispatch, and other operations and communications center operator functions

1h. Information Service Provider Operator:

This actor is the human entity that may be physically present at the ISP to monitor the operational status of the facility and provide human interface capabilities to Travelers and other ISP subsystems. This actor is responsible for the initiations of tuning and performance enhancement parameters to ISP algorithms.

1i. Intermodal Terminal Operator:

The actor is the human entity that operates the intermodal terminal subsystem. This actor is responsible for the initiations of inputs and control actions by an intermodal terminal operator.

1j. Maintenance Operator:

This actor represents the human entity that directly interfaces with the systems in the Maintenance Management subsystem. This actor is responsible for the initiations of information, control, and dispatching inputs from the maintenance operator.

1k. Maintenance Vehicle Driver:

This actor represents the human entity that operates any maintenance vehicle. This actor is responsible for the initiations of information and control actions provided by the driver of a maintenance vehicle.

1j. Parking Operator:

This actor represents the human entity that may be physically present at the parking lot facility to monitor the operational status of the facility. This actor is responsible for the initiations of local parking operator inputs that query current status and control the operation of the parking management system and request issued by a service provider for current parking service performance data.

1k. Pedestrians:

This actor represents the human entity that provides a specialized form of the traveller, who is not using any type of vehicle as a form of transport. This actor is responsible for the initiations of request for pedestrian crossing and sensed presence of pedestrians and other non-motor Vehicle travelers at roadway crossing or control points.

1l. Toll Administrator:

This actor represents the human entity that manages the back office payment administration systems for an electronic toll system. This actor is responsible for the initiations of instructions indicating the toll fees that are charged.

1m. Toll Operator:

This actor represents the human entity that manages the back office payment administration systems for an electronic toll system. This actor is responsible for the initiations of request for information from toll operators at toll collection sites.

1n. Traffic Operations Personnel:

This actor represents the human entity that directly interacts with vehicle traffic operations. This actor is responsible for the initiations of nominal pollution data compliance (reference) levels for each sector of an urban area, traffic operations requests for information, configuration changes, commands to adjust current traffic control strategies (e.g., adjusting signal timing plans, change DMS messages), and other traffic operations data entry.

1o. Transit Driver:

This actor represents the human entity. He is a special form of the Driver actor that receives and provides additional information that is specific to transit operations. This actor is responsible for the initiations of transit driver availability data that can be used to develop driver assignments and detailed operations schedules and transit driver emergency request as well as fare transaction data.

1p. Transit Fleet Manager:

This actor represents the human entity that is responsible for planning the operation of transit fleets; including monitoring and controlling the transit fleet route schedules and the transit fleet maintenance schedules. This actor is responsible for the initiations of instructions governing service availability, schedules, emergency response plans, transit personnel assignments, transit maintenance requirements, and other inputs that establish general system operating requirements and procedures.

1q. Transit Maintenance Personnel:

The actor represents the human entity that is actively responsible for monitoring, controlling, and planning the schedules for the maintenance of transit fleets. This actor is responsible for the initiations of current maintenance status of Vehicle.

1r. Transit System Operators::

This actor represents the human entities that are responsible for all aspects of the Transit subsystem operation including planning and management. This actor is responsible for the initiations of information and control provided by transit system operators involving many aspects of managing transit operations.

1s. Transit User:

This actor represents the human entities using public transit vehicles. This actor is responsible for the initiations of requests from transit user through an on-board or fixed location traveler information station.

1t. Traveler:

This actor represents any individual who uses the transportation services. This actor is responsible for the initiations of requests by travelers to summon assistance, requests travel information, makes a reservation, or requests any other traveler related service.

2 Environments:

Related Environmental Actors are illustrated in figure 5.6

2a. Environment:

This actor is the operational setting in which the CAHS interfaces operate. This actor is responsible for the initiations of atmospheric pollutant levels as monitored by air quality sensors and detection of specific localized hazards affecting the roadway.

2b. Potential Obstacles:

This actor represents any object that possesses the potential of being sensed and struck, and thus also possesses physical attributes. This actor is responsible for the initiations of the detection of an obstacle by vehicles or roadside equipment. Obstacles could include animals, other vehicles, pedestrians, or rocks in roadway, for example.

2c. Roadway:

This actor represents the physical conditions and geometry of the surface or space on or through which vehicles travel from an origin to a destination. This actor is responsible for the initiations of the detection and measurement of road characteristics such as friction coefficient and general surface and sub-surface conditions, road geometry and markings, for example.

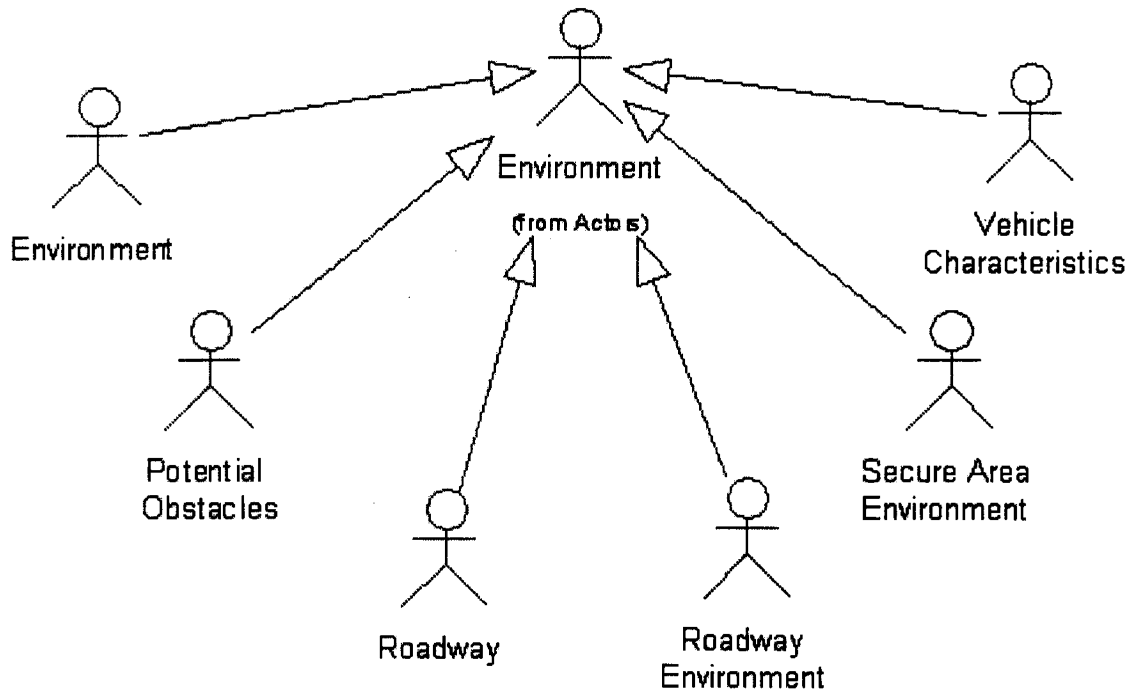

Figure 5.3 Object-oriented visualization of related Environmental actors for CAHS in UML

2d. Roadway Environment:

This actor represents the physical conditions surrounding the roadway itself. This actor is responsible for the initiations of the detection of specific localized hazards affecting the roadway, such as mud slides, avalanches, high winds, flooding, the weather and roadway conditions that serve as environmental data for sensors at the roadside or in maintenance vehicles or other types of vehicles.

2e. Secure Area Environment:

This actor represents public access areas that transit users frequent during trips. Areas include bus stops, park and ride (PAR) facilities, kiosks, and transit transfer and multimodal transfer locations. This actor is responsible for the initiations of characteristics (visual, audible, other) that are monitored by surveillance security systems by sensors.

2f. Traffic:

The traffic actor represents the collective body of vehicles that travel on surface streets, arterials, highways, expressways, tollways, freeways, or any other vehicle travel surface. This actor is responsible for the initiations of physical traffic characteristics that are monitored and translated into macroscopic measures like occupancy, volume, density, and average speed. Point measures support presence detection and individual vehicle measures like speed. This may also include detection of non-vehicular traffic in the roadway, such as bicycles and pedestrians.

2g. Vehicle Characteristics:

This actor represents the external view of an individual vehicle. This actor is responsible for the initiations of the physical or visible characteristics of an individual vehicle that can be measured to classify a vehicle and imaged to uniquely identify a vehicle.

3 Systems:

Related system actors are illustrated in figure 5.7.

3a. Archived Data User Systems:

This actor represents the systems users employed to access archived data. This actor is responsible for the initiation of a user requests that initiates data mining, analytical processing, aggregation or summarization, report formulation, or other advanced processing and analysis of archived data. The requests also include information that is used to identify and authenticate the user and support electronic payment requirements, if any.

3b. Basic Vehicle:

This actor represents the basic vehicle platform that interacts with and hosts CAHS electronics. This actor is responsible for the initiations of information provided to on-board CAHS equipment from the vehicle platform, indicating current vehicle status.

3c. Commercial Vehicle:

The actual commercial vehicle along with the special aspects of large commercial vehicles and vehicles designed to carry cargo that extend beyond the characteristics defined for the Basic Vehicle. This actor is responsible for the initiations of commercial vehicle, driver, and cargo safety status measured by on-board CAHS equipment.

3d. Customs Agency:

This border inspection agency performs the primary regulatory inspection function at an international Port of Entry. In Canada, this is generally the Canada Customs and Revenue Agency, but can also include Immigration, Agriculture, and other inspection agencies. This actor is responsible for the initiations of customs request for the electronic manifest associated with a container; requires proper authentication.

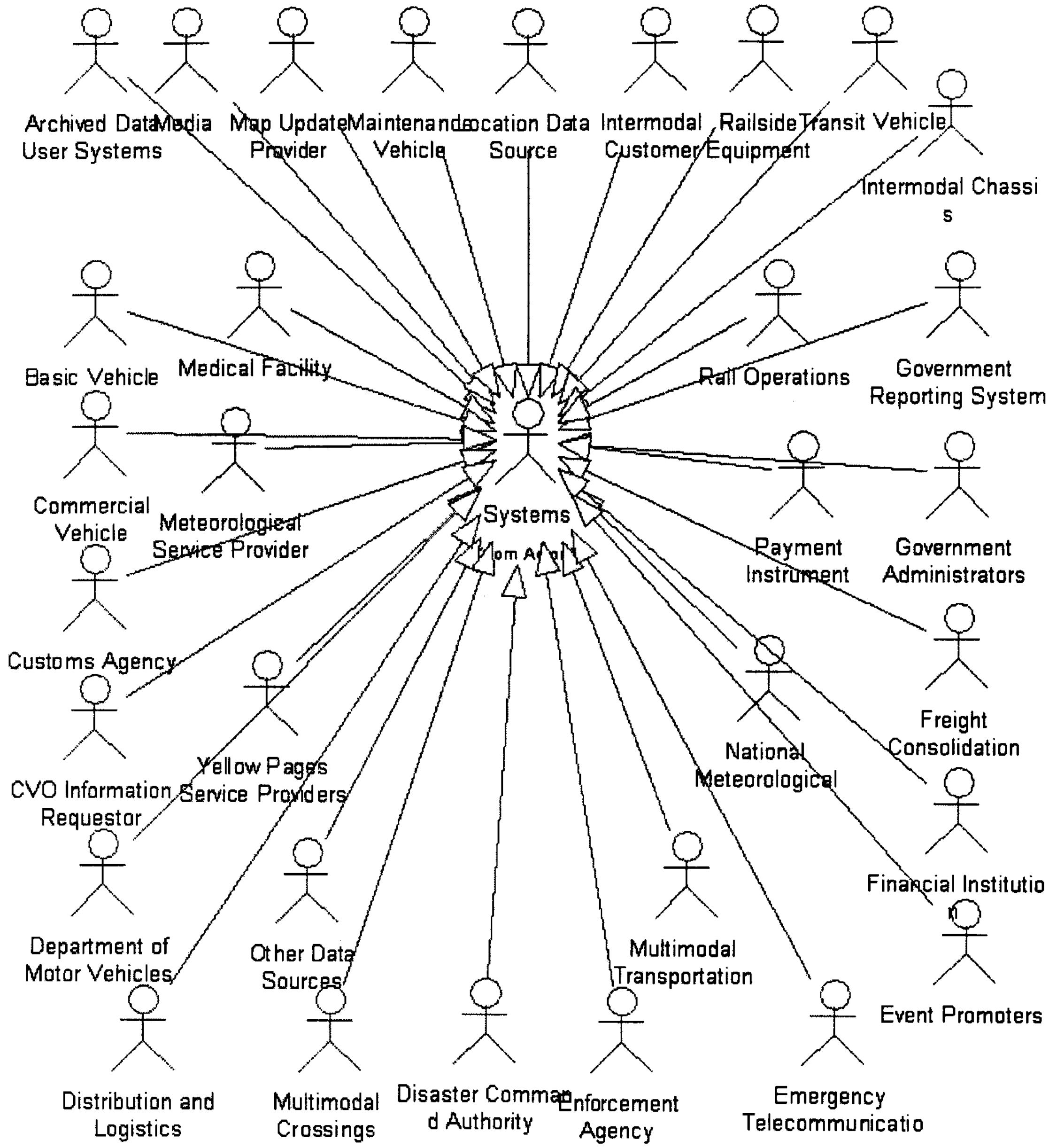

Figure 5.4 Object-oriented visualization of related Systems actors for CAHS in UML

3e. Commercial Vehicle Operator Information Requestor:

This actor represents any organization requesting CVO information. This actor is responsible for the initiations of instructions to commercial vehicle management and/or information systems indicating which vehicles are to be allowed to pass and which are out of service or has not been credentialed.

3f. Department of Motor Vehicles:

This actor represents a specific (provincial) public organization responsible for registering vehicles, e.g., the Ministry of Transportation or Municipalities. This actor is responsible for the initiations of request supporting registration data based on license plate read during violation.

3g. Disaster Command Authority:

The Disaster Command Authority terminal represents the systems used by authorities that provide command-and-control leadership for coordinated disaster response. This actor is responsible for the initiations of information supporting the coordination of disaster and emergency response assets and activities.

3h. Distribution and Logistics Management Provider:

This actor represents system that provides intermodal logistics support, and support for the efficient distribution of freight across transport systems and modes. This actor is responsible for the initiations of request to a container to provide its location and may require the requestor to authenticate their identity.

3i. Emergency Telecommunications System:

This actor represents the human entity that monitors all CAHS emergency requests, (including those from the E911 Operator) and sets up pre-defined responses to be executed by an emergency management system. This actor is responsible for the initiations of the notification of an incident including its nature, severity, and location.

3j. Enforcement Agency:

This terminator represents an external entity which receives reports of violations detected by various CAHS facilities, e.g. individual Vehicle emissions, toll violations, speed or red light running violation, CVO violations, etc. This actor is responsible for the initiation of response from law enforcement agencies to violations notification request.

3k. Event Promoters:

This actor represents external special event sponsors that have knowledge of events that may impact travel on roadways or other modal means. This actor is responsible for the initiations of plans for major events possibly impacting traffic.

3l. Financial Institution:

This actor represents the organization that handles all electronic fund transfer requests to enable the transfer of funds from the user of the service to the provider of the service. This actor is responsible for the initiations of response to transaction requests normally dealing with a request for payment.

3m. Freight Consolidation Station:

An intermediate point (usually an Intermodal terminal located at a port) prior to (or after) container-based shipping, where less-than-container load or less-than-truckload cargoes are consolidated into full container loads (or full containers are disbursed), for cost-effective Intermodal shipping. This actor is responsible for the initiations of location within an Intermodal facility that a container is to be received at or delivered to; and may include guidance as well as location.

3n. Government Administrators:

This terminal represents those public organizations responsible for regulating commercial vehicle operations, e.g., provincial commerce offices, and provincial agency responsible for transportation, provincial finance department, and Transport Canada. Regulatory Agencies are envisioned to be an integral part of the CAHS Commercial Vehicle Operations (CVO) as they will be directly involved with issuance of licenses, permits and other credentials for pre clearance, provide database information to support most CVO services, and will receive, distribute, and audit CVO related taxes. This actor is responsible for the initiation of regulations imposed on Commercial Vehicle Administration agencies including safety ratings, facility locations and credential fee structure.

3o. Government Reporting Systems:

This actor represents the system and associated personnel that prepare the inputs to support the various local, provincial, and federal government transportation data reporting requirements using data collected by CAHS systems. This actor is responsible for the initiation of the acknowledgement of satisfactory receipt of information used as input to government data systems or a report identifying problems or issues with the data submittal.

3p. Intermodal Chassis:

This actor represents the chassis, which is the frame on wheels that an intermodal container is secured to for roadway transport by a truck. This actor is responsible for the initiation of measurement from chassis systems like brakes, tires, and fasteners and indication of operational readiness of a chassis.

3q. Intermodal Customer:

This actor represents the originator of an order to move or the final recipient of a cargo shipment. This actor is responsible for the initiations of the notice confirming the arrival and transfer of control of a container at a container handling facility, for example an intermodal terminal, request from a shipper for services to handle shipping of a container or freight load, and request from consignor or consignee for the current location and transit status of a freight shipment.

3r. Location Data Source:

This actor represents an external entity, which provides accurate position information. This actor is responsible for the initiation of the information, which the geographical position of a traveler or a vehicle.

3s. Maintenance Vehicle:

This actor represents a specialized form of the basic vehicle used by maintenance fleets. This actor is responsible for the initiation of maintenance vehicle status to be measured by on-board CAHS equipment.

3t. Map Update Provider:

This actor represents a third-party developer and provider of digitized map databases used to support CAHS services. This actor is responsible for the initiation of Map update, which could include a new underlying static or real-time map or map layer(s) update.

3u. Media:

This actor represents the information systems that provide traffic reports, travel and weather conditions, and other transportation-related news services to the travelling public through radio, television, and other media. This actor is responsible for the initiation of request from the media for current traffic information and incident information that is collected by the media through a variety of mechanisms (e.g., radio station call-in programs, and air surveillance).

3v.Medical Facility:

This actor represents Medical Facilities such as hospitals, trauma centres, field emergency treatment facilities and any other location capable of receiving injured persons and providing emergency care. This actor is responsible for the initiation of specific care capabilities and available space in a medical facility.

3w. Meteorological Service Provider:

This actor represents the providers of value-added sector specific meteorological services. This actor is responsible for the initiation of prediction of weather information, customized to a particular users needs' and the region.

3x. Multimodal Crossings:

This actor represents the control equipment that interfaces to a non-road based transportation system at an interference crossing with the roadway. This actor is responsible for the initiation of prediction of indication of operational status and pending requests for right-of-way from equipment supporting the non-highway mode at multimodal crossings.

3y. Multimodal Transportation Service Provider:

This terminator provides the interface through which transportation service providers can exchange data with CAHS. This actor is responsible for the initiation of operational information from alternate passenger transportation modes including air, rail transit, taxis, and ferries. Content may include a catalog of available information, the actual information to be archived, and associated metadata that describes the archived information.

3z. National Meteorological Service:

This actor provides weather, hydrologic, and climate information and warnings of hazardous weather including thunderstorms, flooding, hurricanes, tornadoes, winter weather, tsunamis, and climate events. This actor is responsible for the initiation of accumulation forecast and current weather data (e.g., temperature, pressure, wind speed, wind direction, humidity, precipitation, visibility, light conditions, etc.).

3aa. Other Data Sources:

This actor represents the myriad systems and databases containing data not generated from subsystems and terminators represented in the national CAHS architecture that can provide predefined data sets to the CAHS archive. This actor is responsible for the initiation of data extracted from other data sources. A wide range of CAHS and non-CAHS data and associated metadata may be provided.

3ab. Payment Instrument:

This actor represents the entity that enables the actual transfer of funds from the user of a service to its provider. This actor is responsible for the initiation of information passing from a payment instrument (e.g., smart card) to a payment device to provide electronic payment of some kind (e.g., toll, parking, fare) by traveller. In most cases the payment can be credited.

3ac. Rail Operations:

This is roughly the railroad equivalent to a highway traffic management centre. This actor is responsible for the initiation of real-time notification of railway-related incident or advisory.

3ad. Railside Equipment:

This actor represents train interface equipment (usually) maintained and operated by the railroad and (usually) physically located at or near a grade crossing. This actor is responsible for the initiation of information for a train approaching a highway-rail intersection that may include direction and allow calculation of approximate arrival time and closure duration.

3ae. Transit Vehicle:

This actor represents a specialized form of the basic vehicle used by transit service providers. This actor is responsible for the initiation of transit vehicle status measured by on-board CAHS equipment.

3af. Yellow Pages Service Providers:

This actor represents the individual organizations that provide any services oriented towards the traveller. This actor is responsible for the initiation of information supplied by a service provider (e.g., a hotel or restaurant), identifies the service provider and provides details of the service offering. This flow covers initial registration of a service provider and subsequent submittal of new information and status updates so that data currency is maintained.

4 Other Systems:

Other related system actors are illustrated in figure 5.5.

4a. Other Archives:

This actor represents distributed archived data systems or centres whose data can be accessed and shared with a local archive. This actor is responsible for the initiation of catalog data, metadata, published data, and other information exchanged between archives to support data synchronization and satisfy user data requests.

4b. Other Commercial Vehicle Administration Subsystem:

This actor is intended to provide a source and destination for CAHS data flows between peer (e.g. inter-regional) commercial vehicle administration functions. This actor is responsible for the initiation of tax and credential fee information exchanged between cooperating commercial vehicle administration offices and instructions to commercial vehicle managing and/or information systems indicating which vehicles are to be allowed to pass and which are out of service or have not been credentialed.

4c. Other Emergency Management:

Representing other emergency management centres, systems or subsystems, this terminal provides a source and destination for CAHS data flows between various communications centres operated by public safety agencies as well as centres operated by other allied agencies and private companies that participate in coordinated management of highway

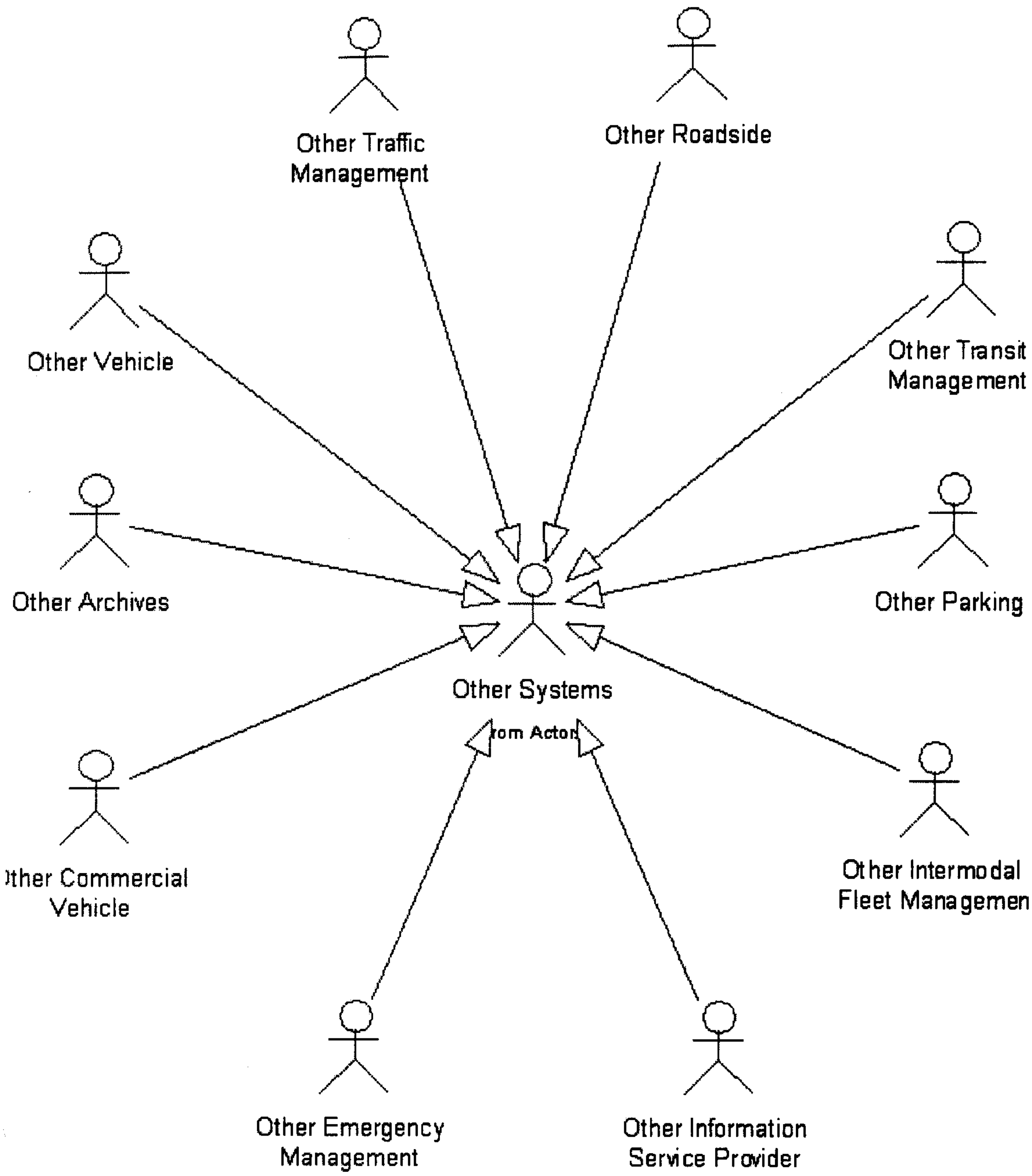

Figure 5.5 Object-oriented visualization of related Other system actors foe CAHS in UML

related incidents. This actor is responsible for the initiations of reports of an identified incident including incident location, type, severity and other information necessary to initiate an appropriate incident response. It is also responsible for the start of incident response procedures, resource coordination, and current incident response status that are shared between allied response agencies to support a coordinated response to incidents. This flow also coordinates a positive hand off of responsibility for all or part of an incident response between agencies.

4d. Other Information Service Provider:

This actor represents other distinct information service providers, this actor is intended to provide a source and destination for CAHS data flows between peer information and service provider functions. This actor is responsible for the initiation of coordination and exchange of transportation information between centres.

4e. Other Intermodal Fleet Management System:

This actor represents another intermodal fleet management system that manages systems for individual fleets of intermodal transport companies. This actor is responsible for the initiation of exchange of information between different intermodal freight management centers regarding cargo or intermodal container movement and information regarding whether a container has been released.

1.16.7 4f Other Parking:

This actor represents another parking facility, system or subsystem, this terminator provides a source and destination for information that may be exchanged between peer parking systems. This actor is responsible for the initiation of information that enables parking management activities to be coordinated between different parking operators or systems in a region.

4g. Other Roadside:

This actor represents another roadside element or system. This actor provides a source and destination for information that may be exchanged between peer roadside elements. This actor is responsible for the initiation of control data for roadside devices that is exchanged between roadside devices or from a maintenance management system to the roadside.

4h. Other Traffic Management:

This actor represents another traffic management centre. This actor is intended to provide a source and destination for CAHS data flows between peer traffic management functions. This actor is responsible for the initiation of traffic information exchanged between TMC's. Normally such information would include traffic incidents, congestion data, traffic data, signal timing plans, and real-time signal control information.

4i. Other Transit Management:

This actor represents another transit management centre. This actor is intended to provide a source and destination for CAHS data flows between peer transit management functions. This actor is responsible for the initiation of co-ordination information between local/regional transit organizations including schedule, on-time information, connection co-ordination, and ridership.

4j. Other Vehicle:

This actor represents a vehicle (of any four vehicle types) that is neighboring the basic vehicle, where the basic vehicle is equipped to support vehicle-to-vehicle communication and coordination. This actor is responsible for the initiations of any type of advanced vehicle-to-vehicle communication.

5.3.1.2 Created Context diagram

The detailed description of the various types of actors that may interact with the system and their rolls help us to identify the system boundary. The boundary in a complex system as Canadian Automated Highway System (CAHS) is difficult and challenging task. The developers have to have a clear idea about the space where they have authority to change in order to implement the need improvements and check its impact. Figure 5.6 illustrates the CAHS context.

5.3.1.3 Created Use Cases and diagrams:

In order to prepare the use case view of a CAHS, the requirements were subject to analysis and visualization in UML to extract the possible required capability of the system. Figure 5.7 to 5.16 are the decomposition of system use cases. The degree of use case decomposition is related to what are the capabilities that are intended from this partial development. The goal as we state earlier is to capture the problems within the ITSC architecture in order to find solution that will increase the effectiveness of the architecture in forming the base of development of any ITS system in Canada. The numbers that are given to the use cases are the same number of the related functional requirement given in the ITSC architecture documentation.

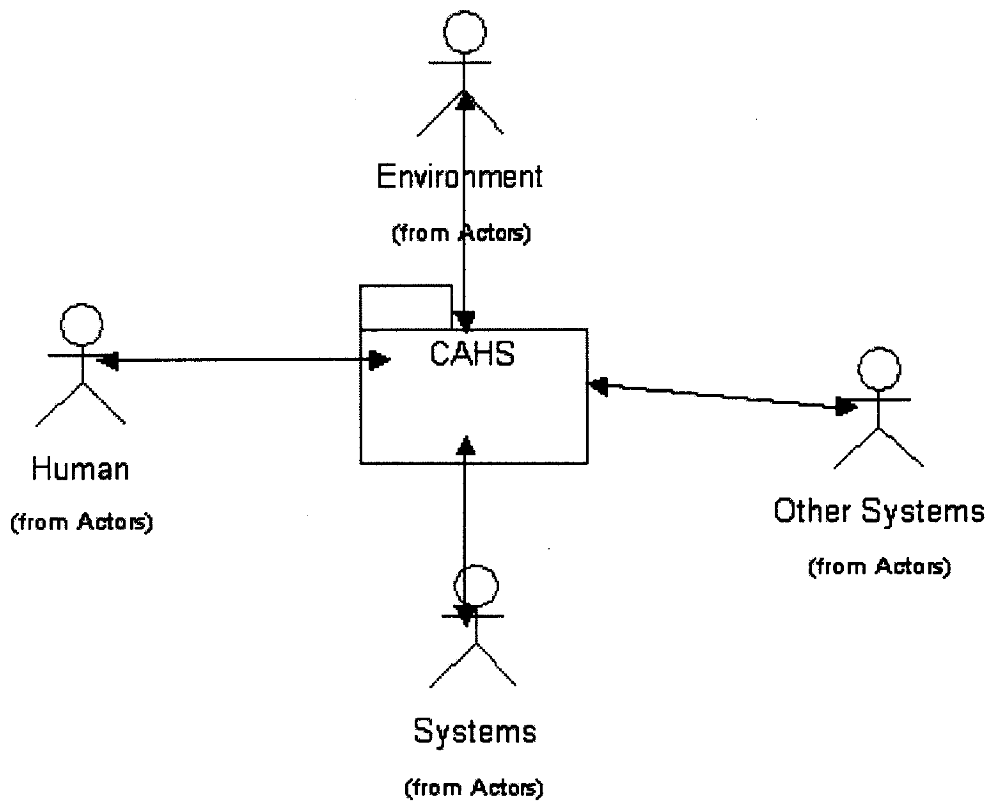

Figure 5.6 Object-oriented visualization of CAHS's context in UML

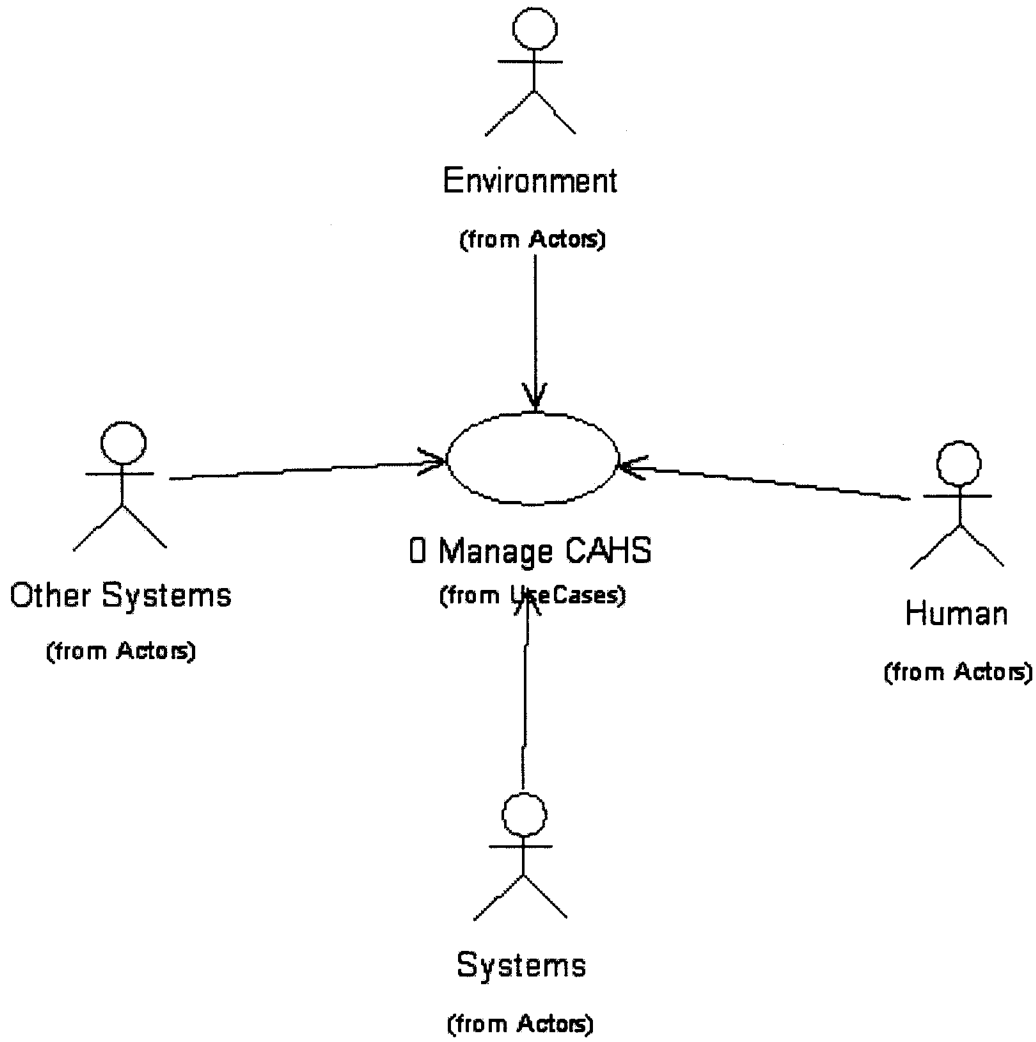

Figure 5.7 Object-oriented visualization of 0.0 Manage CAHS use case for CAHS in UML

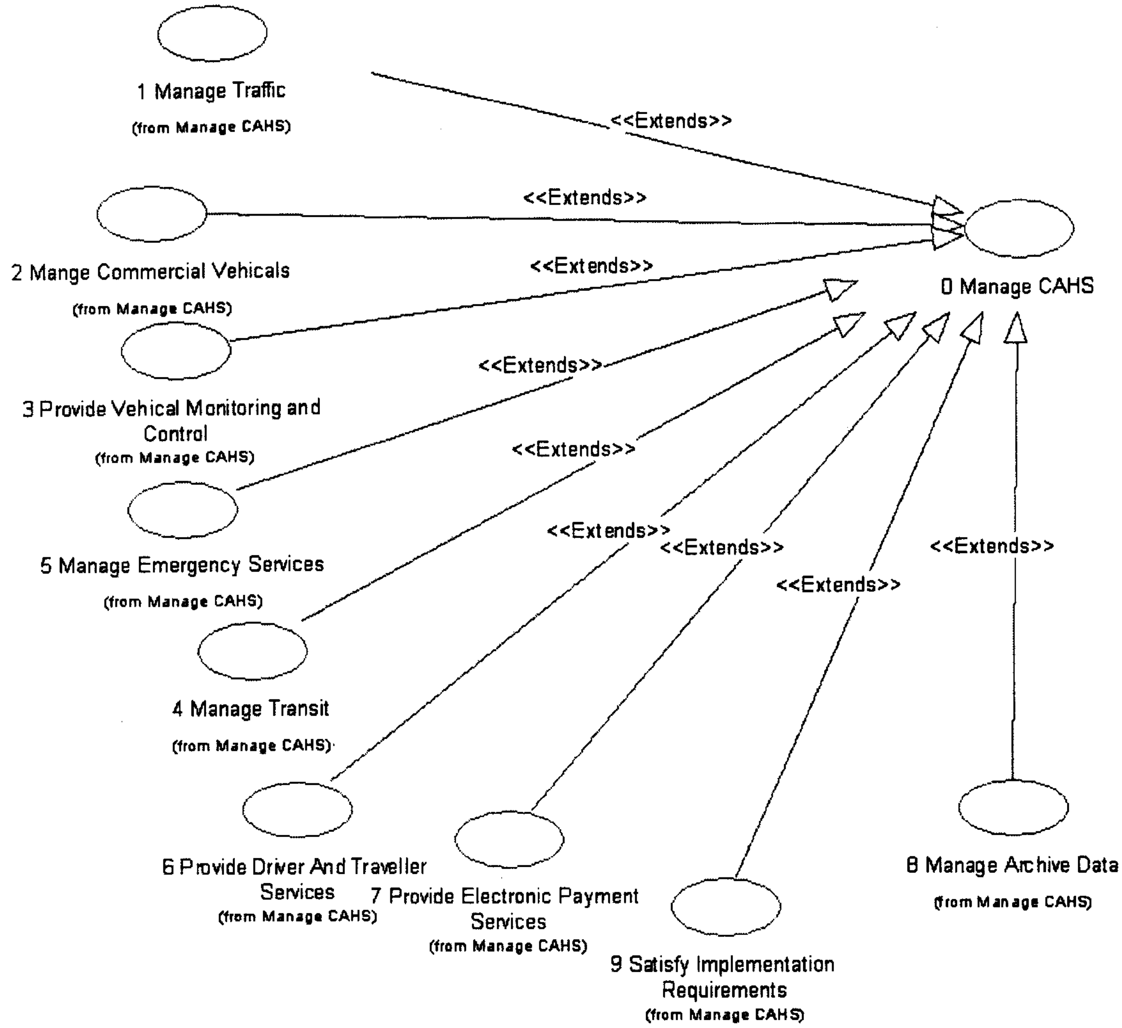

Figure 5.8 Object-oriented visualization of related 0.0 Manage CAHS use case for CAHS in UML

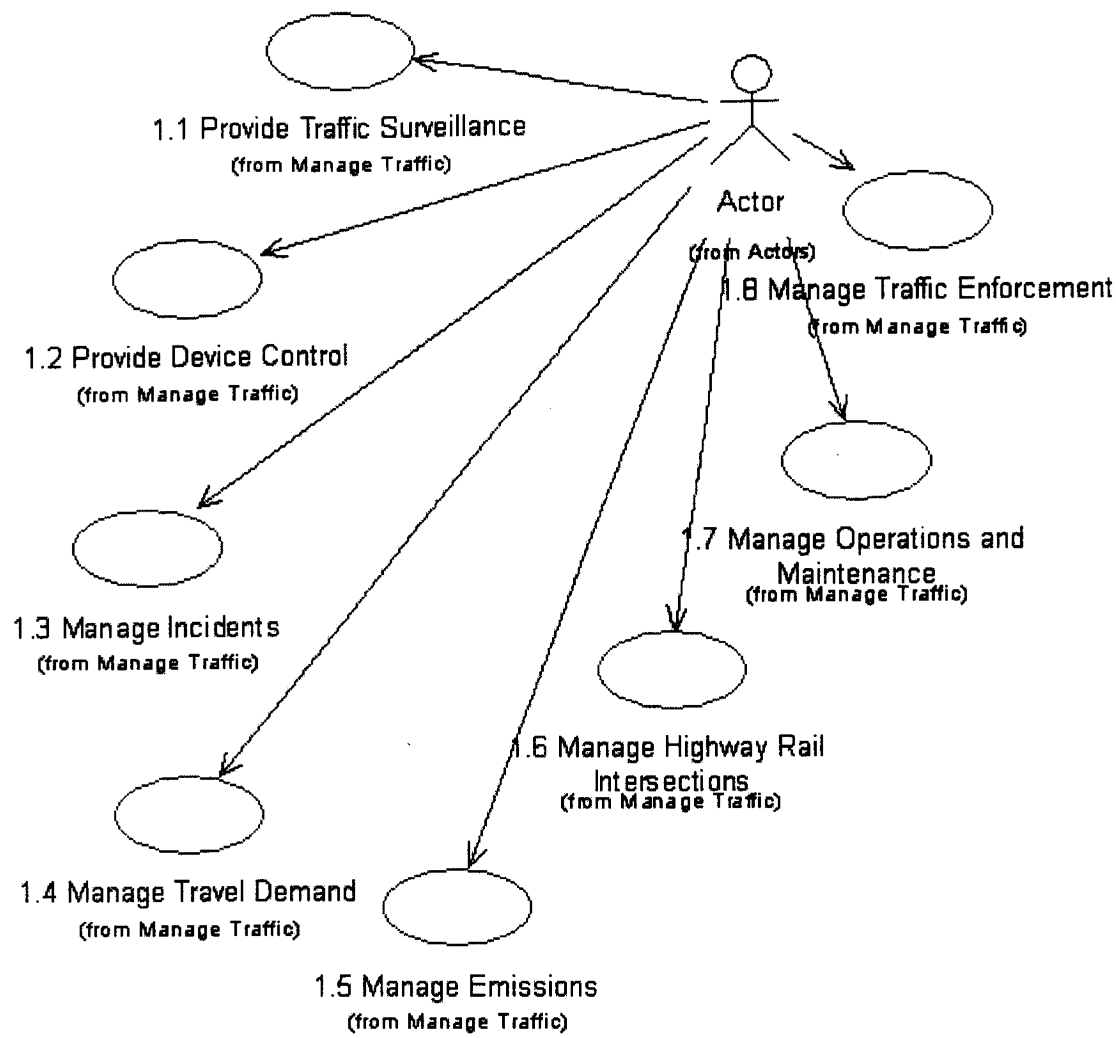

Figure 5.9 Object-oriented visualization of 1 Manage Traffic use case for CAHS in UML

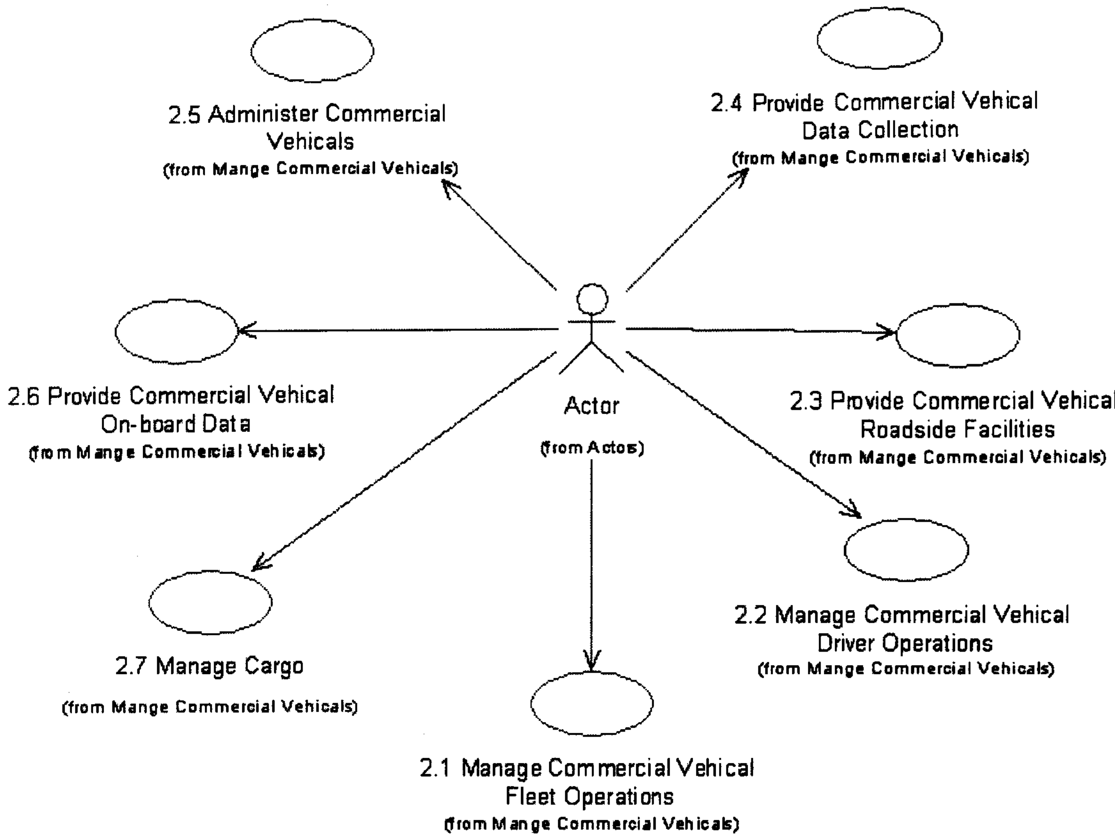

Figure 5.10 Object-oriented visualization of related 2 Manage Commercial Vehicle use case for CAHS in UML

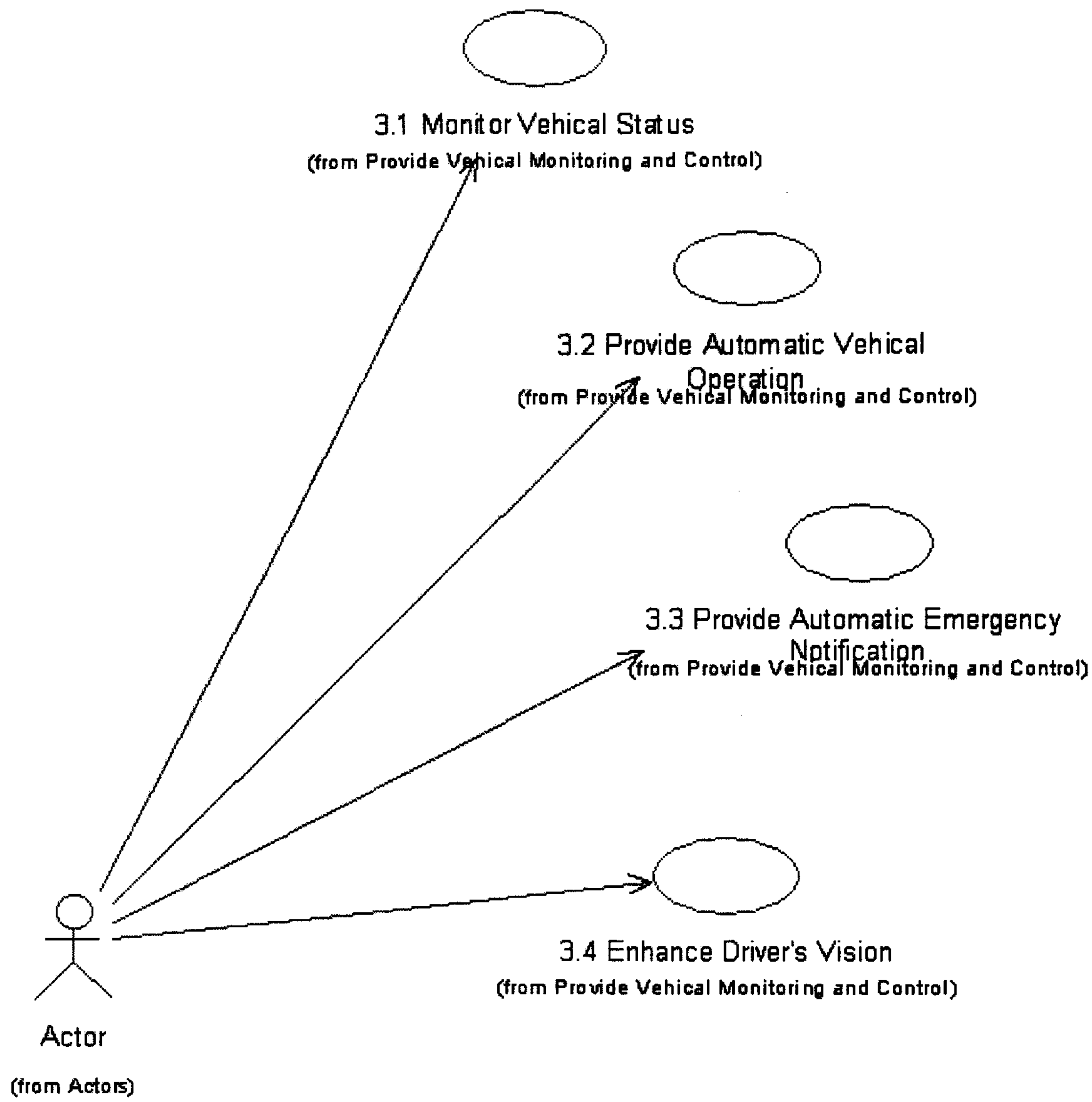

Figure 5.11 Object-oriented visualization of related 3 Provide Vehicle Monitoring and Control use case for CAHS in UML

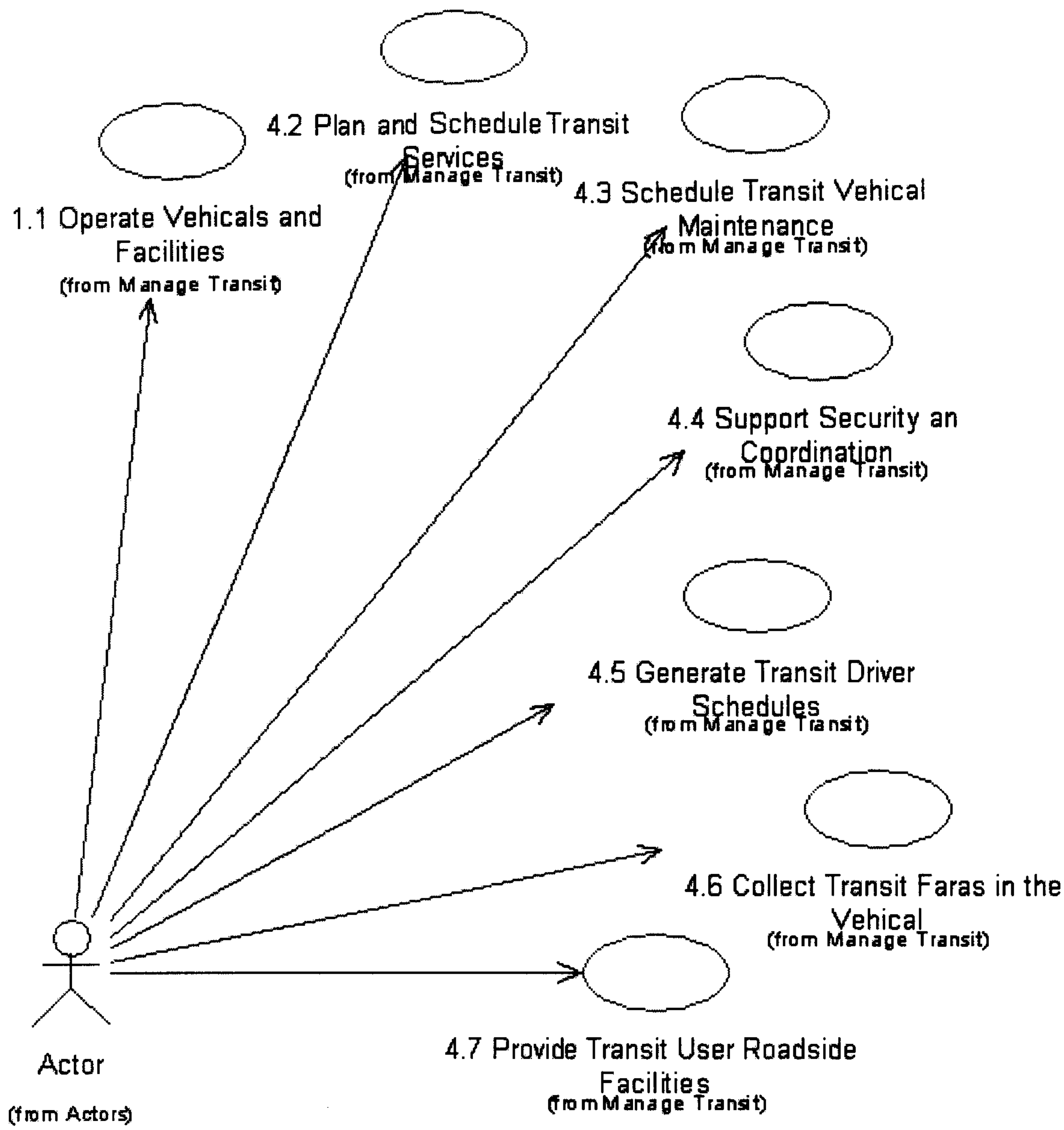

Figure 5.12 Object-oriented visualization of related 4 Manage Transit use case for CAHS in UML

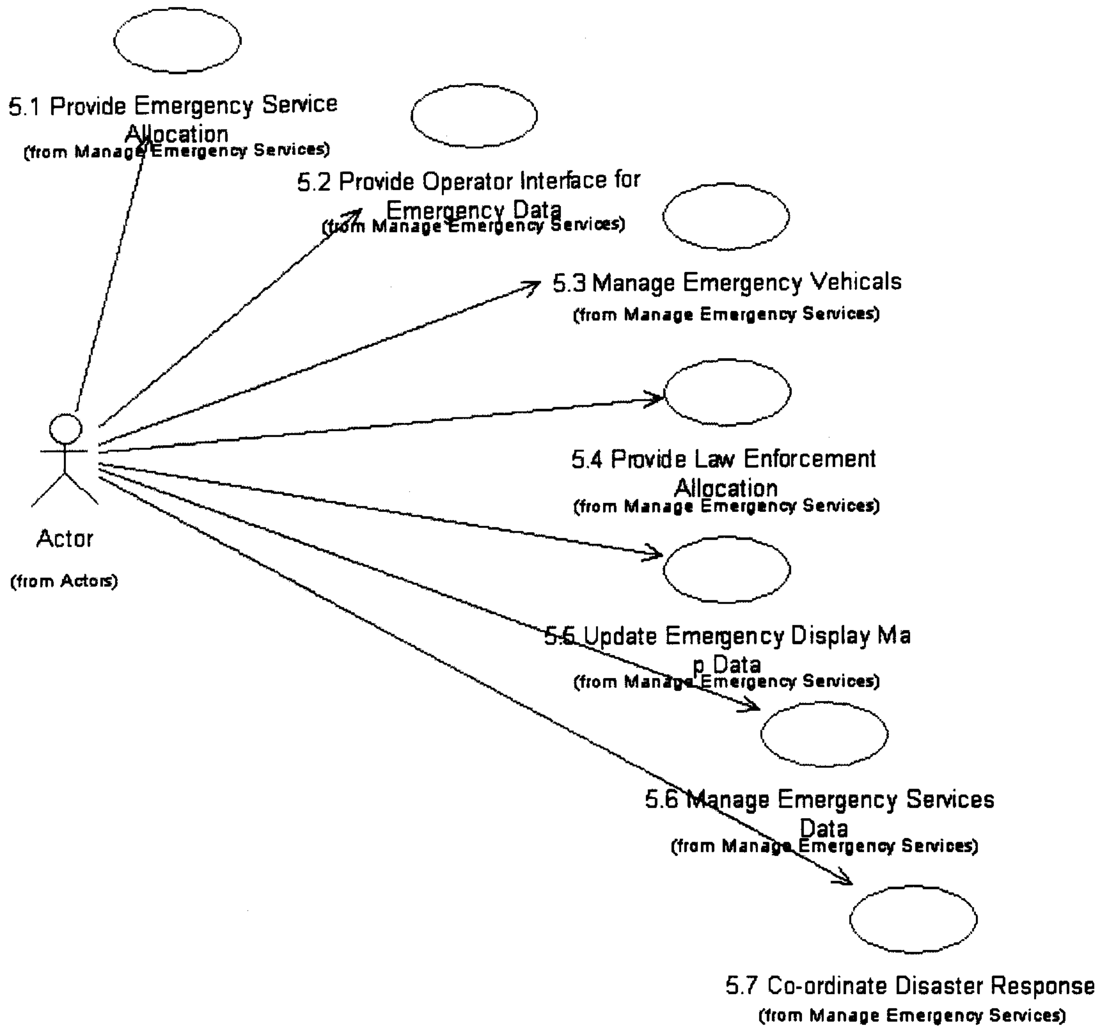

Figure 5.1 3 Object Oriented visualization of related 5 Manage Emergency Services use case for CAHS in UML

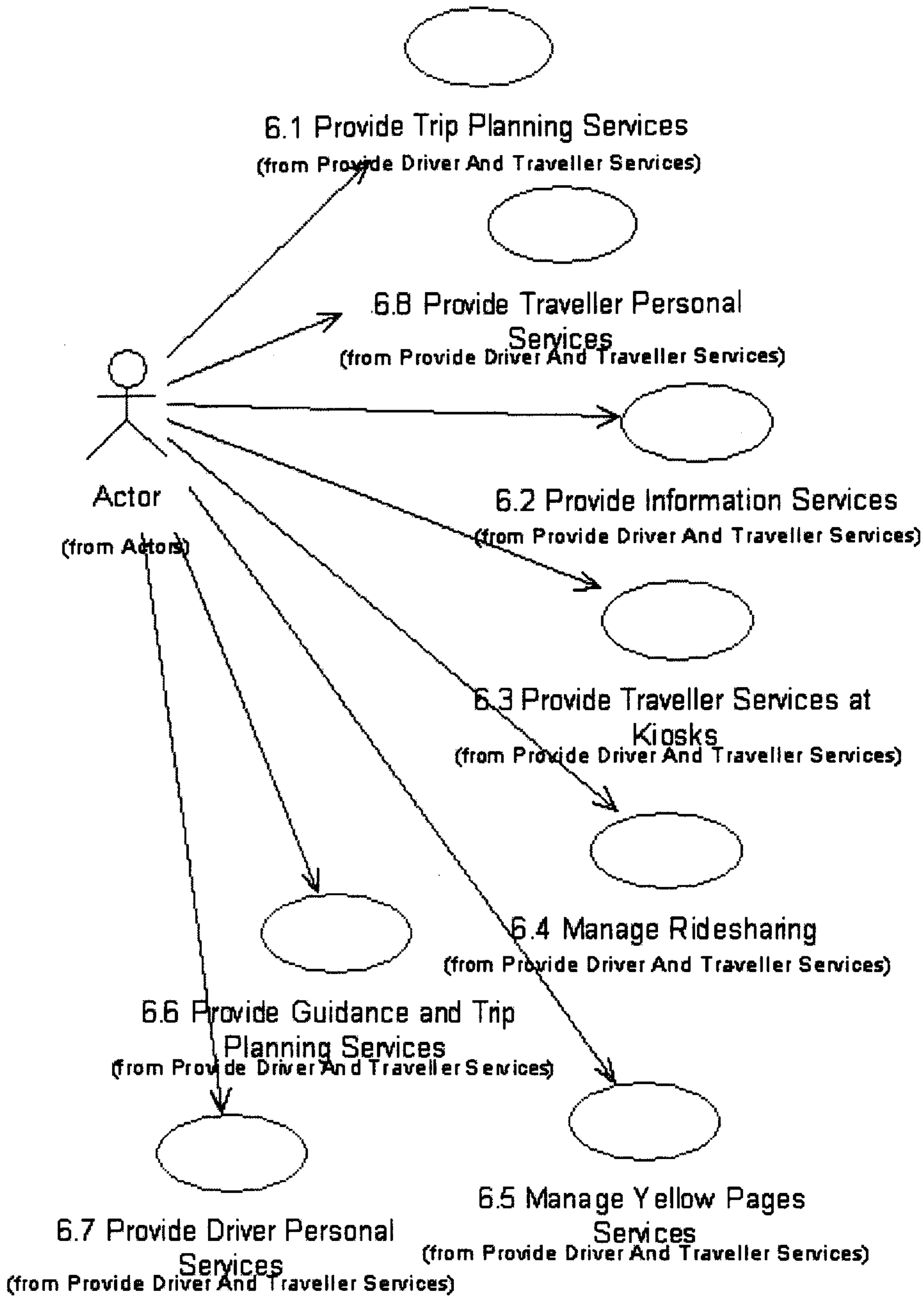

Figure 5.14 Object1-oriented visualization of related 6 Provide Driver and Traveller Services use case for CAHS in UML

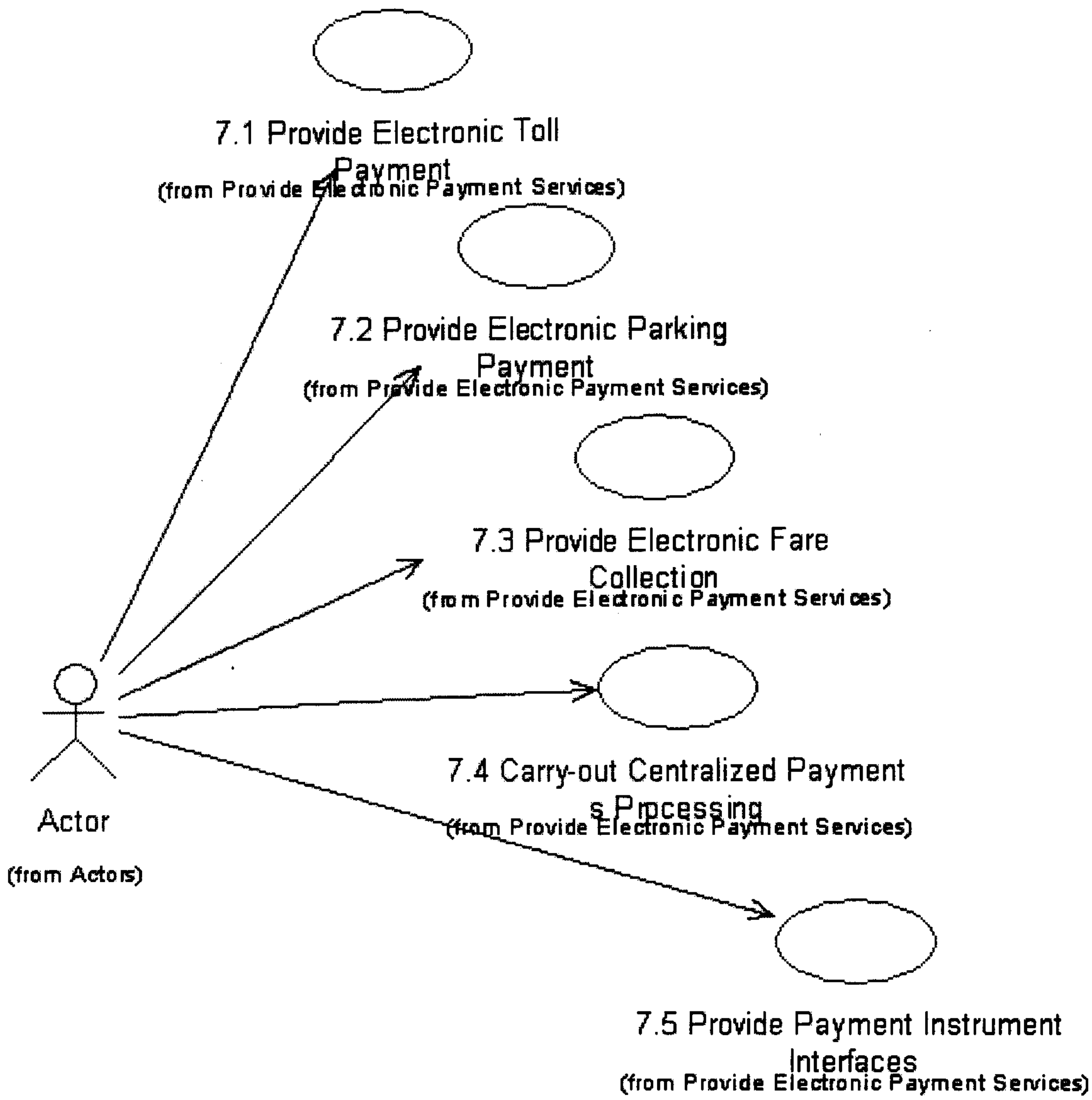

Figure 5.15 Object-oriented visualization of related 7 Provide Electronic Payment Services use case for CAHS in UML

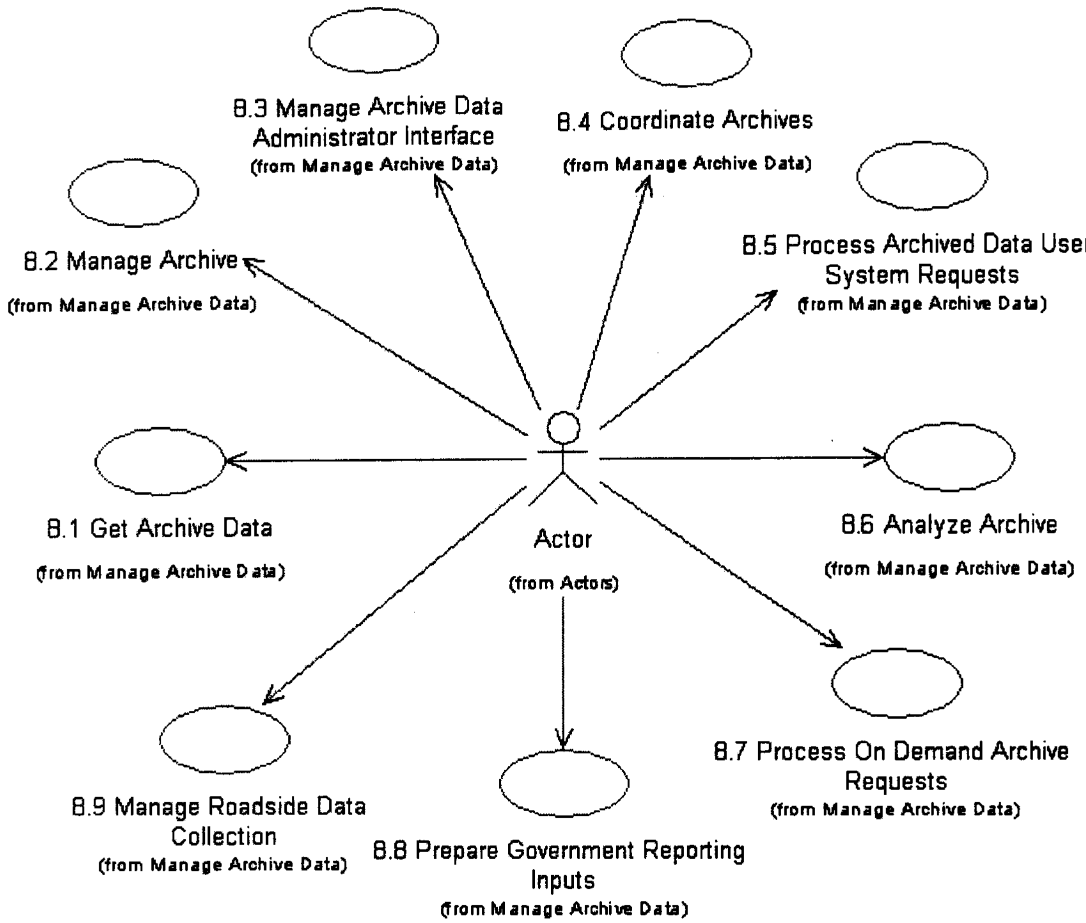

Figure 5.16 Object-oriented visualization of related 8 Manage Archive Data use case for CAHS in UML

5.3.2 Classes in the CAHS Problem

The requirement analysis of CAHS targeting the capture of a common description of related objects. These objects have common attributes, operations, relations, and semantics. The class that represents related groups within CAHS was identified using object-oriented techniques. The initial list of classes with a general description is illustrated in Table 5.2.

5.3.3 Visualizing Class Diagram of CAHS conceptual project

Class Diagram of the CAHS that Represent the Physical Architecture Overview of the System are Illustrated in Figure 5.17. The rest of CAHS classes can be found in APPENDIX C.

5.3.4 Visualizing Scenarios of the CAHS conceptual project

The scenarios of the CAHS are captured using Sequence Diagrams illustrated in Figure 5.18 to Figure 5. 22. The rest of CAHS sequence diagrams can be found in APPENDIX C.

5.3.5 Visualizing attributes and functions of CAHS conceptual project classes

The attributes and functions of CAHS conceptual project classes illustrated in Figure 5.23 to Figure 5. 52. The rest of CAHS conceptual project classes can be found in APPENDIX C.

Table 5.2 CAHS classes with a general description

Class name		Class General Description
Automated Highway System	1	Manage traffic information and control.
Centres	2	Location of the centres subsystems.
Wayside	3	Location of the Wayside subsystems.
Travellers	4	Location of the Travellres subsystems.
Vehicles	5	Location of the Vehicles subsystems.
Backbone Communication	6	Manage traffic information and control communication
Archived Data Management	8	Manage traffic information and control. Archived Data
Commercial Vehicle Administration	9	Manage traffic information and control. Commercial Vehicle Administration.
Emissions Management	10	Manage traffic information and control. Emission Management
Fleet and Freight Management	12	Manage traffic information and control. Fleet and Freight Management
Information Service Provider	13	Manage traffic information and control. Information Service Provider
Maintenance Management	14	Manages traffic information and control Maintenance Management.
Toll Administration	15	Manages traffic information and control Toll Administration.
Traffic Management	16	Manages traffic information and control Traffic Management.
Transit Management	17	Manages traffic information and control Transit Management.
Emergency Management	18	Manage traffic information and control. Emergency Management.
Commercial Vehicle Check	19	Manages traffic information and control Commercial Vehicle Check.
Intermodal Terminal	20	Manages traffic information and control Intermodal Terminal.
Parking Management	21	Manages traffic information and control Parking Management.
Roadway	22	Manages traffic information and control Roadway.
Toll Collection	23	Manage traffic information and control. Toll Collection.
Personal Information Access	24	Manages traffic information and control Personal Information Access.
Remote Traveller Support	25	Manages traffic information and control Remote Traveller Support
Commercial Vehicle	26	Manages traffic information and control Commercial Vehicle.
Emergency Vehicle	27	Manage traffic information and control.
Intermodal Container	28	Manages traffic information and control Intermodal Container.
Maintenance Vehicle	29	Manage traffic information and control. Maintenance Vehicle.
Transit Vehicle	30	Manage traffic information and control Transit Vehicle
Vehicle	31	Manage traffic information and control Vehicle.

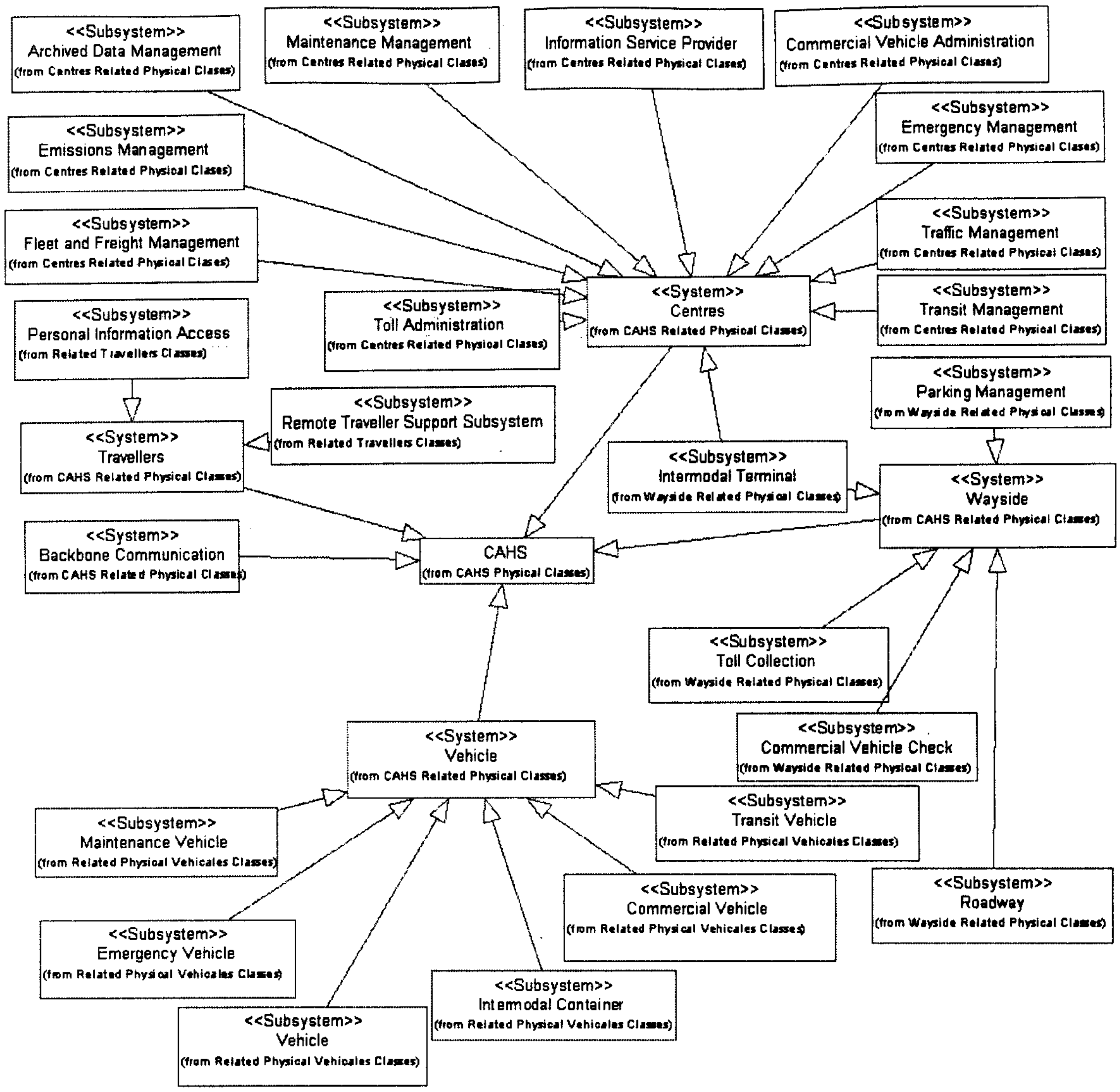

Figure 5.17 Object-oriented visualization of CAHS's Physical Classes in UML

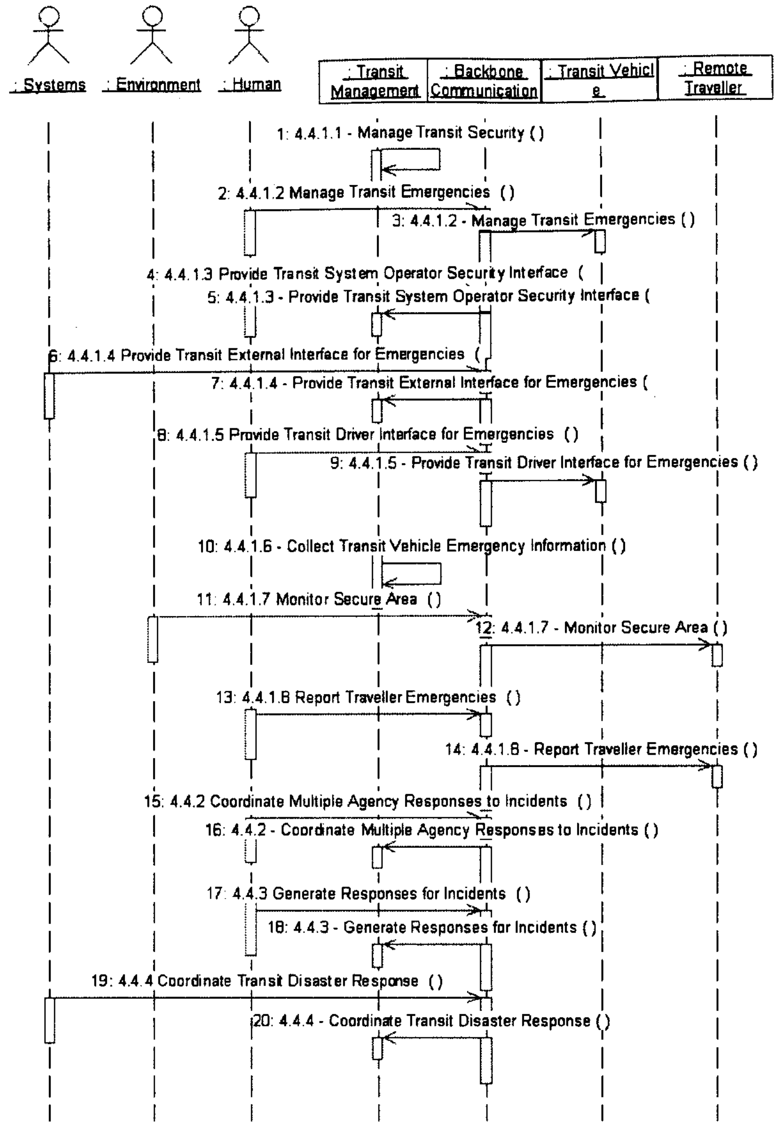

Figure 5.18 Object-oriented visualization of Sequence Diagram 4A.4 Support Security and Coordination for CAHS in UML

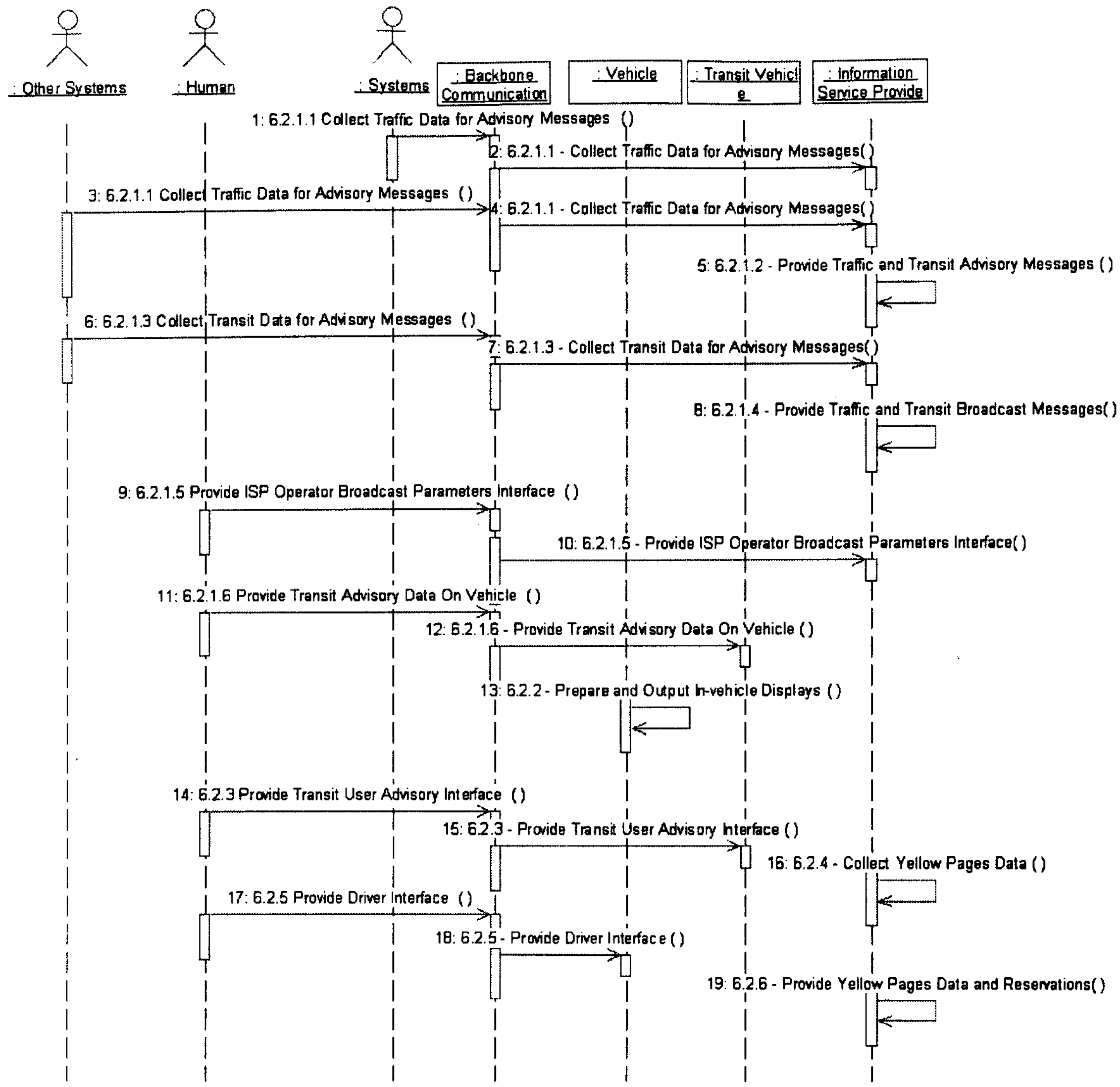

Figure 5.19 Object-oriented visualization of Sequence Diagram 6A.2 Provide Information Services for CAHS in UML

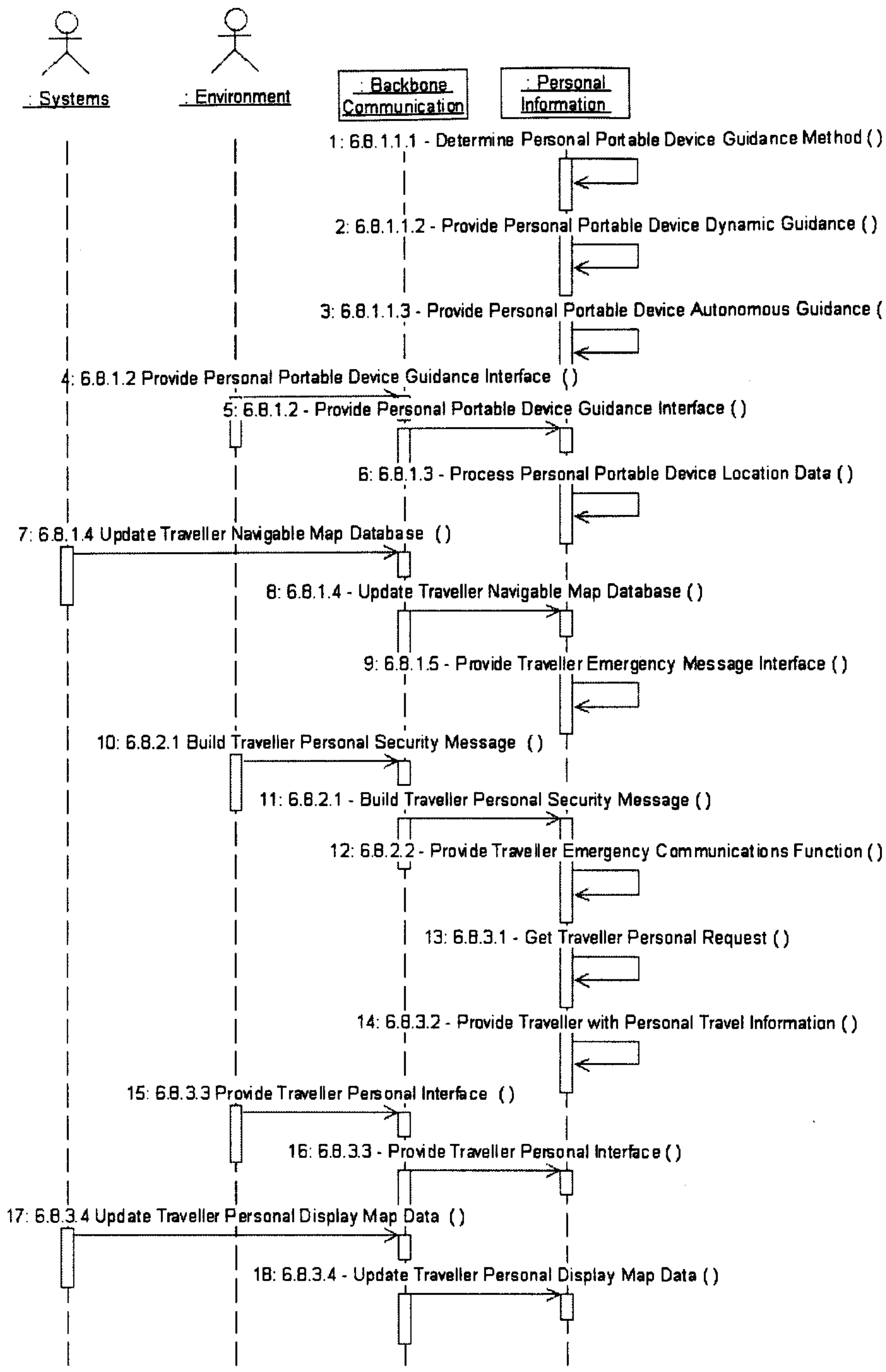

Figure 5.2 0 Object-oriented visualization of Sequence Diagram 6A.8 Provide Traveller Personal Services for CAHS in UML

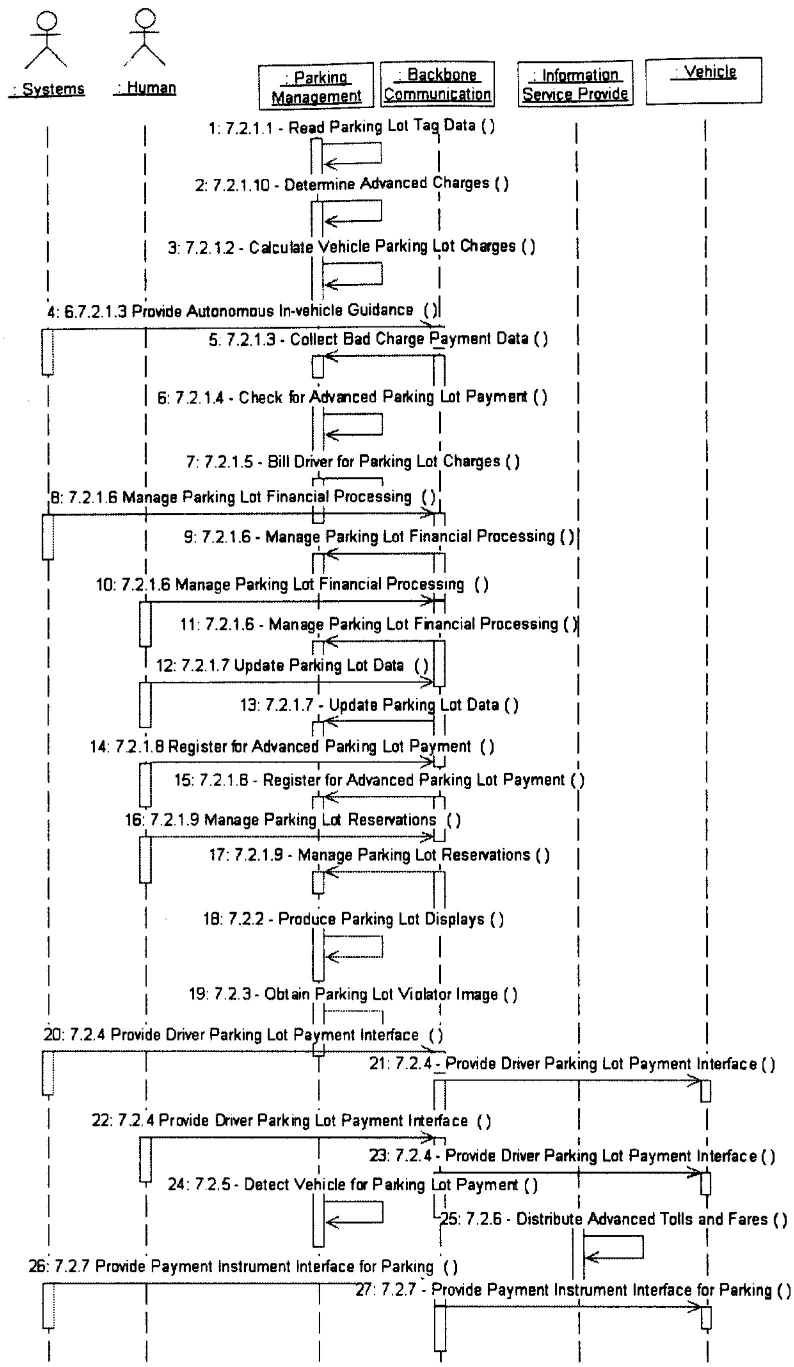

Figure 5.21 Object-oriented visualization of Sequence Diagram 7A.2 Provide Electronic Parking Payment for CAHS in UML

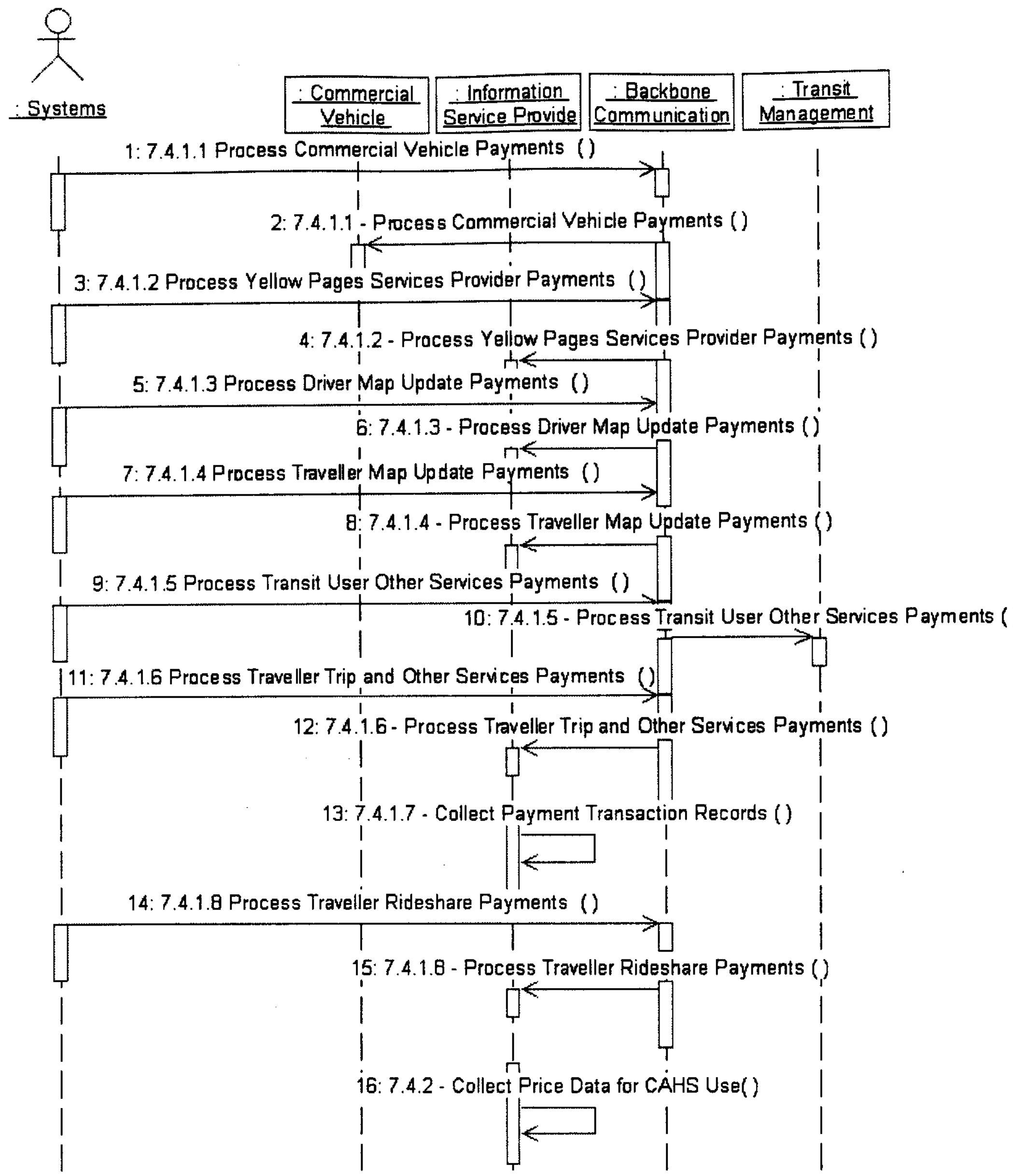

Figure 5.2 2 Object-oriented visualization of Sequence Diagram 7A.4 Carry-out Centralized Payments Processing for CAHS in UML

<<System>>
Backbone Communication

- archive management requests
- commercial vehicle archive data
- emergency archive data
- emissions archive data
- government reporting data receipt
- transaction status
- traveller archive data
- intermodal freight archive data
- construction and maintenance archive data
- map updates
- multimodal archive data
- volume weather information
- archive coordination
- other data source archive data
- parking archive data
- roadside archive data
- toll archive data
- traffic archive data
- transit archive data
- archive analysis requests
- archive requests
- citation data
- credentials and safety information request
- information on violators
- credential application**
- information request

Figure 5.2 3 (Part 1) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

information on violators
credential application**
information request
tax filing, audit data
Commercial Vehicle Administration Subsystem information exchange
credentials and safety information response
archive status
pollutant levels
pollution data
pollution state data request
pollution data parameters
activity reports
compliance review report
electronic credentials
fleet manager inquiry
driver and vehicle information
intermodal status
on-board vehicle data
vehicle location
container location request
container status request
HAZMAT information request
tax filing, audit data
container delivery confirmation
container pickup request
route plan
freight consignment request **
shipment status request

Figure 5.24 (Part 2) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

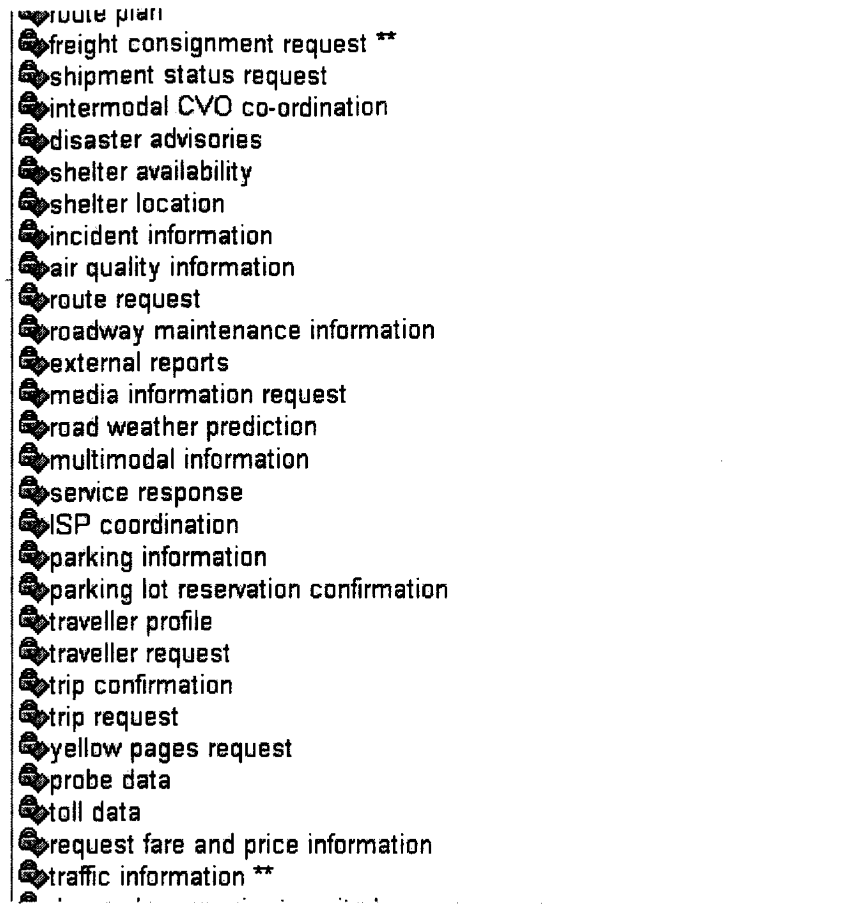

Figure 5.25 (Part 3) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

traffic information **
demand responsive transit plan
transit and fare schedules
transit incident information
transit request confirmation
transit trip plan
vehicle probe data
provider profile data
travel service info
maintenance operator inputs
maintenance status data
maintenance vehicle status data
roadway micro prediction
device status
environmental conditions
fault reports
roadway information system status
surveillance data
closure coordination
environmental sensor data
maintenance resource request
traffic equipment status
disaster response coordination
disaster response status
incident notification **
emergency dispatch response
emergency vehicle tracking data

Figure 5.26 (Part 4) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

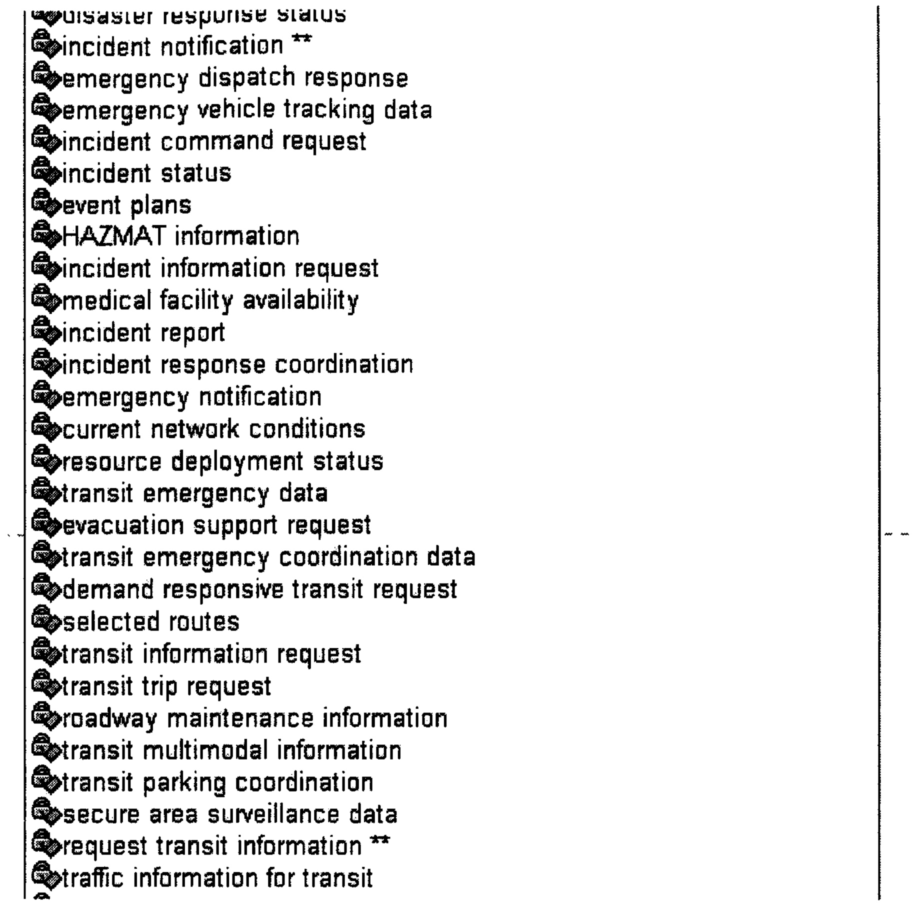

Figure 5.27 (Part 5) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

secure area surveillance data
request transit information **
traffic information for transit
transit demand management request
transit driver availability
transit fleet manager inputs
maintenance status
transit operator management data
fare and payment status
request for bad tag list
transit traveller request
transit user information
transit vehicle conditions
transit vehicle location data
transit vehicle passenger and use data
transit vehicle schedule performance
evacuation support request
transit multimodal information
Transit Management System Co-ordination
transit operator management data
transit user information
transit vehicle conditions
transit vehicle location data
road network status request
emergency traffic control request
incident response status
remote surveillance control **
resource request

Figure 5.28 (Part 6) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

incident response status
remote surveillance control **
resource request
widearea statistical pollution information
fare and price information
logged special vehicle route
request for traffic information
road network use
environmental sensor data
equipment maintenance status
maintenance dispatch status
maintenance resource response
work zone status
external reports
traffic control coordination
traffic information coordination
parking availability
parking demand management response
railroad advisories
railroad schedules
AHS status
emissions data
freeway control status
HOV data
HRI status
incident data
intersection blockage notification
request for right-of-way **
reversible lane status

Figure 5.29 (Part 7) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

intersection blockage notification
request for right-of-way **
reversible lane status
roadway warning status
signal control status
traffic flow
traffic images
violation detection
toll demand management response
transit system data
registration
toll data request
toll administration requests
Toll Transactions
toll demand management request
CVO weight and presence
credentials information
CVO database update
international border crossing data
safety information
electronic clearance data
lock tag data
on-board safety data
screening data
Commercial Vehicle Check override mode
CVO inspector input
container transfer location request
exit request **
vehicle identification

Figure 5.30 (Part 8) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

container transfer location request
exit request **
vehicle identification
container release status
container availability request
container pickup confirmation
container identification
container location
intermodal terminal operator inputs
parking lot data request
parking reservations request
parking operator inputs
request for performance data
parking demand management request
parking instructions
transit parking lot response
vehicle characteristics
tag data
sensor and surveillance control
local signal preemption request
vehicle pollution criteria
environmental hazards
pollutant levels
device control data
roadway information system data
sensor control **
multimodal crossing status

Figure 5.31 (Part 9) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

roadway information system data
sensor control **
multimodal crossing status
device control data
crossing call
non-vehicular presence
physical presence
arriving train information
track status
roadway characteristics
environmental hazards
weather conditions
traffic characteristics
AHS control information
freeway control data
HRI control data
HRI request
roadway warning data
signal control data
local signal priority request
vehicle signal violation
vehicle speed violation
vehicular presence
AHS vehicle data
toll instructions **
toll operator requests
emergency acknowledge

Figure 5.32 (Part 10) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

AHS vehicle data
toll instructions **
toll operator requests
emergency acknowledge
broadcast information
trip plan
yellow pages information
position fix
payment
personal transit information
traveller inputs
traveller information
secure area characteristics
secure area monitoring support
transit fare payment responses
transit traveller information
transit user inputs
traveller inputs
commercial vehicle measures
border clearance event record
clearance event record
electronic clearance request
lock tag data request
on-board safety request
safety inspection record
screening request **
CVO driver initialization

Figure 5.33 (Part 11) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

screening request **
CVO driver initialization
container transport assignment
fleet to driver update
intermodal status request
location request
container transfer location
entry permission
exit permission
chassis data
cargo data
container status
commercial vehicle data request
emergency dispatch requests
incident command information
suggested route
emergency personnel inputs
maintenance vehicle dispatch
maintenance vehicle measures
maintenance driver inputs
environmental hazards
transit driver inputs
bad tag list
driver instructions
fare management information
request for vehicle measures
transit schedule information **
emergency request

Figure 5.34 (Part 12) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

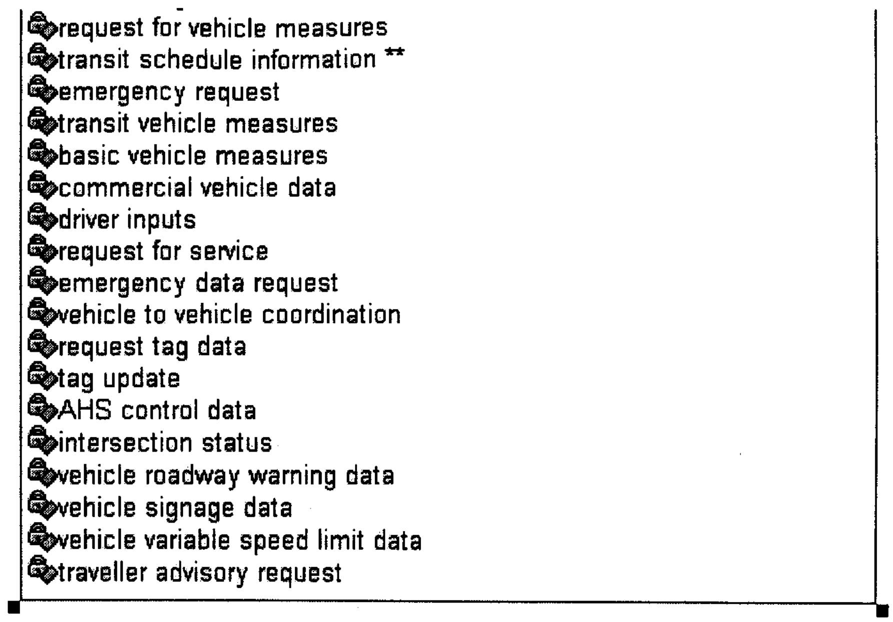

Figure 5.35 (Part 13) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

- 0 Manage CAHS ()
- 1 Manage Traffic ()
- 1.1 Provide Traffic Surveillance ()
- 1.1.1 Process Sensor Data ()
- 1.1.1.1 Process Traffic Sensor Data()
- 1.1.1.2 Collect and Process Sensor Fault Data ()
- 1.1.1.3 Process Environmental Sensor Data ()
- 1.1.1.4 Manage Data Collection and Monitoring ()
- 1.1.1.5 Process Advanced Roadway Data ()
- 1.1.1.5.1 Collect Roadway Condition Data ()
- 1.1.1.5.2 Provide Other Roadside Inputs ()
- 1.1.1.5.3 Process Roadway Warning Data ()
- 1.1.5.4 Process Road User Protection()
- 1.1.2 Process and Store Traffic Data()
- 1.1.2.1 Process Traffic Data for Storage ()
- 1.1.2.2 Process Traffic Data ()
- 1.1.2.3 Update Data Source Static Data ()
- 1.1.2.4 Monitor HOV lane use ()
- 1.1.2.5 Process Tag/AVL Data for Link Time Data ()
- 1.1.2.6 Process Collected Vehicle Smart Probe Data ()
- 1.1.2.7 Monitor Reversible Lanes ()
- 1.1.2.8 Manage Environmental Data for Traffic()
- 1.1.3 Generate Predictive Traffic Model ()
- 1.1.4 Display and Output Traffic Data ()
- 1.1.4.1 Retrieve Traffic Data ()
- 1.1.4.2 Provide Traffic Operations Personnel Traffic Data Interface ()
- 1.1.4.3 Provide Direct Media Traffic Data Interface ()
- 1.1.4.4 Update Traffic Display Map Data ()
- 1.1.4.5 Provide Media System Traffic Data Interface ()
- 1.1.4.6 Provide Traffic Data Retrieval Interface ()
- 1.1.4.7 Manage Traffic Archive Data ()
- 1.1.4.8 Provide Disaster Command Data ()
- 1.1.5 Exchange data with Other Traffic Centres ()
- 1.1.6 Collect Vehicle Tag Data for Link Time Calculations **()
- 1.1.7 Collect Vehicle Smart Probe Data ()

Figure 5.36 (Part 14) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

1.1.6 Collect Vehicle Tag Data for Link Time Calculations **()
1.1.7 Collect Vehicle Smart Probe Data ()
1.2 Provide Device Control()
1.2.1 Select Strategy ()
1.2.2 Determine Road and Freeway State ()
1.2.2.1 Determine Indicator State for Freeway Management ()
1.2.2.2 Determine Indicator State for Road Management ()
1.2.3 Determine Ramp State ()
1.2.4 Output Control Data ()
1.2.4.1 Output Control Data for Roads ()
1.2.4.2 Output Control Data for Freeways ()
1.2.4.3 Output In-vehicle Signage Data ()
1.2.5 Manage Parking Lot State ()
1.2.5.1 Determine Parking Lot State ()
1.2.5.2 Coordinate Other Parking Data ()
1.2.5.3 Provide Parking Lot Operator Interface ()
1.2.5.4 Determine P+R needs for Transit Management ()
1.2.5.5 Manage Parking Archive Data ()
1.2.5.6 Calculate Parking Lot Occupancy ()
1.2.6 Maintain Static Data for TMC ()
1.2.6.1 Maintain Traffic and Sensor Static Data ()
1.2.6.2 Provide Static Data Store Output Interface ()
1.2.7 Provide Roadside Control Facilities ()
1.2.7.1 Process Indicator Output Data for Roads ()
1.2.7.2 Monitor Roadside Equipment Operation for Faults ()
1.2.7.3 Manage Indicator Preemptions ()
1.2.7.4 Process In-vehicle Signage Data ()
1.2.7.5 Process Indicator Output Data for Freeways()
1.2.7.6 Provide Intersection Collision Avoidance Data ()
1.2.7.7 Process Vehicle Smart Probe Data for Output ()
1.2.7.8 Receive Other Roadside Inputs ()
1.2.7.9 Display Roadway Warnings ()
1.2.8 Collect and Process Indicator Fault Data ()
1.2.8.1 Collect Indicator Fault Data ()
1.2.8.2 Maintain Indicator Fault Data Store ()
1.2.8.3 Provide Indicator Fault Interface for C and M **()
1.2.8.4 Provide Traffic Operations Personnel Indicator Fault Interface ()

Figure 5.37 (Part 15) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

- 1.2.8.3 Provide Indicator Fault Interface for C and M **()
- 1.2.8.4 Provide Traffic Operations Personnel Indicator Fault Interface ()
- 1.3 Manage Incidents ()
- 1.3.1 Traffic Data Analysis for Incidents ()
- 1.3.1.1 Analyze Traffic Data for Incidents ()
- 1.3.1.2 Maintain Static Data for Incident Management ()
- 1.3.1.3 Process Traffic Images ()
- 1.3.2 Review and Manage Incident Data ()
- 1.3.2.1 Store Possible Incident Data ()
- 1.3.2.2 Review and Classify Possible Incidents ()
- 1.3.2.3 Review and Classify Planned Events ()
- 1.3.2.4 Provide Planned Events Store Interface ()
- 1.3.2.5 Provide Current Incidents Store Interface ()
- 1.3.3 Respond to Current Incidents ()
- 1.3.4 Provide Operator Interfaces for Incidents ()
- 1.3.4.1 Retrieve Incident Data ()
- 1.3.4.2 Provide Traffic Operations Personnel Incident Data Interface ()
- 1.3.4.3 Provide Media Incident Data Interface ()
- 1.3.4.4 Update Incident Display Map Data ()
- 1.3.4.5 Manage Resources for Incidents ()
- 1.3.5 Manage Possible Predetermined Responses Store ()
- 1.3.6 Manage Predetermined Incident Response Data ()
- 1.3.7 Analyze Incident Response Log ()
- 1.4 Manage Travel Demand ()
- 1.4.1 Provide Traffic Operations Personnel Demand Interface ()
- 1.4.2 Collect Demand Forecast Data ()
- 1.4.3 Update Demand Display Map Data ()
- 1.4.4 Implement Demand Management Policy ()
- 1.4.5 Calculate Forecast Demand ()
- 1.5 Manage Emissions ()
- 1.5.1 Provide Traffic Operations Personnel Pollution Data Interface ()
- 1.5.2 Process Pollution Data ()
- 1.5.3 Update Pollution Display Map Data ()
- 1.5.4 Manage Pollution State Data Store ()
- 1.5.5 Process Vehicle Pollution Data ()
- 1.5.6 Detect Roadside Pollution Levels ()
- 1.5.7 Manage Pollution Data Log**()
- 1.5.8 Manage Pollution Reference Data Store ()

Figure 5.38 (Part 16) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

1.5.7 Manage Pollution Data Log**()
1.5.8 Manage Pollution Reference Data Store ()
1.5.9 Manage Pollution Archive Data ()
1.6 Manage Highway Rail Intersections ()
1.6.1 Manage HRI Vehicle Traffic ()
1.6.1.1 Detect Roadway Events ()
1.6.1.2 Activate HRI Device Controls ()
1.6.1.2.1 Control HRI Traffic Signals ()
1.6.1.2.2 Control HRI Warnings and Barriers ()
1.6.1.2.3 Provide SSR Device Controls ()
1.6.1.2.4 Provide HSR Device Controls ()
1.6.1.2.5 Manage Device Control ()
1.6.1.2.6 Maintain Device State (()
1.6.1.3 Perform Equipment Self-Test ()
1.6.1.4 Provide Advisories and Alerts ()
1.6.1.4.1 Generate Alerts and Advisories ()
1.6.1.4.2 Provide Closure Parameters ()
1.6.1.4.3 Report Alerts and Advisories ()
1.6.1.4.4 Report HRI Status on Approach ()
1.6.1.5 Detect HRI Hazards ()
1.6.1.6 Provide Advance Warnings ()
1.6.1.6.1 Close HRI on Detection ()
1.6.1.6.2 Detect Imminent Vehicle/Train Collision ()
1.6.1.7 Execute Local Control Strategy ()
1.6.1.7.1 Control Vehicle Traffic at Passive HRI ()
1.6.1.7.2 Control Vehicle Traffic at Active HRI ()
1.6.1.7.3 Close HRI on Command ()
1.6.2 Interact with Rail Operations ()
1.6.2.1 Exchange Data with Rail Operations ()
1.6.2.2 Manage Alerts and Advisories ()
1.6.2.3 Manage Rail Traffic Control Data ()
1.6.3 Manage HRI Rail Traffic ()
1.6.3.1 Interact with Wayside Systems ()
1.6.3.2 Advise and Protect Train Crews ()
1.6.3.3 Provide ATS Alerts ()
1.6.4 Interact with Vehicle Traffic Management**()
1.6.4.1 Manage HRI Closures ()
1.6.4.2 Exchange Data with Traffic Management ()

Figure 5.39 (Part 17) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

- 1.6.4 Interact with Vehicle Traffic Management**()
- 1.6.4.1 Manage HRI Closures ()
- 1.6.4.2 Exchange Data with Traffic Management ()
- 1.6.5 Monitor HRI Status ()
- 1.6.5.1 Provide Interactive Interface ()
- 1.6.5.2 Determine HRI Status ()
- 1.6.5.3 Maintain HRI Closure Data ()
- 1.7 Manage Operations and Maintenance ()
- 1.7.1 Manage Operations and Maintenance Data ()
- 1.7.1.1 Gather O&M Data ()
- 1.7.1.2 Manage O&M Archive Data ()
- 1.7.1.3 Manage O&M Assets ()
- 1.7.1.4 Manage Environmental Data for Operations and Maintenance ()
- 1.7.1.4.1 Gather and Process Environmental Data ()
- 1.7.1.4.2 Create Road Forecasts ()
- 1.7.1.4.3 Fuse Environmental Data ()
- 1.7.2 Manage Maintenance Activity ()
- 1.7.2.1 Schedule Maintenance Activity ()
- 1.7.2.2 Manage and Control On-board Maintenance Systems ()
- 1.7.2.3 Manage Maintenance Operator Interface ()
- 1.7.2.4 Manage Maintenance Driver Interface ()
- 1.7.2.5 Collect Environmental Data On-board ()
- 1.7.3 Control Roadway Maintenance Devices ()
- 1.7.4 Disseminate O&M Information ()
- 1.8 Manage Traffic Enforcement ()
- 1.8.1 Detect and Classify Speed Violations ()
- 1.8.2 Collect and Verify Speed Violations ()
- 1.8.3 Post Speed Limit ()
- 1.8.4 Set Speed Limit ()
- 1.8.5 Establish Violation Parameters ()
- 1.8.6 Detect and Classify Signal Violations ()
- 1.8.7 Collect and Verify Signal Violations ()
- 2 Manage Commercial Vehicles ()
- 2.1 Manage Commercial Vehicle Fleet Operations ()
- 2.1.1 Manage Commercial Fleet Electronic Credentials and Tax Filing ()
- 2.1.2 Provide Commercial Fleet Static Route ()
- 2.1.3 Provide Flt Mgr Electronic Credentials and Tax Filing Interface**()

Figure 5.40 (Part 18) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

2.1.3 Provide Flt Mgr Electronic Credentials and Tax Filing Interface**()
2.1.4 Provide Fleet Manager Commercial Vehicle Communications ()
2.1.5 Provide Commercial Vehicle Driver Routing Interface ()
2.1.6 Manage Driver Instruction Store ()
2.2 Manage Commercial Vehicle Driver Operations ()
2.2.1 Manage CV Electronic Credential and Tax Filing Interface ()
2.2.2 Provide Vehicle Static Route ()
2.2.3 Provide CV Driver Electronic Credential and Tax Filing Interface ()
2.2.4 Provide Commercial Vehicle Driver Communications ()
2.3 Provide Commercial Vehicle Roadside Facilities ()
2.3.1 Produce Commercial Vehicle Driver Message at Roadside ()
2.3.2 Provide Commercial Vehicle Clearance Screening ()
2.3.2.1 Administer Commercial Vehicle Roadside Credentials Database ()
2.3.2.2 Process Screening Transactions ()
2.3.3 Provide Roadside Commercial Vehicle Safety ()
2.3.3.1 Provide Commercial Vehicle Checkstation Communications ()
2.3.3.2 Provide Commercial Vehicle Inspector Handheld Terminal Interface ()
2.3.3.3 Administer Commercial Vehicle Roadside Safety Datadase ()
2.3.3.4 Carry-out Commercial Vehicle Roadside Safety Screening ()
2.3.3.5 Carry-out Commercial Vehicle Roadside Inspection ()
2.3.4 Detect Commercial Vehicle ()
2.3.5 Provide Commercial Vehicle RoadsideOperator Interface ()
2.3.6 Provide Commercial Vehicle Reports ()
2.3.7 Produce Commercial Vehicle Driver Message on Vehicle ()
2.3.8 Provide Commercial Vehicle Border Screening ()
2.4 Provide Commercial Vehicle Data Collection ()
2.4.1 Communicate Commercial Vehicle On-board Data to Roadside ()
2.4.2 Collect On-board Commercial Vehicle Sensor Data ()
2.4.3 Analyze Commercial Vehicle On-board Data ()
2.4.4 Provide Commercial Vehicle Driver Interface ()
2.4.5 Communicate Commercial Vehicle On-board Data to Vehicle Manager ()
2.4.6 Provide Commercial Vehicle On-board Data Store Interface ()
2.5 Administer Commercial Vehicles ()
2.5.1 Manage Commercial Vehicle Trips and Clearances ()
2.5.2 Obtain Electronic Credential and Tax Filing Payment ()
2.5.3 Update Permits and Duties Store **()
2.5.4 Communicate with Other Commercial Vehicle Administration System ()

Figure 5.41 (Part 19) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

2.5.2 Obtain Electronic Credential and Tax Filing Payment ()
2.5.3 Update Permits and Duties Store **()
2.5.4 Communicate with Other Commercial Vehicle Administration System ()
2.5.5 Manage Commercial Vehicle Credentials and Enrollment ()
2.5.6 Output Commercial Vehicle Enrollment Data to Roadside Facilities ()
2.5.7 Process Commercial Vehicle Violations ()
2.5.8 Process Data Received from Roadside Facilities ()
2.5.9 Manage Commercial Vehicle Archive Data ()
2.6 Provide Commercial Vehicle On-board Data ()
2.6.1 Provide Commercial Vehicle Manager Tag Data Interface ()
2.6.2 Transmit Commercial Vehicle Tag Data ()
2.6.3 Provide Commercial Driver Tag Data Interface ()
2.6.4 Provide Lock Tag Data Interface ()
2.6.5 Manage Commercial Vehicle Tag Data Store ()
2.7 Manage Cargo ()
2.7.1 Manage Intermodal Terminal Interface ()
2.7.1.1 Manage Terminal Access ()
2.7.1.2 Manage Container Pickup and Delivery ()
2.7.1.3 Manage Container Release ()
2.7.1.4 Provide Intermodal Terminal Operator I/F ()
2.7.1.5 Provide Container Tracking ()
2.7.2 Manage Intermodal Container ()
2.7.2.1 Determine Container Status ()
2.7.2.2 Determine Cargo Status ()
2.7.2.3 Provide Cargo Status Interface ()
2.7.2.4 Provide Container Status Interface ()
2.7.2.5 Manage Customs Interface ()
2.7.3 Manage On-Board Intermodal Applications ()
2.7.3.1 Provide Driver Interface for Intermodal Freight Dispatch()
2.7.3.2 Monitor Container Status ()
2.7.3.3 Monitor Chassis Status ()
2.7.3.4 Manage On-Board Facility Access ()
2.7.4 Manage Intermodal Dispatch ()
2.7.4.1 Monitor Intermodal Elements ()
2.7.4.2 Provide Fleet Manager Interface for Intermodal ()
2.7.4.3 Manage Intermodal Customer Interface**()
2.7.4.4 Manage Other Intermodal FMS Interface ()
2.7.4.5 Manage Distribution and Logistics Management Provider Interface ()

Figure 5.42 (Part 20) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

2.7.4.2 Provide Fleet Manager Interface for Intermodal ()
2.7.4.3 Manage Intermodal Customer Interface**()
2.7.4.4 Manage Other Intermodal FMS Interface ()
2.7.4.5 Manage Distribution and Logistics Management Provider Interface ()
2.7.4.6 Manage Freight Consolidation Station Interface ()
3 Provide Vehicle Monitoring and Control ()
3.1 Monitor Vehicle Status ()
3.1.1 Produce Collision and Crash Avoidance Data ()
3.1.2 Carry-out Safety Analysis ()
3.1.3 Process Vehicle On-board Data ()
3.2 Provide Automatic Vehicle Operation ()
3.2.1 Provide Driver Interface ()
3.2.2 Provide AHS Control ()
3.2.3 Provide Vehicle Control ()
3.2.3.1 Provide Command Interface ()
3.2.3.2 Manage Platoon Following ()
3.2.3.3 Process data for Vehicle Actuators ()
3.2.3.4 Provide Servo Controls ()
3.2.3.4.1 Provide Speed Servo Control ()
3.2.3.4.2 Provide Headway Servo Control ()
3.2.3.4.3 Provide Lane Servo Control ()
3.2.3.4.4 Provide Change Lane Servo Control ()
3.2.3.4.5 Provide Vehicle Control Data Interface ()
3.2.3.5 Process Vehicle Sensor Data ()
3.2.3.6 Communicate with other Platoon Vehicles ()
3.2.4 Process Sensor Data for AHS input ()
3.2.5 Check Vehicle for AHS eligibility ()
3.2.6 Manage AHS Check-in and Check-out ()
3.2.7 Manage AHS Operations ()
3.3 Provide Automatic Emergency Notification ()
3.3.1 Provide Cargo Data for Incident Notification ()
3.3.2 Provide Communications Function ()
3.3.3 Build Automatic Collision Notification Message ()
3.4 Enhance Driver's Vision ()
4 Manage Transit ()
4.1 Operate Vehicles and Facilities ()
4.1.1 Process Transit Vehicle Sensor Trip Data**()
4.1.2 Determine Transit Vehicle Deviation and Corrections ()

Figure 5.43 (Part 21) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

4.1.1 Process Transit Vehicle Sensor Trip Data**()
4.1.2 Determine Transit Vehicle Deviation and Corrections ()
4.1.2.1 Determine Transit Vehicle Deviation and ETA ()
4.1.2.2 Determine Transit Vehicle Corrective Instructions ()
4.1.2.3 Provide Transit Vehicle Driver Interface ()
4.1.2.4 Provide Transit Vehicle Correction Data Output Interface ()
4.1.2.5 Request Transit Vehicle Preemptions ()
4.1.3 Provide Transit Vehicle Location Data ()
4.1.4 Manage Transit Vehicle Deviations ()
4.1.5 Provide Transit Vehicle Status Information ()
4.1.6 Manage Transit Vehicle Operations Data ()
4.1.7 Manage Connection Protection ()
4.1.7.1 Manage Connections with External Systems ()
4.1.7.2 Provide Transit Vehicle Deviation Data Output Interface ()
4.1.7.3 Manage Individual Service Requests ()
4.1.8 Provide Transit Operations Data Distribution Interface ()
4.1.9 Process Transit Vehicle Sensor Maintenance Data ()
4.2 Plan and Schedule Transit Services ()
4.2.1 Provide Demand Responsive Transit Service ()
4.2.1.1 Process Demand Responsive Transit Trip Request ()
4.2.1.2 Compute Demand Responsive Transit Vehicle Availability ()
4.2.1.3 Generate Demand Responsive Transit Schedule and Routes ()
4.2.1.4 Confirm Demand Responsive Transit Schedule and Route ()
4.2.1.5 Process Demand Responsive Transit Vehicle Availability Data ()
4.2.1.6 Provide Demand Responsive Transit Driver Interface ()
4.2.2 Provide Transit Plans Store Interface ()
4.2.3 Generate Transit Routes and Schedules ()
4.2.3.1 Generate Transit Routes ()
4.2.3.2 Generate Schedules ()
4.2.3.3 Produce Transit Service Data for External Use ()
4.2.3.4 Provide Transit Fleet Manager Interface for Services Generation ()
4.2.3.5 Manage Transit Operational Data Store ()
4.2.3.6 Produce Transit Service Data for Manage Transit Use ()
4.2.3.7 Provide Interface for Other TRM Data ()
4.2.3.8 Provide Interface for Transit Service Raw Data ()
4.2.3.9 Update Transit Map Data ()
4.2.4 Manage Transit Archive Data **()
4.3 Schedule Transit Vehicle Maintenance ()

Figure 5.44 (Part 22) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

4.2.3.9 Update Transit Map Data ()
4.2.4 Manage Transit Archive Data **()
4.3 Schedule Transit Vehicle Maintenance ()
4.3.1 Monitor Transit Vehicle Condition ()
4.3.2 Generate Transit Vehicle Maintenance Schedules ()
4.3.3 Generate Technician Work Assignments ()
4.3.4 Monitor And Verify Maintenance Activity ()
4.3.5 Report Transit Vehicle Information (()
4.3.6 Update Transit Vehicle Information ()
4.3.7 Manage Transit Vehicle Operations Data Store ()
4.4 Support Security and Coordination ()
4.4.1 Provide Transit Security and Emergency Management ()
4.4.1.1 Manage Transit Security ()
4.4.1.2 Manage Transit Emergencies ()
4.4.1.3 Provide Transit System Operator Security Interface ()
4.4.1.4 Provide Transit External Interface for Emergencies ()
4.4.1.5 Provide Transit Driver Interface for Emergencies ()
4.4.1.6 Collect Transit Vehicle Emergency Information ()
4.4.1.7 Monitor Secure Area ()
4.4.1.8 Report Traveller Emergencies ()
4.4.2 Coordinate Multiple Agency Responses to Incidents ()
4.4.3 Generate Responses for Incidents ()
4.4.4 Coordinate Transit Disaster Response ()
4.5 Generate Transit Driver Schedules ()
4.5.1 Assess Transit Driver Performance ()
4.5.2 Assess Transit Driver Availability ()
4.5.3 Access Transit Driver Cost Effectiveness ()
4.5.4 Assess Transit Driver Eligibility ()
4.5.5 Generate Transit Driver Route Assignments ()
4.5.6 Update Transit Driver Information ()
4.5.7 Report Transit Driver Information ()
4.5.8 Provide Transit Driver Information Store Interface ()
4.6 Collect Transit Fares in the Vehicle ()
4.6.1 Detect Transit User on Vehicle ()
4.6.2 Determine Transit User Needs on Vehicle ()
4.6.3 Determine Transit Fare on Vehicle ()
4.6.4 Manage Transit Fare Billing on Vehicle **()
4.6.5 Provide Transit User Fare Payment Interface on Vehicle ()

Figure 5.45 (Part 23) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

4.6.4 Manage Transit Fare Billing on Vehicle **()
4.6.5 Provide Transit User Fare Payment Interface on Vehicle ()
4.6.6 Update Transit Vehicle Fare Data ()
4.6.7 Provide Transit Vehicle Passenger Data ()
4.6.8 Manage Transit Vehicle Advanced Payments ()
4.7 Provide Transit User Roadside Facilities ()
4.7.1 Provide Transit User Roadside Information ()
4.7.1.1 Provide Transit User Roadside Data Interface ()
4.7.1.2 Provide Transit User Roadside Vehicle Data Interface ()
4.7.2 Collect Transit Fares at the Roadside ()
4.7.2.1 Detect Transit User at Roadside ()
4.7.2.2 Determine Transit User Needs at Roadside ()
4.7.2.3 Determine Transit Fare at Roadside ()
4.7.2.4 Manage Transit Fare Billing at Roadside ()
4.7.2.5 Provide Transit User Roadside Fare Interface ()
4.7.2.6 Update Roadside Transit Fare Data ()
4.7.2.7 Provide Transit Roadside Passenger Data ()
5 Manage Emergency Services ()
5.1 Provide Emergency Service Allocation ()
5.1.1 Identify Emergencies from Inputs ()
5.1.2 Determine Coordinated Response Plan ()
5.1.3 Communicate Emergency Status ()
5.1.4 Manage Emergency Response ()
5.1.5 Manage Emergency Service Allocation Store ()
5.1.6 Process Mayday Messages ()
5.2 Provide Operator Interface for Emergency Data ()
5.3 Manage Emergency Vehicles ()
5.3.1 Select Response Mode ()
5.3.2 Dispatch Vehicle ()
5.3.3 Track Vehicle ()
5.3.4 Assess Response Status ()
5.3.5 Provide Emergency Personnel Interface ()
5.3.6 Maintain Vehicle Status ()
5.3.7 Provide Emergency Vehicle Route ()
5.4 Provide Law Enforcement Allocation ()
5.4.1 Process TM Detected Violations ()
5.4.2 Process Violations for Tolls ()
5.4.3 Process Parking Lot Violations**()

Figure 5.46 (Part 24) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

5.4.2 Process Violations for Tolls ()
5.4.3 Process Parking Lot Violations**()
5.4.4 Process Fare Payment Violations ()
5.4.5 Process Vehicle Fare Collection Violations ()
5.4.6 Process CV Violations ()
5.4.7 Process Roadside Fare Collection Violations ()
5.5 Update Emergency Display Map Data ()
5.6 Manage Emergency Services Data ()
5.7 Co-ordinate Disaster Response ()
5.7.1 Collect Disaster Response Data ()
5.7.2 Provide Medical Facility Interface ()
6 Provide Driver and Traveller Services ()
6.1 Provide Trip Planning Services ()
6.1.1 Provide Trip Planning Information to Traveller ()
6.1.2 Confirm Traveller's Trip Plan ()
6.1.3 Manage Multimodal Service Provider Interface ()
6.1.4 Provide ISP Operator Interface for Trip Planning Parameters ()
6.1.5 Collect Service Requests and Confirmation for Archive ()
6.1.6 Manage Traveller Info Archive Data ()
6.2 Provide Information Services ()
6.2.1 Provide Advisory and Broadcast Data ()
6.2.1.1 Collect Traffic Data for Advisory Messages ()
6.2.1.2 Provide Traffic and Transit Advisory Messages ()
6.2.1.3 Collect Transit Data for Advisory Messages ()
6.2.1.4 Provide Traffic and Transit Broadcast Messages ()
6.2.1.5 Provide ISP Operator Broadcast Parameters Interface ()
6.2.1.6 Provide Transit Advisory Data On Vehicle ()
6.2.2 Prepare and Output In-vehicle Displays ()
6.2.3 Provide Transit User Advisory Interface ()
6.2.4 Collect Yellow Pages Data ()
6.2.5 Provide Driver Interface ()
6.2.6 Provide Yellow Pages Data and Reservations ()
6.3 Provide Traveller Services at Kiosks ()
6.3.1 Get Traveller Request ()
6.3.2 Inform Traveller ()
6.3.3 Provide Traveller Kiosk Interface ()
6.3.4 Update Traveller Display Map Data at Kiosk ()
6.4 Manage Ridesharing**()

Figure 5.47 (Part 25) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

6.4 Manage Ridesharing**()
6.4.1 Screen Rider Requests ()
6.4.2 Match Rider and Provider ()
6.4.3 Report Ride Match Results to Requestor ()
6.4.4 Confirm Traveller Rideshare Request ()
6.5 Manage Yellow Pages Services ()
6.5.1 Collect and Update Traveller Information ()
6.5.2 Provide Traveller Yellow Pages Information and Reservations ()
6.5.3 Register Yellow Pages Service Providers ()
6.6 Provide Guidance and Trip Planning Services ()
6.6.1 Provide Multimodal Route Selection ()
6.6.2 Select Vehicle Route ()
6.6.2.1 Calculate Vehicle Route ()
6.6.2.2 Provide Vehicle Route Calculation Data ()
6.6.2.3 Provide Route Segment Data for Other Areas ()
6.6.2.4 Update Vehicle Route Selection Map Data ()
6.6.2.5 Provide ISP Operator Route Parameters Interface ()
6.6.2.6 Calculate Vehicle Probe Data for Guidance ()
6.6.3 Update Other Routes Selection Map Data ()
6.6.4 Select Transit Route ()
6.6.5 Select Other Routes ()
6.7 Provide Driver Personal Services ()
6.7.1 Provide Driver Personal Security ()
6.7.1.1 Build Driver Personal Security Message ()
6.7.1.2 Provide Driver In-vehicle Communications Function ()
6.7.2 Provide On-line Vehicle Guidance ()
6.7.2.1 Provide Vehicle Guidance ()
6.7.2.1.1 Determine In-vehicle Guidance Method ()
6.7.2.1.2 Provide Dynamic In-vehicle Guidance ()
6.7.2.1.3 Provide Autonomous In-vehicle Guidance ()
6.7.2.2 Process Vehicle Location Data ()
6.7.2.3 Provide Driver Guidance Interface ()
6.7.2.4 Update Vehicle Navigable Map Database ()
6.8 Provide Traveller Personal Services ()
6.8.1 Provide On-line Traveller Guidance ()
6.8.1.1 Provide Traveller Guidance ()
6.8.1.1.1 Determine Personal Portable Device Guidance Method **()

Figure 5.48 (Part 25) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

6.8.1.1.1 Determine Personal Portable Device Guidance Method **()
6.8.1.1.2 Provide Personal Portable Device Dynamic Guidance ()
6.8.1.1.3 Provide Personal Portable Device Autonomous Guidance ()
6.8.1.2 Provide Personal Portable Device Guidance Interface ()
6.8.1.3 Process Personal Portable Device Location Data ()
6.8.1.4 Update Traveller Navigable Map Database ()
6.8.1.5 Provide Traveller Emergency Message Interface ()
6.8.2 Provide Traveller Personal Security ()
6.8.2.1 Build Traveller Personal Security Message ()
6.8.2.2 Provide Traveller Emergency Communications Function ()
6.8.3 Provide Traveller Services at Personal Devices ()
6.8.3.1 Get Traveller Personal Request ()
6.8.3.2 Provide Traveller with Personal Travel Information ()
6.8.3.3 Provide Traveller Personal Interface ()
6.8.3.4 Update Traveller Personal Display Map Data ()
7 Provide Electronic Payment Services ()
7.1 Provide Electronic Toll Payment ()
7.1.1 Process Electronic Toll Payment ()
7.1.1.1 Read Tag Data for Tolls ()
7.1.1.10 Determine Advanced Toll Bill ()
7.1.1.11 Manage Toll Archive Data ()
7.1.1.2 Calculate Vehicle Toll ()
7.1.1.3 Manage Bad Toll Payment Data ()
7.1.1.4 Check for Advanced Tolls Payment ()
7.1.1.5 Bill Driver for Tolls ()
7.1.1.6 Collect Probe Data From Toll Transactions ()
7.1.1.7 Update Toll Price Data ()
7.1.1.8 Register for Advanced Toll Payment ()
7.1.1.9 Manage Toll Financial Processing ()
7.1.2 Produce Roadside Displays ()
7.1.3 Obtain Toll Violator Image ()
7.1.4 Provide Driver Toll Payment Interface ()
7.1.5 Detect Vehicle for Tolls ()
7.1.6 Distribute Advanced Charges and Fares ()
7.1.7 Provide Payment Instrument Interface for Tolls ()
7.2 Provide Electronic Parking Payment **()
7.2.1 Process Electronic Parking Lot Payment ()

Figure 5.49 (Part 26) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

7.2 Provide Electronic Parking Payment **()
7.2.1 Process Electronic Parking Lot Payment ()
7.2.1.1 Read Parking Lot Tag Data ()
7.2.1.10 Determine Advanced Charges ()
7.2.1.2 Calculate Vehicle Parking Lot Charges ()
7.2.1.3 Collect Bad Charge Payment Data ()
7.2.1.4 Check for Advanced Parking Lot Payment ()
7.2.1.5 Bill Driver for Parking Lot Charges ()
7.2.1.6 Manage Parking Lot Financial Processing ()
7.2.1.7 Update Parking Lot Data ()
7.2.1.8 Register for Advanced Parking Lot Payment ()
7.2.1.9 Manage Parking Lot Reservations ()
7.2.2 Produce Parking Lot Displays ()
7.2.3 Obtain Parking Lot Violator Image ()
7.2.4 Provide Driver Parking Lot Payment Interface ()
7.2.5 Detect Vehicle for Parking Lot Payment ()
7.2.6 Distribute Advanced Tolls and Fares ()
7.2.7 Provide Payment Instrument Interface for Parking ()
7.3 Provide Electronic Fare Collection ()
7.3.1 Process Electronic Transit Fare Payment ()
7.3.1.1 Register for Advanced Transit Fare Payment ()
7.3.1.2 Determine Advanced Transit Fares ()
7.3.1.3 Manage Transit Fare Financial Processing ()
7.3.1.4 Check for Advanced Transit Fare Payment ()
7.3.1.5 Bill Transit User for Transit Fare ()
7.3.1.6 Collect Bad Transit Fare Payment Data ()
7.3.1.7 Update Transit Fare Data ()
7.3.2 Distribute Advanced Tolls and Parking Lot Charges ()
7.3.3 Get Transit User Image for Violation ()
7.3.4 Provide Remote Terminal Payment Instrument Interface ()
7.3.5 Provide Transit Vehicle Payment Instrument Interface ()
7.4 Carry-out Centralized Payments Processing ()
7.4.1 Collect Advanced Payments ()
7.4.1.1 Process Commercial Vehicle Payments ()
7.4.1.2 Process Yellow Pages Services Provider Payments ()
7.4.1.3 Process Driver Map Update Payments ()
7.4.1.4 Process Traveller Map Update Payments **()
7.4.1.5 Process Transit User Other Services Payments ()

Figure 5.50 (Part 27) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in

7.4.1.2 Process Yellow Pages Services Provider Payments ()
7.4.1.3 Process Driver Map Update Payments ()
7.4.1.4 Process Traveller Map Update Payments **()
7.4.1.5 Process Transit User Other Services Payments ()
7.4.1.6 Process Traveller Trip and Other Services Payments ()
7.4.1.7 Collect Payment Transaction Records ()
7.4.1.8 Process Traveller Rideshare Payments ()
7.4.2 Collect Price Data for CAHS Use ()
7.4.3 Route Traveller Advanced Payments ()
7.5 Provide Payment Instrument Interfaces ()
7.5.1 Provide Vehicle Payment Instrument Interface ()
7.5.2 Provide Transit User Roadside Payment Instrument Interface ()
7.5.3 Provide Personal Payment Instrument Interface ()
7.5.4 Provide Commercial Fleet Payment Instrument Interface ()
7.5.5 Provide Traveller Kiosk Payment Instrument Interface ()
8 Manage Archived Data ()
8.2 Manage Archive ()
8.3 Manage Archive Data Administrator Interface ()
8.4 Coordinate Archives ()
8.5 Process Archived Data User System Requests ()
8.6 Analyze Archive ()
8.7 Process On Demand Archive Requests ()
8.8 Prepare Government Reporting Inputs ()
8.9 Manage Roadside Data Collection ()
9 Satisfy Implementation Requirements ()
8.1 Get Archive Data ()
5.7.3 Provide Disaster Coordination Authority Interface ()

Figure 5.51 (Part 28) Object-oriented visualization of CAHS's Class Backbone Communication with it attributes and Functions in UML

5.3.6 Findings of the Visual Analysis

The monitoring of a class 6 (Backbone communication class) that is a generalization of the four-communication channels, which are available in ITSC architecture. It is active in the fulfillment of the sequence diagrams functionality. The communication where coming in and out of class no. 6 and without it no functionality can be assured. This is true for almost all the sequence diagrams of CAHS. **What will happen if a malfunction occurs in the backbone communication class?**

5.4 Conclusion of CAHS Three Analysis Study

We proved in this chapter that we need a **backup communication system** within CAHS. Since CAHS posses all ITSC requirements, this fact can be generalized to ITSC architecture.

Chapter 6

A SATELLITIC AUTOMATED HIGHWAU SYSTEM (SAHS): ANALYSIS OF ITSC REQUIREMENTS RELATED TO COMMUNICATION PROBLEM SOULTION

The main objectives of this chapter are to prove that our solution for a backup communication is adequate and object-oriented architecture provides stability and flexibility. Stability is the ability of the architecture to meet the present and future needs of user, designer, and the environment. Flexibility is the ability of the architecture to respond to the demands of the user, designer, and environment changes. In order to reach this goal, we set a conceptual system called the Satellitic Automated Highway System (SAHS). This system is integration between an Automated Highway concept and a Satellitic Backup Communication System. It should be clear that we are not developing a new system due to limited time and resources, but rather scratching the surface of architecture to the point whereby the solution to the problem of the lack of a backup communication would manifest itself. SAHS is a partial object-oriented development that will allow us through localization of change offer to have a stable and flexible architecture. Since SAHS possess all ITSC architecture requirements plus five (Backup Communication) architecture requirements, then the analysis result can be directly generalized to the ITSC. In the previous chapters and throughout the example modules we developed a realization of the serious problems that are within the documentation and visualization of the Canadian Intelligent Transportation Architecture (ITSC). These serious problems are shared with most of the ITS architectures around the world, and eventually, they will face extreme pressures to change their architecture due to the unbalanced proportion that they hold between flexibility and stability. The pressure to change will come from the challenges of change that comes with the change of user and developer needs and the change in technology. The impact of change would be minimized if ITS Canadian developers would choose an object-oriented development methodology and provide a Satellitic Backup Communication System.

6.1 SOLUTION VISION

The solution that will provide ITS Canada the ability to promote a stable and flexible architecture is by adapting an object-oriented methodology in developing the documentation and visualization of our architecture. Any change will affect small number of classes that can be easily changed without noticeable effects on the rest of the architecture. The other solution would be to provide a seamless integration of the automated highway system with a redundant Satellitic wireless backup communication system. The conceptual project model for such system is given in Figure 6.1.

One of the main components of the Redundant Satellitic wireless backup communication system is the stratospheric satellite. An example of such a satellite is shown in Figure 6.2. This unmanned solar-driven satellite can be used as a wireless communication platform in the Satellitic Backup Communication System. More detail regarding these satellites can be obtained from the NASA reference (NASA Dryden Flight Research Center, 2002). The other important part of the Satellitic Backup Communication System is the Intelligent Digital Modem (IDM). The intelligent modem is located inside every system and subsystem of the Satellitic Automated Highway system. The main duty for the IDM will be to monitor the Satellitic Automated Highway system Communication channels. In the event of malfunctioning of the SAHS communication, the IDM will divert the communication to the stratospheric satellite communication platform. In the case of failure of the stratospheric satellite communication platform, the IDM will initiate a message to the actor to take over the related operation. We will develop the IDM to a near market level of readiness.

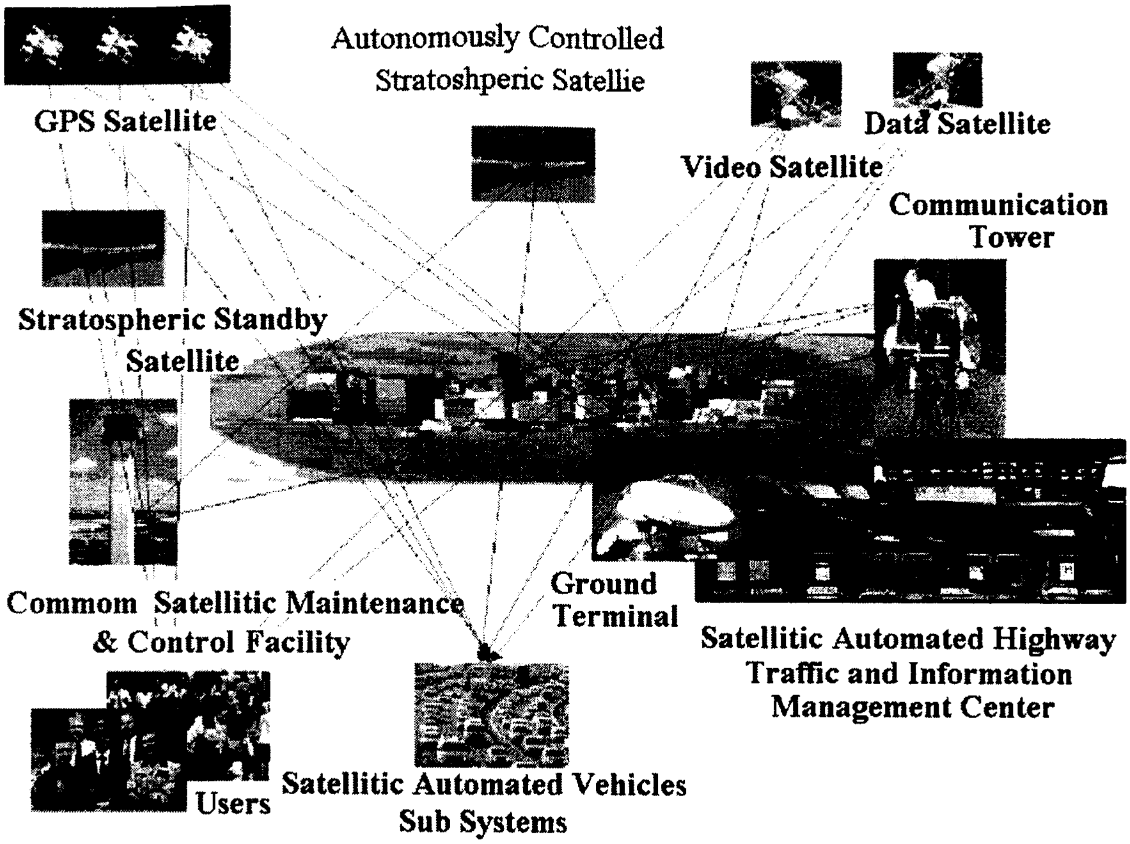

Figure 6.1 Satellitic Automated Highway System
(SAHS) Initial Conceptual Model.

Figure 6.2 An example of a stratospheric Satellite (NASA Dryden Flight Research Center, 2002)

6.2 The Analysis of SAHS Conceptual Model

Three-solution analysis:

- Scientific Solution Analysis
- Documental Solution Analysis
- Visual Solution Analysis

6.3 Scientific Solution Analysis

The scientific study will contain both the functional requirements and the user requirements of the ITSC plus five (Backup Communication) architecture requirements, then the analysis of result can be directly generalized to the ITSC. In order to do the study we created both use case and class system. The study is about tracing each requirement to the class or classes that are responsible for the functionality of the requirement. SAHS the proposed partial architecture development efforts for a system that will demonstrate clearly that our solution for backup communication is adequate and object-oriented architecture provides Stability & Flexibility. The following were performed:

1. The description of functional requirement of the ITSC architecture was derived and called Requirement Description (RDescription).

2. The functional requirement number of the ITSC architecture was obtained and called the Functional Requirement Number (FunctionalR#).

3. The user requirement number of the ITSC architecture was obtained and called the User Requirement Number (UserR#).

4. The functional requirement number of the ITSC architecture was obtained and assigned to the Use case number and called Use Case number (UseCA #) for normal operations (where Backbone Communication is functioning) and (UseCB #) is for when a failure occurs in the Backbone Communication System. Then, the Satellitic Backup Communication takes over the operation.

5. A creation of class number was done in order to be able to trace the requirements to the responsible class and was called the Class Responsibility number (ClassR#)

6. An automated number was created to keep truck of the study, and we called it the Satellitic Automated Highway System number (SAHS #).

The results were documented in SAHS requirements class responsibility shell. The shell is a database that help us to track the various related relations. The scientific analysis of SAHS requirement led us to trace requirements to its classes. Since SAHS posses, all ITSC architecture requirements plus five any conclusions from the scientific analysis may directly generalize to the ITSC architecture.

The analysis shows that CAHS physical classes can be modeled by thirty-one classes; their names and numbers are given in Table 6.1. We created a class called Backup communication class (7) to present the Satellitic backup for the four-communication channels that are available in ITSC architecture and which the backbone communication class (6) is in its general form. We will keep a close attention on classes 6 and 7 in order to monitor its participation in the fulfillment of the requirement functionality.

The study covers 694 requirements back to their responsible classes. The analysis report details are given in Appendix C. A snap shot of that report is shown below.

SAHS #	RDescription	UserR#	FunctionalR#	UseCA#	UseCB#	ClassR#
6	The SAHS shall provide travellers with instructions on turns and other manoeuvres to reach selected destinations.	1.2	1.1.1.3 1.1.3 6.2.1.1 6.2.1.2 6.2.1.3 6.2.1.4 6.2.2 6.2.3 6.2.4 6.2.5 6.2.6 6.6.2.1 6.7.2.1.1 6.7.2.2	1A.1.1.3 1A.1.3 6A.2.1.1 6A.2.1.2 6A.2.1.3 6A.2.1.4 6A.2.2 6A.2.3 6A.2.4 6A.2.5 6A.2.6 6A.6.2.1 6A.7.2.1.1 6A.7.2.2	1B.1.1.3 1B.1.3 6B.2.1.1 6B.2.1.2 6B.2.1.3 6B.2.1.4 6B.2.2 6B.2.3 6B.2.4 6B.2.5 6B.2.6 6B.6.2.1 6B.7.2.1.1 6B.7.2.2	6 7 16 13 31 30
7	The SAHS shall Enable route planning and detailed route guidance based on static, stored information collected through in-vehicle sensory, location	1.2.1	6.2.2 6.2.3 6.6.2.1 6.7.2.1.1 6.7.2.2	6A.2.2 6A.2.3 6A.6.2.1 6A.7.2.1.1 6A.7.2.2	6B.2.2 6B.2.3 6B.6.2.1 6B.7.2.1.1 6B.7.2.2	6 7 31 30 13

Table 6.1 SAHS Classes and its related numbers

SAHS class name	Class Number
Automated Highway System	1
Centers	2
Wayside	3
Travelers	4
Satellitic Vehicles	5
Backbone Communication	6
Satellitic Backup Communication	7
Archived Data Management	8
Commercial Satellitic Vehicle Administration	9
Emissions Management	10
Fleet and Freight Management	12
Information Service Provider	13
Maintenance Management	14
Toll Administration	15
Traffic Management	16
Transit Management	17
Emergency Management	18
Commercial Satellitic Vehicle Check	19
Intermodal Terminal	20
Parking Management	21
Roadway	22
Toll Collection	23
Personal Information Access	24
Remote Traveler Support Subsystem	25
Commercial Satellitic Vehicle	26
Emergency Satellitic Vehicle	27
Intermodal Container	28
Maintenance Satellitic Vehicle	29
Transit Satellitic Vehicle	30
Satellitic Vehicle	31

In the SAHS solution, **we now have two classes that will fulfill the communication aspects of the architecture. Backbone communication class No. 6, and Satellitic Backup communication class No.7 will handle the communication in case class 6 is malfunctioning.**

6.3.1 Analysis

1. We were able to trace most of the requirement to the responsible class.

2. The study shows that in order for the requirements to fulfill its functionality, it must use the communication system to perform its duty. Therefore, the communication subsystem is the backbone of the system. One of the most important findings of this study is the need for a backup communication system to increase the reliability and functionality of the system. In the SAHS solution, we now have two classes that will fulfill the communication aspects of the architecture. Backbone communication class No. 6, and Satellitic Backup communication class No.7 will handle the communication in case 6 has a fault.

3. The scientific analysis enabled us to model and to document in an OOAD methodology, the findings of the study. We were able to illustrate successfully the two deficiencies of the ITSC architecture. These deficiencies are the development methodology and the lack of backup communication system. SAHS present a solution for such deficiencies.

4. This study enabled us to find a solution for the two major problems that this architecture inherited from the experience of the other ITS architectures. It showed how the entire system relies upon its communication subsystem and how important it is to the overall functionality and reliability. Furthermore, in order to increase the flexibility and stability of the system the architecture must be based on an object-oriented methodology. The study provides us with a

clear picture of the need for a documental and visualizes solution analysis for the two problems that we have mentioned by partial development of SAHS.

6.4 DOCUMENTAL SOLUTION ANALYSIS

In the documental analysis, the flow of events is captured in this partial development of SAHS in text. The numbering for this documentation is related to the number of the process in the ITSC architecture. The documentations captured the communication from the actors to the various classes that are needed to perform the desired functionality. For each third level uses case. We focused only on one-way communication in the documental analysis only the actors are communicating with the system and subsystems. The system and subsystems will meet most requirements of the actors, but will not directly give specific answers to the actors. This documental analysis is based on the requirements that are provided by ITSC architecture, we analyzed and documented the following:

- For each use case of SAHS document Characteristic Information
- For each use case of SAHS document Main Success Scenario
- For each use case of SAHS document Scenario Extensions
- For each use case of SAHS document Related Information
- For each use case of SAHS document Open Issues
- Analyse the extent to which the backbone communication class and thee backup communication class participates in fulfilling all use cases in order to have a successful scenario

The documentation analysis for SAHS third level use cases are given Appendix D. For an example of this document analysis, ten use cases were chosen:

- 4A.4: CAHS Use Case Name: Support Security and Coordination
- 6A.2: CAHS Use Case Name: Provide Information Services
- 6A.8: CAHS Use Case Name: Provide Traveller Personal Services
- 7A.2: CAHS Use Case Name: Provide Electronic Parking Payment
- 7A.4:Use CAHS Case Name: Carry-out Centralized Payments Processing

- 4B.4: CAHS Use Case Name: Support Security and Coordination but with the use of the Satellitic Backup Communication System
- 6B.2: CAHS Use Case Name: Provide Information Services but with the use of the Satellitic Backup Communication System.
- 6B.8: CAHS Use Case Name: Provide Traveller Personal Services but with the use of the Satellitic Backup Communication System
- 7B.2: CAHS Use Case Name: Provide Electronic Parking Payment but with the use of the Satellitic Backup Communication System
- 7B.4:Use CAHS Case Name: Carry-out Centralized Payments Processing but with the use of the Satellitic Backup Communication System

The documentations are as follows:

4A.4: SAHS Use Case Name: Support Security and Coordination

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the Use Case.

Goal In Context:	Support Security and Coordination
Scope:	SAHS
Level:	Strategic
Pre-Condition:	System is ON and the Backbone Communication System is Functioning
Success End Condition:	Security is Coordinated and Supported
Failed End Condition:	Security is not Coordinated and Supported
Primary Actor:	32, 33, and 34
Trigger Event:	System Clock

Main Success Scenario

This Scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1.	17	4.4.1.1 Manage Transit Security	17
2.	33	4.4.1.2 Manage Transit Emergencies	6
3.	6	4.4.1.2 Manage Transit Emergencies	30
4.	33	4.4.1.3 Provide Transit System Operator Security Interface	6
5.	6	4.4.1.3 Provide Transit System Operator Security Interface	17
6.	34	4.4.1.4 Provide Transit External Interface for Emergencies	6
7.	6	4.4.1.4 Provide Transit External Interface for Emergencies	17
8.	33	4.4.1.5 Provide Transit Driver Interface for Emergencies	6
9.	6	4.4.1.5 Provide Transit Driver Interface for Emergencies	30
10.	17	4.4.1.6 Collect Transit Vehicle Emergency Information	17
11.	32	4.4.1.7 Monitor Secure Area	6
12.	6	4.4.1.7 Monitor Secure Area	25
13.	33	4.4.1.8 Report Traveller Emergencies	6
14.	6	4.4.1.8 Report Traveller Emergencies	25
15.	33	4.4.2 Coordinate Multiple Agency Responses to Incidents	6
16.	6	4.4.2 Coordinate Multiple Agency Responses to Incidents	17
17.	33	4.4.3 Generate Responses for Incidents	6
18.	6	4.4.3 Generate Responses for Incidents	17
19.	34	4.4.4 Coordinate Transit Disaster Response	6
20.	6	4.4.4 Coordinate Transit Disaster Response	17

Scenario Extensions

This is a listing of how each step in the Main Success Scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the Main Success Scenario is rejoined or the Failed End Condition is met. The Step refers to the Failed Step in the Main Success Scenario and has a letter associated with it. I.E if Step 3 fails the Extension Step is 3a.

Step	Condition	Action Description
2a	Backbone Communication System Failure	Switch to Satellitic Backup Communication System

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	Variable	Possible Variations
2a	Backbone Communication System Failure	Switch to Satellitic Backup Communication System

Related Information

The following table gives the information that is related to the Use Case.

Schedule:	ASAP
Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Manage Transit
Sub Use Case(s):	N/A
Channel To Primary Actor:	Backbone Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	Problem solved

6A.2: Use Case Name: Provide Information Services

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the Use Case.

Goal In Context:	Provide Information Services
Scope:	SAHS
Level:	Strategic
Pre-Condition:	System is On and Backbone Communication System is functioning
Success End Condition:	Information Services is provided
Failed End Condition:	Information Services is not provided
Primary Actor:	33, 34 and 35
Trigger Event:	Request from 34

Main Success Scenario

This Scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1.	34	6.2.1.1 Collect Traffic Data for Advisory Messages	6
2.	6	6.2.1.1 Collect Traffic Data for Advisory Messages	13
3.	35	6.2.1.1 Collect Traffic Data for Advisory Messages	6
4.	6	6.2.1.1 Collect Traffic Data for Advisory Messages	13
5.	13	6.2.1.2 Provide Traffic and Transit Advisory Messages	13
6.	35	6.2.1.3 Collect Transit Data for Advisory Messages	6
7.	6	6.2.1.3 Collect Transit Data for Advisory Messages	13
8.	13	6.2.1.4 Provide Traffic and Transit Broadcast Messages	13
9.	33	6.2.1.5 Provide ISP Operator Broadcast Parameters Interface	6

10.	6	6.2.1.5 Provide ISP Operator Broadcast Parameters Interface	13
11.	33	6.2.1.6 Provide Transit Advisory Data On Vehicle	6
12.	6	6.2.1.6 Provide Transit Advisory Data On Vehicle	30
13.	31	6.2.2 Prepare and Output In-vehicle Displays	31
14.	33	6.2.3 Provide Transit User Advisory Interface	6
15.	6	6.2.3 Provide Transit User Advisory Interface	30
16.	13	6.2.4 Collect Yellow Pages Data	13
17.	33	6.2.5 Provide Driver Interface	6
18.	6	6.2.5 Provide Driver Interface	31
19.	13	6.2.6 Provide Yellow Pages Data and Reservations	13

Scenario Extensions

This is a listing of how each step in the Main Success Scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the Main Success Scenario is rejoined or the Failed End Condition is met. The Step refers to the Failed Step in the Main Success Scenario and has a letter associated with it. I.E if Step 3 fails the Extension Step is 3a.

Step	Condition	Action Description
1a	Backbone Communication System Failure	Switch to Satellitic Backup Communication Sysytem

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	Variable	Possible Variations
1a	Backbone Communication System Failure	Switch to Satellitic Backup Communication Sysytem

Related Information

The following table gives the information that is related to the Use Case.

Schedule:	ASAP

Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Provide Driver And Traveller Services
Sub Use Case(s):	N/A
Channel To Primary Actor:	Satellitic Backup Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	Problem solved

6A.8: AHS USE CASE NAME: PROVIDE TRAVELLER PERSONAL SERVICES

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the Use Case.

Goal In Context:	Provide Traveller Personal Services
Scope:	SAHS
Level:	Strategic
Pre-Condition:	System Must be On and the Backbone Communication is functioning
Success End Condition:	Traveller Personal Services is provided

Failed End Condition: Traveller Personal Services is not provided
Primary Actor: 32 and 34
Trigger Event: System Clock

Main Success Scenario

This Scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1.	24	6.8.1.1.1 Determine Personal Portable Device Guidance Method	24
2.	24	6.8.1.1.2 Provide Personal Portable Device Dynamic Guidance	24
3.	24	6.8.1.1.3 Provide Personal Portable Device Autonomous Guidance	24
4.	32	6.8.1.2 Provide Personal Portable Device Guidance Interface	6
5.	6	6.8.1.2 Provide Personal Portable Device Guidance Interface	24
6.	24	6.8.1.3 Process Personal Portable Device Location Data	24
7.	34	6.8.1.4 Update Traveller Navigable Map Database	6
8.	6	6.8.1.4 Update Traveller Navigable Map Database	24
9.	24	6.8.1.5 Provide Traveller Emergency Message Interface	24
10.	32	6.8.2.1 Build Traveller Personal Security Message	6
11.	6	6.8.2.1 Build Traveller Personal Security Message	24
12.	24	6.8.2.2 Provide Traveller Emergency Communications Function	24
13.	24	6.8.3.1 Get Traveller Personal Request	24
14.	24	6.8.3.2 Provide Traveller with Personal Travel Information	24
15.	32	6.8.3.3 Provide Traveller Personal Interface	6
16.	6	6.8.3.3 Provide Traveller Personal Interface	24
17.	34	6.8.3.4 Update Traveller Personal Display Map Data	6
18.	6	6.8.3.4 Update Traveller Personal Display Map Data	24

Scenario Extensions

This is a listing of how each step in the Main Success Scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the Main Success Scenario is rejoined or the Failed End Condition is met. The Step refers to the Failed Step in the Main Success Scenario and has a letter associated with it. I.E if Step 3 fails the Extension Step is 3a.

Step	Condition	Action Description
4a	Backbone Communication System Failure	Switch to Satellitic Backup Communication System

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	Variable	Possible Variations
4a	Backbone Communication System Failure	Switch to Satellitic Backup Communication System

Related Information

The following table gives the information that is related to the Use Case.

Schedule:	ASAP
Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Provide Driver And Traveller Services
Sub Use Case(s):	N/A
Channel To Primary Actor:	Satellitic Backup Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	Problem solved

7A.2: SAHS Use Case Name: Provide Electronic Parking Payment

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the Use Case.

Goal In Context:	Provide Electronic Parking Payment
Scope:	Automated Highway System
Level:	Strategic
Pre-Condition:	The system is ON and the Backbone Communication System is functioning
Success End Condition:	Electronic Parking Payment is Provide
Failed End Condition:	Electronic Parking Payment is not complete
Primary Actor:	34 and 33
Trigger Event:	System Clock

Main Success Scenario

This Scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1	21	7.2.1.1 Read Parking Lot Tag Data	21
2	21	7.2.1.10 Determine Advanced Charges	21
3	21	7.2.1.2 Calculate Vehicle Parking Lot Charges	21
4	34	7.2.1.3 Collect Bad Charge Payment Data	6
5	6	7.2.1.3 Collect Bad Charge Payment Data	21
6	21	7.2.1.4 Check for Advanced Parking Lot Payment	21
7	21	7.2.1.5 Bill Driver for Parking Lot Charges	21
8	34	7.2.1.6 Manage Parking Lot Financial Processing	6
9	6	7.2.1.6 Manage Parking Lot Financial Processing	21

10	33	7.2.1.7 Update Parking Lot Data	6
11	6	7.2.1.7 Update Parking Lot Data	21
12	33	7.2.1.8 Register for Advanced Parking Lot Payment	6
13	6	7.2.1.8 Register for Advanced Parking Lot Payment	21
14	33	7.2.1.9 Manage Parking Lot Reservations	6
15	6	7.2.1.9 Manage Parking Lot Reservations	21
16	21	7.2.2 Produce Parking Lot Displays	21
17	21	7.2.3 Obtain Parking Lot Violator Image	21
18	34	7.2.4 Provide Driver Parking Lot Payment Interface	6
19	6	7.2.4 Provide Driver Parking Lot Payment Interface	31
20	33	7.2.4 Provide Driver Parking Lot Payment Interface	6
21	6	7.2.4 Provide Driver Parking Lot Payment Interface	31
22	21	7.2.5 Detect Vehicle for Parking Lot Payment	21
23	13	7.2.6 Distribute Advanced Tolls and Fares	13
24	34	7.2.7 Provide Payment Instrument Interface for Parking	6
25	6	7.2.7 Provide Payment Instrument Interface for Parking	31

Scenario Extensions

This is a listing of how each step in the Main Success Scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the Main Success Scenario is rejoined or the Failed End Condition is met. The Step refers to the Failed Step in the Main Success Scenario and has a letter associated with it. I.E if Step 3 fails the Extension Step is 3a.

Step	Condition	Action Description
4a	Backbone Communication System failure	Switch to Satellitic Backup Communication System see use case number 7B.2

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	Variable	Possible Variations
4a	Backbone Communication System failure	Switch to Satellitic Backup Communication System see use case number 7B.2

Related Information

The following table gives the information that is related to the Use Case.

Schedule:	ASAP
Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Provide Electronic Payment Services
Sub Use Case(s):	N/A
Channel To Primary Actor:	Backbone Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	The problem is solved

7A.4:Use Case Name: Carry-out Centralized Payments Processing

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the Use Case.

Goal In Context:	Carry-out Centralized Payments Processing
Scope:	SAHS
Level:	Strategic
Pre-Condition:	The System is On and Backbone communication is functioning
Success End Condition:	Centralized Payments Processing is carried-out
Failed End Condition:	Centralized Payments Processing is not carried-out
Primary Actor:	34

Trigger Event: Request from 34

Main Success Scenario

This Scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1	34	7.4.1.1 Process Commercial Vehicle Payments	6
2	6	7.4.1.1 Process Commercial Vehicle Payments	26
3	34	7.4.1.2 Process Yellow Pages Services Provider Payments	6
4	6	7.4.1.2 Process Yellow Pages Services Provider Payments	13
5	34	7.4.1.3 Process Driver Map Update Payments	6
6	6	7.4.1.3 Process Driver Map Update Payments	13
7	34	7.4.1.4 Process Traveller Map Update Payments	6
8	6	7.4.1.4 Process Traveller Map Update Payments	13
9	34	7.4.1.5 Process Transit User Other Services Payments	6
10	6	7.4.1.5 Process Transit User Other Services Payments	13
11	34	7.4.1.6 Process Traveller Trip and Other Services Payments	6
12	6	7.4.1.6 Process Traveller Trip and Other Services Payments	13
13	13	7.4.1.7 Collect Payment Transaction Records	13
14	34	7.4.1.8 Process Traveller Rideshare Payments	6
15	6	7.4.1.8 Process Traveller Rideshare Payments	13
16	13	7.4.2 Collect Price Data for AHS Use	13

Scenario Extensions

This is a listing of how each step in the Main Success Scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the Main Success Scenario is rejoined or the Failed End Condition is met. The Step refers to the Failed Step in the Main Success Scenario and has a letter associated with it. I.E if Step 3 fails the Extension Step is 3a.

Step	Condition	Action Description
1a	Backbone Communication System failure	Switch to Satellitic Backup Communication System

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	Variable	Possible Variations
1a	Backbone Communication System failure	Switch to Satellitic Backup Communication System

Related Information

The following table gives the information that is related to the Use Case.

Schedule:	ASAP
Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Provide Electronic Payment Services
Sub Use Case(s):	N/A
Channel To Primary Actor:	Backbone Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	Problem solved

4B.4: SAHS Use Case Name: Support Security and Coordination With Satellitic Backup Communication System

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the Use Case.

Goal In Context:	Support Security and Coordination

Scope:	SAHS
Level:	Strategic
Pre-Condition:	System is ON and the Backbone Communication System is Functioning
Success End Condition:	Security is Coordinated and Supported
Failed End Condition:	Security is not Coordinated and Supported
Primary Actor:	32, 33, and 34
Trigger Event:	System Clock

Main Success Scenario

This Scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
21.	17	4.4.1.1 Manage Transit Security	17
22.	33	4.4.1.2 Manage Transit Emergencies	7
23.	7	4.4.1.2 Manage Transit Emergencies	30
24.	33	4.4.1.3 Provide Transit System Operator Security Interface	7
25.	7	4.4.1.3 Provide Transit System Operator Security Interface	17
26.	34	4.4.1.4 Provide Transit External Interface for Emergencies	7
27.	7	4.4.1.4 Provide Transit External Interface for Emergencies	17
28.	33	4.4.1.5 Provide Transit Driver Interface for Emergencies	7
29.	7	4.4.1.5 Provide Transit Driver Interface for Emergencies	30
30.	17	4.4.1.6 Collect Transit Vehicle Emergency Information	17
31.	32	4.4.1.7 Monitor Secure Area	7
32.	7	4.4.1.7 Monitor Secure Area	25
33.	33	4.4.1.8 Report Traveller Emergencies	6
34.	6	4.4.1.8 Report Traveller Emergencies	25
35.	33	4.4.2 Coordinate Multiple Agency Responses to Incidents	7
36.	7	4.4.2 Coordinate Multiple Agency Responses to Incidents	17
37.	33	4.4.3 Generate Responses for Incidents	7
38.	7	4.4.3 Generate Responses for Incidents	17

39.	34	4.4.4 Coordinate Transit Disaster Response	7
40.	7	4.4.4 Coordinate Transit Disaster Response	17

Scenario Extensions

This is a listing of how each step in the Main Success Scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the Main Success Scenario is rejoined or the Failed End Condition is met. The Step refers to the Failed Step in the Main Success Scenario and has a letter associated with it. I.E if Step 3 fails the Extension Step is 3a.

Step	Condition	Action Description
2a	Satellitic Backup Communication System Failure	Switch to conventional vehicle highway system

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	Variable	Possible Variations
2a	Satellitic Backup Communication System Failure	Switch to conventional vehicle highway system

Related Information

The following table gives the information that is related to the Use Case.

Schedule:	ASAP
Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Manage Transit
Sub Use Case(s):	N/A
Channel To Primary Actor:	Satellitic Backup Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	Problem solved

6B.2: SAHS Use Case Name: Provide Information Services With Satellitic Backup Communication System

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the Use Case.

Goal In Context:	Provide Information Services
Scope:	SAHS
Level:	Strategic
Pre-Condition:	System is On and Satellitic Backup Communication System is functioning
Success End Condition:	Information Services is provided
Failed End Condition:	Information Services is not provided
Primary Actor:	33, 34 and 35
Trigger Event:	Request from 34

Main Success Scenario

This Scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1	34	6.2.1.1 Collect Traffic Data for Advisory Messages	7
2	7	6.2.1.1 Collect Traffic Data for Advisory Messages	13
3	35	6.2.1.1 Collect Traffic Data for Advisory Messages	7
4	7	6.2.1.1 Collect Traffic Data for Advisory Messages	13
5	13	6.2.1.2 Provide Traffic and Transit Advisory Messages	13
6	35	6.2.1.3 Collect Transit Data for Advisory Messages	7

7	7	6.2.1.3 Collect Transit Data for Advisory Messages	13
8	13	6.2.1.4 Provide Traffic and Transit Broadcast Messages	13
9	33	6.2.1.5 Provide ISP Operator Broadcast Parameters Interface	7
10	7	6.2.1.5 Provide ISP Operator Broadcast Parameters Interface	13
11	33	6.2.1.6 Provide Transit Advisory Data On Vehicle	7
12	7	6.2.1.6 Provide Transit Advisory Data On Vehicle	30
13	31	6.2.2 Prepare and Output In-vehicle Displays	31
14	33	6.2.3 Provide Transit User Advisory Interface	7
15	7	6.2.3 Provide Transit User Advisory Interface	30
16	13	6.2.4 Collect Yellow Pages Data	13
17	33	6.2.5 Provide Driver Interface	7
18	7	6.2.5 Provide Driver Interface	31
19	13	6.2.6 Provide Yellow Pages Data and Reservations	13

Scenario Extensions

This is a listing of how each step in the Main Success Scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the Main Success Scenario is rejoined or the Failed End Condition is met. The Step refers to the Failed Step in the Main Success Scenario and has a letter associated with it. I.E if Step 3 fails the Extension Step is 3a.

Step	**Condition**	**Action Description**
1a	Satellitic Backup Communication System Failure	Switch to conventional vehicle highway system

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	**Variable**	**Possible Variations**
1a	Satellitic Backup Communication System Failure	Switch to conventional vehicle highway system

Related Information

The following table gives the information that is related to the Use Case.

Schedule:	ASAP

Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Provide Driver And Traveller Services
Sub Use Case(s):	N/A
Channel To Primary Actor:	Satellitic Backup Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	Problem solved

6B.8: AHS USE CASE NAME: PROVIDE TRAVELLER PERSONAL SERVICES

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the Use Case.

Goal In Context:	Provide Traveller Personal Services
Scope:	SAHS
Level:	Strategic
Pre-Condition:	System Must be On and the Backbone Communication is functioning
Success End Condition:	Traveller Personal Services is provided
Failed End Condition:	Traveller Personal Services is not provided
Primary Actor:	32 and 34

Trigger Event: System Clock

Main Success Scenario

This Scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1	24	6.8.1.1.1 Determine Personal Portable Device Guidance Method	24
2	24	6.8.1.1.2 Provide Personal Portable Device Dynamic Guidance	24
3	24	6.8.1.1.3 Provide Personal Portable Device Autonomous Guidance	24
4	32	6.8.1.2 Provide Personal Portable Device Guidance Interface	7
5	7	6.8.1.2 Provide Personal Portable Device Guidance Interface	24
6	24	6.8.1.3 Process Personal Portable Device Location Data	24
7	34	6.8.1.4 Update Traveller Navigable Map Database	7
8	7	6.8.1.4 Update Traveller Navigable Map Database	24
9	24	6.8.1.5 Provide Traveller Emergency Message Interface	24
10	32	6.8.2.1 Build Traveller Personal Security Message	7
11	7	6.8.2.1 Build Traveller Personal Security Message	24
12	24	6.8.2.2 Provide Traveller Emergency Communications Function	24
13	24	6.8.3.1 Get Traveller Personal Request	24
14	24	6.8.3.2 Provide Traveller with Personal Travel Information	24
15	32	6.8.3.3 Provide Traveller Personal Interface	7
16	7	6.8.3.3 Provide Traveller Personal Interface	24
17	34	6.8.3.4 Update Traveller Personal Display Map Data	7
18	7	6.8.3.4 Update Traveller Personal Display Map Data	24

Scenario Extensions

This is a listing of how each step in the Main Success Scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the Main Success Scenario is rejoined or the Failed End Condition is met. The Step refers to the Failed Step in the Main Success Scenario and has a letter associated with it. I.E if Step 3 fails the Extension Step is 3a.

Step	Condition	Action Description
4a	Satellitic Backup Communication System Failure	Switch to conventional vehicle highway system

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	Variable	Possible Variations
4a	Satellitic Backup Communication System Failure	Switch to conventional vehicle highway system

Related Information

The following table gives the information that is related to the Use Case.

Schedule:	ASAP
Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Provide Driver And Traveller Services
Sub Use Case(s):	N/A
Channel To Primary Actor:	Satellitic Backup Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	Problem solved

7A.2: SAHS Use Case Name: Provide Electronic Parking Payment but with the use of Satellitic Backup Communication System

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the Use Case.

Goal In Context:	Provide Electronic Parking Payment
Scope:	Automated Highway System
Level:	Strategic
Pre-Condition:	The system is ON and the Backbone Communication System is functioning
Success End Condition:	Electronic Parking Payment is Provided
Failed End Condition:	Electronic Parking Payment is not complete
Primary Actor:	34 and 33
Trigger Event:	System Clock

Main Success Scenario

This Scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1	21	7.2.1.1 Read Parking Lot Tag Data	21
2	21	7.2.1.10 Determine Advanced Charges	21
3	21	7.2.1.2 Calculate Vehicle Parking Lot Charges	21
4	34	7.2.1.3 Collect Bad Charge Payment Data	7
5	7	7.2.1.3 Collect Bad Charge Payment Data	21
6	21	7.2.1.4 Check for Advanced Parking Lot Payment	21
7	21	7.2.1.5 Bill Driver for Parking Lot Charges	21
8	34	7.2.1.6 Manage Parking Lot Financial Processing	7
9	7	7.2.1.6 Manage Parking Lot Financial Processing	21

10	33	7.2.1.7 Update Parking Lot Data	7
11	7	7.2.1.7 Update Parking Lot Data	21
12	33	7.2.1.8 Register for Advanced Parking Lot Payment	7
13	7	7.2.1.8 Register for Advanced Parking Lot Payment	21
14	33	7.2.1.9 Manage Parking Lot Reservations	7
15	7	7.2.1.9 Manage Parking Lot Reservations	21
16	21	7.2.2 Produce Parking Lot Displays	21
17	21	7.2.3 Obtain Parking Lot Violator Image	21
18	34	7.2.4 Provide Driver Parking Lot Payment Interface	7
19	7	7.2.4 Provide Driver Parking Lot Payment Interface	31
20	33	7.2.4 Provide Driver Parking Lot Payment Interface	7
21	7	7.2.4 Provide Driver Parking Lot Payment Interface	31
22	21	7.2.5 Detect Vehicle for Parking Lot Payment	21
23	13	7.2.6 Distribute Advanced Tolls and Fares	13
24	34	7.2.7 Provide Payment Instrument Interface for Parking	7
25	7	7.2.7 Provide Payment Instrument Interface for Parking	31

Scenario Extensions

This is a listing of how each step in the Main Success Scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the Main Success Scenario is rejoined or the Failed End Condition is met. The Step refers to the Failed Step in the Main Success Scenario and has a letter associated with it. I.E if Step 3 fails the Extension Step is 3a.

Step	**Condition**	**Action Description**
4a	Satellitic Backup Communication System failure	Switch to Conventional Vehicle highway System

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	**Variable**	**Possible Variations**
4a	Satellitic Backup Communication System failure	Switch to Conventional Vehicle highway System

Related Information

The following table gives the information that is related to the Use Case.

Schedule:	ASAP
Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Provide Electronic Payment Services
Sub Use Case(s):	N/A
Channel To Primary Actor:	Satellitic Backup Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	The problem is solved

7B.4:Use Case Name: Carry-out Centralized Payments Processing but with the use of the Satellitic Backup Communication System

Characteristic Information

The following defines information that pertains to this particular use case. Each piece of information is important in understanding the purpose behind the Use Case.

Goal In Context:	Carry-out Centralized Payments Processing
Scope:	SAHS
Level:	Strategic
Pre-Condition:	The System is On and Backbone communication is functioning
Success End Condition:	Centralized Payments Processing is carried-out
Failed End Condition:	Centralized Payments Processing is not carried-out
Primary Actor:	34
Trigger Event:	Request from 34

Main Success Scenario

This Scenario describes the steps that are taken from trigger event to goal completion when everything works without failure. It also describes any required cleanup that is done after the goal has been reached. The steps are listed below:

Step	From	Action Description	To
1	34	7.4.1.1 Process Commercial Vehicle Payments	7
2	7	7.4.1.1 Process Commercial Vehicle Payments	26
3	34	7.4.1.2 Process Yellow Pages Services Provider Payments	7
4	7	7.4.1.2 Process Yellow Pages Services Provider Payments	13
5	34	7.4.1.3 Process Driver Map Update Payments	6
6	7	7.4.1.3 Process Driver Map Update Payments	13
7	34	7.4.1.4 Process Traveller Map Update Payments	7
8	7	7.4.1.4 Process Traveller Map Update Payments	13
9	34	7.4.1.5 Process Transit User Other Services Payments	6
10	7	7.4.1.5 Process Transit User Other Services Payments	13
11	34	7.4.1.6 Process Traveller Trip and Other Services Payments	7
12	7	7.4.1.6 Process Traveller Trip and Other Services Payments	13
13	13	7.4.1.7 Collect Payment Transaction Records	13
14	34	7.4.1.8 Process Traveller Rideshare Payments	7
15	7	7.4.1.8 Process Traveller Rideshare Payments	13
16	13	7.4.2 Collect Price Data for AHS Use	13

Scenario Extensions

This is a listing of how each step in the Main Success Scenario can be extended. Another way to think of this is how can things go wrong. The extensions are followed until either the Main Success Scenario is rejoined or the Failed End Condition is met. The Step refers to the Failed Step in the Main Success Scenario and has a letter associated with it. I.E if Step 3 fails the Extension Step is 3a.

Step	Condition	Action Description
1a	Satellitic Backup Communication System failure	Switch to conventional Highway Vehicle System

Scenario Variations

If a variation can occur in how a step is performed it will be listed here.

Step	Variable	Possible Variations
1a	Satellitic Backup Communication System failure	Switch to conventional Highway Vehicle System

Related Information

The following table gives the information that is related to the Use Case.

Schedule:	ASAP
Priority:	Must
Performance Target:	N/A
Frequency:	N/A
Super Use Case:	Provide Electronic Payment Services
Sub Use Case(s):	N/A
Channel To Primary Actor:	Satellitic Backup Communication System
Secondary Actor(s):	N/A
Channel(s) To Secondary Actor(s):	N/A

Open Issues

The following table provides insight to any unresolved problems or questions. These are the things that seem to apply but could not be fit into this use case on this pass.

Issue ID	Issue Description
1	Problem solved

The rest of SAHS Documentations can be found in APPENDIX D

Summary of the Documental Analysis Findings

In the documental analysis, the extent to which the backbone communication class and backup communication class participate in fulfilling a use case successful scenario is monitored. All the examples use cases documentations shows that backup communication class is taking over when the backbone communication class is in malfunction status (See use case B type), which, provided us with a successful scenario. This is true for almost all the documentation use cases of SAHS. **This proves that our solution for backup communication is adequate.**

6.5 Visual Solution Analysis

In the previous sections, we conducted a scientific and documental analysis of the requirements of SAHS. It is clear in our initial estimates that the problem of communications is solved by the provided Satellitic Backup Communication System. We will visualize the solution in order to provide the necessary proof that the communication problem is solved. SAHS will be partially modeled, visualized, and detailed enough for the satisfaction of solving the communication problem and that object oriented architecture provided the stability and flexibility the original ITSC architecture lacks. Since SAHS possesses all ITSC architecture requirements plus five backup requirements, we can generalize any finding back to ITSC architecture. We will follow the one way the messages from actors to the identified classes through the backbone communication system, and what steps that were taken for the partial development of an object -oriented modeling of the conceptual SAHS. These are as follows:

6.5.1 Creating Use Case

6.5.1.1 Identified Actors of the System

The ITSC architecture provides us with the required information about the SAHS's Actors. Actors interact with the proposed system asking for service or participating in providing some of the information required to fulfill the system capability.

The general-purpose Actor illustrated in Figure 6.3, which captures characteristics common to all users with its subclasses, Human, Environment, Other Systems, and Systems.

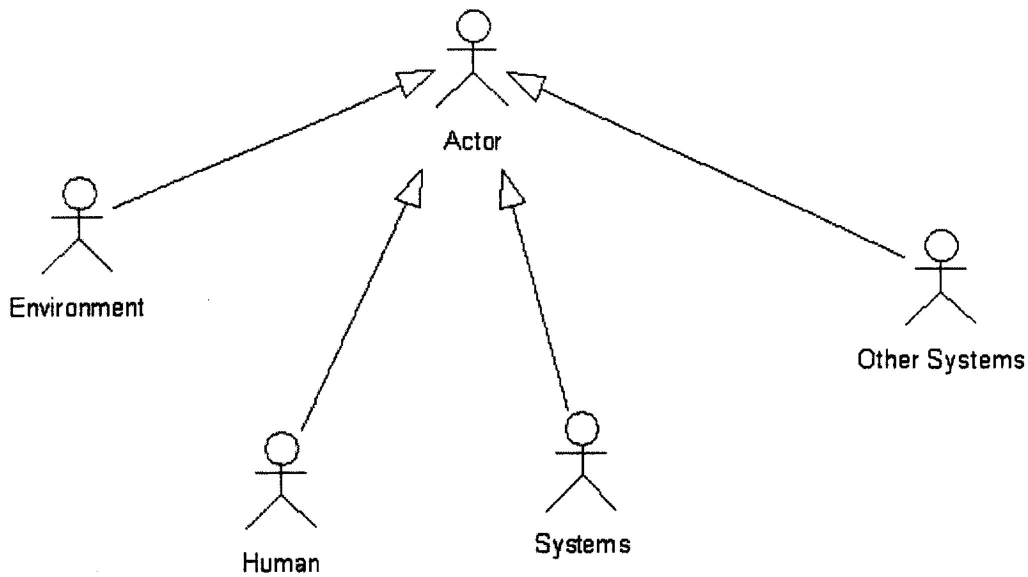

Figure 6.3 Object-oriented visualization of actor's categories in SAHS using UML

1 Human:

In Figure 6.4 Human Actors Categories are illustrated.

1a. Archived Data Administrator:

This actor represents the human operator who provides overall data management, administration, and monitoring duties for the SAHS data archive. This actor is responsible for the initiation of commands, requests, and queries that support the administration and management of an SAHS data archive.

1b. Vehicle Driver:

This actor represents the human entity that operates vehicles transporting goods. This actor is responsible for the initiations of Commercial Satellitic Vehicle driver and Satellitic Vehicle information and requests to the Commercial Vehicle Managing System.

1c. Commercial Satellitic Vehicle Manager:

This actor represents the human entities that are responsible for the dispatching and management of Commercial Satellitic Vehicle fleets. This actor is responsible for the initiations of inquiries from fleet manager requesting data from Commercial Satellitic Vehicle Management System.

1d. Commercial Satellitic Vehicle Operation Inspector

This actor represents the human entities that perform regulatory inspection of Commercial Satellitic Vehicles in the field. This actor is responsible for the initiations of manual override by the Commercial Satellitic Vehicle Roadside Facility Inspector of automated pass/pull-in signage information and Requests from the commercial vehicle inspector to operate the commercial vehicle inspection station.

1e. Driver:

This actor represents the human entity that operates a Satellitic Vehicle on the roadway. This actor is responsible for the initiations of driver commands to the Satellitic Vehicle and a traveller service request initiated by a driver or traveler. The request may result in a

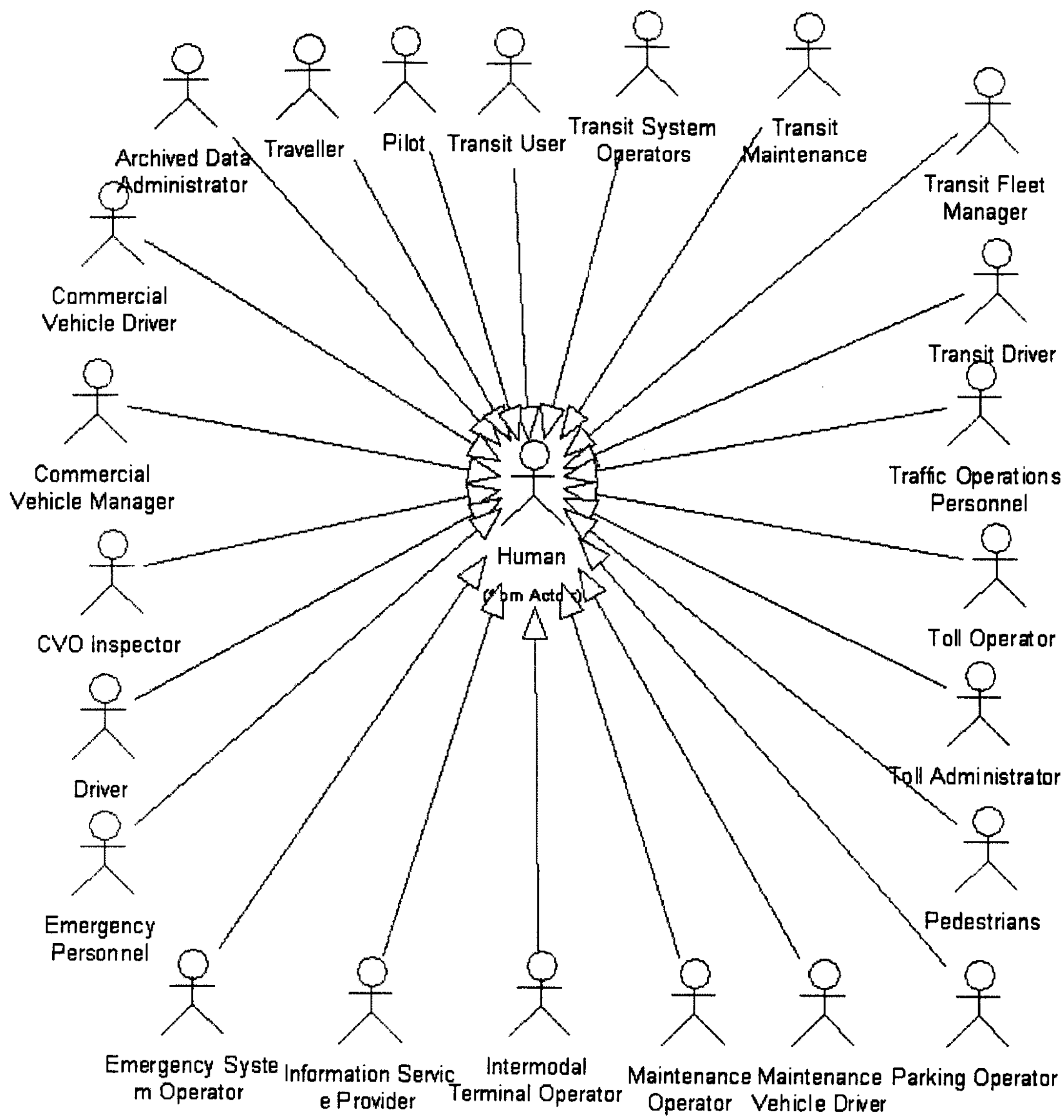

Figure 6.4 Object Oriented visualization of related Human actors for SAHS in UML

financial transaction, summon an emergency response, or initiate another service at the behest of the driver.

1f. Emergency Personnel:

This actor represents personnel that are responsible for police, fire, emergency medical services, towing, and other special response team activities at an incident site. This actor is responsible for the initiation of current incident status information and requests from emergency personnel in the field for information and/or resources.

1g. Emergency System Operator:

This actor represents the human entity that monitors all SAHS emergency requests, and sets up pre-defined responses to be executed by an emergency management system. This actor is responsible for the initiations of emergency operator inputs supporting call taking, dispatch, and other operations and communications center operator functions.

1h. Information Service Provider Operator:

This actor is the human entity that may be physically present at the ISP to monitor the operational status of the facility and provide human interface capabilities to Travelers and other ISP subsystems. This actor is responsible for the initiations of tuning and performance enhancement parameters to ISP algorithms.

1i. Intermodal Terminal Operator:

The actor is the human entity that operates the intermodal terminal subsystem. This actor is responsible for the initiations of inputs and control actions by an intermodal terminal operator.

1j. Maintenance Operator:

This actor represents the human entity that directly interfaces with the systems in the Maintenance Management subsystem. This actor is responsible for the initiations of information, control, and dispatching inputs from the maintenance operator.

1k. Maintenance Satellitic Vehicle Driver:

This actor represents the human entity that operates any maintenance Satellitic Vehicle. This actor is responsible for the initiations of information and control actions provided by the driver of the maintenance Satellitic Vehicle.

1j. Parking Operator:

This actor represents the human entity that may be physically present at the parking lot facility to monitor the operational status of the facility. This actor is responsible for the initiations of local parking operator inputs that query status and control the operation of the parking management system and request issued by a service provider for current parking service performance data.

1k. Pedestrians:

This actor represents the human entity that provides from a specialized form of the traveler, who is not using any type of Satellitic Vehicle as a form of transport. This actor is responsible for the initiations of request for pedestrian crossing and sensed presence of pedestrians and other non-motor Satellitic Vehicle travelers at roadway crossing or control points.

1l. Toll Administrator:

This actor represents the human entity that manages the back office payment administration systems for an electronic toll system. This actor is responsible for the initiations of instructions indicating the toll fees.

1m. Toll Operator:

This actor represents the human entity that manages the back office payment administration systems for an electronic toll system. This actor is responsible for the initiations of request for information from toll operators at toll collection sites.

1n. Traffic Operations Personnel:

This actor represents the human entity that directly interacts with Satellitic Vehicle traffic operations. This actor is responsible for the initiations of nominal pollution data compliance (reference) levels for each sector of an urban area, traffic operations requests for information, configuration changes, commands to adjust current traffic control strategies (e.g., adjusting signal timing plans, changing DMS messages), and other traffic operations data entry.

1o. Transit Driver:

This actor represents the human entity that is a special form of the Driver actor that receives and provides additional information that is specific to transit operations. This actor is responsible for the initiations of transit driver availability data that can be used to develop driver assignments and detailed operations schedules and transit driver emergency request as well as fare transaction data.

1p. Transit Fleet Manager:

This actor represents the human entity that is responsible for planning the operation of transit fleets; including monitoring and controlling the transit fleet route schedules and the transit fleet maintenance schedules. This actor is responsible for the initiations of instructions governing service availability, schedules, emergency response plans, transit personnel assignments, transit maintenance requirements, and other inputs that establish general system operating requirements and procedures.

1q. Transit Maintenance Personnel:

The actor represents the human entity that is actively responsible for monitoring, controlling, and planning the schedules for the maintenance of transit fleets. This actor is responsible for the initiations of current maintenance status of Satellitic Vehicle.

1r. Transit System Operators:

This actor represents the human entities that are responsible for all aspects of the Transit subsystem operation including planning and management. This actor is responsible for the initiations of information and control provided by transit system operators involving many aspects of managing transit operations.1s. Transit User:

This actor represents the human entities using public transit Satellitic Vehicles. This actor is responsible for the initiations of requests from transit user through an on-board or fixed location traveler information station.

1t. Traveler:

This actor represents any individual who uses transportation services. This actor is responsible for the initiations of requests by a traveler to summon assistance, request travel information, make a reservation, or request any other traveler service.

2 Environments:

Related Environmental Actors are illustrated in Figure 6.5.

2a. Environment:

This actor is the operational setting in which the SAHS interfaces operate. This actor is responsible for the initiations of atmospheric pollutant levels as monitored by air quality sensors and detection of specific localized hazards affecting the roadway.

2b. Potential Obstacles:

This actor represents any object that possesses the potential of being sensed and struck and thus possesses physical attributes. This actor is responsible for the initiations of the detection of an obstacle by Satellitic Vehicle or roadside equipment. Obstacles could include animals, other Satellitic Vehicles, pedestrians, rocks in roadway, etc.

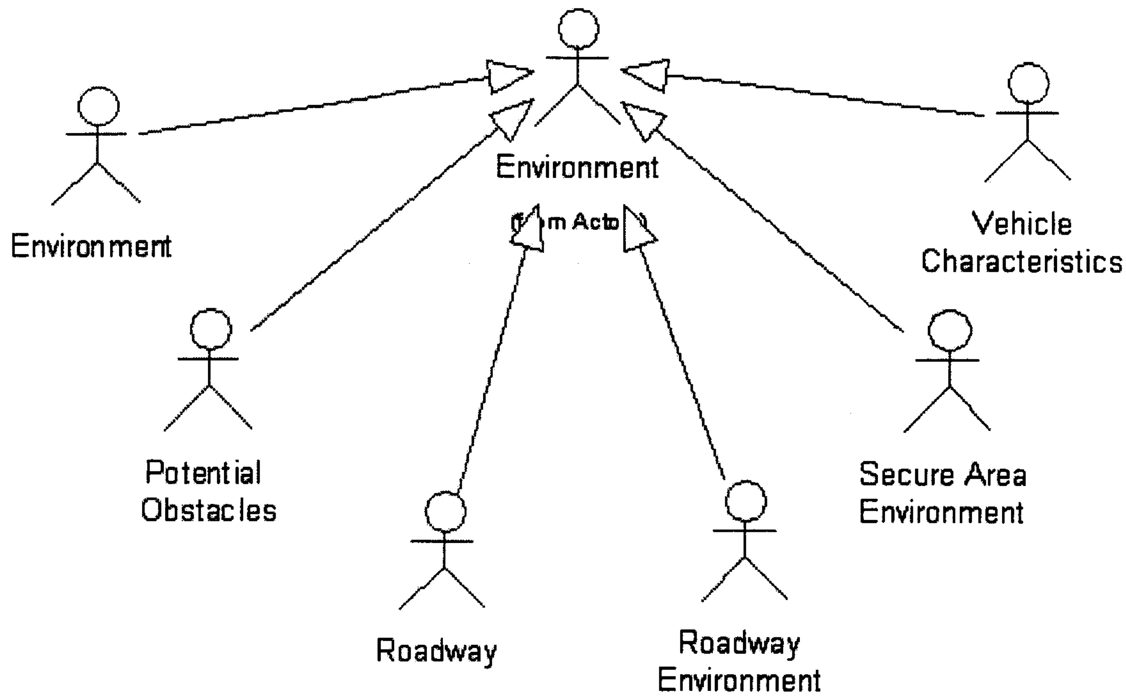

Figure 6.5 Object-oriented visualization of
related Environmental actors for SAHS in UML

2c. Roadway:

This actor represents the physical conditions and geometry of the surface or space on or through which Satellitic Vehicles travel from an origin to a destination. This actor is responsible for the initiations of the detection and measurement of road characteristics such as friction coefficient and general surface and sub-surface conditions, road geometry and markings, etc.

2d. Roadway Environment:

This actor represents the physical conditions surrounding the roadway itself. This actor is responsible for the initiations of the detection of specific localized hazards affecting the roadway, such as mud slides, avalanches, high winds and flooding and the weather and roadway conditions, that serve as the input environmental data for sensors at the roadside, This actor can be used in maintenance and other Satellitic Vehicles.

2e. Secure Area Environment:

This actor represents public access areas that transit users frequent during trips. Areas include bus stops, park and ride (PAR) facilities, kiosks, and transit transfer and multimodal transfer locations. This actor is responsible for the initiations of characteristics (visual, audible, and other) that are monitored by surveillance security systems via sensors.

2f. Traffic:

The Traffic actor represents the collective body of Satellitic Vehicles that travel on surface streets, arterials, highways, expressways, tollways, freeways, or any other vehicle travel surface. This actor is responsible for the initiations of physical traffic characteristics that are monitored and translated into macroscopic measures like occupancy, volume, density, and average speed. Point measures support presence detection and individual Satellitic Vehicle measures include determining speed for example. This may also include detection of non-vehicular traffic in the roadway, such as bicycles and pedestrians.

2g. Satellitic Vehicle Characteristics:

This actor represents the external view of an individual Satellitic Vehicle. This actor is responsible for the initiations of the physical or visible characteristics of an individual Satellitic Vehicle that can be measured to classify a vehicle and imaged to uniquely identify a Satellitic Vehicle.

3 Systems:

Related System Actors are illustrated in Figure 6.6.

3a. Archived Data User Systems:

This actor represents the systems users employed to access archived data. This actor is responsible for the initiation of a user requests that initiates data mining, analytical processing, aggregation or summarization, report formulation, or other advanced processing and analysis of archived data. The requests also include information that is used to identify and authenticate the user and to support electronic payment requirements, if any.

3b. Basic Satellitic Vehicle:

This actor represents the basic Satellitic Vehicle platform that interacts with and hosts SAHS electronics. This actor is responsible for the initiations of information provided to on-board SAHS equipment from the vehicle platform indicating current vehicle status.

3c. Commercial Satellitic Vehicle:

This is the actual commercial Satellitic Vehicle along with the special aspects of large commercial Satellitic Vehicles and Satellitic Vehicles designed to carry cargo that extend beyond the characteristics defined for the Basic Satellitic Vehicle. This actor is responsible for the initiations of commercial Satellitic Vehicle, driver, and cargo safety status measured by on-board SAHS equipment.

3d. Customs Agency:

This border inspection agency performs the primary regulatory inspection function at an international Port of Entry. In Canada, this is generally the Canada Customs and Revenue

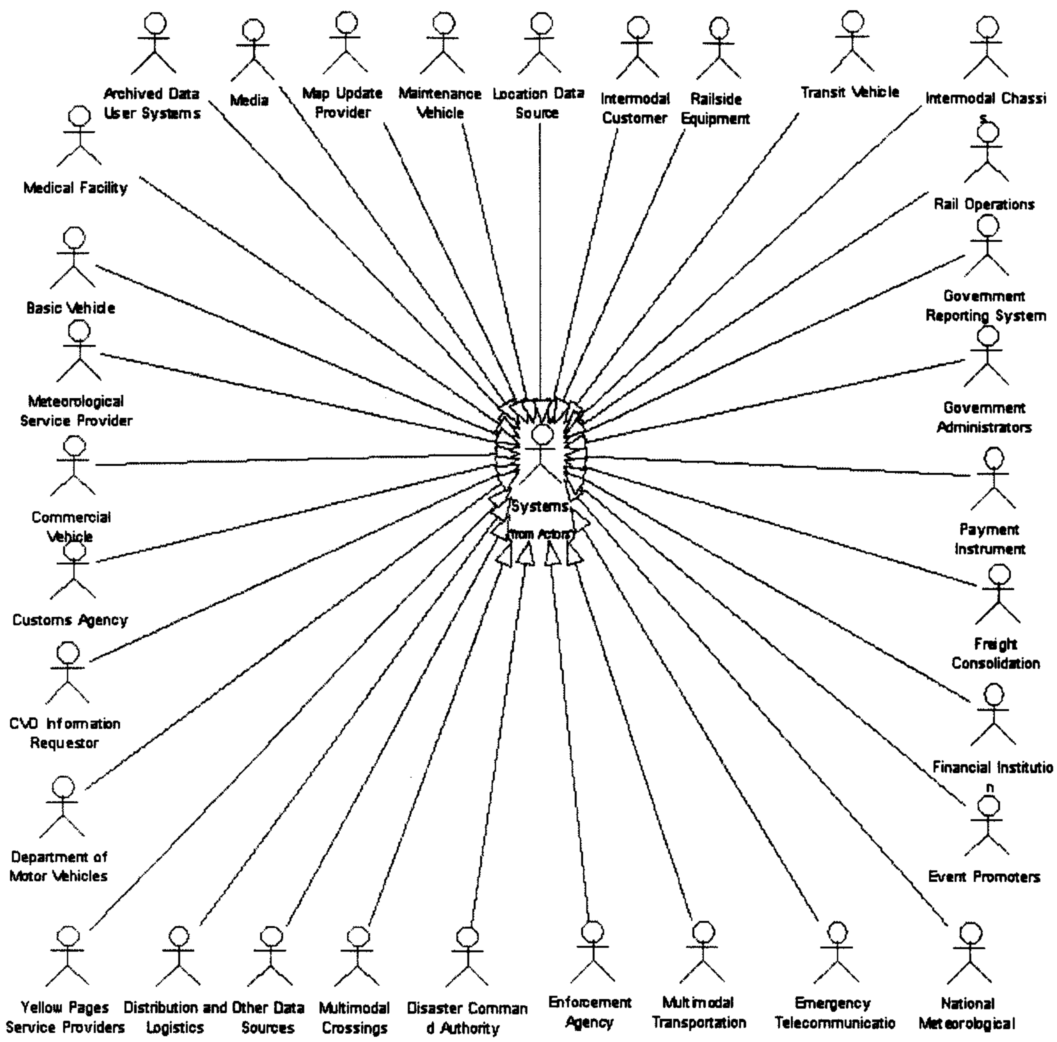

Figure 6.6 Object-oriented visualization of related Systems actors for SAHS in UML

Agency, but can also include Immigration, Agricultural agencies. Other inspection agencies can also be actors. This actor is responsible for the initiations of customs request for the electronic manifest associated with a container; requires proper authentication.

3e. Commercial Satellitic Vehicle Operator Information Requestor::

This actor represents any organization requesting CSVO information. This actor is responsible for the initiations of instructions to commercial Satellitic Vehicle management and/or information systems indicating which Satellitic Vehicles are to be allowed to pass and which are out of service. This actor can determine of which vehicles do not possess the sufficient credentials.

3f. Department of Motor Satellitic Vehicles:

This actor represents a specific (provincial) public organization responsible for registering Satellitic Vehicles, e.g., the Ministry of Transportation. This actor is responsible for the initiations of request supporting registration data based on license plate read during violations.

3g. Disaster Command Authority:

The Disaster Command Authority terminal represents the systems used by authorities that provide command-and-control leadership for coordinated disaster response. This actor is responsible for the initiations of information supporting the coordination of disaster and emergency response assets and activities.

3h. Distribution and Logistics Management Provider:

This actor represents systems that provide intermodal logistics support, and support for the efficient distribution of freight across transport systems and modes. This actor is responsible for the initiations of request to a container to provide its location; may require the requestor to authenticate their identity.

3i. Emergency Telecommunications System:

This actor represents the human entity that monitors all SAHS emergency requests, (including those from the E911 Operator) and sets up pre-defined responses to be executed by an emergency management system. This actor is responsible for the initiations of the notification of an incident including its nature, severity, and location.

3j. Enforcement Agency:

This terminator represents an external entity which receives reports of violations detected by various SAHS facilities, e.g. individual Satellitic Vehicle emissions, toll violations, speed or red light running violation, CVO violations, etc. This actor is responsible for the initiation of response from law enforcement agency to violations notification request.

3k. Event Promoters:

This actor represents external special events sponsors that have knowledge of events that may impact travel on roadways or other modal means. This actor is responsible for the initiations of plans for major events possibly impacting traffic.

3l. Financial Institution:

This actor represents the organization that handles all electronic fund transfer requests to enable the transfer of funds from the user of the service to the provider of the service. This actor is responsible for the initiations of response to transaction request, normally dealing with a request for payment.

3m. Freight Consolidation Station:

An intermediate point (usually an Intermodal terminal located at a port) prior to (or after) container-based shipping, where less-than-container load or less-than-truckload cargoes are consolidated into full container loads (or full containers are disbursed), for cost-effective intermodal shipping. This actor is responsible for the initiations of location within an intermodal facility that a container is to be received at or delivered to; and may include guidance as well as location.

3n. Government Administrators:

This terminal represents those public organizations responsible for regulating commercial Satellitic Vehicle operations, e.g., provincial commerce offices, and provincial agencies responsible for transportation, provincial finance departments, and Transport Canada. Regulatory Agencies are envisioned to be an integral part of the SAHS Commercial Satellitic Vehicle Operations (CSVO) as they will be directly involved with issuance of licenses, permits and other credentials for pre -clearance, provide database information to support most CSVO services, and will receive, distribute, and audit CSVO related taxes. This actor is responsible for the initiation of regulations imposed on Commercial Satellitic Vehicle Administration agencies including safety ratings, facility locations and credential fee structure.

3o. Government Reporting Systems:

This actor represents the system and associated personnel that prepare the inputs to support the various local, provincial, and federal government transportation data reporting requirements using data collected by SAHS systems. This actor is responsible for the initiation of the acknowledgement of satisfactory receipt of information used as input to government data systems or a report identifying problems or issues with the data submittal.

3p. Intermodal Chassis:

This actor represents the chassis, which is the frame on wheels that an intermodal container is secured to for roadway transport by a truck. This actor is responsible for the initiation of measurement from chassis systems as brakes, tires, and fasteners. This actor indicates operational readiness of a chassis.

3q. Intermodal Customer:

This actor represents the originator of an order to move or the final recipient of a cargo shipment. This actor is responsible for the initiations of the notice confirming the arrival and transfer of control of a container at a container handling facility, for example, an intermodal terminal, request from a shipper for services to handle shipping of a container or freight load, and request from consignor or consignee for the current location and transit status of a freight shipment.

3r. Location Data Source:

This actor represents an external entity, which provides accurate position information. This actor is responsible for the initiation of the information, of the geographical position of a traveler or a Satellitic Vehicle.

3s. Maintenance Vehicle:

This actor represents a specialized form of the basic Satellitic Vehicle used by maintenance fleets. This actor is responsible for the initiation of maintenance Satellitic Vehicle status to be measured by on-board SAHS equipment.

3t. Map Update Provider:

This actor represents a third-party developer and provider of digitized map databases used to support SAHS services. This actor is responsible for the initiation of map updates, which could include new underlying static or real-time maps or map layer(s) update.

3u. Media:

This actor represents the information systems that provide traffic reports, travel weather conditions, and other transportation-related news services to the travelling public. The information is broadcasted through radio, television, and other media. This actor is responsible for the initiation of requests from the media for current traffic information and incident information. Such information is is collected by the media through a variety of mechanisms (e.g., radio station call-in programs, and air surveillance).

3v.Medical Facility:

This actor represents medical facilities such as hospitals, trauma centres, field emergency treatment facilities and any other locations capable of receiving injured persons and providing emergency care. This actor is responsible for the initiation of specific care capabilities and available space in a medical facility.

3w. Meteorological Service Provider:

This actor represents the providers of value-added sector specific meteorological services. This actor is responsible for the initiation of prediction of weather information, customized to a particular users needs and region.

3x. Multimodal Crossings:

This actor represents the control equipment that interfaces to a non-road based transportation systems at an interference crossing with the roadway. This actor is responsible for the initiation of prediction of indication of operational status and pending requests for right-of-way from equipment supporting the non-highway mode at multimodal crossings.

3y. Multimodal Transportation Service Provider:

This terminator provides the interface through which transportation service providers can exchange data with SAHS. This actor is responsible for the initiation of operational information from alternate passenger transportation modes, including air, rail transit, taxis, and ferries. Content may include a catalog of available information, the actual information to be archived, and associated metadata that describe the archived information.

3z. National Meteorological Service:

This actor provides weather, hydrologic, and climate information and warnings of hazardous weather including thunderstorms, flooding, hurricanes, tornadoes, winter weather, tsunamis, and climate events. This actor is responsible for the initiation of accumulation forecast and current weather data (e.g., temperature, pressure, wind speed, wind direction, humidity, precipitation, visibility, light conditions, etc.).

3aa. Other Data Sources:

This actor represents the myriad systems and databases containing data not generated from subsystems and terminators represented in the national SAHS architecture that can provide predefined data sets to the SAHS archive. This actor is responsible for the initiation of data extracted from other data sources. A wide range of SAHS and non-SAHS data and associated metadata may be provided.

3ab. Payment Instrument:

This actor represents the entity that enables the actual transfer of funds from the user of a service to its provider. This actor is responsible for the initiation of information passing from a payment instrument (e.g., smart card) to a payment device to provide electronic payment of some kind (e.g., toll, parking, fare) by traveller. In most cases, the payment can be linked to credit accounts.

3ac. Rail Operations:

This is roughly the railroad equivalent of a highway traffic management center. This actor is responsible for the initiation of real-time notification of railway-related incident or advisory.

3ad. Railside Equipment:

This actor represents train interface equipment (usually) maintained and operated by the railroad and (usually) physically located at or near a grade crossing. This actor is responsible for the initiation of information for a train approaching a highway-rail intersection that may include direction and allow calculation of approximate arrival time and closure duration.

3ae. Transit Satellitic Vehicle:

This actor represents a specialized form of the basic Satellitic Vehicle used by transit service providers. This actor is responsible for the initiation of transit Satellitic Vehicle status measured by on-board SAHS equipment.

3af. Yellow Pages Service Providers:

This actor represents the individual organizations that provide any services oriented towards the traveller. This actor is responsible for the initiation of information supplied by a service provider (e.g., a hotel or restaurant), and identifies the service provider, and provides details of the service offering. This flow covers initial registration of a service provider and subsequent submittal of new information and status updates so that data currency is maintained.

4 Other Systems:

Other related system actors are illustrated in figure 6.7.

4a. Other Archives:

This actor represents distributed archived data systems or centres whose data can be accessed and shared with a local archive. This actor is responsible for the initiation of catalog data, metadata, published data, and other information exchanged between archives to support data synchronization and satisfy user data requests.

4b. Other Commercial Satellitic Vehicle Administration Subsystem:

This actor is intended to provide a source and destination for SAHS data flows between peer (e.g. inter-regional) commercial Satellitic Vehicle administration functions. This actor is responsible for the initiation of tax and credential fee information exchanged between cooperating commercial Satellitic Vehicle administration offices and instructions to commercial Satellitic Vehicle managing and/or information systems indicating which Satellitic Vehicles are to be allowed to pass and which are out of service or do not meet the necessary requirements.

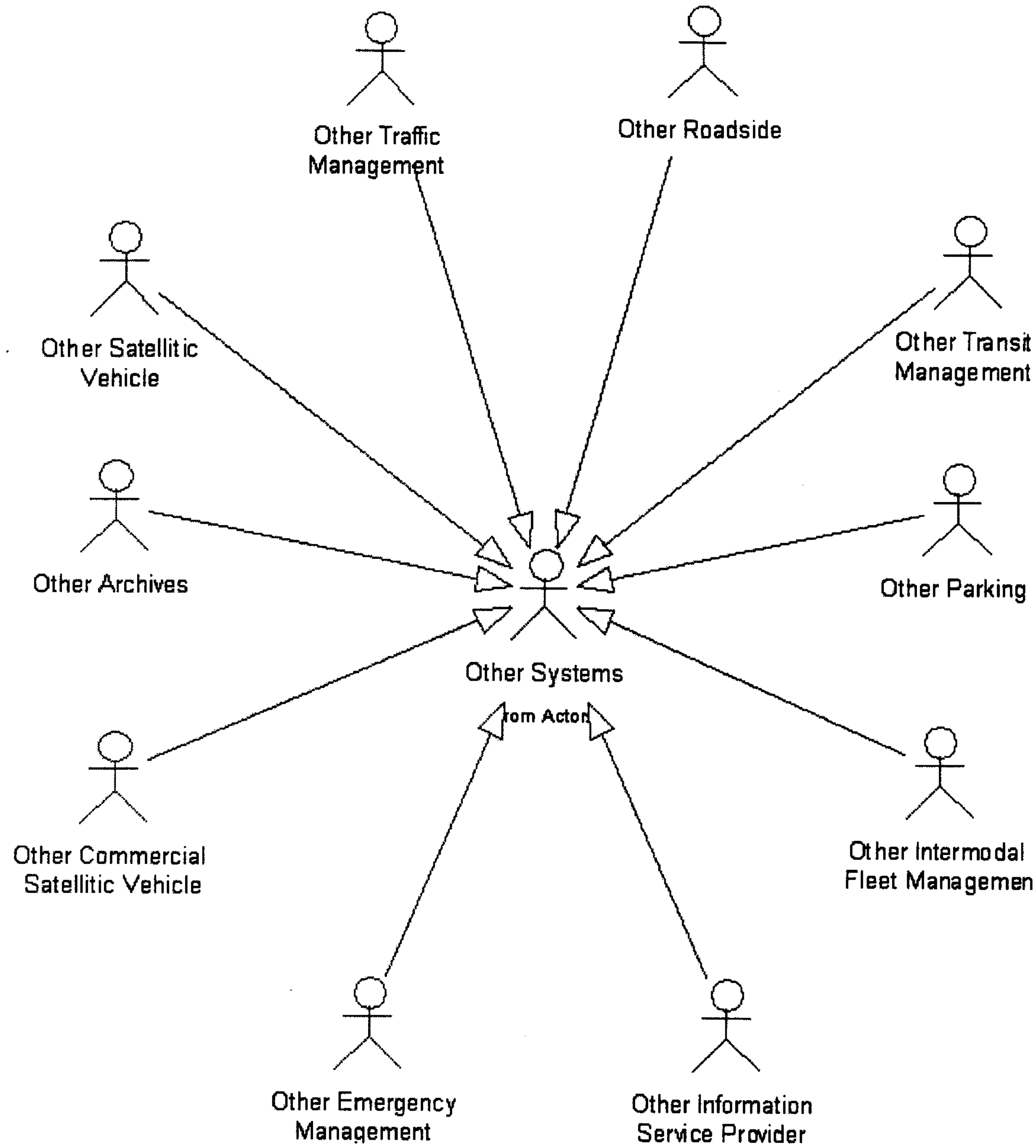

Figure 6.7 Object Oriented visualization of
related Other System actors foe SAHS in UML

4c. Other Emergency Management:

Representing other emergency management centres, systems or subsystems, this terminal provides a source and destination for SAHS data flows between various communications centres operated by public safety agencies as well as centres operated by other allied agencies and private companies that participate in coordinated management of highway-related incidents. This actor is responsible for the initiations of reports of an identified incident including incident location, type, severity and other information necessary to initiate an appropriate incident response. It is also responsible for the start of incident response procedures, resource coordination, and current incident response status that are shared between allied response agencies to support a coordinated response to incidents. This flow also coordinates a positive hand off of responsibility for all or part of an incident response between agencies.

4d. Other Information Service Provider:

This actor represents other distinct information service providers, this actor is intended to provide a source and destination for SAHS data flows between peer information and service provider functions. This actor is responsible for the initiation of coordination and exchange of transportation information between centres.

4e. Other Intermodal Fleet Management System:

This actor represents another intermodal fleet management system that manages systems for individual fleets of intermodal transport companies. This actor is responsible for the initiation of exchange of information between different intermodal freight management centers regarding cargo or intermodal container movement and information regarding whether a container has been released.

4f. Other Parking:

This actor represents another parking facility, system or subsystem. This terminator provides a source and destination for information that may be exchanged between peer parking systems. This actor is responsible for the initiation of information that enables parking management activities to be coordinated between different parking operators or systems in a region.

4g. Other Roadside:

This actor represents another roadside element or system. This actor provides a source and destination for information that may be exchanged between peer roadside elements. This actor is responsible for the initiation of control data for roadside devices that is exchanged between roadside devices or from a maintenance management system to the roadside.

4h. Other Traffic Management:

This actor represents another traffic management center, system or subsystem. This actor is intended to provide a source and destination for SAHS data flows between peer traffic management functions. This actor is responsible for the initiation of traffic information exchanged between TMC's. Normally, traffic information would include incidents, congestion data, traffic data, signal timing plans, and real-time signal control information.

4i. Other Transit Management:

This actor represents another transit management center, system or subsystem. This actor is intended to provide a source and destination for SAHS data flows between peer transit management functions. This actor is responsible for the initiation of co-ordination information between local/regional transit organizations including scheduling, on-time information, connection co-ordination, and ridership.

4j. Other Satellitic Vehicle:

This actor represents a Satellitic Vehicle (of any 4 Satellitic Vehicle types) that is neighboring the basic Satellitic Vehicle, where the basic Satellitic Vehicle is equipped to support vehicle-to-vehicle communications and coordination. This actor is responsible for the initiations of any type of advanced vehicle-to-vehicle communications.

6.5.1.2 Created Context diagram

The detailed description of the various types of actors that may interact within the system and their roles help us to identify the system boundary. The boundary in a complex system as SAHS is a difficult and challenging task. The developers must have a clear idea about the boundaries of their authority with regard to making changes prior implementing the improvements and then check the impact of these changes. Figure 6.8 illustrates the SAHS context.

6.5.1.3 Created Use Cases and diagrams

In order to prepare the use case view of a SAHS, the requirements were subject to analysis and visualization in UML with the view to extract the possible required capability of the system. Figures 6.9 to 6.18 are the decomposition of system use cases. The degree of use case decomposition is related to what we what intend from this partial development. The goal, as we stated earlier, is to capture the communication problems solution to the ITSC architecture. The numbers that are assigned to the use cases are the same numbers pertaining to the related functional requirements given in the ITSC architecture documentation. A is related to Backbone Communication System class function, and B is related to the cases where the Backup Communication system Class is taking over as an integrated substitute to the backbone Communication System Class.

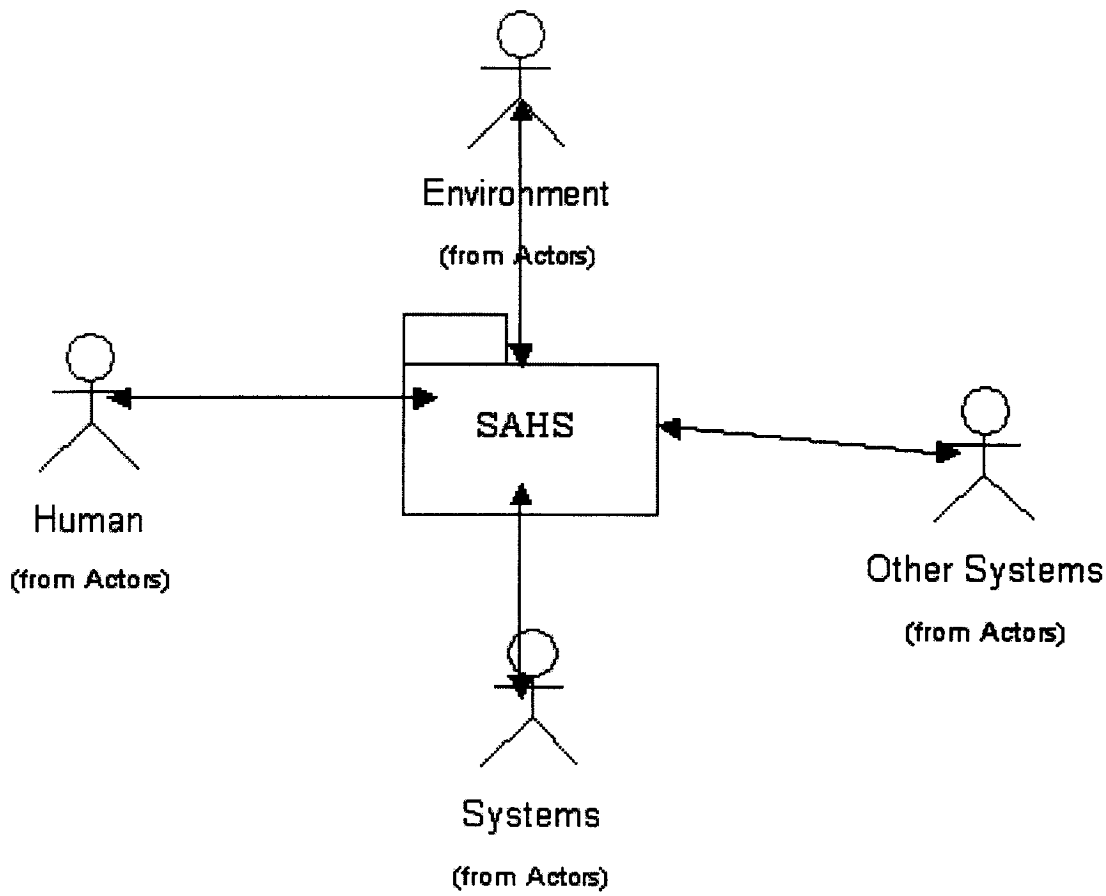

Figure 6.8 Object-oriented visualization of SAHS's context in UML

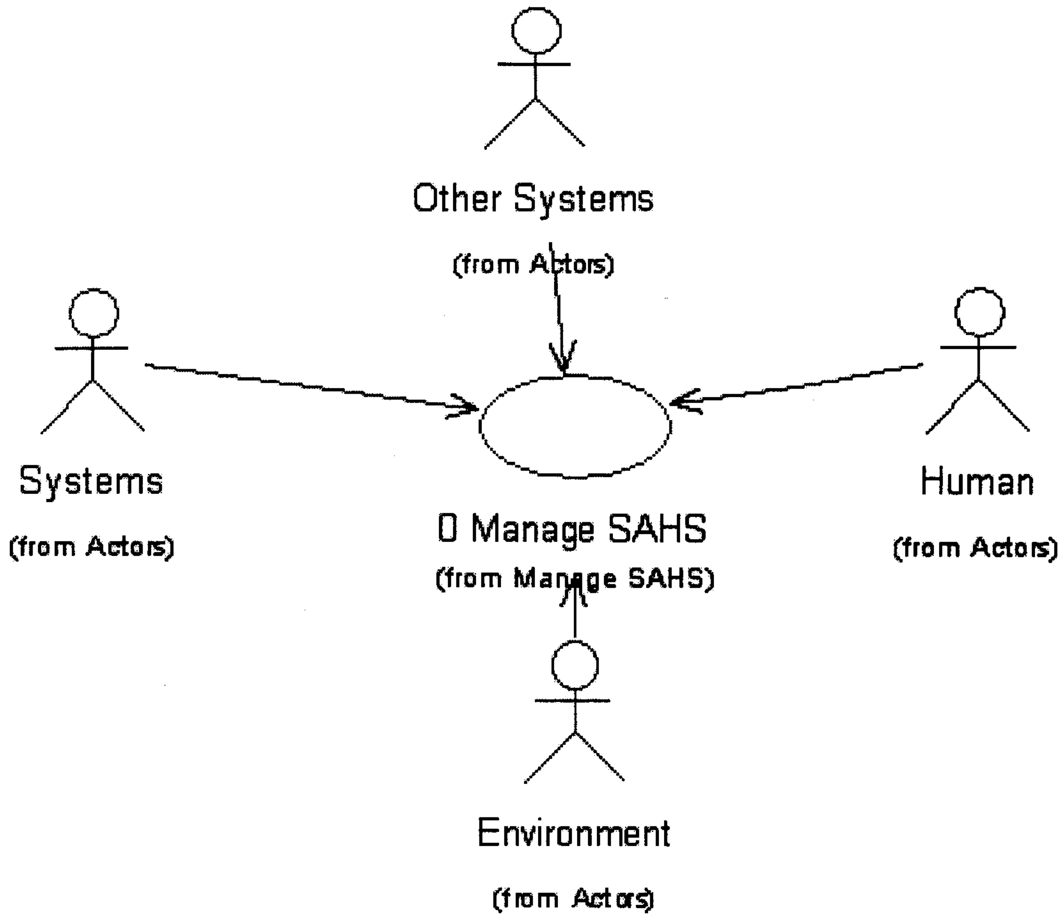

Figure 6.9 Object-oriented Visualization of 0
Manage SAHS use Case diagram in UML

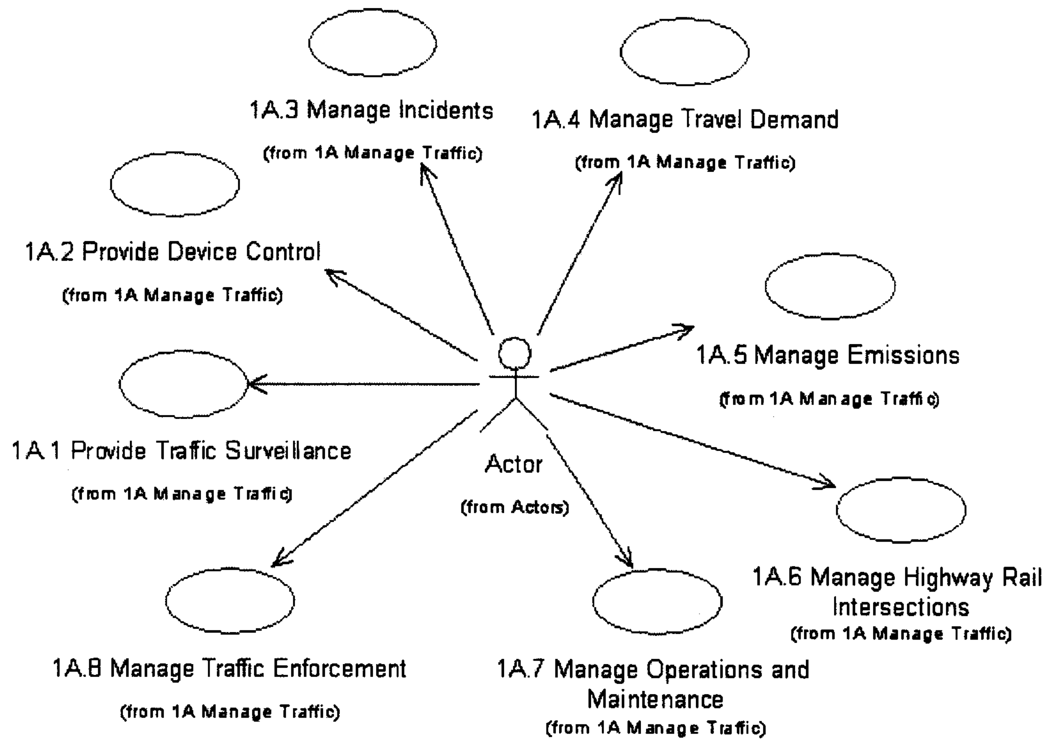

Figure 6.10 Object-oriented visualization of related 1A Manage Traffic use cases for SAHS in UML

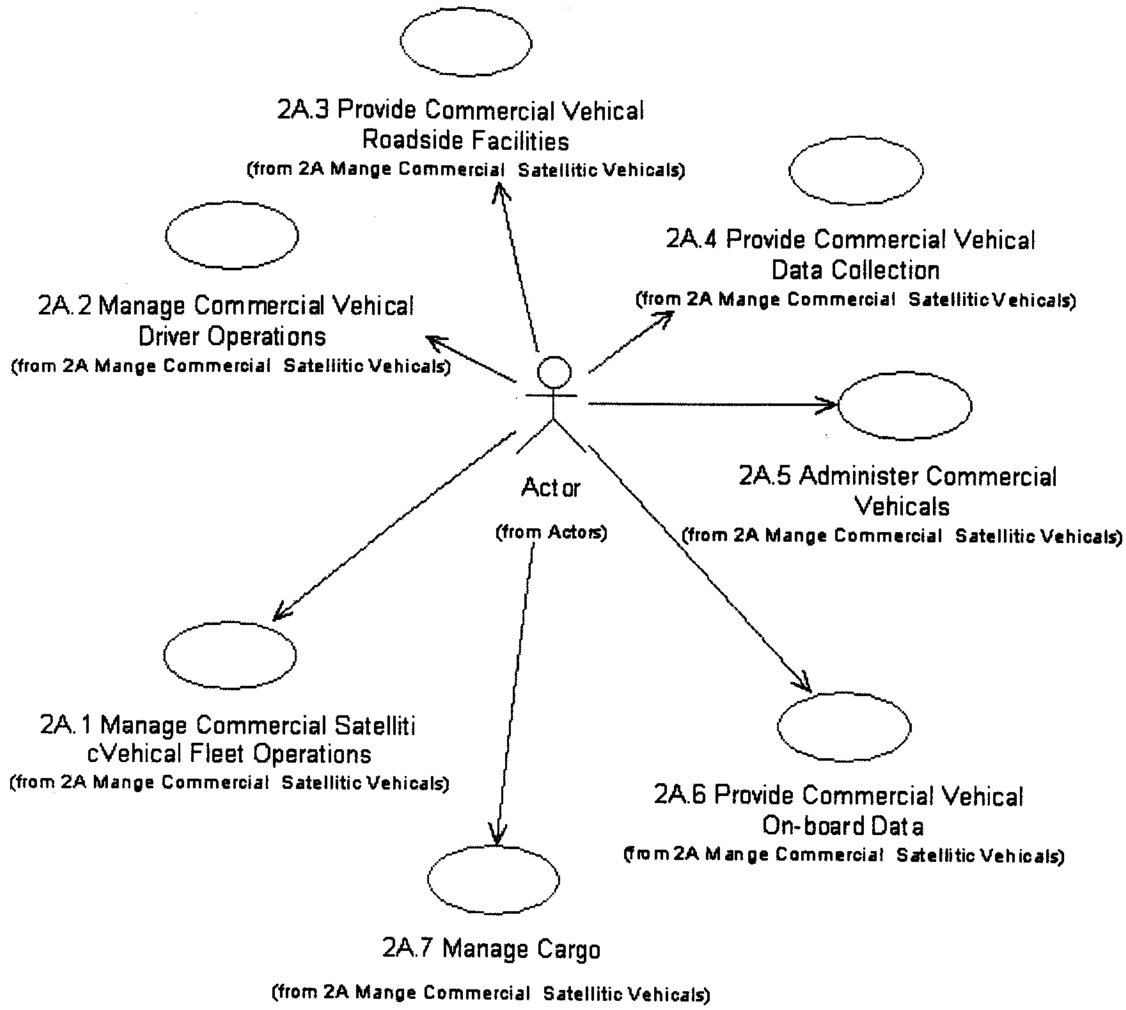

Figure 6.11 Object-oriented visualization of related 2A Manage Commercial Satellitic Vehicle use cases for SAHS in UML

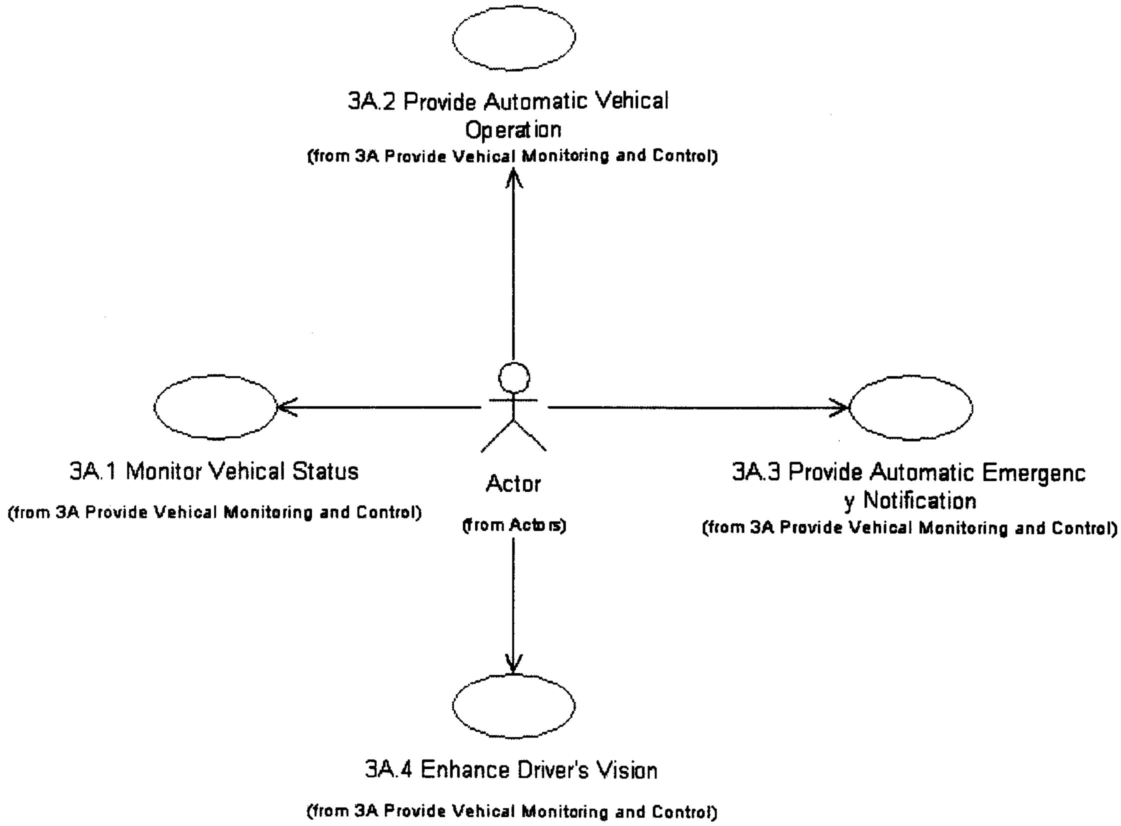

Figure 6.12 Object-oriented visualization of related 3A Provide Vehicle Monitoring and Control uses cases for SAHS in UML

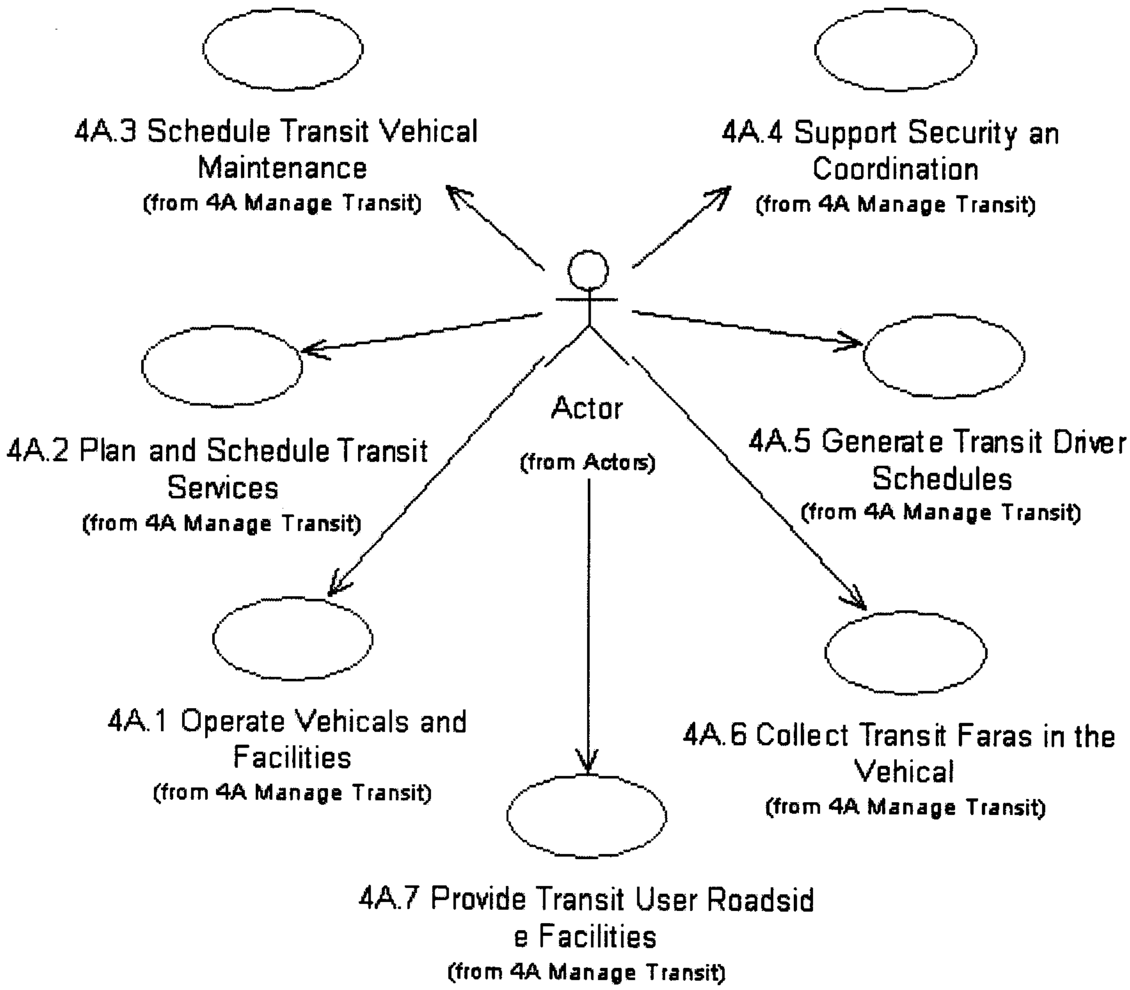

Figure 6.13 Object-oriented visualization of related 4A Manage Transit use cases for SAHS in UML

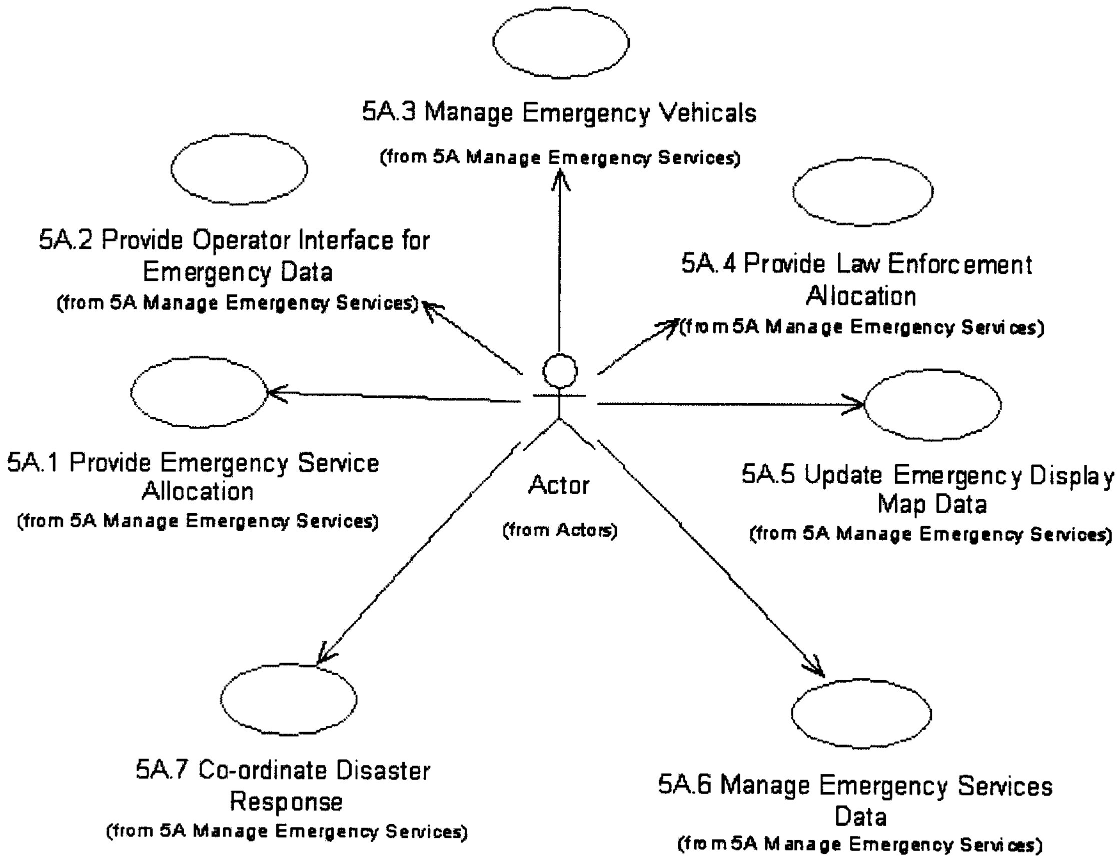

Figure 6.14 Object-oriented visualization of related 5A Manage Emergency Services use cases for SAHS in UML

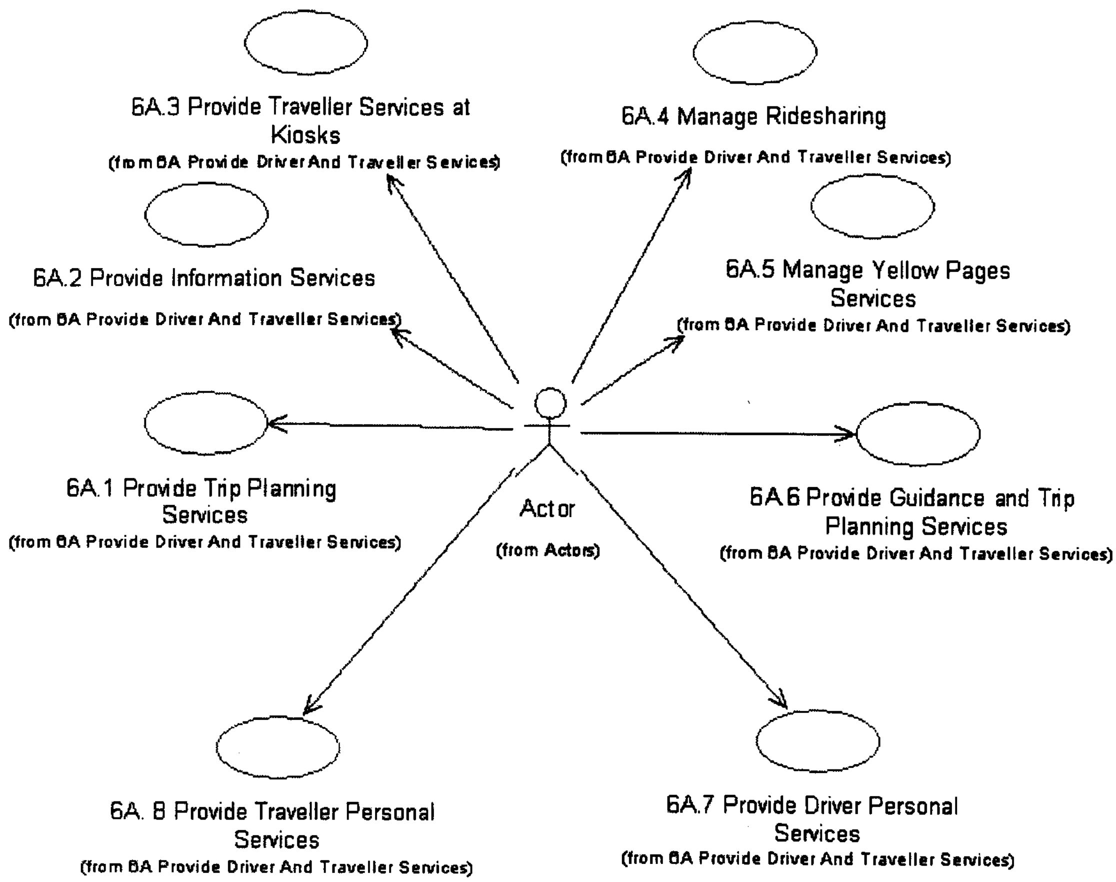

Figure 6.15 Object-oriented visualization of related 6A Provide Driver and Traveller Services use cases for SAHS in UML

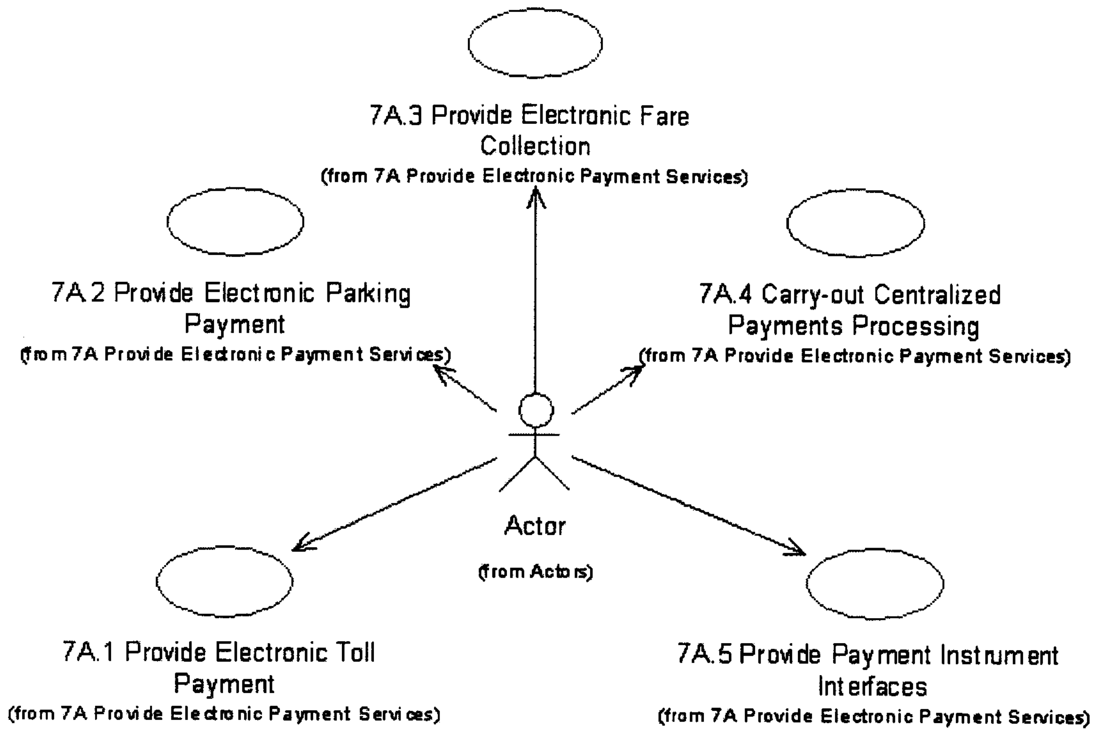

Figure 6.16 Object-oriented visualization of related 7A Provide Electronic Payment Services use cases for SAHS in UML

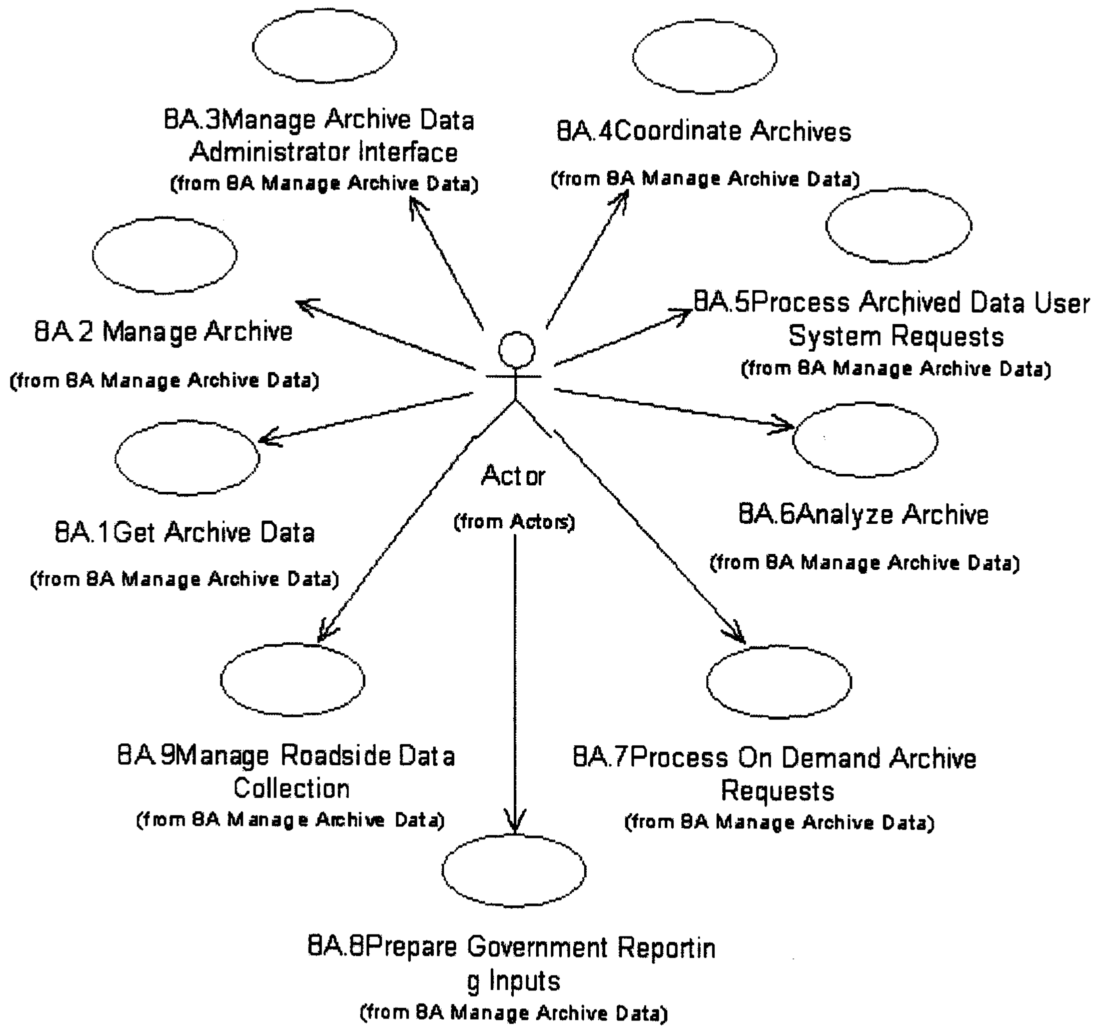

Figure 6.17 Object-oriented visualization of related 8A Manage Archive Data use case for SAHS in UML

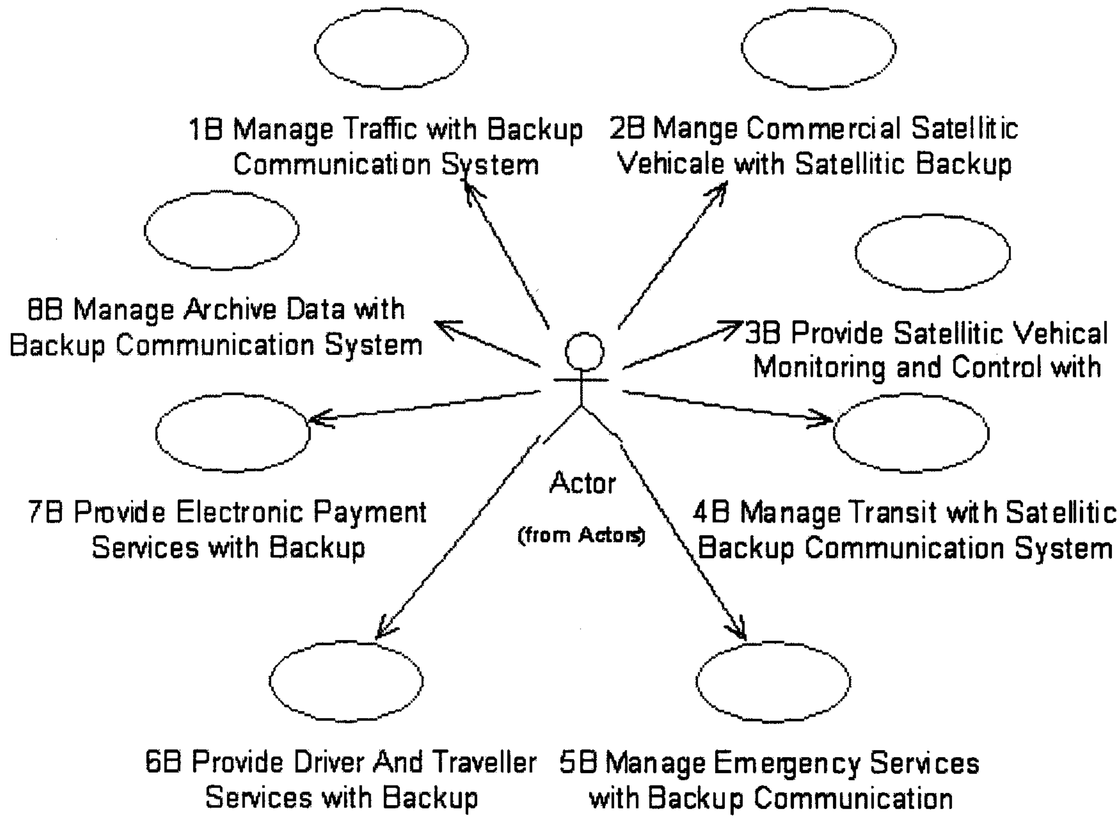

Figure 6.18 Object-oriented visualization of related 0B Manage SAHS use cases in UML

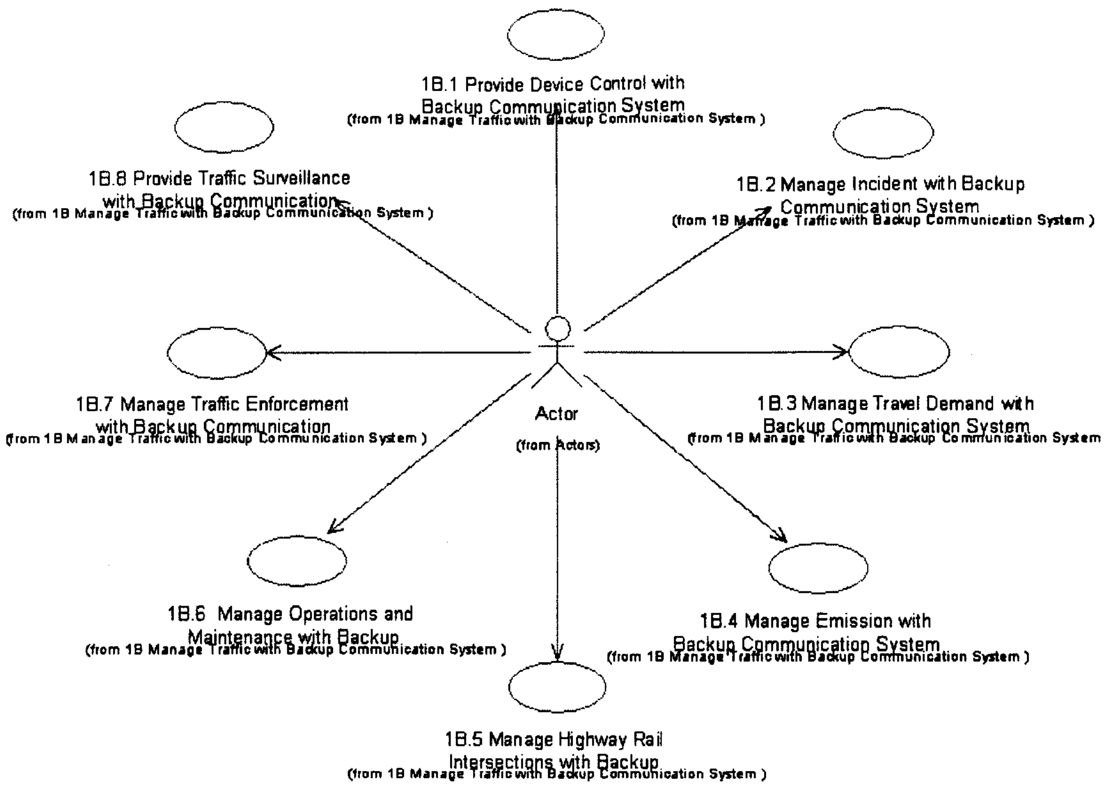

Figure 6.19 Object-oriented visualization of related 1B Manage Traffic use cases for SAHS in UML

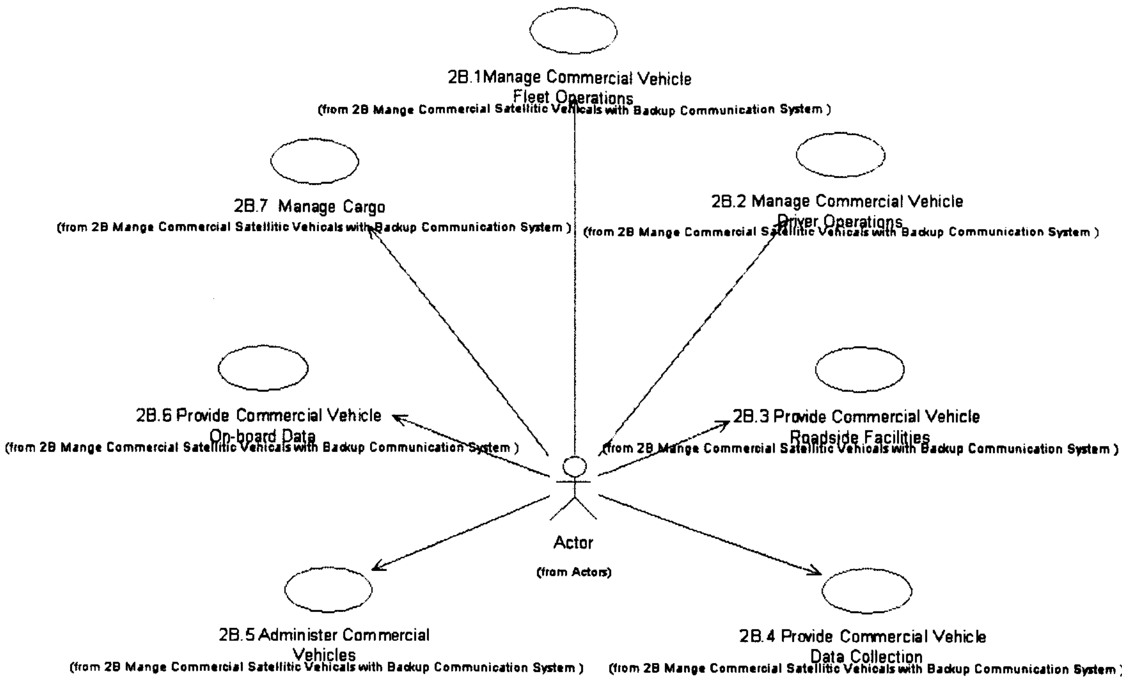

Figure 6.20 Object-oriented visualization of related 2B Manage Commercial Satellitic Vehicle for SAHS in UML

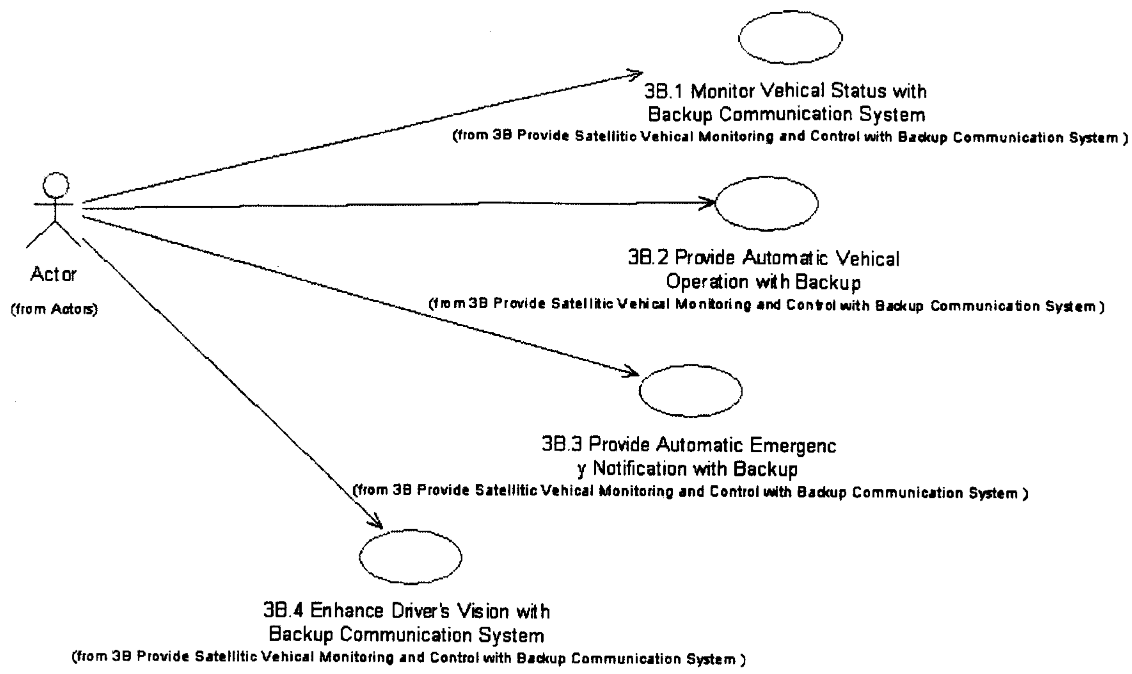

Figure 6.21 Object-oriented visualization of related 3B Provide Satellitic Vehicle Monitoring and Control use cases for SAHS in UML

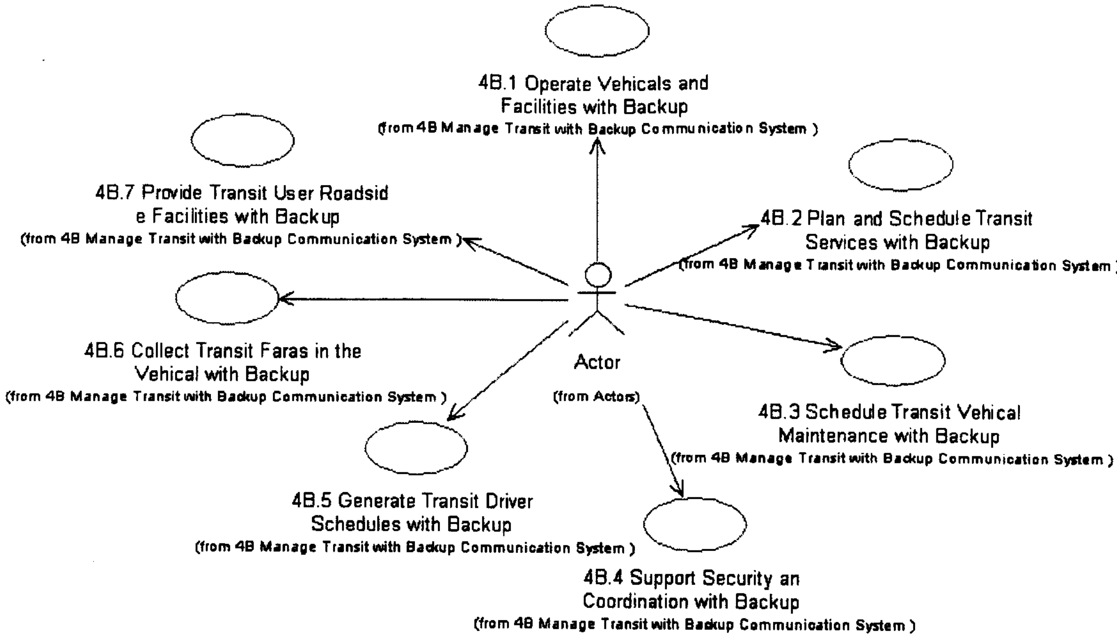

Figure 6.22 Object-oriented visualization of related 4B Manage Transit use cases for SAHS in UML

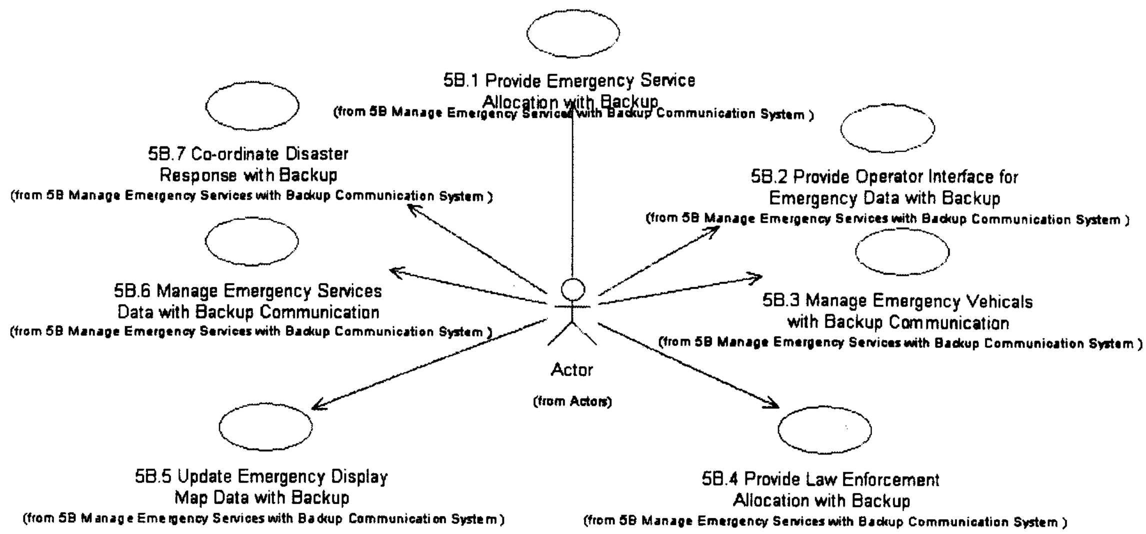

Figure 6.23 Object-oriented visualization of related 5B Manage Emergency Services for SAHS in UML

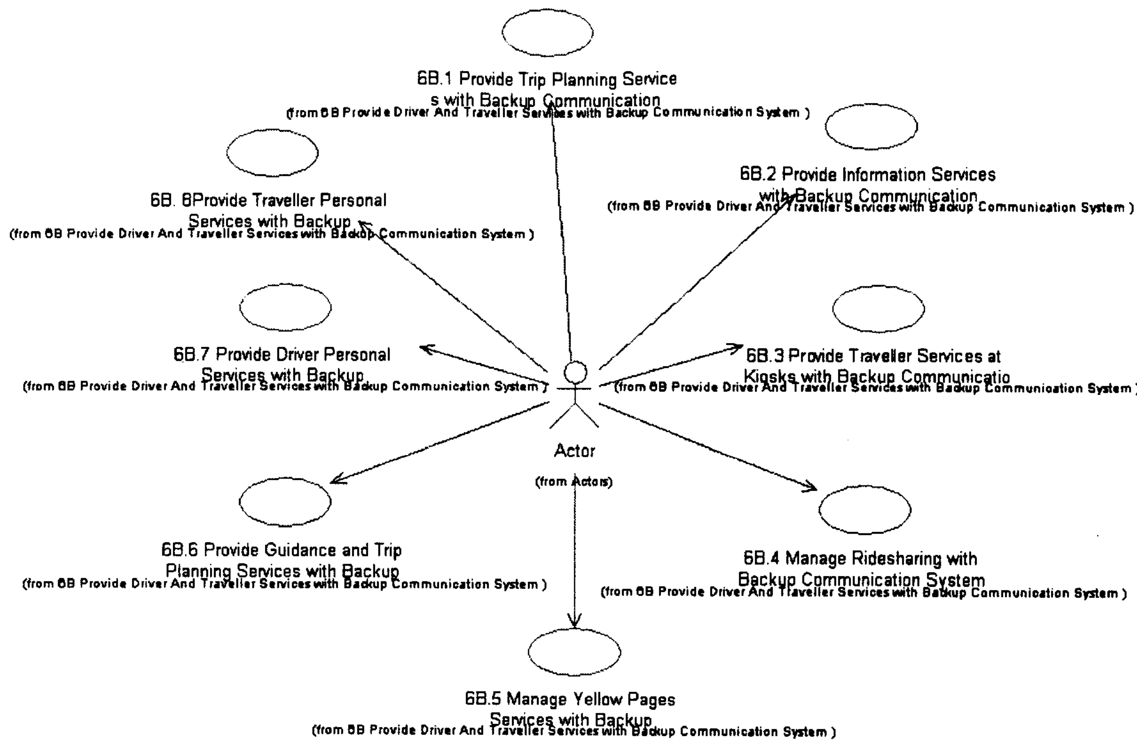

Figure 6.24 Object oriented visualization of related 6B Provide Traveller Personal Services for SAHS in UML

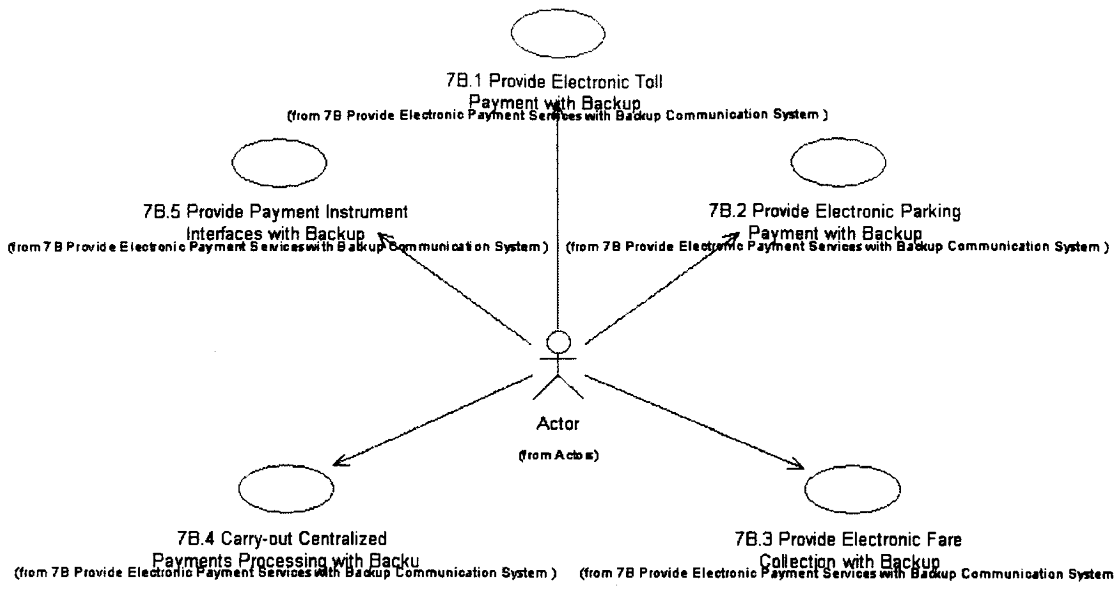

Figure 6.25 Object-oriented visualization of related 7B Provide Electronic Payment Services for SAHS in UML

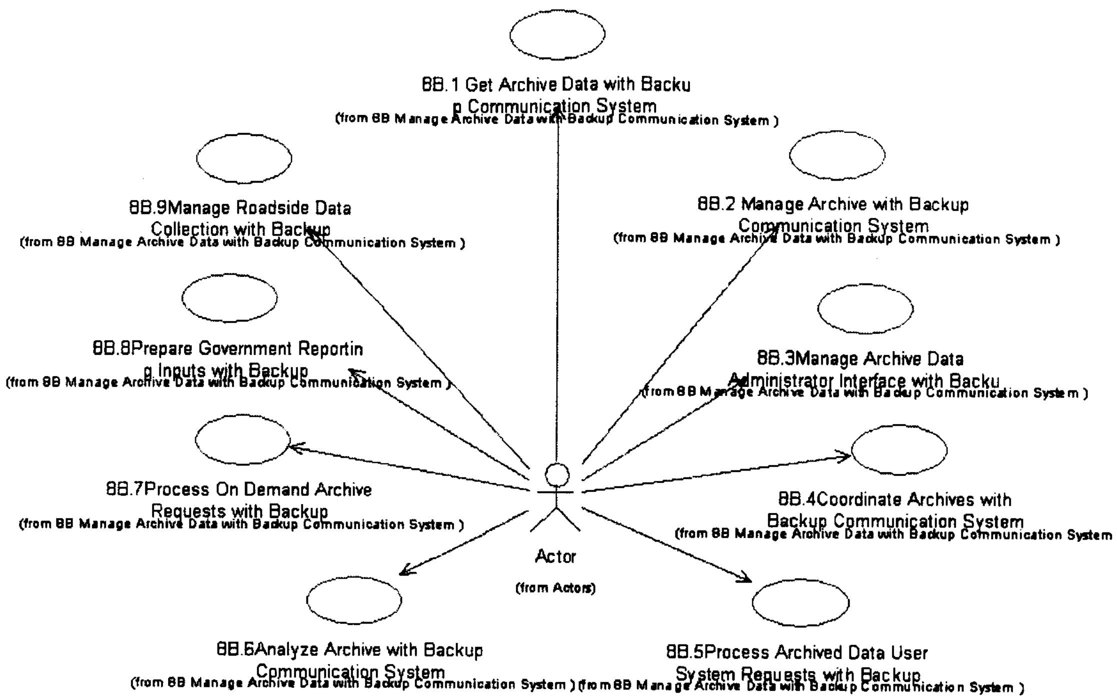

Figure 6.26 Object-oriented visualization of related 8B Manage Emergency Services for SAHS in UML

6.5.2 Classes in the SAHS Conceptual Project

The requirement analysis of SAHS targeting the capture of a common description of related objects. These objects have common attributes, operations, relations, and semantics. The class that represents related groups of within SAHS was identified using object-oriented techniques. The initial list of classes with a general description is illustrated in Table 6.2.

6.5.3 Visualization Class Diagram of the SAHS Conceptual Project

Class diagram of the SAHS that represent the Physical Architecture overview of the system are illustrated in figure 6.27.

6.5.4 Visualization Scenarios of SAHS Conceptual Project

The scenarios of the SAHS are captured using Sequence Diagrams. The figures 6.28 to figure 7.37 are presentation of such visualization. The rest of SAHS sequences Diagrams can be found in APPENDIX E.

6.5.5 Visualizing attributes and functions of SAHS Conceptual Project Classes

The attributes and functions of SAHS conceptual project classes illustrated in Figure 6.38 to Figure 6.100, the rest of SAHS conceptual project classes can be found in APPENDIX E.

Table 6.2 SAHS list of classes with a general description

Class name		Class General Description
Satellitic Automated Highway System	1	Manage traffic information and control.
Centres	2	Location of the centres subsystems.
Wayside	3	Location of the Wayside subsystems.
Travellers	4	Location of the Travellres subsystems.
SatelliticVehicles	5	Location of the Vehicles subsystems.
Backbone Communication	6	Manage traffic information and control communication
Backup Communication	7	Manage Backup traffic information and control communication
Archived Data Management	8	Manage traffic information and control. Archived Data
Commercial Satellitic Vehicle Administration	9	Manage traffic information and control. Commercial Vehicle Administration.
Emissions Management	10	Manage traffic information and control. Emission Management
Fleet and Freight Management	12	Manage traffic information and control. Fleet and Freight Management
Information Service Provider	13	Manages traffic information and control. Information Service Provider
Maintenance Management	14	Manages traffic information and control Maintenance Management.
Toll Administration	15	Manages traffic information and control Toll Administration.
Traffic Management	16	Manages traffic information and control Traffic Management.
Transit Management	17	Manages traffic information and control Transit Management.
Emergency Management	18	Manages traffic information and control. Emergency Management.
Commercial Satellitic Vehicle Check	19	Manages traffic information and control Commercial Vehicle Check.
Intermodal Terminal	20	Manages traffic information and control Intermodal Terminal.
Parking Management	21	Manages traffic information and control Parking Management.
Roadway	22	Manages traffic information and control Roadway.
Toll Collection	23	Manages traffic information and control. Toll Collection.
Personal Information Access	24	Manages traffic information and control Personal Information Access.
Remote Traveller Support Subsystem	25	Manages traffic information and control Remote Traveller Support Subsystem.
Commercial Satellitic Vehicle	26	Manages traffic information and control Commercial Vehicle.
Emergency Satellitic Vehicle	27	Manages traffic information and control.
Intermodal Container	28	Manages traffic information and control Intermodal Container.
Maintenance Satellitic Vehicle	29	Manages traffic information and control. Maintenance Vehicle.
Transit Satellitic Vehicle	30	Manages traffic information and control Transit Vehicle
Satellitic Vehicle	31	Manages traffic information and control. Vehicle.

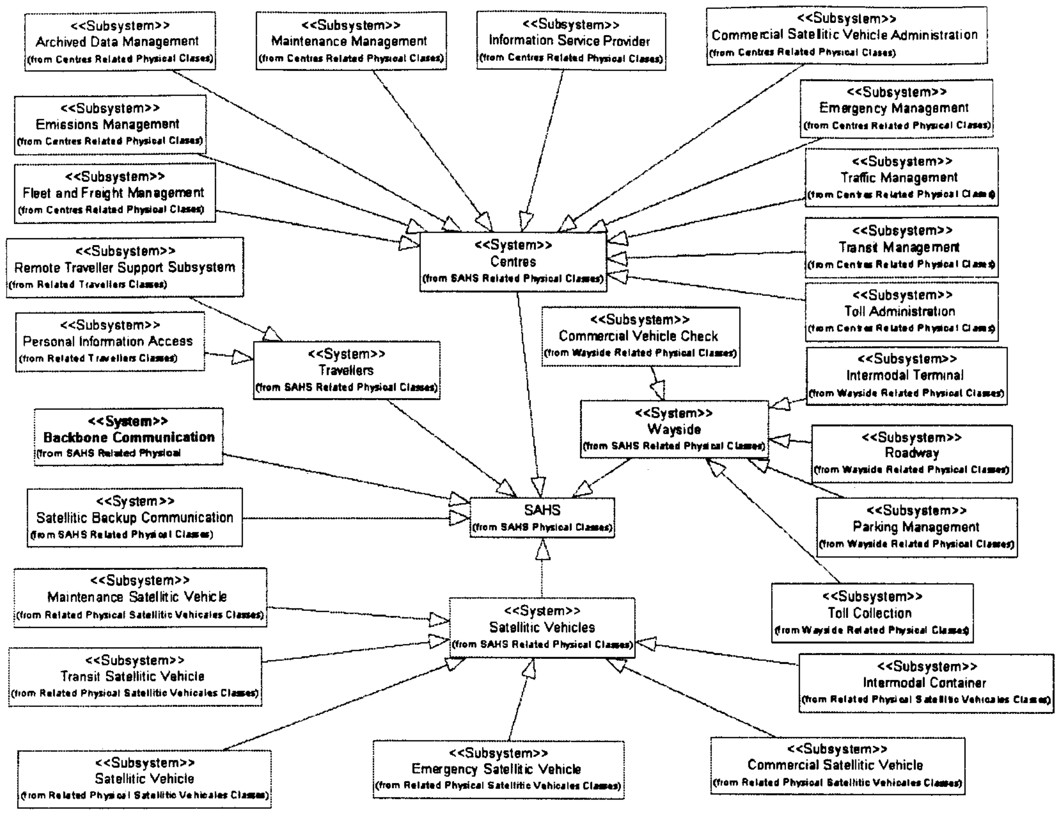

Figure 6.27 Object-oriented visualization of Physical Architecture overview of SAHS in UML

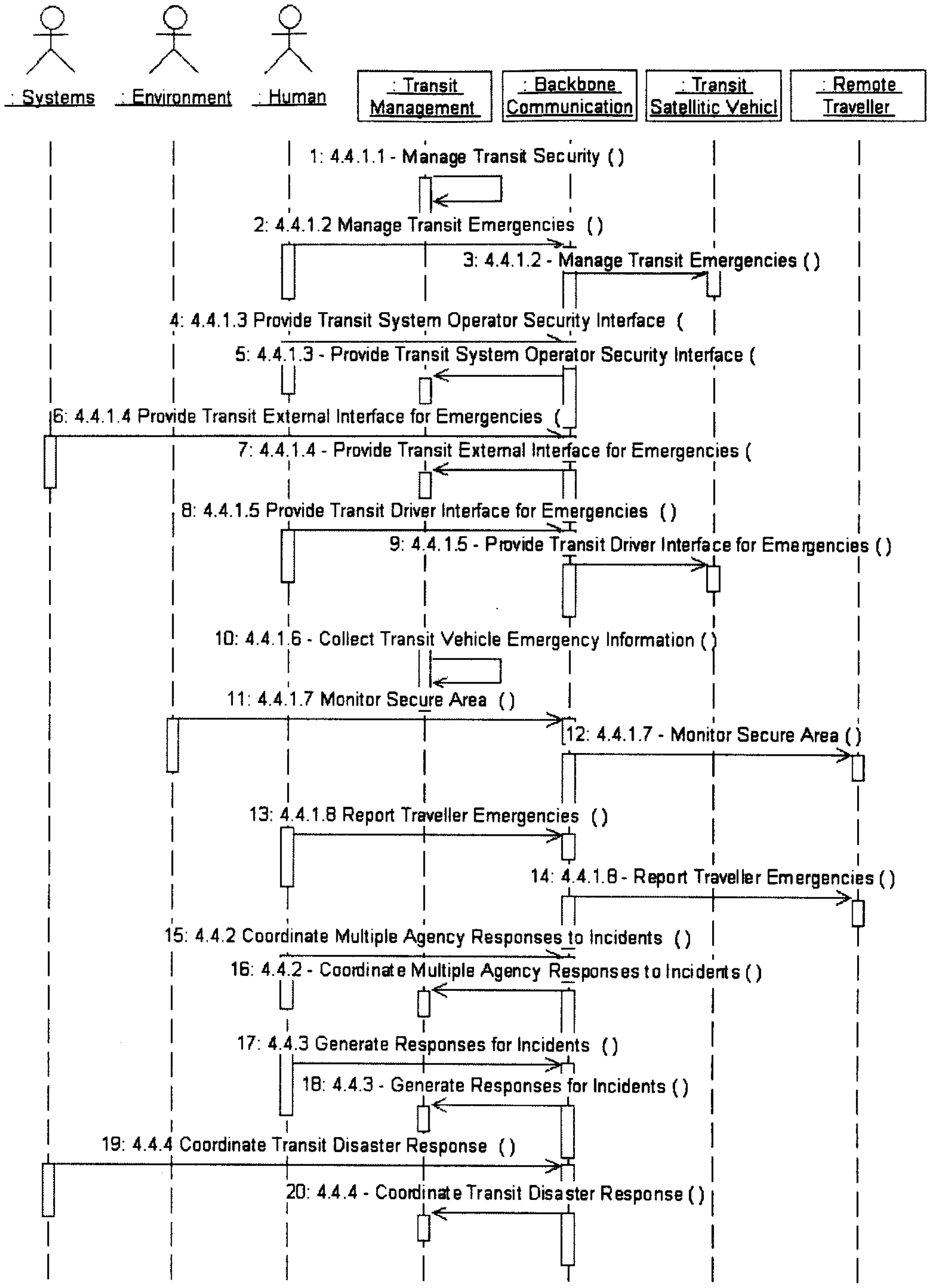

Figure 6.28 Object-oriented visualization of Sequence Diagram 4A.4 Support Security and Coordination for SAHS in UML

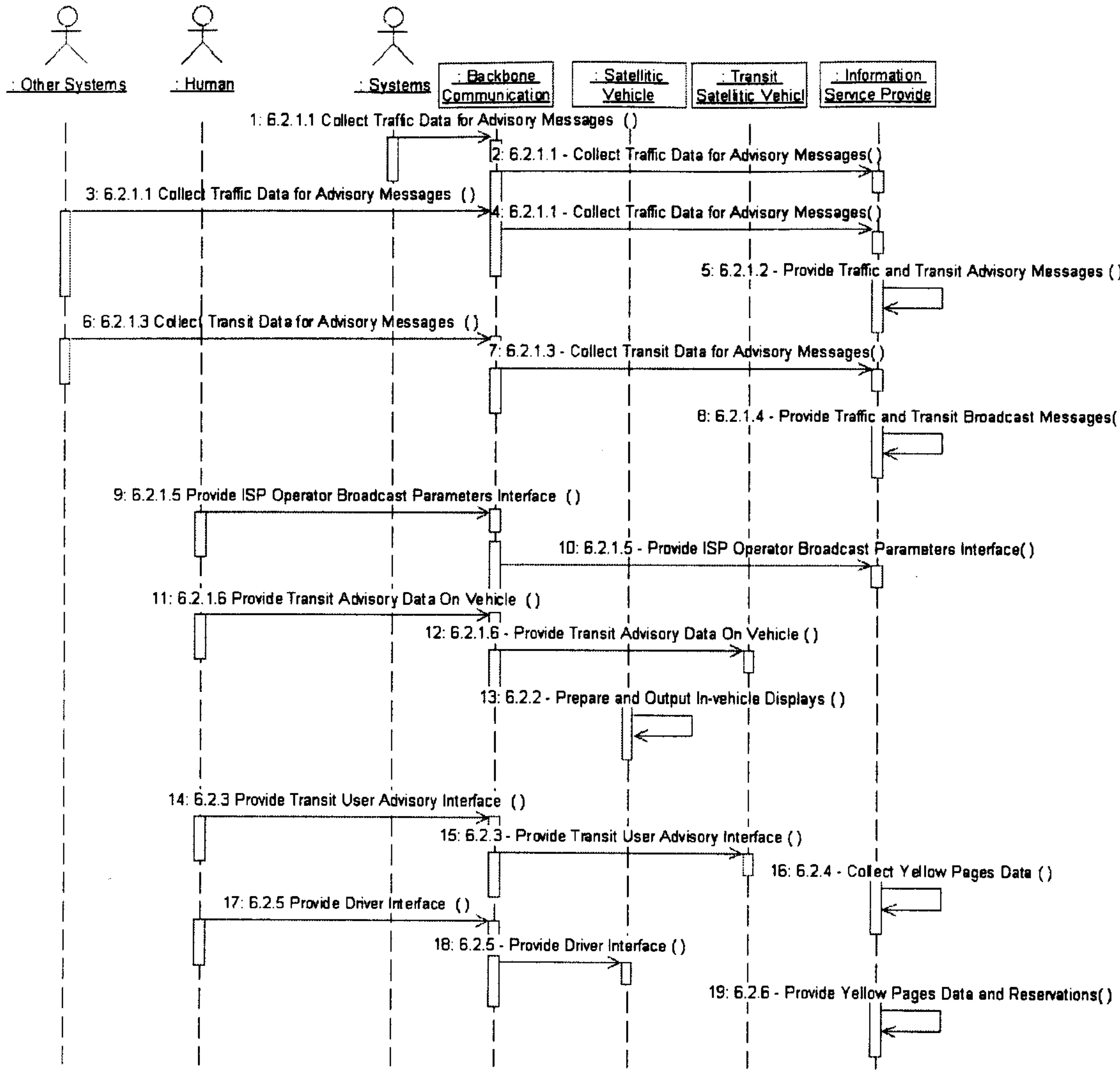

Figure 6.29 Object-oriented visualization of Sequence Diagram 6A.2 Provide Information Services for SAHS in UML

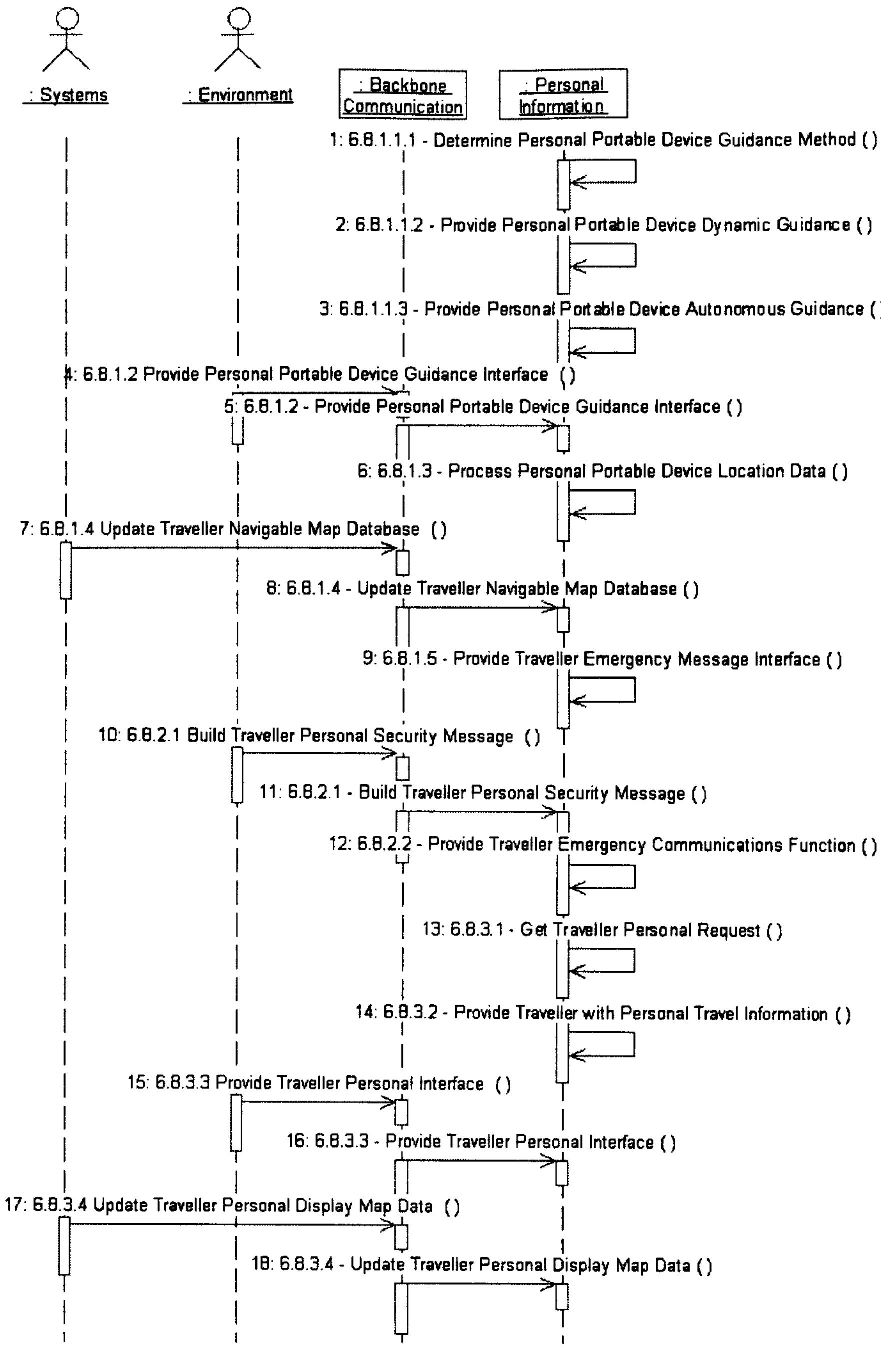

Figure 6.30 Object-oriented visualization of Sequence Diagram 6A.8 Provide Traveller Personal Services for SAHS in UML

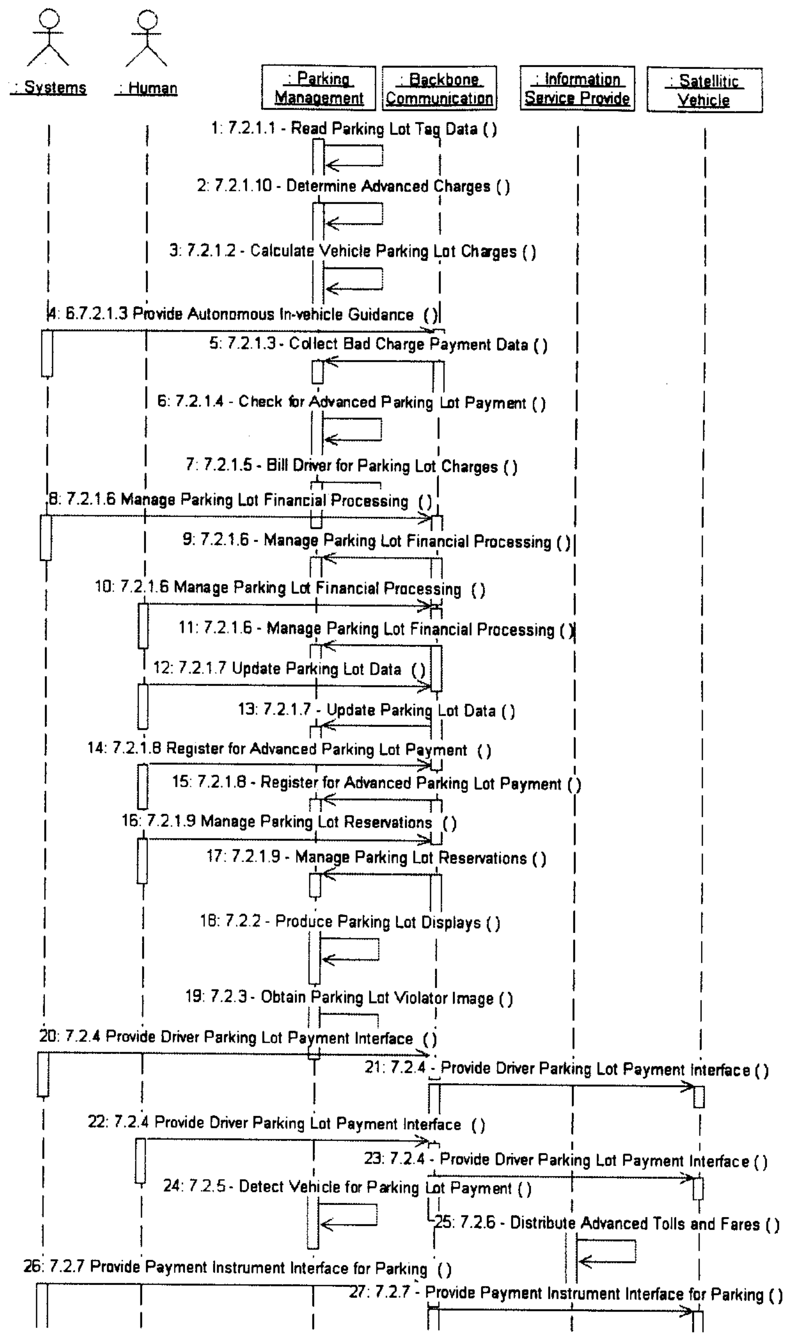

Figure 6.31 Object-oriented visualization of Sequence Diagram 7A.2 Provide Electronic Parking Payment for SAHS in UML

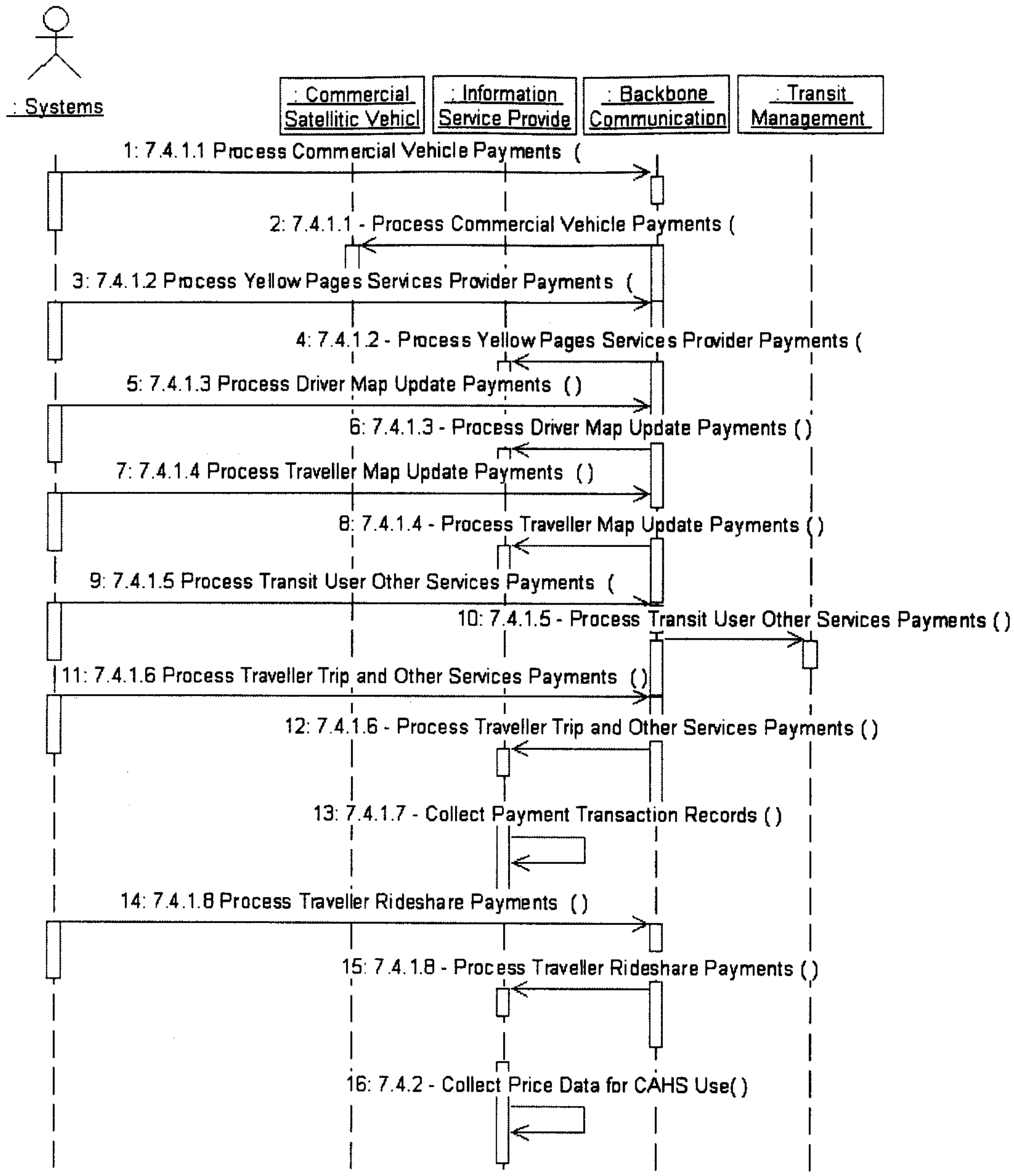

Figure 6.32 Object-oriented visualization of Sequence Diagram 7A.4 Carry-out Centralized Payments Processing for SAHS in UML

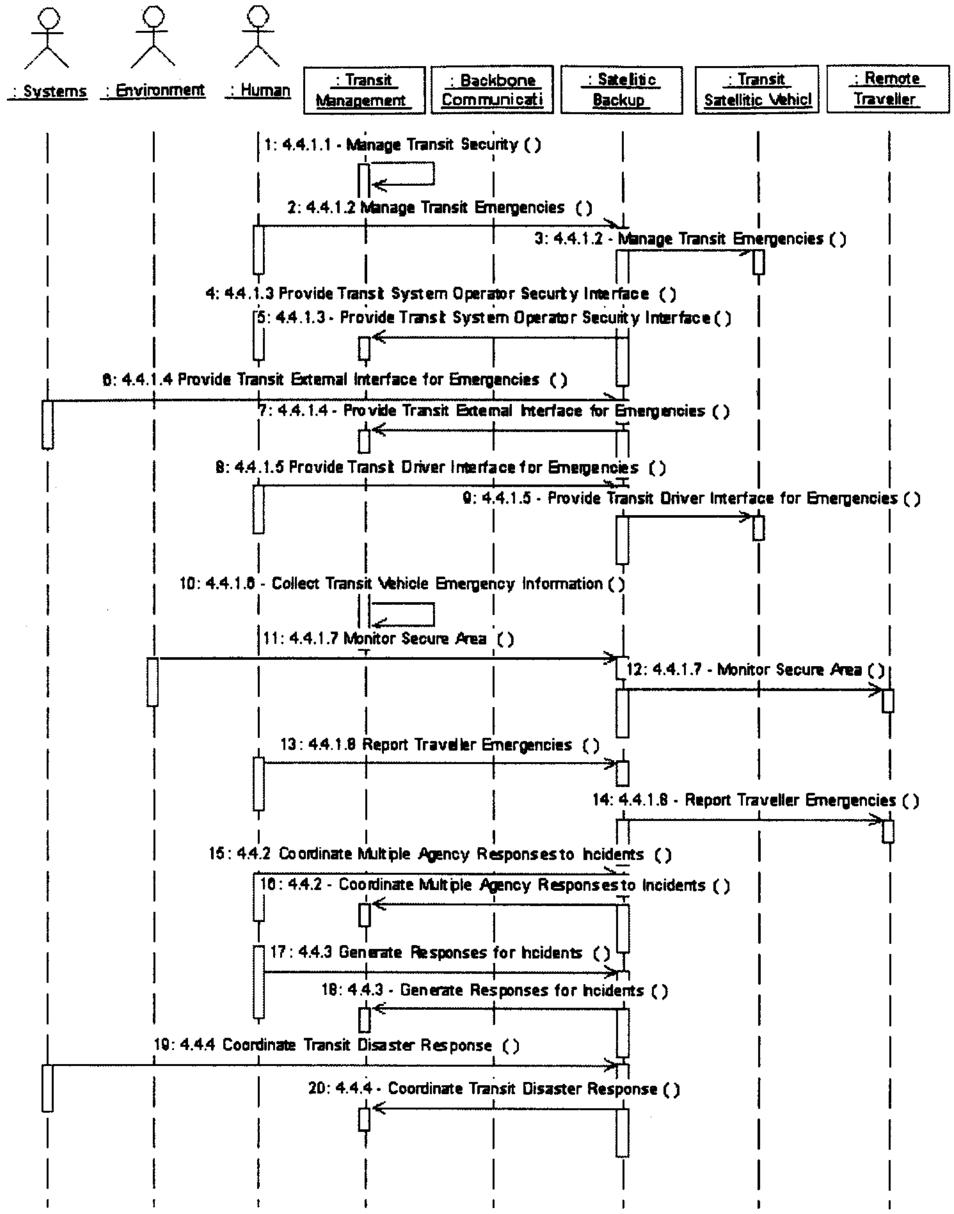

Figure 6.33 Object-oriented visualization of Sequence Diagram 4B.4 Support Security and Coordination for SAHS in UML

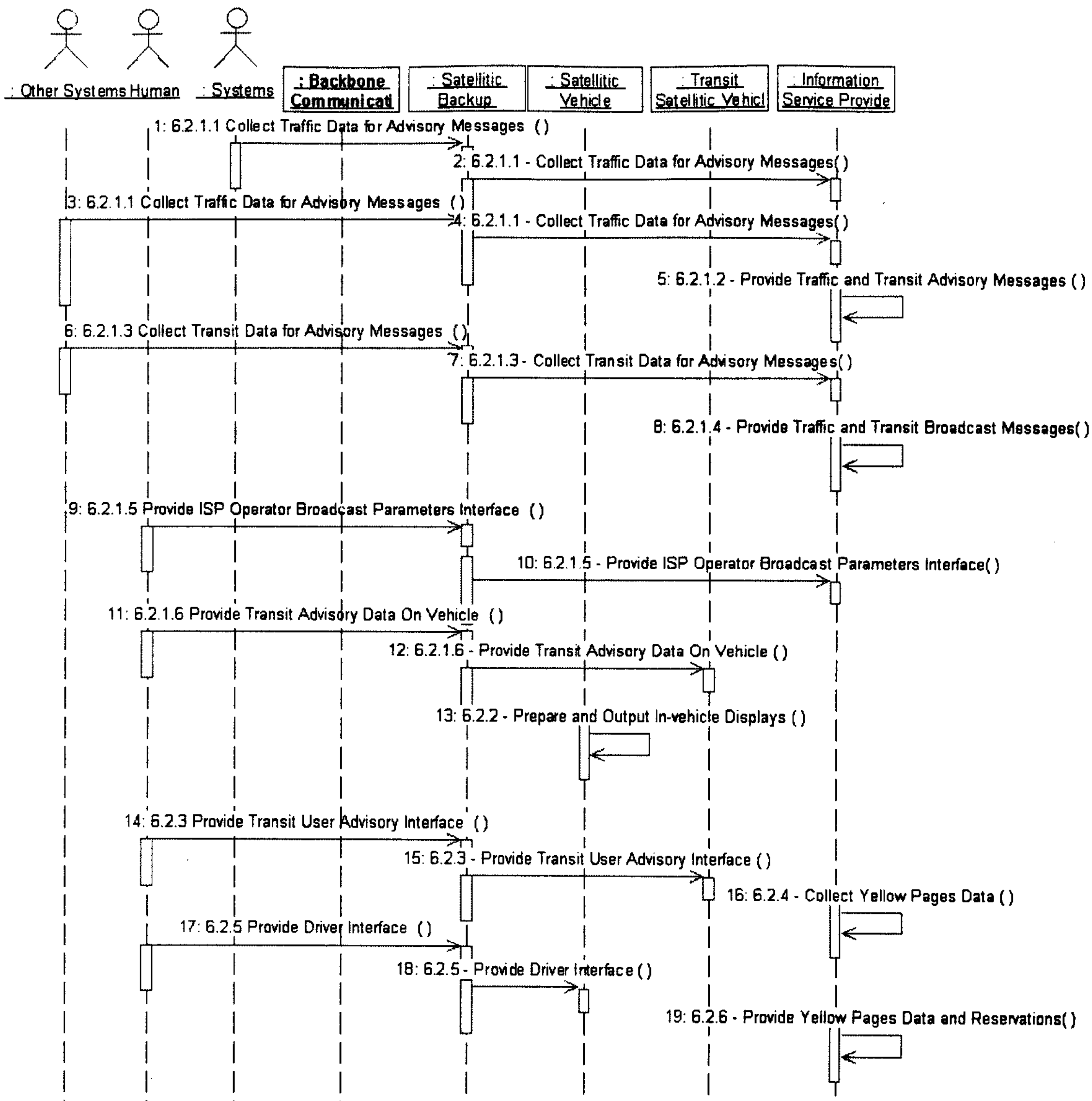

Figure 6.34 Object-oriented visualization of Sequence Diagram 6B.2 Provide Information Services for SAHS in UML

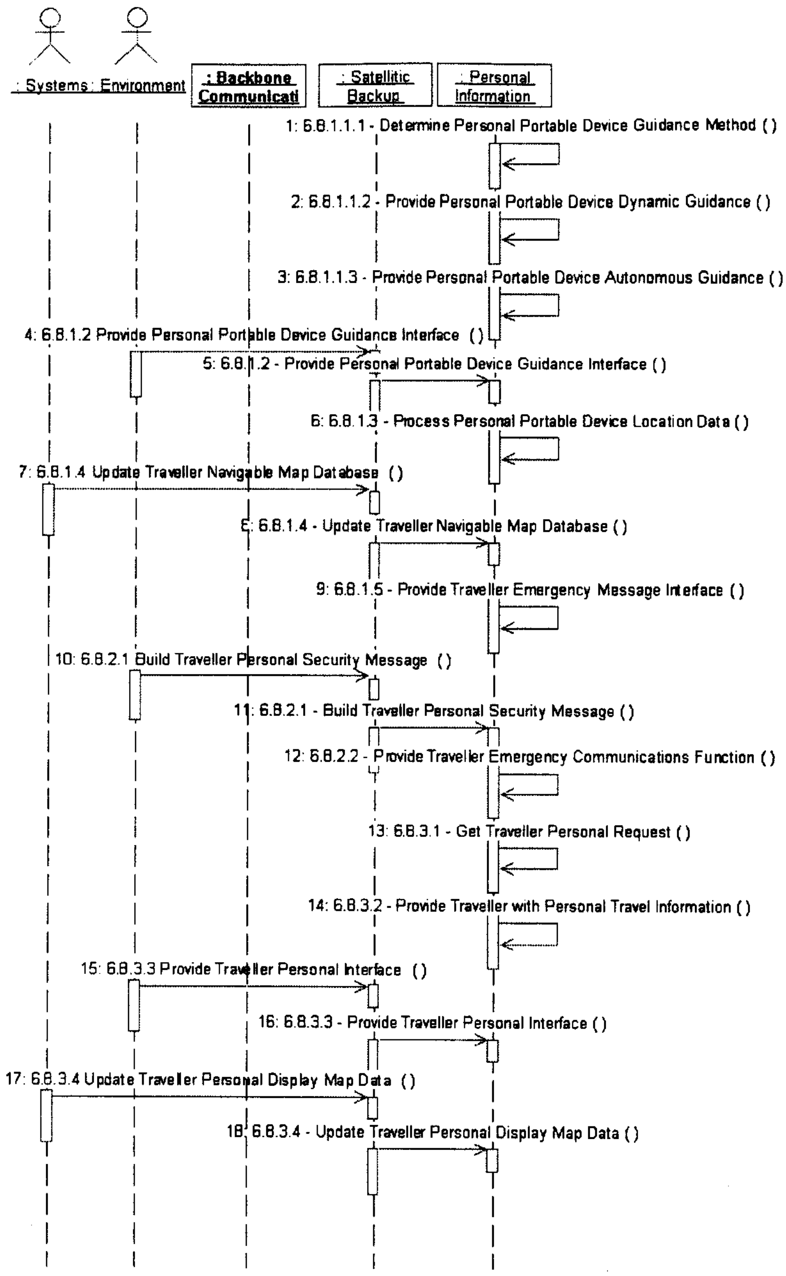

Figure 6.35 Object-oriented visualization of Sequence Diagram 6B.8 Provide Traveller Personal Services for SAHS in UML

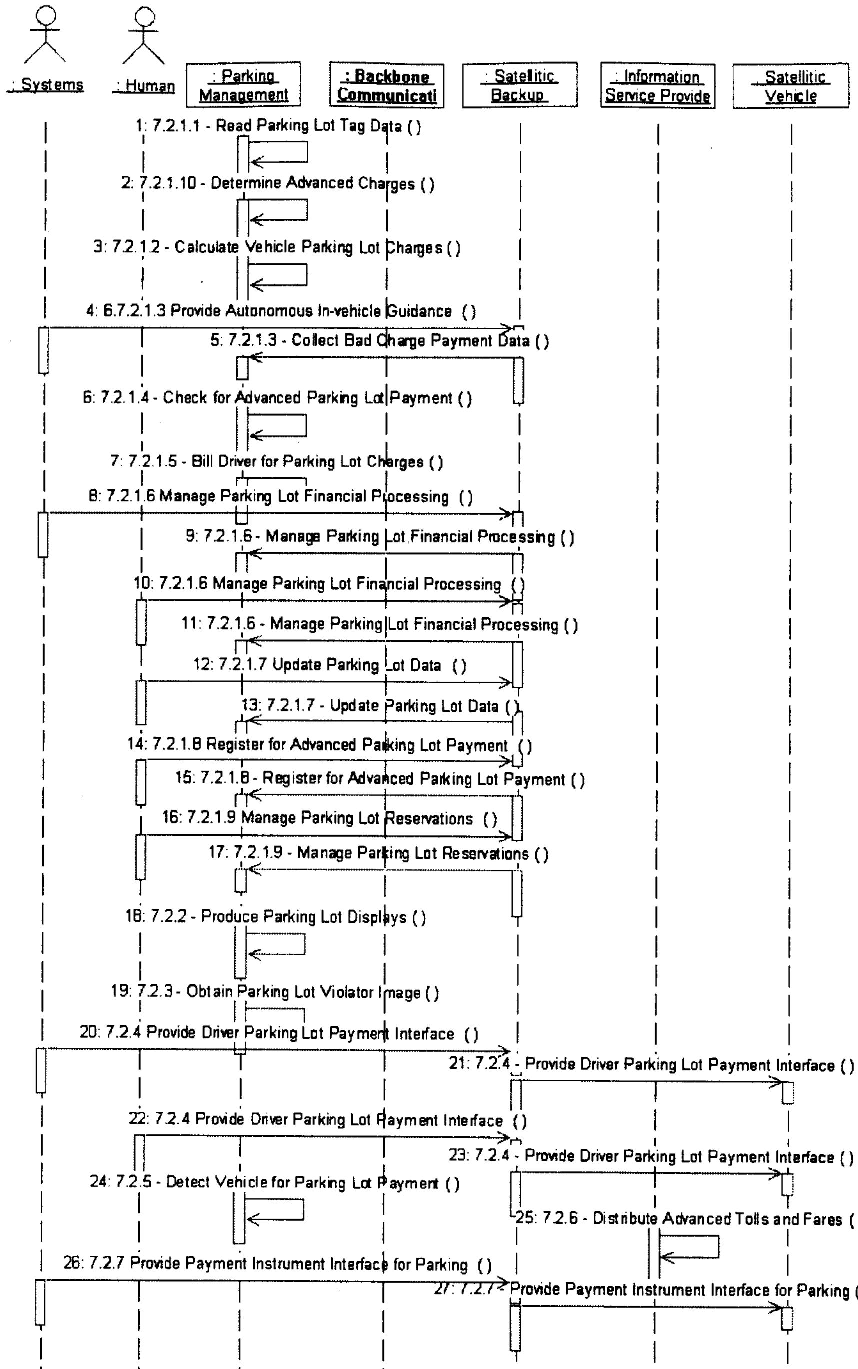

Figure 6.36 Object-oriented visualization of Sequence Diagram 7B.2 Provide Electronic Parking Payment for SAHS in UML

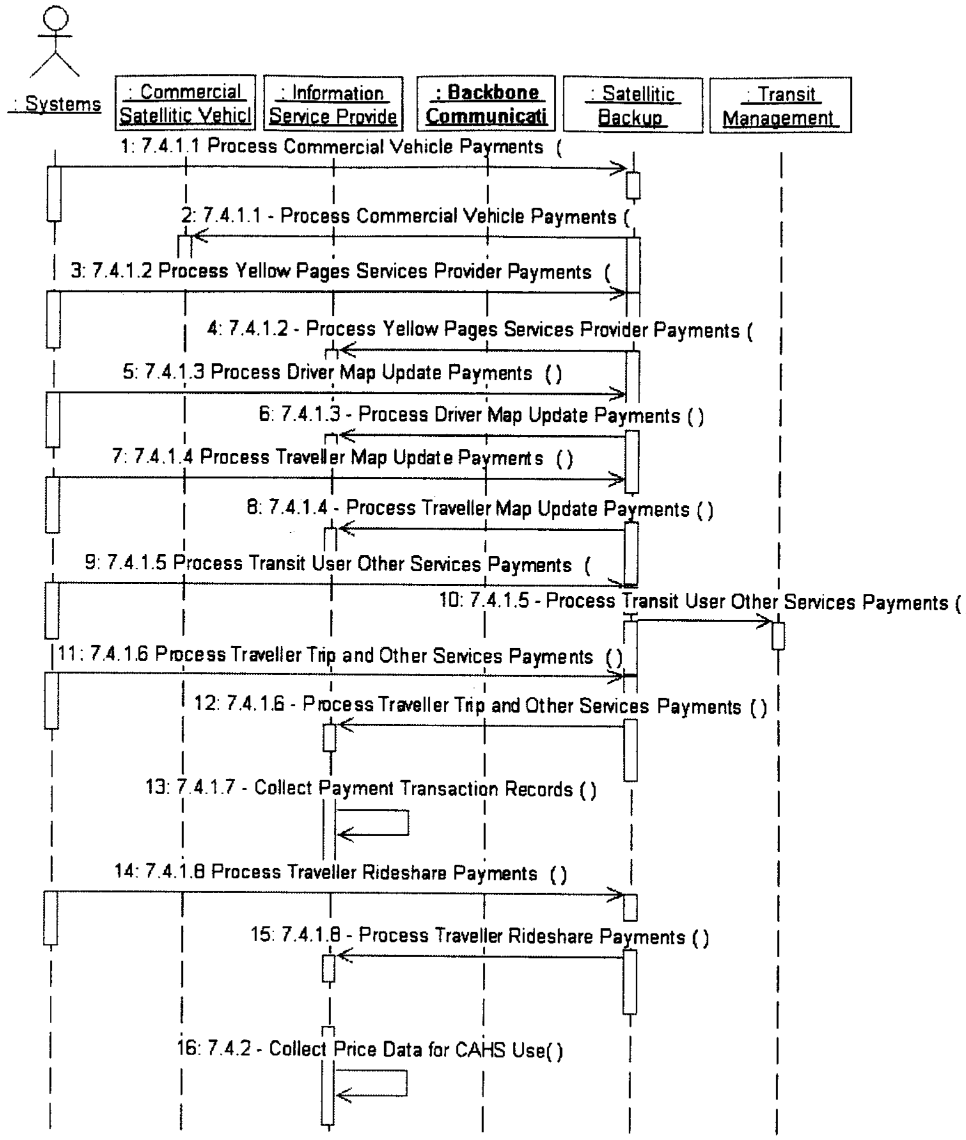

Figure 6.37 Object-oriented visualization of Sequence Diagram 7B.4 Carry-out Centralized Payments Processing for SAHS in UML

<<System>>
Backbone Communication

- archive management requests
- commercial vehicle archive data
- emergency archive data
- emissions archive data
- government reporting data receipt
- transaction status
- traveller archive data
- intermodal freight archive data
- construction and maintenance archive data
- map updates
- multimodal archive data
- volume weather information
- archive coordination
- other data source archive data
- parking archive data
- roadside archive data
- toll archive data
- traffic archive data
- transit archive data
- archive analysis requests
- archive requests
- citation data
- credentials and safety information request
- information on violators
- credential application**
- information request

Figure 6.38 (Part 1) Object-oriented visualization of SAHS's Class Backbone Communication with its attributes and Functions in UML

information on violators
credential application**
information request
tax filing, audit data
Commercial Vehicle Administration Subsystem information exchange
credentials and safety information response
archive status
pollutant levels
pollution data
pollution state data request
pollution data parameters
activity reports
compliance review report
electronic credentials
fleet manager inquiry
driver and vehicle information
intermodal status
on-board vehicle data
vehicle location
container location request
container status request
HAZMAT information request
tax filing, audit data
container delivery confirmation
container pickup request
route plan
freight consignment request **
shipment status request

Figure 6.39 (Part 2) Object-oriented visualization of SAHS's Class Backbone Communication with its attributes and Functions in UML

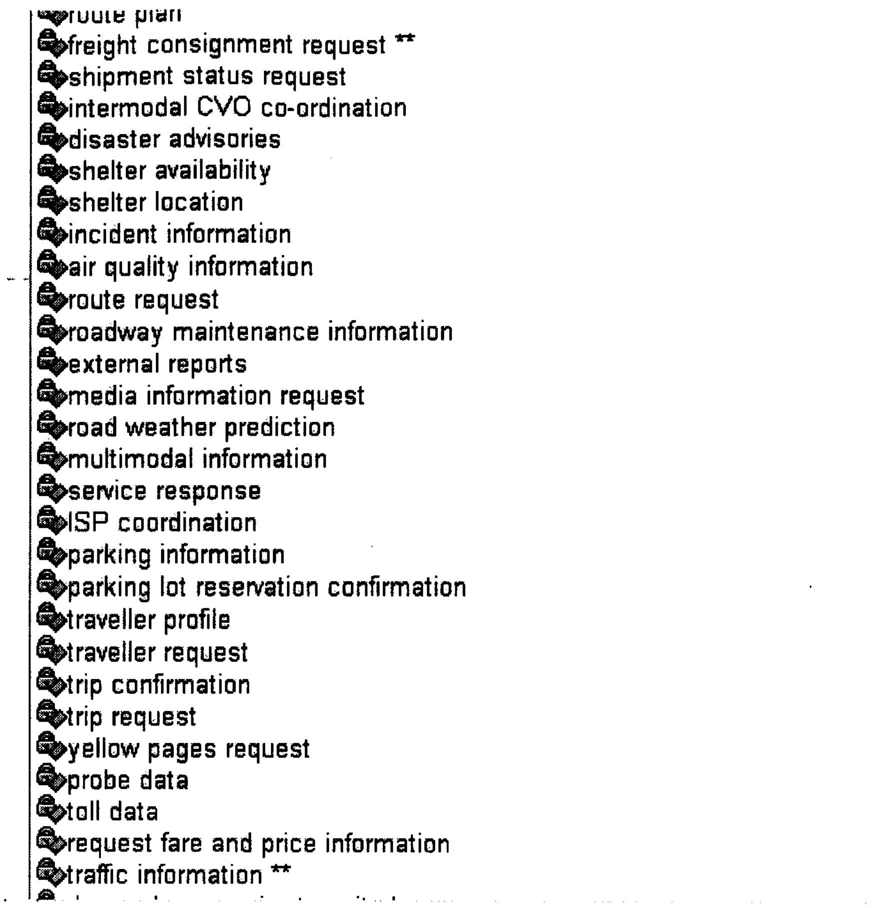

Figure 6.40 (Part 3) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

traffic information **
demand responsive transit plan
transit and fare schedules
transit incident information
transit request confirmation
transit trip plan
vehicle probe data
provider profile data
travel service info
maintenance operator inputs
maintenance status data
maintenance vehicle status data
roadway micro prediction
device status
environmental conditions
fault reports
roadway information system status
surveillance data
closure coordination
environmental sensor data
maintenance resource request
traffic equipment status
disaster response coordination
disaster response status
incident notification **
emergency dispatch response
emergency vehicle tracking data

Figure 6.41 (Part 4) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

disaster response status
incident notification **
emergency dispatch response
emergency vehicle tracking data
incident command request
incident status
event plans
HAZMAT information
incident information request
medical facility availability
incident report
incident response coordination
emergency notification
current network conditions
resource deployment status
transit emergency data
evacuation support request
transit emergency coordination data
demand responsive transit request
selected routes
transit information request
transit trip request
roadway maintenance information
transit multimodal information
transit parking coordination
secure area surveillance data
request transit information **
traffic information for transit

Figure 6.42 (Part 5) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

secure area surveillance data
request transit information **
traffic information for transit
transit demand management request
transit driver availability
transit fleet manager inputs
maintenance status
transit operator management data
fare and payment status
request for bad tag list
transit traveller request
transit user information
transit vehicle conditions
transit vehicle location data
transit vehicle passenger and use data
transit vehicle schedule performance
evacuation support request
transit multimodal information
Transit Management System Co-ordination
transit operator management data
transit user information
transit vehicle conditions
transit vehicle location data
road network status request
emergency traffic control request
incident response status
remote surveillance control **
resource request

Figure 6.43 (Part 6) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

incident response status
remote surveillance control **
resource request
widearea statistical pollution information
fare and price information
logged special vehicle route
request for traffic information
road network use
environmental sensor data
equipment maintenance status
maintenance dispatch status
maintenance resource response
work zone status
external reports
traffic control coordination
traffic information coordination
parking availability
parking demand management response
railroad advisories
railroad schedules
AHS status
emissions data
freeway control status
HOV data
HRI status
incident data
intersection blockage notification
request for right-of-way **
reversible lane status

Figure 6.44 (Part 7) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

intersection blockage notification
request for right-of-way **
reversible lane status
roadway warning status
signal control status
traffic flow
traffic images
violation detection
toll demand management response
transit system data
registration
toll data request
toll administration requests
Toll Transactions
toll demand management request
CVO weight and presence
credentials information
CVO database update
international border crossing data
safety information
electronic clearance data
lock tag data
on-board safety data
screening data
Commercial Vehicle Check override mode
CVO inspector input
container transfer location request
exit request **
vehicle identification

Figure 6.45 (Part 8) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

container transfer location request
exit request **
vehicle identification
container release status
container availability request
container pickup confirmation
container identification
container location
intermodal terminal operator inputs
parking lot data request
parking reservations request
parking operator inputs
request for performance data
parking demand management request
parking instructions
transit parking lot response
vehicle characteristics
tag data
sensor and surveillance control
local signal preemption request
vehicle pollution criteria
environmental hazards
pollutant levels
device control data
roadway information system data
sensor control **
multimodal crossing status

Figure 6.46 (Part 9) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

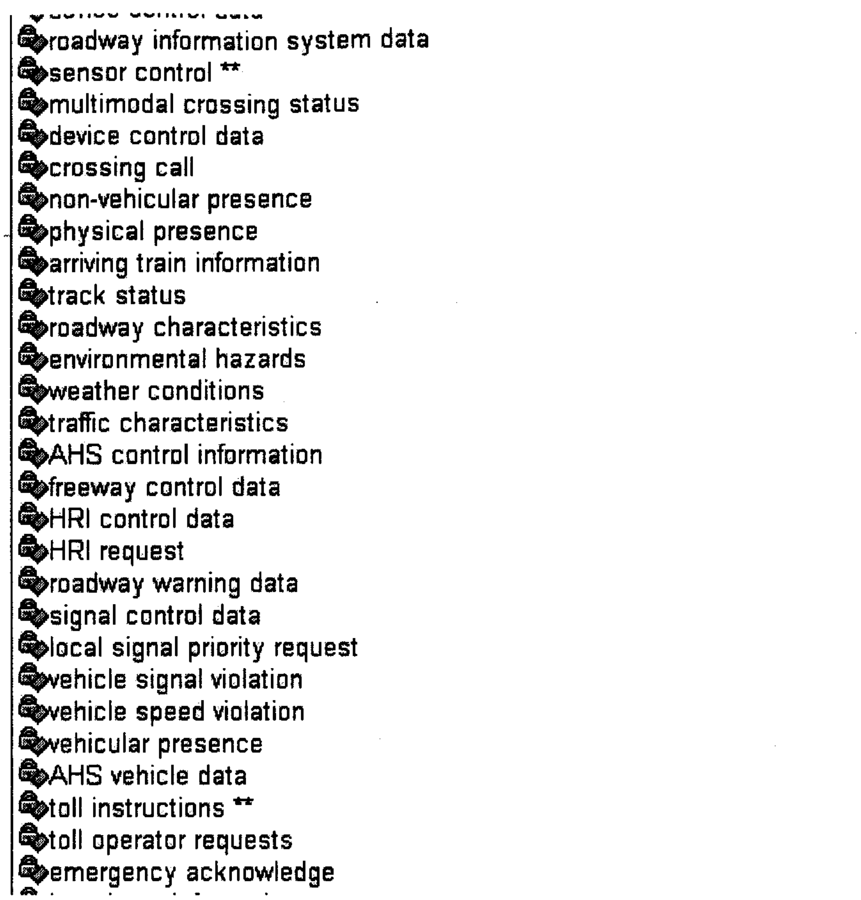

Figure 6.47 (Part 10) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

AHS vehicle data
toll instructions **
toll operator requests
emergency acknowledge
broadcast information
trip plan
yellow pages information
position fix
payment
personal transit information
traveller inputs
traveller information
secure area characteristics
secure area monitoring support
transit fare payment responses
transit traveller information
transit user inputs
traveller inputs
commercial vehicle measures
border clearance event record
clearance event record
electronic clearance request
lock tag data request
on-board safety request
safety inspection record
screening request **
CVO driver initialization

Figure 6.48 (Part 11) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

screening request **
CVO driver initialization
container transport assignment
fleet to driver update
intermodal status request
location request
container transfer location
entry permission
exit permission
chassis data
cargo data
container status
commercial vehicle data request
emergency dispatch requests
incident command information
suggested route
emergency personnel inputs
maintenance vehicle dispatch
maintenance vehicle measures
maintenance driver inputs
environmental hazards
transit driver inputs
bad tag list
driver instructions
fare management information
request for vehicle measures
transit schedule information **
emergency request

Figure 6.49 (Part 12) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

request for vehicle measures
transit schedule information **
emergency request
transit vehicle measures
basic vehicle measures
commercial vehicle data
driver inputs
request for service
emergency data request
vehicle to vehicle coordination
request tag data
tag update
AHS control data
intersection status
vehicle roadway warning data
vehicle signage data
vehicle variable speed limit data
traveller advisory request

Figure 6.50 (Part 13) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

0 Manage CAHS ()
1 Manage Traffic ()
1.1 Provide Traffic Surveillance ()
1.1.1 Process Sensor Data ()
1.1.1.1 Process Traffic Sensor Data()
1.1.1.2 Collect and Process Sensor Fault Data ()
1.1.1.3 Process Environmental Sensor Data ()
1.1.1.4 Manage Data Collection and Monitoring ()
1.1.1.5 Process Advanced Roadway Data ()
1.1.1.5.1 Collect Roadway Condition Data ()
1.1.1.5.2 Provide Other Roadside Inputs ()
1.1.1.5.3 Process Roadway Warning Data ()
1.1.5.4 Process Road User Protection()
1.1.2 Process and Store Traffic Data()
1.1.2.1 Process Traffic Data for Storage ()
1.1.2.2 Process Traffic Data ()
1.1.2.3 Update Data Source Static Data ()
1.1.2.4 Monitor HOV lane use ()
1.1.2.5 Process Tag/AVL Data for Link Time Data ()
1.1.2.6 Process Collected Vehicle Smart Probe Data ()
1.1.2.7 Monitor Reversible Lanes ()
1.1.2.8 Manage Environmental Data for Traffic()
1.1.3 Generate Predictive Traffic Model ()
1.1.4 Display and Output Traffic Data ()
1.1.4.1 Retrieve Traffic Data ()
1.1.4.2 Provide Traffic Operations Personnel Traffic Data Interface ()
1.1.4.3 Provide Direct Media Traffic Data Interface ()
1.1.4.4 Update Traffic Display Map Data ()
1.1.4.5 Provide Media System Traffic Data Interface ()
1.1.4.6 Provide Traffic Data Retrieval Interface ()
1.1.4.7 Manage Traffic Archive Data ()
1.1.4.8 Provide Disaster Command Data ()
1.1.5 Exchange data with Other Traffic Centres ()
1.1.6 Collect Vehicle Tag Data for Link Time Calculations **()
1.1.7 Collect Vehicle Smart Probe Data ()

Figure 6.51 (Part 14) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

1.1.6 Collect Vehicle Tag Data for Link Time Calculations **()
1.1.7 Collect Vehicle Smart Probe Data ()
1.2 Provide Device Control()
1.2.1 Select Strategy ()
1.2.2 Determine Road and Freeway State ()
1.2.2.1 Determine Indicator State for Freeway Management ()
1.2.2.2 Determine Indicator State for Road Management ()
1.2.3 Determine Ramp State ()
1.2.4 Output Control Data ()
1.2.4.1 Output Control Data for Roads ()
1.2.4.2 Output Control Data for Freeways ()
1.2.4.3 Output In-vehicle Signage Data ()
1.2.5 Manage Parking Lot State ()
1.2.5.1 Determine Parking Lot State ()
1.2.5.2 Coordinate Other Parking Data ()
1.2.5.3 Provide Parking Lot Operator Interface ()
1.2.5.4 Determine P+R needs for Transit Management ()
1.2.5.5 Manage Parking Archive Data ()
1.2.5.6 Calculate Parking Lot Occupancy ()
1.2.6 Maintain Static Data for TMC ()
1.2.6.1 Maintain Traffic and Sensor Static Data ()
1.2.6.2 Provide Static Data Store Output Interface ()
1.2.7 Provide Roadside Control Facilities ()
1.2.7.1 Process Indicator Output Data for Roads ()
1.2.7.2 Monitor Roadside Equipment Operation for Faults ()
1.2.7.3 Manage Indicator Preemptions ()
1.2.7.4 Process In-vehicle Signage Data ()
1.2.7.5 Process Indicator Output Data for Freeways()
1.2.7.6 Provide Intersection Collision Avoidance Data ()
1.2.7.7 Process Vehicle Smart Probe Data for Output ()
1.2.7.8 Receive Other Roadside Inputs ()
1.2.7.9 Display Roadway Warnings ()
1.2.8 Collect and Process Indicator Fault Data ()
1.2.8.1 Collect Indicator Fault Data ()
1.2.8.2 Maintain Indicator Fault Data Store ()
1.2.8.3 Provide Indicator Fault Interface for C and M **()
1.2.8.4 Provide Traffic Operations Personnel Indicator Fault Interface ()

Figure 6.52 (Part 15) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

♦1.2.8.3 Provide Indicator Fault Interface for C and M **()
♦1.2.8.4 Provide Traffic Operations Personnel Indicator Fault Interface ()
♦1.3 Manage Incidents ()
♦1.3.1 Traffic Data Analysis for Incidents ()
♦1.3.1.1 Analyze Traffic Data for Incidents ()
♦1.3.1.2 Maintain Static Data for Incident Management ()
♦1.3.1.3 Process Traffic Images ()
♦1.3.2 Review and Manage Incident Data ()
♦1.3.2.1 Store Possible Incident Data ()
♦1.3.2.2 Review and Classify Possible Incidents ()
♦1.3.2.3 Review and Classify Planned Events ()
♦1.3.2.4 Provide Planned Events Store Interface ()
♦1.3.2.5 Provide Current Incidents Store Interface ()
♦1.3.3 Respond to Current Incidents ()
♦1.3.4 Provide Operator Interfaces for Incidents ()
♦1.3.4.1 Retrieve Incident Data ()
♦1.3.4.2 Provide Traffic Operations Personnel Incident Data Interface ()
♦1.3.4.3 Provide Media Incident Data Interface ()
♦1.3.4.4 Update Incident Display Map Data ()
♦1.3.4.5 Manage Resources for Incidents ()
♦1.3.5 Manage Possible Predetermined Responses Store ()
♦1.3.6 Manage Predetermined Incident Response Data ()
♦1.3.7 Analyze Incident Response Log ()
♦1.4 Manage Travel Demand ()
♦1.4.1 Provide Traffic Operations Personnel Demand Interface ()
♦1.4.2 Collect Demand Forecast Data ()
♦1.4.3 Update Demand Display Map Data ()
♦1.4.4 Implement Demand Management Policy ()
♦1.4.5 Calculate Forecast Demand ()
♦1.5 Manage Emissions ()
♦1.5.1 Provide Traffic Operations Personnel Pollution Data Interface ()
♦1.5.2 Process Pollution Data ()
♦1.5.3 Update Pollution Display Map Data ()
♦1.5.4 Manage Pollution State Data Store ()
♦1.5.5 Process Vehicle Pollution Data ()
♦1.5.6 Detect Roadside Pollution Levels ()
♦1.5.7 Manage Pollution Data Log**()
♦1.5.8 Manage Pollution Reference Data Store ()

Figure 6.53 (Part 16) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

- 1.5.7 Manage Pollution Data Log**()
- 1.5.8 Manage Pollution Reference Data Store ()
- 1.5.9 Manage Pollution Archive Data ()
- 1.6 Manage Highway Rail Intersections ()
- 1.6.1 Manage HRI Vehicle Traffic ()
- 1.6.1.1 Detect Roadway Events ()
- 1.6.1.2 Activate HRI Device Controls ()
- 1.6.1.2.1 Control HRI Traffic Signals ()
- 1.6.1.2.2 Control HRI Warnings and Barriers ()
- 1.6.1.2.3 Provide SSR Device Controls ()
- 1.6.1.2.4 Provide HSR Device Controls ()
- 1.6.1.2.5 Manage Device Control ()
- 1.6.1.2.6 Maintain Device State (()
- 1.6.1.3 Perform Equipment Self-Test ()
- 1.6.1.4 Provide Advisories and Alerts ()
- 1.6.1.4.1 Generate Alerts and Advisories ()
- 1.6.1.4.2 Provide Closure Parameters ()
- 1.6.1.4.3 Report Alerts and Advisories ()
- 1.6.1.4.4 Report HRI Status on Approach ()
- 1.6.1.5 Detect HRI Hazards ()
- 1.6.1.6 Provide Advance Warnings ()
- 1.6.1.6.1 Close HRI on Detection ()
- 1.6.1.6.2 Detect Imminent Vehicle/Train Collision ()
- 1.6.1.7 Execute Local Control Strategy ()
- 1.6.1.7.1 Control Vehicle Traffic at Passive HRI ()
- 1.6.1.7.2 Control Vehicle Traffic at Active HRI ()
- 1.6.1.7.3 Close HRI on Command ()
- 1.6.2 Interact with Rail Operations ()
- 1.6.2.1 Exchange Data with Rail Operations ()
- 1.6.2.2 Manage Alerts and Advisories ()
- 1.6.2.3 Manage Rail Traffic Control Data ()
- 1.6.3 Manage HRI Rail Traffic ()
- 1.6.3.1 Interact with Wayside Systems ()
- 1.6.3.2 Advise and Protect Train Crews ()
- 1.6.3.3 Provide ATS Alerts ()
- 1.6.4 Interact with Vehicle Traffic Management**()
- 1.6.4.1 Manage HRI Closures ()
- 1.6.4.2 Exchange Data with Traffic Management ()

Figure 6.54 (Part 17) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

- 1.6.4 Interact with Vehicle Traffic Management**()
- 1.6.4.1 Manage HRI Closures ()
- 1.6.4.2 Exchange Data with Traffic Management ()
- 1.6.5 Monitor HRI Status ()
- 1.6.5.1 Provide Interactive Interface ()
- 1.6.5.2 Determine HRI Status ()
- 1.6.5.3 Maintain HRI Closure Data ()
- 1.7 Manage Operations and Maintenance ()
- 1.7.1 Manage Operations and Maintenance Data ()
- 1.7.1.1 Gather O&M Data ()
- 1.7.1.2 Manage O&M Archive Data ()
- 1.7.1.3 Manage O&M Assets ()
- 1.7.1.4 Manage Environmental Data for Operations and Maintenance ()
- 1.7.1.4.1 Gather and Process Environmental Data ()
- 1.7.1.4.2 Create Road Forecasts ()
- 1.7.1.4.3 Fuse Environmental Data ()
- 1.7.2 Manage Maintenance Activity ()
- 1.7.2.1 Schedule Maintenance Activity ()
- 1.7.2.2 Manage and Control On-board Maintenance Systems ()
- 1.7.2.3 Manage Maintenance Operator Interface ()
- 1.7.2.4 Manage Maintenance Driver Interface ()
- 1.7.2.5 Collect Environmental Data On-board ()
- 1.7.3 Control Roadway Maintenance Devices ()
- 1.7.4 Disseminate O&M Information ()
- 1.8 Manage Traffic Enforcement ()
- 1.8.1 Detect and Classify Speed Violations ()
- 1.8.2 Collect and Verify Speed Violations ()
- 1.8.3 Post Speed Limit ()
- 1.8.4 Set Speed Limit ()
- 1.8.5 Establish Violation Parameters ()
- 1.8.6 Detect and Classify Signal Violations ()
- 1.8.7 Collect and Verify Signal Violations ()
- 2 Manage Commercial Vehicles ()
- 2.1 Manage Commercial Vehicle Fleet Operations ()
- 2.1.1 Manage Commercial Fleet Electronic Credentials and Tax Filing ()
- 2.1.2 Provide Commercial Fleet Static Route ()
- 2.1.3 Provide Flt Mgr Electronic Credentials and Tax Filing Interface**()

Figure 6.55 (Part 18) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

2.1.3 Provide Flt Mgr Electronic Credentials and Tax Filing Interface**()
2.1.4 Provide Fleet Manager Commercial Vehicle Communications ()
2.1.5 Provide Commercial Vehicle Driver Routing Interface ()
2.1.6 Manage Driver Instruction Store ()
2.2 Manage Commercial Vehicle Driver Operations ()
2.2.1 Manage CV Electronic Credential and Tax Filing Interface ()
2.2.2 Provide Vehicle Static Route ()
2.2.3 Provide CV Driver Electronic Credential and Tax Filing Interface ()
2.2.4 Provide Commercial Vehicle Driver Communications ()
2.3 Provide Commercial Vehicle Roadside Facilities ()
2.3.1 Produce Commercial Vehicle Driver Message at Roadside ()
2.3.2 Provide Commercial Vehicle Clearance Screening ()
2.3.2.1 Administer Commercial Vehicle Roadside Credentials Database ()
2.3.2.2 Process Screening Transactions ()
2.3.3 Provide Roadside Commercial Vehicle Safety ()
2.3.3.1 Provide Commercial Vehicle Checkstation Communications ()
2.3.3.2 Provide Commercial Vehicle Inspector Handheld Terminal Interface ()
2.3.3.3 Administer Commercial Vehicle Roadside Safety Datadase ()
2.3.3.4 Carry-out Commercial Vehicle Roadside Safety Screening ()
2.3.3.5 Carry-out Commercial Vehicle Roadside Inspection ()
2.3.4 Detect Commercial Vehicle ()
2.3.5 Provide Commercial Vehicle RoadsideOperator Interface ()
2.3.6 Provide Commercial Vehicle Reports ()
2.3.7 Produce Commercial Vehicle Driver Message on Vehicle ()
2.3.8 Provide Commercial Vehicle Border Screening ()
2.4 Provide Commercial Vehicle Data Collection ()
2.4.1 Communicate Commercial Vehicle On-board Data to Roadside ()
2.4.2 Collect On-board Commercial Vehicle Sensor Data ()
2.4.3 Analyze Commercial Vehicle On-board Data ()
2.4.4 Provide Commercial Vehicle Driver Interface ()
2.4.5 Communicate Commercial Vehicle On-board Data to Vehicle Manager ()
2.4.6 Provide Commercial Vehicle On-board Data Store Interface ()
2.5 Administer Commercial Vehicles ()
2.5.1 Manage Commercial Vehicle Trips and Clearances ()
2.5.2 Obtain Electronic Credential and Tax Filing Payment ()
2.5.3 Update Permits and Duties Store **()
2.5.4 Communicate with Other Commercial Vehicle Administration System ()

Figure 6.56 (Part 19) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

2.5.2 Obtain Electronic Credential and Tax Filing Payment ()
2.5.3 Update Permits and Duties Store **()
2.5.4 Communicate with Other Commercial Vehicle Administration System ()
2.5.5 Manage Commercial Vehicle Credentials and Enrollment ()
2.5.6 Output Commercial Vehicle Enrollment Data to Roadside Facilities ()
2.5.7 Process Commercial Vehicle Violations ()
2.5.8 Process Data Received from Roadside Facilities ()
2.5.9 Manage Commercial Vehicle Archive Data ()
2.6 Provide Commercial Vehicle On-board Data ()
2.6.1 Provide Commercial Vehicle Manager Tag Data Interface ()
2.6.2 Transmit Commercial Vehicle Tag Data ()
2.6.3 Provide Commercial Driver Tag Data Interface ()
2.6.4 Provide Lock Tag Data Interface ()
2.6.5 Manage Commercial Vehicle Tag Data Store ()
2.7 Manage Cargo ()
2.7.1 Manage Intermodal Terminal Interface ()
2.7.1.1 Manage Terminal Access ()
2.7.1.2 Manage Container Pickup and Delivery ()
2.7.1.3 Manage Container Release ()
2.7.1.4 Provide Intermodal Terminal Operator I/F ()
2.7.1.5 Provide Container Tracking ()
2.7.2 Manage Intermodal Container ()
2.7.2.1 Determine Container Status ()
2.7.2.2 Determine Cargo Status ()
2.7.2.3 Provide Cargo Status Interface ()
2.7.2.4 Provide Container Status Interface ()
2.7.2.5 Manage Customs Interface ()
2.7.3 Manage On-Board Intermodal Applications ()
2.7.3.1 Provide Driver Interface for Intermodal Freight Dispatch()
2.7.3.2 Monitor Container Status ()
2.7.3.3 Monitor Chassis Status ()
2.7.3.4 Manage On-Board Facility Access ()
2.7.4 Manage Intermodal Dispatch ()
2.7.4.1 Monitor Intermodal Elements ()
2.7.4.2 Provide Fleet Manager Interface for Intermodal ()
2.7.4.3 Manage Intermodal Customer Interface**()
2.7.4.4 Manage Other Intermodal FMS Interface ()
2.7.4.5 Manage Distribution and Logistics Management Provider Interface ()

Figure 6.57 (Part 20) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

2.7.4.2 Provide Fleet Manager Interface for Intermodal ()
2.7.4.3 Manage Intermodal Customer Interface**()
2.7.4.4 Manage Other Intermodal FMS Interface ()
2.7.4.5 Manage Distribution and Logistics Management Provider Interface ()
2.7.4.6 Manage Freight Consolidation Station Interface ()
3 Provide Vehicle Monitoring and Control ()
3.1 Monitor Vehicle Status ()
3.1.1 Produce Collision and Crash Avoidance Data ()
3.1.2 Carry-out Safety Analysis ()
3.1.3 Process Vehicle On-board Data ()
3.2 Provide Automatic Vehicle Operation ()
3.2.1 Provide Driver Interface ()
3.2.2 Provide AHS Control ()
3.2.3 Provide Vehicle Control ()
3.2.3.1 Provide Command Interface ()
3.2.3.2 Manage Platoon Following ()
3.2.3.3 Process data for Vehicle Actuators ()
3.2.3.4 Provide Servo Controls ()
3.2.3.4.1 Provide Speed Servo Control ()
3.2.3.4.2 Provide Headway Servo Control ()
3.2.3.4.3 Provide Lane Servo Control ()
3.2.3.4.4 Provide Change Lane Servo Control ()
3.2.3.4.5 Provide Vehicle Control Data Interface ()
3.2.3.5 Process Vehicle Sensor Data ()
3.2.3.6 Communicate with other Platoon Vehicles ()
3.2.4 Process Sensor Data for AHS input ()
3.2.5 Check Vehicle for AHS eligibility ()
3.2.6 Manage AHS Check-in and Check-out ()
3.2.7 Manage AHS Operations ()
3.3 Provide Automatic Emergency Notification ()
3.3.1 Provide Cargo Data for Incident Notification ()
3.3.2 Provide Communications Function ()
3.3.3 Build Automatic Collision Notification Message ()
3.4 Enhance Driver's Vision ()
4 Manage Transit ()
4.1 Operate Vehicles and Facilities ()
4.1.1 Process Transit Vehicle Sensor Trip Data**()
4.1.2 Determine Transit Vehicle Deviation and Corrections ()

Figure 6.58 (Part 21) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

4.1.1 Process Transit Vehicle Sensor Trip Data**()
4.1.2 Determine Transit Vehicle Deviation and Corrections ()
4.1.2.1 Determine Transit Vehicle Deviation and ETA ()
4.1.2.2 Determine Transit Vehicle Corrective Instructions ()
4.1.2.3 Provide Transit Vehicle Driver Interface ()
4.1.2.4 Provide Transit Vehicle Correction Data Output Interface ()
4.1.2.5 Request Transit Vehicle Preemptions ()
4.1.3 Provide Transit Vehicle Location Data ()
4.1.4 Manage Transit Vehicle Deviations ()
4.1.5 Provide Transit Vehicle Status Information ()
4.1.6 Manage Transit Vehicle Operations Data ()
4.1.7 Manage Connection Protection ()
4.1.7.1 Manage Connections with External Systems ()
4.1.7.2 Provide Transit Vehicle Deviation Data Output Interface ()
4.1.7.3 Manage Individual Service Requests ()
4.1.8 Provide Transit Operations Data Distribution Interface ()
4.1.9 Process Transit Vehicle Sensor Maintenance Data ()
4.2 Plan and Schedule Transit Services ()
4.2.1 Provide Demand Responsive Transit Service ()
4.2.1.1 Process Demand Responsive Transit Trip Request ()
4.2.1.2 Compute Demand Responsive Transit Vehicle Availability ()
4.2.1.3 Generate Demand Responsive Transit Schedule and Routes ()
4.2.1.4 Confirm Demand Responsive Transit Schedule and Route ()
4.2.1.5 Process Demand Responsive Transit Vehicle Availability Data ()
4.2.1.6 Provide Demand Responsive Transit Driver Interface ()
4.2.2 Provide Transit Plans Store Interface ()
4.2.3 Generate Transit Routes and Schedules ()
4.2.3.1 Generate Transit Routes ()
4.2.3.2 Generate Schedules ()
4.2.3.3 Produce Transit Service Data for External Use ()
4.2.3.4 Provide Transit Fleet Manager Interface for Services Generation ()
4.2.3.5 Manage Transit Operational Data Store ()
4.2.3.6 Produce Transit Service Data for Manage Transit Use ()
4.2.3.7 Provide Interface for Other TRM Data ()
4.2.3.8 Provide Interface for Transit Service Raw Data ()
4.2.3.9 Update Transit Map Data ()
4.2.4 Manage Transit Archive Data **()
4.3 Schedule Transit Vehicle Maintenance ()

Figure 6.59 (Part 22) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

4.2.3.9 Update Transit Map Data ()
4.2.4 Manage Transit Archive Data **()
4.3 Schedule Transit Vehicle Maintenance ()
4.3.1 Monitor Transit Vehicle Condition ()
4.3.2 Generate Transit Vehicle Maintenance Schedules ()
4.3.3 Generate Technician Work Assignments ()
4.3.4 Monitor And Verify Maintenance Activity ()
4.3.5 Report Transit Vehicle Information (()
4.3.6 Update Transit Vehicle Information ()
4.3.7 Manage Transit Vehicle Operations Data Store ()
4.4 Support Security and Coordination ()
4.4.1 Provide Transit Security and Emergency Management ()
4.4.1.1 Manage Transit Security ()
4.4.1.2 Manage Transit Emergencies ()
4.4.1.3 Provide Transit System Operator Security Interface ()
4.4.1.4 Provide Transit External Interface for Emergencies ()
4.4.1.5 Provide Transit Driver Interface for Emergencies ()
4.4.1.6 Collect Transit Vehicle Emergency Information ()
4.4.1.7 Monitor Secure Area ()
4.4.1.8 Report Traveller Emergencies ()
4.4.2 Coordinate Multiple Agency Responses to Incidents ()
4.4.3 Generate Responses for Incidents ()
4.4.4 Coordinate Transit Disaster Response ()
4.5 Generate Transit Driver Schedules ()
4.5.1 Assess Transit Driver Performance ()
4.5.2 Assess Transit Driver Availability ()
4.5.3 Access Transit Driver Cost Effectiveness ()
4.5.4 Assess Transit Driver Eligibility ()
4.5.5 Generate Transit Driver Route Assignments ()
4.5.6 Update Transit Driver Information ()
4.5.7 Report Transit Driver Information ()
4.5.8 Provide Transit Driver Information Store Interface ()
4.6 Collect Transit Fares in the Vehicle ()
4.6.1 Detect Transit User on Vehicle ()
4.6.2 Determine Transit User Needs on Vehicle ()
4.6.3 Determine Transit Fare on Vehicle ()
4.6.4 Manage Transit Fare Billing on Vehicle **()
4.6.5 Provide Transit User Fare Payment Interface on Vehicle ()

Figure 6.60 (Part 23) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

4.6.4 Manage Transit Fare Billing on Vehicle **()
4.6.5 Provide Transit User Fare Payment Interface on Vehicle ()
4.6.6 Update Transit Vehicle Fare Data ()
4.6.7 Provide Transit Vehicle Passenger Data ()
4.6.8 Manage Transit Vehicle Advanced Payments ()
4.7 Provide Transit User Roadside Facilities ()
4.7.1 Provide Transit User Roadside Information ()
4.7.1.1 Provide Transit User Roadside Data Interface ()
4.7.1.2 Provide Transit User Roadside Vehicle Data Interface ()
4.7.2 Collect Transit Fares at the Roadside ()
4.7.2.1 Detect Transit User at Roadside ()
4.7.2.2 Determine Transit User Needs at Roadside ()
4.7.2.3 Determine Transit Fare at Roadside ()
4.7.2.4 Manage Transit Fare Billing at Roadside ()
4.7.2.5 Provide Transit User Roadside Fare Interface ()
4.7.2.6 Update Roadside Transit Fare Data ()
4.7.2.7 Provide Transit Roadside Passenger Data ()
5 Manage Emergency Services ()
5.1 Provide Emergency Service Allocation ()
5.1.1 Identify Emergencies from Inputs ()
5.1.2 Determine Coordinated Response Plan ()
5.1.3 Communicate Emergency Status ()
5.1.4 Manage Emergency Response ()
5.1.5 Manage Emergency Service Allocation Store ()
5.1.6 Process Mayday Messages ()
5.2 Provide Operator Interface for Emergency Data ()
5.3 Manage Emergency Vehicles ()
5.3.1 Select Response Mode ()
5.3.2 Dispatch Vehicle ()
5.3.3 Track Vehicle ()
5.3.4 Assess Response Status ()
5.3.5 Provide Emergency Personnel Interface ()
5.3.6 Maintain Vehicle Status ()
5.3.7 Provide Emergency Vehicle Route ()
5.4 Provide Law Enforcement Allocation ()
5.4.1 Process TM Detected Violations ()
5.4.2 Process Violations for Tolls ()
5.4.3 Process Parking Lot Violations**()

Figure 6.61 (Part 24) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

5.4.2 Process Violations for Tolls ()
5.4.3 Process Parking Lot Violations**()
5.4.4 Process Fare Payment Violations ()
5.4.5 Process Vehicle Fare Collection Violations ()
5.4.6 Process CV Violations ()
5.4.7 Process Roadside Fare Collection Violations ()
5.5 Update Emergency Display Map Data ()
5.6 Manage Emergency Services Data ()
5.7 Co-ordinate Disaster Response ()
5.7.1 Collect Disaster Response Data ()
5.7.2 Provide Medical Facility Interface ()
6 Provide Driver and Traveller Services ()
6.1 Provide Trip Planning Services ()
6.1.1 Provide Trip Planning Information to Traveller ()
6.1.2 Confirm Traveller's Trip Plan ()
6.1.3 Manage Multimodal Service Provider Interface ()
6.1.4 Provide ISP Operator Interface for Trip Planning Parameters ()
6.1.5 Collect Service Requests and Confirmation for Archive ()
6.1.6 Manage Traveller Info Archive Data ()
6.2 Provide Information Services ()
6.2.1 Provide Advisory and Broadcast Data ()
6.2.1.1 Collect Traffic Data for Advisory Messages ()
6.2.1.2 Provide Traffic and Transit Advisory Messages ()
6.2.1.3 Collect Transit Data for Advisory Messages ()
6.2.1.4 Provide Traffic and Transit Broadcast Messages ()
6.2.1.5 Provide ISP Operator Broadcast Parameters Interface ()
6.2.1.6 Provide Transit Advisory Data On Vehicle ()
6.2.2 Prepare and Output In-vehicle Displays ()
6.2.3 Provide Transit User Advisory Interface ()
6.2.4 Collect Yellow Pages Data ()
6.2.5 Provide Driver Interface ()
6.2.6 Provide Yellow Pages Data and Reservations ()
6.3 Provide Traveller Services at Kiosks ()
6.3.1 Get Traveller Request ()
6.3.2 Inform Traveller ()
6.3.3 Provide Traveller Kiosk Interface ()
6.3.4 Update Traveller Display Map Data at Kiosk ()
6.4 Manage Ridesharing**()

Figure 6.62 (Part 25) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

6.4 Manage Ridesharing**()
6.4.1 Screen Rider Requests ()
6.4.2 Match Rider and Provider ()
6.4.3 Report Ride Match Results to Requestor ()
6.4.4 Confirm Traveller Rideshare Request ()
6.5 Manage Yellow Pages Services ()
6.5.1 Collect and Update Traveller Information ()
6.5.2 Provide Traveller Yellow Pages Information and Reservations ()
6.5.3 Register Yellow Pages Service Providers ()
6.6 Provide Guidance and Trip Planning Services ()
6.6.1 Provide Multimodal Route Selection ()
6.6.2 Select Vehicle Route ()
6.6.2.1 Calculate Vehicle Route ()
6.6.2.2 Provide Vehicle Route Calculation Data ()
6.6.2.3 Provide Route Segment Data for Other Areas ()
6.6.2.4 Update Vehicle Route Selection Map Data ()
6.6.2.5 Provide ISP Operator Route Parameters Interface ()
6.6.2.6 Calculate Vehicle Probe Data for Guidance ()
6.6.3 Update Other Routes Selection Map Data ()
6.6.4 Select Transit Route ()
6.6.5 Select Other Routes ()
6.7 Provide Driver Personal Services ()
6.7.1 Provide Driver Personal Security ()
6.7.1.1 Build Driver Personal Security Message ()
6.7.1.2 Provide Driver In-vehicle Communications Function ()
6.7.2 Provide On-line Vehicle Guidance ()
6.7.2.1 Provide Vehicle Guidance ()
6.7.2.1.1 Determine In-vehicle Guidance Method ()
6.7.2.1.2 Provide Dynamic In-vehicle Guidance ()
6.7.2.1.3 Provide Autonomous In-vehicle Guidance ()
6.7.2.2 Process Vehicle Location Data ()
6.7.2.3 Provide Driver Guidance Interface ()
6.7.2.4 Update Vehicle Navigable Map Database ()
6.8 Provide Traveller Personal Services ()
6.8.1 Provide On-line Traveller Guidance ()
6.8.1.1 Provide Traveller Guidance ()
6.8.1.1.1 Determine Personal Portable Device Guidance Method **()

Figure 6.63 (Part 26) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

6.8.1.1 Provide Traveller Guidance ()
6.8.1.1.1 Determine Personal Portable Device Guidance Method **()
6.8.1.1.2 Provide Personal Portable Device Dynamic Guidance ()
6.8.1.1.3 Provide Personal Portable Device Autonomous Guidance ()
6.8.1.2 Provide Personal Portable Device Guidance Interface ()
6.8.1.3 Process Personal Portable Device Location Data ()
6.8.1.4 Update Traveller Navigable Map Database ()
6.8.1.5 Provide Traveller Emergency Message Interface ()
6.8.2 Provide Traveller Personal Security ()
6.8.2.1 Build Traveller Personal Security Message ()
6.8.2.2 Provide Traveller Emergency Communications Function ()
6.8.3 Provide Traveller Services at Personal Devices ()
6.8.3.1 Get Traveller Personal Request ()
6.8.3.2 Provide Traveller with Personal Travel Information ()
6.8.3.3 Provide Traveller Personal Interface ()
6.8.3.4 Update Traveller Personal Display Map Data ()
7 Provide Electronic Payment Services ()
7.1 Provide Electronic Toll Payment ()
7.1.1 Process Electronic Toll Payment ()
7.1.1.1 Read Tag Data for Tolls ()
7.1.1.10 Determine Advanced Toll Bill ()
7.1.1.11 Manage Toll Archive Data ()
7.1.1.2 Calculate Vehicle Toll ()
7.1.1.3 Manage Bad Toll Payment Data ()
7.1.1.4 Check for Advanced Tolls Payment ()
7.1.1.5 Bill Driver for Tolls ()
7.1.1.6 Collect Probe Data From Toll Transactions ()
7.1.1.7 Update Toll Price Data ()
7.1.1.8 Register for Advanced Toll Payment ()
7.1.1.9 Manage Toll Financial Processing ()
7.1.2 Produce Roadside Displays ()
7.1.3 Obtain Toll Violator Image ()
7.1.4 Provide Driver Toll Payment Interface ()
7.1.5 Detect Vehicle for Tolls ()
7.1.6 Distribute Advanced Charges and Fares ()
7.1.7 Provide Payment Instrument Interface for Tolls ()
7.2 Provide Electronic Parking Payment **()
7.2.1 Process Electronic Parking Lot Payment ()

Figure 6.64 (Part 27) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

7.2 Provide Electronic Parking Payment **()
7.2.1 Process Electronic Parking Lot Payment ()
7.2.1.1 Read Parking Lot Tag Data ()
7.2.1.10 Determine Advanced Charges ()
7.2.1.2 Calculate Vehicle Parking Lot Charges ()
7.2.1.3 Collect Bad Charge Payment Data ()
7.2.1.4 Check for Advanced Parking Lot Payment ()
7.2.1.5 Bill Driver for Parking Lot Charges ()
7.2.1.6 Manage Parking Lot Financial Processing ()
7.2.1.7 Update Parking Lot Data ()
7.2.1.8 Register for Advanced Parking Lot Payment ()
7.2.1.9 Manage Parking Lot Reservations ()
7.2.2 Produce Parking Lot Displays ()
7.2.3 Obtain Parking Lot Violator Image ()
7.2.4 Provide Driver Parking Lot Payment Interface ()
7.2.5 Detect Vehicle for Parking Lot Payment ()
7.2.6 Distribute Advanced Tolls and Fares ()
7.2.7 Provide Payment Instrument Interface for Parking ()
7.3 Provide Electronic Fare Collection ()
7.3.1 Process Electronic Transit Fare Payment ()
7.3.1.1 Register for Advanced Transit Fare Payment ()
7.3.1.2 Determine Advanced Transit Fares ()
7.3.1.3 Manage Transit Fare Financial Processing ()
7.3.1.4 Check for Advanced Transit Fare Payment ()
7.3.1.5 Bill Transit User for Transit Fare ()
7.3.1.6 Collect Bad Transit Fare Payment Data ()
7.3.1.7 Update Transit Fare Data ()
7.3.2 Distribute Advanced Tolls and Parking Lot Charges ()
7.3.3 Get Transit User Image for Violation ()
7.3.4 Provide Remote Terminal Payment Instrument Interface ()
7.3.5 Provide Transit Vehicle Payment Instrument Interface ()
7.4 Carry-out Centralized Payments Processing ()
7.4.1 Collect Advanced Payments ()
7.4.1.1 Process Commercial Vehicle Payments ()
7.4.1.2 Process Yellow Pages Services Provider Payments ()
7.4.1.3 Process Driver Map Update Payments ()
7.4.1.4 Process Traveller Map Update Payments **()

Figure 6.65 (Part 28) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

7.4.1.2 Process Yellow Pages Services Provider Payments ()
7.4.1.3 Process Driver Map Update Payments ()
7.4.1.4 Process Traveller Map Update Payments **()
7.4.1.5 Process Transit User Other Services Payments ()
7.4.1.6 Process Traveller Trip and Other Services Payments ()
7.4.1.7 Collect Payment Transaction Records ()
7.4.1.8 Process Traveller Rideshare Payments ()
7.4.2 Collect Price Data for CAHS Use ()
7.4.3 Route Traveller Advanced Payments ()
7.5 Provide Payment Instrument Interfaces ()
7.5.1 Provide Vehicle Payment Instrument Interface ()
7.5.2 Provide Transit User Roadside Payment Instrument Interface ()
7.5.3 Provide Personal Payment Instrument Interface ()
7.5.4 Provide Commercial Fleet Payment Instrument Interface ()
7.5.5 Provide Traveller Kiosk Payment Instrument Interface ()
8 Manage Archived Data ()
8.2 Manage Archive ()
8.3 Manage Archive Data Administrator Interface ()
8.4 Coordinate Archives ()
8.5 Process Archived Data User System Requests ()
8.6 Analyze Archive ()
8.7 Process On Demand Archive Requests ()
8.8 Prepare Government Reporting Inputs ()
8.9 Manage Roadside Data Collection ()
9 Satisfy Implementation Requirements ()
8.1 Get Archive Data ()
5.7.3 Provide Disaster Coordination Authority Interface ()

Figure 6.66 (Part 29) Object-oriented visualization of SAHS's Class Backbone Communication with it attributes and Functions in UML

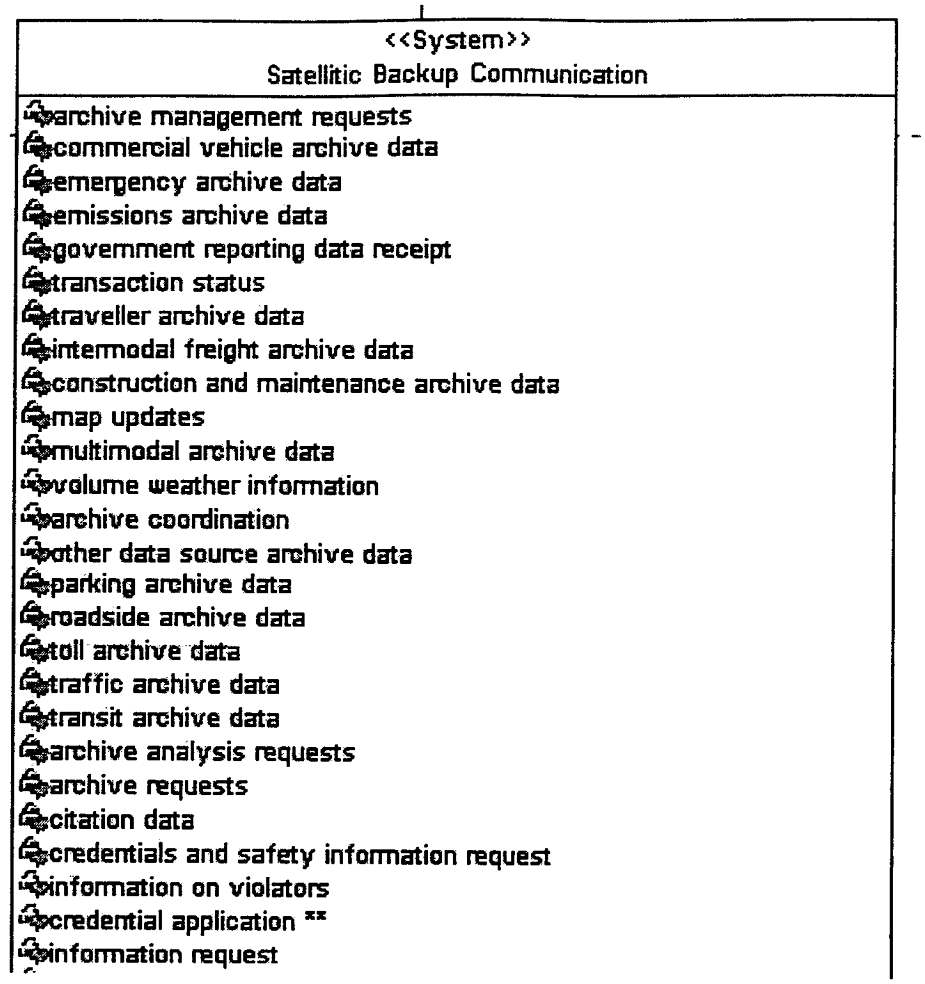

Figure 6.67 (Part 1) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

information on violators
credential application**
information request
tax filing, audit data
Commercial Vehicle Administration Subsystem information exchange
credentials and safety information response
archive status
pollutant levels
pollution data
pollution state data request
pollution data parameters
activity reports
compliance review report
electronic credentials
fleet manager inquiry
driver and vehicle information
intermodal status
on-board vehicle data
vehicle location
container location request
container status request
HAZMAT information request
tax filing, audit data
container delivery confirmation
container pickup request
route plan
freight consignment request **
shipment status request

Figure 6.68 (Part 2) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

route plan
freight consignment request **
shipment status request
intermodal CVO co-ordination
disaster advisories
shelter availability
shelter location
incident information
air quality information
route request
roadway maintenance information
external reports
media information request
road weather prediction
multimodal information
service response
ISP coordination
parking information
parking lot reservation confirmation
traveller profile
traveller request
trip confirmation
trip request
yellow pages request
probe data
toll data
request fare and price information
traffic information **

Figure 6.69 (Part 3) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

traffic information **
demand responsive transit plan
transit and fare schedules
transit incident information
transit request confirmation
transit trip plan
vehicle probe data
provider profile data
travel service info
maintenance operator inputs
maintenance status data
maintenance vehicle status data
roadway micro prediction
device status
environmental conditions
fault reports
roadway information system status
surveillance data
closure coordination
environmental sensor data
maintenance resource request
traffic equipment status
disaster response coordination
disaster response status
incident notification **
emergency dispatch response
emergency vehicle tracking data

Figure 6.70 (Part 4) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

disaster response status
incident notification **
emergency dispatch response
emergency vehicle tracking data
incident command request
incident status
event plans
HAZMAT information
incident information request
medical facility availability
incident report
incident response coordination
emergency notification
current network conditions
resource deployment status
transit emergency data
evacuation support request
transit emergency coordination data
demand responsive transit request
selected routes
transit information request
transit trip request
roadway maintenance information
transit multimodal information
transit parking coordination
secure area surveillance data
request transit information **
traffic information for transit

Figure 6.71 (Part 5) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

secure area surveillance data
request transit information **
traffic information for transit
transit demand management request
transit driver availability
transit fleet manager inputs
maintenance status
transit operator management data
fare and payment status
request for bad tag list
transit traveller request
transit user information
transit vehicle conditions
transit vehicle location data
transit vehicle passenger and use data
transit vehicle schedule performance
evacuation support request
transit multimodal information
Transit Management System Co-ordination
transit operator management data
transit user information
transit vehicle conditions
transit vehicle location data
road network status request
emergency traffic control request
incident response status
remote surveillance control **
resource request

Figure 6.72 (Part 6) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

incident response status
remote surveillance control **
resource request
widearea statistical pollution information
fare and price information
logged special vehicle route
request for traffic information
road network use
environmental sensor data
equipment maintenance status
maintenance dispatch status
maintenance resource response
work zone status
external reports
traffic control coordination
traffic information coordination
parking availability
parking demand management response
railroad advisories
railroad schedules
AHS status
emissions data
freeway control status
HOV data
HRI status
incident data
intersection blockage notification
request for right-of-way **
reversible lane status

Figure 6.73 (Part 7) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

intersection blockage notification
request for right-of-way **
reversible lane status
roadway warning status
signal control status
traffic flow
traffic images
violation detection
toll demand management response
transit system data
registration
toll data request
toll administration requests
Toll Transactions
toll demand management request
CVO weight and presence
credentials information
CVO database update
international border crossing data
safety information
electronic clearance data
lock tag data
on-board safety data
screening data
Commercial Vehicle Check override mode
CVO inspector input
container transfer location request
exit request **
vehicle identification

Figure 6.74 (Part 8) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

container transfer location request
exit request **
vehicle identification
container release status
container availability request
container pickup confirmation
container identification
container location
intermodal terminal operator inputs
parking lot data request
parking reservations request
parking operator inputs
request for performance data
parking demand management request
parking instructions
transit parking lot response
vehicle characteristics
tag data
sensor and surveillance control
local signal preemption request
vehicle pollution criteria
environmental hazards
pollutant levels
device control data
roadway information system data
sensor control **
multimodal crossing status

Figure 6.75 (Part 9) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

roadway information system data
sensor control **
multimodal crossing status
device control data
crossing call
non-vehicular presence
physical presence
arriving train information
track status
roadway characteristics
environmental hazards
weather conditions
traffic characteristics
AHS control information
freeway control data
HRI control data
HRI request
roadway warning data
signal control data
local signal priority request
vehicle signal violation
vehicle speed violation
vehicular presence
AHS vehicle data
toll instructions **
toll operator requests
emergency acknowledge

Figure 6.76 (Part 10) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

AHS vehicle data
toll instructions **
toll operator requests
emergency acknowledge
broadcast information
trip plan
yellow pages information
position fix
payment
personal transit information
traveller inputs
traveller information
secure area characteristics
secure area monitoring support
transit fare payment responses
transit traveller information
transit user inputs
traveller inputs
commercial vehicle measures
border clearance event record
clearance event record
electronic clearance request
lock tag data request
on-board safety request
safety inspection record
screening request **
CVO driver initialization

Figure 6.77 (Part 11) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

screening request **
CVO driver initialization
container transport assignment
fleet to driver update
intermodal status request
location request
container transfer location
entry permission
exit permission
chassis data
cargo data
container status
commercial vehicle data request
emergency dispatch requests
incident command information
suggested route
emergency personnel inputs
maintenance vehicle dispatch
maintenance vehicle measures
maintenance driver inputs
environmental hazards
transit driver inputs
bad tag list
driver instructions
fare management information
request for vehicle measures
transit schedule information **
emergency request

Figure 6.78 (Part 12) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

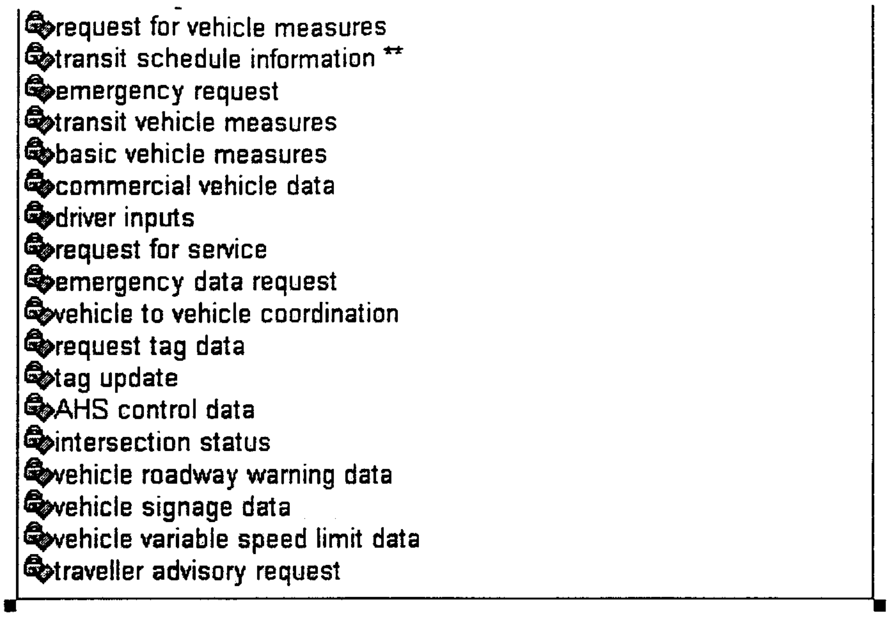

Figure 6.79 (Part 13) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

♦0 Manage CAHS ()
♦1 Manage Traffic ()
♦1.1 Provide Traffic Surveillance ()
♦1.1.1 Process Sensor Data ()
♦ 1.1.1.1 Process Traffic Sensor Data()
♦ 1.1.1.2 Collect and Process Sensor Fault Data ()
♦1.1.1.3 Process Environmental Sensor Data ()
♦1.1.1.4 Manage Data Collection and Monitoring ()
♦1.1.1.5 Process Advanced Roadway Data ()
♦1.1.1.5.1 Collect Roadway Condition Data ()
♦1.1.1.5.2 Provide Other Roadside Inputs ()
♦1.1.1.5.3 Process Roadway Warning Data ()
♦1.1.5.4 Process Road User Protection()
♦1.1.2 Process and Store Traffic Data()
♦1.1.2.1 Process Traffic Data for Storage ()
♦1.1.2.2 Process Traffic Data ()
♦1.1.2.3 Update Data Source Static Data ()
♦1.1.2.4 Monitor HOV lane use ()
♦1.1.2.5 Process Tag/AVL Data for Link Time Data ()
♦1.1.2.6 Process Collected Vehicle Smart Probe Data ()
♦1.1.2.7 Monitor Reversible Lanes ()
♦1.1.2.8 Manage Environmental Data for Traffic()
♦1.1.3 Generate Predictive Traffic Model ()
♦1.1.4 Display and Output Traffic Data ()
♦1.1.4.1 Retrieve Traffic Data ()
♦1.1.4.2 Provide Traffic Operations Personnel Traffic Data Interface ()
♦1.1.4.3 Provide Direct Media Traffic Data Interface ()
♦1.1.4.4 Update Traffic Display Map Data ()
♦1.1.4.5 Provide Media System Traffic Data Interface ()
♦1.1.4.6 Provide Traffic Data Retrieval Interface ()
♦1.1.4.7 Manage Traffic Archive Data ()
♦1.1.4.8 Provide Disaster Command Data ()
♦1.1.5 Exchange data with Other Traffic Centres ()
♦1.1.6 Collect Vehicle Tag Data for Link Time Calculations **()
♦1.1.7 Collect Vehicle Smart Probe Data ()

Figure 6.80 (Part 14) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

1.1.6 Collect Vehicle Tag Data for Link Time Calculations **()
1.1.7 Collect Vehicle Smart Probe Data ()
1.2 Provide Device Control()
1.2.1 Select Strategy ()
1.2.2 Determine Road and Freeway State ()
1.2.2.1 Determine Indicator State for Freeway Management ()
1.2.2.2 Determine Indicator State for Road Management ()
1.2.3 Determine Ramp State ()
1.2.4 Output Control Data ()
1.2.4.1 Output Control Data for Roads ()
1.2.4.2 Output Control Data for Freeways ()
1.2.4.3 Output In-vehicle Signage Data ()
1.2.5 Manage Parking Lot State ()
1.2.5.1 Determine Parking Lot State ()
1.2.5.2 Coordinate Other Parking Data ()
1.2.5.3 Provide Parking Lot Operator Interface ()
1.2.5.4 Determine P+R needs for Transit Management ()
1.2.5.5 Manage Parking Archive Data ()
1.2.5.6 Calculate Parking Lot Occupancy ()
1.2.6 Maintain Static Data for TMC ()
1.2.6.1 Maintain Traffic and Sensor Static Data ()
1.2.6.2 Provide Static Data Store Output Interface ()
1.2.7 Provide Roadside Control Facilities ()
1.2.7.1 Process Indicator Output Data for Roads ()
1.2.7.2 Monitor Roadside Equipment Operation for Faults ()
1.2.7.3 Manage Indicator Preemptions ()
1.2.7.4 Process In-vehicle Signage Data ()
1.2.7.5 Process Indicator Output Data for Freeways()
1.2.7.6 Provide Intersection Collision Avoidance Data ()
1.2.7.7 Process Vehicle Smart Probe Data for Output ()
1.2.7.8 Receive Other Roadside Inputs ()
1.2.7.9 Display Roadway Warnings ()
1.2.8 Collect and Process Indicator Fault Data ()
1.2.8.1 Collect Indicator Fault Data ()
1.2.8.2 Maintain Indicator Fault Data Store ()
1.2.8.3 Provide Indicator Fault Interface for C and M **()
1.2.8.4 Provide Traffic Operations Personnel Indicator Fault Interface ()

Figure 6.81 (Part 15) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

1.2.8.3 Provide Indicator Fault Interface for C and M **()
1.2.8.4 Provide Traffic Operations Personnel Indicator Fault Interface ()
1.3 Manage Incidents ()
1.3.1 Traffic Data Analysis for Incidents ()
1.3.1.1 Analyze Traffic Data for Incidents ()
1.3.1.2 Maintain Static Data for Incident Management ()
1.3.1.3 Process Traffic Images ()
1.3.2 Review and Manage Incident Data ()
1.3.2.1 Store Possible Incident Data ()
1.3.2.2 Review and Classify Possible Incidents ()
1.3.2.3 Review and Classify Planned Events ()
1.3.2.4 Provide Planned Events Store Interface ()
1.3.2.5 Provide Current Incidents Store Interface ()
1.3.3 Respond to Current Incidents ()
1.3.4 Provide Operator Interfaces for Incidents ()
1.3.4.1 Retrieve Incident Data ()
1.3.4.2 Provide Traffic Operations Personnel Incident Data Interface ()
1.3.4.3 Provide Media Incident Data Interface ()
1.3.4.4 Update Incident Display Map Data ()
1.3.4.5 Manage Resources for Incidents ()
1.3.5 Manage Possible Predetermined Responses Store ()
1.3.6 Manage Predetermined Incident Response Data ()
1.3.7 Analyze Incident Response Log ()
1.4 Manage Travel Demand ()
1.4.1 Provide Traffic Operations Personnel Demand Interface ()
1.4.2 Collect Demand Forecast Data ()
1.4.3 Update Demand Display Map Data ()
1.4.4 Implement Demand Management Policy ()
1.4.5 Calculate Forecast Demand ()
1.5 Manage Emissions ()
1.5.1 Provide Traffic Operations Personnel Pollution Data Interface ()
1.5.2 Process Pollution Data ()
1.5.3 Update Pollution Display Map Data ()
1.5.4 Manage Pollution State Data Store ()
1.5.5 Process Vehicle Pollution Data ()
1.5.6 Detect Roadside Pollution Levels ()
1.5.7 Manage Pollution Data Log**()
1.5.8 Manage Pollution Reference Data Store ()

Figure 6.82 (Part 16) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

1.5.7 Manage Pollution Data Log**()
1.5.8 Manage Pollution Reference Data Store ()
1.5.9 Manage Pollution Archive Data ()
1.6 Manage Highway Rail Intersections ()
1.6.1 Manage HRI Vehicle Traffic ()
1.6.1.1 Detect Roadway Events ()
1.6.1.2 Activate HRI Device Controls ()
1.6.1.2.1 Control HRI Traffic Signals ()
1.6.1.2.2 Control HRI Warnings and Barriers ()
1.6.1.2.3 Provide SSR Device Controls ()
1.6.1.2.4 Provide HSR Device Controls ()
1.6.1.2.5 Manage Device Control ()
1.6.1.2.6 Maintain Device State (()
1.6.1.3 Perform Equipment Self-Test ()
1.6.1.4 Provide Advisories and Alerts ()
1.6.1.4.1 Generate Alerts and Advisories ()
1.6.1.4.2 Provide Closure Parameters ()
1.6.1.4.3 Report Alerts and Advisories ()
1.6.1.4.4 Report HRI Status on Approach ()
1.6.1.5 Detect HRI Hazards ()
1.6.1.6 Provide Advance Warnings ()
1.6.1.6.1 Close HRI on Detection ()
1.6.1.6.2 Detect Imminent Vehicle/Train Collision ()
1.6.1.7 Execute Local Control Strategy ()
1.6.1.7.1 Control Vehicle Traffic at Passive HRI ()
1.6.1.7.2 Control Vehicle Traffic at Active HRI ()
1.6.1.7.3 Close HRI on Command ()
1.6.2 Interact with Rail Operations ()
1.6.2.1 Exchange Data with Rail Operations ()
1.6.2.2 Manage Alerts and Advisories ()
1.6.2.3 Manage Rail Traffic Control Data ()
1.6.3 Manage HRI Rail Traffic ()
1.6.3.1 Interact with Wayside Systems ()
1.6.3.2 Advise and Protect Train Crews ()
1.6.3.3 Provide ATS Alerts ()
1.6.4 Interact with Vehicle Traffic Management**()
1.6.4.1 Manage HRI Closures ()
1.6.4.2 Exchange Data with Traffic Management ()

Figure 6.83 (Part 17) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

- 1.6.4 Interact with Vehicle Traffic Management**()
- 1.6.4.1 Manage HRI Closures ()
- 1.6.4.2 Exchange Data with Traffic Management ()
- 1.6.5 Monitor HRI Status ()
- 1.6.5.1 Provide Interactive Interface ()
- 1.6.5.2 Determine HRI Status ()
- 1.6.5.3 Maintain HRI Closure Data ()
- 1.7 Manage Operations and Maintenance ()
- 1.7.1 Manage Operations and Maintenance Data ()
- 1.7.1.1 Gather O&M Data ()
- 1.7.1.2 Manage O&M Archive Data ()
- 1.7.1.3 Manage O&M Assets ()
- 1.7.1.4 Manage Environmental Data for Operations and Maintenance ()
- 1.7.1.4.1 Gather and Process Environmental Data ()
- 1.7.1.4.2 Create Road Forecasts ()
- 1.7.1.4.3 Fuse Environmental Data ()
- 1.7.2 Manage Maintenance Activity ()
- 1.7.2.1 Schedule Maintenance Activity ()
- 1.7.2.2 Manage and Control On-board Maintenance Systems ()
- 1.7.2.3 Manage Maintenance Operator Interface ()
- 1.7.2.4 Manage Maintenance Driver Interface ()
- 1.7.2.5 Collect Environmental Data On-board ()
- 1.7.3 Control Roadway Maintenance Devices ()
- 1.7.4 Disseminate O&M Information ()
- 1.8 Manage Traffic Enforcement ()
- 1.8.1 Detect and Classify Speed Violations ()
- 1.8.2 Collect and Verify Speed Violations ()
- 1.8.3 Post Speed Limit ()
- 1.8.4 Set Speed Limit ()
- 1.8.5 Establish Violation Parameters ()
- 1.8.6 Detect and Classify Signal Violations ()
- 1.8.7 Collect and Verify Signal Violations ()
- 2 Manage Commercial Vehicles ()
- 2.1 Manage Commercial Vehicle Fleet Operations ()
- 2.1.1 Manage Commercial Fleet Electronic Credentials and Tax Filing ()
- 2.1.2 Provide Commercial Fleet Static Route ()
- 2.1.3 Provide Flt Mgr Electronic Credentials and Tax Filing Interface**()

Figure 6.84 (Part 18) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

2.1.3 Provide Flt Mgr Electronic Credentials and Tax Filing Interface**()
2.1.4 Provide Fleet Manager Commercial Vehicle Communications ()
2.1.5 Provide Commercial Vehicle Driver Routing Interface ()
2.1.6 Manage Driver Instruction Store ()
2.2 Manage Commercial Vehicle Driver Operations ()
2.2.1 Manage CV Electronic Credential and Tax Filing Interface ()
2.2.2 Provide Vehicle Static Route ()
2.2.3 Provide CV Driver Electronic Credential and Tax Filing Interface ()
2.2.4 Provide Commercial Vehicle Driver Communications ()
2.3 Provide Commercial Vehicle Roadside Facilities ()
2.3.1 Produce Commercial Vehicle Driver Message at Roadside ()
2.3.2 Provide Commercial Vehicle Clearance Screening ()
2.3.2.1 Administer Commercial Vehicle Roadside Credentials Database ()
2.3.2.2 Process Screening Transactions ()
2.3.3 Provide Roadside Commercial Vehicle Safety ()
2.3.3.1 Provide Commercial Vehicle Checkstation Communications ()
2.3.3.2 Provide Commercial Vehicle Inspector Handheld Terminal Interface ()
2.3.3.3 Administer Commercial Vehicle Roadside Safety Datadase ()
2.3.3.4 Carry-out Commercial Vehicle Roadside Safety Screening ()
2.3.3.5 Carry-out Commercial Vehicle Roadside Inspection ()
2.3.4 Detect Commercial Vehicle ()
2.3.5 Provide Commercial Vehicle RoadsideOperator Interface ()
2.3.6 Provide Commercial Vehicle Reports ()
2.3.7 Produce Commercial Vehicle Driver Message on Vehicle ()
2.3.8 Provide Commercial Vehicle Border Screening ()
2.4 Provide Commercial Vehicle Data Collection ()
2.4.1 Communicate Commercial Vehicle On-board Data to Roadside ()
2.4.2 Collect On-board Commercial Vehicle Sensor Data ()
2.4.3 Analyze Commercial Vehicle On-board Data ()
2.4.4 Provide Commercial Vehicle Driver Interface ()
2.4.5 Communicate Commercial Vehicle On-board Data to Vehicle Manager ()
2.4.6 Provide Commercial Vehicle On-board Data Store Interface ()
2.5 Administer Commercial Vehicles ()
2.5.1 Manage Commercial Vehicle Trips and Clearances ()
2.5.2 Obtain Electronic Credential and Tax Filing Payment ()
2.5.3 Update Permits and Duties Store **()
2.5.4 Communicate with Other Commercial Vehicle Administration System ()

Figure 6.85 (Part 19) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

2.5.2 Obtain Electronic Credential and Tax Filing Payment ()
2.5.3 Update Permits and Duties Store **()
2.5.4 Communicate with Other Commercial Vehicle Administration System ()
2.5.5 Manage Commercial Vehicle Credentials and Enrollment ()
2.5.6 Output Commercial Vehicle Enrollment Data to Roadside Facilities ()
2.5.7 Process Commercial Vehicle Violations ()
2.5.8 Process Data Received from Roadside Facilities ()
2.5.9 Manage Commercial Vehicle Archive Data ()
2.6 Provide Commercial Vehicle On-board Data ()
2.6.1 Provide Commercial Vehicle Manager Tag Data Interface ()
2.6.2 Transmit Commercial Vehicle Tag Data ()
2.6.3 Provide Commercial Driver Tag Data Interface ()
2.6.4 Provide Lock Tag Data Interface ()
2.6.5 Manage Commercial Vehicle Tag Data Store ()
2.7 Manage Cargo ()
2.7.1 Manage Intermodal Terminal Interface ()
2.7.1.1 Manage Terminal Access ()
2.7.1.2 Manage Container Pickup and Delivery ()
2.7.1.3 Manage Container Release ()
2.7.1.4 Provide Intermodal Terminal Operator I/F ()
2.7.1.5 Provide Container Tracking ()
2.7.2 Manage Intermodal Container ()
2.7.2.1 Determine Container Status ()
2.7.2.2 Determine Cargo Status ()
2.7.2.3 Provide Cargo Status Interface ()
2.7.2.4 Provide Container Status Interface ()
2.7.2.5 Manage Customs Interface ()
2.7.3 Manage On-Board Intermodal Applications ()
2.7.3.1 Provide Driver Interface for Intermodal Freight Dispatch()
2.7.3.2 Monitor Container Status ()
2.7.3.3 Monitor Chassis Status ()
2.7.3.4 Manage On-Board Facility Access ()
2.7.4 Manage Intermodal Dispatch ()
2.7.4.1 Monitor Intermodal Elements ()
2.7.4.2 Provide Fleet Manager Interface for Intermodal ()
2.7.4.3 Manage Intermodal Customer Interface**()
2.7.4.4 Manage Other Intermodal FMS Interface ()
2.7.4.5 Manage Distribution and Logistics Management Provider Interface ()

Figure 6.86 (Part 20) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

2.7.4.2 Provide Fleet Manager Interface for Intermodal ()
2.7.4.3 Manage Intermodal Customer Interface**()
2.7.4.4 Manage Other Intermodal FMS Interface ()
2.7.4.5 Manage Distribution and Logistics Management Provider Interface ()
2.7.4.6 Manage Freight Consolidation Station Interface ()
3 Provide Vehicle Monitoring and Control ()
3.1 Monitor Vehicle Status ()
3.1.1 Produce Collision and Crash Avoidance Data ()
3.1.2 Carry-out Safety Analysis ()
3.1.3 Process Vehicle On-board Data ()
3.2 Provide Automatic Vehicle Operation ()
3.2.1 Provide Driver Interface ()
3.2.2 Provide AHS Control ()
3.2.3 Provide Vehicle Control ()
3.2.3.1 Provide Command Interface ()
3.2.3.2 Manage Platoon Following ()
3.2.3.3 Process data for Vehicle Actuators ()
3.2.3.4 Provide Servo Controls ()
3.2.3.4.1 Provide Speed Servo Control ()
3.2.3.4.2 Provide Headway Servo Control ()
3.2.3.4.3 Provide Lane Servo Control ()
3.2.3.4.4 Provide Change Lane Servo Control ()
3.2.3.4.5 Provide Vehicle Control Data Interface ()
3.2.3.5 Process Vehicle Sensor Data ()
3.2.3.6 Communicate with other Platoon Vehicles ()
3.2.4 Process Sensor Data for AHS input ()
3.2.5 Check Vehicle for AHS eligibility ()
3.2.6 Manage AHS Check-in and Check-out ()
3.2.7 Manage AHS Operations ()
3.3 Provide Automatic Emergency Notification ()
3.3.1 Provide Cargo Data for Incident Notification ()
3.3.2 Provide Communications Function ()
3.3.3 Build Automatic Collision Notification Message ()
3.4 Enhance Driver's Vision ()
4 Manage Transit ()
4.1 Operate Vehicles and Facilities ()
4.1.1 Process Transit Vehicle Sensor Trip Data**()
4.1.2 Determine Transit Vehicle Deviation and Corrections ()

Figure 6.87 (Part 21) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

4.1.1 Process Transit Vehicle Sensor Trip Data**()
4.1.2 Determine Transit Vehicle Deviation and Corrections ()
4.1.2.1 Determine Transit Vehicle Deviation and ETA ()
4.1.2.2 Determine Transit Vehicle Corrective Instructions ()
4.1.2.3 Provide Transit Vehicle Driver Interface ()
4.1.2.4 Provide Transit Vehicle Correction Data Output Interface ()
4.1.2.5 Request Transit Vehicle Preemptions ()
4.1.3 Provide Transit Vehicle Location Data ()
4.1.4 Manage Transit Vehicle Deviations ()
4.1.5 Provide Transit Vehicle Status Information ()
4.1.6 Manage Transit Vehicle Operations Data ()
4.1.7 Manage Connection Protection ()
4.1.7.1 Manage Connections with External Systems ()
4.1.7.2 Provide Transit Vehicle Deviation Data Output Interface ()
4.1.7.3 Manage Individual Service Requests ()
4.1.8 Provide Transit Operations Data Distribution Interface ()
4.1.9 Process Transit Vehicle Sensor Maintenance Data ()
4.2 Plan and Schedule Transit Services ()
4.2.1 Provide Demand Responsive Transit Service ()
4.2.1.1 Process Demand Responsive Transit Trip Request ()
4.2.1.2 Compute Demand Responsive Transit Vehicle Availability ()
4.2.1.3 Generate Demand Responsive Transit Schedule and Routes ()
4.2.1.4 Confirm Demand Responsive Transit Schedule and Route ()
4.2.1.5 Process Demand Responsive Transit Vehicle Availability Data ()
4.2.1.6 Provide Demand Responsive Transit Driver Interface ()
4.2.2 Provide Transit Plans Store Interface ()
4.2.3 Generate Transit Routes and Schedules ()
4.2.3.1 Generate Transit Routes ()
4.2.3.2 Generate Schedules ()
4.2.3.3 Produce Transit Service Data for External Use ()
4.2.3.4 Provide Transit Fleet Manager Interface for Services Generation ()
4.2.3.5 Manage Transit Operational Data Store ()
4.2.3.6 Produce Transit Service Data for Manage Transit Use ()
4.2.3.7 Provide Interface for Other TRM Data ()
4.2.3.8 Provide Interface for Transit Service Raw Data ()
4.2.3.9 Update Transit Map Data ()
4.2.4 Manage Transit Archive Data **()
4.3 Schedule Transit Vehicle Maintenance ()

Figure 6.88 (Part 22) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

4.2.3.9 Update Transit Map Data ()
4.2.4 Manage Transit Archive Data **()
4.3 Schedule Transit Vehicle Maintenance ()
4.3.1 Monitor Transit Vehicle Condition ()
4.3.2 Generate Transit Vehicle Maintenance Schedules ()
4.3.3 Generate Technician Work Assignments ()
4.3.4 Monitor And Verify Maintenance Activity ()
4.3.5 Report Transit Vehicle Information (()
4.3.6 Update Transit Vehicle Information ()
4.3.7 Manage Transit Vehicle Operations Data Store ()
4.4 Support Security and Coordination ()
4.4.1 Provide Transit Security and Emergency Management ()
4.4.1.1 Manage Transit Security ()
4.4.1.2 Manage Transit Emergencies ()
4.4.1.3 Provide Transit System Operator Security Interface ()
4.4.1.4 Provide Transit External Interface for Emergencies ()
4.4.1.5 Provide Transit Driver Interface for Emergencies ()
4.4.1.6 Collect Transit Vehicle Emergency Information ()
4.4.1.7 Monitor Secure Area ()
4.4.1.8 Report Traveller Emergencies ()
4.4.2 Coordinate Multiple Agency Responses to Incidents ()
4.4.3 Generate Responses for Incidents ()
4.4.4 Coordinate Transit Disaster Response ()
4.5 Generate Transit Driver Schedules ()
4.5.1 Assess Transit Driver Performance ()
4.5.2 Assess Transit Driver Availability ()
4.5.3 Access Transit Driver Cost Effectiveness ()
4.5.4 Assess Transit Driver Eligibility ()
4.5.5 Generate Transit Driver Route Assignments ()
4.5.6 Update Transit Driver Information ()
4.5.7 Report Transit Driver Information ()
4.5.8 Provide Transit Driver Information Store Interface ()
4.6 Collect Transit Fares in the Vehicle ()
4.6.1 Detect Transit User on Vehicle ()
4.6.2 Determine Transit User Needs on Vehicle ()
4.6.3 Determine Transit Fare on Vehicle ()
4.6.4 Manage Transit Fare Billing on Vehicle **()
4.6.5 Provide Transit User Fare Payment Interface on Vehicle ()

Figure 6.89 (Part 23) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

4.6.4 Manage Transit Fare Billing on Vehicle **()
4.6.5 Provide Transit User Fare Payment Interface on Vehicle ()
4.6.6 Update Transit Vehicle Fare Data ()
4.6.7 Provide Transit Vehicle Passenger Data ()
4.6.8 Manage Transit Vehicle Advanced Payments ()
4.7 Provide Transit User Roadside Facilities ()
4.7.1 Provide Transit User Roadside Information ()
4.7.1.1 Provide Transit User Roadside Data Interface ()
4.7.1.2 Provide Transit User Roadside Vehicle Data Interface ()
4.7.2 Collect Transit Fares at the Roadside ()
4.7.2.1 Detect Transit User at Roadside ()
4.7.2.2 Determine Transit User Needs at Roadside ()
4.7.2.3 Determine Transit Fare at Roadside ()
4.7.2.4 Manage Transit Fare Billing at Roadside ()
4.7.2.5 Provide Transit User Roadside Fare Interface ()
4.7.2.6 Update Roadside Transit Fare Data ()
4.7.2.7 Provide Transit Roadside Passenger Data ()
5 Manage Emergency Services ()
5.1 Provide Emergency Service Allocation ()
5.1.1 Identify Emergencies from Inputs ()
5.1.2 Determine Coordinated Response Plan ()
5.1.3 Communicate Emergency Status ()
5.1.4 Manage Emergency Response ()
5.1.5 Manage Emergency Service Allocation Store ()
5.1.6 Process Mayday Messages ()
5.2 Provide Operator Interface for Emergency Data ()
5.3 Manage Emergency Vehicles ()
5.3.1 Select Response Mode ()
5.3.2 Dispatch Vehicle ()
5.3.3 Track Vehicle ()
5.3.4 Assess Response Status ()
5.3.5 Provide Emergency Personnel Interface ()
5.3.6 Maintain Vehicle Status ()
5.3.7 Provide Emergency Vehicle Route ()
5.4 Provide Law Enforcement Allocation ()
5.4.1 Process TM Detected Violations ()
5.4.2 Process Violations for Tolls ()
5.4.3 Process Parking Lot Violations**()

Figure 6.90 (Part 24) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

5.4.2 Process Violations for Tolls ()
5.4.3 Process Parking Lot Violations**()
5.4.4 Process Fare Payment Violations ()
5.4.5 Process Vehicle Fare Collection Violations ()
5.4.6 Process CV Violations ()
5.4.7 Process Roadside Fare Collection Violations ()
5.5 Update Emergency Display Map Data ()
5.6 Manage Emergency Services Data ()
5.7 Co-ordinate Disaster Response ()
5.7.1 Collect Disaster Response Data ()
5.7.2 Provide Medical Facility Interface ()
6 Provide Driver and Traveller Services ()
6.1 Provide Trip Planning Services ()
6.1.1 Provide Trip Planning Information to Traveller ()
6.1.2 Confirm Traveller's Trip Plan ()
6.1.3 Manage Multimodal Service Provider Interface ()
6.1.4 Provide ISP Operator Interface for Trip Planning Parameters ()
6.1.5 Collect Service Requests and Confirmation for Archive ()
6.1.6 Manage Traveller Info Archive Data ()
6.2 Provide Information Services ()
6.2.1 Provide Advisory and Broadcast Data ()
6.2.1.1 Collect Traffic Data for Advisory Messages ()
6.2.1.2 Provide Traffic and Transit Advisory Messages ()
6.2.1.3 Collect Transit Data for Advisory Messages ()
6.2.1.4 Provide Traffic and Transit Broadcast Messages ()
6.2.1.5 Provide ISP Operator Broadcast Parameters Interface ()
6.2.1.6 Provide Transit Advisory Data On Vehicle ()
6.2.2 Prepare and Output In-vehicle Displays ()
6.2.3 Provide Transit User Advisory Interface ()
6.2.4 Collect Yellow Pages Data ()
6.2.5 Provide Driver Interface ()
6.2.6 Provide Yellow Pages Data and Reservations ()
6.3 Provide Traveller Services at Kiosks ()
6.3.1 Get Traveller Request ()
6.3.2 Inform Traveller ()
6.3.3 Provide Traveller Kiosk Interface ()
6.3.4 Update Traveller Display Map Data at Kiosk ()
6.4 Manage Ridesharing**()

Figure 6.91 (Part 25) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

6.4 Manage Ridesharing**()
6.4.1 Screen Rider Requests ()
6.4.2 Match Rider and Provider ()
6.4.3 Report Ride Match Results to Requestor ()
6.4.4 Confirm Traveller Rideshare Request ()
6.5 Manage Yellow Pages Services ()
6.5.1 Collect and Update Traveller Information ()
6.5.2 Provide Traveller Yellow Pages Information and Reservations ()
6.5.3 Register Yellow Pages Service Providers ()
6.6 Provide Guidance and Trip Planning Services ()
6.6.1 Provide Multimodal Route Selection ()
6.6.2 Select Vehicle Route ()
6.6.2.1 Calculate Vehicle Route ()
6.6.2.2 Provide Vehicle Route Calculation Data ()
6.6.2.3 Provide Route Segment Data for Other Areas ()
6.6.2.4 Update Vehicle Route Selection Map Data ()
6.6.2.5 Provide ISP Operator Route Parameters Interface ()
6.6.2.6 Calculate Vehicle Probe Data for Guidance ()
6.6.3 Update Other Routes Selection Map Data ()
6.6.4 Select Transit Route ()
6.6.5 Select Other Routes ()
6.7 Provide Driver Personal Services ()
6.7.1 Provide Driver Personal Security ()
6.7.1.1 Build Driver Personal Security Message ()
6.7.1.2 Provide Driver In-vehicle Communications Function ()
6.7.2 Provide On-line Vehicle Guidance ()
6.7.2.1 Provide Vehicle Guidance ()
6.7.2.1.1 Determine In-vehicle Guidance Method ()
6.7.2.1.2 Provide Dynamic In-vehicle Guidance ()
6.7.2.1.3 Provide Autonomous In-vehicle Guidance ()
6.7.2.2 Process Vehicle Location Data ()
6.7.2.3 Provide Driver Guidance Interface ()
6.7.2.4 Update Vehicle Navigable Map Database ()
6.8 Provide Traveller Personal Services ()
6.8.1 Provide On-line Traveller Guidance ()
6.8.1.1 Provide Traveller Guidance ()
6.8.1.1.1 Determine Personal Portable Device Guidance Method **()

Figure 6.92 (Part 26) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

0.8.1.1.1 Determine Personal Portable Device Guidance Method **()
0.8.1.1.2 Provide Personal Portable Device Dynamic Guidance ()
0.8.1.1.3 Provide Personal Portable Device Autonomous Guidance ()
0.8.1.2 Provide Personal Portable Device Guidance Interface ()
0.8.1.3 Process Personal Portable Device Location Data ()
0.8.1.4 Update Traveller Navigable Map Database ()
0.8.1.5 Provide Traveller Emergency Message Interface ()
0.8.2 Provide Traveller Personal Security ()
0.8.2.1 Build Traveller Personal Security Message ()
0.8.2.2 Provide Traveller Emergency Communications Function ()
0.8.3 Provide Traveller Services at Personal Devices ()
0.8.3.1 Get Traveller Personal Request ()
0.8.3.2 Provide Traveller with Personal Travel Information ()
0.8.3.3 Provide Traveller Personal Interface ()
0.8.3.4 Update Traveller Personal Display Map Data ()
7 Provide Electronic Payment Services ()
7.1 Provide Electronic Toll Payment ()
7.1.1 Process Electronic Toll Payment ()
7.1.1.1 Read Tag Data for Tolls ()
7.1.1.10 Determine Advanced Toll Bill ()
7.1.1.11 Manage Toll Archive Data ()
7.1.1.2 Calculate Vehicle Toll ()
7.1.1.3 Manage Bad Toll Payment Data ()
7.1.1.4 Check for Advanced Tolls Payment ()
7.1.1.5 Bill Driver for Tolls ()
7.1.1.6 Collect Probe Data From Toll Transactions ()
7.1.1.7 Update Toll Price Data ()
7.1.1.8 Register for Advanced Toll Payment ()
7.1.1.9 Manage Toll Financial Processing ()
7.1.2 Produce Roadside Displays ()
7.1.3 Obtain Toll Violator Image ()
7.1.4 Provide Driver Toll Payment Interface ()
7.1.5 Detect Vehicle for Tolls ()
7.1.6 Distribute Advanced Charges and Fares ()
7.1.7 Provide Payment Instrument Interface for Tolls ()
7.2 Provide Electronic Parking Payment **()
7.2.1 Process Electronic Parking Lot Payment ()

Figure 6.93 (Part 27) Object Oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

7.2 Provide Electronic Parking Payment **()
7.2.1 Process Electronic Parking Lot Payment ()
7.2.1.1 Read Parking Lot Tag Data ()
7.2.1.10 Determine Advanced Charges ()
7.2.1.2 Calculate Vehicle Parking Lot Charges ()
7.2.1.3 Collect Bad Charge Payment Data ()
7.2.1.4 Check for Advanced Parking Lot Payment ()
7.2.1.5 Bill Driver for Parking Lot Charges ()
7.2.1.6 Manage Parking Lot Financial Processing ()
7.2.1.7 Update Parking Lot Data ()
7.2.1.8 Register for Advanced Parking Lot Payment ()
7.2.1.9 Manage Parking Lot Reservations ()
7.2.2 Produce Parking Lot Displays ()
7.2.3 Obtain Parking Lot Violator Image ()
7.2.4 Provide Driver Parking Lot Payment Interface ()
7.2.5 Detect Vehicle for Parking Lot Payment ()
7.2.6 Distribute Advanced Tolls and Fares ()
7.2.7 Provide Payment Instrument Interface for Parking ()
7.3 Provide Electronic Fare Collection ()
7.3.1 Process Electronic Transit Fare Payment ()
7.3.1.1 Register for Advanced Transit Fare Payment ()
7.3.1.2 Determine Advanced Transit Fares ()
7.3.1.3 Manage Transit Fare Financial Processing ()
7.3.1.4 Check for Advanced Transit Fare Payment ()
7.3.1.5 Bill Transit User for Transit Fare ()
7.3.1.6 Collect Bad Transit Fare Payment Data ()
7.3.1.7 Update Transit Fare Data ()
7.3.2 Distribute Advanced Tolls and Parking Lot Charges ()
7.3.3 Get Transit User Image for Violation ()
7.3.4 Provide Remote Terminal Payment Instrument Interface ()
7.3.5 Provide Transit Vehicle Payment Instrument Interface ()
7.4 Carry-out Centralized Payments Processing ()
7.4.1 Collect Advanced Payments ()
7.4.1.1 Process Commercial Vehicle Payments ()
7.4.1.2 Process Yellow Pages Services Provider Payments ()
7.4.1.3 Process Driver Map Update Payments ()
7.4.1.4 Process Traveller Map Update Payments **()

Figure 6.94 (Part 28) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

7.4.1.2 Process Yellow Pages Services Provider Payments ()
7.4.1.3 Process Driver Map Update Payments ()
7.4.1.4 Process Traveller Map Update Payments ()
7.4.1.5 Process Transit User Other Services Payments ()
7.4.1.6 Process Traveller Trip and Other Services Payments ()
7.4.1.7 Collect Payment Transaction Records ()
7.4.1.8 Process Traveller Rideshare Payments ()
7.4.2 Collect Price Data for CAHS Use ()
7.4.3 Route Traveller Advanced Payments ()
7.5 Provide Payment Instrument Interfaces ()
7.5.1 Provide Vehicle Payment Instrument Interface ()
7.5.2 Provide Transit User Roadside Payment Instrument Interface ()
7.5.3 Provide Personal Payment Instrument Interface ()
7.5.4 Provide Commercial Fleet Payment Instrument Interface ()
7.5.5 Provide Traveller Kiosk Payment Instrument Interface ()
8 Manage Archived Data ()
8.2 Manage Archive ()
8.3 Manage Archive Data Administrator Interface ()
8.4 Coordinate Archives ()
8.5 Process Archived Data User System Requests ()
8.6 Analyze Archive ()
8.7 Process On Demand Archive Requests ()
8.8 Prepare Government Reporting Inputs ()
8.9 Manage Roadside Data Collection ()
9 Satisfy Implementation Requirements ()
8.1 Get Archive Data ()
5.7.3 Provide Disaster Coordination Authority Interface ()

Figure 6.95 (Part 29) Object-oriented visualization of SAHS's Class Satellitic Backup Communication with it attributes and Functions in UML

6.5.6 Finding of SAHS Visual Analysis

In SAHS visual analysis, we monitored the participation of backbone communication and backup communication classes. This monitoring is clearly visible in the sequence diagrams. The close examining of the extracted examples are as follows:

1. In Figure 6.28, an object-oriented visualization of Sequence, diagram 4A.4 Support Security and Coordination for SAHS in UML. It can be noted the backbone communication class is active in providing a successful scenario. **What will happen if a malfunctioning occurs in the backbone communication class?** The answer will be determined in Figure 6.33, were the backup communication class is taking over the communication functions in the case of malfunction of the backbone class. We are still having a successful scenario. This is a proof that, **our solution for backup communication is adequate.**

2. In Figure 6.29 Object-oriented visualization of Sequence Diagram, 6A.2 Provide Information Services for SAHS in UML. It can be noted the backbone communication class is active in providing a successful scenario. **What will happen if a malfunctioning occurs in backbone communication class?** The answer will come in Figure 6.34, were the backup communication class is taking over the communication functions in the case of malfunction of the backbone class. We are still having a successful scenario. This is a proof that, **our solution for backup communication is adequate.**

3. In Figure 6.30 Object-oriented visualization of Sequence Diagram, 6A.8 Provide Traveller Personal Services for SAHS in UML. It can be noted the backbone communication class is active in providing a successful scenario. **What will happen if a malfunctioning occurs in backbone communication class?** The answer will come in Figure 6.35, were the backup communication class is taking over the communication functions in the case of

malfunction of the backbone class. We are still having a successful scenario. This is a proof that, **our solution for backup communication is adequate.**

4. In Figure 6.31 Object-oriented visualization of Sequence Diagram 7A.2 Provide Electronic Parking Payment for SAHS in UML. It can be noted the backbone communication class is active in providing a successful scenario. **What will happen if a malfunctioning occurs in backbone communication class?** The answer will be determined in Figure 6.36, were the backup communication class is taking over the communication functions in the case of malfunctioning of the backbone class. We are still having a successful scenario. This is a proof that, **our solution for backup communication is adequate.**

5. In. Figure 6.32 Object-oriented visualization of Sequence Diagram 7A.4 Carry-out Centralized Payments Processing for SAHS in UML It can be noted the backbone communication class is active in providing a successful scenario. **What will happen if a malfunctioning occurs in backbone communication class?** The answer will be determined in Figure 6.37, were the backup communication class is taking over the communication functions in the case of malfunction of the backbone class. We are still having a successful scenario. This is a proof that, **our solution for backup communication is adequate.**

6. The Object-oriented Rational Rose model of SAHS is a clear example of stability and flexibility. We were able to integrate a new communication system, and a backup communication system with no major impact on the architecture it self. The change is only localized, thus the whole architecture is stable. **This is a proof that Object-oriented architecture provides stability and flexibility.**

Chapter 7

Object Oriented ITSC Architecture in a Communication Prospective Within the SAHS

In this chapter, we will have an overview of the object-oriented communication component within the ITSC architecture, which is present within the SAHS. This system has two basic classes that are responsible for the communication activities within the architecture i.e. backbone communication class and backup communication class that will take responsibility in case of malfunctioning of the backbone communication class.

7.1 Backbone Communication Systems

The backbone communication System class is a general representation of the four communication systems channels that were available for CAHS. The four classes that are included in the backbone communication class are as follows:

- Wide Area Wireless Communication Class
- Vehicle-to-Vehicle Communications Class
- Dedicated Short Range Communications Class
- Wireline Communications Class

The backbone communication class diagram is illustrated in Figure 7.1.

7.2 Backup Communication Systems

The backup communication system class is a general representation of the two communication systems channels that were introduced in SAHS. The two classes that are included in the backup communication class are as follows:

- Intelligent Digital Modem Class
- Stratospheric Satellite Class

The backup communication class diagram is illustrated in Figure 7.2.

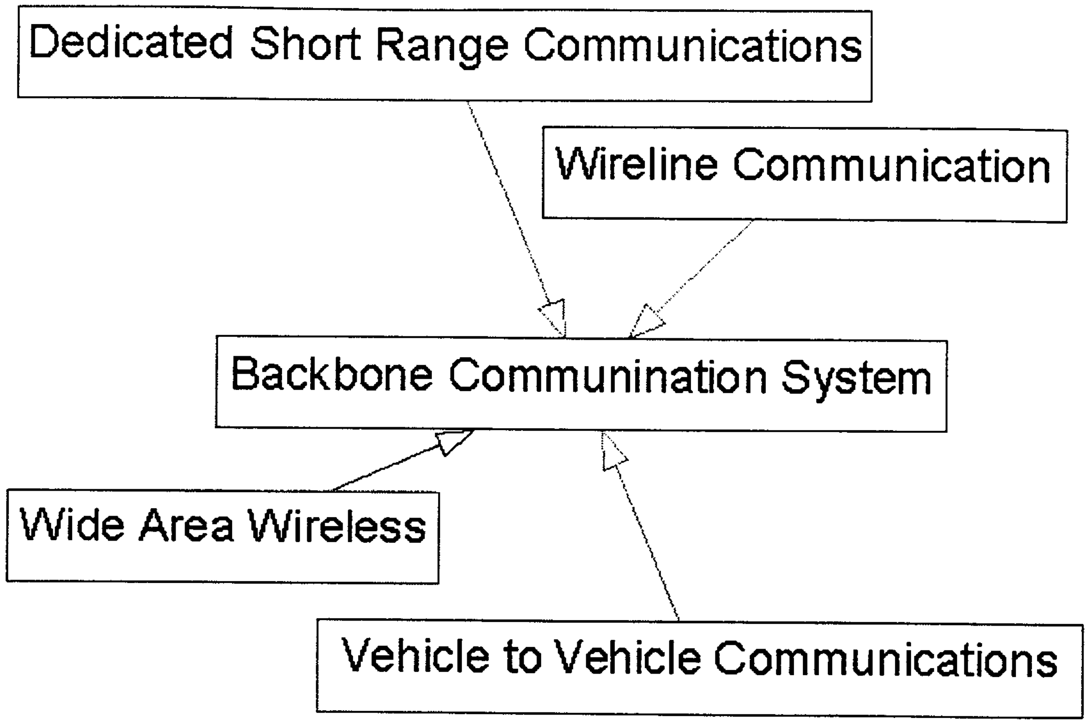

Figure 7.1 The Backbone communication class diagram

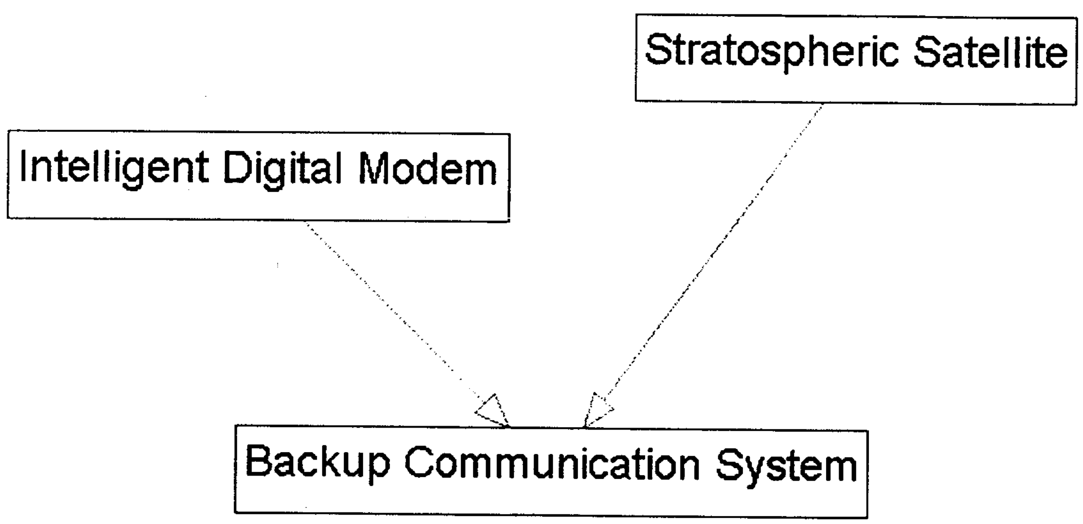

Figure 7.2 The Backup (communication Class) diagram

7.3 SAHS Communication Systems

SAHS communication is an integration of the backbone communication system with the backup communication system that are visualized and modeled by the backbone communication class and backup communication class. The integration is captured in Figure 7.3.

7.4 Intelligent Digital Modem Concept

One of the concepts that was introduced in this dissertation is the Intelligent digital modem concepts. Figure 7.4 represents a flow chart that explains part of its functionality, During a mobile communication session in a Satellitic vehicle, the control can be time out, which is the difference in time between sending and receiving the acknowledgment message. If the time out is not reached, the Intelligent Digital Modem will continue sending, through the mobile unit. In case time out reaches the limit, it will shift the transmitting through the Satellitic communication unit, and continue to monitor while sending through the Satellitic communication unit. If the mobile resumes, it will shift the communication back to the mobile. In case that a time out is reached in the Satellitic communication unit, the Intelligent Digital Modem will send a warning message to the Driver to take over the driving function manually.

Communication between the ITSC actors will be performed using the intelligent digital modem (IDM). To ensure stability and adequate performance of the ITSC requirements, IDM systems will be designed to perform communication tasks through the main Backbone Communication System, and in the case of backbone communication malfunction, through the Satellitic Backup Communication System. IDM will be installed on each Satellitic Vehicle that uses an automatic traffic control system. Its purpose is to:

1. Monitor communications both to and from the Satellitic Vehicle to determine when any of these communications are out of service. This may include but is not restricted to GPS

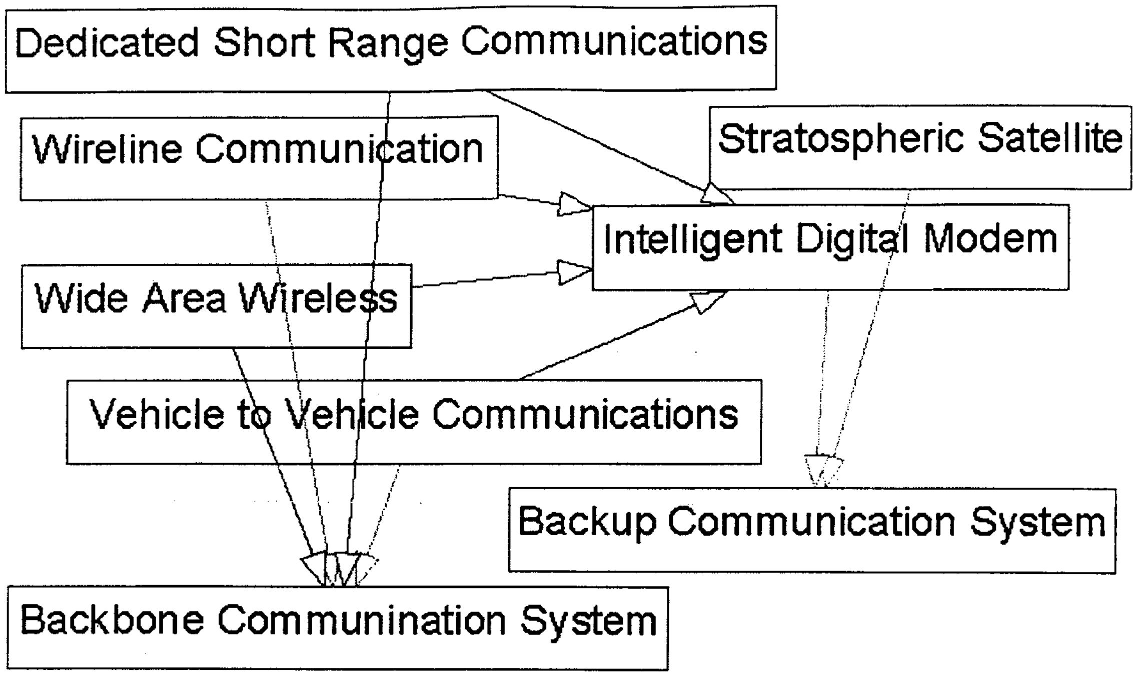

Figure 7.3 SAHS communication class diagram

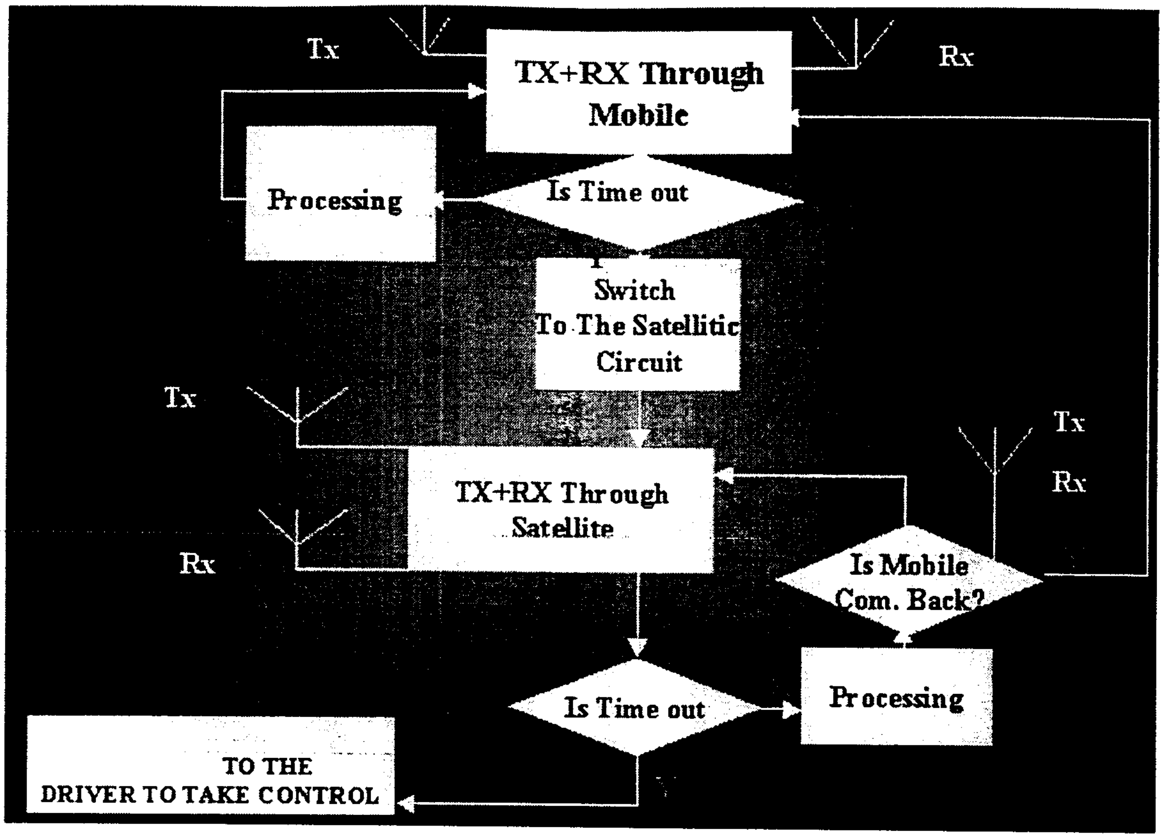

Figure 7.4 Intelligent Digital Modem Time out Concept Flow chart

Satellite messages, radio communication and communication with roadside sensors and other Satellitic Vehicles.

2. In the event that any communication system becomes unserviceable, switch to the designated backup system.
3. If there is no backup, send a message to the driver to take manual control of the Satellitic Vehicle immediately.

It is expected that the vehicles using the IDM could be moving at high speeds, so the time required for communication switchovers to take place is essential. The IDM must have the capacity to execute simple programs and to expand in the future if necessary. The detailed concept of functionality of the Intelligent Digital Modem are illustrated in Figure 7.5

7.5 The need to Modify SAHS requirements

The five requirements that need to be added to SAHS requirements and ITSC requirement are as follows:

1. SAHS shall monitor backbone communication systems at all times.

2. SAHS shall take over if and when malfunction occurs in the backbone communication.

3. SAHS shall direct communication to the Satellitic backup wireless communication system when a malfunction occurs in the backbone communication.

4. SAHS shall continue monitoring all communication channels if and when the backbone system regains its functionality should discontinue the Satellitic backup wireless communication system.

5. SAHS shall send a message to the user when the malfunction occurs in the backup wireless communication system if necessary. In fact the Intelligent Digital Modem should be responsible for such functionality.

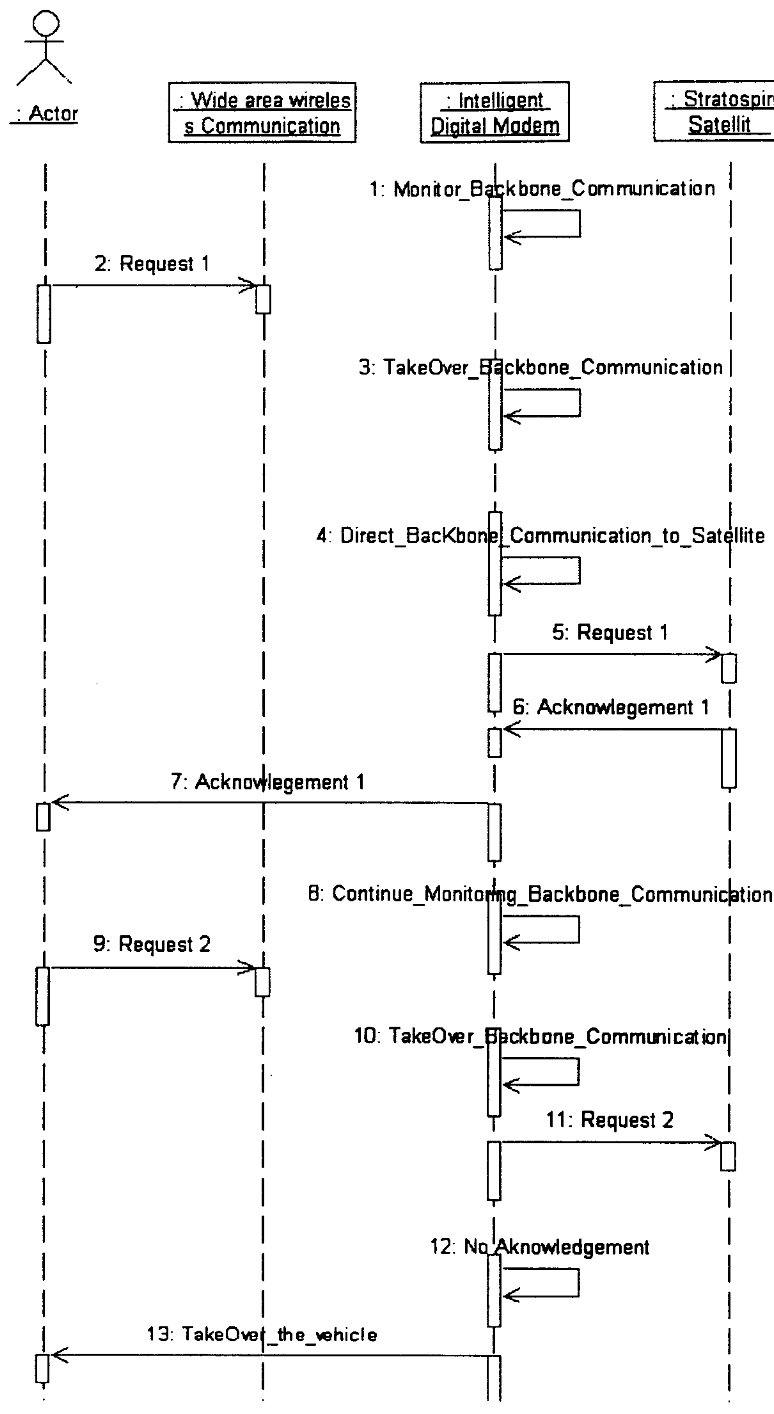

Figure 7.5 Intelligent Digital Modem in Action

7.6 Intelligent Digital Modem Software

Object-oriented design is essential to the modem software because:

1. The IDM will be one part of a traffic control system, which has many components. Object- oriented design allows communication among many components with minimum confusion.
2. Although individual communication components are of varying types, the actions taken by the software to check their statuses are essentially the same. Object- oriented designs allow easy reuse of codes.
3. The future will no doubt bring many other types of communications; the modularity inherent in object-oriented design allows them to be easily integrated into the software.

The computer language chosen for this application is C++, because it is object -oriented, and because its "parent" language, C, is commonly used as control language for all sorts of hardware, including the micro-controller chosen for this task.

Safe operation of the traffic control system requires uninterrupted communication between the Satellitic Vehicles and satellites, ground stations, roadside sensors, and other vehicles, etc.

Thus, it is essential that the IDM operate on a timely basis. For this reason, the IDM software will be event-driven rather than be continuously polling each communication system. Any communication failure will generate an interrupt, which causes the micro-controller to execute codes to handle that failure.

The safest method of assigning interrupts is to give the highest priority interrupts to those communication channels, which have no backup, then to channels with one backup, and two backups, etc. This ensures that the IDM will not waste time trying to switch from satellite to radio, for example, when The last backup for communication with other Satellitic Vehicles just fails and the driver needs to take control of the Satellitic Vehicle.

All transmitters and receivers, which are connected to the IDM, are expected to test themselves, either through self-diagnosis routines or appropriate digital coding which can detect a communication failure. Having the IDM test each device attached to it is time-consuming and would complicate the software to the point that a microprocessor may be necessary. At the most, a failure should be indicated by the setting or clearing of one bit at the interrupt request register of the IDM. More sophisticated devices would also indicate an error number which may be displayed for the benefit of the driver.

All interrupt routines will have the four basic steps:

1. Activate the designated backup system, or tell the driver to take manual control.
2. Inform the backup system that a failure has occurred so that it may retransmit or request a retransmit.
3. Read the error number from the failed system, if possible.
4. Display a message including which system has just failed, and the error number if one exists.

Typical routine is shown in figure 7.6

This software implementation will necessitate the following requirements on the hardware:

1. Programmable using C++.
2. High execution speed.
3. On-board memory.

On-board interrupt controller

```
if(DesignatedBackup) {
        SetCommunicationSystem(DesignatedBackup);
        SendStatus(DesignatedBackup, FAILURE_HAS_OCCURED);
}
else
        Display(EmergencyMessage);
ErrorNumber=ReadErrorNumber(InterruptedCommunicationSystem);
Display(InterruptedCommunicationSystem, ErrorNumber);
```

Figure 7.6 Basic Structure of a Micro-controller Interrupt Routine

7.7 INTELLIGENT DIGITAL MODEM HARDWARE

Three types of devices are considered as the IDM controller: programmable logic arrays, micro-controllers and microprocessors.

The programmable logic array consists of a "switch matrix" which is programmed by the user. A fixed number of inputs have logic operations performed upon them to arrive at a fixed number of outputs. The PLA is inexpensive, but does not have the capacity to execute the programs necessary to control the IDM.

A microprocessor is the central processing unit of a computer, and normally needs peripheral devices such as ROM and RAM chips, serial and parallel ports and interrupts controllers to perform a given task. Microprocessors can operate very quickly (clock speeds over 1 GHz) and have large memory capacity to store large, complex programs. They tend to be relatively expensive ($300 and up), and the necessity of including peripherals can make the design unwieldy.

A microcontroller contains all of the functions of a microprocessor and peripherals on one chip, but its speed (up to 25 MHz) and capacity (up to 64 kilobytes) are smaller than those of a microprocessor. Microcontrollers are less expensive and much more compact than microprocessors. They come equipped with interfaces that allow them to be connected to a personal computer and programmed in a high level language such as C.

The software requirements for the IDM indicate that a microcontroller is the best device to use because it is programmable in C++, operates at reasonably high speed, has enough memory for small programs, and is equipped with an interrupt controller, all at a reasonable price. Hardware devices evolve so quickly that choice of the actual microcontroller to be used for such a futuristic application is indeterminate.

7.8 Intelligent Digital Modem Demonstration Model

The IDM may be demonstrated using a personal computer to act as the vehicle display. To prove the viability of using different communication channels and the ability to switch to a backup, four channels are to be simulated:

1. A vehicle-to-vehicle radio channel.
2. A second radio channel between the vehicle and roadside stations.
3. A cellular telephone channel between the vehicle and a traffic control centre.
4. A satellite communication channel, which backs up any of the first three.

It is anticipated that the two radio channels will be simulated by actual digital radios set to different frequencies. Two cellular phones would set up a cellular phone channel, while a laptop computer with infrared network communication, located in a different room would simulate the satellite communication channel.

The four transmitters and receivers associated with the four channels will have status lines which connect to the microcontroller's interrupt lines. Additionally, some of the channels will have error number outputs to I/O ports on the microcontroller (e.g. the cellular phone may send the byte 0x01 for a low battery, 0x02 for total loss of power, etc.), although the exact implementation has not yet been decided. The microcontroller itself will be connected to the PC's parallel port in order to display messages on the monitor.

The program used by the microcontroller will take the form shown in Appendix 1, and will be used on the microcontroller, with addresses of I/O ports and interrupt numbers changed to fit the microcontroller architecture. When compiled and linked on a PC, the code size will specify a minimum memory size for the microcontroller. The other requirements for the microcontroller are:

1. Eight interrupt lines, one for each transmitter and receiver,

2. Some I/O ports (not necessarily eight, because some transmitters and/or receivers will not produce error numbers)

3. A high enough speed to display the emergency "manual control" message within a typical human reaction time. This should not be a problem for any microcontroller given a typical clock speed of 25 MHz.

7.9 The Microcontroller Object Oriented C++ Program

```
#define ACTIVE 1
#define INACTIVE 0
#define FAILURE_HAS_OCCURED 2
#define DISPLAY_PORT 0x0001
#define NOBACKUP 0
#define NOERROR_NUMBERS -1
#include <stdio.h>
#include <string.h>

class CommunicationChannel {
public:
int ErrorPort;
int IRQNumber;
int Status;
CommunicationChannel *Backup;
CommunicationChannel(int, int, int, CommunicationChannel*);
int Display(char *);                          // Returns the number of bytes sent.
};

CommunicationChannel::CommunicationChannel(int w, int x, int y,
CommunicationChannel *z) {
// Sets the communication channel parameters.
ErrorPort=w;
IRQNumber=x;
ststus=y;
Backup=z;
}

int CommunicationChannel::Display(char *Message) {
// Displays an error message for the user.
int i=0;
while(Message[i])
outportb(DISPLAY_PORT, Message[i]);
```

```
return i;
}
void interrupt Interrupt0() {
char *ErrorMessage;
int ErrorNumber;
char buffer[20];
if(NOBACKUP)
SatelliteTransmitter.Display("Take manual control now!!!");
else
SatelliteTransmitter.Backup->Status= ACTIVE || FAILURE_HAS_OCCURED;
ErrorMessage=strcat("Channel #", itoa(SatelliteTransmitter.IRQNumber, buffer, 10));
if(NOERROR_NUMBERS) {
ErrorMessage=strcat(ErrorMessage, "reports an error.");
else {
ErrorNumber=inportb(SatelliteTransmitter.ErrorPort);
ErrorMessage=strcat(ErrorMessage, "reports error #");
ErrorMessage=strcat(ErrorMessage, itoa(ErrorNumber, buffer, 10));
}
SatelliteTransmitter.Display(ErrorMessage);
// Seven more interrupt service routines here.
int main(int argc, char* argv[])
{
// Set channel parameters.
CommunicationChannel SatelliteTransmitter(NOERROR_NUMBERS, 0, INACTIVE,
NOBACKUP);
CommunicationChannel SatelliteReceiver(NOERROR_NUMBERS, 1, INACTIVE,
NOBACKUP);
CommunicationChannel Vehicle2VehicleTransmitter(NOERROR_NUMBERS, 2,
ACTIVE, &SatelliteTransmitter);
CommunicationChannel Vehicle2VehicleReceiver(NOERROR_NUMBERS, 3, ACTIVE,
&SatelliteReceiver);
CommunicationChannel Vehicle2RoadsideTransmitter(NOERROR_NUMBERS, 4,
ACTIVE, &SatelliteTransmitter);
CommunicationChannel Vehicle2RoadsideReceiver(NOERROR_NUMBERS, 5,
ACTIVE, &SatelliteReceiver);
CommunicationChannel Vehicle2CentreTransmitter(0, 6, ACTIVE,
&SatelliteTransmitter);
CommunicationChannel Vehicle2CentreReceiver(1, 7, ACTIVE, &SatelliteReceiver);
// Code to install the interrupt service routines goes here.
return 0;
```

Chapter 8

CONCLUTIONS AND RECOMMENDATIONS

This chapter opens with a brief discussion of the object- oriented development methodology. The necessity of having a backup communication clearly mentioned (analysis, documentation and the visualization) in the object-oriented ITSC architecture was proposed. Both of these requirements were proposed as a solution to the flexibility and stability quality attributed concerns. The results and contributions are listed in the second section. The chapter concludes with the recommendation for future research.

8.1 Discussion

The object- oriented concept is used not only as a tool for the partial development of alternative ITSC architecture, but also as a way to identify and isolate areas of concern within the architecture. This area of concern will push the next wave of changes in the structure of our Canadian ITS architecture. The current developmental practices i.e. process-oriented development methodologies and its presentation and documentation will make it difficult to respond to the changes that are needed for both the present and the future.

The dissertation professes that the object -oriented concept is an effective tool to isolate the communication problems within ITSC architecture and lead the research in this particular area toward the solution. These inherited problems that we got due to our dependency on what the United States and Europe used for the development of the United States and the European ITS architecture.

The object- oriented methodology that was used in the partial development of CAHS and SAHS captured the hierarchical decomposition of the system's physical components and its functionality. The method is not flat. Object-orientation defines the interface of the system based on the architecture by identifying the actors and their communication within the system and the ability to include attributes of performance within its presentation.

The two main objectives that the research has are to provide the Intelligent Transportation System of Canada (ITSC) a stable and flexible architecture based on object-oriented (OO) methodology, and to integrate the backup communication concept with the new ITSC architecture, which will enhance reliability and dependability of systems that will be developed based on the architecture. There is no standard method of analysis for determining the influence of quality architecture attributes on system architecture. The lack of such method hinders the development of a clear relationship between architectural stability and flexibility, and the system usability and functionality, represented by user and functional requirements. Therefore, meaningful results and physical insight into the influence of architectural attributes are better obtained through scientific, documental, and visual analysis. The use of OOAD in modeling can ease changes in the architectural design when a backup communication system is introduced. The objectives were satisfied by a partial object-oriented Canadian Automated Highway System (CAHS) conceptual project development in which we proved the need for backup communication and a partial object-oriented Satellitic Automated Highway System (SAHS) conceptual project development in which we proved, that our solution for backup communication is adequate and object-oriented (OO) based architecture provide stability and flexibility.

The three-part analyses that were conducted on CAHS conceptual project identified the participation of Backbone communication class in almost every, functionality of the system. What will happen if a malfunction occurs in the system that is presented by the backbone communication class? This proofs that **there is a need for a backup communication system.**

We integrated a backup communication with CAHS and we called it SAHS. Now we have two classes to under take the communication functions. If a malfunction occurs in the backbone communication class, the backup communication class will take over. This solved the problem as we proved in the three-part solution analysis of SAHS. This was the proof, **that our solution for backup communication is adequate**. Because of the **localization of change** that object orientation offers, OO based architecture provide stability and flexibility to the architecture. The integration of a backup communication system effects a small area, thus, the architecture remains stable and the way the integration happens proves that an object-oriented architecture provides the flexibility to the architecture so, **OO based architecture provide stability and flexibility to the architecture.**

The research introduces the concepts of Intelligent Digital Modem (IDM) and the use of Stratospheric Satellite as a communication platform. A generalization of the above two classes were modeled within the backup communication class.

8.2 Results and Contributions

1. Identification of a backup communication problem within the ITSC architecture.

2. Application of the OO methodology to ITSC architecture enhances the stability and the flexibility.

3. Analytical, documental and visualization needs for a backup communication system.

4. Integration of a backup communication concept into ITSC architecture by using an OO methodology.

5. Analytical, documental and visualization demonstrations for a backup communication solution.

8.3 Contributions to Knowledge

1. Demonstrates a defect of the ITSC architecture by developing a partial CAHS architecture.

2. Demonstrates a solution to the ITSC architecture by developing a partial SAHS architecture.

3. Developed a conceptual backup communication system with state of the art technology.

8.4 Recommendations for future research

1. Full development of the ITSC architecture using an object-oriented methodology.

2. Investigate the influence of the ITSC architecture characteristics on quality attributes, especially the ones not addressed by this dissertation, such as safety and environmental soundness.

3. Characterize the mutual influence and dependency of quality attributes on one another to ensure satisfactory fulfillment of ITSC requirements.

4. More detailed analysis of the ITSC requirements is needed for the complete development of an ITSC architecture. Such an analysis can include case-specific requirements and features.

5. Although not addressed, the ideal ITSC architecture must address the economical, environmental and social impact of the implementation of an ITSC architecture. Thus, research is required in this area.

6. Study the impact of the integration of the atmospheric vehicle (SkyCar) and atmospheric highway concepts into the development efforts of a Satellitic Automated Highway System (SAHS) concept.

Bibliography

Bachmann, Flex, Len Bass, Jeromy Carrire, Paul Clements, David Garlan, James Ivers, Robert Nord, and Reed Little, *Software Architecture Documentation in Practice: Documenting Architectural Layers*, Software Engineering Institute, Carnegie Melon University, [CMU/SEI-2000-SR-004], 2000.

Barbacci, Mario, Mark Klein, Thomas Long staff, and Charles Weinstock, *Quality Attributes*, Software Engineering Institute, Carnegie Melon University, [CMU/SEI-95-TR-021/ESC-TR-95-021], 1995.

Bement, Arden L., et al. *National Automated highway System Program-A review /Committee for a review of the national Automated Highway System Consortium Research Program. (Special report: 253).* Washington, D.C.: Transportation Research Board, National Research Council, 1998.

Booch, Grady. *Object-Oriented Analysis and Design, Second Edition.* Redwood City, Calif.: Benjamin/Cummings, 1994. ISBN 0-8053-5340-2.

Booch, Grandy. *Object-Oriented Analysis and Design: with Application.* Reading, Massachusetts: Addison Wesley Longman, Inc., 1997.

Coad, Peter & Yourdon, Edward. *Object-Oriented Analysis, Second Edition.* Englewood Cliffs, NJ: Yourdon Press, 1991. ISBN 0-1362-9981-4.

Coleman, D.; Dollin, C.; & Jeremaes, P. "Fusion: A Second Generation Object-Oriented Analysis and Design Method," pp. 189-193. *Proceedings: Object EXPO Europe Conference*. London, U.K., July 12-16, 1993. New York, New York: ACM SIGS Publishing, 1993.

Dahl, O. and K. Nygaard; "Simula: An algol-based simulation language". *Communications of the ACM*, 9, 1966.

DeMarco, T., "Controlling Software Projects", Englewood Cliffs, and N.J.: Prentice-Hall, 1982.

Easa, Said M. " Automated Highways." *Encyclopedia of Electrical and Electronic Engineering*. By Ed.John G. Webster. New York: John Wiley & Sons Ltd., 1999. Webster 3-3034/ Article 2603.

Gane, C. and Sarson, T., *Structured Systems Analysis*, Englewood Cliffs, and N.J.: Prentice-Hall, 1979.

Hatley, D.J. and Pirbhai, I.A., *Strategies for Real-Time System Specification* - New York, N.Y.: Dorest House Publishing Co., 1987.

ITSA, "National ITS Architecture." Version 4.0. 17 April 2002. *ITS America*. 9 May 2000 < http://itsarch.iteris.com/itsarch/index.htm >.

ITSAu, PPK (Environment &Infrastructure), CSIRO, and Intelligent Transportation Specialists (1999). *A National Reference Architecture For Intelligent Transport System (ITS)*. Australia: ITS (Australia) lnc.

ITSC, "ITS Architecture for Canada". *Transport Canada*. Ver1.1. 2000. 9 May 2002 < http://www.its-sti.gc.ca/en/static/content.htm >.

ITSE, "European Framework Architecture." *First issue (KAREN)*. Sept 2000. European Commission. 9 May 2002 < http://frame-online.net/library.htm>.

ITSJ, "System Architecture for ITS in Japan -Detailed Materials-."System Architecture for ITS in Japan". 15 October 2001. *VERTIS*. 9 May 2002 < http://www.vertis.or.jp/e-sa/index.html>.

Jacobson, I.; Christerson, M.; Jonsoon, P.; & Overgaard, G. *Object-Oriented Software Engineering*. Reading, Me.: Addison-Wesley, 1992. ISBN 0-2015-4435-0.

Jacobson, I.; Christerson, M.; Jonsoon, P.; & Overgaard, G. *Object-Oriented Software Engineering*. Reading, Me.: Addison-Wesley, 1992. ISBN 0-2015-4435-0.

Jesty P.H., J.F Gaillet, J.F. Giezen, G. Franco, I. Leighton, H. J. Schultz, " Guidelines for the Development and Assessment of Intelligent Transportation System Architectures", *The European Union*, DSA2.3, 1998.

Minsky, M., "A framework for representing knowledge"; In P. Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill, 1975.

NASA Dryden Flight Research Center. "Helios". NASA. 1 June 2002 <http://www.dfrc.nasa.gov/>.

OMG. "OMG Unified Modeling Language." Version 1.4. September 2001. *OMG*. 1 June 2002 <http://cgi.omg.org/docs/formal/01-09-68.pdf>.

Rational Software Corporation. *Rational Rose 98: Using Rational Rose*. U. S. A.: Rational Software Corporation. 1998

Rumbaugh, J. et al. *Object-Oriented Modeling and Design.* Englewood Cliffs, NJ: Prentice-Hall, 1991. ISBN 0-1362-9841-9.

Shaldover, Steven E. "Why WE Should Develop A Truly Automated Highway System." *Transportation Research Record* 1651, 98-0641-1998: 66--73.

Stevens, W.P., G.J. Mayers and L.L. Constantine, "Structured Design," *IBM System Journal*, Vol. 13, No. 2, May 19974, pp. 115—139. Reprinted in P. Freeman and A.I. Wasserman, Editors, *Tutorial on Software Design Techniques, Forth Edition, IEEE Computer Society Press* (Catalog number EH0205-5), Silver Spring, Maryland, 1983, pp. 328-352.

Transport Canada, "An Intelligent Transportation Systems (ITS) Plan for Canada: En Route to Intelligent Mobility" *Transport Canada*. November 1999. 8 May 2000 < http://www.its-sti.gc.ca/en/docs/its_plan.pdf >.

U.S. DOT, "Precursor System Analysis of Automated Highway Systems", *U.S. Department of Transportation*, 1997.

Wirfs-Brock, Rebecca; Wilkerson, B; & Weiner, L. *Designing Object-Oriented Software.* Englewood Cliffs, NJ: Prentice-Hall, 1991. ISBN 0-1362-9825-7.

Yourdon, E., *Modern Structured Analysis*, Englewood Cliffs, and N.J.: Prentice-Hall, 1991.