SENG 609.22 Agent-based Software Engineering
Instructed by Dr. Far

Laboratory Project

Police Patrol Assistant System (PPAS)

Designed by:
Vani Mudigonda
Jialin Shen
Oct 25, 2005

Contents

1. Introduction

The role and responsibilities of a policeman have become extremely challenging in solving present day crimes because our societies and communities have become dynamic. The advancements in the communication technologies provide a lot of information related to various crimes and criminals. Such vast amount of information needs to be effectively used in a short time to solve crimes. This poses additional technological challenges in order to organize the information to help police officers in their job roles as "Uncontrolled and unorganized information is no longer a resource in an information society, instead it becomes the enemy [1]."

The present work proposes to develop a Police Patrol Assistant System (PPAS) to help a traffic policeman in the identification of vehicles and their drivers. When a police officer stops a suspicious vehicle, the officer needs to quickly identify the history of driver as well as the registration information of the vehicle to plan for his further investigation. This situation requires information from multiple databases including criminal, 911 emergency centers, and insurance databases to enable the police officer to quickly assess the situation and to formulate further course of action. We propose to develop the PPAS application by employing an agent-based software engineering methodology proposed by Gaia [2] considering the following factors:

- PPAS is a complex system and requires information from multiple databases maintained by different organizations and a multiagent system (MAS) is required to handle their interactions.
- PPAS uses collaborative software agents to solve complex distributed problems. It adopts the use of ontology services when accessing information from different data sources.
- It is very difficult to design and implement the same system using any other object-oriented technique.
- PPAS is to be designed using MAS because it can be easily extended later by adding more agents for assisting the police officers in solving other crimes.

This report is organized as follows. Section 2 discusses PPAS system requirement analysis including system configuration and the assumptions for the system. Section 3 presents a brief description of the developmental approach for the PPAS using Gaia methodology. Section 4 presents analysis and design of the system. Section 5 presents conclusion and discusses the scope for future work.

2. Requirements analysis

An agent-based Police Patrol Assistant System (PPAS) will facilitate the police officer with the information needed to handle the situations efficiently and effectively. In this system, intelligent agents have a great potential in helping police officer to make a decision. Table 1 shows both the high level and derived requirements analysis for the PPAS system.

2.1 Requirements:

Table 1. Requirement specifications for the PPAS system.

Requirement ID		Requirement Text
R1		PPAS shall provide a web-based user interface to the operator.
	R1.1	PPAS shall authenticate registered users logon information including user id and password.
R2		PPAS shall acquire information from multiple databases including Registration, Criminal/Prisoner and Vehicle stolen center in a maximum time of 20 minutes.
	R2.1	PPAS shall use vehicle ID, vehicle model, any other user input provided by the user to search for details in registration database.
	R2.2	PPAS shall report the information back to the user through user interface.
	R2.3	PPAS shall use personal ID of the driver provided by the user to search for the details about the driver in the criminal database.
	R2.4	PPAS shall use vehicle ID, vehicle model, and personal ID provided by the user to search in the 911 emergency database and check if the vehicle or the driver or both have involved in an accident recently.
	R2.5	PPAS shall use the vehicle ID provided by the user to search the stolen vehicle database if the vehicle had been stolen.
	R2.6	PPAS shall use the vehicle ID and personal ID provided by the user to generate a ticket for the traffic violations.
R3		PPAS shall present to the user all the information acquired in a consistent manner as per a prespecified format.
R4		PPAS shall flash appropriate error messages in case of communication problems or any other problems.
	R4.1	PPAS shall flash a login error message in case of login failure.
R5		PPAS shall update a log file that includes information about login information, date, time, and any other user actions/requests.
R6		In case of unreliable wireless connection, PPAS shall provide an alternate way for the client to resume the session.

2.2 system configuration

Figure 1 shows the building blocks of the Police Patrol Assistant System (PPAS). Police officer on patrol can access the PPAS located at the base control station through wireless network. The officer requests the PPAS for the vehicle or driver-related information and the PPAS contacts multiple databases distributed across the web to obtain the information required. Then it processes the information and presents it to the user in an organized manner.

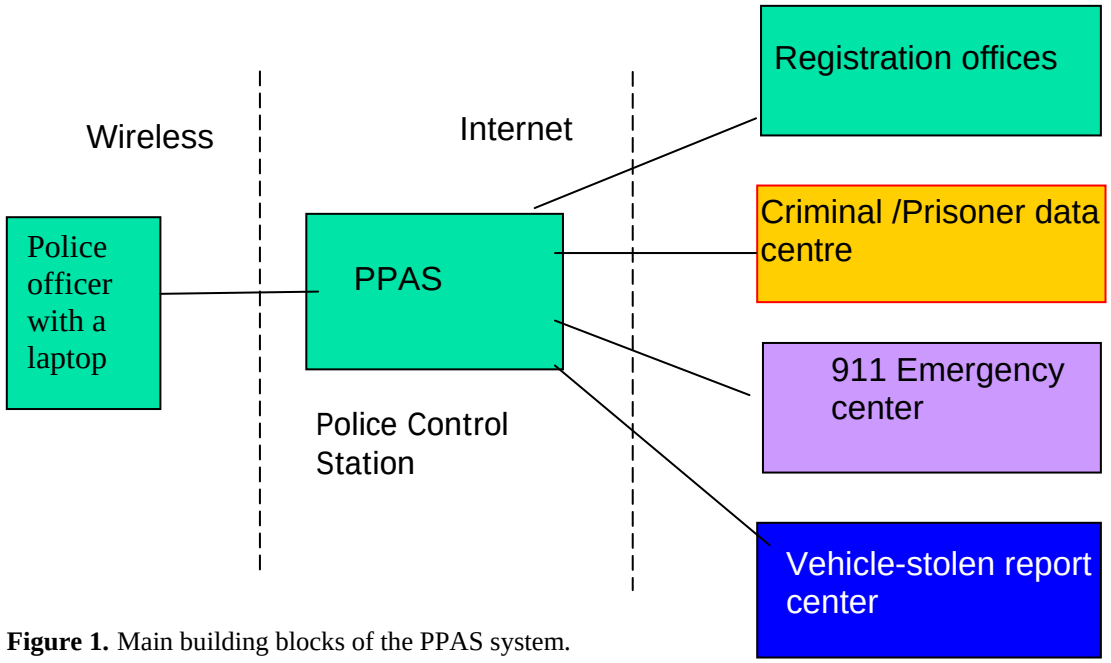

Figure 1. Main building blocks of the PPAS system.

2.3 Assumptions

All data centers provides web services interface for PPAS to retrieve information.
The user requests a check through a user interface. The user interface captures the vehicle and driver information. There are already predefined web services the system could make use of. These web services have the same WSDL description meaning their input and out messages have a standard syntax.

2.4 Wish list

a. In case of unreliable wireless connection, PPAS should provide alternate ways to allow the clients to resume their session. When communication connection is resumed, required information should not be discarded and the information is sent to the user by other means of communication.
b. System has an authentication mechanism. Only granted privileged users could use this system.

3. Development Approach

According to the high level requirements, we have derived some characteristics of the PPAS system:

- The PPAS needs to adopt a top-down approach to develop an agent-based system.
- PPAS system is heterogeneous system—there are different kinds of computers in the system, and also agents may be implemented using different programming languages, architectures and techniques.
- In the PPAS system, inter-agent relationships, abilities and services they provide do not change at run-time. The organization structure is static.
- The system contains small number of different agent types.

Based on these characteristics, Gaia methodology is very suitable for its analysis and design.

Gaia methodology allows an analyst to go systematically from a statement of requirements to the design of the system in a step by step manner. Each successive step introduces more concrete details and shrinks the space of possible system components. The models used in Gaia and their relationships are illustrated in Figure 2. In Gaia methodology, building agent-based systems resembles the process of organizational design [2].

The designing of a multiagent system in Gaia methodology is done in two phases. The abstract aspects of the system are captured in the analysis phase. The analysis phase includes roles, permissions, responsibilities, protocols, activities, liveliness properties and safety properties. The concrete aspects of the system are captured in the design phase that include agent types, services, and acquaintances. It has two analysis models and three design models.

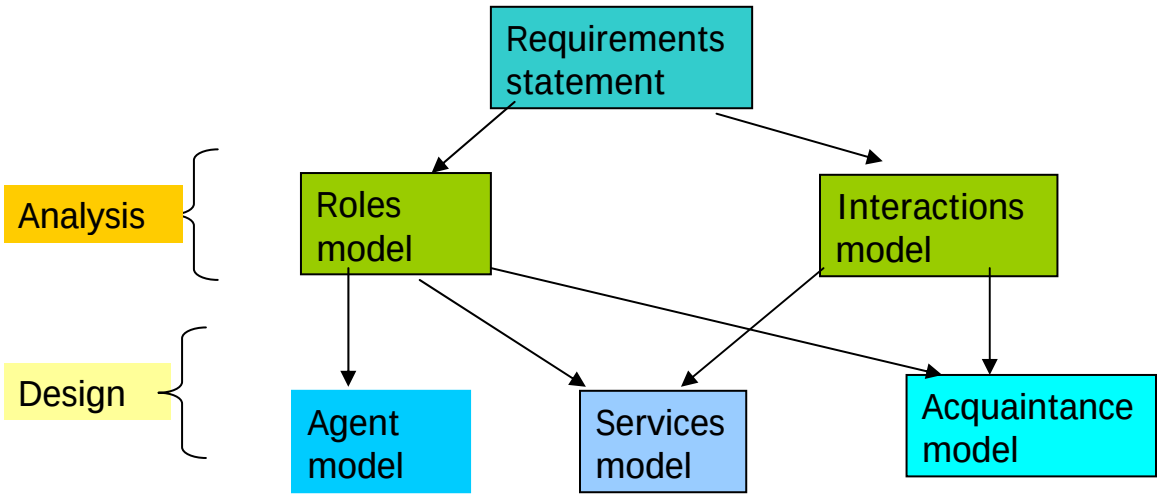

Figure 2: Models and relationships in Gaia

The analysis and design process of the Gaia methodology is described below.

In the analysis phase, roles in the system are identified and their informal descriptions are produced from the requirements of the system. For each role, associated protocols are identified and documented. Based on the protocol model elaborate the roles and document their associated permissions, responsibilities and protocols. In the design phase, create an agent model by aggregating roles into agent types. Refine the agent types to form an agent type hierarchy and document the instances of each agent type using instance annotations. In the next step develop a service model by examining activities, protocols, safety, and liveliness properties of roles. Then develop an acquaintance model from the interaction model and agent model.

Gaia methodology only provides analysis and design techniques for a system. It doesn't provide implementation procedure. The implementation ideas are provided in the design model. Object oriented methodologies or any other classic methodologies can be used for the implementation. For detail design and implementation, we propose to use OO programming model, XML as the content language, CORBA as development platform and IIOP as its communication protocol.

4. Analysis and Design of PPAS

As shown in figure 3, the PPAS system mainly consists of Police Assistant Agent (PAA), Criminal agent, Registration agent, Ticket agent and Police Assistant (PA) database. The PA database stores all the management and application related data, such as the logon information, audit data, traffic rules, and fine.

The clients use the web-based user interface to input the particulars of vehicle as well as driver and request to verify the vehicle registration details as well as the drivers' history. Only registered users are authenticated to logon to the system. The logon information, date, time and the requests are stored in the PA database for a later audit or review. The registration agent checks the registration web services for the vehicle registration information. It acts as an ontology agent and the information is returned in an organized and prespecified format. The criminal agent checks the history of the driver in three different web services: stolen vehicle center, criminal/prisoner services, and 911 emergency service. This agent provides an ontology service as well and the information is presented in the prespecified format. The ticket agent automatically generates ticket with the fine amount, date, location, vehicle, and driver details. Based on the user input and the preexisting traffic regulation stored in the PA database, fine amounts for the ticket are decided.

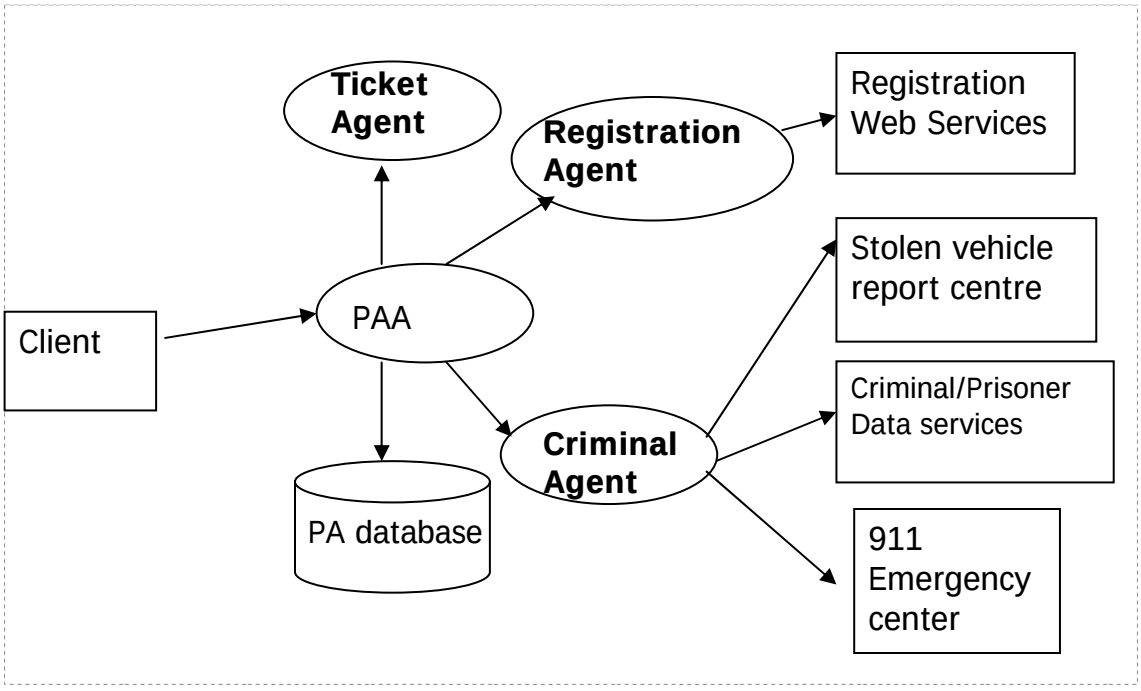

Figure 3 System Architecture of PPAS

4.1 Analysis

Different Police patrol teams will access the PPAS through wireless network and PPAS needs to serve all requests of the police officers on patrol by contacting

different web services to retrieve the necessary information. In the roles model of the analysis phase various roles of the PPAS system are identified. In the interaction model the interactions and relationships among these roles are identified and represented in their protocol definition.

4.1.1 Roles Model

From the requirement specifications and the system architecture, different roles are identified for the PPAS system. These roles along with their permissions, responsibilities and protocols are captures in the role schemas shown the figures (4 to 9). Next the interactions of these roles are captured in the figures (10 to 15).

4.1.1.1 Roles

The following are the various roles identified for the PPAS system:

- Police assistance role: PoliceAssistant
- Web-services handling roles
  - Registration web service: RegistrationChecker
  - Stolen vehicle report centre: StolenReporter
  - Criminal/Prisoner Data service: CriminalHunter
  - 911 Emergency centre: AccidentInvolver
- Ticket generating role: TicketIssuer

Police assistance role:
The police assistance (PA) role is the central role for the system. The responsibilities associated with the PA role are to check the user authentication against user id and password. PA uses the inputs from the user and contacts the other roles such as RegistrationChecker role, StolenReporter role, AccidentInvolver role and CriminalHunter role, which obtain the information from various web services and also provide ontology services to PA. These roles search the web services/resources for the requested information and report to PA that processes them and presents them to the user. This helps the police officer in making quick decisions.

Web services role includes RegistrationChecker role, StolenReporter role, AccidentInvolver role and CriminalHunter role.

Registration Check role:
The responsibilities associated with the RegistrationChecker role are to use the registration plate number, vehicle identification number and contact all registration web services. It will report the information about vehicle such as what province the registration number was issued, whether the registration was renewed and all other details pertaining to vehicle and driver to the user through the user interface.

Criminal Check role:
The responsibilities associated with the CriminalCheck role are to contact the criminal/prisoner database services and check if the driver is a suspect given the drivers' identity by PA. It uses the Vehicle Identity Number (VIN), model, color or any other

details pertaining to the vehicle and contacts the stolen vehicle report center to check whether the vehicle is legal. It checks the 911 emergency services to verify if the vehicle had been involved in an accident recently.

Ticket generating role:
According to the record of speed and violation of traffic laws, and the driver's history, the ticket role responsibility includes to automatically generating fine suggestions to policemen and also issuing the ticket. The user may modify the fine amounts or cancel the ticket. The PA database is updated.

Stolen Reporter role:
The responsibility of the StolenReporter role is to use the VIN, model, color and other vehicle related information and search the Stolen Vehicle Center to check if there is any vehicle reported stolen and returns the information to the user with an 'yes' or 'no'.

Accident Involver role:
The responsibility of the AccidentInvolver role is to uses the VIN, model, color and other vehicle-related information and the driver's description to search the 911 Emergency Center and check whether the vehicle is reported involved in an accident.

4.1.1.2 Role Schemas
Each role is associated with some permissions and responsibilities and their interactions are defined using protocols and activities. A role schema is specified for each role that captures their permissions, responsibilities and protocols. The following figures show the role schemas for all the roles identified.

Figure 4 describes the role schema for the police assistant role. This schema shows that the police assistant has the permission to read the supplied police request and generate reports with the information requested by the user. The responsibilities of the police assistant role include interacting with the other roles and responding to the user request.

Role Schema: PoliceAssistant
Description: Using the inputs from the user search the web services through other roles like registration role and criminal role and report to user with the information requested.
Protocols and Activities: AwaitRequest, CheckVehicle, CheckPerson, IssueTicket, CancelTicket, GenerateReport. ResponseInquiry
Permissions: Reads Supplied PoliceRequest *// details requests* Generates Report *// Give report according to the request.* Tickets *// issue tickets according to the violation types.*
Responsibilities Liveness: PoliceAssistant = (AwaitRequest. CheckVehicle.

CheckPerson.GenerateReport. ReponseInquiry) IssueTicket = (RequestTicket, CancelTicket, ResponseTicket) Safety: ResponseTimeout = 20 mins

Figure 4. Schema for the role of Police Assistant.

Figure 5 describes the role schema for the role of Registration checker. The permissions of the registration checker include reading the supplied registration inquiry, contact the web services, and generate the reports with the response to the inquiry in a predefined format.

Role Schema: Registration Checker
Description: The Registration Checker role checks all the registration details of the vehicles.
Protocols and Activities: AwaitRegistrationRequest, CheckVehicleRegistration, ResponseRegistrationInquiry
Permissions: Read Supplied *PoliceRequest* *//request for vehicle information* Generates VehicleIsLegal // *boolean*. VehicleReport // *respond the information*
Responsibilities Liveness: RegistrationChecker = (AwaitRegistraionRequest. CheckVehicleRegistration. ReponseRegistrationInquiry) Safety: VehicleIsLegal = False ➔ report

Figure 5. Schema for role of Registration Checker.

Figure 6 describes the role schema for the stolen report role. The permissions of this role includes to read the supplied vehicle information and contact the stolen vehicle center for related vehicle information and generate reports with the confirmation of the requested information.

Role Schema: StolenReporter
Description: Contact the Stolen Vehicle Center to check if there is any vehicle reported stolen and get the information of the stolen vehicles – its registration number, model, color.
Protocols and Activities: AwaitRequest, CheckStolenRecord, ResponseInquiry
Permissions: Reads Supplied VehicleID // vehicle ID Generates Rreport // Give report according to the request.
Responsibilities

Liveness: StolenReporter= (AwaitRequest. CheckStolenRecord.ResponseInquiry)* Safety: true

Figure 6. Schema for the role of Stolen Reporter.

Figure 7 describes the schema for the Criminal Hunter role. The permissions for this role include reading the supplied personal information and contacting the criminal web data source and responding to the request with the information obtained and processed.

Role Schema: CriminalHunter
Description: Contacts the criminal database services and prisoner database to check whether the driver is a suspect according to the supplied driver's identification, personal identification and the description of the features.
Protocols and Activities: AwaitRequest, CheckCriminalRecord, ResponseInquiry
Permissions: Reads Supplied PersonalSocialID // Personal ID Generates Rreport // if the person is a suspect or criminal.
Responsibilities Liveness: Criminal Hunter = (AwaitRequest. CheckCriminalRecord.ResponseInquiry)* Safety: personIsCriminal = true ➔ call for help

Figure 7. Schema for role of Criminal Hunter.

Figure 8 describes the role schema for the accident involved role. The permissions for this role include reading the supplied driver and vehicle information and checking with the 911 emergency database and responding to the user request with a conformation if the vehicle had been involved in an accident recently.

Role Schema: Accident Involver
Description: Uses the VIN, model and other vehicle related information and/or the driver information to search the 911 Emergency database and check whether the vehicle had been recently reported of involvement in an accident.
Protocols and Activities: AwaitRequest, Check911Record, ResponseInquiry
Permissions: Reads Supplied Driver information // Personal ID Vehicle information //model, type, vehicle ID Generates Rreport // if the vehicle is reported as involved in accident
Responsibilities Liveness:

AccidentInvolver = (AwaitRequest. Check911Record.ResponseInquiry)* Safety: true

Figure 8. Schema for role of Accident Involver.

Figure 9 describes the role schema for the ticket issue role. The permissions of this role include reading the supplied driver details and generating a ticket.

Role Schema: TicketIssuer
Description: This role involves accepting the request, collecting necessary information to issue a ticket to requestor.
Protocols and Activities: AwaitRequest, CollectInformation, GenerateTicket, CancelTicket, ReplyGenerateRequest, ReplyCancelRequest
Permissions: Reads Supplied DriverDetails // Detailed driver's violation Changes TicketHistoryRecord //delete/add record Generates Ticket // generate a ticket according violation type.
Responsibilities Liveness: TicketIssuer = (AwaitReq. CollectInfo. GenerateTicket/ CancelTicket. ReplyReq)* Safety: TicketRecordNumber>=0

Figure 9. Schema for role of Police Assistant.

4.1.2 Interaction Model

In this model interactions and relationships between the various roles are captured. This is represented by the protocol definition, which includes initiator role, responder role and the processing. The inputs and outputs of these interactions are also captured. The following figures represent the definition of protocols associated with each role. Figure 10 represents the definition for the protocols initiated by the Police Assistant role and responded by other roles like Registration role, CriminalHunter role or the Ticket issuer role to obtain the information about the vehicle and the driver and to generate or cancel ticket requested by the user. These protocols provide CheckVehicle, CheckPerson, GenerateTicket, CancelTicket service.

CheckVehicle		
PA	RR(Registration Role)	Supplied Vehicle information
Check whether vehicle is legal		Legal/Illegal with case description

CheckPerson		
PA	Criminal Roles	Supplied Driver SIN
Check whether person is a suspect		Criminal/suspect or not without report

GenerateTicket		
PA	TicketIssuer	Supplied Violation types and person information
Request a ticket		ticket

CancelTicket		
PA	TicketIssuer	
Cancel a ticket		Delete Success

Figure 10. Definition of protocols associated with the Police Assistant role.

Figure 11 represents the definition for the protocols initiated by the registration role and responded by registration web services with the information pertaining to vehicle records. The registration role uses the vehicle-related inputs from PA to search the registration web services and report the vehicle information back to the PA. CheckVehicleRegistration service, ResponseRegistrationInquiry services are provided by these protocols.

CheckVehicleRegistration		
Registration Role	RegistrationWebSevices	Supplied vehicle information
Request retrieving Vehicle records		Legal with VehicleDetails

ResponseRegistrationInquiry		
RR	PA	
Give the report of vehicle information		VehicleDetails

Figure 11. Definition of protocols associated with the Registraionchecker role.

Figure 12 represents the definition for the protocols initiated by the stolen reporter role to check if the vehicle is reported stolen and responded by stolen report web services if it is recorded as stolen. CheckStolenRecord, ResponseStolenCarInquiry services are provided by these protocols.

CheckStolenRecord		
StolenReporter	Stolen Report Webservices	Supplied Vehicle Information
Request a check for the vehicle whether it is marked as a stolen car		VehicleIsLegal

ResponseStolenCarInquiry		
Stolen Report Centre	Stolen Reporter	
Give the report whether the car is stolen		VehicleIsLegal

Figure 12. Definition of the roles associated with the StolenReporter role.

Figure 13 represents the definition for the protocols initiated by the CriminalHunter role to check if the person is a suspect and responded by the criminal data center with the response to the inquiry and the report is passed on to the CriminalHunter role. CheckStolenRecord, ResponseStolenCarInquiry services are provided by these protocols.

CheckCrimanalRecord		
Criminal Hunter	Criminal Data Centre	Supplied person information
Request the check for whether the person is a suspect		PersonIsSuspect

ResponseCriminalInquiry		
Criminal Data Centre	Criminal Hunter	
Give the report if Person is a suspect		PersonIsSuspect

Figure 13. Definition of protocols associated with the Criminal Hunter role.

Figure 14 represents the definition for the protocols initiated by the AccidentInvolver role to check whether the vehicle is involved in an accident and responded by 911 emergency center with the response to the inquiry. ResponseAccidentInvolveInquiry, Check911Record, are the services provided by these protocols.

Check911Record		Supplied Car Information
AccidentInvolver	911 emergency centre	
Request a check whether the car is involved in an accident		CarIsInvolvedAccident

ResponseAccidentInvolveInquiry		
911 Emergecy centre	AcidentInvolver	
Give the report of vehicle if it is most possibly involved		CarIsInvolvedAccident with report

Figure 14. Definition of protocols associated with the 911 Emergency center role.

Figure 15 represents the definition for the protocols initiated by the ticket issuer role and responded by personal assistant by issuing or deleting a ticket. Reply Generate Request, ReplyCancel Request services are provided by these protocols.

Reply Generate Request		
Ticket Issuer	PA	
Return a ticket		Ticket

ReplyCancel Request		
Ticket Issuer	PA	
Return whether the task is finished or not		DeleteTicket

Figure 15. Definition of protocols associated with the Ticket Issuer role.

4.2 Design

Gaia design process involves generating three models: Agent model and Services model and acquaintance model. The Agent model identifies the agent types that will make up the system, and the agent instances that will be instantiated from these types and that will realize these agent types at run-time. The service model identifies the main services that are required to realize the agent's role. The acquaintance model documents the lines of communication among the agents [2].

4.2.1 Agents model

In PPAS system, we have identified the following four agents:
Police Assistant Agent, Registration Agent, Criminal Agent, and Ticket Agent. The Criminal Agent is assigned with three roles: CriminalHunter, StolenReporter, AccidentInvolver. The responsibilities of these three roles are to identify if the driver had involved in a crime or had a criminal record. They intend to serve the same functionality while searching in different databases. Combining these roles to form one agent optimizes the design of the system and the system becomes more efficient. Other agents have one to one mapping with the roles.
Figure 16 shows how each agent is assigned with roles and the multiplicity of their relationships.

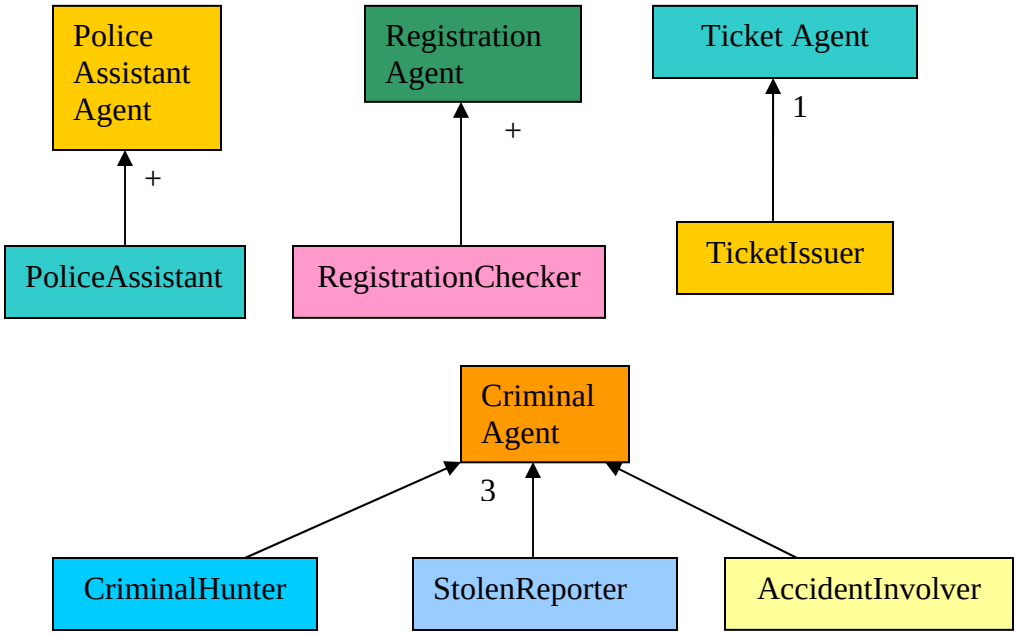

Figure 16. Agent Model for the PPAS system.

4.2.2 Agents descriptions

Registration Agent:

The registration agent uses the VIN, model, and any other vehicle related inputs provided by the PAA and search the web registration services including different regional registration offices. It obtains the information from the web services and reports to the PAA with the information of the owner, registration renewal date, place where the registration plate was issued, and any other details that is of to the interest to the user.

Criminal Agent:

The criminal agent uses the VIN, vehicle registration number, drivers license, drivers personal identity and any other vehicle related or driver related inputs from the PAA and searches the criminal web services, vehicle stolen center, 911 emergency centers. The criminal web center responds whether or not the person is in their database. Similarly the vehicle stolen center responds to the inquiry whether or not the vehicle is in their records and the 911 emergency center responds whether there was an accident recently and the vehicle or the driver or both were involved in the accident. The criminal agent obtains the information from these three different databases and processes the information in a prespecified manner using the ontology services and presents it to the PAA. It also uses police wireless radios to call other police officer in case the driver is a criminal and there is a potential danger to the officer.

Ticket Agent:

The ticket agent uses the vehicle registration number, driver details along with any other inputs from the PAA as well as the prespecified traffic laws and fine amounts specified in

PA database, it automatically generates a ticket. It provides an interface for ticket management and can be modified, deleted or updated by the user.

PAA agent:
PAA agent accepts the request sent by the police patrol officers through the web-based interface. It handles the correspondence with all other agents as a central point of delegation. On behalf of the police officer, it collects the information from different data sources/services and processes and presents it to the user in an organized manner. Using the inputs from the client, it contacts the ticket agent and arranges for issuing a ticket.

4.2.3 Services models

In this model, we need to identify services associated with each agent role, and specify the main properties of these services.
Table 2 presents the details of services for all agent roles. CheckVehicle, IssueTicket, CheckStolenCar, CheckCriminal, CheckAccident are the services identified for all the agents. The inputs, outputs, preconditions, and post conditions for these services are tabulated.

Table 2. **Services Model**

Service	Inputs	Outputs	Pre-condition	Post-condition
CheckVehicle	Vehicle Identity number (VID) and Plate number	Vehicle registration information	VehicleIs Legal = nil	VehicleIsLegal != nil
IssueTicket	Violation Types and Driver history information	Ticket	true	Ticket generated
CheckStolenCar	Car information	IsStolen	CarIsStole n = nil	CarIsStolen != nil
CheckCriminal	Person ID	IsSuspect	Nil	IsSuspect!=Nil
CheckAccident	Car and Person	CarIsInvolved	Nil	Police knows whether the car is involved in an accident.

4.2.4 Acquaintance model

The acquaintance model defines the communication links that exist between agent types. It doesn't define what messages are sent or when messages are sent. It only indicates that communication pathways exist to identify any potential communication bottlenecks that may cause problems at runtime.
Figure 17 illustrates the acquaintance model for the PPAS. PAA handles the correspondence with all other agents (Registration agent, criminal agent and ticket agent) as a central point of delegation. The user sends and receives information through PAA.

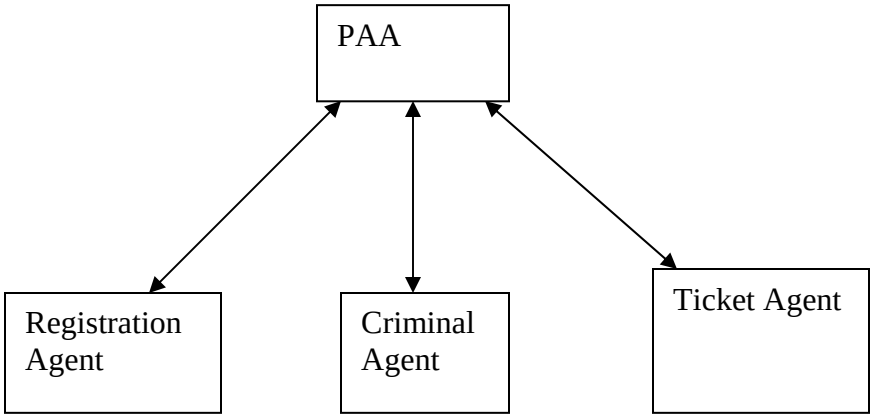

Figure 17. Acquaintance Model for the agents involved in PPAS

4.3 Data / knowledge sharing specification

This section presents use cases for each participating agent, their class diagrams, E-R diagram, and data definition tables.

4.3.1 Use cases: Figures 18, 19, 20 and 21 show the use cases for each agent and their use case definitions are also presented.

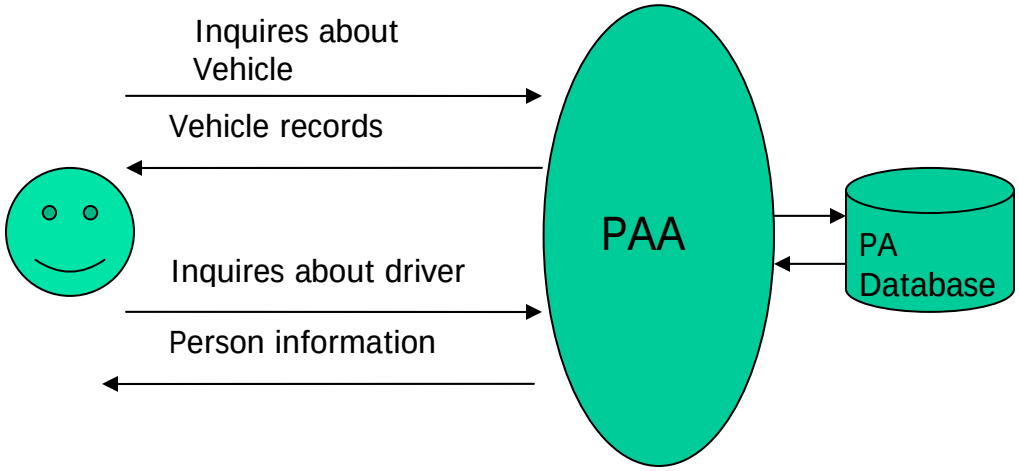

Figure 18. Use case for PAA (Police Assistant Agent)

Use case Definition:

Brief Description: The actor uses this use case to check vehicle and/or driver information
Precondition: Actor should pass system authentication. All actors should have an account with their profiles recorded in the PA database.
Post condition: If actor passes the authentication, he can use the facilities provided by PAA.

Process steps:

a. The Actor inputs the vehicle and personal information and makes a request to check whether the vehicle is legal or a person is a suspect.
b. PAA sends the request to Registration agent, which will collects information from registration web services about that vehicle.
c. PAA uses the vehicle and driver information and requests Criminal Agent to check if the person is a suspect or the vehicle is involved in an accident.
d. If the person is a suspect or the car is a stolen car, PAA will warn the actor.
e. If ticket is needed, issue a ticket through Ticket Agent.

Exceptions: 1a, System is not accessible. Error message is generated stating that system is not accessible at this point. Use case terminated.
2a, system is not accessible. Message is generated, Use case continues to next step.

3a. System is not accessible. Message is generated. Use case terminated.

Relationships:
Initiating: actor
Collaborating: Criminal Agent, Registration Agent, and Ticket agent.
Other Diagram:
Data Requirements:
Data Required: Driver information
Vehicle Information
Traffic rules and laws

Figure 19 shows the use case for the Registration agent.

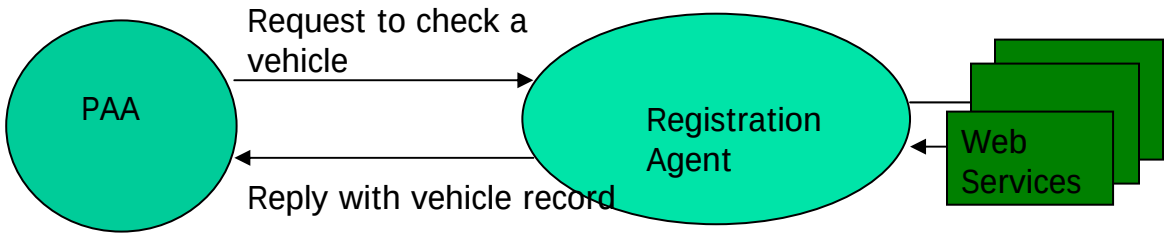

Figure 19. Use case for Registration Agent

Use case Definition:

Brief Description: PAA uses this use case to check vehicle information in the registration web services.
Precondition: Plate Number is provided.
Post condition: Vehicle record returned.
Process steps:

1. PAA requests Registration Agent to check whether the vehicle is legal
2. Vehicle record is returned.

Exceptions: 2a, if the response is not received in 20 minutes, Error Message will be generated.
Relationships:
Initiating: PAA
Collaborating: Registration Agent.
Other Diagram:
Data Requirements
Data Required: Vehicle Information

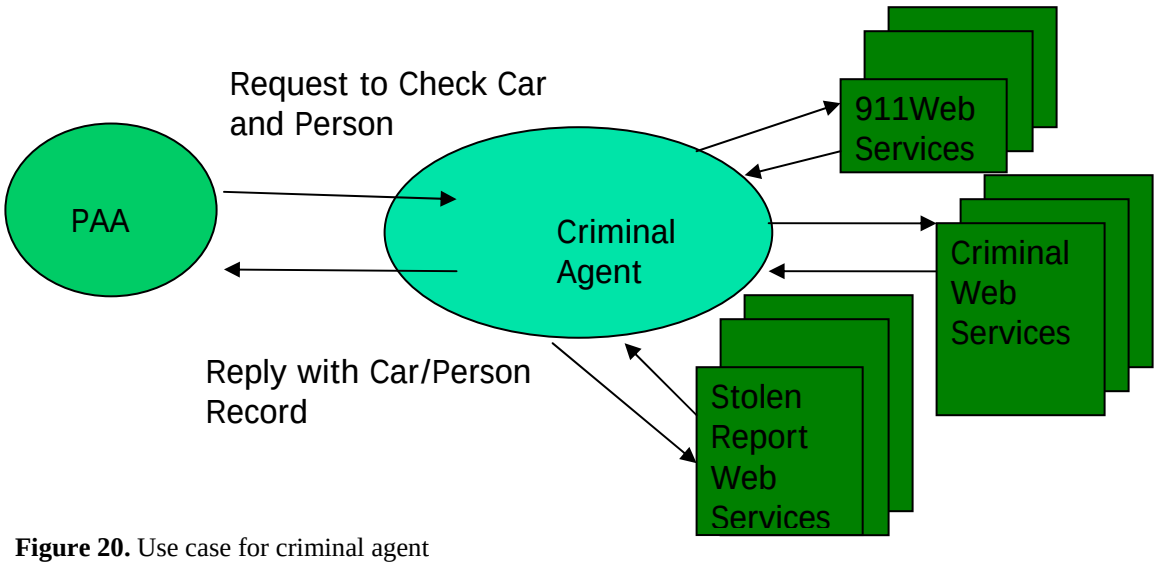

Figure 20. Use case for criminal agent

Use case Definition:

Brief Description: PAA uses this use case to check vehicle and/or driver related information whether the vehicle is involved in an accident or the driver is a suspect.
Precondition: PAA has information about vehicle and person.
Post condition: Personal and/or vehicle information is reported.
Process steps:

1. PAA requests Criminal agent to check if the vehicle is involved in an accident or the driver is a suspect.
2. PAA uses criminal agent to obtain the information from 911 emergency web services to check if the vehicle has recently involved in an accident.
3. PAA uses criminal agent to obtain the information from criminal/prisoner web services to check if the driver is a suspect.
4. PAA uses criminal agent to obtain the information from stolen report web services if the vehicle had been reported as stolen.
5. Criminal agent collects all information and sends back to PAA. If person is a suspect or the car is a stolen car, it will automatically call for help.

Exceptions:

2a, 3a, 4a, system is not accessible. Message is generated, Use case continues to next step. Criminal agent will retry to connect those web services.
5a, reporting all web services is unavailable.

Relationships:
Initiating: PAA
Collaborating: Criminal Agent.
Other Diagram:
Data Requirements
Data Required: Person information
Vehicle Information

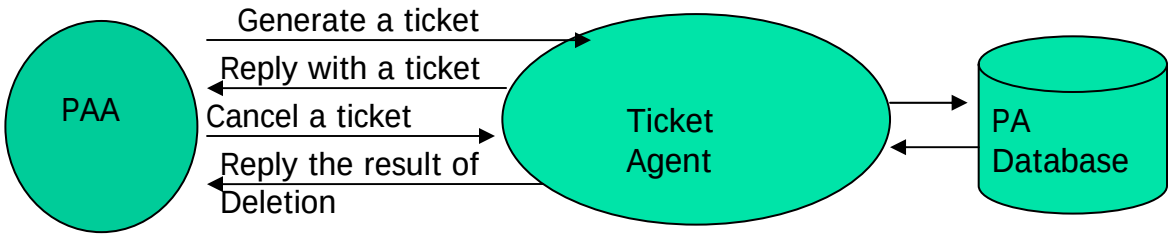

Figure 21. Use case for Ticket Agent

Use case Definition:

Brief Description: PAA uses this use case to generate or cancel a ticket

Precondition: PAA has all violation information and traffic laws and the cancellation number of the ticket. PAA gets the user inputs about the vehicles, location and its driver's information.

Post condition: Generate a ticket or a ticket is deleted in PA database.

Process steps:

1. PAA sends the request to Ticket agent
2. If the request is to issue a ticket, Ticket agent will collect information from PA database about that traffic laws and person's traffic history record, and generate a ticket, return that ticket to PAA.
3. If the request is to cancel a ticket, Ticket Agent will retrieve that ticket and delete it from PA database, return the result of deletion.
   Exceptions: 3a, If the Ticket agent cannot find the ticket to be deleted in the PA database, it flags an error message.

Relationships:

Initiating: PAA

Collaborating: Ticket agent.

Other Diagram:

Data Requirements

Data Required: Person information, Vehicle Information, Violation information, Or Ticket Number

4.3.2 Class diagram

Figure 22 shows the class diagrams for the registration and ticket classes and their relationships. The registration record class is part of the registration class and similarly the ticket record is part of ticket class.

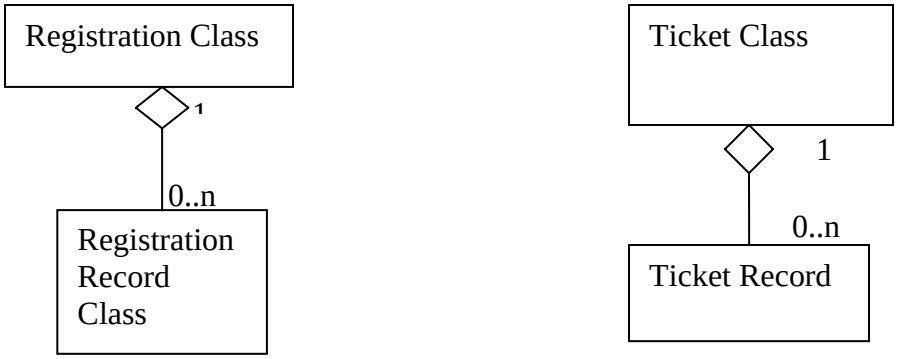

Figure 22 Class Diagrams for Registration and Ticket

Figure 23 shows the class diagram of the classes associated with criminal and their relationships.

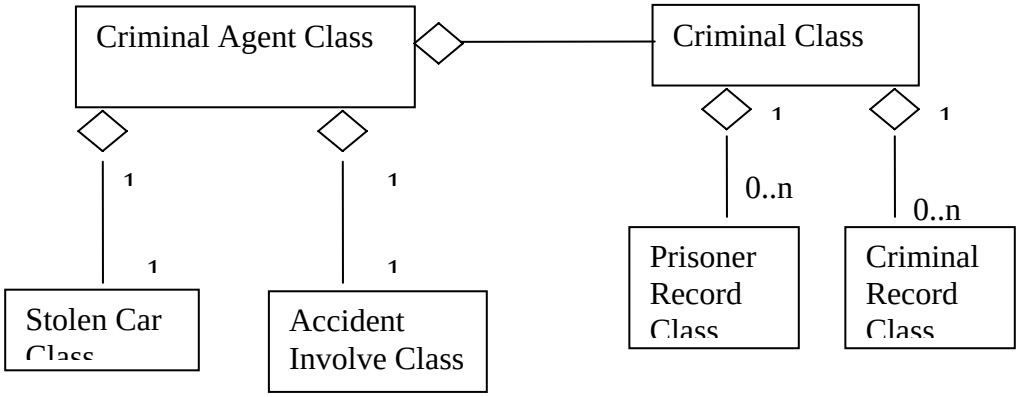

Figure 23. Class diagram for Criminal Agent

Figure 24 shows the class diagram for the PA class and its relationship with the criminal, ticket and registration classes.

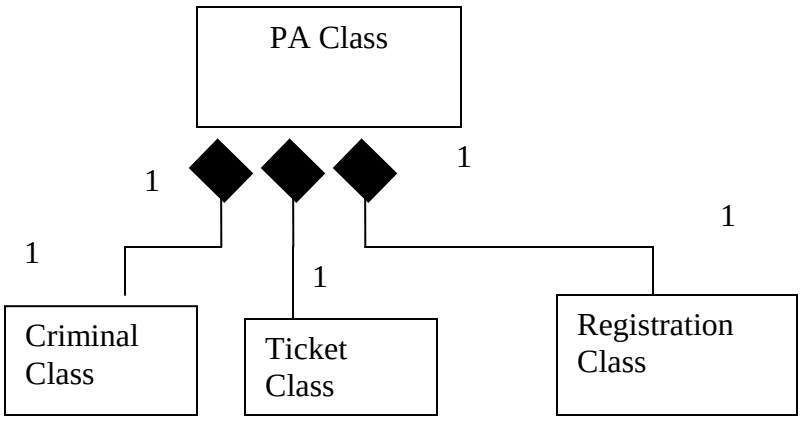

Figure 24. Class diagram for PA.

4.3.3 E-R Diagram: Figure 25 shows the ER diagram for PPAS.

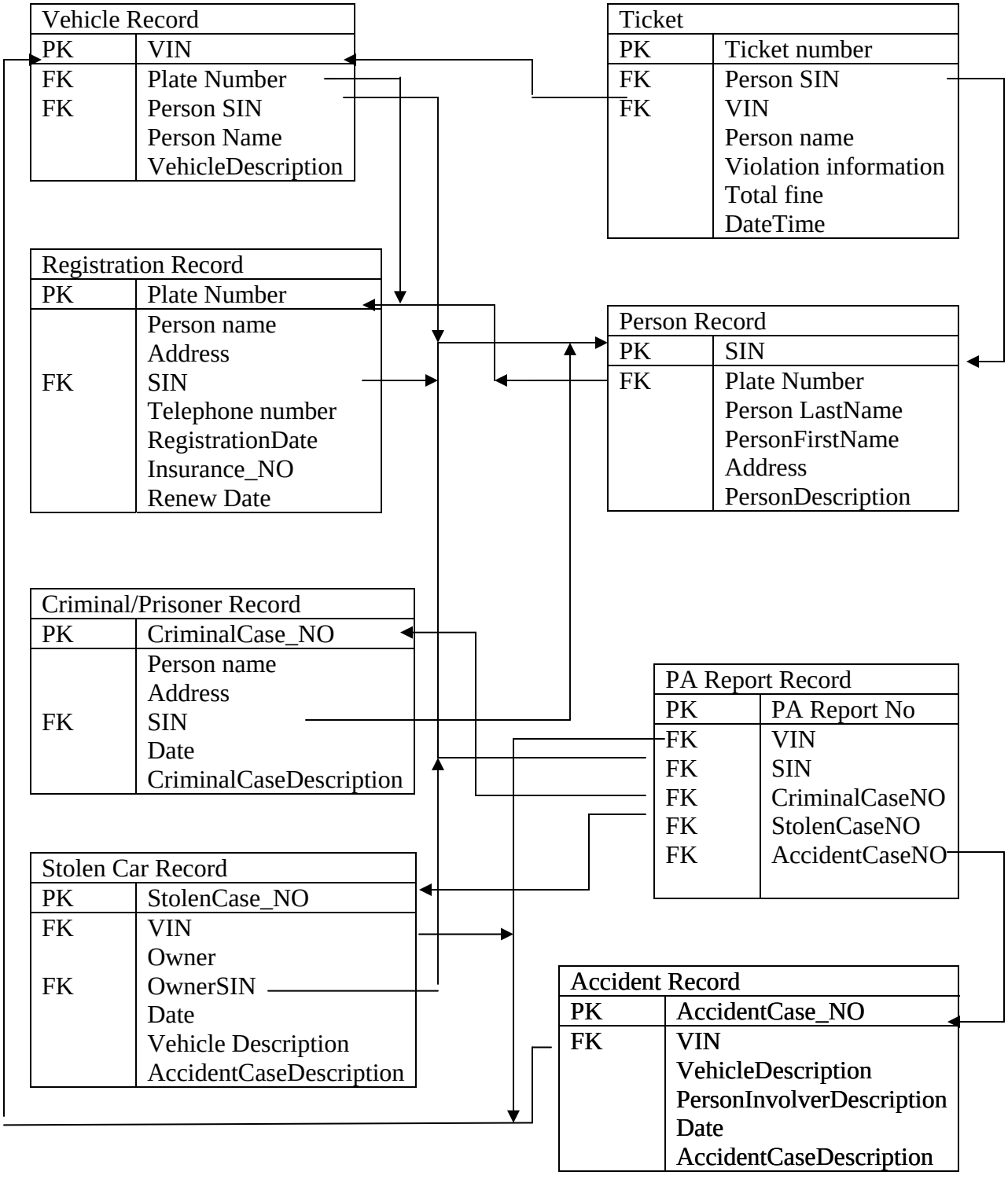

Figure 25. E-R Diagram

4.3.4 Typical Data Definition

The information related to the ticket issued is stored in Ticket record as shown in Table 3, This information is stored in the PA database. Based on the unique ticket number, a ticket is generated with the vehicle and the driver information as well as the fine amount, date, time and the description of the traffic law violation.

Table 3. Data definition of Ticket record.

Field	Description	Type
Ticket number	Unique ID	Long
Person SIN	ID	Long
Person Name	Person name	Varchar (30)
VIN	Vehicle ID	String (20)
Violation information	Violation information	String (60)
Fine	Total fine	Integer
Datetime	Date and time	date

The information regarding to a vehicle is stored in the vehicle record as shown in the Table 4.

Table 4. Data definition of Vehicle Record.

Field	Description	Type
VIN	Unique ID	String(20)
Plate Number	ID	String (20)
Person SIN	ID	Long
Person Name	Person name	Varchar (30)
VehicleDescription	Vehicle model, color and any other description	Varchar (100)

The personal information of the person (driver) is stored in the person record table as shown in Table 5.

Table 5. Data definition of Person record.

Field	Description	Type
Person SIN	Unique ID	Long
Person first name	Persons first name	Varchar (30)
Person last Name	Person last name	Varchar (30)
Person Description	Description of the person	Varchar (100)
Address	Address	Varchar (100)
Plate number	Plate number	String (20)

Table 6 shows the information of different identities used by all the classes registration, person, ticket, criminal, stolen, and accident classes is stored in the PA record table.

Table 6. Data definition of PA Report Record

Field	Description	Type
PA Report NO	Unique Report NO	Long
VIN	ID	String(20)
SIN	ID	Long
CriminalCaseNo	ID	Long
StolenCaseNO	ID	Long
AccidentCaseNo	ID	Long

Table 7 shows the information about the criminal records for a person.

Table 7. Data definition of Criminal/Prisoner record.

Field	Description	Type
CriminalCaseNO	Unique NO	Long
SIN	ID	Long
Personname	Name of person	Varchar(30)
Address	Address of the person	Varchar(50)
Date	Date of Crime occurred	Date
CriminalDescription	Description	Varchar(100)

The information regarding a vehicle accident is stored in the accident record as shown in the Table 8 identified by its unique accident case number.

Table 8. Data definition for Accident Record.

Field	Description	Type
AccidentCaseNO	Unique NO	Long
VIN	ID	String(20)
VehicleDescription	Description	Varchar(100)
PersonInvolverDescription	Description	Varchar(100)
Date	Date of accident occurred	Date
AccidentCaseDescription	Description	Varchar(100)

All the stolen vehicle information is stored in the stolen record and is identified by a unique stolen case number as shown in Table 9.

Table 9. Data definition of Stolen record.

Field	Description	Type
StolenCase_NO	Unique case NO	Long
VIN	ID	String(20)
Owner SIN	ID	Long
Owner Name	Person name	Varchar (30)
VehicleDescription	Vehicle model, color and any other description	Varchar (100)
AccidentCaseDescription	Accident Description	Varchar(200)

All the vehicle registration information along with the owner details is recorded in the registration record as shown in Table 10.

Table 10. Data definition of Registration Record.

Field	Description	Type
PlateNumber	Unique NO	String(20)
SIN	ID	Long
Personname	Name of person	Varchar(30)
Address	Address of the person	Varchar(50)
RenewDate	Registration Date	Date
Telephone Number	Telephone number of person	Long
InsuranceNO	Number of insurance	long

4.4 Communication specification

This section presents communication specification for PPAS represented by message sequence chart shown in the Figure 26 and inter-agent messages shown in the Tables 6,7,8, 9, 10, 11.

4.4.1 Message Sequence Chart

Figure 26 shows the sequence diagram for the internal agent messaging. The sequential communication between the web services and agents is not shown.

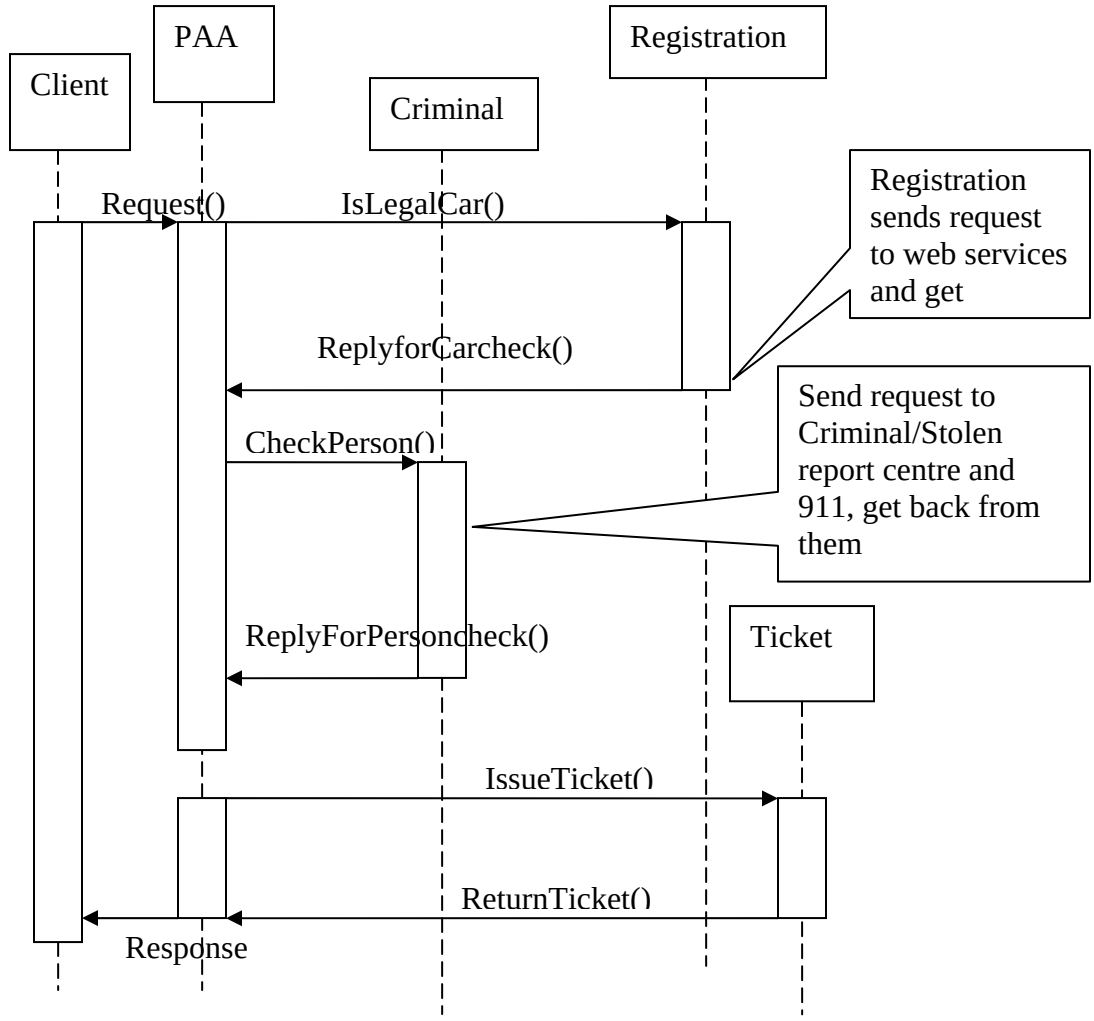

Figure 26. Message Sequence chart for PPAS

4.4.2 Inter-agents messages

As discussed in the development approach, IIOP will be used as a protocol of communication between agents and between agents and Web Services.

For each function, its input and output parameters have an XML format. These XML documents map the data structure defined in the data specification.

Table 6 shows the input parameters for PAA requesting ticket agent to issue a ticket. Figure 7 shows the output parameters for the IssueTicket message. The input/output parameters of the message use XML format.

Table 6. Input parameters for Issue Ticket

Input Parameter	Description
<PAA> <Person> <SIN>Long</SIN> <First Name>String</FirstName> <LastName> String</LastName> </Person> <ViolationDescription> String</ViolationDescription> <Vehicle><VIN>string</VIN></Vehicle> <Date>Date</Date> </PAA >	Ticket Related information

Table 7. Output parameters for Issue Ticket

Output Parameter	Description
<Ticket> <TicketNO> Long </TicketNO> <Person> <SIN>Long</SIN> <First Name>String</FirstName> <LastName> String</LastName> </person> <ViolationDescription> String</ViolationDescription> <Vehicle><VIN>string</VIN></Vehicle> <IssueDate>Date</IssueDate> <TotalFine>Integer</TotalFine> </Ticket>	Ticket and fine

Table 8 shows the input parameters for the PAA requesting criminal agent to check the personal records of the driver.

Table 8. Input parameters for IsCriminal

Input Parameter	Description
<PAA> <Person> <SIN>Long</SIN> <First Name>String</First Name> <Last Name> String</Last Name> </person> <Vehicle> <VIN>String</VIN> <VehicleDescription>Varchar</VehicleDescription> </Vehicle> </PAA>	Input person ID

Table 9 shows the output parameter for the IsCriminal message.

Table 9. Output parameters for IsCriminal

Output Parameter	Description
<Criminal agent> <IsSuspect>Boolean</IsSuspect> <Person> <SIN>Long</SIN> <First Name>String</FirstName> <LastName> String</LastName> <CriminalDescription>string</CriminalDescription> </Person> </Criminal agent>	Personal information, if the driver is a criminal

Table 10 shows the input parameters for IsLegal message where the Registration Agent uses vehicle ID input from PAA and Table 11 shows the output parameters for IsLegal message where vehicle related information is returned.

Table 10. Input parameters for IsLegal

Input Parameter	Description
<PAA > <Vehicle> <VIN>String</VIN> <PlateNumber>String</PlateNumber> </Vehicle> </PAA >	Vehicle ID

Table 11. Output parameters for IsLegal

Output Parameter	Description
<Registration agent> <Vehicle> <VIN>String</VIN> <PlateNumber>String</PlateNumber> <OwnerSIN> Long</OwnerSIN> <VehicleDescription>string</VehicleDescription> </Vehicle> </Registration agent>	Vehicle information

5. Conclusion and Future Work

In this work, we propose to develop a Police Patrol Assistant System (PPAS) by employing multiagent systems (MAS) concepts based on Gaia methodology. The PPAS application is designed in a scaleable manner so that it can easily be extended at a later date to accommodate more agents for assisting the police officers to gather information from other sources that are not covered by the present set of requirements. The analysis of the PPAS has been done using the role model that includes identification of roles and the interactions among the roles. In the design phase, agents have been assigned with roles along with the services they provide. The aspects related to the system specification and knowledge sharing have been captured by formulating a detailed set of use cases, class diagrams, E-R diagram, and data definition tables. Inter-agent communication is represented using sequence diagram and input/output messaging is established using Extended Modeling Language (XML) schema. As part of future work, we propose to implement the PPAS system using OO programming model in conjunction with Orbix (an enterprise CORBA solution) as the development platform. When implemented PPAS will help police officers to make on the spot decisions about various crimes and criminals.

6. References

1. John Naisbitt, author of 1982 bestseller, “Megatrends”
2. M. Wooldride, N.Jennings, and D. Kinny, “The Gaia Methodology for Agent-Oriented Analysis and Design”, Autonomous Agents and Multi-Agent Systems, 3, 285-312,2000
3. B. Far, Sample project of Course 609.22: TAS
4. B. Far, Lecture notes of Course 609.22
5. CORBA --http://www.omg.org/gettingstarted/corbafaq.htm
6. XML –http://ww.w3.org/XML/ and http://www.xmlblaster.org/
7. Orbix -- http://www.iona.com/products/orbix/welcome.htm