

UNIVERSITY OF NEBRASKA - LINCOLN

Driving Intelligence System

CSCE489 Senior Design Project Final Progress Report

Team Helicopter Hats
Dan Cromer, Zach Miller and Jeremy Tate
4/28/2009

ABSTRACT

This report details the final status of Team Helicopter Hats' Senior Design project. Included is the design concept and analysis of design choices and architecture, implementation of the project, testing, difficulties as well as limitations on the project. A cost analysis is also provided.

1	Introduction	3
2	Current State of the Art	3
2.1	“Sentience”	3
2.2	Intelligent Transportation Systems.....	3
2.3	Intelligent Speed Adaption	4
3	Scope.....	4
3.1	In-Vehicle Unit.....	4
3.2	Unit Web Communication Layer	4
3.3	Web Service	4
3.4	Traffic Signal Web Communication (Not in Scope).....	4
3.5	Traffic Controller Web Interface (Not in Scope)	4
3.6	Justification of Scope	5
4	Design.....	5
5	Architecture	6
6	Assumptions.....	6
6.1	City-Wide Wireless Communication	6
6.2	Traffic Information System	7
7	Implementation	7
7.1	Ideal Implementation.....	7
7.2	Actual Implementation	7
8	Testing.....	7
8.1	Test Driven Development (TDD)	7
8.2	JUnit Testing.....	8
9	Difficulties and Limitations	8
9.1	Difficulties	8
9.2	Limitations.....	8
10	Cost Analysis	9
10.1	In-Vehicle Unit.....	9
10.2	Networked Traffic Lights.....	9
10.3	Driving Intelligence Server	9
11	Team Organization.....	9
12	References	10

1 Introduction

The Driving Intelligence System aims to reduce wasteful driving by exposing some traffic light timing information to individual vehicles. Specifically, the system will use the light timing information to tell drivers how fast they should travel in order to hit a green light. This knowledge will allow drivers to make more intelligent decisions and adjust their habits to increase efficiency. Ideally, this project will:

- Increase fuel efficiency by helping drivers optimize their vehicle's speed
- Decrease wasted energy
- Reduce traffic congestion by increasing traffic throughput
- Encourage safe driving habits
- Reduce carbon emissions

This project also falls in the class of mobile location-aware software, leveraging the convenience of mobile technology with the future's robust Internet communications architecture to deliver targeted intelligence to specific mobile devices quickly and efficiently, based on their locations.

2 Current State of the Art

The team researched existing efforts to modernize the world's traffic infrastructure.

2.1 "Sentience"

A study to determine the effectiveness, practicality, cost and efficiency of advanced communication with light intersections to determine ideal vehicle speed for maximal fuel efficiency is being conducted by the U.K. government (1). The *Sentience* project integrates tightly with the vehicle being driven and actually adjusts the speed to optimal. Though *Sentience* accomplishes the same end-goals as the solution developed by our team this year, it has two major drawbacks:

1. Vehicle Control. Instead of simply telling drivers the optimal speed, *Sentience* requires each vehicle to be fitted with a computer that automatically adjusts the vehicle's speed. With the Driving Intelligence System, however, drivers are always in control of their vehicles and are only advised of their optimal speed--which can be easily ignored.
2. Higher cost. Vehicles must be fit with a *Sentience* system, which is a significantly higher cost to entry than the team's proposed driving intelligence system. This is especially true for owners of large fleets of vehicles—it's much easier to fit each truck in a fleet with a cell phone than a system capable of controlling the vehicle.

2.2 Intelligent Transportation Systems

Intelligent Transportation Systems refer to the overarching effort to computerize the transportation industry. Of particular interest is the existing effort to computerize and network traffic signals. Systems that use this idea are starting to be implemented in metropolitan areas around the country (2) (3). The primary effort in this area is to network the intersections together in order to centrally analyze traffic patterns and adjust the light timings to promote greater throughput of traffic. The team decided that

this idea was not as interesting partly for the reason that this is a problem that has already been studied extensively and there are already metropolitan areas that use ITS.

2.3 Intelligent Speed Adaption

Intelligent Speed Adaption refers to the process by which sensors detect the speed of a moving vehicle; and, if the vehicle is found to be speeding, a system is activated to alert the driver that he or she should slow down. These can be found on many major roadways, even in Lincoln, Nebraska. This is obviously not related to our senior design project, but was researched to illustrate that our project is *not* an intelligent speed adaption system.

3 Scope

The project scope can be divided into five modules (as shown in Figure 2), described below.

3.1 In-Vehicle Unit

This portion of the scope comprises the entire Android application, less the web communication piece. It is written in the Google Android environment, using primarily Java.

This application is responsible for:

- Tracking vehicle location
- Initiating web communication
- Identifying upcoming traffic signals
- Performing driving optimization calculations
- Notifying the user of driving adjustments

3.2 Unit Web Communication Layer

This module, also residing on the in-vehicle unit (though considered a separate scope piece), will carry out the actual communication between the in-vehicle unit and the web service. It is also written in Java.

3.3 Web Service

A web service must exist to handle the ultimate exchange of information between the traffic signals and the drivers. It is written in PHP and exchanges SOAP messages.

3.4 Traffic Signal Web Communication (Not in Scope)

A communication layer would need to be written for the traffic signal so that it could publish its timing information to the web service. Since the traffic controller wireless interfaces do not yet exist, this module is out of scope.

3.5 Traffic Controller Web Interface (Not in Scope)

Traffic Signals would require a piece of hardware capable of communicating with the web service over the Internet. This could be using wireless or wired technologies. The team assumes that most cities of the future will have networked stoplights as part of the larger nation-wide traffic information system movement. This piece is out of scope.

3.6 Justification of Scope

Of the five required scope items above, the team had decided to focus on the user-facing items instead of the infrastructure-related items. This is for a number of reasons. First, many successful projects start with the focus on what the user sees and interacts with instead of the infrastructure that is necessary to support it. Second, the algorithms that the team developed reside on the user side of the project, making it a logical progression of the project. Third, the time between idea conception and a working prototype for user devices is much smaller and more rewarding than for infrastructure items. Lastly, the team has concluded that it is a reasonable idea to conclude that cities (or contractors working with the cities) will deploy the necessary support infrastructure to network traffic lights together. The Driving Intelligence System would simply build off of and make use of that infrastructure.

4 Design

The system's high level design can be broken into two main segments: the application running on the phone inside the vehicle, and the web service that stores information about traffic lights. The web service is not particularly complicated; it simply accepts a query string detailing a vehicle's current location and returns a list of nearby traffic lights. The majority of the project's complexity can be attributed to the on-phone, or In Vehicle, application.

The In Vehicle component of the project is the main thread of the program and is responsible for directing the communication between the other components, described below.

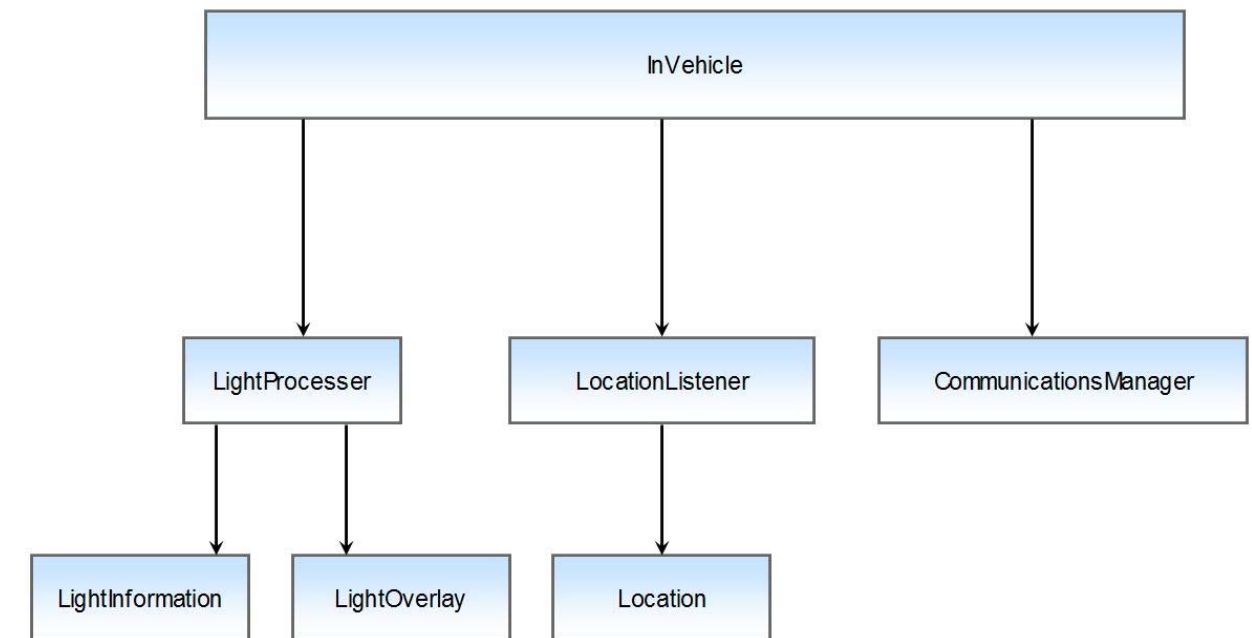


Figure 1: The class structure of the InVehicle application.

1. LightProcessor - The LightProcessor is responsible for determining which light is the closest to the vehicle and using that light's timing information in determining optimal speed.

2. LocationListener - The LocationListener is responsible for communicating the current GPS coordinates of the vehicle to the CommunicationsManager.
3. CommunicationsManager - The CommunicationsManager is responsible for all communications with the web server. It sends the vehicle's GPS coordinates and bearing to the server and handles the returned light timing information.

5 Architecture

The system's overall architecture, at a high level, is fairly simple and can be expressed in the diagram below.

Figure 2: The system architecture.

Communication takes place over the Internet. Traffic lights are responsible for notifying the server of their timing information. The In Vehicle application, running on the phone, queries the driving intelligence server using a web service.

6 Assumptions

6.1 City-Wide Wireless Communication

This project is designed for a "future city" where wireless, broadband internet access is available to all vehicles and traffic signals are networked together. Wi-MAX and 3G are two alternatives that cities may

choose to implement. Regardless of the implemented network, city-wide wireless is essential for the success of this project.

6.2 Traffic Information System

As detailed in section 5, this project assumes that, not only are all traffic lights networked, but all traffic lights will have the means to publish their timing information to a centralized location. We are implementing this centralized location as a web service, and populating it with sample data to simulate the presence of these futuristic traffic lights. While this information is not currently readily available to interface with, there are initiatives to make use of this information for the use of altering the light timing information based on traffic conditions (2). If systems such as that detailed in (2) are available, major roadblocks would be removed.

7 Implementation

7.1 Ideal Implementation

If our assumptions for needed support infrastructure existed, it would be possible to implement the system on actual roads and begin to realize the savings in fuel, energy and emissions that it provides.

7.2 Actual Implementation

Without the necessary support infrastructure for publishing light timing information to the Driving Intelligence Server, the project demonstrated its feasibility through the use of a simulator. Expanding on the traffic simulator that was developed for the mini-project requirement of CSCE 488 in the fall of 2008, the capabilities of the device emulator for the G1 were used to mimic the ideal system and provide large-scale deployment performance estimations. In short:

- The traffic simulator sends GPS coordinates to the phone emulator.
- The phone emulator sends those GPS coordinates to the Driving Intelligence Server.
- The server returns the nearby traffic light information as well as distance to the lights.
- The phone emulator determines the ideal speed for the driver and sends that information to the simulator.
 - The phone emulator displays the ideal speed on the screen.
- The simulator adjusts the speed of the rendered test vehicle such that it appears that the ideal driver is obeying the emulator instructions.

8 Testing

8.1 Test Driven Development (TDD)

Test driven development is the idea that instead of starting a coding project by coding, a series of tests is created first. By creating these tests, several assumptions are made that influence the way the project is created. The code that is then written is only done so in order to help pass a currently failing test. By having this seemingly “backwards” approach to development, bugs, errors and other problems are

found much more quickly and the offensive code much more readily identified. By following this practice for parts of our project, there have been fewer errors introduced into the code, development has been faster and the overall product is of higher quality. In the business world, this translates into a lower development cost and more money staying with the company, both of which are very beneficial.

8.2 JUnit Testing

In conjunction with TDD, JUnit tests have been written for the main algorithm portion of the project in different scenarios to prove the correctness and robustness of our project. The team determined that because we were advising drivers how fast they should drive, possible errors in our application could literally mean people dying. For this reason, unit tests were written to minimize that possibility.

9 Difficulties and Limitations

9.1 Difficulties

The team found that the XML parser used to deserialize web service communication performed poorly when more than 1 kilobyte of data was received. Because unit testing was conducted on a powerful laptop, which could still perform well even when using a poor-performing parser, and load testing was conducted through unit tests, the team did not expect the less-powerful phone hardware to perform significantly worse. A two kilobyte response took over 15 seconds to parse. Because the simulated drivers were not receiving information about traffic lights until 15 seconds after the data was received, they frequently ran red lights. After identifying this bug, the team switched to a better-performing XML parser.

The team also discovered that high latency in the internet connection would significantly delay the results of traffic information from the server and reduce its effectiveness. Because this is outside of the team's control, it could not be resolved.

9.2 Limitations

This solution is intended to be a proof-of-concept, demonstrating that a system like this is both physically realizable and provides the necessary benefits to justify its future implementation.

1. Reliance on Future Technologies - While this solution provides significant benefits to drivers and cities alike, the supporting infrastructure is not yet mature. The team is confident, however, that the assumptions detailed in section 6 will be realized within five to ten years.
2. No Support for Sensor-Driven Lights – Currently, the system only handles timed traffic lights. This was a scope decision intended to simplify the project, as it is a proof-of-concept. Any intersection that changes lights based on the current traffic via some sort of sensor were considered outside of this project's scope, though, it should be noted, the system is designed with this particular extension in mind.
3. Limited Unit Testing - In order to develop a high quality product, test driven development has been implemented for as much of the code base as possible. While this is a good engineering practice to develop quality work in a short amount of time, the Android specific code that is

used in the project is not capable of being unit tested. This leaves only the code responsible for the main algorithm capable of automated testing.

10 Cost Analysis

There are three main pieces that must be put in place to realize an actual driving intelligence system. The individual user would only be responsible for first of these three pieces. The remaining two would be costs handled by a city or organization.

10.1 In-Vehicle Unit

One of the primary innovations of this project was to make use of cell phones that have GPS and wireless broadband internet capabilities. By making use of cell phones, there is almost no cost for the units in the vehicles. The only cost associated here would be the cost of the application that runs on the cell phone itself which would be minimal, less than \$10.

10.2 Networked Traffic Lights

According to reports such as (2) and (3), the cost of updating the traffic lights to be centrally controlled range from \$2,000 to about \$3,000 per intersection. This cost analysis was done with the intention of linking all the intersections to a central computer system to dynamically change the timing information based on time of day as well as current traffic patterns.

Listings such as those in (4) put the cost for a small scale metropolitan traffic light upgrade to be several hundreds of millions of dollars.

10.3 Driving Intelligence Server

While no specific information was found about the cost of the controlling computer system behind the networked intersections, it would be reasonable to assume that the cost of the system was grouped into the overall cost of the project.

11 Team Organization

The team organization and structure was broken down based on strengths of each individual with overlap occurring when certain milestones were achieved in one area.

Dan Cromer: Unofficial team leader and organizer of events and work periods, specialist in Java for the Android and owner of the demonstration platform.

Zach Miller: Programmed some of the InVehicle application and most of the simulator, including integration with the web service.

Jeremy Tate: Created the web service, managed the traffic light database, and debugged traffic light data.. Helped out where and when he could.

It should be noted, however, that the team made extensive use paired programming. Roughly half of the programming tasks were completed in pairs, which helps ensure quality and familiarize other members of the team with sections of the system they wouldn't normally work on.

12 References

1. **Ben Webster, Transport Correspondent.** Sentience, the smart sat-nav that sees the red light before you do. *TimesOnline*. [Online] March 5, 2009. [Cited: April 20, 2009.]
<http://www.timesonline.co.uk/tol/driving/article5847252.ece>.
2. **Denver Regional Council Of Governments.** *Traffic Signal System Improvement Program*. Denver : s.n., 2007.
3. **Metropolitan Transportation Commission.** *Regional Signal Timing Program*. Oakland : s.n., 2008.
4. **Research and Innovative Technology Administration.** Advanced Signal Systems Cost Documents. *Intelligent Transportation Systems*. [Online] [Cited: April 24, 2009.]
<http://www.itscosts.its.dot.gov/its/benecost.nsf/SingleLink?OpenForm&Tax=Intelligent+Infrastructure+Arterial+Management+Traffic+Control+Advanced+Signal+Systems&Location=Cost>.