**Current ITS Product Description**

Current ITS – Red Team

Old Dominion University

CS 411 - Brunelle

Author: Nathan Lutz

Last Modified: September 24, 2012

Version: 3.0

Contents

[1. Introduction 3](#_Toc336290835)

[2. Current ITS Product Description 4](#_Toc336290836)

[3. Key Product Features and Capabilities 6](#_Toc336290837)

[7](#_Toc336290838)

[4. Major Components (Hardware/Software) 7](#_Toc336290839)

[5. Target Market/Customer Base 10](#_Toc336290840)

[6. Current ITS Prototype Description 13](#_Toc336290841)

[7. Prototype Functional Goals and Objectives 14](#_Toc336290842)

[8. Prototype Architecture (Hardware/Software) 15](#_Toc336290843)

[9. Prototype Features and Capabilities 17](#_Toc336290844)

[10. Prototype Challenges and Risk 19](#_Toc336290845)

[Glossary 21](#_Toc336290846)

[References 22](#_Toc336290847)

# Table of Figures

[Figure 1. A sample transit authority web interface 5](#_Toc336291235)

[Figure 2. Major Functional Component Diagram 7](file:///C:\Users\Nathan\Documents\cs411\lab1v3.docx#_Toc336291236)

[Figure 3. Costs of developing a Light Rail system 10](file:///C:\Users\Nathan\Documents\cs411\lab1v3.docx#_Toc336291237)

[Figure 4. Prototype Major Functional Component Diagram 15](file:///C:\Users\Nathan\Documents\cs411\lab1v3.docx#_Toc336291238)

[Figure 5. Prototype Software logic flow 17](#_Toc336291239)

[Figure 6. Sample Business Web GUI 18](#_Toc336291240)

(This space intentionally left blank)

## Introduction

Transit authorities around the country face a multitude of problems when beginning to implement a light rail system to service their host populations. Not the least of these is accurate, real-time communication between the rider, the local businesses that surround the light rail, and the transit authorities themselves. Unfortunately, when planning these systems, often the cost associated with them leaves certain features unfinished or completely left out. In the case of Hampton Roads Transit (HRT) and the Tide, there is no way for riders to accurately know the following: where the trains are relative to them, how occupied the trains are, and any potential problems along the rail line that may impact their travel.

HRT also has no way to accurately track where the trains are, even in downtown Norfolk, a densely populated area. If a train de-rails, human operators have to manually call in the event, which then is relayed to the control center located near Norfolk State University (Messina, 2011). There is no way for those affected by the derailment to reroute themselves to their destination using HRT’s other services, such as the busses. HRT has no way to accurately size their deployment of trains with their existing system, which results in wasted man hours and potentially under/over provisioning of trains.

The local businesses along the line wish to attract customers who ride the light rail, but have no way to target the riders in a meaningful way. This results in a loss of riders who are frustrated about a lack of information on this new transportation service that could otherwise complement the already overburdened highway system in the area, a loss of revenue for HRT, who has to constantly justify their light rail project to the local tax paying populace, and a loss in potential revenues for the local businesses along the rail line (SIR 2011). This leads to potential problems when it comes to expansion of the rail line into neighboring Virginia Beach, whose main motivator for doing so would be in the benefits the rail line could have to its local businesses and large naval population.

Current intelligent transportation system (ITS), is developed with one thing in mind, free flow of information between the three actors in this picture: the rider, HRT, and local businesses. The results are riders who ride more often, transit authorities who have a good handle on their efficiently running system, and prosperous local businesses who benefit from educated riders visiting their stores.

## Current ITS Product Description

Current ITS is a highly modular system, accommodating any existing ITS infrastructure that the transit authority has deployed. Primarily, it works by relaying real-time global positioning system, (GPS) and automatic people counting (APC) information about the train system to the riders and to the transit authority through a web interface. In addition to the web, riders with smartphones will be able to use a mobile application to view the status of trains, local events and places of interest, routes to their destination, as well as real time alerts. This provides a platform for instant feedback to the transit authority. For HRT, collection of this GPS and APC data allows for trending and prediction in order to accurately provision train cars according to demand. This prediction can also show where riders are going to most often, allowing for future expansions to be more quickly determined. Current ITS helps local businesses target riders who are headed near their place of business by providing a backend platform for uploading and tracking advertisements for their business. These adverts are then deployed through the web interface to the various users (i.e. the mobile application and kiosks near stations.) Ultimately, these features save the riders from wasting their time on a transit solution they know nothing about, transit authorities from running an inefficient and incomplete service, and assists local businesses in attracting customers, helping them inject money and life into the local economy.



Figure 1. A sample transit authority web interface

(This space intentionally left blank)

## Key Product Features and Capabilities

As was mentioned previously, Current ITS consists of three interfaces that the end users will interact with. The first is the interface that the transit authority will use to track their trains data. A mockup of this interface can be seen in Figure 1. This is a website that will have an overview of all trains currently in service, along with a visual map with the current positions of the trains so that operators know exactly where the train cars are, their current speed, proximity to the next station, occupancy levels, and any other custom data that they wish to collect. This website will also have a reporting function, to track up-to the minute data on how closely trains are adhering to schedules, occupancy levels per station, per minute, per hour, per route, etc. This data will serve as the basis for the prediction engine that can advise transit authorities on how better to provision trains on a minute-to-minute basis, according to historical data. The second interface serves the local businesses: allowing for the registration of adverts, paying of fees for those ads, statistics on the adverts such as click-throughs, and viewing statistics. Local businesses can also input events that will be happening that might affect ridership numbers (for example, a hockey game, that could result in the light rail system being overwhelmed). The mobile app is the third interface, which serves the riders. This can be used to plan a route, purchase tickets, receive notifications of upcoming events and businesses close to their intended destination, as well as real time alerts of potential issues with the light rail. The riders can then communicate back to the transit authorities through this interface in emergencies or other situations that would call for it.

(This space intentionally left blank)

## http://www.mimicemore.com/hr/wp-content/uploads/2011/02/www.jpg



**DB**

Onboard Unit

Infrared Counters

GPS Transponder

Decision

Engine

GTFS

Web App

Server





Figure 2. Major Functional Component Diagram

## Major Components (Hardware/Software)

As can be seen in Figure 2, the basis for Current ITS begins in the onboard sensor network. This consists of two sensors: the GPS unit and the APC unit. These report both GPS location data and occupancy data on a regular, tunable basis to an onboard computer, the Master PC. This PC is a single board computer, running a FOSS (free and open source software) OS, in this case CentOS Linux. This PC has a reporting agent that communicates with the database server located at the transit authority’s datacenter, over a GSM wireless network. This data is transmitted using the Google Transit Feed Specification, an industry standard in transit reporting. This allows for the web application to be developed quickly, and to fit many transit authorities’ data, even if they are already recording it themselves. The database then stores this information in a format that will facilitate further reporting and trending.

The second step in Current ITS requires prediction – and for that purpose a decision engine will be developed. This decision engine resides on a physical server in the transit authority’s datacenter along with the web application engine. This engine polls the database for the state of the train system, and begins to make predictions based on current and past GPS and occupancy data using a learning algorithm. This happens constantly, so reports can be made by the transit authority at any time. The Decision Engine also keeps an up to the minute picture of the entire network of trains, so routing information can be prepared to be sent to the end user interfaces. This data, along with the current GPS and APC data stored in the database, can be polled by the web application server, the lynchpin of the reporting phase of Current ITS. The web application server will provide the interfaces for the transit authority, the rider, and the local businesses. The rider will have the option to go online, use their mobile Android device, or use a kiosk at a station to access the information. This allows for most riders to be addressed using a single system.

Powering all of this are several algorithms that we will develop. The first is the embedded reporting agent. This agent will receive data from the GPS and APC sensors over a RS232 serial or USB interface, and format it to the GTFS before sending it in plaintext over the Global System for Mobile Wide Area Network (GSM WAN) to the database. To generate reports and advise riders on appropriate routes, a machine learning algorithm will be created to learn from the day to day operation of the light rail. This means polling the database, associating times with occupancy levels, events, incidents, and positions of the trains and then saving these to forecast tables which can be used to trend out future events. These forecasts will be reported to the web application engine.

Secondly, there will be an intelligent routing algorithm, which will establish a network of nodes consisting of the trains and stations by polling the database. It will then assign weights to these nodes, according to constraints put on it such as schedule constraints, capacity constraints, or interruptions of service. It will then update its weighted network and wait for a routing request from the web application server. This request will consist of a beginning point and an endpoint, both GPS coordinates. These coordinates will be associated with the closest nodes in the network. The database will be polled for moving entities that intersect this path, such as moving trains or busses, and then a shortest path algorithm will determine the fittest path using the weighted nodes. Once this path has been determined, it will return it to the web application engine for reporting to various end user interfaces.

The web application will also have several algorithms powering it. The first is syndication to the Google Places API. This will allow riders to view their location and the location of trains through a Google Maps interface, as well as locations of points of interest along their route. The web application engine will also have authentication methods in place to ensure security of data, as well as to facilitate per-user tracking and trending. In order to implement context-aware advertising, the web application engine will also need to associate adverts with the locations they represent, and then couple them to the rider interface appropriately. All of this will be hosted on commodity servers, using largely open source software. This allows us to drive down development costs, while still providing the bleeding edge in functionality to potential customers.

## Target Market/Customer Base

The initial customer for Current ITS is Hampton Roads Transit, as they are a perfect example of an emerging light rail provider. They serve the greater Hampton roads area with a ridership of nearly one and a half million people. They spent over $318 million on developing the Tide over the past 4 years, and hope to expand it throughout Hampton Roads by 2034.

Light Rail Project Costs

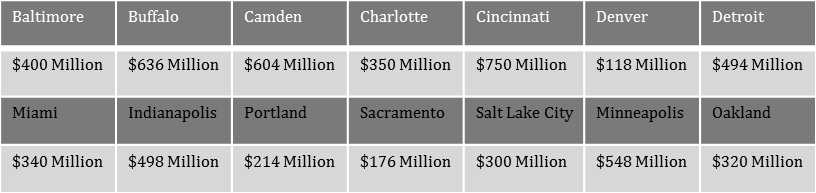


Figure 3. Costs of developing a Light Rail system

A recent study of over 1000 Norfolk residents and business owners shows that though 90 percent of those surveyed were aware of the Tide, 70 percent of those that worked downtown did not know the stop locations, 55 percent of other respondents did not either. (SIR, 2011) In contrast, 69 percent of those surveyed ranked information about stops and 75 percent rated information about schedules as a very important problem. (SIR, 2011) This data shows that though awareness is high, convincing these people that the Tide will benefit them will take some doing. Current ITS hopes to address these deficiencies by providing real-time, accurate information to the riders on behalf of HRT.

Current ITS will benefit HRT in the following ways: first, it will increase ridership. Studies show that rider retention rates increase when availability of information to them is high. This translates directly to higher revenues for HRT. Current ITS will provide an easy access to the information they need to make informed decisions going forward with Tide expansions, allowing them to directly correlate dollars spent to the efficiency of their system. It also provides a two way communication channel between HRT and the rider. Previously, HRT had to conduct town hall meetings to address concerns and plan route changes, which could take weeks to implement. With Current ITS, this feedback is instant, and the changes can be put into production within days.

HRT faces another challenge in expanding the light rail: Virginia Beach is the next logical step in expansion, yet Virginia Beach council members have remained skeptical. This attitude is beginning to change, but HRT has the burden of proving their worth. Current ITS hopes to assist in this purpose by modeling its solution using the Tide in Norfolk as its initial target, then expanding to the greater Hampton Roads area and eventually the entire US. The US market for Light Rail is growing: more than 35 light rail systems are currently active and running, and 60 more are proposed or in development (US DOT, 2011). The Obama administration has invested more than $8 billion in stimulus funding for rail transit, so now is the perfect time to enter the market as many mass transit projects are underway (Brent, 2009). Past the US market, global light rail market is estimated at $7.5 billion by 2015 and is expanding rapidly. Current ITS can address all of the issues that emerging light rail systems face, by providing flexible and affordable integration into existing mass transit systems, while being scalable enough to eventually manage large deployments in cities such as New York and Tokyo.

(This space intentionally left blank)

*Table 1. Side-by-side comparison of Real World Product (RWP) and Prototype*

|  |  |  |  |
| --- | --- | --- | --- |
| **Hardware** | **RWP** | **Prototype** | **Functionality** |
| Automatic Passenger Counter (APC) | IRMA Matrix infrared sensor | Simulated | Partial |
| GPS Antenna | Garmin GPS 18x | Static Android GPS Data | Partial |
| Embedded Computer System | Habey BIS-6620-IV-Z530 | Omitted | N/A |
| 3G Modem | Novatel MC935D | Omitted | N/A |
| Electronic Signage | US Stamp & Sign Electronic LED | Omitted | N/A |
| Physical Server | Dell R710 | CS Dept Virtual Machine | Full |
| **Software** | **RWP** | **Prototype** | **Functionality** |
| Virtualization Software | RHEL KVM | Omitted | Partial |
| Operating System Software | Red Hat Enterprise server | CentOS server | Full |
| Web Application Engine |  |  | Partial |
| Web GUI | Administrative Interface, Schedule Delays, Rail Capacity, Rider Feedback, Module Ridership counts, and Local Event Calender |  | Full |
| General Request Handler | Capacity Check, Accept Feedback, Retrieve Schedule, Local Destinations, and Retrieve Forecast |  | Full |
| Database I/O | Rider Feedback | Rider Feedback | Full |
| Syndication process | Google Places API Checker, and GTFS/AJAX/ETC Publication |  | Full |
| Test Harness | Omitted | Backend GUI to simulate various scenarios - i.e. sensors failure, simulated train problems, controllable occupancy levels, etc |  |
| Mobile Application |  | Omitted |  |
| Local Database | Setting and Shared Preferences | Omitted |  |
| GUI | Schedule Delays, Rail Capacity & Delay, Rider Feedback Module, Ridership Counts, Local Places, Local Event Calender |  |  |
| **Software** | **RWP** | **Prototype** | **Functionality** |
| Processes | UI Event Handler, GPS/Triangulation on Checker, WAE Request Interface, Rider Feedback Submission |  |  |
|  |  |  |  |
| Decision Engine | Prediction | Modified for trending data | Partial |
| Database I/O | Forecast Tables |  |  |
| Request Handler | Delay Forecast, Ridership Forecast, Optional Routes. |  |  |
| Gradient Descent Algorithm | Rider Features, Historical Features Location Features. |  |  |
| Option Route Detection | Shortest Path, Shortest Time, GTFS Interface | N/A | N/A |
|  |  |  |  |
| Linux reporting Agent | GPS Interface m APC Interface, Database I/O |  | Partial |

*Table 1. Side-by-side comparison of Real World Product (RWP) and Prototype*

## Current ITS Prototype Description

The prototype of Current ITS aims to provide the most functionally complete product it can while remaining within the constraints of the time allotted. It will be at a reduced scope and feature set than the real world product, but it will be functionally complete. This prototype’s objective is to provide the transit authorities with the information they need to make informed decisions, to assist riders in making efficient travel plans, and to provide a location-based ad platform for local business owners. Users will be able to view the current state of the trains, as well as historical data in the form of reports. In order to promote themselves and to notify the transit authorities, business owners will have an interface to post upcoming events along with their location. These features will allow all end users to save time and money.

## Prototype Functional Goals and Objectives

The goals for the prototype are to show the functionality of the RWP in a reduced fashion. Similar to the RWP, the prototype will have several interfaces that the end users will interact with. Transit Authorities will still be provided with a web GUI through which they can view the status of their trains, alerts, train’s current position and occupancy, trends in ridership and in on-time performance. Any upcoming events or trends that indicate a needed change in train provisioning will be posted to the interface for resolution. Business owners will be given access to the web application through a GUI that will allow them to experience the ease at which they can create an advertising profile, upload adverts, and arrange display and payment options. The prototype android mobile application provides the functionality to the riders. They will have the ability to view the current positions and relative occupancy of trains relative to their current position, purchase tickets, and report accidents/feedback. A test harness program will have to be created to simulate the various independent variables, such as GPS coordinates and APC data, as accurately as possible. This program will have a GUI which allows control of these variables, to introduce potential failure into the system for testing purposes.

(This space intentionally left blank)

Figure 4. Prototype Major Functional Component Diagram

CS Dept Virtual Machine



Web App

Server

GTFS

Trending

Algorithms

Simulated GPS Data

Simulated APC Data



**DB**

CS Dept Virtual Machine

## Prototype Architecture (Hardware/Software)

As can be seen in Figure 2, the major component difference between the real world product and the prototype is the lack of physical hardware. This is both for convenience of testing and for cost reasons. For the prototype, GPS sensor data on the Tide will be collected prior to the prototyping process using commodity phones. This data can then be used to generate simulated yet accurate GPS coordinates and timestamps which will be used for the prototype itself. For the APC unit, a simulated population count will be implemented to account for the natural ebbs and flows of occupancy levels on the Tide as time passes. Both of these will be incorporated into the test harness, which can be adjusted at any time to reflect emergency situations such as derailing, or unusual population increases/decreases that may occur during events downtown.

This simulated data will be received by the MySQL database, which will remain largely unchanged from the real world product design. The database will store the GPS and APC data, as well as having defined user roles to incorporate access control into the web application. For example, an Admin role will have the ability to view information about the trains, edit events or business details, and view detailed information about the train system. However, a regular end-user such as a rider will only be able to view basic information about the trains.

The decision engine in the real world product will become a reduced feature, retaining only the ability to trend out and predict ridership levels and on time performance. This gives the transit authority the ability to predict accurately when planning and evaluating. Not being implemented here is the routing algorithm, which would provide riders an alternative route should the train run into problems. However, real time alerts will still be included.

To provide interfaces to the transit authority, the rider, and to businesses, a web application will be developed. The web application will provide a singular access to the information each entity needs, and will have authentication methods in place so that improper access does not occur. This web application will make requests to the database and to the decision engine asynchronously, providing data on demand. Finally, a mobile android application will be created, to facilitate rider-to-transit-authority communication. This application will pull data from the web application server on demand.

As was mentioned, the prototype will require no specialized hardware to complete. The various applications, servers, algorithms and databases will all be hosted locally on the Department of Computer Science virtual machines. Creation of these resources will be tasked to the Systems Group, and their maintenance will fall under the CS411 group’s purview. This means that no expensive hardware will have to be purchased in order to make the prototype a reality. Several departmental PCs will also be utilized for development purposes. A development Android device may also be required, but can be emulated through software.

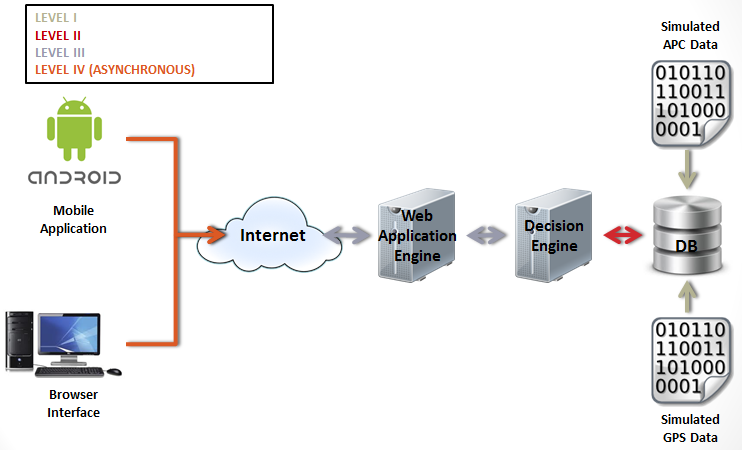


Figure 5. Prototype Software logic flow

## Prototype Features and Capabilities

One of the major features that the prototype will showcase is the ability to associate local businesses and attractions with the end user in order to better coordinate between the two. This will be done through a subscription to the Google Places API. Users can search for endpoints along the route using the same interface. After a single use of Current ITS, users of the mobile application will be able to view their past trips, and receive recommendations for travel endpoints based upon that historical data. This collected data can then be integrated into the business interface to give business owners trends and reports of how effective / popular their store is with the light rail riding populace. An event calendar will be viewable by all entities who use Current ITS. This will give the transit authorities notice to accommodate any increases in traffic that may occur, riders the knowledge of current events in their area, and businesses opportunities to expand their revenue stream more effectively.

A large feature of the prototype is the business backend interface. This is a web page with several sections that will each benefit the local business owner. They will have the ability to upload advertisements, specify what times of day and where they might appear, and view statistics on their advertisements such as click-throughs. The business user can monitor detailed statistics on the station associated with their business, such as trends in embarks/disembarks, peak ridership times, and rider feedback concerning the immediate area.

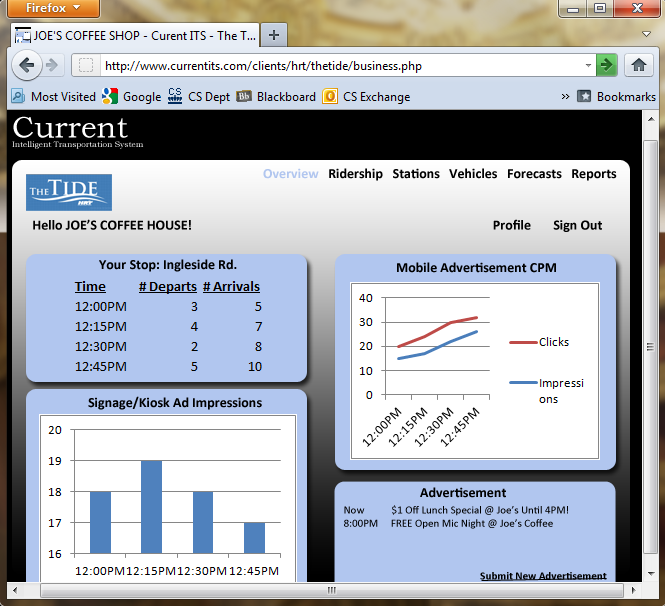


Figure . Sample Business Web GUI

Feedback to the transit authority is a key feature of the prototype and will be implemented across all interfaces. Users will have the ability to authenticate, and then submit general feedback, service ratings, report accidents or outages; or register a complaint. The transit authority will receive notification of all feedback and have the ability to respond either to the user through their profile on the application, or through email. The transit authority will also be able to generate reports based on this feedback data for self-evaluation.

Feedback is not the only report that will be needed for the prototype. Trend analysis will be necessary for the transit authority to have a complete picture of their operating environment. These trends include: ridership levels on a tunable granularity, station embarks/disembarks, feedback associated by train, station, or user; on-time performance of trains, and (Service Level Agreement) SLA compliance. In addition, prediction of future events effects on ridership will be included.

## Prototype Challenges and Risk

In order to complete this prototype, several risks have been identified. Because of the simulated nature of the prototype, it is important to make sure the data the application uses is accurate and timely. Thus, the following risks: data latency being too great, bad data, and inaccuracy of simulated real-world data. To combat data latency issues, timestamps for all sensor data will be adopted. This ensure that even if the data is not received in a timely fashion, transactions dependent on ordered data can still be completed, and that reports can be accurately generated. In addition, part of the test harness will simulate delayed data, so that its effects can be further resolved. Secondly, to mitigate data inaccuracy issues, the test harness will simulate malformed or bad data, and several error-checking processes in the database and in the web application will be constructed to ensure that these do not proliferate throughout the system or cause it to fail. Finally, to ensure that the simulated data is as close to real life as possible, prior to the prototypes development, actual GPS data from the Tide will be collected, and day to day averages of ridership will provide a limit to the range of occupancy values the application will simulate.

A challenge that the team will have to overcome is meeting the relatively short development timeline that is in place while simultaneously completing the integration of the features that makes this approach unique. Current ITS is a large solution to a complicated problem. Thus, proper team communication and task tracking will become a priority in the coming months.

(This space intentionally left blank)

## Glossary

GSM: Global System for Mobile Communications - Technology that is used in cellular devices today to enable wireless communication.

Android Google’s widely-used, open-source platform for mobile applications.

Automatic Passenger Counter (APC) - Sensor that uses infrared cameras to count passengers entering and exiting the vehicle door.

Automatic Vehicle Location (AVL) System that sends GPS coordinates of vehicles.

Database Server Hardware that stores information in a relational database management system (RDBMS).

Decision Engine (DE) Component of Current ITS that can generate trends and predictions of ridership and delays.

Global Positioning System (GPS) Satellite-based navigation system that provides location and time information from space.

Google Places Third-party source for local business locations.

Graphical User Interface (GUI) How the end user will interact with the application.

GTFS - Google Transit Feed Specification, an industry standard in transit reporting.

Intelligent Routing Algorithm An algorithm inside Current ITS that determines the best path to take in the event of a delay or other incident.

Intelligent Transportation System (ITS) general term for advanced management applications in transportation systems.

GNU/Linux A free, widely-available open-source operating system.

CentOS – Community Enterprise Linux.

Machine Learning A branch of artificial intelligence concerned with programming computers to change behaviors based on historical data and make predictions based off many types of algorithms.

MySQL – the worlds most used RDBMS.

Structured Query Language (SQL) Language used to make queries to a database server.

Tables - Logical collections of data stored on a database server.

Training Set - A collection of historical, actual data used to make predictions about future events.

Web Application Engine (WAE) - The main interactive component of Current ITS that can process various requests including DE predictions, checking delays, etc.

RDBMS – Relational Database Management System.

## References

Shapiro, C. (2012, Feb 20). *Some stores near Norfolk light rail stations see boost*. Retrieved Apr 18, 2012, from Pilot Online: http://hamptonroads.com/2012/02/some-stores-near-norfolk-light-rail-stations-see-boost

Southeastern Institute of Research, Inc. (2011). *Hampton Roads Transit: Light Rail Marketing Research Study.* Norfolk: Southeastern Institute of Research, Inc.

Katrin Dziekan, Karl Kottenhoff, *Dynamic at-stop real-time information displays for public transport: effects on customers, Transportation Research Part A: Policy and Practice*, Volume 41, Issue 6, July 2007, Pages 489-501, ISSN 0965-8564, 10.1016/j.tra.2006.11.006.

(http://www.sciencedirect.com/science/article/pii/S0965856406001431)

Keywords: Public transport traveller information; Real-time information; User reaction; Customer satisfaction; Wait time; Evaluation

HRT Ridership levels by month: http://www.gohrt.com/public-records/Operations-Documents/Rail/Monthly-Ridership/Rail-Ridership-Current.pdf

Lang, Brent. (2009, April 16). *Obama Touts High Speed Rail*. From CBS News: http://www.cbsnews.com/8301-503544\_162-4949672-503544.html

Messina, Debbie (2011, July 31). *Control room at NSU servers as brains for light rail*. From the Virginian Pilot: http://hamptonroads.com/2011/07/control-room-nsu-serves-brains-light-rail

U.S. Dept of Transportation Research and Innovative Technology Administration (2011). *Intelligent Transportation Systems Benefits, Costs, Deployment, and Lessons Learned: 2011 Update.* From itscosts.its.dot.gov: http://www.itscosts.its.dot.gov/its/benecost.nsf/ByLink/CostDocs#ITS2011

Gulf Coast Institute (2006, Aug 2) *The impact of Light Rail on Local Businesses.* From Houston Tomorrow: http://www.gulfcoastinstitute.org/university/LightRail\_BusinessImpact.pdf