ABSTRACT

AN ELECTRONIC ON-BOARD RECORDER FOR INTERMODAL DRAYAGE OPERATIONS

By

Andrew Browning

May 2012

This paper makes the case for the development of a drayage specific electronic on-board recorder that can be tested in and around the twin ports of Long Beach and Los Angeles. The goal of this device is to gather data that can then be processed and used to reduce port congestion, reduce drayage cost and reduce port-related air pollution. This paper will describe drayage operations and the problem to be solved. It will then review the background literature on drayage optimization, the requirements and specifications for the device, the design choices made during development, the software's underlying algorithms and data structures, device testing, possible future data analysis and then present conclusions.

AN ELECTRONIC ON-BOARD RECORDER FOR INTERMODAL DRAYAGE OPERATIONS

A THESIS

Presented to the Department of Computer Engineering and Computer Science

California State University, Long Beach

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Engineering

Committee Members:

Shui Lam, Ph.D.
Min He, Ph.D.
Kristen Monaco, Ph.D.

College Designee:

Burkhard Englert, Ph.D.

By Andrew Browning

B.A., 2000, California State University, Long Beach

May 2012

ACKNOWLEDGEMENTS

To Jennie, Will and Jamie, without whom I would be lost.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABREVIATIONS

AB 2650	Assembly Bill 2650
API	Application Programming Interface
COTS	Consumer Off-The-Shelf
FMCSA	Federal Motor Carrier Safety Administration
GE	Google Earth
GPS	Global Positioning Satellite
KML	Keyhole Markup Language
MRRP-FT	Multi-Resource Routing Problem With Flexible Tasks
PDPTW	Problem Denoted Pickup Times With Time Windows
SDK	Software Development Kit
SGT	Samsung Galaxy Tab
UI	User Interface
UML	Unified Modeling Language
XML	Extensible Markup Language

CHAPTER 1
INTRODUCTION

The idea behind Ubiquitous Computing is that the small things that make up the world around us will eventually become computational. As this happens, it will no longer become necessary to bring something to a computer to process information about it, because the thing itself will be able to process its own information. As all of these little things around us begin to communicate, we will realize the true benefits of computerization; we will have the machines without being conscious of their presence. Recent advances in cell phones and tablet computers, for example, have allowed us to take computers with us everywhere and gather and process financial, social and retail information in real time. This allows us to be conscious of the machines when we need them but not conscious of them when we do not, a far different paradigm from the desktop or laptop computer. Leveraging this technology can allow us to computerize modules of the global supply chain to increase efficiency and decrease cost and pollution. One area in which this is keenly relevant is in trucks. By using a tablet computer, known in the trade as an Electronic On-Board Recorder (EOBR), a truck and its cargo can in effect become a computer that reads its own state, processes data and communicates with other machines in an effort to mine the gathered data for information that can both grease the supply chain and also contribute to sustainable urban development. One of the areas of supply chain management where this can be particularly useful is in drayage, where intermodal freight is transported over short distances prior to being loaded onto trains and long haul trucks for a much longer journey to its destination. This paper seeks to make the case for a drayage-specific EOBR. After providing the reader with the problem description, the current literature on drayage optimization will be reviewed and an EOBR called Drayage Recorder will be proposed to gather drayage-specific information to help uncover areas of inefficiency. Design choices will then be reviewed and the device's data structures and algorithms will be fully explained. Next, test procedures and results will be reviewed followed by data extraction procedures, future analysis and finally conclusions.

EOBR Problem Description

When intermodal containers arrive at the Ports of Long Beach and Los Angeles, they are often required to make an over the road journey to semi-local distribution centers from which they are then dispatched to their final destination. This journey is frequently short, usually within a hundred miles of the twin ports complex, but is responsible for a disproportionately high portion of the total cost to ship the container from its origin to its destination [1]. It is also responsible for higher than normal levels of diesel pollution at the ports and their surrounding corridors because the volume of trucks is so high. Drayage is the process of moving intermodal freight between the ports and these semi-local distribution centers. By gathering data using EOBR technology, drayage operations can be analyzed for inefficiencies that increase cost and lead to higher than normal levels of air pollution.

One of the challenges with EOBRs, however, is that they are not well designed for drayage operations. Current manufacturers design their devices as a digital alternative to the paper based hours of service (HOS) logbook that long-haul drivers are required to keep to prove they are in compliance with Federal Motor Carrier Safety Administration (FMCSA) HOS regulations. HOS regulations require a driver to log when he is on duty but not driving, on duty and driving, in a sleeper birth, and off duty. Depending on a combination of the amount of hours spent in these four states, a driver is deemed to be in or out of compliance with FMCSA safety regulations, which are in place to prevent fatigued drivers from compromising public highway safety. Drivers under this regulation are required to keep and update these logs throughout the entirety of their trip and present them to authorities upon request.

While drayage operators are required to follow FMCSA rules for the amount of hours worked in a day, they are not required to keep a log for that purpose. To wit, current EOBR solutions are not applicable to drayage operators and do not provide a facility to gather the information this study requires to analyze the current state of drayage operations. Moreover, while PierPass and Digital Geographic Research Corporation performed a turn time study at the Ports of Los Angeles and Long Beach in 2011, their study only focused on turn times and GPS logging; the study did not log categories of drayage work and did not seek to collect data outside of the port terminals. As such there is a need for an EOBR that gathers drayage specific data not addressed by the current state of the art or recent studies. That data includes information about the real-time location of vehicles, details about the origin and the destination of the freight, real-time information about congestion and wait times and a driver's holistic interpretation of a given trip. By gathering this data, it will be possible to identify where efficiency may be increased and cost and pollution decreased.

Background Literature

Heavy-duty truck miles in the Los Angeles region have more than doubled since 1982, which is a growth rate greater than population, employment or total vehicle miles traveled; the Port of Long Beach alone generates more than 403,000 heavy-duty truck trips annually. Some consequences of this growth are a higher than normal cancer risk in and around the ports, a large increase in nonrecurrent highway congestion and negative public sentiment toward over the road freight transportation [2].

As a result some planning has been put in place to affect these problems. In the early part of the last decade, drivers at California ports typically waited more than an hour outside port terminals waiting for access and frequently spent more than an hour inside ports waiting for cargo [3]. However, California Assembly Bill 2650 (AB 2650) has had an impact in mitigating this type of congestion at the ports. The bill offered marine terminal operators a choice between increasing their operating hours from 45 hours per week to 70 hours per week (a program known as the PierPass off peak gate program) and implementing appointments for specific cargo pickup and drop-off. The regulation also provided for a $250 fine when trucks were forced to idle for more than 30 minutes outside a marine terminal gate. As a result, the authors found that total turn times (the queue time + the transaction time) was significantly lowered with the median values for a bobtail in/container out being 42 minutes, a container in/bobtail out being 38 minutes and a container in/container out being 61 minutes [2].

While AB 2650 led to a significant decrease in port wait times, it could not provide technology inside marine terminals to track vehicles and determine inefficiencies there. Lam, Park and Pruitt addressed this by capturing the movement of marine terminal traffic using digital cameras. With six digital cameras positioned throughout marine terminal X, they used time-stamped photography to track trucks as they made their way from the queue, through the gate, to the pedestals and then to the exit. While the project was successful in identifying trucks and wait times, the identification of photographs was complicated by human factors and camera limitations, specifically the color of the truck as seen by the camera and the human eye in changing light conditions. Moreover, a truck that may make repeated trips to the port in a single day might introduce errors in identification, linking the entry of one trip to the exit of another. This led to an average truck matching accuracy rate of 90% [4]. While high, the use of global positioning satellite technology can dramatically improve the process of identification by uniquely identifying each truck with near 100% accuracy.

This is important because Monaco and Grobar report that there is pressure to increase throughput in the nation's port systems as US imports increase. Part of that throughput involves drayage-related port terminal traffic. Approximately three-quarters of drivers surveyed in their study reported that they were subjected to strict deadlines for the pickup and delivery of containers and roughly two-fifths reported that there were penalties associated with not meeting those deadlines. Moreover, almost 50% of drivers surveyed indicated that they had recently received chassis that were not roadworthy, which required waiting for a new chassis, waiting for a repair or taking it anyway to avoid financial penalties. If this data can be gathered with a drayage specific EOBR it may be possible to test some of the pickup and delivery algorithms that have recently been applied to drayage optimization but only simulated by computer [5].

For example, Namboothiri and Erera applied a variant of the problem denoted pickup with time windows (PDPTW) algorithm and found that terminal congestion resulted in 29.5% more travel time and 35.4% more vehicles on average, resulting in costs to the drayage operators [3]. Escudero et al. (2010) argued that drayage represents 40% of intermodal transport costs and that by implementing centralized management of drayage operations, there can be a cost savings of 43% - 63% as well as an overall improvement in service. They modeled drayage operations using a Multi-Resource Routing Problem with Flexible Tasks (MRRP-FT) algorithm to schedule resources and an evolutionary algorithm combined with a GPS location system to process the stochastic travel times of each resource with the goal of minimizing operating costs. Their conclusion is that an operating cost reduction of 30% is indeed possible. However, since these pickup and delivery algorithms have only been simulated, it is not possible to know for certain if they accurate or worthy of implementation. A drayage specific EOBR may be helpful in determining the accuracy and effectiveness of these algorithms and whether or not they should be implemented on a larger scale as part of a centralized port dispatch system or other efficiency system [1]. In light of these considerations, Lam and Monaco (2011) proposed to develop a device that could capture drayage specific data for further in depth study of these issues.

This research addresses how such a COTS tablet-based EOBR specifically designed for drayage operations is well suited to gather data to increase port efficiency, decrease port related pollution and further some of the research put forth in the background literature. Testing occurred during the week of February 5th, 2012. A driver working for a drayage operator with a fleet of 30 trucks and 70 chassis gathered the test data during 18 different drayage moves. The driver and the company will be referred to in this paper as Driver X and Company Y respectively and the Requirements and Specifications will outline the process by which the recorder was developed.

CHAPTER 2

REQUIREMENTS AND SPECIFICATIONS

Functional Requirements

Over the last few years the persistent growth in the touch screen phone and tablet computer market has grown so substantially that they are now ubiquitous. Not only are they user-friendly, but they also allow users to customize their experience to a degree that was not previously possible, features that are highly beneficial to the academic researcher. Indeed, phones and tablets have become physical extensions of users themselves, allowing them to organize the world around them in a way that makes sense to them and their immediate community. Most importantly, a great deal of time and energy has been put in to creating powerful standardized user-interfaces because "every time our ability to access information and to communicate it to others is improved, in some sense we have achieved an increase over natural intelligence [6]."

As outlined in the software requirements specification, an EOBR for drayage operations must be a hand-held device with touch screen and be simple and intuitive to use by someone familiar with drayage operations. The user-interface must group related operations together and separate out unrelated operations. The design must provide simple meaningful shortcuts in the operator's language and be consistent with the lexicon of drayage transportation. Finally, the device must be able to sit in the user's periphery until such time as the user needs to focus on it and then be returned to the periphery. The simplicity of the interface will allow the driver to concentrate on driving and relate to the device as just another piece of dashboard hardware. The interface will provide feedback to the user on changes of state, error conditions, (etc). The design will be tolerant and allow the user to undo and redo operations when necessary. Internal and external components will be reused for consistency.

Specifications

At the beginning of each trip the user must create a unique name for each trip to be stored in human readable form. At the same time, the system will generate a unique key-value for each trip allowing the user to then initiate a time stamp and logging process. Once the process is started, the user will indicate the origin type and destination type of the trip to be a rail yard, transload/storage facility, a warehouse/distribution center or other location. The user must also specify whether the trip is a pickup empty, pickup full, deliver empty or deliver full at the time the trip begins. The software will log and timestamp the location of the truck at regular intervals throughout its journey. The driver will interact with the recorder only if relevant details about the trip need to be recorded, such as delays. Upon arriving at the destination, the driver will initiate a timestamp and location log for the end of the trip. The above requirements have greatly informed the design choices made for this device.

CHAPTER 3

DESIGN CHOICES

Lam and Monaco (2011) require a device that allows for the design and layout of a screen for the logging and time stamping of drayage work at the push of a button. They also require the device to be able to log the location of the truck using GPS and to be able to overlay that data on GPS mapping software for route analysis. Finally, the device must have sufficient storage for large datasets that may be the result of long trips.

Apple iPad

I initially reviewed four different tablets across three different platforms. The first tablet I looked at was the Apple iPad. The Apple iPad had the benefit of being the most widely used tablet. Its software development environment is Xcode, which comes as part of the Apple OS X operating system. Xcode allows for every aspect of the iPad development process including providing a software development kit (SDK) for iPhone and iPad applications and device emulators to allow for realistic functional testing of applications before they are installed on the hardware. Also, iPad applications are written in objective C, a high-level object-oriented extension of the C programming language, which is attractive from a development standpoint. The iPad also comes equipped with built-in Assisted GPS (AGPS), Wi-Fi, and 64 gigabytes of flash memory and supports a subscriber identification module (SIM) card tray for external storage. However, there were downsides to the iPad, which made it undesirable, the most prominent being its size.

With a 10-inch screen, the device is too large to mount and does not easily rest within the driver's periphery. It also requires an AT&T service plan, instead of allowing for a choice of providers. Also, Apple heavily regulates the distribution of applications making it difficult to develop for third parties. Finally, at $499.00 for the least expensive model, it was not as competitively priced as other tablets.

<u>Archos 9 PC Windows 7 Starter Tablet</u>

The Archos 9 PC Windows 7 Starter Tablet has all the benefits of a Windows platform device. It can be programmed in C++ or any .Net language and is compatible with the Microsoft Visual Studio development environment. It also comes Wi-Fi enabled, has AGPS and a 60-gigabyte hard drive. However, its screen size (9 inches), its cost ($549.00) and its reputation for being unstable made it a poor choice for this project.

<u>CUWINN 5500</u>

The CUWINN 5500 was the first device identified for this project. It has a 7-inch, waterproof front panel, runs windows CE 6.0, which can be programmed in C++ or any .Net language and is compatible with the Microsoft Visual Studio development environment. Moreover, it is the one device built for the hard knocks of industrial applications. Its weaknesses are that it has a starting price of $429.00, no built in GPS, requiring an extra chipset and extra cost and is 2.2 inches deep making it unwieldy to handle.

<u>Samsung Galaxy Tab</u>

The tablet being used is the Samsung Galaxy Tab (SGT) running the Android Honeycomb Operating System. Unlike some of the other tablets we looked at, the SGT had an attractive price point at just over $200.00 and a choice of service providers to choose from. It is Wi-Fi enabled, comes with a built-in AGPS chipset, has a 7-inch screen, 16 gigabytes of flash memory and supports a SIM card tray for external storage. Finally, it allows us to leverage Google services such as cloud to data messaging and Google Earth. This will allow the truck's location to be logged and remove the complexity of two pieces of hardware by having them integrated into one. The SGT also has a programmable color touch controller allowing for the programming of customized layouts required for this research.

Android

Android is a Linux-based operating system with support for Java libraries and the Dalvik virtual machine. Because it is a popular tablet computer platform and is supported by a robust SDK and APIs that provide access to the hardware resources of Android devices, it is the perfect choice for this project. Moreover, Its low cost, support for rapid development and a large developer community make it a better choice than other platforms, some of which require third party testing before applications can be installed. Because it is a Google operating system, the application being developed for this research will be able to leverage Google Earth (GE) to fulfill the requirement that the GPS positions be easily converted to map locations.

Eclipse

The software development platform is Eclipse, which is the premier open source development platform for Java-based devices. Google has created a SKD specifically for Android devices that includes all the required libraries for writing, testing and debugging Android applications. Samsung has also provided a Galaxy Tab Emulator, allowing for the application to be simulated prior to being installed on the physical tablet.

Google Earth

GE is free mapping software that allows users to overlay Keyhole Markup Language (KML) files on a map of the earth. KML is a variant of Extensible Markup Language (XML) that allows for the markup of geographical locations. KML allows the GE user to view a map of location points and then create a time animation of events. KML files also support altitude data types, photographic data types and custom data types allowing the user to define their mapping experience [8]. Figure 1 is a sample KML file. Placemarks are GPS time stamped locations.

Software Design

Android supports component-based software development. Each Android application is made up of one or more software components that typically run in a single thread. The components used for the Drayage Recorder are the activity and the service.

Android activities are windows presented to the user that hold the user interface (UI). Each time a new activity is launched a new screen is presented to the user. Because memory is often limited on small devices, activities have lifecycle states specifically designed to preserve the instance and the data of the currently running activity while at the same time ensuring resources are available for other programs. When an activity starts, its onCreate() method is called to initialize the activity followed by onStart(), which is called as the window becomes visible to the user.