

**Navigational and Delivery System in the
USPS Conceptual Truck**

by

Boris I. Paskalev

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degrees of
Bachelor of Science in Electrical [Computer] Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

May 21, 2003

Copyright 2003 Boris I. Paskalev. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and
distribute publicly paper and electronic copies of this thesis
and to grant others the right to do so.

Author _____
Department of Electrical Engineering and Computer Science
May 16, 2003

Certified by _____
Ted Selker, Ph.D.
Thesis Supervisor

Accepted by _____
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

**Navigational and Delivery System in the
USPS Conceptual Truck**

by

Boris I. Paskalev

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degrees of
Bachelor of Science in Electrical [Computer] Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

May 21, 2003

Copyright 2003 Boris I. Paskalev. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and
distribute publicly paper and electronic copies of this thesis
and to grant others the right to do so.

ABSTRACT

A custom USPS specific mobile navigational and delivery system is presented. This system aids in the everyday delivery of mail and enhances the services provided by the USPS. The system uses wireless network connectivity to receive real-time delivery specific updates and to transmit package tracking information to a centralized server. Varieties of new services are implemented and many new opportunities are presented. The system provides a more efficient, cost effective, mail delivery process and adds to the quality and range of services provided.

Thesis Supervisor: Ted Selker, Ph.D.

Title: Associate Professor, Context-Aware Computing, MIT Media Lab

Contents

1. Introduction

1.1. Background of the Project

1.1.1. Motivation behind the Navigational and Delivery System

1.2. Potential Users

1.3. Similar Systems and Known Implemented Components

1.3.1. Package Tracking Systems

1.3.2. Navigational Systems

2. Design of the System

2.1. Hardware of the System

2.1.1. The LCD Display

2.1.1.1. Package Tracking and Scanning -- Laser Scanner

2.1.2. Positioning System

2.1.3. Wireless 802.11b Network Connectivity

2.1.4. Onboard Computer and Central Server

2.2. Goals of the Software System

2.2.1. Major Parts and Inter-System Communication

2.2.2. The Server Software

2.2.3. Onboard Software

3. Features of the Software System

3.1. Health of the Truck

3.2. Package Tracking

3.3. Navigational and Delivery Map

3.4. Office Mode

3.5. Sensors

4. Software Modules and Implementation

4.1. Graphic User Interface (GUI)

4.1.1. Sizing of the GUI Frames

4.2. Frames Implementation

4.2.1. Map Frame

4.2.2. Scan Frame

4.2.3. Health Frame

- 4.2.3.1. Actual Health Frame
- 4.2.3.1 Sensors Frame
- 4.2.4. Status Frame
- 4.3. SOAP Communication Interface**
- 4.5. Barcode-scanning Interface**
- 4.6. Map Interface**
- 4.7. Truck Health and Sensors Interface**
- 5. Initialization of the System**
- 6. System Usability and Field Testing**
- 7. Future Work and Extensions**
 - 7.1 Voice Activated Commands and Directions**
 - 7.2. Opportunistic Mobile 802.11b Wireless Network Collection**
 - 7.3. Alternative Positioning Systems**
 - 7.4. System Testing**
 - 7.5. Map Server**
- 8. Conclusion**

Appendix A.

- JFrameGUI.java**
- MappointControl.java**
- IMappointCtrl.java**
- PopUpInput.java**
- MenuHandler.java**
- DeliveryAndPackageStatus.java**

Appendix B. Metrologic MS900 Series Laser Scanner

Appendix C. Digital View LCD Interface Controller

List of Figures

1. Schematic representations of the hardware configuration build in the USPS conceptual truck to accommodate the navigational and delivery system.
 - a. Mocha P4 (E7042B) onboard mobile computer (2003, Cappuccino PC)
 - b. Standard Dual 12V Battery Supply part of the USPS conceptual Truck
 - c. Lucent Technologies, Orinoco Gold Wireless LAN PCMCIA card.
 - d. Metrologic MS951 laser Scanner (Appendix XXX)
 - e. 600W (1500w peak) Power Inverter 12V to 110V
 - f. GARMIN GPS II Plus (2003, Garmin)
 - g. USPS server. Running Apache Tomcat Server and SOAP2.2
 - h. Mounted LCD display
2. Implemented systems with the LCD displays provided by Vert Inc.
 - a. The LCD display implemented and placed on a mounting bracket on the dashboard of the USPS Conceptual Truck.
 - b. Vert's Intelligent Display mounted on a taxi cab, showing a superb quality and brightness of the picture displayed. (Vert Inc. 2003)
3. Printed transistor for RFID on transparent sheet
4. A star network model with seven end nodes (mobile trucks) and a central USPS server
5. The indicator for the "health" of the truck as seen by the driver
6. The navigational and delivery information area for the USPS conceptual truck (the map frame)
7. Diagram of the main software modules
8. Difference between decorate and non-decorated Java frames
9. Layout of the static GUI frames and one pop-up frame
10. A Scan Frame that appears as a pop-up window to scan the barcode of a package that is being delivered
11. Three basic states of the health frame
 - a. Smiling face of the USPS truck and status bar, indicating good health
 - b. Neutral face of the USPS truck and status bar, indicating mediocre health
 - c. Sad face of the USPS truck and status bar, indicating poor health

1. Introduction

In the spirit of the Context-Aware Computing Group at the MIT Media Laboratory, we have decided to change people's perception of the mail delivery system and the way it fits into our society. In simple words, we are designing the delivery vehicle of the future. Many innovative ideas have been proposed and considered as additions to or upgrades of the United States Postal Service (USPS) Conceptual Truck but only a dozen features have been selected, because they radically change or enhance the truck at minimal cost. The three aspects in which the truck will be enhanced are efficiency, safety and capacity to perform new services. The USPS truck is a vital part of the society. This fact led to the idea that the truck should be able to communicate to the community and also help the society. A major aim for improving the truck is increasing the driving and reliability security of the truck. Another direction for improvements, which this thesis concentrates on, is enhancing the service provided. Enhancing the service is an approach that would allow information about tasks that a postal worker might perform while delivering the mail to be available to him within minutes of his destination. Another sub goal is to ensure that the postal worker is oriented geographically at all times and able to deliver orders from any warehouse. This approach expects that all delivery trucks would have small amounts of items that have not yet been ordered but are very likely to be ordered from contracted warehouses. These items would be loaded on the truck in advance. The selection of these items would be based on the information for most demanded items from each counteracted retailer. In addition, the postal worker will be able to communicate with his supervisors and coordinators from the truck as well as to work on his administrative tasks there. The ultimate goal is the postal truck to be the mobile office for the driver.

1.1. Background of the Project

Working under the supervision of Professor Ted Selker at the Media Lab's Context-Aware Computing Group, I designed and built a computer system for an interactive conceptual USPS truck. The truck has enhanced USPS-specific navigational

features and operates as a marketplace on wheels. A Global Positioning System (GPS) is connected to a real-time interactive map, displaying the current location of the truck. The map also displays real-time delivery orders, traffic conditions, route-specific updates and information about various community services that USPS will be involved in. Some of the societal services that the USPS Conceptual Truck could be involved in are hospice check-in, parole signing-in, medical alerts, emergency help, determining accident locations and providing real-time video feeds about traffic and weather status. Using the real-time GPS navigation and real-time updates, the truck can find the best routes out of traffic jams, accidents and construction sites to lead the driver more rapidly and safely to his destinations.

Many navigational systems use GPS. However, we are designing a more interactive, USPS-specific, bi-directional real-time system. The system will not only provide navigational information about driving but will also aid both in standard mail delivery and in introducing novel real-time delivery systems like the same-day delivery option. Preliminary research shows this delivery feature to be very attractive for the end users, the retailers and the USPS. This thesis uses the research and implementation already done on GPS-aided navigational systems like Microsoft MapPoint 2002 and the advanced “GARMIN” GPS units to create a unique, profound and useful mobile office system.

1.1.1. Motivation behind the Navigational and Delivery System

Some of the main reasons for creating this mobile office system are to increase the productivity of the postal workers, to provide even more reliable tracking of packages and to minimize the possibilities of lost or misplaced packages. The office mode of the system will provide capabilities for the postal worker to perform all of his paper work and communicate with his managers or other office workers from the truck. This communication will decrease the time spent in the postal office and will increase the USPS presence in local communities. The actual office space of each worker will be moved to his truck, thus provide for more sentimental attachment to the truck. This effect should lead to greater caution in the use of the postal equipment and increase its lifetime

while decreasing its maintenance cost.

Higher efficiency in the mail delivery will also be provided by the customized navigational system, currently based on Microsoft's MapPoint 2002 (MP2002) software. This navigational system will receive real-time updates from a dedicated USPS server about traffic conditions, constructions and specific postal demands. The opportunity for same-day delivery is based on the fact that on average, each truck is only 30% full of mail. To utilize the rest of the space, popular selling items from contracted online retailers will be pre-loaded on the trucks. Once a retailer receives a same-day delivery order for an item that is available for same-day delivery, the central USPS server sends a message to the closest USPS truck that holds this item and the truck delivers it within minutes. This service will provide extra revenue for the USPS and the online retailer, as well as unmatched convenience for the consumer.

Package tracking capabilities will be based on either Radio Frequency Identification (RFID) (AIM, Inc, July 2002.) or on the current barcode scanning system. The main feature will be that all packages and pre-loaded inventory will be scanned while the truck is being loaded and this information will be stored on the USPS server database. The postal packages will be available for instantaneous retrieval of package tracking information and the pre-loaded inventory will be available for scheduling and navigating the same-day delivery orders as well as package tracking. Upon each scan, the database will be updated so that up-to-date information will be available for all packages at all times. This feature will allow USPS to satisfy the demand for real-time package tracking and will be able provide additional guarantees for the service provided. In particular, the RFIDs will be a novel way to provide up-to-the-minute package tracking information and approximate time of delivery based on the route of the particular truck delivering the item and the RFID's readings. The RFIDs also provide effortless scanning of all packages at the same time and could be used to decrease the chances of a package being lost, forgotten or misplaced. When the package leaves the truck for its final delivery, its destination address will be matched to the current location of the truck. This tracking method will provide unique and unmatched reliability and minimize the possibility of human error in the delivery process.

1.2. Potential Users

The potential users of this system range from any courier, postal service, delivery establishment or any organization that operates a fleet of vehicles in the world. This system is sponsored by USPS and is specifically designed to meet particular needs and challenges that the USPS fleet is facing. Considering the integration of USPS specific needs in the early design of the system, the first and most prominent user of this system is the USPS mobile fleet of delivery vehicles. The system is designed and implemented in a modular fashion to provide for high quality and ease of maintenance, extensions and modifications. This general design of the system allows various modification of the system to ensure wider range of possible users and integration opportunities. For example, the current version of the systems is based on barcode-scanning for package tracking. However, a simple implementation of a new module, which will scan RFIDs, could replace the old module and provide a major hardware upgrade of the system.

1.3. Similar Systems and Known Implemented Components

1.3.1. Package Tracking Systems

No single system that encapsulates all the features and capabilities in the system presented in this thesis is commercially available. What we can talk about is various partial solutions to what we can call the ultimate delivery and navigational system embedded in a mobile office. The idea of this multi-functional system is to navigate, assist and monitor the whole process and communication of the postal truck. The system will also provide extra security and assistance to the postal worker in his manual tasks: by double-checking the drop-off location for packages and even by reminding the driver of packages he failed to deliver. This comprehensive coverage of the mail delivery process, that the system provides, makes it so unique and novel. Partial implementations or systems with smaller spans have been commercially available and used in various businesses, mainly in the leading expeditors like USPS, Federal Express (FedEx) and United Parcel Services (UPS). Both FedEx and UPS provide real-time package tracking,

while USPS provides package tracking updates every twenty-four hours.

FedEx's newest data collection device for couriers incorporates a micro radio for hands-free communication with a printer and mobile computer in the courier's delivery vehicle. Called the PowerPad, the devices use Bluetooth wireless technology that allows them to communicate with each other within 30 feet. (FedEx, 2002)

The quote shows that FedEx is using a custom device to scan packages and then to transmit that information to a computer in the vehicle. Information about communication between the vehicle and the main database is not publicly disclosed. The most probable communication solution is to use FedEx's offices for wireless download, once the trucks reload and/or pass by FedEx offices or a nationwide wireless network solution with the same capabilities as the UPS network described later in this paragraph. Similarly to FedEx, UPS has proprietary device called "Delivery Information Acquisition Device or DIAD [DIAD allows] ...drivers to collect and send delivery information at virtually the same time" (UPS, 1999). What is known about UPS's DIAD connectivity is that UPS have started a "Package Level Detail (PLD) initiative that captures information about each package and moves the information through UPS's operational network" (UPS, 2002). This private network also does not have a publicly disclosed specification but it is known that this is a wireless network using satellites for virtually continuous coverage, similar to the cell phones' coverage. Evidently, this network provides the ability to transfer information about scanned packages to a central database that keeps track of all packages. Both FedEx and UPS have their own systems which scan and transmit the information for tracking packages. Both systems have close to real-time performance but they represent only a single aspect of the functionality of the USPS navigational and delivery system presented in this thesis. The package tracking part of this new system will replace the current USPS system which uses barcode-scanning wands with internal memory. These wands are downloaded via short-range radio communication at the end of the day when the truck is docked at the post office. This is why the data about all packages is updated at the end of the day, when all downloads are completed. The new system will allow the use of these wands or any other scanning device or schema; the information will be "immediately" downloaded from the wand and sent to the centralized

USPS server for real-time package tracking. The actual time to transmit the information from the mobile truck to the USPS server will depend on the wireless coverage. The specification of the system approximates that within a five-minute window will be within wireless coverage and sent all updates to the central server. Further explanations of this process are presented in the **System Design** chapter, section 2.1.4. *Wireless 802.11b Network Connectivity*

1.3.2. Navigational Systems

Navigational systems are standard features in most new cars. These systems are GPS guided navigational systems that use only static pre-deployed maps and GPS readings. Such a system is one-directional and receives information only from GPS satellites. It determines the position of the vehicle and calculates driving direction from the current position to designated destinations. Map databases and software products for driving directions could be updated with information about constructions and road updates. Although these mapping products are rapidly developing, they often lack new and small streets and reconstructed infrastructures. Considering the lack of details in the mapping programs each navigational and delivery USPS software will add an additional layer for updates and custom services, while displaying the current location and route for the truck. For this project, the MP2002 package was chosen because of its ActiveX components capabilities, which allow users to customize and call components of this package using any programming language. Further discussion of this choice can be found later in the **Software Design** chapter. Each truck will be able to collect information about traffic situations and construction sites. This information will be collected and processed by the USPS server to update a custom real-time road and traffic map. This map will allow each truck retrieve updated information about its surrounding region and then recalculate its optimal path.. The availability of such a realistic map of the routes and traffic would provide competitive advantage and would open possibilities for new markets and services for USPS.

2. Design of the System

The design of our systems included many stages of planning and considering various possible solutions. The first issues were purely hardware problems, such as what kind of cables should be used and how they should be installed in the truck. Choosing the hardware was a long process. For example, the navigational display, that would show the navigational and delivery status of the truck, could have been an out-of-the-box touch screen display or just a regular LCD display. After numerous discussions and trials, an already tested but not commercially available and packaged display was chosen for its brightness and durability. For the main onboard computer we chose the Mocha P4 (E7042B) patented design of Cappuccino PC (Cappuccino PC, 2003). This particular choice was based on the size and weight constraints for the equipment used in the USPS conceptual truck. With a set of hardware in place, the software design and development started. This process required even longer and harder considerations than the process of selecting and installing all of the hardware for the system. Without further discussing the design choices here, the following paragraphs will go over the details of the hardware and software design and implementation.

2.1. Hardware of the System

The hardware of the navigational and delivery system of the USPS conceptual truck constitutes of a bright 17-inch LCD display flexibly mounted on the dashboard of the conceptual truck. The Mocha P4 computer that runs the navigational and delivery system of the USPS truck communicates with the central USPS server over local 802.11b networks. The hardware installation of the LCD display was performed directly on the actual conceptual truck provided by USPS (the final output could be seen on Figure 1a). The rest of the hardware, including the onboard computer, is portable and easy to deploy and remove from the truck. A GARMIN GPS II Plus unit (Garmin, 2003) is connected to the onboard computer through its serial port for global positioning data readings. The

main input device for package delivery and pick-up is a Metrologic MS951 laser barcode scanner (Appendix B). Figure 1 shows the schematics of the implemented hardware configuration of the system.

Figure 1. Schematic representations of the hardware configuration build in the USPS conceptual truck to accommodate the navigational and delivery system.

- a) Mocha P4 (E7042B) onboard mobile computer (Cappuccino PC, 2003)
- b) Standard Dual 12V Battery Supply part of the USPS conceptual truck
- c) Lucent Technologies, Orinoco Gold Wireless LAN PCMCIA card.
- d) Metrologic MS951 laser Scanner (Appendix B)
- e) 600W (1500w peak) Power Inverter 12V to 110V
- f) GARMIN GPS II Plus (Garmin, 2003)
- g) USPS server. Running Apache Tomcat Server and SOAP2.2
- h) Mounted LCD display

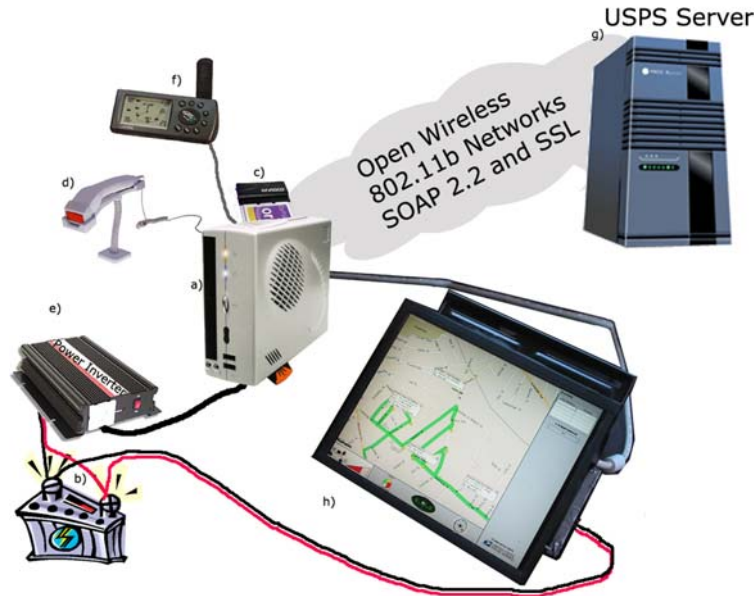


Figure 1 shows the main hardware configuration used for the USPS navigational and delivery conceptual system. The Mocha computer is the central computational unit onboard the USPS truck. The main input device is a laser scanner for reading the barcodes of delivered and picked-up packages. The scanner is connected with the onboard computer through a standard PS1 cable, as an extension of the PS1-keyboard. The GPS unit is connected to the computer through a serial port and is mainly powered by a set of four, size AA, batteries (Garmin, 2003.) In the serial implementation of this system an inexpensive, commercially available, GPS will be integrated in the truck. This

integrated GPS would be identical to the one used in the serial automotive production lines. The 802.11b wireless card is plugged into the PCMCIA slot of the Mocha computer. To drive the LCD we use the internal Mocha video card. The power inverter provides 110 Volts for the computer operation and it is directly connected to the truck batteries. High voltage supply for the backlight of the LCD comes from a dedicated power inverter directly connected to the truck's batteries as well.

2.1.1. The LCD Display

The most visually aesthetic point of this system's design is the implementation of the LCD screen and subsequently the location of the onboard computer. By walking into the truck the first feature of this project that an observer will note is the location and the mounting mechanism of the LCD. In short, the realization of the display will provide the first impression about this project. This is why the display was flexibly mounted on the dash-board with a mounting bracket that supports the LCD from the bottom. The mounting bracket allows the screen to be adjusted in two horizontal directions, so that the visibility could be improved and also to allow easy retraction when the display is not in use. The mounting bracket is a modified TV stand and supports the display only at its four sides. This mounting provides the largest possible open surface on the back of the LCD for cooling. A dedicated cooling system might be needed for extended hours of operation in extremely warm conditions. Such a cooling system was tested but proved to be unnecessary in the conceptual vehicle. The LCD screen has an extra bright display body, provided by Vert Inc., specifically designed to be easily viewable under extremely bright conditions, like direct sun light. This same display is the building block used in the Vert's Intelligent Display (VID) technologies that are fitted on the top of taxi cabs (Vert Inc. 2003) (Figure 2b.) To drive the bare-bone packaged active-matrix LCD display, a Digital View Ltd. AC-1024V3 LCD interface controller (Appendix C) was used. This controller receives the analog VGA input from the video card of the Mocha P4 computer and drives the multi-sync active matrix interface. The building of the working LCD display required a power inverter supplying 2005Vrms for the backlight. The power inverter is directly connected to the 12V DC array of batteries in the truck. The power inverter outputs close to 4 kV and is the source for the extreme brightness of the picture

of this display. Figure 1a shows the built system for the LCD display in the USPS conceptual truck and Figure 1b shows how Vert Inc. builds their VID systems and mounts them on taxi cabs.

Figure 2. Implemented systems with the LCD displays provided by Vert Inc.

a) The LCD display implemented and placed on a mounting bracket on the dashboard of the USPS Conceptual Truck.

b) Vert's Intelligent Display mounted on a taxi cab, showing a superb quality and brightness of the picture displayed. (Vert Inc. 2003)



b)



The mounting bracket that supports the LCD screen allows the power inverter and the small Mocha computer to be attached behind it. The small Mocha computer could also be placed under the dashboard to decrease the weight on the mounting bracket and allow for better cooling of the LCD display and the onboard computer.

2.1.2. Package Tracking and Scanning -- Laser Scanner

The choice of laser scanner was a long discussed issue as it would be the instrument that all postal workers will use on daily. Each package is being scanned multiple times on the way to its destination. This scanning allows for accounting and tracking of all packages. Considering the enormous amount of scans that would be performed on a daily basis, Professor Ted Selker from the MIT Media Lab, supervisor and leader of this project, suggested the usage of automated scanning technology. The design of this technology is based on RFID tagging of all packages. This means that each package will be labeled with a sticky RFID tag rather than a barcode. The RFID technology have evolved over the years and now an RFID tag could be printed on any material, including paper, using organic or non-organic materials for the active diodes. The organic materials can lower the cost per RFID in mass production to a fraction of a cent. Figure 3 shows one such RFID produced at Infineon Technologies AG (RFID Journal, 2003.)

Figure 3. Printed transistor for RFID on transparent sheet



During a presentation at MIT, representative from Infineon Technologies AG demonstrated organic RFID tags printed on paper, aluminum and plastic foil (RFID Journal, 2003.) Clearly, there is a huge benefit of using the RFID tagging, allowing wireless scanning of all packages at the same time. Implementation of this system would decrease the man-hours needed for scanning and also increase the reliability of package tracking and delivery. During loading of the postal vehicles, all packages will be scanned-in in the truck and then upon delivery will be checked-out as leaving the truck. The system will be able to determine at all times the location of a package and whether it

is being delivered to the designated destination. Imagine a package is being stolen or just it falls out of a truck this will lead to instantaneous detection and increased possibility of recovery. The detection (scanning) of RFID is done by a device called reader that beams a short range radio signal. This radio signal induces electricity in the RFID and transmits some information back to the reader. The information transmitted could be identical to the numeric barcodes currently used. This technology is extremely promising and this is why the design of our system is modular to allow for any kind of a scanning device to be ported to the system. USPS currently operates a proprietary system, developed by Motorola, that uses specialized scanning units. These units are the most economically feasible scanning device for the production deployment of the USPS navigational and delivery system. For the purpose of testing and demonstrating the system, a Metrologic MS951 laser Scanner (see Appendix B) is installed and used for data acquisition.

2.1.3. Positioning System

To provide realistic navigation at all times, our system requires a global positioning technology that provides information about the coordinates of the vehicle. The current implementation of the system includes a GARMIN GPS II Plus (Garmin, 2003) (Figure 1f.) This GPS provides location information to the Microsoft MapPoint 2002 software and also to the USPS records that keep package deliveries and truck locations data. This location data is used in three main features of the system -- optimizing the routes of the trucks, locating trucks for same-day delivery orders by the central USPS server and double-checking the accurate package deliveries. The optimization of the truck route is based on the current location and the delivery objectives of the truck; in addition, the route optimization is based on the updated map information received from the USPS server. The information about the position of the truck, together with each truck's unique ID, will be constantly transmitted to the central USPS server, thus the server will have a realistic map of the location of all delivery vehicles. Based on this location map and the known pre-loaded inventory on each truck, the USPS server will be able to schedule the same-day delivery orders among the trucks and alert each driver when new orders would arrive. The known locations of all trucks will also allow

the USPS to help in emergencies and other community services. The third feature that uses the location information received from the GPS unit on board of each truck is to double-check whether each delivered package is delivered to its designated destination. This check is based on the bi-directional database of valid addresses and geographical points that USPS maintains and uses in the daily mail delivery. Similar, but not as detailed and up-to-date database, is available in the MP2002 software.

Although the current positioning system is based on a GPS device, there are other alternatives to determine the location of each truck at all times. One of them is a schema based on the wireless 802.11b routers that the truck uses to connect to the Internet. Based on Media Access Control (MAC) addresses of these routers, the current location of the trucks could determine. This project, by itself, is a profound one and on-going research will determine how successful it will be in the future. Another possible solution is to use the knowledge of each truck's driver. This schema will assume that the driver knows where he is and will deliver most packages to their right location. Then, every time he scans a package for delivery the system will know where the truck is based on the delivery address of this package. This approach is not perfect but provides frequent enough updates for scheduling the same-day delivery options. In addition, this approach does not require additional hardware or actions taken by the driver to determine the location of the truck. Without further discussing possible positioning system in this thesis, I will just mention that a simple inexpensive commercial GPS receiver will be enough for the mass deployment of this system.

2.1.4. Wireless 802.11b Network Connectivity

The wireless Internet network is the main communication line between the conceptual truck and the central USPS server. In the current design the Internet connectivity, will be through the local 802.11b networks that are already installed and in use in most administrative and private buildings. The 802.11b wireless networks usually have a range of 250 meters. This range is sufficient to cover all streets around buildings, having wireless access, where the delivery vehicle will be passing through. A research vehicle was driven around Boston and Cambridge, Massachusetts to randomly monitor

the wireless coverage, and showed that the open 802.11b wireless coverage is rather thick. More specifically, while driving one can find a local 802.11b network on at least every block. It is important to note that the design will not require a continuous coverage but will actively detect open local networks and connect to the USPS server to receive updates. Open networks are private or business owned wireless networks that can be used by anyone who have a computer and a wireless card, capable of sending and receiving 802.11b packages, within the range of the wireless transmitter. Such transmitters are often used in private homes and most businesses to provide wireless connectivity in and around buildings. This paper will not further deal with the availability of the 802.11b networks or other specific network topologies that are suitable for the USPS system, as the main concentration of this thesis is the implementation of a working navigational and delivery mobile office system. This system requires the use of some kind of wireless network connectivity. The 802.11b network is just one of the most prominent and popular possible implementations for the wireless network used in the USPS system. In addition, the USPS trucks can use the new national high-speed wireless Internet network, announced for deployment by Cometa Networks, a new cooperation company of AT&T, IBM and Intel (The New York Times, December 2002.) According to the company, there will be "secure Internet access within either a five-minute walk in downtown areas or a five-minute drive in outside areas" (Cometa Networks, January 2003.) This solution will allow the USPS trucks to be able to connect often to the main server, and update their navigational and delivery system as well as to provide real-time package tracking information. For performing office tasks, like video-conferencing or any task that requires streaming data transfer, we assume that the truck will be stationary at a point where reliable network coverage is available. This requirement is also a safety requirement concerning the attention and concentration of the postal worker while driving.

In the current implementation of the system, the onboard computer (Figure 1a) connects to the open wireless networks using a wireless PCMCIA card using the "Wi-Fi" standard. The "Wi-Fi Alliance is ... [an] international association to certify interoperability of wireless Local Area Network products based on IEEE 802.11 specification" (Wi-Fi Alliance, 2003.) These international specifications provide for the

ease of access to open wireless 802.11b networks. The particular PCMCIA wireless card used in this system is shown in Figure 1c.)

2.1.5. Onboard Computer and Central Server

For the testing purposes of this system, two computers are needed: an onboard computer for the conceptual truck and a testing USPS server. The onboard computer (Figure 1a) will not perform computationally intensive calculations but will mainly display information received from the central USPS server. This specification will only require a small and not very expensive computer to be installed in each truck. The cost of the computer and all needed peripherals including the laser scanner, wireless card and display, would cost no more than one thousand dollars, which is a good price in comparison the UPS DIAD, which costs in the range of one thousand and five hundred dollars. The price for the DIAD device is obtained from a UPS delivery person and is not officially confirmed.

2.2. Goals of the Software System

The software for the navigational and delivery USPS system is a flexible and extendible product for facilitating various services. This modularity of the system is the key component allowing for advancing the technology used and upgrading both the software and the hardware of the system. The development is done mainly in Java 1.4 as it provides good grounds for web-services development and reliable security packages for authentication and authorization of two or more parties communication over the Internet. Each truck will be running the customized client software (the mobile navigational and delivery office software), which will be initialized with the truck's unique ID. The unique ID is used in the authentication of the trucks and also to provide each truck with the ability to receive personalized information for their route, status and any real-time orders and information. The goal of the system is to aid the driver, with minimum interference from his side, in the everyday delivery process.

2.2.1. Major Parts and Inter-System Communication

The two major parts of the navigational and delivery system – the server and the onboard computer, communicate in a star type of a network, sketched on Figure 4.

Figure 4. A star network model with seven end nodes (mobile trucks) and a central USPS server



Figure 4 shows the star network consisting of a central node and seven end nodes. The number of nodes is a variable and does not have a set maximum. All communication links (the lines in the figure) are based on the open wireless network as already discussed. The center of the star-network is the USPS server and each node of the network is a delivery truck running the mobile navigational and delivery system. The server could handle the communication between the trucks, if such communication is needed. There is no need to add additional communication links and computational tasks on the mobile nodes (truck) for end node to end node direct communication. Such an additional burden on the mobile nodes will increase the computational needs, which will increase the minimal requirements for the onboard computer. The star network creates a single point of failure possibility at the central server and it could also be the bottleneck of the system. To overcome these limitations, the requirements for the USPS server should be extremely rigid. The server should be replicated and backed up at all times to guarantee reliability and availability at all times.

2.2.2. The Server Software

This thesis does not go into the details of building the actual central server for the USPS navigational and delivery system but highlights the technology and a test server for system. The actual server and choice of technology will be based on USPS specific cost and benefit analysis. The USPS server side shown on Figure 1g) and Figure 2, will run a standard production web server like iPlanet. A web server is a piece of software, running on a dedicated machine, called server, that is constantly connected to a network, in our case to the Internet. This server has a particular network address and other applications or computers, connected to this same network, can communicate with it based on its address. The USPS server will be able to securely authenticate and communicate with all mobile trucks and perform various updates and requests. This same server will receive all updates about package deliveries, truck locations and miscellaneous information from all mobile nodes. For the testing purposes of the system, the server uses the free Apache Tomcat server, since it is fast, extremely flexible and provides the same functionality as the commercial iPlanet server, at no cost. The free Apache Tomcat server can be used as a production server as well, in the case that it is finely tuned on a regular basis. The actual communication between the server and the mobile nodes will be encrypted using Secure Sockets Layer (SSL) connection. The protocol to transfer the data between the central server and the delivery trucks is the Simple Object Access Protocol (SOAP). SOAP accommodates for the possibility of various wireless networks usage. "SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment." (Don Box et al., May 2000) This is a very general protocol that allows the creation of web-services and communication between a running server and any number of mobile or static users. The protocol provides a wide variety of extensions, so that any kind of data packages could be transmitted between the server and the USPS trucks. This allows the system to be extended on demand. For example, one of the future ideas is to transfer still images, taken from the six cameras installed on the truck, back to the server in case of accidents or emergencies. Those images could be used by the authorities to coordinate actions and provide specific help when and where needed.

The Java programming language has build in libraries for development, based on SOAP calls. Both the creation of the SOAP based web-services and the SOAP based calls to these web-services, is embedded in Java and this is one of the reasons Java is the language of choice for this system.

2.2.3. Onboard Software

The mobile side of this system is a USPS-specific customized software product build with Java 1.4. The Operating System (OS) independent nature of Java, allows the USPS trucks to exchange or even run different types of systems simultaneously. The only requirement for the onboard computers is capability to run the Java Virtual Machine (JVM) to compile the software. The actual development was done using the *Sun ONE Studio 4 update 1, Community Edition* environment running the Java 1.4.1_02 version, which is the latest Java 1.4 release. Other applications that were used and directly embodied in the development of the software were Microsoft MapPoint 2002 and J-Integra® 1.5.5 (Intrinsyc Software, 2003.) These applications will be discussed later in this work in more details together with the modules they supplement. The onboard system is primarily designed to navigate and track the mail delivery process in each truck. The main features are real-time tracking of all packages and aiding in their prompt and reliable delivery. Additional societal function of USPS can be added to the system with little or no changes in the software at all. These additional features will mainly emphasize on alerting the driver of the truck for a particular societal task he would need to perform on a particular location. All alerts and information that the driver receives are shaped and scheduled on the USPS server side and then transmitted to the particular truck and displayed on his onboard computer.

3. Features of the Software System

The navigational and delivery system for the USPS conceptual truck has five

main features, which are picked based on their significance and impact in the postal delivery process. Those features are: health of the truck, package tracking, navigational map, office mode and sensors. Each of these features is targeting one or more of the goals of the USPS conceptual truck, mainly -- efficiency, safety and services. The health of the truck feature is mainly aiming at the safety and efficiency issues as it monitors all hardware and software systems in the truck and provides a general view of how "healthy" the truck is. Once the overall status or "health" of the truck drops below certain point, a mandatory maintenance service would be required for this particular truck. The sensors of the truck are the main data source for computing the overall "health" of the truck. These sensors also contribute to the general safety features of the truck. The most critical sensor-readings are displayed at all times in a dedicated frame on the screen for the driver to observe. Package tracking is one of the most important features that impact the efficiencies and the service provided by USPS. Package tracking provides real-time information on the location and status of all packages in the system. This feature is a high consumer priority feature and thus receives a lot of attention in the conceptual delivery system. The navigational map enhances the efficiency of the mail delivery and the possibilities for new services. In addition, the navigational map can alert for possible accidents or extreme road conditions and thus ensures the safety of the postal driver. The following paragraphs will further explain the details and benefits of the main features mentioned here.

3.1. Health of the Truck

The "health" of the truck is an idea that ensures that the truck is in acceptable operating condition at all times. The onboard software system receives readings from all sensors in the truck and then, based on a pre-determined function of these readings, determines the overall "health" of the truck. The software also checks the readings received against a compliance table with acceptable safe operational values. If any reading is determined to be out of the safe operational range, an alert is issued to both the driver and the central USPS server. Once an alert is received, the truck might be

automatically removed from the operational fleet by not allowing it to start the engine again until a maintenance crew removes the alert. This blockage of the engine or automated activation of the emergency brake could be controlled by the central USPS server or the driver himself, once the engine is not running or once a manual control is activated. The specific logistic of such procedures will be cleared in the future but the main outcome is that the central USPS server and the driver are both alerted of any occurring problems to allow for their prompt resolution. It is obvious that this feature is highly effective in increasing the safety of the truck and the efficiency in recovering from possible brake downs of trucks. Figure 5 shows what the indicator for the "health" of the truck looks like. The indicator has an image of an USPS delivery truck that has smiling, sad or neutral face and a scale, indicating the "health" of truck.

Figure 5. The indicator for the "health" of the truck as seen by the driver



The picture above is the exact copy of what the driver concerning the health of the truck. This particular view is showing the truck in mediocre "health". Based on this information the driver can obtain more details about the sensors' readings and what are the inputs for the current health of the truck. This same information could also be used for maintenance of the truck and for fixing particular problems. The detailed information about the "health" of the truck is only accessible from the truck when it is not running or is in office mode. This requirement is necessary to assure that the driver cannot be distracted while driving. The central USPS server can also read all sensors at any time upon request.

3.2. Package Tracking

Real-time package tracking is one of the main requirements for this system. As already discussed, the real-time package tracking will match USPS's customer needs and

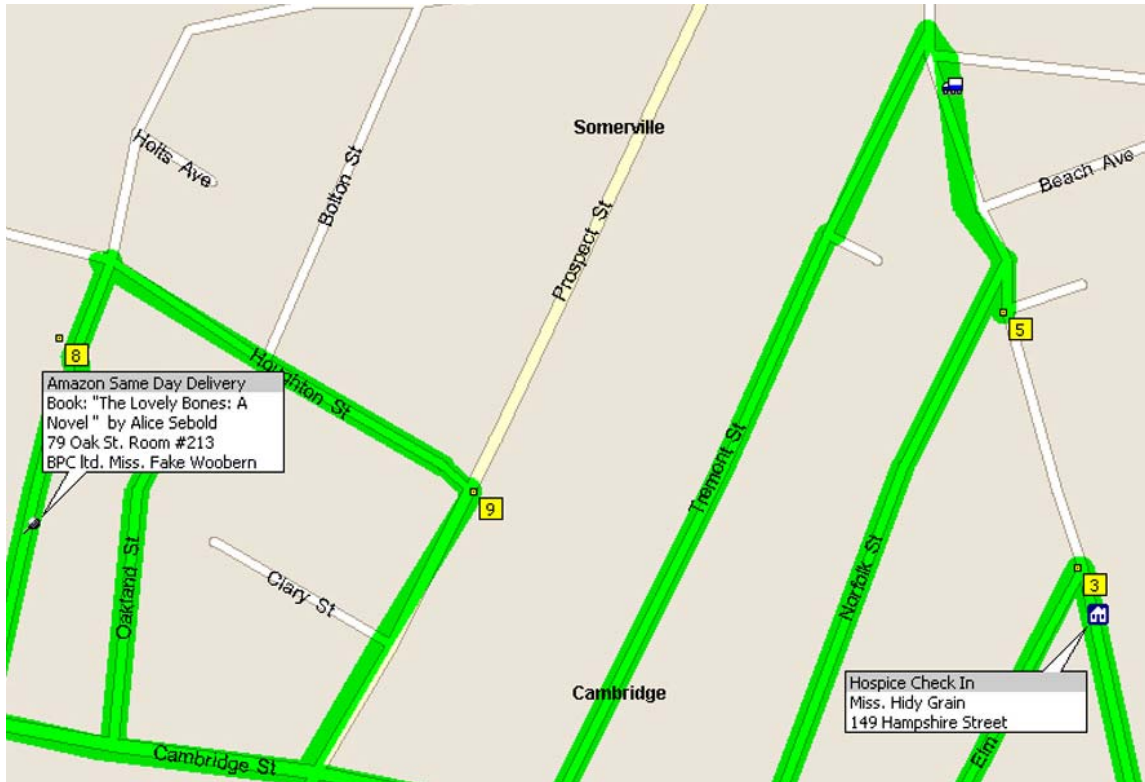
the service provided by other package delivery companies. The real-time package tracking provides extra security for the customers. The availability of the information of where a package is and approximately when it is going to arrive at its final destination allows customers to double-check the progress of the delivered item and to make sure there is no confusion to where this package is going. In the process of package tracking, each package gets a unique identification which is stored in central database together with specific delivery information. As discussed earlier in this paper, the identification of the package could be barcode, RFID or some other media. As the package is being handled during the delivery process, the ID of the package is scanned and the delivery status of the package is updated in the database together with the location and time of the scan details. Interface to this database is provided to all customers to check the status on their packages as often as they like. As USPS already has a reliable and accessible database for package tracking, this system will only provide the possibility to update the information in this database in real-time rather than updating it daily. Recording the barcodes of all packages in a truck, as they are being processed, allows their tracking information to be updated in real-time on the USPS server. As discussed in section 2.1.2. *Package Tracking and Scanning -- Laser Scanner*, the method for scanning the packages could vary. Independent of the type of scanning device, the information about all packages is recorded and transmitted to ensure real-time server updates.

3.3. Navigational and Delivery Map

The navigational map is the main feature displayed to the driver. The map shows the current position and direction of the truck as well as detailed information about the surrounding road-map and conditions. The map also displays the route that the truck should follow. This suggestive route is dynamically updated, based on road-condition updates, the location of the truck and any new tasks added to the delivery route of the truck. All information for updating the map is received from the central server with exception for the current position of the truck, which is based on the onboard positioning system as described in chapter 2.1.3. *Positioning System*. The Microsoft MapPoint 2002

software performs the optimal route calculations as well as the graphical representation of the navigational map. The positioning system readings are directly inputted into the map as parameters. All the calls to the map software are external and coordinated by the navigational and delivery software, as well as the USPS central server. All of the updates to the map are stored at the onboard computer and the USPS server to create a customized realistic map of the road conditions at all times. This map could be used for additional USPS services like traffic watch systems in rural areas. To provide efficient mail delivery, the map lists the next drop-off or pick-up locations for the carrier. Next to the scheduled stops, the driver can see a description of the items he is supposed to deliver or pick up. The map will also alert the driver if he leaves a location without delivering all items designated to this location. This feature ensures that all items for a particular address are delivered while the truck is there. Figure 6 shows a snap-shot of the navigational area of the running delivery and navigational system deployed in the USPS conceptual truck.

Figure 6. The navigational and delivery information area for the USPS conceptual truck (the map frame)



The picture above shows only the actual navigational map area of the system. The location of the truck is shown as a small truck on the map. The thick green line is the currently calculated optimal path. The numbers on the green route are pre-determined pick-up locations that are part of the every day route of this truck. The picture also shows extra locations for community services to be performed by the driver of the truck.

3.4. Office Mode

The office mode has nothing to do with the delivery and is just an extension of the system that is not presented in this thesis in details as we concentrate on the delivery and navigational system. In brief, the office mode allows the driver to communicate with his managers and to complete paperwork. The actual specifics of the tasks that are to be performed during office mode will be determined upon the complete implementation of the system as USPS have not provided any particular requirements. The only interesting

feature of the office mode is that it will be triggered only when the truck is stationary. It will require wireless network coverage for video conferencing and submission of data. The current implementation of a sample office mode has the Adobe Acrobat tools and the Microsoft Office tools for word processing solution for completing postal paperwork from the mobile office. These commercially available packages are accessed in the same way the Microsoft Map Point 2002 software is accessed from the mobile navigational and delivery USPS system.

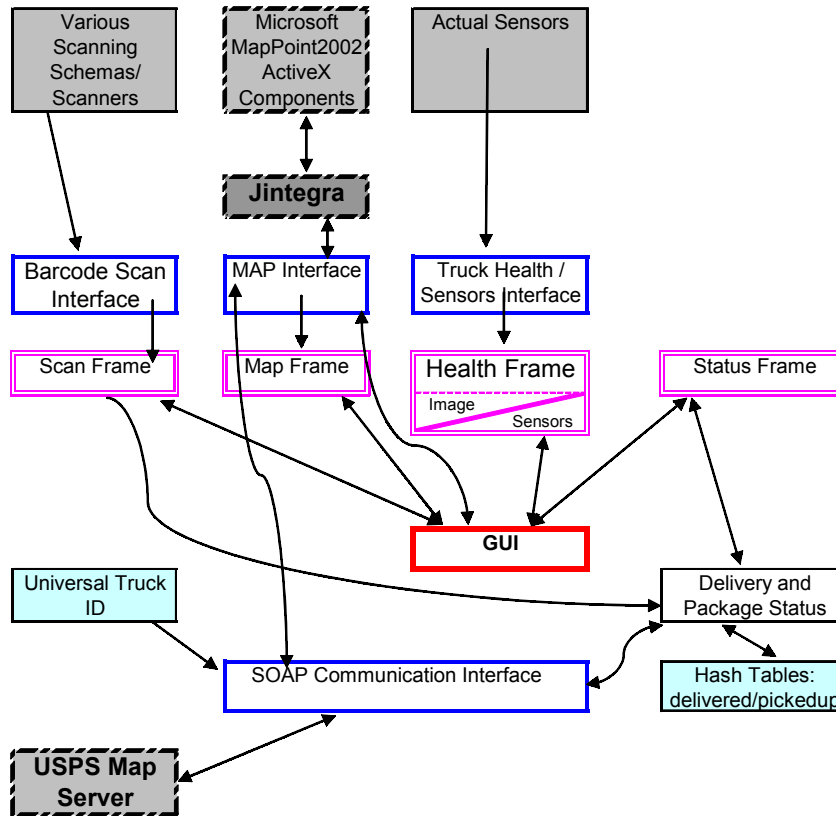
3.5. Sensors

The sensors in the conceptual truck are extremely important for the safety of the truck. The "health" of the truck is mainly based on sensor readings. All sensors will be connected to a converter and the converter to the onboard computer. The function of the converter is to collect the data from all sensors and direct it onto a single connection to the onboard computer. The onboard computer will interpret the data, calculate the overall "health" of the truck and display the results. Information about the "health" of the truck is also stored on the central USPS server for future references during maintenance.

4. Software Modules and Implementation

This chapter will concentrate on the specifics of the implementation of the system. First, the general structure and main modules of the system will be presented. Second, a detailed explanation of these modules functions and implementations will be discussed. One of the by-products in our system comes directly from the choice of implementation language -- the system is modular and object oriented, as any Java based system. Considering the Java inherited modularity of the system, Figure 7 shows an overview of the main modules and their interaction as parts of the system.

Figure 7. Diagram of the main software modules



The main module is the Graphic User Interface (GUI) which is the facilitator of the bi-directional communication between the system and the users of the system, in our case postal delivery personnel. The GUI is actually a container for all the frames displayed to the truck driver. There are four main frames and each of them concentrates on one or more specific functions of the system. The main frames are: navigational map (map frame), health frame, status frame and scan frame. The particular functions and implementations of these frames are presented in subsequent paragraphs. All functionality of the system that requires some kind of external input or output operations is exposed to these external modules through interfaces. The interface structure allows changing or even using more than one external module or system for the same internal functionality. For example, the *Map Interface* connects the external mapping software to the navigational and delivery system, in our case Microsoft MapPoint 2002 is the external product. The interface allows any other mapping software to take the place of the Microsoft MapPoint 2002 or even just to perform only some of the functionality and

supplement the main mapping product. This modularity, provided by the interfaces of the system allows fast and easy extensions and upgrades of the system, without changing its internal structure. Figure 7 also shows the main interactions in the system. One can see that the GUI has a central role as it compiles all the data that is displayed on the screen. There is no particular direction of the information flow as this system gets inputs from the user, the sensors, the navigational software, the central server and from internal computations. The actual information flow is discussed in the following sections that represent the details and the implementation of each internal and external module of this system.

4.1. Graphic User Interface (GUI)

The GUI is the module that provides the interactions between the user of the system and the system itself. The GUI presents to the user what the status of the whole system is and what is the progress of the mail delivery process and the plan for subsequent delivery actions. An extension of the GUI provides a verbal representation of the main delivery and navigational information that the delivery person uses. This external verbal communication module is not embedded in the current version of the product, because it is not representative of the innovative functionality of the system. The verbal communication module is just another possible way to present the data to the driver. Moreover, some of the mapping navigational systems provide verbal communication and the USPS system can just be upgraded to such a mapping product and use its communicational capabilities, rather than cluttering the system with various external extensions.

The implementation of the GUI is done through Java frames, using the `javax.swing.JFrame` class. The actual java file implementing the GUI class is called `JFrameGUI.java` and could be found in Appendix A. The `javax.swing` package provides “a set of ‘lightweight’ (all-JavaTM programming language) components that, to the maximum degree possible, work the same on all platforms.”(Sun Java, November 1998.) The current implementation of the system has four static frames and additional pop-up

frames for scanning delivered or picked-up packages. The static frames are the building blocks of the GUI. They are static in a sense that they fill the whole visible area on the screen. To maintain this integrity of the visible area, which is to maintain the integrity of the user interface (UI) of this software, all of the static frames are created using the following procedure:

```

1      private JFrame createFrame(String name) {
2          JFrame frame = new JFrame(name);
3          frame.setResizable(false);
4
5          if (NoDecoration) {
6              frame.setUndecorated(true);
7          }
8          frame.addWindowListener(new WindowAdapter() {
9              public void windowClosing(WindowEvent e) {
10                 closeAllWindows();
11             }
12         });
13
14         if (numActFrames++ >= FRAME_NUMBER ) {
15             System.out.println("FATAL ERROR ID:023985dkf09 \n The Maximum Number of active Frames was
16             number");
17             Toolkit.getDefaultToolkit().beep();
18         }
19         else {
20             frameArr[numActFrames-1] = frame;
21         }
22         return frame;
23     }

```

This procedure creates a generic JFrame with a set of specific properties. The first property is to remove the resizable capabilities of the frame (shown on line 3 of the procedure.) This property allows the UI of the software to maintain its shape and look throughout the operation of the software. The initial sizing of all frames is explained below, in *Sizing of the GUI Frames* section. The second property of each frame is determining whether the frame will be decorated or not. The difference between decorated and non-decorated frames is shown in Figure 8.

Figure 8. Difference between decorate and non-decorated Java frames

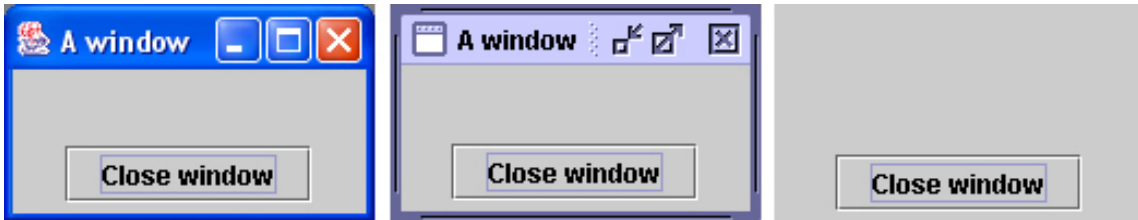


Figure 8 above shows three frames. The first two frames from right-to-left are decorated using the Windows XP and the Java standard decorations. The third frame uses non-decorated frame. To determine whether the static frames will be decorated or not, lines 5 through 7 of the `createFrame()` procedure perform a check whether a global Boolean parameter is set. The static GUI frames are by default set to non-decorated to create a uniform single-window look of the GUI.

To guarantee that all frames behave as a single software product, all frames are connected so that they can be closed at the same time. This behavior is achieved by creating a common `closeAllWindows()` procedure that is called upon the window closing event on any of the static frames. The actual call is embedded into a *WindowListener* event for each frame (lines 8 through 12 above.) The `closeAllWindows()` procedure can also be found in Appendix A as part of the `JFrameGUI.java` file.

The `createFrame` procedure also keeps track of the number of open frames and alerts the user for possible errors. This is a software internal check that makes sure that a pre-set global variable for the maximum number of frames is not exceeded. This setting for the maximum number of frames is part of the initialization parameters of this software.

4.1.1. Sizing of the GUI Frames

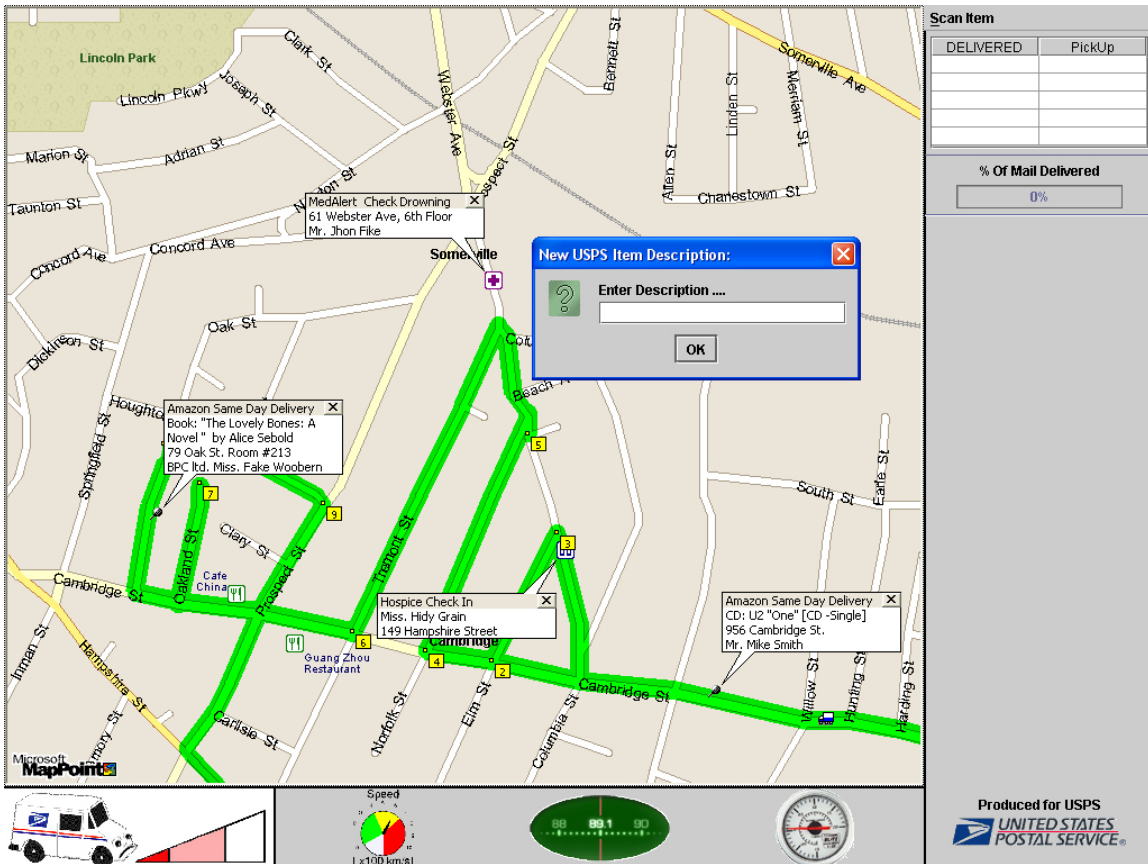
The actual sizing of all of the static frames is done as part of the initialization process of the software. The call of the frame-sizing procedure is done before any of the frames is created to ensure the integrity of the UI. The actual frame sizes are determined by a stand-alone procedure called *calculateFrameSizes()*. This procedure can be found in Appendix A as part of the `JFrameGUI.java` file. The role of this procedure is to set global

variables for the sizes and locations of all static frames based on the size of the screen and internal variables and constraints. The main constraints are the non-vector objects like static images.

4.2. Frames Implementation

After the layout for the display is calculated in the initialization process of the system, all frames for the GUI are created by calling the `createFrame()` procedure discussed above for each one of them. Once the static JFrames are created, each one of them is further manipulated to its designated shape, position and function. In addition to the static frames, which are the building blocks of the look of the GUI, there are pop-up frames that are temporary frames used to collect or display data. These frames are called scan frames, as collecting the scanned barcode information is their main function. The detailed description of the scan frame is presented later in a separate section. Figure 9 shows how all of the frames are laid out to form the GUI.

Figure 9. Layout of the static GUI frames and one pop-up frame



The picture above highlights the four main static frames forming the GUI and a sample of a scan frame which is a temporary pop-up frame. The pop-frame is the only decorated frame and it waits for description information of a package to be entered. The map frame is the biggest frame positioned in the top left corner. The health frame is the frame with white background and positioned in the bottom left corner. The sensors frame is the middle bottom frame displaying three sensors readings. The last frame is the status frame, occupying the whole right side of the GUI. The following paragraphs describe the details of creations and functions of each frame used in the navigational and delivery USPS system.

4.2.1. Map Frame

The map frame is the main navigational and delivery frame. This frame is a static

frame that is visible at all times, as part of the GUI. The background of the map frame is based on a location map provided by the MP2002 software. The map frame displays an area map centered at the location of the truck, determined by the navigational system and, in our case, by the GPS unit. The actual width of this frame is four fifths of the whole screen and the height is the height of the screen minus the pre-set size of the health frame. These values for the size are calculated upon initialization of the system. This is the only frame that directly displays information from an off-the-shelf software product. This whole frame uses the MP2002 engine to create and display the interactive map. What actually happens is that MP2002 has ActiveX Controls “specifically designed to facilitate distribution of components” (Microsoft, March 1999.) “With MapPoint 2002 and the ActiveX Control, Microsoft offers developers a tool to include maps and location-based information in their solutions” (Andrews Sandra, August 2001.) The map frame indeed includes MP2002’s map and location-based information into the navigational and delivery system of the USPS. In addition, on top of the MP2002 map layer, our system adds USPS specific information about package deliveries and services. What the map frame provides is all of the information about package delivery and pick-up that the postal worker might need as well as navigational information and customized information about various services. All of the information is provided by creating a layer of labels and information boxes displaying the specific tasks the postal worker should perform.

To access the ActiveX controls and all objects and procedures in MP2002 from the Java programming language I used a bridge produced by J-Integra® 1.5.5 (Intrinsyc Software, 2003.) J-Integra provides is a pure Java to COM bridge. This bridge allows accessing COM Objects of MP2002 from our pure Java applications. The specifics are that for each COM class in MP2002, J-Integra creates a Java class that can access the corresponding MP2002 COM class. The creation of the Java bridge-classes requires using the J-Integra®'s *com2java* tool (Intrinsyc Software, 2003.) This tool creates proxy classes to access the COM classes of MP2002. The actual proxy classes can be found in Appendix A section MP2002 Proxy Classes. The following lines of code are from the *JFrameGUI.java* and represent the life cycle of a *MappointControl* object which is a MP2002 object.

```

1     private final mappoint2.MappointControl mpctrl = new mappoint2.MappointControl();
2     mpctrl.setSize(w_map, h_map);
3     frame.getContentPane().add(mpctrl, BorderLayout.CENTER);
4     mpctrl.openMap("Path to this truck's defeault route");
5     ....
6     mpctrl.closeMap();

```

Line 1 of the code calls the MappointControl class, a bridge interface class create by J-Integra), and creates an instance of a MappointControl object. The mappoint2 is the package of proxy classes generated by the *com2java* tool. After the MappointControl object is created, we can access all functionalities of the MP2002 COM bridged-classes. Lines 2, 4 and 6 are the basic steps to open, to size and to close a map. Between these lines one can manipulate the map in any fashion available in MP2002. This includes finding optimal routes, adding pushpins (delivery objectives) or displaying particular locations information in bubbles around the points of interests. One can go through the mappoint2 package generated by J-Integra to familiarize with the wide range of capabilities provided. This package is available in Appendix A MP2002 Proxy Classes. Using this various operations over an active map allows this navigational and delivery software for the USPS conceptual truck to customize the directions and data displayed on the map and to create a customized solution from the MP2002 product.

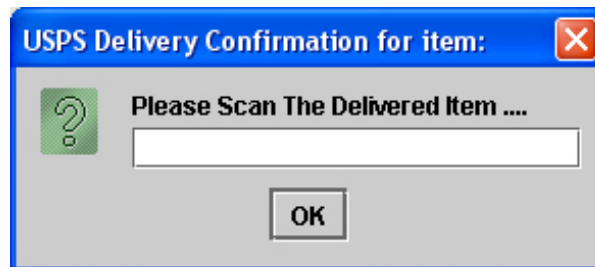
The above paragraph provided an overview of how the navigational and delivery map in the map frame is created and modified. After the initial map for the map frame is loaded the map frame can receive any number of updates based on the GPS readings and server updates and then calls the MappointControl to apply the changes to the active map. This combination of a stationary off-the-shelf software product and its encapsulation in customized constantly updating system provides for the functionalities of the navigational and delivery map of this USPS custom product.

4.2.2. Scan Frame

Scan frame are the only type of non-static frame displayed in the GUI. This type of frame is a pop-up window that appears only when a package is to be scanned. Once the package is scanned, the frame records the scanned information and closes automatically.

The implementation of this pop-up frame uses a JOptionPane from the javax.swing package. The actual implementation can be found in Appendix A, the PopUpInput.java. The procedure called to invoke a pop-frame is getString(String prompt, String frameTitle). Figure 10 below shows the actual pop-up window that appears when a package is to be scanned before delivery.

Figure 10. A Scan Frame that appears as a pop-up window to scan the barcode of a package that is being delivered



The actual input for this window comes directly from the laser barcode scanner. After the barcode is read by the scanner, this automatically closes the pop-up window and the data is recorded. Based on the barcode scanned, the information about the package corresponding to this barcode is updated. Simultaneously, the information for the delivered package appears in the Status Frame, which displays the information for the last five delivered packages. The scan frame is used in other functions of the navigational and delivery system in a similar fashion. For example, the pick-up package status uses two scan frames. The first frame, exactly as the delivery scan frame, reads the barcode of the package being picked-up. Then a second scan frame is used to input any additional information about the package that might be needed. Usually, this additional information is a second barcode that the USPS uses to code the exact destination and type of the package. The size of the scan frame is determined by the OS running the software and is based on the size of the text displayed in the frame. In general, we can conclude that the function of the scan frame is to collect information from the user and then record this information for further processing.

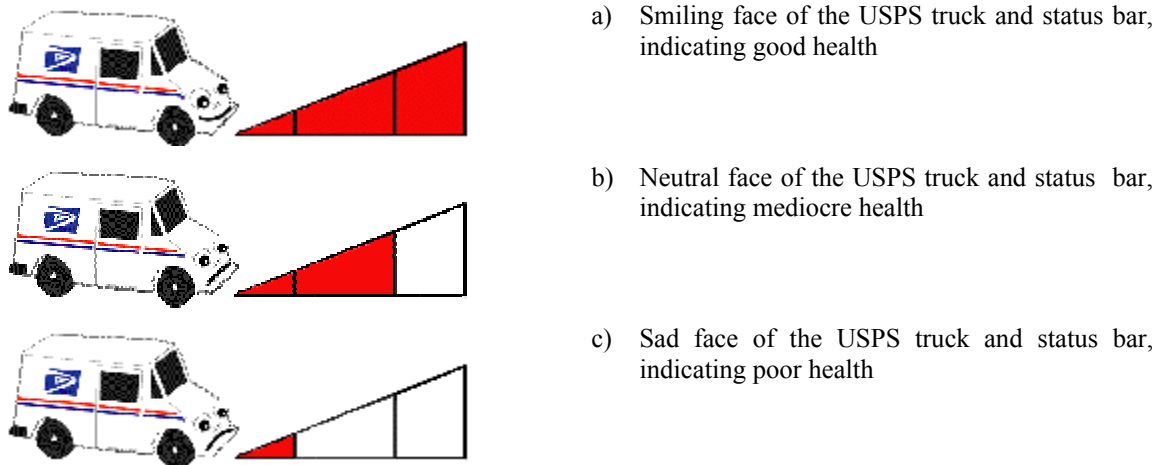
4.2.3. Health Frame

The health frame is the static frame that appears at the bottom of the GUI. The health frame is directly below the map frame and constitutes of two sub frames – the sensors frame and the actual health frame or image displaying the “health” of the truck.

4.2.3.1. Actual Health Frame

The actual health frame is positioned in the left-most lower corner of the GUI. The left part of the health frame displays an image of the USPS postal truck with smiling, neutral or sad face. Next to the “happy” truck image is a status bar displaying how healthy the truck is. Figure 10 shows the three main possible states of the health frame. These states include the images of the truck and the corresponding graphic displays associated with each “happy” truck picture.

Figure 10. Three basic states of the health frame



Parts a), b) and c) of the figure above show the three main possible health statuses displayed in the health frame. By interpolating these images, the health frame can display additional intermediate health statuses for the truck. One of them is shown on Figure 5 in the *Health of the Truck* section. The size of this frame is constrained by the size of the USPS “happy” truck images which is taken in consideration in the `calculateFrameSizes()`

procedure executed during initialization. As a static frame the health frame is created in the JFrameGUI class from the createFrame() procedure and then added to the visible static GUI. The actual call that builds this frame is called createHealthFrame() and can be found in Appendix A JFrameGUI.java

4.2.3.1 Sensors Frame

The sensor frame is another static frame that displays information about the health of the truck. This frame is a container for all sensors readings. The frame monitors whether any of the sensors are beyond a critical point and then displays the readings and the information about particular sensors. Various numbers of sensors can be displayed simultaneously. An initial setup can determine which, if any, sensor's readings will be constantly displayed. For example, a sensor reading is considered the laser barcode scanner and thus this frame can display the current package information details. In addition, the sensor frame can be set up to display readings from the onboard GPS unit translated to driving directions and details about the next delivery task. The actual setup of this frame will be determined when all needed sensors are installed and a decision is made on how critical each one of them is.

4.2.4. Status Frame

The status frame is the last building block of the static GUI look. The position of the status frame is in the right side of the display. The status frame displays delivery specific information. In particular, it monitors the DeliveryAndPackageStatus module (see Figure 7 of the main software modules) which holds the local information about all packages in the process of being delivered and all packages already delivered. This module also keeps track of the percentage of the daily mail delivered. This information is displayed on the screen to inform the driver about the progress of his daily mail delivery tasks. The DeliveryAndPackageStatus class keeps a java.util.HashMap with all packages that were processed by this delivery truck during the day and their status. There are also daily

counters for the number of delivered, picked-up and in the process of delivery packages. This information provides constant updates about the daily progress of the mail delivery. The java procedure below highlights the general steps for creating the status frame.

```
1      protected JFrame createStatusFrame() {
2          JFrame frame = createFrame("Status Frame");
3
4          MenuHandler menuHandler = new MenuHandler();
5          JMenuBar menuBar = menuHandler.CreateJMenuBar();
6          frame.setJMenuBar(menuBar);
7
8          JScrollPane scrollpane = getTableColumn().tableColumn.getJSP(w_statf, 100);
9          contentPane.add(scrollpane);
10
11         JSeparator jsep = new JSeparator();
12         contentPane.add(jsep);
13
14         JLabel text = new JLabel("% Of Mail Delivered");
15         contentPane.add(text);
16
17         status_progressBar= ProgressBar.GetProgressBar(JProgressBar.HORIZONTAL);
18         contentPane.add(status_progressBar );
19
20         JSeparator jsep2 = new JSeparator();
21         contentPane.add(jsep2);
22
23         ImageIcon uspsLogo = new ImageIcon("../images/uspsLogo.gif", "");
24         JLabel label1 = new JLabel("Produced for USPS",uspsLogo, JLabel.CENTER);
25         label1.setVerticalTextPosition(JLabel.TOP);
26         label1.setHorizontalTextPosition(JLabel.CENTER);
27         contentPane.add(label1);
28
29         frame.pack();
30         frame.setSize(w_statf, h_statf);
31         frame.setVisible(true);
32         return frame;
33     }
```

The above procedure, `createStatusFrame()` is part of the `JFrameGUI.java` file. The full procedure can be seen in Appendix A. The procedure presented above omits most of the minute details due to space limitation. The `createStatusFrame()`, as all static frames part of the GUI starts by calling `createFrame("Status Frame")` procedure, described in the beginning of the **Graphic User Interface (GUI)** chapter. This call, on line 2, creates a generic empty static frame for the GUI. The next step in the creation of the status frame is the generation and addition of a `MenuBar`. This is shown in lines 4 through 6. The `MenuBar`

is created in the MenuHandler class by calling its CreateJMenuBar() procedure on line 5. The implementation of the MenuHandler class is available in Appendix A for reference. The function of the MenuBar is to provide for the ability to call various functions like scanning new items, adding their descriptions and also customizing the GUI. This is the only MenuBar in the GUI and it provides small number of functions and settings for the navigational and delivery USPS system. This is the only way to modify and customize the software, while it is running. Future testing might lead to the conclusion that this MenuBar should not be available as part of the active delivery GUI. After the MenuBar is added to the status frame, lines 8 and 9 create a table for displaying the status of the last five delivered and picked-up packages. This table is created in a class called TableColumn, which is also available in Appendix A. The settings for the table could be adjusted to display the history of more than 5 items. The table allows obtaining more detailed information about the status of the displayed packages by clicking on the particular item. Line 11, 12, 20 and 21 just add frame separators to localize the different functions displayed in the status frame. Lines 14 through 18 create a textual input using a JLabel and a progress bar that displays the percentage of the daily mail delivered so far. The progress bar creation is done by calling the ProgressBar class (see Appendix A) and then the value of this progress bar is updated by calling an internal procedure setSPB(new_value). As discussed before, this new value is calculated in the DeliveryAndPackageStatus class by calling its percentageDelivered() method. Finally, the status frame displays the logo of USPS which is loaded from a .gif file and then added to a JLabel to be displayed in the status frame. This is shown in lines 23 through 27. The rest of the procedure shown sets the size of the status frame, makes it visible and returns the frame to the caller of the procedure.

4.3. SOAP Communication Interface

The SOAP communication interface provides the Internet data channels between the central USPS server and the mobile system. The interface uses the SOAP protocol, which was already mentioned in section 2.2.2. *The Server Software*. The SOAP communication interface is the place where all requests and updates to the USPS central

server are conducted. The truck first authenticates to the server using his unique truck ID. After the truck is authenticated, all of the information that was stored after the previous update gets send to the central USPS server. Once all updates are confirmed to be received by the central server, the truck receives all new updates needed for the USPS navigational and delivery system. The SOAP calls are Remote Procedure Calls (PRC) generated form the Java org.apache.soap package. This package is an extension of Java that supports the SOAP communication. This same package should be running on the central USPS server to guarantee that the RPCs are formed by the strict specification of SOAP and thus the two parties will be able to communicate. The following code demonstrates how a SOAP call is invoked to retrieve information from the USPS central server.

```
1      // Build the Call object
2      org.apache.soap.rpc.Call call = new org.apache.soap.rpc.Call( );
3      call.setTargetObjectURI("urn:USPS_truck_server");
4      call.setMethodName("getUpdates");
5      call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC);
6
7      // Set up parameters
8      Vector params = new Vector( );
9      params.addElement(new Parameter("TRUCK_ID ", String.class, TRUCK_ID, null));
10     params.addElement(new Parameter("typeOfUpdatesRequested ", String.class, typeOfUpdatesRequested, null));
11     call.setParams(params);
12
13     // Invoke the call
14     org.apache.soap.rpc.Response response;
15     response = call.invoke(url, "");
16
17     //Read the response
18     Parameter returnValue = response.getReturnValue( );
```

The first step of creating a SOAP RPC is to build the SOAP call by creating a new org.apache.soap.rpc.Call() on line 1. After the call is created, there are some mandatory parameters that need to be set. Line 3 specifies the name of the SOAP service running on the server. This service name in our case is urn:USPS_truck_server, the urn: identifies that this is indeed SOAP service. All SOAP services' names begin with the urn: identifier. This service name is provided by the server side administration and is determined upon deployment of the service. The actual service is a class that the USPS central server runs

and this class can perform particular services and calculations. This class can have a number of methods that are declared as accessible through the SOAP service. In the code example above, the method requested is passed as parameter to the call object in line 4. The name of the method requested from the urn:USPS_truck_server service is getUpdates. This method is part of the USPS_truck_server class deployed on the USPS central server and will return all of the updates designated to the truck requesting the updates (the truck making the SOAP RPC). Line 5 determines what kind of encoding will be used to transfer the input and output parameters of the SOAP RPC. The next step is to add the input parameters to the call object. In our case, there are two parameters added on lines 9 through 11. The input parameters are TRUCK_ID and typeOfUpdatesRequested. These parameters will be added to the SOAP RPC and passed to the getUpdates method of the USPS_truck_server class running on the USPS server and a Java object with all updates requested by truck with the specified TRUCK_ID will be returned as in an org.apache.soap.rpc.Response. This response is received after the invocation of the SOAP call in line 15. The invocation of the call requires to specify the network address of the USPS server, which is passed in the url parameter on line 15. After the response is received it is parsed and all updates are used to update the onboard navigational and delivery system of the truck. This process, although seemingly long and complicated, takes just a fraction of a second for a reasonable size request. The actual set of the SOAP call is mechanical process and is easily automated for sequence of calls.

The parameters needed for the creation of the SOAP call, like the server address, the name of the SOAP service and all possible method invocations, are provided by the server side administration in a well-structured .wsdl file (WSDL). To automate the process of SOAP call creation, a WSDL2Java utility is provided in the org.apache.axis package that “transforms WSDL documents into Java interfaces ... and then invokes the [rmi2soap](#) compiler to generate SOAP stubs and skeletons” (SilverStream Software, Inc., 2001-2001.) The SOAP stubs and skeletons are accessed from the Java interfaces created and to perform the actual SOAP call, one need only to call the interface of choice and pass on the required parameters. This utility allows the process of creating SOAP RPCs to be automated.

4.5. Barcode-scanning Interface

The barcode-scanning interface provides flexibility to change the type of scanning device used in the system. In addition, it is possible to have more than one type of scanning device operated in different locations without changing a line of code in the navigational and delivery system. The interface could be implemented by as many scanning devices or techniques as needed. For example, the future integration of the RFID scanning technology in the system could easily be added as a parallel scanning technique while it is being tested and tuned for full production scales. The implementation of additional scanning class, which reads barcodes in some fashion, requires only that these classes send the obtained barcodes directly into the `DeliveryAndPackageStatus` class. In fact, the only implemented procedure that is mandatory is `thisNewClass.getString()` and a call to this method should return a valid barcode obtained by this new class. This method is called as part of the `MenuHandler` class, which then updates the HashTables for all packages in the `DeliveryAndPackageStatus` class. Both classes can be found in Appendix A.

4.6. Map Interface

The map interface provides for the interconnectivity between the navigational and delivery USPS system and the off-the-shelf mapping product used in the map frame of the system. In this implementation of the system, as discussed in section *4.2.1.Map Frame*, MP2002 is the external mapping product and its interface is created through J-Integra. The actual interface class is the `IMappointCtrl` interface part of the `mappoint2` package created by J-Integra and available in Appendix A. The class that implements the `IMappointCtrl` interface is the `MappointControl` class in the `mappoint2` package, discussed in section *4.2.1.Map Frame*. To change the mapping software used in the USPS system, one needs to guarantee that the new mapping software has comparable functionalities and should build a new class similar to the `MappointControl` class that implements the

IMappointCtrl interface and connects to the new mapping software. The new mapping software might be based on totally different technology and not use ActiveX controls and J-Integra as a bridge.

4.7. Truck Health and Sensors Interface

The truck health and sensors interface is the collection box for all sensorial readings in the truck. This interface was designed and created with the purpose for any number of sensors and their inputs to be included as part of the system. The intended use is that a hardware network will connect all sensors and then package this information for the onboard computer. The actual specifications and complexity of this sensorial network is to be designed and implemented by knowledgeable entities in this field. The inputs from this network will be received from the truck's health and sensors interface for compilation of the data. As already discussed in the **Health of the Truck** section, all sensorial data is being stored for future maintenance and also used to calculate the overall "health" of the truck and to update the central USPS server with information about the status of the truck. The GPS readings are the only sensorial readings currently received by the navigational and delivery software system. After actual network for the sensors is implemented in the truck the GPS readings will come as part of the data received from the sensors converter as described in section 3.5.*Sensors*. Nevertheless, the design and implementation of the software maintains the truck health and sensors interface to allow for connectivity between the software system and any future implementations of the network of sensors.

5. Initialization of the System

The flawless operation of the system requires a proper initialization of the system. The standard operation of the system is set up as a continuously running central USPS server and many mobile navigational and delivery systems connect and disconnect to the central server. The initialization of the server requires that all software and hardware on

the server side is implemented and connected. This includes the deployment and distributing specifications of all possible web-services that the mobile trucks will access using the server specifications in .wsdl created. The details of this process are provided in section 4.3 *SOAP Communication Interface*. Having a running server available, each truck will theoretically initialize its mobile navigational and delivery system before leaving the docking postal station. This initialization allows all necessary updates to the system to be made while reliable wireless connection is available. During this initialization process, the download of large software and database updates could be conducted. For example, the most recently updated roadmap information will be updated and stored on the onboard computer. During operation, only small packages with new updates of the map will be received by the truck. This requirement for a reliable wireless connection during initialization is not mandatory but decreases the traffic load on the open wireless 802.11b networks and the time required for the onboard system to have fully updated information database.

The particular steps needed to start the initialization of the current implementation of the system, require to compile the Java source code and to simply execute the JFrameGUI class with no additional parameters. A compiled package is also available but this package is not platform independent. Details for obtaining this package could be found in Appendix A. The compilation and execution of the actual code of the system guarantees the successful operation of the system on any OS running the Java Virtual Machine. A .jar file containing all Java classes, internal packages and media files used in the software system is also available for the distribution of the system (see Appendix A). The only additional requirement is that the computer running the navigational and delivery software needs to have the MP2002 software installed, which is the only external software package used in this implementation. The J-Integra bridge, used to connect to the MP2002, is included in the .jar file for convenience. Appendix A provides information how to obtain the .jar file and all documentation for the latest release of the USPS navigational and delivery system.

6. System Usability and Field Testing

To ensure that the system is designed and implemented in a useful and approachable manner, a series of tests are needed. To conduct any kind of field testing of the implemented system, it needs to pass through USPS' quality control units. Then USPS should mandate a possible field testing of the system. Unfortunately, the conceptual USPS truck is a mobile laboratory and is rapidly changing which have not yet allowed the whole vehicle to go through the quality control of USPS. In addition, as the USPS conceptual truck is property of the government there are very strict regulations on who can drive truck. It is considered felony for someone other than a USPS employee, with a valid license for this particular type of truck, to operate this conceptual or any of the USPS's fleet trucks. These limitations have prevented the planned field testing of the system to take place so far. The only realistic testing of the system was done by a postal worker who was responsible for the transportation of the conceptual vehicle around Cambridge, Massachusetts for various demonstration sessions. The input received from him was very helpful and led to some changes in the GUI design and the functionality of the system in the implementation stages of the project. According to him, the system is easy to understand and manageable to operate. To actually receive a statistically significant results a broader study is considered in the near future. The results of this and future studies will be available on the project website (see Appendix A.) Input from full-time postal delivery drivers was received in the very early stages of the design to obtain their objectives and particular needs. All features of the currently implemented system are coordinated with USPS postal workers and officials to guarantee an optimal performance and value from the system provided.

Usability testing of the system is of an extreme importance before moving the system into any kind of production state. The usability testing will provide the fine details for the GUI layout and the input devices used by the system. This testing will provide a valuable feedback on how easy it is to understand and use the system. In addition, the tests will also provide more ideas of what should be added to or removed from the system to make it more efficient and productive. The users of this system are the USPS truck drivers and their daily experience with the system will determine its benefits in the future.

7. Future Work and Extensions

There are many possible extensions to this system and some of them are highlighted in the paragraphs below. A possible future extension of this system is the implementation of the system in a programming language like Visual Basic that directly supports ActiveX controls. This will remove the need of J-Integra like technology to bridge the COM components into Java.

7.1 Voice Activated Commands and Directions

Safety rulings might require that future releases of the navigational and delivery software for USPS to include a voice activated controls and verbal announcements of driving and delivery information to the postal worker. The speech recognition module might be an external product ported to the system in fashion similar to the way the MP2002 software is. Such a module might be required to allow the driver to receive all necessary information in a verbal form rather than by looking at the LCD display. This extension to the communication capabilities of the system will allow the driver not to lose sight of the road while he is driving. In addition, verbal communication between the system and the driver will provide more mobility and efficiency of the delivery personnel using the system.

7.2. Opportunistic Mobile 802.11b Wireless Network Collection

A very promising extension of the USPS system is to harvest the open 802.11b wireless networks. Based on the pattern and the data from the usage of the open networks a real-time aerial wireless coverage map could be created. Building such a state-of-the-art system that harvests open 802.11b wireless networks is ongoing project (awaiting patent)

in Professor Ted Selker's Context-Aware Computing group in the MIT Media Laboratory. The information from all of the USPS trucks locating and using the open wireless networks could be used to create a real-time map of the available open wireless networks including their strength and bandwidths. The availability of this map is an extremely valuable market commodity and could provide a service for various network providers or other interested businesses and individuals. One possible drawback of this idea is the legal issues concerning the ownership of the open wireless Internet connections and the question whether one could legally use them. Another possible setback of this design is the possible termination of the usage of the Wi-Fi wireless technology. This termination of technology possibility does not seem to be very probable in the near future, considering the rapid increase in the usage and deployment of wireless network technologies.

7.3. Alternative Positioning Systems

Although the current design of the USPS navigational and delivery system uses a GPS device for positioning information, other possible alternatives could provide real-time information about the position of the delivery truck. One possibility is to use the driver's knowledge of his delivery route. This approach relays on the delivery and pick-up address information connected to each package. Once a package is being scanned, the delivery destination or the pick-up location of this package is used to update the current position of the truck. This approach assumes that all deliveries are accurate and the pick-up locations for all packages are pre-determined or explicitly entered into the system. Based on the address information, the truck can determine its approximate longitude and latitude coordinates. These approximate coordinates are used in the navigational system of the truck and by the central USPS server to schedule new tasks for all mobile delivery vehicles. This alternative system to the GPS requires no extra hardware for the system and no specific procedures in addition to the regular delivery process.

Another very promising positioning technology is based on the 802.11b wireless network that the truck harvests while delivering the mail. Each wireless transmitter has a

unique network address called Media Access Control (MAC) address. The MAC address is a “unique number encoded in the circuitry of a device to identify it on a local area network” (Bay Networks Inc., 1998.) Based on this MAC address of the wireless transmitter, the truck can determine its approximate location using the registration information of each unit. In addition, the trucks will be able to determine and update location information for transmitters that are not registered properly on the Internet. Ongoing research exploits the possibility of calculating exact location coordinates based on three or more wireless transmitters in the same vicinity. This method could be used in the future to gather precise location information for the navigational system and fully remove the need for an integrated GPS device.

7.4. System Testing

Field testing and usability tests are one of the most important parts for the future of this system. The system should be thoroughly tested on its accuracy in matching the goals and needs of USPS. Parallel testing will provide vital user feedback on the look and interaction of the current system implementation. Based on the specific results and findings from these studies, a final setup and tuning of the system will prepare the system for its production integration.

7.5. Map Server

For the optimized route calculations of the USPS postal vehicles, a unique map server could be implemented to create a customized map database. This map server will collect real-time information from all USPS trucks and build a map with very specific information about constructions, traffic and condition of the road structure in areas of interest. The idea is that the postal trucks could automatically detect the speed of the traffic around them and provide visual images from the actual road conditions. The speed detection will be based on a sensor detecting the speed of the passing traffic based on the

speed of the truck. The visual images provided by the already installed cameras on the truck can be processed by an image-recognition system and detect the conditions of the roads. These same images could also be used in accidents or emergencies by various authorities to determine the actions and resources needed to resolve particular problems. The map database will not only supply the USPS with up-to-date information but could also provide additional services to drivers and navigational systems. The knowledge of the real traffic conditions is very valuable to navigate the general street traffic, especially during rush hours. This map server could also use inputs from all state and local authorities to update scheduled road construction and to monitor its progress and conditions.

Many other services and updates are possible and might appear in the future releases of the mobile USPS navigational and delivery system. All of these updates will be based on the needs of USPS and the feedback received from the deployment of the current implementation.

8. Conclusion

The customized mobile USPS navigational and delivery system presented in this optimizes and solves a number of specific tasks that are part of the daily mail delivery process. This system provides a robust and expandable software solution together with a sample hardware implementation. The implemented software uses the Operating-System-independent Java programming language. The communication solutions used in this product are based on the secure Simple Object Access Protocol for the Internet. The current release of this system contains two additional embedded software packages. The first one is the Microsoft Streets and Trips 2002 mapping solution and the second one is the J-Integra package that allows access to the ActiveX controls of the Microsoft Streets and Trips 2002. The hardware implementation was built in the USPS conceptual truck and consists of an onboard computer, custom display, sensors and communication

hardware. This system is designed to be easily extendable and upgradeable to allow fast and effective future releases based on new business needs and feedback received from previous implementations of the system.

Acknowledgements

Professor Ted Selker is the main initiator and supervisor of this project. He is an endless source of great ideas. Thanks for your support and attention to this project.

Ken Paul for his help in synchronizing our communication with the USPS officials, for his valuable consulting functions in this project and for his help in making the USPS conceptual truck available for this project.

Betty Lou for her support, interest and logistics help in this project

Jordan Braynov for his help with the hardware system implementation.

Vilislava Petrova for her support and for accepting my long hours of work and frustration.

USPS for providing the conceptual truck and the requirements for this project.

References

1. The New York Times, December 5th 2002. *High-Speed Wireless Internet Network is Planned*.
2. Cometa Networks, January 29th 2003. [WWW Document] URL <http://www.cometanetworks.com/> (Visited March 20th, 2003)
3. Wi-Fi Alliance, 2003. Wi-Fi Alliance index. [WWW Document] URL <http://www.weca.net/OpenSection/index.asp> (Visited May 03, 2003)
4. FedEx, May 18th, 2002. About FedEx, FedEx Technology Overview [WWW Document] URL <http://www.fedex.com/us/about/overview/technology.html?link=4>. (Visited April 19, 2003)
5. UPS, 1999. UPS Pioneers Real-Time Package Tracking in Hong Kong. [WWW Document] URL <http://pressroom.ups.com/pressreleases/archives/archive/0,1363,3612,00.html> (Visited April 19, 2003)
6. UPS, 2002. UPS Receives "CIO-100" Award for Technology, Process Integration. [WWW Document] URL <http://pressroom.ups.com/pressreleases/archives/archive/0,1363,4143,00.html> (Visited April 19, 2003)
7. Vert Inc., 2003. Vert Intelligent Display (VID). [WWW Document] URL <http://www.vert.net/> (Visited April 28, 2003)
8. Cappuccino PC, 2003. Cappuccino : Mocha P4 PC [WWW Document] URL <http://www.cappuccinopc.com/mochae7042b.asp> (Visited April 2, 2003)
9. Garmin, 2003. Garmin: GPS II Plus Specifications [WWW Document] URL <http://www.garmin.com/products/gpsIIp/spec.html> (Visited April 2, 2003)
10. RFID Journal, 2003. Breakthrough on 1-Cent RFID Tag [WWW Document] URL <http://www.rfidjournal.com/article/articleview/273> (Visited April 29, 2003). Also talk by Infineon Technologies AG at MIT room 2-190 on April 17th 2003.
11. Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, Dave Winer May 8th, 2000., Simple Object Access Protocol (SOAP) 1.1 [WWW Document] URL <http://www.w3.org/TR/SOAP/> (Visited October 12th, 2002)
12. Intrinsyc Software, 2003. ntroducing J-Integra® 1.5.5. [WWW Document] URL <http://www.intrinsyc.com/support/j-integra/> (Visited April 5th, 2003)

13. Intrinsic Software, 2003. The com2java Tool [WWW Document] URL <http://www.intrinsic.com/support/j-integra/doc/com2java/index.html> (Visited April 5th, 2003)
14. Sun Java, November 5th 1998. Swing 1.1 API Specification: Package javax.swing [WWW Document] URL http://java.sun.com/products/jfc/swingdoc-api-1.1/javax/swing/package-summary.html#package_description (Visited April 15th, 2003)
15. Microsoft, March 30th, 1999. ActiveX Controls [WWW Document] URL <http://www.microsoft.com/com/tech/ActiveX.asp> (Visited April 1th, 2003)
16. Andrews Sandra Microsoft Corporation, August 2001. Introducing the New Microsoft MapPoint 2002 ActiveX Control [WWW Document] URL <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmap02/html/introactivex.asp> (Visited April 1st, 2003)
17. SilverStream Software, Inc., 2001-2001. wsdl2java- jBroker Web WSDL to Java compiler [WWW Document] URL <http://www.novell.com/documentation/lg/workbench4/docs/jbroker-web/docs/reference/tools/wsdl2java.html> (Visited April 1st, 2003)
18. Bay Networks, Inc., 1998. MAC [WWW Document] URL http://support.baynetworks.com/library/tpubs/terms/gloss_m.htm#1017571 (Visited April 8th, 2003)
19. The Association for Automatic Identification and Data Capture Technologies (AIM, Inc.) July 18, 2002. What is Radio Frequency Identification (RFID)? [WWW Document] URL http://www.aimglobal.org/technologies/rfid/what_is_rfid.htm (Visited May 23, 2003)