Lab 1 – AIR Tracker Description

Rahil Patel

CS411

Janet Brunelle

March 04, 2009

Table of Contents

List of Figures

1 Introduction

Mishandled luggage costs the airline industry as a whole $3.8 billion a year [SITA]. Returning mishandled luggage wastes the traveler's time and airline's money. If a bag is sent to the incorrect destination, too late for the passenger's itinerary or lost altogether, the bag is labeled mishandled. Travelers can become frustrated after their bag is mishandled for several reasons. People may have necessities, such as clothes or business documents, in their bag. They may have invaluable items that cannot be replaced by a small sum of money. Also, waiting for the bag to be retrieved may cause the traveler anxiety. In the future, the traveler may reconsider booking a flight with that airline next time.

Airlines suffer even more and there are government provided statistics that prove this. According to the U.S. Department of Transportation, in August 2008, bags were mishandled at a rate of 8,700 a day by American-based airlines. Each mishandled bag costs the airport an average of $90 per bag [SITA]. In order to recover bags they airlines must to hire more personnel and use more resources to save them. Some airlines are known to have a high rate of mishandled bags which customers have realized. For example, Delta Airlines has been dubbed with the following acronym: Don't Expect Luggage to Arrive!

To solve this problem, a better quality assurance system is needed. The system would need to use current technology such as Radio-Frequency Identification (RFID) scanning to track the bags. The system should have more checkpoints to pinpoint the exact location of errors. Since 61% of all mishandled bags are caused by transfers, areas involving transfers such as gates and carts should be the main focal point [SITA]. The solution should provide complex reports through an easy-to-use GUI. Together, these tasks create the four modules of the solution; routing, cart, gate, and reporting.

2 Product Description

AIR Tracker does all of the previously stated and more. In addition to the checkpoints current quality assurance systems have (check-in and sorting), AIR tracker is able to track bags when they are loaded and unloaded from the cart which is crucial during transfers. Not only does the system track the bags, it can also sense when a bag has strayed away from its correct route and can safely send an alert to the correct staff member. Furthermore, the cart scans would eliminate the need of scanning the bag at the bottom of the belt loader, thus effectively reducing the time for all flights, including transfers.

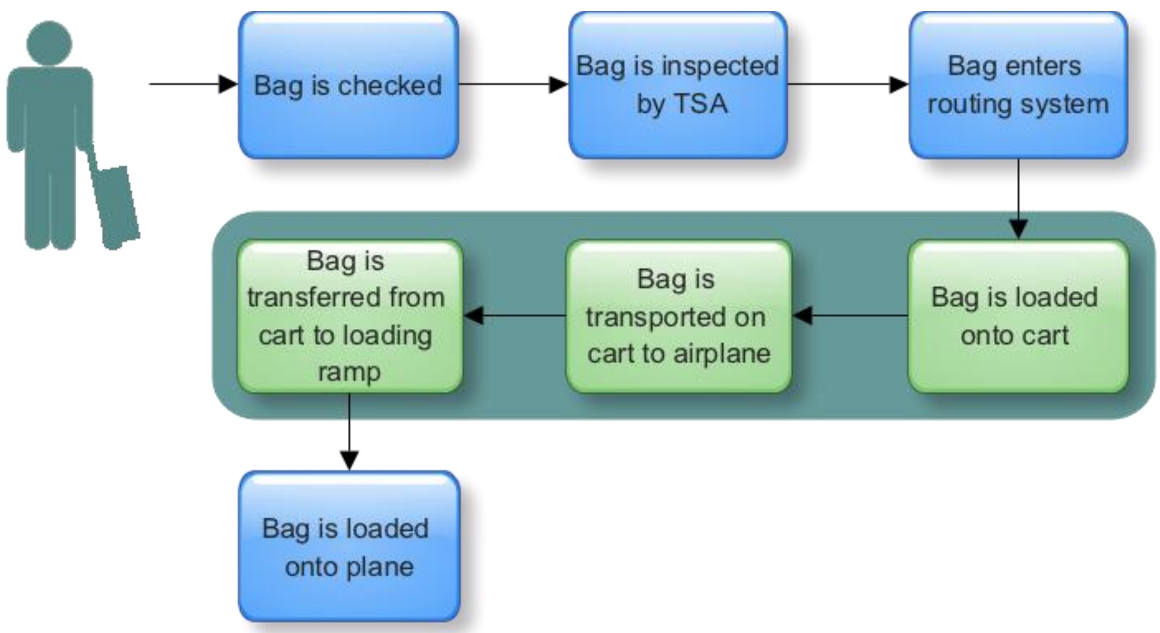

Figure 1. Bag Flow

AIR Tracker uses five well placed checkpoints to gather data. They are check-in, TSA inspection, loading the pusher, loading the cart, and loading the airplane. The first three checkpoints are provided by most existing baggage handling systems while the highlighted three are provided exclusively from AIR Tracker due to the innovative use of cart sensors and wireless antennas. During transfers, the two exclusive checkpoints are used twice as much, which will provide crucial data and more possibilities to save mishandled bags with real-time alerts.

Before AIR Tracker can attain its goals, there are many objectives it must conquer. It must provide real-time tracking of bags throughout the Ground-level Routing Process (GRP). It must alert the airport baggage handling staff when bags get off track. Lastly, it must provide a historical summary of alerts to assist the airport in finding problem areas within the baggage handling system. Only after completing these objectives, the goals can be attained.

The ultimate goal of AIR Tracker is to reduce the rate of mishandled baggage. Immediately it will only save a few bags in comparison to how much the airport can save if the product is taken full advantage of. After a fair amount of time has passed, AIR Tracker can create reports based on the data it has been gathering since the day it was implemented. These reports are the key to save mishandled bags because they will tell the customer exactly where in the GRP bags are being mishandled.

2.1 Key Product Features and Capabilities

AIR Tracker has several features, many of which are unique and put the product above any of the competition. The main attraction is the cart reader. The cart reader scans bags at the gate and the airplane, sending bag data to the applications server wirelessly. This removes the need for a scan before the cargo bay, which effectively saves time. The readers also send alerts to the driver if a bag falls off, which would directly save mishandled bags.

For the cart reader to figure out where the bag's destination is, the cart reader needs memory and firmware. The memory is needed to hold a cartload of bags' information and the firmware is needed to tell where the bag is supposed to be. The following diagram illustrates all of the possible states of a bag on a cart.

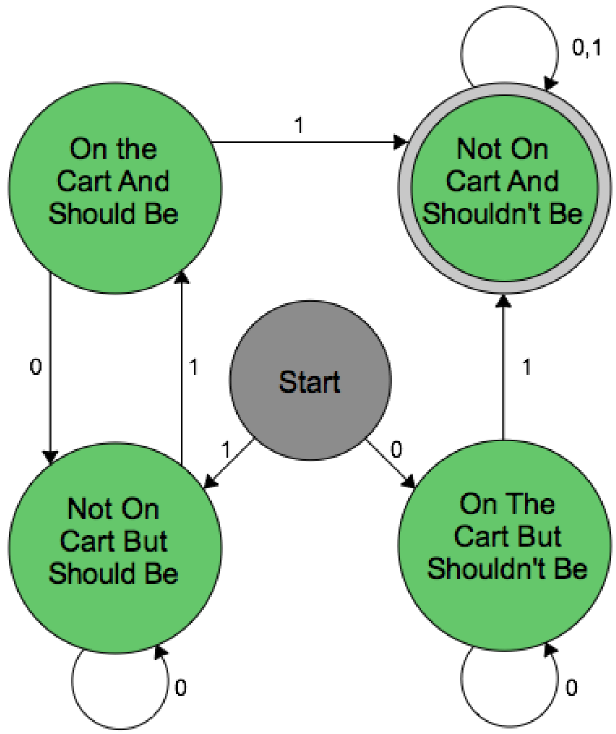

Figure 2. Cart State Diagram

Routing algorithms are used to determine whether a bag is on the right track or not. A similar diagram can be used to determine the possible states a bag can be at when it reaches a gate. Depending on the state, the system may send an alert. Since AIR Tracker is custom-built to fit the airport, the routing system will be able to generate more accurate reports than existing systems. This tight integration allows the AIR Tracker to offer real-time alerts. The alerts can be displayed and reported in many different ways, utilizing all historical data from the time AIR Tracker was implemented.

2.2 Major Components (Hardware/Software)

Figure 3 displays the flow of data throughout major components of the AIR Tracker. Whenever a bag passes through a checkpoint, it is scanned by a RFID scanner. The data is then sent to an applications server via high speed Ethernet cables. The applications server continuously gathers data from both the scanners and the airport's check-in application. The data is then computed and used to send real-time alerts to the staff's handheld devices. Finally, it stores the data in the AIR Tracker database to create reports in the future.

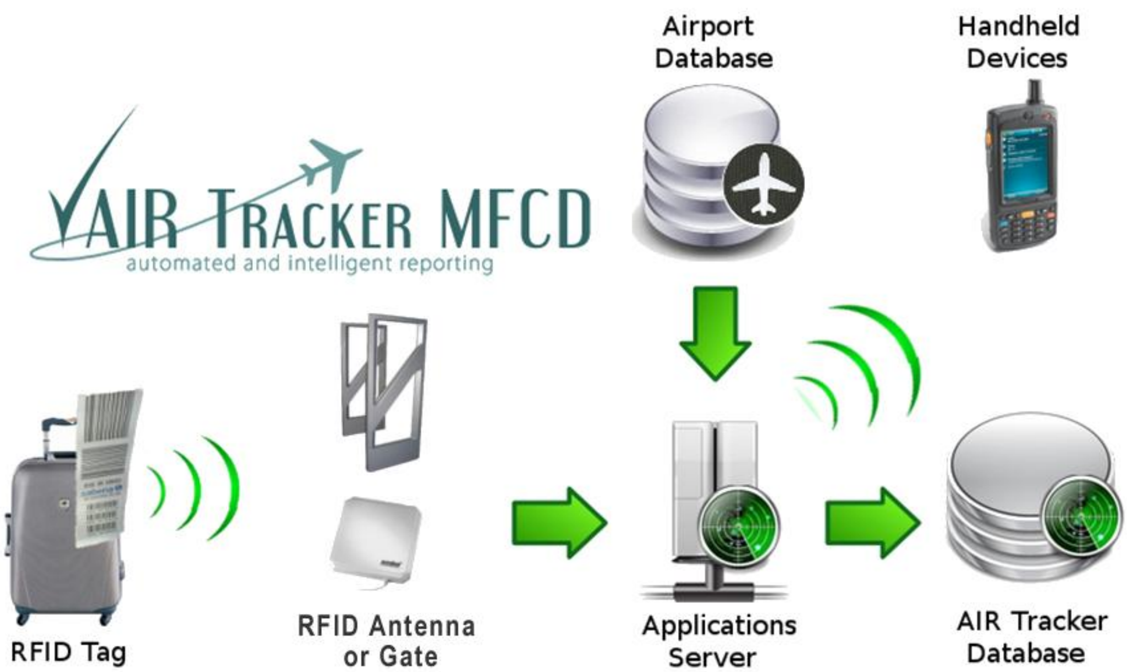

Figure 3. AIR Tracker Major Functional Components Diagram

All of the major hardware components are listed and detailed below. For clarification, the AIR Tracker application server contains the entire AIR Tracker application which will be used for both, gathering bag information and reporting it. It is also worth it to note that there are two different kinds of RFID readers. The first is the standard readers inside the airport that scans bags along the GRP, and the second are the custom-made readers with onboard memory that will reside in the cart.

Hardware

- Application server - The physical storage and computational hardware on which the AIR Tracker application is deployed
- Database server - The physical storage of an AIR Tracker database
- RFID tags (passive) - The unique tags applied to every passenger's luggage within a single airport

- RFID reader/scanner - The devices used to read RFID tags, relay alerts, and receive baggage destination information. The RFID scanners on the cart contain a Wi-Fi module to send information wirelessly.
- Wireless routers - A means of wireless communication between the Application Server and the RFID readers

Most of the software will be used in conjunction with the AIR Tracker server. The AIR Tracker application, the RFID reader interface, the AIR Tracker database interface, and the bag algorithms are all associated with the server. The only software not near the server is the RFID firmware that resides in the cart reader's memory. Also, depending on the size of the airport, either MySQL or Oracle will be used in the AIR Tracker database. All of the software used in AIR Tracker is listed below.

Software

- Oracle Database 11g - Stores bag identity information, destination information and keys to foreign databases. Also stores alerts sent from RFID readers.
- MySQL - Same function as Oracle Database
- AIR Tracker application - Also a back-end for data transfer to databases and RFID readers.
- AIR Tracker to airport application interface - Required to retrieve baggage destination information
- AIR Tracker application to RFID readers interface - Required to relay and update the list of bags associated with a particular cart. Also required to send alerts.
- AIR Tracker application to AIR Tracker database interface - Required to insert, delete, and update baggage information and alerts
- RFID firmware - Maintains that each bag on a particular cart is accounted for and routed correctly. Alerts the baggage handler and the AIR Tracker application of baggage mishandling.

- Bag routing algorithms - Algorithms that determines if a bag is off the correct route in the GRP and on the cart

2.3 Target Market/Customer Base

AIR Tracker is customized and built for the airport. Airports of all sizes can benefit from AIR Tracker, but the primary targets are airline hubs. Hubs are where many transfers take place and consequently where many bags are mishandled. The perfect target would be hubs with high rates of mishandled bags. Although airlines own the baggage handlers and are responsible for any bag-related problems, the airport is still the customer because the system will be implemented at specific locations within the airport. The airport will have to convince airlines that they need to get a new baggage handling quality assurance system such as AIR Tracker in place. Indirectly, airlines may pay for the system, but the check for the Group will come from the airport.

Airports and airlines are not the only ones to benefit from AIR Tracker either. Baggage handlers, passengers, administrators all receive upgrades over the current way things work. Baggage handlers don't need to search for bags with their eyes anymore because bags are alerted in real time providing the bag's location. Administrators will be equipped with an easy-to-use GUI, which takes less clicks to do more work. Lastly passengers, the ultimate customer, benefit the most due to the higher standards in baggage handling.

3 Product Prototype Description

The prototype of the AIR Tracker is designed to demonstrate the feasibility of the product using very limited resources. The prototype should serve as a scale of what the real world product (RWP) should be able to do. In this case, the prototype will be greatly scaled down to a simulation ran on a computer which will run an application with access to two databases. That is all that is needed to show routing algorithms, alert generation and reports. The other portion of the prototype is to demonstrate how ID readers can be used on a cart to gather information on-the-fly. The PC and the cart reader together are enough to show the power of AIR Tracker.

3.1 Prototype Functional Goals and Objectives

First and foremost, the prototype's main goals are to show the ease of use, to demonstrate that the product is feasible, and be able to scale to the real world product. As seen in Figure 4, the real world product and prototype have many differences in hardware, which will directly reflect the prototypes overall scalability. The main differences are that the prototype will be using RuBee instead of RFID and virtual databases instead of separate servers.

Features	Real-World Product	Prototype
tracking device	RFID tag on bag	RuBee tag and simulated tags
scanners	RFID scanners	RuBee scanner
alert mechanism	handheld assistant	simulated alert
airport database	airport database on airport database server	simulated MySQL database on laptop
product database	AIR Tracker database on database server	simulated MySQL database on laptop

Figure 4. RWP and Prototype Comparison

Although it seems quite different from the RWP, the prototype should still be able to complete a myriad of objectives. The objectives will be completed in the order of the process flow itself, starting with the RFID scanner and ending with reports. First, the scanner has to be able to grab data from a tag and send it to the simulated AIR Tracker application. Then, a simulation of the GRP needs to be created, which will have to display bags going through normally and off track which then would be alerted. The data from the GRP simulation will have to be sent to the virtual AIR Tracker database, which then can be queried to create a report.

3.2 Prototype Architecture (Hardware/Software)

Extremely similar to the RWP's MFCD, the prototype MFCD only replaces servers and the handheld device with virtual machines. Again, the process flow begins with a RuBee tag being read by a RuBee reader. The data is inputted into the simulated application server, which is constantly tracking bags in a simulated GRP. Another source for data can be coming from a simulated airport application which will act as the check-in counter. The data from the GRP simulation will be stored in the virtual AIR Tracker database. Finally the simulated application server may create reports by querying the database.

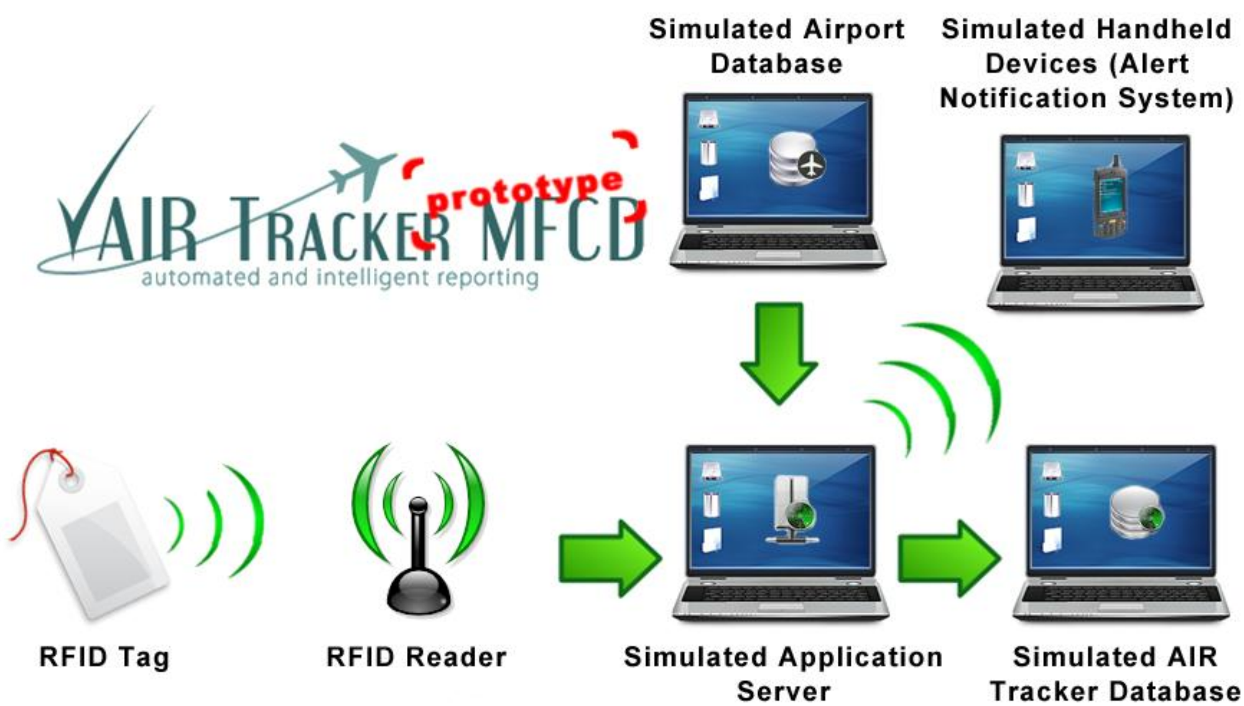

Figure 5. Prototype Major Functional Components Diagram

All of the major hardware components for the AIR Tracker prototype are listed and detailed below. RuBee Tags will be very helpful when prototyping the cart reader because the cart needs to read tags going on and off a cart, but is not needed. RFID tags may be used to demonstrate a more realistic prototype because it uses passive scanning. The simulated server

Hardware

- RuBee Tags - The unique tags applied to model luggage
- RuBee Reader - The device which inputs data from the RuBee Tags and sends transmissions out
- Simulated Application server - The physical storage and computational hardware on which the AIR Tracker application is deployed
- Simulated Virtual database server - The physical storage of an AIR Tracker database.
- Simulated mishandled baggage alert system - Simulated system with graphical notification of mishandled luggage

Similar to the RWP for small airports, MySQL will be used for the database. The simulated AIR Tracker application will contain almost everything for the prototype. This includes a simulation of the airport with routing of bags and generation of real-time alerts. The simulation will also allow the user to create reports and modify factors of the simulation.

Software

- MySQL - Stores bag identity information, destination information and keys to foreign databases. Also stores alerts sent from RFID readers.
- Simulated AIR Tracker application - Front-end for baggage data input and data manipulation. Also a back-end for data transfer to databases and RFID readers. The scaled version of this software wrapper uses simulated interfaces, which are non-disparate to the AIR Tracker application's source code.
- Simulated AIR Tracker to airport database - Required to retrieve baggage destination information
- Simulated AIR Tracker to RFID readers - Required to relay and update the list of bags associated with a particular cart. Also required to send alerts.

- Simulated AIR Tracker to AIR Tracker database - Required to insert, delete, and update baggage information
- Simulated RFID firmware - Maintains that each bag on a particular cart is accounted for and routed correctly

3.3 Prototype Features and Capabilities

If the prototype is able to create reports and generate alerts about mishandled bags, it proves that the product can and should work. The success of the prototype will increase the chance of successful marketing in the future. If the prototype is not successful, the product may never become a retail product.

During the development of the AIR Tracker prototype, there may be problems before the project has even started that must be looked into. Even further, each risk should have mitigations or even alternatives. The first and most severe possibility is that required hardware is too expensive to prototype AIR Tracker accurately. To mitigate, the prototype will have to succumb to combining cheaper alternatives of the technology with planned simulations to display at least the procedural workings of the product. The second most severe is the risk that information on airport routing or data management may be unavailable or not easily acquired. To alleviate the problem, the Group must perform in-depth research to the limits of what is available and then attempt to accurately simulate the rest. Lastly, the risk that simulated processes may not reflect real world events with complete accuracy is always a concern with all prototypes in general. To mitigate, the Group must perform in-depth real-world analysis of every simulated aspect of the project in conjunction with all related actual processes.

3.4 Prototype Development Challenges

Due to the scale of AIR Tracker, the prototype will be missing or substituting a few features that the RWP has. The prototype will be substituting passive RFID tags for more functional active RuBee tags. The prototype will not be able to demonstrate cluster scanning done on the cart due to budget constraints. Since the prototype will be using virtual machines, wireless communication is trivial. Also, since the airport database will be simulated by the Group, it is assumed (for the prototype) that this is what the airport uses. All of the mitigations were agreed upon by the Group and should not deter the prototype from achieving its goals.

Glossary

Cart: Special containers that carry bags between the gate and the airplane

'Cluster Scan': Scan of multiple objects (in this case, bags) within proximity

Gate: A specified location for boarding or leaving an aircraft in an airport

Green Group (Group): Creators of the AIR Tracker

Ground-level routing process (GRP): Conveyors equipped with junctions and sorting machines automatically route the bags to the gate

Hub: An airport that serves as a central connecting point through which many flights of a particular airline are routed

Graphic User Interface (GUI): An interface for issuing commands to a computer utilizing a pointing device, such as a mouse, that manipulates and activates graphical images on a monitor

Pusher: A machine called a pusher either lets it pass or pushes it onto another conveyor

RFID Reader/Scanner: Scan the labels on the luggage

RuBee: A two way, active wireless protocol that uses Long Wave magnetic signals to send and receive short data packets in a local regional network.

Tag: The RFID tag attached to every bag at an airport

Truck: The automobiles that attach to carts and commute between gates and airplanes

Wi-Fi: The name of a popular wireless networking technology that uses radio waves to provide wireless high-speed Internet and network connections based on the IEEE 802.11 standards

Wireless Access Point (WAP): A device that allows wireless communication devices to connect to a wireless network using Wi-Fi, Bluetooth, or related standards

References

SITA. (2009). Baggage Report 2008. Retrieved October 25, 2008, from SITA Web site: http://www.sita.aero/content/baggage-report-2008

U.S. Department of Transportation. (2008, August). Air Travel Consumer Report. Retrieved September 4, 2008, from U.S. Department of Transportation Web site: http://airconsumer.dot.gov/reports/2008 ... 08atcr.pdf