

The Taxi Management System

FAA Design Competition for Universities

University of Southern California

CSCI 477

Brian Abraham, C.J. Windisch, Donald Allen, Matt Gilewski,
Annie Chi, Fernando Arreola, Julio Villegas, Chin-Kai Chang,
Osman Ahmed

Table of Contents:

Executive Summary	Pg. 3
Problem Statement	Pgs. 4-7
• Current Solutions	Pg. 6
Literature Review	Pgs. 8-9
Problem Solving Approach	Pgs. 10-14
• History	Pg. 10
• Lessons Learned	Pg. 12
• Roles, Timeline, User Interaction	Pg. 13
Technical Design	Pgs. 14-28
• Requirements	Pg. 15
• Use Cases	Pg. 16
○ Login and Registration	Pg. 16
○ Real Time Usage	Pg. 16
○ Data Analysis	Pg. 18
○ Training	Pg. 18
• Architecture	Pg. 19
○ ATC Graphical Airport Model	Pg. 19
○ Multi-Directional Runway Lighting	Pg. 20
○ Multi-Variable Taxi Path Database	Pg. 20
○ Automated Radio Instruction	Pg. 20
• Component Design	Pg. 21
○ ATC Graphical Airport Model	Pg. 21
○ Multi-Directional Runway Lighting	Pg. 22
○ Multi-Variable Taxi Path Database	Pg. 22
○ Automated Radio Instruction	Pg. 23
• Technical Aspects of Solution	Pg. 24
○ Updating Flight Locations	Pg. 25
○ ATC Drawing a Path	Pg. 25
○ Path Suggestion	Pg. 26
Safety Risk Assessment	Pgs. 29-33
Interactions with Airport Operators and Industry Experts	Pgs. 34-37
Projected Impact	Pgs. 38-40
Appendix A	Pg. 41
Appendix B	Pg. 42
Appendix C	Pg. 43
Appendix D	Pg. 44
Appendix E	Pgs. 45-57
Appendix F	Pgs. 58-62
Appendix G	Pgs. 63-71

The Taxi Management System project is the work of the computer science department of the University of Southern California's capstone design class, CSCI 477 (Design of Large Software Systems). This one-semester class was held in the Spring 2008 semester.

A runway incursion is when an airplane, vehicle, person or object on the ground creates a collision hazard with an airplane that is taking off or landing. Runway incursions are an increasing problem as there have been 410 occurrences already in 2008 as oppose to the 370 occurrences in the entire 2007 year. In this project we are reducing runway incursions by focusing on one of the major causes of incursions, being pilot error.

Our goals are to:

- Decrease the negative effects caused by poor visibility
- Assist pilots who are unfamiliar with the airport
- Improve the communication between pilots and air traffic controllers
- Assist the air traffic controller in making route planning decisions

The Taxi Management System (TMS) has four major components:

- **ATC Graphical Airport Model:** A touch screen panel with the layout of the runways and taxiways that acts as a user interface for the ATC. Incoming planes and outgoing planes will appear on the screen where the ATC can select them. Once a plane is selected, the ATC will draw a taxi path for the plane which will be communicated to the pilot via audio and visual feedback. The TMS will also suggest possible routes for the ATC to either use or disregard.
- **Multi-Directional Runway Lighting:** Our runway lighting system is comprised of arrays of lights in the shape of arrows embedded in the taxiways and runways. These arrows are located on the taxiways and runways just before every intersection. There are corresponding arrows for every direction a plane could proceed to from their current position at the intersection. The arrow of lights will guide the pilot from their initial position, a runway or gate, to their destination.
- **Multi-Variable Taxi Path Database:** Our database will collect real world data during every takeoff and landing. This data will be analyzed every 6 months to determine suggested routes to the ATC for each plane.
- **Audio Feedback System:** Our text to speech server will convert the path drawn by the ATC into an audio description of the path for the pilot and ATC to hear.

The Taxi Management System will be deployed in two phases:

- **Phase 1:** The first 6 months of operation will focus on the ATC drawing routes and the database collecting data from the routes for analysis. Path suggestion will not be possible in this phase.
- **Phase 2:** The ATC will have the option of selecting a route from suggested paths generated by our system or drawing their own path like in Phase 1. These paths are generated for each airport every 6 months by analyzing path data from the previous 6 months.

1.0 Problem Statement

Mistakes involving aircrafts carry the potential for massive loss of life. Although catastrophic events in this domain are very rare, and air-travel is considered among the safest modes of transportation, the profound value of the human lives at stake continues to make safety improvements worthwhile. Some of the deadliest and most common mistakes occur on the ground in the form of runway incursions.

A runway incursion is defined as:

Any occurrence at an aerodrome involving the incorrect presence of an aircraft, vehicle, or person on the protected area of a surface designated for the landing and take off of aircraft [1].

Historically, there have been numerous runway incursions that have spanned the spectrum from no casualties to hundreds of casualties. Examples include:

- Los Angeles International Airport – March 27, 2008
 - The pilot of an American Eagle flight was given instructions to stop before entering a runway occupied by a private jet. The pilot read back the instructions, showing his acknowledgement, but failed to stop before entering the runway. Since the private jet had not received takeoff clearance, he was stopped at the beginning of the runway and was not hit by the American Eagle flight [10].
- Chicago O'Hare International Airport – April 1, 1999
 - An Air China aircraft had just landed on a runway and was taxiing. A Korean Air aircraft was instructed to taxi into position and hold. The Air Traffic Controller directed the Air China aircraft on a taxi path and cleared the Korean Air aircraft for takeoff. The Air China aircraft diverged from its instructed route and began to

taxi on the runway on which the Korean Air flight was taking off. The Korean Air pilot noticed this and lifted off early, missing the Air China aircraft by 80 feet [6].

- Tenerife North Airport – March 27, 1977
 - A KLM flight had just refueled and was anxiously awaiting takeoff at the beginning of a runway. Simultaneously, a PanAm flight was taxiing on the same runway towards an exit ramp. The Air Traffic Controller wanted the PanAm flight to take the exit C3 (“Charlie Three”), and told the pilot to “take the third exit.” Since the pilot had already passed the exit C1, he assumed that the instruction was to take the C4 exit. The KLM pilot meanwhile, mistook a hold short instruction as clearance for takeoff. After seeing the PanAm flight still on the runway, he tried to pull up, but was unable to. As a result, the KLM aircraft collided with the PanAm aircraft, resulting in casualties of over 500 [48].

These instances have reminded us that not only are runway incursions a serious problem, but they are also a growing problem. In the first half of the fiscal year 2008, the FAA logged 410 runway incursions, while the year long total in fiscal 2007 was 370 [49].

There are many factors, both known and unknown, that raise the risk of runway incursion. This shows that the problem cannot be adequately addressed with a single one-dimensional solution. Thus, a truly effective solution will take examine and tackle several of the factors associated with runway incursions. The three general categories we have chosen to address are as follows:

- Pilot–ATC miscommunication
- Inadequate pilot-side situational awareness due to:

- Poor weather conditions
- Lack of familiarity with the destination airport
- Taxi-route planning errors

1.1: Current Solutions

LAX Runway Status Lights

In February 2008, the Los Angeles International Airport announced its plan to install runway status lights. The lights are embedded in the runway, and flash red when a pilot should hold, and green when it is safe to go. A radar system detects which runways are being used and toggles the color based on the airplane's position [50]. While this system is an intuitive way to reduce runway incursion, it was also the cause of one during its initial rollout. At Dallas Airport, a pilot was instructed to hold short and red lights were illuminated. Due to a glitch in the system, the lights went off and the pilot proceeded to enter the runway, on which another aircraft was taking off [51].

ASDE-3

The Airport Surface Detection Equipment is comprised of monitor displaying airport traffic and is intended for the ATC to use. It serves as a visual aid for tower personnel. A primary function of the system is to assist ATC during situations where visibility is low due to environmental reasons. This system can be very beneficial, however, it lacks the ability to detect or alert the ATC of any possible RI. In addition, we found that the system can better utilize the monitor to provide a more effective visual aid.

Airport Movement Area Safety System (AMASS)

The Airport Movement Area Safety System is an improvement of the ASDE-3. In addition to displaying airport traffic, it also predicts when potential RI may occur through the use of radar

data. Radar data is used to calculate the possible RI by taking speed, direction, and the planes' position relative to other traffic. The ATC receives a visual and audible warning if this occurs. This system is much better improvement on the ASDE-3, but its high rate of false warnings diminishes its reliability and makes it be not very practical to use.

ASDE-X

This system is a further improvement on the ASDE system by identifying traffic on its monitor using tags. The tags make it much easier for the ATC to identify the different traffic on the airport. In addition, this system gathers information from various sources including Surface Movement Radar, Automatic Dependent Surveillance-Broadcast, Terminal Automation Systems, and aircraft transponders. Since it relies on various sources for data, it is much more accurate than AMASS and less prone to false alarms. A major factor keeping this system from being very effective is its high cost.

Runway Incursion Prevention System (RIPS)

The Runway Incursion Prevention Systems is NASA's approach to solving the RI problem. It focuses on increasing the situational awareness of pilots on the runway. It does so through the use of various cockpit displays that allow the pilot to have a similar view of the airport as the ATC. Similar to other promising systems, this solution comes at a high cost.

Autonomous Runway Incursion Prevention System (ARIPS)

The Autonomous Runway Incursion Prevention System focuses on building an affordable system. It uses ultraviolet lights as sensors to detect whenever traffic is in the path or area of a plane. An alert is given to the pilot whenever ground traffic is operating within the planes planned route. The approach of this system on reducing costs increases its likelihood of having long term success.

2. Literature Review

The RI situation is a difficult problem to tackle due to the complexity of the interaction between the varying components involved in an airport. Our initial inquiry into the problem began with a review of published airport standard operating procedures (SOP). We then analyzed several data and statistics outlining RI and compared that to the SOP's. This comparison gave us the information we needed to understand the key factors leading to RI and provided the context in which they occur. As a final step before designing our solution, several existing technologies were taken into account with the intent of making our solution practical to integrate. Our final design was largely influenced by the information found in airport SOP's, literature on RI, and both current and proposed RI prevention technology.

A domain analyses was performed using the Los Angeles International Airport (LAX) SOP's [22] as our primary source of information. This entailed a thorough review of the responsibilities of airport personnel and the method in which their roles interact. Following a review of SOP literature was an onsite visit to the LAX control tower. We used the information learned from those experiences to narrow our focus to the interaction between the air traffic controller and the aircraft pilot. FAA air traffic control SOP's [23] and a review of airport communications phraseology [24] provided us with valuable information on these interactions. This information provided evidence that there are elements of this interaction that are better suited for an automated system.

Various resources were used to gather information on RI. The data gathered from the FAA runway safety report [1] was central to our understanding of RI. Other sources included literature on RI prevention by Aircraft Owner and Pilots Association, the International Civil Aviation Organization, and the National Transportation Safety Board [2]-[8]. News articles,

academic publications, published airport personnel interviews, and various media provided a large amount of information describing in full detail RI incidents [9]-[21]. Furthermore, a case study on the RI at Tenerife in 1977 provided insights on the technical, environmental, and human factors that can lead to deviating from the SOP's. This study included a presentation from a subject matter expert on RI human factors, University of Southern California Professor Dr. Najmedin Meshkati, and a review of a published report on the incident [48]. Our findings on all RI data impacted our design by highlighting the need to address issues of miscommunication, lack of effective use of technology, and human factors to prevent RI. Specifically, preventing incorrect pilot and ATC actions due to language barriers, work fatigue, and unreliable radio frequencies were some of our primary concerns. An emphasis on our design was also given to preventing pilot error due to unfamiliarity with an airport's layout.

Part of our inquiry into the problem involved examining existing RI prevention technologies. We began by reviewing information on the most common technologies used by ATC's and pilots, such as ground positioning devices, radar, transponders, and automatic terminal information service [38]-[40] to determine where our design would fit in. A review of literature describing current and proposed RI prevention systems [26]-[35] was also performed with an assessment of both possible improvements and strengths of the systems. As a result, we narrowed our design to rely on a more specific set of technologies followed by a review on literature detailing this technology. This included runway status lights [36] [37], voice generation [42] [43], and the use of visual aids to automate time sensitive tasks. Evidence from the data we gathered has demonstrated these technologies to be very promising in solving the RI problem.

3. Problem Solving Approach

3.1 Problem Solving Approach: History

Billboards. Electronic billboards just off the runway to show the pilot where to go. Our class slaughtered us for presenting this brilliant idea – “Aren’t there restrictions on what can be sticking up in the runway areas?” Yes.

That’s how the Taxiway Management System got started – after a phase of researching the core causes of incursion and choosing to focus on the most common cause, pilot error, we jumped into finding solutions. After the disastrous billboard idea, we regrouped and discussed a more realistic solution. No idea went unheard. Everything from cost to prototyping feasibility got thoroughly debated. Eventually we came up with a way to expand pilots’ situational awareness that didn’t involve objects that planes might crash into – directional runway lights.

From our research, we knew about existing systems and LAX’s standard operating procedures for pilot-controller interactions. So we worked to get a high level architecture of where our system would fit in place with other systems and how the controllers and pilots would interact and benefit from the Taxiway Management System.

Weekly team meetings coupled with on going research meant a steady flow of fresh ideas to refine and improve our system. Debates on features and their cost kept our system realistic and constantly checked our individual assumptions. Every few weeks we presented our progress to our computer science class, enabling us to get outside feedback to perfect our design.

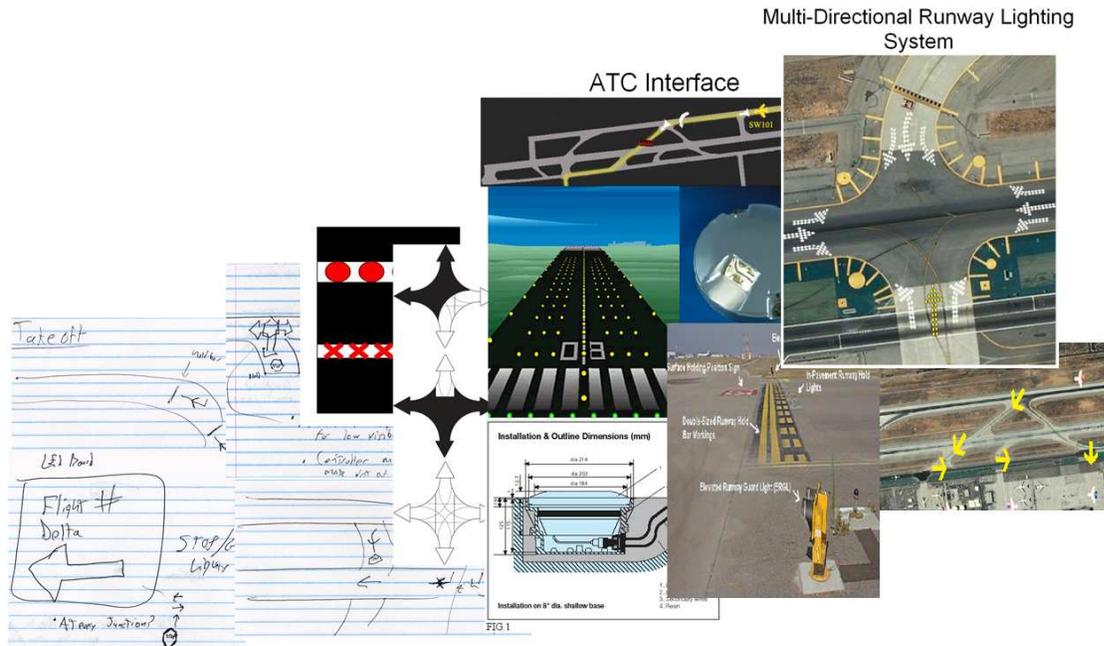
At meetings we discussed requirements for what exactly our system should do, including deliberations on the associated costs and affects on stakeholders. After agreeing on a feature, we would chalkboard ideas on what that feature should look like to the ATC or pilot and how they

would interact with it. We would also figure out how that feature fit into our existing system architecture and whether it was worthwhile to prototype. That decision was made on feasibility of rapid implementation and whether the feature was unique enough to our system that a prototype could benefit the design and make it easier to communicate our solution. For example we decided to prototype a real world view of the runway showing how the lights would light up, but not the user login system because it's so generic.

In conjunction with our live meetings, we did research and design work individually. Our individual research, brainstorming, sketches, and Photoshop mock-ups were posted at an online wiki called brainkeeper.net. We had a wiki just for our team to communicate and one for the entire class to share research, incursions in the news, and ideas. When changes were made to one of the wikis, an e-mail was automatically generated to alert our team that someone added something, so we could all stay on the same page come time for meetings.

Although chalk boarding, sketching, brainstorming, and discussing were all great ways to flesh out requirements and design, nothing quite showed us places for improvement in our system like building the prototype. Being able to interact with a live version of our system allowed us to discover small features that made the Taxiway Management System much easier to use. The best example is highlighting the starting and ending points of plane paths. Initially, the ATC would just draw the route where he wanted a pilot to taxi on a map of LAX, however, we found it hard to know where to start drawing (the start of the runway or the start of the taxiway?). A 10-minute implementation by Fernando highlighted the possible starting and ending points which made it clear where to start and stop drawing. Continuously improving our prototype by actually using it let us see the system from the perspective of the people who'd actually be using

it, which helped us make the Taxiway Management System faster to use and easier to learn.



3.2 Problem Solving Approach: Lessons Learned

Stumbling Points

We always had great meetings, but we were not always good at documenting what went on at them, and with 7 days between meetings, we often forgot some of our decisions – as a result, it took us 3 weeks to design how the runway lights would look. However, by the end of our time, we were much better at documenting decisions on our wiki (or at least in e-mail communications); this often meant diagramming in Visio and Photoshop, or scanning in notes – something that helped us progress much faster and with less backtracking.

Successes

Our design meetings were always full of ideas because of our inclusive atmosphere and the team’s willingness to participate beyond their said role descriptions. Considering every idea

no matter how obvious it seemed kept our approach simple and feasible, while allowing everyone to contribute uninhibitedly. Rapid prototyping helped us make our system real to our audiences and allowed us to improve its user-friendliness.

Visual communication of our system in our presentation via diagrams and versions of the prototype forced us to be concrete and more detailed in our design, moving us beyond vague ideas and into the details of making the Taxiway Management System happen.

3.3 Problem Solving Approach: Roles, Timeline, User Interaction

When our team was formed by Professor Wilczynski in January, we each choose primary and secondary roles. Although, we had specific roles, responsibilities were organically defined and each team member contributed beyond their role. Our design progressed in 2 major phases: 1) Research and Core Solution Definition, 2) Design Iterations.

In phase 1, we were still getting to know each other as a team, and we worked mostly on our own to come up to speed on what runway incursion actually was. As a class, we discussed the core causes of incursion and watched film on the famous incursion at Tenerife, while we began brainstorming solutions as a team. Early on we visited LAX to see how controllers worked first hand. This completely changed our approach because we soon realized many of the problems that happened at Tenerife have been solved or don't apply to LAX, which had its own set of problems. Also, the controllers explained all the work they do to prevent incursions and how pilot error was overwhelmingly the #1 cause, often because pilots fly around the world and might only see LAX once a year – pilot unfamiliarity.

Before going to LAX our researchers analyzed LAX's Standard Operating Procedures for Controllers and converted it to non-controller lingo. Once we visited LAX, Osman and Julio

posted research about incursion and existing technologies on our team wiki. We discussed the runway incursion problem and broke it down into its core causes. From there, we decided to limit our scope and focus on the problem of pilot error, and focus our solution on expanding pilots' situational awareness. During this solution definition phase we passed around ideas (like the billboards) and decided on directional runway lights as our driving component. We also created a high level architecture of how our system fit in with the existing systems and processes.

In transitioning to phase 2, we made the key decision to go with the directional runway lights idea and began designing how the Taxiway Management System would look. In phase 2 we designed iteratively, meaning we brainstormed features, discussed them, and decided which ones to keep. Then, we'd design how they'd fit into our system architecture, how they'd look to the users, and how we'd make it work in code. Since we were focusing on benefiting the users, we spent a lot of time drawing and debating design for the user interface. What shape would the runway lights be? What color? How would the ATC interact with the system in a way that minimized intrusiveness? Would this change the SOP's? The interface drove our system and also gave us ideas for features, such as the path suggestion user interface you will see later. Throughout phase 2 we constantly reviewed our previous design work, questioned it, and refined it. To keep our design in check, we used an abbreviated version Barry Boehm's MBASE methodology for documenting our goals, requirements, and design. As the weeks progressed we became faster at designing and began making decisions more quickly as to which ideas would become features and which would be deferred. Decisions also became quicker because once we had a prototype, the cost of changing it limited what new features we could add. By the end of our time, we successfully created a working prototype and full design of our vision: the Taxiway Management System.

4. Technical Design

Runway incursion has many causes and no silver bullet solution. Our system, The Taxi Management System, will focus on a subset of these causes to reduce the number of runway incursions at any given airport.

4.1 Technical Design: Requirements

The Taxi Management System goals are to:

1. **Reduce the negative effects of poor visibility:** Bad weather as well as low lighting can cause poor visibility for both pilots and air traffic controllers. We believe that by decreasing the negative effects of poor visibility we will be able to reduce human errors that lead to runway incursions significantly.
2. **Assist pilots who are unfamiliar with an airport layout:** Pilot unfamiliarity with an airport is a contributing factor to runway incursions because of the pilot's possible misunderstanding of directions or specific routes. By assisting pilot's who may be unfamiliar with a particular airport we believe that we can increase situational awareness, which will lead to fewer runway incursions.
3. **Improve the way air traffic controllers communicate with pilots:** Air traffic control to pilot communication is done primarily over radio which can have static or interference. These issues can lead to misunderstanding of directions or instructions which can lead to a possible runway incursion. By changing or improving the way controllers and pilots communicate we believe that we can reduce that number of runway incursions that occur due to misinterpreted orders.

4. **Assist the air traffic controllers in making wise taxi route planning decisions:**

Taxi route planning is one of the main jobs of an air traffic controller. During high traffic times it may be difficult for the controller to keep track of all of the paths he has set for planes and may tell two planes to travel on intersecting paths which could lead to a runway incursion. By assisting the air traffic controllers and making their path decision easier we believe that they will reduce runway incursions due to poor taxi path planning.

4.2 Technical Design: Use Cases

4.2.1 Use Cases: Login and Registration

- We will use the best available login and registration software.
- Typical users would be air traffic controllers, control supervisors, systems administrators, and data analysts.
- Further details are unavailable at this time.

4.2.2 Use Cases: Real Time Usage

ATC Draws Path (Arriving or Departing):

- The air traffic controller will select a plane using our system.
- They will determine the planes taxi path by physically drawing it out on the real-time airport map.
- Once the path is determined, it will be sent to a central server that will convert the path data into text instructions.

- Once converted to text, the path data is sent to our text to speech server (TTS) for conversion into an audio file. Simultaneously, the server will be collecting real world data about every arrival and departure for logging.
- When the TTS has finished creating the audio path file it will return it to the central server.
- The central server will broadcast the audio to both the controller and the corresponding pilot. Corresponding runway lights will be illuminated at this time.

Path Suggestion (Arriving or Departing):

- Initially an air traffic controller will select a plane using our system.
- The controller requests one of the path suggestions from our database.
- Suggested paths will be determined and returned for the controller to view and chose between.
- Once the path is determined, it will be sent to a central server that will convert the path data into text instructions.
- Once converted to text, the path data is sent to our text to speech server (TTS) for conversion into an audio file. Simultaneously, the server will be collecting real world data about every arrival and departure for logging.
- When the TTS has finished creating the audio path file it will return it to the central server.
- The central server will broadcast the audio to both the controller and the corresponding pilot. Corresponding runway lights will be illuminated at this time.

Missed Taxiway (Arriving or Departing):

- Plane deviates from their specified taxi path.

- Our System alerts the controller that the specific plane has deviated from its course.
- The central server will instruct the pilot to hold at the next hold bar.
- The air traffic controller will determine a new route for the plane (see previous Use Cases).

Emergency:

- An air traffic control supervisor disables the system via an external emergency switch.
- All controllers are instructed through our system to give all further instructions to pilots through the radio.

Override:

- An air traffic control supervisor can close certain runways or taxiways by logging into our system.
- Controllers are no longer able to route planes through those closed taxiways or runways.

4.2.3 Use Cases: Data Analysis

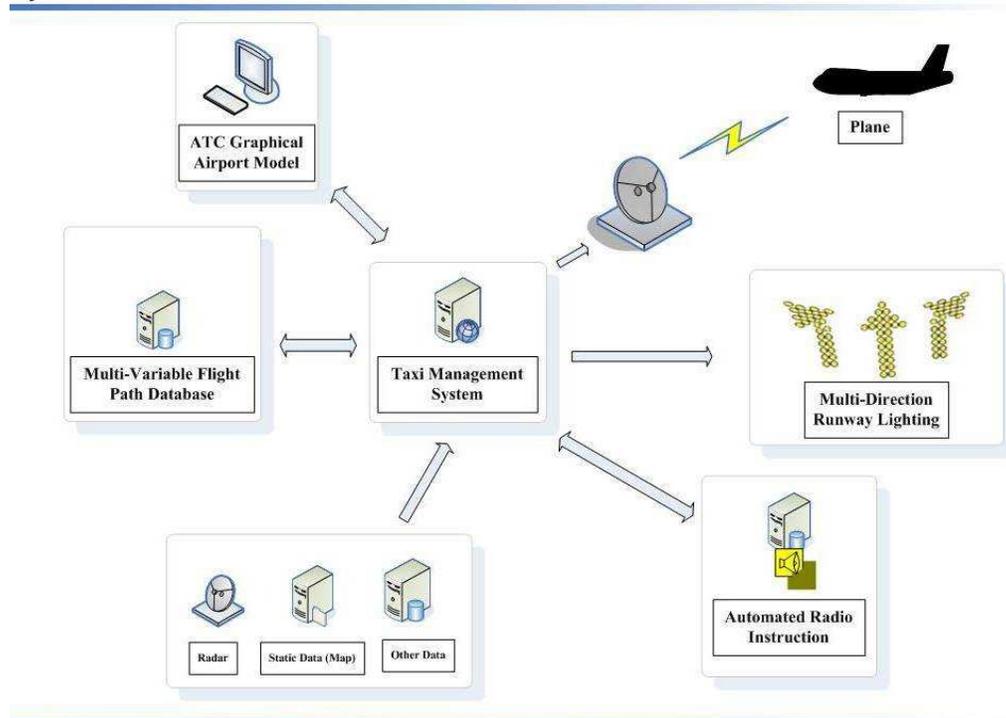
Real time data will be collect during every arrival and departure. To analyze this data, third-party data analyzers will be brought in to make sense of the data and create useful information. More information is unavailable at this time.

4.2.4 Use Cases: Training

- System training will be necessary; however details are unavailable at this time.

4.3 Technical Design: Architecture

System Architecture



4.3.1 Architecture: ATC Graphical Airport Model

The ATC Graphical Airport Model is the interface for the air traffic controller to view and manipulate operations of the taxiways and runways. The interface includes a real-time updated map of the taxiways and runways, lists of all active arriving and departing flights, and a list of flights that will be entering the system momentarily. All interactions with the ATC Graphical Airport Model will be done with a touch screen panel. The system will be used for keeping track of all taxiing flights and determining their taxiing path. When a plane becomes active and part of the system it will be added to either the arrivals or departures list, whichever is appropriate, and a plane icon will appear in a corresponding position on the taxiway map. The paths of each plane will be determined either by the air traffic controller drawing a path on the

map or by the path suggestion system. Once the pathway is determined the system will communicate it to the pilot both visually (4.2.2) and verbally (4.2.4).

4.3.2 Architecture: Multi-Directional Runway Lighting

The Multi-Directional Runway Lighting system will be comprised of arrays of white lights placed in the shape of arrows. These lights will be physically embedded into all of the runways and taxiways at an airport. The arrows will occur just prior to every entrance at every intersection throughout the taxiways. Each entrance can be fitted with as many or few arrows positioned in any number of directions, as the complexity of the intersection requires.

4.3.3 Architecture: Multi-Variable Taxi Path Database

The Multi-Variable Taxi Path Database will be used as a means for collecting and organizing variables surrounding every planes taxi to takeoff or from landing. The data is analyzed to determine suggested routes for the ATC Graphical Airport Model.

4.3.4 Architecture: Automated Radio Instruction

The Automated Radio Instruction is an audio representation of the taxi path the air traffic controller has drawn for a specific plane. The audio is played back for both the air traffic controller and the corresponding pilot.

The Taxi Management System's strength does not come from all of its subsystems independently, but from the combination of all four. The ATC Graphical Airport Model (4.2.1), Multi-Directional Runway Lighting (4.2.2), Multi-Variable Taxi Path Database (4.2.3), and

Audio Feedback System (4.2.4) all must work together to make the Taxi Management System effective.

4.4 Technical Design: Component Design

4.4.1 Component Design: ATC Graphical Airport Model

Initially the ATC Graphical Airport Model will have a list of inactive planes, which will soon be added to the system, at the top of its screen. Once these planes enter the system the air traffic controller will be able to select a plane by touching either the plane icon or its description in the arrivals or departures list. When a plane is selected the air traffic controller can either draw a path from one of the, highlighted, selectable starting positions and continuing to a runway or the gate area, whichever is applicable, or they can choose to use one of the path suggestion types. Path suggestion can be determined a number of ways. These include: shortest distance (from the beginning of the path to the destination), a saved favorite route of that particular controller, or from our Multi-Variable Taxi Path Database (4.2.3). Once a path is selected the ATC Graphical Airport Model will communicate the path to the pilot. At any point during the path determination process the controller can clear any planes path and create a new one from scratch. The controller can also select any plane at any time to view their determined path.

If a pilot deviates from their pre-determined route, the ATC Graphical Airport Model will alert the controller of the deviation and instruct the pilot of the deviated plane to stop at the closest hold bar. Once acknowledged the controller can re-select the deviated plane, clear their old route, and determine a new one from their current position.

In the event of an emergency at the airport, an air traffic control supervisor can activate emergency mode (via an external switch). While in emergency mode all Taxi Management

System functionality is suspended and the controllers are all told, through the ATC Graphical Airport Model, to give all taxi path instructions to pilots via the radio.

In the event of a downed taxiway or runway, an air traffic control supervisor can override the system (via an override button) and disable any taxiway or runway as needed. Controllers will be unable to route planes through the downed taxiway or runway.

The typical users for this system would be an air traffic controller, control supervisor, and a system administrator. The controller would use the system as stated above. They do not have access to emergency or override modes. Control supervisors could use the system as stated above, but they have extra capabilities such as activating emergency or override modes. The system administrator would perform maintenance on the system and would not have the full functionality of the system available.

ATC Graphical Airport Model: Draw Taxi Path

The screenshot shows the 'Taxi Management System' interface. At the top, there is a blue header with the title 'Taxi Management System'. Below the header is a map of an airport taxiway system. A yellow line is drawn on the map, representing a taxi path. To the right of the map, there are two buttons: 'Confirm Route' and 'Clear Route'. Below the map, there are two tables: 'Departures' and 'Arrivals'.

Departures		
Airline	FlightNumber	Sequence
MGA	211	1

Arrivals		
Airline	FlightNumber	Sequence
AAL	777	1
NWA	333	2
SWA	145	3
ASA	899	4

ATC Graphical Airport Model: Suggest Taxi Path

The screenshot shows the 'Taxi Management System' interface. At the top, there is a blue header with the title 'Taxi Management System'. Below the header is a map of an airport taxiway system. Multiple colored lines (yellow, purple, red, blue) are drawn on the map, representing suggested taxi paths. To the right of the map, there are four buttons: 'Draw Route', 'Route 1', 'Route 2', and 'Route 3'. Below the map, there are two tables: 'Departures' and 'Arrivals'.

Departures		
Airline	FlightNumber	Sequence
MGA	211	1

Arrivals		
Airline	FlightNumber	Sequence
AAL	777	1
NWA	333	2
SWA	145	3
ASA	899	4

4.4.2 Component Design: Multi-Directional Runway Lighting

Once an air traffic controller has use the ATC Graphical Airport Model (4.2.1) to designate a path for an arriving or departing plane and the specified plane has begun their taxi the arrows will begin to illuminate. The arrows that will illuminate correspond with the direction the plane is traveling into each intersection and which direction they are supposed to proceed from out of the intersection. Once a plane has taxied through an intersection, effectively driving over one of the arrows, that specific arrow will turn off. As a plane progresses through the taxiways if they are required to stop at a hold bar at any point, only the arrows up to that hold bar will be illuminated. The continuing arrows to the planes destination will only illuminate once the plane has been cleared from their hold bar.



4.4.3 Component Design: Multi-Variable Taxi Path Database

During every taxi to takeoff or landing our system will gather real world data from a number of sources such as radar, weather, runway conditions, etc. This data is sent to and stored in our Multi-Variable Taxi Path Database to be analyzed. Third-party data analysts will be used to analyze all collected data. After analysis the data can be used by the ATC Graphical Airport Model to determine suggested routes based on current conditions.

Sample Data:

UserId	Date	Time	Airline	Flight#	Path	Emergency?	RunwayOrientation
5678	2008-04-05	13:11	JAL	227	24R-Z-E-E12	No	East-west
5678	2008-04-05	13:37	ASA	477	D10-E-V-24L	No	East-west
5678	2008-04-05	13:42	HAL	103	D7-V-24L	No	East-west
5678	2008-04-05	13:50	DAL	259	24R-AA-E-D9	No	East-west
5678	2008-04-05	13:55	UAL	583	24R-Y-E-D9	No	East-west
5678	2008-04-05	14:02	COA	930	D8-E-V-24L	No	East-west
5678	2008-04-05	14:15	VRD	672	24R-BB-E-E11	No	East-west

Aircraft	Weather-AFC	Wind (knots)	Wind dir	Surface Temp (°C)	SIG/AIRMET
B747-400	MVFR	7	230	15.0	none
B737-800	MVFR	7	230	15.0	none
B767-300	MVFR	7	230	15.0	none
B767-400	MVFR	13	230	16.1	none
B737-300	MVFR	13	230	16.1	none
B767-400	MVFR	13	230	16.1	none
A319	MVFR	13	230	16.1	none

4.4.4 Component Design: Automated Radio Instruction

When an air traffic controller specifies a taxi route for a plane using the ATC Graphical Airport Model (4.2.1) the path sent through a text-to-speech (TTS) converter that creates an audio pathway instruction. This instruction is played back for both the pilot and controller.

4.5 Problem Solving Approach: Technical Aspects of Solution

As mentioned in section 4 the Taxi Management System (TMS) is divided into four different components: the ATC Graphical Airport Model, Multi-Directional Runway Lights, Multi-Variable Taxi Path Database and an Audio Feedback System. These four components are represented internally by four main classes: Airport class, Runway class, Light class and Plane class. Along with these four main classes there are also event handlers that listen to ATC input from the user interface and an input abstraction layer which grabs information from the radar and converts it into data that is usable by the classes mentioned before.

4.5.1 Updating Plane Locations

The radar will feed the location of all the planes that it has information on to the input abstraction layer. The input abstraction layer will parse this information into an array of planes, called `LocationDataInterface`, containing the airline, plane number and the current location of all the planes. The `LocationDataInterface` array will update the location of all the planes in the airport using the `Airport` class. If there is a plane located inside the `LocationDataInterface` array that is not currently in the planes array inside the `Airport` object this means that the plane has just entered the scope of the airport and must be added to the airport object in order to keep track of it.

4.5.2 Technical Aspects: ATC Drawing a Path

Path Drawing and Turning Lights On

The event handlers listen to what is entered from the ATC user interface. This component listens to when the buttons and data table rows are touched, as well as anytime the touch happens on the map display. When the map is touched in order to draw a path the Taxi Management System calculates the latitude and longitude of the location on the screen being touched, using the method mentioned above, and draws a line on the closest runway of the screen, based on the latitude and longitudes of the runway corners. In order to determine which lights should be turned on the event handlers keeps track of the previous runway that touched as well as the current one. If the runway is the first one pressed for the given route then no lights should be turned on. When a new runway is pressed the current runway gets moved into the previous

runway variable and the new runway is saved as the current runway and the state of the light for that direction is changed to ON:

```
prevRunway = currRunway;  
currRunway = pressedRunway;
```

Once the confirm route button is pressed the system will turn on the lights on the runways for that specific plane route. The event handler keeps a reference to the current plane in question. When the confirm route button is touched the event handlers iterates through the current planes route, accessing the state of each light to determine if the light should be switched on or off.



The current taxiway is Z and the previous taxiway is 2 therefore the middle light's state is set to ON and turned on when the route is confirmed.

Database Logging

As the path is confirmed, the system also logs the information entered into a database. The system saves the route that was entered, as well as variables such as the weather, the airline the plane was, date, time, etc. This is done simply by iterating through the current route and noting the runway numbers, by concatenating them in a string, of the runways in that specific route. The weather information is obtained from the radar while the airline is obtained from the Plane object.

Automated Radio Instruction

Aside from logging the string of runway numbers into a database the system also uses it to communicate the route to the pilot as well as reiterate to the ATC after the route is confirmed. We plan on using third party software in order to create this string into a wav file that can be sent to the pilot and played for the ATC. There are many examples of this type of software that we could use such as AT&T's Text-to-Speech Demo. A link to this software is provided below:

<http://www.research.att.com/~ttsweb/tts/demo.php>

Plane Deviation Detection

It is very possible that a plane does not follow the path it is assigned for one reason or another. For this reason the Taxi Management System, after updating the positions of all the planes, checks whether any of the planes are deviating from the path assigned. To do this the input abstraction layer calls the checkDeviation method inside the airport class. This method iterates through all the planes currently in the system obtaining access to the route the plane was assigned to. The Taxi Management System then iterates through all the runways in the route obtaining the polygon objects that is used to represent the space which the runways occupy. In order to verify that the plane is within the polygon the system then checks that the latitude of the current position is between the minimum and maximum latitude of the polygon, and check the same logic with the longitude of the plane position and the polygons. As soon it is found that a polygon contains the position of the plane then the system proceeds to check the next plane. If no polygon is found to contain the current position of the plane then the system will open a message window right above the icon of the respective plane on the ATC User Interface informing the ATC that the plane is not following directions.



Display shows message box stating that the plane is deviating from its assigned route

4.5.3 Technical Aspects: Path Suggestion

The information that is logged in the database will be given to third party real data analysts which will determine the best routes to take using the variables mentioned above.

From the analysis we will develop an XML file containing the information of the best routes. This XML file will include the runway numbers that the route crosses in sequence, as well as the light from those runways that must be turned on and the weather, airline or other conditions in which the route is fitted to be optimal or preferred over other routes. The event handler object listens to two touches on the ATC user interface, which are the starting and ending points of the route. The system then reads in the information from the XML file parsing out all the entries that do not start and end with the given points. The system takes the routes that match and display them on the ATC user interface and the ATC would then be allowed to choose a path by pressing it or can also choose to draw a different path. When a path is selected the information obtained from the XML file for that path, i.e. the runways and lights that must be turned on, are passed to the current plane object and stored in the route attribute of the class.

After a route is selected the ATC would have to confirm the route. The system will then proceed to the same steps as when the ATC draws the path and confirms.

5. Safety Risk Assessment

The FAA promotes a culture of safety. TMS has been implemented with this in mind. In fact, total failure of the system would only result in a "Hazardous" but not "Catastrophic" severity rating. Before the TMS would be deployed at any airport, it would undergo the rigorous Safety Risk Management (SRM) process as described in Advisory Circular 150/5200-37 and the FAA Safety Management System Manual. This process involves identifying the potential hazards of the system, assessing the impact and probability of those hazards occurring, and mitigating the hazards through a reduction in the probability or severity of hazards to acceptable risk levels. (page 8, FAA SMS Manual) As a full SRM is beyond the scope of this document and the time constraints of this competition, we will only address very high level hazards of the TMS. Though many of these hazards are worst-case examples, we include only the hazards that are also credible, per the SRM guidelines. The general mitigations are system redundancy to reduce the chance of a total failure of any single component as well as normal ATC training in the procedures for communicating and guiding pilots without using the system. Again, since a full SRM process is beyond the scope of this project, we did not have a chance to iterate through the risk assessment and mitigation to the point where all risks are at an acceptable level. All severities and likelihoods are from the FAA SMS Manual, pages 42-44. Likelihood definitions are from the "ATC Operational, per Facility" column in Table 4.3.

Affecting Entire System

hazard: power loss

effect: all components would be offline, along with entire airport.

severity: hazardous (large reduction in functional capability and safety margin)

likelihood: extremely remote (once every 10-15 years)

risk: medium

mitigation: Redundant power systems would reduce the likelihood. TMS is designed to be an enhancement to and not a replacement for pre-deployment procedures.

ATC Graphical Airport Model

hazard: slightly inaccurate location data

effect: less than optimal paths, potential for false positive warnings about deviations.

severity: minor (significant increase in ATC workload)

likelihood: probable (once per month)

risk: medium

mitigations: use several data sources to verify location, check location visually

hazard: total component failure (Graphical Airport Model and User Interface offline renders whole system offline)

effect: ATC would need to coordinate lots of traffic without any system aid. Sudden new workload increase makes stress and mistakes likely.

severity: hazardous (large reduction in functional capability and safety margin)

likelihood: remote (once every 2-3 years)

risk: high

mitigations: develop and train in contingency procedures. Train ATCs in procedures for directing pilots without use of the system. Have redundant user interfaces to minimize recovery time. With adequate training, severity would be reduced to major.

Multi-Directional Runway Lights

hazard: lights illuminate incorrectly (and are promptly turned off, per contingency strategy)

effect: pilots become confused and distracted by incorrect lights. If unmitigated, may **cause** incursions.

severity: minor (increase in ATC workload)

likelihood: probable (once per month)

risk: medium

mitigations: automatic or manual cut-off switch; rather than light up to guide planes the wrong way, prevent parts of the system from lighting up at all. ATC can direct pilots as they did before system deployment. Pilots should be trained to contact ATC for instructions if the runway lights go dark.

hazard: total component failure (entire multi-directional runway light system offline)

effect: pilot situational awareness would no longer be enhanced. ATC will need to direct pilots as they did before system deployment.

severity: minor (significant increase in ATC workload)

likelihood: remote (once per year)

risk: low

mitigations: train ATCs in procedures for directing pilots without use of the system.

Multi-Variable Taxi Path Database

hazard: inaccurate or missing data

effect: (if widespread) path data would be less than optimal. Path suggestion quality could slightly decrease.

severity: no safety effect (inconvenience, not noticeable by ATC or pilots)

likelihood: frequent (several times per week)

risk: low

mitigations: When analyzing data, discard data that is not statistically significant. TMS paths are generated from six months of collected data, so even a bad batch of data would at most be a minor annoyance until the next group of paths are deployed.

hazard: Total component failure. Databases inaccessible or offline.

effect: Path data would not be logged. Path suggestion would be unavailable. ATC workload would increase significantly compared to accustomed workload.

severity: minor (slight reduction in ATC capability)

likelihood: remote (once per year)

risk: low

mitigations: redundant databases reduce likelihood of total failure as well as reduce the time to recover the system from such a failure.

Audio Feedback System

hazard: unintelligible or misheard instructions, instructions not understood by pilot

effect: ATC would need to clarify or repeat instructions to pilot.

severity: no safety effect (slight increase in ATC workload)

likelihood: frequent (once per day) at beginning, probable (once per month) as system matures.

risk: low

mitigations: as pilots and ATCs become accustomed to the computer voice, and as voice technology improves, the voice of the system will be understood more readily.

hazard: total component failure

effect: ATC would need to handle all communications with pilots.

severity: minor (significant increase in ATC workload)

likelihood: remote (once per year)

risk: low

mitigations: redundant servers to handle conversion of paths to voice, training of ATC in pre-deployment procedures for voice communication.

6. Interactions with Airport Operators and Industry Experts

Much of our research involved personal interaction with experts and professionals in the field.

This included taking a tour on the control tower at Los Angeles International Airport (LAX) and corresponding with professors and airport officials.

Los Angeles International Airport (LAX) Experts

Sherry Avery
Chief of LAX Control
Tower

Steve Ramirez
Procedures, LAX Tower

Tony DiBernardo
LAX Control Tower

Our class had the opportunity to go to LAX airport and speak with the experts there. At the airport, Sherry Avery and Tony DiBernardo briefed us about LAX operations and gave us a tour of the control tower. From this tour we learned a few very important points that we used as the foundation for our system.

1. Pilot errors are the primary cause of runway incursions.

LAX is very strict about who they hire as controllers. Each controller must have at least 3-5 years of experience and must go through a year long training course before they can actually start working in the control tower.

2. Their current system is effective.

LAX uses a system which involves passing around paper strips with flight information on it amongst the control tower operators. Information such as its runway, taxiway, and gate are handwritten onto the flight strip. Although this is a very low tech system, we saw that it was very effective.

3. New technologies are already on its way to being implemented.

There are existing plans to implement changes. They plan on having a new radar system called ASD-X added by the summer of 2008. A new runway is going to be built and existing runways are being moved further apart from each other. There are also plans to implement a runway status light system that acts like a stop light to pilots.

With this information in mind, our team decided to focus our system on reducing pilot errors, since we saw that this was the major cause of runway incursions. We also borrowed from the idea of runway status lights to create our runway directional lights.

A few times during our design process, our team had further questions regarding LAX. Steve Ramirez taught us about the location of hold bars on the runways. Tony DiBernardo gave us information on the functionality of the hold bars and how ATC's deal with multiple planes on the runways and taxiways.

Dr. Najmedin Meshkati

Professor Meshkati, a professor of Civil/Environmental Engineering at the University of Southern California also gave a presentation in our class. Dr. Meshkati is a subject matter expert on human factors and aviation safety related issues. His presentation provided our team with details of the contributing human factors that caused runway incursions at Tenerife, Chicago, and Milan airports. In his presentation, he focused on how the ATC is under constant stress, which can lead to errors. We realized that it is important for our system to address ATC problems and help make their job easier.

Professor David Wilczynski

Our professor, David Wilczynski, was our primary source for general information and our expert on software systems. Professor Wilczynski introduced us to all of our other professional contacts. He also gave us the idea of having the system generate speech from the route and automatically transmit that to the pilot.

Presentation

Sherry Avery
Chief of LAX Control
Tower

Jason A. Ragona
Manager of Aviation
Safety Investigations
Delta Air Lines, Inc.

Robert Jones
Assistant Manager
LAX Control Tower

One of our final milestones in our design process was presenting our system to our professor and a group of industry experts. A total of five teams in our class presented. During another teams' presentation, Robert Jones suggested adding a feature to include predetermined taxi paths to the system, which was already part of our system. Robert Jones also agreed with our design decision to avoid completely automating taxi path generation due to the many factors involved in the process. An example Robert Jones gave was that an ATC knows Southwest airplanes tend to taxi faster than planes from American Airlines, and thus the ATC will adjust taxi sequences accordingly. These types of optimizations are informal, dynamic, and organic, making them difficult to automate with a computer. Sherry Avery liked our idea of the system generating an audio message to the pilot from the controller. She commented that controllers at LAX often talk very fast because of their high workload and pilots may have difficulty understanding them. Jason Ragona thought our multi-directional runway lights was a great idea, but the cost of embedding lights may be high. We considered this high cost, and decided that the

benefit outweighed the cost. Also, we thought that LAX could install our new lights at the same time their new runways are being built, which would reduce the installation costs. Jason also reminded us to consider pilot and ATC training for the system.

7. Projected Impact

- **Tier 1** – Initial funding to start the company and develop all necessary software and hardware.
- **Tier 2** – Second round funding to do a full scale implementation in one airport.
- **Tier 3** – Third round funding to do full implementations nationwide as well as to periodically update each airports taxi path suggestion algorithms.

Phase 1: Phase 1 will be the initial deployment of The Taxi Management System. It will include all of the components and all of the functionality with the exception of the path suggestion. During this phase the air traffic controller will be required to manually draw every taxi path for all arriving and departing planes on the ATC Graphical Airport Model. During this time the TMS will collect data in the Multi-Variable Taxi Path Database that will later be analyzed and can be used to suggest paths.

Phase 2: Phase 2 will begin once sufficient amounts of data have been collected from taxi paths to be effectively analyzed and turned into path suggestions. This phase will help the TMS become more airport specific. During phase 2, periodic updates to the path suggestion algorithms will be made based on new data that has been collected.

Milestone 1: Software and Hardware Development (Tier 1 funding)

Develop and create a fully functional touch screen interface for the ATC Graphical Airport Model. Test and debug our text to speech converter. Determine all appropriate real world data to collect as well as means to collect it. Develop software interface between ATC Graphical Airport Model and Multi-Direction Runway Lighting. Prepare runway lights for mass

production. Based on a \$200,000 cost per employee/year we believe over a software development time period of 3 years the cost will be \$6million. In year 1 we will have 5 employees, 10 employees during year 2, and 15 employees during year 3.

Milestone 2: Single Airport Phase 1 Integration (Tier 2 funding)

Integrate The Taxi Management System in the tower of a single airport. Make the system specific to that particular airport with accurate runway mapping, airlines, and all other pertinent airport data. Have the system collect real world data for future Phase 2 departure. The estimated airport cost \$135,000 for equipment costs and \$157,500 for runway hardware (see Appendix G).

Milestone 3: Single Airport Phase 2 Integration

The real world data that was collected during Phase 1 will need to be analyzed prior to Phase 2 departure. Once completed, the analyzed data can be used for path suggestion in the ATC Graphical Airport Model. Periodic data analysis and updates for the path suggestion formulae will occur.

Milestone 4: National Phase 1 Integration (Tier 3 funding)

Once success is achieved at one airport we will deploy The Taxi Management System to airports across the nation. Each will need to be specific to the particular airport and therefore will need to undergo Phase 1 to collect relevant data.

Milestone 5: National Phase 2 Integration

As each airport collects enough real world data they will have the data analyzed for an introduction of Phase 2.

Cost Benefit Analysis:

We lack sufficient data to make a reasonable estimate of how much our system will save airports. However, from examining how comparable systems have been reported to reduce runway incursions, we predict that our system will reduce the number of runway incursions by 70%.

The Taxi Management System is designed to enhance situational awareness of both ATCs and pilots. By using the Graphical Airport Model with path suggestions, ATCs will benefit from decreased head down time and spend less time figuring out where to direct planes. By implementing the Multi-directional Runway Light System, pilots unfamiliar with the layout of an airport will be more able to localize themselves by looking at the runway, not by peering at a tiny screen in the cockpit. Moreover, by embedding the lights in the runway, all planes are able to benefit from the system at an airport without requiring airlines to retrofit their fleets. By collecting data about paths taken in various weather and traffic conditions through the Multi-Variable Taxi Path Database, we will be able to analyze and suggest optimal paths for any occasion. By using the Audio Feedback System, we will standardize a portion of the ATC-pilot communication, resulting in fewer miscommunications and less ambiguity in directions. Though the Taxi Management System has a somewhat steep initial investment, it will significantly reduce the severity, number, and rate of runway incursions. The return on that investment will be peace of mind among the Industry, the FAA, and the flying public.

Appendix A: Contact Information

Team Members:

<p>Allen, Donald, Brooks Coopman donaldal@usc.edu (831) 706-6093 311 Liberty St. Santa Cruz, CA 95060</p>	<p>Gilewski, Matthew, Francis mfgilews@usc.edu (818) 434-5533 604 West Kenneth Road Glendale, CA 91202</p>	<p>Windisch, Christopher, James windisch@usc.edu (323) 309-2962 2658 Menlo Ave. Apt. 6 Los Angeles, CA 90007</p>
<p>Chi, Ann, K annchi@usc.edu (469) 235-7298 4408 Ringgold Ln. Plano, TX 75093</p>	<p>Arreola, Fernando jfarreol@usc.edu (310) 283-1286 1237 S. Grevillea Ave. Inglewood, CA 90301</p>	<p>Villegas, Julio, Gonzalo jgvilleg@usc.edu (213) 359-7774 225 N rd St. La Puente, CA 91744</p>
<p>Abraham, Brian babraham@usc.edu (213) 793-1708 10129 Abbotshire Village Pl. Powell, OH 43065</p>	<p>Chang, Chin-Kai chinkaic@usc.edu (408)921-6666 21445 Millard Ln, Cupertino,CA,95014</p>	<p>Ahmed, Osman osmanahm@usc.edu (978) 376- 9240 3 Andover Country Club Ln. Andover, MA 01810</p>

Advisor

Prof. David Wilczynski
 Phone: (213) 740-4507
 Fax: (213) 740-7285
dwilczyn@usc.edu
 University of Southern California
 Computer Science Department
 941 W. 37th Place
 Los Angeles, CA 90089

Appendix B: Description of the University

The University of Southern California, located in the University Park neighborhood in Los Angeles, California, USA, was founded in 1880, making it California's oldest private research university.

U.S. News & World Report ranked USC 27th among all universities in the United States in its 2008 ranking of "America's Best Colleges", also designating it as one of the "most selective universities" for admitting 8,550 of the 33,754 who applied for freshman admission in 2007 for a 25% admissions rate. According to the freshman profile, 18% of admissions were associated with legacy preferences. USC was also named "College of the Year 2000" by the editors of TIME magazine and the Princeton Review for the university's extensive community-service programs. Residing in the heart of a global city, USC ranks among the most diverse universities in the United States, with students from all 50 United States as well as over 115 countries.

USC is also home to Nobel Prize winning Chemistry Professor George Olah, director of the Loker Hydrocarbon Research Institute. The university also has two National Science Foundation–funded Engineering Research Centers—the Integrated Media Systems Center and the Center for Biomimetic Microelectronic Systems. In addition, The U.S. Department of Homeland Security selected USC as its first Homeland Security Center of Excellence. Since 1991, USC has been the headquarters of the NSF and USGS funded Southern California Earthquake Center.

USC is the largest private employer in Los Angeles and the third largest in the state of California and is responsible for \$4 billion in economic output in Los Angeles County; USC students spend \$406 million yearly in the local economy and visitors to the campus add another \$12.3 million.

Appendix C - Non-University/ Industry Partners

Sherry Avery

LAX Control Tower

Manager

Phone: (310) 342-4912

Sherry.Avery@faa.gov

Robert Jones

LAX Control Tower

Assistant Manager

Phone: (310) 342-4912

Robert.Jones@faa.gov

Najmedin Meshkati

Phone: (213) 740 8765

Fax: (213) 744 1426

meshkati@usc.edu

University of Southern California

Civil/Environmental Engineering Department

Industrial and Systems Engineering Department

Kaprielian Hall (KAP Bldg.), Room 224 D

Los Angeles, California 90089-2531 U.S.A.

Jason A. Ragogna

Delta Air Lines, Inc.

Manager, Aviation Safety Investigations

Phone: (404) 773-7787

Mobile: (404) 375-3446

FAA University Design Competition Design Proposal Submission Form (Appendix D)

Note: This form should be included as Appendix D in the submitted PDF of the design package. The original with signatures must be sent along with the required print copy of the proposal.

University University of Southern California

List other partnering universities if appropriate None

Proposal Developed by: Individual Student Student Team

If Individual Student

Name _____

Permanent Mailing Address _____

Permanent Phone Number _____ Email _____

If Student Team:

Student Team Lead C.J. Windisch

Permanent Mailing Address 2355 Ansonia Ave. SW

Grand Rapids, Michigan 49507

Permanent Phone Number (323)309-2962 Email windisch@usc.edu

Competition Design Challenge Addressed:

Runway Incursion

I certify that I served as the Faculty Advisor for the work presented in this Design Proposal and that the work was done by the student participant(s).

Signed David Wilczynski Date April 10, 2008

Name David Wilczynski

University/College USC

Department(s) Computer Science

Street Address 941 W. 37th Place

City Los Angeles State CA Zip Code 90089

Telephone (213)740-4507 Fax (213)740-7285

Appendix E: Evaluation of Educational Experience

Advisor's Evaluation (Professor Wilczynski):

Last year (2007) our class of about 45 students all worked on the same entry. As a result I provided strong technical guidance and the class meetings were quite coherent. This year I divided the class in 5 teams of about 10 members each. I presented a straw man proposal of trying to analyze aviation standard operating procedures and see if we could write tools to “enforce” them in the tower cab and the cockpit. Some teams took that challenge, others didn’t. That, of course, made our class meetings less coherent. As a result (and also to keep up on what the groups were doing) I focused on presentations. Most of the personal time I spent with the teams was reviewing their presentations and commenting on how to give their designs more “pop.” It was during these meetings that we brainstormed on design alternatives, technical extensions, and so forth. In addition, last semester I influenced the editing of the document; this year, with 5 documents, I won’t as much.

So which experience did I prefer? Not really clear, but in the future I will do multi-team approaches mostly because it forces each team member to do more. I think some students last year could “coast” a little because the large team offered so much backup. Not this semester, any failure by an individual team member showed up much more clearly.

Teams Evaluation:

Fernando Arreola

The FAA Airport Design Competition provided a very meaningful learning experience for me. In the working environment engineers mostly work in teams and this competition allowed me to work with a group of my fellow students. This experience helped me learn how to work with other people and contribute to a project in a positive way.

The main challenges that we went through would have to be the changes that were made to certain of the design components as well as the addition of a few features. In the process of attempting to improve our design we had to make changes throughout and that put pressure on me since I was in charge of developing a prototype. This caused me to have to add more features to the prototype as well as modify things that I already had completed. We were able to overcome this by working hard and working together as a team in order to complete all facets of the project we were developing.

In developing our thesis we first paid a visit to the Los Angeles Airport. From this visit we concluded that many runway incursions occur because the pilot is not familiar with the airport. In order to assist the pilots we came up with a couple ideas, GPS in the cockpit and directional runway lights. We decided to go with directional runway lights because we felt that with this approach we would not be restricted by planes that would not have our system on board. After this decision we would meet as a group to refine our idea and also would meet with the professor to get input from him.

The fact that we were able to get input and information from industry experts was great. They really provided us with a great access to information that we may not necessarily be able to

obtain from other resources. The visit to LAX was a great example of this since we got to experience the dynamics in a control tower.

This project was a great representation of what I would experience once I go out to the workforce. Like I mentioned the design was constantly changing based on different demands which occur regularly in the workforce based on the customer. We also got to work in teams which are a common structure for engineering companies. Since I was part of the prototype team, this project also helped me refine my HTML and Javascript skills which are becoming very prominent technologies in the workforce.

Chin-Kai Chang

Through the FAA Airport Design Competition, I earned a lot of valuable learning experiences. Working in a team is a lot more fun than working individually. This competition gave me the chance to learn how to work in a team and do the work more efficiently.

In the beginning of the design, each person has all different kinds of ideas and thinking styles for the competition. To sum up of various kinds of ideas is a big challenge. We go through every detail of our design to make it perfect. Sometimes we need to change our design over and over again because we come up with new questions and new ideas. It really helped me to learn how to communicate with team members.

Throughout the design process, I learned how to apply engineering approaches to solve the problem and understand which kinds of problems we can solve and which we can not solve. For example, we tried to add the speech recognition system into our system, but we found that the current speech recognition is still not robust enough to apply to this design. To choose the right technology is not an easy task to complete during the design. I realized that the most

suitable technology might not necessarily be the newest technology. Just like our light guide system is using the simplest approach to helping pilots find right paths on the runway.

This project gave me the chance to touch the industry problem and work on the solution with the other team members. It helped me to learn how to construct a large design with different approaches. Since I am working for the Text-To-Speech system in our design, I also learned how to convert the text to voice and how to interact with AJAX webpage by the Perl CGI script.

Osman Ahmed

The FAA design competition provided me with a meaningful learning experience. I became well versed in working with a software engineering team, and learned a great deal about product development.

Overall, the team worked extremely well together with every member contributing actively. The one challenge we faced was in the design phase, as we strove to design a system that was practical, feasible, and cost-effective. We solved this by pinpointing the factors we wanted to address and consulting with the professor and with airport personnel on how convenient they foresaw our ideas to be.

In developing our hypothesis, our team first visited Los Angeles International Airport, to become familiar with the ATC environment. We then brainstormed what we thought were the factors leading to runway incursions and how best to solve them. We also decided to take an approach that would minimize modifications in the cockpit, since that would be costly for the different types of aircraft present in an airport.

Industry participation was extremely helpful in our project. By speaking with LAX and FAA personnel, we were able to solicit feedback on our ideas and incorporate that feedback into the formulation of our product.

This project was an excellent application of the skills we have learned in our engineering classes. By working in a team and being involved in all phases of development – from brainstorm to prototype – we were able to gain first-hand knowledge on how the software engineering process is utilized in industry.

Brian Abraham

The FAA Airport Design Competition was a very meaningful learning experience for me as a student about to graduate. This experience allowed me to get a taste of what a full software development cycle is. This is a very valuable lesson for me as I will be expected to develop software from scratch shortly after graduation in a few weeks.

Our team faced a number of challenges while working on this project. Initially we had to work out our differences based purely on our personalities and design approach since very few of us have had experience working on a large project, such as this, in a team of more than 2 or 3. Once we were all seeing eye to eye we had to determine which approach we wanted to take. Runway incursion is such a broad topic with many causes and we realized early that it would be impossible to handle all of them. We toyed around with a number of approaches but finally agreed on the approach we took.

Input from industry experts was very helpful in our design. Many, if not all, of us were very unfamiliar with the details surrounding the problem of runway incursion. If not for the vast

amount of industry input we received, I believe that we would have had a much more difficult time with our design and probably would not have come up with one as affective.

I do believe that this project will help to develop the skills I need to be successful in the workforce. As I stated earlier, shortly I will be required to handle software development from start to finish, which is similar to what we did on this project. The experience I gained will be invaluable as I make my transition from college to the workforce.

Julio Villegas

1. Participating in the FAA competition provided a meaningful experience through the research and industry interactions I had the chance to be part of. Designing a solution required me to have a thorough understanding of the problem, state of the art technologies, and know the underlying factors leading to runway incursions.
2. A big challenge was designing a solution to that was practical to integrate in real life. In addition, deciding on which technology to use was another challenge. Our group was able to overcome these challenges by conducting extensive research on state of the art runway incursion prevention systems.
3. Our team developed a hypothesis after research and having had interactions with airport industry personnel.

4. Participation with industry helped clarify airport SOP. Visiting the LAX tower gave me a better sense for the responsibilities of the ATC and the environment they work in. Presenting our teams ideas to industry was extremely important in narrowing our focus to a specific solution.

5. I gained an extensive amount of knowledge about airports and runway incursions through the domain analysis that we conducted. This included airport SOP's, causes for runway incursions, and state of the art technologies. With this experience I do feel prepared to pursue further study on the matter.

Matt Gilewski

The FAA Airport Design Competition provided a wonderful learning experience as we were able to interact with industry. I particularly found it useful to learn how to translate real-world requirements to system requirements in a setting outside of a dull exam question.

Working as part of a team means that one must deal with scheduling team meetings and keeping the team focused on developing a cohesive project. We overcame the scheduling challenges through email contact with members who could not meet.

We had a general ideas brainstorm, settled on the idea of runway lights to improve situational awareness, then refined the situational awareness idea through further brainstorm sessions. We tried and succeeded in keeping the design as a supplemental system, so that if the system were to fail completely, airports would still be able to function safely and normally. Our professor suggested that the design needed more "oomph" so we added the path logging and suggestion after another group brainstorm.

We received very good and helpful feedback from industry officials in our visit to the LAX tower and our presentation to LAX officials and a Delta manager. We had tried to minimize "head down time" and it was good to know that industry officials were at least intrigued by our ideas.

This program was definitely an interesting and useful experience. I may not go into aviation, but learning to meet and exceed requirements while optimizing system safety, usability, and cost will be invaluable in any job as a software engineer.

Donald Allen

1. One common complaint from engineering students and employers alike is the disconnect between formal education and industry experience. Too often, diligent students are nonetheless ill-prepared for the work environment of their future jobs because of the impractical, academic nature of their curriculum. As the focus of our senior-year capstone software course, the FAA Design Competition was an effective bridge between our university education and the “real world” of team-based software design for actual industry concerns.

The development of our Taxi Management System was a crash-course in task delegation, interdependence, self-organization, and responsibility; each member of our team felt an unwavering drive to match the excellent work of their peers.

2. Aside from the benefits of working with other dedicated engineers, the team paradigm presented its own set of challenges. Each team member was a full-time student, so we had to contend with many other obligations while scheduling meetings. It helped to have the dedicated

class time to meet as a group and work on the project; without the academic sponsorship of approaching the project as accredited coursework, it would have been much more difficult to follow through in the midst of the busy spring semester.

3. To produce a well thought-out design, we first went through a thorough research phase.

During this research we read public documentation of the state of air traffic control, visited an LAX control tower, and met with several industry experts. Once we had a good understanding of the problem and domain, we began discussing requirements, what our system would have to be capable of to address the problems we'd found. Once these were made, the design came from many long meetings of the entire team, all collaborating in brainstorm sessions with notes and a blackboard. The design went through several versions as we fleshed out which features were truly beneficial and which should be cut. We are indebted to our Professor Wilczynski for his wisdom and enthusiasm for the importance of our task.

4. We had the benefit of industry input at all stages of our project, from research to design. Early on, our trip to LAX gave us the opportunity to form a realistic view of the ATC's working environment, and which factors were most important to the job. Overall, this was an invaluable experience, but I believe we would have gleaned more from it if we had had more time prior, during which to study the domain's common-knowledge and learn to ask the right questions. Toward the end of the design phase we were able to show our system to a panel of industry experts in a 45 minute PowerPoint presentation and Q&A. This served as the final "yea or nay" on our ideas' feasibilities. Although it was a great relief when they approved, it might have had a

more substantive impact on our final product if the feedback had been solicited once we had our first-draft design.

5. I don't know what the timing of other colleges' spring semester is, but the Competition ends over a week from the end of ours; it would have been nice to have the full semester to work on our submission, and turn it in shortly after final exams.

C.J. Windisch

The dean of USC's Viterbi School of Engineering, Yannis Yortsos, has argued that the future of engineers will be interdisciplinary and will require engineers that have a "seamless blending of left-brain and right brain skills" to interact with non-technical people. The FAA's Airport Design Competition has proved this too me.

In creating our solution I have used just about every skill I've learned during my years in college. I have used graphic design skills that I learned from working at the USC Roski School of Fine Arts to create graphics for presentations and user interface mock-ups. I have used the visual communication and story telling skills that I learned from my first major in college, Cinema Television, to create this document and present our team's ideas. I have used my writing skills learned throughout college to organize our team's ideas and activities. I have used my organizational and management skills that I've learned in from my studies of business to help make our design happen. I have used my user interface design and implementation skills to help create a prototype. I have used my knowledge in the fields of artificial intelligence, psychology, and neuroscience to do user interface design and stakeholder analysis. I have used my skills in software and systems engineering to help direct our design process and ensure a top quality

design. And lastly I have used my people skills in interacting with stakeholders to get information to make a system that truly adds value to them. All of this went into the analysis and design of our system.

This learning experience, in the context of my Computer Science “Capstone Design” course, has not only tried my skills, but has given me extremely valuable experience. It has given me the opportunity to use my skills for something more than just a class project. The experience I’ve gained from utilizing all of my skills has taken me beyond just the book smarts I’ve learned from hours in the library and online, and has truly prepared me to make the transition from a student to a professional Software Engineer.

Throughout the competition, we encountered many challenges, especially in turning our vague ideas into concrete system designs. We found that what sounded like simple ideas required a lot of time and effort to design precisely.

We also encountered many communication challenges. We communicated online and in live meetings, and found that it was essential to record our decisions online in words and in pictures (diagrams) to progressively design such a complex system.

To develop our solution, we brainstormed. We constantly had brainstorm and design meetings to come up with solutions to core incursion problems that were not excessively costly and were realistically developable today. This meant drawing sketches, making mock-ups, drawing on the chalkboard, and presenting our ideas to our class, experts, and professor to gain feedback.

Industry participation was essential to understanding the problems. Our initial assumptions were completely off base. Originally we assumed that a system LAX had in place that used paper flight strips was bad, however, after watching and consulting with actual ATC’s,

we found the paper system to make a lot of sense and not to be a major problem. Plus from industry we found that we should focus on pilot error.

Annie Chi

Since working on the FAA Airport Design Competition, I believe I have been able to expand my technical skills and my ability to work with other people. Our team worked on this competition as part of a school project. In general, I've found that it is difficult to glean many practical skills from class projects. However, this competition provided a very different experience for our class.

I started working at an internship at around the same time I started working on this competition. The skills I developed while working on this competition prepared me to work with my team at my internship. These skills did not just involve programming, but also people skills within a technical setting. Being a graduating senior, I was also interviewing for full time positions while I working on this project. Many interviewers were very interested in my experience in this competition. The knowledge and experience I gained regarding the software life cycle seemed to be exactly what recruiters were looking for in potential hires.

One of the most challenging aspects of this competition was working with a large group. Prior to this class, the largest group I'd worked with on a computer science project involved three people – there were nine people in the group I worked with for this project. Having ten smart, talented, and motivated students on one project does have its benefits, but trying to organize and moderate between ten different people also had its own set of difficulties. Fortunately, we were able to decide on various meeting times early on so that everyone in the

group had a chance to meet together. Also, every student was in charge of a specific aspect of the project so the work was evenly divided.

Throughout our design process, I believe our team did a great job in getting our own work done while helping each other out as well. For each step in the system design there were two students who were in charge, and they led the rest of the team through each stage. Our process was collaborative and iterative. We went through each part of our system again and again, refining it each time as different people came up with new ideas and input.

Having experts in the industry participation in our project was very useful. They provided us with information and insight that would have been very difficult to get just from doing our own research. I believe that it would have been impossible to design a system to the level that we did without the help of our industry contacts.

Overall, I had a great time working on this project. It was educational and meaningful. I was honored to be able to do a small part in helping the FAA solve the problem of runway incursions.

Appendix F: References

- [1] Federal Safety Administration, "Runway safety report," September 2007,
<http://www.faa.gov/runwaysafety/pdf/rireport6.pdf>

- [2] Aircraft Owners and Pilots Association, "Runway Incursion Analysis," October 2005,
http://www.aopa.org/asf/accident_data/incursions.html

- [3] Aircraft Owners and Pilots Association, "Runway Safety - Incident Categories"
<http://www.aopa.org/asf/categories.html>

- [4] International Civil Aviation Organization, "Manual on the Prevention of Runway Incursions," http://www.icao.int/fsix/res_ans.cfm

- [5] National Transportation Safety Board, "Runway Incursion Forum: Promoting Runway Safety," March 27, 2007, http://www.nts.gov/Events/symp_ri/symp_ri.htm

- [6] By M. Kirby, "Animation: NTSB shows just how close Boeing 737 and Boeing 747 really were at O'Hare," National Transportation Safety Board, November 14, 2006,
<http://www.nts.gov/Events/2006/MostWantedFed/AnimationDescription.htm>

- [7] National Transportation Safety Board, "Runway Incursion – NTSB," June 13, 2000,
http://www.nts.gov/Events/2000/incursion/incur_video.htm

- [8] National Transportation Safety Board, "Runway Incursion Accidents, Serious Incidents, Recommendations, and Statistics," December 2007,
<http://www.nts.gov>

- [9] "Airport Maps and Runway Incursions" July 2006, [online]. Available:
http://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0002NW

- [10] KNBC News, "FAA Says LAX Logged First Runway Incursion of Year," March 25, 2008,
<http://www.knbc.com/travelgetaways/15702996/detail.html>

- [11] NBC Nightly News with Brian Williams, "Secrets of Air Traffic Controllers," March 8, 2008, <http://www.msnbc.msn.com/id/3032619/vp/23534592#23534592>
- [12] NBC Nightly News with Brian Williams, "FAA skimping on safety?," February 16, 2008, <http://www.msnbc.msn.com/id/3032619/#23200059>
- [13] MSNBC, "Near Misses - Runway Incursions on the Rise," March 29, 2008, <http://www.msnbc.msn.com/id/21134540/vp/23858254#23858254>
- [14] B. Pfister, "FAA, controllers differ over claims crash 'seconds' away," March 7, 2008, http://www.pittsburghlive.com/x/pittsburghtrib/news/cityregion/s_555963.html
- [15] Gawande, "The Checklist-If something so simple can transform intensive care, what else can it do?" The New Yorker, December 10, 2007, http://www.newyorker.com/reporting/2007/12/10/071210fa_fact_gawande
- [16] "lax_runwayincursion," [online].Available: http://www-scf.usc.edu/~csci477/FAA/lax_runwayincursion.wmv
- [17] N. Meshkati and C.R.T. Ricafort, "Stop the bulldozers," Los Angeles Times, December 17, 2007, <http://www.latimes.com/news/opinion/la-ow-meshkati17dec17,0,641729.story?coll=la-promo-opinion>
- [18] "Runway Incursion - The Next Major Accident?" April 15, 2002, [online].Available: http://findarticles.com/p/articles/mi_m0UBT/is_15_16/ai_84811773
- [19] "LAX Runway Incursions 2003-3005," <http://www.lawa.org/airops/runway.cfm>
- [20] "LAX Runway Incursions – 2006," <http://www.lawa.org/airops/runway.cfm>
- [21] "LAX Runway Incursions – 2007," <http://www.lawa.org/airops/runway.cfm>
- [22] "Los Angeles Tower Standard Operating Procedures," June 11, 2007

- [23] Federal Safety Administration, "Air Traffic Control," February 14, 2008,
http://www.faa.gov/airports_airtraffic/air_traffic/publications/atpubs/Atc/index.htm
- [24] Federal Safety Administration, "Radio Communications Phraseology and Techniques"
http://www.faa.gov/airports_airtraffic/air_traffic/publications/ATpubs/AIM/Chap4/aim0402.html
- [25] Wikipedia, "NATO phonetic alphabet,"
http://en.wikipedia.org/wiki/NATO_phonetic_alphabet
- [26] Aircraft Owners and Pilots Association, "AOPA Air Safety Foundation Online Courses"
http://www.aopa.org/asf/runway_safety
- [27] S. Keller, "Investigation of Runway Incursion Prevention Systems," Cornell University
ECE Dept. May 14, 2003,
http://132.236.67.210/engrc350/ingenuity/Keller_S_issue_1.pdf
- [28] N.S.Dorigi and B. Rabin, "NASA Uses Virtual Reality to Target Runway Incursions at
LAX," NASA Ames Research Center, 2002,
http://www.simlabs.arc.nasa.gov/library_docs/tech_papers/nasa_lax.pdf
- [29] R. Cassell, C. Evers, B. Sleep and J. Esche, "Initial Test Results of Pathprox- A Runway
Incursion Alerting System," Digital Avionics Systems, 2001,
<http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/7625/20788/00963334.pdf?tp=&isnumber=&arnumber=963334>
- [30] K. Barnstorff, "No More Blind Flying," NASA Langley Research Center, October 5, 2004,
http://www.nasa.gov/vision/earth/improvingflight/svs_reno.html
- [31] S.D. Young and D.R. Jones, "Runway Incursion Prevention: A Technology Solution,"
NASA Langley Research Center, November 2001,
http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20070030080_2007025910.pdf
- [32] S.D Young and D.R. Jones, "Runway Incursion Prevention Using an Advanced Surface
Movement Guidance and Control System (A-SMGCS)," Digital Avionics Systems

Conferences, 2000,
<http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/7093/19162/00886913.pdf?arnumber=886913>

- [33] "DARPA advances artificial intelligence program for air traffic control," February 11, 2008, [online]. Available: <http://www.networkworld.com/community/node/24924>

- [34] "Beyond Radar – Pathprox," [online]. Available: <http://www.erabeyondradar.com/pathprox/>

- [35] C.L. Scovel III, "Actions Needed To Improve Runway Safety," Department of Transportation,
http://www.oig.dot.gov/StreamFile?file=/data/pdfdocs/WEB_IG_Statement_on_Runway_Safety.pdf

- [36] M.L. Wald, "F.A.A. Wants Stoplights Added to Runways," New York Times, March 26, 2008,
http://www.nytimes.com/2008/03/26/business/26runway.html?_r=1&hp&oref=slogin

- [37] Siemens, "Airfield Lighting," <http://www.industry.siemens.com/adb/en>

- [38] Wikipedia, "Transponder (Aviation),"
http://en.wikipedia.org/wiki/Transponder_%28aviation%29

- [39] Wikipedia, "Automatic Terminal Information Service,"
http://en.wikipedia.org/wiki/Automatic_Terminal_Information_Service

- [40] Wikipedia, "Airport Surveillance Radar,"
http://en.wikipedia.org/wiki/Airport_surveillance_radar

- [41] Wikipedia, "Pushback," <http://en.wikipedia.org/wiki/Pushback>

- [42] Wikipedia, "Speech Recognition," http://en.wikipedia.org/wiki/Speech_recognition

- [43] Carnegie Mellon Sphinx Project
<http://cmusphinx.sourceforge.net/html/download.php#sphinx4>

- [44] Map Overlays, <http://code.google.com/apis/maps/documentation/overlays.html>
- [45] Google Maps API Reference,
<http://code.google.com/apis/maps/documentation/reference.html>
- [46] Google Maps API Tutorial, <http://economy.googlepages.com/index.htm>
- [47] Aircell, <http://www.aircell.com/>
- [48] Wikipedia, "Tenerife Disaster," http://en.wikipedia.org/wiki/Tenerife_disaster
- [49] FAA Runway Incursion Data, http://www.faa.gov/runwaysafety/data/ri_tot.cfm
- [50] Sturgell, Robert. "*Red Means Stop.*"
http://www.faa.gov/news/speeches/news_story.cfm?newsId=10177
- [51] LATimes, "Runway System Being Tested Could Save Lives"
<http://www.latimes.com/technology/la-me-runway25feb25,1,78247.story?page=1>

APPENDIX G: Extra

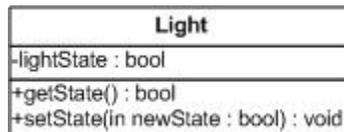
Class Design Code:

Prototype

During our process of designing the Taxi Management System we were able to develop a working prototype. We used the Google Maps API in order to draw routes and simulate seven planes arriving or leaving the Los Angeles Airport North runways. Since this is a web based application we had to replace the touch screen functionalities with clicking on the runways to draw the routes. A link to the prototype is provided below:

<http://csci351.usc.edu:8105/prototype.html>

Light Class



```
Class Light{
    bool lightState; //true for ON false for OFF

    bool getState(){ return lightState; }
    Void setState(bool newState){
        lightState = newState;
    }
}

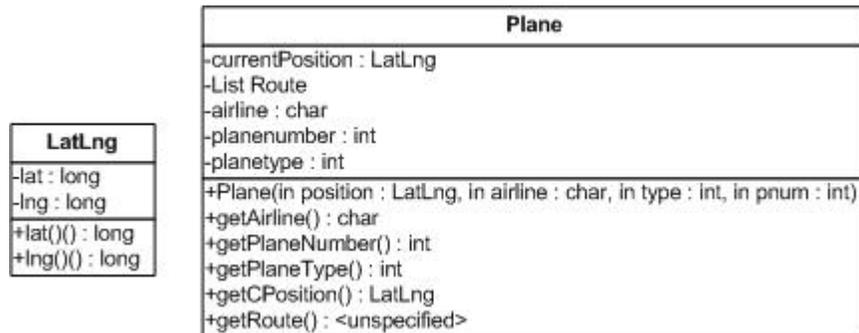
Class HoldBarLight extends Light{

}

Class ArrowLight extends Light{
```

```
}
```

Plane Class



```
Class Plane{
```

```
LatLng currentPosition; //where the plane is in Latitude and Longitude  
List<Runway> route; //list of the runways in the route that the plane must take  
String airline;//the airline of the plane  
Int planenumber;//the plane number of the plane  
Int planetype;//0 for arrivals, 1 for departures
```

```
//constructs a plane object and initializes the components
```

```
Plane(positon, airline, fnum, type){  
    This.currentPosition = position;  
    This.airline = airline;  
    This.planenum = fnum;  
    This.planetype = type;  
}
```

```
//gives back the airline
```

```
String getAirline(){  
    Return airline;  
}
```

```
//gives back the plane number
```

```
Int getPlanenumber(){  
    Return planenumber;  
}
```

```

//gives back the plane type
Int getPlaneType(){
    Return planetype;
}

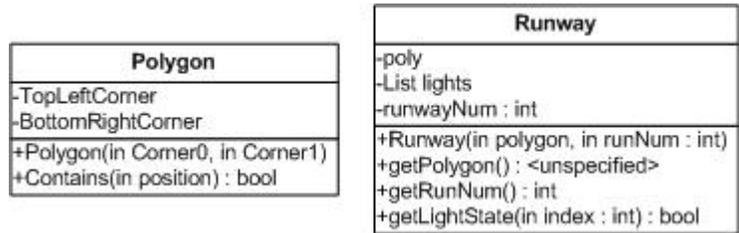
//gives back the route
LatLng getCposition(){
    return currentPosition;
}

//gives back the route
void setCposition(LatLng pos){
    currentPosition = pos;
}

//gives the route that the plane is suppose to take on the runway
List<Runway> getRoute(){
    Return route;
}
}

```

Runway and Polygon Class





Taxiway Z has references to the circled arrow lights and is responsible for determining which one is lit when drawing a route.

```
Class Runway{
```

```
  Polygon poly; //the space the runway takes up represented by the four corners in
                //Latitude and Longitude
```

```
  List<ArrowLights> lights;
```

```
  Int runwayNum;
```

```
    //constructs runway object and initializes the polygon
```

```
  Runway(polygon, runNum){
```

```
    This.poly = polygon;
```

```
    This.runwayNumber=runNum;
```

```
  }
```

```
    //gives access to the polygon
```

```
  Polygon getPolygon(){
```

```
    Return poly;
```

```
  }
```

```
    //gives access to the runway number
```

```
  Int getRunNumber(){
```

```
    Return runwayNum;
```

```
  }
```

```
    //returns the state of the lights
```

```
  bool getLightState(int index){
```

```

        Return lights[index].getState();
    }
}

```

Class Polygon{

```

    LatLng TFcorner, BRcorner; //top left and bottom right corners in Latitude and
                               //Longitude

```

```

    //construct a polygon by initializing the top left and bottom right corners
    Polygon(cor0,cor1){
        This.TFcorner=cor0;
        This.BRcorner=cor1;
    }

```

//checks whether or not the position passed in is inside the polygon

```

    Bool Contains(LatLng Position){
        Bool inside = false;
        //check if it is between the latitudes of the top left and bottom right corners
        If(Position.lat() <= TFcorner.lat() AND Position.lat()>=BRcorner.lat()){
            //check if it is between the longitudes of the top left and bottom right corners
            If(Position.lng() <= TFcorner.lng() AND Position.lng()>=BRcorner.lng())
                Inside = true; //if it passed both checks set inside to true
        }
    }
    Return inside;//if it did not pass both tests inside would be false
}

```

Airport Class

Airport
-List flights
-List runways
-Map<key:Airline+FlightNumber, Element:Flight> fmap
+getFlight(in airline : char, in flightnum : int) : <unspecified>
+addFlight(in flight) : void
+checkDeviations() : void

Class Airport{

```

    List<Flight> flights;//list of the flights that are active
    List<Runway> runways;
    Map<Key:Airline+FlightNumber, Element:Flight> fMap;

```

```

    //get access to a flight in the system
Flight getFlight(String airline, Int flightnum){
    Return fmap.get(airline+flightnum);
}

//add a flight to the list of flights in the system
void addFlight(Flight flight){
    flights.push(flight); //add the flight to the end of the list
    fMap.add(flight.getAirline()+flight.getFlightNumber(), flight); //create an entry to
                                                                    //easily find a flight
                                                                    //based on airline and
                                                                    //flight number

//check if any flight is deviating from its route
    Void checkDeviations(){
        List<Runway> tmpRoute;
        Bool inside;
        Int j;
        LatLng pos;
        For(int i=0; i<flights.length; i++){ //go through all the active flights
            j=0;
            inside=false;
            tempRoute = Flights[i].getRoute();
            pos = Flights[i].getCposition();
            while(inside!=true AND j<tempRoute.length){ //go through the runways
                                                                    //for the route of the flight
                inside = tempRoute[j].getPolygon().contains(pos); //check if the
                                                                    //plane is in the
                                                                    //runway
                j++;
            }

            If(inside == false){ //if the plane was not in any of its route runways show
                                                                    //show on the map that it is deviating
                Map.showDeviation(flights[i]);
            }
        }
    }
}
}
}
}

```

```

Class InputAbsLayer{

    Flight [] LocationDataInterface;
    Airport airport; //reference to the airport class

    //...
}

```

```

//RADAR updates LocationDataInterface

//...

void update(){

    for(int i=0; i<LocationDataInterface.length; i++){
        String airl = LocationDataInterface[i].getAirline();
        Int fnum = LocationDataInterface[i].getFlightNumber();
        Flight ftemp = airport.getFlight(airl, fnum);//get the flight from the airport object
        If(ftemp == NULL){//if the plane is not currently in the system
            airport.addFlight(LocationDataInterface[i]);
        }
        else{//the plane is already in the system
            ftemp.setCposition(LocationDataInterface[i].getCposition());//update the
            }
            }
            }
            airport.checkDeviation();//see if any of the flights are deviating from the path
            }
            // assigned
        }
    }
}

```

Team Roles and Responsibilities:

Team Member	Primary, Secondary Roles	What they actually did
C.J. Windisch	Requirements, Documentation	Wrote the MBASE requirements and goals document, organized team meetings and delegated responsibilities, created presentation graphics and photoshopped user interface mock-ups, defined use cases
Brian Abraham	Documentation, Prototype	Lead the documentation and presentation efforts, wrote documentation and fleshed out formal use cases, created the final designs for the runway lights
Kai Chang	Prototype, Design	Researched voice generation and recognition (dropped feature), prototyped voice generation
Julio Villegas	Research, Design	Researched runway incursions, condensed the LAX Controller SOP's, researched runway light hardware, create formal design diagrams, translated object code into design diagrams
Fernando Arreola	Prototype, Research	Prototyped the runway lights and ATC user interface with Google Maps. Integrated the final prototype. Made incremental UI suggestions and improvements. Prototyped the back-end database.
Don Allen	Design, Documentation	High level design, presentation creation, object level design, wrote the design portion of our MBASE system document
Annie Chi	Design, Prototype	Architecture level design, object level design, designed and implemented the final look of the ATC interface
Osman Ahmed	Research, Requirements	Researched runway incursions, condensed the LAX Controller SOP's, interfaced with our LAX contacts, researched various questions as they came up
Matt Gilewski	Documentation, Requirements	Came up with the runway light idea, research, wrote documentation, designed the statistical database

System Implementation Costs:

Inventory Analysis			
Item	Quantity	Cost	
	per runway	unit	per runway
touchscreens	3	2,000	6,000
thin clients	3	500	1,500
TTS server	1	10,000	10,000
light clusters	100	1,000	100,000
path suggestion DB (and server)	2	10,000	20,000
main server	2	10,000	20,000
Total Runway Equipment Cost:			\$157,500
Item	Quantity	Cost	
	per airport	unit	per airport
logging DB(and server)	2	10,000	20,000
networking materials	1	5,000	5,000
redundant power system	1	10,000	10,000
hardware certification	1	50,000	50,000
Airport Software Setup	1	50,000	50,000
Total Airport Equipment Cost:			\$135,000

Cost Analysis for Los Angeles International Airport (LAX)			
Amount of Runways at LAX:	4		
Cost	Year		
	1	2	3
Software Engineering Labor ¹	1,000,000	2,000,000	3,000,000
touchscreens	24,000		
thin clients	6,000		
TTS server	40,000		
light clusters	400,000		
path suggestion DB (and server)	80,000		
main server	80,000		
logging DB(and server)	20,000		
networking materials	5,000		
redundant power system	10,000		
hardware certification	50,000		
Airport Software Setup	50,000		
Year End Total	\$1,765,000	\$2,000,000	\$3,000,000
Total System Cost:			\$6,765,000
<i>Footnotes:</i>			
1. Assumes \$200,000 per year per person with 5 people in Year 1, 10 in Year 2, 15 in Year 3			