Airport Communication-Based Incursion Detector (ACID)

Designed By (Undergraduate Students):

Ryan Berti

Ryan Brown

Tommy Holford

Michael Hsu

Zhongyuan Li

Farid Nobakht

Denis Tulskiy

Advised By:

Dr. David Wilczynski

University of Southern California

Viterbi School of Engineering

April 2010

1. Executive Summary

Our Airport Communication-based Incursion Detector (ACID) provides software to help increase the situational awareness of both pilots and controllers. The system analyzes the spoken dialogue inherent in airport operations between controllers and pilots. Safety logic is used to make inferences about what has been said and alerts are distributed. The goal of the system is to provide alerts before incursions occur that have a significant potential for collision.

ACID takes audio transmissions and produces from them a set of statements to be asserted in a knowledge base that has a model of the airport. Then, ACID applies logic rules that can: (1) detect operational errors that might result in incursions (for example, we can tell if controllers have given access to the same runway to different airplanes); and (2) turn on runway safety lights as soon as clearance is given to a pilot. We verified our design by constructing a prototype and showing it to the LAX staff who lauded it for its ability to increase the situational awareness in both controllers and pilots.

The system we propose will be cost efficient in its use of technology currently in place in some airports, implementable while adhering to current practices, and unique in its ability to recognize potential safety hazards in time for mitigating options. ACID provides an essential service that is not currently available.

Contents

2. Problem Statement and Background on Design Challenge

The Federal Aviation Administration (FAA) design competition has challenged its participants to reduce runway incursions by expanding the situational awareness of pilots and controllers on the airfield. Our proposed system achieves this by analyzing the dialogue between pilots and controllers and making inferences about a plane's current and future location. Our system will interface with current technologies to alert airport operators to incursion risks and forewarn pilots to in-use runways. In many cases the inferred data will give operators and pilots time to correct mistakes on the aerodrome in ways beyond what the current technologies offer.

The FAA adopted the International Civil Aviation Organization (ICAO) definition of runway incursion in late 2007 to be "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." Recently, two new technologies have led to a reduction in the number and severity of runway incursions: (1) the ASDE-X system alerts controllers to runway incursion risks and imminent collisions, while (2) the Runway Safety Lights system (RWSL) forewarns pilots to runways being used by arriving and departing aircraft. Both systems base their safety logic on the real time position and velocity data of aircraft in the airport environment. Only the most technologically advanced airports in the United States have these systems, but their use will become more widespread. We hope to augment their performance with our work.

ASDE-X is a combination of GPS sensors and multilaterated ground radar. This system produces accurate data regarding a plane's position and velocity, and this data is displayed in real time on screens in the control tower. In addition to effectively eliminating blind spots and constraints on visibility, ASDE-X provides a layer of safety logic that is used to warn ground controllers to imminent collisions on the runway. The safety logic uses an airplane's position and velocity to compute the acceleration of the vehicle, and if this value is high enough the system understands that the airplane is either taking off or landing. Once this inference is made, the system checks for other vehicles in the path of the airplane and alerts controllers if collision risks are present. By 2011 there will be 35 airports in the United States using the ASDE-X system (*Airport Surface Detection Equipment, Model X*).

The RWSL system is a more experimental technology that forewarns pilots to situations leading to runway incursions rather than alerting them once an incursion has occurred. RWSL uses the data produced by the ASDE-X system to turn on lights on in-use runways. When a runway has an arriving or departing airplane on it, red lights along the sides of the runway called runway entrance lights are turned on that act as stop signs for airport traffic that may be crossing the in-use aerodrome. When an airplane is crossing the runway the opposite occurs, lights in the center of the runway called takeoff hold lights are turned on that act as stop signs for departing flights. There are currently two airports in the US that have both types of warning lights installed, and a third airport that has runway entrance lights but not takeoff hold lights in place. These airports all attribute lower numbers of incursions to the presence of the RWSL system, and more development of this technology can be expected (*Runway Safety Lights System*).

Both the ASDE-X and RWSL systems focus on preventing runway incursions that stem from pilot and vehicle deviations. A pilot deviation occurs when an aircraft enters a runway without clearance from a controller. A vehicle deviation is similar, but can involve any unauthorized vehicle or pedestrian on a runway. In the time between 2003 and 2006, pilot and vehicle deviations accounted for more than 70 percent of all runway incursions. The remaining 30 percent of incursions stemmed from operational errors, or controller actions that resulted in less than the minimum separation between two or more aircraft or the entrance of an airplane into an in-use runway (*Federal Aviation Administration's Runway Incursion Program*.). Our proposed system hopes to alert controllers to these operational errors before they turn into runway incursions while interfacing with current technologies to provide a fluid solution for incursion prevention.

Besides interfacing to these technologies, our system conforms to the FAA system of operational “checks and balances” for communications between pilot and controller that are meant to foster a shared understanding of the airplane’s role in the larger airport environment. Because the most important subsets of communication are well documented by the FAA, we can analyze audio data and produce assertions about pilot and controller intent. These assertions about the current and future location of an airplane will be used to produce alerts about potential incursion risks. Along with providing a system for advanced incursion analysis, we propose to log our communication and inference data for future review. Here are three concrete examples of how our system can help:

1. In the situation where a controller clears an aircraft to move across a runway that is occupied by an aircraft with clearance to takeoff, our system can trigger an alert about the possible incursion well before it is detected by the ASDE-X acceleration based safety logic.
2. When clearance for takeoff is given to an airplane the runway entrance lights can be turned on before the airplane has reached takeoff speed and ASDE-X recognizes the vehicle as a departure.
3. When clearance for takeoff or landing is given to an aircraft for a runway that is designated "closed" on the ASDE-X, our system can trigger the alert on the closed runway operation as soon as the clearance is issued, well before the ASDE-X Safety Logic System will detect the operation and generate an alert.

We believe that analyzing communications is a resource that is largely untapped by the current runway safety systems. Its integration into the larger safety logic of the ASDE-X and RWSL systems will be beneficial to both controllers and pilots alike, and would require little to no modification of current FAA procedures. Our solution was produced using classic software engineering methods and is both commercially viable as well as environmentally scalable. The integration of communication based intent is the logical next step in airport safety technology.

3. Summary of Literature Review

During our research, we referred to a set of documents to verify that our system's design would solve the problem proposed and adhere to the FAA's guidelines. To understand runway incursions and their causes, we consulted the set of Runway Safety Reports produced by the FAA from 2006 to 2009. To better understand the standard operating procedures outlined for pilots and controllers, we consulted the FAA's 7110.65S Air Traffic Control document. Information about current safety systems was mined mainly from their respective distributor's websites, and examples of pilot-controller communication streams were taken from sites such as LiveATC.net.

The FAA's Annual Runway Safety Report was integral in our in our research of safety hazards on the aerodrome. We referenced all five reports produced from 2004 to 2009, but held a focus on the most recent publication due to the updated definition of runway incursion adopted in 2008. In these reports, we identified operational errors as the type of runway incursion not currently being mitigated with safety technology. These reports also included statistics about the frequency of runway incursions in the US and incidences where collision occurred or was narrowly avoided.

Once we decided that we would address operational errors, or more specifically communication-based incursions, we did research into what procedures were currently in place to maintain coordination between pilots and controllers. The FAA's 7110.65s details the phraseology of air traffic operations and served as the basis for our voice analysis. This document was recommended to us by LAX personnel and became integral to the design of ACID.

Finally, we referenced a set of industry websites for information about current safety technologies. Sensis corporation's website provided us with a detailed description of the ASDE-X system's components, and the Lincoln Laboratory website offered a set of facts about current implementations of the RWSL system. Once we had a thorough understanding of the current state of safety technology, we searched for real world examples of pilot-controller dialogue. The website LiveATC.net allowed us to listen to real time communication streams, while the website AviationSafety.net has a database of communication streams leading to runway incursions. Our original analyses came from dialogues provided to us by Mr. Thomas Bennett, a controller at LAX.

This combination of literature allowed us to fully understand the problems the FAA faces, the methods they have for mitigating them, and the untapped resources available for future safety technologies.

4. Team's Problem Solving Approach To Design Challenge

Our proposed system, Airport Communication-Based Incursion Detector (ACID), hopes to mitigate runway incursions that stem from operational errors and forewarn pilots to in-use runways while logging data about communication between controllers and pilots. To do this effectively, ACID will obtain audio data through a direct radio link to pilot-controller communication streams. It will then translate voice to a text format, create assertions about the communication, and distribute inferences about incursion risks to controllers through the ASDE-X interface. The system as a whole will alert controllers and pilots to runway incursion risks before location-based systems have the ability to do so. The system architecture is shown in figure 1.

ACID is a real time system in its continuous extraction of meaningful data from the communication stream between pilots and controllers. From the radio interface (RDVS), the system receives the audio communication. ACID's design makes use of a speech recognition system to "listen" to this audio stream and infer any potential issues. It alerts pilots and controllers as needed.

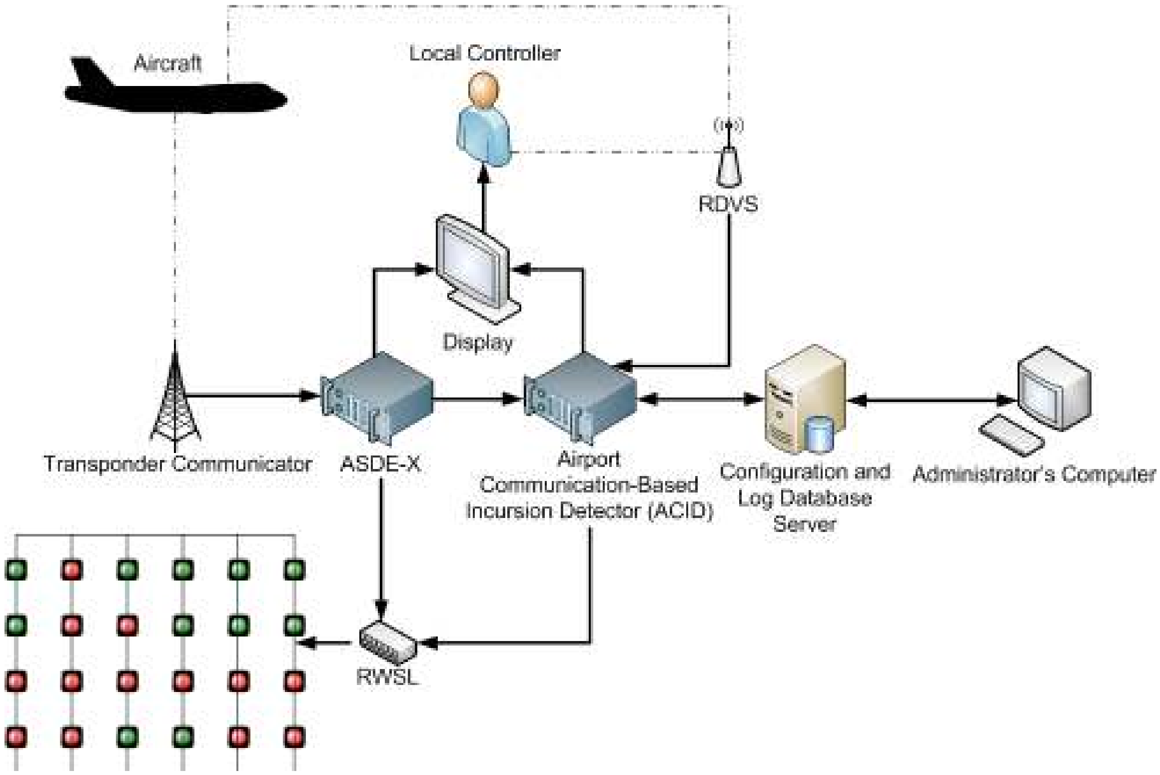


Figure 1: ACID Architecture

To explain ACID, we will explore two distinct scenarios. The first will provide a detailed description of the system's processes while the second will give insight into the complexity and scalability of our safety logic.

4.1. Runway Entrance Lights

One of the most beneficial, and simple, aspects of our system's design is its ability to illuminate runway entrance lights (RELs) before current systems do so. The illumination of these RELs will be used to show how the components of ACID perform their tasks. A common scenario at Los Angeles International Airport (LAX), shown in figure 2 below, involves a departure that is cleared to takeoff on an inner runway (24L) and an arrival on an outer runway (24R) that is told to hold short of the inner runway.

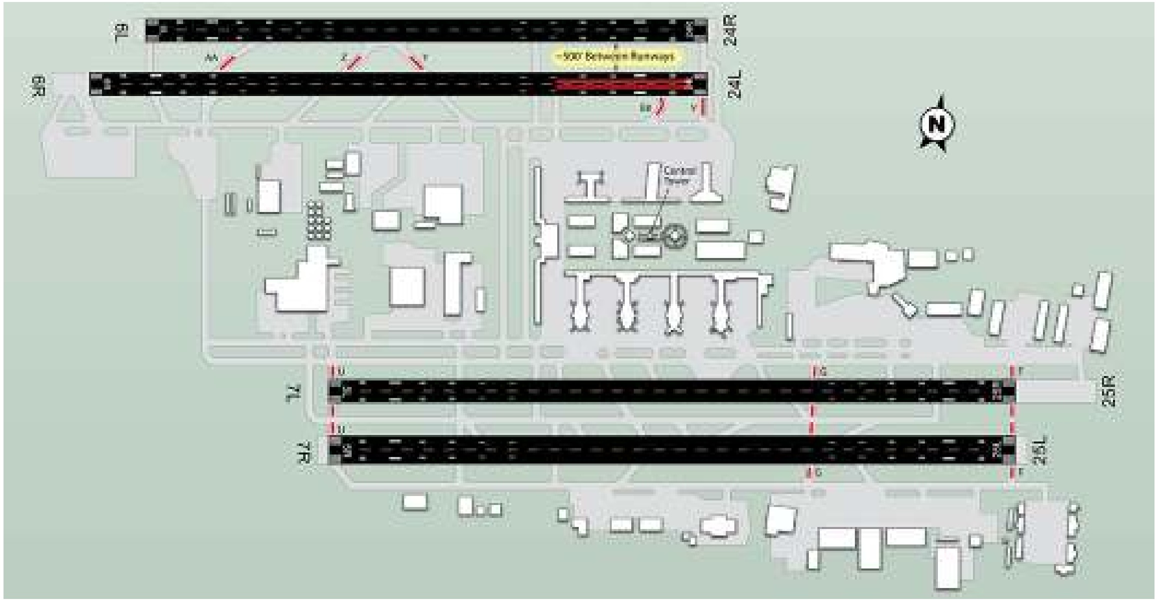


Figure 2: LAX Schematic

Often times, the pilot that is supposed to hold short doesn't realize where the crossing runway is and enters the runway without clearance. The sooner the runway entrance lights are activated, the better chance the pilot has of actually holding short of the crossing runway. In this example, ACID will become aware that the departing flight has takeoff clearance and as a result turn the runway entrance lights on before the plane even begins to move--a distinct improvement over ASDE-X, which turns on the lights only after it detects a significant amount of movement.

To simplify the scenario we will ignore any communications with arriving planes and focus on the processes in place to illuminate RELs. Figure 3 gives a component view of how ACID interfaces with the current set of airport technologies.

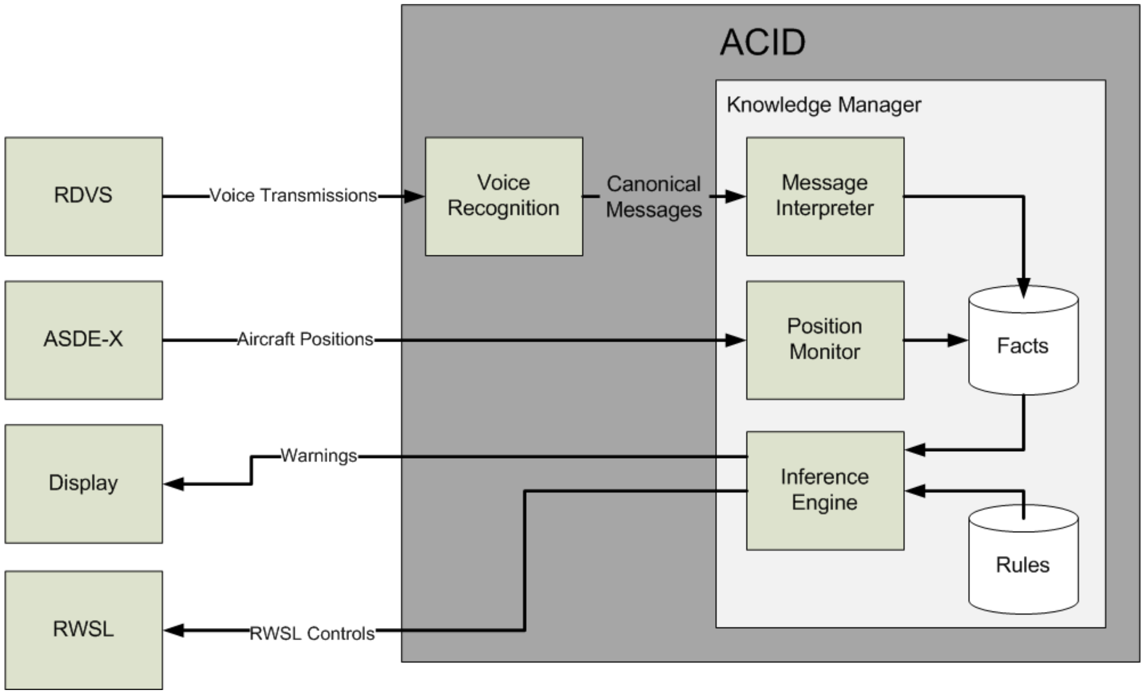
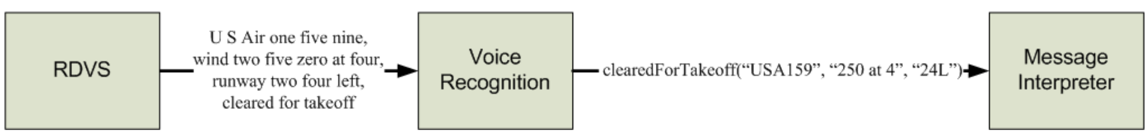


Figure 3: Component View of ACID

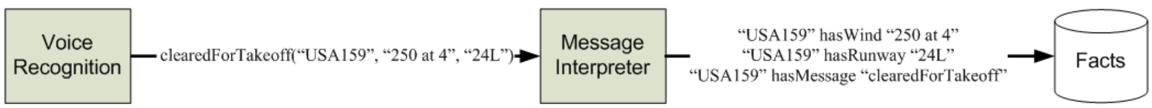
Step 1: The pilot of USAir 159 is given clearance for takeoff on runway 24L.



The controller's clearance of the departing flight is captured and parsed by Voice Recognition for known airport identifiers. For example, the term "two four left" is extracted due to the preceding identifier "runway." Each voice stream is parsed into a canonical message, which will then be processed by the Message Interpreter in step 2. In this case the clearedForTakeoff message contains parameters for flight number, wind, and designated runway. We referenced the phraseology section of the FAA's 7110.65S to create a formal set of canonical messages.

Step 2: Creating assertions from the canonical message.

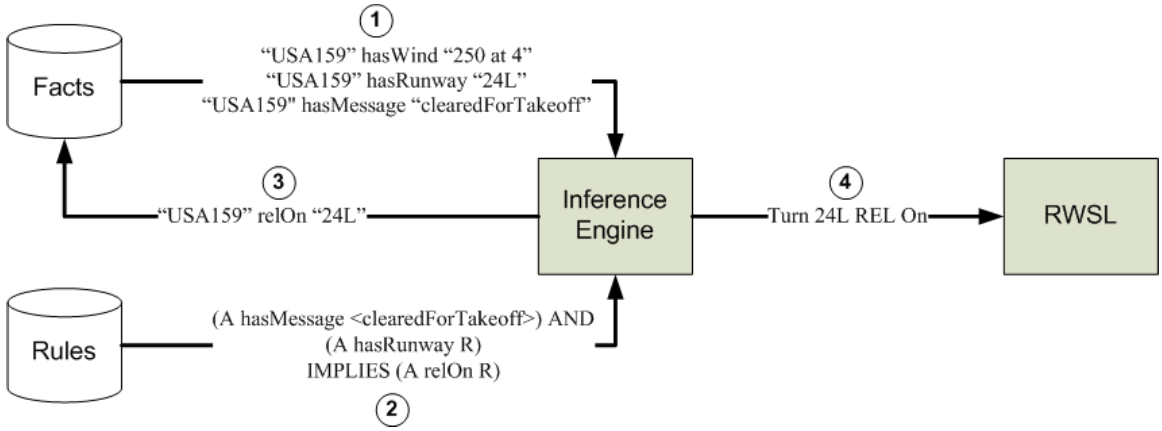
The message interpreter creates assertions about the airport environment from the information in the message. An assertion is a subject-predicate-object (SPO) statement that is stored in a knowledge base. Each canonical message is associated with a set of assertions.



As shown above, the clearedForTakeoff message is associated with three assertions: hasWind, hasRunway, and hasMessage along with their appropriate parameters. It is important to note that an assertion is created regarding the type of the last received message; these time-dependent assertions are used to make inferences about the plane's current and future location.

Step 3: Making inferences about newly added assertions.

The knowledge base uses rules to make inferences about the airport environment. A rule is a statement that links multiple assertions and entities together (the left-hand side) to add new assertions (the right-hand side) in the knowledge base. Every time new assertions are added, ACID's inference engine checks which rules "fire", i.e., those whose left-hand side matches leading to the new assertions that are in its right-hand side. In our example, such a rule is found. It will turn on the runway lights.



The rule states that if there is an airplane that has received a "clearedForTakeoff" message and this same plane is associated with a runway, then the RWSL system should be turned on for the given runway. Due to the existence of the hasMessage and hasRunway assertions about flight "US Air one five nine," a new statement regarding REL illumination is added to the knowledge base.

Step 4: Turning on the runway lights.

The knowledge base is continuously checked for statements of this nature, and ACID notifies the ASDE-X system when the relOn assertion is found. Our proposed interface to the ASDE-X system is a modified version of the current ASDE-X end user interface, discussed in the FAA's Interface Control Document Number 790-010712 Version 5. The modified interface would allow ACID to remotely turn on the runway entrance lights (REL) for runway 24L before ASDE-X recognizes the departing airplane.

4.2. Incursion Risks

The mitigation of runway incursions stemming from operational errors is another goal of ACID. Again, the scenario at the LAX airport involving an arrival and a departure will be used to show the complexity and scalability of ACID's safety logic. In this case, the controller will clear the arriving aircraft to cross the inner runway rather than issuing a hold short. The controller has made an operational error that could result in a runway incursion when the departing aircraft is simultaneously given clearance for takeoff. In this example, the controller will be alerted by ACID as soon as the operational error takes place allowing the controller to mitigate the imminent runway incursion.

Step 1: Canonical messages are formed from speech.

The controller first clears the arrival to cross the inner runway then immediately clears the departure for takeoff. This only requires the addition of a single line of dialogue to the preceding example, but results in a much more complex system behavior.

Local Controller: "Southwest One Eleven, at the reverse high-speed, cross Runway Two Four Left"

Local Controller: "U S Air one five nine, wind two five zero at four, runway Two Four Left, cleared for takeoff"

The two statements above are parsed and canonical messages are formed to encompass their information. The first line of dialogue results in the creation of a new message of type crossRunway with data regarding the arrival's clearance to cross runway "Two Four Left" at taxiway "reverse high-speed." The second line of dialogue is parsed in the exact same way as in the previous example, creating a message of type clearedForTakeoff with flight, wind, and runway parameters.

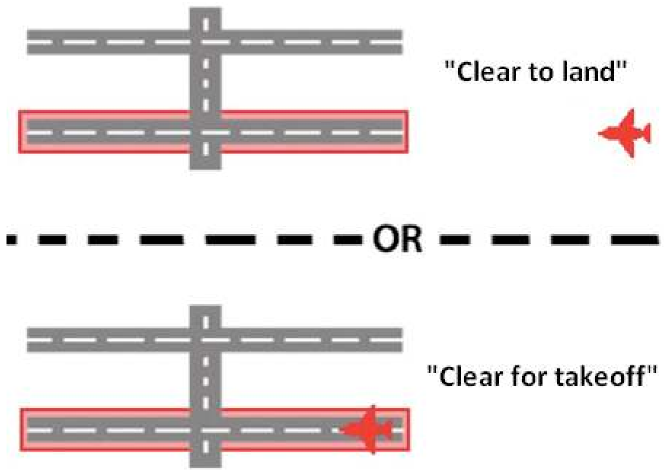
Step 2: Creating assertions from the canonical message.

These two canonical messages are then turned into sets of assertions for the knowledge base. The crossRunway message is mapped to a set of assertions about flight Southwest 111 with predicates "hasCrossRunway" and "hasCrossTaxiway." Again, the clearedForTakeoff message maps to the exact same assertions as it did in the runway lights scenario. The two sets of assertions are inserted into the knowledge base.

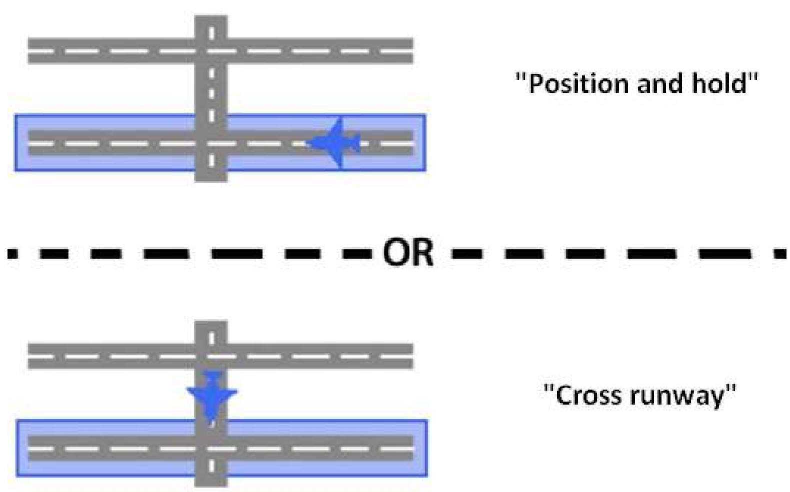
Step 3: Making inferences about newly added assertions.

In this example a set of rules regarding incursions are used. These rules need more explanation than the runway lights rule since they are designed to be abstract enough to encompass all runway incursions.

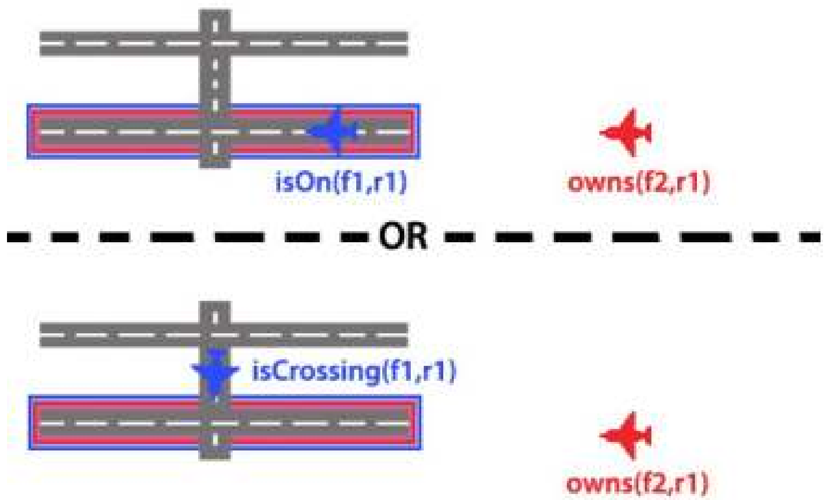
To describe the different ways a runway incursion could take place, we first went about decomposing the definition of runway incursion. For an incursion to take place, a plane must be arriving or departing on a runway. We abstracted this idea to develop the concept of "owning" a runway. An airplane "owns" a runway when it is arriving or departing on that runway. The "owns" rule is fired when an airplane has been associated with a runway and the last message it has received is either "clearedToLand" or "clearedForTakeoff."



A runway incursion takes place when a second vehicle is present on an “owned” runway. During the research phase, we classified two situations where a second vehicle’s presence led to a runway incursion between the two vehicles - the presence of a vehicle on the “owned” runway, and the presence of a vehicle on a runway or taxiway that will cross the “owned” runway. The vehicle in this case is classified as “isOn” the “owned” runway. An airplane “isOn” a runway if the coordinates from ASDE-X are on the runway.



With these definitions, we produced a basic rule about runway incursions. The incursion rule states that if an airplane “owns” a runway, and a second distinct vehicle “isOn” the same runway, both vehicles are “incursingOn” the “owned” runway.



Returning to our example, the arriving plane has received the crossRunway message with the value “Two Four Left.” This fires the “isOn” rule for this airplane. The departing plane has received the clearedForTakeoff message with the runway “Two Four Left.” and the “owns” rule fires. Once this assertion is added the knowledge base, the “incursingOn” rule fires and both airplanes receive the predicate “incursingOn” with the object “Two Four Left” in the knowledge base.

These abstractions of the incursion definition are integral to our safety logic, but must be augmented by ASDE-X coordinates so that alerts are not superfluous, i.e. the arriving plane may be far enough away to not present an incursion danger.

4.3. Closed Runways

Even though a runway is closed, controllers accidentally let aircraft takeoff or land on that closed runway. We can easily model a runway closure and then could listen for:

- Runway XX, position and hold.
- Runway XX, cleared for takeoff.
- Runway XX, cleared to land.

Any of these transmissions, when detected, could signal an alert. This is very similar to disallowing any flights from being on a runway when another flight owns that runway.

4.4 Other components of ACID

4.4.1 Airport Builder

The Airport Builder is used by an engineer to describe an airport to ACID so that a rendering like that of the LAX airport shown in Figure xxx and its attendant model can be added to the knowledge base. The airport description is stored as part of the configuration data. Professor Wilczynski had previous FAA teams actually design such a builder and because he wanted us to focus on the speech analysis part of ACID, we did no design of the builder.

4.4.2 Logging

Similarly, it is obvious that ACID can log information about what it is doing. A Log Viewer can provide a human interface to search through the Configuration and Log Database in order to pull out desired information. Such logging systems are critically important, but somewhat orthogonal to our design, so we have ignored it in this document.

4.5. Software Engineering

Our team used a simplified version of USC Prof. Barry Boehm's Model Based Architecting and Software Engineering (MBASE) approach to designing our system. In MBASE we first define an Operational Concept Description (OCD) for the project. This includes organizational goals, identification of key stakeholders, and the enumeration of the system's capabilities. Next a full set of requirements, both functional and non-functional, was produced along with any external interface requirements. Finally, a design was produced in which the system's architecture is expanded upon and system services elaborated. Of course, each phase was reviewed by our faculty mentor and industry experts.

In our case we identified the key stakeholders to be the FAA and the LAX airport administration. A year ago, Professor Wilczynski first broached the voice recognition idea to Mr. Tony DiBernardo, LAX Support Manager. He was highly supportive of this idea and thought it would make a good project. The design was first meant to be wholly separate from ASDE-X and RWSL, but after multiple meetings with the LAX administrators the ACID design became integrated with current technologies in hopes of making its implementation as fluid as possible. A prototype was produced and showed to the LAX staff, and the final design has been shown to be effective at providing alerts to controllers regarding communication based incursion risks.

5. Safety Risk Assessment

LAX currently uses following systems to ensure runway safety:

- ASDE-X (Airport Surface Detection Equipment, Model X) – tracks plane velocity, type, position and activates Runway Safety Lights system
- Airport Radar Tracking System (ARTS) – determines distance from airport of plane that's on course to land, also tracks plane type
- Runway Safety Lights(RWSL) – lights along runways to alert crossing planes and vehicles
- Rapid Deployment Voice Switching (RDVS) – intercom system ATC uses

ACID was designed to integrate with ASDE-X and RWSL, and furthermore, ACID was designed to require minimal ATC involvement thus limited distraction from ATC's normal tasks. Risk analysis has been performed on ACID to demonstrate ACID's compliance with LAX's existing integrated runway safety system. Sections below discuss four potential failures in ACID: failure to activate RWSL, incorrect activation of RWSL, failure to generate incursion alert, incorrect incursion alert. In each of the following section, a specific failure is discussed, including its impact on LAX's runway safety, and potential solution to each failure.

1. In the event of ACID's failure to activate RWSL, ASDE-X is still capable of activating the RWSL. When this failure occurs, LAX runway safety changes from being pro-active to reactive, which is what LAX currently has. The integration of ACID does not stop ASDE-X from activating the RWSL, but instead, ACID activates RWSL before ASDE-X does. Therefore, when ACID fails to activate the RWSL,

ASDE-X becomes the primary RWSL activation controller. This specific malfunction does not introduce any unexpected or additional safety risks, as ASDE-X takes over automatically without ATC interaction.

2. In the event of ACID's incorrect activation of RWSL, ATC interaction is required, as there is no automatic fall back from this malfunction. The ATC would have to first shutdown ACID and then reset RWSL. Since RWSL is a preventive measure, the incorrect activation of RWSL does not introduce additional safety risks because incorrectly stopping runway crossing traffic does not cause any safety issue, though the airport efficiency will decline during repair.
3. In the event of ACID's failure to activate incursion alert, ATC's normal tasks are minimally affected because ACID was not designed to replace any of the existing runway safety systems nor taking over any of the ATC tasks, but instead being an aid to the ATC and to generate early incursion alerts. This malfunction causes minimal safety risk because it is assumed that LAX runway should still operate correctly without ACID.
4. In the event of ACID's incorrect activation of incursion alerts, ATC is distracted, and recovery from this malfunction is for the ATC to shutdown ACID and resume his or her tasks. Similar to situation discussion in previous section, this malfunction in ACID does not introduce additional safety risks.

In conclusion, even though ACID is not fail-proof, the four identified failures do not introduce additional safety risks at LAX or other airports with less technology.

Voice Recognition Issues

ACID relies heavily on the accuracy of its voice recognition system, a technology known for its inaccuracy. Realize that we do not plan to build a voice recognition system capable of perfect transcription; instead we plan to license a state-of-the-art voice recognition system. Microsoft and Google, among others, with an enormous amount of resources are working very hard on the accuracy problem. The Open Source community is also making a strong effort with Carnegie Mellon University's Sphinx (which is what we used in our prototype) (*CMU Sphinx – Speech Recognition Toolkit*). It is widely accepted that the demand for *good* voice recognition technology will continue to increase as more and more uses become apparent. As a result, it is safe to assume that voice recognition will continue to improve, even to the point in which it is safe to use in such a high risk environment.

In our discussions with LAX personnel, we were made aware of speech recognition work done by Adacel. They built an ATC trainer that received voice from an ATC trainee and responded with simulated pilots. This kind of application is clearly valuable, but Adacel's implementation failed in operation because the voice recognition part of their system was too inaccurate. They had to replace the simulation with an actual person.

We recognize that state-of-the-art in speech recognition technology cannot be put in critical operation. However, as we have mentioned, there are powerful players driven to make this technology mainstream.

6. Technical Details about Our Prototype

In the design section of this report, two scenarios were described. In this section we explore some of the technical details that were implemented in our prototype.

6.1. Prototype with off the shelf components

Our prototype made use open systems that had large communities. This made it possible to implement a highly complex system in the time allotted. Our prototype featured a knowledge base and voice recognition, both of which were implemented with off the shelf components.

JENA implements the Semantic Web framework in Java. Our Subject-Predicate-Object triples are the foundation of its Resource Description Framework (RDF). JENA provided all of the functionally that one would expect from a knowledge base system (assertions, inferences, etc.).

SPHINX, CMU's voice recognition toolkit. Sphinx takes the specification of a grammar and allowed us to quickly recognize key airport phrases.

6.2. Language Model

The language model describes sets of words that can occur in the speech and also defines message protocol – order in which words will occur. The Language model used in the system is in Java Speech Grammar Format (JSGF), a standard BNF notation that allows optional words and skipping words that are of no interest to the system. This improves chances of recognizing ambiguous speech input. The grammar for the example in section 4.2 is defined by the following BNF:

```
<runway> = Runway <digit> (<digit>) (Right | Left)
<flight_id> = (K L M | Lufthansa | Southwest | U S Air) <digit> <digit>
               (<digit>) (<digit>)
public <crossRunway> = <flight_id> at <taxiway> cross <runway>
public <clearedForTakeoff> = <flight_id> (<wind>) <runway>, cleared for takeoff
```

Here the keyword 'public' indicates that the BNF will be reported to the user as opposed to other BNF that are used only internally. Following the public keyword is the name that will be used to create a canonical message for the knowledge base. Definition can include other rules (indicated by angle brackets) or plain text. The "|" symbol divides alternatives. Internal BNF used in this example are:

- <flight_id> - matches input to a flight id of the form: 'Airline name' followed by a two, three, or four digit number. Example: Southwest One Eleven
- <taxiway> - matches taxiway name or type. Depends on airport configuration. Example: reverse high-speed.
- <runway> - matches the word 'runway' followed by runway number and type. Depends on airport configuration. Example: Runway Two Four Left
- <wind> - optional rule (surrounded by brackets) that matches description of wind conditions. Example: wind two five zero at four

If an input signal matches any of this BNF, it is reported to the system along with set of tags that represent parameters given in the message. So messages in example 4.2 will be transformed into the following canonical messages and sent to the knowledge base:

```
crossRunway("SWA111", "reverse high-speed", "24L")
```

```
clearedForTakeoff("USA159", "250 at 4", "24L")
```

If some of the parameters are optional and were not found in the input, the system will report these parameters as `NULL`. Having optional parameters or skipping parts of the messages that do not bring any useful information to the system contributes to higher chances of recognizing messages and allows deviations in the message protocol.

JSGF grammar also allows assigning probabilities of rules occurring at the given time period. This allows the speech recognition engine to receive feedback from either ASDE-X or the knowledge base. It can assign higher probability of recognizing flight ids that are currently at the airport. The system has to include all airline names while searching for best fit, and eliminating most of them will improve speed and quality of recognition. It can also assign higher chance of recognizing messages based on current position in the dialog.

6.3. Message Interpreter – Asserting Facts from Canonical Messages

The Message Interpreter uses message definitions to transform a canonical message into assertions. The message definitions language is straightforward and best explained by looking at the examples from our prototype. In the following two examples, the first line names the canonical message and its parameters, while the body (the text between the braces) leads to the set of assertions generated:

```
crossRunway(flight, taxiway, runway) {
      hasID(F, flight);/* find the flight obj with id flight; assign it to F */
      hasFlight(A, F);/* find the arrival obj with flight F; assign it to A */
      hasCrossTaxiway(A, taxiway); /* make this assertion */
      hasCrossRunway(A, runway); /* make this assertion */
      hasMessage(A, "crossRunway"); /* make this assertion */
}
```

```
clearedForTakeoff(flight, wind, runway) {
     hasID(F, flight); /* find the flight obj with id flight; assign it to F */
     hasFlight(D, F); /* find the arrival obj with flight F; assign it to D */
     hasWind(D, wind); /* make this assertion */
     hasRunway(D, runway); /* make this assertion */
     hasMessage(D, "clearedForTakeoff"); /* make this assertion */
}
```

'A' corresponds to the Arrival flight and 'D' corresponds to the Departure flight. Next we show two examples of canonical messages and the facts (in Subject-Predicate-Object format) that are asserted in the knowledge base due to their respective message definitions.

```
crossRunway("USA159", "reverse high-speed", "24L"):

     A hasCrossTaxiway "reverse high-speed"
     A hasCrossRunway "24L"
     A hasMessage "crossRunway"

clearedForTakeoff("USA159", "250 at 4", "24L"):

     D hasWind "250 at 4"
     D hasRunway "24L"
     D hasMessage "clearedForTakeoff"
```

By using the message definitions, the system is able to capture all of the knowledge that can be gained from any one single message. In section 6.5, the system will make inferences on all of the facts currently known to the knowledge base in order to learn more about the current state of the airport.

6.4. Position Monitor – Additional Facts from Aircraft Positions

The Position Monitor interfaces with ASDE-X to monitor the position of each aircraft. Since we were unable to obtain the specification of the interface of ASDE-X, we made the following assumptions about the interface:

- ASDE-X can output a list of all known aircraft positions along with their flight names (e.g. "SWA111 X=152.5 Y=75.3")
- ASDE-X can answer queries about whether or not an aircraft is on a specific

runway or taxiway

- ASDE-X can answer queries about whether or not an aircraft is within landing range or takeoff range (i.e. the range at which no other aircraft should be on the runway)

In the case that ASDE-X doesn't support one or more of these features, it is feasible to acquire the necessary data by other means. Having visited the LAX control tower, we know that the current technology at the airport has all of the required information. Given that a flight is within landing range, the Position Monitor can make an assertion to the knowledge base about it (e.g. inLandingRange(A, R), which would mean that the flight with arrival object A is in landing range of runway R). The Position Monitor is also responsible for asserting when a flight is on a runway or taxiway (e.g. asdexOn(A, W), which means that the flight corresponding to object A is on way W). In section 6.5, the system will make inferences on all of the facts currently known to the knowledge base in order to learn more about the current state of the airport.

6.5. Inference Engine - Making Inferences with Inference Rules

In order to make inferences about the existence of incursion risks and whether or not runway entrance lights should be on, our system utilizes a set of inference rules. An inference rule is a statement that identifies when a certain fact is true. For example, if you know that one flight owns a runway (see 4.2) and another flight is on that same runway, then there is an incursion risk. This is the definition of an incursion and can be represented by the following inference rule in JENA's general purpose rule format (*JENA 2 Inference Support*):

```
(?A owns ?R), (?B isOn ?R), notEqual(?A, ?B)
```

```
-> (?A incursionOn ?R), (?B incursionOn ?R)
```

In English this rule says, “If there exists a flight A that owns runway R, and there exists a flight B that is on runway R, and flight A is not the same flight as flight B, then there is an incursion risk for both flight A and flight B on runway R.” In an inference rule, variables (?A, ?R, ?B) refer to any object that satisfies each of the statements that the variable occurs in. This means that with one rule, we can detect simultaneous incursion risks at the airport. This also means that we need to be careful when defining the owns rule and the isOn rule because it only makes sense for a flight to own a runway and not some other thing. Given that we can correctly infer owns and isOn, you can see how we can detect incursion risks. Here are all the rules in our prototype:

```
(?A hasMessage <clearForTakeoff>), (?A hasRunway ?R), (?A inTakeoffRange ?R)
-> (?A owns ?R)
```

“If there exists a flight A that has received takeoff clearance for runway R, and flight A is in takeoff range of runway R, then flight A owns runway R.”

```
(?A hasMessage <clearToLand>), (?A hasRunway ?R), (?A inLandingRange ?R)
-> (?A owns ?R)
```

“If there exists a flight A that has received landing clearance for runway R, and flight A is in landing range of runway R, then flight A owns runway R.”

```
(?A owns ?R) -> (?A runwayEntranceLightsOn ?R)
```

“If a flight A owns runway R, activate the runway entrance lights to alert pilots that they must not go onto runway R.”

```
(?A asdexOn ?W) -> (?A isOn ?W)
```

"If ASDE-X believes that flight A is on way W, then so do we."

```
(?A owns ?R) -> (?A isOn ?R)
```

"If a flight A owns runway R, then treat it as also being on runway R."

```
(?A hasMessage <crossRunway>), (?A hasCrossRunway ?R), (?A isOn ?W),
(?W crosses ?R)
-> (?A isOn ?R)
```

"If flight A has clearance to cross runway R and it's currently on way W which crosses runway R, then treat it as also being on runway R."

```
(?A owns ?R1), (?R1 crosses ?R2)
-> (?A on ?R2)
```

"If flight A owns runway R1 and runway R1 crosses runway R2, then treat flight A as being on runway R2"

7. Interactions with Industry and Operators

7.1. Interactions with FAA Personnel

Over the course of the design and prototyping process, the team maintained contact with key personnel from the Federal Aviation Administration stationed at Los Angeles International Airport.

Our team first visited the LAX control tower in fall 2009. As part of our visit we were given a tour of the facilities, including a trip to the ATC operations room and the balcony outside. After the tour, Mr. Larry Sweeney, an FAA employee, gave a presentation to us about the general flow of operations at LAX. Among other things, we learned about how traffic is handled on the four runways at LAX, what the ATC training process is like, and the types and frequencies of safety incidents that occur every year at LAX.

The information we gathered in this session gave us a "real world" to model in our design and prototype and force our interface to be as usable and intuitive as possible. Moreover, after spending a substantial amount of time in the ATC room atop the tower, the team realized that several key changes had to be made to the proposed ATC user interface. In short, the lessons we learned from this first onsite meeting allowed us to develop a polished functional prototype.

In March of 2010, the team once again visited LAX to present a working prototype of our system. As before, we benefitted substantially from the input and feedback of Mr.

Herbert T. King, an ATC quality control supervisor and Ms. Sherry Avery, the ATC manager at LAX. For example, Mr. King pointed out three things:

1. Some of the incursions we detected with our "owns" and "ison" rules are superfluous if the arriving flight is far enough away. That caused us to modify our rule to consider location data as well.

2. If we could integrate our system with Lincoln Labs' Runway Safety Lights (RWSL), then upper level management both at LAX and within the FAA would be quite interested in our solution. He told us that as soon as clearance is given to a plane, the runway lights can be turned on.

3. Finally, he alerted us to the closed runway scenario.

Furthermore, after demonstrating the speech recognition technology of our software, both Ms. Avery and Mr. King seemed excited about the potential immediate benefits our system could bring.

An additional example of the valuable feedback we received from Mr. King and Ms. Avery involved the design of our user interface for the ATCs. After proposing several possible designs of the touch screen device we plan to install in the control tower, they encouraged us to select a design with both auditory and visual runway incursion alerts for maximum safety.

In summary, upon seeing our system in action, both Mr. King and Ms. Avery offered several constructive criticisms that ultimately guided us to make several key design decisions for our project.

8. Projected Impact of Design

Runway incursions are caused by human errors and are hard to prevent. They can lead to a high number of casualties and loss of money. Current technological solutions, such as ASDE-X and RWSL, assist in preventing many situations when an incursion can occur, but their potential is limited without the information from the dialog between pilots and controllers. Our proposed design will complement existing solutions by filling the information gap with the data from the communication.

The system targets air traffic controller errors by checking that given instructions will not result in an incursion and is capable of indentifying pilot's mistakes by ensuring that aircraft's position corresponds to the given instructions. This accounts for approximately 85% of runway incursions. Therefore, successful installation of the system may significantly decrease the number of runway incursions.

8.1. Incursions that could have been prevented

The FAA safety report mentions two Type A incursions that happened at the Chicago O'Hare and Fort Lauderdale/Hollywood airports. The first incursion involved giving a clearance to takeoff of an airplane on runway 4L, and approximately 30 seconds later, clearance to takeoff to an airplane on an intersecting runway. The controller noticed the risk only after the planes started the takeoff procedure, so the pilots had to take serious measures to avoid the collision. Closest proximity was 100 feet. Our system would automatically identify a runway incursion risk right after the second pilot was given a clearance to takeoff. This would give the controller enough time to issue a position-and-hold instruction and avoid the incursion risk. In the second situation, an A320 missed a turn to a taxiway and ended up on runway without proper clearance. A Boeing 757 was about to land on that runway. B757 was issued a go around instruction and over flew the A320 by only 50 feet. Although our current design does not include rules that will check if the pilot is on a wrong runway, the system will illuminate Runway Entrance Lights (part of the RWSL system) on so that the A320 would see that the runway is unsafe to cross.

8.2. Meeting the FAA Requirements

The design meets the FAA's goals for improving runway safety. Our system expands situational awareness of the pilots and improves airports visual aids by operating the RWSL more efficiently. It gives direct warnings to the air traffic controllers in situations that may lead to a runway incursion. Warnings are given ahead of time, while it is still possible to mitigate the situation easily.

8.3. The system is affordable and easy to install

Our design is software-based which gives it many benefits:

- Ease of installation – the system does not require much new hardware and will integrate into current solutions seamlessly.
- Ease of maintaining – the system allows simple adapting to different airports. Updating the system to new requirements is also simple and can be done remotely.

- Low cost and affordability compared to the current solutions.

8.4. Implementation Schedule

The time line for design to maintenance is approximately five years, and it will complete in 2015 as shown below in figure 1:

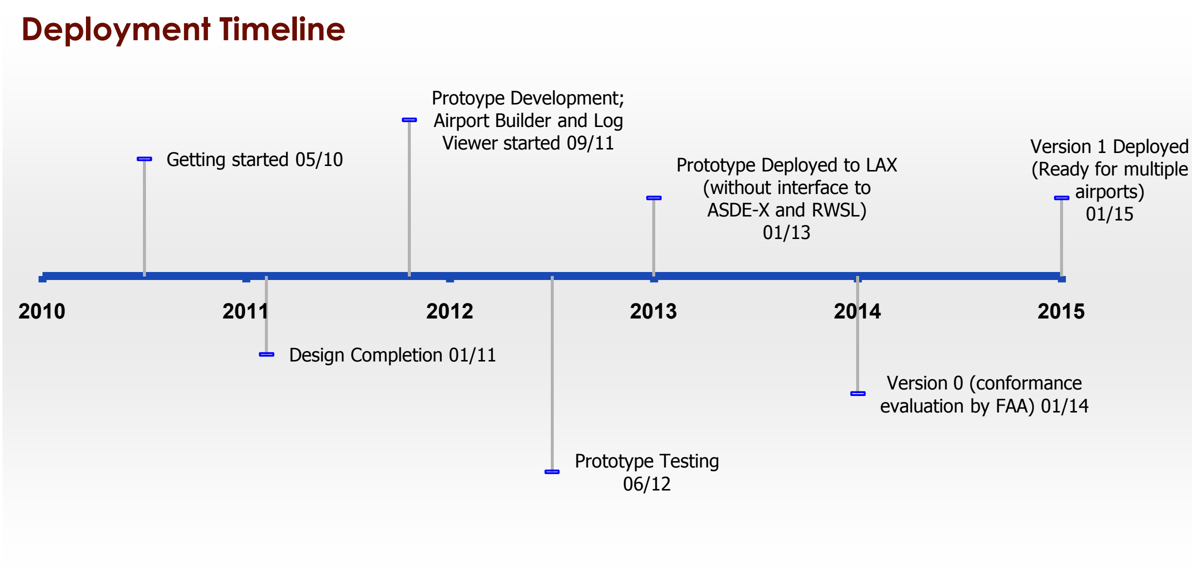


Figure 1: Deployment Timeline

8.5. Cost Estimation:

As discussed above the final implementation of this project is purely affordable because proposed system will work with current technology. But as like any new technology's design and implementation, there are some apparent and hidden charges are attached. In this section we will give the approximate estimation of explicit cost of different phases of this Knowledge base system.

For smart estimation of cost we have to consider many things like number of employs and their salaries, the use of devices, computers, software and their prices, office space etc. In this regard table 1 gives a rough estimate of cost of different phases of process like design to development, development to installation and installation to operational and operational to maintenance.

Table 1: Financial summary:

2010- 2013	Requirement	Qty	Annual Cost	Total Cost
Phase I: (2010-2013)	***Software Design & Development***			
	Project Manager	1	$118, 710 (mean wage)	$356,130
	Software Engineer, applications	1	$87,900 (mean wage)	$263,700
	Engineer, system software	1	$94,520 (mean wage)	$283,560
	Programmer	1	$73,470 (mean wage)	$220,410
	Drafter/Designer	1	$47,290	$141,870
	Software, Computers & servers etc (Fixed Cost)		$100,000 approx.	$100,000 approx.
Total Cost				**$1,365,670**
Phase II: (2013-2014)	***Implementation & Evaluation***			
	Project Manager	1	$118, 710 (mean wage)	$118,710
	Software Engineer, applications	1	$87,900 (mean wage)	$87,900
	Engineer, system software	1	$94,520 (mean wage)	$94,520
	Database & system administrator	1	$72,900 (mean wage)	$72,900
	Voice Recognition system	1	$1,199	$1,199

	Installation cost			$ 200,000
Total Cost				$575,229
Phase III: (2014-2015)	***Improvement & Maintenance***			
	Maintenance			$300,000 approx.
Final Cost				**$2,240,899**

Table 1 clearly shows that our knowledge base system is not only useful and efficient but very much affordable for FAA. However in above estimations the charges which are not countable and cant approximate at this stage like electricity, telephone, traveling etc are not included. We end on this note, with this low cost and useful system FAA can increase the potential time to resist the risk for pilot and controller and this will ultimately decrease the pressure and work load on them which will definitely show the positive results in their performances.

Appendix A – Contact Information

Prof. David Wilczynski
Phone: (310) 372-2332
E-Mail Address: dwilczyn@usc.edu
Permanent Address:
1637 Gates Ave.
Manhattan Beach, CA 90266

Ryan Berti
E-Mail Address: ryanbertiwork@gmail.com

Ryan Brown
E-Mail Address: ryanbrow@usc.edu, ryan@ryanleebrown.com

Tommy Holford
E-Mail Address: tholford@usc.edu

Michael Hsu
E-Mail Address: hsu2@usc.edu

Zhongyuan Li
E-Mail Address: zhongyul@usc.edu

Farid Nobakht
E-Mail Address: fnobakht@usc.edu

Denis Tulskiy
E-Mail Address: tulskiy@gmail.com, tulskiy@usc.edu

Appendix B – University of Southern California

From Wikipedia:

The University of Southern California (commonly referred to as USC) is a private, nonsectarian, research university located in the University Park neighborhood in Los Angeles, California, USA. USC was founded in 1880, making it California's oldest private research university.

The university enrolled 16,384 undergraduate and 17,024 graduate students and awarded 4,676 bachelor's and 5,380 advanced degrees in 2007. USC's four year, full-time undergraduate program is classified as "more selective, higher transfer-in" by the Carnegie Foundation and was ranked 26th among national universities by U.S. News & World Report, which classified it as one of the "most selective universities" for admitting 21% of the 35,809 who applied for freshman admission in 2008. According to the 2007 freshman profile, 18% of admissions were associated with legacy preferences USC was also named "College of the Year 2000" by the editors of Time and The Princeton Review for the university's extensive community-service programs. USC students hail from all 50 states in the United States as well as over 115 countries.

USC employed 3,127 full-time faculty, 1,363 part-time faculty, and about 8,200 staff members in 2007. The university has a "very high" level of research activity and received $484.6 million in sponsored research in 2007. USC is home to two National Science Foundation–funded Engineering Research Centers: the Integrated Media Systems

Center and the Center for Biomimetic Microelectronic Systems. 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.

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 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. USC and its partner institutions have recently completed or soon will be constructing 27 new buildings, which will provide nearly 8.1 million square feet (750,000 m^2) of new space for research, teaching, patient care, and student life enrichment.

Appendix C – Non-University Partners

None.

Appendix E – Educational Experience

Ryan Berti's Experience

Did the Competition provide a meaningful learning experience for you? Why or why not?

Yes, this competition allowed our group to work in a delivery-oriented manner very similar to what we will see in industry. The research I took part in was exciting in its applicability and the designing and prototyping of such a complex system was a rewarding experience.

What challenges did you encounter in undertaking the Competition? How did you overcome them?

Working with a large team to do research, development and presentations was inevitably a tedious process as coordination among the group was at times lacking. To overcome this, we used multiple different systems to assign tasks and keep each other updated including a wiki, Google wave, and e-mail.

Describe the process you used for developing your hypothesis.

The problem was outlined to us by Professor Wilczynski, but we were given free reign on how we would solve it. We referenced airport safety documents to better understand the frequency of communication-based incursions, and this played a large role in our formation of the hypothesis that if voice can be captured and analyzed runway incursions can be reduced.

Was participation by industry in the project appropriate, meaningful and useful? Why or why not?

Industry participation was not only appropriate and meaningful, it drove our project to success.

What did you learn? Did this project help you with skills and knowledge you need to be successful for entry in the workforce or to pursue further study? Why or why not?

I learned how to manage the disparate stages of system development, a skill that I believe will be pertinent once I enter the workforce. I also gained knowledge about the software design process through the successful integration of well-designed software components.

Ryan Brown's Experience

Did the Competition provide a meaningful learning experience for you? Why or why not?

The competition was one of the most meaningful experiences in my entire four years at USC. Having interned at Microsoft, this project was one of the only projects that felt like the real world. In designing our solution we went through most of the same processes that I went through at Microsoft. Since I plan on working in the industry (instead of staying in school), this experience has immediate benefits.

What challenges did you encounter in undertaking the Competition? How did you overcome them?

I would say that half of the challenges were technical, and the other half were not. The technical challenges were overcome simply by iterating and evolving our design. The non-technical challenges (which were mostly due to team-dynamics) were more difficult. Teams are hard to get right in school because there is no screening process; random people are thrown with other random people. When half of the people are just trying to skate by, it puts a lot of stress on those who really care about making great software.

Describe the process you used for developing your hypothesis.

Our Professor told us in the beginning that we needed to utilize the voice transmissions to improve the safety of airports. We got to the heart of the issue and gathered the requirements for such a system.

Was participation by industry in the project appropriate, meaningful and useful? Why or why not?

I enjoyed the industry participation. It acted as a guide to keep our project meaningful. Without it, I don’t think we would have known exactly how our system could help.

What did you learn? Did this project help you with skills and knowledge you need to be successful for entry in the workforce or to pursue further study? Why or why not?

I think that learning first-hand about voice recognition systems it extremely important as it will be one of the most widely used technologies in the near future. Knowledge bases will probably be used less often but it is still useful to have another *solution* in my toolbox.

Tommy Holford's Experience

Without a doubt, the FAA Design Competition provided a meaningful learning experience for me. Although the academic environment is a great place to learn about new concepts and ideas, nothing has prepared me for work in the "real world" quite like this project.

One of the biggest challenges our team faced was communication. In short, it was sometimes difficult to coordinate meetings among a group with a wide variety of schedules, particularly because we used too many forms of communication - email, texting, Google Wave, and a project wiki. As a result, sometimes important information ended up being scattered and fragmented around in different places.

To develop our hypothesis, the team visited the LAX air traffic control tower to observe real time airport operations and spent many hours researching FAA safety data. Combined with information we gathered from operators and employees from within the air safety industry and from the FAA, our team was able to synthesize data from the myriad sources to formulate our guiding hypothesis.

I wholeheartedly believe that industry participation is not only appropriate and useful but also absolutely necessary. The industry contacts that we were in touch with throughout this process proved to be invaluable sources of information and ultimately influenced significantly the final iteration of our project design.

I learned a great deal from this project. As I mentioned earlier, the "real world" experience gained from this project is impossible to receive inside a traditional classroom setting. Coordination of meetings, performing industry research, and meeting project deadlines were three valuable skills the team was able to practice in preparation for our future careers.

Michael Hsu's Experience

Did the Competition provide a meaningful learning experience for you? Why or why not?

Yes, I have learned a lot from my participation in the FAA Design Competition. As our team worked on this project, I was able to both visually and conceptually connect and apply all of my software engineering training in making a real world design. Even though a lot of examples were given to us during software engineering classes, working on a real world problem and interacting with working professionals gave me a sense of responsibility and urgency.

What challenges did you encounter in undertaking the Competition? How did you overcome them?

One challenge we faced was the amount of information, both relevant and irrelevant, that were available during our early phase research. Especially at the early phase when I was not very familiar with runway incursion, it was very challenging to process all those information. Fortunately, our first of the two LAX visits gave me a chance to meet and chat with FAA and LAX professionals, and they helped clarified some of my misconceptions.

Describe the process you used for developing your hypothesis.

Our team is composed of CS students whom all have prior design and programming in creating a multi-threaded airport simulator. With that said, we started with a very simple, modular based incursion defection design, and its design was improved as we gave three presentations to our class. However, our design idea did not solidify until we began putting together a prototype. Our prototype design got its overhaul when we made our second visit to LAX, which we received valuable input about which design ideas were valid and which were irrelevant.

Was participation by industry in the project appropriate, meaningful and useful? Why or why not?

Our team visited LAX tower twice and both visits were meaningful. On our first visit to LAX tower gave me a visual context in understand the flow of ATC communication. On top of an airport simulation project we did in our CSCI 201 class which we learned about and then implemented ATC communication, seeing an actual controller performing airport traffic management definitely enhanced my experience. Furthermore, during our second LAX tower visit, we met and shared our design draft with two FAA professionals, and they provided valuable feedback on our draft ideas. One thing that's for sure is, without those two LAX visits, our design would take longer to finish because during early design stage, we were overwhelmed by informations that we later found irrelevant to our design.

What did you learn? Did this project help you with skills and knowledge you need to be successful for entry in the workforce or to pursue further study? Why or why not?

From this project, I learned that working as a professional is far more serious than regular school work. I realized that a design mistake may no longer be a simple simulator crash, and an oversight can be very costly in real work application. When performing risk analysis for our prototype, I learned that a student's work mentality of "good enough" is just not good enough, and when time permits, safety and risk analysis should be double, even triple, checked.

Zhongyuan Li's Experience

FAA Design Competition has provided me a meaningful learning experience. Through this design, I learned teamwork. Different team members are assigned to complete different tasks and cooperated with each other to merge separate works together. This is the part where we learned how important teamwork is and negotiation among members gave everyone a chance to learn the problem one more time and see in a different perspective. During undertaking of the Competition, we run into problem like inaccurate voice recognition. We overcame this by integrating a more sophisticated algorithm when receiving input. Our team developed the hypothesis through extensive research, we have did a lot of related research on such incidents and by comparing the similar situation of these incidents we concluded the hypothesis of our design. FAA has showed great help and support on this project and we have had a visit to the LAX tower to experience the real environment. The most important thing I learned from this project is that design is far more important than the code itself. Communication between stakeholders and clients could be very essential to the result of the project and misunderstanding between the two parties could cause huge loss for both of them. Then after understanding the problem, a proper time line and task distribution are also very important. I learned more non-technical knowledge in this project and it will definitely help me in terms of career.

Farid Nobakht's Experience

Did the Competition provide a meaningful learning experience for you? Why or why not?

Certainly it was a meaningful learning experience as competition is something that takes out the best in competitors to set and achieve goals and make standards. This design competition not only realized me, my best skill areas but also polished my analytical skills. It gives me a fruitful experience to work on a high profile project for a giant organization like FAA and also taught me the value and importance of team work and effective teambuilding. Last but not least it trained me about time management and working under pressure.

What challenges did you encounter in undertaking the Competition? How did you overcome them?

The first challenge was the how to select a distinct and creative idea. This matter solved by asking every member to come up with an inventive proposal, so that we can select a best one. Secondly a challenge was effective work distribution, this issue handled by our adviser and he assigned every member a task according to his/her expertly and interest, which resulted in an efficient and successful teamwork. One of the most challenging aspects for me was time management to achieve my targets and I overcome it with choosing small targets.

Describe the process you used for developing your hypothesis.

First thing was to select a project design, in this regard FAA’s objectives and targets helped us in true sense because it gave us the right direction. Keeping in mind FAA’s perception about runway incursions we select the proposal of Knowledge Based system. Then our adviser assigned the tasks according to skills and expertly of each member.

Was participation by industry in the project appropriate, meaningful and useful? Why or why not?

Industry plays a very positive role for the development of this project. Just on the base of literature review and research we could not develop such a fine project. Industry participation gave us the practical information to refine system of this project.

What did you learn? Did this project help you with skills and knowledge you need to be successful for entry in the workforce or to pursue further study? Why or why not?

It gives me an opportunity to learn, how to apply newly attained skills in consequential and productive ways. It also broad my thinking to develop and maintain a new idea, which is definitely require in practical life. Hence this competition gave me productive field experience in my education life which will help me to pursue successfully my practical life. Furthermore, this experience also improved my writing and analytical skills which will help me in further studies.

Denis Tulskiy's Experience

Did the FAA Design Competition provide a meaningful learning experience for you? Why or why not?

Yes, it was important to know what systems and methods are currently used to prevent runway incursions. I learned a lot about how airports are operated and it was interesting to study latest technological solutions. Also, working in a team and designing a real system gave me great experience.

What challenges did you and/or your team encounter in undertaking the Competition? How did you overcome them?

Main challenge was designing the system that would be flexible and would allow simple integration with existing systems. We tried to provide the design with pluggable interface. Designing logic for identifying incursion risks was challenging, because it had to work for different airport setups. Speech recognition field was also new to us, so researching it and understanding ways to make it reliable took some time.

Describe the process you or your team used for developing your hypothesis.

I prefer to test my ideas directly with code. I wrote sample code to test speech recognition and team members enhanced it and turned into a working prototype. We worked with specialists from LAX to identify messages that our system will target. We designed the knowledge base part by analyzing reports about previous runway incursions.

Was participation by industry in the project appropriate, meaningful and useful? Why or why not?

Help from LAX was very useful. They gave us great comments about our design and motivated us to continue the development.

What did you learn? Did this project help you with skills and knowledge you need to be successful for entry in the workforce or to pursue further study? Why or why not?

Systems that we worked with gave me great knowledge that I hope to use in the future. Speech recognition is widely used today. Knowledge base paradigm is used in business applications. So knowledge about how these systems work will be advantageous.

Professor's Experience

I have taught the capstone design course for many years. Since 2007, we have entered the FAA competition. That project (and others in the class) have always had a high degree of sophistication to them. The runway incursion project is large, serious, and hard—the focus of significant research efforts. Finding an approach was a challenge in and of itself. In the past we have had ideas from Delta Airlines, the head of USC Airlines safety, and this year the LAX control tower itself gave our idea the go-ahead.

Though I helped organize the teams and helped develop the design, I never looked at any code or knowledge base work directly. I tried to play the role of a non-technical manager. Obviously, I got involved when help was needed, but mostly I listened at our meetings, which by the way are often held during class time. Students have problems with team projects because scheduling meetings is difficult; their class schedules are so different. I used many class times for just this purpose. The meetings unified the class, for sure; students heard what others were up to.

I make class attendance "mandatory" (without taking role or punishing absence). How can you have meetings if people don't show up? I learned (again) that the student's documentation skills are weak. This is no surprise, especially in engineering. Though our engineering students take writing classes, this is a skill that takes years to master, and most don't. Most of their problems fall into three categories:

- Making sure the industry terminology was precise.
- Keeping the text from getting so technical that non-engineers would have trouble understanding it.
- Keeping the context of the writing clear.
- And make the wording simple and clear.

My colleague, Prof. Michael Crowley, and I talk often about the problem of how to make students take pride in their work. It's difficult when the work is usually small school problems. This project is different. The look on the students faces when presenting their prototype told it all. They were beaming, especially the lead programmer, Ryan Brown. His smile was worth the whole project. In summary, the experience works for me. I look forward to the next competition.

Appendix F – References

(1999). *Airport Surface Detection Equipment, Model X.* Retrieved from http://www.sensis.com/docs/128/

(1996). *ASN Airport Safety Database.* Retrieved from http://aviation-safety.net/database/

(2010). *CMU Sphinx – Speech Recognition Toolkit*. http://cmusphinx.sourceforge.net/

(1997). *Federal Aviation Administration's Runway Incursion Program*. Retrieved from http://www.johnsonaviation.com/downloads.htm

(2010). *JENA 2 Inference Support*. http://jena.sourceforge.net/inference/

(2003). *Live ATC.net Live Air Traffic – From Their Headsets To You.* Retrieved from http://www.liveatc.net/

(2008). *Order JO 7110.65S Air Traffic Control*. Retrieved from www.faa.gov/documentLibrary/media/Order/7110.65S.pdf

(2004). *Runway Safety Lights System.* Retrieved from http://rwsl.ll.mit.edu/

(2004). *Runway Safety Report*. Retrieved from http://www.faa.gov/airports/runway_safety/media/pdf/report4.pdf

(2005). *Runway Safety Report*. Retrieved from http://www.faa.gov/airports/runway_safety/media/pdf/report5.pdf

(2006). *Runway Safety Report*. Retrieved from http://www.faa.gov/airports/runway_safety/media/pdf/rireport06.pdf

(2008). Runway Safety Report. Retrieved from http://www.faa.gov/airports/runway_safety/media/pdf/RSreport08.pdf

(2009). *Runway Safety Report*. Retrieved from
http://www.faa.gov/airports/runway_safety/media/pdf/Annual_Runway_Safety_Report_2009.pdf