NextGen Air Traffic Control Display System

Bespectacled Builders

Elena Corpus, Lucas Iverson

CSCI 363 - User Interface Design

December 11, 2019

**Abstract** This project will provide a baseline of what to expect from our version of the NextGen Air Traffic Display System. This new system allows for anticipated adaptations for different airports along with allowing for the controller to personalize to ensure for their most comfortable view. This aims to provide a simple screen yet provides all the information needed for the controller along with ensuring that it is able to be used safely over many different monitors including touchscreen. We strive to provide the best services and software for our customers and thus building a simple yet adaptive Air Traffic Control Display System.

1. Introduction

Being awarded a contract from the United States Federal Aviation Administration (FAA) to develop a new graphical user interface for the Next Generation (NextGen) Air Traffic Control (ATC) system, our team began the process of analyzing the requirements, designing the system, and creating a prototype to demonstrate to the stakeholders of the FAA project team. This project had a timeline of approximately three months, culminating in a demonstration of our design. Our project objectives guided us through many choices. These objectives were to keep the interface relatively simple, yet flexible and to focus on the needs of the end user, the Air Traffic Controller. With these basic objectives, we began the task of fulfilling the expectations and requirements put upon us.

1. Background

To fully understand where we needed to take our project, understanding where the FAA was coming from was an important step to ensure we aligned with the intentions of the contract. To do this, we learned about the large umbrella of the FAA’s project to improve air travel. The Next Generation, or NextGen as designated by the FAA, is “a wide-ranging series of improvements that will transform the air transportation system, encompassing new air traffic management technologies and procedures; airport infrastructure improvements; and environmental, safety and security-related enhancements.” [1] Previously implemented systems relied on ground based radar systems fed to viewing stations monitored by controllers. These crude radar displays, while relatively accurate, only provided minimal data for the controllers to understand the situation in their designated airspace. With NextGen, many systems have been and will continue to be upgraded with new technologies, improved methodology, and greater efficiencies. For example, a very important development has been to the TCAS system, which has been utilized in the United States “for more than 20 years.” In which, TCAS “has provided a last line of defense.” [2] To summarize, TCAS was originally an in cockpit alarm system, warning pilots of a danger, and no other details were provided. Over the years, it has developed into an automated system, so much so that TCAS can automatically guide aircraft away from each other without pilot intervention. This system also notifies air traffic controllers of imminent danger and provides details as to what is happening in real time. NextGen as an organization (FAA) wide project continues to improve upon all the varying aspects of air travel. Our project focuses specifically on the user experience centered around the Air Traffic Control system, namely for the air traffic controllers. Some of these systems have been implemented already and our goal is to continue advancing these improvements to ensure that air travel continues to be statistically one of the safest modes of travel. Just for reference, an article on Fortune.com analyzed information from the National Highway Traffic Safety Administration and the National Transportation Safety Board to find that there is “a 1 in 114 chance of dying in a car crash” and a “1 in 9,821” chance of dying in air transport. [3]

1. **The Project**

During these last few months we were tasked with the opportunity to create an Air Traffic Control Display System, that would be able to simulate the radar where the Air Traffic Controller would be able to observe airplanes that come into the airspace of the tower within a 50 mile radius. In order to begin this task, the team had to meet to determine how we wanted to begin this task along with how we intend on ensuring that our interface was parallel with what the contractor wanted. The best way to begin with programming this system was using Microsoft Visual Studio. This software allowed us multiple, diverse tools that helped with the design and ensured that our interface is user friendly. However, before even beginning this project, we were able to use multiple different resources, including some of what the contractor provided before and during the project construction time to help begin building the foundation of our interface.

While building the foundation for the interface, we were able to use multiple different diagrams in order to better prepare ourselves with how we wanted our system to run. We were able to use the Use Case diagram to organize the functionality of our system, by using the controller and the actions that this system would need to do in order for it to function the way the controller needed it too. In addition to the Use Case diagram, we used an Activity diagram in order to provide the control flow from start to finish and including all the decision paths that exists while the system is running. With this diagram we were able to learn how each of our steps should be implemented and how we wanted the timing to work during our demonstration for the contractor. However, before putting together any of these other diagrams, we first had to build a Class Diagram to build a foundation of the system and what we were wanting to include within it.

When designing out interface during our first stages, our main goal is to allow for the system to be adaptable, and simple. With some general knowledge about Air Traffic Controllers, we knew that this was a stressful job with lots of screen time, the controllers needed to be able to watch the screen in order to allow for a smooth running air space for work, and travel. Aiming for a simple but adaptable design we opted for two separate screens in preparation for dual views so the user is always able to have the screen view even if they decided to change a few settings, request reports, or view alert messages. However, in order for the screen to be adaptable we have minimal options on the screen as to leave room for necessary components the user would like to have, like larger messages, weather updates, etc. Also in preparations for updated hardware, we wanted to ensure that our software would be able to not only work on regular monitors, but for touch screens as well, thus we allowed for larger buttons and more space in between so the likelihood of accidentally pressing the wrong thing was reduced.

Over the course of the few months we were working on this project, we were faced with multiple problems. However, the primary issues that were encountered were programming related including learning how to use Visual Studios and to program in C#. Both were solved quite quickly, leading to running multiple test runs of our program to ensure it was up to standard with what the contractor was asking. In addition to learning, we had to make multiple decisions about how we were wanting to display the airplanes along with how to make the radar move in a circle.

Ultimately, learning this new language and software made the task of building this system difficult yet enjoyable. However, we wanted to include that post-presentation, our largest concern if we were to implement this system worldwide, is rather than just testing this system on a few different systems, we would rather test on more to ensure that our system would run smoothly. This system has the potential to hold multiple lives in its hands, thus having the errors that took place during the presentation would be unacceptable, thus providing us with feedback that showed the team we always needed to make sure that if there were minor differences between software we would want to ensure our system ran smoothly on all updates.

1. **Conclusion**

In a project such as this one, there is often a lot to be learned. In this case, we not only learned a great amount in the programming phase, but also the sales side of things as well. We were exposed to many new things in building this system, especially using C# and Visual Studio and determining the best way to have our airplanes move around the screen. In addition to programming, we discovered how to utilize different UML diagrams to help determine and build the foundation for our system. Thus proving these diagrams to be more useful than first understood.

From the aspect of sales, we found that gathering and getting a cursory understanding of the history of the systems we were expected to improve was a huge benefit. Not having direct access to the target user, the controllers in this case, did hinder our understanding of which features and options would initially be included. However, after discussing how to approach this lack of concrete information, we returned to one of our main objectives and decided to rely on flexibility, allowing for future modifications and customizations. We learned that we may need to open to the possibility that designs and plans need to account for open-ended solutions where there is not a concrete answer. Creativity in the end result is definitely a strong possibility.

Ultimately, this project taught us that there are many approaches to solving challenges. While requirements may appear to be exact when first encountered, there are several ways to interpret even the most absolute statements. Following concepts and processes that guide the development of user experiences, these variations in interpretation become much more manageable. Ensuring acceptance, and even appreciation, of new interfaces depends a lot on working with and/or studying the behaviors of the end users. Understanding people and what they want (or *think* they want) can play a larger role than understanding the programming language and coding to accomplish the goal.

Bibliography

1. James T. Eck, *Update to the Business Case for the Next Generation Air Transportation system based on the Future of the NAS Report* PDF Report found at https://www.faa.gov/nextgen/media/BusinessCaseForNextGen-2016.pdf*,* July 2016
2. Unknown, *“Performance Success Stories”* <https://www.faa.gov/nextgen/snapshots/stories/?slide=27>, December 2019
3. Aric Jenkins, *“Which is Safer: Airplanes or Cars?”* <https://fortune.com/2017/07/20/are-airplanes-safer-than-cars/>*,* July 20, 2017

|  |  |
| --- | --- |
| Class Diagram | Activity Diagram |
|  |  |
| Use Case Diagram | |
|  | |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |