ote technical note technical

Concept Development and Design Description of Electronic Flight Data Interfaces for Airport Traffic Control Towers

Todd Truitt, Ph.D., Human Factors Team – Atlantic City, ATO-P

December 2006

DOT/FAA/TC-TN-06/17

U.S. Department of Transportation
Federal Aviation Administration

William J. Hughes Technical Center
Atlantic City International Airport, NJ 08405

NOTICE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT/FAA/TC-TN-06/17		

4. Title and Subtitle	5. Report Date
Concept Development and Design Description of Electronic Flight Data Interfaces for Airport Traffic Control Towers	December 2006
	6. Performing Organization Code AJP-6110
7. Author(s) Todd Truitt, Ph.D., Human Factors Team – Atlantic City, ATO-P	**8. Performing Organization Report No.** DOT/FAA/TC-TN-06/17
9. Performing Organization Name and Address Federal Aviation Administration Human Factors Team – Atlantic City, ATO-P William J. Hughes Technical Center Atlantic City International Airport, NJ 08405	**10. Work Unit No. (TRAIS)** **11. Contract or Grant No.**
12. Sponsoring Agency Name and Address Federal Aviation Administration Human Factors Research and Engineering Group 800 Independence Avenue, S.W. Washington, DC 20591	**13. Type of Report and Period Covered** Technical Note **14. Sponsoring Agency Code** ATO-P

15. Supplementary Notes

16. Abstract

This report documents and describes the development process, design rationale, and design description for two prototype Electronic Flight Data Interfaces (EFDIs) for an Airport Traffic Control Tower (ATCT). The author designed the EFDIs as part of a concept research program to examine the feasibility of using Electronic Flight Data (EFD) in an ATCT instead of paper Flight Progress Strips. The author designed the EFDIs based on literature review, working group and subject matter expert input, task analyses, low-risk usability tests, and a rapid prototype process. The Integrated EFDI incorporates EFD with the Airport Surface Detection Equipment - Model X (ASDE-X). The Perceptual-Spatial EFDI does not rely on ASDE-X, but presents EFD that controllers can arrange spatially on an airport surface map. Both EFDIs include separate displays for the local and ground controller positions and provide controllers with the ability to record, manage, and transfer flight data. The EFDIs will be used to automate some flight data management tasks, to provide new tools designed to reduce controller workload and improve safety, and to improve controller efficiency by integrating information. A provisional patent application is pending for the EFDIs.

17. Key Words	18. Distribution Statement
Airport Traffic Control Tower Electronic Flight Data Flight Progress Strips Interface Design	This document is available to the public through the National Technical Information Service, Springfield, Virginia, 22161. A copy is retained for reference at the William J. Hughes Technical Center Library.

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	49	

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized**

Acknowledgments

A very special thanks to the members of the Airport Traffic Control Tower (ATCT) Electronic Flight Data working group for their insight, expert advice, support, and patience: Steve Cooley, Federal Aviation Administration (FAA) Human Factors Lead for ATO-T; Ricky N. Dillbeck, Evansville Regional ATCT; Dr. Francis T. Durso, Texas Tech University; Ronald Hubbard, Traverse City ATCT; Robert A. Hutson, Milwaukee Mitchell ATCT; Dr. Carol A. Manning, FAA Civil Aero Medicine Institute; Robert Oliver, FAA William J. Hughes Technical Center; and Ric Wunn, FAA Quality Assurance Specialist. I would like to thank Jim McGhee and Joe Toland (Northrop Grumman IT) who served as Subject Matter Experts and members of the interface design team. Also, I would like to thank Dino Piccione (FAA ATO-P, Human Factors Research & Engineering Group) for his program support and for understanding the importance of this concept research. Thanks also to the technical leadership of Drs. Pamela Della Rocco and D. Michael McAnulty of the FAA Human Factors Team – Atlantic City. Many thanks to the expert software developers who dealt with the numerous requirements and constant time pressures associated with our rapid prototyping and development process: Nick Marzelli and Dallas Shiles, Barco/Orthogon as well as Valentina Velez and Joel Grochowski, L-3 Communications Titan Corporation. Special thanks are due to the FAA William J. Hughes Technical Center's Simulation Group for their many hours of effort to support the underlying aircraft target generation and ATCT simulation infrastructure: Albert Rehmann, Dan Warburton, Scott Doucett, Jim Hamilton, Jim Mauroff, George Chachis, and Samantha Fullerton. I would like to thank John Dilks and Ralph "Ed" Little (L-3 Communications Titan Corporation) and Albert Macias (FAA, Future Laboratory Development Group) for their excellent support and maintenance of the laboratory hardware and network infrastructure. Finally, a note of thanks to Linda Johnson, April Jackman, and Jean Dunn (Northrop Grumman IT) for processing this paper through technical editing and publication.

Table of Contents

List of Illustrations

Tables Page

Executive Summary

Projected increases in air traffic, along with modernization efforts, have led the Federal Aviation Administration (FAA) to consider replacing paper Flight Progress Strips (FPSs) with an electronic alternative. Electronic Flight Data (EFD) alternatives have the potential to increase a controller's ability to acquire, track, and record information as well as communicate and coordinate that information with others. More importantly, EFD presents the possibility of improving controller efficiency by providing new methods of flight data management that integrate information into a single source, reduce staffing requirements, and enhance safety.

Previous efforts at implementing EFD have focused on replicating the appearance and functions of FPSs without significant changes to the task itself. In this report, we present two alternative Electronic Flight Data Interface (EFDI) designs that consider the basic tasks and information needs of controllers working at Airport Traffic Control Towers (ATCTs). We designed the prototype EFDIs based on the existing literature, task analyses, low risk usability testing, and subject matter expert opinions without allowing historical artifacts to constrain our design. We designed the EFDIs to provide the right information to controllers at the right time and in accordance with a controller's mental model of the task. The first prototype interface, the Integrated EFDI, combines textual EFD with an airport surface situation display provided by Airport Surface Detection Equipment - Model X (ASDE-X) radar. The second prototype interface, the Perceptual-Spatial (P-S) EFDI, does not rely on ASDE-X radar capabilities, but combines textual EFD with an airport surface map. The P-S EFDI also functions as a backup flight data management system to the Integrated EFDI, if ASDE-X capabilities were to fail.

In this report, we present our prototype development process and decision rationale for both the Integrated and the P-S EFDIs. We also present design descriptions of each prototype interface to explain their respective functionality. We plan to submit both of the prototype EFDIs to usability testing in the near future. A provisional patent application is pending for the EFDIs.

1. INTRODUCTION

Airport operations logged by the 449 Federal Aviation Administration (FAA) Airport Traffic Control Towers (ATCTs) were projected to increase from 63.1 million in 2004 to 68.8 million in 2008 (FAA, 2005). In anticipation of the increase in air traffic, the FAA is investigating the potential effects of implementing an electronic flight data system (EFDS) in ATCTs. The EFDS would replace the paper Flight Progress Strips (FPSs) used by certified professional controllers (hereafter referred to simply as controllers) since their inception in the 1930s and 1940s. The FAA should consider an EFDS because the FPS has become an historical artifact that limits the usefulness of flight data and consumes valuable cognitive resources (Durso & Manning, 2002). In the Air Traffic Control (ATC) environment of today, controllers must manually update information, record clearances, and physically pass FPSs from one controller to another within the ATCT. All of these activities require cognitive and sensory resources that may be relieved by automation or other less subtle changes in standard operating procedures. The inherent physical limitations of FPSs also restrict the controllers' ability to communicate flight data information with other facilities such as the Terminal Radar Approach Control (TRACON), Air Route Traffic Control Center, and Airline Operations Center (AOC). Currently, controllers must perform most communication and coordination between the ATCT and other facilities via landline. In some instances, controllers can pass FPSs from the ATCT to the TRACON with a gravity-fed drop tube. However, with the modernization of FAA facilities and the advent of the Electronic Flight Strip Transfer System (EFSTS), drop tubes are becoming outdated. Bar code scanners located at the controllers' workstation and bar codes printed on each FPS enables the EFSTS. Although the EFSTS allows the electronic transfer of information between remote facilities, the EFSTS also has a number of limitations. The EFSTS requires the FAA to print duplicate FPSs in multiple locations, that is, the ATCT and TRACON. Changing or updating FPS information that controllers must pass between the ATCT and TRACON is also difficult or impossible with the EFSTS.

When considering the transition to an EFDS, one primary interest is preserving the current benefits of FPSs while enhancing the performance of air traffic controllers and the National Airspace System (NAS). To do so, we must understand the similarities and differences among ATCTs as well as all of the tasks involving FPSs, flight data, and the communication of information among controllers. Researchers can contribute to the success of an EFDS if they address some major gaps in the existing research by developing a better understanding of the actual cognitive benefits afforded by FPSs in the ATCT. System designers must then preserve or enhance these benefits with an EFDS. Researchers must also address long-standing organizational norms during the design process to ease the transition from FPSs to an EFDS. Given the proper design of the interface and automation, an EFDS should maintain some of the basic functionality and benefits of the FPSs. It should reduce workload related to flight data entry, tracking, and sharing, and provide new features that will enhance controller performance and encourage usage.

The aesthetic quality of the overall EFDS, especially the user interface, will influence initial user acceptability. The user interface must also be helpful and easy to operate, and it must not create new problems or reintroduce old problems. This is not to say that any proposed EFDS must look and act exactly like the current system, just in an electronic form. We must be willing to conduct concept research to explore new ways of handling air traffic and its associated flight data. If the predicted increase in the level of air traffic actually occurs, then the FAA must find ways to help controllers keep pace. One approach for system designers is to make a few minor adjustments to the overall system without changing the fundamental way that work is accomplished. Such an approach appears to be without risk. However, there is the risk that without fundamental changes to the ATC task, controller workload may become unbearable and prove a limiting factor to the overall NAS performance. New approaches must be attempted with all of the risks, including user acceptance, in mind.

2. SCOPE

This report presents the research and development work involved in the creation of two prototype Electronic Flight Data Interfaces (EFDIs) for the FAA ATCTs. The scope of this report focuses on the work we conducted during the year 2005. This report describes only the initial design of the prototype EFDIs; this report does not reflect any changes resulting from subsequent usability testing.

We limited the scope of the prototype EFDIs' development and functionality to common events rather than attempting to account for all possible functions that an EFDS might encompass. We limited the scope of the EFDIs' functionality to simplify the task so that we could examine the basic concept of using Electronic Flight Data (EFD) in an ATCT.

3. PURPOSE

The purpose of this report is to document the processes and methodologies used to create the initial prototype EFDIs. This report also presents functional design descriptions of the prototype EFDIs.

4. LITERATURE REVIEW

The current project began during 2004 with a literature review (Truitt, 2006). The literature review examined prior task analyses, published literature, and field observation data to explore the basic functionality of FPSs and flight data in ATCTs. Surprisingly, researchers have conducted very little work during the past 20 years in the ATCT domain, especially in the United States. Some of the earliest relevant research was conducted by CTA, Inc. (Alexander et al., 1989; Ammerman, Becker, Jones, Tobey, & Phillips, 1987) who completed comprehensive job analyses of both the then current and future Advanced Automation System (AAS). The CTA, Inc. job analyses are still a valuable source of information today. Only two other field studies have provided relevant information and unique data regarding how controllers work in an ATCT; one was by Bruce (1996) and one was by Dattel, Johnson, Durso, Hackworth, and Manning (2005). Researchers in other countries have already expended considerable effort to examine how controllers work and to estimate the utility of EFD (e.g., Berndtsson & Normark, 1999; Mertz, Chatty, & Vinot, 2000; Mertz & Lecoanet, 1996; Mertz & Vinot, 1999; Pavet, 2001; Ryan, van Schyndel, & Kitchin, 2003). Unfortunately, these studies have lacked the data required to evaluate hypotheses regarding the cognitive effects of an EFDS in FAA ATCTs.

Even though the extent of prior research is limited, the literature review provided insights into the potential risks and benefits associated with EFD. A primary benefit of EFD is the ability to share and transfer important flight data such as clearance amendments, aircraft location on the airport surface, posting and updating expected departure clearance times (EDCTs), and wake turbulence warnings. Other benefits of EFD include the ability to (1) present the right flight data to the controller at the right time while still preserving the ability to access all information about a flight; (2) simplify or automate data input; (3) integrate with other systems; (4) reduce workload associated with flight data management; (5) integrate information sources to reduce visual shifts of attention; (6) increase time for controllers to observe the airport surface; (7) increase awareness of other controllers' actions through linked displays; (8) eliminate expenses associated with FPS papers, holders, and printers; and (9) automate recording and reporting of activity, such as number and duration of departure delays. Potential disadvantages of EFD include initial training, increased criticality of data entry, and changes to controller selection and training methods. We attempt to incorporate these findings into the present design of two prototype EFDIs.

5. WORKING GROUP

After completing the literature review, we formed a working group that included ATCT controllers, cognitive psychologists, and software developers. We formed this multidisciplinary team to provide information and ideas for the development of prototype EFDIs for an ATCT. The core members of the working group are representative of the users themselves – Certified Professional Controllers (CPCs). The controllers served as the task Subject Matter Experts (SMEs) on the working group. The SMEs were from Milwaukee Mitchell ATCT, Evansville Regional ATCT, Traverse City ATCT, and one was an FAA Quality Assurance Specialist in the ATCT domain. Cognitive psychologists from the Human Factors Team – Atlantic City, the Civil Aerospace Medical Institute, and Texas Tech University analyzed the ATCT task with consideration for the relevant human cognitive abilities and limitations. Software developers from the FAA William J. Hughes Technical Center's Future Laboratory Development Group helped to constrain the development and implementation of the prototype EFDIs within the laboratory's simulation capabilities. The research sponsor, representing human factors interests for the Air Traffic Organization-Terminal (ATO-T) office, was also a member of the working group. All members of the working group will contribute to the eventual usability testing and refinement of the prototype EFDIs.

The working group met June 7-9, 2005, at the FAA's Research, Development, & Human Factors Laboratory (RDHFL). At the beginning of the meeting, the working group members learned about the overall research objectives and the general direction of the research program. The purpose of the working group meeting was not to design a Graphical User Interface (GUI). Rather, the working group spent the three days discussing the task of ATCT controllers.

The working group began by defining the entire scope of the interfaces to be designed. There are a number of controller positions (i.e., functions) in the ATCT. The most commonly staffed positions are local, ground, and flight data/clearance delivery. The working group decided to limit the scope of this first project to the local and ground positions because these two positions are involved with aircraft and vehicles on the airport surface during safety-critical operations. Although the flight data/clearance delivery position is concerned with flight data, this function is performed before an aircraft even pushes back from the airport terminal gate. Therefore, by focusing on the local and ground positions, we would be able to have the largest impact on airport operations and the flow of information between controllers in operational positions.

Another initial decision made by the group was to use a generic airport as the simulated environment. For the purposes of concept research, a generic airport has a number of advantages and disadvantages compared to a real airport. First, the generic airport allows us to test a general concept without associating the results with a particular location. The generic airport eliminates any political affiliation between the data we gather and any particular facility. Collecting data based on a generic airport operation also prevents others from applying the results too narrowly by focusing on a known airport and its idiosyncrasies. A generic airport allows broader application of results because it gives us the flexibility to capture a number of prototypical elements contained in many real airports. Furthermore, it is much easier to select participant controllers for testing because they can be sampled from a large variety of facilities. Given that the generic airport should contain prototypical elements, the working group agreed to use Boston-Logan Airport (BOS) as a template to build upon because it contains both parallel and crossing runways. Furthermore, we could use previous work of the Airway Facilities Tower Integration Laboratory (AFTIL) which has already modeled BOS in their simulation platform including realistic out-the-window scenery including terminal buildings, taxiways, and runways.

Once the working group determined the scope and airport environment, we then implemented a modified version of The Bridge methodology (Dayton, McFarland, & Kramer, 1998) for user interface design. The Bridge provides a structured method for developing user interfaces. It advocates a multidisciplinary team including system users, cognitive psychologists, software developers, and a facilitator trained in the methodology. The first step in The Bridge is to define the scope of a task and document its task flow using low risk materials such as chart paper, sticky notes, and markers. The design team then redesigns the task flow into a "Blue Sky" task flow that streamlines the task and eliminates bottlenecks. The design team then identifies and categorizes task objects in the optimized task flow and identifies actions that users will have to take upon these task objects. The design team then translates the task objects and their associated actions into GUI objects. The GUI designs begin as paper prototypes created by the design team using chart paper, sticky notes, and markers. Throughout the process of creating the Blue Sky task flow and the paper prototype, the design team conducts usability testing to ensure that tasks are not omitted and that the GUI operates in a user friendly and logical way. Once the paper prototype is developed, then actual construction of the GUI begins and the design team can conduct further usability testing.

We modified The Bridge methodology for a number of reasons. First, none of the working group members were trained facilitators in using The Bridge. We conducted a pilot study using The Bridge with a facilitator who was not trained in the methodology. The pilot study showed that without an in-depth knowledge and formal training in The Bridge, the facilitator was not able to guide the group in a way that makes the methodology effective. Second, Dayton et al. (1998) designed The Bridge for rapid development of GUIs for nominal tasks. The task of ATC is a considerably more formidable task, much larger in scope than could be adequately addressed during the three days of the working group meeting. The high level of task complexity and the large number of different subtasks in an ATCT dictated that we modify the original methodology. Nevertheless, we used parts of The Bridge to provide structure for GUI development that began with understanding the tasks and task elements of ATCT controllers before attempting to design a GUI.

We began by examining the tasks of the ATCT controller. Typically in The Bridge methodology, participants start by generating a task flow describing their current task using markers, chart paper, and sticky notes. This activity using low risk materials is intended to encourage participants to understand and document the entire task flow fairly rapidly (i.e., within the first day). This would be an overwhelming activity for the ATC task even if the focus was solely on the ATCT or even one position in the ATCT. Rather than starting from nothing, we decided to examine the existing task flows of Alexander et al. (1989) and a predicted future

ATCT based on the AAS concept. The working group quickly discovered that these task flows, although almost 20 years old, are still highly accurate. The working group identified a number of tasks that appeared to be incomplete. We included extra information as needed, but this was the exception rather than the norm.

Our examination of the task flows focused on the most common activities for both the local and ground controller positions. For the local controller position, we examined the most relevant 4 of the 7 primary tasks and 24 of the 28 subtasks (see Table 1). For the ground controller position, we examined the most relevant 3 of the 6 primary tasks and 10 of the 17 subtasks (see Table 2).

Table 1. Local Controller Primary Tasks (T1.X) and Subtasks (T1.X.X)

Task Number	**Task Name**	**Examined**[a]
T1.1	Perform Local Situation Monitoring	√
T1.1.1	Establishing Positive Aircraft/Vehicle Identification	√
T1.1.2	Checking & Evaluating Separation	√
T1.1.3	Receiving Airport & System Equipment Status Info	
T1.1.4	Housekeeping	√
T1.2	Resolve Conflict Situations	√
T1.2.1	Performing Conflict Resolution	√
T1.2.2	Performing Minimum Safe-Altitude Resolution	√
T1.2.3	Performing Airspace/Movement Area Violation Resolution	√
T1.2.4	Issuing Unsafe Condition Advisories	√
T1.2.5	Suppressing/Restoring Alerts	√
T1.3	Manage Air Traffic Sequences	√
T1.3.1	Processing Deviations	√
T1.3.2	Issuing Departure Information/Instructions	√
T1.3.3	Issuing Arrival & Landing Information/Instructions	√
T1.3.4	Monitoring Non-Controlled Objects	
T1.3.5	Responding to Imposed Airspace/Movement Area Restrictions	√
T1.3.6	Requesting Temporary Release of Airspace/Movement Areas	√
T1.3.7	Responding to Requests for Temporary Release of Airspace/Movement Areas	√
T1.3.8	Responding to Runway/Taxiway Changes	√
T1.3.9	Managing Airborne Departures	√
T1.3.10	Managing Aircraft Takeoff Termination	√
T1.4	Route or Plan Flights	√
T1.4.1	Planning Clearances	√
T1.4.2	Responding to Special Conditions/Emergencies	
T1.4.3	Responding to Special Operations	
T1.4.4	Processing Flight Plan Amendments	√
T1.4.5	Responding to Requests for Transfer of Control	√
T1.4.6	Initiating Transfer of Control/Radar Identification	√
T1.4.7	Issuing Pointouts	√
T1.4.8	Responding to Pointouts	√
T1.4.9	Issuing Clearances	√
T1.5	Assess Weather Impact	
T1.6	Manage Local Controller Position Resources	
T1.7	Respond to System/Equipment Degradation	

[a] The check mark indicates that the working group examined the task.

Table 2. Ground Controller Primary Tasks (T2.X) and Subtasks (T2.X.X)

Task Number	Task Name	Examined[a]
T2.1	Perform Ground Situation Monitoring	√
T2.1.1	Establishing/Maintaining Positive Aircraft/Vehicle Identification	√
T2.1.2	Checking & Evaluating Traffic Movement	√
T2.1.3	Receiving Airport & System Equipment Status Information	
T2.1.4	Housekeeping	√
T2.2	Control Aircraft/Vehicle Ground Movement	√
T2.2.1	Responding to Flow Constraints	√
T2.2.2	Processing Ground Traffic Deviations	√
T2.2.3	Managing Departure Traffic	√
T2.2.4	Responding to Movement Area Closures/Reopening	
T2.2.5	Responding to Ground-Movement Requests	√
T2.2.6	Responding to Requests for Temporary Release of Movement Areas	√
T2.2.7	Responding to Runway/Taxiway Usage Changes	
T2.2.8	Monitoring Non-Controlled Objects	
T2.3	Route or Plan Flights	√
T2.3.1	Planning & Issuing Clearances	
T2.3.2	Responding to Special Conditions/Emergencies	
T2.3.3	Responding to Special Operations	
T2.3.4	Transferring Control Responsibilities – Departure Aircraft	√
T2.3.5	Observing Arrival Aircraft	√
T2.4	Assess Weather Impact	
T2.5	Manage Ground Controller Resources	
T2.6	Respond to System/Equipment Degradation	

[a] The check mark indicates that the working group examined the task.

After examining and amending the task flows for both the local and ground controller positions, we identified each of the task objects. We made a list of all the nouns contained in the task flows, which became the initial set of task objects. We reorganized this set by categorizing many of the task objects as object attributes – a subordinate category to a task object. We selected the higher-level task objects to correlate with the controllers' mental model of the task: EFDI, arrival aircraft, departure aircraft, and vehicles on the airport movement area. We categorized the remaining task objects as subordinate object attributes associated with one or more of the task objects. Perhaps not surprisingly, the remaining object attributes are comprised of essential flight data (e.g., aircraft call sign, aircraft type, first departure fix, proposed departure time, runway assignment, etc).

For both the local and ground controller positions, the EFDI is the highest-level task object. The EFDI contains all of the other task objects and their respective object attributes. The EFDI combines EFD with an aircraft and vehicle situation display of an airport movement area. The EFDI not only presents information, but it also provides for user access and input of flight data. Attributes of the EFDI object include the airport name, runway/taxiway layout, flight data, departure/arrival sequence list, reminders, and timers. Arrival aircraft, departure aircraft, and surface vehicles comprise the three task objects within the EFDI object. Each of these objects has a set of associated object attributes, although many of the attributes are common to all of the task objects within the EFDI.

For the arrival task object, attributes gleaned from the task flows include aircraft location, hold short clearance, ground speed, possession indicator, reminder, gate assignment, aircraft type, call sign, deviation/conflict indicator, and other flight data. The arrival object attributes were the same for both the local and ground controller positions.

The departing aircraft object contained more attributes than any other object. The local and ground controller positions shared most of the attributes for departing aircraft including destination/first fix/route, location on the surface, aircraft type, EDCT/delay information, number in sequence, runway assignment, hold short clearance and location, ground hold restrictions, Traffic Management Unit (TMU) restrictions, Automatic Terminal Information Service (ATIS) code, release time/timer, ground speed, proposed departure time, possession indicator, reminder, call sign, and deviation/conflict indicator. Altitude, heading, and Taxi-into-Position-and-Hold (TIPH) clearances were attributes of the departing aircraft object that were unique to the local controller position. Only taxi start time was unique to the ground controller position. For the surface vehicle object, we identified object attributes associated with both the local and ground controller positions including vehicle location, vehicle type, ground speed, possession indicator, reminder, call sign, deviation/conflict indicator, and hold short clearance.

The three task objects (arrival aircraft, departure aircraft, and surface vehicles) share a number of common attributes such as aircraft/vehicle type, location on airport surface, hold short clearance, ground speed, possession indicator, reminder, call sign, and deviation/conflict indicator. Once we identified all of the task objects and their respective attributes, we determined the common actions that controllers would need to act upon each of the task objects. The identified actions included view, move, edit, start, create, and change possession. These actions relate to all of the task objects and would be implemented in the GUI.

6. PROTOTYPE DEVELOPMENT

Once we understood the task flow and identified all the task objects, object attributes, and actions, we had the foundation of task requirements necessary to build any type of GUI. One possible approach was to design an EFDI that preserved the prominent features of a FPS such as the size and information format. Preserving the FPS in an electronic format is a reasonable approach because it seeks to take advantage of expertise that controllers have already developed. For example, preserving the format of information among different media should maintain a controller's ability to search for and find information. Rearranging the format of information, on the other hand, should make it more difficult for a controller to find information (at least until expertise for the new format is developed). Although the positive transfer of skill is desirable, preservation of old features can lead to the negative transfer of skill, especially when those features do not work as the user expects (Besnard & Cacitti, 2005).

A number of interface designers have already taken the approach to preserve historical features. In 1996, Eurocontrol constructed one of the first EFDS prototypes for Gardermoen Tower in Norway. Since then, many other countries including Brazil, Canada, and Germany have adopted some type of EFDSs. These countries developed their systems independently, but they are all similar in that they have retained many features of the FPS including the information format. Although the electronic medium could allow a highly customized and flexible interface, there is little difference between FPSs and fielded EFDIs.

An alternative approach is to redesign the task altogether. Although this approach requires the most change and does not allow much opportunity for the positive transfer of skill in terms of flight data management, it does present an opportunity to create new, possibly more efficient, methods of operation. Furthermore, relieving the design of an EFDI from historical precedents also allows the opportunity to integrate flight data information systems in the ATCT. According to FAA (2003), the implementation of numerous independent systems in the ATCT has led to equipment clutter, numerous displays, and inconsistent user interface designs. The FAA has implemented new systems in ATCTs without considering the impact on the controllers' task and cognitive task load. In fact, the current situation in ATCTs prompted the FAA Associate Administrator for Air Traffic Services to implement a policy on February 13, 2003, restricting the acquisition or implementation of new equipment in an ATCT. We elected to design a more efficient GUI and integrate it into the existing system.

6.1 Integrated Electronic Flight Data Interface

An initial suggestion for consolidating existing information displays came from the Air Traffic Terminal Enhancement and Modernization (ATEAM) group, a group of ATC experts who obtain user input on display integration. The ATEAM suggested that we integrate EFD with the ASDE-X display. Integrating EFD with the ASDE-X display makes some intuitive and logical sense in that it places flight data closer to aircraft positions. Instead of controllers (1) examining FPSs for flight data; (2) shifting their visual attention to the ASDE-X and out-the-window to verify aircraft position; and then (3) mentally correlating the disparate sources of information, the integration of EFD and the ASDE-X allows the controller to gain both flight data and aircraft position information from a single source. A system that combines related information into a well designed single source should improve controller efficiency by reducing the controllers' cognitive task load and need to shift visual attention.

Given the potential advantages of integrating information and the opportunity to discard the historical artifacts of FPSs, especially for large airports where ASDE or ASDE-X already exists, the first prototype EFDI we constructed, the Integrated EFDI, used ASDE-X as the foundation. We reduced the time to develop an ASDE-X simulation platform by using existing software code for the ASDE-X prototype tool provided by Barco/Orthogon. We translated the existing code into the RDHFL's highly flexible Distributed Environment for Simulation, Rapid Engineering, and Experimentation (DESIREE) ATC simulation platform. Once the simulated ASDE-X foundation was in place, we began mapping the task objects and attributes into GUI objects.

We formed an interface design team consisting of two ATCT SMEs, a lead software developer, and an engineering research psychologist to implement the GUI objects that would eventually comprise the Integrated EFDI. The software developer and psychologist were both members of the EFDI working group. The interface design team used the materials generated by the working group, including the lists of task objects and attributes for each controller position and the generic airport layout and runway configuration.

6.1.1 Integrated Electronic Flight Data Interface Hardware

The first step of designing the Integrated EFDI required the design team to select the interface hardware. This was an important decision because it determined how much area would be available to display both flight data and ASDE-X information. The team decided to use a VarTech Systems, Inc. touch sensitive display. The 21.3″ display has an active display area of 17″ (432 mm) wide and 12.75″ (324 mm) high with a 1600 x 1200 pixel format. The display has a viewing angle of 85 degrees. The display uses resistive technology to enable a touch screen that can be activated by either a stylus or by a person's fingertip. The fact that the display was touch sensitive did not limit us to using only touch as a means of interacting with the interface. The touch display provided many more possibilities than a standard display and in fact, it did not require us to use touch at all. We mounted the display on a stand that supports the weight of the 30.4 lb (13.8 kg) display and allows the user to adjust the horizontal and vertical viewing angle. The display stand was designed to be simple to operate and easy to manufacture. We also acquired an actual ASDE-X keyboard and a trackball/keypad as additional input devices. Figure 1 shows the hardware used to implement the first prototype Integrated EFDI.

Figure 1. Hardware used to implement the prototype Integrated EFDI including resistive touch display, display mount, ASDE-X keyboard, and trackball/keypad.

6.1.2 Integrated Electronic Flight Data Interface Design Description

The Integrated EFDI is actually two separate interfaces: one for the ground controller position and one for the local controller position. We designed slightly different interfaces for each position based on the respective tasks, the task objects, and the attributes identified by the working group. The local EFDI and the ground EFDI share many task objects, but also have some distinct differences. The shared task objects translate into shared GUI objects. The main shared GUI objects are the surface situation display, EFD lists, readout area, clock alarm window, and map buttons. Figures 2 and 3 show an overall view of the Integrated EFDI and its primary elements for the ground and local controller positions. The primary difference between the EFDIs is the type of lists that appear. The following sections present detailed descriptions of the shared and unique objects as well as attributes of the ground and local Integrated EFDIs.

Figure 2. The primary elements of the ground controller's Integrated EFDI including the surface situation display, EFD lists, and readout area.

Figure 3. The primary elements of the local controller's Integrated EFDI including the surface situation display, EFD lists, and readout area.

6.1.2.1 Electronic Flight Data Lists

The EFD lists are similar to FPS bays in that they contain individual flight data elements (FDEs) that can be stacked and sorted. Each FDE contains only the most important and relevant information, as identified by earlier design processes. The user can locate the EFD lists on either the left or right side of the surface situation display. Each list has a header that identifies the contents of the list. The list headers are also touch sensitive buttons that allow the user to move FDEs between lists, controller positions, and other facilities such as a TRACON or AOC. To move an FDE to another list or to transfer it to another position or facility, the controller selects the FDE to move and then selects the appropriate list header. The list headers provide visual feedback to the controller to indicate activation. Whenever an FDE appears in a list or the controller moves an FDE to a list using a header or button, the FDE appears at the top of the new list. The user can also resequence FDEs in a list by selecting and dragging an FDE. All flight data for an aircraft are associated such that selecting either an FDE or data block will cause all other related flight data to display highlighted in green (see Figure 4). For example, selecting an FDE will highlight the FDE and the associated data block. Likewise, selecting a data block will highlight the associated FDE.

Figure 4. A selected FDE in an EFD list and the selected data block for the same aircraft.

We implemented a noun-verb command style throughout the EFDIs. That is, the controller first selects an object to act upon (noun), and then selects an action to perform (verb). We decided to use the noun-verb command style to reduce errors, increase speed, and for simplicity and reversibility (Raskin, 2000). Apple Computer (1985), Hewlett-Packard (1987), IBM (1988), and Microsoft (1995) also recommend the noun-verb command style. The noun-verb command style reduces errors because commands take effect when the user issues them and the user is focusing attention on the command. In contrast, the verb-noun style makes the user chose a command and then executes that command as soon as the user makes a noun selection. Any interruption between the user's selection of a command and selection of what to act upon may redirect the user's attention and cause the user to forget the already selected command and that the system is waiting for the user to select an object. Likewise, the user may select the correct command but inadvertently select the wrong object. Thus, the verb-noun style creates the potential of applying the command to an unintended object. The preferred noun-verb style increases speed by minimizing the number of times that the user must refocus attention. The noun-verb style allows the user to select the object to act upon and then redirect attention to select the proper command; a single shift in the focus of attention. In contrast, the verb-noun style requires the user to select the object to act upon, redirect attention to the proper command, and then redirect attention back to the object to act upon; two shifts in the focus of attention. The verb-noun style also requires system designers to implement a feature that allows the user to cancel a command. Once the user selects a command in the verb-noun style and the user decides not to use that command, then the user must inform the system by canceling the selected command. Conversely, with the noun-verb style, if the user selects the wrong object, they just make another selection of the correct object.

Another primary design decision concerned the size (width and height) of the FDEs. The size of the FDEs, and all touch sensitive elements for that matter, is critical because controllers must be able to reliably select a number of different elements including an FDE, data block, header, or button. Touch sensitive displays have been in use for some time in relatively simple applications such as information kiosks and automated teller machines. Woodson, Tillman, & Tillman (1992) recommend that the actuation area (i.e., button) of a touch screen should be 0.63 in. (16 mm) to 1.5 in. (38 mm) in both width and height. They also suggest that interface designers separate the buttons from one another by 0.13 in. (3.3 mm) to 0.25 in. (6.4 mm). The FAA's Human Factors Design Standard (HFDS) (Ahlstrom & Longo, 2003) recommends that buttons be at least 0.75 in. (19 mm) and not more than 1.5 in. (38 mm) in height and width. The HFDS recommends the same button separation as Woodson et al. These standards for touch sensitive displays have been in place since at least 1989 (Department of Defense, 1999).

Although a number of design standards have been in use, researchers are currently gathering empirical evidence regarding the optimal button size on touch sensitive displays. Rogers, Fisk, McLaughlin, and Pak (2005) compared a touch screen (a direct input device) to a rotary encoder (an indirect input device) to examine how task demands and user age affected performance. Their first experiment showed that the task performed and the age of the user interacted to determine the optimal input device. For pointing tasks, Rogers et al. recommend a direct device such as a stylus, light pen, or fingertip. This finding supports the research of Charness, Holley, Feddon, and Jastrzembski (2004), who found that participants performed better at a pointing task when they used a light pen (i.e., compared to when they used a mouse). Charness et al. also found that the light pen helped younger users, but was particularly helpful for older users during a pointing task.

In a second experiment, Rogers et al. (2005) manipulated button size and button separation along with a number of other factors. Again, task demands and user age moderated performance. While accounting for errors, they found that participants responded faster to buttons without space between them (i.e., stacked buttons) compared to buttons that were separated from one another. Although these results are not in agreement with Woodson et al. (1992) or the HFDS (Ahlstrom & Longo, 2003), the superiority of stacked buttons is supported by other research (e.g., Walker, Meyer, & Smelcer, 1993; Walker, Philbin, & Fisk, 1997).

Not surprisingly, Rogers et al. (2005) also found that response times and the variability of response times increased as button size on the touch sensitive display decreased. They examined five different button sizes: 11, 13, 16, 18, and 21 mm (< 0.5 in. to < 1 in.). For younger adults, the mean response times ranged from 379.32 ms to 426.89 ms for buttons of size 21 to 13 mm, respectively. However, with 11 mm buttons, the younger adults' mean response time increased to 511.63 ms. Older adults exhibited the same pattern of results as the younger adults; however, their response times were slower overall. The older adults' mean response times ranged from 490.69 ms to 608.68 ms for buttons of size 21 to 13 mm, respectively. Their mean response time increased to 815.48 ms when using the 11 mm buttons. Both the younger and the older adults exhibited a larger increase in response time when going from a 13 mm button to an 11 mm button than they had experienced when going from a 21 mm button to a 13 mm button. These results accounted for errors and suggest that there is a significant decrease in pointing performance when buttons are smaller than 13 mm.

Taken together, the literature suggests that there are no concrete design solutions that provide an easy and correct solution every time. Rather, interface designers must consider all of the demands of the task at hand. We took the approach of conducting rapid prototyping and usability tests while keeping the known recommendations and research in mind. We did not want to restrict our design to accommodate disparate design standards, nor did we want to create an interface that was difficult and frustrating to use. Therefore, the prototype Integrated EFDI contains FDEs that are height-adjustable for purposes of initial usability testing. The FDEs in the initial prototype are about 0.5 in. (13 mm) in height and about 4 in. (100 mm) in width. The height we selected was a reasonable starting point based on the research of Rogers et al. (2005) because this is where the largest increase in response times occurred for both younger and older adults in their study. The research of Rogers et al., Walker et al. (1993), and Walker et al. (1997), although perhaps counterintuitive and in disagreement with some design guidelines, supports our decision to design the FDEs in a list or stacked button style.

Controllers can move FDEs within a list by selecting an FDE and dragging it to a new location. When dragging an FDE, it remains in its original location and only a frame of the FDE moves vertically across the display. The controller uses the top of the frame as a reference to determine FDE placement. To place an FDE above another FDE, the top of the frame must move above the top of the other FDE. To place an FDE below another FDE, the top of the frame must move below the top of the other FDE. Once the controller moves the frame to the desired location and releases the frame, by removing his fingertip from the interface, the FDE will occupy the new location. We animated FDE movement so that when a controller moves an FDE within a list or inserts an FDE in a new list, there is a visual affordance to indicate what happened. For example, when the controller moves an FDE, the FDE will occupy its new position and the FDEs below it will move out of its way simultaneously.

6.1.2.1.1 Pending List

The pending list appears on the ground controller's EFDI. The pending list contains aircraft that have pushed back from the terminal gate and are waiting on the ramp to contact ground control. The flight data attributes that appear in the pending list are call sign, aircraft type, runway assignment, proposed departure time or EDCT (indicated in hours and minutes [HH:MM] Universal Coordinated Time (UTC)), and ATIS update indicator (see Figure 5). The FDEs in the pending list appear muted (gray) to indicate that these aircraft have not contacted the ground controller yet. When an aircraft makes initial contact with the ground controller, the ground controller verifies that the pilot has the current ATIS code by selecting the ATIS update indicator box on the right side of the aircraft's FDE (see section 6.1.2.6.1 for details about the ATIS update indicator). When the controller issues a taxi clearance, he moves the FDE into the outbound list by selecting the aircraft's FDE or data block and then selecting the outbound list header.

Figure 5. FDEs in the pending list on the ground control Integrated EFDI.

6.1.2.1.2 Outbound List

The outbound list appears on the ground controller's EFDI. FDEs appear in their active color (white) at the top of the outbound list when the ground controller moves them from the pending list. Once an aircraft is in the outbound list, this indicates that the controller has issued a taxi clearance and the aircraft is taxiing from the ramp to the departure runway. When the ground controller places an FDE in the outbound list, the EFDS automatically records a taxi time (indicated in hours and minutes [HH:MM] UTC) and the taxi time replaces the proposed departure time in the FDE. The flight data attributes that appear in the outbound list are call sign, aircraft type, first departure fix, runway assignment, taxi time or EDCT, and ATIS update indicator (see Figure 6). If an FDE contains an EDCT, placing the FDE in the outbound list will also automatically record a taxi time, but the EDCT will remain visible. When the ground controller is ready to transfer responsibility of an aircraft, the controller selects the aircraft's data block or FDE from the outbound list and then selects the local header to transfer it to the local controller.

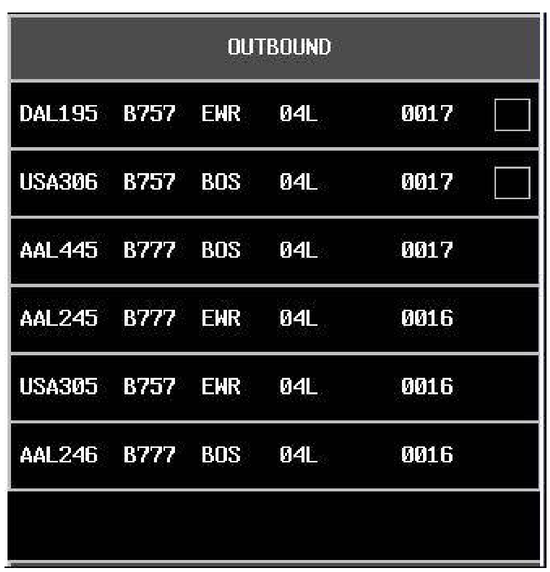

Figure 6. FDEs in the outbound list on the ground control Integrated EFDI.

6.1.2.1.3 Departure List

When the ground controller transfers an FDE, it appears at the top of the departure list on the local controller's EFDI. The departure list contains the flight data attributes of call sign, aircraft type, first departure fix, runway assignment, taxi clearance time or EDCT, and ATIS update indicator (see Figure 7). If necessary, the controller can adjust the departure sequence, as indicated by FDE order, by dragging an FDE to a new position within the departure list. Once an aircraft begins its takeoff roll, the EFDS automatically replaces the time field in the FDE with a timer that begins to increment (displayed in minutes and seconds [mm:ss]). The controller can use this timer for departure spacing (see details in section 6.1.2.6.2 for details on timers). Once the local controller has instructed the aircraft to contact the departure controller, the local controller then selects the appropriate FDE or data block followed by the TRACON button to transfer the FDE to the departure controller.

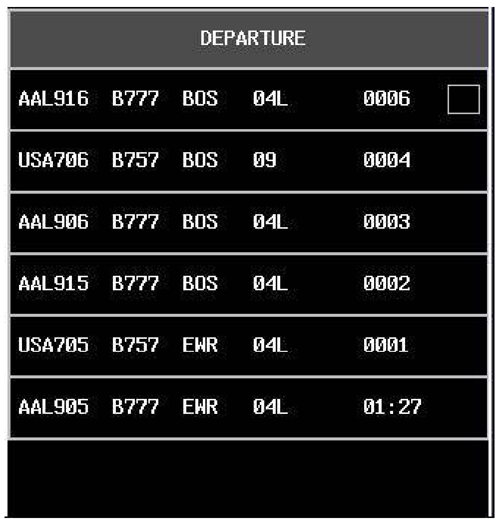

Figure 7. FDEs in the departure list on the local control Integrated EFDI.

6.1.2.1.4 Arrival List

The arrival list is located on the local control EFDI and shows aircraft that are on approach to the airport. The arrival list appears ordered by the predicted time of arrival at the runway. An aircraft's FDE appears at the top of the arrival list when the aircraft reaches the outer marker. The flight data attributes contained in the arrival list are call sign, aircraft type, runway assignment, and ATIS update indicator (see Figure 8). The local controller can change the landing runway assignment for an aircraft by selecting the aircraft's FDE or data block, followed by the appropriate runway button. If necessary, the controller can adjust the indicated arrival sequence by dragging an FDE to a new position within the arrival list. Once an aircraft has landed and the local controller instructs the pilot to contact ground control, the local controller transfers the FDE to the ground controller by selecting the aircraft's FDE or data block followed by the ground button.

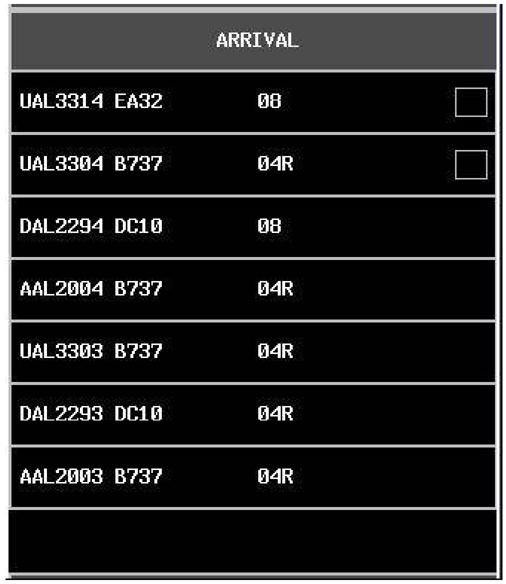

Figure 8. FDEs in the arrival list on the local control Integrated EFDI.

6.1.2.1.5 Inbound List

When the local controller transfers an FDE to the ground controller, the FDE appears at the top of the inbound list on the ground control EFDI. The flight data attributes contained in the inbound list are call sign and aircraft type (see Figure 9). The ground controller will provide the aircraft with a taxi clearance back to the ramp area and then select the aircraft's FDE or data block followed by the ramp button. This sequence of actions causes the aircraft's flight data to appear muted (gray) to indicate that the aircraft is leaving the airport surface. The aircraft's data block and FDE will automatically disappear once the aircraft reaches the ramp. An aircraft's data block and FDE will not disappear if the ground controller does not transfer the FDE to the ramp position. This ensures that the ground controller does not forget to transfer communications along with the flight data.

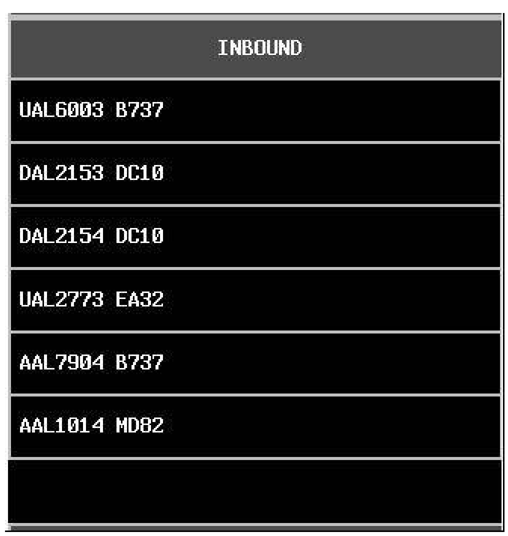

Figure 9. FDEs in the inbound list on the local control Integrated EFDI.

6.1.2.2 Data Blocks

We decided to change the ASDE-X data block format to reflect only the most important and relevant information for the controller. The departure data block contains four flight data attributes including aircraft call sign, aircraft type, first departure fix, and runway assignment (see Figure 10). The arrival data block contains only two flight data attributes; namely, aircraft call sign and aircraft type (see Figure 11). We used these formats for the arrival and departure aircraft data blocks on both the ground and local controller Integrated EFDIs. Like the FDEs, controllers can select data blocks by touch to perform flight data functions. Data block management is another new feature of the Integrated EFDI described in section 6.1.2.8.

Figure 10. Departure aircraft position symbol, leader line, and data block on the local control Integrated EFDI.

Figure 11. Arrival aircraft position symbol, leader line, and data block on the local control Integrated EFDI.

6.1.2.3 Readout Area

The readout area is located above the EFD lists. Three different types of information may appear in the readout area; full flight data for an arrival aircraft, full flight data for a departure aircraft, or a list of the most recent FDEs transferred to another controller position or facility.

When a controller selects a data block or an FDE from one of the lists, the full set of flight data attributes appears in the readout area. Different attributes appear depending on whether the associated aircraft is an arriving or departing flight. When a controller selects an arrival aircraft's FDE, the aircraft's call sign, type, computer identification (CID), runway assignment, and remarks appear in the readout area (see Figure 12). For arrival aircraft, the readout area also contains a missed approach button. When the controller selects the missed approach button, the EFDS automatically enters a standard altitude and heading in the aircraft's flight data information based on runway assignment.

Figure 12. The readout area information shown for an arrival aircraft.

The readout area works in the same general way for departure aircraft. The primary difference is that departure aircraft have more flight data attributes than arrival aircraft. When the controller selects a departure aircraft's FDE or data block, the readout area displays the aircraft's call sign, type, CID, beacon code, proposed departure time, taxi time, EDCT, assigned heading, assigned altitude, assigned runway and intersection departure, full route of flight, and remarks (see Figure 13).

Figure 13. The readout area information shown for a departure aircraft.

The readout area can also show a history of recent FDEs that a controller transferred to another position or facility. For example, the ground controller can display in the readout area the last four FDEs transferred to the local controller by selecting the local controller button. When the ground controller selects the local controller button, the FDEs appear muted in the readout area. The ground controller can recall any of the FDEs displayed in the readout area by selecting the FDE and then selecting a list header to place the FDE in the top of the appropriate list. Likewise, the local controller can recall an FDE from either the ground controller or the TRACON controller in the same manner. The local controller can select either the ground or TRACON header to see a list of the most recently transferred FDEs in the readout area. The local controller then selects an FDE and the appropriate list header to place the FDE at the top of the appropriate list.

When the controller selects an FDE or data block, they may change the altitude or heading assignment by typing "a" for altitude or "h" for heading followed by a three-digit number and the "Enter" key. The controller can change both the altitude and heading assignments at the same time by linking the commands. For example, when the controller selects an FDE or data block and the flight data appears in the readout window, the controller can type “a120h350” and press the Enter key to change the altitude assignment to 12,000 ft and the heading to 350 degrees. The controller can link the commands in the opposite order (e.g., “h350a120”) to obtain the same result. The controller may include spaces, but entries that violate the syntax rule or exceed the range of possible values return an “Invalid Entry” message to the preview area on the surface situation display. When a controller changes an altitude or heading assignment, an asterisk will appear on the right-hand side of the aircraft’s FDE (see Figure 14) and appears highlighted in the readout area (see Figure 15). When a controller transfers the FDE to another controller, the asterisk notifies the receiving controller that there has been a change to either the altitude or heading assignments. The controller can select the FDE displaying an asterisk and examine the flight data in the readout window. The changed flight data attributes appear highlighted until the controller acknowledges the change by touching the readout area. Acknowledging the change turns off the highlighting in the readout area and removes the asterisk from the FDE.

Figure 14. FDE indicating that there has been a change to the assigned altitude and/or heading.

Figure 15. Amended altitude and heading assignments as depicted in the readout area.

6.1.2.4 Buttons

Each list header also functions as a button that the user can select. Buttons, similar to FDEs and data blocks, provide visual feedback to the user when activated. Both the ground and local controllers can change an aircraft’s runway assignment or assign an intersection departure by selecting the aircraft’s FDE or data block and then selecting the appropriate button located on the surface situation display. The EFDS prevents the controller from assigning invalid combinations of runway and intersection assignments. Controllers can place the runway and intersection assignment buttons anywhere on the situation display. Figure 16 shows buttons on the surface situation display that the controller can use to assign and record runway and intersection departure assignments. Assigning a new runway or an intersection departure will update the information in the aircraft’s FDE, data block, and the readout area.

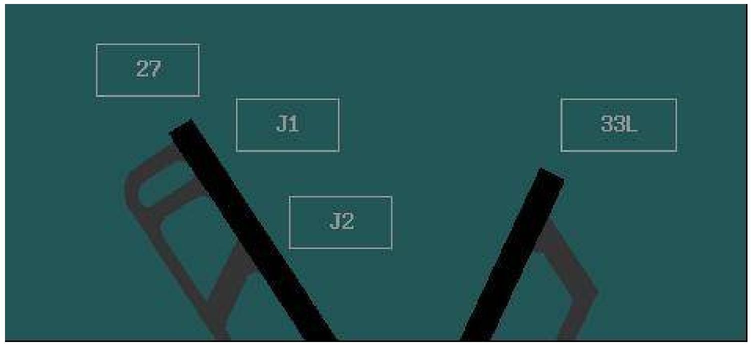

Figure 16. Buttons on the Integrated EFDI situation display for runway and intersection departure assignments.

6.1.2.5 System Information Window

The system information window is transparent and is located on the surface situation display. The system information window contains the current date, time (displayed in hours, minutes, and seconds [HH:MM:SS] UTC), coast/suspend track list, and ATIS code. The user invokes typical ASDE-X display functionality to locate this window anywhere within the situation display. The controller cannot place the system information window over the EFD lists or the readout area. Figure 17 shows the system information window and its elements.

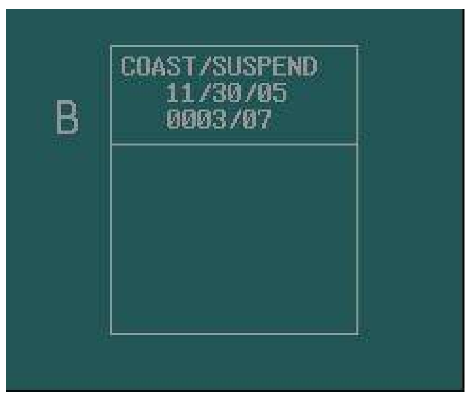

Figure 17. The system information window and elements.

6.1.2.6 Reminders

The working group's task analysis indicated that a set of reminders would be useful. The rationale and description for each reminder follows.

6.1.2.6.1 Automatic Terminal Information Service Update Indicator

The ATIS is a continuous broadcast of recorded or automated non-control information. Typically, the FAA places ATIS equipment at high activity terminal areas to improve controller effectiveness and to reduce frequency congestion. The ATIS usually updates about once an hour, but may update more often when special circumstances arise or when weather conditions change rapidly. According to the FAA Order 7110.65P (FAA, 2004a), controllers must use a procedure on initial contact with an aircraft to verify that the pilot has the most recent ATIS information. If the pilot does not have the most recent information, the controller will provide it or request that the pilot get it before receiving any further ATC clearances.

A condition arises when the ATIS information changes after initial contact with an aircraft but before the aircraft leaves the airspace. In this situation, the controller must provide the updated information to the pilots or at least notify them that new information is available. We decided to implement an ATIS update indicator given the FAA requirement for controllers to ensure that all pilots have the most recent ATIS information, the identification of the ATIS code by the working group as an important flight data attribute, and the potential impact of new ATIS information on an aircraft's operation.

The ATIS update indicator on the Integrated EFDI works by alerting the controller whenever the ATIS changes. An ATIS change automatically causes the ATIS code to flash near the system information window. The ATIS code appears yellow for 1.5 s then white for 1.5 s for a total duration of 15 s. This flash rate is in accordance with established design standards (e.g., Department of Defense, 1999; and Woodson et al., 1992). The controller can acknowledge the ATIS change by touching the flashing ATIS code, at which time the ATIS code stops flashing and displays normally (gray). After 15 s, if the controller does not acknowledge the ATIS change, the ATIS code stops flashing and is displayed in yellow. The ATIS code remains displayed in yellow until the controller acknowledges the ATIS change by touching the ATIS code near the system information window.

In addition to alerting the controller to ATIS updates, the Integrated EFDI also indicates which aircraft to advise of the change. For example, on the ground controller's EFDI, all FDEs appear in the pending list with a box indicator on the right-hand side (see Figure 18). This indicator reminds the controller to ensure that the pilot of the aircraft has the current ATIS information. Once the controller provides the current ATIS information to the pilot, the controller touches the box indicator in the aircraft's FDE to make it disappear. When the ATIS information updates, all FDEs in the outbound, departure, and arrival lists will display the box indicator once again.

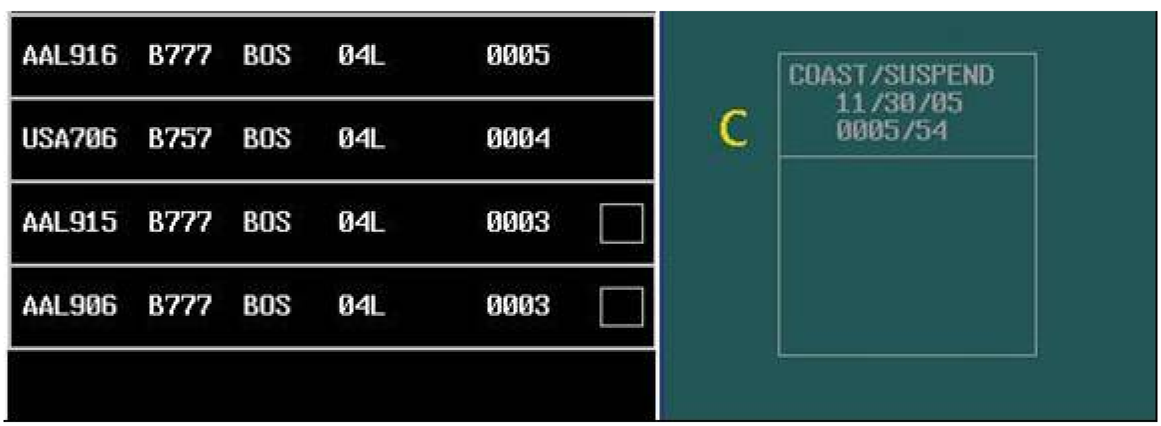

Figure 18. FDEs with ATIS update indicator and associated indicator near system information window.

The ATIS update indicator consolidates general ATIS information with the EFD and may improve controller efficiency by reducing the controller's need to shift attention to another information source. The ATIS update indicator also provides a means for controllers to easily verify and record which aircraft have the current ATIS information.

6.1.2.6.2 Timers

Timing is very important in ATC. Controllers must use timing to merge traffic streams and to ensure proper separation during takeoff and landing. The working group indicated that associating a timer with the EFD would reduce a potential bottleneck in the task flow by relieving the controller of some mental workload. For example, when trying to ensure separation for wake turbulence on departure, controllers must track the time between takeoffs from the same runway. Currently, controllers have no easy way to estimate the time between takeoffs so they must allocate a portion of their attention to record and monitor the passing time. The working group suggested that controllers could also use the timer as a generic reminder rather than associating it with any particular aircraft.

On the Integrated EFDI, the timer works automatically with departure aircraft to assist the controller in determining appropriate departure spacing. Once an aircraft begins its takeoff roll (as sensed by surface radar), the time field in the aircraft's FDE is automatically replaced by a timer that begins to increment from zero (as explained in section 6.1.2.1.3). Once the timer reaches the appropriate time, the controller can release the next aircraft for departure. This timer should improve controller efficiency because it removes the controller's need to remember or record the departure time. The timer also reduces the cognitive workload associated with keeping track of how much time has elapsed since the previous departure. To use the timer generically, the controller selects the current time/date field in the system information window to activate the timer interface (see Figure 19).

Figure 19. The Integrated EFDI Timer Interface.

The controller can drag the timer interface to a preferred location on the Integrated EFDI situation display by using either the upper-left or upper-right corner as a handle. To set a generic timer, the controller selects the amount of time desired and then selects the start/stop button to begin a countdown. The controller can select one of the numbered buttons to add 1, 2, or 5 min to the timer, or they can select the up arrow to add 1 min or the down arrow to subtract 1 min. Selecting the reset button will reset the timer to zero. When the controller selects a button on the timer interface, the button provides visual feedback by appearing with a white background when activated. Once a controller starts a generic timer by selecting the start/stop button, the timer itself appears to the right of the system information window. When the timer is running, the border around the start/stop button is green and the timer counts down. Selecting the start/stop button while a timer is running will pause the timer. When the timer is paused, the border around the start/stop button is red and the count down pauses. Figure 19 also shows a timer that the controller has paused with 1 min 23 s remaining. The timer interface remains visible until the controller selects the current date/time field in the system information window. The controller can make the timer interface visible again by selecting the timer located to the right of the system information window (see Figure 20). Once the timer expires, the Integrated EFDI notifies the controller by flashing the expired timer. As with the ATIS update indicator, the expired timer will flash yellow for 1.5 s and then white for 1.5 s for a total duration of 15 s. After 15 s of flashing, the expired timer displays in yellow text. The controller can acknowledge the elapsed timer while it is flashing or after it stops flashing. Acknowledging the elapsed timer by selecting it causes the timer to disappear from the display.

Figure 20. Generic timer information located to the right of the system information window.

The controller can associate a timer with a particular aircraft by selecting the aircraft's FDE or data block and then activating the timer interface. The timer interface appears and operates as when using the timer generically. Once the controller associates a timer with an aircraft, a timer icon appears in the aircraft's FDE (see Figure 21). When the timer expires, the Integrated EFDI notifies the controller by flashing the timer icon in the associated aircraft's FDE. The controller can acknowledge the elapsed timer by selecting the timer icon, causing it to disappear. Both the generic and the aircraft-associated timers are specific to the controller's display. In other words, a generic or aircraft-associated timer set by a controller on their Integrated EFDI is not shared with any other controller position. Likewise, a timer associated with a particular aircraft will only operate on the controller position where it was established.

Figure 21. FDE for an aircraft that has an associated timer indicated by the icon on the right side.

6.1.2.6.3 Generic Highlighting

The working group's task analysis and recently collected observational field data (Dattel et al., 2005) show that controllers use a number of techniques to make certain pieces of flight data more conspicuous. For example, a common technique is to offset an FPS so that it stands out from the other FPSs in the bay. Controllers may also highlight particular pieces of information that are unusual or especially critical to operations. We wanted to preserve the controller's ability to create conspicuity as a reminder. Highlighting information is a relatively easy way to create conspicuity. Controllers can highlight flight data on the Integrated EFDI by selecting an

FDE or data block and then selecting the readout area. Highlighting flight data causes the text for the selected FDE and data block to appear in light blue (see Figure 22). To remove the highlighting, the controller selects a highlighted FDE or data block and then selects the readout area.

Figure 22. A highlighted FDE and the associated data block.

We also use automatic highlighting in the Integrated EFDI to indicate situations that need special attention. Automatic highlighting occurs when an aircraft has taxied into position and is holding on the runway, when there is expired time information such as departure delay, or when an aircraft has an EDCT.

6.1.2.6.4 Taxi into Position and Hold

Aircraft waiting on an active runway present a potential problem that is inherent in the procedure known as TIPH. Controllers use the TIPH procedure to maximize the efficiency of runway usage. TIPH works because it allows controllers to clear an aircraft for takeoff, and then place another aircraft on the runway in position and ready to take off as soon as possible. However, controllers must remember when an aircraft is holding on the runway to prevent possible runway incursions and collisions. The Integrated EFDI uses the surface surveillance capabilities to automatically detect and indicate when an aircraft is holding or stopped on an active runway by highlighting the aircraft's FDE and data block in orange (see Figure 23). Once the aircraft begins its takeoff roll, the Integrated EFDI automatically removes the highlighting.

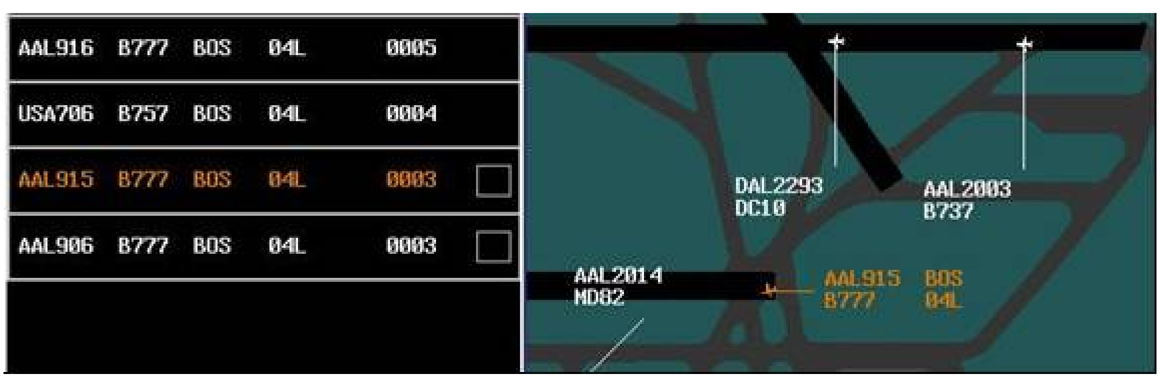

Figure 23. An aircraft that has taxied into position and is holding on the runway causing highlighting of the aircraft's FDE and data block.

6.1.2.6.5 Departure Delay Timer

As soon as the Integrated EFDI records a taxi time for an aircraft, a departure delay timer is automatically associated with the FDE and begins a countdown. We implemented the departure delay timer to provide an indication of aircraft that are "delayed" on the airport surface. According to the FAA Order 7210.55C (FAA, 2004b), an airport surface delay occurs when an aircraft remains on the airport surface for 15 min or longer after entering FAA jurisdiction (i.e., the aircraft calls ready) accounting for normal taxi time for the runway configuration in use. Based on the airport configuration we used to develop our prototype interfaces, we determined that it would take about 5 min for an aircraft to execute an unimpeded taxi route to a departure runway. Therefore, if 20 min (5 min taxi time + 15 min surface delay) elapse before the local controller clears the aircraft for departure, the time field on the FDE will appear highlighted with a yellow background and black text to indicate that the aircraft has entered a delay status. The Integrated EFDI automatically records the number and duration of each departure delay for subsequent reporting.

6.1.2.6.6 Expected Departure Clearance Time

We decided to implement highlighting and an automatic reminder associated with aircraft that have an assigned EDCT. Highlighting the EDCT and creating a reminder associated with a nearly expired EDCT is necessary because of the undesirable consequences that may occur if an EDCT expires. An aircraft must depart at or near the EDCT to maintain its position in the scheduled traffic flow or else incur a delay. If an aircraft does not depart within 30 min after an assigned EDCT, then the NAS automatically deletes the aircraft's filed flight plan from the system. Therefore, if an aircraft has an EDCT, that field appears highlighted in the FDE and the letter "E" is appended to the time (see Figure 24). The Integrated EFDI automatically alerts the controller when an aircraft has not departed within 10 min before an assigned EDCT time by flashing the time field on the aircraft's FDE. The flashing EDCT operates in the same cycle as previously described. If the controller does not acknowledge the flashing EDCT during the 15 s flashing period, the flashing stops and the EDCT displays highlighted with a yellow background and black text. This reminds the controller to take action to depart the aircraft in the near future. A second EDCT warning occurs at the EDCT. If an aircraft has not departed by its assigned EDCT, the Integrated EFDI alerts the controller by flashing the EDCT again. We designed this second reminder to convey more urgency to the controller by flashing the EDCT between white and orange for 15 s. If the controller does not acknowledge the second EDCT reminder within the 15 s time period, then the EDCT appears highlighted with an orange background and black text.

Figure 24. FDE with highlighted time field to indicate that an aircraft has an EDCT.

6.1.2.7 Touch Screen Integration

As part of the prototype interface design and testing process, we need to understand how controllers are using the Integrated EFDI. By making the Integrated EFDI aware of whether the controller is making input via touch or the trackball, we can record usability data to analyze how the Integrated EFDI changes the task and to determine what new demands the interface places upon the controller. We also designed the Integrated EFDI to operate simply and with a minimal number of required interactions. Therefore, we implemented an overall principle that separates EFD actions from ASDE-X actions and prevents controllers from having to switch between different modes of entry to complete a single action. We designed the keyboard and trackball to be used primarily for adjusting the ASDE-X display whereas the touch screen is used primarily for interacting with flight data.

Using a touch sensitive display is a relatively new approach for FAA ATCTs. One potential problem with using a touch sensitive display is the possibility of inadvertently making a selection. We also did not want to interfere with controllers' existing knowledge of ASDE-X operations. To address these potential problems we designed the Integrated EFDI elements so that some respond only to touch and others respond only to trackball input. Touches will only interact with an EFD element and trackball picks will only interact with an ASDE-X element. For example, if a data block on one aircraft is overlying a position symbol of another aircraft, a touch over a data block will select the data block, not the position symbol. We have prevented the controller from selecting a position symbol via touch because the position symbol is very small and difficult to select with touch, and because selecting the position symbol already performs a particular function in the ASDE-X system. Therefore, touching inside the surface situation display only affects EFD elements such as data blocks, ATIS update indicator, timer, or buttons. Touching a button on the ASDE-X menu bar only moves the trackball pointer to that location, but does not act as a trackball selection.

6.1.2.8 Data Block Management

Our design also reduces inadvertent selections by automatically eliminating clutter from the display and reducing the chance that two elements occupy the same location on the display. It is difficult to avoid data block overlap on the ASDE-X situation display. Clutter is not completely unexpected given the display size, the number of aircraft that may be located near one another on the airport surface, and the size and number of data blocks on the display. The ASDE-X system does make some provisions for controllers to manage data blocks, such as the ability to define areas where data blocks will behave in a particular manner. For example, a controller may define an area on the display where the data blocks shift to a preferred orientation. The controller may even define an area where the data blocks are not visible at all. Unfortunately, data block overlap still occurs on the ASDE-X situation display even with the help of existing tools. To address the problem of clutter and data block overlap in particular, we implemented initial orientation of the leader lines and an automatic data block management algorithm.

When an aircraft representation first appears on the surface situation display, the Integrated EFDI automatically orients the leader line based on the aircraft's runway assignment. The aircraft assigned to Runway 27 will appear with the leader line to the left (i.e., position "4") of the aircraft position symbol. The aircraft assigned to Runway 33L or 33R will appear with the leader line to the right (i.e., position "6") of the aircraft symbol. The initial leader line position is not fixed and will be affected by the automatic data block management algorithm.

In addition to initial leader line orientation, the automatic data block management algorithm also dynamically separates data blocks as they traverse the display. This is done by using a force equation to predict where each data block will be in the near future and then automatically adjusting the length and orientation of each data block's leader line to maintain their separation. We have refined the parameters of the algorithm to prevent the data blocks from moving unnecessarily. (The author dubbed an earlier version of the algorithm "dancing data blocks" due to the excessive movement it produced.) Unlike the ASDE-X situation display, leader line positions in the Integrated EFDI are not limited to just the eight cardinal positions and can occupy any orientation. Automatic data block management should reduce controller workload and help ensure that data block information is available to the controller at all times.

Controllers can also manually select and drag a data block to move it to a new position. Controllers in the en route domain often use leader line length or orientation as a reminder (e.g., Truitt, Durso, Crutchfield, Moertl, & Manning, 2000). Being able to easily adjust data block position, including the leader line length and orientation, provides the same potential benefit to ATCT controllers. Once the controller moves a data block manually, that data block will no longer be subject to the algorithm. Simply selecting a data block does not remove the data block from the algorithm. The controller can return all of the data blocks to the algorithm by typing "5" followed by the Enter key. The controller can return a single data block to the algorithm by selecting the aircraft's FDE or data block and then typing "5" followed by the Enter key.

6.2 Perceptual-Spatial Electronic Flight Data Interface

We designed the Perceptual-Spatial (P-S) EFDI as an alternative to the Integrated EFDI for ATCTs that do not have surface surveillance capability. The P-S EFDI also serves as a backup system for the Integrated EFDI in case surface surveillance fails. We used the same task objects and attributes generated by the working group to build the P-S EFDI. In fact, the P-S EFDI shares most of the features of the Integrated EFDI, but presents them in a fundamentally different way. Because the P-S EFDI shares many features with the Integrated EFDI, as explained in previous sections of this paper, this design description focuses on the unique aspects of the P-S EFDI.

The fundamental concept of the P-S EFDI is to provide a means to track, record, communicate, and organize EFD spatially in relation to the airport surface. Although controllers could use this tool in conjunction with an ASDE-X display as an alternative to the Integrated EFDI, unlike the Integrated EFDI, it does not rely on surface surveillance capability. Like the Integrated EFDI, we designed the P-S EFDI to help controllers correlate flight data more closely with the actual aircraft they represent. In other words, we wanted to create a physical, observable relationship between real aircraft on the airport surface and their abstract representation in the form of flight data. By strengthening this relationship between flight data and their associated aircraft, we hope to enhance the controller's ability to maintain awareness of the traffic situation, remember critical information, and perform more efficiently.

6.2.1 Perceptual-Spatial Electronic Flight Data Interface Hardware

The P-S EFDI uses the same hardware as the Integrated EFDI (see section 6.1.1 – Integrated EFDI Hardware).

6.2.2 Perceptual-Spatial Electronic Flight Data Interface Design Description

We designed separate interfaces for the ground control and local control positions to accommodate the information needs of each position. The primary elements common to both the ground and local control positions include the airport representation, FDEs, the readout area, buttons, the system information window, and reminders including the ATIS update indicator, generic timer, aircraft specific timer, and generic, TIPH, EDCT, and delay highlighting (see Figure 25). We present descriptions of the P-S EFDI's primary elements in the following sections.

Figure 25. The primary elements of the P-S EFDI including FDEs, the readout area, buttons, and system information window.

6.2.2.1 Flight Data Elements

Because the P-S EFDI operates independently of any surface surveillance capability, it does not contain any data blocks, however, the FDEs are similar in form to that of a data block. Like the Integrated EFDI, the FDEs occupy various lists or areas on the display. In the P-S EFDI, FDEs do not compose a list per se. Rather, FDEs occupy space on the airport surface map and the controller can visually categorize them by their color, shape, location, and flight data attributes. FDEs in the P-S EFDI include different types for pending, outbound, inbound, arrival, and departure aircraft. The P-S EFDI FDEs use the same flight data attributes as the Integrated EFDI FDEs. All of the EFD functions operate in a manner similar to the Integrated EFDI. However, with the P-S EFDI, the controller can move each FDE and place it on the airport surface map.

Placing an FDE in certain areas of the map has different effects including time records and highlighting, as explained in later sections of this design description.

We designed the FDEs to minimize their size while still maintaining usability. Along with the other primary elements, the outbound, inbound, and arrival FDEs appear on both the ground and local controller positions. The pending FDEs only appear on the ground controller's position and the departure FDEs only appear on the local controller's position. Although many of the FDEs appear on both the ground and local controllers' displays, the FDEs do not appear the same on both displays. Namely, FDEs that are in the possession of a controller appear with white text, but pending FDEs or FDEs possessed by another controller appear with gray text. With the exception of pending FDEs on the ground controller interface, a controller can only move FDEs that are in their own possession. This possession rule preserves the FDEs' usefulness for communicating information about aircraft location and prevents controllers at different positions from interfering with each other's actions.

6.2.2.1.1 Pending Flight Data Elements

Figure 26 shows the pending FDEs for departure aircraft that are waiting on the ramp. A pending FDE contains aircraft call sign, type, first departure fix, runway/intersection assignment, and proposed departure time. A pending FDE may also contain an EDCT instead of a proposed departure time and an ATIS update indicator. The pending FDEs appear on their designated ramp spot with gray text to indicate to the ground controller that he has not yet contacted the aircraft. Pending FDEs only appear on the ground controller's EFDI. The ground controller must move a pending FDE when he makes initial contact with the aircraft and provides a departure taxi clearance.

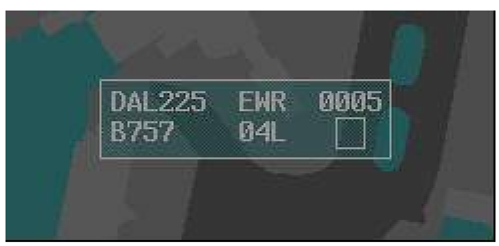

Figure 26. A pending FDE on the ground control P-S EFDI.

6.2.2.1.2 Outbound Flight Data Elements

When the controller moves an FDE out of the ramp area onto the airport surface map it becomes an outbound FDE. Figure 27 shows an outbound FDE. An outbound FDE contains the flight data attributes of call sign, aircraft type, first departure fix, runway/intersection assignment, and assigned taxi time. An outbound FDE may also contain an EDCT in lieu of taxi time, ATIS update indicator, and flight data update indicator or timer indicator. When the ground controller moves an FDE out of the pending list, the EFDS automatically assigns and records a taxi time and the FDE text changes from gray to white. To transfer the outbound FDE to the local controller, the ground controller selects the FDE and then selects the local controller button. This causes the FDE on the ground controller's interface to display with gray text and the associated FDE on the local controller's departure list will display with white text.

Figure 27. Outbound FDEs showing the flight data attributes.

6.2.2.1.3 Departure Flight Data Elements

The departure FDEs appear white on the local controller's EFDI and gray on the ground controller's EFDI. A departure FDE contains the flight data attributes of call sign, aircraft type, first departure fix, runway/intersection assignment, and assigned taxi time (see Figure 28). A departure FDE may also contain an EDCT in lieu of taxi time, ATIS update indicator, and flight data update indicator or timer indicator.

Figure 28. Departure FDEs on the local control P-S EFDI showing the flight data attributes.

The controller can indicate a TIPH clearance by selecting an FDE and then selecting the appropriate TIPH button. To prevent the controller from placing an FDE on the wrong runway, the controller can only select the TIPH button that matches the aircraft's runway assignment as indicated in the FDE. When the controller indicates a TIPH clearance, the FDE occupies the location of the appropriate TIPH button and displays with orange text to remind the controller that an aircraft is holding on the departure end of the runway (see Figure 29).

Figure 29. Unoccupied and occupied TIPH buttons for two intersecting runways.

When the controller clears an aircraft for departure, the controller selects the FDE and then selects the departure clearance button (see Figure 30). Because only the local controller can clear an aircraft for TIPH or departure, the TIPH and departure clearance buttons only appear on the local controller's EFDI.

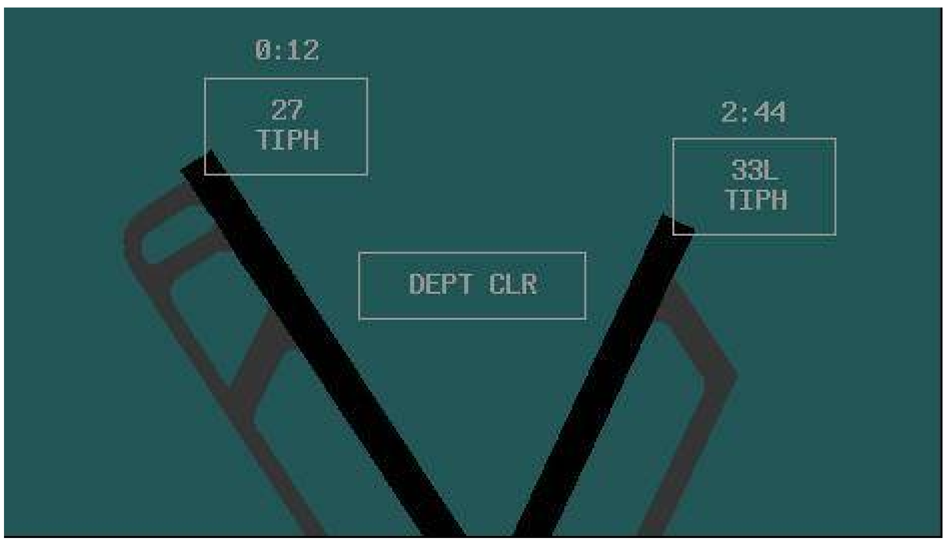

Figure 30. The departure clearance button located near the departure end of intersecting runways on the local control P-S EFDI.

Assigning a departure clearance to an FDE causes the P-S EFDI to automatically assign a departure time for that aircraft and the FDE occupies the departure list beneath the appropriate runway assignment button (see Figure 31). Once the local controller clears an aircraft for departure and takes the appropriate FDE action, the FDE in the departure list displays a timer in the FDE time field. The timer, displayed in minutes and seconds (mm:ss), begins incrementing as soon as the FDE occupies the departure list.

Figure 31. FDEs for aircraft that the local controller has cleared for departure appearing in the departure list on the local control P-S EFDI.

When the P-S EFDI assigns a departure time to an aircraft, a runway spacing timer also appears above the TIPH button for the appropriate runway to indicate the time since the last aircraft departed from that runway (as shown in Figures 29 and 30). The runway spacing timer indicates minutes and seconds (m:ss) and counts up from 0 to 5 min. The runway spacing timer disappears once it reaches 5 min (5:00) to prevent the display of extraneous data. In other words, a controller does not care if it has been more than 5 min since the last aircraft departed from a particular runway. The controller can transfer a departure FDE to the TRACON departure controller by selecting an FDE and then selecting the departure header in the departure list. Transferring a departure FDE to the TRACON causes the FDE to disappear from the local controller's EFDI. The runway spacing timer resets and begins counting up from zero again when the local controller clears the next aircraft to depart from the same runway.

6.2.2.1.4 Arrival Flight Data Elements

The arrival FDEs appear in an arrival list on the airport surface map (see Figure 32). The FDEs enter at the top of the list and are ordered by time sequence over the outer markers. Arrival FDEs appear with white text on the local controller's EFDI and in gray text on the ground controller's EFDI. The arrival FDEs contain aircraft call sign, type, runway assignment, and ATIS update indicator. The FDEs for aircraft assigned to land on different runways automatically offset from one another in the arrival list. Offsetting FDEs in this manner replicates the way that controllers use FPSs and provides an obvious visual indicator of aircraft landing on different runways that may intersect. The controller can resequence the arrival FDEs by selecting an FDE and dragging it to a new location within the arrival list.

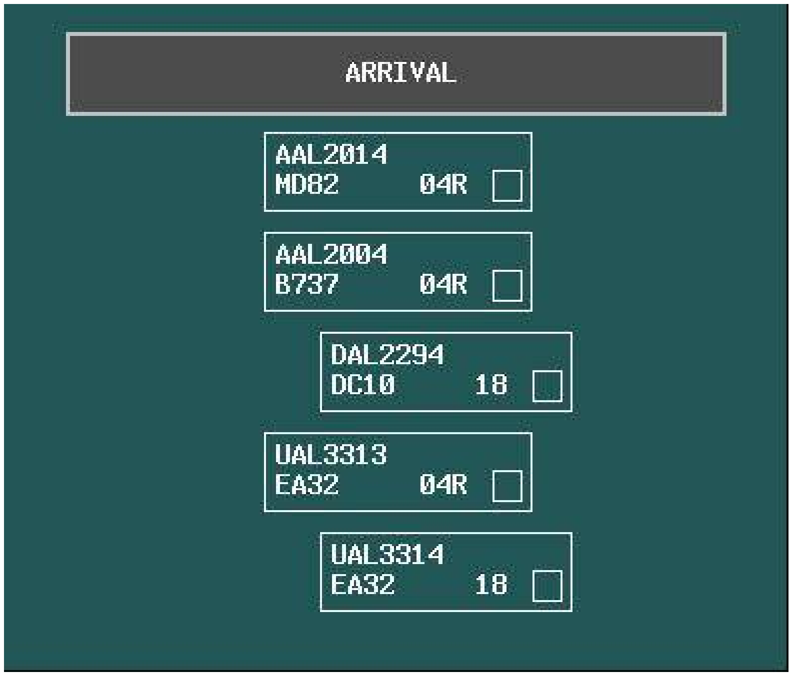

Figure 32. Arrival FDEs in the arrival list on the local control P-S EFDI.

Once an aircraft has landed, the local controller can drag the aircraft's FDE to the appropriate location on the airport surface map to indicate its approximate location. The local controller can transfer control of an FDE to the ground controller by selecting the FDE and then selecting the ground button. Transferring an arrival FDE to the ground controller causes the FDE to appear with gray text on the local controller's EFDI and with white text on the ground controller's EFDI.

6.2.2.1.5 Inbound Flight Data Elements

Once the local controller transfers an arrival FDE to the ground controller, the FDE becomes an inbound FDE. An inbound FDE shows only the flight data attributes of aircraft call sign and type and appears with white text on the ground controller's EFDI and with gray text on the local controller's EFDI (see Figure 33). The ground controller can transfer inbound FDEs to a ramp controller or AOC by selecting an FDE and then selecting the ramp button.

Figure 33. Inbound FDEs on the local control P-S EFDI.

6.2.2.1.6 Flight Data Element Chains

We discovered a number of usability issues related to FDE management during initial, low-risk usability testing and rapid prototyping. One issue we identified relates to sequences, or stacks, of FDEs positioned on the airport surface map of the P-S EFDI. A stack of FDEs is especially likely to form for the outbound and departure FDEs. Given our airport configuration as shown in Figure 25, the outbound and departure FDEs may stack up on the airport surface map while aircraft are waiting to cross an active runway or waiting to depart at the end of a runway. Because each FDE is independent of all other FDEs, controllers must use the FDEs in a way similar to FPSs. For example, when a controller removes one or more FPSs from a FPS bay, gravity closes the space created between the FPSs in the bay. Therefore, when a controller removes an FDE from a stack formed on the P-S EFDI airport surface map, we must simulate the effects of gravity or else the controller would continuously have to move numerous FDEs to maintain each stack.

To maintain stacks of FDEs, we designed three zones on the airport surface map where the P-S EFDI links FDEs together in a chain (see Figure 34). One zone is located on the common taxiway leading to the primary departure runways. This zone is located short of the active runways and will contain FDEs possessed by both the ground and local controllers. The other two zones are located on the taxiways on the opposite side of the active runways. Each of these two zones leads directly to the respective departure end of the runways and typically contain only FDEs possessed by the local controller (given the local controller is responsible for aircraft crossing an active runway). All three of these zones where FDEs form chains are like the lists contained in the Integrated EFDI, but their contents may be mixed (i.e., contain FDEs possessed by both the ground and local controller) and they are spatially anchored to the airport surface map. The zones are not visible to the controller. However, when the controller moves an FDE into one of the zones, the FDE will display a stem on top of the FDE to indicate that that FDE is in a zone. If the controller moves an FDE into one of the zones, the FDE will become part of a chain when the controller releases it (see Figure 35). The first FDE in a chain occupies an anchor position at the top of the zone. When the controller breaks an FDE chain by moving an FDE that is part of a chain, the remaining FDEs in the chain automatically move up the chain to close gaps, as appropriate, and the FDE at the front of the chain occupies the anchor position. For example, suppose a controller has five FDEs in a chain on the taxiway at the departure end of a runway. When the controller moves one of these FDEs into the TIPH position or indicates a departure clearance, the remaining FDEs in the chain will move up towards the departure end of the runway thereby automatically making space for more FDEs to occupy the chain.

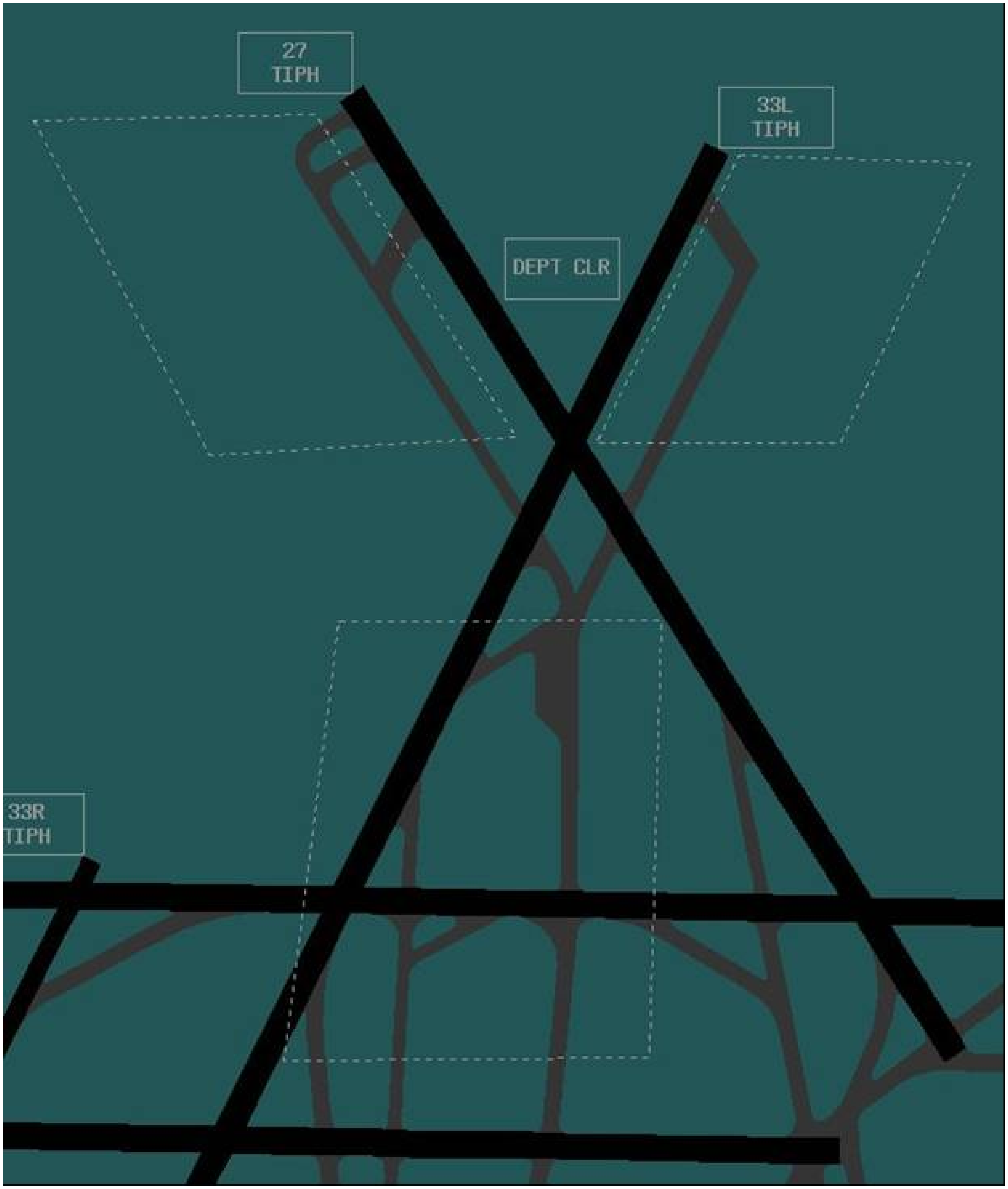

Figure 34. FDE zones on the local control P-S EFDI.

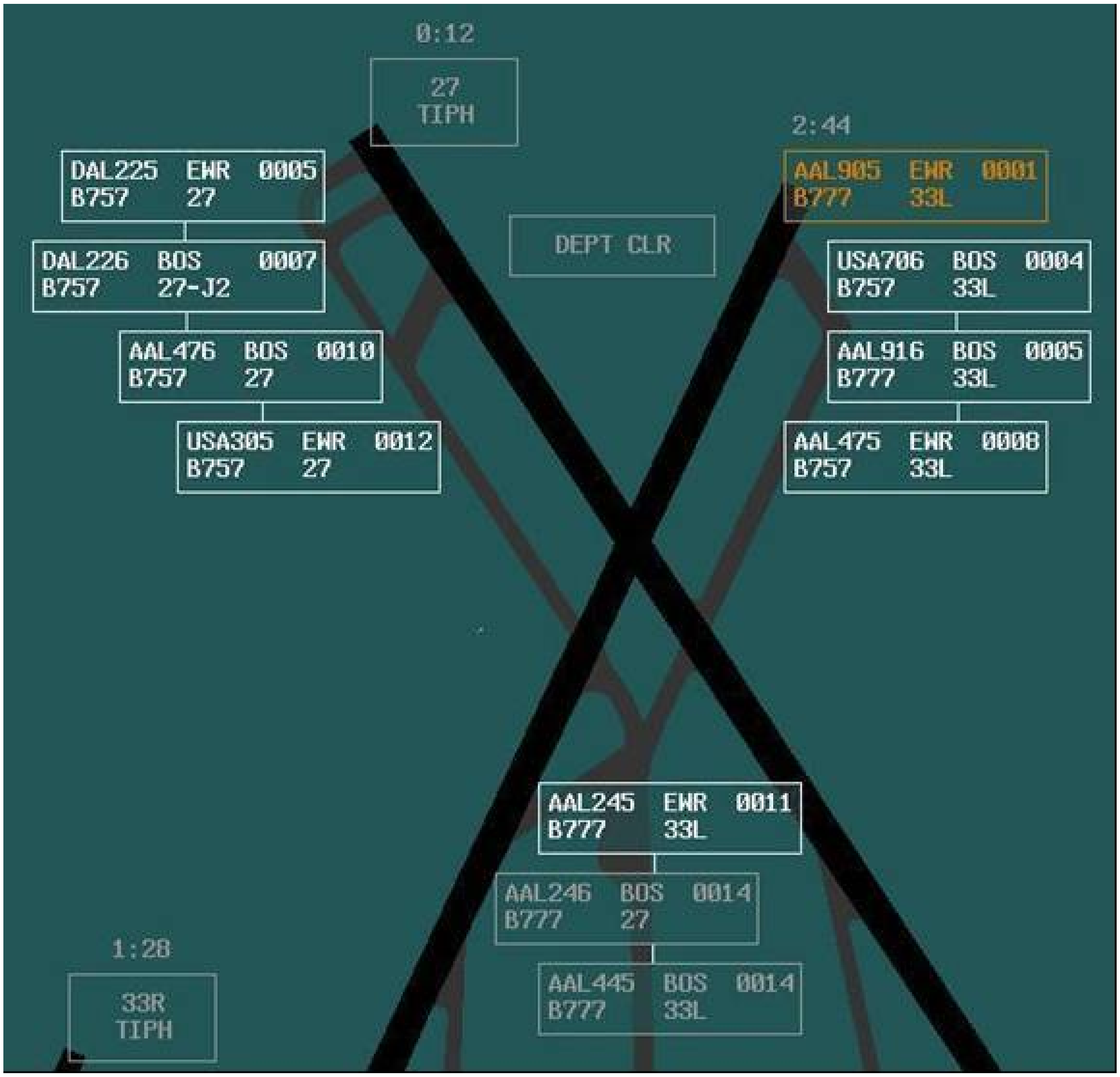

Figure 35. Three chains of FDEs on the local control P-S EFDI.

6.2.2.2 Reminders

We included all of the reminders in the P-S EFDI including the ATIS update indicator, generic and departure timers, generic highlighting, and delay status indicators. All of these reminders function as they did in the Integrated EFDI. We could not implement the automatic highlighting of EFD for aircraft receiving TIPH clearances because that function relied on the surface surveillance capabilities. We did, however, provide means for controllers to highlight an FDE to indicate a TIPH clearance, as previously described in section 6.2.2.1.3 on departure FDEs.

7. SUMMARY AND CONCLUSION

We developed both the Integrated and P-S EFDI prototypes based on the same foundation of information gleaned from the existing literature, task analyses, and SME opinion. Despite their common foundation, each prototype EFDI provides controllers with fundamentally different ways to manage EFD. In the near future, we will submit each of the prototype EFDIs to rigorous usability testing in the laboratory. We will modify each of the prototype EFDIs based on the results of such testing. Additionally, we will establish their usability in low-to-medium fidelity simulations. We recommend continued testing in more complex high-fidelity simulations that will empirically test the prototype EFDIs, compare them to FPSs, and assess their effect on controller workload, communication, situation awareness, and overall performance. A provisional patent application is pending for the EFDIs.

References

Ahlstrom, V., & Longo, K. (Eds.). (2003). *The human factors design standard: Human interface guidelines* (HF-STD-001). Atlantic City International Airport, NJ: Federal Aviation Administration, William J. Hughes Technical Center.

Alexander, J. R., Alley, V. L., Ammerman, H. L., Fairhurst, W. S., Hostetler, C. M., & Jones, G. W., et al. (1989). *FAA air traffic control operations concepts. Vol. 7: ATC tower controllers* (FAA Report No. DOT/FAA/AP-87-01). Washington, DC: Federal Aviation Administration.

Ammerman, H. L., Becker, E. S., Jones, G. W., Tobey, W. K., & Phillips, M. D. (1987). *FAA air traffic control operations concepts. Vol. 1: ATC background and analysis methodology* (FAA Report No. DOT/FAA/AP-87-01). Washington, DC: Federal Aviation Administration.

Apple Computer. (1985). *Inside Macintosh, Vol. 1.* Apple Computer: Cupertino, CA.

Berndtsson, J., & Normark, M. (1999). The coordinative functions of flight strips. In *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work* (pp. 101-110). New York: ACM Press.

Besnard, D., & Cacitti, L. (2005). Interface changes causing accidents: An empirical study of negative transfer. *International Journal of Human-Computer Studies, 62,* 105-125.

Bruce, D. S. (1996). *Physical performance criteria for air traffic control tower specialists.* Washington, DC: Federal Aviation Administration.

Charness, N., Holley, P., Feddon, J., & Jastrzembski, T. (2004). Light pen use and practice minimize age and hand performance differences in pointing tasks. *Human Factors, 46,* 373-384.

Dattel, A. R., Johnson, B. R., Durso, F. T., Hackworth, C., & Manning, C. A. (2005). Flight progress strips in towers: Frequency index and perceived psychological benefits. In *Proceedings of the 13th International Symposium on Aviation Psychology*. Oklahoma City, OK.

Dayton, T., McFarland, A., & Kramer, J. (1998). Bridging user needs to object oriented GUI prototype via task object design. In L. E. Wood (Ed.), *User Interface Design.* CRC Press.

Department of Defense. (1999). *Human engineering design criteria for military systems, equipment and facilities* (MIL-STD-1472F). Philadelphia, PA: Navy Publishing and Printing Office.

Durso, F. T., & Manning, C. A. (2002). Spinning paper into glass: Transforming flight progress strips. *Human Factors and Aerospace Safety, 2*(1), 1-31.

Federal Aviation Administration. (2003). *Tower Display Integration Plan.* Washington, DC: Author.

Federal Aviation Administration. (2004a). *Air Traffic Control* (DOT/FAA Order 7110.65P). Washington, DC: Author.

Federal Aviation Administration. (2004b). *Operational Data Reporting Requirements* (DOT/FAA Order 7210.55C). Washington, DC: Author.

Federal Aviation Administration. (2005, August). *Administrator's fact book.* Washington, DC: Author.

Hewlett-Packard. (1987). *User interface design rules for the new wave office system.* Cupertino, CA: Hewlett-Packard Personal Software Division.

IBM. (1988). *System application architecture, common user access, panel design, and user interaction.* Boca Raton, FL: IBM.

Mertz, C., Chatty, S., & Vinot, J. (2000). The influence of design techniques on user interfaces: The DigiStrips experiment for air traffic control. *Proceedings of HCI-Aero 2000 International Conference on Human-Computer Interaction in Aeronautics.* Toulouse, France.

Mertz, C., & Lecoanet, P. (1996). Gesture recognition on interactive graphical radar image. GRIGRI: *Proceedings of the Gesture Workshop'96.* York, United Kingdom.

Mertz, C., & Vinot, J. (1999). Touch input screens and animations: More efficient and humanized computer interaction for ATC(O). *Proceedings of the 10th International Symposium on Aviation Psychology* (pp. 615-621). Columbus, OH.

Microsoft. (1995). *The Windows interface guidelines for software design.* Redmond, WA: Microsoft Press.

Pavet, D. (2001). Use of paper strips by tower air traffic controllers and promises offered by new design techniques on user interface. *Proceedings of the 4th USA/Europe Air Traffic Management R&D Seminar*. Santa Fe, NM.

Raskin, J. (2000). *The Humane Interface: New directions for designing interactive systems.* ACM Press.

Rogers, W. A., Fisk, A. D., McLaughlin, A. C., & Pak, R. (2005). Touch a screen or turn a knob: Choosing the best device for the job. *Human Factors, 47,* 271-288.

Ryan, C., van Schyndel, R., & Kitchin, G. (2003). Design and usability evaluation of a GAAP flight progress monitoring system in a simulated air traffic environment. In *Proceedings of the Australasian Computer-Human Interaction Conference* (OzCHI03) (pp. 147-156). Brisbane, Australia.

Truitt, T. R. (2006). *Electronic flight data in airport traffic control towers: Literature review* (DOT/FAA/CT-05/13). Atlantic City, NJ: U.S. Department of Transportation, Federal Aviation Administration.

Truitt, T. R., Durso, F. T., Crutchfield, J. M., Moertl, P. M., & Manning, C. A. (2000). Test of an optional strip posting and marking procedure. *Air Traffic Control Quarterly, 8*(2), 131-154.

Walker, N., Meyer, D. E., & Smelcer, J. B. (1993). Spatial and temporal characteristics of rapid cursor-positioning movements with electromechanical mice in human-computer interaction. *Human Factors, 35,* 431-458.

Walker, N., Philbin, D. A., & Fisk, A. D. (1997). Age-related differences in movement control: Adjusting submovement structure to optimize performance. *Journal of Gerontology: Psychological Sciences, 52B,* 40-52.

Woodson, W. E., Tillman, B., & Tillman, P. L. (1992). *Human factors design handbook: Information and guidelines for the design of systems, facilities, equipment, and products for human use* (2nd ed.). New York: McGraw-Hill, Inc.

Acronyms

AAS	Advanced Automation System
AFTIL	Airway Facilities Tower Integration Laboratory
AOC	Airline Operations Center
ASDE-X	Airport Surface Detection Equipment – Model X
ATC	Air Traffic Control
ATCT	Airport Traffic Control Tower
ATEAM	Air Traffic Terminal Enhancement and Modernization
ATIS	Automatic Terminal Information Service
ATO-T	Air Traffic Organization - Terminal
BOS	Boston-Logan Airport
CID	Computer Identification
CPC	Certified Professional Controller
DESIREE	Distributed Environment for Simulation, Rapid Engineering, and Experimentation
EDCT	Expected Departure Clearance Time
EFD	Electronic Flight Data
EFDI	Electronic Flight Data Interface
EFDS	Electronic Flight Data System
EFSTS	Electronic Flight Strip Transfer System
FAA	Federal Aviation Administration
FDE	Flight Data Element
FPS	Flight Progress Strip
GUI	Graphical User Interface
HFDS	Human Factors Design Standard
NAS	National Airspace System
P-S	Perceptual-Spatial
RDHFL	Research, Development, & Human Factors Laboratory
SME	Subject Matter Expert
TIPH	Taxi-into-Position-and-Hold
TMU	Traffic Management Unit
TRACON	Terminal Radar Approach Control
UTC	Universal Coordinated Time