FUTURE EN ROUTE AIR TRAFFIC CONTROL WORKSTATION: BACK TO BASICS

Ben Willems, Federal Aviation Administration, Atlantic City International Airport, New Jersey

Abstract

The expected increase of air traffic by at least 33% by 2015 to 2020 will require more than an evolutionary change from the way air traffic controllers work today in more than an evolutionary manner. One way to do this is to free up individual air traffic controller physical and mental resources. If controllers can apply the increase in available resources to air traffic control, we expect that they will have more capacity to absorb an increase in air traffic. To make these resources available we will use human factors principles to integrate available data and provide that data to controllers in an efficient presentation format.

We report on the development of a concept software platform that integrates data obtained from existing automation tools with available National Airspace System (NAS) data. The integration takes place at the Human Computer Interface and attempts to make that interface easy to use by applying human factors principles and leveraging existing air traffic controller expertise. We will discuss why we must present National Airspace Data in an integrated manner. We will also present how we intend to assess if our approach has succeeded in freeing individual air traffic controller resources.

Introduction

The air traffic controller occupation has gone through a long evolution since the use of bonfires and flags to direct traffic (for an excellent history of air traffic control in the United States, we refer the reader to [1]). Although controllers used maps, rulers, and radio communications, the mental model of the organization of airspace and aircraft within it resided mostly in the controllers' head. The use of maps and radio communication was probably the earliest attempt to provide controllers with information that could help them understand the airspace and air traffic situation. With the invention and introduction of radar, we provided controllers with additional information. The radar data displayed on horizontal scopes gave controllers a much more accurate idea of the location of an aircraft.

The radar displays presented aircraft position as well as video maps of the airspace. Controllers needed more information than just the location. As a result they developed "shrimp-boats". Shrimp-boats are small pieces of plastic that controllers used to document pertinent information such as an aircraft callsign, altitude, and speed. They moved these along the radarscope following the movement of the primary radar target. Linking the position of aircraft with other flight data was the responsibility of controllers until computers made correlation possible. Recognizing the need for a more automated system to keep track of the aircraft state, into the aircraft data block replaced the shrimp-boat. Several types of data blocks exist. They all provide an easy means for controllers to determine aircraft information at the time of an automated update.

Through most of the evolution we have supported controllers by automating routine tasks and assisting information integration where necessary. Many of the more advanced tools that we have introduced over the last decade, however, attempt to assist controllers by removing or supporting cognitive tasks (for an example of a possible evolution of the en route sector see [2]). Examples of these tools are conflict probes and metering tools. Most of the tools had an entirely separate development cycle. As a result the Agency is now implementing tools with automation functions that the NAS at some point, needs to integrate into the controller workstation. One example is the absorption of many aspects of the User Request Evaluation Tool (URET) into the En Route Automation Modernization (ERAM) system [3]. Our Agency is aware of the challenges integrating diverse technologies will create and we have developed concepts on how that can be done (e.g., URET integration with data link [4]). One of the challenges of this integration is to decide on how to integrate automation functions at the human-system interface.

In this paper we will discuss the approach we have taken in developing a concept for the integration of existing automation functions and available data at the user interface with the NAS. This approach takes advantage of available automation and data. We are not creating new tools or adding additional data to the NAS. Instead we use what is already available, however, use it in a way that supports controllers when and where needed as recommended for multi-function displays [5]. Our focus in developing the integration concept is on providing support for primary ATC tasks while off-loading secondary tasks where possible. We thereby attempt to enable controllers to go back to basics, i.e. to the control of air traffic.

We will present four areas where going back to basics may prove useful: information presentation, information integration, controller scope of operations, and human factors considerations in automation. In each of these areas we are looking for opportunities to reduce the time and effort to get exchange relevant information with the NAS.

Information Presentation

For a tactical controller, the display of data where and when needed, often means that we need to present data on the radar display close to or in the aircraft representation. Our philosophy is to stay as close as possible to the aircraft representation that controllers have used for several decades. When we evaluate information presentation we ask ourselves if we can provide (in a very basic manner) an indication that information is available. The information we present to controllers has to be consistent between information displays and connect information related to the same object.

Primitive Status Indicators

We need to provide controllers with an indication that new information is available, but leave it to the controller to decide when to access and how to use that information. The indication of availability of new information reflects a status change of the aircraft representation. In the controller pilot data link communications (CPDLC) environment, for example, the aircraft representation indicates the fact that an aircraft has switched to the sector frequency by changing the CPDLC status indicator. The indicator is primitive in the sense that it is a basic geometric shape and shape and location coding indicates the CPDLC state of the aircraft. In the NAS we have used such primitive coding techniques for many years, although we may not have recognized it as such. One example is the change from an aircraft being within its conformance boundaries along its route to it having deviated from its route. The only change in the aircraft representation is that the position symbol changes from a diamond to a triangular shape. The use of such primitive indicators enables controllers to quickly determine the state of the aircraft and to decide if the situation calls for more detailed information. In our approach to displaying status information to controllers we have adopted the use of primitive indicators as well.

Present Information on Demand

We then make more detailed information available when and where a controller needs it. More pertinent data is available with little effort while less pertinent and more detail is available with a little more effort.

An example of how we could improve display data when and where needed is the display of indicated airspeed. Currently controllers either intuitively know the indicated airspeed when they absorb groundspeed and aircraft data from the display or they call the pilot to ask what an aircraft's indicated airspeed is. In the former case, controllers perform a mental transformation to go from groundspeed to indicated airspeed; in the latter case, controllers have to contact the pilot, request the indicated airspeed, determine what indicated airspeed will correspond to the desired groundspeed and finally call the pilot with an instruction to change the indicated airspeed. Some automation tools calculate indicated airspeed based on aircraft characteristics, groundspeed, weather data, and altitude. If we use the data available in the automation tools, we can provide controllers directly with the indicated airspeed when and where needed.

Consistency between Information Displays

When different displays present information on the same objects in different formats, the operators need to perform a translation of one or both formats to a mental representation. We therefore suggest to maintain information presentation formats identical across information displays. In the current environment, for example, flight plan information on flight progress strips, computer readout device (CRD), and URET's aircraft list (ACL) are all in a different format.

Connect Related Information

If we display data related to the same object across different displays or across different locations within a display, connecting these representations will enable operators to quickly find that data. This reduces the search time needed when controllers need to move from one information display to another. Researchers at the National Aeronautics and Space Agency (NASA) presented a good example on their Center TRACON Automation System (CTAS) tool. On the CTAS plan view graphical user interface (PGUI), for example, the EDA presentation of information includes a timeline as well as a two dimensional display of the traffic situation [6]. NASA created the PGUI as a research interface used in lieu of the plan view display NAS (PVD). When a controller uses the PGUI and selects an aircraft on either the timeline or the traffic display, the other representation will show an emphasis as well.

The underlying concept to connect related information eliminates some of the searching that controllers need to do when moving from the traffic display to a list or another display. When extrapolating this principle, we can choose to emphasize all representations of a selected aircraft. If we no longer restrict ourselves to one and the same flight to simultaneously emphasized objects, we can further assist controllers in their tasks by extending the principle to features other than the callsign. For example, we have created an emphasis function that enables controllers to quickly display aircraft that have a particular feature (e.g., altitude) in common. Such a temporary emphasis supports controller perception, because it reduces the amount of scanning for information a controller needs to find aircraft that share the same feature.

Interactive Full Data Block (FDB)

Until one of the recent upgrades to the Display System Replacement (DSR), controllers only interacted directly with aircraft representations when they either picked (a left trackball button click on the position symbol) an aircraft, or selected an aircraft (a center trackball button click on the position symbol). With the recent DSR upgrades, the aircraft representation has become much more interactive. Examples include the emphasis of an FDB by hovering the trackball curser over the FDB; requesting a flight plan readout by hovering over the FDB and clicking on it with the center trackball button; choosing a different interim or assigned altitude by clicking on an altitude field; and choosing a coordinated heading or speed through clicking on the CID and groundspeed fields respectively. In the current implementation of DSR, controllers can make changes to interactive fields by both keyboard entries and use of the trackball or by using the trackball to click on the field. When using the interactive field, the system displays a small interactive menu off the FDB with the current value emphasized and three values lower and higher values above and below the current value. Initially developed with the CPDLC in mind, the FAA introduced the non-CPDLC flyout windows in one of the recent DSR upgrades.

We can find the idea of using a menu similar to Figure 1 in research conducted at Eurocontrol and other research groups and implemented in several countries. The ATC workstations used in the systems that use such a menu often do not have keyboards and it therefore makes sense to create an interface that is a fully Windows Icons Menus Pointer (WIMP) system. The advantage of a full WIMP system is that it can support direct manipulation of objects on the display. A drawback, of course, is that alphanumeric input that the user cannot select from a menu becomes awkward (by using a screen-based keyboard for example).

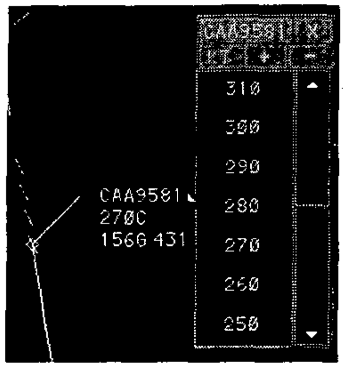


Figure 1. Example Flyout for Coordinated Speed

So why are we using it in the US? In the CPDLC Build 1A interface, the flyout window for CPDLC equipped aircraft had an option to simultaneously update NAS and uplink a message to an aircraft. Almost at the same time, CAASD, the developer of URET, published some material on the Assisted Resolution Tool (ART). ART used color coding of menus to indicate if changing an aircraft altitude and other interactive fields would result in a potential conflict. So, at first glance the use of flyout windows may be beneficial. The literature, however, reports that menus are especially useful for novice users, but are too slow for expert users. When we evaluated some of the existing WIMP techniques to change a field we noted two things. First, the flyout window is part of a continuum of menus and lists (Figure 2). The URET altitude window displays many or all altitudes simultaneously and is at one extreme of this continuum. The flyout window sits somewhere in the middle of that continuum (Figure 2). The other extreme is an interactive presentation of a single value. If we then anchor that window in the same location as the original field, we have created an interactive field. We have seen the use of such elements in the STARS CHI [7]. If we use a similar interaction scheme as controllers and human factors specialists chose for the STARS CHI (albeit not for interaction with FDBs, but with some of the fields in the toolbar), controllers click on a field, then move the trackball up and down to scroll through the list of values.

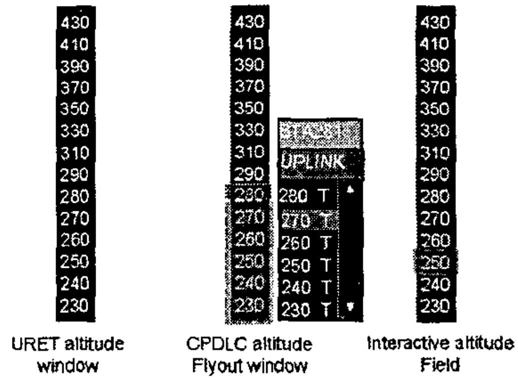
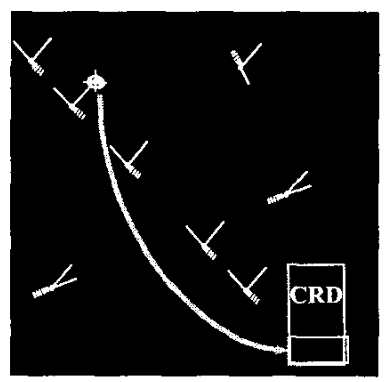


Figure 2. Flyout Windows as One Window on a List of Values

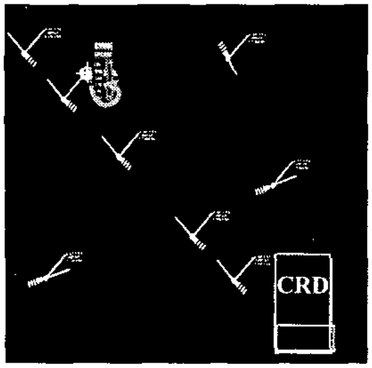
There is a clear advantage of moving from showing all values to the right side of the continuum. In Figure 3 we have depicted a schematic version of a DSR display. The location of the computer readout device (CRD) feedback area is often quite a distance from the focus of attention. Therefore, to move between the feedback area and the aircraft that is of interest to the controller requires substantial effort. In addition, to change a field, the controller must use the keyboard as well.



For a tactical controller, the location of the CRD and other windows and lists will require large eye movements, potentially interrupting the visual scan

Figure 3. Focus Before Interactive FDBs

With the introduction of the interactive FDB, it became possible to keep the visual attention close to the aircraft representation (Figure 4). We can go even further and make it unnecessary to move the focus of attention during interaction scrolling up and down through the list of values in the interactive field itself.



The time to jump (a saccade) depends directly on the distance traveled. The flyout windows will save time, because they potentially shorten the distance to get to information.
Although interactive FDBs have potential and may be thought of as reducing heads-down time, it will still interrupt the flow of the visual scan. Therefore always provide controllers with the option to enter data directly from the keyboard.

Figure 4. Focus with Interactive Flyout Windows

Although the introduction of scrollable fields may reduce the number of interactions and the number of times a controller needs to refocus, there is still a drawback to using the interactive fields. The use of interactive fields requires the controller to focus on the field. When using the keyboard, the data entry task is using the motor channel. When a controller needs to lock the focus of attention onto an interactive field, there will be a corresponding reduction of sampling other areas of the display. The reduction in scanning the display for information potentially leads to less awareness of the overall traffic situation. Controllers refer to that as tunnel vision. Tunneling of attention occurs when controllers focus on one area so intently that they forget to update information present at other locations of the display.

Information Integration

The NAS, as it currently exists, contains a wealth of data. Although we are using some of the data to support controllers in their task to keep aircraft separated and guide them along efficient routes, we have limited ourselves unnecessarily. We can leverage many of the information integration functions that the NAS currently uses. Through extrapolation or generalization of the current functionality we can better support controllers. The following sections will address several of these functions.

Emphasis Function

Controllers currently have a Quick Look (QL) function available that enables them, as the name implies, to quickly look at air traffic that is under control of another sector. NAS has extended that functionality by providing flow sectors that seem to have only aircraft going to a particular airport. The integration function in the QL is to briefly present extra detail where and when controllers need it and by using a common feature of aircraft (sector ownership). We can take advantage of this principle by using other features for a QL. Altitude, for example, is another feature that aircraft representations carry along. By applying the QL principle, we have created an emphasis function that enables controllers to briefly emphasize aircraft sharing the same altitude. We have not generated extra data, but have taken advantage of existing data to assist controllers to perceptually group aircraft sharing a feature for a limited time from other aircraft representations on the display. Controllers can use the emphasis to reduce the number of eye movement fixations necessary to find which aircraft are at same altitude as an aircraft that is about to enter the sector airspace. We have extended the emphasis function to other aircraft features such as destination, a navigational point on the filed route, etc. To not overwhelm controllers with new functionality we have integrated the emphasis function by creating a key that replaces the flight identity in the controller input grammar. Currently a controller would enter:

QU WPT ACT123

to indicate that the controller instructed an aircraft ACT123 to change its route (the QU command) to fly direct to waypoint WPT. To emphasize all aircraft that have WPT in their route, in the new interface a controller would enter:

QU WPT <EMPHASIZE>.

In the example above <EMPHASIZE> indicates the use of a special function key labeled "EMPH."

Conflict Probe

The FAA is currently implementing a medium term conflict probe (MTCP). The MTCP concept has a research history of several decades, but has not been available to controllers in the field until 1995 as a prototype and now as an operational tool [8]. The MTCP that the FAA is implementing is part of the URET. URET is currently available on the Radar Associate position and provides strategic guidance to resolve potential loss of separation, but controllers cannot use the URET data to tactically separate aircraft. The location of the URET display of data forces controllers to integrate data within the mental picture that controllers have of the traffic situation. URET has become more and more integrated with air traffic control system. URET had a separate keyboard and mouse during its introduction as a prototype, the keyboard functions and pointing device are now part of the radar associate keyboard and trackball. The conflict probe data now are part of the DSR, but we have not integrated them into the main radar display yet.

Controller Scope of Operations

Controllers have been able with the assistance of a large technical support network to maintain an extremely safe system. The NAS limited the amount of effort needed to maintain that level of safety by providing controllers with relatively small pieces of airspace called sectors. Within the NAS the traffic management units (TMU) attempt to ensure that a sector will not receive more than the limit set for that sector. The maximum number of aircraft that a sector can control depends among others on the size of the sector and the complexity of the flows of traffic within in the sector. A controller team is responsible only for the traffic in the sector, for separation assurance between aircraft and between aircraft and airspace, and for coordination with adjacent sectors or facilities (e.g. [9]).

Our Agency often receives criticism that use of the sector-based approach can lead to inefficiencies in traffic patterns. However, facilities created sectors around the route structure and the routes depended on ground based navigation equipment. The inefficiencies therefore are more the result of using the route structure than of using sectors. Airlines, of course, would prefer the most fuel efficient flight path from airport of origin to airport of destination while flying on-time every time. Changes in efficiency directly affect an airline's profit margin.

The flying public experiences the inefficiencies in delays or increased ticket prices. To address these concerns a movement started within the aviation community that supported a change from sector-based to trajectory-based air traffic control [10 - 12]. Such a change, however, would drastically change the controller's job, because most of the proposals suggest that controllers will need to handle aircraft that are well beyond the sector boundaries. Concepts like a multi-sector planner, an airspace coordinator, upstream D-sides and the like were the result of the trajectory-based school of thought [13].

When we take a look at the sector distribution in the NAS, we will see that sectors become smaller when getting closer to airports. Although not expressed by any of the airspace designers, it very much resembles a finite element mesh used in other domains to model non-linear behavior by linearization within cells. In our case the sectors form our cells. Each of the sectors has a design that enables controllers to move traffic safely and efficiently through its airspace. This does not mean that trajectories that cut through these sectors need to be inefficient, but it does mean that quite a bit of coordination is necessary to get an aircraft from the airport of origin to its destination. One of the assumptions made in the trajectory-based approach is that to be able to create and maintain efficient trajectories controllers will need to change their operations from sector-based to trajectory-based. In reality, what is necessary is a system that optimizes the full trajectory. Currently that is in the hands of Airline Operations Centers (AOC), the Air Traffic Control System Command Center (ATCSCC) and the TMUs at the air traffic facilities.

We suggest that we can integrate a trajectory-based approach into sector-based ATC. In our concept of the future en route sector we go back to basics by maintaining sector-based control. Controllers are very familiar with this concept, have a clearly geographically defined area of control, and have a portion of airspace that is manageable. Trajectory-based control can take place at a higher level and, in fact, some of the automation tools already provide such a function. In a future sector-based concept the distribution of the roles and responsibilities among controllers within a sector may change, but the sector structure stays in place. Under current procedures, controllers manage ATC events. One type of event originates from within the sector (a potential conflict, local weather conditions, or an aircraft that needs to make vertical transition through the airspace for example).

Another set of events are external to the sector (an adjacent sector or facility requests assistance or the supervisor tells the controller to implement a flow restriction). The actors in these events are pilots, controllers, supervisors, and traffic management coordinators.

We suggest extending the current sector-based procedures to include an extra actor, i.e. the NAS automation. NAS automation requests could arrive at the sector for several reasons. For example, if the TMU wants aircraft rerouted, a controller could receive that as an external request. The reroute could be for weather, reduction of traffic complexity, or to accommodate a change in airport acceptance rate. Controllers in our view of the future sector-based NAS have control of the sector and receive requests from pilots, other controllers, traffic and flow management, and the automation system.

Human Factors Considerations in Automation

The fourth area that we try to bring back to basics concerns itself with human factors considerations in automation. One of the most difficult topics in automation is to decide what to automate and what not. Fitts [14] provided us with some guidance by listing functions that he allocated either to a human operator or an automation system. The implementation of his advice has been far from trivial or has been absent altogether. Fitts' list [15] may have changed a little as far as data storage capabilities in machines, but other than that, the list is still applicable to allocation of functions in the human/automation environment (see Table 1).

Table 1. Fitt's List Adapted From [15]

Humans appear to surpass present-day machines with respect to the following:

- Ability to detect small amounts of visual or acoustic energy;
- Ability to perceive patterns of light or sound;
- Ability to improvise and use flexible procedures;
- Ability to store very large amounts of information for long periods and to recall relevant facts at the appropriate time;
- Ability to reason inductively;
- Ability to exercise judgment.

Present-day machines appear to surpass humans with respect to the following:

- Ability to respond quickly to control signals, and to apply great force smoothly and precisely;
- Ability to perform repetitive, routine tasks;
- Ability to store information briefly and then to erase it completely;
- Ability to reason deductively, including computational ability;
- Ability to handle highly complex operations, that is, to do many different things at once.

In our approach to applying human factors we have attempted to use as much as possible the things humans are good at and automate the other activities. One way to free up available resources is to automate repetitive routing tasks.

Repetitive Routine Tasks

In air traffic control we have introduced many automation systems that the current users of the system take for granted. The availability of aircraft data on the radar display other than the position derived from the radar reflection is such an example. Before the integration of beacon code, callsign, altitude, speed, and heading, controllers maintained that data either on artifacts (shrimp boats) or in memory. The NAS has many more automation features that assist controllers in removing repetitive routine tasks to free controller resources. A few examples are:

- Automatic handoff initiation to the next sector if an aircraft is following its current flight plan route within certain conformance bounds.

- Automatic data block orientation for a certain sector is selectable in the adaptation Host Computer System
- Automatic generation of flight progress strips at a sector when the HCS projects that aircraft will fly through a fix posting area belonging to that sector
- A change of the position symbol based on the state of aircraft and its position data

While the NAS evolved and assisted controllers in keeping up with increases in traffic volume and complexity through automation changes, the agency foresaw that the human operators would need more assistance to cope with the continued increase in traffic. Plans to create a system that would support controllers in conflict detection, conflict resolution, and efficient metering of traffic into airports suggested that automation could replace or augment a large portion of the cognitively more challenging controller tasks. In our focus on assisting in those tasks that required higher cognitive skill, however, we have lost sight of the opportunities to further alleviate the demand on controller resources for administrative or menial tasks.

What repetitive tasks are potential candidates for automation? Our simulations indicate that controllers participating in our experiments use about 25 percent [16, 17] of their interactions with the system to move data blocks. Because such high numbers could be an artifact of our simulation environment, we have taken a brief look at data on controller activities in ARTCCs before we introduced the DSR. Although we have only had the opportunity to take a cursory look at the data, the distribution of controller interactions with the system shows clearly the bulk of the interactions that accepting and initiating handoffs combined with moving full data blocks or toggling full data block display on and off (Figure 5). In Figure 5 QP represents actions like creating a halo around an aircraft for separation; QF a flight plan readout; QU a route display or change; QZ an assigned altitude change; QQ an interim altitude removal or change; and QN data block offsets, handoff acceptance or initiations, and forcing data blocks visible onto the display.

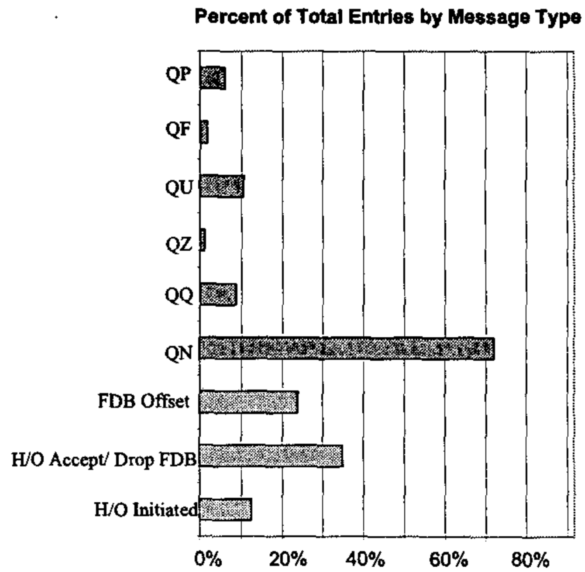


Figure 5. Example of Percent of Total Pre-DSR

Transfer of Control

As you can see from Figure 5, the number of handoffs initiated by controllers is much lower than the number of handoffs accepted. Three sources are responsible for this difference. First, controllers can force the display of a full data block by entering a flight ID through the keyboard or a click on a position symbol. Secondly, controllers tend to drop the FDB when they are done with an aircraft. That is, the next sector or facility has accepted the handoff on the aircraft, instructed the pilot to switch frequency, and the aircraft has physically left the sector. Thirdly, the automatic handoff feature that currently exists in the HCS is partly responsible for that difference. The principle behind automating handoff of aircraft that are conforming to their flight plan (maybe not stated explicitly) is to automate the repetitive and routine actions while providing options to intervene when exceptions occur. So, why have we not automated handoff acceptance? Most of the time controllers will accept the handoff on an aircraft that will enter their airspace. Controllers, of course, will need the option to interrupt an automated acceptance similar to what is now available for automatic handoff initiation.

The CPDLC program could result in a drastic reduction of verbal communications depending on how many airlines equip their aircraft. The introduction of CPDLC promises to reduce frequency congestion by eliminating voice communication related to altimeter settings, initial contact, and switching to the next sector's or facility's frequency. Together with automatic handoff and automatic acceptance this could result in a seamless transition from one sector to another without radio contact or controller display interaction. Currently, however, CPDLC only exists in an automatic handoff and manual transfer of control (TOC) configuration. This still requires controllers to physically accept a handoff and release a held TOC. Although this may not be an issue at current traffic levels, it will become an issue once traffic levels increase.

A word of caution is appropriate here. When we automate repetitive routine tasks, we still need to inform controllers that automation has completed such tasks. The design of the CPDLC system has given great care to providing controllers with information about the status of tasks that controllers have handed over to the automation. For automating other routine tasks such as automatically accepting handoffs and frequency switching we must provide controllers with information about the state of the task that controllers now expect to take place automatically. For example, the initiating controller still needs to be able to see that an aircraft changes to handoff mode, the next sector has accepted the handoff, the aircraft is switching to the next frequency, and has switched to the next sector.

Most controllers currently drop the FDB after the aircraft is the full responsibility of the next sector and has left their sector. Once the aircraft have entered that phase, however, NAS knows that the aircraft has left the sector and with CPDLC will know that the frequency has switched. We can therefore automate the drop of the FDB as well and do that in a similar fashion as URET currently does that for flight plans on the URET aircraft list. In URET, however, flight plans that the next sector has accepted will grey out and disappear automatically after several minutes. Some of these repetitive tasks may be candidates for automation.

Ensuring Proper Information Display
Although this task includes ensuring that tracked aircraft within the physical sector boundaries display FDBs, most of the activities related to proper display of information involve offsetting of FDBs to ensure that they do not obscure pertinent data of other aircraft. In the terminal ATC environment automatic FDB offset is available, but many controllers turn that automation function off, because the algorithm uses the cardinal orientations of the leader line, resulting in FDBs jumping from one position to another. At Eurocontrol Experimental Center, Dorbes [18] developed a requirements document for the automatic resolution related to FDB overlap. Dorbes assumed that FDBs move in a fluid motion, but this is currently not done in the US NAS. To implement such a system, FDBs will need to be able to move smoothly to avoid overlap and to prevent a jump of the FDB. The use of an automatic FDB offset function could reduce the number of controller interactions dramatically.

The current trend in the evolution of the aircraft representation on the ATC display seems to be to include data that was previously only available on flight progress strips as controller annotations. Examples include coordinated speed and heading, free text, aircraft destination, and aircraft type. The inclusion of the extra data will make the aircraft representation unwieldy as shown by Potter [19].

In Figure 6, controllers have detailed information about the current state of the aircraft while other information depicts the status of communications with the aircraft through CPDLC and the advisories from automation tools. To fulfill their primary task, i.e., provide separation services, controllers need the current state of the aircraft and possibly predicted conflict information. If controllers continue to work in sectors similar to those that we currently have, in Figure 6 we have potentially three requests from two different sources. First, the controller received a "Stand by" message related to an earlier uplink. Secondly, the pilot has requested to fly heading 250 and climb to flight level 370. Thirdly, the metering system requests that the aircraft loose one minute and ten seconds.

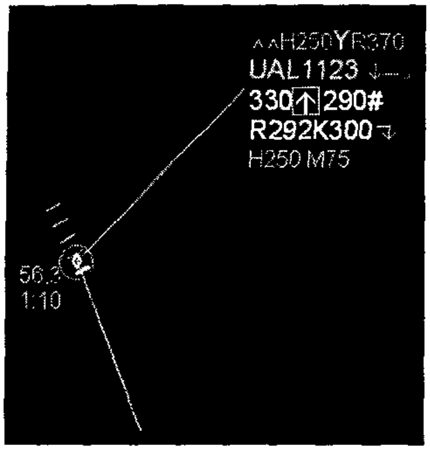


Figure 6. Potential Evolution of FDB

We can see of course that the FDB in Figure 6 is just a hypothetical example, but the aircraft in fact only has two requests at this point. One request comes from the pilot and one indirectly comes from a traffic management entity. Instead of providing controllers with detailed information, we suggest to redesign the interface to clearly indicate that the sector has received external requests or advice. This approach reduces the amount of clutter on the display thereby reducing the chance that one data block obscures data contained in another data block. Once a controller has time to look at external requests, s/he can bring up the detailed information needed to decide which request to address first. Conflict probe results have a similar function, i.e., they provide controllers with information that, if the controller does not take action, the system has detected a potential separation violation.

The advantage of reducing the chance of information overload by providing only basic status indicators is that the aircraft representation stays much closer to the stimulus that controllers have used for decades thereby taking advantage of the expertise that current controllers have in processing the stimulus information.

In Figure 7 we have depicted the aircraft representation that we will use in the future en route workstation experiment (FEWS). For the aircraft depicted in Figure 7 a controller can see that this aircraft has a potential conflict (the red dot at the end of the first line), is CPDLC equipped, logged in, and on the sector frequency (filled in rectangle at the beginning of the first line), has coordinated data (a heading of 250 and a Mach speed of 0.75 in line 4), and is climbing (up arrow in the center of line 2) through flight level 290 (Mode C indicated on the right hand side of line 2) to flight level 330 (Assigned altitude indicated on the left hand side of line 2). The aircraft is flat tracking (indicated by the diamond position symbol). The system will display additional information only when and where a controller needs it.

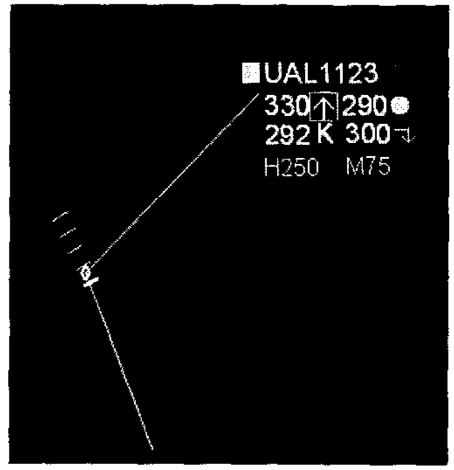


Figure 7. Basic FDB in the FEWS Experiment

Information Filtering and User Preferences

En route controllers have for quite some time now used a digital representation of aircraft position and related data. That has given them the opportunity to filter the information they receive. Controllers can, for example choose not to display aircraft that are outside of an altitude stratum that includes their airspace. This capability removes a lot of visual clutter, because it eliminates aircraft representations below and above the sector altitude stratum. So, how far should we go with the ability to filter data? On DSR almost everything has toggle and brightness settings. But because we can turn all callsigns off on the display, does that mean that we should? Consensus on what to display and how will probably never occur. The answer, however, is not to make everything user selectable [citation]. Filtering of aircraft that a controller currently has under control and on the frequency by using color or intensity, for example, has led to problems that Eurocontrol has documented. By allowing end-users (in our case controllers) to use presentation features to set a group of information carrying objects apart from other object on the display, we set them up to implicitly learn to ignore objects that they may feel are less relevant. In the case of ATC, controllers may have implicitly learned many processes, but we need to take care not to trigger that behavior when it has unwanted consequences. Counter arguments of course include that ignoring certain objects may be the goal of setting them apart. We can do that, however, without causing implicit learning by giving controllers the option to emphasize certain groups of aircraft, but to remove that emphasis after a brief display.

Discussion

The projected increase air traffic by 2015 will result in many challenges. The current NAS still has potential to free up resources if we use available data in more creative ways. We have analyzed the current workstation and presented concepts for enhancing controller interactions in a future environment. Although at first glance we seem to remove time and steps necessary to interact with the NAS, thereby enabling controllers to focus on separating aircraft and moving aircraft through the airspace, only a formal experiment will provide us with data to determine if our concepts have the anticipated effect. To objectively determine the effects of changing the interface to support controllers, we have instrumented our simulation environment with measures that capture the time and number of events involved in controller interactions with the system. The anticipated benefits of the changes we are introducing are a reduction in workload and an increase in situation awareness, safety, and efficiency. In an experiment scheduled for early 2005 we have implemented changes to the en route workstation that should enable controllers to handle current traffic better and control traffic at higher levels than with the current workstation design.

References

[1] John Schamel, 2003, FAA history, The early years, Retrieved from http://www.ama500.jccbi.gov/afss/History/FAA.htm

[2] Celeste G. Ball. & G.J. Jacobs, 1999, Recommendations for R-side evolution: Initial candidates for evaluation (MP 99W000000018), McLean, MITRE Center for Advanced Aviation System Development.

[3] Federal Aviation Administration, 2004, NAS Architecture 5.0, Mechanism Data Report: User Request Evaluation Tool National Deployment, Retrieved from http://www.nas-architecture.faa.gov/cats/mechanism/mech_data.cfm?mid=687

[4] Brestle, Ed, Rich Bolczak, Joe Celio, Karol Kerns, Dave Winokur, 2001, Concept of use for integration of the user request evaluation tool (URET) with aeronautical data link system (ADLS) (MTR 01W0000081), McLean, MITRE Center for Advanced Aviation System Development.

[5] Mejdal, S., M.E. McCauley, & D. B. Beringer, 2001, Human Factors Design Guidelines for Multifunction Displays (DOT/FAA/AM-01/17), Washington, DC, Office of Aerospace Medicine.

[6] Richard Lanier, 2004, EDA Milestone 5.0, Unpublished manuscript.

[7] Raytheon, 2003, Standard Terminal Automation Replacement System, Technical Manual Instruction Book (Contract No. DTFA01-96-D-03008), Marlborough, MA, Author.

[8] Post, Joseph, David Knorr, 2003, Free Flight Program update, 5th USA/Europe Air Traffic Management R&D Seminar, Budapest, Hungary.

[9] Federal Aviation Administration, 2004, Air Traffic Controller's Handbook, FAA Order # 7110.65P, Retrieved from http://www.faa.gov/ATpubs/ATC/

[10] Ken J. Leiden, Steven M. Green, 2000, Trajectory orientation: A technology-enabled concept requiring a shift in controller roles and responsibilities, In Proceedings of the 3rd USA/Europe Air Traffic Management R7D Seminar.

[11] Vivona, Robert A., Mark G. Ballin, Steven M. Green, R.E. Bach, & B.D. McNally, 1996, A system concept for facilitating user preferences in en route airspace (NASA TM 4763), Moffet Field, CA, NASA Ames Research Center.

[12] Couluris, G.J., 2000, Detailed description for CE6, En route trajectory negotiation (NAS2-98005 RTO-41), Moffet Field, CA, NASA Ames Research Center.

[13] Ken J. Leiden, 2000, En route controller roles and responsibilities in support of en route descent advisor inter-sector planning, Moffet Field, CA, NASA Ames Research Center.

[14] Fitts, P.M., 1954, The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement, Journal of Experimental Psychology, 47, p. 381-391.

[15] Fitts P. M., 1951, Human engineering for an effective air navigation and traffic control system, Washington, DC, National Research Council.

[16] Willems, Ben, Michele Heiney, 2002, Decision support automation research in the en route air traffic control environment (DOT/FAA/CT-TN02/10), Atlantic City International Airport, Federal Aviation Administration, William J. Hughes Technical Center.

[17] Willems, Ben, Michele Heiney, Randy Sollenberger, 2002, Study of an ATC Baseline for the Evaluation of Team Configurations: Information Requirements (DOT/FAA/CT-02/17), Atlantic City International Airport, Federal Aviation Administration, William J. Hughes Technical Center.

[18] Dorbes, A. 2000, Requirements for the implementation of automatic and manual label anti-overlap functions, Bretigny-Sur-Orge Cedex, France, Eurocontrol Experimental Centre Publication Office.

[19] Potter, Robert, 2003, Presentation at the 3rd ICNS Conference, Fairfax, VA.