The Magellan User Interface Design Approach

Introduction

Magellan by Mercury Scheduling Systems is a real-time, "day of operations" airline crew scheduling system. I would like to describe the approach used to design the Magellan user interface.

Please note, although I am describing several steps in the design of the Magellan user-interface, it is important to remember that there was some iteration between the various steps.

1. Understand crew scheduling and get to know the users

The first step in designing the User Interface was to understand crew scheduling and to get to know our users. To achieve this, we conducted a series of interviews with real life airline crew schedulers. As a result of these interviews, we learned how scheduling disruptions occurred, the types of problems that caused the disruptions, the ramifications of the disruptions, how the resolution of these disruptions were prioritized, and how these disruptions were resolved. Scenarios were defined based on this series of interviews.

2. Define the Views and a navigation map describing how the user navigates between the views and the data they bring with them when navigating to the views

Based on the scenarios, we identified the main views that were needed. We also defined a navigation map that defined how the user would navigate between the various views and the data they would bring with them during view navigation.

We identified the following views:

- **Monitor** – Monitors your day and alert the user to any schedule disruptions.
- **Trip Repair** – Airline scheduling is based on Trips. Trip Repair is where Trips are created and managed for assignment to crewmembers. Trips are containers which contain work items such as Flights. A Trip might last a few hours, or it might last several days. A Trip also has one or more crewing requirements. E.g. one Captain, one First Officer, and three Flight Attendants.
- **Roster Repair** – This is where the user manages crew Rosters(crewmember schedules). Trips are assigned to crewmembers here.
- **Trip/Flight Search** – A search for Trips/Flights based on temporal criteria, qualification criteria, location criteria, assignment needs etc.
- **Crew Search** – A search for Crew Members based on temporal criteria, availability, location, qualifications etc.

Some of the View navigation we identified included:

- When the Monitor identifies a crewmember related disruption, the user can navigate from the Monitor to either the Roster Repair view with the problem crewmember(s).

- When working in Roster Repair, if other crewmembers are needed to solve the problem, the user navigates to the Crew Search view to find crewmembers that might help solve the problem. If Trips are needed to solve the problem, the user navigates to the Trip/Flight Search view.
- When the Monitor identifies a Trip related problem, the user navigates to the Trip Repair view with the problem Trip(s). If Flights are needed the user navigates to the Trip/Flight Search.
- Search results from Trip/Flight search can be brought into the Roster Repair and the Trip repair view.
- Search results from Crew Search can be brought into Roster Repair.

3. Divide the views into sections

The next step in the UI design was to take the various views and divide these views into sections. For example, the Roster Repair view was divided into an upper section and a lower section. The upper section contains one or more Crew Rosters(crewmember schedules). The lower section acts as a bin containing one or more Trips that can then be assigned to one of the Rosters in the upper section. When items are removed from a Roster, they appear in the bin.

The Trip Repair view was divided into similar sections as the Roster Repair view. An upper section containing one or more Trips, and a lower bin that would contain Flights that could be assigned to Trips in the upper section. When items are removed from a Trip in the upper section, the item appears in the bin.

The Trip/Flight search and Crew Search views were both divided into upper and lower sections. The upper sections of both views to contain search criteria. The lower sections to contain search results.

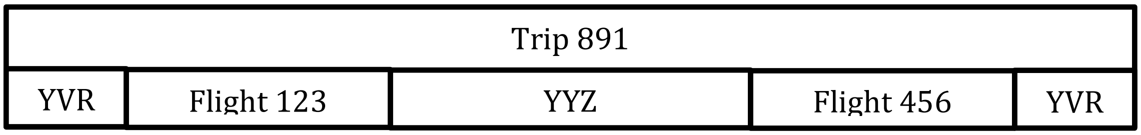
The Monitor just had one section to contain Trips presenting the current state of the schedule.

4. Define the detail that will appear in the various sections of the views

Given the fact that this system was a scheduling system, I wanted to show a time-based view with screen objects.

Trips are important as all airline scheduling is based on Trips. Trips need to appear in the Monitor, in crewmember Rosters, in the bin section of Roster Repair, in the upper Trip section of Trip Repair, as well as in the results area of the Trip/Flight Search view. Trips must contain Flight screen objects. As well we want to show the airports the Flights in a Trips started and ended at.

I designed a Trip screen object first. In the end, the format of the Trip screen object was along this line:



<table>
<tr><td colspan="5">Trip 891</td></tr>
<tr><td>YVR</td><td>Flight 123</td><td>YYZ</td><td>Flight 456</td><td>YVR</td></tr>
</table>

The next step was designing what a crewmember's Roster would look like. Rosters will be time-based and contain several Trip screen objects representing items in a crewmember's schedule. Rosters will be found in the upper area of Roster Repair and the search result area of Crew Search. In both cases we want to present several Rosters. Aside from schedules Trips, I laid out where the crewmember's name, qualifications etc. would appear in each Roster.

The Roster Repair bin area would contain Trip screen objects. Trips that are removed from a Roster appear in the bin. Trips in the bin can also be assigned to Crew Rosters in the upper section.

The next step was to think about the details that would appear in the Trip Repair view. The upper area would contain several Trip screen objects. The lower bin area would contain Flight screen objects. When editing Trips, Flights that are removed from Trips appear in the lower bin. Flights from the lower bin area could be assigned to Trips in the upper area.

The next view we needed to think about was the Monitor view. Essentially the Monitor needed to contain Trip screen objects. We also needed a way to alert the user to problems. Essentially we added color coded alert circles to the corners of the Trip screen objects.

The next view we needed to think about was the Crew search. The upper area of the view contains search criteria. This criteria was laid out in the upper area. The lower area needs to contain Rosters representing the search results. We used the same Roster design from the upper area of the Roster repair view.

The next view we needed to think about was The Trip/Flight search. In the upper area we laid out the search criteria. The lower area search results would contain Trips presented in the same way as they were presented in the upper area as Trip Repair. Flights would also be presented in the search result area in the same format as they appear in the bin area of Trip Repair.

5. Define a tight set of primitive Commands

It was time to start thinking about commands. The concept of "Cut" and "Paste" seemed very natural. From Roster Repair, cut a Trip out of a Roster and the Trip appears in the lower bin. Paste a Trip from the lower area bin into a Roster in the upper Roster area.

From Trip Repair, cut a flight from a Trip in the upper area, and it appears in the lower bin. Paste a flight from the lower bin into a Trip in the upper section.

The user would be warned of any illegal assignments via our alert mechanism (color coded circles in the upper corner of Trips).

There were a few more specialty commands, but the concept of “cut” and “paste” covered many of our bases.

We also decided how navigation would work. For example, when a disruption was identified in the Monitor view, we provided a Trip context sensitive menu item that allows the user to navigate to either Roster Repair or Trip Repair with the relevant data.

We also needed a way to transfer search results from the Crew Search and Trip/Flight Search views to both Roster Repair and Trip Repair. Crew Search results can be transferred to the upper area of Roster Repair. Trip search results from Trip/Flight Search could be transferred to the upper section of Trip Repair and the lower section of Roster Repair. Flight Results from Trip/Flight Search could be transferred to the lower section of Trip Repair. A selection method was provided in the search result area in both Search views where a user could select one or more results. The selected items were then transferred via a context sensitive menu item.

6. Define dialogs that were needed

Given that our user-interface was a time-based user-interface containing screen objects, we wanted to provide a detail dialog for every type of screen object appearing in the various views. Dialogs for the various screen-objects were designed.

We also designed a dialog that provided details on any Alerts.

7. Refinements

We came up with a few specialty commands to help save time. An example of one such a command was the Trip Trade command. This is an important command as it is quite common for two crewmembers to trade Trips. The user would select a Trip in one Roster in Roster Repair, select another Trip in another Roster, and via a context-sensitive menu item, the user could perform a quick Trip trade.

During a typical day, it is not unusual for a crew scheduler to put aside a problem they are working on to deal with a more pressing problem. A concept we came up with to deal with this was called Repair Focus. Repair Focus represented all of the items populating their various views. We provided a mechanism for the user to save a Repair Focus. If a more pressing problem came up, the user could save their current Repair Focus. The user could then work on the more pressing problem. When the more pressing problem is resolved, the user could then reload their previously saved Repair Focus and return to the original problem.

Conclusions

The Magellan user interface was successful. After we finished the design, we presented a static prototype of the user interface to some of our potential customers. In every case, after a quick presentation, these potential customers understood the user interface. It was very satisfying watching them figure out how they would solve the real life problems they faced everyday with Magellan via our static prototype.

Magellan Customers included British Airways and Maersk Air.