A Comparison of two Decision Support Systems Designed for a Dispatching Problem

Christopher S. Cutrona, Craig Pratsch, Camila Valcarengh, Anelise Welter, Lia Buarque de Macedos Guimarães, and Stephanie Guerlain, Senior Member, IEEE

Abstract—This paper compares two decision support system designs to aid in the daily dispatching of trucks delivering oil from Tenoas, an oil distributor, to hundreds of locations in the southernmost state of Brazil. On site research and interviews were conducted in order to determine the needs of the system and the greater implications for drivers, managers, and dispatchers. This analysis led to the functional requirements which, at the highest level, are to create a fair yet understandable and editable procedure for determining the daily dispatching of truckers. Both designs display customer orders and available trucks, and aid the user to manually or automatically assign drivers to routes by displaying relevant information needed for the decision making process. One uses a matrix layout and the other a map layout. The two designs are compared in their workflow and the likely impact they would have on the workers. The final system, if implemented, will have implications for many of the workers at Tenoas, including the programmers, who can create truck/location assignments more efficiently, the truck drivers who do not have to wait as long and are subjects of a fairer system, and the managers who are adopting a technology that is more socially responsible.

I. INTRODUCTION

The goal of any dispatching activity is to assign a resource to a specific destination in order to promptly complete a transaction or meet a particular need. Dispatching is a resource allocation problem with temporal and spatial components. Resources get allocated and scheduled and dispatched, but become available again to meet new requests that continue to come in. An example of this is the dispatching of trucks from an oil distribution center in order to deliver requested petroleum products to clients. In this example, the dispatchers decide which trucks (resources) are used to meet the amount and type of oil requested from a client in a particular location (demand). The dispatchers must manage time constraints, special requests, priority deliveries, capacity and compartmental constraints, and driver shift changes—which is considerably complex since this is a sequential decision making process and the dispatcher is handling multiple other tasks at the same time.

The scope of this project was to design a decision support system (DSS) to aid in the dispatching process. Previous research on this subject includes the Resource Allocation Planning System (RAPS) which uses a color-coded matrix to allocate resources for a demand. The RAPS also discusses the use of a map when visualizing spatial relationships important in the decision making [1].

The approach used in this project was to have all authors conduct a cognitive task analysis and define the functional requirements, but then to split into two independent design teams to create a solution to meet the same requirements. The intention of this approach was to compare the two resulting designs’ strengths and weaknesses and combine the best features. It was determined a priori that one design would focus on a matrix representation of the data and the other a geospatial representation, based on the recommendations of design approaches presented in [1] for problems of this type.

II. THE CASE STUDY

A. Understanding the Domain

We performed a cognitive task analysis through interviews and observations with the dispatchers in order to gain a complete view of the cognitive tasks, objectives and constraints involved in dispatching.

The Tenoas oil distribution center is responsible for the delivery of a variety of petroleum products throughout the southernmost states of Rio Grande do Sul and Santa Catarina by trucks and trains. Many customers have their own trucks and will call ahead to schedule a time to pick up their requested order. For those customers who do not have their own trucks, Tenoas has contracted two trucking companies, Stefani and Dalçoquoi to perform these deliveries. According to the contract, Stefani is to service approximately 40% of the monthly business and Dalçoquoi the other 60%. There are 31 total contracted trucks: 13 from Stefani and 18 from Dalçoquoi. The Stefani and Dalçoquoi dispatchers have been working there for 30 and 11 years.
respectively. At the end of each day, one of them (they trade off) performs the dispatching for the following day. This initial dispatching consists of “Round 1” in which they decide which of the 31 drivers goes to what posts, depending on the requests made that day. Only “Round 1” is scheduled the night before because the drivers return from their trips at different times and are dispatched again.

The dispatching is especially difficult because of the system constraints and other non-tangible factors. For example, the dispatchers have to consider the capacity of the requests, the number of compartments of each truck and their capacities, which trucks only go long distances, which routes cannot handle large trucks, whether the route is a “good” or “bad” route (a “good” route is a desired route, meaning it has a higher $/km traveled rate or has favorable road quality conditions), and the value of each route. The dispatchers try to balance the amount of money that the drivers have made to date. They also try to deliver product to multiple locations on one trip because the companies receive payments as if the total distance traveled was solely for each request.

This dispatching of trucks is not the dispatchers’ only job; they have many other tasks that they need to perform on a daily basis. The current dispatching system is performed manually and takes an unnecessarily long amount of time to complete. Drivers often have to wait for more than an hour to receive their dispatching orders. A decision support system should aid the dispatchers in their decision making process, without removing their power to make final dispatching decisions, and ultimately reduce the time it takes to make these decisions. A systematic process would also ensure that the dispatching task can continue efficiently as the current manual system would require intensive training for a new dispatcher to learn.

B. Functional Requirements

The functional requirements of the system were created to ensure that all of the cognitive tasks dispatchers need to perform would be met. In this problem, the DSS must allow the dispatcher to be able to add, delete, and edit all requests, clients, and drivers and display all current unfilled requests (including requests not delivered from the previous day). The system should show all possible drivers for a given request and show all possible routes for a given driver. It must calculate the value of a route, enable the assignment of a driver to a route, show when delivering to multiple locations is advantageous, delete a request after the driver returns, and log all actions. Additionally, the system should include a search function that allows the dispatcher to search for a city, driver, or client and sorting functions to sort drivers based on waiting time, estimated time of return, earnings to date, capacity, and company. Furthermore, the system should show the driver availability and unavailability, the driver waiting time at Tenoas, the estimated time of arrival at client sites and time of return to Tenoas for drivers en route, and any relevant details about each driver, request and client.

The system also needs to generate reports, e.g. to show history and relationships between data elements at a high level. The system must show the revenue balance between Dalçoquoii and Stefani while allowing the dispatcher to see this over selectable time periods, e.g. from the previous two weeks, month-to-date, year-to-date, and a custom display. The system should also show the details of drivers’ deliveries including the locations, dates, times, earnings, client, amount, and type for the same time periods. Likewise, the system must show the clients’ requests for these time periods and include the requested types, amounts, dates, drivers, and cost.

The data elements are the basic informational components of the system and can be represented using many forms (propositional, iconic, and analogical) [2]. These data elements are listed below:

- **Truck**: TruckID, license number, driver name and phone number, capacity of truck and number of compartments, money earned to date, average distance in city market, log of deliveries, earnings
- **Client**: Client ID, client name, city, phone number, contact name, log of deliveries made, distance, estimated delivery time, rate, preferred route
- **Request**: RequestID, client, city, amount, type, date/time constraint, date/time made, requested driver
- **Delivery**: DeliveryID, RequestID, TruckID
- **Route**: RouteID, DeliveryID, list of waypoints, total earnings, total time

III. DESIGN A – MATRIX REPRESENTATION

The matrix interface, shown in Fig. 1, is based on the design elements defined for resource allocation planning system as specified in [1]. It displays all possible driver/location assignments at all times to provide the user with a general overview of the system. The matrix interface concentrates on effectively displaying the driver wait time and evenly distributing the ‘good’ and ‘bad’ roads to support the scheduler in the decision making process.

To aid the schedulers in their decisions, the locations, whose requests are waiting a driver assignment, are listed along the top of the matrix in columns and the drivers that are currently waiting at Tenoas are listed along the left of the matrix in rows. If the intersection of a driver and location within the matrix is black, it is infeasible for a driver to deliver to that particular location. Intersections in the matrix that are color coded represent information about the drivers who are able to deliver oil to that location due to the size and capabilities of their trucks to hold various products.

Above the matrix, a function toolbar provides the user with further information or the ability to change the matrix. Some of the functions are to display driver pay histories, display the Delcoquoii/Stefani earnings ratio, change route assignments, or mark a driver as unavailable (e.g. if sick or truck is in maintenance). The sorting functions enable displaying the rows and columns in different orders to create new perspectives of the data. The row and columns can be sorted by driver wait time, truck capacities, road quality conditions, etc. There are also information boxes displayed along the bottom which provide additional information when a particular truck or location is selected. Information boxes
contain further information about the route assignments, locations and drivers.

Fig. 1 displays the color coding feature of the matrix display to denote the “good” and “bad” locations. The drivers are automatically color coded based on the number of ‘good’ and ‘bad’ routes they have received within the City Market. If a driver has delivered oil to a higher than average number of ‘good’ locations, they are displayed in green, if they have delivered oil to an average number of ‘good’ locations they are displayed in yellow, and if they have delivered oil to a low number of ‘good’ locations they are displayed in red. Those client locations within the City Market (which are all paid at the same rate) are also color coded to complement the driver’s color. Locations that are inside the City Market and close to the distribution center are displayed in green, (meaning that they are a ‘good’ location to deliver to) locations at an average distance from the distribution center are displayed in yellow and a location far from the distribution center is displayed in red. The locations that are outside the City Market are displayed in grey because they are priced based on the distance from the distribution center. The intention is to assign a driver that has delivered to a higher number of ‘bad’ locations to a green, or good location, and vice versa.

The rest of this section discusses an assignment scenario using the matrix design. In this scenario, the dispatcher wants to assign a driver to the Gramado location. To begin this procedure, the dispatcher selects the Gramado location column along the top of the matrix design which displays Location Information at the bottom of the screen and highlights the possible drivers that can deliver to this location, as shown in Fig. 2.

The highlighted drivers within the Gramado column adjust to show the amount of time that they have been waiting at Tenoas. In this scenario driver LXX-9065 has been waiting 45 minutes and driver LZS-3545 has been waiting 25 minutes. Since both drivers have the same capacity truck, meaning they are equally able to deliver to the Gramado location, the driver that has been waiting the longest is selected. Once driver LXX-9065 has been selected, a dialogue box appears as shown in Fig. 3.

The dialogue box titled “Possible Route Combinations for Gramado” appears and provides the dispatcher with multiple delivery options. Since the selected driver’s truck is 30,000L and Gramado is only a 15,000L request, the driver can deliver to another location as well. The dialogue box in Fig. 3 shows the possible locations that the driver can deliver to while delivering to Gramado. In this scenario, the location Gramado/Feliz/Parobe combination is selected by clicking on the “Assign Route” button next to the combination in the dialogue box (presumably chosen because a) it fills the truck, b) delivering to multiple locations in one trip is more profitable and c) the return time is earlier than just delivering to two locations).

<table>
<thead>
<tr>
<th>Drivers</th>
<th>Capacity</th>
<th>Location</th>
<th>Delivery Time</th>
<th>D/S</th>
<th>Route</th>
<th>Capacity</th>
<th>Location</th>
<th>Delivery Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>LXX-9065</td>
<td>30</td>
<td>Feliz</td>
<td>15</td>
<td>D/S</td>
<td>30</td>
<td>Parobe</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2: Location selection using matrix.
IV. DESIGN B – MAP REPRESENTATION

Fig. 4 is a screenshot showing the map design during the day when drivers are waiting or currently en route. There are 24 red circles, each of which represents a request. Each circle is positioned on the map at the city of the client that made the request. The legend on the left side indicates the amount of oil (liters) represented by the size and saturation of each circle. The 20 black circles represent requests that are currently being service (the driver is on route). The box in the lower left hand corner displays the month-to-date ratio of the total revenue of Dalçoquoi to Stefani using a configurable moving average calculation and includes a vertical bar representing the goal of the 60/40 contract. Positioned at the bottom middle is a box that the dispatcher uses to add a request to the map and a search function that is used to search for a city, client, or driver. The search returns the entry highlighted on the interface. Also located at the bottom of the screen is a box which filters what is displayed on the map; multiple items can be marked at the same time (all requests are currently being displayed).

On the right hand side of the screen is the Driver List which lists all of the drivers by their license number, size of truck, and company. The color of each driver indicates his company affiliation where green represents Dalçoquoi and orange represents Stefani (those are the company colors). There is a Sort function on top which sorts the drivers based on drivers’ earnings-to-date (running total revenue of the last two weeks), wait time, estimated time of return (ETR), truck size, and company. In Fig. 1, the drivers are currently sorted by earnings-to-date and the numbers correspond to the amount that each driver has earned below or above the average. For example, driver BWT-4973 has earned R$198 more than the average over the past two weeks and should receive less routes and/or less-valued routes. Furthermore, drivers that are currently delivering gas are “grayed out” because they are unavailable and an estimated time of return is shown. Drivers that are currently waiting and available...
are shown with a horizontal blue bar where the length of this bar is proportional to the amount of time the driver has been waiting. Also, the light gray oval represents City Market (an area around Tenoas where all locations are delivered at the same rate), the light gray rectangle on top shows the five cities that are located in other states, and the black “x” is the location of the distribution center.

The remainder of this section describes the process of assigning a driver to a route given the same circumstances as described in the previous scenario. Assuming the dispatcher wants to satisfy the 15,000L demand in Gramado, the request on the map is selected and outlined in white, while nearby requests are displayed on the map, as shown in Fig. 5. These nearby requests are in cities that are located close enough in which the combinations of delivering gas to Gramado and the other cities is more profitable. At this point, a Request Profile appears containing information unique to that request (client name, city, amount, type, preferred driver, and time constraint) and gives delivery options ranked by their value. Fig. 5 shows the Gramado/Feliz/Parobe option highlighted which is represented by the black lines on the map. Also, all drivers that are unavailable to drive (e.g. currently delivering, sick, etc.) are “grayed out” and all drivers that are available to drive are shown.

Selecting one of the delivery options in the Request Profile, shown by the yellow highlighted bar in Fig. 5, filters all of the available drivers by “graying out” all of the drivers that fail to meet the capacity constraint of the request. In this figure, only the available drivers with 30,000L trucks are shown since the requests for Gramado, Feliz, and Parobe require a 30,000L truck. If the Gramado option was selected, all available 15,000L trucks would be shown and the 30,000L trucks would become unavailable. Also, the black lines on the map signify the route of the combination that is selected in the Request Profile. At this point, the dispatcher can click on any of the available drivers in the Driver List in order to access their Driver Profile. The Driver Profile, located to the right of the Request Profile in Fig. 5, displays all of the driver’s important information: license number, company, capacity and compartments of truck, current status (e.g. en route, waiting, sick) and graphs about a driver’s earnings compared to the average total of all drivers. From this profile, the dispatcher can view the driver’s log of deliveries by clicking on the “History” option. Also, the dispatcher can choose to view all routes that the driver could be assigned by clicking on the “All Routes” option. In order to assign a driver to a route, the dispatcher clicks on the “Assign” button located between the Driver Profile and Request Profile (Fig. 5).

When the assignment is made, a blue confirmation box with a trip itinerary appears with all pertinent information about the delivery, as shown in Fig. 6. This trip itinerary includes the RouteID, license number, company, capacity of truck, value of route, and the estimated time of arrival at each location. It also details the filling procedure, the Request Profile for each city, and the estimated departure and arrival times from Tenoas. In order to confirm the assignment, the dispatcher clicks the “Dispatch” button and the itinerary is printed and given to the driver to begin the filling process. The dispatcher can also click the “Reverse Direction” button in order to change the delivery sequence (i.e. Parobe, Feliz, Gramado) if there is a time constraint. When the “Dispatch” button is clicked, the screen returns to the current state and the requests from the previous assignment change from red to black on the map and remain black until the driver has returned and the deliveries are complete.

V. MATRIX VS. MAP COMPARISON

Although the two interfaces are both decision support systems, they emphasize different aspects of the system while presenting “information” instead of just raw “data”, which allows the dispatchers to have a complete understanding of the state of the system [3]. The approach to the resource allocation problem is the same in the sense that they provide possible driver/location combinations when either a driver or a location is selected. When these combinations are displayed, comparisons can be made to determine the benefits and drawbacks of a particular assignment. Both interfaces aid the dispatcher in their decision making process by providing reporting functions and information boxes to determine why a particular combination is more advantageous than another, depending...
on the particular needs at the time.

A major strength of the matrix design is the ability for the user to see all possible driver/location combinations at once. All of the routes to where a driver can deliver are displayed along with all of the drivers that can deliver to a particular location. A strength of the map design is the ability to show the spatial relationships between requests. This is a valuable feature because the user can visualize where the driver will be going and the relative distances between the locations. This method likely creates a greater sense of trust between the user and the system because the user can view the route that has been assigned on a map, unlike matrix design which displays text boxes which can create a sense of uncertainty of where the truck is physically going. While the matrix interface displays more information at once, the map interface gives the represents the information in a more tangible way.

VI. Conclusion

Each interface was designed by a different design team and could be compared for further improvements. Similar aspects of each interface, such as the driver or route information and dialogue boxes, can be compared to determine the most effective attributes. Once the most effective attributes of each interface are determined, the less adequate design features can be changed to mimic the more effective attributes. These changes would also make the two interfaces more consistent so that the user would feel as if they are using one system instead of two interfaces should they be compared in a side by side experiment.

These interfaces effectively complement each other because they each aid the user in the three phase iterative process: 1) perceiving relevant information, 2) generating “situational assessments” and hypotheses about the information and 3) selecting choices to take [4]. In general, the matrix interface design concentrates on the even distribution of ‘good’ and ‘bad’ roads and the map interface design concentrates on balancing the drivers’ salaries and the Dalçoquoi/Stefani ratio. Displaying both screens would be advantageous because the user can choose which method of decision making is appropriate at the time. Since the matrix design displays all of the driver/location combinations, the user could choose the driver/location assignment on the matrix design and the map design would display the route that the driver would take, providing a sense of confidence that the dispatcher selected an optimal route assignment. The bottom line is that both designs were “designed to improve decision making of its user by extending the user’s cognitive decision-making abilities” [5]. The combination of both interfaces would provide different views of the same system and allows the dispatcher to make decisions based on the factors that are most important in a particular situation.

The approach of having two independent design teams work to meet the same functional requirements led to the realization that the dispatching task has some fundamental characteristics. Users are likely to start a dispatching decision by either selecting a destination (demand) or driver (resource) to begin the assignment process. Comparisons are then made to pick a resource that can meet that demand or vice versa. Details might need to be accessed about each of these data elements, and thus there is a need for details on demand. History data is required on demand, and support is needed for navigating through the screens, tracking decisions made, and changing and justifying those decisions as circumstances dictate. The two design teams met most of these demands, but different teams focused on different details that will ultimately be important in the final design. For example, the team working on the map design determined the need to have a detailed itinerary report once a particular assignment was made (Fig. 6) which would be a useful feature to add to the Matrix design. Meanwhile, aspects about the design approach taken led to functions that are more amenable in one approach than the other. For example, the Matrix design team ended up with more functions for sorting and filtering locations, since a dedicated list of these (acting as columns in the matrix) make it amenable to do so. The Map display has functions for filtering cities by request amount (which is represented by the size and saturation of the circles representing each city) but has no functions for sorting or searching for cities based on time of delivery. This would likely be a useful feature to add into the Map system in some way. Thus, the independent design team approach is a recommended one for future user interface design teams since it helps to uncover features that one team might have found independent of the other, while verifying functions that both teams identified as necessary.

Acknowledgment

We would like to thank the manager, dispatchers, and truck drivers at Tenoas for allowing us to conduct research and interviews on a weekly basis. We would also like to thank Lucimara Ballardin and Cleyton Vargas, the industrial engineering graduate students from the Universidade Federal do Rio Grande do Sul that helped us with our research.

References