Visualization in Resource Allocation Tasks

Pearl Pu & George Melissargos
Laboratory of ergonomics of Intelligent Systems
École Polytechnique Fédérale de Lausanne
DMT/IMT, MT-Ecublens
1015 Lausanne, Switzerland

May 4, 1998

Introduction

Allocation of resources, whether man power or tools, are ubiquitous tasks in factories, hospitals, airline and communication network companies. It is also an example of many decision-making problems involving complex and changing criteria. It presents unusual challenges to information systems both from modeling and problem solving points of view. This paper presents an application of information visualization techniques in the resource re-allocation domain and in particular flight rescheduling. In collaboration with Swissair, our work concentrates on human-computer problem solving and how visualization techniques can help users perceive the entire solution space in four abstraction models in order to make the "right" decision. We present a technique called *coordinated visualization*.

In the following text, we describe the domain, followed by a task analysis of the flight rescheduling problem. We then focus on the method of coordinated visualization and discuss how that achieves the goal of helping users perform tradeoffs in decision making.

Flight reallocation

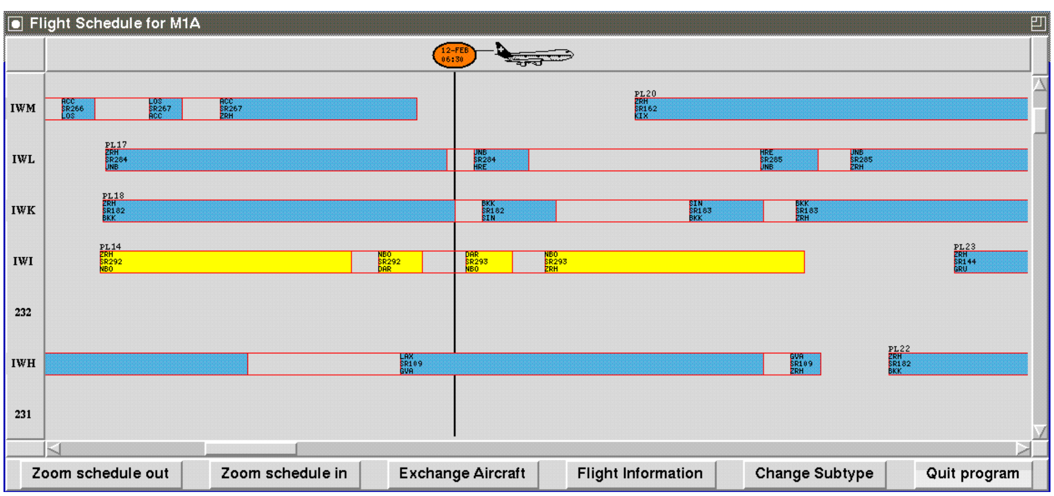

Figure 1: Pairlists as connected bars and their preassigned airplanes (left column

A flight route consists of a list of pairs (hence pairlist) where the first element of each pair is the departure point of a flight and the second the destination. Each pairlist is assigned to an airplane in the original schedule (Figure 1). A pairlist is usually longer than an individual flight. For instance, one route contains the flight from Zurich to Johannesburg (the long yellow bar in Figure 1), then a short hop from Johannesburg to Cape Town and back, and another flight back to Zurich. An airplane is normally assigned to an assortment of flights to allow varieties of weather conditions and take-off and landing characteristics. Flight scheduling, that is which airplane flies which route, is decided several weeks in advance. Three main reasons lead to the rescheduling of flights. Marketing personnel at various occasions offer promotion deals, thus want to upgrade an preassigned airplane to a bigger one. Technical personnel for reasons of routine maintenance request to change the route of an airplane so that it ends up in a destination where maintenance plant is kept. Finally and more urgently, an airplane has to be replaced by another one because of mechanical problems.

Reallocation requests are demanding because a small change in one airplane assignment causes other assignments to be changed as well, resulting in a chain reaction of exchanges. Further, these requests require machines to respond in real time because of their urgent nature and the fact that several people from different departments may be using the system at the same time.

Techniques from artificial intelligence (AI) were used to develop an automatic flight reassignment system which was previously employed at Swissair. It ran in real time and handled reallocation of airplanes in approximately two-week's time. User requests were handled rather flexibly. One could specify which airplane is to be exchanged, which airplane(s) are not to be moved in the process of rescheduling, and which ones can be moved. While there were many solutions found by the system, most users took the first one returned by the machine because of the highly textual nature of the solutions. Most optimization criteria and dynamic constraints were not handled by the system. For example, some solutions found later may involve a fewer number of airplanes to be exchanged. Some solutions may appear to be less desirable, but satisfy dynamic constraints such as flying a particular airplane to a maintenance destination.

Task analysis of flight rescheduling system

After careful analysis of user tasks, the following steps have been identified for possible user actions:

- task 1: browse existing assigned pairlists
- task 2: select a target pairlist to change
- task 3: ask the system to perform search to find potential swaps
- task 4: look for a right candidate
- task 5: play out the exchange before commitment

The existing Swissair flight rescheduling system consists of three main steps: 1) access data base and prepare pairlists for search; 2) search and 3) return results five at a time.

A simple comparison of user tasks and the existing system model reveals that because users are not involved in solution selection, the majority of solution space, thus the search effort, has not been optimally used. While building interfaces for supporting task 1, 2, 3, and 5 is straightforward, solving task 4 (that is deciding on a right solution) becomes not only a visualization task, but also the question of interactive human-computer problem solving.

Four visualization models of solution space

The main difficulty arises from the question "what is a right choice in a decision making system?" Classical AI techniques do not offer satisfactory answers since they are based on notions of crisply defined optimization criteria [2]. Fuzzy logic helps to a certain degree in modeling ranges of data using fuzzy sets. But the inherit problem in decision making is that often people do not have a clear definition of optimality until a set of solutions have been explored. This explains why humans can spend a lot of time in shopping around in order to decide what cars to buy.

In traditional scientific visualization systems, data is mapped to a single abstraction model whose geometry is displayed and analyzed. But complex information systems deal with data of a much higher dimensionality [1]. To present the solution space for exploration, we invented a technique based on 4 abstraction models, each covering an aspect of decision making. Any one of the models allows the selection of the final winner, but together they converge much more quickly towards the right solution.

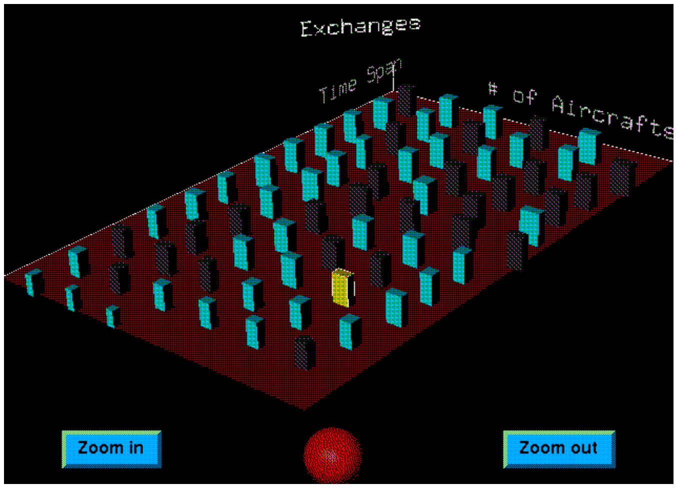

Figure 2: Three basic decision criteria modeled as highrises

The first abstraction model is implemented by a highrise visualization metaphor (Figure 2). Each highrise box (Figure 3) represents a solution in terms of three basic evaluation criteria: the total number of exchanges (EX), the number of aircraft involved (NA), and the span time (ST). EX denotes the number of pairlists to be swapped in the new assignment. NA is not necessarily equal to EX because airplanes often fly several routes. Thus while a solution may involve a high number of EX, it may only require few airplanes to be reassigned. Finally the span time indicates how long it takes to re-settle the original schedule. Only absolute dimensions are used, leaving the x,y,z coordinates in the 3D plane free for other information or criteria. Any of the three criteria can be used to sort the highrises. In Figure 2, the number of exchanges is the chosen one. By clicking on span time, the display will show a sorted set of highrises along that dimension.

As mentioned before, various criteria not accounted in the highrise metaphor can also play important roles in choosing the solution. These criteria contain neither structure, nor any information which allows direct mapping to geometrical objects such as highrises. Most of them cannot even be formulated until users explore the solution details.

Thus in addition to the highrise metaphor, we have implemented two more abstraction models in terms of exclusion and inclusion. That is, all solutions can be viewed either as good or no good, leaving users to judge what is good and what is not good.

In the exclusion model, a window of clickable buttons is provided (Figure 4). The left-most column and top row represent respectively the name of airplanes and flight numbers involved in the solutions. The numbered squares indicate how many times the pair, (airplane flight-number), participates in the solution space. A click on the numbered square excludes all solutions containing that pair, while a click on an airplane or flight number excludes the respective solutions. Several forms of criteria/constraints can be expressed in terms of exclusion. An airplane not fit for a particular flight route can be deleted from the solution space.

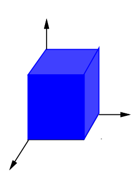

Figure 3: Dimensions of each box in highrise model

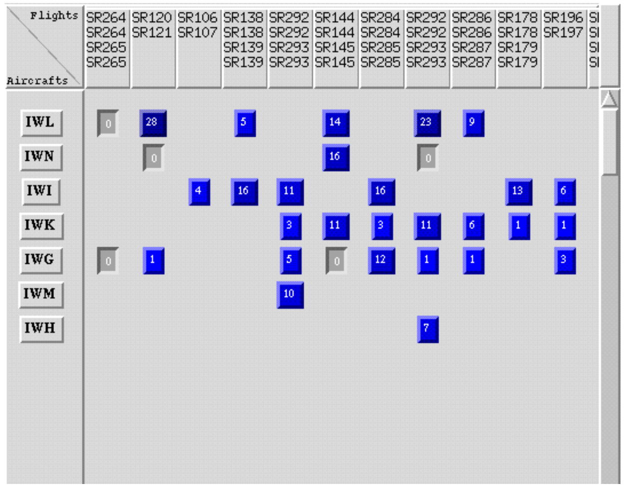

Figure 4: Exclusion visualization models

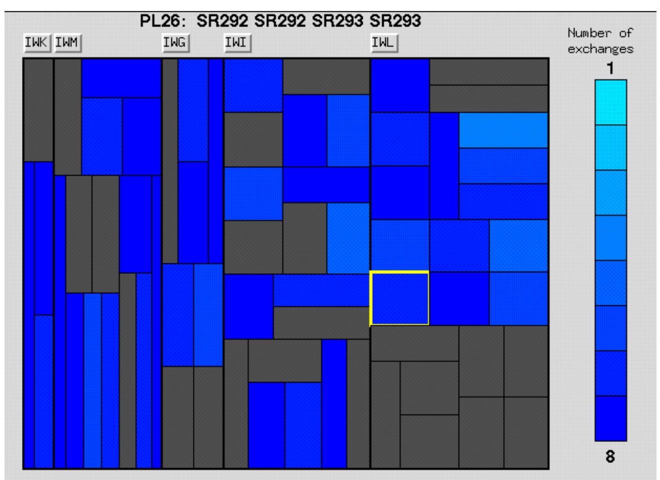

Figure 5: Inclusion visualization models

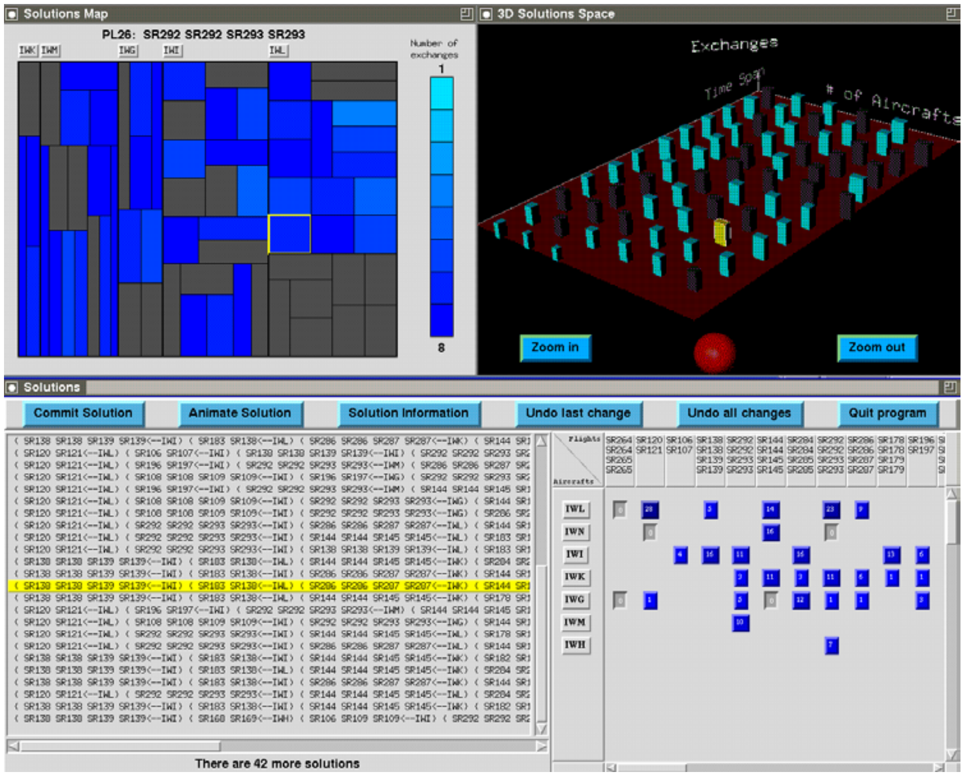

Figure 6: Coordinated visualization of 4 models for decision making

For instance, a jet without a special collision detection device cannot fly to a specific country. A flight route containing a specific airport not fit for rescheduling can be deleted. Finally, an airplane whose pilots are not able to change its original schedules can be deleted.

The inclusion model (Figure 5) is implemented by the treemap [4] visualization technique. However, instead of representing simple notions such as the size of files, our treemap is dynamic and corresponds to a decision tree found by the ID3 algorithm [3]. That is, the solution set is divided into minimally inhomogeneous subsets and essentially gives users an indication of the main subsets of the solution space. Hence they are able to choose an area of the treemap to further explore desirable solutions, thus the name inclusion model. The first-level subsets are characterized by the (airplane, pairlist) pairs. For instance, the squares under the name IWI are all reassignments including the airplane IWI and the pairlist PL24. If the title IWI is clicked on, users get down to the subtree to further explore the next-level subsets. Such a dynamic treemap allows users to step into a sub solution space to zoom in on details. At any level of the tree, users can select a single tile which represents a solution. The color coding reflects the number of exchanges involved in each solution.

The fourth model is a detailed textual list of solutions and is simply implemented by a listbox.

Coordinated interaction in visualization

As Gibson pointed out, humans' behavior and perception are two tightly coupled actions. People perceive in order to behave, and they behave in order to perceive better. In performing problem solving tasks with the machine, humans not only need to visualize their mental road maps, but also interact with their maps in order to find their destination. By coordinating all user actions in the four models, we allow them to explore different solution subspaces using different knowledge and criteria. Each manipulation enhances the user's experience with the solutions and eventually builds a conceptual structure that helps the reasoning process.

Once this goal is clear, the actual implementation of coordinated visualization is rather simple. In Figure 6, a click in the models where manipulation means selection, the corresponding tile, highrise or list all become yellow respectively. When the exclusion model is used, as soon as some of the squares have been clicked (meaning deletion), highrise boxes, tiles and lists become darkened respectively. Finally when only one solution is selected, the system can do a play out of that exchange using animation to show how the targeted flight moves to another spot.

Results

Many techniques from artificial intelligence and operation research have been proposed to either solve general resource allocation and reallocation problems or tackle specific aspects of them. However, two main obstacles still impede the full potential of these techniques from being widely used. First the search space associated with a RA problem can be enormous thus without some kind of constraints the method is mostly infeasible. Second the solution space can be very large, but selection criteria vary depending on the situation. In industry where human operators demand to be increasingly involved in the decision loop, a fully automatic system does not appeal to buyers. On the contrary, visualization techniques and more importantly a system design strategy to achieve interactive reasoning and decision making can help remove these obstacles and make AI systems more marketable. Our version of the system is being ported to the local machines of Swissair. Even though user studies were performed on a limited number of people, the prototype already helped Swissair sale their system to other airline companies.

Acknowledgment: We like to thank Klaus Bena, Elina Rantakallio, and Kun Sun from the Swissair expert system development team, and Boi Faltings and Stephan Monnier from the artificial intelligence laboratory at the Swiss Institute of Technology, Lausanne. This work has been supported by the Swiss Commission pour l'Encouragement de la Recherche Scientifique (CERS), with collaboration from Swissair.

References

[1] M. Chalmers. Design perspectives in visualising complex information. In S. Spaccapietra and R. Jain, editors, *Visual Database Systems 3: Visual Information Management, Proc 3rd IFIP Visual Databases Conference (VDB.3)*, pages 103–111. UBILAB, UBS bank, Switzerland, Chapman and Hall, 1995.

[2] G. Melissargos and P. Pu. Employing interactivity and visualization to augment the process of machine-based rescheduling. In *Proceedings of the First International Workshop on Approximate Reasoning in Scheduling (ARS'97) of the International ICSC Symposium on Fuzzy Logic and Applications (ISFL'97)*, 1997.

[3] J.R. Quilan. Learning efficient classification procedures and their application to chess end games. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 463–482. Mogan Kaufmann, 1983.

[4] Ben Shneiderman. Tree visualization with tree-maps: A 2-D space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, January 1992.