An Information Visualization Approach to Hospital Shifts Scheduling

Carmelo Ardito, Paolo Buono, Maria F. Costabile, Rosa Lanzilotti, and Adalberto L. Simeone

Dipartimento di Informatica, University of Bari
Via E. Orabona 4, Bari 70125, Italy
{ardito,buono,costabile,lanzilotti,simeone}@di.uniba.it

Abstract. Scheduling staff shift work in a hospital ward is a well-known problem in the operation research field but, as such, it is very often studied from the algorithmic point of view and seldom from the human-computer interaction perspective. In most cases, the automatic solutions that operations research may provide do not satisfy the involved people. After discussing the inconveniences of an automatic approach with physicians, we have designed a staff scheduling system that combines an expert system with an information visualization (IV) system; in this way the schedule generated by the expert system is presented through the IV system to the schedule manager, who can modify the results if last minute changes are necessary, by directly manipulating the visualized data and obtaining immediate feedback about the changes made.

Keywords: Information Visualization, Shift Scheduling**.**

1 Introduction

The problem of scheduling staff shift work in a hospital has been thoroughly examined in literature but it is still challenging due to the critical nature of people and aspects involved. In fact, unlike other organizations or businesses, the healthcare industry differentiates itself because, in most cases, its services can neither be offered at select times nor postponed. Its employees must maintain a coverage of the whole twenty four hours period and also during bank holidays. The employees' overall well-being is an important factor to consider when planning schedules, because it can have a deep impact on employees' performance, job satisfaction, and most importantly, the safety of those who they ought to care after [1].

In this paper, the focus is on the problem of scheduling physicians' shifts, instead of the broader field concerning nurses and other personnel. Manually finding a solution that can satisfy all parties involved in the process can be a complex operation since it is often difficult to satisfy each individual's needs. Common problems that are faced in this field include, for example, the need of minimizing overtime hours and making sure that there is always the minimum number of physicians required. There are several other constraints since physicians want to minimize the occurrences of working during nights, weekends and holidays such as Christmas and New Year's Eve. Therefore, staff scheduling must keep history of who worked, so that physicians

J.A. Jacko (Ed.): Human-Computer Interaction, Part IV, HCII 2009, LNCS 5613, pp. 439–447, 2009.

can alternate appropriately. For these reasons, it is useful to employ a computer system able to store the history of work shifts for the last few years. However, fully automatic solutions cannot effectively cope with last minute changes: if someone gets sick, the overall schedule needs to be redefined. A mixed approach, combining algorithms that automatically compute a schedule with Information Visualization (IV) techniques capable of involving users to drive decision making process appears valuable. Visualizations can thus become the mean through which users and computers can cooperate to obtain efficient results with a semi-automatic process that leaves final decisions to users.

This paper presents the development of a system composed by two modules: 1) an expert system, 2) an IV system. The first module automatically generates schedules, which are then visualized on the computer screen, so that users can interact and modify them. The user interface also allows users to specify some employees' constraints and preferences, which the expert system tries to satisfy.

According to User-Centered Design methodologies [2], the system has been developed by taking into account the needs of final users, which have been involved in our design. We have collaborated with physicians of the Paediatric Hospital “Giovanni XXIII” in Bari, Italy. After an initial requirement gathering phase, some paper prototypes were devised and evaluated with the medical staff at the hospital. This led to several refinements of the prototypes that contributed to enhancing the usability of the system. The final system is the result of all the feedback received during the prior stages of the design phase.

The paper has the following organization. Next section reports related work in this field. Section 3 describes how the system requirements have been gathered. Section 4 describes the developed scheduling system. Finally, Section 5 we report our conclusions.

2 Related Work

Different works exist in literature that have explored the issue of hospital personnel scheduling. As shown in Burke et al. [3], most analyses focus on comparing different algorithmic solutions with the results obtained from other approaches and less on the day-to-day reality of hospitals. Another literature survey is provided in [4]. Other works study different aspects of the schedule generation process such as satisfying work rules, taking into account personal preferences, and performance in general. These systems are often developed for testing special purpose algorithms and not with the intent of being actually deployed in hospitals.

In this work, the focus is on those studies who detailed experiences of complete systems being used in real world situations. Among these, Scheduler solves the generation problem algorithmically [5]: the application permits both total or partial schedule generation and manages unlimited work contracts. The schedules obtained with Scheduler can be visualized either as Gannt-diagrams, text or histograms, whereas our system offers different views, depending on the time granularity. Scheduler has been in use at several sites that could make use of its features: at Kolmårdens Djurpark, a Swedish zoo, users reported positively on its adoption, after an initial testing period. Very appreciated was the fulfilment of all work and law obligations, a requirement that is open to be much more error prone if worked on by humans, that may overlook or forget certain rules. In H&M Rowells' call center, a mail order subsidiary of Hennes & Mauritz (a Swedish based clothes manufacturer and retailer), before the introduction of Scheduler, staff was used to get the breaks they wanted (with people they knew), while the optimised breaks generated by the application have maximised staff availability for high demand periods.

Ghosh et al in their study [6] propose a computer aided system that supports managers in their work by computing optimal staff allocation based on a number of parameters. They report that the adoption of the system brings several benefits such as ensuring optimal nurse availability and simulate different staffing scenarios. The multitude of controls present in the interface disorients people not accustomed to the particular domain and would probably require some training before being able to use it with proficiency. From a human-computer interaction perspective, there is certainly room for improvement.

The application presented by Beliën et al [7] enables managers to allocate the resources needed for the master surgery unit, through a graphical user interface. Both the schedule and the available resources are shown in the same screen. The schedule area is further subdivided by days per week and according to the various operating rooms. Schedules have to be built manually, by assigning each surgeon to the desired time slot. The application helps the schedule manager by instantly visualizing the impact of each choice: in this way conflicts can be more easily identified and solutions found. This system has been tested at the Belgian university hospital Gasthuisberg and, according to the authors, it was deemed to be very promising for the purpose of facilitating and improving the schedule generation process. The interface itself, again, could require some training to be used to its fullest.

Among commercial systems, we examined Kappix's DRoster [8] and PlanningPME [9] have been examined: while they both do not allow automatic generation of schedules, they offer several features that allow users to specify work shifts through an interface similar to calendar applications. In DRoster, an automated rule engine spots violations or inapplicable schedules. An interesting feature is the possibility of specifying "places" and authorize staff from this point of view. DRoster allows schedules to be viewed both in a daily and in a hourly resolution. PlanningPME visualizes schedules in several different resolutions: morning or afternoon shift and daily, weekly or monthly.

3 The Requirement Gathering Process

With the aim of designing the scheduling system according to real users' needs, we involved some physicians at the Pediatric Hospital "Giovanni XXIII" in Bari. They were happy to collaborate with us because they are open to new technologies. We have successfully collaborated with them also in the development of a mobile system for supporting physicians treating epilepsy [10] and a system supporting remote collaboration among physicians [11].

A field study with contextual interviews has been performed during the requirements analysis to better understand the medical domain, how the physicians operate and the main features of the application to be implemented. User observation methods and principles of participatory design have proven to be effective in user interface design. The initial investigation was carried out by HCI researchers who performed contextual analysis at the hospital. They observed how physicians define the schedule and discussed the current process with them through interviews and focus groups. The familiar workplace helped the interviewed to recall certain work anecdotes that could have otherwise been neglected should the interview had taken place in another context. We discovered how the scheduling process was handled: despite technological advancement, schedules were still built manually via several paper forms, during a 30-day period. The different agreements that have to be made in order to guarantee the fulfilment of preferences and special requirements are the results of personal negotiations between the interested parties and the schedule manager. All the material was collected for later examination and for a preliminary description of the interaction possibilities that would have had to be managed. This material was used to formulate interaction scenarios that could depict the most frequent tasks users should be able to perform.

The requirement gathering phase highlighted that the system to develop is very complex from the functional viewpoint. In fact the resulting schedules have to satisfy several constraints, some of which related to regulations, others to the history of previous schedules and others to individual physicians' needs or preferences. For this reason, it was decided to rely on an expert system for the management of the schedule generation process, but leaving users the possibility of making changes, should special occasions arise that would make certain staff members unexpectedly unavailable; for instance, it could obviously happen that a physician cannot get to work for some days because s/he must attend a professional meeting or because s/he might be ill; it is thus necessary to insert this information and have the system recompute schedules. Therefore, it was necessary to design an intuitive interface that provided a) an overview representation of the overall situation and b) the possibility of interacting directly with visualized data.

4 The Scheduling System

The scheduling main components are: the expert system module and the visualization module. The former is responsible of analyzing the facts provided in input by the schedule manager (in this paper we assume to be a male physician) and computing a solution that tries to satisfy the constraints. The latter visualizes the computed schedules and allows the operator to modify them. These modules are described in detail in the following sections.

4.1 The Expert System Module

The expert system is developed in Java and JClips [12] and uses XML for data interoperability. It exploits one of the most commonly used techniques to develop expert systems, namely rule-based programming. With this paradigm, a set of rules (which represent heuristics) indicate how the system should behave in a variety of situations. Each rule is composed by an "if" and "then" portion. The former contains a series of patterns (or facts) that, if satisfied, will cause the rule to be applicable. The latter represents a set of actions that will be executed when the rule is applicable.

To begin the scheduling process, the system must know some data that are inserted by the user such as: start and end date of the period to be scheduled, physician's days off, etc. Other data that is needed is those which is computed internally by the system and is related to the amount of duty physicians and working and festive availabilities assigned to physicians in the Operative Unit. The insertion of facts in the knowledge base allows the expert system to activate the rules that will produce the final state, and hence the full schedules for the considered period of time. The expert system models these constraints that emerged during the requirement analysis phase, e.g.:

- A physician can work only one shift per day.
- A physician assigned the night shift must not be assigned further shifts on the next day and the day after that.
- A physician who worked on a particular bank holiday (e.g.: New Year's Eve) will not be expected to work on the same day of the following year.
- Some physicians may be exonerated from working night shifts

Among the different strategies that can be used to compute the schedules within CLIPS, our module uses the *Simplicity* and *Random* ones, that in the performed experiments produce better results in the distribution of physicians in the resulting schedule. Simplicity means that a rule is chosen depending on the number of facts necessary for it to be satisfied: less is better. Random, as the name implies, means that rules with the same salience value will be chosen randomly. The execution of a rule may change the list of applicable rules by adding or removing facts. Therefore, after a rule is executed, the environment selects the next applicable rule, until no more remain.

4.2 The Information Visualization Module

The information visualization module is the result of a series of refinements which contributed to its overall improvement. This was obtained through various iterations of paper prototypes. Developing paper prototypes before any implementation phase is fundamental in the software life-cycle. Through paper prototypes it is possible for final users to experience the interface before it is actually implemented. Should problems arise, they can be fixed and the interface modified in a quick and cheap way, according to the user's feedback and criticism. For our system, we built several paper prototypes before getting to the final interface. Each prototype was discussed with the physicians at the hospital in order to gain their feedback about our proposals. One of the main improvements was the introduction of a data entry interface (see Fig. 1.) that facilitated physicians in entering personal information and adjusting their preferences and requirements for the schedule generation. For example, in order to specify working preferences, the schedule manager simply has to click on the corresponding colored buttons. The feedback of this action is immediately observable visually in the interface.

From the interviews conducted after examining paper prototypes, we were able to gather some interesting insights: physicians appreciated the feature which let them have an overview of schedules (i.e., yearly) rather than shorter time periods (i.e., monthly or weekly).

4.3 A Walkthrough in the System

In this section we describe a walkthrough of the information visualization module in order to describe how the schedule manager inserts data for starting the schedule process (data-entry interface), and how he interacts with the scheduling interface.

Data-entry interface

As discussed earlier, the scheduling process starts by requiring some data from the schedule manager. The system shows the interface in Fig. 1. The user indicates which period he may want to generate schedules for. By using the two colored bars (referred to as "Preferenze Generali", Italian for "General Preferences") in the upper part of the interface, the operator may insert the preferences of each physician. General preferences concern which days of the week (and in which shifts) a physician would like to work or those to avoid working. The operator may assign different preference levels on each item by clicking the respective colored button. For example, for each day the operator can select the green button for expressing the best preference, orange for neutral, or violet to express that the physician dislikes to work in that day. Similarly, he can indicate the preferred shift among morning, evening and night. In Fig. 1 the physician, whose name is highlighted in the leftmost column, would like to work on Wednesday, Friday and Saturday (green colored), avoiding Tuesday, Thursday and Sunday (violet colored); about the shifts, he would like to work in the afternoon. The bottom of the screen (referred to as "Preferenze Dettagliate", Italian for "Detailed Preferences") allows the operator to assign to specific days preferences about the

Fig. 1. The data entry interface

desired shift or time off. For example, the operator can specify that a physician would like to have a week off in this month. After having inserted the desired information about general and/or detailed preferences for the physicians to be considered, the schedule generation process may begin.

Scheduling interface

The monthly scheduling, computed by the expert system module, is visualized in Fig. 2. It allows the schedule manager to compare each physician's schedule to those of colleagues of the same Operative Unit. The example shown in Fig. 2. refers to the October 2008 schedule. Each row refers to one of seven physicians belonging to the Operative Unit, while the columns represent days. By using colors and labels we are able to codify different activities the physician has to carry out during each day. For example, on October the 30th, the light blue icon with an "A" inside indicates that the first physician, whose name is Arcangelo Clemente, must work in the surgery unit of the hospital, while the fifth physician, Anna Mele, must work during the evening shift, as shown by the blue icon. In the monthly visualization, physicians are not interested in seeing the schedules belonging to physicians outside their unit in this interface; instead, they prefer to reserve this feature for the yearly overview, where the comparison between different Operative Units was felt to be important.

In the yearly visualization, as it can be seen in Fig. 3, there is a row for each month and they are aligned depending on the first day of the month. In the shown example, August (abbreviated AGO, for Agosto, in Italian) starts on Tuesday (abbreviated MA, for Martedì, in Italian) while September (SET) starts on Friday (GIO) and so on. Each row is further divided in as many columns as there are days in that particular month. Each day is then divided in three further rows which represent the three different shifts there are in a given day: morning 8 a.m. – 2 p.m., evening 2 p.m. – 8 p.m., night

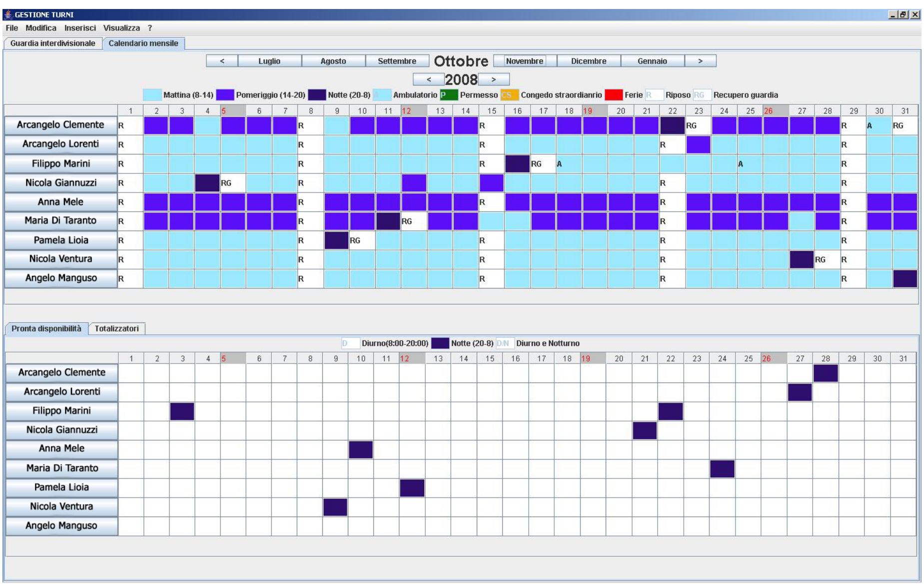

Fig. 2. Monthly overview

8 p.m. – 8 a.m.. Thanks to this visualization, holidays can be immediately spotted. In fact, physicians strongly stressed the need to highlight holidays and holiday eves. From our interviews it came to light the need of introducing the so called "totalizers", that is, counters which represent the sum of the total amount of shifts performed by each operative unit. These are displayed as an optional sliding panel on the right, which displays the totals both as a number and as a proportional bar. If the schedule manager is not satisfied with the computed output, or needs to adjust some last minute details of the schedule, he may customize the results. For example a problem has occurred with the first physician: he cannot work on Wednesday. The operator chooses him and selects "Modify" from the menu. The data entry interface will be displayed showing the current schedule for the chosen physician. The operator can now insert the new preference (i.e.: avoid working on Wednesday) and the overall schedule will be recomputed to satisfy the updated situation.

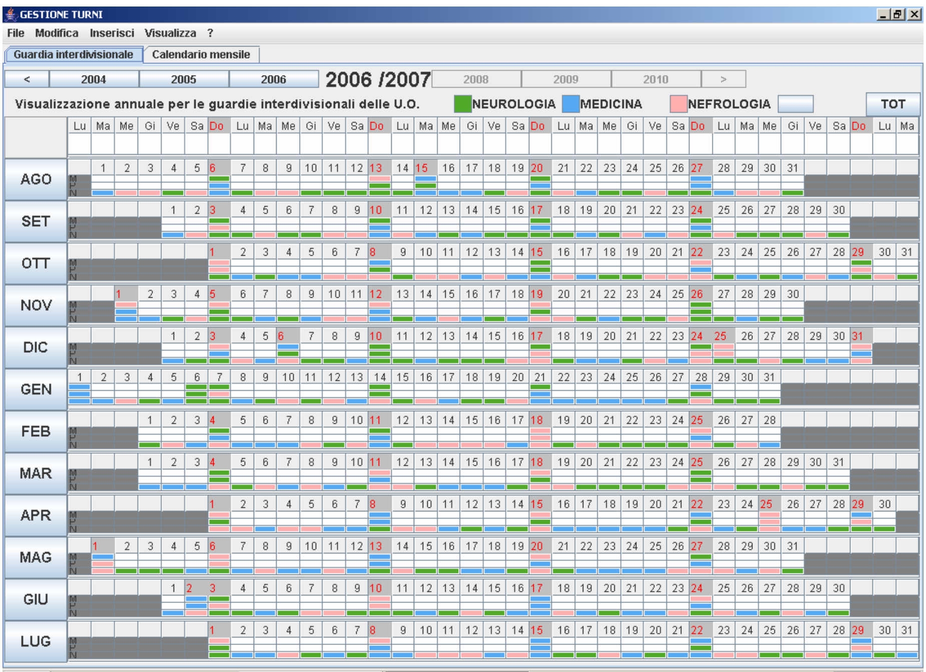

Fig. 3. Yearly schedules overview

5 Conclusions

In this paper we presented a system which supports staff shift scheduling. It is a challenging activity, because several aspects must be considered and it is even harder to satisfy all the constraints. Algorithms that automatically generate such schedules do not satisfy people in most cases, especially when last minute modifications are necessary, as it is often the case in hospitals. It is useful to employ a computer system able to generate schedules. The approach adopted in this paper combines an expert system that generates the schedules on the basis of rules specific for the hospital with Information Visualization techniques in order to give users the possibility to drive the decision making process and possibly perform the necessary adjustments.

Since the early stages of system design, we have involved physicians of a paediatric hospital. Thanks to the feedback received during the various steps of the development, including the revisions of paper prototypes, the system resulted being usable by the schedule managers.

Among the system features, the interactive overview was much appreciated because it represented an innovative idea, since users were accustomed to working with paper schedules. The rapidity with which users are able to deal with several unexpected situations gives the scheduling system an edge over old-fashioned manual approaches. These advantages come from the adopted visualization techniques, which enable users to quickly understand the status of a particular schedule, just by looking at the visualization.

Acknowledgments. This work was partially supported by University of Bari ("ex-60%" funding) and by EU and Regione Puglia under grant "DIPIS". We like to thank Ass. Prof. Stefano Ferilli for his valuable collaboration and the students of the Computer Science curriculum who participated to the development of the system.

References

1. Aiken, L.A., Clarke, S.P., Sloane, D.M., Sochalski, J.A., Silber, J.H.: Hospital nurse staffing and patient mortality, nurse burnout, and job dissatisfaction. J. of the American Medical Association 288(16), 1987–1993 (2002)
2. ISO 13407, Human-centered design process for interactive systems. International Organization for Standardization (1998)
3. Burke, E.K., De Causmaecker, P., Van Den Berghe, G., Van Landeghem, H.: The State of the Art of Nurse Rostering. J. of Scheduling 7(6), 441–499 (2004)
4. Ernst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: A review of applications, methods and models. European J. of Operational Research 153(1), 3–27 (2004)
5. Eveborn, P., Rönnqvist, M.: Scheduler – A System for Staff Planning. Annals of Operations Research 128(1), 21–45 (2004)
6. Ghosh, B., Cruz, G.: Nurse requirement planning: a computer-based model. J. of Nursing Management 13(4), 363–370 (2005)
7. Beliën, J., Demeulemeester, E., Cardoen, B.: Visualizing the Demand for Various Resources as a Function of the Master Surgery Schedule: A Case Study. J. of Medical Systems 30(5), 343–350 (2006)
8. DRoster, `http://www.kappix.com/`
9. Planning PME, `http://www.planningpme.com/`
10. Ardito, C., Buono, P., Costabile, M., Lanzilotti, R.: Two Different Interfaces to Visualize Patient Histories on a PDA. In: MobileHCI 2006, Espoo, Finland, pp. 37–40. ACM Press, New York (2006)
11. Costabile, M.F., Fogli, D., Lanzilotti, R., Mussio, P., Piccinno, A.: Supporting Work Practice through End-User Development Environments. J. of Organizational and End User Computing 18(4), 43–65 (2006)
12. JClips, `http://sourceforge.net/projects/jclips/`