An Implementation of a Scheduling Tool in a Medical Information System

Ivica M. Marković[1], Stevica S. Cvetković[2] and Dragan S. Janković[3]

***Abstract* – In order to achieve high quality in providing healthcare services with limited medical personnel resources an appropriate scheduling strategy for both staff and patients is required. Therefore a carefully designed IT solution for this purpose which is described in this paper can significantly improve scheduling process in a healthcare institution and make it more efficient.**

***Keywords* – Medical information system (MIS), Staff and patients scheduling, Appointment scheduling.**

I. Introduction

High demands in providing healthcare services together with limited medical personnel resources require a proper scheduling of staff and patients. It is known that efficient utilization of manpower has always been a key concern in any organization and one of the most important means to achieve productivity gain. For this reason, the staff and patient scheduling is an area that has become increasingly important in healthcare services. Also the working conditions of the staff in the health services have an important effect on quality of provided services and overall health of the patients [1], [2].

Scheduling process in a healthcare institution is an activity that consumes a large amount of administrative resources and, above all, time of the users themselves. Also appointments for certain diagnostic examinations are planned for 2-3 months in advance. Therefore primary contribution of this study is to focus on real world problems in medical scheduling services and to suggest a complete scheduling solution that can be configured and understood easily. Such a solution should have the following properties:

- Simple, clear and intuitive user interface suitable for use by medical personnel with a limited IT knowledge;
- System configurability based on the parameters stored in the database;
- Preventing conflicts during appointment scheduling and taking care of matching patients with their chosen doctor;
- Centralized scheduling system that provides insight into the status of all participants within the system;
- Generic reporting mechanism.

Due to complexity of the problem, entire model is broken down into two manageable tasks: (1) creating work time schedule for medical personnel and (2) scheduling of appointments for patients.

Our scheduling solution, Medis.NET Scheduler, is a part of a medical information system (MIS) which is being developed at The Faculty of Electronic Engineering in Niš (http://medisnet.elfak.ni.ac.rs) in cooperation with The Health Care Center in Niš [3]. Development of medical information system is a central part of a project supported and funded by The Ministry of Science and Technological Development of Republic of Serbia.

In the rest of the paper we first describe main use cases of the application and after that we present an overview of our system architecture.

II. Application's Main Use Cases

As already mentioned above we split scheduling problem into two parts and therefore two main groups of use cases can be analyzed – medical personnel work time scheduling and patient appointment scheduling as shown on Fig. 1.

A. Medical personnel work time scheduling

Work time scheduling tool in Medis.NET Scheduler is a tool which could be used for creating work time schedule for employees in any organization or company although our focus here is on its application within health care institutions. It is suitable for creating schedule for employees who do not have fixed work hours but instead change their work time by shifts. Moreover this tool supports changing of shifts by different complex patterns which we call work schemes (or work patterns). Main feature of work time scheduling tool is automated generating of work time schedule based on preconfigured data. An effort to enter configuration data should be made only once and after that schedule for employees is generated automatically at certain time periods. Later user actions should only include modifications of configuration data and entering exceptions from configured pattern such as absences from work.

Initial configuration of the tool should include the following actions:

- Connecting employees to the offices in which they work;
- Entering data about shifts (for example first shift from 7 AM to 2 PM, second shift from 1 PM to 8 PM);
- Entering data about breaks (break times are not fixed for a shift but depend on organizational unit also);
- Creating work pattern which defines the sequence of shift changing (for example first shift during one week and

[1]Ivica M. Marković is with the Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: ivica.markovic@elfak.ni.ac.rs

[2]Stevica S. Cvetković is with the Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: stevica.cvetkovic@elfak.ni.ac.rs

[3]Dragan S. Janković is with the Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: dragan.jankovic@elfak.ni.ac.rs

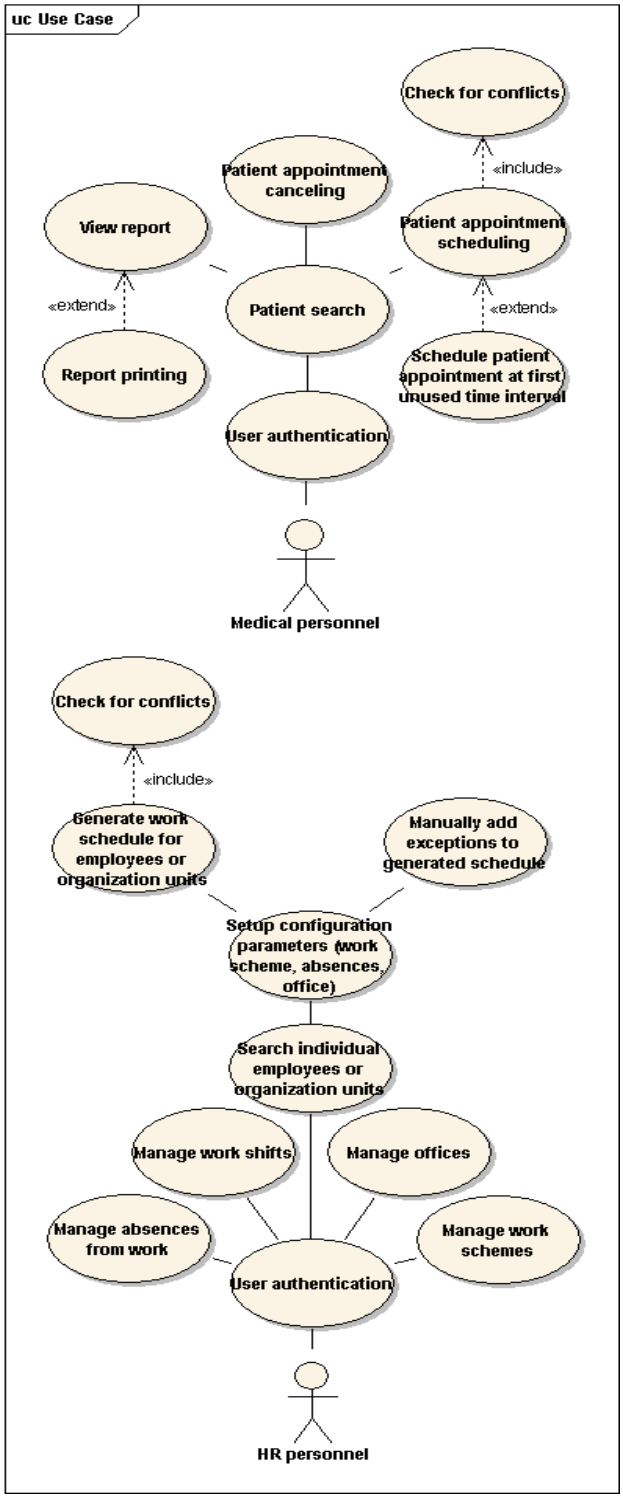

Fig. 1. Use case diagram

second shift during the second week, or maybe always work in first shift);

- Choosing an appropriate work pattern for each employee.

After initial configuration of the tool operator might only need to perform the following actions:

- Handle eventual changes in initial configuration (this should be very rare);
- Insert data about planned holidays of the employees;
- Change generated work schedule in case of employee's sudden absence from work.

This configuration tool, besides managing work time for employees, can also be used for managing configuration parameters required for patient appointment scheduling. One such parameter is average estimated examination time. This parameter is provided by Serbian Ministry of Health, and is defined for each medical specialization. For example, average estimated examination time of gynaecologist is 15 minutes, while for general practitioner is 10 minutes.

B. Patient appointment scheduling

After logging to the system, a medical personnel performs appointment scheduling in two successive stages. The initial stage is performed in the main GUI window that is used to find the patient in the database. At the second stage, doctor's appointment scheduling for the selected patient is carried out. This stage is conducted on a separate GUI window (AppointmentSchedulingWindow).

In order to enable efficient and visually intuitive scheduling tool, we developed custom TimeGrid visual control. It occupies central part of the AppointmentSchedulingWindow and offers to the user ability to assign the patient to doctor by simple mouse click (Fig. 2). The TimeGrid control displays all available appointment time intervals as grid cells, whereby already appointed time intervals are pink colored, while free intervals have white color. During software design phase, system configurability was considered as an important issue. As a consequence of this, all available time intervals are generated during application run-time, based on the doctor's working time stored in the database and maintained by HR personnel. Default layout displays TimeGrid for selected doctor in the next seven days, but it is possible to display time intervals for arbitrary selected period. A mouse click on a white cell results in doctor's appointment in the time interval specified by the cell. Appointment cancelling could be done by double click on already appointed time interval cell. As additional functionality, automatic appointment of the first not scheduled time interval is offered to the user. Generic report mechanism offers wide range of reports to be generated and printed, including a report which contains list of appointed patients for the next working day.

It is important to emphasize that our software is able to successfully prevent different scheduling conflict situations. There are two conflict scenarios, which must be prevented:

- Scheduling of a patient at one time interval for the two or more doctors;
- Scheduling of a doctor at one time interval for the two or more patients.

First step in the conflict prevention is implemented through GUI (TimeGrid). Actually, all time intervals which are already appointed (pink colored cells) are disabled for further scheduling. However, since the system is used in demanding multi-user environment, it is necessary to perform additional conflict checks immediately before database inserts. Therefore, our approach is to check for conflicts on temporarily locked tables and perform insert into database, all in a single transaction.

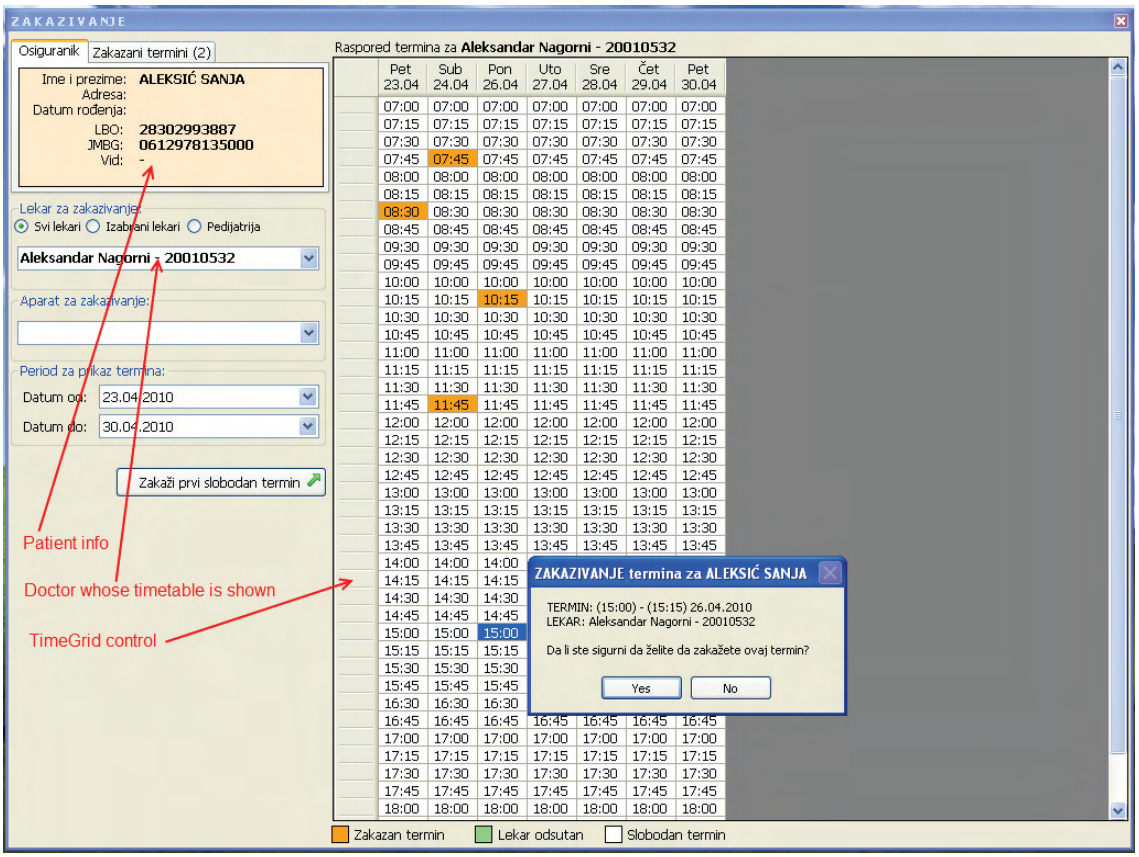

Fig. 2. Appointment scheduling window

III. System Architecture

Application Medis.NET Scheduler, as well as whole Medis.NET system, is a three-layer application which consists of the following three layers: data access layer, business logic layer and presentation layer. It is implemented using Microsoft .NET Framework 3.5 and integrated development environment Microsoft Visual Studio 2008 SP1. Application's database is operated on Microsoft SQL Server 2005.

For implementation of **data access layer** we used ADO.NET Entity Framework [4], a solution for object-relational mapping (ORM) within .NET Framework. ORM resolves a well known "impedance mismatch" problem between object oriented data layer classes and relational database. Microsoft Visual Studio 2008 SP1 contains auxiliary tools for automatic generating .NET classes which represent object model for target database. Mapping between relational database and automatically generated object model is defined through Entity Data Model (EDM) which uses XML syntax. In addition, Entity Framework introduces an extension (Language Integrated Queries - LINQ) to .NET programming languages which enables defining database queries as a part of standard language constructs.

Business logic layer consists of set of classes implemented in C# programming language. These classes contain application specific logic:

- User authentication and checking user rights level;
- Managing configuration parameters for generating work time schedule;
- Correcting generated schedule;
- Creating appointments for patients;
- Creating appointments reports etc.

Presentation layer defines end user's view of the application. It was carefully designed in order to achieve uniform look and simplicity of use because it is intended to be used by medical personnel who, in general, have minimal knowledge of information technology. Because of this we combined well known mechanism of Windows forms and new libraries from Windows Presentation Foundation (WPF). More details on using WPF in Medis.NET information system can be found in [5].

Database scheme was designed keeping in mind configurability of the solution as a very important request in Medis.NET Scheduler. This request was fulfilled by storing all configuration parameters in the database. Tables relevant for Medis.NET Scheduler application are shown in Fig. 3. Central table in the diagram is *employee* and it contains data about personnel (both medical and non-medical) in a healthcare institution. The rest of the tables can be divided into two groups – tables related to patient appointments (*appointment*, *specialization*, *employee_specialization*) and tables related to work time scheduling (*work_scheme*, *employee_work_scheme*, *work_scheme_shift*, *shift*, *employee_shift*, *break*, *absence_type*, *absence*). Some more explanations of these tables are listed here:

- *appointment* – contains data about appointed time periods and data about eventually canceled appointments;
- *specialization* – contains data about doctors' specializations and planned values of patient examination times for each specialization;
- *employee_specialization* – contains relations between tables *employee* and *specialization*;
- *work_scheme* – contains data about work schemes (work patterns);
- *employee_work_scheme* – connects employees with their work schemes;
- *work_scheme_shift* – connects work scheme with shifts which are included in the work scheme and contains frequency of changing the shift with other one;

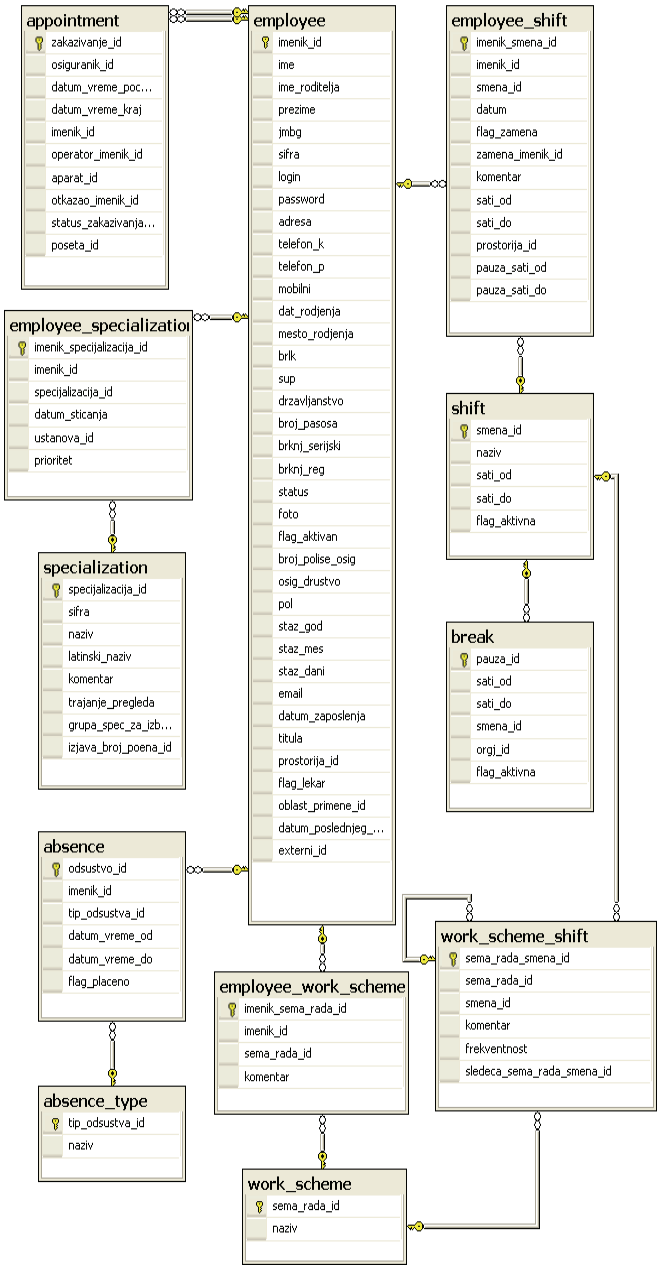

Fig. 3. Database diagram

- *shift* – contains data about work shifts such as start and end time;
- *employee_shift* – connects employees with their shifts, redundant table containing one record for employee per day which enables making exceptions from generated schedule;
- *break* – contains data about breaks during work time for each shift and organization unit;
- *absence_type* – contains data about different absence types such as holidays, sick leaves etc;
- *absence* – connects employees with corresponding absence types for certain time periods.

IV. Conclusion

In this paper we presented a solution for scheduling employees work time and patients appointments which is a part of a medical information system. By using latest technology achievements it aims to make efficient utilization of medical personnel resources and comprehensive scheduling of patients' appointments at The Health Care Center in Niš. Nevertheless there is still significant room for improvements in this area. Our future research will be targeted towards optimizing utilization of available healthcare equipment by involving various optimization techniques [6] in our software and improvement of the set of offered search criteria requirements for empty terms

Acknowledgement

Presented solutions for medical information systems are results of projects supported and funded by the Ministry of Science and Technological Development of the Republic of Serbia (Project Nr TR13015).

References

[1] D. Gupta, B. Denton, "Appointment scheduling in health care: Challenges and opportunities", IEEE Transactions, vol. 40, no. 9, pp. 800-819, 2008.

[2] S. N. Ogulata, M. Koyuncu, E. Karakas, "Personnel and Patient Scheduling in the High Demanded Hospital Services: A Case Study in the Physiotherapy Service", Journal of Medical Systems, vol. 32, no. 3, pp. 221-228, 2008.

[3] P. Rajković, D. Janković, V. Tošić, "A Software Solution for Ambulatory Healthcare Facilities in the Republic of Serbia", Healthcom 2009, Conference Proceedings, pp.161-168, Sydney, Australia, 2009.

[4] A. Adya, J. Blakeley, S. Melnik, S. Muralidhar, "Anatomy of the ADO.NET Entity Framework", ACM SIGMOD International Conference on Management of Data, Conference Proceedings, pp. 877-888, Beijing, China, 2007.

[5] I. Marković, S. Pešić, D. Janković, "Using XAML in Representation of Dental Charts in Electronic Health Record", ICT Innovations 2009, Conference Proceedings, pp. 247-256, Ohrid, Macedonia, 2009.

[6] A.T. Ernst, H. Jiang, M. Krishnamoorthy, D. Sier, "Staff scheduling and rostering: A review of applications, methods and models", European Journal of Operational Research, vol. 153, no. 1, pp. 3-27, 2004.