PREDICTION OF BLOOD DEMANDS IN A HOSPITAL

Christian Fischer [(a)], Lukas Bloder [(b)], Christoph Neumüller [(c)], Sebastian Pimminger [(d)], Michael Affenzeller [(e)], Stephan M. Winkler [(f)], Herbert Stekel [(g)], Rupert Frechinger [(h)]

[(a – f)] Upper Austria University of Applied Sciences
School for Informatics, Communications, and Media
Heuristic and Evolutionary Algorithms Laboratory
Softwarepark 11, 4232 Hagenberg, Austria

[(g – h)] General Hospital Linz
[(g)] Central Blood Laboratory / [(h)] Medical Controlling
Krankenhausstraße 9, 4021 Linz, Austria

[(a)] christian.fischer@students.fh-hagenberg.at, [(b)] lukas.bloder@students.fh-hagenberg.at,
[(c)] christoph.neumueller@students.fh-hagenberg.at, [(d)] sebastian.pimminger@students.fh-hagenberg.at,
[(e)] michael.affenzeller@fh-hagenberg.at, [(f)] stephan.winkler@fh-hagenberg.at,
[(g)] herbert.stekel@akh.linz.at, [(h)] rupert.frechinger@akh.linz.at

ABSTRACT
In this paper we describe the use of genetic programming for the prediction of blood demands. As blood bags for Hospitals are provided by blood banks on demand, predicting the needed amount of those should be as precise as possible. In order to achieve such an accurate prediction we have used genetic programming for data based modeling in order to find a mathematical model which predicts the blood bag demand of a hospital. This model should allow the hospital to minimize storage costs and the probability of running out of certain types of blood bags. In addition to the anonymized patient data provided by the General Hospital Linz, Austria we have also considered supplemental data such as weather and historical data such as the blood demand of the last few days which might lead to a more accurate model.

Keywords: blood demand prediction, machine learning, regression, structure identification

1. INTRODUCTION

1.1. Blood Demand in a Hospital

Every hospital needs a certain amount of blood bags for various medical activities throughout a day or a week. This blood consumption consists of demands from scheduled events (e.g. such as planned surgeries, ...) and from unpredicted events (e.g. some type of traffic accident followed by a treatment in the emergency room).

The blood bags are provided by a blood bank operated by the Austrian Red Cross on a "by demand" basis. If the hospital needs blood bags they are delivered by the blood donation service. In every hospital there is a need to predict the optimal demand of blood bags to minimize storage cost and the risk to run out of certain types of blood bags.

1.2. Research Goal

The research goal was to create a model which is able to predict the amount of blood bags of a specific type. There are different types of blood bag demands to predict, like demand per day, demand per week or demand per medical activity.

In this paper we present the research results achieved by analyzing the data of thousands of medical activities in the General Hospital Linz, Austria using data based modeling methods (namely genetic programming with offspring selection) in order to identify mathematical models for predicting blood bag demands.

In the following section (Section 2) we describe the data basis we have used for our research work. In Section 2.2 we describe the data preprocessing steps we performed to make the data more useful and complete for the model identification task. In Section 3 we give an overview over the modeling methods used in this project as well as the parameter settings applied, and in Section 4 we present and analyze the modeling results we have achieved. In the last section (Section 5) we give a conclusion of this project.

2. DATA BASIS

2.1. Available Patient Data

The data is provided by the Central Blood Laboratory of the General Hospital Linz, Austria and has been measured in the years 2005-2009. The following data tables are being used for the blood demand prediction:

- Laboratory Data: Contains every single blood value measured with a unique id for every pa-

tient, the label and the date of the value as well as the value itself. There are 27 routinely measured blood values of thousands of patients available, but not all values are measured at one examination.

- Medical Activities: Contains data about which medical activities were performed in which treatment. One treatment is identified by a case ID. A treatment can last several days or weeks and a number of blood measurements can be made during a treatment. A treatment typically ends by the discharge of the patient from the hospital.
- Blood Consumption: Contains records about the amount and type of the used blood bags in one treatment. However there is no direct connection between one medical activity and the used types of blood bags. Our approach to solve this problem is described in Section 2.2.1.

Patients' personal data (as for example name, date of birth and so on) where at no time available to the authors except the head of the laboratory and medical controlling.

2.2. Data Preprocessing

For most heuristic classification and regression tasks, the preprocessing of the raw input data has a big impact on the final model quality. The preprocessed data should end up containing both meaningful and complete information which is suitable for model identification using heuristic methods. Depending on the domain and the quality of the raw data, this can be quite a challenging task.

Figure 1 shows a brief overview of all the steps performed in the data preprocessing stage. Selected steps are described in one of the following sections. After all the input data is converted, the actual model identification can be performed. The desired prediction model should estimate the demand of blood bags for the General Hospital Linz for a given day or week as precisely as possible.

In the first preprocessing stage there are a number of input files, which contain multiple file types and have inconsistent column naming. For the modeling phase, the output of this stage should be a comma separated file, which has a consistent column naming and is usable for further processing and modeling tasks.

As a very practical problem, it appeared that the input data is scattered among multiple spreadsheet files with possibly different column names or even file formats. This problem is solved by a tool, which can merge an arbitrary number of heterogeneous spreadsheet files while taking into account that equivalent – but differently named – columns have to be merged. Those equivalences have to be defined manually.

Another problem is that in the laboratory data, every record represents one measured blood value. For blood demand prediction, one record has to represent an entire blood examination. For this reason, the records are transposed in a way that one record contains all blood values from a single examination.

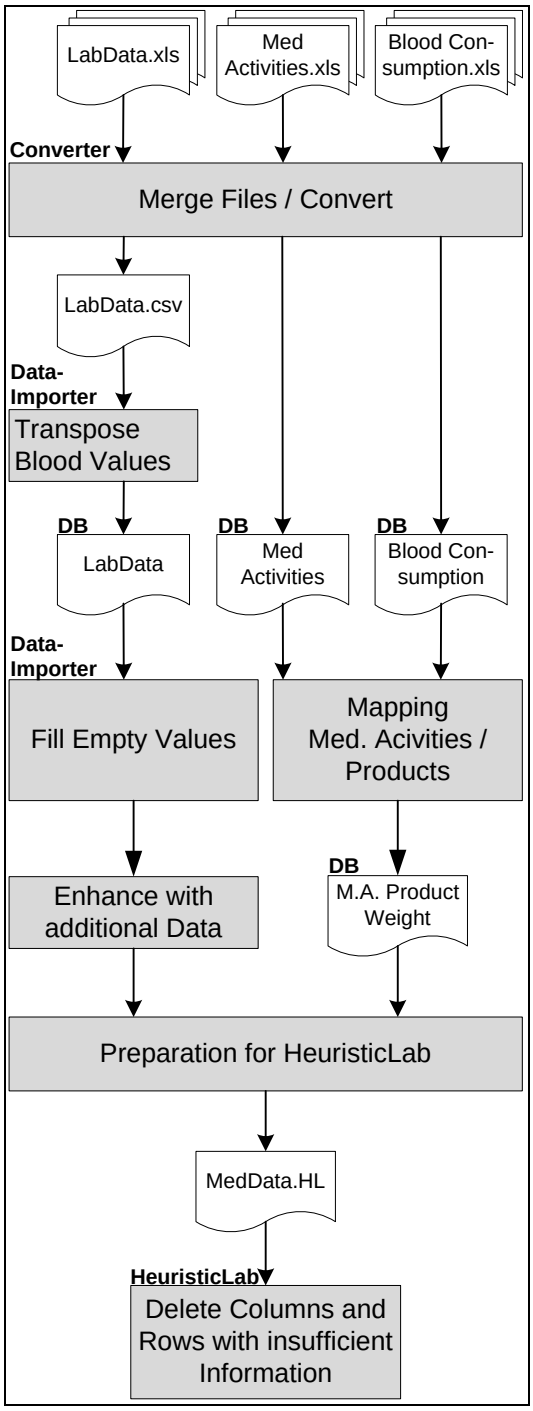

Figure 1: Workflow of data preprocessing phase

2.2.1. Mapping Medical Activities to Blood Products

For application of the blood demand prediction in the hospital a schedule of medical activities in the next days or week, for example derived from the planned surgeries, can be used as input data. The desired output is the amount of certain blood bags, grouped by different blood products.

In the data provided by the General Hospital Linz, Austria there is currently no unique mapping between

medical activities and issued blood products. But by using a schedule of medical activities as input for the blood demand prediction such a mapping is deemed necessary.

Therefore we distinguish the data in two sets: already unique mappings and non-unique mappings. For the first set we calculate a weight based on the amount of issued blood bags and the number of different patients who received these blood bags. For the second set we assign a constant weight of one and then combine the two sets again. With this new mapping we can bridge the gap between medical activities and different blood products to enable a blood demand prediction grouped by different blood products as desired.

2.3. Strategies for Improving Data Completion

In order to overcome the problem of empty feature values, samples from within a certain time span are used to fill up the missing values. This time span, e.g. one day, is used by a search function that determines which samples belong together and thus can provide features to each other.

Two strategies namely fill and merge, can be used to populate missing feature values of the given samples. The fill strategy fills up missing feature values for a given sample if a value exists within the list of samples chosen by the time span function. This strategy does not delete any samples but allows more complete samples to be increased in weight as their values are used more often. In contrast to that the merge strategy merges the selected samples to one. This results in fewer samples but does not favor feature-rich samples. The two strategies can be combined with one of the following aggregation functions which define how to compute the new value if more than one values are found by the search function:

- Min: fills up the missing values with the smallest value found in search space
- Max: fills up the missing values with the greatest value found in search space
- Mean: fills up the missing values with the mean value found in search space
- First (Merge only): takes the value from the sample with the oldest timestamp found in search space
- Last (Merge only): takes the value from the sample with the youngest timestamp found in search space
- Nearest (Fill only): takes the value with the minimum time distance to the empty value

Table 1: Input Data

Pat ID	Date	Val.1	Val.2	Val.3	Val.4
1	20/12/08		35		25
1	21/12/08	8.6		19	
2	21/12/08	10		30	8
2	22/12/08	5	15		13

Table 2: Merge-Max

Pat ID	Date	Val.1	Val.2	Val.3	Val.4
1	20/12/08	8.6	35	19	25
2	21/12/08	10	15	30	13

Table 3: Fill-Max

Pat ID	Date	Val.1	Val.2	Val.3	Val.4
1	20/12/08	8.6	35	19	25
1	21/12/08	8.6	35	19	25
2	21/12/08	10	15	30	8
2	22/12/08	5	15	30	13

The algorithm can be parameterized with a threshold which defines the size of the search space in days and a number of grouping columns, which also constrain the search space. In the case of Laboratory Data, the Patient ID would be such a grouping column, since only samples from the same patient shall be merged or filled.

3. MODELING METHODS

3.1. Artificial Neural Networks

Besides the use of genetic programming (GP) for system identification also artificial neural networks (ANN) can be utilized. For a regression or classification task a feed-forward neural network with one output neuron and backpropagation can be used; theoretical background and details can for example be found in (Gurney 1997, Priddy 2005).

But in contrast to GP, where the actual size and height of the tree containing the operators and feature variables can grow and shrink during the run, the number of neurons and their connections however has to be fixed before each training of an ANN. In addition an activation function for each neuron or for all neurons in each layer has to be chosen for ANN, whereas the operators in the tree for GP are chosen randomly during initialization.

We limited our work to GP in finding a model for the prediction of blood demands. The various modeling approaches are discussed in the following section.

3.2. Genetic Programming

Our main approach towards the blood demand prediction is genetic programming (GP). This section gives a theoretical background on genetic programming and shows how it can be applied to solve problems.

3.2.1. Introduction to Genetic Programming

Genetic programming is inspired by the Darwinian principles of selection, crossover and mutation. It can be seen as a specialized form of genetic algorithms (GA) to generate computer programs and therefore to solve problems automatically.

Historically the field of genetic programming began with the evolutionary algorithms. In the 1990s, John R. Koza pioneered the application of genetic programming. Over the years the idea was expanded and gained foothold both in the academic and industrial

field. As described in (Koza 1992) virtually all problems in artificial intelligence, machine learning, adaptive systems, and genetic programming provides a way to successfully conduct the search in space of computer programs.

As mentioned before, GP is a specialization of genetic algorithms. Each individual is a computer program that receives input, performs computations and generates output. In (Buchberger et al. 2009) GP is described as a machine learning technique used to optimize a population of computer programs according to a fitness landscape determined by a program's ability to perform the given task. The concept of GP is domain-independent, so it is important to find a good problem representation schema that can be effectively manipulated by the two main operators, namely crossover and mutation. This is critical to the success of genetic programming. The most common representation type is the point-labeled structure tree as seen for example in (Koza 1992; Koza 1994; Koza et al. 1999; Koza et al. 2003; Langdon and Poli 2002).

The crossover operator is applied on individuals to exchange a node of the structure tree with another node in another population. Due to the tree representation this can mean that a whole branch is replaced. As an effect the resulting new program structure can differ strongly from its parents.

The mutation operator is applied on a randomly chosen node. It can either alter the information of a node, or replace it completely, depending on the problem and tree representation.

3.2.2. Data Based Modeling and Structure Identification

In structure identification, solution candidates represent mathematical models; these models are applied to the given training data and the so generated output values are compared to the original target data. The left part of Figure 2 visualizes how the GP cycle works: As in every evolutionary process, new individuals (in GP's case, new programs) are created and tested, and the fitter ones in the population succeed in creating children of their own; unfit ones die and are removed from the population (Langdon and Poli 2002).

Within the last years the Josef Ressel Centre for Heuristic Optimization has set up an enhanced and problem domain independent GP based structure identification framework that has been successfully used in the context of various different kinds of identification problems for example in mechatronics, medical data analysis, and the analysis of steel production processes. One of the most important problem independent concepts used in this implementation of GP-based structure identification is offspring selection (Affenzeller et al. 2005), an enhanced selection model that has enabled genetic algorithms and genetic programming implementations to produce superior results for various kinds of optimization problems. As in the case of conventional GAs or GP, offspring are generated by parent selection, crossover, and mutation. In a second (strict offspring) selection step (as shown in the right part of Figure 2), only those children become members of the next generation's population that outperform their own parents; the algorithm repeats the process of creating new children until the number of successful offspring is sufficient to create the next generation's population (Winkler et al. 2009).

"Using Genetic Programming for data-based modeling has the advantage that we are able to design an identification process that automatically incorporates variables selection, structural identification and parameters optimization in one process" (Buchberger et al. 2009).

3.2.3. Modeling for Blood Demand Prediction

For our blood demand prediction we follow two approaches:

1. Blood Bag demand per medical activity (grouped by day and blood product)
2. Blood bag demand per day (grouped by blood product)

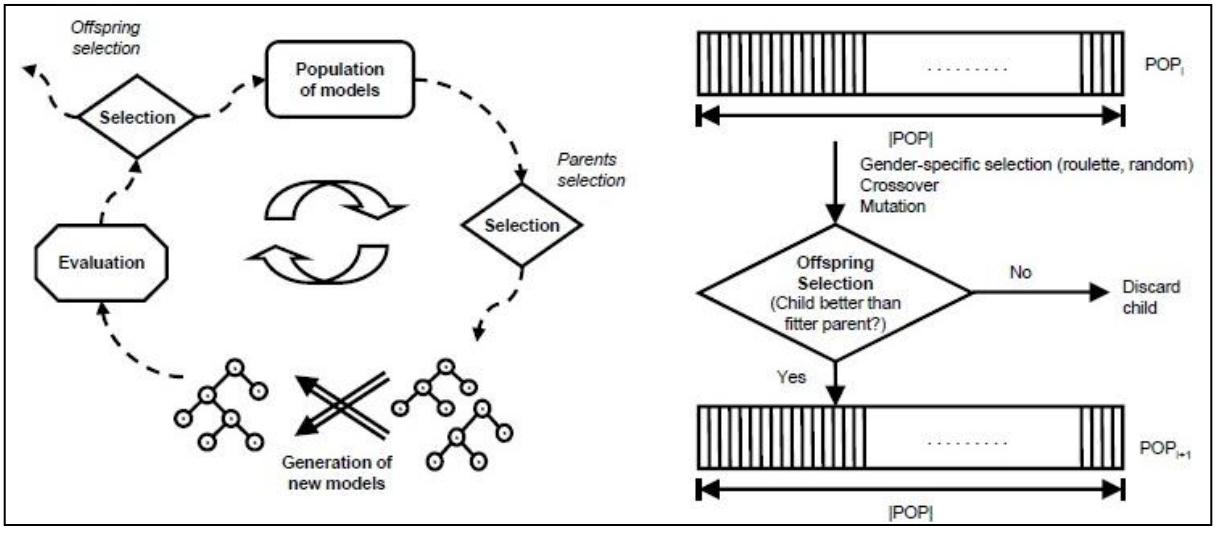

Figure 2: Left: The extended genetic programming cycle including offspring selection; Right: Strict offspring selection as used here within the GP process

Since the prediction depends on the blood product, both approaches are based on the mapping from medical activities to blood products described in Section 2.2.1. An additional feature with historical data is added, which sums up the blood consumption in the last one and last seven days. Now it should be possible to predict for example the blood demand for the next day.

In addition to splitting the given data into training and test data, the GP based training algorithm in HeuristicLab (Affenzeller et al. 2009, Wagner 2009) has been implemented in such a way that a part of the given training data is not used for training the model and serves as validation set; in the end, the algorithm returns those models that perform best on validation data. This approach has been chosen because it is assumed to help to cope with overfitting; it is also applied in other GP based machine learning algorithms as for example described in (Banzhaf and Lasarczyk 2004).

4. RESULTS

In this section we report the best test results we have achieved. Table 4 shows the most promising parameter settings for each scenario, which were found by running a number of test runs using different parameter settings. The parameters have been determined experimentally, since there is no golden rule for the optimal settings.

The tree complexity of the models has been restricted by defining an upper limit for tree height and tree size. This was done to keep the solutions interpretable and to avoid overfitting. Scenario 1 represents blood bags per day and Scenario 2 blood bags per medical activity.

Table 4: Parameter settings for test runs

Parameter	Scenario 1	Scenario 2
Population size	500	500
Mutation rate	0.05	0.05
Parents selection	Random & Proportional	Random & Proportional
Offspring selection	Strict	Strict
1-Elitism	Yes	Yes
Selection Pressure	200	300
Generations	1,000	1,000
Tree Size / Height	70 / 8	100 / 10

The results displayed in table 5 and 6 show the mean value and standard deviation of the model qualities achieved in nine test runs. In scenario 1 the 60,784 available samples were partitioned in 2,000 training, 43,589 validation and 15,195 test samples. In scenario 2 the 59,968 available samples were partitioned in 2,000 training, 42,977 validation and 14,991 test samples**.**

Table 5: Results Blood bags per day

	μ	σ
Training %	2.1929	0.0268
Validation %	1.5244	0.0027
Test %	1.8746	0.0151

Table 6: Results Blood bags per medical activity

	μ	σ
Training %	1.1897	0.0444
Validation %	1.2678	0.0388
Test %	1.1514	0.0311

5. CONCLUSION AND OUTLOOK

In this paper we have described the use of genetic programming to identify structure models that describe the blood bag demands in a hospital. The used data was provided by the General Hospital Linz, Austria. We have also described the necessary preprocessing steps in order to prepare the data to be useable for structure identification with genetic programming.

As seen in the results the introduction of features representing a medical activity leads to a significant improvement of the model quality.

Future goals in this research project include new modeling scenarios like the prediction of blood demand per week and for other time intervals. Additionally, further parameter optimization will be conducted.

ACKNOWLEDGMENTS

The work described in this paper was done within the Josef Ressel Centre for Heuristic Optimization *Heureka!* (http://heureka.heuristiclab.com/) sponsored by the Austrian Research Promotion Agency (FFG).

REFERENCES

Affenzeller, M., Wagner, S., Winkler, S., 2005. Goal-oriented preservation of essential genetic information by offspring selection. *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), volume 2*, pages 1595–1596. June 25-29, 2005, Washington DC, USA.

Affenzeller, M., Winkler, S., Wagner, S., Beham, A., 2009. *Genetic Algorithms and Genetic Programming - Modern Concepts and Practical Applications*. Chapman & Hall/CRC. ISBN 978-1584886297.

Banzhaf, W., Lasarczyk, C., 2004. Genetic programming of an algorithmic chemistry. In O'Reilly, U., Yu, T., Riolo, R., Worzel, B., eds. *Genetic Programming Theory and Practice II*. Ann Arbor, pages 175-190.

Buchberger, B., Affenzeller, M., Ferscha, A., Haller, M., Jebelean, T., Klement, E.P., Paule, P., Pomberger, G., Schreiner, W., Stubenrauch, R., Wagner, R., Weiß, G., Windsteiger, W., 2009. *Hagenberg Research. 1st edition*. Dordrecht, Heidelberg, London, New York: Springer, ISBN 978-3-642-02126-8.

Gurney, K., 1997. *An introduction to neural networks*. London: CRC Press.

Koza, J. R., 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press.

Koza, J. R., 1994. *Genetic Programming II: Automatic Discovery of Reusable Programs*. The MIT Press.

Koza, J. R., Bennett III, F.H., Andre, D., Keane, M.A., 1999. *Genetic Programming III: Darvinian Invention and Problem Solving*. Morgan Kaufmann Publishers.

Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., Lanza, G., 2003. *Genetic Programming IV: Routine Human-Competitive Machine Learning*. Kluwer Academic Publishers.

Langdon, W.B., Poli, R., 2002. *Foundations of Genetic Programming*. Berlin, Heidelberg, New York: Springer Verlag.

Priddy, K.L., Keller, P.E., 2005. *Artificial neural networks: an introduction*. Washington: The International Society for Optical Engineering.

Wagner, S., 2009. *Heuristic Optimization Software Systems - Modeling of Heuristic Optimization Algorithms in the HeuristicLab Software Environment*. PhD thesis, Johannes Kepler University Linz.

Winkler, S., Hirsch, M., Affenzeller, M., Re, L., Wagner, S. 2009. Virtual Sensors for Emissions of a Diesel Engine. *Produced by Evolutionary System Identification. In Computer Aided Systems theory (EUROCAST)*, February 15-20, 2009, Las Palmas de Gran Canaria, Spain.

AUTHORS BIOGRAPHIES

CHRISTIAN FISCHER received his BSc in software engineering in 2009 from the Upper Austria University of Applied Sciences, Campus Hagenberg. He is currently pursuing studies for his master's degree. In the course of his studies he is involved in the project team for the prediction of blood demands in a hospital in cooperation with the Josef Ressel Centre Heureka! and the General Hospital Linz.

LUKAS BLODER received his BSc in internet technologies in 2009 from the Joanneum University of Applied Sciences, Kapfenberg. He is currently pursuing studies for his master's degree in software engineering. In the course of his studies he is involved in the project team for the prediction of blood demands in a hospital in cooperation with the Josef Ressel Centre Heureka! and the General Hospital Linz.

CHRISTOPH NEUMÜLLER received his BSc in software engineering in 2010 from the Upper Austria University of Applied Sciences, Campus Hagenberg. He is currently pursuing studies for his master's degree. In the course of his studies he is involved in the project team for the prediction of blood demands in a hospital in cooperation with the Josef Ressel Centre Heureka! and the General Hospital Linz.

SEBASTIAN PIMMINGER received his BSc in software engineering in 2009 from the Upper Austria University of Applied Sciences, Campus Hagenberg. He is currently pursuing studies for his master's degree. In the course of his studies he is involved in the project team for the prediction of blood demands in a hospital in cooperation with the Josef Ressel Centre Heureka! and the General Hospital Linz.

MICHAEL AFFENZELLER has published several papers and journal articles dealing with theoretical aspects of evolutionary computation and genetic algorithms. In 2001 he received his PhD in engineering sciences from JKU Linz, Austria. Dr. Affenzeller is professor at the Upper Austria University of Applied Sciences, Campus Hagenberg, and head of the Josef Ressel Center *Heureka!* at Hagenberg.

STEPHAN M. WINKLER received his MSc in computer science in 2004 and his PhD in engineering sciences in 2008, both from Johannes Kepler University (JKU) Linz, Austria. His research interests include genetic programming, nonlinear model identification and machine learning. Since 2009, Dr. Winkler is professor at the Department for Medical and Bioinformatics at the Upper Austria University of Applied Sciences, Campus Hagenberg.

HERBERT STEKEL received his MD from the University of Vienna in 1985. Since 1997 Dr. Stekel is chief physician at the General Hospital Linz, Austria, where Dr. Stekel serves as head of the central laboratory.

RUPERT FRECHINGER received his MD from the University of Vienna. He obtained his apprenticeship as general practitioner at the hospitals of Upper Austria. Subsequently Dr. Frechinger was consultant at the Computer Science Cooperation Austria. Since 2000 he has been the head of medical controlling department at the General Hospital Linz, Austria.