International Institute for Applied Systems Analysis
IIASA www.iiasa.ac.at

Qualitative Decision Models for Structured Modeling Technology

Predki, B.

IIASA Interim Report
September 2004

International Institute for
Applied Systems Analysis
Schlossplatz 1
A-2361 Laxenburg, Austria

Tel: +43 2236 807 342
Fax: +43 2236 71313
E-mail: publications@iiasa.ac.at
Web: www.iiasa.ac.at

Interim Report **IR-04-050**

Qualitative Decision Models for Structured Modeling Technology

Bartłomiej Prędki (Bartlomiej.Predki@cs.put.poznan.pl)

Approved by

Marek Makowski (marek@iiasa.ac.at)
Senior Research Scholar, Risk, Modeling and Society Program

September 2004

Contents

Foreword

This paper presents the results the author achieved during his participation in the Young Scientists Summer Program (YSSP) 2004.

However, the impact of these results is wider than it can be seen from this paper. This is because of the synergy resulting from team work.

Three participants of the YSSP 2004: Bartłomiej Prędki, Cezary Chudzian, and Vladimir Molchanov were members of the team working on the development of the Structured Modeling Technology (SMT). The other two members of the team were Michał Majdan (who spent five months at IIASA on leave from the National Institute of Telecommunications, Warsaw, Poland) and me.

The development of SMT is a long-term challenging undertaking that requires collaborative work of researchers that have experience not only in methods and tools for advanced modeling but also knowledge and skills in DBMSs (Data Base Management Systems), XML (Extensible Markup Language), and object-oriented programming of Web-based applications.

Michał Majdan has designed the user interface to, and basic data structures of SMT. He had been coordinating the design of elements developed by other colleagues in order to be able to smoothly combine all elements into one system. This work has not been documented yet.

The contributions of the other three members of the team have been described in three Interim Reports (IRs), which constitute a kind of virtual set describing the collaborative work. I briefly summarize the scope of each IR encouraging the reader to become familiar with all of them:

- Bartłomiej Prędki (IR-04-050) has implemented an extension of SMT (originally designed for algebraic models) by implementing a prototype handling of decision rule models; he has adapted a suite of software supporting applications of decision rules for analysis of qualitative data to work with SMT. Moreover, he tested the concept using a medical case study developed in collaboration with the Ottawa University.

- Cezary Chudzian (IR-04-051) has developed the key elements of SMT that support a part of the modeling process composed of instance definition, specification of preferential structure for various types of model analysis, and efficient handling of underlying complex and large data structures (e.g., for parametric optimization, and diversified sets of results, both composed of huge amounts of data).
- Vladimir Molchanov (IR-04-052) has explored possibilities of using XML for automatic documentation of the modeling process, and implemented a prototype of automatic documentation of model specification, which is the most difficult element of the documentation due to the complexity of the structure of the symbolic specification and the requirement for supporting gradual modifications of the descriptive part of the documentation (which is added to the part resulting from the interactive model specification).

Finally, I would like to stress that it has been a pleasure to be the leader of the SMT team during the Summer of 2004. Each member of the team not only has very good professional skills but also abilities necessary for team work, strong dedication to achieve good results, and to have fun during the short periods spent away from the keyboard.

We plan to make the SMT publicly available in 2005. Therefore, I invite the readers to not only become familiar with the IRs mentioned above, but also to visit http://www.iiasa.ac.at/~marek in Spring 2005 to check on the further developments of SMT.

Marek Makowski

Abstract

This paper presents the decision rule extension to the Structured Modeling Technology based on a case study. SMT was developed at the International Institute for Applied Systems Analysis in Laxenburg, Austria. It is a tool for modeling complex algebraic models. Some of the models require the use of qualitative data, so it is necessary to represent such data. The proposed approach is based on the decision rule paradigm used in machine learning and knowledge discovery.

It was verified on a real-life case study based on the modeling of a care map. Care map is a multidisciplinary plan of best clinical practice for specified groups of patients with a particular diagnosis that aids the coordination and delivery of high quality care. The care map used in the work documented by this report was developed at the Ottawa Hospital, Canada, for radical prostatectomy procedure. The research was based on the data from about 125 patients, obtained from the same hospital.

SMT web application was modified to include models based on decision rules, which allow storing of data in a form applicable for rule generation algorithms, as well as storing decision rules as the results of model analysis.

Software application and class library was written for the rule generation phase, which is able to access the data from the SMT application and store decision rules in its data-warehouse. Decision rules are generated using external modules, based on the Rough Set theory.

Acknowledgments

The research described in this paper has been made during my participation in the Young Scientists Summer Program (YSSP) 2004 at the International Institute of Applied Systems Analysis.

I would like to thank Dr Wojtek Michałowski from the School of Management, Ottawa University, Canada, for his cooperation and Dr Marek Makowski for his supervision.

I would also like to thank Michał Majdan and Cezary Chudzian for their cooperation on SMT modifications.

Finally, I would like to thank the Polish National Member Organization of IIASA for the financial support which has made my participation in the YSSP possible.

About the Author

Bartłomiej Prędki received his M.Sc. in Computer Science from Poznań University of Technology in 1996. The title of his thesis was “Application of the rough set theory in the experiment concerning treatment of urological disease”. He is presently employed at the Laboratory of Intelligent Decision Support Systems and is currently a final year Ph.D. student at Poznań University of Technology. The title of his Ph.D. thesis is “Induction of preferential information for relational model from the set of decision rules in multicriteria choice problem”.
His main fields of scientific interest include decision support systems, machine learning and knowledge discovery with specific interest in rough set theory.

Qualitative Decision Models for Structured Modeling Technology

Bartłomiej Prędki*

1. Introduction

Structured Modeling Technology (SMT) is a general application modeling tool developed at the International Institute for Applied Systems Analysis in Laxenburg, Austria [Makowski 2004]. It is designed to support model-based analysis of complex decision problems that are not suitable for the general purpose tools available. It is built using the basic concept of Structured Modeling methodology introduced by Geoffrion [Geoffrion 1987] with the current computing technologies like data base management systems, object oriented programming and web applications.

One of the reasons for creating the SMT is to meet the requirements of modeling activities developed to support intergovernmental negotiations aimed at improving European air quality which constitute the RAINS model.

SMT is based on a paradigm of separating the model specification from its data. The Model specification contains semantic descriptions of model entities – variables or parameters and the relations between them. Based on the model specification it is possible to generate the data structures in the data-warehouse to store the model data. An approach based on the divide, conquer and integrate paradigm has the following advantages:

- possibility to amalgamate specifications for separate models,
- one repository used during the entire modeling process, that may include many approaches to analysis of different problem instances along with automatically generated documentation,
- possibility of data reuse for different model specifications and for defining the different sets of data for the same model specification, thanks to the database management system,
- possibility of recreating any given model analysis thanks to the data-warehouse approach.

SMT was designed for modeling of very complex quantitative problems, but decision support involves also qualitative data. This paper presents a qualitative decision model extension to Structured Modeling Technology based on the case study for modeling hospital caremaps in order to optimize the operations of the hospital.

* Institute of Computing Science, Poznań University of Technology, Poznań, Poland
Bartlomiej.Predki@cs.put.poznan.pl

2. Representing qualitative models

Every model is a representation of some knowledge about considered reality. In quantitative models this knowledge is usually represented in the form of equations. Because it's not possible to store information about the qualitative data as equations, also other forms of knowledge representation have to be supported.

One such form for knowledge representation for qualitative data is known as decision rules.

Decision rules are one of the tools used in machine learning and knowledge discovery domains. The basic assumption when using decision rules is that the data is described using attributes. Attributes can be of different types, e.g. numeric attribute or categorical attribute, where values are members of the set of labels (e.g. good, sufficient, bad). Such attributes are called condition attributes. We also assume that there is at least one attribute, the called decision attribute, that assigns each object to a decision class.

Decision rules are used to classify objects into decision classes, thus together with the classification procedure they constitute the classifier, where rules are used as the knowledge base. There is a possibility that classification obtained from the decision rules will be wrong, this is called a misclassification. Very good overview of decision rules generation approaches and classification techniques can be found in [Stefanowski 2001].

Decision rules represent knowledge in the form of "*if* … *then* …" clauses. The part between "*if*" and "*then*" is called a condition part and what follows after "*then*" is called a decision part.

For example, it may be possible to state the following decision rules:

if temperature=high *then* patient=ill (1)

if temperature=normal *and* cough=no *then* patient=healthy (2)

The condition part of the rule consists of so called "elementary conditions", for example in rule 1 there is one elementary condition "temperature=high", where "temperature" is the attribute name and "high" is one of its values. In rule 2 there are two elementary conditions joined by the logical "and" operator.

An elementary condition consists of three elements – the name of the attribute, the relation, and the attribute value. In general, relation can be one of the following: $=$, $\geq$, $\leq$, $>$, $<$ or a set member.

We assume that the condition part is always made up of the conjunction of the elementary conditions (only the logical "and" operator is used). Logical "or" (disjunction) is equivalent to using several decision rules with the same decision part.

In most cases the decision part of the rule consists of a single decision (outcome), which is similar to the elementary condition, except when the considered attribute is a decision attribute. If this is the case, such a decision rule is called an exact decision rule. There can also be more possible decisions connected by the logical "or" operator. Such

decision rules are called possible decision rules. All possible decisions in such a rule use the same decision attribute.

There are two ways of acquiring knowledge:

- supervised learning,
- unsupervised, automatic learning.

The first process is very complex and time consuming although it usually provides very good results. For example, one of the first expert systems based on decision rules for supporting the selection of antibiotics treatment – MYCIN [Buchanan, Shortliffe 1984] had 500 decision rules, all coming from expert physicians. It took almost 10 years to make the final version.

In the second approach a set of data is obtained, for example from historical data, called a training set. Based on this information several rule induction techniques can be used to generate a set of decision rules.

Because there is a possibility of inconsistency in the training set (for example patients with the same description by all condition attributes have different values of the decision attribute), resulting from the limited description of analyzed reality, one of the very successful approaches to rule generation is based on Rough Set theory.

The Rough Set theory was introduced by Pawlak [Pawlak 1982]. It allows for taking the inconsistency into the data analysis process. It is done by using so called rough approximations of decision classes.

For each decision class two rough approximations are calculated:

- lower approximation – contains only the objects that for sure belong to this decision class,
- upper approximation – contains all objects that may belong to the given decision class.

Having calculated the approximations, exact decision rules are induced from the lower approximations of each decision class, and possible rules are induced either from the upper approximations or from the boundary region between the classes.

There are three approaches to generating rules from rough approximations:

- local minimal covering algorithm, which generates the minimal (from heuristic point of view) set of decision rules, that satisfy the correct classification of data in a training set; the example of such an algorithm is LEM2 [Grzymała-Busse 1992],
- satisfactory rules algorithm, which generates the decision rules that meet some predefined conditions; the example of such an algorithm is Explore [Stefanowski 2001],
- all rules algorithm, which generates all possible (not redundant) decision rules; using Explore with specific parameters generates all decision rules.

Usually each decision rule has some additional information, based on the data it was induced from. Typically this includes:

- length of the rule – number of elementary conditions in the condition part of the rule,
- support – number of objects in the learning set covered by the condition part of the rule and compatible with its decision part,
- strength – quotient of the support to all objects in the training set,
- discrimination level – quotient of the support to the number of all objects covered by the condition part of the rule in the training set,
- relative strength – quotient of the support to the number of all objects in the compatible decision class.

One of the software packages implementing rough set theory is ROSE2, which is an acronym for the Rough Set Data Explorer version 2. It was developed in the Laboratory of Intelligent Decision Support Systems of the Institute of Computing Science at the Poznań University of Technology [Prędki, Wilk 1999]. A limited version (10 attributes, 500 objects) of the software can be download from the web address: http://www-idss.cs.put.poznan.pl/rose/index.html.

Because ROSE2 has modular architecture, rule generation methods used in this paper will be based on the ones included in it.

There is also a freely available, open-source software package for machine learning problems called WEKA, developed at the University of Waikato, New Zeland, which can be obtained from the web page: http://www.cs.waikato.ac.nz/~ml/weka/. It is written in Java, it has several rule generation algorithms built in and can be modified to suit the users needs.

3. Case study

In this paper we present the qualitative extension to SMT based on the case study. It considers modeling of a so called caremap for medical procedures.

In most of the public healthcare systems the amount of money is limited, but the needs and expectations of the beneficiaries are constantly rising due to the aging of population, development of new technologies, etc. Without any additional funding the only improvement that is possible, is optimizing and streamlining the treatment of patients.

One of the attempts to control costs of inpatient care is to reduce the average length of hospital stay (LOS) without hurting patient care. The patient's length of stay after surgery appears to be one of the main components of this cost. If the LOS could be reduced, hospital costs would be much lower. The implementation of a clinical caremap is an attempt to introduce "best clinical practice" so that the hospital stays within prescribed LOS. This in turn is determined on a basis of best practices in a given clinical presentation and "adopted" in the hospital. Therefore, the implementation of caremaps has a positive impact on managing the LOS. Moreover, the caremap promotes quality patient care by incorporating existing standards, practice guidelines, and research findings, and by providing a mechanism to monitor quality care based on patient outcomes.

The caremap (also called clinical pathways, care management, or care planning) is a "multidisciplinary plan of best clinical practice for specified groups of patients with a particular diagnosis that aids the coordination and delivery of high quality care" [Ignatavicius, Hausman 1995]. It consists of four essential components: a timeline, categories of care processes (e.g., assessment, treatment, activity, etc.), a patient's outcome criteria, and the variance record. As a patient-oriented, cost-effective care management system, the caremap is increasingly being used. Despite the intent to define the essential components of care, there still are variations in how care will be delivered and how patients will respond. A patient's outcomes do not always follow the relevant caremap. Process variances will occur when the patient is not progressing at the standard rate through the caremap. Furthermore, these variances can be positive or negative. Positive variances occur when the patient progresses towards projected outcomes earlier than expected. Negative variance occurs when the patient fails to meet projected outcomes; either there is a delay in meeting the outcomes, or there is need for additional interventions previously unplanned.

As improvements in the quality of care are achieved through continuously redefining the caremap to reflect current best practices, variance analysis is a very important audit tool. Any negative variances might influence the patient's conditions; ultimately, they might influence the patient's LOS. Therefore, during the development of a caremap, how to efficiently evaluate the impact of negative variances in the clinical caremap in terms of LOS, is an important subject to consider.

Analysis of variances from the caremap and their impact on the patient's LOS can be a part of modeling process. Our goal is to use the Structured Modeling Technology for optimizing the work of a hospital where knowledge based on caremaps will be used as part of the model.

Our case study considers the caremap for radical prostatectomy procedure (complete surgical removal of the prostate gland) developed at the Ottawa Hospital. This caremap can be found in Appendix A. Based on the earlier analysis of variances in this caremap [Li 2004] we have obtained data about 125 patients treated with this procedure in the past.

The first step in modeling of the caremap is to identify the attributes that can be used to describe patient's state. Li has identified 15 attributes in the radical prostatectomy caremap, which are shown in Table 1.

Table 1 Attributes identified in the radical prostatectomy caremap

No	Attribute name	Description	Attribute values	Days of validity
1	Unders	patient's mental state	Abnormal, Normal	N/A
2	VS	vital signs	Abnormal, Normal	1, 2, 3
3	Actw	activity	Ambulate, No	1, 2, 3
4	Nutriw	nutrient input	Fluid, Regular	1, 2, 3
5	Nutrio	nutrient output	Nausea, Normal, Vomit	1, 2, 3
6	Painr	verbal pain score in rest	Medium, Mild, Nopain	1, 2, 3
7	Resp	respiratory functions	Mild, Normal	1, 2, 3
8	JP	Jackson-Pratt	D/C, Large, Medium, Small	1, 2, 3
9	Hema	evidence of hematoma	Bt, No, Y	1, 2, 3
10	Urineo	urine output	Adequate, Inadequate	1, 2, 3
11	Bowels	bowel sounds	Absent, Present	1, 2
12	Painm	verbal pain score with mobility	Medium, Mild, Nopain	1, 2, 3
13	Temp	body temperature	Abnormal, Normal	2, 3
14	Wound	wound healing	Medium, Mild, Normal	2, 3
15	LOS	length of stay	numeric	N/A

All of the identified attributes, except for the decision attribute LOS, are of the categorical type. The aforementioned caremap states that the LOS for a patient after the procedure is 3 days, 4 days in total. Two of the attributes (Unders, LOS) are not dependent on the day after the procedure, others have different values depending on the day.

Figure 1 shows the distribution of the values of attribute LOS in the obtained data.

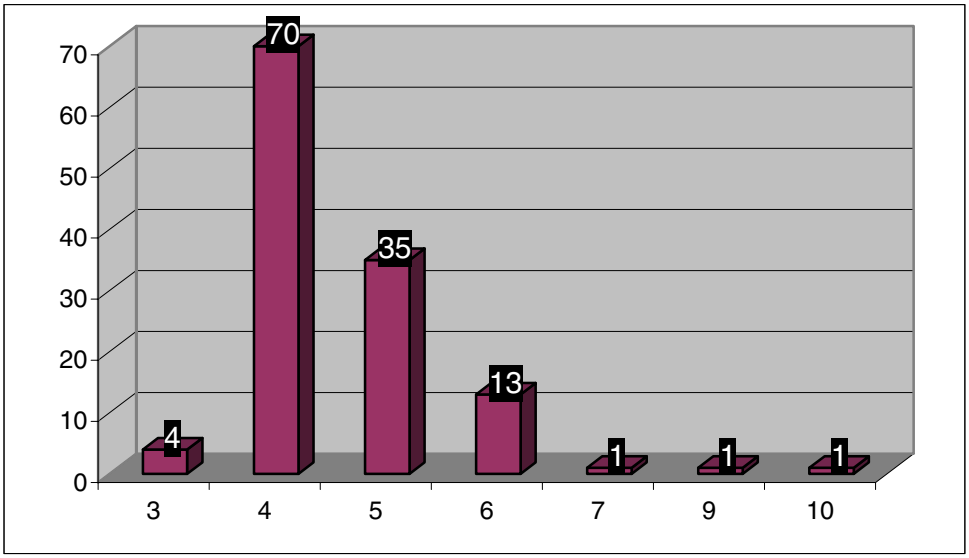


Figure 1 Distribution of the values of decision attribute Length of Stay (LOS)

It can be noted that out of 125 cases, 70 cases meet the expected LOS and 4 are even shorter. Most of the extended length of stay are 5 to 6 days. Only in 3 cases patient stayed for more then 6 days.

4. Qualitative extension to SMT

In order to extend the SMT application for modeling the qualitative data it was necessary to make several modifications.:

- the collection of types of models that can be built was modified to include the decision rule type model.
- the collection of entity types was modified to include a condition and decision attribute, because we have decided to define attributes as entities in the model.
- the list of possible types was extended to include categorical data type.
- SMT data structures for servlets were extended to handle decision rules

By making such modifications, it is possible to store the data that is a training set for the decision rules generation algorithm within the model instance. This data can be periodically updated using the data update mechanism. We assume that data updates will create a linear structure in time, without any branching.

In models of decision rule type, decision rules are stored as the results. Thus, we have adopted the database structure for storing results to also handle decision rules.

Following is the detailed description of the structures in data-warehouse used for the case study of modeling the radical prostatectomy caremap. The important difference from the typical approach, is that most of the attributes identified in the caremap are day dependent, where some, especially decision attribute LOS, are not. Thus it is necessary to use the indexing mechanism in the SMT.

Two indices will be used:

- index *i* to identify the patients,
- index *d* for day dependent attributes to identify the day for which it is valid.

Because of the use of two indices, there are two data tables generated by the SMT application using the corresponding symbolic model specification. Although the data tables are generated automatically by SMT, we are presenting their content here, to illustrate the underlying data structures.

Table storing data for day-independent attributes is named following the schema: "D_"+model_name+"_I", where model_name is the selected model name and "I" comes from the index name. Assuming that model name is "PROSTATE" then the table will be named "D_PROSTATE_I". Table 2 shows the data structure of this table for prostatectomy caremap.

Table 2 Structure of the table for storing day-independent data

Column name	Data type	Description
DATA_ID	Number	used to identify the data update
I	VARCHAR (20)	value of index *i*; identifies the patient
UNDERS	VARCHAR (128)	values of attributes for patient identified by *i*
LOS	VARCHAR (128)	

The table storing data for day-dependent attributes is named in the following way: "D_"+model_name+"_DI", where model_name is the selected model name and D and I are indices. Depending on the sequence order of the indices their order may be reversed. So for the "PROSTATE" model this table would be named "D_PROSTATE_DI" or "D_PROSTATE_ID". Table 3 shows the data structure of this table for the prostatectomy caremap.

Table 3 Structure of the table for storing day-dependent data

Column name	Data type	Description
DATA_ID	Number	used to identify the data update
D	VARCHAR (20)	value of index *d*; identifies the day for the attribute
I	VARCHAR (20)	value of index *i*; identifies the patient
VS	VARCHAR (128)	values of attributes for patient identified by *i* on day *d* (may be null if attribute is not valid for the given day)
ACTW	VARCHAR (128)	
NUTRIW	VARCHAR (128)	
NUTRIO	VARCHAR (128)	
PAINR	VARCHAR (128)	
RESP	VARCHAR (128)	
JP	VARCHAR (128)	
HEMA	VARCHAR (128)	
URINEO	VARCHAR (128)	
BOWELS	VARCHAR (128)	
PAINM	VARCHAR (128)	
TEMP	VARCHAR (128)	
WOUND	VARCHAR (128)	

As mentioned earlier the decision rules are stored as a result of model analysis. Thus they are stored in a table for results. It is named following the schema: "R_"+model_name+"_DI", where model_name is the selected model name and D and I are indices. Depending on the sequence order of the indices their order may be reversed. So for the "PROSTATE" model this table would be named "R_PROSTATE_DI" or "R_PROSTATE_ID". Table 4 shows the structure of this table, which is independent of the data.

Table 4 Structure of the table for storing results

Column name	Data type	Description
ENTITY_ID	Number	used to identify the entity – attribute
D	VARCHAR (20)	value of index *d*;
I	VARCHAR (20)	value of index *i*;
OPERATOR	VARCHAR (30)	operator

VALUE_NUMBER	Number	value if it is numeric
VALUE_STRING	VARCHAR (30)	value if it is string
RTASK_ID	Number	runnable task ID

Such a structure of the data table is sufficient for storing decision rules, but it is necessary to explain how exactly it is done.

Each decision rule is stored in this table in two parts:

- the part containing all elementary conditions and decisions,
- the part containing all meta data describing rules, like rule strength, etc.

Each elementary condition is stored in one table row in the following way:

- ENTITY_ID – identifies the attribute in the elementary condition,
- D – indicates the day for the considered attribute; if equal -1 the day is irrelevant,
- I – uniquely identifies the decision rule,
- OPERATOR – is made up of two characters: second character indicates whether the elementary condition is part of the condition part of the decision rule ("c") or decision part ("d"), first character responds to the relation type in the elementary condition in the following way:
  - =, >, < are self-explanatory,
  - "l" is less or equal,
  - "m" is more or equal,
  - "i" is one of the set of values.
- VALUE_NUMBER – may be the elementary condition value, if the attribute is numeric, otherwise it is null,
- VALUE_STRING – the value of the elementary condition; may be null if VALUE_NUMBER is not null,
- RTASK_ID – the associated runnable task identifier.

All meta data describing rules is stored in one row of the table in the following way:

- ENTITY_ID – is null,
- D – is null,
- I – identifies the decision rule,
- OPERATOR – is null,
- VALUE_NUMBER – is the value of the parameter,
- VALUE_STRING – is the name of the parameter,
- RTASK_ID – the associated runnable task identifier.

The part of the table containing one decision rule is shown on Figure 2.

ENTITY_ID	D	I	OPERATOR	VALUE_NUMBER	VALUE_STRING	RTASK_ID
54	1	52	=c		No	7
57	1	52	=c		Mild	7
51	1	52	=c		Nopain	7
51	2	52	=c		Mild	7
34	-1	52	=d		4	7
		52		1	Strength	7
		52		1	Support	7
		52		1	Confidence	7
		52		2	Day	7
		52		100	Discrimination	7
		52		1	RelativeStrength	7

Figure 2 Part of the screen showing one decision rule stored in the results table

5. Use of qualitative SMT

The entire modeling process for the hospital management system may be divided into the following stages:

1. Identification of all procedures in the hospital with available caremaps.
2. Definition of the main optimization model, for which some of the input variables will be instantiated using decision rules for each caremap.
3. Definition of models for each caremap which are responsible for storing the historical data and for rule induction.

This report describes only the work done for stage 3.

To define the model for the selected caremap it is necessary to follow certain rules:

- all attributes identified in the caremap are divided into two sets: attributes that are time dependent (set A_d) and attributes that are time independent (set A_i).
- exactly two indices are defined in the model:
  - i – used to identify the patient,
  - d – used to describe the timescale in days.
- all attributes in sets A_d and A_i are defined as parameters in the model
- attributes from the set A_i are indexed only by index i,
- attributes from the set A_d are indexed by both indices – i, d
- we consider only one decision attribute responding to the patient's length of stay.

The screen of the SMT application after logon is shown on Figure 3. On the left side of the screen the main application menu is found. At the bottom there is a status line with several application buttons. All operations take place in the central window.

Figure 3 Main SMT Web application window

After logging into the SMT web application tone can define a new mode of the caremap by use of decision rules, by using the "New" button (as on Figure 3). The model description window will appear as shown on Figure 4. Please note that the model class should be selected as "Decision Rules".

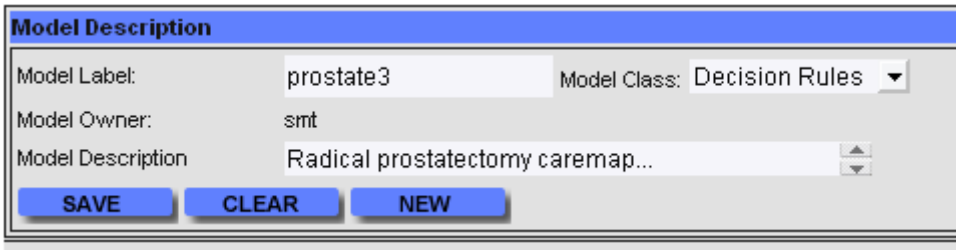


Figure 4 Model description window

The first step after the creation of the new model is to define indices. We define two indices, named respectively *d* (for day) and *i* (for patient id). In order to do this it is necessary to use the "Indices" command from the "Model specification" section in the main menu. Screen for index definition is shown on Figure 5.

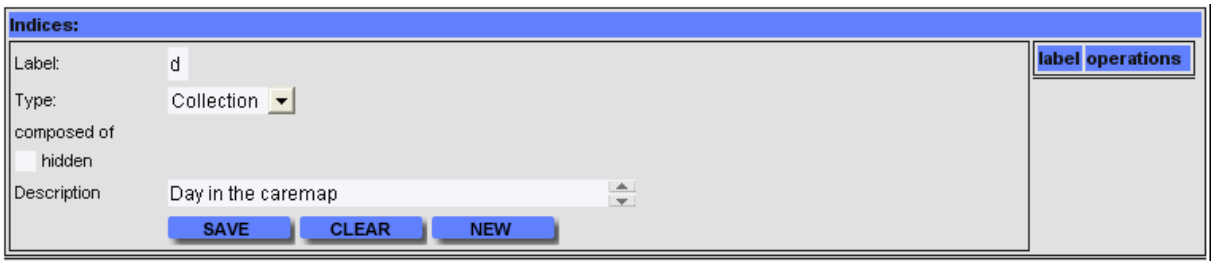


Figure 5 Index definition window

After defining the indices *d* and *i* the screen should look like on Figure 6.

Figure 6 Indices window with two indices defined

When both indices are defined it is possible to input information about attributes used in the caremap. Attributes are defined in the SMT model as entities, by using the "Entities" command in the main menu. Figures 7, 8 and 9 show the screens during the definition of the attributes. Figure 7 shows the attribute "Unders" from the prostatectomy caremap, that is day independent, thus it is indexed only by index *i*. On the right there is a list of all defined attributes (parameters).

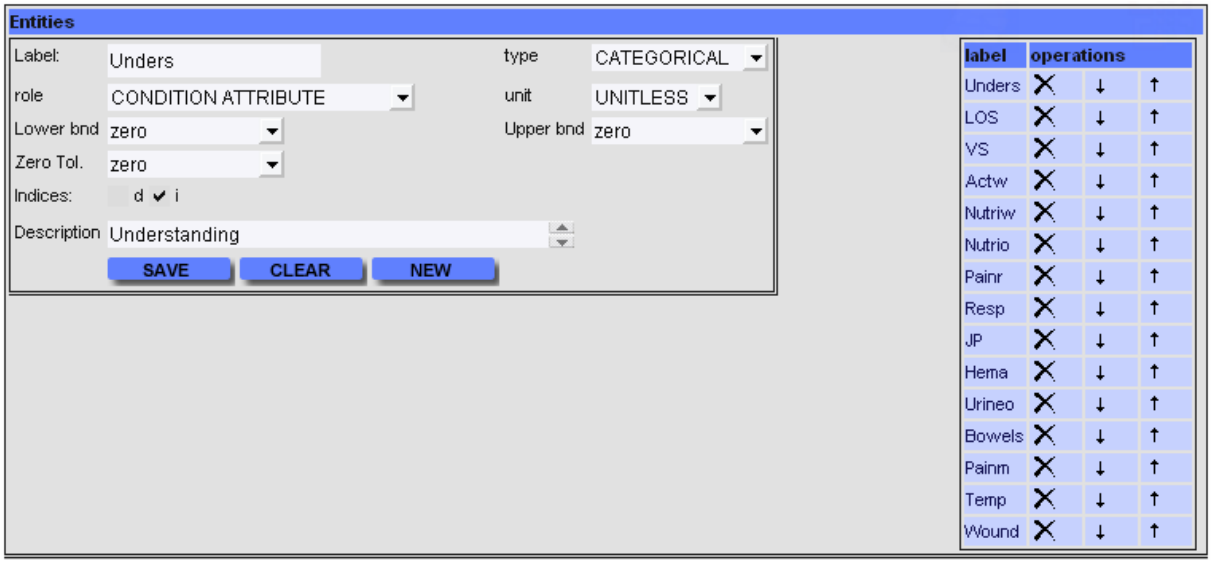


Figure 7 Entity definition window with defined attribute that is day-independent

Figure 8 shows the definition of the attribute "VS", which is day dependent, thus it is indexed by both indices: *d* and *i*.

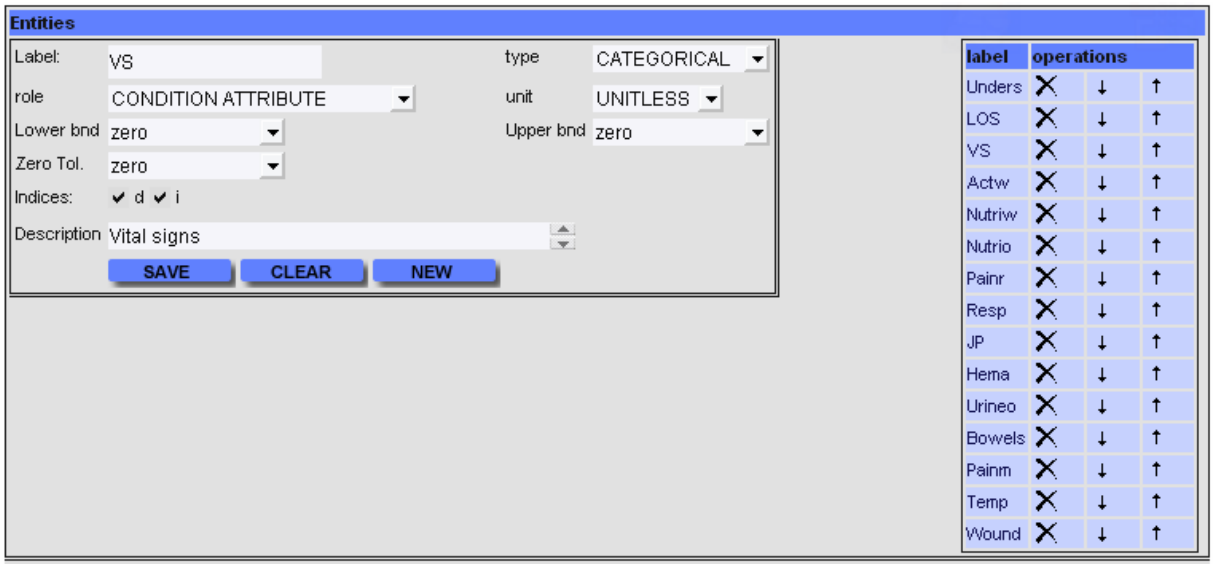


Figure 8 Definition of entity which is a day-dependent condition attribute

Figure 9 shows the definition of the decision attribute "LOS". As a decision attribute it is day independent and indexed only by index *i*. Please note the different roles for the attributes on Figures 6 through 8.

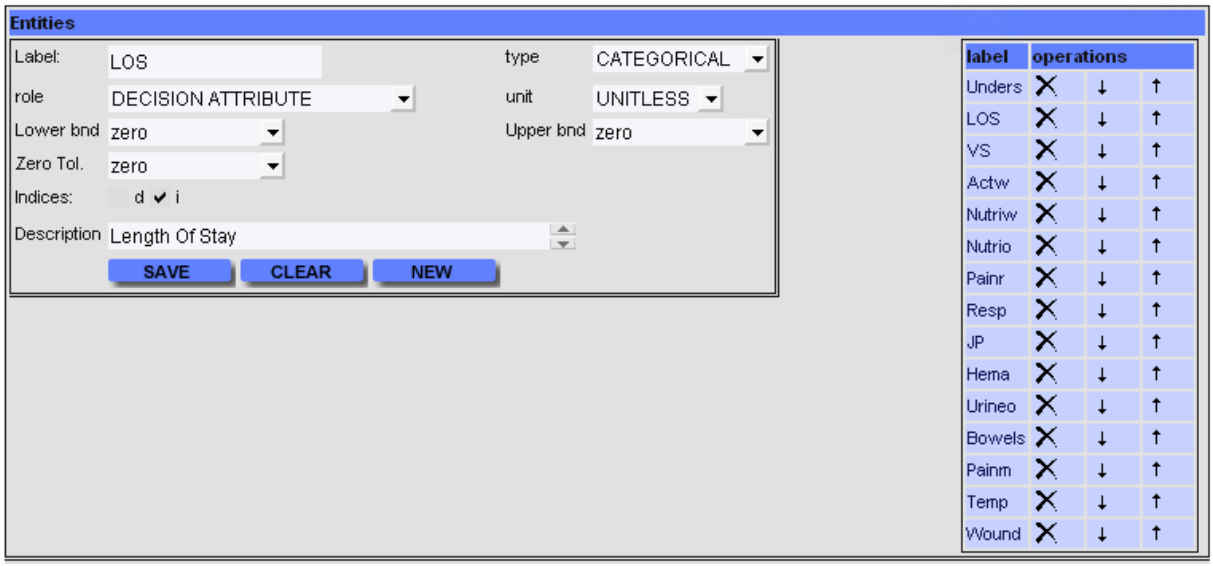


Figure 9 Definition of entity which is a decision attribute

After defining all attributes as parameters in the model it is necessary to generate the data-warehouse structures. In order to do this the model must be locked, by going to the model definition screen (Figure 10) and using the "Lock" button. Then the "Generation" command in "Data Warehouse" becomes active.

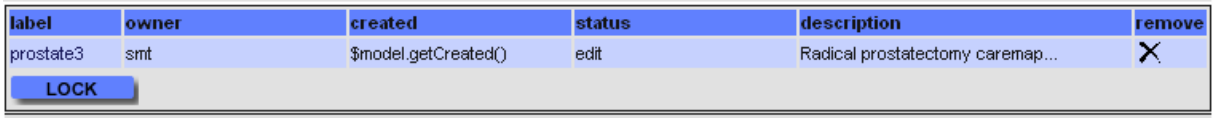


Figure 10 Window with model definition

After using the "Generation" command the user is asked for confirmation as on Figure 11.

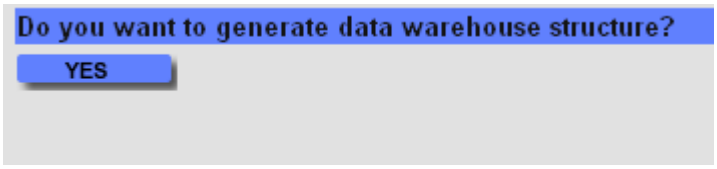


Figure 11 Confirmation screen before data warehouse generation

Historical data concerning patients that were already released is put in the SMT data warehouse using the data import mechanism. In this case it is necessary to prepare two files: file with the attributes from the set A_i (Figure 12) and files with the attributes from the set A_d (Figure 13).

```
Unders,LOS,i
Normal,4,0
Normal,4,1
Normal,6,2
Normal,4,3
Normal,5,4
```

Figure 12 Beginning of the CSV file containing data for day-independent attributes

```
d,VS,Actw,Nutriw,Nutrio,Painr,Resp,JP,Hema,Urineo,Bowels,Painm,Temp,Wound,i
1,Normal,Ambulate,Fluid,Normal,Nopain,Normal,Medium,Bt,Adequate,Present,Nopain,,,0
2,Normal,Ambulate,Regular,Normal,Nopain,Normal,D/C,Bt,Adequate,Present,Nopain,Normal,Normal,0
3,Normal,Ambulate,Regular,Normal,Nopain,Normal,D/C,Bt,Adequate,,Nopain,Normal,Normal,0
1,Normal,Ambulate,Fluid,Vomit,Nopain,Normal,Small,No,Adequate,Present,Mild,,,1
2,Normal,Ambulate,Regular,Normal,Nopain,Normal,D/C,No,Adequate,Present,Mild,Normal,Mild,1
3,Normal,Ambulate,Regular,Normal,Nopain,Normal,D/C,No,Adequate,,Nopain,Normal,Normal,1
1,Normal,Ambulate,Fluid,Normal,Mild,Normal,Medium,No,Adequate,Present,Medium,,,2
2,Normal,Ambulate,Regular,Normal,Medium,Normal,Small,No,Adequate,Present,Medium,Normal,Normal,2
3,Normal,Ambulate,Regular,Normal,Mild,Normal,Small,No,Adequate,,Mild,Normal,Normal,2
```

Figure 13 Beginning of the CSV file containing data for day-dependent attributes

The first line in a CVS file has to contain the names of the parameters responding to the attribute names plus the names of the indices used for indexing these parameters. Values can be separated by any ASCII character (typically it is a comma or semicolon). Next lines contain values for each patient and parameter. If some attributes are meaningless (don't have values) for given day, their responding parameters should be left blank.

After using the "Data import" command from the main menu, the data import window is shown as on Figure 14. The user has to provide information about locations of both

CSV files (“Choose” button) and enter the character used in them as a delimiter (typically comma). The data import process starts after the “Save” button is pressed.

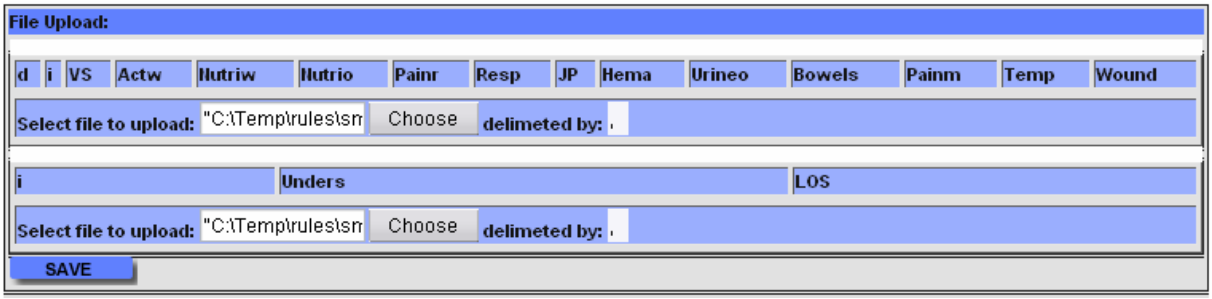


Figure 14 Data import window

In order to generate decision rules as the output of the considered model it is necessary to define a new model instance and generate a runnable task. The definition of the new model instance starts after selecting the “Definition” command in the “Model Instance” submenu. The resulting screen, lists all available instances, as shown on Figure 15.

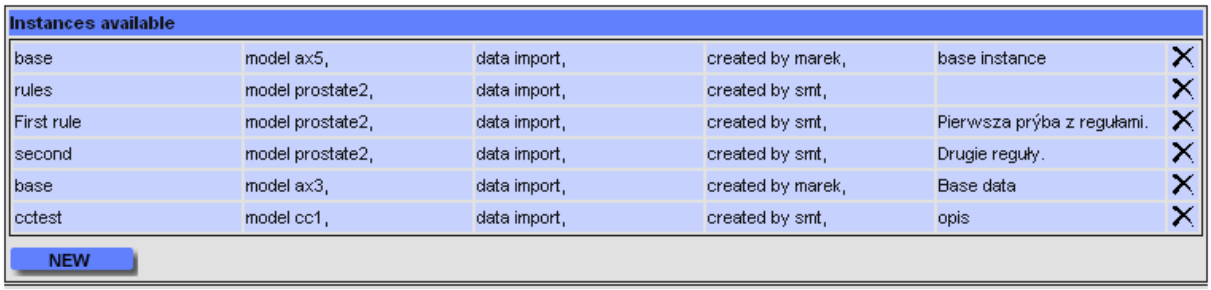


Figure 15 Instance selection window

When it is necessary to create a new model instance, pressing the “New” button will open another screen, shown on Figure 16. Here the instance name and description are defined.

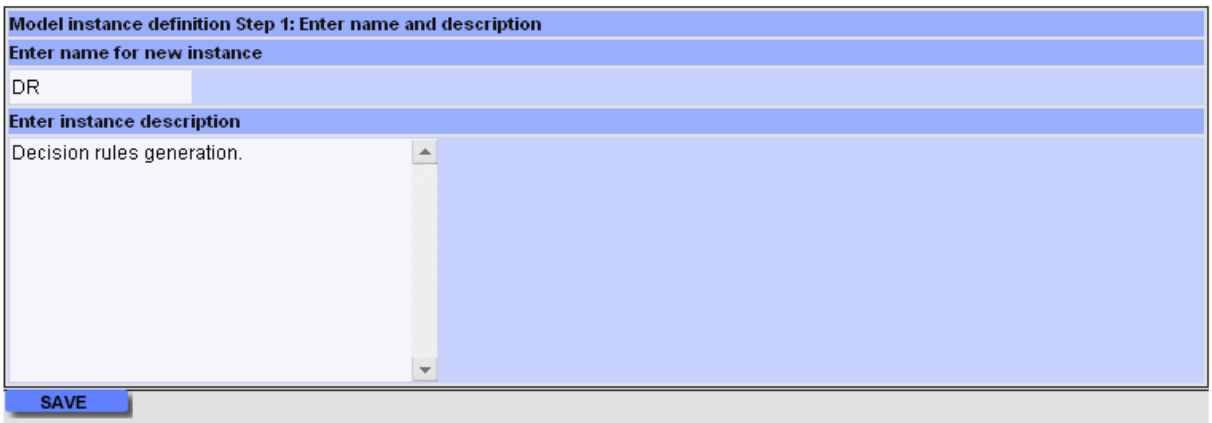
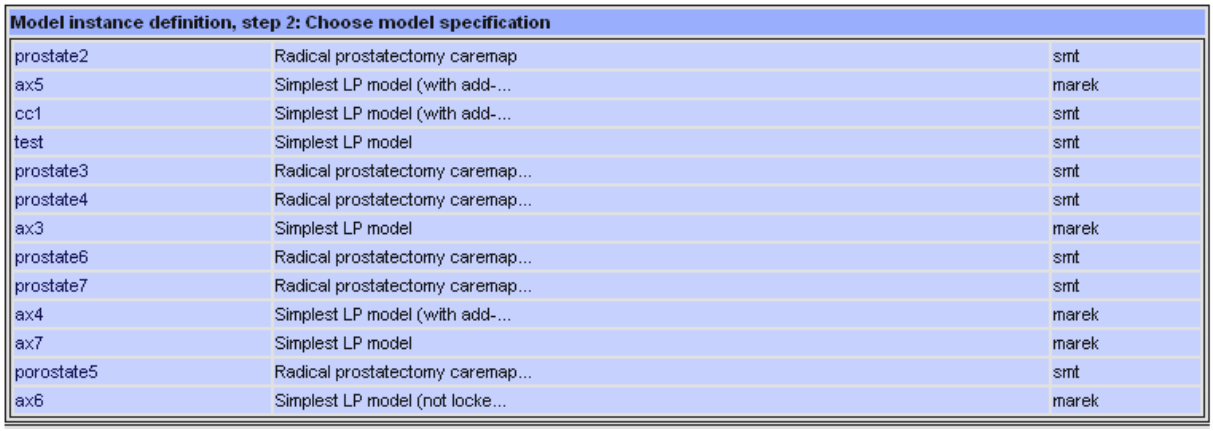


Figure 16 Instance definition Window – basic information

Because each instance has to based on the model specifications it is necessary to select in the next step, one of the existing model specifications, as shown on Figure 17.



Model instance definition, step 2: Choose model specification		
prostate2	Radical prostatectomy caremap	smt
ax5	Simplest LP model (with add-...	marek
cc1	Simplest LP model (with add-...	smt
test	Simplest LP model	smt
prostate3	Radical prostatectomy caremap...	smt
prostate4	Radical prostatectomy caremap...	smt
ax3	Simplest LP model	marek
prostate6	Radical prostatectomy caremap...	smt
prostate7	Radical prostatectomy caremap...	smt
ax4	Simplest LP model (with add-...	marek
ax7	Simplest LP model	marek
porostate5	Radical prostatectomy caremap...	smt
ax6	Simplest LP model (not locke...	marek

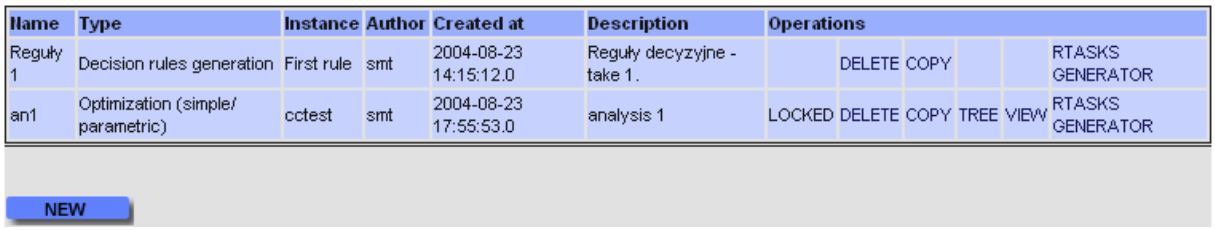
Figure 17 Instance definition window – model specification selection

Model instance joins the model specification with the data, so in the next step it is necessary to select one of the data updates that will constitute the model instance, as shown on Figure 18.

Model instance definition Step 3. Select data update
import

Figure 18 Instance definition window – data update selection

When the model instance is defined it is possible to define the new model analysis. This is done by using the "Analysis" command from the "Model Instance" submenu. The resulting screen shows all available model analysis tasks, as shown on Figure 19.



Name	Type	Instance	Author	Created at	Description	Operations					
Reguły 1	Decision rules generation	First rule	smt	2004-08-23 14:15:12.0	Reguły decyzyjne - take 1.		DELETE	COPY			RTASKS GENERATOR
an1	Optimization (simple/ parametric)	cctest	smt	2004-08-23 17:55:53.0	analysis 1	LOCKED	DELETE	COPY	TREE	VIEW	RTASKS GENERATOR

NEW

Figure 19 Model analysis window

If a new model analysis task is to be defined, after pressing the "New" button, a model instance selection screen appears, as shown on Figure 20.

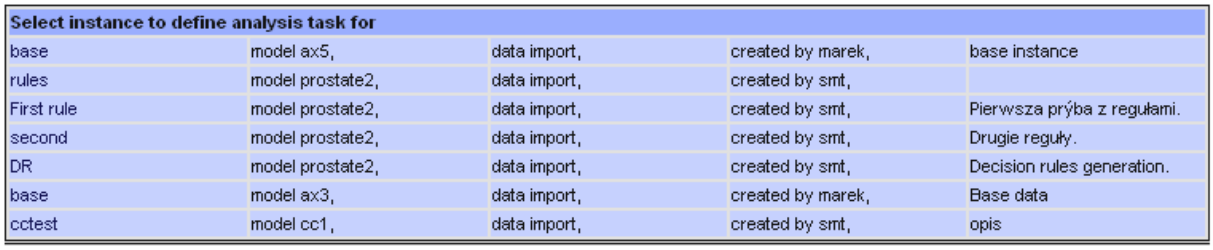


Figure 20 Analysis task window – instance selection

After selecting the model instance it is necessary to select its type. For the considered type of analysis it should be "Decision rules generation", as shown on Figure 21.

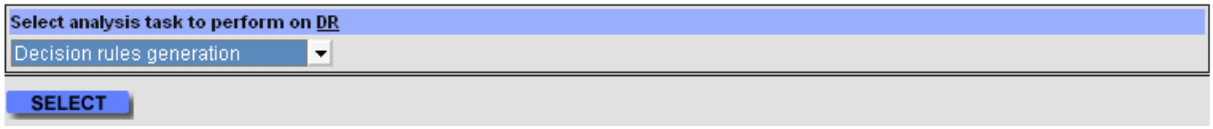


Figure 21 Analysis task type selection

The last element to define, is an analysis name for the group of tasks, as shown on Figure 22.

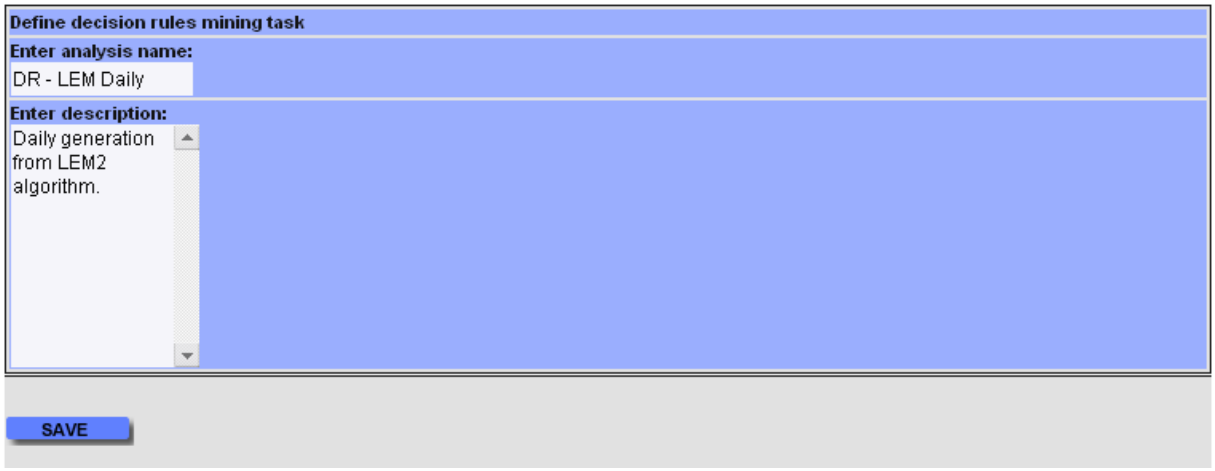


Figure 22 Decision rules analysis task description window

Newly defined analysis will show in the list now, as on Figure 23.

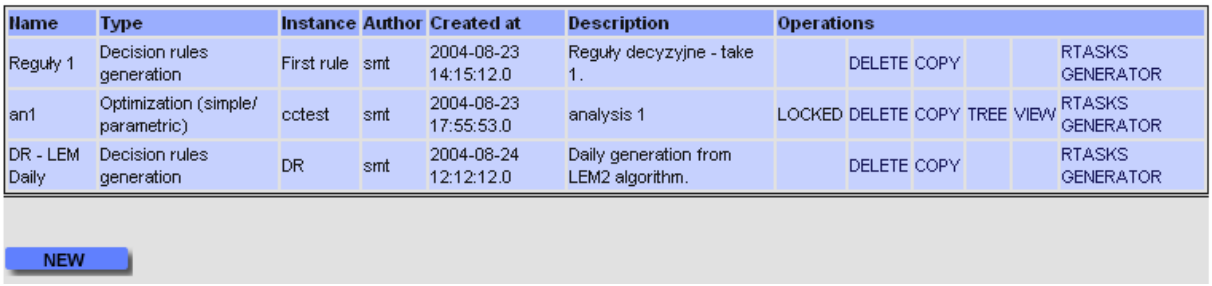


Figure 23 List of defined analysis tasks

To generate runnable tasks it is necessary to use the "RTASKS GENERATOR" command next to the chosen analysis task. The screen shown on Figure 24 will appear.

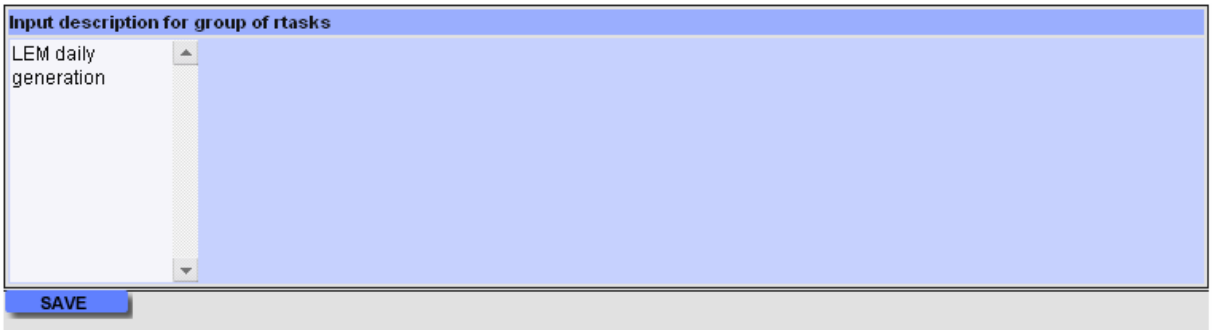


Figure 24 Runnable task description window

Thus the process of generation of runnable tasks for rule generation is finished.

6. DRSMT software

DRSMT (acronym for Decision Rules Structured Modeling Technology) consists in fact of two parts:

- a library of classes designed to read/write data and decision rules in different formats and to interface with rule generation software (namespace DRSMTLib)
- a simple, menu-driven, user interface for Microsoft Windows (namespace DRSMTGen).

6.1. The Application

The Figure 25 shows the main screen after starting the application. The interface is divided in two parts: menu and log window. Menu allows invoking the commands and the log window is where all output information is presented to the user.

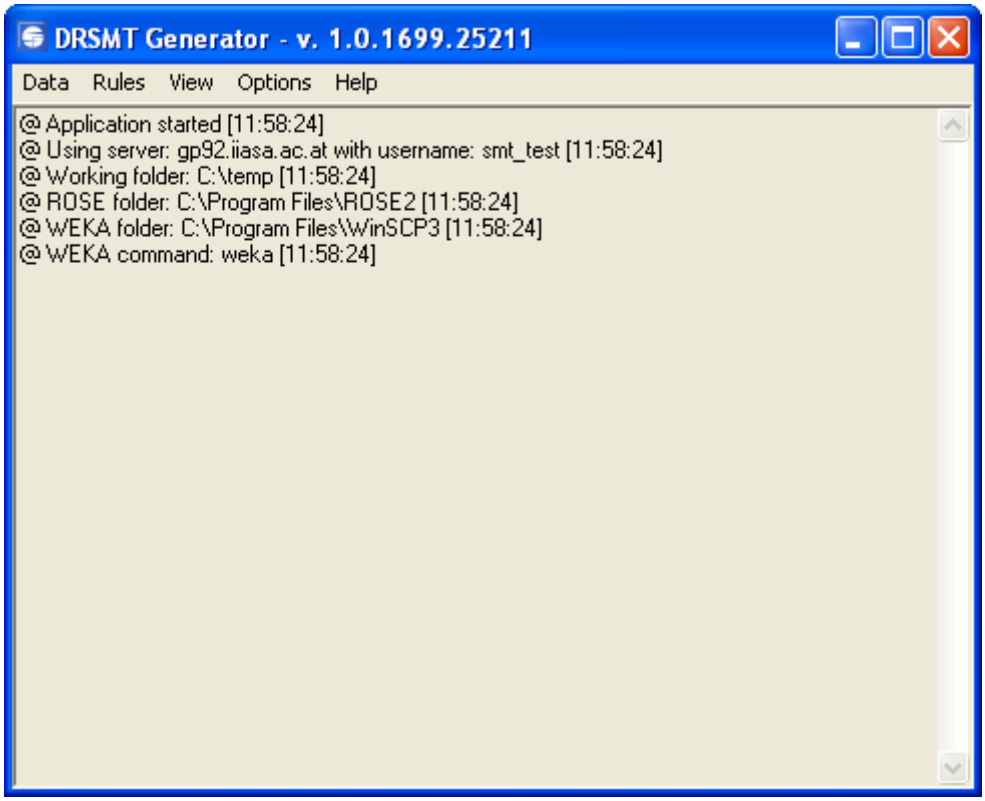


Figure 25 Main screen of the DRSMT application

During the first run of the application several settings should be defined:

- SMT connection settings – server address, user name and password for the server (Figure 26),
- working folder – folder in which all temporary files are stored,
- ROSE folder – folder containing the rule generation modules of the ROSE2 software package (typically c:\Program Files\ROSE2\),
- WEKA folder – folder containing WEKA software,
- WEKA command – command used in WEKA to generate decision rules.

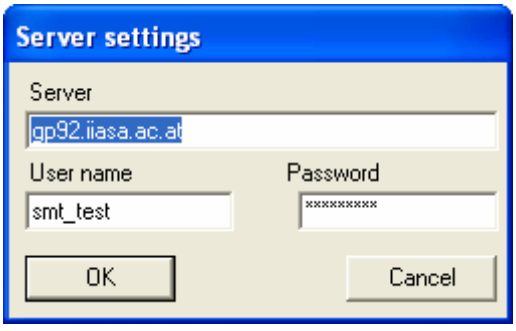


Figure 26 Server settings window In DRSMT

All these settings are available in the Options menu. The settings are stored in the machine's registry. All operations on data are done using the "Data" menu.

DRSMT is able to read data in the following formats:

- proprietary XML format (see Appendix C),
- CSV text format described earlier,
- SMT data-warehouse.

DRSMT is able to write data in the following formats:

- proprietary XML format,
- ISF format used by ROSE2 software,
- ARFF format used by WEKA software,
- CSV files for import to SMT (as described earlier).

DRSMT is capable of saving the data for importing through the SMT application data import mechanism and reading the data from the plain CSV file, where attributes from the set A_d have names followed by underscore and day number. The first line contains the attribute names and the following contain data. Data should be separated using semicolons.

```
Unders_0;VS_1;Actw_1;Nutriw_1;Nutrio_1;Painr_1;Resp_1;JP_1;Hema_1;Urineo_1;Bowels_1;Painm_1;VS_2;Temp_2;Actw_2;Nutriw_2;Nutrio
_2;Painr_2;Resp_2;JP_2;Hema_2;Urineo_2;Bowels_2;Painm_2;Wound_2;VS_3;Temp_3;Actw_3;Nutriw_3;Nutrio_3;Painr_3;Resp_3;JP_3;Hema_
3;Urineo_3;Painm_3;Wound_3;LOS
Normal;Normal;Ambulate;Fluid;Normal;Nopain;Normal;Medium;Bt;Adequate;Present;Nopain;Normal;Normal;Ambulate;Regular;Normal;Nopain;Norm
al;D/C;Bt;Adequate;Present;Nopain;Normal;Normal;Normal;Ambulate;Regular;Normal;Nopain;Normal;D/C;Bt;Adequate;Nopain;Normal;4
Normal;Normal;Ambulate;Fluid;Vomit;Nopain;Normal;Small;No;Adequate;Present;Mild;Normal;Normal;Ambulate;Regular;Normal;Nopain;Normal;D/C;
No;Adequate;Present;Mild;Mild;Normal;Normal;Ambulate;Regular;Normal;Nopain;Normal;D/C;No;Adequate;Nopain;Normal;4
Normal;Normal;Ambulate;Fluid;Normal;Mild;Normal;Medium;No;Adequate;Present;Medium;Normal;Normal;Ambulate;Regular;Normal;Medium;Normal;
Small;No;Adequate;Present;Medium;Normal;Normal;Normal;Ambulate;Regular;Normal;Mild;Normal;Small;No;Adequate;Mild;Normal;6
Normal;Normal;No;Fluid;Normal;Medium;Normal;Small;No;Adequate;Present;Medium;Normal;Normal;No;Regular;Normal;Medium;Normal;Small;No;A
dequate;Present;Medium;Normal;Normal;Normal;Ambulate;Regular;Normal;Mild;Normal;D/C;No;Adequate;Mild;Normal;4
Normal;Normal;Ambulate;Fluid;Normal;Mild;Normal;Small;No;Adequate;Present;Mild;Normal;Normal;Ambulate;Fluid;Normal;Mild;Normal;Small;Bt;Ade
quate;Present;Mild;Normal;Normal;Normal;Ambulate;Regular;Normal;Mild;Normal;D/C;Bt;Adequate;Mild;Normal;5
Normal;Normal;Ambulate;Fluid;Normal;Nopain;Normal;Small;Y;Adequate;Present;Medium;Normal;Abnormal;Ambulate;Fluid;Normal;Mild;Normal;Sma
ll;Y;Adequate;Present;Mild;Medium;Normal;Abnormal;Ambulate;Regular;Normal;Mild;Normal;Small;Y;Adequate;Mild;Medium;7
Normal;Normal;Ambulate;Fluid;Normal;Nopain;Normal;Small;Bt;Adequate;Present;Mild;Normal;Normal;Ambulate;Regular;Normal;Mild;Normal;Small;B
t;Adequate;Present;Mild;Normal;Normal;Normal;Ambulate;Regular;Normal;Nopain;Normal;Small;Bt;Adequate;Mild;Mild;4
Normal;Normal;Ambulate;Fluid;Normal;Mild;Normal;Small;Bt;Adequate;Present;Mild;Normal;Normal;Ambulate;Fluid;Normal;Nopain;Normal;Small;Bt;A
dequate;Present;Nopain;Normal;Normal;Normal;Ambulate;Regular;Normal;Nopain;Normal;D/C;No;Adequate;Nopain;Normal;4
```

Figure 27 Beginning of the CSV file for importing data into DRSMT software

Using the "Rules" menu it is possible to generate, read and write decision rules.

As stated earlier, three rule generation approaches are available:

- Local, minimal covering algorithm (LEM2) from ROSE2 system,
- all rules algorithm (EXPLORE) from ROSE2 system,
- algorithms available in WEKA software (support only for data export in ARFF format and running WEKA commands).

For each approach three aforementioned types of rules can be generated.

DRSMT is able to write decision rules in the following formats:

- proprietary XML format (see Appendix D),
- SMT data-warehouse,
- HTML file for documentation.

DRSMT is able to read decision rules in following formats:

- proprietary XML format,
- SMT data-warehouse,
- RLF format used by ROSE2 software.

When a rule generation module is run its output is captured and displayed in the log window, along with the system messages with timestamp, as shown on Figure 28.

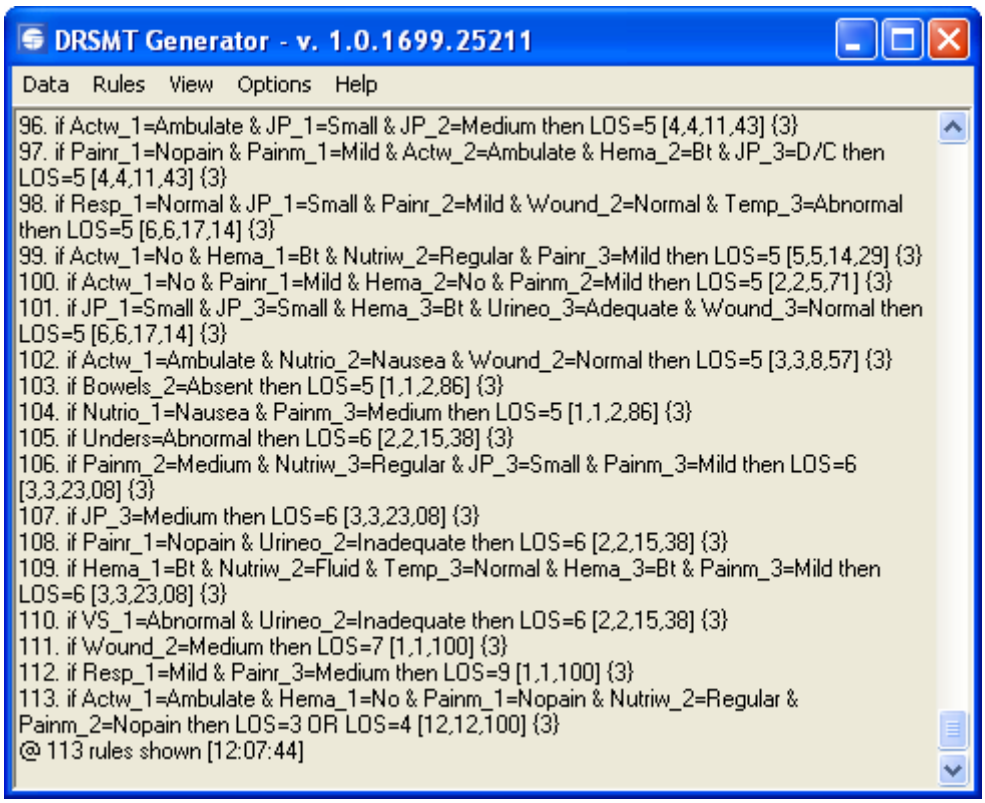


Figure 28 DRSMT screen after rule generation

Because DRSMT is a standalone application, all operations on data and decision rules can be done either offline (without connection to SMT data-warehouse) or online.

For online access it is necessary to define, in the SMT web interface, an analysis task of the decision rule generation type. Only if at least one such task is defined it is possible to read data from SMT because DRSMT identifies the model and corrects data updates, based on the information obtained from the analysis task.

To obtain a connection to the SMT data-warehouse an analysis task must be selected, by using the “Select task” command from the “Data” menu. All analysis tasks of decision rule type are enlisted and the user can select one of them. Based on this selection an appropriate model is also selected. If the data is read from SMT it will only be read up to the data update associated with the analysis task.

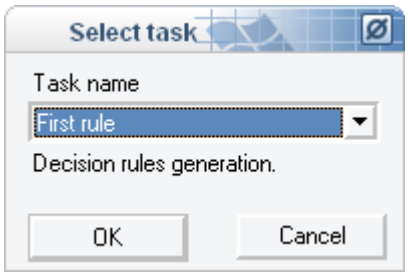


Figure 29 Task selection window in DRSMT

If data is read it is always possible to generate decision rules, but only if the results for the selected analysis task were not updated (rules haven't been written into the data-warehouse) it is possible to store the decision rules in SMT for that task. Otherwise a new analysis task has to be defined. Such a procedural constraint is necessary to achieve a unique correspondence between an analysis task and the results.

After selecting the analysis task there is a message concerning the state of the given task. If the state is "READY" it is possible to write a new set of decision rules. If it is "DONE" a set of decision rules has already been written for this task earlier.

It is possible to read the decision rules for any given task from SMT, although if the task state is "READY" the resulting rule set will be empty.

When data is imported into the SMT data warehouse it is possible to induce decision rules. At this stage it is necessary to use the DRSMT software on the client computer. DRSMT is able to read the data straight from the SMT data warehouse. After reading the data, the user can select one of the rule generation approaches:

1. Daily rule generation.
2. Meta rule generation.
3. Coalition rule generation.

In the first approach the data set is divided so that each partition contains only data from the given day and earlier days. If all attributes were day-independent the data would include an entire data set. Rules are generated for each partition set and they should be used to classify patients on the given day of their stay in the hospital. Such an approach is dictated when we would like to classify the patients for which only a limited set of attributes is known at the time.

In the second approach meta rules are generated. By meta rules we understand that the decision attribute for which rules are generated is not the main decision attribute, but for each day every attribute from the next day becomes a decision attribute. Thus we obtain decision rules that are able to predict the value of some attribute on the next day based on the information from the current day.

In the third approach, based on the assumption that the decision attribute LOS (Length Of Stay) is numeric, decision rules are generated for the coalition of decision classes. This way it is possible to generate rules that in the decision part state that the patient's LOS will be at least or at most as the stated value.

For each approach any of the available rule generation algorithms can be used. Currently it includes:

- full support for "local minimal covering" algorithm (LEM2) from ROSE2,

- full support for “all rule generation” algorithm (Explore) from ROSE2,
- basic support (data export and runtime) for algorithms from WEKA.

After the rule generation process is finished all rules may be written into the SMT data warehouse. They are stored in the table designed to store the results of various types of analysis. Each rule is stored not only in its form but also with additional information, like support, strength, etc.

DRSMT is also able to store and read data from the proprietary XML format files.

6.2. DRSMT library

As it was stated before, DRSMT software is in fact based on the library of classes. This library consists of N classes which can be divided into X categories:

- classes for reading data, based on the abstract class InputDataReader,
- classes for writing data, based on the abstract class InputDataWriter,
- classes for rule generation, based on the abstract class Generator,
- classes for reading rules, based on the abstract class RuleReader,
- classes for writing rules, based on the abstract class RuleWriter,
- classes representing the data – Attribute and Case,
- classes representing the rules – Condition and Rule,
- main library class – DRSMT.

All classes are shown on the UML class diagram in Figure 30.

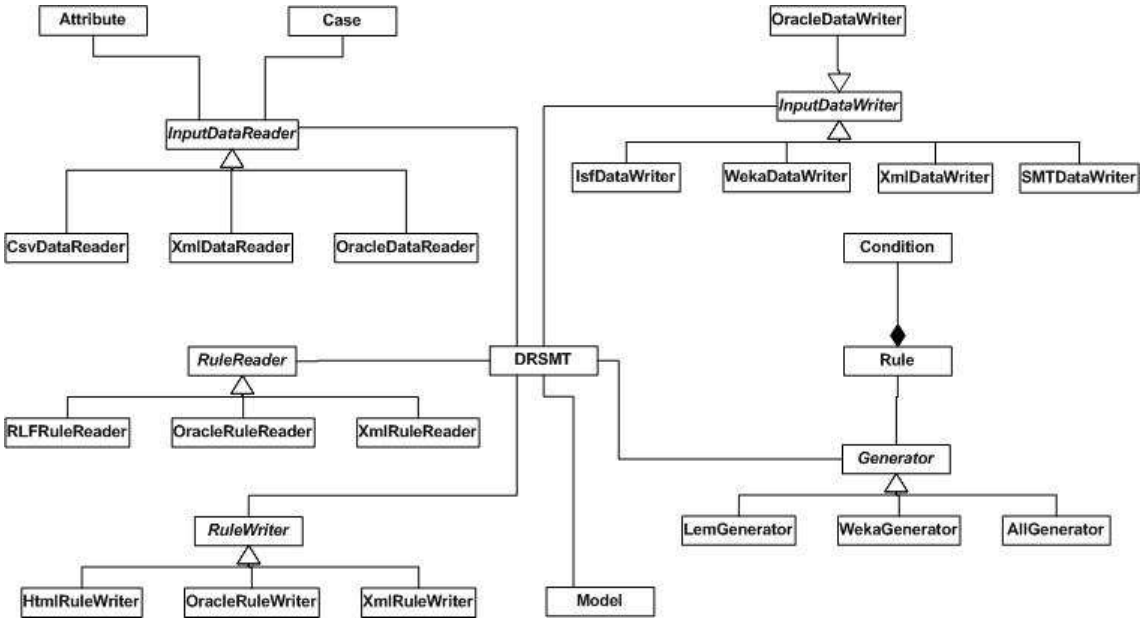


Figure 30 Class diagram of the DRSMTLib library

To enable the library to read or write new data or rule formats, or to use new rule generation techniques it is necessary to derive new classes from the existing ones.

The entire class library is self documented in the code.

The DRSMTLib library is dependent on two external libraries:

- Telperion.Utility library exposing some basic utility methods.
- Oracle.Data library used to access the SMT data-warehouse, downloadable from www.oracle.com.

To use the DRSMT software the client machine has to have the .Net Framework version 1.1 installed. The setup program for the software is provided.

7. Conclusions

Structured Modeling Technology being developed in IIASA for algebraic models has been extended to include models based on qualitative data, especially models using decision rules as knowledge representation. It has been proven that SMT is able to store model specification designed for rule generation, where the model data constitutes the training set for decision rule induction algorithms and decision rules are stored as the result. The extension of the SMT is as generic as possible to suit any kind of decision rules. The entire concept was tested on a case study for modeling the caremap for radical prostatectomy procedure.

Future versions of SMT will support the development of models that will use results and especially decision rules of other models, as their model data.

In case of care map modeling it is necessary to provide structures to store data about patients currently treated at the hospital and for providing the classifier that will read decision rules and data, to provide predictions of their length of stay and possible variations from the care map.

The decision rule paradigm may provide very useful functionality for Algebraic Models (AM) developed for decision support.

We mention here only two areas of such functionality:

1. Data mining and knowledge discovery from the results of AM.
2. Support for selection of sets of data to be used for model parameters.

The paper reports the results of the clinical decision support research which is a part of the collaborative research between the University of Ottawa and IIASA.

The results show advantages of expanding the Structured Modeling Technology with rules-based modeling paradigm. Such expanded SMT was applied to a case study in the area of clinical decision making using data obtained from The Ottawa Hospital in Ottawa, Canada. Modeling of patient management process in a hospital is significant undertaking and thus, due to a limited time available the reported research resulted in a prototype system only.

In order to conduct a prospective evaluation of a system in a hospital, it is necessary to continue the research by incorporating different medical presentations and considering different types of the end users. This research will be conducted and supported within a collaborative framework signed by the University of Ottawa and IIASA and it will involve researchers and the graduate students from both institutions. It is important to stress here that applicability of the expanded SMT is not limited or bounded to the clinical area. Similar in scope and complexity decision support problems exist especially in the area of sustainable development, aquatic systems and water

management and a comprehensive decision support can be developed for respective decision makers from these areas.

Finally for the future, it may be possible to extend SMT in such way, that it will not only include the decision rule paradigm, but for example artificial neural network, decision trees or Bayesian belief networks.

Bibliography

B.G. Buchanan, E.H. Shortliffe (eds.) (1984): Rule-Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, Reading, MA.

A. Geoffrion (1987). An introduction to structured modeling. *Management Science*, 33, pp. 547-588.

J.W. Grzymała-Busse (1992): LERS - a System for Learning from Examples Based on Rough Sets, in R. Słowiński (ed.): *Intelligent Desicion Support. Handbook of Applications and Advances of the Rough Set Theory*, Kluwer Academic Publishers, Dordrecht, pp. 3-18.

D.D. Ignatavicius, K.A. Hausman (1995): *Clinical Pathways for Collaborative Practice*. W.B. Saunders, Philadelphia.

M. Makowski (2004). Structured modeling technology. *European Journal of Operational Research* (to appear).

M. Li (2004): *Application of the Bayesian Belief Network Model to Evaluate Variances in a Clinical Caremap: Radical Prostatectomy Case Study*, Master Thesis, University of Ottawa, Ottawa, Ontario, Canada.

Z. Pawlak (1982): Rough sets. *International Journal of Information & Computer Sciences*, 11, pp. 341-356.

B. Prędki, Sz. Wilk (1999): Rough Set Based Data Exploration using ROSE system, [in] Z.W. Raś, A.Skowron, (ed.) *Foundations of Intelligent Systems,* Lecture Notes in Artificial Intelligence vol. 1609, Springer Verlag, Berlin, pp. 172-180.

J. Stefanowski (2001): *Algorytmy indukcji reguł decyzyjnych w odkrywaniu wiedzy*. Seria rozprawy nr 361, Wydawnictwo Politechniki Poznańskiej, Poznań.

Appendix A – Radical prostatectomy caremap

THE OTTAWA HOSPITAL–CIVIC CAMPUS
CARE MAP
Radical Prostatectomy

Patient Name _____
Unique # _____

CRITICAL PATH

Patient Day # Date	Pre-Admit Unit	Same Day Admit	OR Day (PACU)	Post-op Day 1	Post-op Day 2	Post-op Day 3	Post-op Day 4	Clinic Visit #1	Clinic Visit #2
Consults	• Anaesthesia • PAU assessment			• Home Care					
Medications	• Review patient medications	• Heparin in OR • port IV medications as ordered • IV VCS at 30–120 h for all patients where IV medications are ordered	• Patient specific medications • Analgesia: Intrathecal Protocol • IV as ordered • Oxygen Titration Protocol • s.c. Heparin 5000 units q12h	• Patient specific medications • Analgesia: Intrathecal Protocol • IV as ordered • Oxygen Titration Protocol • s.c. Heparin 5000 units q12h • KSAD as ordered	• Patient specific medications • Analgesia: Intrathecal Protocol • D/C IV if drinking well • Oxygen Titration Protocol • s.c. Heparin 5000 units q12h	• Patient specific medications • Analgesia: Intrathecal Protocol • D/C Heparin	• Patient specific medications • Analgesia: Intrathecal Protocol		
Tests	• Type & screen • CBC • Na, K, Cl, glucose, creatinine • Chest x-ray • ECG • Urine R & M • C & S • If appropriate – sickle cell • If patients on Coumadin – INR/PT	• Patients with chronic renal failure – *Sodium, potassium chloride stat on arrival* – All diabetic patients – glucoscan • If patient on Coumadin – INR/PT *stat on arrival* • 2 autologous blood donor – CBC	• CBC 1 hour post-op		• CBC • Na, K, Cl, creatinine				
Assessments/ Treatments	• Vital signs • Measure for TED stockings • Patient/ surgery specific assessments	• Vital signs • SpO2 • Apply TED stockings • Complete pre-op checklist	• VS q4h • I & O q4h • SpO2 qshift • TED stockings • Monitor dressing, change if draining • Indwelling urethral catheter • Jackson Pratt • Systems assessment respiratory, GU, GI qshift & prn	• VS q4h • I & O q4h • SpO2 qshift • TED stockings • Monitor dressing change if draining • Indwelling urethral catheter • Systems assessment respiratory, GU, GI qshift & prn • Jackson Pratt	• VS qshift • I & O qshift • SpO2 qshift • TED stockings • Remove dressing—open to air • Indwelling urethral catheter • Systems assessment respiratory, GU, GI qshift & prn • Jackson Pratt	• VS qshift • Output qshift • SpO2 qshift • D/C TED stockings • Monitor incision • Indwelling urethral catheter • Systems assessment respiratory, GU, GI q shift & prn • Jackson Pratt • D/C drain if amount < 50 cc previous 24 Hrs – *apply dry dressing over site*	• VS • Monitor incision • Indwelling urethral catheter output • Remove dressing over drain site – *open to air* • Home with clips	• Remove clips • Assess wound, pack with saline soaked gauze if applicable	• Remove indwelling urethral catheter • Assess wound, wound care as applicable
Activity	• AAT	• AAT	• Bed rest • Dangle × 1	• Chair × 3 • Ambulate for short walk	• Progressive ambulation with minimal assistance	• AAT (independent)	• AAT	• AAT	• AAT
Nutrition/ Elimination		• NPO from 2400 hours	• NPO	• Sips clear fluids	• Full fluids to DAT	• DAT	• DAT	• DAT • Assess bowel function • Assess urine output (colour, consistency)	• DAT • Assess bowel function • Assess urine output (colour, consistency)
Patient Teaching	• Initiate Radical Prostatectomy Teaching Booklet • Patient specific medications for morning of surgery • Pre-op preparations: – *Urology bowel prep day before surgery* – Kegel exercises	• Reinforce pre-op teaching • Care of patient belongings	• Reinforce – *deep breathing & coughing* – *leg exercises* – *wound splinting* – *pain management* – *diet*	• Reinforce – *deep breathing & coughing* – *wound splinting* – *activity* – *pain management* – *diet* – *record oral intake*	• Reinforce – *activity* – *diet* • Review – *catheter care & drainage system*	• Review discharge instructions – *activity* – *pain management* – *wound care* – *catheter care and drainage system*	• Reinforce discharge instructions	• Reinforce – *catheter care and drainage system* – *wound care* • Discuss continence care	• Reinforce – *continence care* – *urine elimination* – *Kegel exercises* – *wound care* • Reference Prostate Cancer Support Group
Discharge Planning	• Discuss Care Map and expected length of stay • Initiate Surgical Portfolio Discharge Assessment & Planning Care Map				• Complete Surgical Portfolio Discharge Assessment & Planning Care Map	• Confirm discharge plans with patient/family • Prescription written prn • Follow-up appointment within 7 days at urologist's next clinic	• Complete Discharge Summary Section of Surgical Portfolio Discharge Assessment & Planning Care Map	• Modify Home Care referral if applicable • Follow-up appointment in one week at urologist's clinic	• Requisition for PSA and follow-up appointment
Patient progress corresponds with Care Map	**Nursing** D ☐ Yes ☐ No _____ RN	**Nursing** D ☐ Yes ☐ No _____ RN	**Nursing** A ☐ Yes ☐ No _____ RN P ☐ Yes ☐ No _____ RN	**Nursing** A ☐ Yes ☐ No _____ RN P ☐ Yes ☐ No _____ RN	**Nursing** A ☐ Yes ☐ No _____ RN P ☐ Yes ☐ No _____ RN	**Nursing** A ☐ Yes ☐ No _____ RN P ☐ Yes ☐ No _____ RN	**Nursing** A ☐ Yes ☐ No _____ RN P ☐ Yes ☐ No _____ RN	**Nursing** D ☐ Yes ☐ No _____ RN	**Nursing** D ☐ Yes ☐ No _____ RN

PAGE ONE

THE OTTAWA HOSPITAL–CIVIC CAMPUS

CARE MAP
Radical Prostatectomy

Patient Name _____

Unique # _____

THE OTTAWA HOSPITAL–CIVIC CAMPUS

CARE MAP
Radical Prostatectomy

PATIENT OUTCOMES

Legend

A = 0715 – 1915
D = 0715 – 1515
P = 1915 – 0715
NSAID = Non Steroidal Anti-Inflammatory Drug
PAU = Pre Admission Unit
PSA = Prostatic Specific Antigen

Patient Problem List

1) Impaired health maintenance related to knowledge deficit
2) Potential for pain
3) Potential for impaired respiratory function
4) Potential for alteration in elimination: constipation/urine
5) Potential for alteration in mobility
6) Potential for wound complications

Patient Specific Problems

_____

_____

_____

Addressograph

Patient Day #	Pre-Admission	Same Day Admit	OR Day	Post-op Day 1	Post-op Day 2	Post-op Day 3	Post-op Day 4	Clinic Visit #1	Clinic Visit #2
Date									
Patient Teaching	• Verbalizes understanding of events of pre-operative day • Verbalizes understanding of pre-operative instructions – *bowel prep day before surgery* • Specific concerns addressed	• Verbalizes understanding of events of operative day	• Demonstrates DB & C exercises • Demonstrates wound splinting • Compliant with activity	• Independent with DB & C • Independent wound splinting • Demonstrates understanding of recording oral intake • Demonstrates proper technique in moving in/out of bed • Verbalizes understanding of progressive activity • Verbalizes understanding of progression of diet	• Verbalizes understanding of catheter care and drainage system • Verbalizes understanding of progression of activity and diet	• Demonstrates ability to manage catheter care and drainage system • Verbalizes understanding of discharge plans and discharge care – *wound care* – *activity* – *pain*	• Demonstrates understanding of – *discharge instructions as per booklet* – *follow-up appointment*	• Verbalizes understanding of – *wound care* – *next clinic visit* – *continence care* – *catheter care and drainage systems*	• Verbalizes understanding of – *wound care* – *on going follow up with physician* – *continence care* – *urine elimination* – *kegel exercises* • Specific concerns addressed
Pain	• Verbalizes understanding of post-op plan for pain management		• Pain score: mild to moderate (< 3 at rest, < 7 with activity) • Verbalizes adequate pain control	• Pain score: mild to moderate (< 3 at rest, < 7 with activity) • Verbalizes adequate pain control • Demonstrates progressive ADL related to adequate pain control	• Request analgesia as needed • Pain score: mild to moderate (< 3 at rest, < 7 with activity) • Demonstrates progressive ADL related to adequate pain control	• Requests analgesia as needed • Demonstrates progressive ADL related to acequate pain control	• Patient discharged with adiquate pain control	• Adequate pain control	• Adequate pain control
Respiratory Function	• Verbalizes understanding of potential for respiratory complications and importance of preventative measures		• Respiratory rate, rhythm and effort are stable (RR >8, ≤20) at rest • SpO_2 within limits of titration protocol • Level of sedation: easily arousable • No new evidence of crackles, wheezes, bronchial breath sounds	• Respiratory rate, rhythm and effort are stable (RR >8, ≤20) at rest • SpO_2 within limits of titration protocol • Level of sedation: easily arousable • No new evidence of crackles, wheezes, bronchial breath sounds	• Respiratory rate, rhythm and effort are stable (RR >8, ≤20) at rest • SpO_2 within limits of titration protocol • Level of sedation: easily arousable • No new evidence of crackles, wheezes, bronchial breath sounds	• Respiratory rate, rhythm and effort are stable (RR >8, ≤20) at rest • SpO_2 within limits of titration protocol • Level of sedation: easily arousable • No new evicence of crackles, wheezes, bronchial breath sounds	• Patient discharged with no evidence of impaired respiratory function	• No evidence of impaired respiratory function	• No evidence of impaired respiratory function
Elimination			• Urine output adequate • No evidence of hematuria	• Bowel sounds present • Urine output adequate • No evidence of hematuria	• Bowel sounds present • Urine output adequate • No evidence of hematuria	• Bowel sounds present • Urine output adequate • No evidence of hematuria	• Patient discharged with normal bowel function • Urine output adequate • No evidence of hematuria	• Normal bowel function • Urine output adequate • No evidence of hematuria	• Normal bowel function
Mobility			• Dangles × 1 at bedside	• Up in chair × 3 with assistance • Ambulatory for short walk	• Ambulates in hall with minimal assistance	• Independent ambulation	• Patient discharged with ability to ambulate independently	• Independent ambulation	• Independent ambulation
Wound Complications					• Temp within normal range • No evidence of wound redness, swelling, drainage or dehiscence	• Temp within normal range • No evidence of wound redness, swelling, drainage or dehiscence	• Patient discharged with no clinical signs of wound complications • Able to identify signs of wound complications that should be reported to MD • Discharged home, Jackson Pratt	• No evidence of wound redness, swelling, drainage or dehiscence	• No evidence of wound redness, swelling, drainage or dehiscence
Patient progress corresponds with Care Map	**Nursing** **D** ☐ Yes ☐ No _____ RN	**Nursing** **D** ☐ Yes ☐ No _____ RN	**Nursing** **A** ☐ Yes ☐ No _____ RN **P** ☐ Yes ☐ No _____ RN	**Nursing** **A** ☐ Yes ☐ No _____ RN **P** ☐ Yes ☐ No _____ RN	**Nursing** **A** ☐ Yes ☐ No _____ RN **P** ☐ Yes ☐ No _____ RN	**Nursing** **A** ☐ Yes ☐ No _____ RN **P** ☐ Yes ☐ No _____ RN	**Nursing** **A** ☐ Yes ☐ No _____ RN **P** ☐ Yes ☐ No _____ RN	**Nursing** **D** ☐ Yes ☐ No _____ RN	**Nursing** **D** ☐ Yes ☐ No _____ RN

Appendix B – DTD definition for the XML file format for storing data.

```
<!DOCTYPE InputData [
<!ELEMENT InputData (Attributes,Cases,UniqueAttributes)>
<!ELEMENT Attributes (Attribute*)>
<!ELEMENT Attribute (Name,Day,Role)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Day (#PCDATA)>
<!ELEMENT Role (#PCDATA)>
<!ELEMENT Cases (Case*)>
<!ELEMENT Case (Day,Value)>
<!ELEMENT Day (#PCDATA)>
<!ELEMENT Value (#PCDATA)>
<!ELEMENT UniqueAttributes (Attribute*)>
<!ELEMENT Attribute (Day,Value)>
<!ATTLIST Attribute name PCDATA>
<!ELEMENT Day (#PCDATA)>
<!ELEMENT Value (#PCDATA)>
]>
```

Appendix C – DTD definition for the XML file format for storing decision rules.

```
<!DOCTYPE Rules [
<!ELEMENT Rules (Rule*)>
<!ATTLIST Rules information PCDATA>
<!ATTLIST Rules date PCDATA>
<!ELEMENT Rule
(Strength,Support,Confidence,RelativeStrength,Discrimination,Elementar
y*)>
<!ATTLIST Rule Day PCDATA>
<!ELEMENT Strength (#PCDATA)>
<!ELEMENT Support (#PCDATA)>
<!ELEMENT Confidence (#PCDATA)>
<!ELEMENT RelativeStrength (#PCDATA)>
<!ELEMENT Discrimination (#PCDATA)>
<!ELEMENT Elementary (Left,Relation,Right)>
<!ATTLIST Elementary Role PCDATA>
<!ELEMENT Left (#PCDATA)>
<!ELEMENT Relation (#PCDATA)>
<!ELEMENT Right (#PCDATA)>
]>
```