

Automatic generation of Formal Intervention Plans based on the SDA* representation model

Francis Real, David Riaño, John Bohada
Departamento de Ingeniería Informática y Matemáticas
Universidad Rovira i Virgili
francis.real , david.riano, john.bohada@urv.cat

Abstract

Clinical practice guidelines are important in the work of physicians. These guidelines are manually created by experts using their knowledge and experience. This work gives an approach to automatically develop the clinical guideline charts with the SDA representation model. In addition, this paper details an example of application of the methodology proposed with the treatment of Hypertension.*

1. Introduction

The US National Cancer Institute defines Clinical Practice Guidelines as documents developed to help healthcare professionals and patients make decisions about screening, presentation or treatment of a specific health condition.

In the K4CARE project (www.k4care.net) a Formal Intervention Plan (FIP) is a structure representing health care procedures to assist patients suffering from particular ailments or diseases. They contain indications to all the actors involved in the care process (i.e. healthcare professionals, patients and relatives) in order to provide the best coordinated and effective action plan.

Since Clinical Practice Guidelines are created by experts with the objective of improving the quality of patient care, reducing the variability of treatments and the decision making cost, they are the baseline to develop FIPs.

The components of a FIP are a small group of design elements that are combined to describe a global plan. Figure 1 shows the graphical representation of a FIP on hypertension.

Among the languages and systems to model clinical practice guidelines we stand out PROforma[1], Asbru[2], EON[3], PRODIGY-3[4], GLIF3[5], GUIDE[6] and SDA*[7].

The costs of these manually created FIPs are high, but the knowledge necessary to represent the structures is implicit in the medical databases. Here, the SDA* model is taken as the formalism to represent FIPs which are obtained from data on the treatment of representative samples of patients. In section 2 the SDA* model is briefly introduced. Section 3 contains the description of the new methodology to generate SDA* structures from raw data. The correctness of the approach is discussed in section 4 through the exposition of an experiment in the treatment of hypertension. Some conclusions are provided in section 5.

2. SDA* representation model

The representation model of FIPs that we use in our experiment is the SDA* (State-Decision-Action [7]). This theoretical model has been chosen because of its simplicity. The SDA* model has three types of variables: state variables, decision variables and action variables. **State variables** describe the condition of a patient in a concrete disease. **Decision**

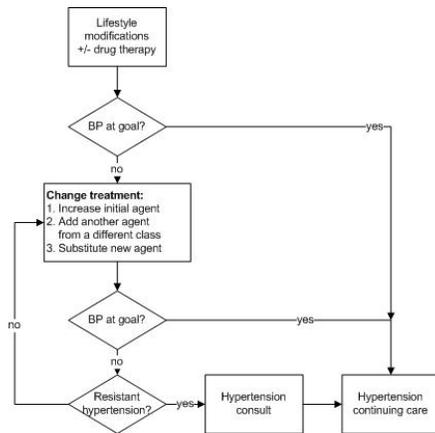


Figure 1: A part of the clinical algorithm "Hypertension diagnosis and treatment" from the Institute for Clinical Systems improvement (ICSI)

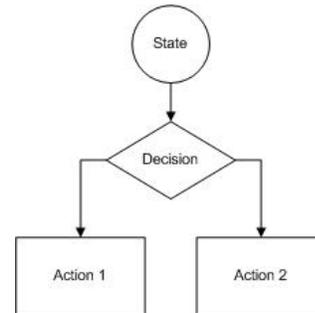


Figure 2: Basic combination of SDA* primitives

variables are those that allow to differentiate the particularities of a patient who requires aspects of treatment with respect to other patients who require other aspects of treatment. **Action variables** represent all of the involved individual medical actions in the treatment of the disease.

These variables are grouped into three primitives in the SDA* model: states, decisions and actions. **State primitives** of the SDA* model contain state variables and represent the different stages of the disease. They are the entry points of the FIP. **Decision primitives** are used to describe that we can follow different treatment procedures depending on some decision variables at the same stage of the disease. Therefore, a decision consists on a question with two or more possible answers that mark the possible alternatives in the FIP. **Action primitives** consist of groups of procedures, recommendations, prohibitions, suggestions, prescriptions, etc. that professionals have to follow in the treatment of the patient. A treatment can be defined by using a combination of SDA* primitives.

Figure 2 shows the *basic combination* of SDA* primitives in a state-decision-action sequence. Theoretically, a treatment should be composed of steps including SDA* primitives or basic combinations of SDA* primitives.

3. Automatic generation of FIPs

Automatic generation of FIPs starts with data about real patients structured in a *data matrix*. With this information we can generate a *decision tree* to decide on the treatment a patient should receive. Each one of the branches in the decision tree gives information to split patients into groups according to the treatment actions that they have to do. We use this information to find out sorts of patient.

Whenever we apply a treatment to each one of those sorts of patients, the patient condition may change. We use a *background knowledge* based in rules to represent these health changes. All these components (data matrix, decision tree induction, and background knowledge) are combined in a methodology to generate FIPs.

3.1. Data matrix

In our approach, the data matrix contains information about the patients visited and their treatments. Patients are described by means of *input* variables and treatments by means of *output* variables. So, input variables can refer to features on the patient current situation (e.g.

“Fever” or “High_Blood_Pressure”), the patient general situation (e.g. “Women” or “Penicillin_Allergy”) or information about patient antecedents (e.g. “Radiotherapy?” or “Antibiotics?”). Output variables represent the actions that professionals and patients have to follow in some phases of the treatment (e.g. “Apply_Radiotherapy” or “Take_Antibiotics”).

In the SDA* representation, input variables can be either decision or state variables and output variables are equivalent to action variables. Here, input and output variables are Boolean.

Input and output variables define the columns of the data matrix, while each row in the data matrix describes a visited patient (in terms of the input variables), and the treatment decisions taken in terms of the output variables. For example, the second row of the data matrix in figure 3 describes a patient that has the signs and symptoms I_1 and I_n . It also indicates that the patient is treated with action O_1 .

3.2. Decision trees to detect (S, A) structures

One of the steps of automatic generation of FIPs is to classify the patients in the data matrix according to the treatment they receive. For this purpose, each different combination of output variables in the matrix (i.e. different treatment) is provided a label, and the resulting data set is used to generate a decision tree. After that, each branch going from the root to a leaf of the tree is transformed into a structure (S, A) where S is the set of conditions on the input variables in the branch (i.e. state of the patient) and A is the treatment in the leaf of the branch that the output variables describe (i.e. action on the patient).

C4.5 [8] is an efficient machine learning algorithm to generate decision trees. It shares the approach of ID3 [9] that consists on repeatedly select the attribute that partitions the set of instances in a more convenient way. For this work we use the WEKA’s approach to the C4.5, the J.48 algorithm.

3.3. Using Background knowledge to introduce the consequences of treatment actions

It could happen that some medical knowledge was available informing about how the application of medical actions would affect the patient state. This knowledge would relate input and output variables by means of rules as the one introduced in figure 4. These rules describe how some output variables may affect some other input variables in the following way: if we have a set of input variables with their values and a set of output variables is applied then, the first set of input variables will be turned in a new set of input variables. For example, the rule in figure 4 states that if a patient has *fever*, and he takes *antibiotic_0106* and *rests*, the patient will evolve to a state in which he *does not have fever* and he is taking the antibiotic.

These rules have three components: an initial group of input variables (tag <Input_variables>), a group of output variables (tag <Output_variables>), and a set of input variables replacing the first group (tag <New_input_variables>).

	Input Vars				Output Vars			
	I_1	I_2	...	I_n	O_1	O_2	...	O_m
Pat. 1	0	0	...	1	0	0	...	0
Pat. 2	1	0	...	1	1	0	...	0

Pat. k	1	1	...	0	0	0	...	1

Figure 3: Example of data matrix

```

<rule>
  <Input_variables>
    <var name="FEVER" value="TRUE" \>
  <\Input_variables>
  <Output_variables>
    <var name=" ANTIBIOTIC_0106" value="TRUE" \>
    <var name="REST" value="TRUE" \>
  <\Output_variables>
  <New_input_variables>
    <var name="FEVER" value="FALSE" \>
    <var name=" ANTIBIOTIC_0106?" value="TRUE" \>
  <\New_input_variables>
</rule>

```

Figure 4: Example of medical knowledge data used by the system

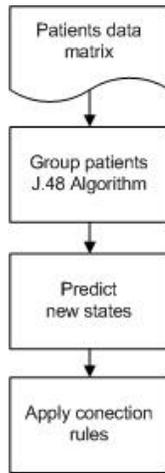


Figure 5: Schedule of the algorithm

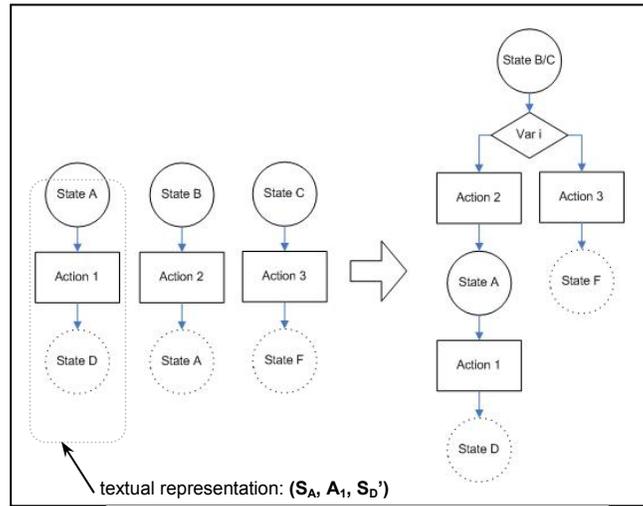


Figure 6: Example of connection rules

3.4. Methodology

The input data matrix and the background knowledge described in sections 3.1 and 3.3 are used to generate a FIP that represents a medical treatment. The methodology employed incorporates the J.48 algorithm in a sequence of steps that figure 5 depicts. Initially, the number of possible patient stereotypes for a disease can be very high, but the number of possible sets of different actions in a treatment is limited and normally it is precisely defined. Therefore, our first step in figure 5 is to group the patients in the data matrix according to their treatments. To reach this objective we use the J.48 algorithm to create a decision tree that will be used to calculate the (S, A) structures. The S components represent SDA* state primitives and the A components SDA* action primitives that are one-to-one connected as figure 6 shows at the left side with straight lines. These connections define SDA*State-Action structures that may need to be connected to the next possible SDA*State-Action in order to provide a treatment as a connected sequence of steps. This is achieved applying the background knowledge to each SDA*State-Action structure (see the second step in figure 5).

We say that a rule is applicable to a SDA*State-Action structure when all the input variables of the rule are contained the state of the structure, and all the output variables of the rule are contained in the action of the structure. The union of the new input variables of all the applicable rules of a SDA*State-Action structure defines a new predicted state that figure 6 represents at the left side with slashed lines.

The last step of the methodology in figure 5 is to connect pairs of SDA*State-Action structures until a final FIP is obtained. Given two SDA*State-Action structures in textual representation (S_1, A_1, S_1') and (S_2, A_2, S_2') (see figure 6), the conditions to connect them are:

- (1) If S_1 and S_2 differs in n or less variables (where n is a parameter of the methodology) then a state with the variables in $S_1 \cap S_2$ is connected to a decision primitive with two branches: one labeled $S_1 \setminus S_2$ and connected to A_1 , and the other one labeled $S_2 \setminus S_1$ and connected to A_2 .
- (2) If $S_1' = S_2$, then we connect the output of A_1 to S_2 .
- (3) If S_1' and S_2 differ in m or less variables (where m is a parameter of the methodology) then we can connect the output of A_1 to S_2 .

The whole process applies conditions (1) and (2) until no changes are possible. Then condition (3) is applied on the disconnected structures.

Input variables					Output variables		
A	First BP at goal?	X	Test var 1	1?	Lifestyle Modific.?	1!	Lifestyle Modific.
B	Last BP at goal?	Y	Test var 2	2?	Change Treatment?	2!	Change Treatment
C	Resistant Hyper.			3?	Hyper. Consult?	3!	Hyper. Consult
				4?	Hyper. Cont. Care?	4!	Hyper. Cont. Care

Table 1: Input and output variables

Figure 6 shows an example of this process in which *State B* and *State C* differ only in variable *var i*. At the end of the application of conditions (1), (2), and (3) the diagram on the right side of figure 6 is obtained.

4. Experiment

We apply this methodology to the diagnosis and treatment of *Hypertension* (see figure 1). We select the variables in *Table 1*. *A*, *B* and *C* are the variables that appear in the decision blocks of the hypertension algorithm. *X* and *Y* are control variables that do not appear in the hypertension algorithm, which are included to see if the methodology can select the proper variables and discard the wrong ones. Variables with numbers are the action blocks in the hypertension algorithm. If the action variable has a question mark (e.g. 1?) it represents an input variable with meaning “has this action been done?” while action variables with an exclamation mark (e.g. 1!) represent output variables with meaning “make this action!”.

For this experiment, a data matrix has been created with the following procedure: Boolean values are randomly assigned to the input variables, simulating a patient description *P* that is been matched with the actions that the hypertension algorithm proposes for *P*. This process is repeated to generate the *N* patients (i.e. rows) of the data matrix. With this process, we generated three data matrices with 1000, 500 and 100 random patient instances. Repeated occurrences were removed from the matrices that reduced to 177, 128 and 52 different random patient.

When we apply the J.48 algorithm to the different data matrices we obtain the results that tables 2a, 2b, and 2c display. The patients that share the same set of actions are grouped as the first columns of the tables indicate. Within each group each row describes a *SDA*State-Action* structure (e.g. in table 2a, the row of the action block **3** is a *SDA*State-Action* structure with state $\{A=0, B=0, C=1, 1?=1, 2?=1, 3?=0\}$ and action $\{3!\}$). We observe that since variables *X*, *Y* and *4?* are not required by the J.48 algorithm, they will not be part of the final FIP. In this experiment, we use a background knowledge in which each rule has as new input variables (tag <New_input_variables>) all the input variables of the rule (tag <Input_variables>) plus the output variables of the rule (tag <Output_variables>) as ? variables, with

	<i>A</i>	<i>B</i>	<i>C</i>	<i>X</i>	<i>Y</i>	<i>1?</i>	<i>2?</i>	<i>3?</i>	<i>4?</i>
1						0			
2	0					1	0		
	0	0	0			1	1		
3	0	0	1			1	1	0	
4	0	0	1			1	1	1	
	0	1				1	1		
	0	0	1			1	1	1	
	0	1	1			1	1		
	1							1	

(a)

(b)

(c)

Table 2: Results of apply J.48 decision tree algorithm to the patient data matrices with 177, 128 and 52 instances.

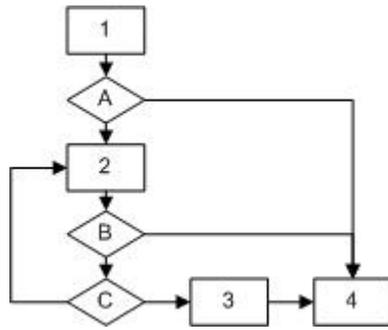


Figure 7: Results for 117 and 128 patients

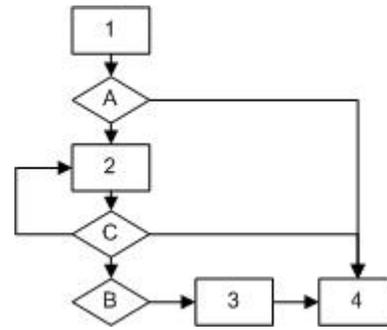


Figure 8: Results for 52 patients

value true. For example, the rule with input variables $A=0$, $I?=1$, and $2?=0$ and output variables $2!=1$ will have as new input variables $A=0$, $I?=1$, $2?=1$.

Figure 7 shows the resulting FIP after applying the methodology to the matrices of 177 and 128 patients. Despite the differences between the initial matrices and the different number of initial connections to the actions (rows in tables 2a and 2b), the final FIP is the same. Moreover, the structure (SDA* representation) is equivalent to the structure of the initial Hypertension algorithm in figure 1. Finally, figure 8 shows the FIP obtained with the third data matrix. The differences with respect to the original algorithm are caused by the reduced number of patients (52 out of 512 possible patients with 9 Boolean variables).

5. Conclusions

This work shows a way to automatically generate FIPs using medical data. The methodology can combine actions in the treatment and definitions of patient prototypes to create new FIPs. The experiment provided in section 4 confirms that it is possible to obtain similar structures to the manually created by experts if the sample of patient in the data matrix is representative of all the alternatives in the treatment of the disease under study. The quality of these structures depends on the amount of data.

6. References

- [1] J. Fox, N. John and A. Rahmzadeh, "Disseminating medical Knowledge: the PROforma approach". *Artificial Intelligence in Medicine*, 14:157-181. 1998.
- [2] Y. Shahar, S. Miksch, and P. Johnson. "The Asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines" *Artificial Intelligence in Medicine* (14): 29-51, 1998.
- [3] M.A. Musen, S.W. Tu, A.K. Das, and Y. Shahar. "EON: A component-based approach to automation of protocol-directed therapy". *Journal of the American Medical Informatics Association*. 3(6):367-388.
- [4] I.N. Purves, B. Sugden, N. Booth, M. Sowerby. "The PRODIGY project - the iterative development of the release one model". *Proceedings AMIA Symp.* 1999;:359-63.
- [5] M. Peleg, A.A. Boxwala, O. Ogunyemi et al. "GLIF3: The Evolution of a Guideline Representation Format". *Proceedings AMIA Annual Symposium*, 2000.
- [6] L. Dazzi, C. Fassino, R. Saracco, S. Quaglini and M. Stefanelli. "A Patient Workflow Management System Built on Guidelines". *AMIA 1997 Annual Fall Symposium*.
- [7] D. Riaño. "The SDA Model v1.0: The set of theory approach", Tech. Report DEIM-RT-07-001, URV, 2007.
- [8] J.R. Quinlan. "C4.5: Programs for Machine Learning". Morgan Kaufmann Publishers, Inc., 1993.
- [9] J.R. Quinlan. "Introduction of decision trees". *Machine Learning*, 1(1):81-106, 1986.

7. Acknowledgments

This work has been possible thanks to the support of the K4CARE project (IST-2004-026968) and the HYGIA project (TIN2006-15453-c04).