Determination of Guidelines Complience: Comparison of Clinical Guidelines with the Patient's Record

Arnošt Veselý[1,2], **Jana Zvárová**[2]

[1] Department of Information Engineering, Czech University of Life Sciences, Prague, Czech Republic

[2] Department of Medical Informatics, Institute of Computer Science AS CR, Prague, Czech Republic

Summary

Many clinical guidelines were elaborated to improve quality of medical care and to achieve standardization of patient's treatment. Originally clinical guidelines are written in everyday language and then they are converted into formal model that can be implemented and processed by computer. If all relevant patient's treatment data are stored in patient's Electronic Health Record, the guidelines formal model may be, in principle, compared with patient's data to determine, if the patient was treated according to the recommended clinical practice. In this article we present an algorithm that enables to compare patient's data record with EGLIF (Enhanced GLIF) model. EGLIF is a simple enhancement of the standard GLIF model and it was devised to render the comparison more transparent and more convenient. Comparing algorithm is proposed for GLIF models with unambiguous decision steps and for patient's data records containing all relevant patient's treatment information. Its modification for arbitrary decision steps can be easily done. However, comparing GLIF or EGLIF model with incomplete patient's data record is more difficult issue. Some suggestions how to tackle this problem are discussed in the conclusion.

Keywords

Clinical guidelines, GLIF model, Electronic Health Record, Reminder facility, Execution engine algorithm

Correspondence to:

Arnošt Veselý
Department of Information Engineering
Czech University of Life Sciences
Address: Kamýcká 129, Prague 6, Czech Republic
E–mail: vesely@pef.czu.cz

EJBI 2012; 8(1):16–28
received: December 12, 2011
accepted: March 12, 2012
published: June 15, 2012

1 Introduction

Clinical guidelines (CG) contain a set of care decisions to assist the physician with patient care decisions about appropriate diagnostic, therapeutic, or other clinical procedures. They are intended to ensure high quality clinical practice [1]. CG are developed as textual recommendations by groups of medical experts on topics selected by a local scientific authority, e.g. expert medical society or by a national health institution. Usually their text focuses on the specific group of physicians or health professionals. Several international organizations create and maintain web repositories with guidelines in different domains [2, 3]. In the Czech Republic the Catalog of Clinical Guidelines as the web repository was created [4, 5].

The development of clinical guidelines is a quite expensive process. First, paper-based guidelines have to be developed. Second, the paper-based guidelines have to be translated into computer based guidelines representation language. This process is described in general in [6, 7, 8, 9] and [10, 11].

More groups have translated paper-based guidelines into computer readable format. Good literature overview of used methods and achieved results can be found in [12]. Here we mention only some of them. The paper [13] describes successfully implemented two cardiovascular guidelines and hypercholesterolemia guidelines into computer system. Guidelines were represented in GLIF format and connected with the Electronic Health Record. GLIF is a tool that was developed specially for formalization of medical guidelines. Alternatively, notions and tools developed in the field of business process modeling also proved to be useful. The paper [14] compared guidelines development with business process modeling and described some of their strengths and weaknesses. Medical guidelines need to be developed by experts from different fields. The papers [15] and [16] designed parallel guidelines development strategy, in which a multidisciplinary

group cooperated in creating both textual and computer-interpretable guidelines. This strategy of parallel guidelines development and formalization appeared to be successful. Parallel development gave the opportunity to eliminate vague concepts or errors already in the early stage of their development.

Overviews of computer-interpretable formalisms and modeling approaches were presented in [1, 17] and [18].

Arezzo for representing guidelines uses PROforma language [19, 20]. Arezzo tool consists of three parts: Composer, Tester and Performer. The Performer inference engine runs the guidelines taking into account the patient data stored in the health care database.

DeGel (Digital Electronic Guidelines Library) is a web based modular and distributed architecture, which facilitate conversion of a text form of guidelines into Asbru representation language [21].

GLARE (Guidelines Acquisition, Representation and Execution) has graph-based representation [22, 23]. The graph nodes represent atomic actions of four kinds: queries that allow to input information into the system, work actions that represent actions to be carry out, decision actions that represent selection among alternative actions according to the set of conditions and conclusions that allow to describe outputs of decisions.

NewGuide framework for modeling clinical guidelines [24, 25] uses a representing language GUIDE, which is based on Petri nets [26]. It allows to model concurrent processes and temporal data.

SAGE (Standards-based Sharable Active Guidelines Enviroment) was created in collaboration among several research groups in the United States [27]. The main goal was to encode guidelines using some standard representation to facilitate its deployment in different clinical information systems. Guidelines representation is based on a set of Protégé classes and plug-ins. Medical care plans are specified by activity graphs that consist of context nodes that specify clinical setting and relevant patient's attributes, decision nodes, action nodes and routing nodes that are used for branching and synchronization.

HeCaSe2 (Health Care Services release 2) is an agent-based platform [28]. There is not any central control. Agents act independently using their own knowledge and data and perform different tasks. Guidelines Agent performs all actions related to the clinical guidelines. Clinical guidelines are represented using the PROforma representation language [20]. Medical terms use UMLS terminology and they are stored in ontology.

In this article we are using Guidelines Interchange Format (GLIF). GLIF is a result of collaboration among different institutions and universities in the United States. The description of its version 2.0 (GLIF2) may be found in [29] and description of the newer version 3.0 (GLIF3) in [30]. Guidelines Execution Engine (GLEE) that is a tool for execution of guidelines encoded in GLIF3 format is described in [31].

GLIF specifies a process-oriented model for guidelines representation. It can be represented in a form of oriented graph. The nodes of the graph are guidelines steps and edges represent continuation from one step to the other one. Guidelines steps are of a different kind. Guidelines step might be: action step, decision step, branch step, synchronization step and patient state step.

Action steps specify clinical actions that are to be carried out. It can be an application of some therapy, carrying out some examination or measurement etc. Action step may also name sub-guidelines that provide greater detail for the action.

Decision steps are used for conditional branching. A decision step specifies criteria for each possible alternative decision.

Branch and Synchronization steps enable parallelism in the model. The guidelines steps that follow the branch step and that are on different branches can be carried out concurrently or in an arbitrary sequence. The branches with root in the branch step eventually converge in the synchronization step, where they are synchronized. It means that the actions that follow the synchronization step could not be carried out unless the synchronization condition is fulfilled. A simple synchronization condition, for example, might require that all actions specified on the branches between the branch step and the synchronization step must have been carried out.

Patient state step names the current state of the patient.

There is a lot of benefits that clinical guidelines may provide. The most evident and the most important are the following.

1. CG can improve the quality of clinical decisions, since CG help physicians to use the clinical knowledge in the appropriate patient clinical state.
2. CG can be effectively used in teaching, since they support rapid dissemination of updates and changes.
3. Health care professionals can use CG for comparing health care standards in different institutions.
4. If all relevant information is stored in patient's Electronic Health Record (EHR), then it is possible to check if the applied treatment procedure complies with recommended treatment standards.

In our paper we focus on acquiring the benefit mentioned in the item 4 above. The first ideas how to compare patient's data and formalized guidelines we examined in [32, 33, 34] and [35]. Here we continue further and we propose algorithm, which is capable of doing it. We assume that all relevant information about the patient's treatment is stored in the patient's Electronic Health Record (EHR). Our task is to contrive a method how to compare patient's data in EHR with medical treatment standards described in clinical guidelines and check if the data in EHR are in compliance with them.

The comparison may be ex post or on-line. In the case of ex post comparison we have at our disposal patient's record from a long time period and we would like to know ex post if the patient was treated according to the

appropriate standard described in CG. On line comparison means that patient's data record is compared with the standard each time when it is updated with a new data item. An on-line remainder system, which warns the physician if his/her action does not comply with the treatment standard, might be based on such on-line comparison.

The algorithm we are proposing in this paper assumes that the following two conditions are fulfilled.

1. The EHR record comprises all relevant information, i.e. all physician actions during patient's treatment are recorded.

2. Each guidelines contain decision points, in which the decision how to continue must be taken. According to the patient's state, which is determined by the values of already examined parameters, some alternatives are recommended and some are not. We assume that recommendation is unequivocal. It means that according to CG in each decision point one and only one alternative must be chosen. Such guidelines we call strict. The designed comparison algorithm generates error message if a physician does not follow the uniquely determined way. Non- strict guidelines usually in a decision point recommend more than one alternative. Comparing algorithm may be adapted to fit this case as well. How to do it we discuss in the conclusion.

In patient's treatment important recommendations concerning the time intervals between two actions or between some action and its repetition are often given. For example some examination must be repeated at least or at most after 1 week, 2 months etc. In GLIF model a condition that interval between two consecutive actions must fulfill can be represented using decision node. However, it is more illustrative and for formulation of our comparing algorithm more convenient, to represent these time conditions with a new type node called Time node.

The paper has the following structure. In the paragraph 2 the principles underlying the comparison of patient's data record and guidelines model are briefly sketched and a rough description of comparing algorithm is given. Then there follows the definition of enhanced GLIF model (EGLIF). The enhancement was necessary so that the comparing algorithm could have been established. In paragraph 3 the comparing algorithm is defined and its behavior is demonstrated on several simple examples. In the conclusion paragraph the possible generalization of the algorithm for non-strict guidelines models and for incomplete patient's data records is discussed.

2 Methods

The objective of this paper is to present an algorithm, which could be used for comparing patient's clinical data with formalized guidelines that prescribe the way the patient should be treated. We suppose that during patient's visits physician examines state of patient's health and prescribes therapies. During examination physician is looking for symptoms or carries out examinations of physiological quantities. We assume that each examination results in determination of a value of some patient's physiological parameter or symptom P in a certain time. The result we will write in the form

$$P(time) = value.$$

For example parameter SBP (systolic blood pressure) has been measured at time t and its value has been 145. Then the result is written as $SBP(t) = 145$. We suppose that also application of some therapy may be written in this way. Then P denotes the used therapy and with $value = 1$ we denote the fact that the therapy was applied. For example $Diet(t) = 1$ means that the therapy $Diet$ at time t has been prescribed. Moreover, by means of the parameter $value$ the further specification of the therapy might be given. For example $Penicilin(t) = Daily - 2mg$ might mean that at time t the $Penicilin$ has been prescribed and that the prescribed dose has been 2mg daily.

We assume that patient's clinical data are stored into patient's EHR. We do not stipulate how the data format of the health record ought to look like, but we assume that this record can be converted into the following sequence of performed examinations and therapy prescriptions (further called data sequence)

$$\mathbf{S} = \{P_1(t_1) = c_1, \ldots, P_n(t_n) = c_n\}, t_1 < \ldots < t_n,$$

where $P_i(t_i)$ denotes the value of the parameter P_i at time point t_i. To simplify notation we suppose that time scale consists of days and that time is written in the date format $day.month.year$, e.g $t = 1.1.02$ or $SBP(1.1.01) = 150$ etc. Of course, in real applications the time scale will be more detailed. For example, time t might be system time.

Table 1: The guidelines data model of the small guidelines from Example 1.

action	**parameter** P	**value type**
Measurement of systolic blood pressure	SBP	numeric
Measurement of diastolic blood pressure	DBP	numeric
High density cholesterol test	HDL	numeric
Low density cholesterol test	LDL	numeric
Prescription of diet regime	$Diet$	Boolean
Medication prescription	$Medication$	Boolean

All physiological parameters and therapies occurring in the clinical guidelines must be specified. It could be done using guidelines data model. The guidelines data model should contain the list of all possible examinations and therapies together with the description of their possible values. Typical parameter value types will be Boolean,

nominal and numerical. Example of a very simple guidelines data model that will be used in further examples is given in the Table 1.

As the formalized guidelines will be compared with data from the patient's health record an existing data model of health record data is assumed. That model must comprise the set of all parameters $\mathbf{P}_G$ occurring in the formalized guidelines model.

To be able to compare the patient's health record with the guidelines, the guidelines must be formalized and coded into computer readable format. For this purpose we use enhanced GLIF model (EGLIF model). As we have already said, we assume that patient's health record can be converted into data sequence **S** described above. The comparison is carried out with comparative algorithm CA specially designed for this purpose (see Fig. 1). The CA algorithm consequently deletes data items from the sequence **S** generated by the EHR system and compares them with the coded EGLIF model. If some data item is not in compliance with the EGLIF model, the CA algorithm warns the user. The EGLIF and CA algorithm are described in detail further. Here we give only their rough description.

EGLIF model is an oriented graph with nodes of different type. To the nodes there are assigned parameters and conditions. The node parameters have nothing in common with patient's parameters mentioned above. The most important node parameters are the parameter *next* that defines EGLIF graph structure and the parameter *token* that enables to follow passing through the model. Some types of nodes have parameters for storing tokens. We say that these nodes are able to store or to catch tokens. When the comparison starts only one token placed in the node $START$ exists in the EGLIF model. During comparison tokens are moving along graph branches among nodes that can store them. The CA algorithm subsequently deletes items $P(t) = c$ from data sequence **S** and compares them with those action steps in EGLIF model that have tokens. Each action node has three main parameters $action, result$ and $time$. The parameter $action$ determines the prescribed action. Its value is compared with P of the current data item. If they are identical, the result c of the current action $P(t)$ is written into the node parameter $result$ and its time t is written into the node parameter $time$. Then the node's token is handed over to the node, which is the nearest node on the same graph branch and can store it. The token passing through a branch node BRN is multiplied. If the branch node has n outgoing branches, then n tokens stem from the passing through token. Newly created tokens continue along different branches. In the synchronization node $SYNC$ with n inputs the incoming tokens are stored in the n token parameters $token_1, \ldots, token_n$. As soon as the synchronization condition of a $SYNC$ node is satisfied, one token comes out of the synchronization node and at the same time all tokens residing in the same $BRN - SYN$ subgraph are removed. The comparative algorithm (CA) is described more rigorously in the §3. To be able to formulate this algorithm we must at first give detailed description of the enhanced GLIF model.

2.1 Description of the Enhanced GLIF Model (EGLIF Model)

The EGLIF model is an oriented graph with nodes of types $A, D, BRN, SYN, TIM, START, STOP, GLF, ERROR$. Nodes of the same type are distinguished using indexes, e.g. A_1, A_2, etc. To each node one or more parameters or conditions are assigned. The parameter *par* or the condition *cond* of the node N will be denoted $N(par)$ or $N(cond)$ respectively, e.g. $A_1(result)$ or $TIM_2(\beta)$. The parameter *next* contains the pointer to the following node and the parameter *token* stores tokens ($token = 1$ indicates that the node contains token and $token = 0$ indicates that it does not).

The EGLIF node types are the following.

Action node $A(token, action, result, time, ref, next)$

Action node represents an action. The kind of the action is specified with parameter *action*. The result of the action is written into parameter *result* and the time when the action has been carried out is written into parameter *time*. Parameter ref contains a pointer to some time node or it contains the null pointer NULL. If ref value is not NULL, then parameter ref points to the time node, whose condition β must be checked.

Decision node $D((\alpha_1, next_1), \ldots, (\alpha_n, next_n))$

Decision node represents the decision that should be done on the base of evaluation of conditions $\alpha_1, \ldots, \alpha_n$, defined for the alternative branches. We assume that conditions are so called strict-in conditions. It means that if α_i condition is fulfilled, then the i-th branch must be chosen. Moreover, we assume that one and only one condition is fulfilled, i.e. we assume that for each decision node the following expressions hold

$$\alpha_1 \vee \ldots \vee \alpha_n = \mathbf{t}(\text{true}),$$

$$\alpha_i \wedge \alpha_j = \mathbf{f}(\text{false}), \text{ for all } i, j = 1, \ldots n \text{ and } i \neq j.$$

Conditions $\alpha_1, \ldots, \alpha_n$ are made up of parameters of action steps by means of basic relational and logical operators. For example an condition might be

$$(A_1(result) > 10) \wedge (A_2(result) < 100).$$

Branch node $BRN(next_1, \ldots, next_n)$

Branch node introduces parallelism into the model. From branch node one may continue taking any branch. Branches can be followed simultaneously, however, sequence of steps on each branch must be retained.

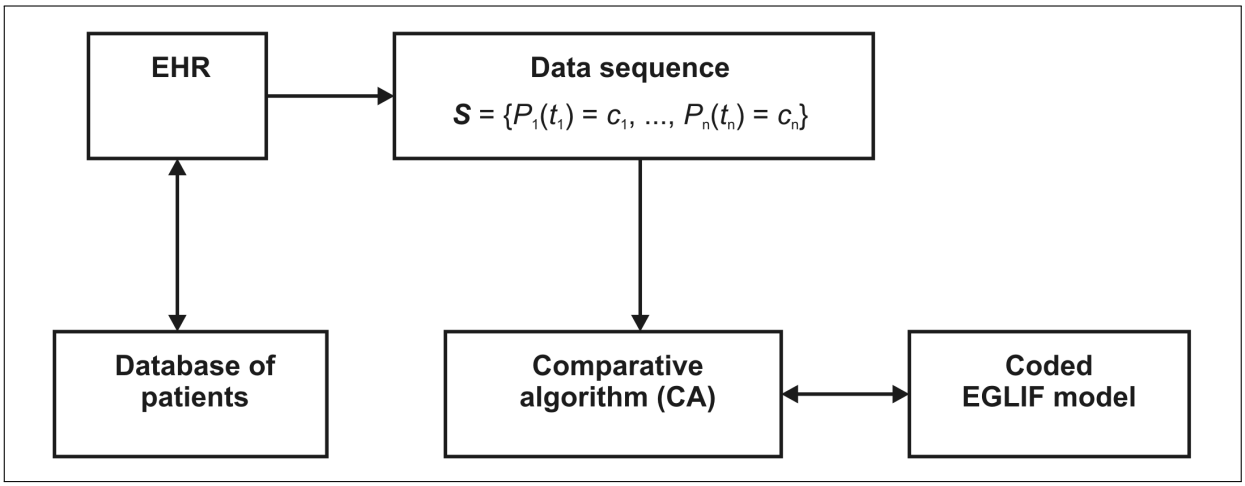

Figure 1: Comparing patient's health record with coded guidelines.

Synchronization node $SYN(token_1, \ldots, token_n, \alpha, \beta, time, next)$

Synchronization node connects and synchronizes branches. If the synchronization node SYN connects n branches, it has n inputs represented with n pointers $SYN(1), \ldots, SYN(n)$.

If a token comes from the i-th branch, then it is stored into the i-th token parameter $token_i$. One can pass through the synchronization node only if the condition α is satisfied. Condition α is a propositional formula made up of the parameters $token_1, \ldots, token_n$. For example,

$$\alpha = (token_1 \wedge token_2) \vee token_3.$$

Whenever a token is stored into the node, the time t taken from the last deleted data item $P(t) = c$ is written into the node parameter $time$. Condition β is the time condition that must be fulfilled for the time parameter value of all actions that take place in the subgraph $BRN - SYN$. In the definition of the β there occurs key word $atime$, whose meaning is obvious from the following example.

Assume the following condition

$$\beta = ((atime - A(time) \leq year).$$

This condition posits that if some node B is contained in the subgraph $BRN - SYN$ and $A(time)$ is the time parameter of a node A, then the condition $(B(time) - A(time)) \leq year$ must hold.

State node $STATE(name, next)$ denotes the patient's current state with string $name$.

Time node $TIM(time, \beta, next)$

Time node sets time limits of the next action. When a token passes a time node TIM, the current time is written down into the parameter $TIM(time)$. Condition β is a time condition that must be valid for the time parameter of the following action. In the definition of the condition β the time parameter of the following action is denoted $ftime$.

For example

$$\beta = ((ftime - TIM(time)) \leq year).$$

Start node represents the starting point

$$START(token, next).$$

Stop node represents the end

$$STOP(token).$$

Error node represents stop after error

$$ERROR(token, text).$$

Call node represents jump into another EGLIF model

$$GLF(name, next).$$

Definition of EGLIF model

EGLIF model is a set of above described nodes that by means of pointers (stored in their parameters $next$) constitute connected oriented graph, if the following conditions $C_1 - C_4$ are satisfied.

- C_1 The set contains just one $START$ node.
- C_2 For each BRN node there exists one SYN node, in which all branches starting in BRN node end. A subgraph of EGLIF model that starts with node BRN and ends with node SYN is called $BRN - SYN$ subgraph of EGLIF model.
- C_3 If G_1 and G_2 are two $BRN - SYN$ subgraphs, than only one of the following assertions can hold:

a) $G_1 \subset G_2$ (i.e. every G_1 node is also G_2 node)

b) $G_2 \subset G_1$

c) $G_1 \cap G_2 = \emptyset$ (i.e. G_1 a G_2 have no common node).

C_4 Topology of the graph is such that during token hand over, the token passes through at most one node TIM and it passes this node at most once.

Example 1

Small guidelines for heart failure prevention

When a patient comes for a visit, his/her physician examines patient's blood pressure parameters SBP, DBP and let to determine his cholesterol parameters LDL, HDL in laboratory.

1. If blood pressure is not normal, i.e. if the condition

$$\alpha = (SBP < 145) \wedge (DBP < 90)$$

is not satisfied, the physician prescribes diet and invites the patient for repeated examination after 1-2 months:

   (a) If patient's blood pressure is not normal again, the physician prescribes medicament treatment.

   (b) If patient's blood pressure is normal, the physician evaluate patient's risk index

$$i_R = (LDL - HDL)/HDL.$$

   If the risk index is small ($i_R < 4.2$), the patient is invited for the next examination not later than after a year. If the risk index is greater than 4.2, the patient is invited not later than after half a year.

2. If blood pressure is normal, i.e. condition α is satisfied, physician evaluates patient's risk index i_R. If the risk index is small ($i_R < 4.2$), the patient is invited for the next examination not later than after a year. If the risk index is greater than 4.2, the patient is invited not later than after half a year.

The EGLIF graph model of Small guideline for heart failure prevention is given in Fig. 2. Its coded form is given in Fig. 3.

3 Results

In this paragraph we present an algorithm for comparing of a strict guidelines EGLIF model with a complete patient's data record. We start with an example.

Example 2

We assume that the physician stores values of all examined patient's parameters (HDL, LDL, SBP, DBP) and all prescribed therapies $(Diet, Medication)$ into patient's health record. We assume further that patient's data stored in EHR can be output in the form of the data sequence described above. Let us suppose that from EHR we have got the following data sequences $\mathbf{S_A}, \mathbf{S_B}, \mathbf{S_C}, \mathbf{S_D}$ for 4 patients A, B, C and D.

$\mathbf{S_A} = \{SBP(1.1.01) = 150, DBP(1.1.01) = 85, HDL(2.1.01) = 1,$
$LDL(2.1.01) = 6, Diet(2.1.01) = 1, DBP(10.2.01) = 85,$
$SBP(10.2.01) = 140, SBP(1.5.01) = 130, DBP(1.5.01) = 85,$
$HDL(2.5.01) = 1, LDL(2.5.01) = 5, SBP(1.4.02) = 130,$
$DBP(1.4.02) = 90, LDL(2.4.01) = 7, HDL(2.4.01) = 2\}$

$\mathbf{S_B} = \{SBP(1.1.01) = 150, DBP(1.1.01) = 85, HDL(2.1.01) = 1,$
$LDL(2.1.01) = 6, DBP(10.2.01) = 85, SBP(10.2.01) = 140,$
$SBP(1.5.01) = 130, DBP(1.5.01) = 85, HDL(2.5.01) = 1,$
$LDL(2.5.01) = 5, SBP(1.4.02) = 130, DBP(1.4.02) = 90,$
$LDL(2.4.01) = 7, HDL(2.4.01) = 2\}$

$\mathbf{S_C} = \{SBP(1.1.01) = 150, DBP(1.1.01) = 85, HDL(2.1.01) = 1,$
$LDL(2.1.01) = 6, Diet(2.1.01) = 1, DBP(1.4.01) = 85,$
$SBP(1.4.01) = 140, SBP(1.5.01) = 130, DBP(1.5.01) = 85,$
$HDL(2.5.01) = 1, LDL(2.5.01) = 5, SBP(1.4.02) = 130,$
$DBP(1.4.02) = 90, LDL(2.4.01) = 7, HDL(2.4.01) = 2\}$

$\mathbf{S_D} = \{SBP(1.1.01) = 150, DBP(1.1.01) = 85, HDL(2.1.01) = 1,$
$LDL(2.1.01) = 6, Diet(2.1.01) = 1, DBP(10.2.01) = 85,$
$SBP(10.2.01) = 140, SBP(1.5.01) = 130, DBP(1.5.01) = 85,$
$HDL(2.5.01) = 1, LDL(2.5.01) = 5.5, SBP(1.4.02) = 130,$
$DBP(1.4.02) = 90, LDL(2.4.01) = 7, HDL(2.4.01) = 2\}$

If the patient's health record is complete we can compare generated data sequence with the guidelines and determine if the patient has been treated according to it. Let us compare data sequences $\mathbf{S_A}, \mathbf{S_B}, \mathbf{S_C}$ and $\mathbf{S_D}$ with the Small guidelines for heart failure prevention described in Example 1.

Comparing $\mathbf{S_A}$ with the guidelines we see that treatment of the patient A complies with the guidelines.

Comparing $\mathbf{S_B}$ with the guidelines we see that treatment of the patient B does not comply with guidelines. The reason is that at the first visit patient's blood pressure was not normal. Therefore, the physician should have prescribed diet, but the diet item is missing in the data sequence $\mathbf{S_B}$.

The treatment of the patient C does not comply with guidelines as well, because at the visit 1.1.01 patient's blood pressure was not normal and therefore the patient should have come for repeated visit not later than after 2 months. Nevertheless, he came later as we can see from the data sequence item $DBP(1.4.01) = 85$.

We can easily see that treatment of the patient D does not comply with guidelines as well. As $HDL(2.5.01) = 1$ and $LDL(2.5.01) = 5.5$, the risk index during patient's visit 2.5.01 had value $i_R = 4.5$. Hence patient's following visit should have been sooner than after half a year. But

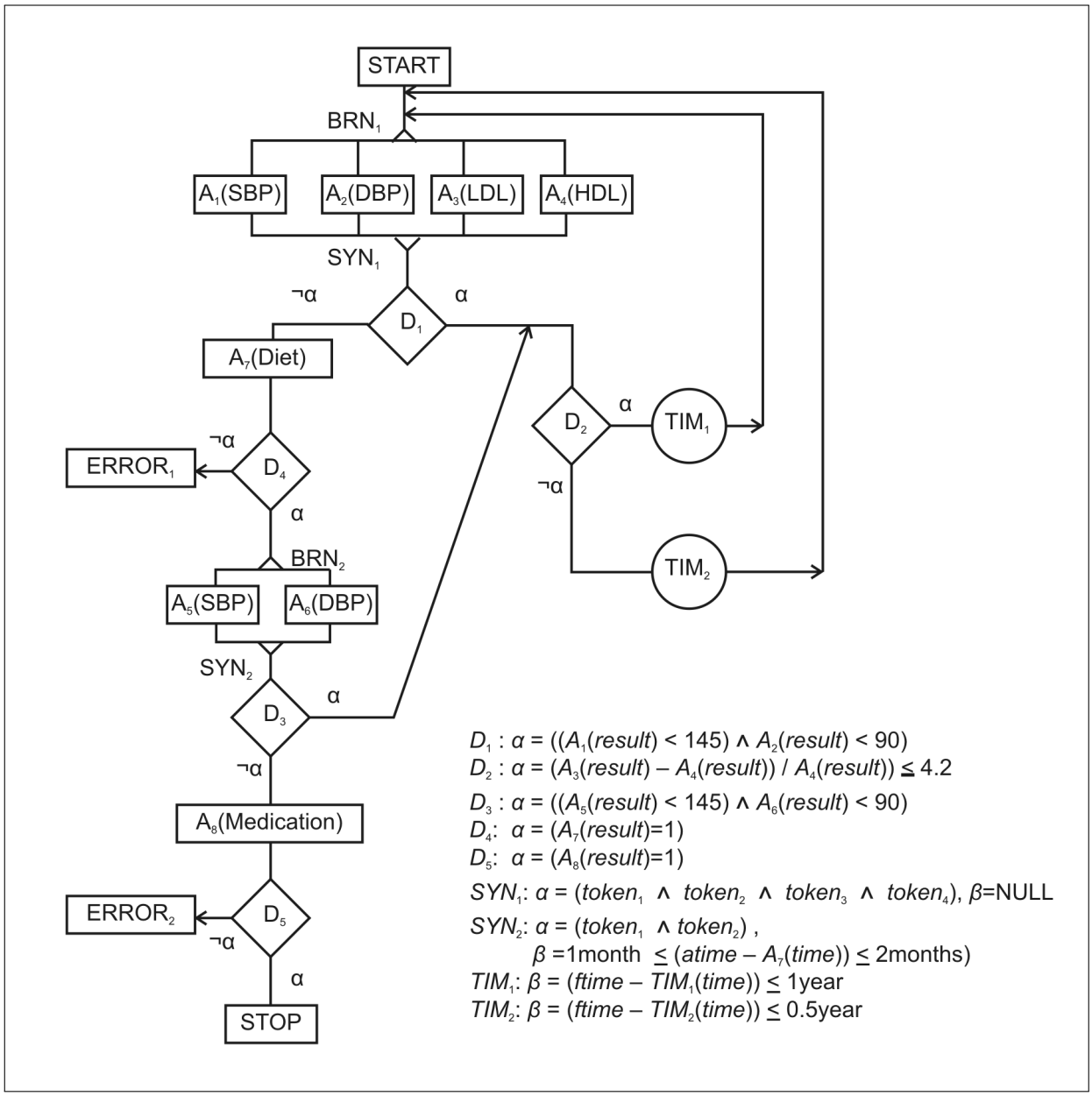

Figure 2: EGLIF graph model of the Small guidelines for heart failure prevention from Example 1. For better readability the value of parameter *action* of each action node is put into brackets and written after node name.

his next visit was 1.4.02 as we can see from the data item $SBP(1.4.02) = 130$.

Algorithm for comparing EGLIF with data sequence (algorithm CA)

Comparing a data sequence with clinical guidelines is a time consuming and tiring process prone to errors. Therefore, having at hand a possibility to do it using computer would be very appealing. For comparing we need an algorithm that would be able to compare a given data sequence with a coded EGLIF model and answer the question if the physician followed the recommended treatment specified in the guidelines or did not. In the following we describe an algorithm that is able to give the answer.

In the §2 we have already given a very rough description of the algorithm to sketch out the principles on which it is based on. Here we will at first shortly remind the main features of the algorithm and then we will describe it in full detail.

The algorithm compares an EGLIF model and a data sequence $\mathbf{S} = \{P_1(t_1) = c_1, \ldots, P_n(t_n) = c_n\}$. The algorithm subsequently deletes the elements from the data sequence. Let us assume that in the i-th step of the algorithm the item $P_i(t_i) = c_i$ of the data sequence is processed. Algorithm will find out all action nodes that have token and whose parameter *action* has value P_i. In each found node the algorithm will write t_i into its parameter *time* and c_i into its parameter *result*. After it the algorithm will propagate tokens from the found *action* nodes

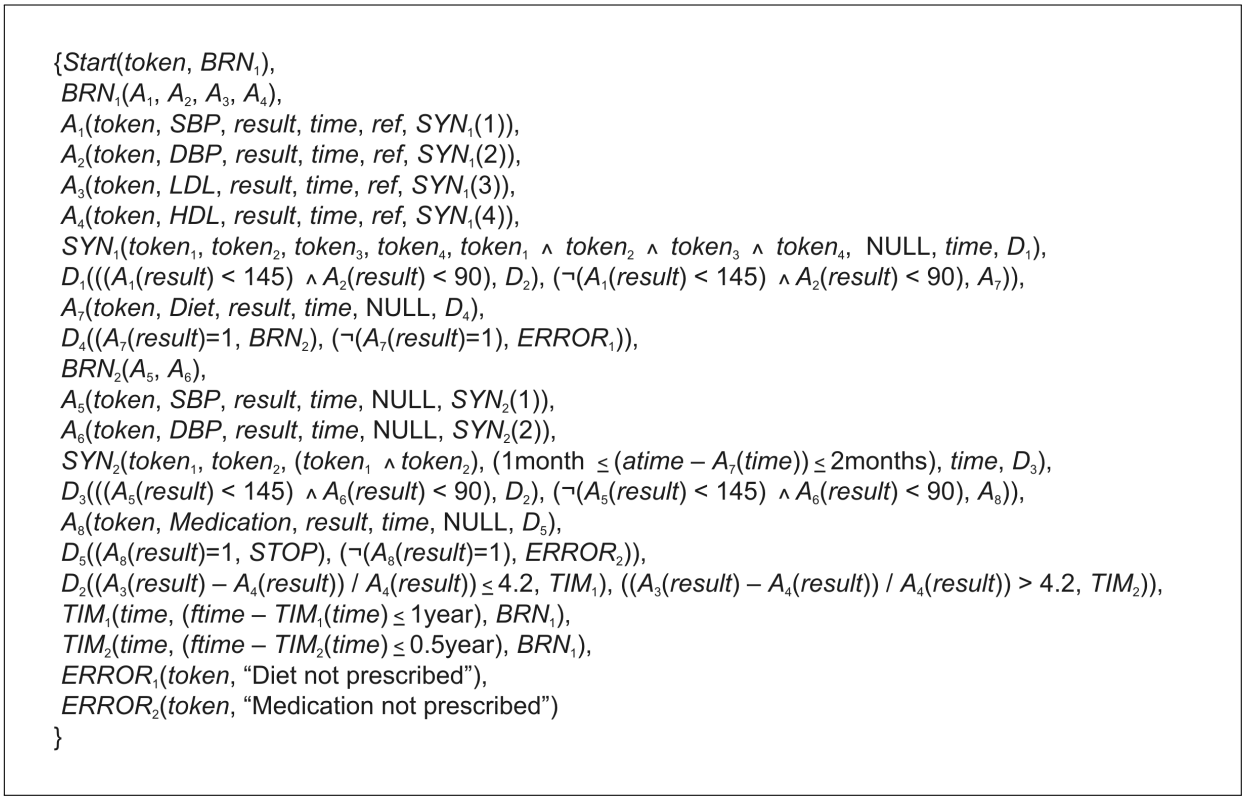

Figure 3: Coded EGLIF model of Small guideline for heart failure prevention.

to the following nodes capable of storing tokens. If no node with properties described above has been found, the error is generated and the comparison ends unsuccessfully.

Definition of the CA algorithm

Definition of the CA algorithm is given in two parts. The run of the algorithm is described in the first part. However, in the description the notion of token propagation is used that also must be precisely specified. The specification of the token propagation is given in the second part.

Part 1

1. In the 0-th algorithm step initialize node parameters:
   (a) In all nodes set parameters $time = 0$, $result = 0$ and $ref = NULL$.
   (b) In all nodes except $START$ set parameter $token = 0$.
   (c) Set $START(token) = 1$ and carry out its propagation from the $START$ node.
2. In the n-th algorithm step delete subsequently from the beginning of the data sequence $\mathbf{S}$ its members $P(t) = c$ until $P \in \mathbf{P_G}$ holds. (It means: actions that are not mentioned in the guidelines are not taken into consideration). Then denote $\mathbf{N_0}$ the set of all action nodes with parameters $token = 1$ and $action = P$.
   (a) If the set $\mathbf{N_0}$ is void, print "Error in sequence of actions" and finish.
   (b) If the set $\mathbf{N_0}$ is non void, then in each action node $A \in \mathbf{N_0}$ set $A(time) = t$ and $A(result) = c$. Here t and c are taken from the last deleted element $P(t) = c$ of the sequence $\mathbf{S}$. Denote as $\mathbf{N_1}$ the set of all nodes A from $\mathbf{N_0}$ that satisfy the following conditions *cond1* and *cond2*.

   cond1 If A is inside a subgraph $BRN - SYN$, then the time condition *cond2* of the node SYN is fulfilled.

   cond2 If the parameter ref of A contains the reference to a TIM node (i.e. $TIM(ref) \neq NULL$), the time condition β of node TIM is fulfilled.

   If the set $\mathbf{N_1}$ is void print "Time error" and finish. Otherwise for each node $A \in \mathbf{N_0}$ set $A(token) = 0$ and for each $A \in \mathbf{N_1}$ propagate its token to the nearest node capable of storing tokens. If the propagated token is caught by a node SYN, then set $SYN(time) = t$. Here t is taken from the last deleted element $P(t) = c$.

   At the end of the n-th step propagate tokens from those SYN nodes, the condition α of which is fulfilled. If α condition is fulfilled for some node SYN, then only one token

Table 2: Comparison of the data sequence $\mathbf{S_A}$ with EGLIF. After the step 15 the data sequence is void and $STOP$ node does not contain token. It means that patient A has been treated according to guidelines and that the treatment has not yet finished.

Step	Data item S_i	A_1	A_2	A_3	A_4	$SYN_1(1)$	$SYN_1(2)$	$SYN_1(3)$	$SYN_1(4)$	A_5	A_6	A_7	$SYN_2(1)$	$SYN_2(2)$
0	Start	•	•	•	•									
1	$SBP(1.1.01) = 150$		•	•	•	•								
2	$DBP(1.1.01) = 85$			•	•	•	•							
3	$HDL(2.1.01) = 1$			•		•	•		•					
4	$LDL(2.1.01) = 6$					•	•	•	•			•		
5	$Diet(2.1.01) = 1$									•				•
6	$DBP(10.2.01) = 85$									•				•
7	$SBP(10.2.01) = 140$	•	•	•	•								•	•
8	$SBP(1.5.01) = 130$		•	•	•	•								
9	$DBP(1.5.01) = 85$			•	•	•	•							
10	$HDL(2.5.01) = 1$			•		•	•		•					
11	$LDL(2.5.01) = 5$	•	•	•	•	•	•	•	•					
12	$SBP(1.4.02) = 130$		•	•	•	•								
13	$DBP(1.4.02) = 90$			•	•	•	•							
14	$LDL(2.4.01) = 7$				•	•	•	•						
15	$HDL(2.4.01) = 2$	•	•	•	•	•	•	•	•					

is propagated from this node. If the propagated token is caught by a node N, then set $N(time) = SYN(time)$.

3. If the stopping condition is not fulfilled, increase the number of algorithm steps $n = n + 1$ and go to 2.

   Stopping condition: The algorithm stops if the following conditions (a) or (b) hold.

   (a) Some $ERROR$ or $STOP$ node contains a token.

   (b) The current data sequence is void.

Part 2 (Rules of token propagation)

1. Tokens are propagated along graph branches. In a decision node a token continues along that branch i, for which the condition α_i is fulfilled. In a branch node with n branches new $n - 1$ tokens are created. Each of n tokens coming of the branch node then continues along a different branch.

2. An action node hands over its token to the nearest node that receives tokens. If a token passes a branch node and new tokens are created, then also each newly created token is caught by the nearest node that is capable to catch it.

3. Synchronization node SYN hands over only 1 token and it does it in the same way as an action node. Moreover, the following actions are done:

   (a) All tokens present in the SYN node are discarded.

   (b) All tokens, stored in the nodes on branches between SYN node and its corresponding BRN node, are discarded.

4. If a node N hands over its token to another node and if the token during its handover passes through a node TIM, then the parameter $time$ of the node TIM is set to be $TIM(time) = N(time)$.

5. If a token was handed over to an action node A and if it passed during its handover a node TIM, then into parameter ref of the node A the pointer to the node TIM is written. If the token has not passed through any time node TIM, then into parameter ref of the node A the pointer $NULL$ is written.

If CA algorithm has not finished with error, it might stop due to one of two following reasons.

1. All data items S_i have been already removed from the data sequence $\mathbf{S}$. In this case the patient has been treated in accordance with guidelines. However, according to guidelines, his/her treatment should continue.

2. In the last step of the algorithm a token has been stored in a node $STOP$. In this case the patient has been treated in compliance with guidelines and the treatment in accordance with guidelines was finished. If all data items have been removed from data sequence $\mathbf{S}$, the physician finished the treatment. On the other hand, if some data items remained in $\mathbf{S}$, the patient has been cured further, perhaps due to some other health problems.

To check behavior of the CA algorithm we can apply it to comparison of the Small guidelines for heart failure prevention from Example 1 with data records of the patients A, B, C and D given above. We suppose that guidelines were formalized using EGLIF model and that data records of patients were transformed into data sequences

Table 3: Comparison of the data sequence $\mathbf{S_B}$ with EGLIF. In the step 5 the error signal "Error in sequence of actions" is generated due to the void set $\mathbf{N_0}$.

Step	Data item S_i	A_1	A_2	A_3	A_4	$SYN_1(1)$	$SYN_1(2)$	$SYN_1(3)$	$SYN_1(4)$	A_5	A_6	A_7	$SYN_2(1)$	$SYN_2(2)$
0	Start	•	•	•	•									
1	$SBP(1.1.01) = 150$		•	•	•	•								
2	$DBP(1.1.01) = 85$			•	•	•	•							
3	$HDL(2.1.01) = 1$			•		•	•		•					
4	$LDL(2.1.01) = 6$					•	•	•	•			•		
5	$DBP(10.2.01) = 85$											•		

Table 4: Comparison of the data sequence $\mathbf{S_C}$ with EGLIF. In the step 6 the error signal "Time error" is generated, because the set $\mathbf{N_1}$ is void.

Step	Data item S_i	A_1	A_2	A_3	A_4	$SYN_1(1)$	$SYN_1(2)$	$SYN_1(3)$	$SYN_1(4)$	A_5	A_6	A_7	$SYN_2(1)$	$SYN_2(2)$
0	Start	•	•	•	•									
1	$SBP(1.1.01) = 150$		•	•	•	•								
2	$DBP(1.1.01) = 85$			•	•	•	•							
3	$HDL(2.1.01) = 1$			•		•	•		•					
4	$LDL(2.1.01) = 6$					•	•	•	•			•		
5	$Diet(2.1.01) = 1$									•	•			
6	$DBP(1.4.01) = 85$									•	•			

$\mathbf{S_A}, \mathbf{S_B}, \mathbf{S_C}$ and $\mathbf{S_D}$. The runs of the algorithm for particular data sequences can be followed in the Tables 2–5. In the tables the movement of tokens is visualized. The presence of a token in some node (or tokens if the node is of type SYN) at the end of the n-th step is represented with a black point. For example a black point in the cell $(step = 0, A_1)$ of the Table 2 means that $A_1(token) = 1$ at the end of the 0-th step, a black point in the cell $(step = 3, SYN_1(2))$ means that $SYN_1(token_2) = 1$ at the end of the third step, void cell $(step = 2, SYN_1(3))$ means that $SYN_1(token_3) = 0$ at the end of the second step and so on. If the row of some step is divided into two sub-rows, then the first sub-row describes token layout after the first phase of the step, it means just before token propagation from SYN nodes takes place.

If the CA algorithm compares the EGLIF model of the guidelines with data sequence $\mathbf{S_A}$ (see Table 2), the comparative process stops at the step 15. The final data sequence is void and $STOP$ node does not contain token. It means that the patient A has been treated according guidelines and that the treatment has not yet finished.

If the CA algorithm compares EGLIF model of the guidelines with the data sequence $\mathbf{S_B}$ (see Table 3), the comparative process ends at the step 5 and the error signal "Error in sequence of actions" is generated. The error is generated because at the beginning of the step 5 only node A_7 has token, the last deleted data item used for comparing is $DBP(10.2.01) = 85$ and $A_7(action) = Diet$. As $Diet \neq DBP$, the set $\mathbf{N_0}$ is void and consequently the error signal "Error in sequence of actions" is generated.

If the CA algorithm compares EGLIF model of the guidelines with the data sequence $\mathbf{S_C}$ (see Table 4), the comparative process stops in the step 6 and the error signal "Time error" is generated. In the step 6 the current data item is $DBP(1.4.01) = 85$. The only action nodes having token at the beginning of the step 6 are the nodes A_5 and A_6. Their parameter $action$ has value $A_5(action) = SBP$ and $A_6(action) = DBP$ respectively. Hence the set $\mathbf{N_0} = \{A_6\}$ and $A_6(time)$ is set to be 1.4.01. The node A_6 is inside subgraph $BRN_2^{\smile}SYN_2$ and β condition of SYN_2 is

$$(1\,\text{month} \leq (atime - A_7(time)) \leq 2\,\text{months}).$$

Therefore the condition

$$(1\,\text{month} \leq (A_6(time) - A_7(time)) \leq 2\,\text{months}).$$

should hold. However, this condition does not hold, because $A_7(time) = 2.1.01$. Hence the error signal "Time error" is generated.

If the CA algorithm compares EGLIF model of guidelines with the data sequence $\mathbf{S_D}$ (see Table 5), the comparative process stops in the step 12 and the error signal "Time error" is generated. At the beginning of the step 12 the nodes with token are the nodes A_1, A_2, A_3 and A_4, but only the node A_1 has value of parameter $action$ equal to SBP. Hence $\mathbf{N_0} = A_1$ and $A_1(time) = 1.4.02$. As the token has been handed over to the node A_1 through the node TIM_2, we have $A_1(ref) = TIM_2$ and $TIM_2(time) = 2.5.01$. The β condition of TIM_2 is

$$ftime - TIM_2(time) \leq 0.5\,\text{year}.$$

As $ftime = 1.4.02$, the condition β is not fulfilled and the error signal "Time error" is generated.

Table 5: Comparison of the data sequence $\mathbf{S_D}$ with EGLIF. In the step 12 error signal "Time error" is generated, because the set N_1 is void.

Step	Data item S_i	A_1	A_2	A_3	A_4	$SYN_1(1)$	$SYN_1(2)$	$SYN_1(3)$	$SYN_1(4)$	A_5	A_6	A_7	$SYN_2(1)$	$SYN_2(2)$
0	Start	•	•	•	•									
1	$SBP(1.1.01) = 150$		•	•	•	•								
2	$DBP(1.1.01) = 85$			•	•	•	•							
3	$HDL(2.1.01) = 1$			•		•	•		•					
4	$LDL(2.1.01) = 6$					•	•	•	•			•		
5	$Diet(2.1.01) = 1$									•	•			
6	$DBP(10.2.01) = 85$									•				•
7	$SBP(10.2.01) = 140$	•	•	•	•								•	•
8	$SBP(1.5.01) = 130$		•	•	•	•								
9	$DBP(1.5.01) = 85$			•	•	•	•							
10	$HDL(2.5.01) = 1$			•		•	•		•					
11	$LDL(2.5.01) = 5$	•	•	•	•	•	•	•	•					
12	$SBP(1.4.02) = 130$		•	•	•	•								

4 Conclusion

In this paper we designed an algorithm that compares a patient's treatment described with a patient's health record with a formal model (EGLIF) of some clinical guideline. The algorithm works correctly under two conditions:

1. The EGLIF model must be strict, which means that the choice of the branch in all decision nodes must be unambiguous.

2. The data record must be complete, which means that all examined patient's parameters and prescribed or applied therapies must be recorded.

However, guidelines often recommend for a particular patient's state more possible ways how to continue with treatment. In a GLIF model it is modeled with in-conditions, strict in-conditions, out-conditions and strict out-conditions. If a strict in-condition or a strict out-condition of the branch is satisfied, this branch is strictly recommended or strictly prohibited and the physician is strictly urged to follow or not to follow it. In-conditions and out-conditions are soft conditions and should be taken only as hints how to continue.

So in practice GLIF models are rarely strict and some specification what the compliance of a non-strict guidelines with a patient's data record actually means is needed.

Here we introduce one possible specification. At first we define the notion of admissibility of a decision branch.

When a token passes a decision node, then each branch originating in this node and satisfying conditions C_1 and C_2 is called admissible.

C_1 All the strict out-conditions and out-conditions on the branch are evaluated as false.

C_2 At least one strict in-condition or in-condition on the branch is evaluated as true.

Subsequently we may define compliance of a patient's treatment with non-strict guidelines. The treatment is in compliance with the given non-strict guidelines if all decisions made resulted in following of admissible branches only.

We may easily modify the comparing algorithm CA introduced above so that it could compare a patient's record and non-strict guidelines and determine the compliance of patient's treatment with it. Modification consists in token multiplication in the decision nodes. If a token passes a decision node new tokens are created so that one token could continue along every admissible branch.

The second assumption is that about completeness of patient's data record. It is clear that in the situation when we know that only some data about patient's treatment are stored in his/her data record, the possibility to test compliance of his treatment with guidelines model is strongly limited. However, in some cases non-compliance can be discovered in spite of the missing data. This happens if the data record would be non-compliant with the guidelines model whatever the missing data were.

If we used the algorithm CA described above, it would generate error for every missing item. However, it might not be our intention. We might admit missing data and we might want to get warnings only if the non-compliance is obvious from the remaining data themselves. If so, some modification of the CA algorithm is needed. To find such modification is much more difficult, than to enhance the algorithm for non-strict guidelines models. At present it is a subject of the further research.

Acknowledgements

The research was supported by the research institutional project MSM 6046070904 and by the project 1M06014 of the Center of Biomedical Informatics of Ministry of Education, Youth and Sports CR.

References

[1] D. Isern, A. Moreno, Computer-based Execution of Clinical Guidelines: A Review, International Journal of Medical Informatics, 77, pp. 787-808, Maastricht, 2008.

[2] National Guidelines Clearinghouse, http://www.guidelines.gov/submit/template.aspx

[3] National Library of Guidelines Specialist Library, http://www.library.nhs.uk/GuidelinesFinder

[4] Catalogue of Clinical Practice Guidelines (neo.euromise.cz/kkdp).

[5] M. Zvolský, The Database of the Catalogue of Clinical Practice Guidelines Published via Internet in the Czech Language – The Current State. European Journal for Biomedical Informatics 6 (2010), 83-89.

[6] V. Patel, V. Allen, J. Arocha, E. Shortliffe, Representing clinical guidelines in GLIF:individual and collaborative expertise, Journal American Medical Inf. Association, 1998, 5(5), 467-483.

[7] V. L. Patel, T. Branch, D. Wang, M. Peleg, A. Boxwala, Analysis of the Process of Encoding Guidelines: A Comparison of GLIF2 and GLIF3, Methods Inf. Med., 2002, no.2, pp. 105-113.

[8] D. Buchtela, J. Peleška, M. Zvolský, J. Zvárová, Medical Knowledge Representation System, In: S. K. Andersen et al., Proceedings of MIE2008 e-Health Beyond Horizon, pp. 377-382, IOS Press, Goteborg, 2008.

[9] D. Buchtela, J. Peleška, A. Veselý, M. Zvolský, J. Zvárová, Formalization of Clinical practice Guidelines, In: S. K. Andersen et al., Proceedings of MIE2008 e-Health Beyond Horizon, pp. 151-156, IOS Press, Goteborg, 2008.

[10] A. Veselý, J. Zvárová, J. Peleška, Formalization of Clinical Practice Guidelines, In: S. K. Andersen et al., Proceedings of MIE2008 e-Health Beyond Horizon, pp. 151-156, IOS Press, Goteborg, 2008.

[11] A. Veselý, J. Zvárová, J. Peleška, Z. Anger, D. Buchtela, Computerized Presentation of Medical Guidelines, In: Proceedings Medinfo 2004 AMIA San Francisco, 2004.

[12] A. Latoschek-Berendsen, H. Tange, H. Herik, A. Hasman, From clinical practice guidelines to computer-interpretable guidelines, Methods Inf Med, 6, 2010.

[13] P. Gillois, G. Chatellier, M. Jaulent, I. Colombet, M. Fieschi, P. Degoulet, From paper based to electronic gidelines: application to French guidelines, Medinfo 2001, 10, 196-200.

[14] J. Fox, E. Black, I. Chronakis, R.Dunlop, V. Patkar, M. South et al., From guidelines to careflows: modeling and supporting complex clinical processes, Stud Health Technol Inform, 2008, 139, 44-62.

[15] P. Shekelle, S. Woolf, M. Eccles, J. Grimshaw, Clinical guidelines:developing guidelines, BMJ 1999,318,593-596.

[16] R. Gould, A. Hasman, A. Strijbis, N. Peek, A parallel guidelines development and formalization strategy to improve the quality of clinical practice guidelines, IntJ Med Inform, 2009, 78 (8), 513-520.

[17] P. DeClerq, K. Kaiser, A. Hasman, Computer-interpretable Guidelines Formalisms, Stud HealthTechnol Inform, 2008, 139, 22-43.

[18] D. Wang, M. Peleg, S.Tu, A.Boxwala, R.Greenes, V. Patel et al, representation primitives, process models and patient data in computer-interpretable clinical practice guidelines: a literature review of guidelines representation models., Int J Med Inform, 2002, 68, 59-70.

[19] J. Fox, S. Das, Safe and sound, 2000, MIT Press.

[20] D.R Sutton, J. fox, The syntax and semantics of the PROforma guidelines modeling language, Journal of the American Medical Informatics Association, 2003, 10, 433-443.

[21] Y. Shahar, O. Young, E. Shalom, M. Galperin, A. Mayaffit, R. Moskovitch, A. Hessing, A hybrid, multiple-ontology clinical guidelines library and automated guideline-support tools, Journal of Biomedical Informatics, 2004, 37(5), 325-344.

[22] L. Anselma, P. Terenziani, S. Montani, A. Bottrighi, Towards a comprehensive treatment of repetitions, periodicity and temporal constrains in clinical guidelines, Artificial Intelligence in Medicine, 2006, 38 171-195.

[23] P. Terenziani, S. Montani, A. Bottrighi, M. Torchio, G.Molino, The GLARE approach to clinical guidelines: main features, in K. Kaiser, S.Miksch, S. Tu (Eds.), Computer-based support for clinical guidelines and protocols, Proceedings of the Symposium on Computerized guidelines and protocols, IOS Press, 2004, Prague, 162-166.

[24] P. Ciccarese, E. Caffi, L.Boiocchi, S. Quaglini, M. Stefanelli, A guideline management system, in M. Fieschi, E. Coiera, Y. Li (Eds.), Proceedings of 11th Word Congress of the International Medical Informatics Association, San Francisco, IOS Press, 2004, 28-32.

[25] P. Ciccarese, E. Caffi, L.Boiocchi, S. Quaglini, M. Stefanelli,Architectuers and tools for innovative health information systems: the guide project, International Journal of Medical Informatics, 74, 2005, 553-562.

[26] S. Quaglini, M. Stefanelli, A. Cavallini, G. Micieli, C. Fassino, C. Mossa, Guidelines-based careflow systems, Artificial Intelligence in Medicine 20 (2000), 5-22.

[27] J. H. Gennari, M.A. Musen, R.W. Fergerson, W.E. Grosso, M. Grubezy, H. Eriksson, N.F. Noy, S.W. Tu, The evolution of Protégé: an enviroment for knowledge-based systems development, International Journal of Human-Computer Studies 58(1), 2003, 89-123.

[28] D. Isern, D Sanchez, A. Moreno, An ontology-driven agent-based clinical guidelines execution engine, in Proceedings of 10th International Conference on Artificial Intelligence in Medicine, Vol. 4594 of Lecture Notes on Artificial Intelligence, Springer, Berlin, 2007, 49-53.

[29] L. Ohno-Machado, J.H. Gennari, S. N. Murphy, N. L. Jain, S. W. Tu S, D. Oliver, et al., The GuideLine Interchange Format: A model for representing guidelines, Journal of the American Medical Informatics Association, 5(4), 1998, pp. 357-372.

[30] M. Peleg, A. Boxwala, et al.: GLIF3: The Evolution of Guideline Representation Format, In: http://smi web.stanford.edu/projects/intermed-web/guidelines, 2000.

[31] D. Wang, M. Peleg, S. V. Tu, A. Boxwala, et al., Design and implementation of the GLIF3 guideline execution engine, Journal of Biomedical Informatics 37(2004) 305-318.

[32] A. Veselý, J. Zvárová, J. Peleška, Medical guidelines presentation and comparing with electronic health record, International Journal of Medical Informatics, 75, pp. 240-245, Maastricht, 2006.

[33] A. Veselý, J. Zvárová, J. Peleška, M. Tomečková, On Direct Comparing of Medical Guidelines with Electronic Health Record, In: Proceedings of MIE2003 IOS San Malo, 2003.

[34] A. Veselý, J. Zvárová, J. Peleška, D. Buchtela, Z. Anger, Medical Guidelines Presentation and Comparing with Electronic Health Record, In: Proceedings of the International Joint Meeting Merkantilie Prague Prague, 2004.

[35] J. Zvárová A. Veselý, P. Hanzlíček, J. Špidlen, D. Buchtela, On Direct Comparing of Medical Guidelines with Electronic Health Record, In: M. Bubak et al.(Eds.): Computational Science-ICCS2004,Part IV Springer-Verlag Berlin Krakow, 2004, 1133-1139.

