Time-Independent Rule-Based Guideline Induction

David Riaño [1]

Abstract. Whereas decision trees are widely studied in the context of knowledge representation and inductive learning, guidelines can be considered as an extension of decision trees that have not received the same amount of attention in the artificial intelligence community, even though they have been proved highly useful in several decision support problems as the therapy process in health-care.

This paper is about the formal definition of guidelines as a rule-based knowledge structure and also about the introduction of an incremental inductive learning algorithm to develop and update time-independent guidelines. These ideas have been tested in the health-care domain of some heart diseases.

1 INTRODUCTION

In artificial intelligence, classification is the process of assigning an object to one of a predefined set of classes according to a list of descriptive properties, and planning is the selection of a sequence of actions that will result in the achievement of a desired goal. Very often, these processes appear in other domains with different names. For example, in medicine the procedure by which a patient is labeled with a particular disease is called *diagnosis*, and the procedure by which a patient is clinically treated is called *therapy*. The most extended representations of a classification (or diagnosis) process are decision trees and production rules [9] [**?**] [13]. In clinical planning (or therapy), knowledge-based plans use to be represented as clinical practice guidelines (CPGs) or protocols [10] [15].

There are many alternative representation models to describe computer-interpretable CPGs: Arden syntax, Asbru [15], EON [5], GLIF [10], GUIDE [14], PRODIGY [12], PRO*forma* [8], etc. A wise comparison of them can be found in [2] and [11].

Decision trees and production rules can be obtained directly from an expert with a knowledge acquisition procedure or from a set of supervised instances with an inductive learning procedure. Unfortunately, nowadays it is not clear how a guideline or protocol can be inductively obtained from an already scheduled set of situations and, therefore, only user defined guidelines are available.

Extensible to the guidelines, there is a global agreement that any CPG representation model must comply with the following requirements [11]: branching, asserting, unfolding, and state representation, among others. *Branching* is the feature that there can be points in the guideline where a selection from a set of alternatives must be taken on the basis of some predefined criterion; *asserting* is the way that a guideline uses to indicate that some actions must be applied at a particular point; *unfolding* is when some parts of the guideline recommend a shift to another guideline, and *state representation* is used to describe specific scenarios of patient's clinical status in the context of a particular point of the guideline.

[1] Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Tarragona, Spain. Email: drianyo@etse.urv.es

Nowadays CPGs are made by expert committees which are created by health-care organizations as the HSTAT in the USA, the SIGN in Scotland, or the NZGG in New Zealand. Therefore, it is expected that a CPG requires on the one hand, a great amount of knowledge investment at the beginning, while the guideline is being made and, on the other hand, periodic updates in order to include new clinical drugs or clinical procedures in the CPG. These are difficulties that cannot be easily overcome with automatic methods since only simple CPGs can be automatically generated and updated [3, 4].

In this paper, we propose a new general purpose guideline representation model that comply with the requirements of branching, asserting, unfolding and state representation, and introduce a learning algorithm to induce time-independent rule-based guidelines. This knowledge model and the learning algorithm have been designed independent of the target domain: science, industry, health-care, etc. and they have been tested and proved useful in the domain of cardiac diseases with the induction of CPGs for bradycardia and tachycardia.

In section 2, we describe the guideline representation model and the inductive algorithm. In section 3, we use the model to represent CPGs as a particular case of guideline. Finally, in section 4 we show some conclusions.

2 GUIDELINE COMPOSITION

A decision tree is a binary tree in which each non-leaf vertex is a yes-no query, each leaf is labeled as a target, the edge from any non-leaf vertex to its left son is labeled 'Yes' (the condition is satisfied), and the one to its right son is labeled 'No' (the condition is not satisfied).

Here, a guideline can be broadly described as a tree structure that combines branching, asserting, unfolding, and state representation vertices in order to represent a hopefully complete description of an action plan. Formally, we describe a guideline as a decision tree whose non-leaf vertices are guarded queries, leaves are discharge reasons or unfold actions, and edges can be attached a set of actions.

Decision trees are commonly used in health-care domains to represent clinical *diagnostic knowledge*, and guidelines or CPGs represent clinical *therapy knowledge*. For example, figure 1 shows a decision tree that diagnoses the sort of emergency cardiac care that must be rendered to a particular Adult patient, and figure 2 describes the CPG that the Maryland Institute for Emergency Medical Services recommends to follow with patients diagnosed with a cardiac emergency.

In the guidelines, branching and state representation are related to the *guarded* query vertices (a guarded query is satisfied only if the state of the system is subsumed by the guard, and the branching decision by the query); asserting is related to the edges that contain actions as drug prescriptions or clinical procedures in CPGs, and unfolding is related to the leaves of the guidelines, where other guidelines can be unfold. For example, some steps of the CPG in figure 2 can be conditioned to the age of the patient and not applicable to

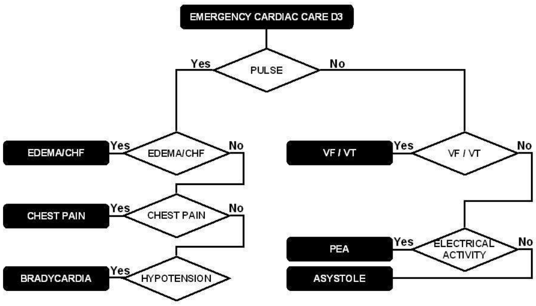

Figure 1. Decision tree for cardiac emergencies.

elder people, in such cases a guard should be related as patient state to the branching conditions to be avoided. Assertions are the square blocks, and unfolding points are those remarked in black.

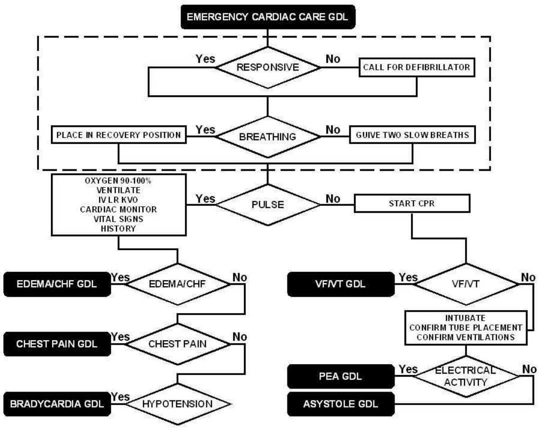

Figure 2. CPG for a cardiac emergency.

Production rules are the natural alternative to graphical decision trees. They sacrifice the graphical representation of decision trees, but they encourage other interesting properties about the represented knowledge as simplicity, scalability, understandability, uncontextuality, flexibility, and reusability. Moreover, converting production rules into decision trees is a solved problem [6].

As we will see, guidelines can also be represented as rules.

2.1 Rule-Based Guideline Representation

The same way that diagnosis can be done with a set of production rules [13], therapies can be represented with cause-effect rules where causes are conditions in the non-leaf vertices and effects are the actions in the edges. In order to this becomes completely true two restrictions must be set on the sort of guidelines that are represented by cause-effect rules: time-independence and order-independence.

Time-independence means that guidelines do not have time constraints. *Order-independence* means that the results obtained by the inference engine do not change if the rules are in a different order.

Let $S = \{s_1, s_2, ...\}$ be a set of state descriptors, $C = \{c_1, c_2, ...\}$ a set of causes, and $E = \{e_1, e_2, ...\}$ a set of effects.

A *single IT act* is defined as a tuple a_i=(S_i, C_i, E_i) where $S_i \subset S$ represents one state that activates the act, $C_i \subset C$ is the cause that fires the act, and $E_i \subset E$ is the effect of the act. Single IT acts can be represented as *if-then* rules; for example, the first step of the CPG in figure 2 represents the rule ”*if not responsive then call for defibrillator*” or the IT act ($\emptyset$, {*not-responsive*}, {*call-for-defibrillator*}). A *single ITE act* is an extension of the single IT act as a tuple a_i=(S_i, C_i, E_i, E'_i) where $E'_i \subset E$ is the counter-effect of the act, i.e. the effects when the cause does not fire the act. Single ITE acts can be represented as *if-then-else* rules. For example, the second step in figure 2 is represented by the rule ”*if breathing then place in recovery position else give two slow breaths*” or the ITE act ($\emptyset$, {*breathing*}, {*place-in-recovery-pos*}, {*give-two-slow-breaths*}).

A complex act or *guideline* is defined as a set of single acts {(S_i, C_i, E_i, E'_i): i=1, 2, ..., n}, with E'_i optional and $n \geq 0$.

Observe that a guideline describes an order-independent act or, in other words, that it is a declarative description of a complex action instead of a procedural one. Moreover, acts are time-independent.

2.2 The Training Set

As in the induction of production rules the training set contains instances of the classes that the rules differentiate, in the induction of cause-effect rules that represent guidelines, the training set must contain instances of single decisions that the experts take during the therapy process.

Experts are expected not only to make right decisions but also to justify them supplying the exact causes that entrust their decisions and sometimes, the consequences that are derived from the absence of such causes. Among the possible causes of a decision we distinguish between *direct* causes and *concurrent* causes. Although this division is conceptual and does not affect the learning algorithm described in section 2.3, the first group is used to contain the causes that are directly related with the training domain, and the second group the rest of the causes indirectly related to the domain. For example, in the domain of cardiac emergencies, decisions can be justified with medical reasons about the disease (e.g. pulse, breathing, VF/VT, etc.) or with other information about the patient (e.g. age, sex, etc.).

Each one of the above expert decisions can be represented by a single IT act (S_i, C_i, E_i), or a single ITE act (S_i, C_i, E_i, E'_i), where S_i stands for the concurrent causes, C_i for the direct causes, E_i for the decisions made, and E'_i for the consequences derived from the absence of C_i. In other words, each single act represents an expert atomic action or an instance of the expert behavior that we want to capture in a guideline. So, ({*Adult*},{*ChestPain, Hypotension, AMI*}, {*antropine*}, {*Dopamine, transcutaneous pacing*}) represents a medical ITE act that the physician decided during a concrete visit of a concrete patient. Then, a complete patient therapy can be represented as a list of medical acts or a guideline structure that can be understood as the adaptation of an ’unknown’ general guideline to that particular patient. The algorithm in the next section describes how to induce a general guideline from patient concrete treatments.

2.3 The Incremental Inductive Learning Algorithm

A single act is *complete* if it contains all the information required to make the decision that it represents. Since S_i and C_i are defined as sets, only conjunctive decisions are possible. When all the single acts of a guideline are complete, the knowledge that each single act represents is self-contained and independent of the rest of the single acts in the guideline.

We work under the assumption that guidelines are both time-independent and order-independent. When a guideline is defined as a list of complete single acts, these single acts describe the best decisions made, according to the circumstances of a particular complex decision which is represented by the whole guideline. A general guideline is constructed as the integration of the best decisions made for many individual complex decisions which act as input instances of the learning algorithm.

Before we introduce the algorithm, we define the combination factor θ_{ij} between two acts a_i and a_j as their similarity (see eq. 1).

$$\theta_{ij} = \frac{|S_i \cap S_j|}{|S_i \cup S_j|} \tag{1}$$

We define the combination operator $\oplus$ that merges two single acts a_i and a_j following the indications in tables 1 and 2. Although in [1] there is an extended description of how these combinations rules are obtained, broadly speaking they are the result of the logical expression $a_i \wedge a_j$, where a_k is represented as $(\neg S_i \vee \neg C_i \vee E_i)$ for IT acts and $(\neg S_i \vee \neg C_i \vee E_i) \wedge (\neg S_i \vee C_i \vee E'_i)$ for ITE acts. Some conflicts are solved with a generalisation step in order to maintain causes as conjunctive expressions.

Table 1. Combination operator $\oplus$ for single IT acts.

a_i	a_j	$a_i \oplus a_j$
(S_i, C_i, E_i)	$(S_j, C_i \cup C, E_i)$	$(S_i \cap S_j, C_i, E_i)$
(S_i, C_i, E_i)	$(S_j, C_i \cup C, E_i \cup E)$	$(S_i \cap S_j, C_i, E_i)$ $(S_j, C_i \cup C, E)$
$(S_i, C_i, E_j \cup E)$	$(S_j, C_i \cup C, E_j)$	$(S_i \cap S_j, C_i, E_j \cup E)$
$(S_i, C_i, E_i \cup E)$	$(S_j, C_i \cup C, E_j \cup E)$	$(S_i \cap S_j, C_i, E_i \cup E)$ (S_j, C_i, E_j)
(S_i, C_i, E_i)	$(S_j, C_i \cup C, E_j)$	$(S_i \cap S_j, C_i, E_i \cup E_j)$
$(S_i, C_i \cup C, E_i)$	$(S_j, C_j \cup C, E_i)$	$(S_i \cap S_j, C_i \cup C_j, E_i)$
$(S_i, C_i \cup C, E_i)$	$(S_j, C_j \cup C, E_i \cup E)$	$(S_i \cap S_j, C_i \cup C_j, E_i)$ (S_j, C_j, E)
$(S_i, C_i \cup C, E_j \cup E)$	$(S_j, C_j \cup C, E_j)$	$(S_i \cap S_j, C_i \cup C_j, E_j)$ (S_j, C_i, E)
$(S_i, C_i \cup C, E_i \cup E)$	$(S_j, C_j \cup C, E_j \cup E)$	$(S_i \cap S_j, C_i \cup C_j \cup C, E)$ (S_i, C_i, E_i) (S_j, C_j, E_j)
$(S_i, C_i \cup C, E_i)$	$(S_j, C_j \cup C, E_j)$	$(S_i, C_i \cup C, E_i)$ $(S_j, C_j \cup C, E_j)$
(S_i, C_i, E_i)	(S_j, C_j, E_i)	(S_i, C_i, E_i) (S_j, C_j, E_i)
(S_i, C_i, E_i)	$(S_j, C_j, E_i \cup E)$	(S_i, C_i, E_i) $(S_i \cap S_j, C_j, E)$
$(S_i, C_i, E_j \cup E)$	(S_j, C_j, E_j)	$(S_i \cap S_j, C_i, E)$ (S_j, C_j, E_j)
$(S_i, C_i, E_i \cup E)$	$(S_j, C_j, E_j \cup E)$	$(S_i \cap S_j, C_i \cup C_j, E)$ (S_i, C_i, E_i) (S_j, C_j, E_j)
(S_i, C_i, E_i)	(S_j, C_j, E_j)	(S_i, C_i, E_i) (S_j, C_j, E_j)

The learning algorithm is designed to integrate all the single acts of a complex decision G_j into a guideline G_i. For all the single acts $a_j \in G_j$, the most similar to a_j single act $a_i \in G_i$ is taken. If θ_{ij} is below a predefined threshold, $a_i \oplus a_j$ replaces a_i in G_i; otherwise, a_j is incorporated to G_i. The incremental algorithm finishes when all the single acts in G_j are treated.

```
algorithm combine_GDL (Gi, Gj: GDL): GDL
 if empty(Gj) then return Gi
 else
  aj := first_single_act(Gj);
  remove_single_act(aj, Gj);
  ai := single_act_with_highest_θij(Gi, aj);
  if θij < threshold then
   remove_single_act(ai, Gi);
   Gi := union(Gi, combine(ai, aj));
  else
   Gi := union(Gi, aj);
  end if;
  return combine_GDL(Gi, Gj);
 end if;
end algorithm
```

Table 2. Combination operator $\oplus$ for single ITE acts.

a_i	a_j	$a_i \oplus a_j$
(S_i, C_i, E_i, E'_i)	$(S_j, C_i \cup C, E_i, E'_j)$	$(S_i \cap S_j, C_i, E_i, E'_i)$
(S_i, C_i, E_i, E'_i)	$(S_j, C_i \cup C, E_i \cup E, E'_j)$	$(S_i \cap S_j, C_i, E_i, E'_i)$ $(S_j, C_i \cup C, E, E'_j)$
$(S_i, C_i, E_j \cup E, E'_i)$	$(S_j, C_i \cup C, E_j, E'_j)$	$(S_i \cap S_j, C_i, E_j \cup E, E'_i)$
$(S_i, C_i, E_i \cup E, E'_i)$	$(S_j, C_i \cup C, E_j \cup E, E'_j)$	$(S_i \cap S_j, C_i, E_i \cup E, E'_i)$ (S_j, C_i, E_j, E'_i)
(S_i, C_i, E_i, E'_i)	$(S_j, C_i \cup C, E_j, E'_j)$	$(S_i \cap S_j, C_i, E_i \cup E_j, E'_i)$
$(S_i, C_i \cup C, E_i, E'_i)$	$(S_j, C_j \cup C, E_i, E'_j)$	$(S_i \cap S_j, C_i \cup C_j, E_i, E'_i \cup E'_j)$
$(S_i, C_i \cup C, E_i, E'_i)$	$(S_j, C_j \cup C, E_i \cup E, E'_j)$	$(S_i \cap S_j, C_i \cup C_j, E_i, E'_i \cup E'_j)$ (S_j, C_j, E, E'_j)
$(S_i, C_i \cup C, E_j \cup E, E'_i)$	$(S_j, C_j \cup C, E_j, E'_j)$	$(S_i \cap S_j, C_i \cup C_j, E_j, E'_i \cup E'_j)$ (S_j, C_i, E, E'_i)
$(S_i, C_i \cup C, E_i \cup E, E'_i)$	$(S_j, C_j \cup C, E_j \cup E, E'_j)$	$(S_i \cap S_j, C_i \cup C_j \cup C, E, E'_i \cup E'_j)$ (S_i, C_i, E_i, E'_i) (S_j, C_j, E_j, E'_j)
$(S_i, C_i \cup C, E_i, E'_i)$	$(S_j, C_j \cup C, E_j, E'_j)$	$(S_i, C_i \cup C, E_i, E'_i \cup E'_j)$ $(S_j, C_j \cup C, E_j, E'_i \cup E'_j)$
(S_i, C_i, E_i, E'_i)	(S_j, C_j, E_i, E'_j)	(S_i, C_i, E_i, E'_i) (S_j, C_j, E_i, E'_j)
(S_i, C_i, E_i, E'_i)	$(S_j, C_j, E_i \cup E, E'_j)$	(S_i, C_i, E_i, E'_i) $(S_i \cap S_j, C_j, E, E'_j)$
$(S_i, C_i, E_j \cup E, E'_i)$	(S_j, C_j, E_j, E'_j)	$(S_i \cap S_j, C_i, E, E'_i)$ (S_j, C_j, E_j, E'_j)
$(S_i, C_i, E_i \cup E, E'_i)$	$(S_j, C_j, E_j \cup E, E'_j)$	$(S_i \cap S_j, C_i \cup C_j, E, E'_i \cup E'_j)$ (S_i, C_i, E_i, E'_i) (S_j, C_j, E_j, E'_j)
(S_i, C_i, E_i, E'_i)	(S_j, C_j, E_j, E'_j)	(S_i, C_i, E_i, E'_i) (S_j, C_j, E_j, E'_j)

3 COMPOSING CPGs IN THE CONTEXT OF CARDIAC DISEASES

In Evidence-Based Medicine, CPGs are defined as "systematically developed statements which assist practitioners and patients make decisions about appropriate health-care for specific clinical circumstances" [7]. CPGs can be graphically represented as in figure 2 or rule-based represented as we described in the previous section.

The above rule-based guideline knowledge-representation model and inductive algorithm are tested in the context of cardiac diseases. In normal Adults, the heart beats regularly at a rate of 60 to 100 times per minute. *Bradycardia* is an abnormally low heart rate of less than 60 beats per minute. On the contrary, *tachycardia* is a heart rate of more than 100 beats per minute. The purpose is twofold, on the one hand we want to analyse the robustness of the algorithm, on the other hand we want to obtain valid CPGs for each one of these cardiac problems. The input of both tests is a list of 15 bradycardia patients (6 Adults and 9 Children) and 7 tachycardia patients, and the therapies followed for each case. In the first test, the set of causes $C = \{\mathtt{AMI}, \mathtt{ChestPain}, ...\}$ was used to make all the possible combinations describing input patients. Each combination was tested with the input CPGs (i.e. those which represent the single patient treatments) and with the CPG resulting from the combination. The result was that the combined CPG proposed a therapy equal to the closest single patient in more than 87% for bradycardia and 91% for tachycardia. The second test is commented in the next sections.

3.1 Bradycardia

The treatments of 6 Adult patients and 9 Children suffering from bradycardia have been used to define a general CPG about bradycardia. Table 3 shows the description of the 15 treatments as complex acts. For example, the first treatment was applied to an Adult that had got pain in his chest, showed hypotension, and acute myocardial infarction (AMI). The physician recommended atropine, and proposed transcutaneous cardiac pacing (TCP) if some of the causes was not present. After that, the patient showed a congestive heart failure (CHF) combined with hypotension and loss of consciousness, and the physician decided to change medication to Dopamine and apply TCP. The treatment concluded when the patient showed pain in his chest, CFH, AMI, and symptomatic, and physician proposed to continue with TCP. Observe that the state of the single actions is only used to distinguish between therapies for Adults and Children.

Table 3. Sample of bradycardia patient treatments.

```
((((Adult) (ChestPain Hypotension AMI) (Atropine) (TCP))
  ((Adult) (CHF Hypotension LostConciousness) (Dopamine TCP) (Atropine))
  ((Adult) (ChestPain CHF AMI Symptomatic) (TCP) (Lidocaine)))
 (((Adult) (ChestPain ShortnessBreath Hypotension CHF) (Dopamine) (Atropine)))
 (((Adult) (ChestPain LostConciousness) (TCP) (Atropine)))
 (((Adult) (PulmonaryCongestion AMI) (Atropine) NIL)
  ((Adult) (AMI CHF LostConciousness) (Dopamine TCP)))
 (((Adult) (Hypoperfusion ChestPain AMI CHF) (Atropine TCP)))
  ((Adult) (CHF Hypotension LostConciousness) (Dopamine)))
 (((Adult) (Hypoperfusion ChestPain AMI) (Atropine) (TCP)))
 (((Child) (ChestPain) (ABC Ventilate90) NIL))
 (((Child) (ChestPain) (ABC Ventilate90) NIL)
  ((Child) (Hypotension) (Epinephrine) NIL))
 (((Child) NIL (ABC Ventilate90) NIL))
 (((Child) NIL (ABC Ventilate90) NIL)
  ((Child) (Hypotension Hypoperfusion) (ChestCompression Epinephrine) NIL))
 (((Child) (ChestPain) (ABC Ventilate90) NIL)
  ((Child) (Hypotension LostConciousness) (Atropine) NIL))
 (((Child) (ChestPain) (ABC Ventilate90) NIL)
  ((Child) (Hypotension) (Atropine) NIL)
  ((Child) (Hypotension LostConciousness) (TCP) NIL))
 (((Child) NIL (ABC Ventilate90) NIL)
  ((Child) (LostConciousness) (Apinephrine TCP) NIL)
  ((Child) (ChestPain Hypotension) (Atropine) NIL))
 (((Child) (LostConciousness) (ABC Ventilate90) NIL)
  ((Child) (Hypotension ChestPain) (Apinephrine) NIL))
 (((Child) NIL (ABC Ventilate90))))
```

When all the knowledge involved in the above clinical episodes is combined with the inductive learning algorithm described in section 2.3, a CPG of bradycardia is obtained. This CPG is shown in table 4.

Table 4. Bradycardia CPG.

```
(((Child) (LostConciousness) (ABC Ventilate90))
 ((Child) (Hypotension ChestPain) (Apinephrine))
 ((Child) (ChestPain) (Atropine))
 ((Child) (LostConciousness) (Apinephrine TCP))
 ((Child) (Hypotension LostConciousness) (TCP))
 ((Child) (Hypotension) (Atropine))
 ((Child) (Hypotension Hypoperfusion) (ChestCompression Epinephrine) NIL)
 ((Child) NIL (ABC Ventilate90))
 ((Child) (Hypotension) (Epinephrine))
 ((Adult) (Hypotension PulmonaryCongestion Hypoperfusion ChestPain AMI CHF) (Atropine))
 ((Adult) (Hypotension ChestPain AMI) (Atropine) (TCP))
 ((Adult) (Hypoperfusion ChestPain AMI CHF) (TCP))
 ((Adult) (AMI CHF LostConciousness) (Dopamine TCP))
 ((Adult) (ChestPain LostConciousness) (TCP) (ATROPHINE))
 ((Adult) (ChestPain ShortnessBreath Hypotension CHF) (Dopamine) (Atropine))
 ((Adult) (CHF Hypotension LostConciousness) (Dopamine TCP) (Atropine))
 ((Adult) (ChestPain CHF AMI Symptomatic) (TCP) (Lidocaine)))
```

3.2 Tachycardia

A CPG has been automatically generated from the data of 7 Adult patient treatments. The data of these treatments are in table 5. The final rule-based CPG is shown in table 6.

Table 5. Sample of tachycardia patient treatments.

```
((((Adult) (NotResponsive) (CallForDefibrillator EMS) NIL)
  ((Adult) (VR>150BPM) (Sedation) (Diltiasem)))
 (((Adult) (SVT) (Ademosine6mg ApplyVFVT_GDL) (Intubate)))
 (((Adult) (NotResponsive Breathing Pulse) (Intubate EMS) (START_CPR))
  ((Adult) (Breathing Edema) (ApplyEdema_CHF_GDL) NIL))
 (((Adult) (NotResponsive) (EMS) NIL)
  ((Adult) (Breathing Pulse) (Intubate) (START_CPR)))
 (((Adult) (NotResponsive Breathing Pulse)(CallForDefibrillator START_CPR) NIL)
  ((Adult) (Breathing Pulse SVT) (ApplyEdema_CHF_GDL) (Ademosine6mg)))
 (((Adult) (Pulse Breathing) (Oxygen90_100) (START_CPR))
  ((Adult) (Pulse Edema) (ApplyEdema_CHF_GDL) (Ademosine6mg)))
 (((Adult) (SVT Pulse) (Ademosine6mg) NIL)
  ((Adult) (SVT NotResponsive) (Oxygen90_100 CallForDefibrillator) NIL)
  ((Adult) (Pulse Edema) (ApplyEdema_CHF_GDL) NIL)))
```

Table 6. Tachycardia CPG.

```
(((Adult) (NotResponsive) (CallForDefibrillator EMS) NIL)
 ((Adult) (Pulse Edema) (ApplyEdema_CHF_GDL) (Ademosine6mg))
 ((Adult) (SVT NotResponsive) (Oxygen90_100))
 ((Adult) (SVT Pulse) (Ademosine6mg))
 ((Adult) (Breathing Pulse SVT) (ApplyEdema_CHF_GDL))
 ((Adult) (NotResponsive Breathing Pulse) (START_CPR))
 ((Adult) (Breathing Pulse) (Intubate Oxygen90_100))
 ((Adult) (Breathing Edema) (ApplyEdema_CHF_GDL))
 ((Adult) (NotResponsive Breathing Pulse) (Intubate))
 ((Adult) (SVT) (Ademosine6mg ApplyVFVT_GDL) (Intubate))
 ((Adult) (VR>150BPM) (Sedation) (Diltiasem)))
```

4 CONCLUSIONS

Applying inductive learning to automatically construct and update guidelines is a difficult task that has been proven possible in this work. First, a rule-based model for guideline representation has been defined that satisfies the properties of time and order-independence, branching, asserting, unfolding, state representation, and completeness. An inductive algorithm has been proposed and tested with two cardiac diseases. The tests are based on the at the moment available *incomplete* data. This explains some troubles with the CPGs obtained that will be solved with the use of more accurate training sets.

Although we have obtained interesting CPGs based on a real-life application domain, the work presented must be considered as part of a work in progress. So, there are still several improvements that must be achieved before the CPGs could be useful. These include extending the guidelines requirements with sequencing, timing, iterative and cyclic flows, and overcome the limitations of time-independence and order-independence.

This work has been developed within the research CICYT projects TIC2001-0633-C03 and TIC2003-07936.

REFERENCES

[1] Riaño D., 'Time-independent context-free clinical practice guidelines composition', Technical Report 04-02, DEIM, URV, (2004).

[2] Wang D., Peleg M., Tu S.W., and Boxwala A.A. et al., 'Representation primitives, process models and patient data in computer-interpretable CPGs', *Medical Informatics*, **68**(1-3), 59–70, (2002).

[3] Abston K. C. et al., 'Inducing practice guidelines from a hospital database', *JAMIA supplement*, 168–172, (1997).

[4] Mani S. et al., 'Two-stage machine learning model for guideline development', *AIMED*, (16), 51–80, (1999).

[5] Musen M.A. et al., 'EON: A component-based approach to automation of protocol-directed therapy', *JAMIA*, **3**(6), 367–388, (1996).

[6] Iman I.F. and Michalski R.S., 'Learning decision trees from decision rules: A method and initial results from a comparative study', *Intelligent Information Systems*, **2**, 279–304, (1993).

[7] *Guidelines for Clinical Practice: From Development to Use*, eds., Field M. J. and Lohr N.K., Washington: Nat Acad Press, 1992.

[8] Fox J. and Rahmanzadeh A., 'Disseminating medical knowledge: the proforma approach', *AIMED*, **14**, 157–181, (1998).

[9] Quinlan J.R., 'Inductive learning of decision trees', *Machine Learning*, **1**, 81–106, (1986).

[10] Peleg M., Boxwala A., and Ogunyemi O. GLIF3: The evolution of a guideline representation format, 2000.

[11] Peleg M. and Tu S. et al., 'Comparing computer-interpretable guideline models: A case-study approach', *JAMIA*, **10**(1), 52–68, (2003).

[12] Veloso M., Carbonell J., Pérez A., Borrajo D., Fink E., and Blythe J., 'Integrating planning and learning', *Exp and Theor AI*, **7**(1), (1995).

[13] Lavrac N., 'Selected techniques for data mining in medicine', *AIMED*, **16**, 3–23, (1999).

[14] Quaglini S., Stefanelli M., and Lanzola M. et al., 'Flexible guideline-based patient careflow systems', *AIMED*, **22**, 65–80, (2001).

[15] Shahar Y., Miksch S., and Johnson P., 'The Asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines', *AIMED*, **14**, 29–51, (1998).