Piloting Signal Processing Algorithms in a Cardiac Monitoring Context

François Portet, Guy Carrault, Marie-Odile Cordier, René Quiniou

Piloting Signal Processing Algorithms in a Cardiac Monitoring Context

F. Portet[1,2], G. Carrault[2], M.-O. Cordier[1], R. Quiniou[1]

[1] Institut de Recherche en Informatique et Systèmes Aléatoires
Campus de Beaulieu, 35042 Rennes, France
{francois.portet, marie-odile.cordier, rene.quiniou}@irisa.fr
[2] Laboratoire de Traitement du Signal et de l'Image, Unité INSERM 642
Campus de Beaulieu, 35042 Rennes, France
guy.carrault@univ-rennes1.fr

Abstract. This work is part of a project dedicated to the development of a cardiac arrhythmia monitoring system. The Calicot system is composed of two distinct parts: a temporal abstraction part, dedicated to the acquisition, the processing and the analysis of the signal, and a medical diagnosis part which computes a diagnosis from the data transmitted by the temporal abstraction. To increase the system performances, we decided to add to our system a pilot whose goal is to choose, according to the context, the type of information which is needed and the best way to get it from the signal. This pilot has three levels: the arrhythmia recognition pilot is in charge of determining the level of detail at which information is needed according to the current diagnosis hypotheses; the temporal abstraction pilot is in charge of (de)activating the abstraction tasks according to the signal characteristics and the recognition requirements; the signal processing pilot is in charge of deciding which algorithm is the most adapted to perform a selected abstraction task and tuning it according to the context and to the other pilots requirements. We present the new architecture of our system in which the pilot plays a central role. Then, the first results, mainly got at a signal processing level, are analyzed and we show the feasibility and the interest of such an approach.

1 Introduction

In medical monitoring, the reduction of false alarms and missed detections is a major objective. Medical monitoring systems, such as Guardian [1] and Calicot [2], are generally composed of two distinct parts: a temporal abstraction part, dedicated to the acquisition, the processing and the analysis of the signal, and a medical diagnosis part which computes a diagnosis from the data transmitted by the temporal abstraction and from a knowledge base. The temporal abstraction errors, even if limited, are transmitted to the arrhythmia recognition and degrade the diagnosis quality because of this false information transmission. It is thus important to select carefully the best suited signal processing algorithms depending on the context. Moreover, the diagnosis part itself does not always

need information with the same level of detail and it can be costly in terms of quality results to be too much demanding when it is not needed. It is thus also important to tune the temporal abstraction task according to the current diagnosis hypotheses. The aim of this work is to improve the cardiac monitoring system Calicot by adding a pilot whose goal is to choose, according to the context, the type of information which is needed and the best way to get it from the signal. This pilot has three levels : the arrhythmia recognition pilot is in charge of determining the level of detail at which information is needed according to the current diagnosis hypotheses ; the temporal abstraction pilot is in charge of activating or deactivated the abstraction tasks according to the signal characteristics and the recognition requirements; the signal processing pilot is in charge of deciding which algorithm is the most adapted to perform a selected abstraction task and tuning it according to the context and to the other pilots requirements.

After a brief description of the Calicot system in section 2, the new architecture and the pilot module are presented in section 3. In section 4, the results obtained in piloting QRS detection algorithms are analyzed. Then, related work are presented in section 5. This paper ends with a short discussion.

2 The Calicot monitoring system

Calicot [2] is a monitoring system devoted to cardiac arrhythmia recognition. Arrhythmias are heart diseases related to heart contraction dysfunction. Heart contractions are ensured by an electric stimulus which goes through the four hearts rooms: the two atria chambers and the two ventricles. Any disturbance in the course of this stimulus is called an arrhythmia. The stimulus can be measured by means of electrodes placed on the skin. The electrocardiogram (ECG) is used in clinical routine to obtain an image of the propagation of the electrical signal inside the heart. The main waves of the ECG signal are the P wave and the QRS complex. They correspond respectively to the depolarization of the atrias and the ventricles which induces the contraction of these chambers. An arrhythmia can be diagnosed from the morphology of the P and QRS waves and their temporal relationships.

Calicot computes a diagnosis from an abstracted representation of the ECG. This monitoring system is composed of two on line main modules (see Figure 1): a temporal abstraction module and a chronicle recognition module.

A chronicle [8] is a temporally constrained pattern which is characteristic of an arrhythmia. It is described by a set of events (in our case, P wave or QRS complex occurrence events) interlinked by time constraints. These chronicles are learned (off line) by an inductive logical programming method. Starting from annotated ECG examples related to an arrhythmia, the learning system produces recognition rules which can be easily translated into chronicle models. Thus, to each arrhythmia corresponds a few (at least one) chronicle models.

The temporal abstraction is achieved by signal processing algorithms (SP algorithms for short) that detect and classify the ECG events (QRS complex or P wave) from the ECG signal. The chronicle recognition module analyzes the

events flow and computes the diagnosis by searching for signal chunks matching the chronicles models.

The Calicot system demonstrated satisfactory performances [2]. However, the system remains sensitive to the temporal abstraction errors which can cause diagnosis errors. It is why we decided to add a pilot in charge of selecting, according to the signal and diagnosis context, the best adapted SP algorithms.

3 The new architecture

Calicot can be improved by piloting its processing chain according to the current context. The different ways to pilot Calicot and the new architecture are exposed. Then, the definition and the analysis of the current context are presented. Lastly, the pilot module and its knowledge bases are detailed.

3.1 Three ways to pilot Calicot

Calicot is piloted in three ways. The pilot selects the level of details that the arrhythmia recognition needs, activates and deactivates temporal abstraction tasks, and chooses and tunes SP algorithms. The architecture of the new system including the pilot is given in the bottom of Figure 1.

Arrhythmia recognition piloting: An arrhythmia can be diagnosed according to several ECG features. In Calicot, all the features are constantly extracted and sent to the arrhythmia recognition, but in some contexts a reduced number of features can be sufficient to recognize an arrhythmia. For example, in the presence of a fast heartbeat rate, one can be in presence of a ventricular tachycardia or a supra-ventricular tachycardia. Arrhythmia recognition based on P waves can discriminate them. But, the analysis of the QRS morphology, which is less time-consuming and more robust than the P wave analysis, is sufficient to discriminate the two arrhythmias. Thus, the arrhythmia recognition piloting consists in choosing the abstraction level of the chronicles to recognize by selecting the corresponding chronicle models, according to the current diagnosis hypotheses. To represent these various abstraction levels, a hierarchy of chronicle models (i.e. arrhythmia models) are learned from a set of examples expressed in the four following languages:

L1 includes the QRS occurrence date plus the temporal interval between QRS occurrence;
L2 adds to L1 the QRS morphology;
L3 adds to L1 the P wave occurrence date;
L4 adds to L2 the P wave occurrence date.

The four levels of chronicle models *C1, C2, C3 and C4* constitute the hierarchical chronicle base (4). Thus, for example, if the P wave is not needed then the chronicle base *C2* is chosen.

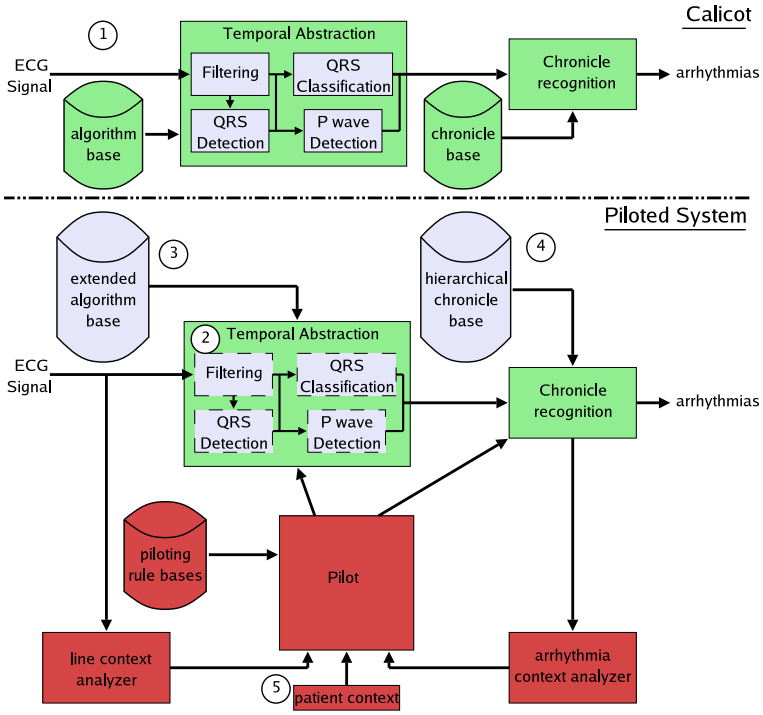

Fig. 1. General architecture of the system. The top chain represents the Calicot system and the bottom chain represents the modules added to Calicot to pilot it.

Temporal abstraction piloting: The temporal abstraction is composed of four linked tasks which extract three main features:

Filtering separates the actual ECG signal from the noisy part of the signal;
QRSDetection identifies QRS occurrence dates;
QRSClassification labels QRS morphologies;
PWaveDetection identifies P wave occurrence dates.

In Calicot (1), each task is always activated. But, if a chronicle base such as *C2* is chosen, then the *PWaveDetection* must be deactivated because it is not needed for arrhythmia recognition. Moreover, in several circumstances, some tasks cannot be achieved. For example, if the line is too noisy to accomplish *PWaveDetection* without errors then *PWaveDetection* must be deactivated. If not, this task penalizes the whole system because it provides false information to the chronicle recognition module. To be more efficient and to base the recognition on reliable information, the new architecture enables the activation and deactivation of the temporal abstraction tasks (2) according to the needs and to specific contexts.

SP Algorithms piloting: The temporal abstraction tasks are performed by SP algorithms. In Calicot, a unique SP algorithm is devoted to a particular task. However, in the literature, there exist several possible algorithms to achieved these tasks whose performances vary according to the context. Our preliminary study, described in [3], showed that the performances of various QRS detection algorithms change with the current context (line noise and QRS morphology). The new extended algorithm base (3) contains several SP algorithms for each task. For example, the *QRSDetection* task can be achieve by three algorithms:

pan : the Pan and Tompkins [10], a standard QRS detection algorithm;
gritzali : the Gritzali's detector [11], another standard algorithm;
*df*2 : the Okada's detector modified by Friesen *et al.*[9].

Thus, the role of the pilot is to choose the best suited algorithm according to the current context and then, to tune its parameters.

These three piloting levels interact in order to have consistent decisions. For example, the arrhythmia recognition pilot cannot select a type of chronicle models if the corresponding needed tasks cannot be activated by the temporal abstract tasks pilot.

3.2 The context and its analysis

The context (5) is composed of three sub-contexts: line context, arrhythmia context, and patient context. The patient context (age, basic ECG rhythm, etc.) is static whereas the line context and the arrhythmia context are dynamic and are regularly updated by the two analyzers:

The line context describes the level and the type of noise on the line at time n. From [3] three mutually exclusive noise types (*bw, ma, and em*) for three signal to noise ratios (5, -5, and -15 dB) are used. The *bw* noise (baseline wander) is mainly low frequency, the *ma* noise (muscle artifact) is mainly high frequency, and the *em* noise (electrode motion artifact) has components at high and low frequencies. The line can also be uncorrupted. The context takes a value among ten values:

$$\forall\ n \in \mathbf{N},\ \ ctxtline(n) \in (\{bw, ma, em\} \times \{5, -5, -15\})\ \cup\ \{no\_noise\}$$

The line context analyzer is connected directly to the input line, which enables a quick communication of the line context to the pilot. The pilot can thus modify the temporal abstraction before the ECG processing has begun.

The arrhythmia context: The arrhythmia context analyzer uses the chronicle recognition assumptions to make a list of the arrhythmias that are most likely to appear. This list is the arrhythmia context. From this arrhythmia context, the pilot can deduce the main ECG waveforms that the temporal abstraction have to process. For example, from the current arrhythmia

context, expert rules infers the main QRS waveforms, which are symbolized by the letters: N (normal beat), A (Atrial premature beat), R (Right bundle branch block beat), L (Left bundle branch block beat),J (Jonctional premature beat), F (Fusion of ventricular and normal beat), V (Premature ventricular contraction), P (Paced beat).

3.3 The pilot

The architecture of the pilot is illustrated by Figure 2.

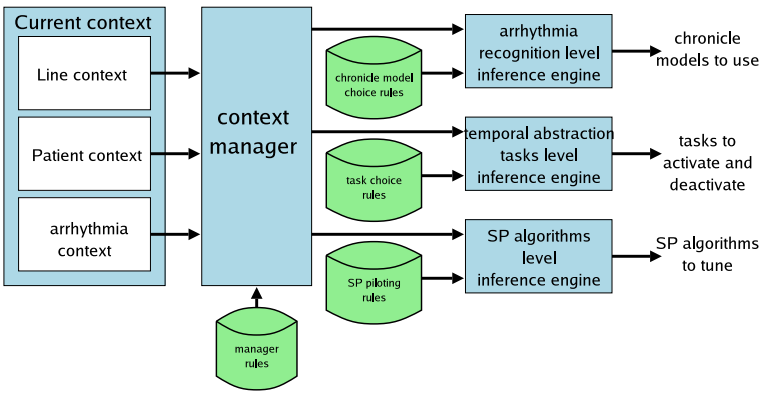

Fig. 2. The pilot architecture.

The pilot is composed of three inference engines which deduce the actions to perform on the system for the three piloting levels. The context manager deduces the information needed by the engines from the current context.

The context manager: The role of the context manager is to instantiate and update useful variables for piloting from the raw information transmitted by the context analyzers. For example, *tooMuchNoiseForPWave* is true only if the line is corrupted. Similarly, from the arrhythmia context, the context manager deduces the main QRS waveforms that will be processed by the temporal abstraction. In this sense, the context manager updates the fact base as in a classical expert system. Its knowledge is represented by expert rules stored as rules of thumbs in *the manager rule base*.

The inference engines: The system is piloted at three levels: the arrhythmia recognition level, the temporal abstraction tasks level, and the SP algorithms level. From the information transmitted by the context manager, the engines infer the actions to perform on the system. Their piloting rules are mainly defined

by an expert and are grouped into: chronicle model choice rules, tasks choice rules, and SP algorithms choice rules.

Chronicle model choice rules: The chronicle recognition adapts the abstraction level to the context. For example, if only the QRS occurrence date and QRS morphology are needed and technically achievable, then the chronicle recognizer must use the definition *C2*.

Tasks choice rules: The temporal abstraction tasks are activated according to the needs and to technical constraints. For example, to activate $PWaveDetection$, it is necessary to have a non disturbed line.

IF $<needPWaveDetection \wedge \neg tooMuchNoiseForPWave>$
THEN
$<start(PWaveDetection)>$

SP algorithms choice rules: The SP algorithms choice rules determines the best suited algorithm according to the temporal abstraction tasks and tunes it. For example, if the *QRSDetection* task is active, then it is necessary to choose the most suitable detector:

IF $<L \wedge bw \wedge SNR \geq -5dB>$
THEN
$<set(QRSDetection,changeQRS(gritzali,param(gritzali,ctxtligne)))>$

IF $<(L \vee F) \wedge no\_noise>$
THEN
$<set(QRSDetection,changeQRS(gritzali,param(gritzali,ctxtligne)))>$

IF $<(F \vee P) \wedge bw \wedge SNR \geq 0dB>$
THEN
$<set(QRSDetection,changeQRS(gritzali,param(gritzali,ctxtligne)))>$

IF $<em \wedge ((N \vee A \vee P \vee R) \wedge SNR = -15dB)>$
THEN
$<set(QRSDetection,changeQRS(df2,param(df2,ctxtligne)))>$

IF $<em \wedge (SNR = -5dB \wedge P)>$
THEN
$<set(QRSDetection,changeQRS(df2,param(df2,ctxtligne)))>$

IF $<default>$
THEN
$<set(QRSDetection,changeQRS(pan,param(pan,ctxtligne)))>$

The first rule says that if the line context has the value *bw noise at -5 dB* and the arrhythmia context informs that it has mainly QRS of form L, then the gritzali's detector is chosen. These rules are derived from [3] and are used only if the corresponding task is activated.

4 Results

In order to test the new piloted system, five ECGs related to different arrhythmias were extracted from the MIT-BIH database[1]. Each ECG lasted from 20 to 30 minutes and about 2 hours in all. Three to four different contexts are introduced in a test ECG in order to assess the system performances in the specific context as well as around the context transitions. Parts of the original ECGs were corrupted with real clinical noise of type *bw*, *ma* and *em*. The accuracy of arrhythmia recognition was assessed by evaluating the ability of the system to detect correctly all the QRSs appearing in the corrupted ECGs. Here follow some details about the ECGs, the kind of QRSs they contained and the type of noise that was added:

ECG_1: 1200 QRSs including 300 of form L without noise, 300 of form V with *bw* at -5dB, 300 of form P with *bw* at -5dB and 300 of form V with *bw* at -5dB
ECG_2: 1800 QRSs of form L including 300 with *ma* at -5dB, 300 without noise, 600 with *bw* at -5dB and 600 without noise
ECG_3: 1200 QRSs of form N including 300 without noise, 300 with *em* at -15dB, 300 without noise and 300 with *em* at -15dB
ECG_4: 1200 QRSs of form R including 300 without noise, 300 with *em* at -15dB, 300 without noise and 300 with *em* at -15dB
ECG_5: 1800 QRSs of form P whose 600 without noise, 600 with *em* at 5dB and, 600 with *em* at -5dB

Figure 3 displays some parts of ECG_2 showing the different contexts that were introduced (here different types of noise).

For any of these contexts, the pilot chooses the most adapted algorithm with the aid of the piloting rules. For each test, FN (the number of False Negatives – missed QRSs) and FP (False Positives – false alarms) are computed to obtain Ne (the Number of errors), $Ne = FP + FN$. The error rate is $Te = \frac{Ne}{N_{QRS}}$ where N_{QRS} is the total number of actual QRSs. In this study, the algorithm thresholds are optimal in the sense that Ne is minimum.

Table 1. Results of the QRS detection task with different detectors and with the pilot.

ECG	1	2	3	4	5	total	
score	Ne	Ne	Ne	Ne	Ne	Ne	Te(%)
pan	***20**	***91**	*240	*312	*367	**1030**	**14,3**
gritzali	**20**	*160	388	360	***295**	1223	17
*df*2	307	278	***174**	***160**	*302	1221	17
pilot	**20**	**88**	**185**	**167**	**304**	**764**	**10,6**

***** algorithms chosen by the pilot

The results of table 1 show that the best algorithm is different for each ECG. In the first case, *pan* and *gritzali* are the best for all the contexts, however

[1] http://ecg.mit.edu/

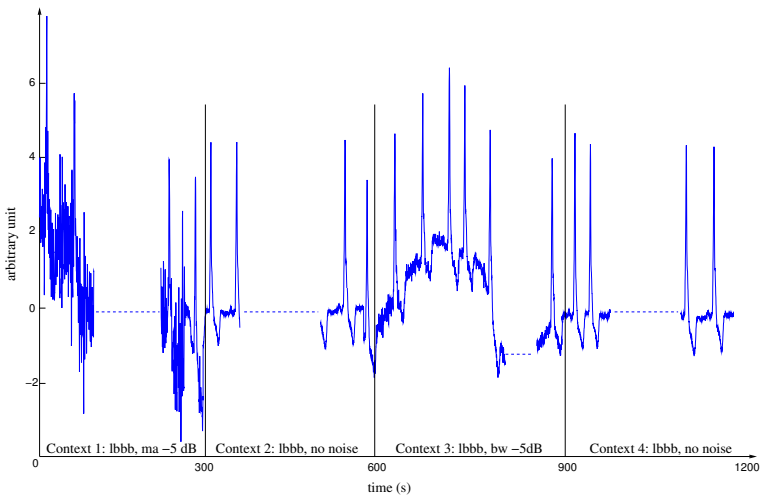

Fig. 3. ECG_2 is composed of four contexts.

according to the piloting rules only *pan* is used by the pilot. ECG_2 exhibits *ma* noise during 300 QRS for which *pan* is the most robust whereas *gritzali* is highly disrupted. But in the *bw* context, *gritzali* exhibits better performances than *pan*. That is why the pilot, by using both *pan* and *gritzali*, has the best results. In a similar way, the pilot uses *pan* and $df2$ for ECG_3 and ECG_4 which exhibit *em* context. In these cases, the pilot exhibits results close to those of $df2$, the best detector. This difference is due to the way the ECG is processed. In real-time processing, the signal is acquired by buffer. After filtering, the buffer is processed by the selected QRS detector. But, a buffer which contained two different contexts (context transition) is processed by only one QRS detector. For example, in ECG_3, the transition between the first context without noise and the second context with *em* noise is contained in one buffer and the pilot has selected *pan* (because of the first context) whereas $df2$ would be a best choice. That is why the pilot has a little bit more errors than $df2$.

The pilot obtains always a score very close to the best detector because it benefits, in each context, from the best algorithm performances. But, because of the nature of the piloting rules, the pilot may not have the best results. Actually, the rules express a general tendency (statistical tendency) and not an absolute behavior. For example, $df2$ is the best in *em* context but sometimes *pan* can be as good as or better than $df2$ (see [3]). However, a smart management of the input signal, as an adaptation of the buffer length to the context length, could improve the pilot results.

Even if the pilot is not always the best in every contexts, the total scores show that the pilot makes globally less errors than the best detector (*pan* here). This demonstrates the value of using a smart piloting of QRS detection algorithms according to a mixture of signal processing domain information (line noise con-

text) and medical information (arrhythmia and patient context), to improve the QRS detection complex which is the most important wave in ECG analysis.

5 Related Work

In [6, 7], Soulas *et al.* proposed a cardiac monitoring system which uses two QRS detection algorithms for temporal abstraction. According to the type of noise, the input signal is processed by the most suitable algorithm. But, the static architecture of this system cannot integrate new algorithms and no medical domain information is used. To modify dynamically the temporal abstraction, we propose to use an approach called *piloting algorithms* whose architecture permits to adapt on line the SP algorithms according to the current context.

Piloting algorithms derived from various work. Shekhar *et al.* [5] introduce the notion of *program supervision* which represents a signal processing task by a plan of primitive operations. A primitive operator achieves a simple goal and has initialization and adjustment methods. A task is represented by a request which is decomposed into simple goals. For each request, the operator plan is executed and then, adapted by an execution control module. But, even if it presents a good formalization of the signal processing application design, it does not merge the technical context (as the signal processing context) and the application domain context (as the medical context) to adjust the operators.

In [4], Karsai *et al.* proposes an architecture for on line self-adaptive software. The basic operators are represented by the nodes of a graph whose edges represent flows of signals. The plan (the graph connecting the operators) is executed under the control of a separate scheduler configured by a control graph. A complex operator is represented by the composition of primitive operators expressed by a graph. This complex operator contains several alternative connection graphs stored in the states of a finite state machine. State transitions are triggered by events (alarms) which induce the plan change, i.e. the complex operator architecture. This approach is well suited to systems whose complex operators are mostly independent to each other. But, for medical systems such as cardiac systems, any contextual information (signal and medical) can be used to adapt the SP algorithms. This high volume of information cannot be reasonably sent to every operators (loss of time, complexity, etc.), that is why a centralized architecture, which mixes signal and medical knowledge, was chosen for the SP algorithms piloting.

6 Discussion and prospects

We have proposed an approach for taking the context into account during signal processing. We propose to choose and adjust the most adapted SP algorithms according to a mixture of low level information (line context) and high-level information (arrhythmia and patient context). Intelligent monitoring systems are generally composed of a low-level part (temporal abstraction) related to the signal processing domain, and a high-level part (arrhythmia recognition)

more related to the artificial intelligent domain [1, 2]. However, the IA reasoning does not generally take into account the errors generated by the low level stage assuming it is only a specific signal processing problem. In the field of signal processing, some improvement have been done to include some kind of reasoning into the algorithms. For example, a rule such as " if nothing is detected during ten seconds then decrease the threshold" can be used. But, these solutions, event if they improve the SP algorithms, take only the local information into account as in [5] with local adjustment rules. In our study, general information as the basic patient rhythm or the current arrhythmia are also used. Using this kind of knowledge – the current context –, the average error rate of 14.3% obtained with the best QRS detection algorithm fell to 10.6% when a pilot was used.

These preliminary results are very encouraging and further work will focus on piloting the arrhythmia recognition and the temporal abstraction tasks to select the level of detail that recognition requires and to deactivate the tasks that are not necessary or that cannot be achieved without lot of errors. This piloting will ensure an more efficient arrhythmia recognition which uses always the reliable information for a robust medical diagnosis even in a noisy context.

References

1. Larsson, J.E., Hayes-Roth, B.: Guardian: An Intelligent Autonomous Agent for Medical Monitoring and Diagnosis. IEEE Intelligent Systems **13** (1998) 58–64
2. Carrault, G., Cordier, M.-O., Quiniou, R., Wang, F.: Temporal abstraction and Inductive Logic Programming for arrhythmia recognition from electrocardiograms. Artificial Intelligence in Medicine **28** (2003) 231–263
3. Portet, F., Hernández, A.I., Carrault, G.: Evaluation of real-time QRS detection algorithms in variable contexts. Medical & Biological Engineering & Computing (to appear in May issue)
4. Karsai, G., Sztipanovits, J.: A Model-Based Approach to Self-Adaptive Software. IEEE Intelligent systems. (1999) **14** 46–53
5. Shekhar, C., Moisan, S., Thonnat, M.: Towards an intelligent problem-solving environment for signal processing. Mathematics and Computers in Simulation. (1994) **36** 347–359
6. Soulas, T., Le Certen, G., Le Pichon, J.P., Carrault, G.: Algorithm switching in real time monitoring. Symposium on Electronics and Telecommunications (ETC) (1998) 145–149
7. Soulas, T., Le Certen, G., Le Pichon, J.P., Mabo, P. : Design of a software architecture for intelligent patient monitoring in critical care environments. 2nd European Medical and Biological Engineering Conference, Vienna (2002) 3.11–3.14.
8. Cordier, M.-O., Dousson, C.: Alarm driven monitoring based on chronicles. Safeprocess'2000 (2000) 286–291
9. Friesen, G.M., Jannett, T.C., Jadallah, M.A., Yates, S.L., Quint, S.R., Nagle, H.T.: A comparison of the noise sensitivity of nine QRS detection algorithms. IEEE Transactions on Biomedical Engineering (1990) **37** 85–98
10. Pan, J., Tompkins, W.J.: A real-time QRS detection algorithm. IEEE Transactions on Biomedical Engineering **BME-32(3)** (1985) 230–236
11. Gritzali, F.: Towards a generalized scheme for QRS detection in ECG waveforms. Signal Processing **15** (1988) 183-192