Strategies of Software Adaptation in Home Care Systems

Piotr Augustyniak[1]

AGH-University of Science and Technology, 30 Mickiewicza Ave., 30-059 Krakow, Poland august@agh.edu.pl

Summary. In this paper the control theory approach to the auto-adaptive telemedical ECG-based surveillance system is presented. The control is implemented as dynamic linking of alternative procedures and aims at optimization of diagnostic result quality. It faces several limitations in a discrete, nonlinear environment, where results cannot be reliably predicted. The paper presents also tests of the prototype implementation showing the feasibility of the idea as well as reasonable data convergence (80.7 %) and response time (17.1 s).

1 Introduction

Two trends are typical for current approach to medical software. First, the standardization, aims at using general-purpose tools and formats what facilitates data exchange and interoperability, but limits adequacy of tools in particular cases [4], [5]. Second, the customization, makes towards software- and, less frequently, hardware-based adaptation of the equipment functionality and algorithms in use to the recognized medical disease.

Following the latter paradigm, we recently proposed a remotely reconfigurable system for seamless cardiac surveillance [3]. This system combines advantages and eliminates drawbacks of two classic approaches: the data reliability is almost as high as in central interpretation-based system, while the load of transmission channel, and resulting operation cost is almost as low as in remote interpretation-based system.

Modification of interpretive software is performed in an unlimited-distance digital wireless feedback, and consequently all rules of control theory are applicable here. Such approach is presented in this paper with a particular consideration of non-linearity and discrete nature of agile software composed automatically on demand.

2 Components of the optimization loopback

2.1 General overview

Following a very typical scenario, the physicist with his experience and knowledge about the particular patient, decides what kind of data is necessary for making the diagnosis precise and accurate, and selects the diagnostic equipment upon necessity and availability. Certain initial patient data (e.g. sex, age or medication) determines the interpretive algorithms or sets normal-abnormal borderline values appropriately for each examined individual.

The auto-adaptive diagnostic process was designed and prototyped to perform by analogy to the scenario above. The remote ECG interpretation is *optimal* in the sense of best available reliability of diagnostic outcome. The optimization loopback (fig. 1) includes: remote interpretation procedures, quality estimation routine and adaptation manager.

The error function represents differences between remotely calculated diagnostic parameters and their reference values occasionally computed by the central server from the same strip of raw signal. The adaptation manager sends a multidimensional modification vector which is applied by the remote operating system to improve the ECG interpretation process. In result of few successive iterations, the adapted remote process issues diagnostic parameters converging to their references and consequently the error value achieves its temporal minimum.

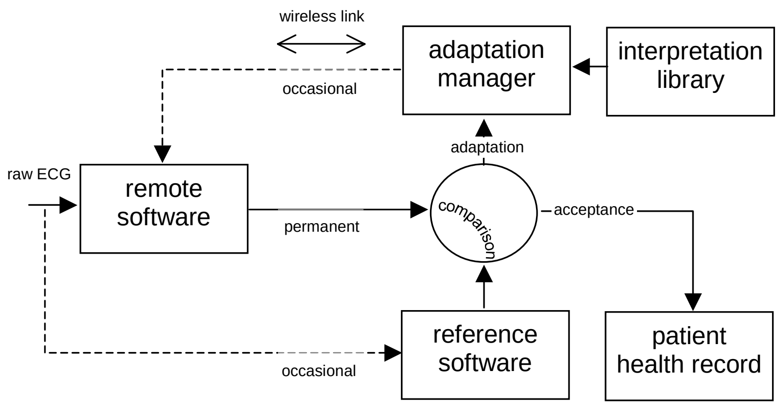

Fig. 1. General scheme of the auto-adaptive system for ubiquitous cardiology

The optimized signal interpretation process may suddenly become suboptimal in result of two factors external to the loopback:

- the patient status may change resulting in modification of relevance ranking of diagnostic parameters errors;
- the availability of resources (battery, memory, connection etc.) may change resulting in worse quality of issued diagnostic parameters;

If the value of error function returned by the quality estimation routine exceeds a specified threshold, information about non-compliant data (error vector) is sent to the adaptation manager (fig. 1).

2.2 Executable code as control argument

The adaptation manager includes modification vector generator and is an expert system for management of the interpretation subroutines. It works with a knowledge base consisting of task-oriented libraries corresponding to all replaceable blocks of the interpretive software (fig. 2a). Within libraries, the subroutines are designed for the same computational purpose, but with different prerequisites for resources requirements and result quality. They also have standardized *gateways* (fig. 2b) enabling replacement between functionally identical subroutines belonging to the same library in course of process of interpretive software optimization. In the generator's expert system each subroutine is described by attributes of quality, *resources requirements* and *external dependency attributes* specifying relations with other elements in signal interpretation tree. Additionally, interpretation subroutines have unified *modification interfaces* providing external access to few parameters used in calculations as factors or thresholds and considered as constants (read only) within the subroutine. Diagnostic data error and resources availability are principal arguments for the modification vector generator to propose variants of possible solution for optimization of the remote interpretive software.

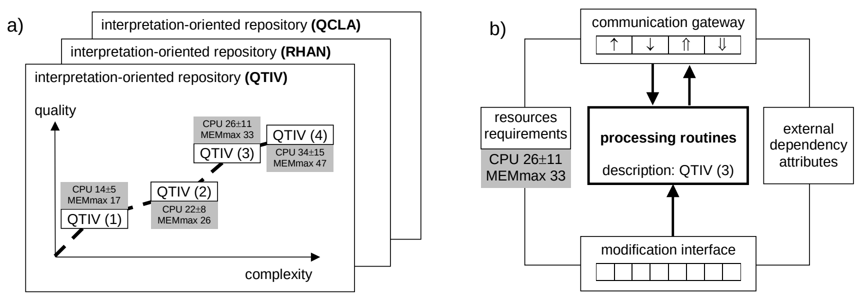

Fig. 2. a) Task-oriented libraries of interpretation subroutines; b) internal structure of replaceable interpretive subroutine

At the beginning, in the inter-procedure dependencies tree an interpretive procedure is identified as most probable common origin of all erroneous outcomes. The procedure influencing maximum number of non-compliant diagnostic data is the first target of software modification. In case when slight changes are required, the *modification interface* (fig. 2b) is used to adjust calculation parameters in working subroutine (software update). Otherwise the alternative subroutine is selected in the knowledge base and thanks to identical *gateways* (fig. 2b) replaces the precedent one in the interpretive software. In case new subroutine requires unavailable resources, the diagnostic parameter relevance ranking is scanned in order to find a procedure yielding medically irrelevant result. Such procedure, disregarding possibly high accuracy of result, is then removed or replaced by its simplest version, making more resources available for new subroutine. In case no irrelevant result could be found, the generator modifies calculation parameters, since this can be done without extra resources requirements. The generator's output is restricted to a set of calculation parameters with limited range of variability or to a subroutine selected as more appropriate from the knowledge space. In any of these cases, the control over the interpreting software is discrete and nonlinear. The effect of software modification can be only roughly predicted.

2.3 Interpreting software as control object

The interpreting software is designed to run in the wearable patient-side recorder, thus strict rules apply for computational complexity and management of resources. This software has *basic knowledge layer* (fig. 3) and elective interpretive procedures linked dynamically to their *communication gateways*. If a particular subroutine is linked, the procedure is enabled and returns results to the basic layer. Otherwise, the procedure is passed over and the result is absent in the report. In prevalence of cases, several procedures are mutually dependent and with use of *external dependency attributes* (fig. 2b), the modification vector generator loads and activates all such components simultaneously. The unlinking or replacement of the subroutine is restricted as long as the processing control (e.g. stack pointers etc.) remains within its executable code. In such cases the software management is performed immediately after the control returns to the calling subroutine.

The quality of automatic interpretation measured through diagnostic parameters requires the processing of the same section of the record to be performed by subsequent versions of agile software. Therefore the ECG has to be buffered in the remote recorder for the time interval sufficient for completion of assumed iteration number of interpretive software modification. The raw signal buffer is thus an important element of the recorder, and allows for seamless acquisition of new ECG while the software adaptation is performed. Fortunately, the automatic interpretation is sufficiently faster than the data acquisition (up to 5 times than real time), so the interpreter can securely keep up with the incoming signal, even if it spends a considerable amount of time on adaptation for a relatively short segment of the record. Nevertheless, buffering the incoming signal for delayed processing affects the recorder re-

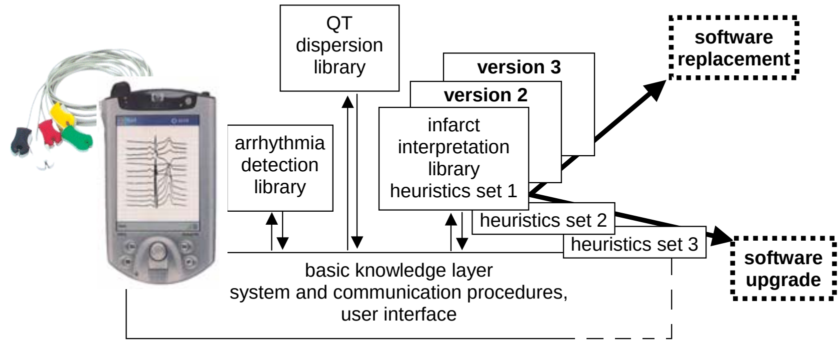

Fig. 3. Methods of control of remote software performance

sponse time. Its value may significantly excess the typical value (2 seconds) [6] what may expose the patient to danger when the adaptation occurs in presence of life-threatening cardiac events.

2.4 Diagnostic parameters quality as control goal

The purpose of the auto-adaptive interpretation is to provide most adequate diagnostic parameters of best possible quality in given patient status. A first-hand approach involves the human assistance for assessment of data quality and for specification of requests for software adaptation. Although computerized, interpretation of the ECG is not able to resolve all medical cases as reliably as the expert, aiming for a practical implementation we had to restrain the necessity of human intervention. Since resources availability is the principal limiting factor for the performance of home-care wearable recorders, we assume the complementary interpretation performed occasionally by the server in unconstrained environment yields diagnostic parameters which, although not absolutely true, are accurate enough to play the role of reference. For the purpose of validation, a strip of raw ECG record accompanies every 20-th or 30-th diagnostic report. Comparing the results of calculations in remote restricted environment with their unlimited counterparts yielded by the server is the first approach to fully automated adaptivity of the system.

This validation method, however, is expected to provide two results:

- quantitative assessment of the severity of remote diagnosis error,
- qualitative description specifying the area of possible improvement.

Although particular diagnostic parameters are not independent in the statistical sense, we defined the global estimate of diagnostic error as the absolute value of errors in the multidimensional space of parameters. To simulate reliably the doctor's assessment, the relevance ranking of diagnostic parameter errors in context of the patient status was applied to modulate the contribution of particular error values. This reflects various expectations the expert has from further diagnostics depending on what he or she already knows about the patient. The hierarchy of diagnostic parameters in context of most frequent diseases was revealed experimentally and described in [1]. This modulation is the crucial procedure of the auto-adaptive system, since it causes relocation of remote resources in order to calculate the most relevant diagnostic parameters with best accuracy and accordingly to the resources availability, the marginal parameters may show lower quality.

2.5 Convergence of the loopback control

As a general estimate of convergence quality, we proposed the value C being a weighted sum of relative errors of 12 most frequently used diagnostic parameters. Weighting coefficients are calculated as normalized relevance ranking results, calculated for diagnostic parameters in 17 most common patient diagnoses [3].

$$C = \sum_{i=1}^{12} \Delta p_i \cdot w_i \quad , where \sum_{i=1}^{12} w_i = 1 \tag{1}$$

The convergence represents the correctness of management procedure decisions about the software components of interpretation processing chain. Taking the analogy from the theory of control, the software adaptation plays the role of feedback correcting the diagnoses made automatically. If the software modification decisions are correct, the outcome altered by the interpreting software modification approaches to the reference value, the modification request signal extincts in consequence of decreasing error and the system is stable. Incorrect decisions lead to the growth of diagnostic outcome error and imply even stronger request for modification. The outcome value may stabilize on an incorrect value or swing the control range in response to subsequent trials. In such case the system is unstable and the only practical solution is relying on server-side calculations as long as the interpretation task is too difficult to be performed by the patient-side recorder.

3 Results

For the testing purpose, the remote recorder has been implemented in a development kit of the PXA270 microprocessor with a widely updated XScale core, which is Marvell's implementation of the fifth generation of the ARM architecture [7]. It provides excellent MIPs/mW ratio of 4.625 at 150 MHz and is currently used in a series of handheld computers. The agile software of remote recorder uses 9 adaptive interpreting procedures, for which the modification vector generator selects one of 2 to 5 specialized subroutines from

Table 1. Inventory of interpreting procedures with respective CPU/memory requirements [%]/[kB]

procedure name	version number				
	1	2	3	4	5
heartbeat detector	7/25	8/32	10/38	15/42	16/48
heartbeat classifier	15/18	17/26	20/35		
wave delimitation procedure	35/22	65/26			
ST-segment assessment	10/10	12/17	13/25		
arrhythmia detector	5/7	8/10	13/12	19/15	
heart rate variability analyzer	30/25	51/38	68/44		
electrical axes calculator	7/8	27/21			
rhythm identification procedure	5/12	6/18	8/26		
QT-segment assessment	13/20	17/31	33/35		

the knowledge base. Table 1 presents these subroutines with the memory and calculation requirements of their versions.

The relevance ranking of diagnostic parameters errors was studied and presented in details in [1]. That survey included 1730 ECG analysis cases and allowed to pursuit the cardiologists' preferences in 12 most frequently observed diseases (normal sinus rhythm, sinus tachycardia, sinus bradycardia, probable AV block, ventricular escape beats, atrial fibrillation, AV conduction defect, myocardial infarction, atrial enlargements, ventricular hypertrophies, left and right bundle branch block). The observations count for these pathologies ranged from 16 to 323 cases.

The rules of software adaptation for the telemedical system was first modeled in order to study the influence of medical and technical dependencies to the diagnostic result quality [2]. For simulation of rapid patient status changes, ECG test signals representing various pathologies and transients were artificially combined from custom-recorded signals.

The total of 2751 one-hour 12-leads ECG records were processed off-line in the prototype system. In case of 857 records (31,2%) the software adaptation was required, next 86 records (3,1%) were found too complicated and interpreted by the server software. Among the software adaptation attempts, 768 (89,6%) were correct, while the remaining 10,4% failed due to incorrect estimation of available resources. The overestimation of resources, resulting in the operating system crash and thus monitoring discontinuity occurred in 27 (1%) cases. Single iteration was sufficient to modify the remote interpretation enough for satisfying the data consistency requirement in 63.1 % of cases. In 19.3% of cases, after four iterations the results were still not converging to the reference values. The adaptation delay measurement was performed with use of a real wireless GPRS connection. The longest value was 6.0 s in case of single iteration and 17.1 s for four iterations.

4 Discussion

Strategies of automatic adaptation of ECG interpretive software in a distributed surveillance system were presented in aspect of control theory. The patient device was build in a test environment designed for prototyping of mobile applications. Values of basic parameters: data convergence (80.7 %) and response time (17.1 s) motivate the hope for future clinical application of the system.

Main reason for limitation of auto-adaptive approach is the discrete control domain (some procedures existed in only two versions) and the lack of results predictability. This could be solved by the application of artificial intelligence-based system in the central server. The analysis of modification results (e.g. data convergence) yields indications for optimal system behavior in presence of similar cases.

The auto-adaptive ECG-based surveillance system is a promising tool for simulation of continuous presence of medical expert at the patient in motion.

5 Acknowledgment

Scientific work supported by the AGH-University of Science and Technology grant No 10.10.120.783.

References

1. Augustyniak P. (2008) How a Human Ranks the ECG Diagnostic Parameters: The Pursuit of Experts' Preferences Based on a Hidden Poll, Proc. Computers in Cardiology; 35, 449-452,
2. Augustyniak P. (2008) Modeling the adaptive telemedical system with continuous data-dependent quality control, Polish Journal of Environmental Studies; 17 no. 2A, 29-32,
3. Augustyniak P. (2008) Diagnostic quality-derived patient-oriented optimization of ECG interpretation, [in:] Ewa Pietka, Jacek Kawa (eds.) Information technologies in biomedicine, Berlin ; Heidelberg : Springer-Verlag, 243-250
4. Chiarugi F. et al. (2002). Real-time Cardiac Monitoring over a Regional Health Network: Preliminary Results from Initial Field Testing Computers in Cardiology 29, 347-350.
5. Gouaux F., et al. (2002). Ambient Intelligence and Pervasive Systems for the Monitoring of Citizens at Cardiac Risk: New Solutions from the EPI-MEDICS Project Computers in Cardiology 29, 289-292.
6. IEC (2003) 60601-2-51. Medical electrical equipment: Particular requirements for the safety, including essential performance, of ambulatory electrocardiographic systems. First edition 2003-02, International Electrotechnical Commission, Geneva,
7. (2007) *http://www.toradex.com/e/Factsheet_Colibri_Intel_Marvell_XScale_PXA_Computer_Modules.php* (visited on March, 31, 2008)