Design of an Intelligent Intensive Care Unit Alarm System

Stephen Schreiber Aaes* Mikkel Sandberg Andersen* Tommy Bitsch* Sanne Christensen*
Tommy Jensen* Rikke Ottesen,* Jan Tvorup*

*Aalborg University
Department of Health, Science and Technology

***Abstract*— The Intensive Care Unit (ICU) accommodates the most severely ill patients in a hospital. These patients are monitored by several medical devices, e.g. ventilators and gas monitors. All of these devices function autonomously giving individual alarms. As a result alarms are often triggered unnecessarily, e.g. in the event of a probe failing, even though another device may measure that the patients condition is unchanged. This article investigates the possibility of processing alarms intelligently through the development of a system, that is capable of evaluating alarms based on data from several medical monitoring devices. This involves detecting equipment failure alarms, trend alarms, and threshold alarms. Furthermore, it is desirable for the hospital staff to be able to monitor the patient from any location in the hospital.**

The emphasis of this article is on developing a monitoring system, providing a network of distributed data collection units, and the ability to monitor measurements and alarms from any given location within the network. The system is comprised of three parts: A bedside unit for each patient, multiple independent monitoring units, and a central data repository. The bedside unit collects data from medical devices monitoring the patient. This data is distributed via TCP/IP to all connected monitoring units and the central data repository. The monitoring unit has three central components: The discovery, the graphical user interface (GUI), and the alarm component. When started, available bedside units are discovered through multicasting. When connected to a bedside unit the monitoring unit displays the collected data and alarms. The central data repository stores measured data from the medical devices connected to the bedside units. The resulting design has proven successful in evaluating alarms based on data from several medical monitoring devices. Test cases show that the design can differentiate the three different alarm types. The design has also proven successful in distributing the measured data from one bedside unit to multiple monitoring units.

This article demonstrates the possiblity of generating intelligent alarms in an ICU by implementing a distributed system, with a bedside unit for each patient and a self-contained monitoring unit. The flexibility of the system can be expanded considerably by implementing XML as the data exchange format between nodes in the system. Furthermore, more advanced alarm types, e.g. decision support systems, can be implemented.

***Index Terms*— Intensive Care Unit, Network, Relational Databases, Distributed System, Multicast, Monitoring system, Java.**

I. Introduction

The Intensive Care Unit (ICU) in hospitals accommodates the most severely ill patients. Because of their condition, patients in these units are monitored carefully using several monitoring devices. Furthermore, a nurse is always present at bedside to respond to alarms. As alarms are not coordinated with respect to measurements by other devices which collects and analyses data autonomously they can be unnecessary.

An unnecessary alarm may occur when a pulse-oxymeter probe fails, an alarm will be given signaling a cardiac arrest, even though another device may measure that the patient is still breathing and blood pressure is within acceptable bounds. With each patient being monitored by several devices these equipment failure alarms are more likely to occur. In one study, alarms in an ICU ward were logged over a duration of 3 weeks. Out of 1455 alarms only 8 alarms were classed as indicating a potentially serious problem. This points out that “a high false alarm rate may distract the attendants attention from an important alarm, as well as making unnecessary noise close to the patient” [1]. Furthermore it is not possible to detect a worsening condition of a patient, e.g. by the use of trend alarms. Making use of these would enable the staff to react on a worsening condition.
An example of a set-up in a thoracic ICU ward can be seen in figure 1.

The aim of this study is to investigate the possibility of processing alarms more intelligently through the development of a system that evaluates alarms based on

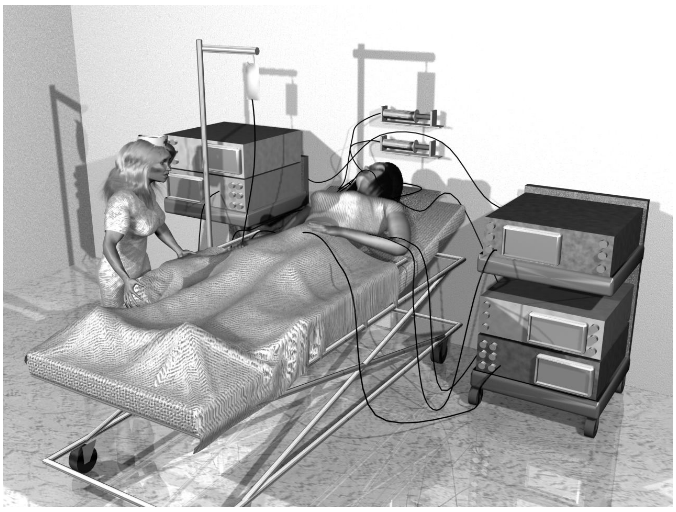

Fig. 1. A set-up in a thoracic ICU ward.

measured data from all monitoring devices connected to the patient. Additionally, monitoring the patient data from a remote location is desirable. This enables the doctor to keep up to date on a patient's condition, without being able to be physically present on the ward. This entails providing the possibility of viewing data simultaneously from several different locations. Due to the fact that the system involves a processing unit by each bed, it can be designed as a distributed system.

In the system, intelligent alarms involve detection of three alarm types. Threshold alarms, which are often acute alarms, are triggered when a threshold value is exceeded. Equipment alarms are triggered when a probe or a device fails, while data from other devices indicate that the patients condition is unaltered. Trend alarms are triggered when the patients condition has worsened over a period of time. To observe a patients condition over a period of time, collected data must be stored. For each patient and device setup, alarms should be customisable due to the fact that different patient conditions require different threshold settings and device setups. For ease of configuration of device-setup, it is desirable to be able to load a profile, defining standard threshold-, and trend-alarm settings for a given patient condition. Furthermore, the system must provide for saving a customised profile for a specific patient. To prove the concept of this study two standard profiles are used: Coronary Artery Bypass Grafting (CABG) and Chronic Obstructive Lung Disease (COLD).

A detailed description of the methods for designing the system will be given in the following section.

II. Methods

A. System overview

The background knowledge for this article has been researched by interviewing a registered nurse at the Thorax Intensive Care Unit (ICU), Aalborg Sygehus Syd and the ICU chief physician at Aalborg Sygehus Nord.

The emphasis of this article is on developing a monitoring system, providing a network of distributed data collection units, the bedside units, and the ability to monitor measurements and alarms, the monitoring units, from any given location within the network.

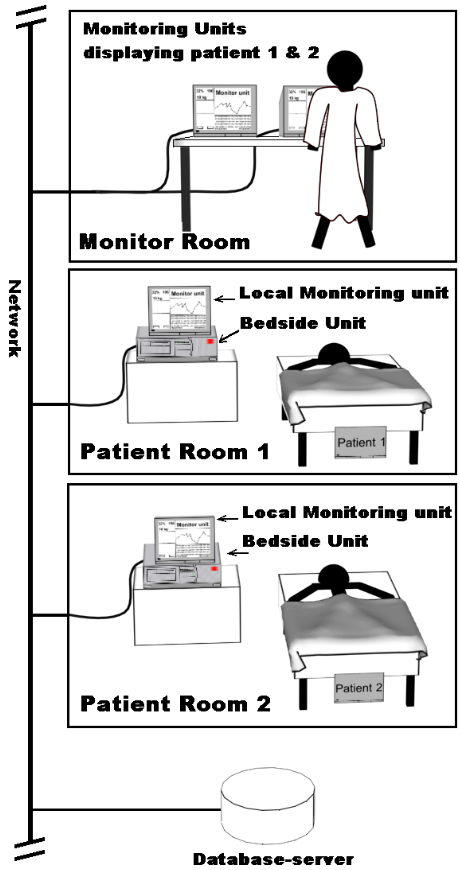

Fig. 2. Overview of the system.

The Intelligent Intensive Care Unit Alarm system (IICUA) system is comprised of three parts: A bedside unit for each patient, multiple independent monitoring units, and a central data repository. All parts communicates through an Ethernet network. The system is illustrated in figure 2. The bedside unit collects data from medical devices monitoring the patient. This data is distributed via TCP/IP to all connected monitoring units and the central data repository. The monitoring unit has three central components. The network communication unit, which detects the available bedside units through multicasting and transfers measured data. The alarm, which determines whether an alarm should be triggered. The graphical user interface (GUI) unit, which displays data and alarms. When connected to a bedside unit the monitoring unit displays the collected data. Furthermore it displays the three alarm types, listed by priority in descending order: threshold-, equipment-, and trend-alarms. The central data repository consists of a database in which patient profiles and measured data from the medical devices connected to the bedside units is stored for later clinical use.

In the following sections the details of the system will be described.

B. Bedside unit

The purpose of the bedside unit is to acquire continuous medical data from the patient and in real time distribute this data to different targets. These targets are one or more monitoring units, which displays the data and processes alarms, and the central data repository which records the data. Additionally the bedside unit calculates trend alarms. The trend alarms calculations are placed in the bedside unit to minimize load on the monitoring unit. The bedside unit runs on a computer connected to the ICU-equipment for each patient. There is one bedside unit for each patient, each patient has a personal ID, and a bedside unit can handle only one patient at a time. When a bedside unit is started, the user is prompted for a patient ID (PID) as to identify the unit on the network, and as an ID when storing medical data in the central data repository.
The bedside unit is implemented using Java.

A class diagram of the bedside unit can be seen in figure 3.

1) Device manager: The device manager is the class that handles the connection to the ICU-equipment. The purpose of the device manager is to act as a transitional layer between the ICU-equipment and the rest of the bedside unit. The device manager handles specific drivers for each medical device connected to the system. The current system uses modified versions of the device drivers for the Brüel & Kjær 1304 and the Siemens Servo 300 developed in the ICUMatic project [2].

The class Device is a superclass for all device drivers, providing generic functions like connect and disconnect. All device drivers must extend this class in order to be recognized by the device manager. The device manager controls all instances of device drivers, allowing these to update the measurements in the datastructure called Patient, see section II-D.

2) Connection pool: The connection pool is the part of the bedside unit that handles all network communication to and from the bedside unit. This includes the multicast discovery and the distribution of the data acquired through the device manager to connected external devices, such as the monitoring unit. The network communication is accomplished through sending serialized objects between nodes. The communication is described in details in section II-D. When the ConnectionPool class is instantiated, a thread handling the multicast discovery is created. When a TCP/IP connection is requested a server-thread is created, listening for connections on a port. After establishing TCP/IP connection, the server-thread will create a separate thread for this connection as not to block the server for further connections. A runtime example of the bedside unit can be seen in figure 4.

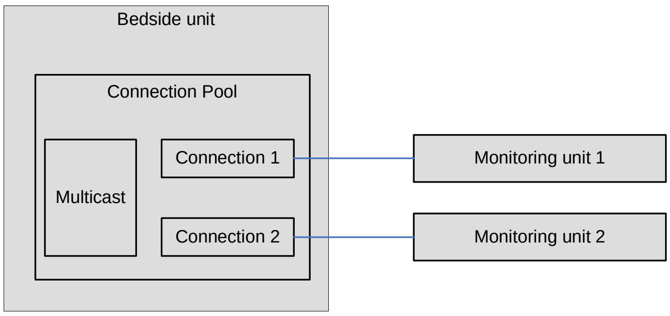

Fig. 4. A runtime example of the processes in the connection pool with two monitoring units connected.

3) Trend alarms: The trend alarms are based on the history of a patients condition. The trend-alarms requires templates in order for them to work. These templates are recorded signals of the patient in a stable condition. Through the GUI the user will be prompted to record these templates. When the user requests a trend-template to be recorded the measured signal is processed and stored as a template in the central data repository. This enables the alarm handler in the monitoring unit to execute trend alarm calculations with a specified time interval.

The trend alarms are implemented using cross correlation [3]. Signals are said to be correlated if the shapes of the waveforms of the to signals $(x(n), y(n))$ are correlated as defined by the cross correlation coefficient as defined by equation 1.

$$\sigma_{XY} = \frac{\frac{1}{N}\sum_{n=1}^{N} x(n)y(n)}{\sqrt{\frac{1}{N}\sum_{n=1}^{N} x(n)^2}\sqrt{\frac{1}{N}\sum_{n=1}^{N} y(n)^2}} \tag{1}$$

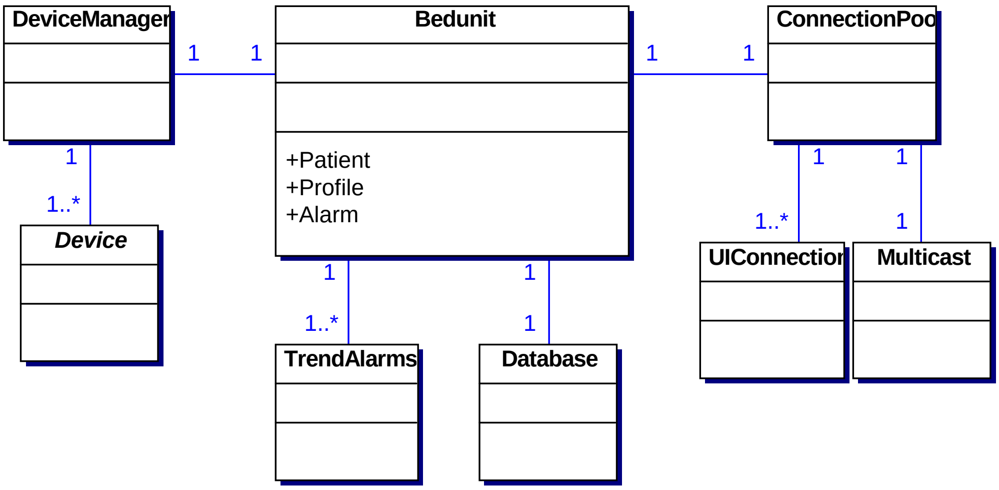

Fig. 3. A class diagram describing the structure of the bedside unit.

The cross correlation is implemented using a template signal recorded from the patient when the patient is stable. The template signal is used as reference signal to all incoming signals. A flow diagram for obtaining the template is seen in figure 5.

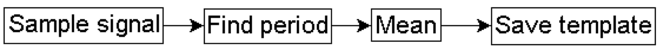

Fig. 5. Flowdiagram showing how the template is obtained.

In order to determine the cross correlation coefficient of the two signals, they must be aligned. The alignment is performed by finding the period of the signals and time-shift the signals. The periods of the signals are calculated using autocorrelation. The autocorrelation is defined by:

$$R_{xx} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x(n) * x(n+k)} \qquad (2)$$

Using cross correlation on signals with same frequency but different amplitudes results in the signals being correlated. This problem is compensated for by calculating the mean value of the difference in amplitude between all samples. Comparing this figure to the highest amplitude gives an estimate as to how well the signals match. The priority level of the trend alarms is 1.

A flow diagram for the cross correlation is seen in figure II-B.3.

C. Monitoring unit

The monitoring unit has three main functions: Displaying the patient data, triggering alarms, and disabling alarms. There can be several monitoring units connected

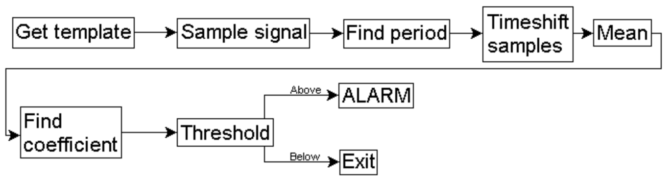

Fig. 6. Flowdiagram showing the cross correlation.

to a given bedside unit. This means that a monitoring unit can show the data for a patient in the same room as the patient. At the same time there can be another monitoring unit, at another location within the hospital, connected to the same bedside unit, showing the same data, see figure 2 in section II-A. The monitoring unit runs on a computer connected to the same network as the bedside unit. The advantage of this separation between the bedside unit, that gathers the data, and the monitoring unit that displays it, is that it is possible for a doctor to view his patient from anywhere in the hospital that is connected the network.
The monitoring device is implemented using Java.

A class diagram describing the structure of the monitoring unit can be seen in fig. 7.

A monitoring unit is not as a standalone program. In order for it to perform its function, it needs to be connected to a bedside unit, providing it with data from a patient. When connected to a bedside unit the user will be prompted to do one of two things: Select a standard profile or a customized profile for the patient. The latter will only be available if a customized profile has been created. The adjustment of a profile is achieved by retrieving the patients personal profile from the database and modifying the values. The profile is described in the

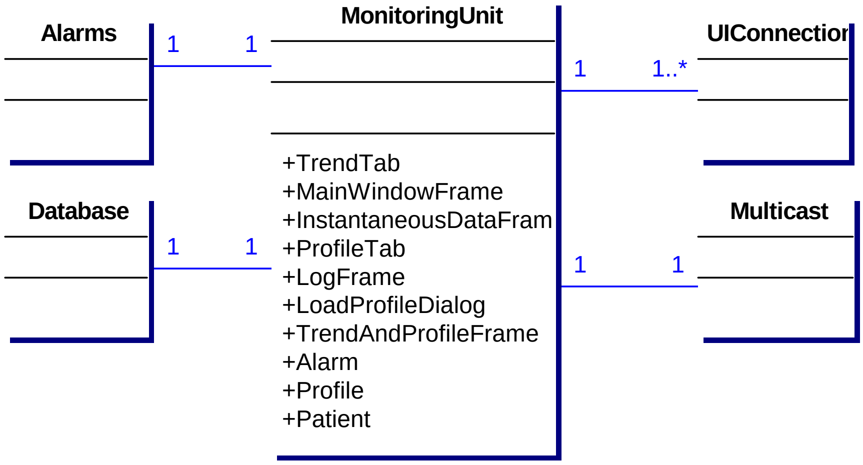

Fig. 7. A class diagram describing the structure of the monitoring unit.

extended markup language (XML).

1) GUI: The GUI displays data received from the bedside unit, and alarms. A screenshot can be seen in fig. 8.

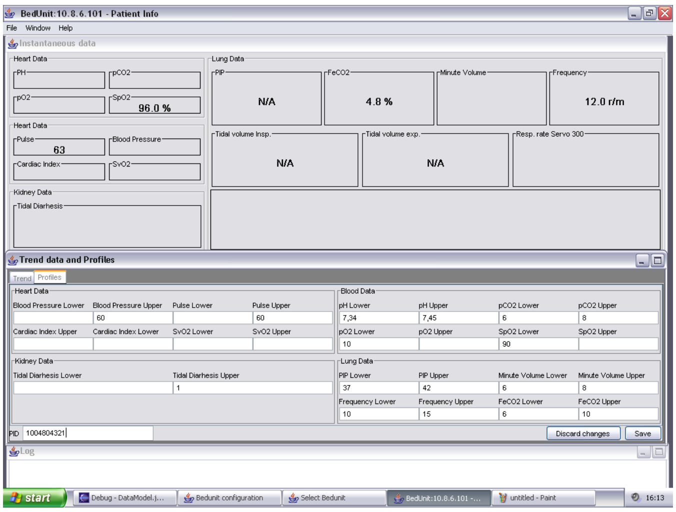

Fig. 8. A screenshot of the monitoring unit.

The GUI consists of three internal frames. The upper frame, instantaneous, in which alarms and the continuous measurements from the bedside unit are displayed. Alarms are displayed as a flashing frame with a color representing the alarm type, red for threshold alarms, orange for trend alarms, and black for equipment alarms. The middle frame, trendgraph and profile editing, in which graphs displaying the trends for a number of measurements, and the current profile settings can be displayed. The lower frame, log, in which detailed information about alarms are printed, e.g. alarm type and the time the alarm was triggered.

2) Alarm handler: The purpose of the alarm-handler is to evaluate the data it is receiving from the network communication, and trigger an alarm if any alarm criteria is met. For each patient there is an individual profile saved, the values in this having been adjusted by the user. When the alarm handler is initialized it determines whether a profile has been created for the patient. If no profile exists the user is prompted to create one via the GUI. Both profiles containing the threshold limits and the equipment alarm definitions are defined in XML.

The following XML example shows an equipment alarm triggered when the pulse oxymeter fails while respiration persists.

```
<alarm>
  <type>equipment</type>
  <name>pulse oxymeter failure</name>
  <condition>
    <data>
      <lost>pulse</lost>
      <exists>respiration</exists>
    </data>
  </condition>
  <fails>pulse</fails>
</alarm>
```

The following XML example shows an excerpt of a profile for a patient with patient ID (PID) 100479-3165.

```
<profile time_stamp=
"Thu Dec 04 12:42:11 CET 2003">
   <pid>1004793165</pid>
      <type>COLD</type>
      <heart>
         <bpmap>
            <lower>63</lower>
            <upper>70</upper>
         </bpmap>
         <pulse>
            <lower>30</lower>
            <upper>90</upper>
```

```
        </pulse>
      </heart>
</profile>
```

When the criteria for a threshold alarm is met. The alarm handler determines if the alarm should be classified as an equipment alarm by evaluating all available equipment alarm definitions. If none of these are met, the alarm is classified as a threshold alarm.

3) Alarm types: The system is implemented with three types of alarms, threshold-, equipment- and trend-alarms. Each alarm has its own alarm level according to the priority of the alarm. The priority levels decides in which order alarms are processed in the alarm handler and are based on an interpretation of the importancy of the alarms. The priorities are denoted with priority 1 as the lowest priority.

Threshold alarms The threshold alarms are based on the threshold values in a profile. The threshold values are defined in a XML document. The priority level of the threshold alarms is 3.

Equipment alarms The equipment alarms are based on logical operations performed on the acute alarms. It can be assumed that an equipment alarm should occur if the blood pressure is constant but the patient has no pulse. The priority level of the equipment alarms is 2.

4) Network communication: The purpose of the network communication is to enable the communications described in section II-D. To satisfy this, the monitoring unit, when started, establishes a multicast thread. When a bedside unit acknowledges a data transfer connection the monitoring unit creates a thread handling the TCP/IP connection.

D. Communication

The network communication in the system can be classified as either multicasting- or data transfer-communication.

1) Multicasting: To provide communication between the bedside unit and the monitoring unit in the discovery phase, multicasting [4] has been chosen. For this a simple protocol has been developed. An example of a monitoring device being connected to the network and requesting connection with a bedside unit can be seen in figure 9.

The data transmitted via multicast is encapsulated in the datastructure shown in fig. 10.

When a monitoring unit is initialized it immediately starts searching for bedside units. This is done by sending an initialization [INIT] message to the multicast group with no recipient address. When receiving this message a bedside unit will issue an initialization ok [INIT_OK] message to the multicast group, to indicate its presence on the network and that it is ready to receive a data transfer connection. To ensure that the reply is detected by the sender of the initialization message the bedside unit specifies the recipient address. Additionally the fields describing the bedside units name, number of current connections and the patients PID is filled in. When establishing a connection between a monitoring unit and a bedside unit, the monitoring unit will issue a connection message [CONNECT] with the address of the bedside unit specified. Upon reception of this message the bedside unit will issue a connection ok

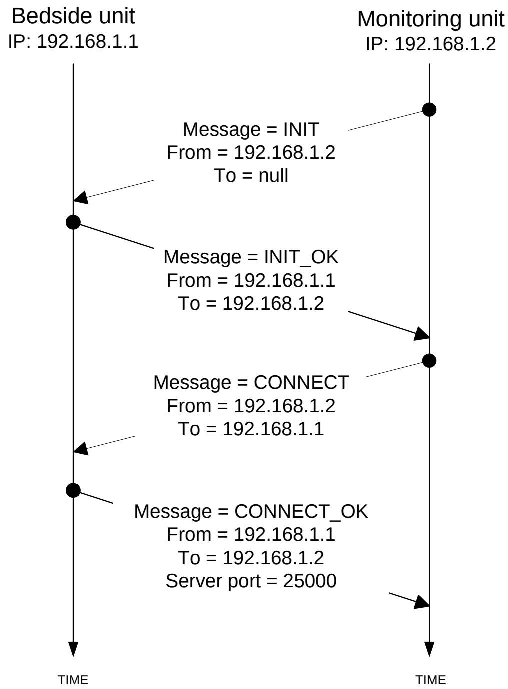

Fig. 9. The multicast connection sequence between a monitoring unit and a bedside unit.

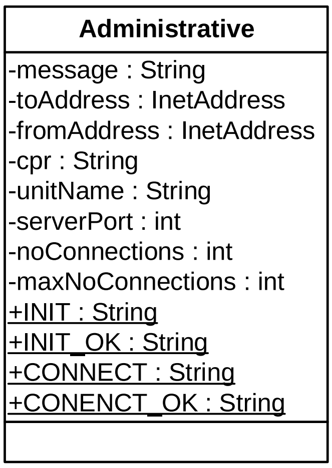

Fig. 10. The datastructure used for multicast discovery communication.

message [CONNECT_OK] and specify a serverport for the monitoring unit to connect to.

2) TCP/IP connection: To provide data transfer communication between a bedside unit and a monitoring unit the datastructure shown in figure 11.

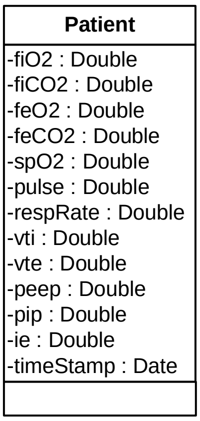

Fig. 11. The datastructure used for TCP/IP data transfer communication.

E. Central data repository

The central data repository is implemented as a database which stores the information of each patient centrally. The data saved on the server are all collected data for all patients in the system. The data are saved for a predefined period of time after it is recorded. This data can be used in retrospective clinical applications. Each patients customized profile and customized alarm are also saved on the server, as well as all the different pre-customized profile-types and the default profiles. By having all the patient-data centrally on a server the system enables the user to freely move the monitoring unit to any location without having to reconfigure the customized profiles. Additionally, storing all patient data centrally will enable the user to exchange any bedside unit without loosing recorded data, e.g. in the case of a defective bedside unit.

The database is implemented by a MySQL server.

III. Testing the Final Design

To evaluate the system the following four test cases were deployed.

A. Connection, display and distribution

Purpose: To verify the functionality of the network discovery and -communication and to verify that multiple monitoring units can connect to a single bedside unit.

Setup: The test setup was comprised by a Brüel & Kjær 1304 Anesthetic Gas Monitor (B&K1304) and a Siemens Servo 300 ventilator (Servo 300) connected via RS232 to a PC running the bedside unit software and the monitoring unit software (PC 1). Additionally a separate computer with the monitoring unit software was used (PC 2). During the test a test person was wearing a pulse-oxymeter probe and a mask connected to the Servo 300 with a capnograph lead to the B&K1304.

Procedure: The bedside unit software was started on PC 1, when ready the monitoring unit software was started on PC 1. It was observed if the monitoring unit listed the available bedside unit. If the unit was listed, a connection was made to it. It was observed whether the monitoring unit displayed correct data collected from the two connected medical devices. To test with multiple monitoring units, PC 2 with the monitoring unit software was connected to the bedside unit running on PC 1. It was observed whether the correct data was displayed. The test would be satisfied if the network discovery and -communication functioned correctly, and that several monitoring units could connect to a single bedunit.

Results: It was observed that the monitoring unit listed the available bedside units after having been intialised and that a connection to the bedside unit was made. Furthermore it was observed that the data was obtained correctly by the monitoring unit and was shown in the appropriate areas of the graphical user interface. When connecting more than one monitoring unit the data was successfully distributed to all connected monitoring devices.

B. Threshold alarm

Purpose: To verify that exceeded threshold limits as set in a patient profile are detected by the system. Secondarily to verify that alarms are displayed correctly on the monitoring unit.

Setup: A B&K, 1304 connected to a PC running both a bedside unit and a monitoring unit. During the test a test person was wearing a pulse-oxymeterprobe and a mask with at capnograph lead.

Procedure: The test person hyperventilated to raise the respiratory frequency above 16 breaths/min. It was observed whether a threshold alarm was displayed on the monitoring unit. The test would be satisfied if, when the respiratory frequency was above 16 breaths/min, a threshold alarm would be displayed on the monitoring unit.

Results: It was observed that the system triggered and displayed a threshold alarm when the test person was hyperventilating, raising the respiratory frequency above 16 breaths/min.

C. Equipment alarm

Purpose: To verify that the system is able to detect an equipment failure.

Setup: A B&K1304 connected to a pc running both a beside unit and a monitoring unit. During the test a test person was wearing a pulse-oxymeterprobe and a mask with at capnograph lead.

Procedure: The pulse-oxymeter probe was disconnected. It was observed whether an equipment alarm was displayed on the monitoring unit. The test would be satisfied if an equipment failure alarm would be displayed in the pulse field on the monitoring unit.

Results: It was observed that the system detected a disconnected probe and that an alarm was shown on the monitoring unit.

D. Trend alarm

Purpose: To verify that a trend alarm is given at the correct time and that the data is analysed correctly.

Setup: This test was performed on simulated data.

Procedure: Simulated data of a patient in a worsening condition was used. It was observed how the trend alarm reacted on this data. For this test the trend-alarm was modified to output the internal results of the analysis of the signal. These results could be held up with the parameters of the simulated data. The test was satisfied if trend-alarm found the period for any given signal and that it was capable of meaning the samples contained in several periods. Furthermore it was observed whether the signals were time-shifted correctly. Finally it was expected that the found cross correlation coefficient and amplitude difference indicated a worsening condition by the patient.

Results It was observed that all internal results from the trend alarm were correct compared to the simulated input signals. The period for the simulated signal was found and the signal was meaned and time-shifted correctly. Furthermore it was observed that the cross correlation coefficient and the amplitude difference were calculated correctly, resulting in correct given alarms.

IV. Results

The results for the test cases are described in the following.

A. Connection, display and distribution

It is observed that the monitoring unit lists the available bedside units after having been intialised and that a connection to the bedside unit is made. Furthermore it is observed that the data is obtained correctly by the monitoring unit and is shown in the appropriate areas of the graphical user interface. When connecting more than one monitoring unit the data is successfully distributed to all connected monitoring devices.

B. Threshold alarm

It is observed that the system triggers and displays a threshold alarm when the test person is hyperventilating, raising the respiratory frequency above 16 breaths/min.

C. Equipment alarm

It is observed that the system detects a disconnected probe and that an alarm is shown on the monitoring unit.

D. Trend alarm

It is observed that all internal results from the trend alarm are correct compared to the simulated input signals. The period for the simulated signal is found and the signal is meaned and time-shifted correctly. Furthermore it is observed that the cross correlation coefficient and the amplitude difference are calculated correctly, resulting in correct given alarms.

V. Discussion

The purpose of this study is to investigate the possibility of designing an intelligent alarm system capable of evaluating alarms based on data from several medical monitoring devices, and thus enable a user to monitor patient data from any given location.
The system has been implemented as a distributed system with bedside units collecting data from patients, and distributing these to multiple monitoring units. The current system is able to collect data from a Brüel & Kjær Anesthetic Gas Monitor, type 1304 and a Siemens Servo Ventilator 300. The profile handling is structured so as to be general. In this study, two different standard profile types, COLD and CABG, are implemented to prove the concept of the system. The flexibility of the system enables users to create additional standard profiles. The profiles can be customised through the use of individual profiles for each patient. These customised profiles describe how alarms must be triggered.

The fact that the system collects data from the measuring devices connected to the patient. These collected data can be distributed to several monitoring units connected through the network. The system can determine threshold alarms when measurements exceed threshold limits defined by the profiles. The system is also able to determine if an alarm is caused by equipment failure. Both alarm types can be displayed on the monitoring unit. The system is not able to trigger trend alarms based on measured medical data. In the case of network link failure, basic functionality of the system is only maintained on the monitoring unit located on the same machine as the bedside unit. However features requiring access to the central data repository are not available. These are: Saving continuous measured patient data, calculating trend alarms, and saving a customised profile.

The system has been implemented in the programming language Java which makes it platform independent. This means that a monitoring unit can run on a portable device, e.g. a tablet PC, connected to a network. This way a user can monitor the patients, via e.g. WLAN[1]. This feature can save the user time in the everyday work, allowing focus on medical support.

The programmed trend alarm module should be implemented as to enable the system to trigger trend alarms based on measured medical data. A feature that could prove useful to incorporate into the system is a monitoring unit capable of displaying several patients data concurrently. This will aid in keeping the amount of equipment needed to monitor several patients down to a minimum.

Instead of using serialized objects, communication between the bedside unit and the monitoring unit could be achieved by communicating via XML-documents. XML is platform independent and will ease future integration into other Hospital Information Systems (HIS). For support of a wider range of medical monitoring devices additional drivers should be implemented. The program has been structured with a device-superclass, that eases the development and integration of new device-drivers.

The structure of the system allows it to be used as an all-purpose monitoring system. A system like this has many areas of applications where the need to homogenise data collection and to compare measurements continuously exists.

Thus, this study has shown that it is possible to collect data from several medical monitoring devices and through a distributed system, display data and evaluate alarms based on these data.

References

[1] C.S. Giles, J. Edworthy, R.D.H. Brown, and C. Davies, "Itu alarm confusion - a smart solution," *IEE Colloquium Medical Equipment Alarms: the Need, the Standards, the Evidence (Ref. No.1998/432)*, p. 44, Oct. 1998.

[2] Simon B. Larsen and Thomas Borup, "Icumatic - udvikling af et informationssystem til praktisk anvendelse i forbindelse med intensiv overvågning," M.S. thesis, Aalborg University - Medical Informatics, 2002.

[3] *Random Signals Detection, Estimation and Data Analysis*, John Wiley and Sons., 1988.

[4] S. Deering, "Host extensions for ip multicasting," Internet RFC, 1989, `http://rfc.sunsite.dk/rfc/rfc1112.html`.

[1] Wireless Local Area Network.