Project: Sapien 190
Date: 03.06.2010
Version: 1.05

Requirement Specification for Sapien 190

Embedded Real-Time Systems (TI-IRTS) Spring 2010

Peter Høgh Mikkelsen (20087291)
Anders Block Arnfast (20085515)
Kim Bjerge (20097553)

PAsien

Version Hisotry

Ver.	Date	Initials	Description
1.00	08.04.10	PHM	Initial Version
1.01	14.04.10	KBE	Removed Traniee from Use Case descriptions and changed use case Execute and Control Simulation added ECG, EDR and pulse signals.
1.02	15.04.10	KBE	Updated according to outputs from project meeting. Use Cases renamed and restructured document. Completed for review to first delivery.
1.03	16.04.10	KBE	Added modifications after review and completed for first delivery.
1.04	24.04.10	KBE	Editorial updates
1.05	03.06.10	KBE	Updated deliveries according to final delivery

Table of Contents

1. INTRODUCTION

1.1 Purpose

The Sapien 190 is a patient simulator, simulating human physiological behaviour according to different patient scenarios. Scenarios are written in an open format and can be downloaded to the Sapien 190 that contains patient records from the PhysioBand database.

In conjunction with the Sapien 110 human doll, the Sapien system provides a complete simulated human interface, with EKG, ECG and respiratory measuring spots and medicine injection spots.

1.2 References

[1] Erich Gamma et al., Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley (GoF)

[2] Bruce Powell Douglass, Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems

[3] PhysioNet and PhysioBank the research resource for complex physiologic signals http://www.physionet.org/

[4] Project specification for PSIMU: Patient Simulator System http://kurser.iha.dk/eit/tiirts/Projekter/PSIMU-project.doc

[5] Project specification for LMON: Local Monitor System http://kurser.iha.dk/eit/tiirts/Projekter/LMON-project.doc

[6] Project specification for IPUMP: Infusion Pump System http://kurser.iha.dk/eit/tiirts/Projekter/IPUMP-project.doc

[7] Project Interface Specification http://kurser.iha.dk/eit/tiirts/Projekter/ProjectInterfaces.doc

1.3 Definitions

- **Model** – A model that represents one physical individual. The model may consist of sub-models for different body subsystems. The model uses an algorithm to compute the output signals based on the input signals or patient records.
- **Model Parameters** – Parameters used in the model.
- **Record** – A patient record file taken from the PhysioBank database
- **Scenario** – Model parameters and a collection of records taken from the PhysioBank database.
- **Scenario Configuration** – A set of files that represents model parameters and patient records.
- **Signals** – Signals in form of waveform files or input from external equipment
- **Simulation** – A continous mode, where the physiological output signals are updated according to the model and the scenario applied to it.
- **ECG** – Electrocardiogram
- **EDR** – ECG–Derived Respiration

OVERALL DESCRIPTION

1.4 System Description

1.4.1 System Overview

The Sapien 190 consists is a compact unit with a graphical user interface (GUI) and interfaces to a range of body monitoring and injection equipment.

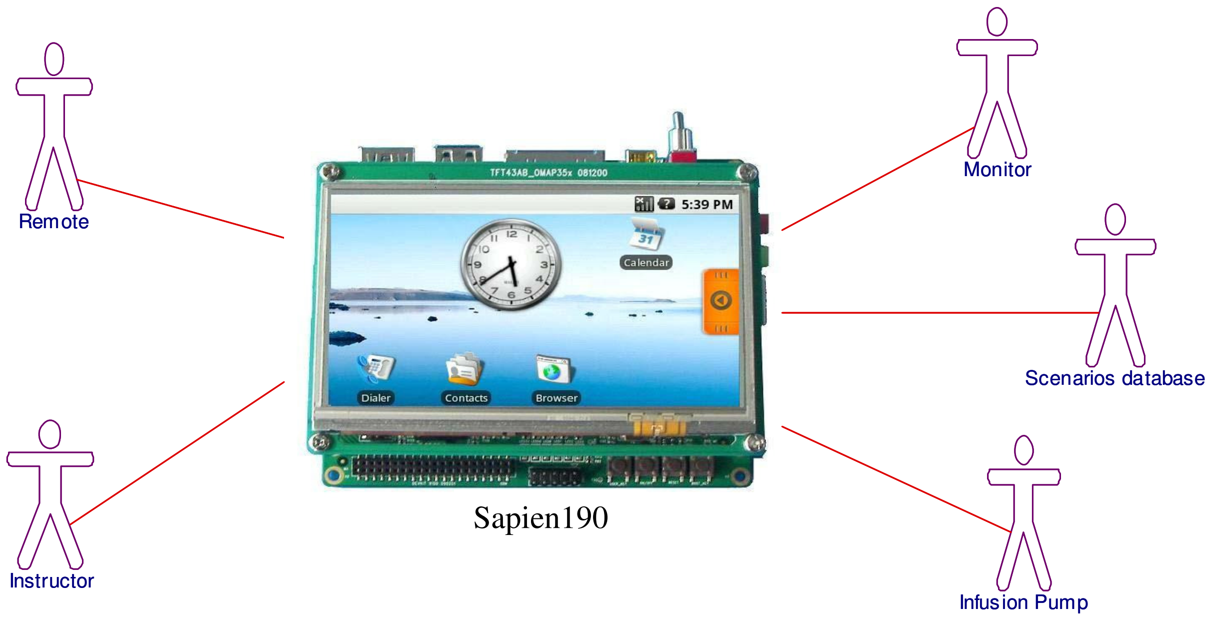
The GUI provides acces to a range of patient scenarios, and a range of adjustment possibilities.

The Sapien 190 provides the following interfaces:

- Two Analogue output signals (0-4V) (EKG / ECG / Persiration)
- One digital interface to an infusion pump

Figure 1. System Overview Diagram

1.4.2 Actor-Context Diagram



Figur 2. Actor-Context Diagram

1.4.3 Actor Descriptions

Actor Name:
Instructor

Type [primary / secondary]:
Primary actor

Description:
The instructor selects scenarios, run simulations and monitors the patient's condition. The instructor can adjust model parameters during simulation like provoking a heart failure or other changes in the patient's condition.

Number of concurrent actors:
Just one

Actor Name:
Scenarios database

Type [primary / secondary]:
Supporting actor

Description:

The scenarious database consists of a number of scenarious that is used as input for execution of simulation. Each scenario consist of records from the PhysioBank database and model parameters.

Number of concurrent actors:
Just one

Actor Name:
Remote

Type [primary / secondary]:
Primary actor

Description:
The remote interface allows uploading of new scenarios and the injection of signals to change the patient's condition.

Number of concurrent actors:
Just one

Actor Name:
Infusion Pump

Type [primary / secondary]:
Primary actor

Description:
The infusion pump inputs medicine type and flow to the Sapien 190. The medicine type and ammount impacts the ECG/EDR/respiratory signals according to the human model used.

Number of concurrent actors:
Just one

Actor Name:
Monitor

Type [primary / secondary]:
Offstage actor

Description:
The monitor takes the analogue output signals and displays them. There is no feedback from the monitor to the Sapien 190.

Number of concurrent actors:
Just one

1.5 System Functions

The purpose of this system is to develop a patient simulator monitor unit (SAPIEN 190). The system should be able to simulate different patient records by reading digitized physiologic signal from a collection of recordings freely available from the PhysioBand database.
Ref. http://www.physionet.org/physiobank/physiobank-intro.shtml

The functions of the patient simulator consist of a number of main use cases listed below:

Manage Scenarios

It should be possible remotely to add and remove new simulation scenarios to the system. A scenario consists of the digitized physiologic signals and a configuration of the patient simulation model. A scenario contains a list of parameters that can be adjusted during the patient simulation.

Adjust Scenario Parameters

The instructor should be able to adjust the parameters for the configuration of the patient simulation model that is specified for a given running simulation scenario.

Select and Initiate Scenario

The instructor should be able to choose between the simulation scenarios and start the patient simulation.

Execute and Control Simulation

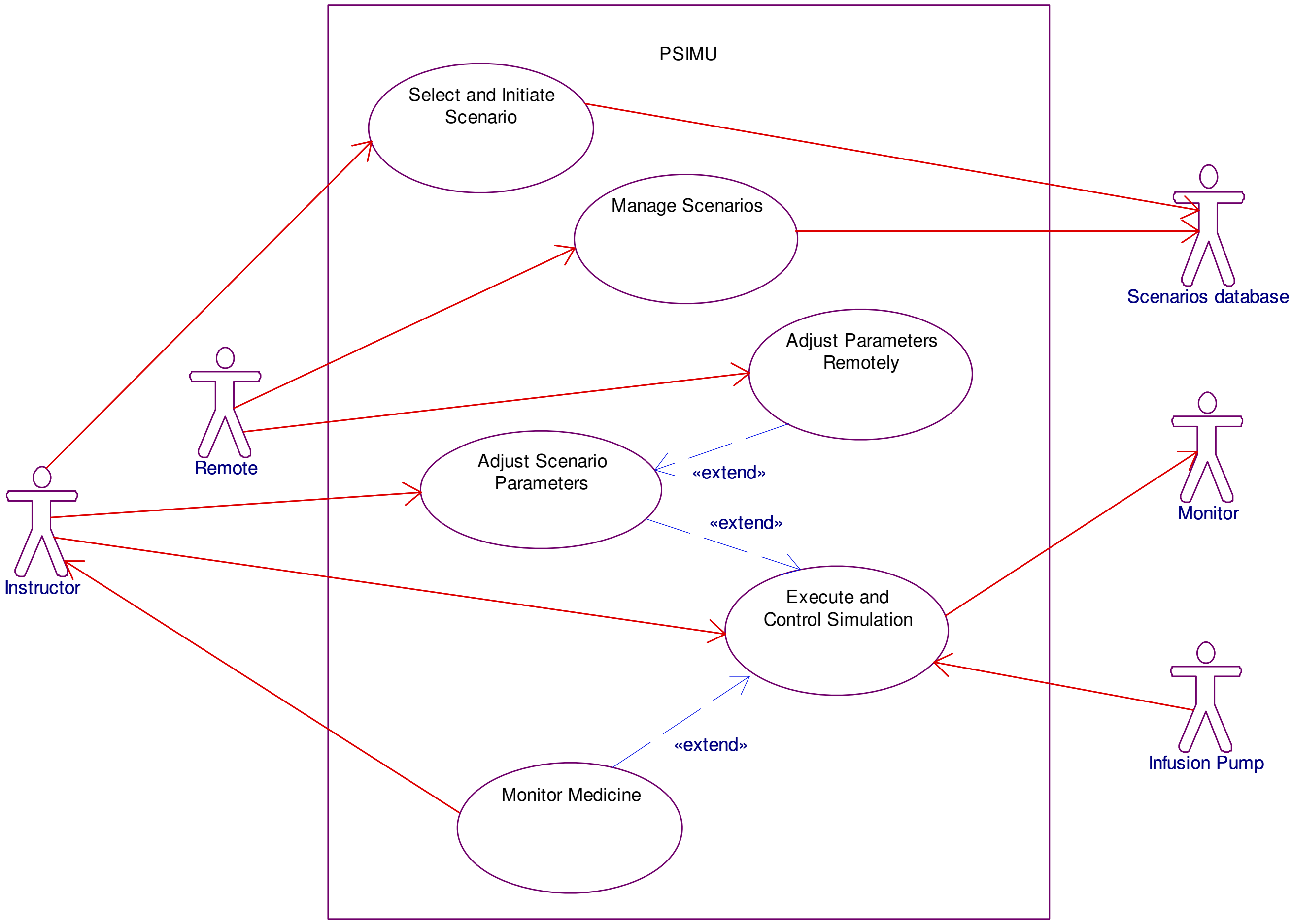
When the patient simulation is running it will perform reading of the digitized physiologic signals and send theses “real time” values as analogue signals to local connected bedside monitoring equipments. Up to 2 analogue channels with different signals is possible to be simulated simultaneously. The simulated signals can be ECG or EDR. The pulse will be signaled to the bedside monitoring equipment as a digital signal.

Monitor Medicine

If part of the simulation scenario the system is updated with readings from the Infusion pump. The instructor should be able to monitor different types of medicaments infused during simulation if the Infusion pump is connected and part of the simulation scenario.

The above system functions is found and described by the Use Case technique. The following diagrams show the system functionality expressed in one or more Use case diagrams. The purpose of these diagrams is to give an overview of the wanted system functionality. Each of the Use Cases on the diagrams is specified in detail in chapter 2.

1.5.1 Use Case Diagram(s)



Figur 3. Use Case Diagram

1.6 System Constraints

No additional information to the monitor is given beside the digital and analogue signals and therefore information available on the simulator and infusion pump cannot be displayed on the connected monitor.

1.7 Systemet Future

It should be possible in the future to add different types of infusion pumps. The systems should be able to be updated with different types of models depending on new type of physiologic signals. The system should be designed being able to be ported to other platforms (OS).

1.8 User Characteristics

The teacher is the main user of the simulator and will be a skilled educated person in operating the system.

1.9 Prerequistes

The target should be running embedded Linux in this first version of the simulator.

2. SPECIFIC REQUIREMENTS (USE CASES)

2.1 Use Case #1: Execute and Control Simulation

Goal:

To simulate signals from the patient based on the selected scenario.
The waveform of signals to monitor and the status of patient are displayed.
The instructor must be able to monitor the simulated patient.

Initiation:

This use case is activated by the system after simulation has been startet by instructor.

Actors and Stake Holders:

Instructor & Monitor & Infusion Pump

No of concurrent instances: 1

Frequency:

Performed continously as long simulation is running.

Non functional requirements:

The analogue ouput signals (ECG and EDR) from the simulator have a resolution of 12 bits and sampled with 250 Hz. Voltage range is 0 - 4.096 volts. The pulse signal is generated as a 1 bit digital output with duration of high in 100 ms. The pulse signal wil be in the interval 0-250 beats/minute.

References:

[5] Project specification for LMON: Local Monitor System
[6] Project specification for IPUMP: Infusion Pump System
[7] Project Interface Specification

Preconditions:

A simulation scenario has been selected and started in use case
#2 Select and Initiate Scenario.

Postconditions by success:

Signal waveform and status is displayed on the screen.
Output signals to monitor are updated continously

Postconditions by failures:

Error message on screen and no ouput signals

Main Scenario:

1. The system replays continuously simulation signals by reading from record files part of the scenario configuration
   [*Exception: Scenario files could not be read*]
   1.A The analogue output signals are updated by the system
      1. ECG signal is update
      2. EDR signal is updated (Derived respiration signal from ECG)
      3. Pulse is updated (0-250 beats/minute)

   1.B The LCD screen is updated with the waveform of the simulated signal that currently has been chosen
2. After a specified time depending on the scenario configuration an alternative simulation file could be programmed to be replayed
3. [*IPUMP present*]

Extensions:

Extension 1: [*Exception: Scenario files could not be read*]

Error messages is displayed on screen and no output signals to monitor

Extension 2: [*IPUMP present*]

If the IPUMP is present and part of the scenario configuration, the type and amount of medicine injected into the patient is used to update the simulation model. The type and amount of medicine has an impact on the model that simulates the ouput signals, like increasing pulse or changing the ECG waveforms.

2.2 Use Case #2: Select and Initiate Scenario

Goal:
Initiate the simulation using a scenario configuration chosen by the instructor.

Initiation:
Sapien is powered-up and user interface is ready

Actors and Stake Holders:
Instructor & Scenarios database

No of concurrent instances:
Just One

Frequency:
Max one minute interval

Non functional requirements:
None

References:
None

Preconditions:
Sapien is powered-up and user interface is ready
A valid scenario is available

Postconditions by success:
Enter “Execute and Control Simulation” state and
shows the “Execute and Control Simulation” view.

Postconditions by failures:
Error message on screen and no ouput signals

Main Scenario:
1. Instructor selects a scenario from a drop-down menu
2. Instructor initiates the simulation by pressing a button

Extensions:

2.3 Use Case #3: Adjust Scenario Parameters

Goal:
Adjust the parameters available in the current model running

Initiation:
During "Execute and Control Simulation",
the instructor may change the model parameters

Actors and Stake Holders:
Instructor

No of concurrent instances:
Just one

Frequency:
May be changed continuoly during "Execute and Control Simulation"

Non functional requirements:
None

References:
None

Preconditions:
Simulation is running

Postconditions by success:
Parameters is changed

Postconditions by failures:
Parameters are not changed and an error message is output

Main Scenario:
1. Instructor picks a parameter
2. Instructor updates its value
   [*Extension 1: Value out of range*]
3. Instructor updates the model by pressing the "update" button

Extensions:

Extension 1: An error is output to the screen, value is NOT updated

2.4 Use Case #4: Monitor Medicine

Goal:
To monitor medicaments infused into the patient from the Infusion Pump.

Initiation:
This use case is activated by the system
after simulation has been startet by instructor.

Actors and Stake Holders:
Instructor & Infusion Pump

No of concurrent instances: 1

Frequency:
Performed continously as long simulation is running.

Non functional requirements:
None

References:
Project specification for IPUMP: Infusion Pump System

Preconditions:
A simulation scenario has been selected and started in
use case #2 Select and Initiate Scenario.

Postconditions by success:
Medicine and amount infused into the patient is displayed on the screen.

Postconditions by failures:
Error messages about condition with Infusion pump are displayed.

Main Scenario:

1. The system reads data from the Infusion pump if it is part of the scenario configuration
   [*Exception: No readings from Infusion pump*]
   1.A The system updates medicine name and volume on the screen based on data received from the Infusion pump.

Extensions:

Extension 1.A: [*Exception: No readings from Infusion pump*]
Message is display that Infusion pump is not connected or could not be read.

2.5 Use Case #5: Manage Scenarios

Goal:
To add and remove new simulation scenarios to the system

Initiation:
This use case is activated remotely by the instructor
after a network connection is established.

Actors and Stake Holders:
Instructor & Scenarios database

No of concurrent instances: 3

Frequency:
Rare

Non functional requirements:
The ftp-server on the device is functional

References:
None

Preconditions:
The simulator has been powered on and a network connection is established.

Postconditions by success:
The scenario database has been updated.

Postconditions by failures:
The scenario database has not been updated.

Main Scenario:
1. User connect to the system
2. The system allow user to see folder of all scenarios
    1. User delete folder containing a scenario of the scenario the user intent to delete
    2. User upload a folder containing a scenario files
3. User disconnect to the system

Extensions:

3. EXTERNAL INTERFACE REQUIREMENTS

3.1 User Interfaces

The user interface shall be written in English.
The GUI must be simple and easy to monitor and operate for the trainer.
The graphic interaction with the users will be based on a GUI written for Linux.
The GUI shall be written in Qt, being able to be ported to other platforms in the future.
The user will operated the system by inputs to a touch screen and therefore the GUI design should be designed according to this constrain. Draft GUI simulator written in Qt:

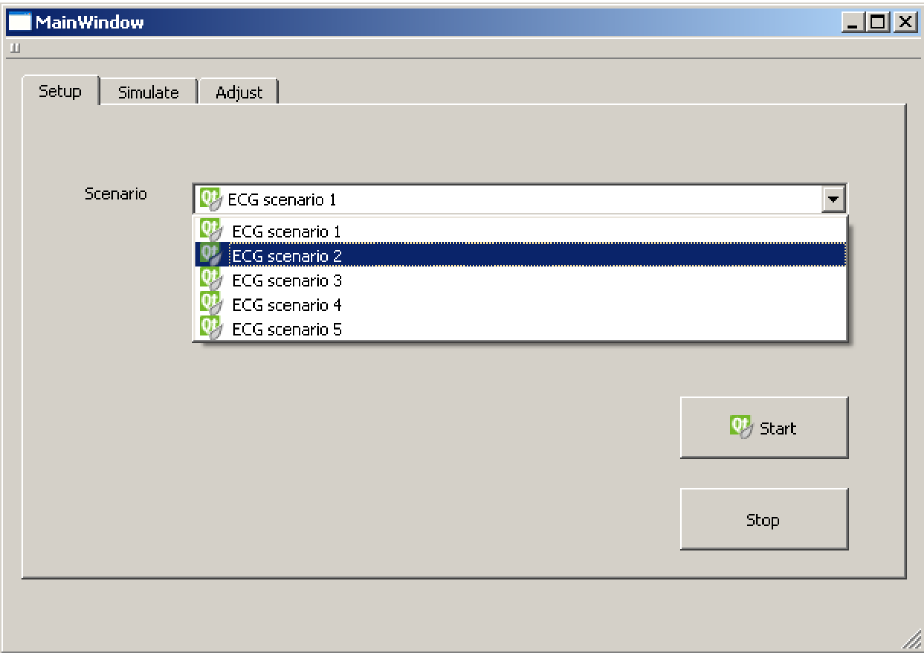


Figure 1. Select and start simulation scenario

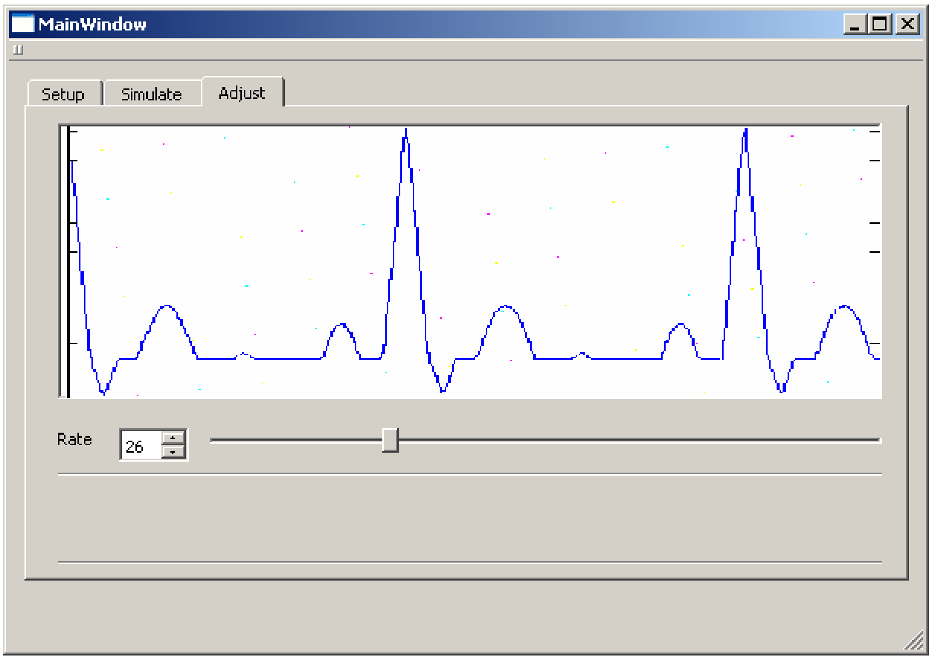


Figure 2. Waveview and adjust of simulation parameters

3.2 Hardware Interfaces

The following interfaces are provided:

- Main processor ARM Cortex A8 running embedded Linux
- 2-channel 12-bit DAC 100KHz Sample Rate 0-4.096 volt
- RS-232, 115200Kbps 8-bits, no parity(Infusion Pump)
- Ethernet 100BASE-T (Remote Interface)

3.3 Communication Interfaces

A user of the system will be able to connect to the system via ftp protocol. The user will have limited access to the system, but will be able to upload and delete scenarios through ftp connection.

3.4 Software Interfaces

3.4.1 Simulation Scenario Format

All simulation scenarios description should be in the same folder as the required files, and the description files shall always be named scenario.xml, and shall be able to be validated by the the XML-scheme specified in Apendix A.

All the required files defined in the headerfile, which are refered through the scenario description, shall exist. The system will not be able to handle corrupted files.

The system will only be able to process databases downloaded from http://www.physionet.org/physiobank/database/#multi%23multi

4. PERFORMANCE REQUIREMENTS

The analogue ouput signal from the simulator has a resolution of 12 bits. Voltage range is 0-4.096 volts. The digitized physiologic signal is sampled in the area of 250 Hz. The system should be able to replay these signal with the in the rate between 0 – 500 Hz. The jitter must be in the area of 1 ms.

5. SYSTEM QUALITIES

Maintablility, correctness and usablility is the most important quality factors for the patient simulator. The simulator application is expected to have a longer life time than the hardware and must be able to be maintained for many years in the future.

System qualities are rated as following: 1: non-critival, 2: not important, 3: important, 4: very important, 5: extremely important

The following qualities are rated:

Maintability: 5
Reliability: 2
Usability: 4
Reusability: 3
Correctness: 5

6. DESIGN CONSTRAINTS

The system must run on the DevKit8000 platform and use the Linux OS.

7. OTHER REQUIREMENTS

7.1 Authority Requirements

The hardware is COTS and has already the required hardware approvals.
Though a medical product, this product is not a life-maintaining or in other way a life-critical product, no medical approval is therefore needed.

8. PART DELIVERIES

The first delivery will contain a design and implementation of use case #1 Execute and Control Simulation where display of signal waveform could be optional. It will be delivered 3 weeks after accepted specification. The delivery will contain a system architecture document and a simulator running on target being able at least to send the simulated ECG channels to the monitor.

The second delivery will contain the use cases #2 Select and Initiate Scenario, #3 Adjust Scenario Parameters and optional #5 Manage Scenario and #4 Monitor Medicine. It will contain the final report and product documentation and simulator being delivered the 4. June 2010.

1. **Delivery – 11. May 2010**
   Updated Requirement Specification
   Draft Product Architecture Document
   Status Report
   Simulator prototype first version (Part of Use Case #1)

2. **Final Delivery – 4. June 2010**
   Requirement Specification
   Product Architecture Document
   Project Report
   Simulator prototype (Part of Use Cases #1, #2 and #3)

9. Appendix A: Scenario Configuration Format

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="Scenario">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="headerfile"/>
            <xs:element name="description" type="xs:string"/>
            <xs:element name="parameters">
                <xs:complexType>
                    <xs:sequence maxOccurs="unbounded">
                        <xs:element name="parameter">
                            <xs:complexType>
                                <xs:attribute name="name" type="xs:string"/>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>
```