Wireless Remote control

For Dialysis Machine

Master's thesis by Felix Reychman, 2003

The division of Ergonomics and Aerosol Technology at Lund Institute of Technology,

and

Gambro

GAMBRO.

Abstract

People with dysfunctional kidneys need dialysis to clean their blood, remove excess fluid, and will die if they do not receive their treatment. That treatment is called dialysis, and is generally performed by a machine, in which the patient's blood is filtered. In order to receive more, and usually better, treatment, some patients choose to perform the treatment at home by themselves instead of at a dialysis clinic.

The intention of this work was to find out whether these home HD patients would be helped by a small, bluetooth connected remote control, implemented on a PDA, propose a GUI design for such a remote control, and build a prototype.

The GUI was designed with the QUAD method, tested first on paper dummies, and later on a laptop running the prototype application.

The tests proved that the application was a good idea, and would indeed help home HD patients by making their home treatment a lot easier, but that the hardware of the iPAQ used in this project is still too immature to support it.

Preface

This report is my Master's thesis, which is the final part of my education in Computer Science at Lund Institute of Technology. The division of Ergonomics and Aerosol Technology at Lund Institute of Technology have supervised the project from the university perspective.

It was made possible by the great support of Gambro Lundia AB in general and the Treatment Systems Research department in particular, where all work was also done.

Although many have contributed, some deserve extra credit, and they are:
Roland Persson (supervisor at Gambro) for almost all help, Gerd Johansson (supervisor at the division of Ergonomics and Aerosol Technology), Gunilla Nordgren & Susanne Pedersen (Gambro) for helping me test, Jonas Svensson (Gambro) for helping me with software for the AK200, my mother and all others criticizing this report, the entire staff at Treatment Systems Research for all help, support, nice coffee breaks and welcoming attitude. I really enjoyed myself here!

Table of Contents

Sammanfattning

Människor vars njurar fungerar illa eller inte alls behöver en behandling som vanligen kallas dialys. Denna behandling går ut på att man på syntetisk väg, med en dialysmaskin, utför njurarnas arbete som i stort består av att rena blodet från slaggprodukter av olika slag som till exempel urinämne. En dialysbehandling varar från ca 4 till 8 timmar, och genomförs i allmänhet minst varannan dag. Om en njursjuk inte får sin behandling, kommer vederbörande att avlida inom någon eller några veckor.

Somliga njursjuka väljer att behandla sig själva hemma med en egen dialysmaskin, eftersom de då får mer behandling, vilket praktiskt taget alltid är detsamma som bättre behandling. Dessa patienter får problem eftersom dialysmaskiner sällan är konstruerade för att köras av patienten, utan i allmänhet är konstruerade för att köras av utbildad sjukvårdspersonal.

Syftet med det här examensarbetet är att ta reda på om dessa patienter skulle vara hjälpta av en liten fjärrkontroll som, via bluetooth, kommunicerade med dialysmaskinen och bestod av en handdator. I arbetet ingick också att konstruera ett användargränssnitt och en prototypmjukvara för att kunna testa det.

Genom att med hjälp av Rapid Prototyping konstruera prototyper och testa dessa, och sedan implementera det hela i java har dessa egenskaper kunnat testas.

Resultatet var att användargränssnittet uppfyllde kraven på användbarhet, men att hårdvaran, och i synnerhet de delar som hanterade bluetooth, inte höll måttet av pålitlighet och stabilitet.

Goals for the Thesis

What should the thesis cover

The goal of the thesis is to establish whether a good graphical user interface (GUI) for the AK200 dialysis machine can be constructed and run via bluetooth from a handheld computer (PDA) such as a Palm Pilot or an HP iPAQ. A GUI is good if it lives up to high standards of usability with all the issues of understanding, safety etc and still supports the actions needed for a dialysis patient performing his/her treatment at home, more or less by him- or herself. The definition of the usability goals is a part of the project.
To validate this user interface, and also to study the basic technical aspects of such an application, a prototype should be constructed.

What should the thesis NOT cover

This thesis will not deal with all the security issues connected with a product like this. Some security fields will be mentioned briefly in the conclusions chapter, as a part of what work lies ahead. Neither will a complete functioning remote control be constructed, but only a prototype.

1. Introduction

1.1 Dialysis patients and treatment

Who needs dialysis? The answer is rather straightforward. Anyone whose kidneys function badly enough is a dialysis patient. The reasons for kidney disorder are several, and among them can be found diabetes, infections and a lot of other diseases. The dialysis machine is little more than a synthetic kidney, and was in the beginning referred to as such. The function of the kidney that needs to be replaced is the production of urine production. Normal people clean their blood and remove the excess water through urine, but with no kidneys, no urine is produced, and nothing is removed from the blood. Water and toxins accumulate, and the patient will die within a few weeks at most, with no treatment.

The dialysis machine does mainly two things. It removes waste products of several different kinds, of which the most important is urea, and it removes excess water. That excess water is about 2-3 litres per day.

Traditional treatment is performed at a clinic, where the patient spends 4-5 hours every second day attached to the machine, and during that time, about 4 litres of water are removed. Removing that amount of fluid from the body in such a short time often causes nausea and fainting, and might be very unpleasant indeed. Many also find the surroundings at the clinics disheartening, since they are treated along with other patients who might be old, very sick or suffer from dementia. The patients are forced to keep a very strict diet with a minimum of drink, water included.

For several reasons, patients are often encouraged to treat themselves at home. From the clinics point of view, it saves a lot of money on personnel on which there is always a shortage, although the machine might be expensive. From the patients' point of view, it is a matter of quality in treatment. By having their own machine at home, they can run it at times that suit themselves, and as long as they want and in an environment that is rather less depressing than a clinic at a big hospital might be. They can run it at night when sleeping, or in the evening while watching TV. They can also run it every day, and much longer than 4-5 hours. Some home HD patients get 8 hours of treatment every day, instead of the 4-5 hours every second day. It is almost four times as much, and thus, they are much healthier. They may also lead a more normal life with a somewhat less strict diet, although they still have to be very careful about what they eat and drink. However, many patients are in so bad a shape, that they are not fit to treat themselves. They will of course have to remain patients at a clinic.

Most dialysis machines available today are big, complex medical machines, designed for nurses who are to treat someone else. Very few are designed to cater to the needs of home HD patients. They are also often rather sad creations concerning the user interface, making them difficult to learn and clumsy to use even for the professionals.

1.2 Bluetooth

Bluetooth is a communication standard, designed by Ericsson Mobile Communications with short-range radio in mind to replace cables. It is useful when building wireless keyboards and wireless computer networks etc. When designing Bluetooth, considerations have been taken to enable (quoted from http://www.swedetrack.com/images/bluet00.htm)

- Global usage
- Voice and data handling
- The ability to establish ad-hoc connections
- The ability to withstand interference from other sources in open band
- Very small size, in order to accommodate integration into a variety of devices
- Negligible power consumption in comparison to other devices for similar use
- An open interface standard
- Competitively low cost of all units, as compared to their non-Bluetooth correspondents.

Today, more and more electronic equipment, like PDAs, cellular phones and other computers is equipped with antennas and software to implement this standard as the prefabricated circuits for Bluetooth are getting cheaper, which greatly improves the possibilities of connecting these devices in a simple manner. The design of the protocol, intended to replace cables such as the serial cable, and the various choices of devices that support it, make it ideal for implementing a remote control. Indeed, remote controls are one of the intended target products for Bluetooth.

1.3 User Interfaces and The Big Conflict: freedom or simplicity

Why is there not one general rule telling us how to build an optimal Graphical User Interface (GUI)? In many sciences an optimum can be calculated, such as the shortest route, the optimal size of a bus to accommodate the most profitable number of passengers, the cost to spend on an advertising campaign calculated by the expected profit, the most economic speed in terms of kilometres per litre of fuel and so on.

In some sciences there is a conflict, forcing the engineer to decide what is more important - in the travelling case, speed versus fuel consumption, in the GUI case, the time it takes to learn a system versus the number of functions that can be achieved by the system versus the effectiveness of the system.

It is easy to form a picture of the system being the easiest to learn. It would have one button, preferably already pressed before the machine left the factory. However, such a system would probably not solve anything for the user. The other extreme is a system that can solve any problem conceivable. Such a system would, however, be equal to the known universe. It can not be built for many reasons, whereof one is that the human world of today does not have every possible answer to every possible question, and even if we did, a machine that recorded our favourite TV show, planted flowers in the garden while feeding the starving people in the third world and calculated the size of every star in the sky would be rather impractical. The third extremity is a system that automatically solves *the right* problem, without the user having to choose, and does so in no time at all, or even before the user was aware of the problem. Such a machine would require telepathic abilities, which, with the science of today, are not possible. This can also be expressed as the choice between the freedom to solve more problems in specialized manners having greater power over the machine, versus the simplicity of a machine that is easy to use, versus spending no time - one button or an infinite amount?

It is obvious we need to find a system that takes a reasonable time to learn, solves the more important and related problems without too many pressings of buttons and does so in a reasonable time. To find what time is reasonable, what functions are important, and how many buttons the user can be bothered to press, we need to study the users to see what they have in common, what problems they have that we wish to solve, how often they will use the system and how good they can be expected to become using it. We will, of course, still not produce a utopic system. It will require learning. In this case, it probably will be somewhat complex, and it will require extra pressings of buttons for reasons of security and so forth.

The Prestudies, including User Profiling, showed that extensive training would be provided for the users, that they would spend hours every day using it and that they have a great interest in learning this system. Thus, they will probably prefer a somewhat faster system that can solve more problems, at the cost of some over-obviousness and tedious explanations of every button. What will be built is what is generally referred to as an expert system – a system for experts.

1.4 QUAD in more detail

Andersson and Reis provided Gambro with the QUAD method in 1999, which has also been recommended in later thesis done at Gambro. It is also the major method mentioned in Gambro's UI Handbook. The QUAD plan is a high level method describing the major steps to go through when designing user interfaces, and was originally brought to Gambro with the obvious aim of pleasing the American FDA.

The QUAD plan has a number of steps and detailed documentation, some of which was rather excessive for this rather simple project. It is largely based on the concept of iteration: "Do it, test it and redo it until it is good enough". All the major steps below are of course closely related, and it might be hard to tell what exact actions belong to what step. They will probably melt together somewhat as you move along.

1.4.1 Prestudies

Prestudies is the phase where you read about your subject, visit clinics, and generally get the feel of the field in which you intend to work. Previously done work should also be studied.

What you get out of it, is understanding and general knowledge, for example field specific vocabulary.

1.4.2 User Profiling

The purpose of the user profiling is to get a good idea of who will be using the product. You should try to get a good cross section of people. If the users are numerous, statistics might be available, and they may carry a lot of information. Interviews are the other extreme, and you might get a lot of information about certain individuals. Be certain to check whether these individuals are representative for the entire population in question.

User might very well be the part that is most often forgotten. Finding out who is going to use the system, is vital to know what design principles to apply, how to apply it, how much education the user has in the particular area, if many users have certain disabilities, what hardware to chose etc.

An enlightening example would be a system aiming towards a group of people whose average age turn out to be over 65. Elderly people often have bad eyesight. Choosing a computer with an extremely small screen, and thus, small fonts, will probably make the end product useless for the targeted user group.

1.4.3 Task Analysis

Task analysis might also be described as the question “What am I going to do?”. Put like that, it sounds rather obvious, but knowing what to do is often a complicated matter.

The task analysis includes not only an understanding of the field, but also what the problem is that you are trying to solve. You have to understand how things are done today and why that is not good enough.

An important result is often a workflow, depicting the steps that make up the work the users perform. It might be something as simple as putting an item into a shopping cart at a web site, but also something rather more complex, like setting up a dialysis machine for a patient, with all the calibrations and actions like rinsing that has to be done. How to get an aircraft of the ground is another typical task that can be described as a flow of actions.

1.4.4 Conceptual Design

The conceptual design is the overall design where you concentrate, not on the little buttons and colours, but on what design principles to use and a general design.

The example of the aircraft is once again an illustrative one. The conceptual design might in this case be something like:

- Check aircraft
- Turn aircraft on
- Accelerate
- Take off

We do not mention all the little buttons that have to be pressed in order to achieve the rather complex task of turning an aircraft on.

Here you will have great advantage of a good task analysis.

1.4.5 Usability Goals

In order to know when you are finished, and if your product at that stage is good enough, you need to define goals. These goals should be measurable. The measurements might be time related or quality related, in general. Your goals might, for example, state that a user with no training should be able to solve case X within Y minutes, with no training, or that someone, who can be described as having bad eyesight and hearing and having only one hand should be able to use the system with the help of a trained dog (such as traffic lights by zebra crossings).

1.4.6 Design/Testing/Evaluation/Proposals (iterated until set goals are achieved)

This is where your main iteration is. You make a design, less detailed in the beginning, and fully detailed by the last iterations, you test this design, if possible with the intended users, you study the results from the test, you modify your design and then you start over testing again. If no proper goals are set, this iteration might be an infinite loop, out of which you will never emerge. Very few things are perfect, and thus, they can always be made better.

1.4.7 Follow-up

Once you have delivered, your work is not done. What if your system does not solve the problems it was intended to solve? You have to review your documentation, analyse user input, and also verify that what you did was done the right way. Did you do any unnecessary work? Was there anything you should have done that you did not do? Why? Document it, to avoid making the same mistakes twice!

2. Prestudies

2.1 Reading previously done thesis and other literature

Lots of work has been done on the fields of User Interface design, Human Factors aspects and other related areas. I was recommended the masters thesis "Human factors and the design process" by Andersson and Reis, [1]. This thesis brought the QUAD plan to Gambro AB, as a result of the need to please the American FDA and get better and more documented methods for UI development. The QUAD plan has since been the standard at Gambro AB and is mentioned in their UI Handbook.

I also read the masters thesis "MUI: Controlling Equipment via Migrating User Interfaces" [2], to see if the software they developed might prove interesting. It was very interesting, and would have been exactly what I needed, had it only been more mature. Unfortunately, it was not stable enough, and not developed enough to prove much help.

The thesis "Usability Guidelines for Wireless Services" [3], gave various insights into the issues concerning small displays.

2.2 Design Principles

Several principles exist that are claimed by the authors of the respective books to make GUIs better. However, what is better or worse depends largely on the user at which the GUI is aimed. I cannot present all the principles here, but I can present the ones I though were relevant to this project.

To be able to prioritise when designing, certain design principles have to be chosen as being important, and those choices have to be motivated. I have selected a few, presented by Jordan, 1998 [5]. To make the device safe, one of the most important principles should, in my opinion, be the one stating:

Consistency of UI

The UI have to be consistent. Similar tasks should be performed in similar ways. Similar tasks can be grouped, so that possible similar workflows might be identified. One example of such grouping is the setting of values. To set the alarm limits for vein pressure is an action that should be very similar to setting the alarm limits of arterial pressure as well as the limits for the TMP. How similar are they really? Can the handling of alarm be grouped? Can all settings of values be handled the same way? Probably not, but they can be designed more or less consistently. The best would be if they all looked the same, but were not possible to confuse with each other.

Choice of colour, fonts, icons and layout should also be consistent throughout the application. An icon should not mean one thing in one part of the application and something else in another.

Feedback
The system has to communicate with the user. Proper feedback is a fundamental issue when designing UIs. Feedback might be sensory (the feeling of a button clicking beneath your finger indicates that the button has been pressed.), it might be visionary (something changing colour or popping up or being written on your screen after a performed action) and it might be audible (you here a beep when a faulty value has been entered, and perhaps small clicks every time you press a button). Not all forms of feedback are available on a PDA. Most sensory input is for example out of the question, since the thing cannot move much by itself.

Consideration of faulty input
Very few systems are fail proof in the true sense of the word. However, the number of possible and probable mistakes the users can/will make might be limited. By considering what the user might do wrong, you might prevent those things from happening. In some cases this is not possible, and ways of correcting mistakes must be available. Backtracking, warnings of strange input, no irrelevant information and allowing correction of errors are the most obvious means to prevent errors. Many more exist.

Easy access to important information
In all systems, some information is more important than other. By grouping information, and prioritising, the system might be faster, easier and safer.

Another aspect is that the information available needs as little interpretation as possible. A quick glance should tell you all you need to know. To achieve this, the choice of fonts, icons, graphs and other graphical objects should be thought through. This is closely related to feedback and consistency of UI.

Obviousness of UI
Ideally, a manual should not be missed if never delivered. This utopia can of course not be achieved, but by designing things right, the user might figure most functions out by him or herself from just looking at the UI. Obvious Icons, the exactly right amount of text in the right size and so on are some of the means to achieve this goal.

Consider the abilities of the users
The users are no more than ordinary human beings. Thus they have limits regarding short-term memory and other intellectual and physical capabilities. These limitations should be considered, so that not too much information is displayed at once, not to many levels of screens and navigation are used, vision, hearing and feeling are all used in an optimal way to take in information and so on.

Consistency with the users' expectations
This is a dangerous principle. Since the UI of many machines are far from optimal, in some cases even downright bad, creating the new interface to resemble it might be a road leading to utter failure. Users might compare something new that is not to far from the first machine with the existing UI on

that first machine, and find it is less good. However, making something brand new, taking no regards to the UI of today, might result in something so different, that it cannot be compared, but be thought of as a new device. Given the UI of the AK200 and the evaluations of it available, I believe this is the best approach. (See next section)

2.3 Basic dialysis studies and the AK200

To get a basic understanding of dialysis treatment in general, I studied the Gambro Basics course material. This material is used as an introduction course attended by all new Gambro employees. I also visited the dialysis ward at the Lund University Hospital and talked to nurses and patients.

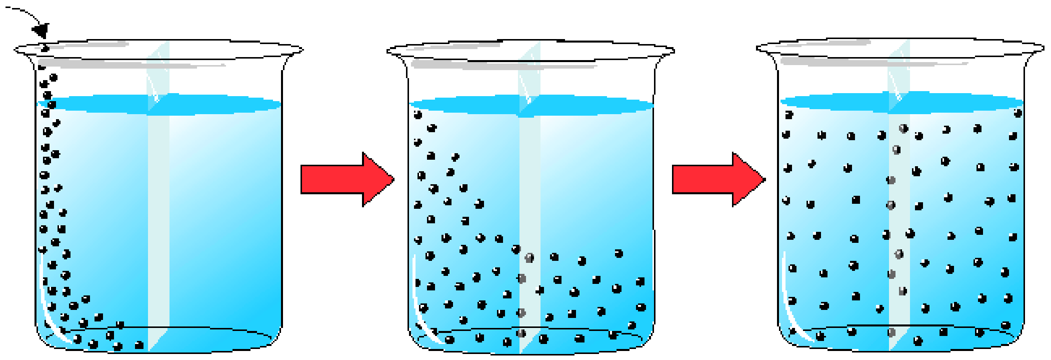
In order to learn the machine for which the remote control was to be designed, I read the manual and simply, under the supervision of an experienced engineer, plugged it in and ran it with a small container of water simulating the patient. I tried causing alarms by clampering the tubes or pushing air in and just causing all possible general problems for it, to see how it reacted. I have also, repeatedly, tried out things I have wondered about, by running the machine and testing, thus gaining what seems to be a sufficient understanding.

2.3.1 The AK200 in more detail

What is basically done in the machine is that blood is pumped from the patient to a filter in which excess water and waste products are drawn from the blood. That process is schematically described below. The blood is then pumped back into the patient.

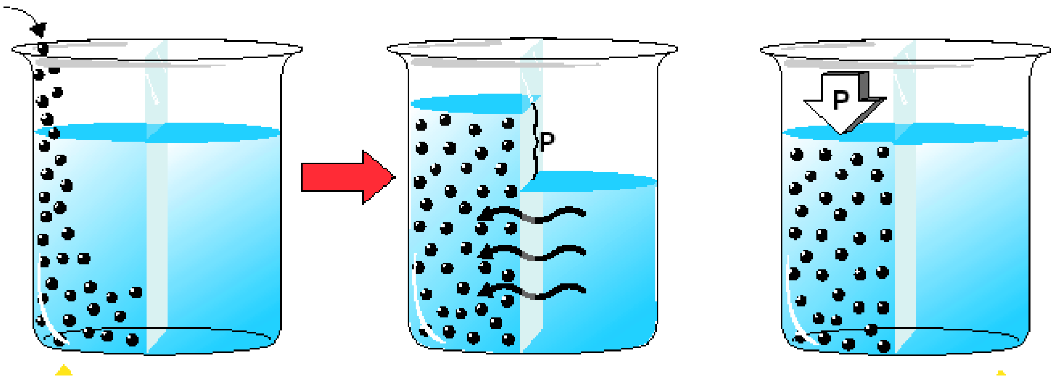
2.3.2 Transport principles

Some chemicals, among them some of the waste products, are drawn from the blood through a chemical process called diffusion. Diffusion [picture 1] is simply the fluid's desire to establish equilibrium, by distributing the solutes from a part with higher concentration to another part with lower concentration, for example from the dialysis fluid to the blood (waste products), or the opposite.



Picture 1. Diffusion

Other chemicals, being too large for the filter, result in what might be considered the opposite, osmosis [picture 2]. When the particles cannot travel through the filter, the fluid itself will move, to establish the much-desired equilibrium regarding concentration. Thus, water moves from the body to the dialysis fluid, and is transported away. This process can be described as a pressure, and this is the pressure, which is later mentioned as TMP.



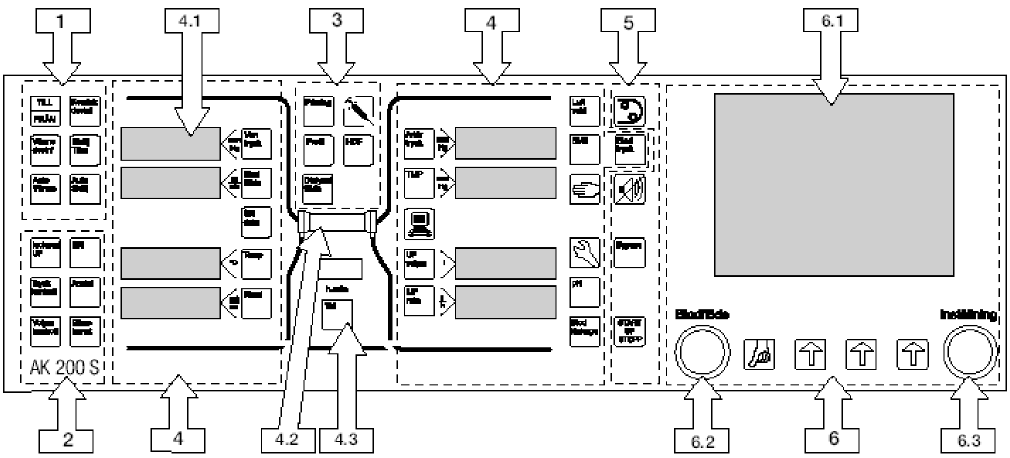
Picture 2. Osmosis

2.3.3 Controlling the machine

A number of parameters can be set and changed during treatment with the AK200 [picture 3, 4]. Many of these values are alarm limits, telling the machine at what state it should stop the pump and sound the alarm. Most of the values available are hardly ever touched, and need not be explained here. The more important values though, are:

- Blood pressure, (actual values and alarm limits)
- TMP and its limits. TMP is the chemical pressure in the filter with which water is drawn from the blood by osmosis
- Treatment time
- Blood flow, which is basically the speed of the blood pump
- UF (Ultra Filtration) Volume, rate and alarm limits, which tell how much fluid should be drawn from the body, and at what rate.

They are all closely linked together, so that changing one might start a chain reaction changing more values.



Picture3. The AK200 Picture 4. User Panel (for a more detailed picture, see appendix)

The AK200 has a serial port on the back, through which you can connect to another computer. Gambro has been using this port mainly for logging and debugging. Using a rather complex protocol, you can access pretty much any data inside that you like. You may, for instance, order data at regular intervals or upon certain events, such as alarms going of and so forth. It is also possible, on a certain model of the AK200 (the AK 200 Ultra S), loaded with a certain version of software, to send data *changing the state* of the machine. This is done by emulating the set of buttons on the physical UI of the machine. You then send packages that look like the ones being sent from the real buttons, and the machine will not be able to tell the difference.

Setting the machine up involves a few stages, which are mainly allowing it to perform a series of self-tests, rinsing it, and putting the tubing, filter and chemicals on.

2.4 Choice of methods

All literature and previously done work point in the same direction on a few subjects. The most important one is, that a lot of testing with real users and improvement of the UI in an iterative manner should be done.

2.4.1 GUI method

I chose the QUAD plan as my major method of GUI development. The main reason is the rather obvious one, that it is the method used by Gambro for developing this kind of products. It is the one method mentioned in the GUI handbook at Gambro. It was developed to suit Gambro's specific needs concerning medical devices. The main parts (see chapter 1.4) are:

- Prestudies
- User Profiling
- Task Analysis
- Conceptual Design
- Usability Goals
- Design/Testing/Evaluation/Proposals (iterated until set goals are achieved)

2.4.2 Software development methods

No special method has been used for the software as such. A general object oriented approach has been applied, and tools that support that philosophy have been used. Many minor parts from the RUP have also been used, such as creating UML diagrams and so on.

However, most software development methods are not really applicable to projects as small as this one. They tend to generate more paper than actual good results. Regarding the main issues of the project, where the UI aspects were the more important ones, I have taken the liberty of developing the software somewhat ad hoc. This is also a result of the knowledge, that it will be rewritten several times, and at least once from the beginning, since java is not really a suitable language, when you consider real-time and other security issues.

2.5 Choice of PDA

Several PDAs were considered, but the two major candidates turned out to be the Palm Tungsten and iPAQ 5450/5455, since these two, as opposed to most others, were equipped with built-in bluetooth communication. The major difference between the two, apart from brand, appeared to be substantially more memory and processor power in the iPAQ. The iPAQ comes with Microsoft Windows Pocket PC 2002 and the Palm runs Palm OS. For both, development tools seem to be rather easy to obtain with little or no cost, but with rather limited choices. However, the tools for Windows appeared to be of higher quality with more elaborate environments. I decided to go for the iPAQ and Microsoft eMbedded Visual Tools 3.0. A major drawback with the iPAQ, however, was that it had just been released on the marked, and might not be tested enough to prove completely reliable. Indeed, reviews available en masse on the Internet, all spoke of this hazard, although the bugs mentioned there were rather limited and should not get in our way. However, the machine, once delivered, broke within a week, just like the replacement, but was repaired within another week.

2.6 Choice of development tools

Once it had been established that the iPAQ was technically very superior to most of its competitors, the choice for development tools was rather limited. Java applications would probably not allow access to the bluetooth stack, needed for the communication, and the only realistic choice left was the Microsoft eMbedded Visual Tools 3.0, which, by a coincidence, was free of charge, and came with a handy PPC 2002 emulator.

After quite some time of writing small applications just to learn the environment, the iPAQ broke. It turned out that it was the debugger in the eVT that crashed the ROM image, something that should not be possible. The bug was confirmed when the temporary replacement iPAQ broke in exactly the same manner. This showed that I could no longer use the eVT. Java once again caught my attention, and after some more research it turned out that my application probably could be built that way after all. The eVT issue set me back several weeks, and CodeWarrior was ordered from Denmark.

2.7 Choice of third party software (packages etc.)

The prestudies included trying to find the main parts of the system and locating available packages and drivers. The only parts that would require anything not found in standard java were the communications part. Here I had to find drivers for the serial ports and base classes with which to access these drivers.

The basic kit for serial communication through serial ports in java is the javax.comm package provided by sun, which, unfortunately, does not support PocketPC 2002. However, I found a third party implementation of javax.comm., modified to suit my needs, called CESerial.

3. UI Development

The following sections are based on the QUAD plan.

3.1 Prestudies

The prestudies of the entire project, UI included, are found in chapter 1, Prestudies.

3.2 User Profiling

This user profiling was limited to Sweden. International studies would have been far to time consuming. Things might be considerably different in other countries.

Three nurses, whereof one is the head nurse at the dialysis clinic at Lund University Hospital, the others employed by Gambro, one employee at Gambro who have tried to do this user profiling before and two home HD patients were interviewed. I tried to find statistics about home HD patients, but none were to be found. According to those interviewed, none exist.

These interviews gave stunningly similar results. However, I was greatly disappointed to find that home HD patients have very little in common apart from their kidney disorders. Everyone interviewed said the exact same things, and told me that what they do have in common is an exceptional knowledge of the machine they run. Some know them so well, they even repair them themselves. They are also in rather good physical shape, their kidney malfunctions excepted, with no general disabilities in common. This is both a result of a better treatment, resulting from being home HD patients, and a necessity for being able to perform that same treatment. One nurse claimed that many home HD patients have diabetes, and as a result, poor eyesight, but this was denied by another nurse, claiming that diabetes patients are usually not well enough for home HD treatment.

Most home HD patients do not use Gambro machines though, but the competitor Fresenius'. This proved them utterly useless as test subjects.

A few more things they appear to have in common, according to the interviewed, are a strong will, lack of fear in general and of technical machines in particular. It was pointed out by that if they did not possess these qualities, they would not dare or be able to perform their own treatment.

3.3 Task Analysis

In most cases, finding out what problems to solve, and what actions to perform is a major part of the work. Task analysis proved a very different thing from what is mentioned in other thesis and the literature. In this case the machine to control was already designed, produced and delivered, leaving the different tasks to be performed very well defined in the AK200 Ultra Manual, which clearly describes the different alarms that might sound and the various values that can be set and any other interactions with the machine.

I decided to limit the tasks to the ones relevant during active treatment. The tasks regarding the starting and finishing of treatments include so many parts that have to be performed at the physical machine, such as setting up and dismantling the tubing and adding chemicals etc, that a remote is no help at all at those stages.

However, the time, as always, proved an obstacle preventing me from implementing all possible tasks even so. As a result, some kind of prioritising among them was necessary, and then the interviews with users were valuable. Indeed, it was the only way to choose! Knowing that task analysis would come and require interviews, I asked a lot of questions regarding the tasks as well, during the interviews concerned mainly with user profiling. Additional interviews and phone calls to the nurses already interviewed gave the extra information needed. This time answers differed somewhat more than concerning user profiling, but not much. They all agreed on the important tasks, and they all prioritised the same general way.

These tasks were the ones that turned out to be the most important, and the ones that would be possible to implement in time.

- Controlling the blood pressure, both regarding veinous and arterial blood, including current values and alarm limits, since blood pressure is a parameter that changes during treatment, as the patient gets up or lies down.
- Controlling the TMP, since this is a parameter that needs to be changed depending on how the patient feels. The TMP is dependent on, and also has effect on, the blood pressures.
- Monitoring the blood flow. The blood flow varies during treatment depending on how much water you want to draw from the blood at the moment.
- Controlling treatment time. You may wish to shorten or lengthen the treatment time, and, which is even more important, you will want to know how much longer you will have to wait

until it is over. If the patient decides to drink some water, the treatment has to compensate for this, for example by prolonging it.

- Controlling the blood pump. Turning of the blood pump is a necessary emergency action. If something goes wrong, for example a leak is detected, turning of the pump is the most effective way to stop everything.
- Controlling the UF. If the patient decides to drink a glass of water containing, for example, 3dl of water, the UF volume have to be raised by 3dl. This is also, of course, closely related to the treatment time.
- Responding to alarms. Many alarms need no more than a little attention. It is rather seldom that anything physical on the machine will have to be remedied. An example is a patient that lies down. When shifting from a sitting to a lying position, the blood pressure will change, and probably set of the alarms for the blood pressure. All that has to be done is changing the alarm levels, and go on with the treatment.
  The alarms could be grouped into alarms that:
  - only needed attention. (Air Guard alarm is generally a good example)
  - were remedied by manipulating values, such as blood flow or TMP

I decided to let these tasks be represented in the remote control, and leave everything else out of it.

3.4 Conceptual Design

3.4.1 Expert or novice users?

The extensive training available to home HD patients is a crucial aspect in this case. They should be considered experts, and the system should be constructed with this in mind.

An expert generally prefers a faster GUI, with more information at a time, than novice users do. This could also be described as fewer, but bigger steps in the flow of events. Thus, a design with relatively few screens, and no unnecessary information or "do you really want to…"-questions was to be preferred.

Expert users tend to be working fast with GUIs, and it is therefore important to make the screens very distinct with the use of headings, colours and other means to prevent the users from changing the wrong values. They cannot be expected to carefully read instructions, since they already know them. These are rather general guidelines, but extra applicable to expert users. However, this is a medical application, and a UI that prevents errors in every way is most important. An error could result in severe medical problems.

I decided to rid my design of as many unnecessary questions and screens as possible, to make it fast. I also decided to give fast handling of the system higher priority than an explanatory interface, where I had to choose.

3.4.2 Separating and grouping functions

Making a clean distinction between different functions is necessary to avoid errors. It has to be obvious for the user what values are being manipulated. The major ways of doing this are colour coding, headings and symbols. How should these means be combined to save space, to be over obvious and to not steal too much attention from the task at hand? The users, being experts, will not read a header, but only give a quick glance. To keep the "knowledge in the world" instead of in the mind, colours are not enough, and symbols might be mistaken unless they are very well chosen.
The functions were grouped according to the main groups of tasks, mentioned above

.

3.5 Usability Goals

The extensive training and the fact that the chosen tasks were rather limited, and in many cases much alike, were the main arguments for these goals.

Primary goal: A user should be able to use the device without making any errors so serious, they cannot be remedied by simply pressing Cancel or otherwise backtracking, after the few weeks of training given at the hospital before the machine is taken home.

Secondary goal: Preferably, no training at all should be necessary, except a few hours of getting acquainted with the UI, if the user has sufficient knowledge of the AK200 to run it.

This can be tested, and if the preferred, higher goal is achieved, it can easily be verified here, at Gambro, provided such patients exist.

3.6 Design

The design limitations were those of the PDA, namely:
A rather small screen (240x360)
No keyboard, except the virtual one on the screen
The limits resulting from the java graphics package AWT.
I decided that one screen per major task, plus a few others containing confirmation, start screen with main menu and overview etc would suffice.

Based on that assumption, I started out making the first designs.

The first design consisted of a few types or groups of screens: The start screen, the screens dedicated to special parameters, such as Veinous Pressure or Blood Flow, and the error, alarm and confirm screens. I decided to make the start screen an overview screen, showing all the important values, but also a navigation screen, that should be able to take you to more specified views of the parameters. As everything went smoothly, you should not need to switch screens. Thus, I added a number of bar graphs representing those values and buttons, one for each parameter, and one for the blood pump, which is an emergency button that shuts of pretty much everything in the machine; hence the red colour. Due to the limited space, I had to choose between text and pictures for the buttons and settled for the pictures, as a result of the users being expert users, who would probably recognize images faster and more reliable than they would read labels. It would require a certain amount of learning, but given that I could pick the pictures well enough, their meaning would not be forgotten since the users used the application so much. However, I was unable to come up with a suitable icon for the ultra filtration.

In the screens for certain parameters, space was not quite as limited, and I could have not only the image, but also an obvious header for each screen. The diagram was put there, since it is present on the AK200, and I thought it carried important information. (Later studies showed that it was seldom used at all.) I decided to use a slide bar early, since there is no physical keyboard on the iPAQ, and the virtual keyboard is not only difficult to use, with very small keys, but also occupies about a third of the screen. A slide bar would give the user a quick way of altering values, when the exact figure was not all that important. Each value could thus be altered both by manipulating the slide bar, and by entering exact values into the boxes. The changes would then take effect when the “OK” button was pushed.

The first version had a lot of confirmation of different kinds. I believed every value had to be confirmed, in order to assure safety. Even though I had already concluded the users were experts, I put rather a lot of text and even a bar graph in the confirmation screens. This was to make it obvious what had changed, and how much.

The first few designs were constructed first with a pencil, and then in PowerPoint, and were not formally tested or evaluated at all. After a while I had something that I considered good enough to take to someone else for evaluation. I concentrated on the tasks, and the principles and guidelines mentioned in the prestudies when constructing the GUI from the beginning. Later on, these principles became less and less obvious in my choices, as the opinions of the users became more dominant

3.7 Testing / Evaluating / Redesigning

The evolution of the GUI can be seen in appendix.

The test subjects were at first planned to be existing home HD patients, but since they were very few, and none of them used the machine in question, that approach was abandoned. Instead, the attention was turned to nurses. A lot of nurses know the AK200 very well, and some of them are employed by Gambro AB, in great part for this kind of work, and are thus easily available.

Paper dummies (also known as Rapid Prototyping [7]) are proposed by many scientists in the MDI field for their obvious qualities of being fast, and thereby cheap. You can easily make changes and have new tests and thus get a lot of feedback. If you were to make the same changes to real, working software or, even worse, hardware, the time to complete the changes would have been several times longer. However, some things cannot be tested on paper dummies.

Animations, changing of values, and the trying and feeling of a system for example, are difficult. Thus, someone saying that one object invites you to try and move it when testing a paper dummy might say something completely different if presented with a user interface on a real screen.

The main steps of testing were as follows:

- Test paper dummies on the nurses
- Test working software on the nurses

3.7.1 Paper dummies

Some of the tested versions, (see 6.6.1, first design iteration), comments and changes can be found in the appendix, along with the cases used for this testing.

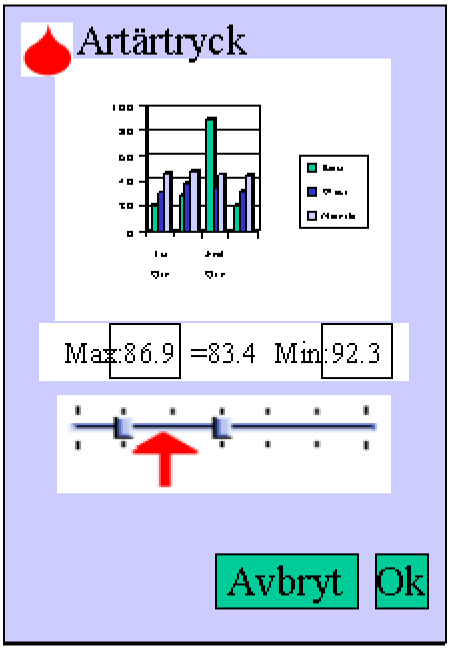
I decided to use a combination of two methods. The first was Think Aloud (suggested by Preece in 1994 [6]), where the test subjects tell the test conductor about what he believes he sees, what he does and why, and what he expects the result to be etc. The other was a simple, non-stereotype interview where I simply discussed the GUI with the test subject.

I constructed a few cases that I presented to my subjects (see appendix). Those cases were always the same, and they were constructed so that I asked the nurse to perform a rather high-level action, that she was familiar with on the AK200. They were of the kind “Adjust the alarm limits of the arterial pressure” and not “push the X-button”. I then studied her as she pretended to press buttons on my paper dummies, which were scale 1:1 and colour, and asked her to continuously tell me what she saw, and how she interpreted it. I asked specific questions only if there was an item on the screen, which she did not explain, so that I got her interpretation of every item in the GUI. I took detailed notes of all misconceptions I noticed. Sometimes the nurse needed an explanation of something that was not well designed to be able to continue, but I avoided interfering with her mental picture of the device as much as possible.

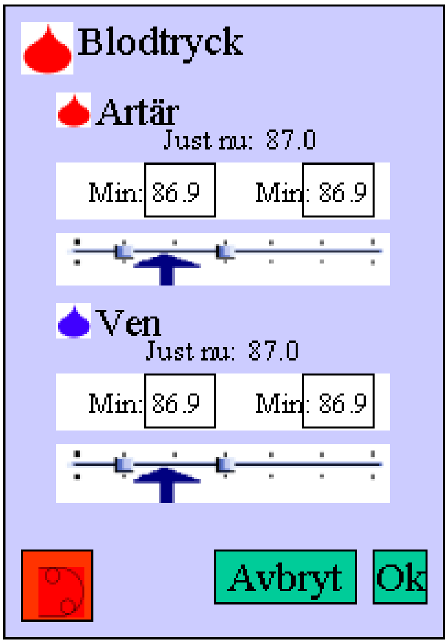
When all cases where finished, I explained everything that had been misunderstood, and then we had a discussion about what she felt was good or bad, and how she would have wanted it to work.

When I got back to my desk, I made a new prototype according to the notes I had made. Some comments I disregarded, since they did not fit in with the opinions of the other test subjects, and could thus be considered local to that user.

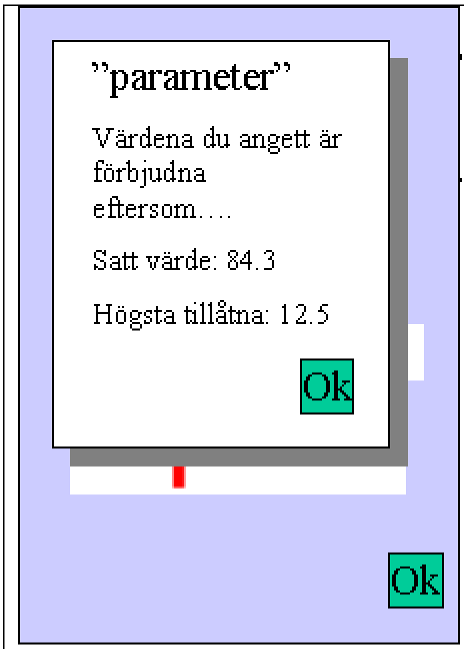
The results of these iterations were many, but I will mention the more important ones here.
The most obvious and important change was everything that disappeared. Between the first version in PowerPoint, and the last, 11 screens had shrunk to 8, and many of these pages were also less crowded. The two blood pressure screens had been merged into one (pictures 5 and 6), the page telling you of erroneous input disappeared (picture 7), as did the page telling you about automatic changes to some values as a result of your changing others (picture 8). The page with erroneous input disappeared when it was discovered that the user could be prevented from entering faulty values from the beginning, and the information about automatically changed values could be moved to the regular confirm page.



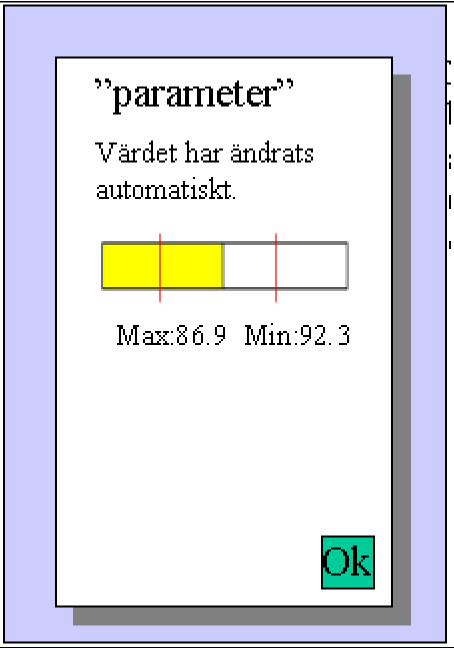
Picture 5 First version of Arterial Pressure, which is identical to the Veinous Pressure



Picture 6 Last version of Blood pressure screen, where both Arterial and Veinous pressure are represented.



Picture 7 Error message speaking about erroneous values. Later removed completely



Picture 8 Message telling you that some values have been automatically changed. Later removed completely.

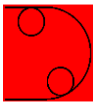
Although many symbols are far from perfect, I believe they are so good, that the users will remember what they represent. However, I have no way of testing this, due to my lack of test subjects.

In the first few iterations, things that were not clear were pointed out more often than later, but fewer opinions on the flow were expressed. Apart from that, some specific changes should be pointed out:

- Several pictures were not understood. One of them was the red arrow symbolizing blood flow (picture 9). It was later changed to the red drop with an arrow, which was better, but far from perfect. (Some have also found the filter with the arrow symbolizing TMP difficult until it was explained, when it was said to be very good. No good alternative could be found, and thus that was not changed!)

Picture 9 The red arrow

- The blood pump button (picture 10) was moved to the lower left corner since a desire to have it represented on every screen was expressed. As a result of the demand for consistency, it should be on the same location on every screen, and I thus had to move it.



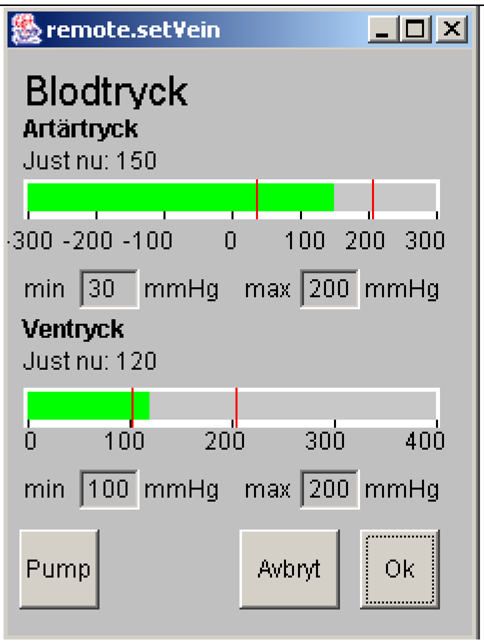
Picture 10 The blood pump button

- The information about the fact that certain values have been automatically changed, as a reaction to the change in other values, and the information on the screen informing that certain values entered by the user are forbidden, could be moved to the confirmation screen, and those two screens removed altogether, respectively.
- The diagrams were removed since they were not necessary, but occupied a lot of valuable space
- The screens concerned with blood pressures could be merged into one, once the diagrams mentioned before, were removed.

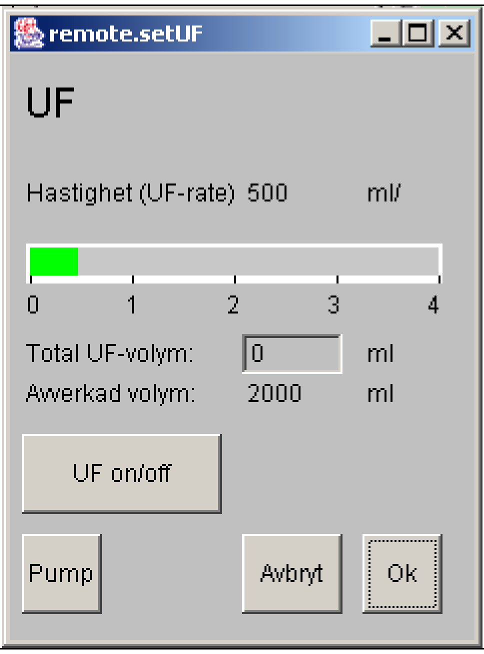
3.7.2 Software implementation of GUI

Implementing this GUI (two examples in pictures 11 and 12, and the entire GUI in appendix) would have proven no problem at all, using the eVT, or java and swing, but none of these were available as had been expected in the beginning. Thus, I had to sacrifice certain things. The slide bar, which I considered a very useful feature, and the symbols on the buttons, could not be implemented within reasonable time. As a result, the buttons on the implementation have text, and the slide bars are simple gone. (Examples in pictures 11 and 12) However, the bar graphs proved very simple indeed, and were implemented.

When implementing on the iPAQ, the ability to adjust fonts and other graphical details was somewhat limited. As a result of this, slight adjustments of the layout had to be done, in order to align the objects to suitable grids. However, this affected only one or two items, and the changes were minor.



Picture 11. The implemented version of blood pressure screen. Bar graphs and icons are not yet added.



Picture 12. The implemented version of the UF screen. Bar graphs and icons are not yet added.

Since we could not get the bluetooth connection between the iPAQ and anything else stable enough to run the application, it had to be tested on a laptop. However, it was the exact same size, looked the exact same way, and run, even without recompilation that way.

The goal of section 3.5 could not be verified literally, since no real patients exist. The nurses working at Gambro, on the other hand, had the necessary knowledge about the AK200, and no knowledge of the remote control, and thus I gave these nurses a short explanation of the UI and asked them to perform the same tasks, where implemented, that were used on the paper dummies when developing those.

The one thing that the migration from PDA to laptop prevented from testing, was the virtual keyboard used to enter values into the fields. This is much easier on a laptop with proper, physical keys. Thus the suspected need for the slide bar could not be verified, but Nurse2 tried to pull the alarm level indicators on the bar graphs to change their values, using them like slide bars, and she also expressed strong desire for them, pointing out the need for speed rather than absolute precision when setting these values.

The nurses that tested it were able to run it all by themselves within about 15 minutes of experimenting and asking of a few questions. This has to be considered, at least regarding those few individuals, as an achievement of the secondary goal in section 3.5.
The only thing that did cause a slight problem in the beginning was that certain buttons needed to be pressed, and kept pressed, for a certain amount of time. This is apparently not a good solution. Rather than letting these buttons work like on the AK200, logic to take care of the timing, which would allow the buttons to work like regular GUI buttons, is necessary.

Detailed opinions about the GUI can be found in the appendix.

3.8 Follow-up

This part was not included in the project, apart from the documentation, of which the major part is this report.

4. Software Development

I chose to make a distinction between the GUI development and the development of the rest of the software. This was partly because the main issue of this thesis was concerned with usability, but also because I believe the presentation would be easier. The main stages of development are rather similar, but the problems encountered are quite different. The technical issues regarding, for example, the communication between the AK200 and the iPAQ have very little bearing on the way users understand the application.

The code was developed and tested a part at a time. It can be divided into the following big chunks:

- User interface
- Decoding packages from the AK200
- Communication on the serial port
- Decoding packages coming in on the serial port
- Displaying the data from the serial port on the UI
- Constructing orders to be sent to the AK200
- Sending those orders over the serial connection
- Creating the orders from the UI
- Complete program

For most of these parts separate test programs were created. Testing a part at a time is a good idea, since smaller parts of code are easier to verify as to their correctness. It is also reassuring to know that most of the code you are using is more or less working, since it allows you to concentrate on the part at hand.

The final software is only a prototype, and it is important to remember this. Implementing a complete remote control would have taken substantially more time, and the main results of the thesis would not have been better or more obvious. Although many parts are designed with a sound object oriented approach and reusing in mind, other parts are rather crudely hard coded. No more work has been done regarding the software, than to make it work in the very limited laboratorial environment in which it will be demonstrated, which is in the labs at Gambro with no real patients and no demands to complete a treatment. Very little error handling of the kind medical devices require is there, and the set of functions actually available is very limited indeed. This is *not* a finished, working product.

Not all the functions originally intended and defined in the task analysis were implemented. This was due to a shortage of time. The severe technical problems discussed below took lots of valuable time, and more prioritising had to be done. Thus, not all values can be set, and only one alarm is handled. This should still be enough to evaluate the usefulness of the application, since most values are set the same way, and most alarms are dealt with the same way.

The functions implemented were:

- Displaying all values intended in the UI development
- Dealing with the Air Detect alarm
- Starting and stopping the blood pump
- Setting the Blood Flow Set value
- Cancelling that setting of blood flow (undo)
- The bar graphs

The functions that were not implemented were:

- No values can be set, except the Blood Flow Set value
- No alarms are handled, except Air Detect
- The Slide Bars are not present at all

4.1 Tools used

The main tools used for the software development were

- Metrowerks' CodeWarrior
- GenteWare's Poseidon.
- Locally developed emulator for the AK200
- Locally developed "COM port splitter"

MetroWerks' CodeWarrior is a complete java environment with compiler and project handling. All compilation was done here. Much, if not most, of the code was also written using this tool.

GentleWare's Poseidon is a freeware tool for creating UML diagrams, with which you can make much of the development through "drag-n-drop", which greatly improves code quality, by remembering everything you tend to forget and generating the standard code parts such as class definitions and so on.

The emulator for the AK200 was built by Roland Persson explicitly for this project, and made testing communication and package handling a lot easier. It has since been proven very useful to others at Gambro as well.

4.2 Design

Much of the design was made in Poseidon. Classes came and went a good deal before I started to write code. The design of the AK200_communicator package can be found in appendix XXX and it is rather similar to the C programs used by Gambro to extract logging data from the dialyser. This package was designed to be reusable. Reusing code has often been found to cost more than it yields, but with this part of the program, it was obvious from the start that it would be easy to construct a reusable package, that could be used in other applications without any extra effort on my part.

4.3 Problems encountered

The problems encountered during the software development were many and complicated. However, many good suggestions were found at Sun's java forums on the Internet.

Very little documentation about the CESerial package could be found, and getting it to work was rather difficult and most solutions were lucky guesses. The main issues here were configuration problems. The package as such follows the standards of the javax.comm., which is rather easy to apply to your own code.

Another source of problems were the rather complex protocols of communication with the AK200. The flexibility of these protocols is great, which naturally makes them complex, and thus rather difficult to code and use. I only use parts of it, and much is hard coded and loaded as static files from disc, to avoid tedious calculations that are not necessary.

Swing, which is the major standard graphics library for java also turned out to be out of the question, as a result of difficulties with different versions of java in different places. The tools for using Swing in CodeWarrior, and also, in part, Swing itself, demands a version of java greater than the one of the virtual machine I had available for the iPAQ. Thus, AWT was all there was.

The communication between the iPAQ and the AK200 proved more than difficult. The biggest issue of all was found when only the final testing was thought to remain. The iPAQ works in a very specific way regarding serial communication over bluetooth, in that it has one com-port for outgoing data, and another for incoming. This is rather unconventional, and in this case, impossible to do something about, since it is hard coded in the iPAQ. The AK200, on the other hand, has only one COM port. This means that the iPAQ can only communicate one way with it, choosing either outgoing or incoming data. The protocol, and the application to for that matter, demands two-way communication. The temporary solution was to construct a small application, capable of passing data on between, in one end, two ports for incoming and outgoing data respectively, and in the other end one port for two-way communication. Thus the iPAQ can talk through the "base station" of the laptop equipped with this "COM port software splitter". This is an ugly solution, which we nevertheless decided to go through with, in order to be able to perform GIU tests with a real machine and iPAQ. It was still important to verify whether the idea of a remote control implemented on a PDA was a good idea from the patient's point of view, regarding usability.

When running the final application, the iPAQ would often, more often than not, stop receiving data. When this happened during the initial communication, it had to be restarted. Otherwise, closing and opening the com-port would suffice. This is believed to be a bug in the bluetooth layer of the iPAQ. In the end the application could not be run on the iPAQ at all.

5. Discussion

5.1 The users

As the user profiling was only applied to the Swedish population, but Gambro is a worldwide company, a more thorough user profiling would be required before a commercial remote control was designed. Who becomes a home HD patient, and who does not, is not as simple as a health issue. It is also to some extent a cultural and political question. In many countries, money might be an important reason to keep patients at the clinic, since the clinic might earn more money that way. However, home HD treatment is cheaper than the clinic equivalent, which might mean we will see more home

HD treatment in the future, as people are ushered out of the clinics with the obvious argument that it saves money. These people will need “user-friendly” machines. Another possibility, of which I have absolutely no facts at all, is that people in some cultures might consider hospital treatment superior to home treatment in general, and therefore prefer not to have the machine at home. One can easily think of several other reasons for and against success such as this one. For example: In many countries, it is far to hospitals. The patients there would probably be happy to be able to get treatment at home.

According to one of the nurses with whom I spoke, the divorce rate among home HD patients is very high, compared to the country in general. Daily treatment and the constant need for someone to help you with it, might very well be straining on a marriage, why possibly many decline the offer to have their own machine at home. They simply wish to spare their spouse that extra strain.

It has been proposed that since apparently many are discouraged by the complex machine and treatment, and simply do not dare run it themselves, a “user-friendly” remote control could be a key to help more patients take that extra step. However, I personally do not believe that to be the case. This opinion, I base on the following facts and assumptions:

- The more complex and discouraging parts of the treatment cannot be replaced by a remote control, since they involve physical actions, such as hooking up the tubing, inserting needles into veins and arteries and connecting filters and chemical containers to the machine.
- A remote control does not eliminate the many values of which one needs to keep track.
- There is also still the possibly frightening fact, that should something go wrong, you will not be at a hospital with lots of competent, medically trained personnel and equipment. You will be more or less on your own at home.
- A remote control does not relieve your spouse of responsibility or the need to be there.
- It does not remove cultural or economic arguments for being at the clinics.

5.2 The tests

I would have liked to have more extensive tests, but the supply of test subjects and the time to perform these tests were limited. Altogether, 5 nurses altogether tested the paper dummies (one nurse tested every version, except the last paper version that was tested by three, and the implemented version, that was tested by four), and that I consider to be too few by a factor of roughly 5. Much literature, but most specifically Nielsen, 1994 [7] mentions numbers of test subjects for every iteration of about 4-8, to catch the diversity of people, of course depending on test method and application. In most cases the tests covering more than 8 give very little more information than 7. The extra test rounds can thus generally be considered a waste of time and money.

I feel I did not get all the feedback I would have wanted. I would have liked to follow specific people as the GUI evolved, to see what they considered improvements and deteriorations, but that was not possible. I was lucky my test subjects managed to squeeze in any time at all into their rather busy schedules.

5.3 The GUI

The differenses between the last PowerPoint design, and the one that was finally implemented are rather important. Some of the things I considered important could not be constructed with the means available. I believe, though I cannot prove this since I cannot test the two kinds under equal circumstances, that these changes made the UI extensively more difficult. I am primarily thinking of the images on the buttons and the slide bars. The result of the lack of slide bars is that the setting of values is no longer neither simple nor fast. You have to bring up the virtual keyboard, enter the new value, and then close the virtual keyboard again.

On the AK200, certain confirmations and other actions are done by pressing a key and holding it down for a certain amount of time, for example 2 seconds. Often, confirmations are made by pressing a button, that otherwise has a different meaning. To avoid this double, and often confusing, meaning of a button, other buttons have been used in the remote application that I thought would be more intuitive. However, the pressing and holding was implemented the same way in the application as on the machine for software technical reasons. It caused problems, since it turned out to be a bad mix of new thinking and keeping the old. While people have no trouble understanding that you keep the button pressed until the noise stops on the AK200, doing the same on a button that says “confirm” is in no way intuitive. People expect buttons in GUIs on computers to need only a slight tap. This is a failure.

Sometimes the argument that things should be changed to look like the UI on the AK200, for the sake of similarity could be heard. However, I strongly doubt that is correct, given the often bad choices made on the AK200. Furthermore, I strongly suspect such similarity, in itself, would not help anyone.

5.4 Problems during development

During the development of this application several things have gone wrong. I have here divided them into hardware and software problems, and left out what is not relevant to this report.

5.4.1 Hardware problems

The hardware problems are all due to the fact that the iPAQ used is quite new and not very well designed regarding hardware. It has been on the market for only a few months (spring 2003). One problem experienced in this respect was that the ROM image was not protected as well as it should be. The result of this was that upon running the debugger in the development environment recommended by both Microsoft and HP, the ROM image was destroyed beyond all recognition. The unit had to be sent back to HP to be rewritten. Thus, the tools to develop the application had to be reconsidered. What had been the obvious choice allowing all I needed, was suddenly no choice at all. By choosing CodeWarrior and java instead, this particular problem was solved, but several others arose. This issue could also possibly be relevant in other cases, such as development with different tools and safety issues. You never know when it might break. At the time when the my work was all but finished, un update for the iPAQ, fixing the security issue with the debugger in eVT was released, which allows Gambro, should they choose to continue this project, much better means to create a good application. However, it was far too late for me.

The communication problem regarding two versus one com-port (see chapter 4.3) was really lethal to the project. The solution with a laptop and a small application passing data on cannot be considered acceptable. The conclusion of this issue is that this application should not be built with today’s existing hardware, without creating extra hardware for the serial port on the AK200. This would be rather expensive, and is not really an option. Other PDAs, such as the Palm Tungsten might provide solutions, but then java is not an option, since the javax.comm. package apparently does not exist for that gadget.

The bug in the bluetooth layer of the iPAQ made it impossible to run the application on it even for tests. Such unstable devices can of course not be used in medical applications. It is possible that this could be solved, by rewriting the application in a more effective way, using C or C++ and coding for speed and low memory usage rather than good object orientation, but this is in no way verified.

5.4.2 Software problems

I was rather limited in the area of graphical packages. Where eVC++ had provided very simple and flexible tools for drawing and typing text, I now had to rely on packages with very strict and limited rule sets. The Swing package, giving a lot more freedom than the AWT package, was not available without fixing a bug in the source code for that package and not really even then. I was also limited to J2ME, which is only a subset of J2SE. It is doubtful whether this application should be built with java at all (see below also). Although this has to be regarded as a failure, I am confident, that once the hardware exists to make this remote control a good idea, the tools for developing will allow all of my intended GUI. Indeed, only days before the project was finished, HP released a software update for the iPAQ, fixing the bug preventing me from using the eVT.

The communication packages available are not many, and most of them are very poorly documented indeed, being open source projects at best. This renders unreasonably long development times, since a lot of it has to be spent on experiments.

Java is a slow language when it comes to execution. The figure 10:1 is often mentioned in the comparison between Java and traditionally compiled C++, although this figure varies widely depending on virtual machine and a lot of other things. This application, being run on a small device with rather limited hardware, needs all the speed and efficiency it can get. I doubt that the real-time and security demands that will be issued for this device by governments can be solved at all with java.

5.5 What is left to be done

5.5.1 Possible improvements

- A possibility to switch between “trainer mode” and “expert mode” as a way to switch different kinds of labelling, error codes, extra steps of confirmation and so on, would probably be a good idea. Users who are training before they take the machine home, might enjoy extra explanations, extra labels and so on, while expert users are only annoyed by things taking unnecessary time.
- Creating a compiled version in C++ would greatly improve execution. Everything would be faster, and thereby better, not only from the usability point of view, but also from real-time perspectives.
- Letting the software handle the “holding down” of certain buttons, so that no buttons ever need be pressed and held would make the GUI a lot more intuitive.
- Blood pressure function
- Summarizing function giving all the values in digits on one page. This would be valuable if the patient wanted to note all the values at certain time for logging. However, a logging function writing all values to file at certain intervals might be even better.
- A function spreading the alarm limits of, for example, Arterial Pressure to let them automatically centre around the current value.

5.5.2 Security issues

Several aspects of security should be considered before this remote control could be a commercial product. Among them are:

- **Integrity of transmitted and calculated values**
  How can we be sure that values are not multiplied by the wrong factors, that decimal points are not placed wrongly and that the packaging and un-packaging of values to be transmitted and received are handled correctly? If, for example, the remaining time or the fluid to be drawn from the body are miscalculated somewhere, it might cause serious harm to the patient.
- **Safety regarding the connection**
  How can we be sure we can trust the connection between the remote control and the dialysis machine? Could another device tamper with the data being transmitted? Could the loss of connection, due to interference of some kind, be a major problem? Can the protocols and other software mechanisms, used by the operating system, be trusted?
- **Hardware issues**
  How do we choose hardware that can be trusted? The first two HP iPAQs supplied for the project broke within a week as a result of a security bug.

- **Real-time issues**
  As a result of the application being written in java, it is rather slow. This leads to real-time issues, which could be a major hazard. It is also a usability issue, since slow responses are a significant source of annoyance among users.
- **Laws and government rules**
  What do government agencies such as Socialstyrelsen or the FDA demand from devices like this one, regarding the above issues? How do we test our product to convince these agencies, customers and ourselves that this technology can be trusted?

6. Final conclusion

My final conclusions are that the idea, and the application would indeed be useful for the home HD patients and make their treatments easier. The prototype I have built appears to have all the necessary functions, even though some of them are simply dummy versions, and all the necessary values displayed. It appears I have added nothing unnecessary to it.

However, after having used several different bluetooth devices from different brands, I have little faith in their quality, and I think it would be hard to find hardware reliable enough to create a commercial bluetooth remote control for the AK200.

7. References

1. Human Factors and the Design Process by Peter Andersson and Håkan Reis, Ergonomics and Aerosol technology Lund University, 1999
2. MUI: Controlling Equipment via Migrating User Interfaces by Torbjörn Eklund and David Svensson, Computer Science, Lund Institute of Technology 2002
3. Gambro AK200 Manual 2002
4. Att utreda, forska och rapportera by Lars Torsten Eriksson and Finn Widersheim-Paul, Liber Ekonomi, 7th edition, 2001
5. An introduction to usability by P. Jordan, London: Taylor & Francis Ltd. 1998
6. Human-Computer Interaction by J. Preece, Addison-Wesley Publishing Company, 1994.
7. Usability Engineering, Jacob Nielsen, Academic Press Inc. 1994, ISBN: 0-12-518406-9

http://www.swedetrack.com/images/bluet00.htm (bluetooth)
http://forums.itrc.hp.com/cm/CategoryHome/0,,273,00.html (hp forum)

Appendixes

I. Dictionary

UI – User Interface. The means of communication with the machine available to a user.
GUI – Graphical User Interface. The set of buttons, symbols and all other graphical ways of manipulating the system.
Home HD – Home Hemo Dialysis.
Bluetooth – Protocol for short-range communication, designed by Ericsson
PDA – "Personal Digital Assistant", a small, handheld computer, generally equipped with calendar, note-writing functions etc.

II. User test tasks

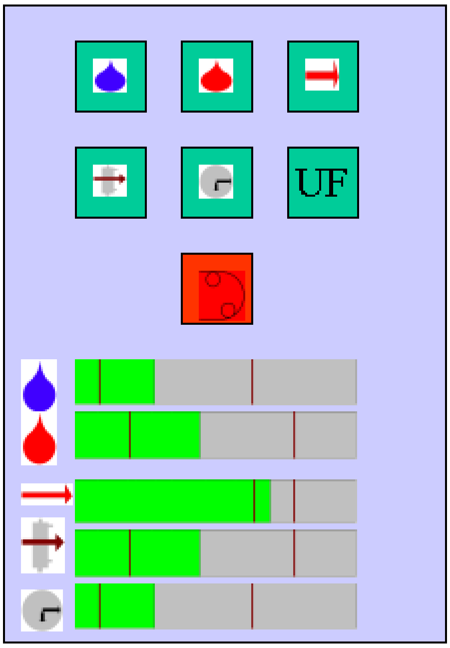
These are the tasks that the test subjects were requested to "perform" when testing. Whenever a screen was supposed to change, I changed the piece of paper accordingly. Most cases started on the main screen, described in section. They were selected to cover every button and screen.

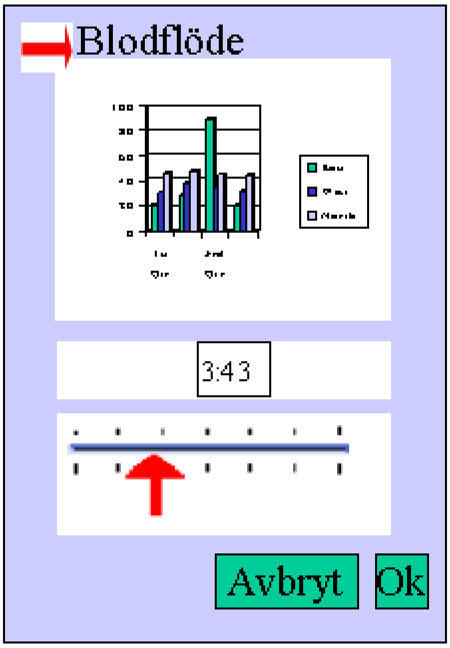
- Adjust the alarm limits for Veinous Pressure
    - Main screen
    - Veinous/Blood pressure screen
    - Confirm screen
    - "Info about consequence on other values" screen (only in the earliest versions)
- Adjust the alarm limits for TMP
    - Main screen
    - TMP screen
    - Confirm screen
    - "Info about consequence on other values" screen (only in the earliest versions)
- Add twenty minutes to the treatment time
    - Main screen
    - TIME screen
    - Confirm screen
    - "Info about consequence on other values" screen (only in the earliest versions)
- Respond to this "Air detector alarm" (starts on the alarm screen)
    - Alarm screen
- Respond to this "Veinous pressure alarm" (starts on the alarm screen)
    - Alarm screen
    - Veinous/Blood pressure screen
    - Confirm screen
    - "Info about consequence on other values" screen (only in the earliest versions)
- Add 0.3 litres to the UF Volume
    - Main screen
    - UF screen
    - Confirm screen
    - "Info about consequence on other values" screen (only in the earliest versions)
- Adjust the alarm limits for Veinous Pressure, and enter a faulty value (only in the earliest versions)
    - Main screen

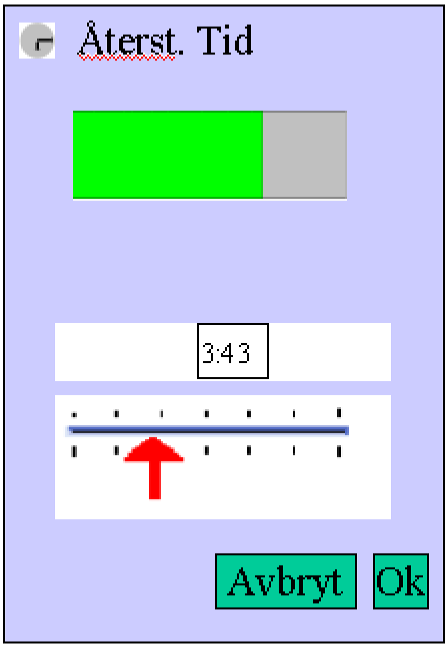
- Veinous Pressure screen
- Erroneous values screen

III. A few of the different version of the UI

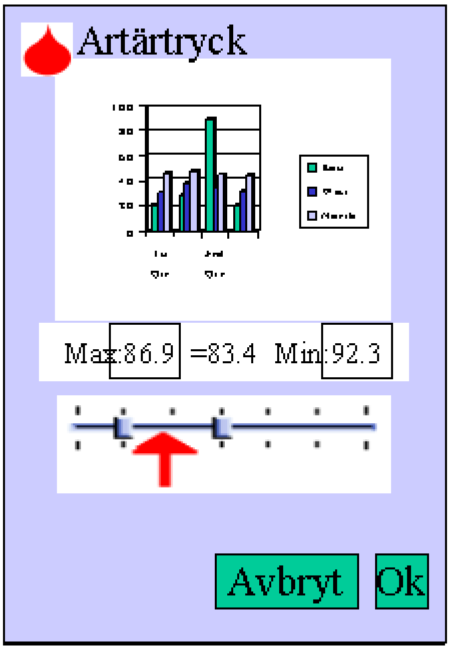
These are not all of the pictures, but merely the representative ones.
Ver: 030224
The diagrams are simply examples from Excel to show the layout. The intended diagrams were of course not bars.

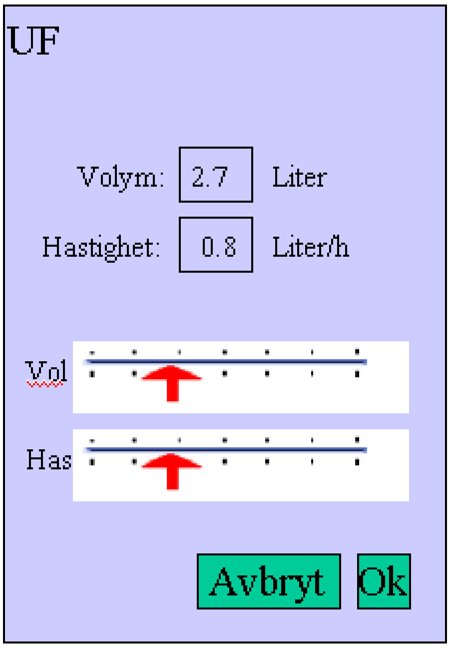


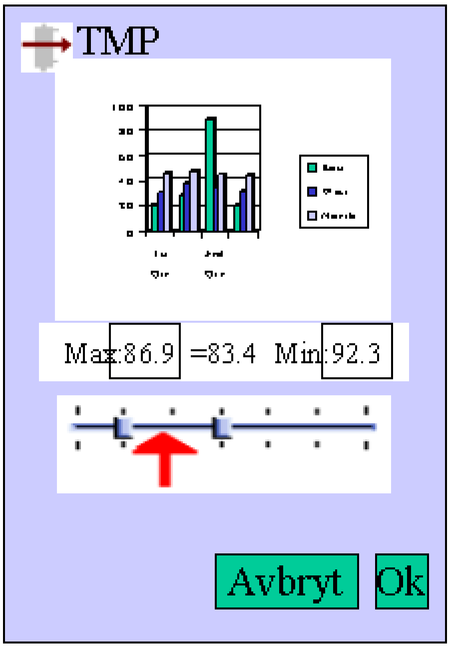




Main screen **Blood flow** **Time**





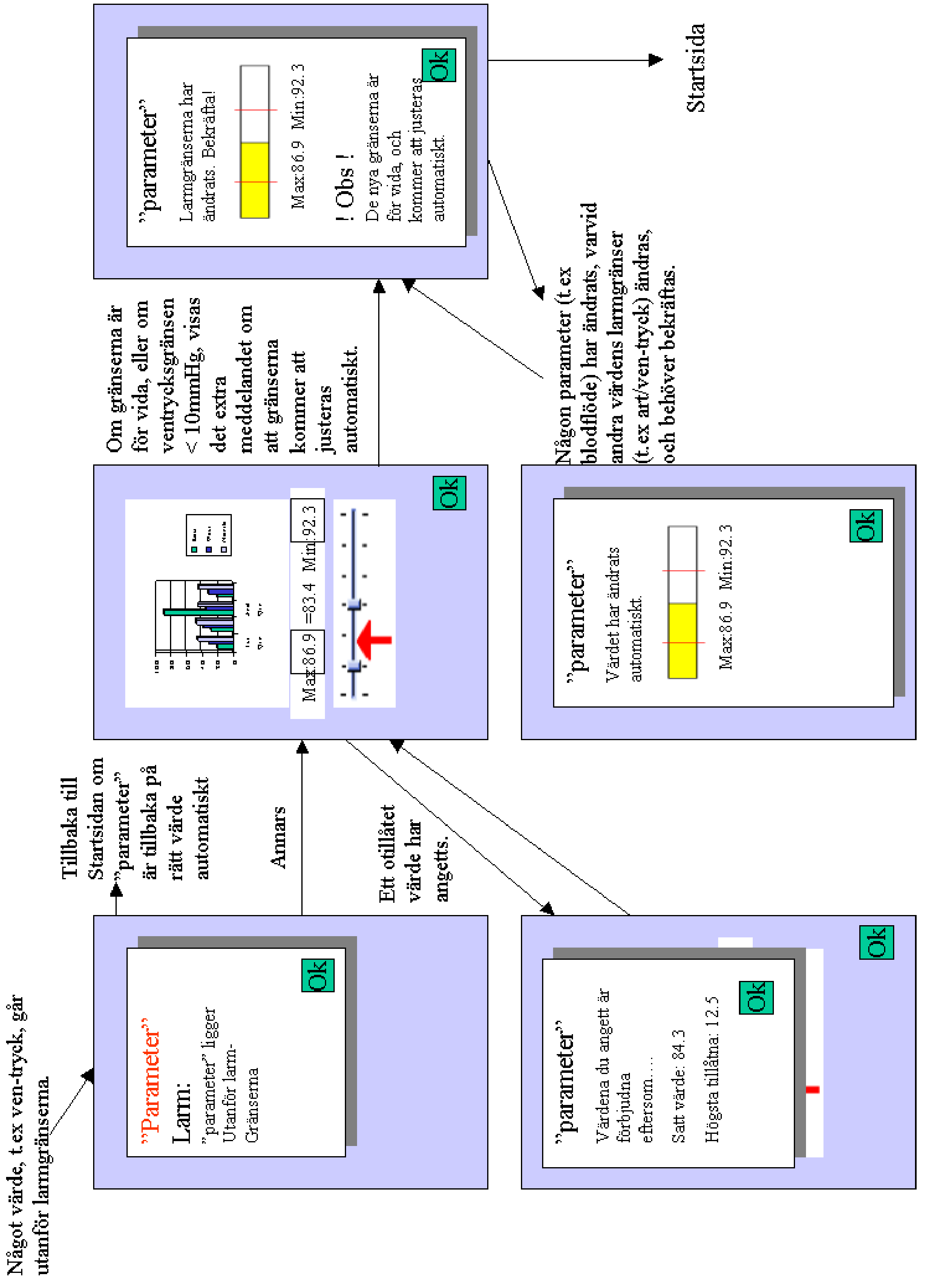


Arterial pressure (Veinous pressure is identical, except for a blue drop instead of a red one)

UF

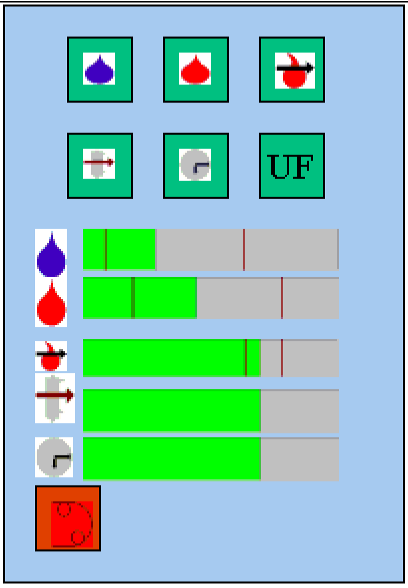
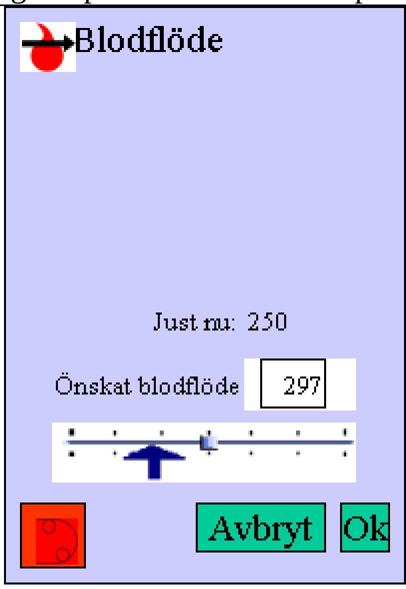
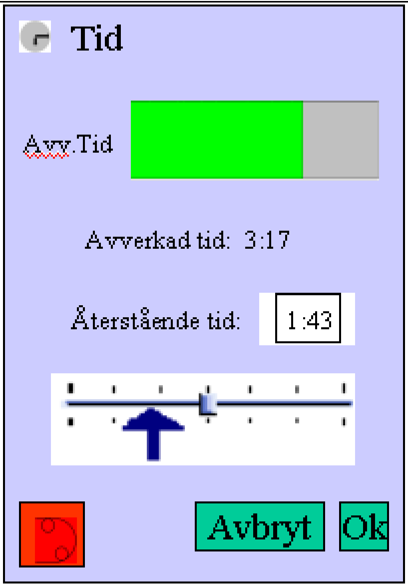
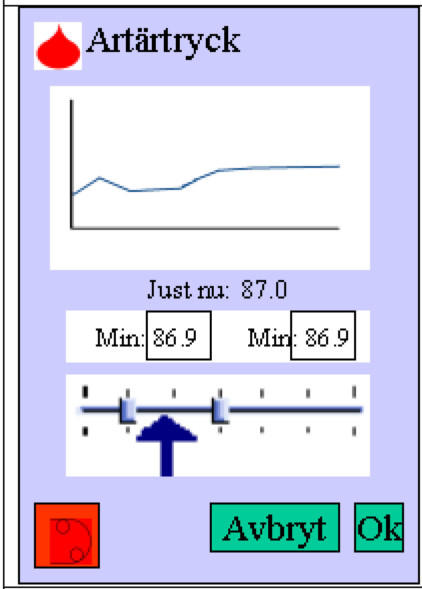
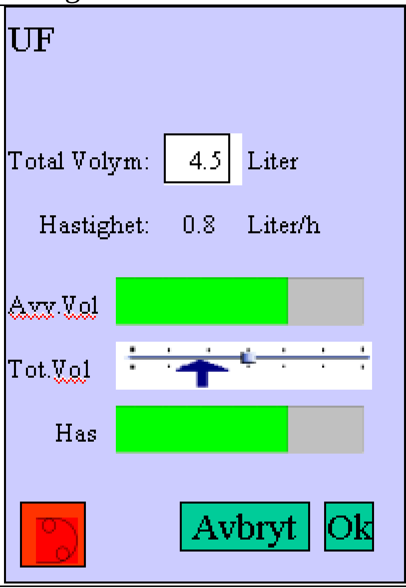
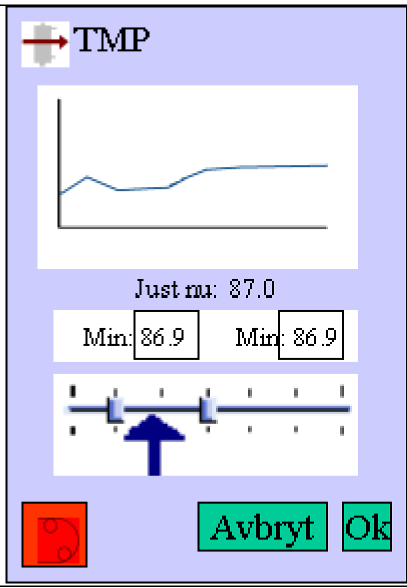
TMP

This picture explains how I originally intended to handle various errors that could arise, including alarms from the machine. It can be applied to any value, which is why the heading on all pictures reads "parameter"

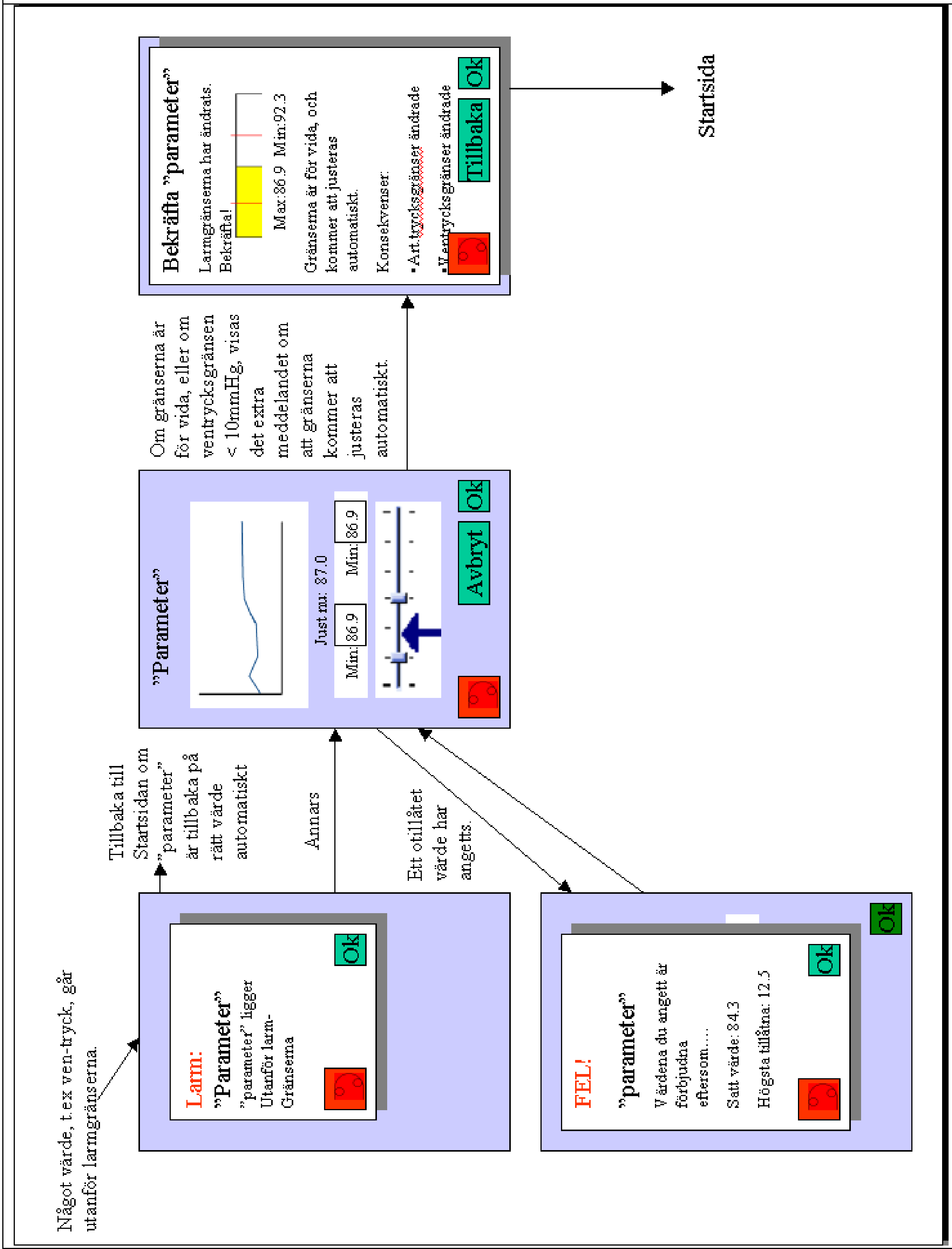


Ver: 030321

The pump button has moved to a fixed position, and is represented on all screens. Better explanations of values have been added. The diagram pictures have been replaced.

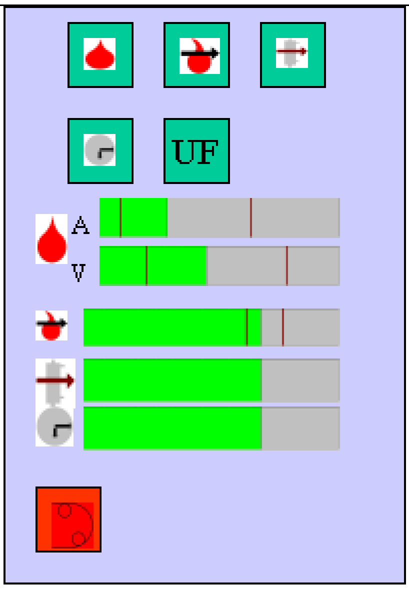
		
Main screen	**Blood flow**. The icon has been changed	**Time**
		
Art. Pressure (Vein. Pressure is still identical)	UF	TMP

In this picture you can see how the information about automatically changed values has moved to the confirm screen.

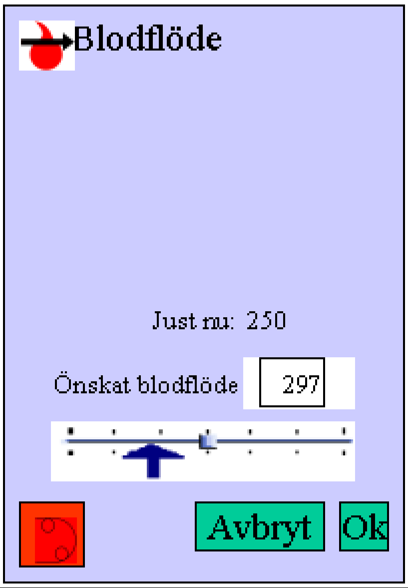


Ver: 030402

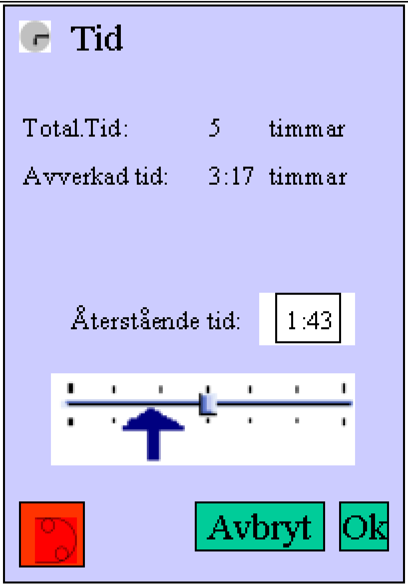
The diagrams have been deemed unnecessary, and removed. A certain regrouping on many screens has been done.



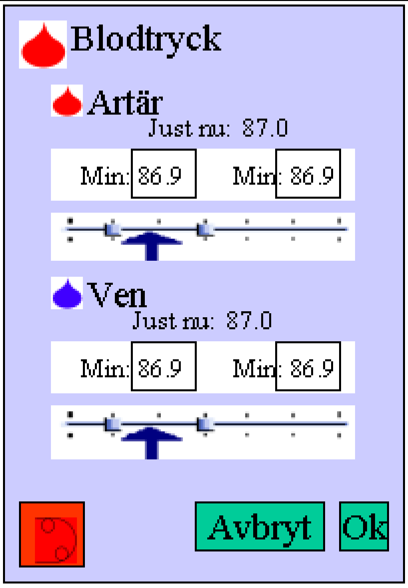
Main screen A blood pressure button was removed. The icons for blood were changed



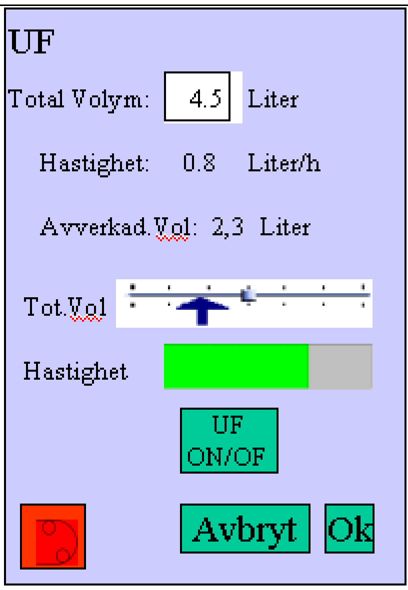
Blood flow



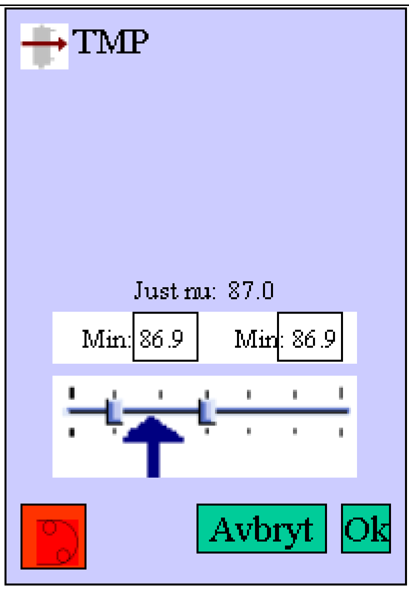
Time



Blood Pressure Veinous and Arterial blood pressure are represented on the same screen.

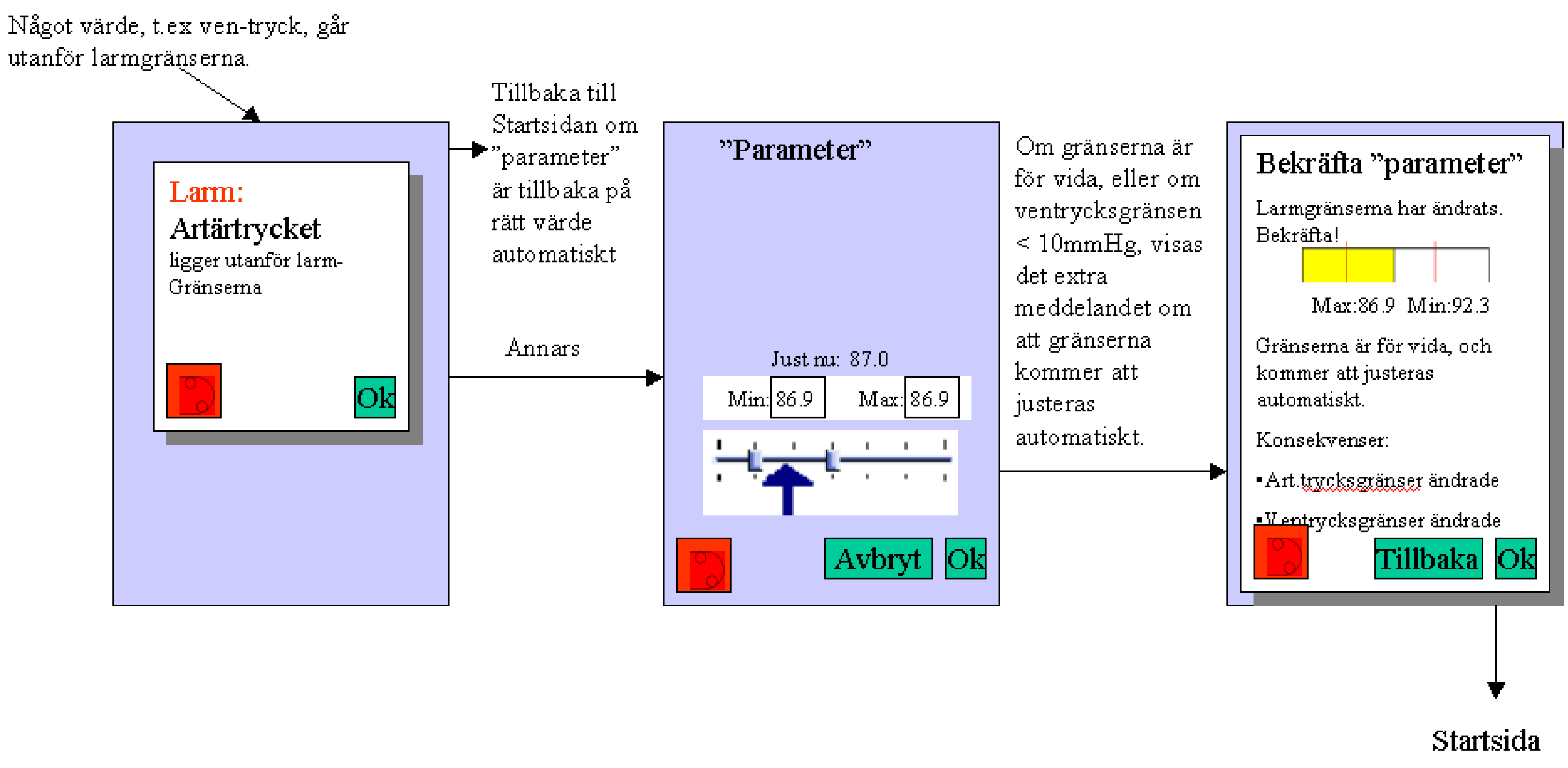


UF A UF ON/OFF has been added. It is still inconsequent regarding choice of language.



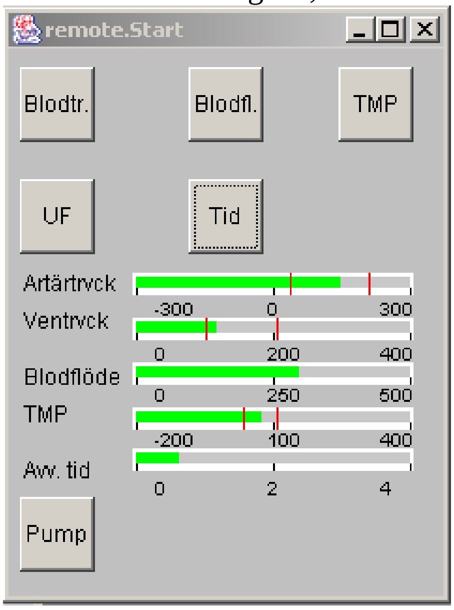
TMP

By removing two unnecessary screens, the alarm flow has been reduced to something rather simple.



Ver: final, implemented version

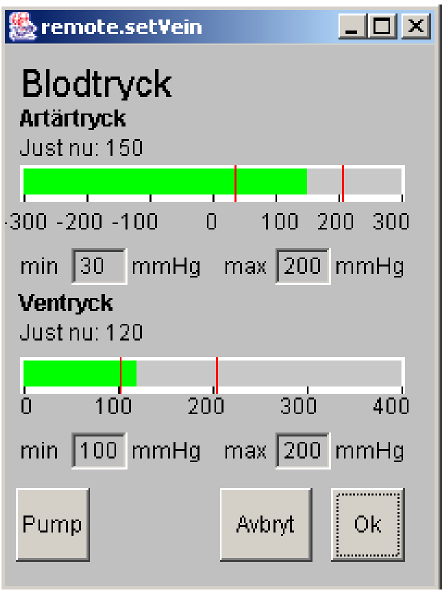
The slide bars are gone, and so are the pictures, which have been replaced by text.



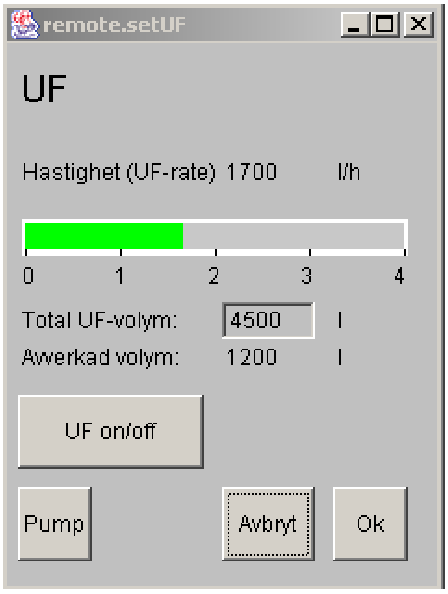
Main screen

Blood flow

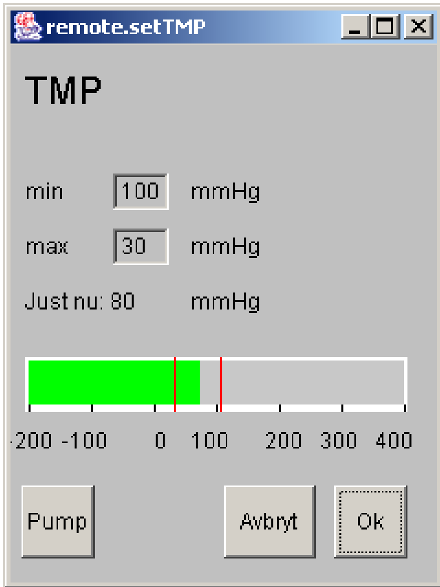
Time



Blood Pressure



UF

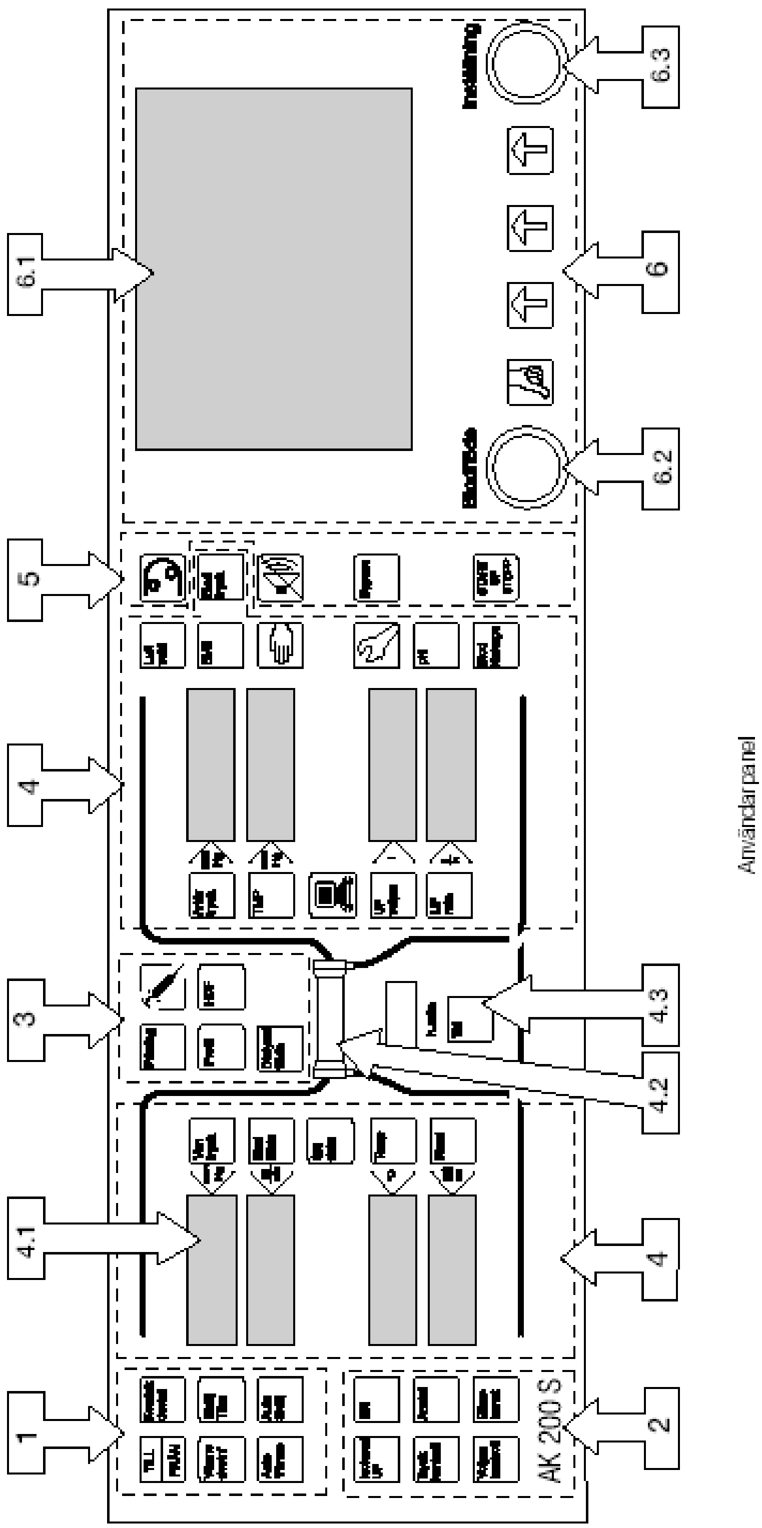


TMP

Alarm

Confirm

User Panel (Translate!!!)



1. **Hygiene group**
2. **Treatment group 1**
3. **Treatment group 2**
4. **Supervision group**

4.1. Graphical display
4.2. Flow diagram
4.3. Time display

5. **Stand-by group**
6. **Display- and settings group**

6.1. Information display
6.2. Blood flow knob
6.3. Settings knob

IV. User Profiling/ Task Analysis interviews

These are the answers from the Home HD patients and the nurses that were interviewed as a part of the user profiling and the task analysis. Naturally, the two Home HD patients were not asked about the tasks, since they know nothing about the AK200 or at least very little.

Below are the two sets of questions on which the interviews were based. Those interviews were rather informal and these questionnaires were not followed blindly.

The text in bold among the patients answers indicates, in large, what question the answer refers to.

1. Home HD Patient questionnaire.

1. Personal data
    - Age
    - Gender
    - General health state

2. Disabilities
    - Bad eyesight
    - Movement
    - Rheumatism or similar
    - Diabetes
    - Other disabilities
3. Where do you live?
4. How well do you know the machine you use?
    - How long have you been using it?
    - How much training have you had?
    - How long have you suffered from kidney failure?
    - Did you use Home HD from the beginning?
    - How much training/education in the science behind the treatment do you have?
        1. Do you know the chemistry?
        2. Do you know the mechanics?
        3. Can you repair it yourself?
5. Who helps you with your treatment?
    - Wife
    - Nurse
    - Other relative
    - Nobody
6. How well is he/she trained and what is the level of his/her technical understanding? (See question 4)
7. Are there any important differences between the treatment performed at a clinic and the treatment performed at home?
    - Time
    - Preparations
    - General types of treatment
8. What do you do to pass time while treating yourself at home?

2. Dialysis nurse questionnaire

1. What is the common home dialysis patient like?
   - Age
   - Gender
   - General health state
   - Disabilities
2. How many people treat themselves at home?
3. Where do they live?
4. How well do they know the machine they use?
5. How long have they been using it?
6. How much training have they had?
7. How long have they been dialysis patients?
   - Are they brand new in the game?
   - Have they been going to clinics for years?
8. How much training in the science behind the treatment?
   - Are they simply trained in what buttons to push?
   - Do they understand what the buttons mean, and what they are doing in a more scientific sense (chemistry, mechanics, biology)?
9. What does a typical home treatment look like?
10. Are there any important differences between the treatment performed at a clinic and the treatment performed at home?
    - Time
    - Preparations
    - General types of treatment
    - Settings on machines
    - Errors / Error handling
11. What do patients do to pass time while treating themselves at home?
12. How often, and what kinds of alarms stop the machine in a typical treatment? (Air alarm, leakage, chemical issues…)
13. What issues could be solved from a PDA, and what issues would a patient have to solve at the physical machine?
14. What information is necessary on a remote control? All information on the AK200 should probably not be displayed. At least not all at once
    - Is there anything that needs to be visible all the time?
    - Is there anything that hardly ever needs to be visible?
    - Are there any values that need changing during the treatment, or are they all set at the beginning, and then only changed in case of unpredicted events?

3. Nurse 1's answers

The HomeD patients are younger and generally in better shape than those being treated at the clinics. This is both a result of the better treatment and a necessity for being able to perform it.

Depending on what kind of treatment is referred to, many might have diabetes and bad eyesight as a result. *(Note: This answer was later argued against by other interviewees)*

In Sweden, there are about 250 home HD patients, very roughly, spread all over the country.

Most patients who are offered home dialysis are new to it, and have not received any treatment before. Those who are treated at the clinics are usually in too bad a shape to be able to treat themselves at home, and the home dialysis programme has been in use for quite some time, which means no patients are being transferred home from the clinics.

The patients taking treatment upon them selves are generally also mentally stronger, which is necessary to dare take care of such a delicate matter. As a result, they could be expected to be somewhat keen on new gadgets or at least not overly cautious of new ideas such as remote controls. They could also be expected to have a thorough knowledge of the machine, since they have a great interest in knowing what they are doing with it. In many cases, they even repair them at home on their own. The extra understanding, compared to just being able to operate it, should make them more susceptible for a GUI that differs some from that found on the machine, since they probably have a mental picture of the AK200 that is very realistic.

The treatment performed at home differs very little from the treatment performed at the clinics in most respects, except for the extent of the treatment. HomeD patients treat themselves either during the nights, at least three times a week, all night (over 8 hours) or for a few hours every day. The mere extra time spent in the machine makes the patients a lot healthier. Those who do not sleep during treatment can be expected to spend their time on anything that they can within the limits of the needles and tubes. (Playing the piano, reading books, watching TV etc)

The most common issues that can be expected to be remedied by a remote control are the artery/vein pressure, and UF adjustment. The artery/vein alarms are usually raised when a needle is not positioned in the best way, causing blocks. This is fixed by the patient, on the patient, and then the procedure can be turned on again. The UF settings control the amount of fluid removed from the patient, and the rate is often adjusted along the treatment depending on how the patient feels, or if he, for example, decides to have something to drink.

Many might suffer from rheumatism or similar conditions due to their kidney failures, which might make handling small devices difficult and/or painful. *(Note: This answer was later argued against by other interviewees, claiming that patients with these conditions are generally treated at the hospital)*

4. Nurse 2's answers

The patients are all ages between 10 and 70, and in rather good physical condition, apart from their kidney conditions. They usually don't have any special disabilities. *(Disagrees with Nurse 1)*

Most patients have a relative or another at home helping them, or at least offering some sort of emergency backup, while a few do not. They generally have some kind of emergency alarm connected to an emergency central.

The clinic at Lund University Hospital cater to 37 patients in all of southern Sweden, although most of them in Skåne and Blekinge. These 37 constitute at least half of the entire group of HomeD patients in Sweden. (The low number might render difficulties when it comes to finding suitable prospects for interviews and user testing.) New patients have been starting home treatment at a steady flow for the last 30 years.

Most of them use Fresenius machines, but a few in Linköping use Gambro. I got the impression, although it was not at all expressively said so, that this might be because of the poor usability of the Gambro AK95 in a home environment. The AK200 is not used for home treatment at all in Sweden.

Most patients treat themselves from the start, and spend, generally, about 25 treatments at the Home HD ward training before they take their machine home. They are exceptionally good at handling their machines, and know them inside out, although they are no longer allowed to repair them themselves, since modern machines are to complex with lots of electronics, and so on.

There is no important differences between home dialysis and dialysis performed at a clinic, when it comes to the actual treatment. “It can only be done one way”.

Most patients spend their time in the machine watching TV or sleeping, or possibly reading a book. It is a quit and peaceful time.

The issues generally appearing during treatment with the AK200 are art/vein pressure alarms and UF adjustments.

What you need to adjust and oversee during treatment is mainly blood pressure, UF (rate and total), blood flow and limits of a few different values. *(Later found out to be TMP, time and blood flow)*

Cond (Conductivity indicating the concentration of the dialysis fluid) and temp (temperature of the dialysis fluid), however, are values that are rarely interesting.

5. Nurse 3’s answers

Dialysis patients are generally in good physical shape apart from their kidney malfunctions. They are also rather mentally strong, which is necessary to go through with such a project. Although many dialysis patients also have diabetes, which is why their kidneys do not work, these patients are usually not home HD patients. *(In response to the statement by Nurse 1)*

They know their machine very well, and have a thorough understanding of its workings when it comes to fluids, chemicals, and principles governing to process. They can be expected to understand new user interfaces relatively good.

Common issues during treatment are air alarms and blood pressure alarms.

Attractive functions on a remote would be

- Clear display of alarms and handling of the same
- Seeing, and adjusting blood flow
- Adjusting alarm limits
- Adjusting TMP
- Adjusting UF

Attractive information on the remote would be

- Remaining time
- Blood flow
- Art/Vein pressures
- UF (Rate & Total)

Completely uninteresting values

- Conductivity

6. Patient 1’s answers

Age: 63
Sex: Male
General Health state (apart from kidney condition): Quite healthy
Disabilities: No disabilities
Lives in Ronneby
Machine: Gambro AK95 (is changing to Fresenius at the moment)

Knowledge of machine
Lars Karlsson has very good understanding of his machine, and, being an electrician, he has even repaired it at times by himself. He also has extensive knowledge of the chemical and mechanical aspects of the treatment, and knows fully well what pushing a button actually does.

Help with treatment
His wife acts as backup at home. She has rather little education on the machine, but has, by way of practice, also come to know it quite well. She helps with certain aspects of the treatment by securing the tubes to Lars’ arm and taking care of various more physical tasks, such as emptying bags and changing chemicals when the containers are empty.

Amount of treatment
The treatment is about 6 hours every night, although, occasionally, the odd night is skipped.

Occupation during treatment
He sleeps

7. Patient 2’s answers

Age: 65
Sex: Male
General Health state (apart from kidney condition): Quite healthy
Disabilities: No disabilities, except a need for glasses while reading and a somewhat aching knee.
Lives in Ängelholm
Machine: Fresenius (Thus, no questions regarding the task analysis)

Knowledge of machine
Jan-Erik believes he has extensive knowledge of the chemical and mechanical aspects of the treatment, and knows fully well what pushing a button actually does. He claims to have a thorough knowledge on the whole of his treatment and the machine.

Help with treatment
His wife aids him, and performs certain tasks, such as “dressing” the machine, securing the tubes to Jan-Erik (not the pinching though), taking blood samples and doing the paperwork.
She had two weeks of training be the end of Jan-Erik’s own training, and has also naturally picked up a great deal of knowledge during the several years the treatment has lasted.

Amount of treatment
The treatment is about 6 hours every other night.

Occupation during treatment
He sleeps

V. Test results, paper dummies

The results of the tests are ordered by screen.

1. Nurse 1, V030320

Start

- The icons symbolizing Blood Flow and TMP were not understood immediately. They will have to be changed or replace with text.
- All other icons and text was fully clear.

Set Art/Vein Pressure

- The red arrow was the first choice of moving. However, with the user interface on a real screen, this might very well not be the case, since it will then move slightly, when the value it symbolizes changes over time. Red is probably not a good choice though. Something more discreet would probably be better. Try blue!
- Next choice for manipulation was the correct little handles. Later on, this was never a problem.
- The possibility of changing the figures directly was also rather obvious, although it might be a result of a leading question on the interviewer's part.
- The Current Value is neither obviously the current value nor easy to see. This will have to be changed dramatically. Maybe the Current Value's proper location is in the lower, right hand corner of the diagram? Try it! (More realistic diagrams will be needed!)

Confirm

- All values to be confirmed should be put on one screen to limit the pressings of buttons. Auto adjusted values should be put in some kind of list. Experts will not need to know them, unless something goes wrong.

Set Time

- Remaining time was interpreted as Total Time. For other parameters, similar problems were detected when on value, being the Current Value and one value, representing the Desired Value, were either not both represented or difficult to tell apart. This has to be changed to be more obvious. One value should represent the Current Value, and should obviously not be changeable and the other should be the desired value, and should easily be recognized as changeable.
- The Set Time picture appeared to be generally difficult, although the well-defined problems were few.

Set UF

- The text labels were inadequate. Some were not even there, and that shall have to be remedied. Remaining Volume was interpreted as the total Volume (see Time, for similar problem)

Set Blood Flow

- The icon for Set Blood Flow was inferior. It could not be guessed, so an alternative will have to be found.
- The same problem regarding what is the Current Value, and what is the Desired Value arose. Thus, two values, clearly labelled, should be put on that page.

2. Nurse 2, V030321

General opinions

- Gunilla believed it to be easier to write the figures, than to handle the slides.
- The OK button was not used consistently. Sometimes it took you to the Start page, and sometimes somewhere else. Some clarification on some of the buttons marked OK would be a good idea.
- The blood pump icon should always be accessible for safety reasons.
- The drops symbolizing vein/art pressure should perhaps be shaped differently, to avoid the misconception that they symbolized fluids. Perhaps just round "blobs".

Start

- The icons symbolizing Blood Flow (changed) and TMP were understood immediately.
- The blood drops were at first misunderstood as meaning d.fluid and blood, but she understood by herself very quickly. They will do, for now.

Set Art/Vein Pressure

- No comments

Confirm

- No comments

Set Time

- No comments

Set UF

- Apparently the UF rate is hardly ever set, except in special kinds of treatments (Isolated UF). This is mostly just a calculated result from time and total UF volume. It'll have to go. In the next version, this value will be read only. It could be added as a new function, but that new function seems to be rather unnecessary and would thus be more of a nuisance than an aid.

Set Blood Flow

- The new icon was understood immediately.
- *Personal reflection:* Can the puller be dangerous? Can the blood flow by accident be set far to high?

3. Colleagues, V030321General

The two colleagues interviewed are both very familiar with the machine and how it works, being engineers involved in the development of it. However, they are only moderately familiar with the dialysis treatment at the clinics.

- The diagrams were considered not very necessary, and will be removed, until there is lots of time to fix them.

Start

- The icons symbolizing Blood Flow (changed) and TMP were understood immediately.
- The blood drops were at first misunderstood as meaning d.fluid and blood, but she understood by herself very quickly. They will do, for now. They will be put together as one button.
- A possible major change would be to keep the buttons at the top, and changing only the bar graphs for the settings of values. This has to be considered, and perhaps tested parallel to the GUI of today.

Set Art/Vein Pressure

- No comments

Confirm

- No comments

Set Time

- No comments

Set UF

- Apparently the UF rate is hardly ever set, except in special kinds of treatments. This is mostly just a calculated result from time and total UF volume. It'll have to go. In the next version, this value will be read only.
- UF Start/Stop was requested, and will be there in the next version.

Set Blood Flow

- The new icon was understood immediately.
- *Personal reflection:* Can the puller be dangerous? Can the blood flow by accident be set far to high?

Alarm

The flow when handling several alarms is not clear. Should every alarm be handled separately? Should the system prioritise? Should there be a list of what other alarms are at hand? Should the user be able to make his/her own choices in the handling of the alarms, and thereby overriding the systems priorities? The answer to all these questions is "Yes, but of course!" That will be added in the next version.

4. MDI expert at Gambro, V030328

This was not a proper user test, but more of a "feedback meeting". Popup windows for input might be a good idea to help people see what they enter, and also to provide buttons for figures, and thus avoiding the regular keyboard, which is way to big.

General

- She believed that it might be necessary to include a "UF Start/Stop" button for every screen.
- She also mentioned the already known problem of the screen being small, giving people with weak eyesight problems.

Start

- The icon for Vein has to go, although both pressure bars are necessary.
- A bar for UF is needed, but the TMP is not all that acute. This has to be checked with nurses…

Set Art/Vein Pressure

- No comments

Confirm

- The confirm page is not always needed. When, for example, the limit window for the veinous pressure is made smaller, no confirmation is needed.
- The message has to be rephrased, to show that the automatic changes to any values (if any) *will* take place once the button is pressed, but are not yet handled. Possibly this should go away by itself after a certain time.

Set Time

- The bar graph could be gotten rid of. No need for it.
- A discussion without conclusion about what times should be seen was held. We do not want to many values, but at the same time, we have to show what's necessary. After checking the physical AK200, it was shown that remaining time and spent time are the values you are given.

Set UF

- The bar graph representing completed UF volume should perhaps be a figure instead. The abbreviation "Avv." was not clear.
- The "UF On/Off" button should perhaps be put on every page, just like the pump.

- The abbreviation “Has” was not recognized.

Set Blood Flow

- The blood flow screen could probably benefit from the current values of blood pressures. They are apparently desired values when setting this value.

Alarm

- The alarm page was considered a good idea, and the ideas from Jan & Roland about prioritising the alarms was also welcomed.

Error

- If the rest of the system is done cleverly, the Error page will loose it reason to live. This might be a setting available from some kind of setup menu, where you choose novice or expert interface.

5. Park Dialys, V030402

As very little useful criticism was brought forth, I think it is time to start testing the GUI on the PDA instead. Thus, that GUI needs to be built. Practically all comments were a result of the GUI being printed on paper, and thus somewhat difficult. Input fields were not distinguishable from other text and there was very little difference between pictures and buttons.

There is the issue of weather “fluid flow” show be represented in some way. However, we decided that it should not, at the moment, be included. Only the standard treatment should be implemented now.

6. Park Dialys Nurse 1

My personal guess is that this nurse had a rather limited experience of computers. In the future, a few data of this kind (age, sex, computer experience) should be recorded.

Once the general concept of what was buttons and what was not and what the point was with the sliders, she had no comments.

General

- A helping text on the buttons would be good. (*I doubt that is in any way necessary, since the users will be experts, and will learn the icons very fast)*

Start

- The “Blood Flow” icon was mistaken for “fluid flow”.
- The clock was not immediately understood. This was due to bad printing quality.

Set Art/Vein Pressure

- The entering of values was not understood. It took some time before the concept of multiple ways to enter values homed. Once again, the limits of PowerPoint as GUI creation tools were obvious. Indeed, the possibility of entering values here at all was hardly grasped at all.
- The pictures of blood drops, indicating where in flow we were, were mistaken for buttons.
- The slides were interpreted as preset values.
- A scale for the slide was requested. Of course there should be figures present.

Confirm

No comments.

Set Time

No comments.

Set UF

No comments.

Set Blood Flow

No comments.

Alarm

No comments.

Error

No comments.

7. Park Dialys Nurse 2

This woman was younger than the previous, and appeared to have a far better grasp of computers in general. She had no problem with buttons, slides and the entering of numbers.

General

No comments

Start

- The “TMP” icon was mistaken for “fluid flow”.
- “Blood Flow” was mistaken for “Blood path/tubing”
- The “Blood” icon could not be fitted in, after the other guesses were made, and a simple “I don’t understand it” was given.
- The clock was immediately understood.

Set Art/Vein Pressure

- Once it was understood that nothing needed to be activated, but that it was the controls that were visible right there. I believe this was due to a prototype conceived in PowerPoint. A proper GUI on a real machine would probably be more obvious regarding affordance.

Confirm

No comments.

Set Time

- Expected to be taken to the setting of blood pressure limits, until she saw that there were mentioned, if needed, on the Confirm screen.

Set UF

No comments.

Set Blood Flow

No comments.

Alarm

No comments.

Error

No comments.

8. Park Dialys Nurse 3

General

- Do we need the UF settings to be easier available?
- We should be able to see the blood pressure data given by the meter for that. We do not know weather that tool is even available to the HomeD patients.

Start

- Blood drop mistaken for “Blood Flow” or possibly “start/stop pump”.

- The “Blood Flow” was not at all understood
- TMP was mistaken by “fluid flow”

Set Art/Vein Pressure

- A button for automatic setting of limits is requested. This might be a very good idea.

Confirm

No comments.

Set Time

- Expected to be taken to the setting of blood pressure limits, until she saw that there were mentioned, if needed, on the Confirm screen.

Set UF

Slide was misinterpreted for “total” in some way. Once again we appear to suffer from the limits of a paper dummy created in PowerPoint.

Set Blood Flow

No comments.

Alarm

No comments.

Error

The label “OK” should probably be different. “OK” might mean that you not only acknowledge the alarm, but also simply turn it off without any action. This might be true in some cases, like the air alarm, but in most cases you probably want to show that this is only the alarm – the fix will be next step.

VI. Test results, software application

The results of the tests are ordered by screen. The tests were not run on the iPAQ, since it simply did not work. Instead it was run on a laptop.

1. Nurse 1, implemented version

General

- Nurse 1 expected some kind of connection between the pump and the blood flow values, although no such connection in the UI of the AK200 exists.
- All values, functions and buttons necessary for such an application are present and easy to find
- Nurse1 could run the application within 10 minutes of experimentation, and thus, the goals for the GUI are met, regarding her.
- Appears to cover all needs.

Start

- The first page gives a good overview, containing the right things

Set Art/Vein Pressure

- No comments

Confirm

- No comments

Set Time

- No comments

Set TMP

- No comments

Set UF

- No comments

Set Blood Flow

- No comments

Alarm

- Nurse 1 expected the Confirm-button on the alarm page to work like computers usually do, and it took a few tries, before she noticed that she needed to hold the button down, just like on the AK200

2. Nurse 2, implemented version

General

- All values, functions and buttons necessary for such an application are present and easy to find, with two exceptions listed below.
- Nurse2 could run the application within 10 minutes of experimentation, and thus, the goals for the GUI are met, regarding her.
- Appears to cover all needs.
- The Pump button was immediately understood.
- The bar graph was interpreted as a slide bar. Nurse2 expressed a strong desire for a slide bar. Entering figures into boxes is too slow and complicated by far. The values hardly ever need to be that exact, while setting them fast is important.

Start

- The first page gives a good overview, containing pretty much right things, except:
- TMP should be replaced with UF volume. The wish to view the TMP is a heritage from the days when the UF values were not available. The home HD patients of today do not have that heritage, and probably prefers to see the direct value of the UF.
- “Bloodtr.” (Eng: Blood Pressure) was not a suitable label for that button. Possibly something along the lines of “Art/Ven tr.” (Eng: Art/Vein pr.) would have been better. Blood Pressure is easily confused with the regular blood pressure.
- Possibly some of the bar graphs could use a slightly different interval. Apparently, it is wrong on the AK200 as well, which is where these values come from.
- The buttons were in the wrong order. Nurse2 would have wanted them to be in the order of:
    - Blood flow
    - Veinous Pressure
    - The other values in no particular order

Set Art/Vein Pressure

- No comments

Confirm

- No comments

Set Time

- No comments

Set TMP

- No comments

Set UF

- How can one tell whether the UF is on or off? I could not answer that question.

Set Blood Flow

- No comments

Alarm

- Nurse 2, just like Nurse1, expected the Confirm-button on the alarm page to work like computers usually do, and it took a few tries, before she noticed that she needed to hold the button down, just like on the AK200. Furthermore, she did not get it right, even when she had expressively been instructed, and understood, how it worked.

3. Nurse 3, implemented version

General

- Nurse3 could run the application within 10 minutes of experimentation, and thus, the goals for the GUI are met, regarding her.
- All values, functions and buttons necessary for such an application are present and easy to find, with two exceptions listed below.
- The bar graph was interpreted as a slide bar. Nurse3 expressed a strong desire for a slide bar. Entering figures into boxes is too slow and complicated by far. The values hardly ever need to be that exact, while setting them fast is important.
- A possibility to read the regular blood pressure mentioned as being a good idea.
- Some summarizing screen with all the values presented as figures might be handy.
- A button spreading the alarm limits automatically, so that they may centre automatically around the current value would be nice. Thus, the limits would not have to be manually set to reasonable values. Apparently this function is available on other machines.
- Nurse3 claims that a Mute button is absolutely necessary to avoid the stress of the alarm sounding on the AK200.

Start

- The first page gives a good overview, containing pretty much right things, except:
- TMP could probably be removed altogether, or be replaced with one of the UF values (rate or volume could not be settled upon)
- "Bloodtr." (Eng: Blood Pressure) was not a suitable label for that button. Possibly something along the lines of "Art/Ven tr." (Eng: Art/Vein pr.) would have been better. Blood Pressure is easily confused with the regular blood pressure.
- Some of the scales were not good. A marking for 0 would be much better than the one saying 100, which is in the geometrical middle of the scale.
- Digits telling the remaining time would be better than the bar graph. The argument for this is that this is how it is on the AK200.

Set Art/Vein Pressure

- No comments

Confirm

- No comments

Set Time

- No comments

Set TMP

- No comments

Set UF

- No comments

Set Blood Flow

- No comments

Alarm

- Nurse 3, just like the other two, expected the Confirm-button on the alarm page to work like computers usually do, and it took a few tries, before she noticed that she needed to hold the button down, just like on the AK200.