………………………………………………………..
Dr. Andreas Holzinger (Betreuer)

MAGISTERARBEIT

Using Work Lists on Mobile Devices as an Example of Mobile Computing in Health Care Applications

Ausgeführt am Institut für
Softwaretechnik und interaktive Systeme (IFS)
der Technischen Universität Wien
und
am Institut für
Medizinische Informatik, Statistik und Dokumentation (IMI)
der Medizinischen Universität Graz

unter der Anleitung von
Univ. Doz. Ing. Mag.rer.nat. Mag.phil. Dr.phil. Andreas Holzinger

durch
Jürgen Trauner
3362 Mauer, Amselstrasse 19

14. März 2009

………………………………………………………..
Jürgen Trauner

Using Work Lists on Mobile Devices as an Example of Mobile Computing in Health Care Applications

Diploma Thesis

Current: 2009-03-14

Jürgen TRAUNER
(0225883)

Supervisor: Prof. Dr. Andreas HOLZINGER
Institute for Software Technology and Interactive Systems (IFS) Vienna and
Institute of Medical Informatics, Statistics and Documentation (IMI) Graz

Abstract: The primary goal of this thesis is to find out how to support nurses during the transfer of the patients from the patient's room to the radiology department and back by the use of mobile devices (e.g. PDA's, mobile phones, Tablet-PCs). Currently, Siemens provides a system which is running on Personal Computers (PC). The basic element of the system is a work list. The end-users can apply various actions to the work list, including commands in order to get more detailed information and for marking an action as completed. The system can be used by nurses, who transport the patients through the hospital. Consequently, they can always see when and where the next patient is waiting. The new application will improve the hospital to get towards a modern service oriented company. On the PDA the nurses can also use filters to get the correct list for their work. An automatic update of the lists must also be possible. The reason for this is that a nurse may forget to update there list and then the patient, which he / she wanted to pick up has already been transported. Later it should be possible to integrate a Voice over IP connection to make it possible that the nurses can inform each other. This is not part of the work but the later integration of voice connection should not be blocked. The technical requirement is that the system can be used on a wide variety of available devices. Subsequently, the system has to be hardware and operation system independent; the decision for Java was therefore obvious. The graphical user interface of the system is built with the help of property files. The specifications, which are read out from files, are the colors, the texts and the messages of the user interface. This is necessary to exactly fit all the different needs and requirements of any hospital department. To satisfy all the needs it is necessary to follow a User Centered Development (UCD) process, including studies on how the nurses work in real-life settings. In arrangement with Siemens the application will be tried it out at a hospital to test which (additional) requirements are available so that we can satisfy them.

The final product is an application for mobile Clients. Siemens has already implemented an Application based on a client - server architecture for PCs. This server can also be used for the mobile application.

Keywords: Mobile User Interfaces, User Centered Development, Medical Workflow optimization, Mobile Computing in Health Care, Medical Documentation

Abstract (de): Das Ziel dieser Diplomarbeit ist es herauszufinden, wie die Krankenschwestern und die Krankenpfleger beim Transport der Patienten zur Radiologie und wieder retour durch mobile Endgeräte (PDAs, Handys, Tablet-PCs) unterstützt werden können. Derzeit hat Siemens nur eine System, dass auf PCs läuft. Das Grundelement, das in diesem System verwendet wird, ist die Worklist. Der Benutzer kann verschiedene Aktionen mit dieser Liste ausführen. Dazu gehört zum Beispiel die Möglichkeit detaillierte Informationen anzufordern oder eine Arbeit als erledigt zu markieren. Die Krankenschwestern und -pfleger können im System immer sehen, welche Patienten wann und wo darauf warten abgeholt zu werden. Mithilfe von mobilen Endgeräten können sie jederzeit genau sehen, welcher Patient als nächstes abgeholt werden soll. Durch diese Applikation kommt ein Krankenhaus wieder einen Schritt näher ein moderner service-orientierter Betrieb zu sein. Auf den PDAs ist es, genau wie auch jetzt bereits auf dem PC, möglich Filter zu definieren um die Einträge in der Liste einzuschränken. Außerdem muss es möglich sein ein automatisches Update zu aktivieren. Der Grund dafür ist ganz einfach, dass die Krankenschwester oder der -pfleger möglicherweise darauf vergisst, die Liste zu aktualisieren und so einen Patienten abholen will, der bereits von jemand anderem transportiert wurde. In einer späteren Version soll es auch möglich sein eine Sprachverbindung mit Voice over IP zwischen den einzelnen Endgeräten aufzubauen. Das ist allerdings nicht teil dieser Arbeit. Von der technischen Seite her sind die Anforderungen, dass es auf möglichst vielen Geräten eingesetzt werden kann. Daraus ergibt sich, dass es hardware- und betriebsystemunabhängig sein muss. Aufgrund dieser Anforderungen ergibt sich die Entscheidung für die Programmiersprache Java von selbst. Die graphische Benutzeroberfläche wird mithilfe von Property Dateien erstellt. Aus diesen Dateien werden die Spezifikationen für die Farben und auch die Texte gelesen, die in der Applikation verwendet werden. Das ist notwendig um die Oberfläche ohne großartige Veränderungen an die einzelnen Anforderungen der Krankenhäuser anzupassen zu können. Um wirklich allen Ansprüchen an das mobile System gerecht zu werden, ist es notwendig eine benutzerorientiere Entwicklung (User Centered Development) durchzuführen. Dies beinhaltet auch eine Studie darüber, wie die Krankenschwestern und -pfleger derzeit arbeiten. In Absprache mit Siemens wird das System auch in einem Krankenhaus getestet um weitere Anforderungen zu finden.

Das Endprodukt ist eine Applikation für mobile Endgeräte. Dazu wird der gleiche Server, der bereits existierenden PC Lösung von Siemens, weiterverwendet.

Table of contents

Glossary

This section describes some essential abbreviations and other terms which are used in this document.

Nurse in this document means the user who works with the program. The two nouns, user and nurse, are used similar and mean the same group of persons.

RMI is short for Remote Method Invocation. This is the Remote Procedure Call which is implemented in Java. Using RMI a client can call a method on a server like it was implemented local. The Java Virtual Machine handles the communication and cares for the success.

BSF is the Bright Side Framework which is used for the communication to the server to allow dynamic calls.

SOAP is the Simple Object Access Protocol which is based on XML. It is used to communicate with a web server over the default HTTP protocol.

PEAP is the Protected Extensible Authentication Protocol which is a method to transport the authentication data securely. It an open protocol that only authenticates a client into a network but it is not an encryption protocol.

GPRS is short for General Packet Radio Service which is a mobile data service for GSM cellular phone users. It is settled between the second and the third generation of mobile phones. It provides moderate speed by using unused GSM channels.

WTLS is short for Wireless Transport Layer Security which is a part of the Wireless Application Protocol (WAP). It is derived from TLS, which is the non wireless version. It is a cryptographic protocol which provides secure communication that is made for low bandwidth mobile devices because of the compression.

TTLS is the Tunneled Transport Layer Security which is a wireless security Protocol similar to PEAP.

Wi-Fi Alliance is an alliance of nearly all companies which produce hardware which is compatible to the IEEE-802.11 standard to guarantee interoperability. This is the standard of the WLAN hardware.

WPA is a class of certified protocols to secure wireless networks. It stands for Wi-Fi Protected Access. Currently the second version called WPA2 was deployed. But the older version will also stay valid for the next years because not all network interface card are able to work with WPA2.

DECT is short for Digital Enhanced Cordless Telecommunications. This is a European standard for wireless telecommunication which was first defined in the year 1992.

JVM is the Java Virtual Machine. This is a short program which is needed to run Java Applications. It is an Interpreter for the Java class files which are built by compiling the source code.

J2SE is the current default of Java on the PC. It is short for Java2 Standard Edition.

J2ME is the current default of Java on mobile devices. It is a smaller version which does only support the basic functionality which is generally needed for applications. But there are different degrees of it depending on the power of the device.

AWT is the Abstract Windowing Toolkit which provides the basic functionality to build graphical user interfaces in Java.

Swing it based on AWT and provides advanced functionality for graphical user interfaces. It is not included in the J2ME.

PDA is short for Personal Digital Assistant. This is a small computer with a fast booting operation system. In the beginning the main functionally was the administration of addresses and dates and time. Nowadays a huge supply of different programs is available.

Layout Managers are helpers to build the user interface. They take over the action to render the elements on the screen. There are many different layout managers. Every one of them has different rules how to build up the user interface. Examples are the PercentLayout or the BorderLayout. The first says that anything has to be placed with constraint using percents of the screen. The other one splits the whole space into five regions. In every region, which is called one of the following, North, South, East, West or Center, a component can be placed.

View is used for one way of combined widgets on the screen. It simply means the actually shown window with its panel which holds the widgets.

syngo® is a software system from Siemens. It is a complete solution for medical applications and image processing tasks. All computers, from thin clients to fat workstations, have the same intuitive user interface (SiemensAG Österreich, 2005b).

JBoss is an application server which is special designed for Java applications.

TKIP is short for Temporal Key Integrity Protocol. It is based on the RC4 stream cipher and uses 128 bit (Cisco Systems Inc., 2005). After transmitting an amount of data the key is changed. Because of this it is called temporal.

MAC address is the physical address of a network interface card. This is a unique address which is used for the physical transport of data. It is short for Media Access Control.

SSID is the Service Set Identifier. This is just the name of the WLAN network.

WEP was the former encryption within a wireless local area network. It stands for Wired Equivalent Privacy.

WPA stands for WiFi Protected Access. It is the encryption standard within wireless networks. The used protocol is TKIP.

RFP is short for Radio Fixed Part. This is the basis station within a DECT network. It provides the logic to handle the requests.

PP is the Portable Part which means a cordless phone which is used within a DECT network.

TDD means Time Division Duplex. This is the method which is used to share a bandwidth of a RFP to connect multiple PPs. The bandwidth is split into short timeslots and every PP gets one of them.

HF-Burst is a high frequency signal which is used to synchronize a channel within a DECT network.

SSL is short for Secure Socket Layer. It is a protocol to encrypt the transmitted data. Therefore different encryption algorithms are supported.

UTF stand for Unicode Transformation Format. It is a method to transform Unicode characters to a series of bytes.

GSM stands for Global System for Mobile Communications. It is the current communication standard of cellular phones.

SMS is short for Short Message Service. This is a service within the GMS network which can be used to send short text messages.

Tomcat always means the Apache Tomcat. It is a web server which makes it possible to use Java code on the server side to process the incoming requests.

EAP means the Extensible Authentication Protocol. It supports many different authentication protocols and is used to control the access to a network.

MSCHAPv2 is the Microsoft Challenge/Reply Handshake Protocol in version two. It is used within EAP to handle the access control.

GUI means the Graphical User Interface. It consists of multiple textboxes, buttons and lists and allows the user to interact with the system.

Siemens IWM is the Siemens Imaging Workflow Management (SiemensAG Österreich, 2004). It is the system which is developed as a solution for the radiology department in hospitals.

SHA1 is the Secure Hash Algorithm. It is used to sign a message with an identical hash code. The code is used as an inspection value to check the message.

serialVersionUID is the version number which is used to identify serialized objects in Java.

1. Introduction

This paper is about the transportation of patients in hospitals, about how this is done now and how PDA's can assist the nurses. The general goal of this paper is a system which helps the nurses to do there job more efficient, what this means that the transportation is done faster. Siemens already has a system for the PC which is used to assist the nurses. When the nurse starts her working day, he / she logs on to the system and enters the criterions of the patients he / she has to transport today. The list, which is fetched from the system, is shown on the screen. If it looks correct to the nurse, the list is printed out on a sheet of paper. This sheet is the procedure documentation of the nurse. Changes to this must be told him / her explicitly by another nurse or by telephone. This was the intention to build a new system. This paper describes the build process of the new system. It documents the problems which occurred during the engineering process and how they were solved. As already told, the basic concept was the use of PDA's. They are small enough so that the nurses can take one with them. Of course there must be a benefit because otherwise the nurses may not want to use them. The new solution contains an automatic notification of transport requests which were added during the working day. Nothing must be done to inform the nurse. Another benefit is that the transported patients can be marked as transported directly on the PDA. This has two side effects. The first is that the nurse must not mark them later as transported when he / she has time to do that. And the other one is that nurses can help each other easily because the get informed of the transports of the others. Therefore it must be able and it is easily possible for the nurse to change the criterions of the patients he / she has to transport without going back to the office of the nurses.

1.1. The technical specification

Now the focus is on how the system works currently. Siemens already developed a system for the personal computer which is called "Web Order Entry". Therefore a server was set up with the application server JBoss and the web server Tomcat. The data is stored into a Sybase database. There are three different methods of communication. These are RMI, SOAP and BSF. The next figure views this architecture.

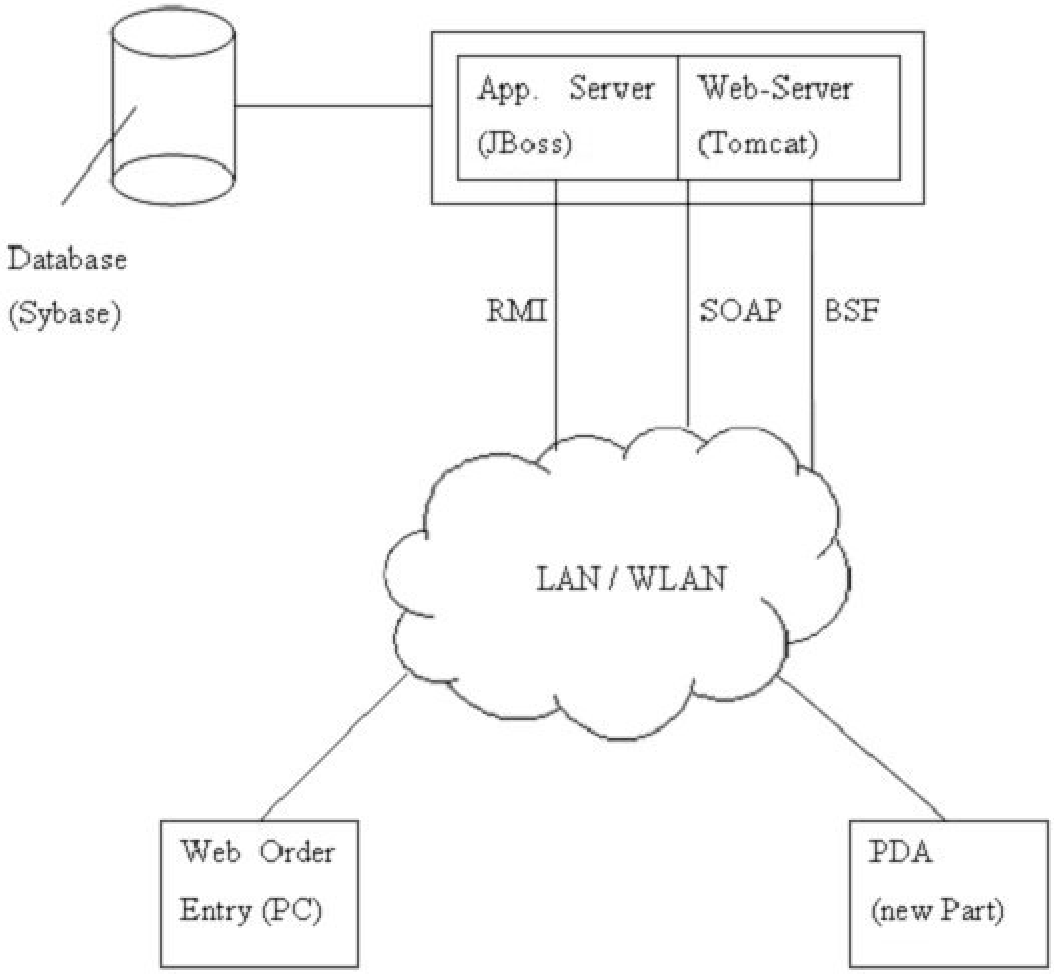
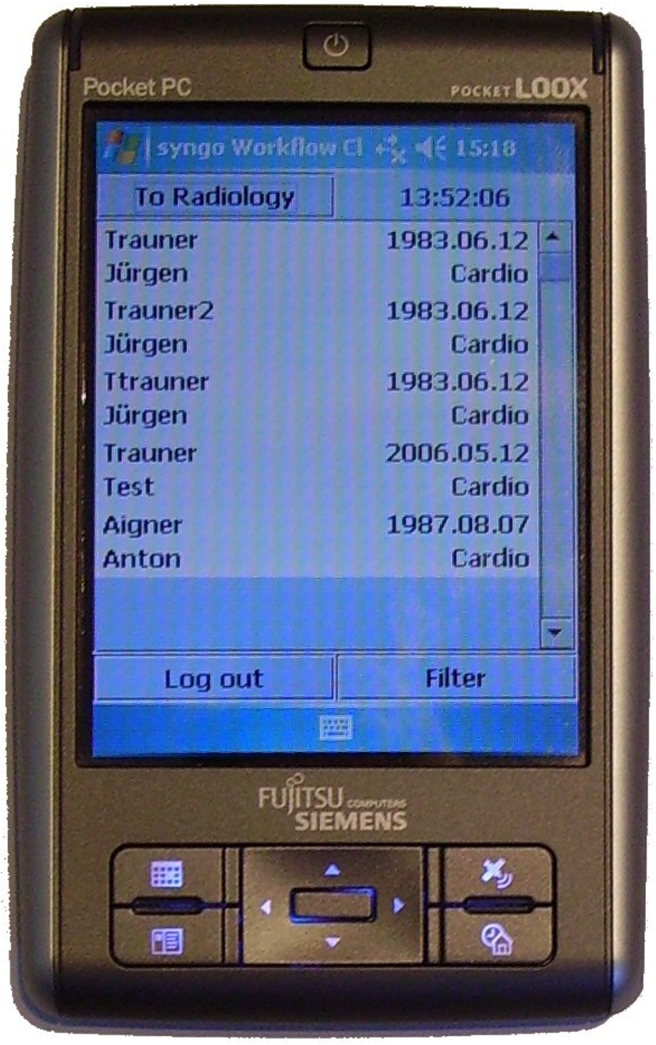


Figure 1: Architecture

The new Part will be built PDA independent. A Fujitsu Siemens Pocket LOOX N520, which can be seen on the next figure, is used for testing. Because Java is used for the Application it is necessary to install the Java Runtime Environment on the PDA. The technical data which reads as follows, show, that this should not be a big problem.

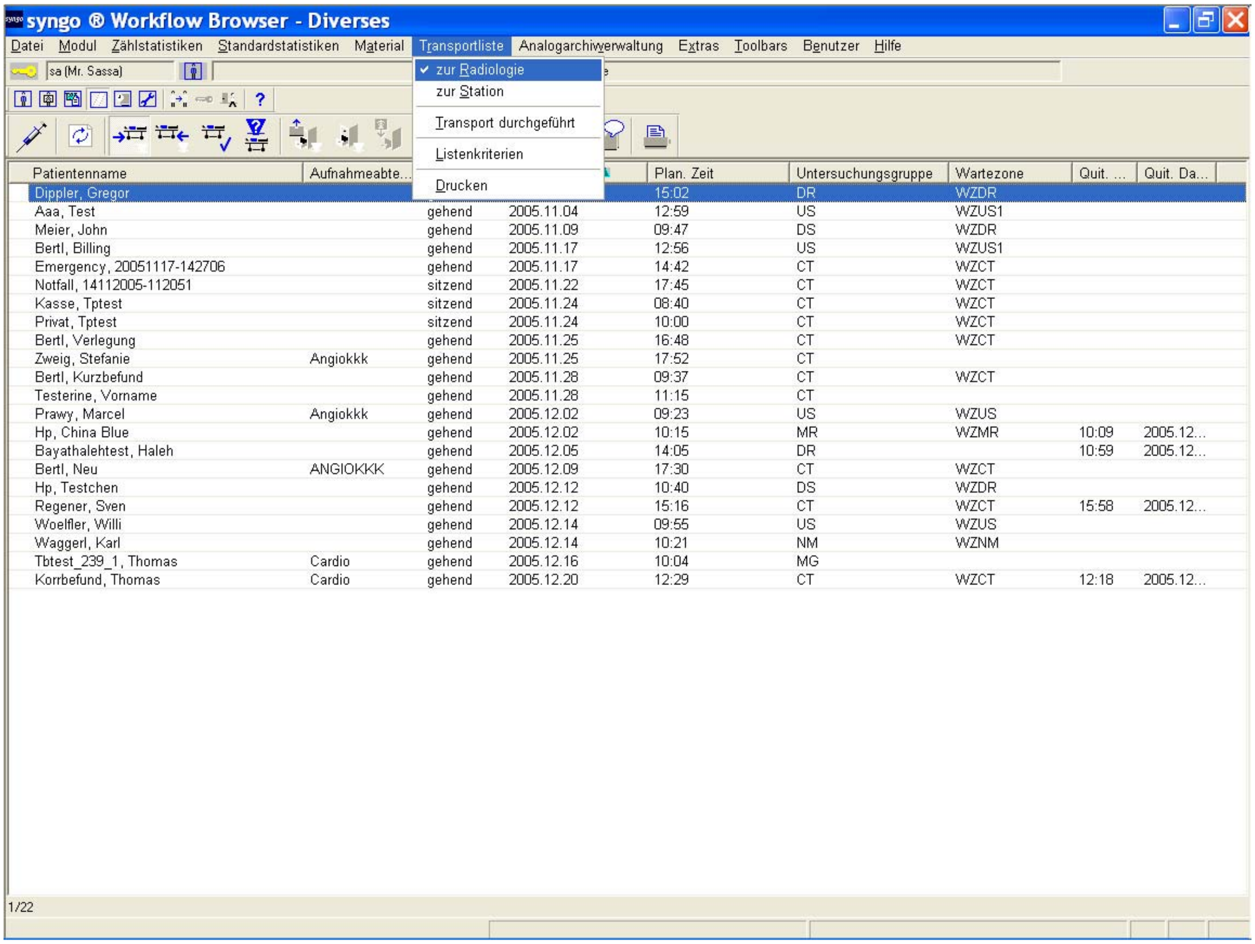
- Intel Pentium Processor with 312 MHz
- 128 MB RAM
- 64 MB ROM
- WLAN 802.11g
- 3.5” TFT, 65536 colors, 240 x 320 pixel
- Microsoft Windows Mobile 5.0



Trauner	1983.06.12
Jürgen	Cardio
Trauner2	1983.06.12
Jürgen	Cardio
Ttrauner	1983.06.12
Jürgen	Cardio
Trauner	2006.05.12
Test	Cardio
Aigner	1987.08.07
Anton	Cardio

Figure 2: Fujitsu-Siemens Pocket LOOX N520 (Source: Photo taken 1.9.2006)

Beside the Web Order Entry system, there is an older application called “syngo® Workflow Browser”. These two applications exist together because the new system does not jet cover the whole range of needed tasks. The transport lists, which are treated in this paper, are part of the old system. These lists show which patients have to be transported from where to where. As shown in the next figure, a transport list contains the name of the patient, if he can walk and when he / she has to be there and some other additional information.



syngo ® Workflow Browser - Diverses
Datei Modul Zählstatistiken Standardstatistiken Material Transportliste Analogarchivverwaltung Extras Toolbars Benutzer Hilfe
sa (Mr. Sassa)
✓ zur Radiologie
zur Station
Transport durchgeführt
Listenkriterien
Drucken
1/22

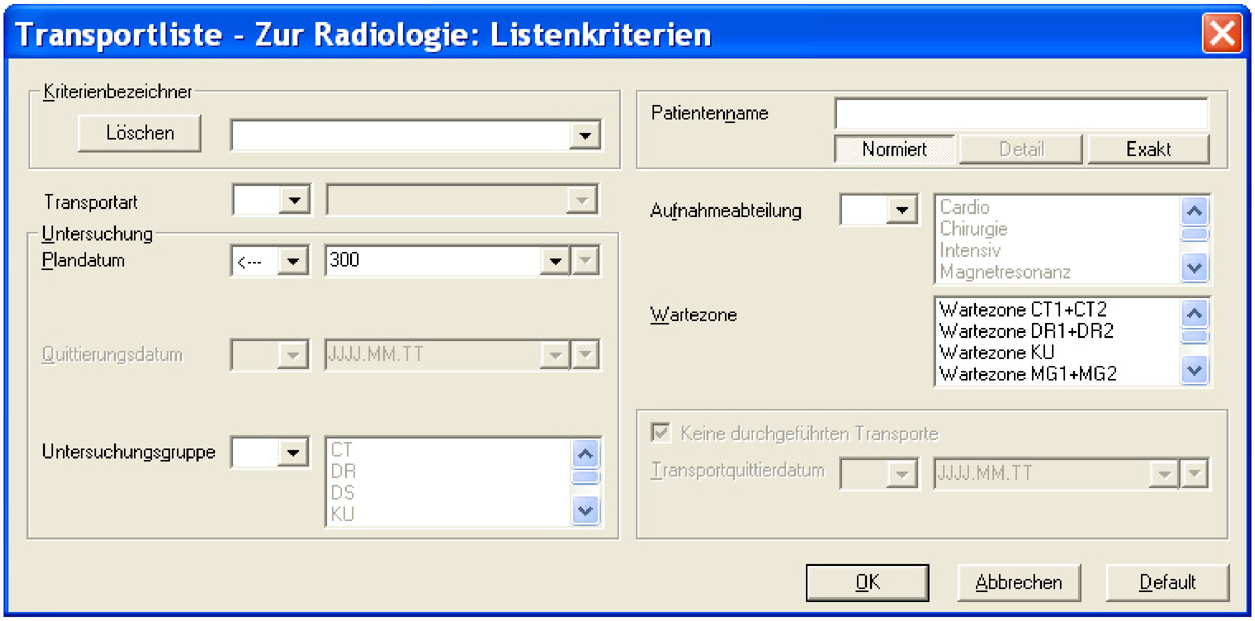
Patientenname	Aufnahmeabte...			Plan. Zeit	Untersuchungsgruppe	Wartezone	Quit. ...	Quit. Da...
Dippler, Gregor				15:02	DR	WZDR		
Aaa, Test		gehend	2005.11.04	12:59	US	WZUS1		
Meier, John		gehend	2005.11.09	09:47	DS	WZDR		
Bertl, Billing		gehend	2005.11.17	12:56	US	WZUS1		
Emergency, 20051117-142706		gehend	2005.11.17	14:42	CT	WZCT		
Notfall, 14112005-112051		sitzend	2005.11.22	17:45	CT	WZCT		
Kasse, Tptest		sitzend	2005.11.24	08:40	CT	WZCT		
Privat, Tptest		sitzend	2005.11.24	10:00	CT	WZCT		
Bertl, Verlegung		gehend	2005.11.25	16:48	CT	WZCT		
Zweig, Stefanie	Angiokkk	gehend	2005.11.25	17:52	CT			
Bertl, Kurzbefund		gehend	2005.11.28	09:37	CT	WZCT		
Testerine, Vorname		gehend	2005.11.28	11:15	CT			
Prawy, Marcel	Angiokkk	gehend	2005.12.02	09:23	US	WZUS		
Hp, China Blue		gehend	2005.12.02	10:15	MR	WZMR	10:09	2005.12...
Bayathalehtest, Haleh		gehend	2005.12.05	14:05	DR		10:59	2005.12...
Bertl, Neu	ANGIOKKK	gehend	2005.12.09	17:30	CT	WZCT		
Hp, Testchen		gehend	2005.12.12	10:40	DS	WZDR		
Regener, Sven		gehend	2005.12.12	15:16	CT	WZCT	15:58	2005.12...
Woelfler, Willi		gehend	2005.12.14	09:55	US	WZUS		
Waggerl, Karl		gehend	2005.12.14	10:21	NM	WZNM		
Tbtest_239_1, Thomas	Cardio	gehend	2005.12.16	10:04	MG			
Korrbefund, Thomas	Cardio	gehend	2005.12.20	12:29	CT	WZCT	12:18	2005.12...

Figure 3: Transport list (Source: Personal communication from 22.12.2005)

The goal of the work on this thesis is to create an application for mobile devices which shows transport lists. Of course the lists can not look alike the lists on personal computers. The reason therefore is simply the small screen size. A main task is to find out which of the information are the most important because these parts have to be shown in the list. The other information can only be viewed on another detailed page. The nurses, for whom the system on the mobile devices is designed, can get the detailed information by selecting the patient record in the list. There has to be found out which is the best way to do the selection. One possibility is a simple click, another one is double click. This has to be found out in the real life because this will show whether a simple click is a problem or not. If the nurses sometimes touch the PDA unintentional then a simple click cannot be used. Another decision which has to be made is, how to mark a patient as transported. If a single click can be used for viewing the details a double click can be used for this. A different way is to place the button, which marks a patient as transported, on the page with the details. But then a nurse always has to open the details page before she can mark a patient as transported. So there are many things which have to be cleared up.

The transport list is not a fixed list. The list has to change automatically when a patient has been transported. Therefore updates have to be made in short intervals.

But the list can also be parameterized. The next figure shows the available options.



Transportliste - Zur Radiologie: Listenkriterien

Plandatum: <--- 300

Quittierungsdatum: JJJJ.MM.TT

☑ Keine durchgeführten Transporte

Transportquittierdatum: JJJJ.MM.TT

Figure 4: Criterions of transport lists (Source: Personal communication from 22.12.2005)

An example for the options is that the nurse wants to view the patients who have to be transported to or from a certain waiting area. Another one is to show all the patients who need a computer tomography or who have to be transported tomorrow. In the current application there is a special dialog therefore. The problem is that the current dialog is very large. On handhelds the screen size is too small to show the whole dialog at once. In principle there are two possibilities. One is to make a high dialog with a scrollbar on the right to scroll down. The other possibility is to make a selection where the nurse can select which criterions she needs now. This can be made with a dropdown list at the top of the dialog to permit a fast switch between the criterions without reopening the dialog. Another fast switching possibility is to use a button and a dialog. After a click on the button he / she gets a list of the possible criterions where he / she can select one by clicking on it. But only a test in real life can show what a nurse really wants to use.

2. Mobile Computing in Health Care: Existing Systems

There are some software systems witch makes use of PDA's in hospitals. They are made for different applications. Some of them help to prepare diagnosis other can be used as a drug dictionary. Architectural questions are also discussed in the papers of the existing systems. A description about existing solutions follows.

2.1. The Ward-in-Hand Project

This is a project which makes use of PDA's (Ancona et al., 2001). The system consists of four main components listed below.

- Patient Record Manager
- Workflow Manager and Personal Organizer
- Legacy System Interface
- Security Manager

The system is build to be used with an existing legacy system. The main functions to operate with patients are supplied by the Patient Record Manager. Examples for such functions are the vital signs or the view of the results of clinical tests.

The Workflow Manager is to define roles and activities. Afterwards they can be combined to a workflow, which can be monitored. The Personal Organizer is designed for remote access. The users can take a look at their work list items when they are at home. But this part of the system can also be used to start new predefined workflows. Important is that the user only has to authenticate once to use the whole system.

The Legacy System Interface is necessary, because the wireless PDA system works in common with an existing system. This part provides the connectivity and the conversion of the data of the legacy system.

And of course security is a big reason in a wireless system. The last module called Security Manager provides this by making use of the Secure Socket Layer (SSL) which is the standard for secure connections.

The system has already been tested within a pilot trial at a hospital in San Martino, Barcelona and Offenbach. Only two departments where involved in the test. The first thing that had to be discovered was, if there is any interference by the access points. After the secure, that there will be no trouble about the access points, 15 medical doctors and 20 nurses started to test the system. *"Because doctors were inexperienced in working with the Ipaq handheld devices, they found it uncomfortable to work with them at the very beginning. But this changed very fast as they got familiar to them and as they got the feeling of the flexibility they can have using them"* (Ancona et al., 2001). In the beginning the users needed much more time than before but they got faster and faster. The final result was that it is possible to save up to 40% of the time. Maybe in real more will be possible because they get familiar with the PDA and the system (Ancona et al., 2001).

2.2. Desktop and Mobile Software Development for Surgical Practice

The goal of this project, called iIncise, was to create a system for Desktops and PDA's (Oyama et al., 2002). The main attention goes to internet able technologies. Therefore an internet server in cooperation with a database was set upped. The client software can be run in two modes. One is online. Then every change can be made immediately. The second mode uses replication. The client requests some data from the server and makes a copy to the local disk. Then the user can operate with this data. The next time when there is a connection to the server, the data is update. This offline mode of the system makes it possible to work everywhere. This is necessary because pediatric surgeons often operate at several different hospitals. And also, if doctors are not within the clinic it is possible to get the data because default internet techniques were used. But this case is also very risky because data is transferred over the internet and / or saved on the local disk of the computers. Maybe the transfer can be secured by using SSL. But it is very difficult to control the usage of the replicated data because of the amount of devices. Since the use of standard software the costs for the system are very low. But the use of the system is also not that high as it could be by using special software. The savings are only about $30.000 a year.

2.3. Ubiquitous Mobility in Clinical Healthcare

The main attention within this project was to provide a reference architecture for a mobile solutions within hospitals based on the eHospital system (Pavlovski et al., 2004). For this system an authentication server for the wireless devices was used. To guarantee the security mutual authentication is used. Therefore Tunneled Transport Layer Security (TTLS) or Protected Extensible Authentication Protocol (PEAP) is used. A radius server handles the connection requests form the wireless devices.

Security is a basic requirement in all mobile applications, especially within applications for Hospitals. Ubiquitous devices lack many of all the familiar security features known in desktop computing. However, as ubiquitous devices are increasingly applied in health care industry, security aspects need to receive more and more attention (Weippl et al., 2006).

The base of the above mentioned security protocols are the Extensible Authentication Protocol (EAP) or the EAP-TLS protocol, which stands for Transport Layer Security. EAP is not a real protocol, it more a framework which provides some basically functions for the authentication. EAP-TLS is a protocol which can be used within the EAP framework. Since TLS is the successor of SSL it is really good. Another advantage of TLS is that it is an open standard, which means that everybody can use it for free and that the devices can be provided form different vendors. It uses the private / public keying system to secure the connection. The authentication is done through a radius server. But for this protocol a client certificate is needed. This is a big disadvantage. The Tunneled Transport Layer Security (TTLS) has been developed on the base of TLS. It also uses a private / public key system to build up a secure connection between the device and the network with the radius server to which the client has to authenticate. The big difference is that a certificate is only used on the server side. The client does not need to have one. It is also an open standard and it is easier to be used in real life. The other protocol, short called PEAP, is in principle similar to EAP-TTLS. But there are different versions of PEAP.

Because most people do not know about this, PEAPv0/EAP-MSCHAPv2 is meant when somebody speaks about PEAP. The second subtype is PEAPv1/EAP-GTC. With this protocol it is possible to use an inner authentication protocol except Microsoft’s CHAP protocol. But the difference between these two versions is rare. All three protocols described above are certified with Wi-Fi Protected Access (WPA) or WPA version 2 (Koren, 2003, SiemensAG Österreich, 2005a, Wikipedia, 2004-2006).

The reason why this system has been developed was the improvement of the patient satisfaction by faster diagnosis. One method to do this is to reduce the quantity of the paperwork. Instead of many different forms on paper, the data is directly inserted into an electronic form. That is why redundant data must not be inserted anymore. This reduces the time which is needed to fill the forms. This time can be used to examine patients.

Another Requirement was the possibility to have access to the medical data within the whole hospital. When realizing the WLAN a problem occurred. It was very difficult to realize this within the whole hospital. It is uncomplicated to do this for a room, but it is not simple to make it possible to roam between many access points without any gap.

Not only the functionally and the usability are criterions for the acceptance by doctors and nurses. Ergonomic factor are also very important. This means, that the size and the weight of the PDA may cause problems in acceptance. It is very difficult that the personal accepts the new mobile devices, if they do not fit their expectations.

One thing, which surely is not standard, is a GPRS gateway. The architecture which is presented here has such a gateway. The incoming calls get directed to the wireless gateway to authenticate through the radius server. After a trusted connection to the wireless Device is established, the user can logon through the Authentication Server. The next figure shows the architecture of the eHospital system in detail.

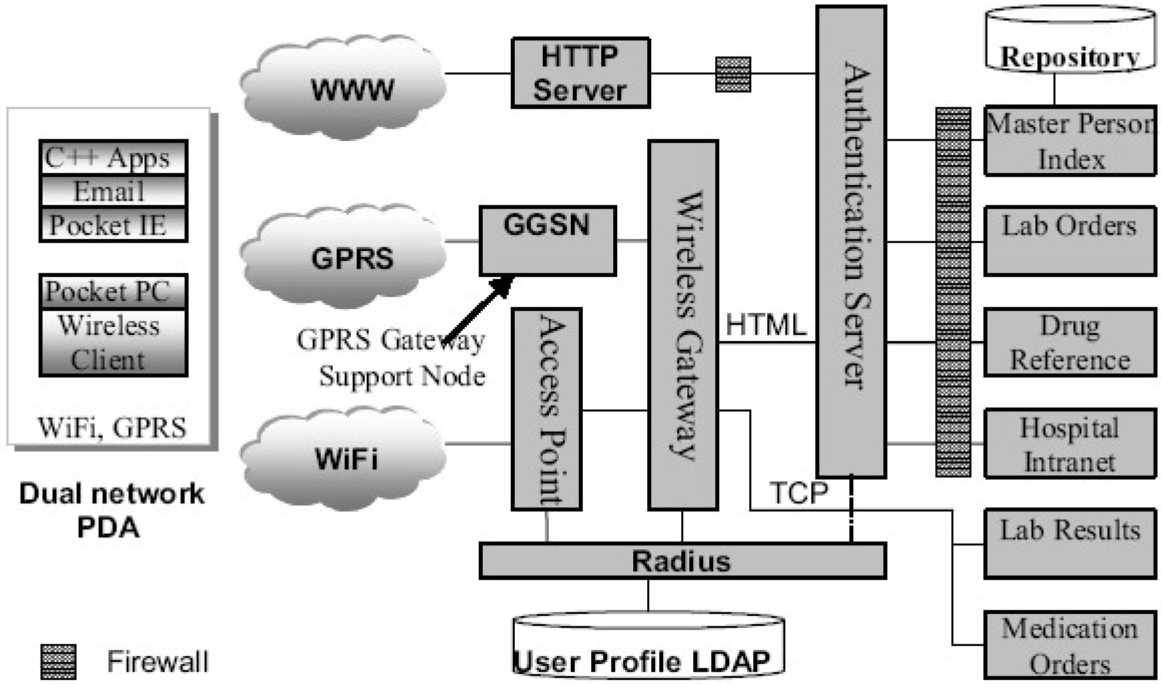


Figure 5: Architectural overview (Pavlovski et al., 2004)

The GPRS calls and the connections through WLAN and Wi-Fi are roamed together. So if the device supports a change between these networks it is possible to switch to the other network. This is called seamless roaming. To have the same network addresses the GPRS Gateway Support Node assigns an IP to the cellular mobile device. To roam between different networks it is necessary, that the client has a consistent IP address. Therefore a special client has been deployed to the PDA which provides the needed service. The second service the client provides is the needed security. Because the system allows connections through public networks the security is a big question. The client uses SSL or WTLS to encrypt the connection (Pavlovski et al., 2004).

2.4. Experienced clinicians in the palm of your hand

The following lines will give an overview of PDA technologies and relevant medical applications. Taking a look to the market shows, that there are only two major companies producing an operation system for PDA’s. Palm OS is made by PalmSource Inc. The second is

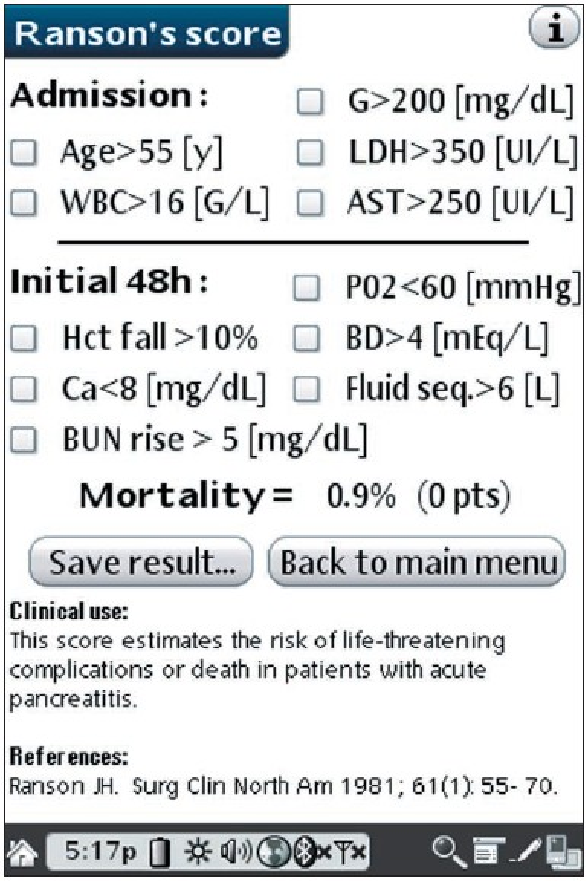
Windows Mobile which is made by Microsoft. Any other system for mobile devices can be neglect. Since 1999 the wireless connection capabilities became more and more recently and so the interest to also use them in clinical healthcare has grown up.

Because of an ever-increasing amount of constantly changing information about patients, diagnostics and therapy the decision finding of the medical staff gets constantly harder. PDA's can help to increase the availability of information everywhere. This means that the physicians always get the latest details of the patient findings. By using this information the PDA can also help to consider new therapeutic recommendations.

That PDA's always get more common in clinical health care can be seen in the results of a study on how many physicians are using a PDA. About 40% are using such a device. The most common applications are for drug reference, scheduling and medical calculations. But one thing that also can be seen in the study of the American Medical Association that mostly the younger stuff uses mobile devices because they thing that this reduces medical errors (Baumgart, 2005). Otherwise the older stuff does not use PDA's. Maybe they are not familiar with them. So for mobile solutions it is very important that they are very easy to use because otherwise they will not be accepted widely.

Just take a look at the first field of usage which is the drug reference. The PDA's can replace the drug reference books. The comparison of drugs can be done very fast. It is also possible to make drug interaction checks very fast. The efficiency can be enlarged by involving the nursing staff. The US Food and Drug Administration (FDA) has published a final role which helps to avoid treatment errors. Therefore every patient and every drug gets a bar code. The system starts when a patient is first admitted to the hospital. Then a bar coded is generated to identify the patient. Also every drug has a bar code on its label. Before a drug is given to the patient the bar code of the patient is scanned with a PDA - based scanner. Afterwards the bar code of the drug is scanned. The computer checks if the combination of patient and drug matches with the patient's medical record. As a result of this the wrong drug or wrong patient problem can be solved.

The next paragraph deals with the patient scheduling, tracking and charting. Daily progress notes are the crucial clinicians work. But the quality is often inadequate, what can be improved by using PDA's. At a study within a pediatric critical care unit with a before and after comparison significant fewer documentation discrepancies could be found after the use of PDA's. Another study was made in an orthopedic surgery. At this study two groups were built. The test group got PDA's with a special software for bedside use. The control group worked on as before which means that they used paper notes which were entered into the hospital computer subsequently. The results were examined by an expert team which found out the number of diagnoses per patient increased significant. Also the quality of the documentation could be improved by using handheld devices. An example of possible information which could be provided through such a device can be seen on the next figure.



Ranson's score

Admission:

□ Age>55 [y]
□ WBC>16 [G/L]
□ G>200 [mg/dL]
□ LDH>350 [UI/L]
□ AST>250 [UI/L]

Initial 48h:

□ Hct fall >10%
□ Ca<8 [mg/dL]
□ BUN rise > 5 [mg/dL]
□ P02<60 [mmHg]
□ BD>4 [mEq/L]
□ Fluid seq.>6 [L]

Mortality = 0.9% (0 pts)

Clinical use:
This score estimates the risk of life-threatening complications or death in patients with acute pancreatitis.

References:
Ranson JH. Surg Clin North Am 1981; 61(1): 55- 70.

Figure 6: Example of a clinical decision support system (Baumgart, 2005)

But it also produced some redundant items. In general all these studies say that the use of PDA's improves the quality. Because all the data which is already within the computer system can also be used for decision support. Such clinical decision support systems can help the doctors making their diagnoses. Pilot studies in which the doctors used such systems on a PDA were perceived as helpful.

But the recording and the interchange of data always contains a security risk. There are special viruses for PDA's which run on Windows Mobile and also for the Palm OS. This can be solved by using a PDA virus scanner. The Center of Medicare and Medicaid Services (CMS) has published security requirements of patient data. With the use of wireless networks new risks come along. The data has to be encrypted and an authentication is necessary to prevent unauthorized use or to weaken the effect. The best would be if the hardware and the software take care of this.

In the future the PDA will get familiar to the doctors. As more and more of them use such devices the industry will produce some which exactly fit the needs of the physicians. In the further future the devices will be able to recognize the speech to simplify the input. Maybe the PDA is also able to combine many different sources like textbooks and public health information with the patient's record to give the needed information to the doctors. But no computer will ever be able to replace a dedicated, experienced clinician (Baumgart, 2005).

2.5. Hospitals in Austria

Many Hospitals use the Digital Enhanced Cordless Telecommunications (DECT) system to communicate wireless between the nurses and doctors (Holzinger, 2002). The standard was defined from the European Telecommunications Standards Institute in 1992. Three years later it has been widely improved. Meanwhile DECT is not only used in Europe but also in many other countries in the world. When talking about DECT two terms come up. They are called Radio Fixed Part (RFP) and Portable Part (PP). The RFP is simple the basis station, which provide the logic to handle the internal requests and external requests to the telecommunication network, and the PP is the hand part. Because it is digital it has some essential advantages in comparison to analog communication techniques which are listed in parts below.

- Audibly tone quality improves
- High hearing security
- Simultaneous use of many mobile parts
- Internally calls are free of charge
- Mobile parts can be used with many basis station
- Saving of frequencies
- Handover

2.5.1. Technical Description of DECT

Most of these points above are clearly. But the last, the handover, has to be explained. The meaning of it is that the channel can be switched. The switching can be done within the same basis station or between two stations. The reason for a switch is that the signal gets worse. The sense is that the quality of the connection can be hold, when it is getting worse. The handover is always stated by the PP. But the RFP can indirectly start the handover. From time to time the RFP transmits the quality data to the PP. So if it sends very bad data then the PP notices that and initiates a handover. The first step is that a second connection is established. Then the data is sent over both connections redundantly for some time. After a short period of time the quality is compared and the worse connection is terminated by the PP (ELektronik-KOmpendium, 1997-2006).

2.5.1.1 The Transmission

In most of the European countries the frequency range form 1880 MHz to 1900 MHz is reserved for DECT. This frequency range is divided into ten channels with a distance of 1728 kHz between the channels. To provide the service to many PP's every channel uses the Time Division Duplex (TDD) method (Wölfle, 2006). This method has a big advantage in systems with an asymmetry of uplink and downlink because as the amount of the uplink increases more bandwidth is used for the uplink and lesser bandwidth is spent to the downlink. This means if less data has to be transmitted then the bandwidth of the up- and downlink is equal. But if one side needs more bandwidth than the asymmetry is exploit (Kowalk, 2002, Wikipedia, 2005-2006a, Deciveforce.com, 2005).

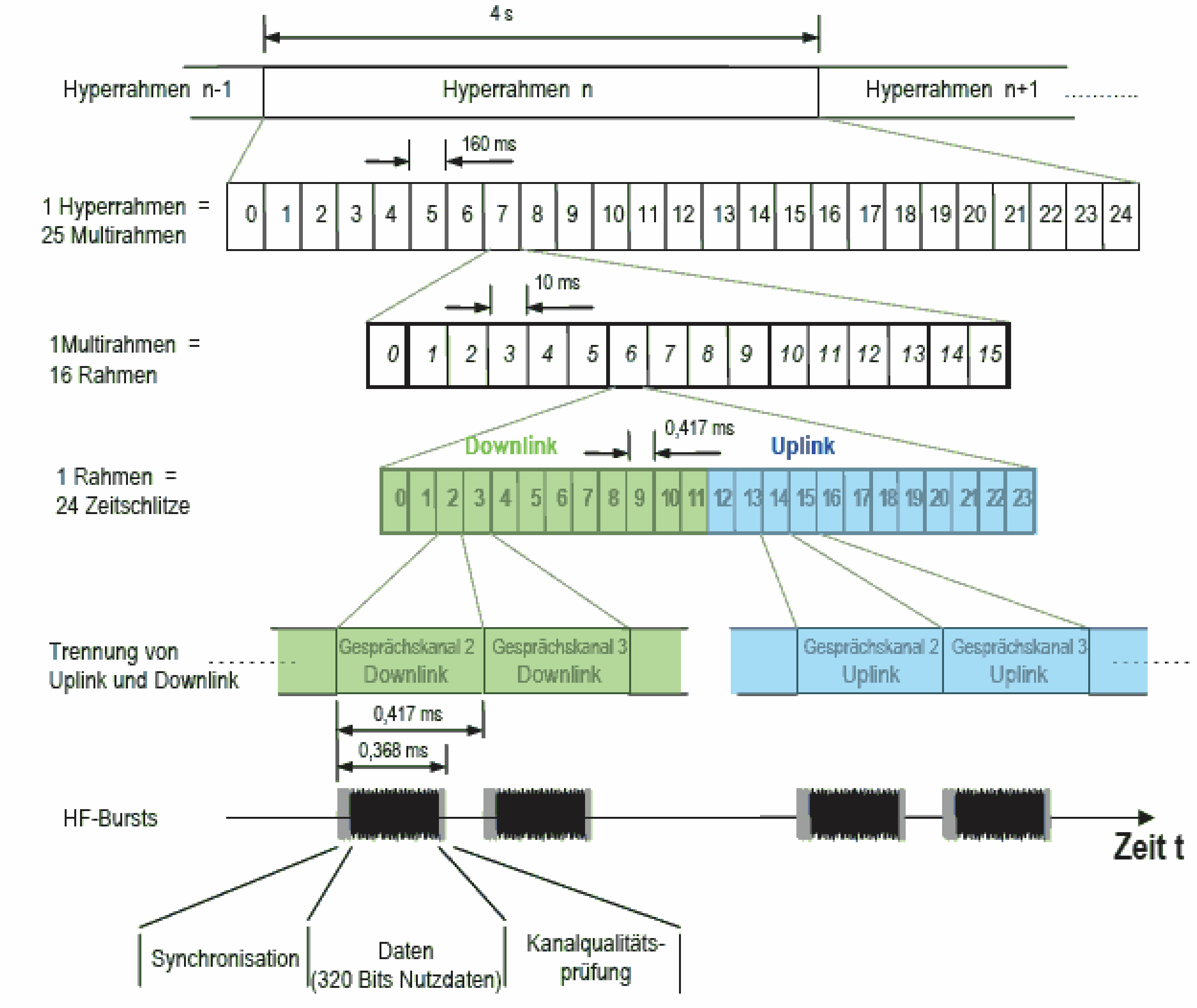


Figure 7: Scheme of the standard frames and the timing structur using DECT (Wölfle, 2006)

The main size in timing is the frame which is 10 ms long. 16 of these frames can be combined to a multi-framework. And 25 of these 160 ms long multi-frameworks themselves can be combined to a hyper-framework which is 4 seconds long. But go back to the main size in timing, the frame.

A frame can be further divided into 24 time slices which are about 417 µs long. By default 12 slices are used for the uplink and 12 are used for the downlink. The transmission itself within one of the 24 slices is done with a short, about 368 µs long, HF-Burst. Beside synchronization and system information 320 Bits are available to transmit user data. The difference between the length of the time slice and the burst is needed to prevent overlapping. The data rate can be calculated easily by multiplying which results in 32 kbps. But for some application this can be different because it is possible to change the timing. It is feasibility that two calls share a time slice. Then the data rate is lower but more PP's can be serviced. In the other way time slices can be combined. So the data rate can increase because of the two time slices and the omission of the gaps. And as mentioned before it is possible to allocate the time slices asymmetric. For example this means 23 time slices are used for uplink and only 1 for downlink. With this modification a data rate of 552 kbps is reachable.

The PP's are organized in cells. Each cell belongs to a RFP. As long as the PP's do not want to do a call they stay passive. The mobile parts do not send any signals until they send a call or response to a call request. The PP's only observe the relevant channels to watch if there is a waiting call for them. The RFP on the other hand sends permanent signals with synchronization and identification data. These signals are called Beacon signals. If a call is waiting for a PP then the RFP sends a Paging signal to initiate the connection. The steps which are necessary to establish a connection between the PP and the RFP are always the same. They are described in the following points.

1. The PP discovers which channel is best for the communication.
2. The PP sends a call to the RFP which channel or channels should be used. This is called Outgoing Call Request.
3. The RFP is always waiting for call request on empty channels.
4. After a half frame the RFP sends the confirmation. This is called Outgoing Call Confirmation. This message contains a list of best channels for the RFP. With this the connection is established.
5. If the PP needs more channels it sends a Physical Channel Request.
6. This is confirmed by the RFP with a Physical Channel Confirmation.

DECT does not have a central system administration. Because of this the channel allocation is always done dynamically. The PP is responsible for the channel selection. The dynamical channel selection is also used for intra cell handovers. Because of the decentralized structure it is relatively cheap to build up a DECT network. If the PP comes to another cell an inter cell handover is started by the PP. The steps for this are very similar to the steps to establish a connection. But DECT is not very tolerant if the user moves fast. The tolerance is by about 20 km/h which is a big difference to modern GSM networks (Wölfle, 2006).

2.5.1.2 Transmitting power

The maximum transmitting power of DECT devices is 250 mW. An active PP sends in standard DECT a 368 µs long HF-Burst with maximum power. This is repeated for each frame which means every 10 ms. The middle achievement can simple be calculated using the next formula.

```
Transmitting power * HF-Burst length / Rate of repetition
```

By inserting the values form above the middle achievement is 9,65 mW. But when the PP does not have an active connection it does not send anything.

The transmitting power of the RFP's can not be calculated as simply because it depends on the count of the active connections. The extreme would be if 23 time slices and double slot burst are used. In this case approximately 9 ms of the 10 ms frame would be used by the RFP to send data. Inserting these values in the formula above by using the maximum transmitting power the result is 225 mW. The minimum would be a single HF-Burst with a length of 83 µs which results in around 2,1 mW. Because a regulation of the transmitting power is not planed the scenarios with 250 mW are realistic. For RFP's this is definitely, for PP a regulation can be added optional. But nearly all devices do not have a regulation for the transmitting power (Wölfle, 2006).

2.5.2. DECT in operation

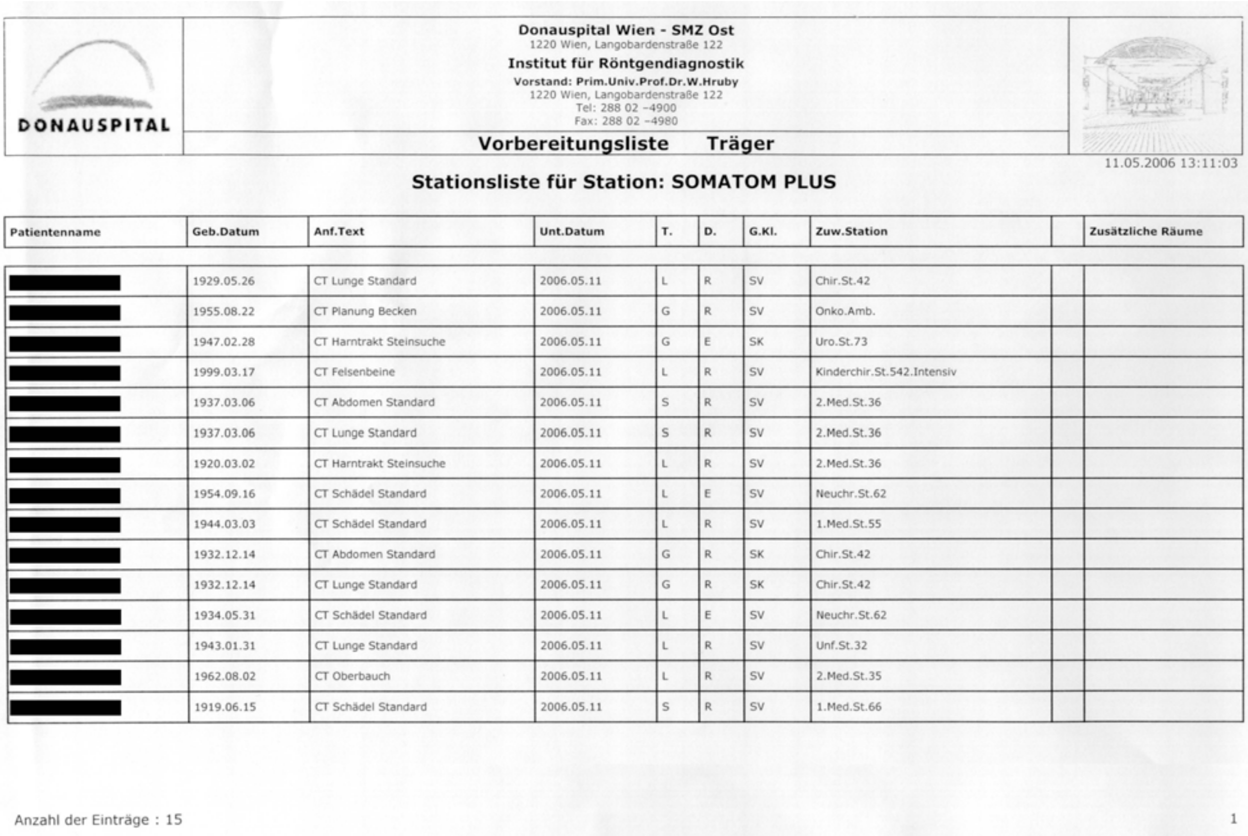
The University Hospital in Graz uses DECT to organize there transportation of the patients. Because it is a very large hospital they have an own transportation service. The physician enters the needing of transportations into the computer and the rest is done by the transportation service. An employee sends the requests of transportation via DECT to a free colleague who carries out the transport. After finishing a notification is sent back to the transportation service station by using the DECT system. They are content with there system and do not think over to change this system.

The Danube Hospital in Vienna is a smaller one. Every employee of transport service has its own ward. First they print a list on paper which patients have to be transported at which time from where to where. Then they use this sheet of paper to do there work. But they know that it is not state of the art and they want to change it. The have planed to make a write out for a new system which has to use the available infrastructure, what means that the new system to support the transportation has to use the DECT system.

3. User Centered Development in Real-Life Setting

In this chapter the workflow of the nurses who transport the patients will be discussed. This is done on the example of two hospitals. The first is the Danube Hospital in Vienna and the second is the University Hospital in Graz. It was essential to concentrate on the workflow of the end-users in order to adapt the user interface exactly to their needs (Holzinger, 2003, Holzinger, 2004, Holzinger, 2005).

At the Danube Hospital there are some nurses who do only transport the patient from and to the X-ray unit and the other medical units like the computerized axial tomography. At the hospital there are 14 rooms with different medical instruments. For each room there is a list of appointments. An anonymous version of this can be seen on the figure below.



DONAUSPITAL

Donauspital Wien - SMZ Ost
1220 Wien, Langobardenstraße 122
Institut für Röntgendiagnostik
Vorstand: Prim.Univ.Prof.Dr.W.Hruby
1220 Wien, Langobardenstraße 122
Tel: 288 02 -4900
Fax: 288 02 -4980

Vorbereitungsliste Träger

11.05.2006 13:11:03

Stationsliste für Station: SOMATOM PLUS

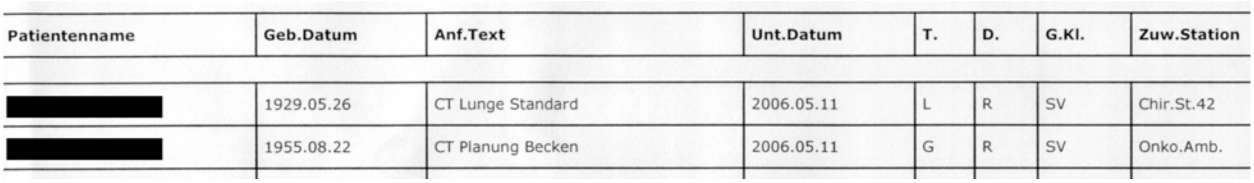
Patientenname	Geb.Datum	Anf.Text	Unt.Datum	T.	D.	G.Kl.	Zuw.Station		Zusätzliche Räume
[redacted]	1929.05.26	CT Lunge Standard	2006.05.11	L	R	SV	Chir.St.42		
[redacted]	1955.08.22	CT Planung Becken	2006.05.11	G	R	SV	Onko.Amb.		
[redacted]	1947.02.28	CT Harntrakt Steinsuche	2006.05.11	G	E	SK	Uro.St.73		
[redacted]	1999.03.17	CT Felsenbeine	2006.05.11	L	R	SV	Kinderchir.St.542.Intensiv		
[redacted]	1937.03.06	CT Abdomen Standard	2006.05.11	S	R	SV	2.Med.St.36		
[redacted]	1937.03.06	CT Lunge Standard	2006.05.11	S	R	SV	2.Med.St.36		
[redacted]	1920.03.02	CT Harntrakt Steinsuche	2006.05.11	L	R	SV	2.Med.St.36		
[redacted]	1954.09.16	CT Schädel Standard	2006.05.11	L	E	SV	Neuchr.St.62		
[redacted]	1944.03.03	CT Schädel Standard	2006.05.11	L	R	SV	1.Med.St.55		
[redacted]	1932.12.14	CT Abdomen Standard	2006.05.11	G	R	SK	Chir.St.42		
[redacted]	1932.12.14	CT Lunge Standard	2006.05.11	G	R	SK	Chir.St.42		
[redacted]	1934.05.31	CT Schädel Standard	2006.05.11	L	E	SV	Neuchr.St.62		
[redacted]	1943.01.31	CT Lunge Standard	2006.05.11	L	R	SV	Unf.St.32		
[redacted]	1962.08.02	CT Oberbauch	2006.05.11	L	R	SV	2.Med.St.35		
[redacted]	1919.06.15	CT Schädel Standard	2006.05.11	S	R	SV	1.Med.St.66		

Anzahl der Einträge : 15

1

Figure 8: Transport list as used at the Danube Hospital (Source: Personal communication from 11.05.2006)

At the begin of his / her shift the nurse prints out the list of the patients on a sheet of paper to get to know which patients he / she has to transport today. The total sum of the transportation nurses who work at the same shift is about 12. In other words this means that some of them have two lists with patients. The next figure shows a part of the list. Well, the most important is the name of the patient. The date of birth is also important to know how old the patient is. This helps the nurse to find out the correct patient fast, if there is more than one patient in a room. A not so important column shows, which type of checkup is planned. Other important details which can be found out from the list are when the checkup is planned and if the patient has to lie in his / her bed or if he / she can walk. And of course the station where the patient lies is shown because otherwise it would not be possible for the nurse to find the patient.



Patientenname	Geb.Datum	Anf.Text	Unt.Datum	T.	D.	G.Kl.	Zuw.Station
[redacted]	1929.05.26	CT Lunge Standard	2006.05.11	L	R	SV	Chir.St.42
[redacted]	1955.08.22	CT Planung Becken	2006.05.11	G	R	SV	Onko.Amb.

Figure 9: Main parts of the transport list (Source: Personal communication from 11.05.2006)

But the nurse does not transport the patients in the order which is given by the list. Each transport is agreed with the assistant at the medical unit. This means that before a patient is fetched from a station the assistant has approved to this. Some changes of the order are made by the assistant and some are initiated by the nurse. A reason for this is that some patients are at the same station and they may be fetched together to save time. A further reason is that a patient is fetched from a station to which another patient is returned. If patients are added within the shift of the nurse he / she does not get notice of this automatically. So the assistant tells the nurse about the new patients which were added to the list. Sometimes the nurse also gets called by telephone to get notice about an added patient. But most of the time the information comes from the assistant. The transport of the patients back to the stations is very simple. Every time the nurse brings someone to the medical unit he / she sees if someone is ready to be brought back to a station.

The second example is the University Hospital Graz. There an own transportation service is available. The doctors enter all transport requests into the system. This means that he / she enters the request for a checkup at a medical unit. The transportation service carries out the request. Any nurse who transports patients has a DECT mobile phone. Another nurse is sitting at the central station in front of a computer where he / she sees all the requested transports. This nurse sends the requests as SMS to the mobile phone of a nurse. After getting the SMS the nurse picks up the patient and brings him / her to the correct medical unit. After completing the transport the nurse confirms the transport. The central station of the transportation service gets noticed of this to get to know that this nurse is free for the next transport. Then the next SMS can be sent to this nurse. The transport back the stations, is also organized trough the computer system. The assistant at the medical unit enters the request therefore into the system and the transportation service handles the request like any other.

4. Workplace Study Findings

The results of the studies of the paper based transport list are discussed below. But first the scenario of the PDA is described.

4.1. Paperless scenario

When the nurse starts his / her work he / she takes the PDA and starts the transport list application. After the application is loaded the nurse enters his / her username and password and logs on to the Siemens Radiology Information System. After that the default transport list is shown. If he / she is only responsible for some patients the options dialog makes it possible to specify the criterions for them. After that the nurse has a list which contains the patients he / she needs to do his / her job. All the needed data is available in this list. There the patients name, the patients station, the time when the patient has to be transported and if the patient can walk is shown by default. After marking an item in the list additional data, like the date of birth, is shown. Now the nurse is able to start into his / her working day. This means that the nurse can transport the first patient. After completing this transport the nurse marks the list item as transported. Then he / she works on with the next patient.

4.2. The Comparison

Now a look on how the work is done and how it can be in the future. The following lines will discuss the differences and the benefit of this.

4.2.1. Preparation

The first part is the preparation before the nurse can start the transportation of the patients. Using the paper based lists the nurse has to print out the needed lists. Therefore he / she has to log on to the Siemens Radiology Information System. Then the transport list window is shown. After this the nurse can specify some criterions in an options dialog. Then the needed list is shown. Now the printer is used to bring the list on a sheet of paper. Using the new system based on the PDA the first steps are all the same. The only difference is that the printing is not done any more. The second difference is that the login and the specification of the criterions may be done somewhere in the hospital where a wireless connection is available. There is no more imperative to do this at the computer at the transportation office.

4.2.2. Doing the Work

When doing the transportation itself the difference is much bigger. Now the nurse goes with his / her sheet of paper to the different stations to fetch the patients. When he / she found the right patient the nurse takes him / her to the radiology or back to the station. Then the name of the patient is marked as transported what means that it is crossed out. With the new system the nurse looks on her PDA. There he / she sees the next patient who is waiting for transportation. On the PDA also the station from which or to which the patient has to be transported. No action is needed for this. This means that the nurse has the same needed information as on the sheet of paper. And if he / she sometimes needs additional information like the date of birth or the weight these data can simply be provided on the PDA by clicking on an item. After the transportation the nurse makes a double click on the patient to mark him as transported. After a few minutes the patient is automatically removed from the list. Because of this the next patient moves to the top of the list. So no scrolling must be done by the nurse at a normal working day. But the real benefit is described in the following lines. If more then one nurse transports patients the other nurses get informed automatically. Because by default many nurses work in a hospital to transport patients this is an interesting aspect. Now the nurses have to agree which patient every one of them transports within her working day when they start. The reason for this is simple that there is no real good way to inform the others about the change. With the use of the PDA the nurse can take some patients to where they are needed. Then he / she marks them as transported and every other nurse automatically gets informed. Another reel big benefit is that it now is possible to change the list during the day. When there is a new transportation request the nurses have to be informed of this. With the paper based list there are only two possibilities. The first is simple that the nurse goes from time to time, say about every hour, to the PC and looks if any new requests are available. But then it is not possible to add immediate requests. The other way is that the nurse gets informed through a phone call. But this means additional work for someone who has to make this calls. With the PDA based transport list this is no problem any more. Any new request is added automatically and right away to the list what means that the nurses are automatically informed about the new requests. In other words, this means that a little bit of time can be saved every day.

4.2.3. Changing the list

The last which had not been described until now is the possibility to change the list during work. Using the paper based list the nurse must print out any list he / she may need during the day. An example will show best what is meant with this. Take a nurse who is primarily only responsible for X-ray number one and two. At the beginning the transports are done as normal but then a very long checkup is needed at both x-rays. With the paper based list now he / she had to go to the PC to look if he / she can help someone of the other nurses. But after that, there is the question, how the nurse informs the other one that he / she has transported a patient. With the transport list on the PDA he / she just enters the options dialog to specify some other criteria. Then the nurse can see where help is needed. And also the information about the transport is no problem. It is done automatically.

5. Technical Development

By developing the application several problems appeared. One problem is the security risk when using wireless connection for the data transfer. There is an additional risk because the system is designed to be used in a hospital. A high security is needed when the data of patients is transferred wireless because this data is very sensitive (Egea-Lopez et al., 2005, Babb, 2004, Mupparapu and Arora, 2004, Miller, 2004, Ackermann, 2005, Sims, 2004).

5.1. Wireless Connection

The connection using radio has many security risks. Some of them can be eliminated easily and others need more work. The best way for a secure connection is to prevent the network form unauthorized access. This means that a bad client should not be ably to book in into the network.

5.1.1. SSID

Therefore it is good if a client even doesn't know that there is a network. This can be done by disabling the SSID broadcast. But it is also possible to find out the SSID by analyzing the radio signals (Bundesamt für Sicherheit in der Informationstechnik Projektgruppe Local Wireless Communication, 2003).

5.1.2. MAC address

The next step to a secure network is to restrict the hardware address of the clients. But it is feasible to fake a MAC address to have one that can book in into the network.

5.1.3. Wired Equivalent Privacy

Because of these problems it is necessary to crypt the wireless traffic. A way to do this is the use of Wired Equivalent Privacy (WEP). But a problem is that every client has to use the same key.

From this it follows that the key management is a big problem. To change the key, the change has to be done at each client. Following this the key will never be changed. So if a client gets lost or hacked the key gets public. The WEP algorithm also has other problems. The key length is enough but the initialization vector is to short. So it is likely that two packages have the same bits to crypt. Then the key can be cracked. Additional it is possible to crack the authentication protocol by recording a full process. The integrity check is also very weak because it is linear. And like this is not enough the design is weak. It is possible to crack the algorithm with a statistical analyze. With the latest findings only one million packets are needed for the analysis and within a hospital network it is no problem to get this amount of packets in a short time. With enough traffic this is possible below an hour.

5.1.4. Wi-Fi Protected Access

Because of this weakness the Wi-Fi Protected Access (WPA) has been developed. WPA has the same architecture as WEP with the difference of dynamic keys based on the Temporal Key Integrity Protocol (TKIP). WPA uses the RC4 algorithm which has also been used for WEP. Additional security mechanisms are a by packet key mixing function and a message integrity check. For the small office and home office area it is possible to use pre-shared keys. If a weak password is used then there is a big security risk. An attacker can generate all possible pre-shared keys based on the password. Then only one client has to sign in to find out the right key because the key exchange is only protected by a MD5-Hash. Since 2004 there are programs for MacOS X and Linux so it can be seen that it is a real problem. This makes it very important to use a strong password. In big companies the Extensible Authentication Protocol (EAP) can be used in common with radius servers. But radius products are different if they are from different vendors. Because of this also big companies sometimes tend to use pre-shared keys. The result of this is that the selection of the passwords is a central point. It is important that an experts helps the users that they choose strong passwords (Wikipedia, 2005-2006b, Wi-Fi Alliance, 2005, Informit, 2005). Because of the low speed of the CPU from the PDA an asymmetric cryptography algorithm can not be used. Only a simple symmetric algorithm can be applied. This would make the connection vulnerable. To solve this problem WPA is used to secure the connection.

5.2. The Relay Connection

This subchapter shows how the connection between the relay and the server is secured and which technologies are used therefore.

5.2.1. SSL

This connection form the relay to the server is built up by using the SSL coding (Wikipedia, 2004-2006, Cisco Systems Inc., 2002). This standard has proved of success through the last years. It is a hybrid procedure what means that first a connection is established with the use of public / private keying. This connection is used to negotiate for a symmetric session key which is used for further connection to save CPU time. With this the connection between the relay and the server should be secured. But here a problem with the certificate became visible. The relay said that no trusted certificate was found. This means that the SSL connection could not be established because the client was not able to check out whether the server can be trusted or not. The problem was solved by copying the certificate file to the relay.

5.2.2. Bright Side Framework

Over this SSL Connection the Bright Side Framework (BSF) is used. It also has the side effect, that the communication is done firewall friendly because of the use of a standard port. So take a look on how it works. The call is transparently encapsulated on the HTTP protocol. The Framework allows the program to specify interfaces at runtime. This is done by using dynamic proxies. These proxies show the interface’s to the client but in reality an instance of HttpServiceInvocationHandler receives and handles the request. The framework is designed for thin and rich clients to get the results of a SQL statement from the server. The client tells the framework which result it wants and gets the result block by block. The real benefit of BSF is that it is very powerful on handling big amounts of data which is stored in a database. And it is open source so it can be used for free. The next figure shows how the framework operates (BrightSideFactory, 2003-2004).

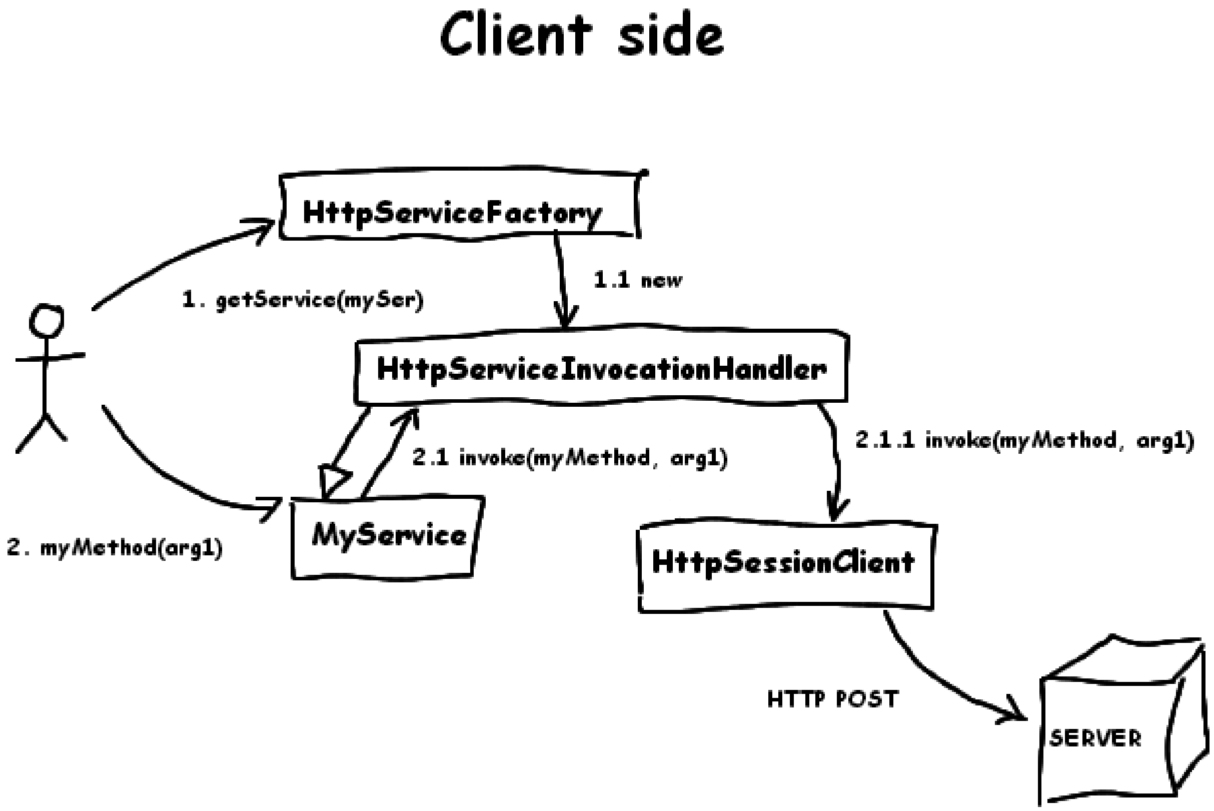


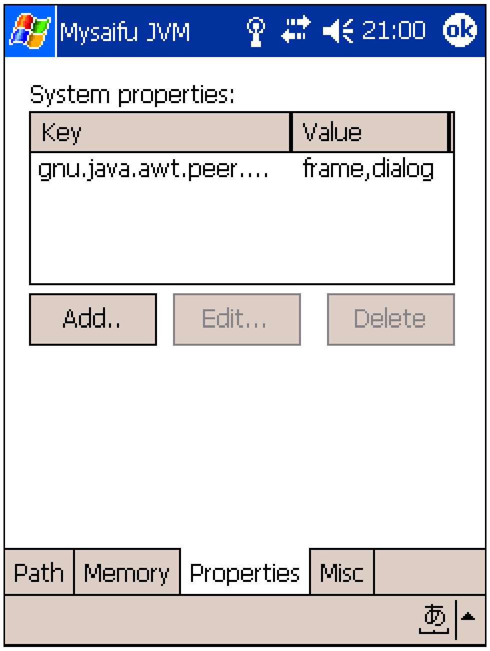
Figure 10: BSF client interaction diagram (BrightSideFactory, 2003-2004)

First an HttpServiceInvocationHandler must be created. This is done by the HttpServiceFactory. The handler implements the dynamic interfaces for the client. In the figure this is called MyService. When such a service is called the HttpSessionClient generates an Http Post call to the server. There a real SQL statement is executed and the result is sent back to the client (Sims, 2004).

5.3. The Java Virtual Machine

Because of the decision to use Java, a Virtual Machine is needed on the Pocket PC. Sun, the father of Java, does not have such a Virtual Machine for consumers to download. The argument for this is that the PDA's, mobile phones and other mobile devices capable to Java are very different. There opinion is that the vendors only should install an optimized Virtual Machine.

But Sun is not the only manufacturer of Virtual Machines. Everybody can build one because of the official specifications. There is a free Virtual Machine for every Windows Mobile 2003 and above called Mysaifu JVM. This is not a Java 2 Micro Edition but a Java 2 Standard Edition (J2SE) conform Virtual Machine which is built as a Source Forge project that can be downloaded under the GNU Public License Version 2 (GPLv2). The developing of it started in the in the beginning of the year 2005. Version 0.2 is used for this work. This version supports the Abstract Windowing Toolkit (AWT) and Swing. Small sample programs were used to ensure this. Of course there are some bugs by now but it support much more then a JVM for J2ME (Freebeans, 2005). The following figure shows the Mysaifu Java Virtual Machine.



Mysaifu JVM

System properties:

Key	Value
gnu.java.awt.peer....	frame,dialog

Figure 11: Mysaifu JVM (Mysaifu, 2005)

The main goal is that it is a Standard Edition JVM. This has some main benefits. The first is that it is very simple to test the application before copying the files to the PDA. The other thing is that J2ME is not J2ME. This means that there are many different versions depending on the power of the device. So it is not possible to build an application for all wireless platforms unless you only use the minimum of all which is not much (Sun Microsystems, 2005). The next figure shows the assembly of the Java 2 Platform.

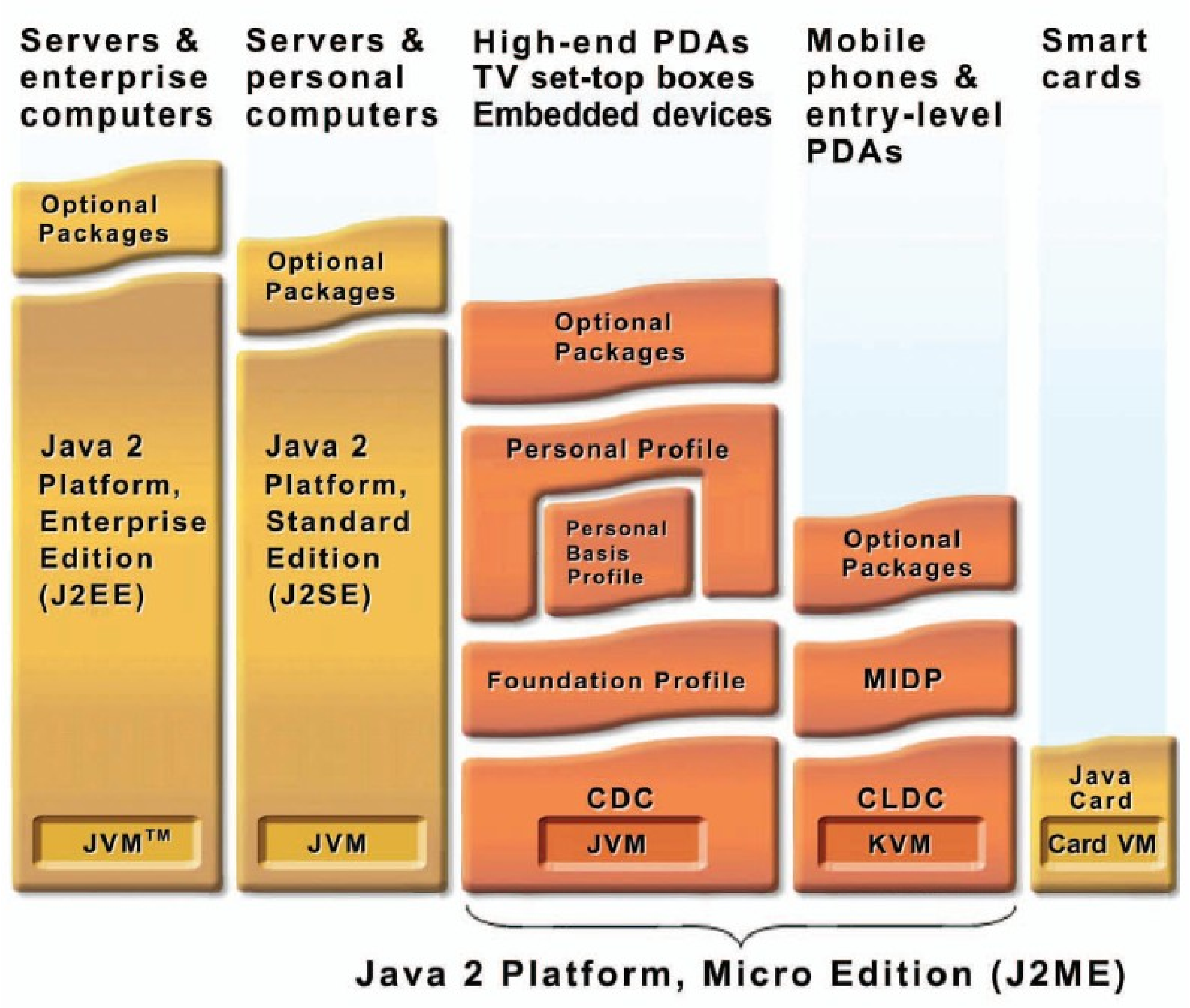

Servers & enterprise computers
Servers & personal computers
High-end PDAs TV set-top boxes Embedded devices
Mobile phones & entry-level PDAs
Smart cards
Optional Packages
Optional Packages
Optional Packages
Optional Packages
Java 2 Platform, Enterprise Edition (J2EE)
Java 2 Platform, Standard Edition (J2SE)
Personal Profile
Personal Basis Profile
Foundation Profile
MIDP
CDC
CLDC
Java Card
JVM™
JVM
JVM
KVM
Card VM
Java 2 Platform, Micro Edition (J2ME)

Figure 12: The J2ME, CLDC, and MIDP Specifications (Sun Microsystems, 2005)

Because of the different restrictions of J2ME the J2SE is the best way to build the application. That provides absolute compatibility to the actual system of Siemens call Web Order Entry. This makes it possible to reuse parts of the code to build the application for the PDA. And in the other way a reuse of code is also possible because the patient transport is not jet implemented in the Web Order Entry system. Maybe a change in the user interface would be good, to use the whole space of the screen.

There are also first efforts of a Virtual Machine for Java 2 Standard Edition for Palm PDA's which are running on the operation system Palm OS.

5.4. Setting up the Server

As already described above the server system, called syngo® Workflow Manager, is needed for development. This is a combination of a Sybase database, the Apache web server the Commsub which has been developed by Siemens. The first idea was to install the server within a VMware Workstation, which simulates a virtual PC. This did work fine, but the Computer was very slow, too slow to work. Cause of this, another PC was required and found. But the installation was full of problems. First there were some hardware troubles, like an old CD-COM which had problem reading the burned CD-R. After solving this problem the next came up. The reason for this was that the Computer on which the server should be installed had an AMD K5 processor which made troubles with the database server. But also this problem could be solved by finding a newer processor. After the change to the new main board with an Intel Pentium 4 processor the installation was really simple. First putting in the CD, then configuring some values, installing the license and the server was ready.

5.5. Building the Client

Building the Client is not simple because it is not easy to use the given structure. Also a problem was to bring the libraries to the client.

5.5.1. Porting the Libraries

The current system, the IWM Client, uses 40 different libraries which take about 13 MB. The problem was that the PDA could not load so many libraries. The reason therefore is that the length of the path is limited. Because of this it was necessary to join the libraries to one archive file, so that the length of the path is not exceeded. The problem of the combination was that some of the libraries needed a digital signature. Because there were only two libraries which had a digital signature the solution was to join only the others. These two libraries were used as they are.

5.5.2. Building the User Interface

Let’s take first a look on the Siemens IWM Client for PC’s and how the user interface is built there. Every dialog or window is defined within an XML file. Inside this file anything of the dialog is specified. Beside the standard attributes, like name, type, position, size and color, also the events are defined within the XML file. The following code is an example for this and shows the definition of the login dialog. The code shows how the events, which are triggered on action, are connected to the widgets.

```
<?xml version="1.0" encoding="UTF-8"?>
<dialog id="logon" caption="Logon.DialogTitle" modal="true" x="200" y="100"
            w="400" h="185" backgroundcolor="dialog.background"
            xs:noNamespaceSchemaLocation="gui.xsd"
            xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">
  <event class="de.siemens.med.pacs.iwm.client.events.SASClientEvent"
            name="TerminateEvent" trigger="onclosing"/>
  <panel id="logonPanel" x="0" y="0" w="100%" h="100%"
            backgroundcolor="dialog.background">
    <label id="usernameLabel" caption="Logon.Username" x="10%" y="20" w="30%"
            h="20" fontsize="11" fontstyle="bold"
            foregroundcolor="general.labeling" />
    <textfield id="Logon.UsernameTextField" x="40%" y="20" w="50%" h="20"
            fontsize="11" fontstyle="plain"
            foregroundcolor="dialog.foreground"
            backgroundcolor="general.labeling" onfocusselect="true" >
      <source type="nosource" />
    </textfield>
    <label id="passwordLabel" caption="Logon.Password" x="10%" y="55" w="30%"
            h="20" fontsize="11" fontstyle="bold"
            foregroundcolor="general.labeling"/>
    <passwordfield id="Logon.PasswordField" x="40%" y="55" w="50%" h="20"
            fontsize="11" fontstyle="plain"
            foregroundcolor="dialog.foreground"
            backgroundcolor="general.labeling" onfocusselect="true" >
      <source type="nosource" />
      <event class="de.siemens.med.pacs.iwm.client.events.SASClientEvent"
            name="LogonEvent" trigger="onaction">
        <parameter name="usernameFieldID" value="Logon.UsernameTextField" />
        <parameter name="passwordFieldID" value="Logon.PasswordField" />
      </event>
    </passwordfield>
    <button id="Logon.OkButton" caption="Logon.Ok" x="3%" y="125" w="25%"
            h="25" fontsize="11">
      <event class="de.siemens.med.pacs.iwm.client.events.SASClientEvent"
            name="LogonEvent" trigger="onaction">
        <parameter name="usernameFieldID" value="Logon.UsernameTextField" />
        <parameter name="passwordFieldID" value="Logon.PasswordField" />
      </event>
    </button>
    <button id="Logon.ExitButton" caption="Logon.Exit" x="30%" y="125"
            w="25%" h="25" fontsize="11">
      <event class="de.siemens.med.pacs.iwm.client.events.SASClientEvent"
            name="TerminateEvent" trigger="onaction" />
    </button>
    <button id="Logon.HelpButton" caption="Logon.Help" x="72%" y="125"
            w="25%" h="25" triggerhelpaction="login" fontsize="11">
    </button>
  </panel>
</dialog>
```

The first intention to build the user interface of the mobile client was to use XML files. But there are two difficulties. The first is that many libraries have to be loaded to use the same scheme as on the PC's. The loading of the libraries takes a lot of time and after completion the system is slowly. And the next problem was that the Java Virtual Machine on the PDA does not support XML parsing.

Because of these problems the user interface has to be built static like a standard Java program. But some things can be made variable. The first is the color. Therefore a property file is used. Within this file the colors of the different widgets and the different dialogs are defined. This is necessary because the Siemens Radiology Information System can be customized in this way and it should be possible to give the mobile client the same look-a-like feeling as the user typically has on the PC. The second thing, which is variable, is the text. Because of the fact that the system is used in many countries it must be possible to adapt the text to the according languages. Another property file is used for this. A short example for such a property file can be seen below.

```
switch_user.background=8080FF
switch_user.panel_background=BDBDFF
switch_user.label_foreground=FFFFFF
switch_user.box_foreground=00003B
switch_user.box_background=FFFFFF

widgets.background=BDBDFF
widgets.foreground=000000
widgets.selection_background=00003B
widgets.selection_foreground=FFFFFF
widgets.inputfieldbackground=FFFFFF
widgets.inputfieldforeground=6666FF
```

To be compatible with the Siemens System where anything is defined within XML file a special labeling system is used. The label of a property is composed by two parts. The first part is the dialog or the type of widget and the second is the property. They are combined by using a dot. This format makes it possible to auto generate the property file for the mobile client. For this only a XML parser is needed, which simply has to transform the XML structure which its sub elements into a dot connected structure. This parser has only to be built once and then any property which is needed for the mobile client can be transferred very easy. This prevents someone who customizes the system from doing this twice. Only the system for the PC has to be customized and then the XML transforming tool is used to copy the current configuration with the correct structure to the property files.

Beside the two property files for the GUI settings a third file is used for some basic settings. The structure of this file is similar to the others. This basic property file has to be on a predefined place in the file system. In this file the communication type and the location of the other files is defined.

5.5.3. Before using the PDA

Before the application for the mobile client can be tested the Java Virtual Machine has to be installed. As described above a Java 2 Standard Edition JVM called the Mysaifu Java Virtual Machine is used on the client. The installation is simple. Just download the compressed binaries, extract them then copy them to the PDA and execute the installer. The installation is done directly on the PDA.

5.5.4. Solution finding for the PDA application

The first idea was, as already mentioned above, that the user interface should be built up with XML files. This was not possible because the current version of the Java Virtual Machine has no XML support. It was simple to find this out because the JVM displayed an error message because of a thrown exception which said that there is no native XML support.

The next idea was making a static user interface and to reuse as much as possible from the client which is designed for the PC. This was a relative simple approach. Possible reuse components were anything which is not used for building the user interface of the client. An example for such a component is the Communication Proxy. But there were troubles with this approach. The first problem was that the most components need another component so that it is not simply possible to reuse only some of them. So the idea was born to copy the whole application of the PC client and the new built part for the PDA to the mobile device. In principle this would work. But in real there are some problems. The first is that the program for the PDA is very large. This is because of the program itself and the needed libraries. Counting them offers a number of more then 40. The first problem of loading them is that the length of the path is restricted. But this can be compensated with a work around. For this all libraries are put together into a single library file. But that's not that simple as it seams. Because of the libraries witch are signed digitally. There are a few libraries which are signed with the SHA1 algorithm. These libraries can not be included in the general library archive for the PDA. But there are only a few so that it is enough to combine the others. Now there are only five libraries witch have to be loaded on the PDA. By trying this out the next problem appears. The JVM needs much time to load the program. The reason is simply that all libraries must be loaded before starting the application. And the libraries have more then ten megabytes. This is very much for a mobile device. And after loading them all, the real program can be started. And also the program is not small. All in all the time to start is about some minutes what is really much too long. If the nurses in the hospital have to wait some minutes when they need short information they will not use the software on the PDA because they can not wait so long. This shortly tells us that this can not be used in praxis in this way. Another problem with this approach is the logger. In nearly every file a logging is possible by using the log4j library. The problem with this is that the configuration file has a XML structure. This makes troubles on the PDA. But this would be the smallest problem of all. Because log4j is a freeware project of at SourceForge.net the source code is available. This makes it possible to change the code so that the configuration can be read out of a property file or something like this. For the configuration of the color or anything else which can be configured within a static user interface property files should be used.

5.5.5. The working solution

On the basis of the written above another solution was necessary. In this one the findings of the previous tries have to be considered. So there are some essential points which have to be considered of to get a working solution. These four important points can be seen in the list below.

- No XML
- Size
- Speed
- Logging

The application has to be built on this basis. The first point, shortly defined as No XML, means that the graphical user interface has to be defined without the use of XML. To prevent a totally static interface property files are used for the general setting, the color and the language. How this works is already described above.

The next point, called Size, is in principle not such a big problem. If the size gets too large it is possible to use a memory card. But of course a smaller program is better because of the purchase costs in hospitals. One memory card is not really expensive but a hospital would need one for each PDA. In the sum this is much money. Another possibility is the use of PDA's with a bigger memory. But such devices are also more expensive than such with lesser memory. What also has to be in the view is that the application in the future should be extended and so the size of program will increase. So if the actual size is already is very large than an increase of it hurt in the wallet because of the amount of memory which is needed more.

The Speed is very important. If a nurse has to wait for minutes after she clicks somewhere then she will not use the software. So the goal is to make the program fast. And it will be fast when there is not much to load and when there are no long difficult operations. The consequence of this is that the Speed and the Size are directly connected. If the program is small-sized then the running speed will be fast enough.

The logging currently is done by the log4j framework. The configuration of it is done with XML files. This would not work on the PDA. But there is another version of log4j called log4jme. This is a very small and fast version of the framework which is compatible to the normal version. The only difference is that some advanced features are not implemented. So this problem can be solved very easy.

5.5.6. Making the application run

Based on the results above a running solution is built. In this solution the communication classes should be taken over. To do so, some other classes were also needed. These classes are the exceptions which can appear and the user classes to authenticate the user before using the program. The model classes are also needed. They wrap the data which is transferred between the server and the client. What this means in packages can be seen in the next list.

- de.siemens.med.pacs.iwm
    - client
        - exceptions
        - proxy
    - model
    - server
        - exceptions
        - interfaces
    - user
    - util

Mostly not the whole package was taken. Only the classes, which were needed for the package proxy, which encapsulates the communication, were transferred to the PDA client's project. Of course many of the classes could not be used one by one. They had to be adopted. The reason therefore is that the XML defined part had to be deleted or replaced by a non XML part.

After adapting the classes of the PC client the classes of the mobile client had to be built. These classes are built under the package mobile within the given Siemens system. The package definition below shows the structure.

- de.siemens.med.pacs.iwm
    - mobile
        - gui
            - components
            - interfaces
        - main
        - model
        - relay
        - util
            - interfaces

In the following paragraphs the package main, gui and so on means the defined directly above. The main classes of the applications without any graphical components are in the package called main. The graphical components are under gui.components, whole windows or a combination of components are under the gui package. Interfaces for common functions which are needed for the GUI are in gui.interfaces. The package model is used for classed which are designed for the data exchange between classes. The relay package is for the classes which build up the functionality of the relay. The other classes with needed utility are under the util package.

The main class is called IWM. This class initializes the logger, the connection and creates the application. The IWMMainFrame is the class which contains the graphical user interface. Therefore it consists of a collection of the different components which are defined in the gui package. It consists of following three panels.

- Login panel
- Transport list panel
- Options panel

The first is the login panel. This only consists of two labels and two text fields to enter the username and the password. To login the user, this means the nurse, enters the login data and presses OK. After this the login on the server is executed and the result is evaluated.

After the login the transport list panel is shown. There the nurses can see the patients which have to be transported form where to where. To add restrictions to the list the options panel can be used.

It is shown by pressing the options button within the transport list panel. There the nurse can for example select that he / she only wants to see the patients who have to be transported to the x-ray with the label “R1”. After pressing OK the transport list panel is updated and shown.

This first version of a prototype is described in the next chapter. Also the problems with this solution are discussed there.

5.6. Client - Server Connection

The general idea was to build up the connection with the use of the Bright Side Framework. The following subchapters will discuss the problems which occurred on the way to the final communication solution.

5.6.1. The Connection using BSF

As already said the first idea for the communication was the use of BSF. To make that possible some modifications in the log4jme framework were necessary. Because the log4j and also the mini edition are distributed under the Apache 2 license, the source code is available and can be adopted. The reason for this is that the framework is built to use the standard log4j. The differences are some additional functions. This three needed functions are all called log. This means that one functions are overloaded twice. The differences of the functions are the caller string and the parameter of a throw able object. The code below shows the maximum of the possible parameters.

```
public void log(String callerFQCN, Priority priority, Object message,
          Throwable t){
  if(hierarchy.enableInt >  priority.level)
    return;
  if(priority.isGreaterOrEqual(this.getChainedPriority()))
    forcedLog(priority, message, t);
}
```

The string of the caller is also logged in the standard log4j framework. But as it can be seen in the code above the string argument is not used in the mini edition of the framework.

But after the elimination of the problems with the log4j framework another one came up. The error is in the java.lang.reflect.Proxy class within the getProxyClass function. To find out this was not as simple as it seams. The reason for this was that the same code, which did not work on the PDA, worked fine on the PC. The error that occurred was a NullPointerException. Because of this the first idea was an error in the BSF framework. But after a long search, to find and repair the error, no problem could be found. The final realization is that the error itself is within the virtual machine or in the common java classes.

5.6.2. The Connection using RMI

After that finding it was clear that another way of communication had to be found. The logical consequence was to use RMI because this is another way that is already implemented at the server. The server side end of the communication is the JBoss application server. Because of this its libraries are needed. They include the transaction description and the security concept which is much more important. But there were also problems. The first again was with the log4j framework which was once more used for the logging. This time the needed changes were much more. Also the structure of the class hierarchy had to be changed. The required changes resulted in three additional classes. The following list shows all files which had to be adopted.

- org.apache.log4j.Category
- org.apache.log4j.Level
- org.apache.log4j.Logger
- org.apache.log4j.LogManager

The Category class is only needed as super class for Logger. The Level is just a synonym for the Priority class because the standard log4j framework has such a class for a more detailed specification of the priority of the log message. The class is a sub class of the Priority class. The LogManager class has some synonyms for functions of the Logger class. In the standard framework these functions are generally in the class of the LogManager.

But after these modifications the next problem occurred. This time it was an InvalidClassException. It appeared at the RMI lookup. The reason therefore is that the class which was sent from the server was not compatible with the local class. The serialVersionUIDs of the ObjectStreamClass, which identify classes clearly, were different. The finding out that the Mysaifu JVM is not completely compatible to the Java 2 Standard Edition was the logical consequence of this.

5.6.3. Building up a Relay

So another way had to be found to establish a connection to the server. Because no more possible ways for the communication were available another concept was necessary. And it was found using a relay. A relay is the only possibility for the communication, where no change on the server is needed. On the server side of the relay the connection is built using BSF or RMI. On the PDA side of the relay the communication is based on an object stream. At first take a look to the server side. The BSF and the RMI connections were given ways to access the data on the server. Both communication ways had to be adjusted to fit the needs of this application. The communication is no longer done over a static connection. The reason is simply that the relay has to be able to open many connections. Also some additional functions were necessary. The first is the function which returns the session id. The other changes are functions which were overloaded. The whole communication between the relay and the server is done through

SASTransferObjects. And the relay also gets such objects from the PDA client. These made it necessary to add some function to send them on without any manipulation. An example therefore is the logon function. The logon data is stored within such an object. But the function takes the username and the password as its arguments. So it was necessary to adjust the communication way that it is possible to do anything directly with such objects. On the other side there is the client. The new built communication way is also sub classed from the SASServerProxy. This makes it possible to change the communication way in the future by just changing the properties file. The first idea was the use of an object stream in both ways. But there some difficulties occurred. Generally the communication with objects from the client to the relay does not work.

But that is not that simple as it normally is on PCs because of the different JVM. Any class which is serialized with the default serialization operation of Java writes the class name followed by the version number, called serialVersionUID. As already noticed above, the Mysaifu JVM is not fully compatible to the java defaults by now. The problem is that it does not consider fixed version numbers which are implemented in the source code. It always calculates them. And exactly this is the problem. Every class which has a fixed version number causes an error at the relay when it is going to be deserialized. The InvalidClassException which is thrown points out that the serial versions are different. But in reality they are the same and only one number has been calculated wrong. The solution for this is to change the version number before deserialization. Any data which is sent trough a stream is sent as a series of bytes. In an object stream this is done with a special protocol which is public(Sun Microsystems, 2004). There is exactly described how the class and the serialVersionUID are printed out. Some general information is followed by the class and super class information which is succeeded by the class name which is printed out in UTF. Directly after the name the version number is wrote to the next eight bytes followed by some flags and the variables which of course can also be classes. And this serial number must be changed. Because of the knowledge of the object serialization stream protocol this is no more a big problem. The right version numbers are known by the relay. So it is no big problem to correct the versions numbers. Before the deserialization the byte stream is changed. The serialVersionUID of any class which is defined in the properties file is changed. After this change there is no more problem to regenerate the objects from the received byte stream. Now, after this adjustment it is possible to send SASTranferObject, with all classes it includes, from the PDA client to the relay.

The other direction, the communication from the relay back to the PDA client, can not be made possible that simple. The reason therefore is simply that the JVM on the PDA throws an InvalidClassException for each class. With each class, really each class is meant, except arrays of simple data types like byte. Theoretically it would also be possible to change all the version numbers. But the amount of classes which are transferred is too much to change all number on the PDA. Therefore the costs would be too height. The adjustment of the serialVersionUIDs would need too much time. So the idea was born to send strings. Of course not as an Object, just as a series of bytes. Therefore a very simple protocol was invented where the string is divided into fields with field terminators. The PDA just has to split up the field and rebuild the result set. This method works fine.

The big disadvantage of the relay is that this made it necessary to think over about the encryption of the data. But it also has an advantage. The number of needed libraries shrinks. Because the communication is built up with such easy methods, the size of the needed libraries is less then 100 KB.

5.6.3.1 The Encryption

Generally the WPA encryption which is supported by the WLAN hardware is enough to secure the data. But the client on the PDA is used within a hospital what means that very sensitive data is transferred. Because of this it was decided to add an additional encryption. The following two tables give a short overview about symmetric and asymmetric encryption algorithms.

Name	Type	Key length	Speed	Security	Use	Comment
One-Time-Pad	Stream	Text length	Height	Perfect	Special cases	The only algorithm with guaranteed security

DES	Block	56 Bit	Low	Special hardware needed to break	Standard algorithm for 20 years	The key length is the only practical problem
3DES	Block	112 Bit	Very low	No known attack	Commonly used	Too slow in software. Old.
IDEA	Block	128 Bit	Faster than DES	Very high	Commonly used	Patented; also in Europe
RC4	Stream	Variable	Height	Height	Commonly used (i.e. SSL)	Security problems with bad implementations
A5	Stream	64 Bit	Height in Hardware	Broken	Cellular phones	Breaking is quite easy
RC5 RC5a	Block	Variable	Height	Practically secure	Rarely used	Principle of variable rotation. Patented in US.
RC6	Block	Variable	Height	No known attack	Not known	Improvement of RC5. Same US patent.
Blowfish	Block	Variable	Height	No known attack	Open-Source Software	Freeware
Twofish	Block	Variable	Height	No known attack	Not known	Improvement of Blowfish.
AES	Block	128-256 Bit	Very Height	Only theoretical weakness	Commonly used. New standard algorithm	Successor of DES.

Table 1: Symmetric encryption algorithms (Wobst, 2003)

Name	Type	Key length	Speed	Security	Use	Comment
RSA	-	Variable; up to several KBit	Very low	Secure up to now	Most important Public Key algorithm	Based on the problem of factorization
ElGamal	-	Variable; up to several KBit	Very low	Secure up to now	Commonly used	Based on discreet logarithm
Diffie-Hellmann	-	None	Very low	Secure up to now	Commonly used (i.e. SSH)	Based on discreet logarithm

Table 2: Asymmetric encryption algorithms (Wobst, 2003)

The first idea, of course, was an asymmetric method because of the non broken security up to now. But as it was to be expected, a speed test of the RSA algorithm has shown that this is not possible. For the test a very short stream about 50 bytes was used. The decryption of this few bytes already needed nearly a second. Within the real application often data with more than one kilo byte is sent. This would cause the application to freeze for many seconds because of the work on the decryption of the data. And this is simply too long. Other asymmetric algorithms are also as slow as the RAS algorithm what means that they can also not be used for the client application on the PDA (Salomaa et al., 1996, Schneider, 1997).

This has shown that only a symmetric method can be used. But this is no real problem because the WPA encryption is also used. And this algorithm is hard to crack. In principle this additional encryption would not be necessary. But it enhances the security of the transferred data. And it is always better to have more then less encryption. Maybe someone deactivates the WPA algorithm unintentionally. Then the data is also not transported in plain text what would be a disaster. Because of the limited power of the PDA and the hard WPA algorithm only a very fast encryption algorithm can be used. The table of the symmetric encryption algorithms shows the possible methods. But some of them are not suitable (Wobst, 2003, Schneider, 1997) for this application. The reason therefore are simple that a fast algorithm is needed which is not patented anywhere in the world because Siemens wants to sell it to customers everywhere. The One-Time Pad would be best but it is not practicable. DES and 3DES are old and quite slow. IDEA, RC5, RC5a and RC6 are protected through patents. And the worst scenario would be A5 which is already broken. The result of this is that only RC4, Blowfish, Twofish and AES can be used. Of course there are some other unknown or proprietary algorithms which would also do the job. But AES is a common and very fast encryption algorithm. This new standard algorithm has already been analyzed towards security vulnerables. This nearly ensures that there are no chances to crack the algorithm in the next years. These are the reasons why this algorithm has been chosen for the application.

The implementation of the algorithm was not done within this project. A free implementation of the ACME Laboratories was chosen to encrypt the data. They have a free Java implementation of many encryption algorithms which can be used within business applications. In this project it is used with a 16 byte key which is equal to 128 bit. The key is calculated from a pass phrase which has to be specified in the property file. The used block size is 32 bytes. That is the maximum which is supported by this algorithm. But this is ok, because the calculation time which is needed for the encryption, decryption and the lookup for the data on the server is up to 5 seconds. But it is not felt so long, because the nurse sees the old list until the new one is loaded. Maybe Siemens or the customer will decide that no encryption beside the WPA is needed and change this in the property file to save time. It is also possible to change the whole algorithm, because this part was built to support other algorithms. If the AES algorithm is broken, it can be simply replaced by another one.

5.6.3.2 Different Character sets

Interesting was also the test with none Standard English characters like the German “äöüß”. They are encoded differently on the PDA. For example, on the PC the “ü” is represented by a byte with the value -4. On the PDA the value for the same character is -127. Other languages with additional character will also have such problems. To add a support for these languages, an adjustment of the characters can be done after the transfer. Which changes have to be done can again be specified in the property file which contains all general settings. This makes it simple to adopt the system for many different character sets.

6. The Prototype

In the following pages the evolution of the Graphical User Interfaces can be pursued. This chapter will give an overview on how the final GUI was developed.

6.1. The first Try

This solution was defined to find out the problems of the PDA. The idea was simply to create an interface that the nurse makes remind the user interface of the PC. Now take a look on the different windows, how they are defined and the related problems. At the very first the general structure is the point of view.

6.1.1. General Structure

The Application is started by the IWM class. This class changes the current path to the application directory, initiates the Logger class and starts the application. The last means that an instance of the IWMMainFrame class is created. This class consists of three panels. The switching is done by the CardLayout. This means that the whole application is loaded when the application is started. The panels are logically the Logon, the Transport List and the Options panel. By default the Logon panel is shown. But there is the possibility to specify the username and the password in the property file which contains all the general mobile settings.

Because of the possibility that the colors of the widgets can be defined in a property file, there are many classes which are able to take care of this by there own. For every type of the used widgets a subclass of the original swing component was built. The names of these classes are built by replacing the “J” by the string “PDA”. In Example the swing class for labels is called “JLabel” and because of that the subclass is called “PDALabel”. These subclasses get the settings for there colors from the PropertyFetcher class. To build up the user interface these classes are used. The advantage of this is simply that no further work is needed for the simple setting of the colors.

6.1.2. Logon

This panel is used to identify the user and to perform the logon. It is built up with the PercentLayout. This means that the existing space is used to display the components. Therefore the whole available space is divided into percents. Any component, which is placed into a panel with this layout manager, must specify its size and location in percent. For example the label with the caption "Benutzername" is located at ten percent from left and five percent from the top. The size is 80 percent of the width and the height is seven percent of the total window height. The benefit of this is that the dialog always has the same look. At present this is not so important because most PDA's have the same resolution of 320 x 240 pixels. But in the future there will be PDA's of the double resolution or even more. The first devices of this future are already get going. The next figure shows this panel. Also the specification in percent can be compared to the example above.

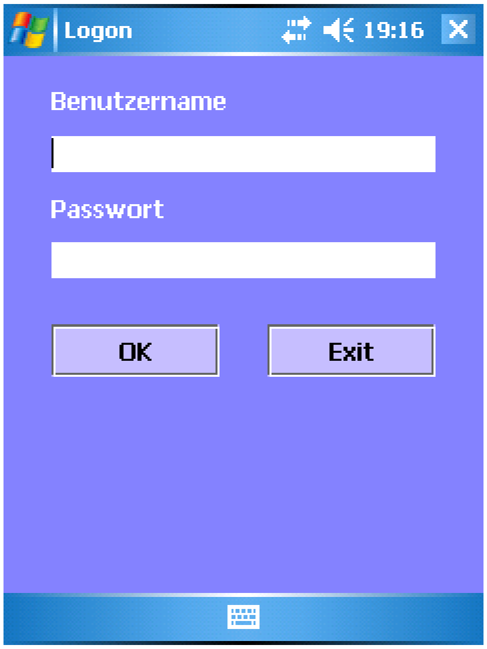
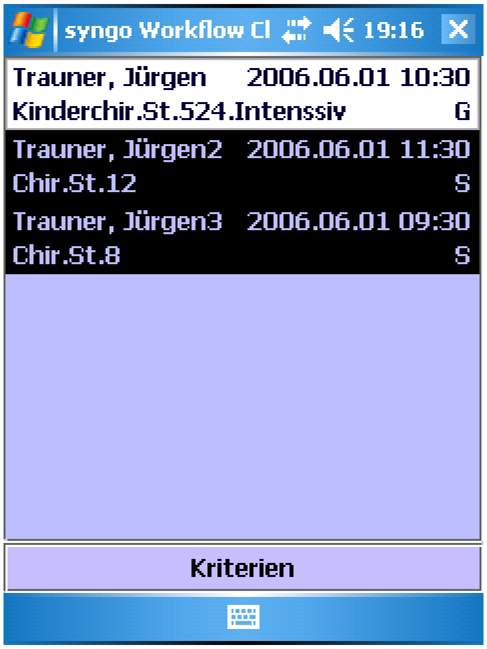


Figure 13: Logon (first try)

6.1.3. Transport List

After a successful logon this panel is viewed. The layout manager which is used here is the BorderLayout. In the south there is the button to switch to the options panel. And in the center there is the main element of the application, the transport list. The patients, and how they have to be transported, are shown there. In this version the four main fields, the name, the station, the date and time and the transportation type, are shown. In the final version it is possible to show more details. This looks like a very easy to build list but that is not true. It is simple to make a list with one line for each entry. But here we need more than one line. And as if that was not enough there are differences between the PDA and the PC. One difference is that the calculation of the preferred size. The calculation on the PDA results in the double height if the string contains a new line character. On the PC this has no effect. As a consequence of this an own calculation had to be programmed. The next figure shows the design of this panel.



```
Trauner, Jürgen      2006.06.01 10:30
Kinderchir.St.524.Intenssiv         G
Trauner, Jürgen2     2006.06.01 11:30
Chir.St.12                          S
Trauner, Jürgen3     2006.06.01 09:30
Chir.St.8                           S
```

Figure 14: Transport List (first try)

The much bigger problem was that the list on the PDA does not support containers like panels. It only supports simple components, like labels. Because of this it was necessary to build an own element rendering class. For this a helping class was built. The JLabel was extended therefore. This class overrides the paintComponent method of the super class. The rendering class sets the things like the color and the border and returns the component which is the same as the elements of the list.

In the final version two extensions will be necessary. One is the logout button and the second is the date and time of the last update.

6.1.4. Options

The options dialog is shown when the options button of the transport list panel is pressed. This is a very large panel so that it is not possible to show the whole panel at once. To make it possible that all the criterions can be used a scrollbar was added. The layout manager which is used is the PercentLayout again. The data for the components come from the database or can be entered by the nurse. An example for a free text component of course is the patient’s name. An example for a selection is the waiting room. The next two figures show all the widgets which are available to specify the citations which can be used to filter the shown rows of the transport list. To use the citations the nurse has to press the OK button. After pressing one of the two buttons the transport list panel is shown again and updated by using the selected citations.

But the scrolling of the panel is a big problem. It is too slow. Maybe this depends on the combination of the PercentLayout and the scroll bar which was added in the east of the underlying BorderLayout. Every time a scroll event appears the dialog must be repainted. Because of this the panel has to be changed. Maybe only changing the layout is not enough because a scrolling event always needs to repaint the whole screen.

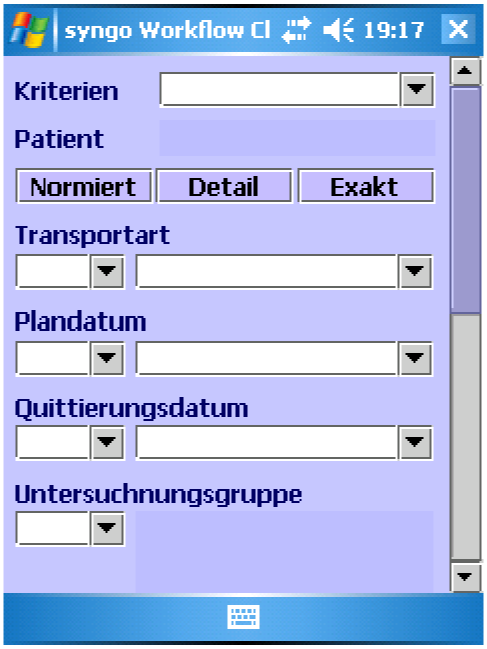


Figure 15: Options, first part (first try)

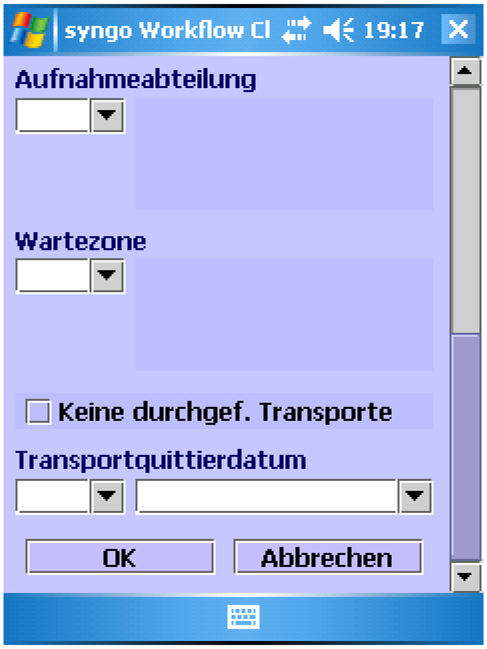


Figure 16: Options, second part (first try)

6.1.5. Problems and possible Solutions

With this first solution several problems appeared as described in the following.

- Keyboard size problem
- Speed problem
- Not default mistake
- Options scroll problem
- Other known mistakes

The keyboard size problem means the problem that the keyboard reduces the screen size. Every time the keyboard is chosen the screen size shrinks. The program then has to manage with the rest of the space. In general this does not sound like a problem. But the Logon and the Options panel use the PercentLayout. In this case it gets to a problem. As the name of the layout manager let expect the whole space of the screen is use and the widgets are placed on the screen with constraints that specify the place and the position in percent of the screen. Up to now this seems very good because of the independence of the physical screen size. But when selecting the keyboard, the screen size gets smaller because of the space which is needed by the keyboard. After showing it, the program gets notified to repaint its screen to fit the new situation. Then the percent layout gets in action. Because of the fact that the screen size is now smaller the layout manager renders all the components new. Because of this every component gets smaller. And it is necessary to render the size and the position again. This causes a lot of work for the PDA. The logon panel with only six components does not make a problem. But the options panel makes big troubles. After selection the keyboard while the logon panel is shown the layout manager makes the rendering work and repaints the screen. This is done below one second what shows that it is not a real problem. The only problem here is that the components get smaller. In real this may be a problem for some nurses who have little problems with there eyes, maybe because of there age. And on this way the simple looking problem can get to a big one because if the nurse cannot read the screen he / she will not want to use the program. But at the Options panel the problem is much bigger. There are many components in this panel which make the necessary time to render the components position and size much more. And every time the keyboard is selected or also if it is close afterwards the layout manager gets in action. Because of the need to change the coordinates of the components and repaint them afterwards the use of the keyboard is a challenge to the patience. The reason therefore is that this need about three seconds. And of course the problem that the components get smaller also appears like before.

The speed problem means that the application starts to slow. And as already mentioned before the resizing of the panels after selecting the keyboard is also to slow. But this should be solved by using another layout manager. But the starting problem still has to be solved by now. This should be done in three steps.

The first step is to review the needed libraries. Maybe not all classes of the library are needed. If this is so then a new library should be built which only contains the necessary classes. The reason for this is simple to shrink the size of the libraries which have to be loaded. And if the size is smaller then the load time is also less. This is not very difficult because the dependency within a package can simple be found out because the class files contain the information of their needed super class and their used interfaces. And the types of the arguments of the methods can also be seen. So it is not a big problem to find out which classes are needed to fit all dependencies.

The second step is the loading process itself. This prototype loads all elements which are shown on the display at the beginning. This mean that all three panels, the logon, the transport list and the options, are loaded before anything is painted on the screen. A better solution is that only the first panel where the user enters his / her username and password is loaded at the beginning. For this the user needs some seconds of time. Within these seconds the next panel, the transport list, can be loaded. The benefit of this is simple that the user can start to work after a shorter load time. For the users view on the running application this makes no big difference because the next panel is already loaded when it is needed. For the user this has no disadvantage but a benefit in the running speed. For the options panel the same strategy can be used. While the user looks at the screen and uses the functions of the transport list panel the options panel can be loaded. The only disadvantage of this is on the programmer’s side. The reason for this is simple that it is more work to load it in the background than to load it all at once. In this case it is necessary to use an own thread for the loading in the background. And if the next panel is needed then a check is necessary to look if it is already loaded. And unless the loading has completed the thread, which is responsible to show the panel on the screen, has to wait. In reality this will appear very rarely. But it can and so it is necessary to take care of this case.

The third step is to use own frames for every panel. This helps to reduce the needed memory. Because after the user is logged in the logon frame can simply be removed from the memory. Another important argument to use panels is discussed in the next paragraph.

The not default mistake means that it is the default on the PDA that the close button in the title bar is used for okay or cancel buttons. An X in the title bar means cancel and the symbol for okay is simply represented by these two characters OK. Because of this that the cancel button generally will not be used very often it could be banished into the title bar. Experienced users will be common with this. Also inexperienced users will not have many problems because they normally only need the OK button. And this one will stay as a button like a PC user would expect it. And if the cancel button is needed a short look over the whole screen will solve this difficulty. As already addressed in the last paragraph, own frames for the different panels should be used. And every frame already has an X in the title bar for closing the window. This can be used for the cancel button. This is a very simple way to bring the cancel button into the title bar. Because different actions are needed, when pressing the cancel button in the different panels, it is better to use own windows for them instead of using one window and changing the action when the view is changed to another panel.

The options scroll problem is really one of the bigger ones. This view shows the different options which the nurse can set up. As already described in the options paragraph above, every time the user activates a scrolling action the whole window must be laid out again. If there are only a few components then the processor of the PDA is sufficient to perform the necessary actions of the layout manager. Because of this scrolling should not be used. Instead of this the panel can be split up into two panels or two windows. Then the scrollbar is not needed any longer. For the switching between the two panels a button is needed. This makes sense because in this way it is possible to give the more important widgets on one panel and the remaining components to the other panel. So it often will simply not be necessary to load the second panel. Another possibility is to shrink the possible options to the important and to cut off the others. In other words this means to remove the second panel completely.

Other known mistakes mean the difficulties which are not real problems. But if it is possible to avoid them the working with the program will get easier. Examples for this are the information about the last update or a logout button.

The last update is necessary because if the nurse is somewhere in the hospital without a wireless connection to the server it is not possible to update the transport list. And if the nurse usually gets an update of the list every five minutes he / she may think that the program does not work anymore because the list had not changed for more the ten minutes. In this case the time of the last update indicates a problem with the connectivity what tells the nurse to change somewhere else with a wireless support. This will mainly be interesting in the beginning of the wireless revolution within the hospitals. The reason therefore is simple that the support will start in small areas and later on be expanded to the whole hospital. After the wireless connection is available in the whole area of the hospital this will be no more subject of discussion.

The logout button is a service for the nurses to save the time of restarting. The nurses do not work around the clock. Some of them have the day shift and some have the night shift. And of course not every nurse has his / her own PDA because that would be too expensive. The count of PDA's is simple the amount of simultaneous working nurses. When a nurse goes out of duty he / she just logs off and the next nurse can login instantly. This saves the time which is needed to start the application. Since large PDA applications do not start as fast as PC applications, the time needed for this is more then a few seconds. This does not sound as if this is much time because this few seconds but in the sum this is much time.

Another thing which is not directly a mistake is the cancel button. Because this button has been moved into the title bar it is no longer a real widget which needs space within a panel or window. So the okay button should be moved to the middle or to the right of the available space to fit the user's expectations.

6.2. Applying the solutions

The general idea was to fix the problems like described in the subchapter before. But that was not that simple as it seamed. The reason therefore is simple that not all assumptions were correct. All these will be discussed below.

As first the assumption is to use only standard layout manager. This means that the PercentLayout should be replaced by the BorderLayout and other layout manager which are available by default. There are two main reasons to do this. The first is that the PercentLayout is a additional class which has to be loaded. The conclusion of this is that the loading time and the space can be saved by only using the default layout managers. The second reason is that the PercentLayout changes the view when the keyboard is selected. Because the keyboard shortens the available space for the application all the components have to be reduced in their height. Other layout managers would also solve this problem. The simplest dialog is the first, the logon dialog. This makes it obvious to use this dialog to try this out. The dialog consists of two labels and two input boxes to enter the username and the password. These components are placed into the dialog with a big gap between them to use the whole space of the dialog. Below them the login button is place to start the login process, which uses the entered text. To make it possible to show the keyboard without changing the dialog all elements should be placed into the north of a BorderLayout. So a cascading BorderLayout was the first idea. This means that one component is placed into the north and a panel is placed into the center. The next component is placed into this panel and so on until all components are placed. To give the dialog a better look a surrounding BorderLayout is used. A label without any text is placed in the north, the east and the west. This is necessary because the gap which can be defined at the BorderLayout is only used between the components and not at the margins. Another idea was the use of the GridLayout in the north of a BorderLayout. The gaps to the margins must also be adjusted with an additional panel below this. It does not matter which of these two methods is used, the result is very similar. The rendering of the components on the PDA needs much more time than before. The reason for this is also very simple. The problems are the cascading panels with their layout managers. The painting of a component must be coordinated with all the panels. This is much more difficult then the use of only one layout manager which defines where the whole components of the panel are painted. The consequence of this is that the PercentLayout will also be used for the next versions of the prototype because the running speed is the most important factor on the PDA. The few more kilobytes which are needed for the additional class are not very important because the storage location is cheaply nowadays.

The problem that the selection of the keyboard causes the layout manager to repaint the screen can be solved by defining a fixed screen size. This means that the layout manager is initiated with the width and the height to which the percents match. And if a repaint is needed after selecting the keyboard the screen size is the same as before and so there is no effect to the components in the window. In one sentence this means that the PercentLayout must be changed in a way to never change the components size or position.

Another assumption is that it is better to use own windows for each component. But as tests have shown this was not true. The actions would be encapsulated better because the cancel operation would be next to the components. This would also be a benefit for the programmers. The reason why this is not used is simple that the process of showing the window needs more time than a switch of the panel. Also a nice to have is that the panel switching does not shimmer like the use of own frames. The only disadvantage is the additional time which is needed by the programmer because of the necessity of different actions when the user presses the X in the title bar. On the options panels it means cancel, transport list panel it means logout and on the logon panel it means exit.

The other assumptions how the problems can be solved for the client are used as the described in the last chapter.

6.3. The second version

This is the next step in the development process. All the findings above are included in this solution.

6.3.1. General

The workflow of the application can be simply described with a few words. The first that the nurse has to do is logging on. The logon information is sent to the server to perform a check if the entered data is correct. For the communication the BSF framework is used. It is a free communication framework which provides the necessary security. This means that it encrypts all patient data. After the successful logon the transport list is shown with the default filter. To change this, the two options panels are available. The next paragraphs will describe this in a more detailed way.

6.3.2. Logon

The Logon panel is the first that the user sees after starting the application. This panel is built up with the PercentLayout as all other panels to. The idea to use cascaded panels with default layout managers which are already included in the J2SE had to be turned down because of the speed problems which occurred trough this.

The important components of the panel are the two fields for the username and the password. There the nurse enters her personal user data. After this he / she press the OK button to start the logon process. The application sends a request to the server to perform the operation. If anything is ok, the transport list panel is shown. Otherwise the panel stays in the foreground and the user gets an error message. The logon screen can be seen on the figure below.

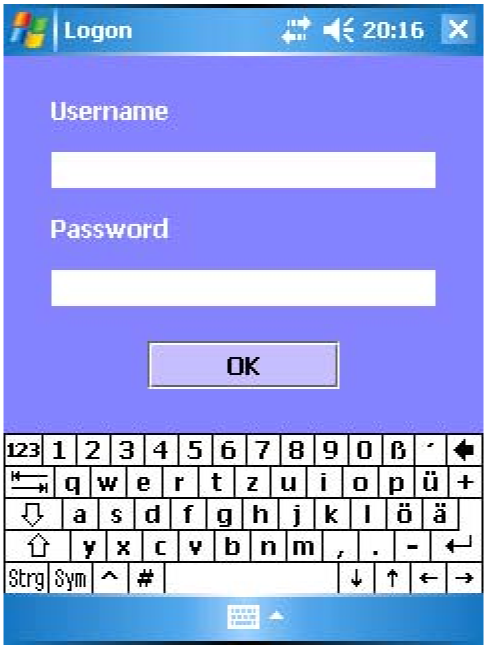
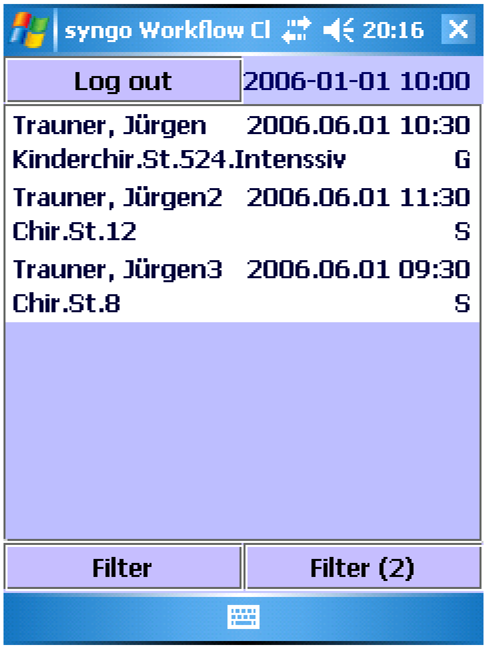


Figure 17: Logon (second version)

6.3.3. Transport List

After the successful logon the Transport List panel which can be seen below, is shown. This is the main panel of the whole application. The patients which have to be transported are shown in the list. The default parameters are used to initialize the list at the beginning. When the user clicks on an entry additional information to this patient is shown. After a nurse has transported a patient he / she can enter this directly on the PDA. For this the patient has to be selected, which means that all information is shown. Then the nurse can mark the patient as transported by making a double click on the entry. The reason why the patient has to be selected is simply that the nurse controls the patient data to ensure that he / she has the right entry. On the top and right of the panel there is a caption which shows when the last update was made. This is necessary because the PDA may loose the wireless connection because of some gaps in the coverage. Especially at the beginning of the use of PDA's in the hospitals there will be many gaps where the connection will be lost. This helps the nurse to decide if the list is up to date. Beside the already mentioned components there are three buttons. The log out simply logs out the current user and displays the logon panel. The other two buttons can be used to show one of the two filter panels to change the criterions of the list to display other patients which match to the filter rules. How this is done can be seen in the next subchapter below.



2006-01-01 10:00

Trauner, Jürgen 2006.06.01 10:30
Kinderchir.St.524.Intenssiv G
Trauner, Jürgen2 2006.06.01 11:30
Chir.St.12 S
Trauner, Jürgen3 2006.06.01 09:30
Chir.St.8 S

Figure 18: Transport List (second version)

6.3.4. Options

When the Button with the caption “Filter” is pressed in the transport list panel the first options panel is shown. There the nurse can specify different options for the filter as it can be seen on the next figure. For example, he / she can specify the transport type or the examination date. The Transport list button is like a back button to the transport list panel. When the nurse presses this button the expected panel is shown and the contained list is updated to fit the current filter settings. The “Next” button is used to get to the second options panel.

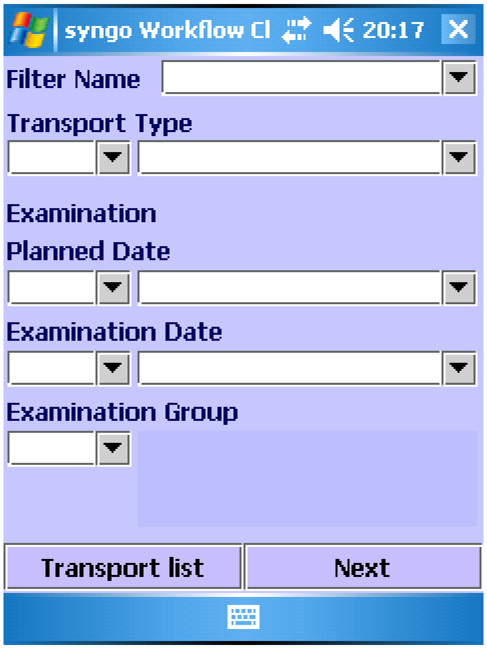


Figure 19: Options, first panel (second version)

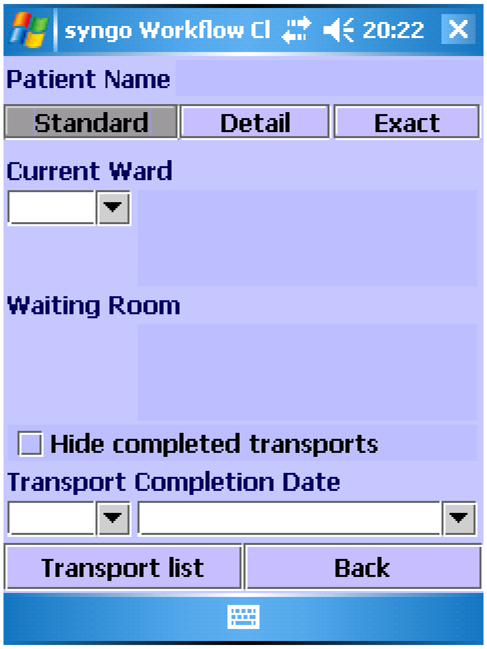


Figure 20: Options, second panel (second version)

The second options panel, which can be seen on the last figure, can be accessed directly from the transport list or indirectly through the first options panel. It is built up like the first panel. The whole available space is used for components which are used to specify different filter criterions. Examples for filter options on this panel are the patient name or the handling of already completed transports. In the bottom of the panel there is again a button to return to the transport list and update it with the specified options to match the different selected filters. And the other button with the caption “Back” is used to change to the first options panel.

The design was made to change all filter criterions and update the list afterwards. The change to the different criterions is done by the “Next” and the “Back” button. Only after pressing the Transport list button the new filter is used to update the transport list.

6.3.5. Problems and possible Solutions

Within the second version also several problems appeared as described in the following.

- No Close Events
- To Radiology / to Station
- Other known mistakes

The last idea was to use different panels for the different screens and to perform the cancel actions on the “X” in the title bar. But this was not possible. The reason for this is simple, that the default action on the PDA is to hide the window and to show the next one which is below the current. On the PC it is possible to execute own code when the closing button is pressed. On the PDA this is not possible because no event is generated to perform some actions which prevent the window of being hided. So some changed were necessary. The first was that every panel gets its own window, except the two options panels which share one. The options window is the only panel which has a default cancel operation which is performed by pressing the “X” in the title bar. This is possible because then the transport list window, which is the window below, is shown. The problem that the switching of the windows takes longer than the switching of the panels could also be eliminated by preloading the windows. If a window is needed it just gets shown on the screen but it is already in the memory. The shimmer is also prevented using this way.

The other two windows no longer have the close button in the title bar to prevent user mistakes. To perform the needed actions standard buttons are used. This means that the exit button, which was just removed, is again added to the logon panel. Also the logout button within the transport list panel can not be removed.

In the transport list panel one very important button is absent. There is no way to change the direction of the transportation. The patients have to be transported to the radiology and they must also be transported back to there stations. This means that there are two lists which have to be displayed. To save space on the screen the button is added instead of one options button. The loss is that there is now no way to get directly to the needed options panel. To minimize the effects of this disadvantage the options dialog always shows the panel which was used the last time. If this is the wrong panel the user has to use the next and the back button within the options dialog.

Other modifications which are needed are the change of the logout and the direction button and the change of the last update time string. The change of the two buttons has two reasons. The first is that the application should be consistent. In the options dialog the position of OK button to go back to the transport list is the left bottom. The logout button is the same within the transport list window. The second reason is that the direction is like headline. And so the logical position is at the top of the window.

6.4. The final Client

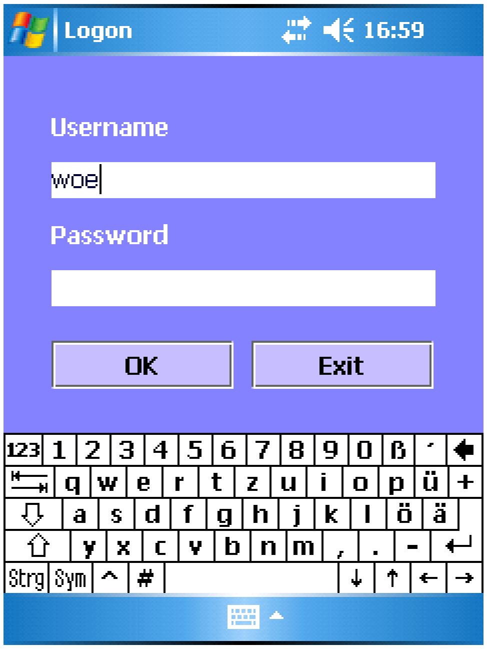
This is the final step in the development process of the user interface. All the findings above are included in this solution.

6.4.1. General

The workflow is the same which has already been described above in this chapter. For the communication the relay is used. The security of the communication is described in the chapter Building up a Relay.

6.4.2. Logon

As already mentioned above, the logon is done in an own window. Within the title bar no “X” is shown. The closing of the application cannot be done trough it, so there is no reason for showing it. It would only cause user mistakes. The exit button was added again to give the user a possibility to close the application.



Username

woe

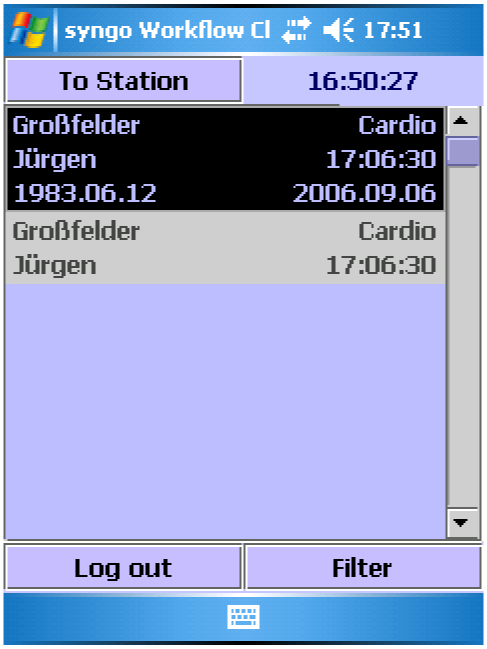
Password

Figure 21: Logon (final client)

6.4.3. Transport List

The most changes where made to the view of the transport list. As already told, an own window is used for the list. There is also no “X” in the title bar as in the logon window to prevent user mistakes. In the top left of the window there is a button to select the direction. The title of the button shows the currently selected direction. Possible values are “To Radiology” and “To Station”. By pressing it the direction is changed. Right to this button there is a label which tells the user when the last update of the list was done. This is necessary because of the automatically update of the list. The time interval of the update is specified in the property file of the client.

The list itself has also new options which were not described until now. For a better understanding the whole functions will be described here. After the start the list consists of items with two rows. There the patients name, the time to transport, from where and if the patient can walk is shown. After selection one of the items, the item pops up and shows some additional information, like the date of the birth. If one patient is transported the nurse makes a double click on the patients item in the list. After that the item is marked as transported. This is shown trough a light gray background. After selecting another item the marked item pops in and the color of the font of the selected item turns dark gray. This can be seen on the next figure.



16:50:27

Großfelder Cardio
Jürgen 17:06:30
1983.06.12 2006.09.06

Großfelder Cardio
Jürgen 17:06:30

Figure 22: Transport List (final client)

By the next update of the list the update is done on the server and the item is removed from the list. But it is possible to declare a minimal time in the property file, which an item is not updated. This is necessary to prevent mistakes. Because if the nurse realizes that he / she has marked the wrong patient it is possible to change this. Therefore it is only necessary to make a second double click on the marked item. Then the item state turns back to normal. At the bottom of the list we have two buttons. The logout button does what it says. The filter button opens the options window to specify some citations for the list.

6.4.4. Options

Only a few changes were done to the options view. It is also in an own window now. But here we need the “X” in the title bar. It is used to perform the abort click. After this the options window is brought to the bottom of all windows what means that the default action is performed. But that is no problem because the window below the options window is the transport list window. And this is exactly what has to be shown after pressing cancel. And the next time the options window is needed, it only has to be initialized again and all is like it has to be. No other changes where made to this window and its two panels.

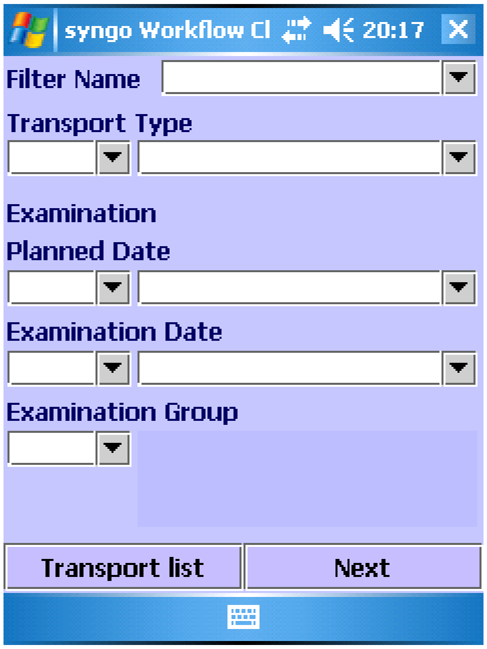


Figure 23: Options, first panel (final client)

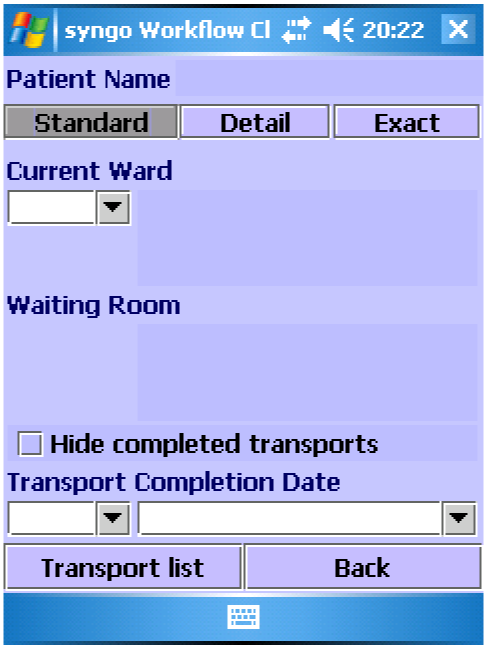


Figure 24: Options, second panel (final client)

6.5. Usability

First of all a definition, what we are talking about, is necessary. *"Usability has multiple components and is transitionally associated with these five usability attributes."* (Nielsen, 1993)

- Lernability
- Efficiency
- Memorability
- Errors
- Satisfaction

The learnability is simple that the system should not need much time until the user can work with it. The efficiency is the next step. It means that the user should be able to work with the system fast after he / she had learned it. The memorability is that the user should remember how he / she used the system to go working on without learning it again after some period of time of not having been confronted with it. The point errors mean that principal no errors should occur and if one occurs it must be ease to recover from it. The last point Satisfaction means that the user should like to work with the system. Best would be if he / she feels entertained (Nielsen, 1993).

6.5.1. Methods

The methods can be classified (Zhang and Vora, 1993) into three parts.

- Testing
- Inspection
- Inquiry

For all of these parts some of the methods are described exemplary.

6.5.1.1 Testing

In generally all methods in this sub chapter have high or quite high costs. But some of them can produce quantitative data (Holzinger, 2005).

Coaching Method

It is mostly used to design a better help or documentation. Three persons are involved in the test. These are the tester, the observer and the participant. It is also possible that the tester and the observer is the same person. While participant is working he can ask and the tester answers the questions. The observer records this and if the answer was helpful. After some participants a better documentation can be generated (Nielsen, 1993).

Co-discovery Learning

When this method is used then two participants work together. This should simulate a team work environment. The participants tell the tester what is okay and where they have problems within there given scenario. The whole conversation is recorded and used to increase the usability of the tested system (Nielsen, 1993, Dumas and Redish, 1999, Rubin, 1994).

Performance Measurement

The participants perform tasks given by the tester. The tester or a program records the results of the participants. Interactions are not allowed to prevent false results. Examples for the quantitative results are the number of user errors, the number of interactions or the needed time for recovering from errors (Nielsen, 1993).

Teaching Method

It can be used to find out how learnable a system is. Therefore two participants are needed. The first is instructed by the tester. After this he performs some tasks within the system. If he / she got quite familiar with the system he / she instructs the second participant how the system works. The second participant now also makes some tasks. If the system is easy to use for this participant than the system is easy to learn otherwise it is hard to learn (Anzai et al., 1995).

Thinking Aloud

This method gives the tester an impression how the participant feels. The participant performs some tasks. While he / she is working he / she permanently talks about what he / she is doing, what problems occur and how he / she feels. If the tasks are very complex the reports can also be at fixed points (Nielsen, 1993, Holzinger, 2004, Holzinger, 2005).

6.5.1.2 Inspection

The methods which are explained below are only performed trough software developers and area experts. No real user is needed (Holzinger, 2005).

Cognitive Walkthrough

This method is done by several experts. The experts have to evaluate the interface considering the background of the users. He / she has to tell a credible story why the user will do a certain action. The expert asks several questions to find out if the user will notice the action he / she needs is available and where it is available (Nielsen and Mack, 1994, Holzinger, 2004, Holzinger, 2005).

Feature Inspection

It is used to test features of the system. Each feature is analyzed for itself. But this does not mean that only one feature can be tested. If you are writing a letter you need some features, for example entering text and spell-checking. All of these features are tested one after another by using the same criterions. Example usability criterions witch can be tested for are the availability or the understandability (Zhang and Vora, 1993, Achatschitz, 2005).

Heuristic Evaluation

This method is used by experts to evaluate the user interface. Thereby they use guidelines which tell them how the user interface has to be. They have guidelines for the combination of colors or for the navigation. The inspection shows to what extent the guidelines match the system (Nielsen and Mack, 1994, Holzinger, 2005).

Perspective-based Inspection

The difference to other inspection methods is that more than one round is necessary. This means that three inspection sessions are made. Any one of this has another focus. The first is newbie use, the second is expert use and the last focuses on the error handling. The user interface domain is also beard in mind (Zhang et al., 1998).

6.5.1.3 Inquiry

Contextual Inquiry

This is not a real usability method. It is more a discovery of the future use of the system. It is used before the system is designed to get an idea what the user's background is and in with context he / she will operate (Achatschitz, 2005).

Field Observation

Using this method the tester gets directly to the work place where the user works. There the tester can see which problems a user has and how he / she manages them. Traditionally this is coupled with other techniques to find out as much as possible (Nielsen, 1993, Holzinger, 2005).

Focus Groups

For this method six to nine participants are necessary. Additional a moderator is needed who has a list of issues which are discussed in the group. Typically the tester listens to what they say and watches what they do. As result you get the reaction to ideas or to a prototype. What you do not get is how the users really work (Nielsen, 1993).

Interviews

As the name says participant are interviewed directly by the tester. The information which will be gathered must consist of data which can only be gotten in that detail trough interviews. Otherwise the interview would be the wrong method. Depending on the required information structured or unstructured interviews can be used (Nielsen, 1993, Holzinger, 2005).

6.5.2. Using Heuristic Evaluation

For this development the Heuristic Evaluation was used. The reason therefore is that it is a very cheap method. Only few testers are needed to find most usability problems. The tester may also not be very experienced because the testing is done using guidelines. These points make it easy to use. Another reason is based on the law. To use the PDA client in a real hospital an authorization from a supreme Federal State authority is needed. Therefore a long authorization procedure has to be passed. And finally this would cost a lot of money. This makes it impossible to test the client with the real users. This makes it clear why this method was chosen. The Guidelines which are used therefore can be seen below.

- High functional design: Do not use fancy designs
- Consistent usage: Consider the users mental model
- Reduce HCI interaction: Reduce especially typing text
- Error prevention: Only show errors if absolutely necessary
- Match between system and real world: What does the user see in real

The logon window is consistent to what the user would expect. Two input boxes for username and password. The interaction can not be reduced. These two fields have to be typed in.

The transport list window has also a high functional design. Nearly any component can be used trough a click on it to perform an action. On the top there is no drop down element to change the direction. Because there are only two possible values, a button was use to reduce the needed interaction to a minimum. If a list element is selected then it pops up. This is like defined in Siemens syngo®. And a double click is used to mark elements as transported. No extra dialog interaction to ensure this is needed. Because of this error preventions was necessary. After marking a patient’s item, it is displayed for at least one minute to correct the choice. But error prevention was also used if the wireless connection is no longer reachable. The user does not even notice this in the first moment and can work on as if noting happened. And because the list consists of patients the connection to the real world is also available.

Also the options dialog is consistent with this what the user would expect. He / she expects the same as on the PC. And it is nearly the same. Many drop downs and list boxes are used to minimize the users interaction. This is also good to prevents errors. The real world has also been considered. Examples are the waiting room or the stations which can be selected from a list.

7. Conclusion

After all I can say, that it was partial very difficult to build a useable application. The general idea to use a J2SE Virtual Machine to make it much easier could not be kept. The reason therefore is simply that the compatibility is not give for all needed functions. This made it necessary to change some attitudes. One point was the server communication which could not be done trough the already given methods. This made it necessary to build up a relay which makes a forward of the inquiries to the server. That made it possible to convert the result to a string which is sent as a byte stream. Because of the sensible data the encryption was also a big question. Primarily the WPA encryption is used. It is a secure and hard to crack encryption which is done by the hardware of the PDA. Additionally the data is encrypted by the application. After a comparison the AES implementation of the ACME Laboratories won the race. Because it is the new standard algorithm for symmetric data encryption, the algorithm can be graded as secure. The user interface of the PDA client also caused some troubles. It is not possible to build up a little more complicated interface which just uses the default layout managers. It was necessary to program an own one to fit all the needs within an adequate time. Another problem was that the PDA had a different window handling. It is not possible to get any window events within a Java application what must also be considered. Another big problem on the PDA which is totally unknown on the PC is the scrolling of complicated interfaces. The scrolling time is much to long because a simple scroll request needed up to some seconds. Anything had to be optimized to save as much computing time as possible. Then it is possible to build an application for the PDA which can be used in reality.

In comparison between an application for the PDA and one for the PC it can be said that it is much more difficult to build an application for the PDA than for the PC. The reason for this is the restricted computing time and other features which are not supported.

8. Acknowledgements

The work on this paper and the belonging application was in between very challenging and providing me a very rewarding experience. I want to use this chapter to thank the people who gave me support throughout composing this thesis.

First of all I want to thank Dr. Andreas Holzinger. He gave me support the whole time from the very beginning to the end. Thank you for your time which you spent through the last eleven months to produce this qualitative thesis. I also want to thank Dr. Stefan Biffl who was the first supporter for my work. Further on I want to thank the Institute for Software Technology and Interactive Systems (IFS) Vienna and Institute of Medical Informatics, Statistics and Documentation (IMI) Graz which made it possible for me to write this paper.

And of course I thank the Siemens AG Austria. I especially want to thank Herbert Kaindl who did the necessary administrative work and provide this topic to me. Special thanks to Thomas Birbach who supported me with all necessary information about the Siemens system which I needed during my work. Thank you for spending so much time to support me. I also want to thank Wolfgang Gratzl who helped me solving a problem.

I also want to thank my family, especially my father and my mother, for their understanding and their encouragement during this time.

9. Index of Figures

10. Index of Tables

11. References

ACHATSCHITZ, P. (2005) Mobile Usability Evaluation within a software project - Workflow Optimization trough a moblie application. 17-27.

ACKERMANN, M. J. (2005) Wireless networking-security. *The Journal Of Medical Practice Management: MPM,* 19**,** 299-300.

ANCONA, M., COSCIA, E., DODERO, G., GIANUZZI, V., MINUTO, F. & VIRTUOSO, S. (2001) Wireless Connections in a Hospital Ward: The Ward-In-Hand Project.

ANZAI, Y., OGAWA, K. & MORI, H. (1995) Symbiosis of Human and Artifact. 375-380.

BABB, D. (2004) User requirements for security in wireless mobile systems. *Information Security Technical Report,* 9**,** 51-59.

BAUMGART, D. C. (2005) Personal digital assistants in health care: experienced clinicians in the palm of your hand?

BRIGHTSIDEFACTORY (2003-2004) Bright Side Framework - Overview. http://www.bs-factory.org/components/remotingDoc/architecture.html

BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK PROJEKTGRUPPE LOCAL WIRELESS COMMUNICATION (2003) Sicherheit im Funk-LAN. http://www.bsi.de/literat/doc/wlan/

CISCO SYSTEMS INC. (2002) Introduction to Secure Sockets Layer. 5-11. http://www.cisco.com/en/US/netsol/ns340/ns394/ns50/ns140/networking_solutions_white_paper09186a0080136858.shtml

CISCO SYSTEMS INC. (2005) Cisco Wireless LAN Security Overview. 10. http://www.cisco.com/en/US/products/hw/wireless/ps430/prod_brochure09186a00801f7d0b.html

DECIVEFORCE.COM (2005) DECT wireless telephony arrives in the US. http://www.deviceforge.com/news/NS9790688989.html

DUMAS, J. S. & REDISH, J. C. (1999) A Practical Guide to Usability Testing. 31.

EGEA-LOPEZ, E., MARTINEZ-SALA, A., VALES-ALONSO, J., GARCIA-HARO, J. & MALGOSA-SANAHUJA, J. (2005) Wireless communications deployment in industry: a review of issues, options and technologies. *Computers in Industry,* 56**,** 29-53.

ELEKTRONIK-KOMPENDIUM (1997-2006) DECT - Digital Enhanced Cordless Telecommunications. http://www.elektronik-kompendium.de/sites/kom/0505231.htm

FREEBEANS (2005) Mysaifu JVM. http://www2s.biglobe.ne.jp/~dat/java/project/jvm/index_en.html

HOLZINGER, A. (2002) Basiswissen IT/Informatik Band 1: Informationstechnik.

HOLZINGER, A. (2003) Experiences with User Centered Development (UCD) for the Front End of the Virtual Medical Campus Graz.

HOLZINGER, A. (2004) Application of Rapid Prototyping to the User Interface Development for a Virtual Medical Campus. *IEEE Software,* 21**,** 95-99.

HOLZINGER, A. (2005) Usability Engineering for Software Developers. *Communications of the ACM,* 48**,** 71-74.

INFORMIT (2005) Wi-Fi Protected Access (WPA). http://www.informit.com/guides/content.asp?g=security&seqNum=84&rl=1

KOREN, D. (2003) PEAP & EAP-TTLS. http://www.cs.huji.ac.il/~sans/

KOWALK, P. D. W. (2002) Rechnernetze. 24. http://einstein.informatik.uni-oldenburg.de/rechnernetze/seite24.htm

MILLER, A. (2004) PDA security concerns. *Network Security,* 2004**,** 8-10.

MUPPARAPU, M. & ARORA, S. (2004) Wireless networking for the dental office: current wireless standards and security protocols. *J Contemp Dent Pract,* 5**,** 155-162. http://www.sciencedirect.com/science/article/B6WVB-4DW36Y8-4KM/2/c8a63fac2918142019678f124a4795f1

MYSAIFU (2005) Mysaifu JVM. http://www2s.biglobe.ne.jp/~dat/java/project/jvm/index_en.html

NIELSEN, J. (1993) Usability Engineering. 23-58, 115-226.

NIELSEN, J. & MACK, R. (1994) Usability Inspection Methods. 63-76.

OYAMA, L., TANNAS, H. S. & MOULTON, S. (2002) Desktop and Mobile Software Development for Surgical Practice. *Journal of Pediatric Surgery,* 37**,** 477-481.

PAVLOVSKI, C., KIM, H. & WOOD, D. (2004) Ubiquitous Mobility in Clinical Healthcare. *IEEE: Proceedings of the IDEAS Workshop on Medical Information Systems: The Digital Hospital (IDEAS-DH'04).*

RUBIN, J. (1994) Handbook of Usability Testing. 240.

SALOMAA, A., ROZENBERG, G. & W., B. (1996) Public-Key Cryptography. 125-153.

SCHNEIDER, B. (1997) Angewandte Kryptographie. 252-260.

SIEMENSAG ÖSTERREICH (2004) SIENET Imaging Workflow Management VA10A. http://www.medical.siemens.com/siemens/en_US/rg_marcom_FBAs/files/brochures/DICOM/rs/Sienet_IWM_DCS_Internet.pdf

SIEMENSAG ÖSTERREICH (2005a) Stichwort: EAP-SIM. http://www.pse.siemens.at/apps/pseauftritt/ge/pseinternet.nsf/CD_Index?OpenFrameset&Bookmark&/0/PK273DB499F20BA28CC1256FA40037D82F

SIEMENSAG ÖSTERREICH (2005b) Stichwort: syngo. http://www.pse.siemens.at/apps/pseauftritt/ge/pseinternet.nsf/0/PK9165260E5ED3BC10C1256C87002A28FA

SIMS, B. (2004) Moving from liability to viability. Hospitals, health plans and physician practices can outsmart hackers with policy, a comprehensive security infrastructure and wireless monitoring. *Health Management Technology,* 25**,** 32-35.

SUN MICROSYSTEMS (2004) Object Serialization Stream Protocol. http://java.sun.com/j2se/1.5.0/docs/guide/serialization/spec/protocol.html

SUN MICROSYSTEMS (2005) CLDC HotSpot™ Implementation Virtual Machine. http://java.sun.com/j2me/docs/pdf/CLDC-HI_whitepaper-February_2005.pdf

WEIPPL, E., HOLZINGER, A. & TJOA, A. M. (2006) Security aspects of ubiquitous computing in health care. *Springer Elektrotechnik & Informationstechnik, e&i,* 123**,** 156-162.

WI-FI ALLIANCE (2005) Wi-Fi Security at Work and on the Road. http://main.wi-fi.org/OpenSection/secure.asp?TID=2

WIKIPEDIA (2004-2006) Transport Layer Security. *Wikipedia*. http://de.wikipedia.org/wiki/Secure_Sockets_Layer

WIKIPEDIA (2005-2006a) Duplex (telecommunications). *Wikipedia*. http://en.wikipedia.org/wiki/Time_division_duplex

WIKIPEDIA (2005-2006b) Wi-Fi Protected Access. *Wikipedia*. http://de.wikipedia.org/wiki/Wi-Fi_Protected_Access

WOBST, D. R. (2003) Know-How Verschlüsselungs-Algorithmen. *c't*, 17**,** 200-206. http://www.heise.de/security/artikel/39275/6/

WÖLFLE, R. D. (2006) Digital Enhanced Cordless Telecommunications (DECT). http://www.ralf-woelfle.de/elektrosmog/redir.htm?http://www.ralf-woelfle.de/elektrosmog/technik/dect_2.htm

ZHANG, Z., BASILI, V. & SHNEIDERMAN, B. (1998) An empirical study of perspective-based usability inspection. 1346-1350.

ZHANG, Z. & VORA, P. (1993) Usability Evalutaion. http://www.usabilityhome.com/