

Martin ULBEL

EKG-Darstellungssoftware zur telemedizinischen Herzschrittmacher-Nachsorge

DIPLOMARBEIT



TECHNISCHE UNIVERSITÄT GRAZ

Institut für Genomik und Bioinformatik
Petersgasse 14, 8010 Graz

Vorstand: Univ.-Prof. Dipl.-Ing. Dr.techn. Zlatko Trajanoski

Betreuer:

Dipl.-Ing. Dr. Dieter Hayn, Austrian Research Centers GmbH

Begutachter:

Univ.-Prof. Dipl.-Ing. Dr.techn. Zlatko Trajanoski

Graz, September, 2008



AUSTRIAN RESEARCH CENTERS

Diese Arbeit wurde bei der

Austrian Research Centers GmbH - ARC
Biomedical Engineering, eHealth systems

Leitung:
Univ.-Doz. Dipl.-Ing. Dr. Günter Schreier, MSc

durchgeführt.

Danksagung

Hiermit möchte ich mich bei allen recht herzlich bedanken, die mich bei der Entstehung dieser Diplomarbeit tatkräftig unterstützt haben. Danken möchte ich auch allen Mitarbeitern der Austrian Research Centers GmbH - ARC für die Unterstützung und gute Zusammenarbeit.

Besonderer Dank gebührt meinen Eltern, die mir dieses Studium und somit auch die Diplomarbeit ermöglicht haben.

Hiermit erkläre ich, die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Graz, 12. September 2008

Unterschrift

Kurzfassung

Internationalen Standards zufolge müssen bei Herzschrittmacherpatienten in regelmäßigen Abständen Nachsorgeuntersuchungen durchgeführt werden, in deren Verlauf die grundlegenden Funktionen des Schrittmachers und der Ladezustand seiner Batterie geprüft werden. In einer am LKH Graz laufenden Studie soll derzeit untersucht werden, ob durch den Einsatz von Telemedizin für die Herzschrittmachernachsorge Krankentransporte reduziert, Spezialambulanzen entlastet und insgesamt Kosten eingespart werden können. Im Zuge der vorliegenden Arbeit wurde ein web-basierter Viewer zur Betrachtung von Elektrokardiogrammen von Herzschrittmacherpatienten entwickelt, mit dessen Hilfe die telemedizinische Nachsorge von Herzschrittmachern optimiert wurde. Spezielles Augenmerk wurde dabei einerseits auf die Darstellung aller für die Nachsorge entscheidenden Parameter und andererseits auf die einfache Benutzbarkeit im Klinikalltag gelegt. Der Viewer wurde in ein bestehendes Studiensystem zur telemedizinischen Herzschrittmachernachsorge integriert und entsprechend den Rückmeldungen der Mediziner optimiert. Die Anwendung hat sich als nützlich für die telemedizinische Herzschrittmachernachsorge erwiesen und ist in einer Studie mit aktuell über 300 Patienten am LKH Graz im Einsatz.

Schlüsselwörter: *Telemedizin, Herzschrittmacher, Nachsorgeuntersuchung, EKG*

Abstract

According to international standards, patients carrying a cardiac pacemaker have to go through periodic follow-up examinations, in which the basic functions of the pacemaker as well as the discharge level of its battery are tested. In an ongoing study at the LKH Graz it is currently investigated, whether - by the use of telemedicine for pacemaker follow-ups - patient transportations can be reduced, specialized pacemaker ambulances can be disburdened, and overall costs can be reduced. In the course of the present thesis, a web-based viewer for displaying electrocardiograms of pacemaker patients has been developed, which helps optimizing the telemedical aftertreatment of pacemaker patients. A special focus was put on the visualization of all parameters needed for telemedical pacemaker follow-ups, as well as on the simple usability in the daily clinical routine. The viewer was integrated into an existing research network for telemedical pacemaker follow-ups and optimized according to the feedback of the medical personnel. The application has shown to be useful for telemedical pacemaker follow-ups and is currently utilized in a study with 300 patients at the LKH Graz.

Keywords: *Telemedicine, Pacemaker, Follow-up, ECG*

Inhaltsverzeichnis

1	Einleitung	1
1.1	Herzschrittmacher allgemein	1
1.1.1	Herzschrittmacher-Typen	2
1.2	Zahl der Herzschrittmacherpatienten	4
1.3	Herzschrittmacher Nachsorgeuntersuchung	5
1.4	Basisnachsorgeuntersuchung	5
1.4.1	Magneteffekt	6
1.5	Das Elektrokardiogramm (EKG)	7
1.5.1	Erregungsleitungssystem	7
1.5.2	Entstehung des Signals	8
1.5.3	Ableitungen	10
1.5.4	Abtastrate	12
1.6	H.ELGA	12
1.6.1	Extramularer Bereich	13
1.6.2	Telemedizinische Servicezentrale	15
1.6.3	Kardiologie	15
1.6.4	Technologie	15
1.6.5	Aufgabenstellung	16
1.6.6	Recherche	17
2	Methoden	19
2.1	Schnittstelle Matlab-Python	19
2.1.1	Python	19
2.1.2	Matlab	20
2.1.3	PyMat	21
2.2	Dateiformate	22
2.2.1	Dateiformat SCP(Standard Communication Protocol)	22
2.2.2	Dateiformat XML FDA	22
2.2.3	Dateiformat MAT	23
2.2.4	Dateiformat WAV	23
2.2.5	Dateiformat GDF	23

2.2.6	Dateiformat EDF	24
2.3	Anforderungen an das Dateiformat	24
2.4	Entwicklungsumgebung	25
2.4.1	Java	25
2.4.2	Applet	28
2.4.3	JFreeChart	31
2.4.4	andere Chart Packages	33
2.4.5	Eclipse	33
2.5	Schnittstelle Zope	35
2.5.1	Datenspeicherung	35
2.5.2	File laden	35
3	Ergebnisse	37
3.1	Lösungsansätze	37
3.2	EKG-Viewer	37
3.2.1	Dateiformat	37
3.2.2	Dateigröße	40
3.2.3	Viewer	40
3.2.4	Voraussetzungen	43
3.2.5	allgemeiner Programmaufbau	43
3.2.6	EKG-Viewer für mehrere Kanäle	45
3.2.7	EKG Darstellungszeiten	47
3.2.8	EKG-Viewer für Langzeit EKGs	47
3.2.9	Raster	49
3.3	Geschwindigkeitstest	50
3.4	Einbindung in H.ELGA	51
3.5	Sicherheit	51
3.6	Anforderungen	52
4	Diskussion	54
4.1	Viewer	54
4.2	Dateiformat	55
4.3	Einsatzmöglichkeiten des EKG-Viewers	56
4.4	Ausblick	56
4.5	Fazit	57
	Literaturverzeichnis	58

Abbildungsverzeichnis

1.1	Herzschriftmacher der Firma Vitatron	2
1.2	EKG mit Magnetfrequenz	7
1.3	Erregungsleitungssystem des Herzens	9
1.4	EKG mit Bezeichnungen	9
1.5	Übersicht H.ELGA	14
1.6	Technische Übersicht der H.ELGA	16
1.7	EKG Bild in der „alten“ H.ELGA	17
2.1	Übersicht des Sicherheitskonzeptes bei Java	27
2.2	Applets in der Java Sicherheitsumgebung	32
3.1	Beispiel eines EKGs in der H.ELGA	41
3.2	Klassendiagramm	46
3.3	Viewer für Mehrkanal EKGs	47
3.4	Vergleich des Rasters	49
3.5	Workflow des Viewers in der H.ELGA	51
3.6	H.ELGA und EKG Viewer	53

Tabellenverzeichnis

1.1	HSM Statistik Deutschland	4
1.2	HSM Statistik 2005	4
1.3	EKG Ableitungen	11
1.4	Übersicht von Eigenschaften derzeit erhältlicher open-source Viewern und den Anforderungen für den Viewer für die H.ELGA	18
3.1	Dateigröße	40
3.2	Zeiten der EKG Darstellung von Dateien mit verschiedener Größe	48
3.3	Zeiten der EKG Darstellung für Zweikanal EKGs mit verschiedener Größe	48
3.4	Ladezeiten lokal gespeicherter Dateien	48
3.5	Größe der Packages	50
3.6	Darstellungszeiten mit originalen und verkleinerten Packages .	50
3.7	Vergleich bestehende EKG Viewer mit dem programmierten Viewer; 1) Einfache Möglichkeit zur Erweiterung auf beliebige andere Formate	52

Abkürzungsverzeichnis

EDF	European Data Format
EKG	Elektrokardiogramm
EU	Europäische Union
FTP	File Transfer Protocol
GDF	Dateiformat für Biosignale
GUI	grafische Benutzeroberfläche
H.ELGA	Herzschrittmacher . Elektronische Gesundheitsakte
HL7	Health Level 7, internationaler Standard
HSM	Herzschrittmacher
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICD	implantierbarer Defibrillator
JAR	Dateiformat von Java
JRE	Java Runtime Environment
JVM	Java Virtual Machine
MAT	Dateiformat von Matlab
PDA	Personal Digital Assistent
PDF	Portable Document Format
SCP	Standard Communication Protocol, Dateiformat
SSL	Secure Sockets Layer
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
USB	Universal Serial Bus
WAV	Dateiformat
XML	Extensible Markup Language
ZODB	Zope Objekt Database

Kapitel 1

Einleitung

Das Zeitalter der Telemedizin hat schon vor Jahren begonnen. Unter Telemedizin versteht man die Überbrückung der Distanz zwischen Arzt und Patient oder Arzt und Arzt mittels Telekommunikation zum Zwecke der Durchführung einer medizinischen Handlung. Der Bereich der Telemedizin umfasst ein sehr großes Gebiet, von Telemonitoring über Teledermatologie bis zur Telechirurgie. Sie bringt vor allem für den Patienten zahlreiche Vorteile: Er kann vom Arzt ständig überwacht werden, erspart sich lange Anfahrtswege und Wartezeiten beim Spezialisten und muss nicht lange auf einen Termin oder seinen Befund warten. Für die Ärzte hingegen ist es eine neue Herausforderung, ohne den Patienten zu sehen - alleine auf Basis von Messwerten - einen Befund zu erstellen. Problematisch ist derzeit allerdings noch die offene Frage der Bezahlung von telemedizinischen Behandlungen, denn diese Leistungen können in Österreich noch nicht mit der Krankenkasse verrechnet werden [27]. Eine mögliche Anwendung der Telemedizin, die Inhalt der vorliegenden Arbeit ist, liegt in der telemedizinischen Herzschrittmachernachsorge. Auf die Grundlagen dieser Nachsorgen wird im Folgenden näher eingegangen.

1.1 Herzschrittmacher allgemein

Es wird grundsätzlich zwischen Herzschrittmacher (HSM) und implantierbarem Defibrillator (ICD) unterschieden. Ein HSM greift bei Herzrhythmusstö-

rungen ein und versucht das Herz wieder in einen normalen Rhythmus zu bringen. Ein ICD gibt zusätzlich bei Herzkammerflimmern einen Schock von 10-40 J ab und versucht damit das Herz wieder in den Takt zu bringen. Bei allen neuen und modernen HSM wird die Herzaktivität überwacht. Falls keine Eigenaktivität erkannt wird, wird vom Schrittmacher ein elektrischer Impuls gesendet, damit der Herzmuskel stimuliert wird. Ein Schrittmacher, wie in Abbildung 1.1 abgebildet, besteht aus einem miniaturisierten elektronischen Schaltkreis und einer Batterie. Die Verbindung zum Herzen übernehmen ein bis drei Elektroden. Diese Elektroden werden im Vorhof oder in der Kammer des Herzens platziert und dienen der Stimulation durch den Schrittmacher (*Pacing*) und zur Messung der elektrischen Aktivität (*Sensing*). Beim Einkammer-Schrittmacher liegt die Elektrode im rechten Vorhof oder in der rechten Kammer, je nach Indikation. Der Zweikammer-Schrittmacher hat zwei Elektroden, eine befindet sich im rechten Vorhof die andere in der rechten Kammer.

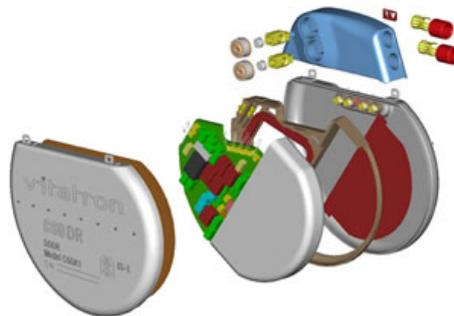


Abbildung 1.1: Herzschrittmacher der Firma Vitatron (Vitatron, Arnheim, Niederlande) entnommen von www.vitatron.de

1.1.1 Herzschrittmacher-Typen

Mittlerweile gibt es über 20 verschiedene Hersteller für Herzschrittmacher, diese folgen alle dem seit 1988 und 2002 revidierten NBG-Code [4] (NASPE/BPEG Generic Pacemaker Code).

NASPE: North American Society of Pacing and Electrophysiology

BPEG: British Pacing and Electrophysiology Group

Der NBG-Code besteht aus maximal fünf Buchstaben, wobei meistens aber nur die ersten drei verwendet werden und dient der Funktionsbeschreibung des HSM.

1. Buchstabe, beschreibt den Stimulationsort:

- 0...keiner
- A...Atrium
- V...Ventrikel
- D...Dual (Atrium und Ventrikel)
- S...Single (Atrium/Ventrikel)

2. Buchstabe, beschreibt den Wahrnehmungsort:

- 0...keiner
- A...Atrium
- V...Ventrikel
- D...Dual (Atrium und Ventrikel)
- S...Single (Atrium/Ventrikel)

3. Buchstabe, beschreibt die Betriebsart:

- 0...keine. Bei dieser Art wird weder der Modus getriggert, noch der Modus inhibiert unterstützt.
- T...getriggert. In diesem Modus führt ein wahrgenommenes Signal zur Impulsabgabe.
- I...inhibiert. In diesem Modus kann der Schrittmacher die Abgabe eines Impulses bei eigener Herzaktivität unterdrücken.
- D...Dual (getriggert und inhibiert). Bei dieser Art werden beide Modi unterstützt.

4. Buchstabe, beschreibt die Frequenzadaption:

- 0...keine
- R...frequenzmoduliert. Mit dieser Funktion ist eine Anpassung des Schrittmachers an ein belastungsinduziertes Signal möglich.

5. Buchstabe, beschreibt die multifokale Stimulation:

- 0...keine. Es erfolgt keine Multisite Stimulation.

A...Atrium. Die Stimulation erfolgt an mehr als einer Stelle im Atrium.

V...Ventrikel. Die Stimulation erfolgt an mehr als einer Stelle im rechten Ventrikel oder in beiden Ventrikeln.

D...Dual (Atrium und Ventrikel). Es erfolgt eine Multisite Stimulation in Atrium und Ventrikel.

1.2 Zahl der Herzschrittmacherpatienten

Die Zahl der Patienten mit einem HSM ist in den letzten Jahren steigend. Dies beweist z.B. auch die Statistik [8] aus Deutschland über einen Zeitraum von drei Jahren, zu sehen in Tabelle 1.1.

	Neuimplantationen	Austausch
2003	51.904	12.484
2004	62.382	14.622
2005	65.447	17.040

Tabelle 1.1: HSM Statistik Deutschland

Die folgende Statistik in Tabelle 1.2 zeigt die Zahl der Neuimplantationen, Patientenanzahl deren Schrittmacher getauscht wurde und die Anzahl an Patienten mit einem Herzschrittmacher im Jahr 2005.

	Neuimplantationen	Austausch	HSM-Patienten
Österreich	5.209	1.691	50.000
Deutschland	65.447	17.040	230.000
Schweiz	3.382	1.248	22.330

Tabelle 1.2: HSM Statistik 2005 [25] [2] [8]

In Österreich gibt es derzeit ungefähr 50.000 Patienten mit einem implantierten HSM und die Zahl wird immer größer. Gründe für die steigende Patientenzahl sind unter anderem die steigende Lebenserwartung [3] und dass auch noch im hohen Alter Schrittmacher implantiert werden.

1.3 Herzschrittmacher Nachsorgeuntersuchung

Bei der HSM Nachsorgeuntersuchung wird sowohl der technische Zustand des Schrittmachers als auch der Zustand des Patienten beurteilt. Die Nachsorgeuntersuchungen werden nach den Richtlinien [1] in vier verschiedene Typen eingeteilt:

Typ 1: postoperativ

Die erste Nachsorgeuntersuchung sollte innerhalb der ersten 72 Stunden nach der Operation stattfinden. Bei dieser Untersuchung werden folgende Parameter untersucht: Sensing, Pacing, Thoraxbild, Wundkontrolle.

Typ 2: innerhalb der ersten 6 Monate nach der Operation

In den ersten 6 Monaten danach sollen ein bis zwei Untersuchungen stattfinden, bei denen eine Anpassung der Programmierung an die Patientenbedürfnisse und an die Empfindlichkeits- und Reizschwellen durchgeführt wird.

Typ 3: Zeit nach den ersten 6 Monaten

Danach wird in der Regel die Nachsorgeuntersuchung alle 6 bis 12 Monate durchgeführt. Falls Probleme mit dem HSM auftreten, der Patient Schmerzen oder andere Beschwerden hat, ist eine Kontrolle in kurzen Abständen notwendig.

Typ 4: intensive Kontrollen bei bevorstehender Batterieerschöpfung

In dieser Phase muss der Zeitrahmen der Nachsorgeuntersuchung verkürzt werden, damit der Zeitpunkt der Batterieerschöpfung und ein Ausfall des HSM nicht übersehen werden.

1.4 Basisnachsorgeuntersuchung

Die Basisnachsorgeuntersuchung [1] ist jene, die dem Typ drei entspricht. Dabei wird der Patient nach seinem Gesundheitszustand und eventuellen

Beschwerden befragt. Die Untersuchung des HSM findet mit einem herstellereigenen Programmierer statt. Mit Hilfe dieses Programmierers können verschiedene Parameter des HSM kontrolliert und gegebenenfalls verändert werden. Weiters wird eine kurze EKG-Sequenz aufgezeichnet, bei der der Magnet-Effekt auftritt. Hierbei wird dem Patienten ein Magnet auf den Schrittmacher gelegt und der HSM, herstellerabhängig, schaltet in den Magnet-Mode. Die nun aufgezeichnete Frequenz wird auch Magnetstimulationsfrequenz genannt und mit ihrer Hilfe kann auf den Batterieladezustand geschlossen werden. Es werden folgende drei Batterieladezustände unterschieden:

BOL (Begin of Life): Dies ist der Zustand von Beginn einer Implantation bis zum Zustand ERI. Am Anfang ist die Batteriekapazität voll vorhanden. Je nach Programmierung des Schrittmachers und Aktivität verringert sich mit der Zeit die Kapazität der Batterie.

ERI (Elective Replacement Indication): Wenn diese Schwelle erreicht ist, garantiert der Hersteller noch die volle Funktionsfähigkeit des HSM für eine herstellerabhängige Zeitspanne. Es werden aber gewisse Sonderfunktionen deaktiviert um Energie zu sparen. Ab diesem Zeitpunkt entscheidet der Kardiologe, ob die Zeitspanne für die Nachsorgeuntersuchung verringert oder der HSM ausgetauscht wird. Meistens wird der HSM gleich ausgetauscht um für eine maximale Sicherheit des Patienten zu garantieren.

EOL (End of Life): Bei Erreichen dieser Schwelle muss der HSM spätestens ausgetauscht werden. Der HSM wird noch mindestens drei Monate in einem Energiesparmodus betrieben.

1.4.1 Magneteffekt

Im Rahmen einer HSM Nachsorgeuntersuchung wird ein EKG bei temporären Magnetauflage, dem so genannten „*Magnettest*“, aufgezeichnet [23]. Wird auf den Schrittmacher ein Magnet gelegt, schaltet dieser automatisch in den Magnet-Mode. Jeder Schrittmacher verfügt über eine bestimmte Magnetstimulationsfrequenz, die auch als Magnetfrequenz bezeichnet wird. Diese ist

vom Schrittmachertyp und dem Batteriezustand abhängig. Aus dem aufgezeichneten EKG kann diese Magnetfrequenz herausgelesen werden und damit der Ladezustand der Batterie des Herzschrittmachers beurteilt werden.

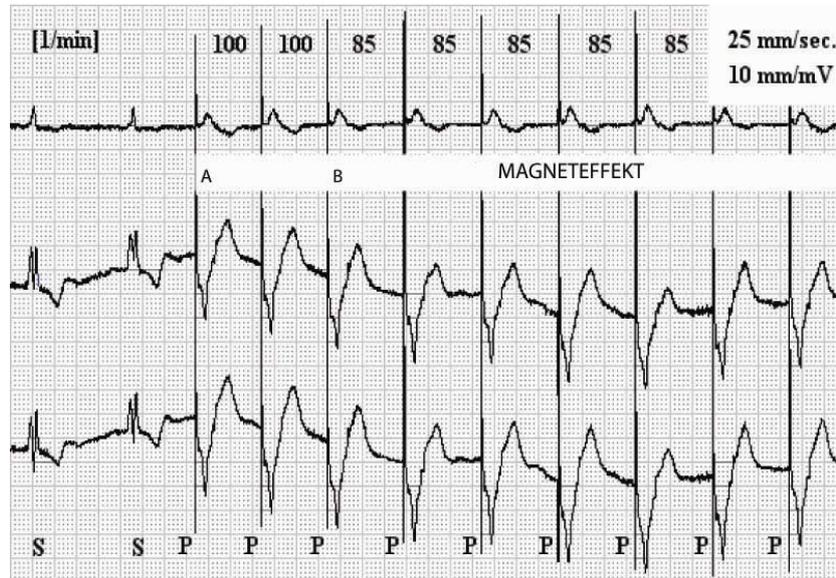


Abbildung 1.2: EKG mit Magnetfrequenz

In Abbildung 1.2 ist ein EKG während einer Magnetauflage ersichtlich. Dabei handelt es sich um einen Herzschrittmacher der Firma Medtronic (Modell „Kappa“); (Medtronic Inc., Minneapolis, USA). Ab dem Punkt A ist der Magneteffekt erkennbar, es erfolgen drei Stimulationen mit einer Frequenz von 100 min^{-1} und danach Stimulationen mit einer Frequenz von 85 min^{-1} . Diese Stimationsfrequenzen sind direkt vom Ladezustand der Batterie abhängig, in diesem Fall hat die Batterie den Zustand BOL.

1.5 Das Elektrokardiogramm (EKG)

1.5.1 Erregungsleitungssystem

Das EKG-Signal ist eines der wichtigsten und am häufigsten gemessenen Biosignale. Die Erregung des Herzens entsteht im Sinusknoten [24], dieser wird

auch als *primärer Schrittmacher* bezeichnet. Der Sinusknoten erzeugt ohne externe Einflüsse einen Sinusrhythmus von 60-80 Schlägen pro Minute. Die Erregung läuft während der späten Füllungsphase über den rechten Vorhof in Richtung Herzspitze, diese aktiviert die Muskulatur beider Vorhöfe und bewirkt eine aktive Ventrikelfüllung kurz vor der neuen Systole. Der AV-Knoten wirkt wie ein Zeitzünder, der die Erregungsausbreitung um ca. 40 ms (20-60ms) verzögert. Diese Verzögerung dient dazu, der Vorhofmuskulatur Zeit zur aktiven Füllung der Ventrikel zu lassen. Fällt die Erregung des AV-Knoten durch den Sinusknoten aus, kann dieser die Erregung des Herzens übernehmen. Er arbeitet dann als *sekundärer Schrittmacher* mit einer Eigenfrequenz von ca. 40 Schlägen pro Minute. Nach einem kurzen gemeinsamen Verlauf im Kammerseptum, His-Bündel genannt, trennen sich die Fasern des Erregungsleitungssystems in einen rechten und zwei linke Tawara-Schenkel. Diese verlaufen bis zur Herzspitze und gehen dort in die Purkinje-Fasern über, die das Kammermyokard erregen. Bei Ausfall des Sinusknoten und AV-Knoten können diese Fasern die Erregung des Herzens mit einer Frequenz von 30 Schlägen pro Minute übernehmen und werden deshalb auch *tertiärer Schrittmacher* genannt.

In Abbildung 1.3 sind die Positionen der „Schrittmacher“ ersichtlich. Wenn es nun zu einer Störung des Reizleitungssystems kommt sind verschiedene Therapiemöglichkeiten gegeben, darunter fallen eine medikamentöse Therapie oder die Behebung des Problems mit einem HSM.

1.5.2 Entstehung des Signals

Das Elektrokardiogramm ist eine Messung der elektrischen Aktivitäten der Herzmuskelfasern. In Abbildung 1.4 ist ein EKG-Signal dargestellt. Es werden nur von der Arbeitsmuskulatur messbare Ströme erzeugt, die Masse des Erregungsleitungssystems ist dafür zu klein. Ein EKG Ausschlag kommt zustande, wenn sich eine elektrische Erregung ausbreitet oder wieder zurückbildet. Mit Hilfe eines EKGs können unter anderem eine Rhythmus Diagnose, verschiedene Zeit- und Amplitudenparameter und der Lagetyp festgestellt werden. Häufige Parameter sind z.B. Herzrate, PR-Intervall, QRS-Dauer oder

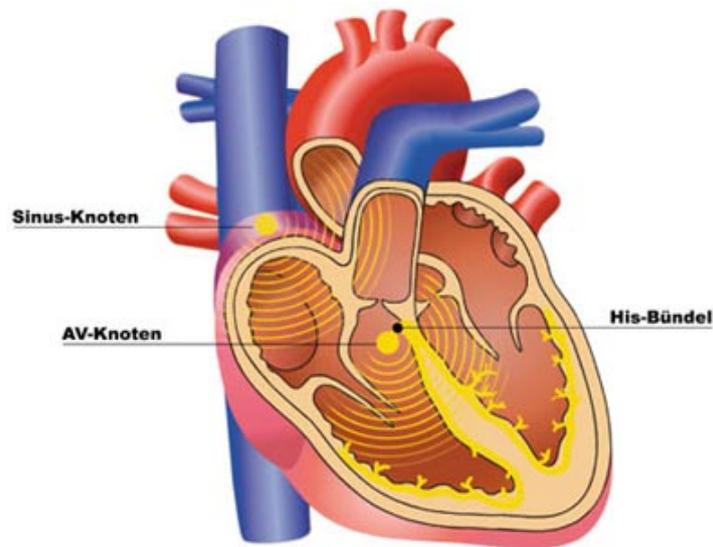


Abbildung 1.3: Erregungsleitungssystem des Herzens, aus www.kardionet.de

QT-Intervall. Das EKG ist ein wichtiges Hilfsmittel zur Beurteilung des Reizleitungssystems und Herzerkrankungen.

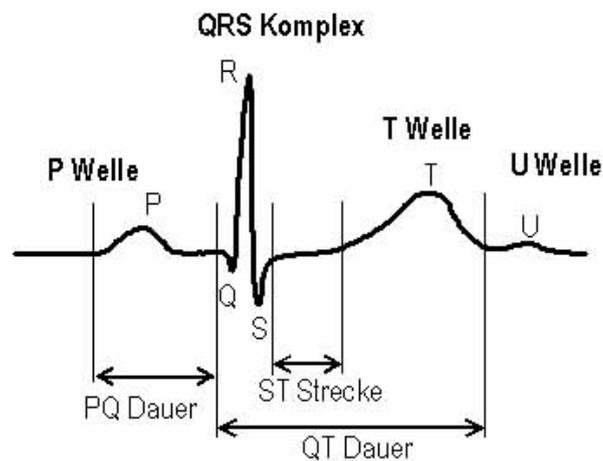


Abbildung 1.4: EKG mit Bezeichnungen

Im folgenden werde die Grundmerkmale eines EKG-Signals erläutert:

P-Welle: Erregung der Vorhofmuskulatur. Der Beginn der P-Welle kenn-

zeichnet den Zeitpunkt, an dem der Sinusknoten seine Aktionspotentiale erzeugt.

PQ-Strecke: Vollständige Erregung der Vorhofmuskulatur. Die Erregung hat am Ende der P-Welle den AV Knoten erreicht. Die Länge der PQ-Strecke ist ein Maß für die Verzögerung im AV-Knoten.

QRS-Komplex: Erregung der Kammermuskulatur. Die Q-Zacke ist der erste negative Ausschlag, die R-Zacke ist immer positiv und die S-Zacke wieder negativ. Gleichzeitig bildet sich die Vorhoferregung zurück, dieser Vorgang geht aber im QRS-Komplex unter.

ST-Strecke: Vollständige Erregung der Kammermuskulatur, die Länge hängt stark von der Herzfrequenz und von Einflüssen des vegetativen Nervensystems ab.

T-Welle: Erregungsrückbildung der Kammermuskulatur. Da diese nicht an allen Stellen gleichzeitig erfolgt, ist sie breiter und niedriger als der QRS-Komplex.

1.5.3 Ableitungen

Es gibt verschiedene Ableitungsarten, grundsätzlich wird zwischen den bipolaren und unipolaren Ableitungen unterschieden. Bei den bipolaren Ableitungen wird die Spannung zwischen zwei Elektroden an der Körperoberfläche gemessen. Bei den unipolaren Ableitungen wird die Spannung zwischen einer differentiellen und indifferenten (nahezu potentialkonstanten) Elektrode gemessen. Die drei bekanntesten Ableitungssysteme sind Einthoven, Goldberger und Wilson.

Einthoven: I, II, III, bipolare Ableitungen mit drei Elektroden und einer Neutralelektrode

Goldberger: aVL, aVR, aVF, Extremitätenableitungen, unipolare Ableitungen mit drei Elektroden und einer Neutralelektrode

Wilson: V1, V2, V3, V4, V5, V6, Brustwandableitungen, unipolare Ableitungen mit neun Elektroden und einer Neutralelektrode

Tabelle 1.3 gibt eine Übersicht der Punkte, wo die Elektroden aufgeklebt werden. Zur genaueren Untersuchung des Herzens gibt es noch andere Ableitungsschemas mit mehreren (z.B. 60) Ableitungen. Hierbei bekommt der Patient z.B. eine Weste mit mehreren Elektroden angezogen, dies wird derzeit aber nur für wissenschaftliche Zwecke eingesetzt.

Bezeichnung	Minuspole	Pluspole
I	rechter Arm	linker Arm
II	rechter Arm	linkes Bein
III	linker Arm	linkes Bein
aVL	rechter Arm und linkes Bein	linker Arm
aVR	linker Arm und linkes Bein	rechter Arm
aVF	linker Arm und rechter Arm	linkes Bein
V1	li. Arm + re. Arm + li. Bein	4. ICR rechts parasternal
V2	wie bei V1	4. ICR links parasternal
V3	wie bei V1	zwischen V2 und V4
V4	wie bei V1	5.ICR Medioclaviculärlinie
V5	wie bei V1	5.ICR vordere Axillarlinie
V6	wie bei V1	5.ICR mittlere Axillarlinie

Tabelle 1.3: EKG Ableitungen; ICR ist der Intercostalraum, darunter versteht man den Raum zwischen zwei benachbarten Rippen.

EKG-Signale werden nach Aufzeichnungsdauer und Art unterschieden. Das Ruhe-EKG wird beim liegenden Patienten durchgeführt, es werden meistens die 12 Standard Kanäle aufgezeichnet und dauert nur wenige Sekunden. Dies wird z.B. bei internistischen Untersuchungen, beim Hausarzt, im Krankenhaus oder in der Notfallmedizin verwendet. Das Langzeit-EKG wird über mehrere Stunden aufgezeichnet und kommt zur Anwendung, wenn der Patient Beschwerden angibt, die sehr unregelmäßig oder unter bestimmten Zuständen vorkommen. Das Belastungs-EKG wird unter körperlicher Belastung aufgezeichnet, meistens wird dafür ein Fahrrad-Ergometer verwendet.

1.5.4 Abtastrate

Unter dem Begriff Abtastrate, auch Abtastfrequenz genannt, versteht man die Häufigkeit, mit der ein Signal pro Zeitintervall abgetastet wird. Der Abstand zwischen zwei Abtastzeitpunkten ist das Abtastintervall. Die Abtastrate soll bei einem Signal konstant sein, um eine einfache Weiterverarbeitung des Signals möglich zu machen. Um Artefakte zu vermeiden, darf die Abtastrate nicht zu klein werden. Wird sie zu hoch gewählt, wird nur zusätzlich Speicherplatz verbraucht, aber es gibt keine qualitative Verbesserung. Nach dem Nyquist-Abtasttheorem [18] $f_{max} < f_a/2$ muss die Abtastfrequenz größer als die doppelte maximale Frequenz sein, um keinen Informationsverlust zu erhalten. In der Formel steht f_{max} für die maximale Frequenz die im Signal auftritt und f_a für die Abtastfrequenz.

1.6 H.ELGA

Das Projekt Herzschrittmacher . Elektronische Gesundheitsakte (H.ELGA) entstand im Jahre 2002 im Rahmen einer Diplomarbeit [11] bei der Austrian Research Centers GmbH - ARC. Um die Schrittmacherambulanz und den Patienten zu entlasten wurde dieses Projekt gestartet und ist derzeit in der zweiten Studie im Einsatz. Die ersten klinischen Untersuchungen begannen im Jahre 2002. Eine klinische Pilotstudie [12] wurde im Jahre 2005 erfolgreich abgeschlossen. An dieser ersten Studie waren 24 Patienten mit 17 verschiedenen HSM Modellen beteiligt.

Laut internationalen Richtlinien [1,9,15] muss bei Herzschrittmacherpatienten eine regelmäßige Nachsorgeuntersuchung durchgeführt werden. Die Basisnachsorgeuntersuchung findet im Rahmen der H.ELGA aber nicht mehr in einer Schrittmacherambulanz statt, sondern als telemedizinische Untersuchung. Diese kann in einem beliebigen Krankenhaus oder vom Hausarzt mit der notwendigen Ausrüstung (EKG Aufzeichnungsgerät, PDA) durchgeführt werden.

Von diesem Projekt profitieren Patienten und Schrittmacherambulanz.

Vorteile für Patienten:

- Untersuchung wird vom Hausarzt durchgeführt
- keine weiten Fahrtstrecken in die Ambulanz
- kein Warten in einer Schrittmacherambulanz

Vorteile für die Ambulanz:

- weniger Patienten zur Untersuchung
- keine Patienten der Basisnachsorgeuntersuchung
- Kostenersparnis wegen einer geringeren Anzahl an Untersuchungen

Abbildung 1.5 stellt einen Überblick der telemedizinischen Nachsorgeuntersuchung des Projektes H.ELGA dar.

1.6.1 Extramularer Bereich

Im extramularen Bereich geht der Patient zum Hausarzt oder in ein Krankenhaus in seiner Umgebung, dort wird der allgemeine Gesundheitszustand erfragt. Weiters werden Blutdruck, Puls und ein EKG aufgezeichnet. Das EKG wird mittels mobilem, PDA-basiertem EKG Aufzeichnungsgerät während einer temporären Magnetauflage aufgezeichnet [12]. Derzeit wird ein Standard PDA (Modell iPAQ 3970, Hewlett-Packard Development Company, L.P., Wilmington, USA) mit einer seriellen Schnittstelle verwendet. An dieser seriellen Schnittstelle ist das mobile EKG Aufzeichnungsgerät (g.MobiLab, g.tec Guger Technologies OEG, Graz, Österreich) angeschlossen, über Bluetooth und ein UMTS-fähiges Mobiltelefon wird das EKG an die telemedizinische Servicezentrale gesendet. Das g.MobiLab System ist in der Lage, ein Zwei-Kanal-EKG mit einer Abtastrate von 1024 Hz aufzuzeichnen.

Am PDA läuft eine auf Java basierende Software, die den Arzt bei der Aufnahme der Patientendaten, der Aufzeichnung des EKG und der Datenübertragung unterstützt. Beim Start werden von der Software alle notwendigen Verbindungen und Systemeinstellungen initialisiert, danach wird der Arzt Schritt für Schritt durch die Nachsorgeuntersuchung geleitet. Nun werden

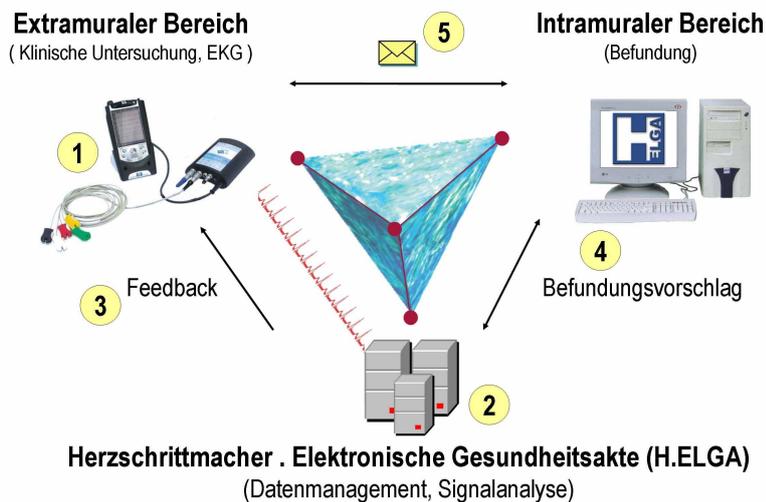


Abbildung 1.5: Übersicht H.ELGA; 1. Aufzeichnung des EKGs mittels PDA z.B. beim Hausarzt; 2. Monitoringzentrale, Analyse des EKG; 3. Feedback an den Hausarzt, ob das EKG erfolgreich gesendet wurde; 4. Befundvorschlag wird an den Kardiologen übermittelt; 5. der endgültige Befund wird dem Hausarzt übermittelt

vom Arzt die Anamnesedaten erhoben und ein EKG während einer temporären Magnetauflage aufgezeichnet. Alle Daten und das EKG werden zuerst lokal am PDA gespeichert. Danach wird das EKG-File mittels FTP und die Daten mittels HTTP an die telemedizinische Servicezentrale übermittelt. Das Mobiltelefon, welches mittels Bluetooth mit dem PDA gekoppelt ist, dient dabei der Übertragung mittels UMTS.

Dieses Szenario wurde in der erwähnten Pilotstudie [12] erfolgreich angewandt. Statt der Bluetooth-Kopplung und Übertragung via UMTS, kann der PDA mittels USB an einen Computer in der Schrittmacherambulanz angeschlossen werden. Danach wird das aufgezeichnete EKG Signal an die Telemonitoring Zentrale übermittelt. Diese Methode wird in der derzeitigen

Studie [21] verwendet.

1.6.2 Telemedizinische Servicezentrale

Wurde ein EKG erfolgreich an die telemedizinische Servicezentrale übertragen, beginnt automatisch die Signalanalyse. Diese wird mit dem Programm MatLab¹ (The Math Works Inc., MA, USA) durchgeführt. Im ersten Schritt werden die Stimulationsspiques detektiert und daraus das Stimulationsmuster kodiert. Danach wird das Stimulationsmuster mit den Mustern in der Datenbank verglichen, um den Zustand der Batterie (BOL, ERI und EOL) festzustellen. Der Arzt bekommt per SMS das Ergebnis der Signalanalyse, bei schlechter Signalqualität oder keinem erkannten Magneteffekt wird er aufgefordert, die Aufzeichnung zu wiederholen. Ein Vorbefund wird in der H.ELGA abgelegt und steht dem Kardiologen zur Verfügung.

1.6.3 Kardiologe

Der Kardiologe kann über das WEB-Portal ersichtlich, alle Daten und früheren Untersuchungen und Befunde des Patienten einsehen. Mit Hilfe des vorherigen Befunde, der Krankengeschichte des Patienten, des aktuellen EKGs und der aktuellen Anamnesedaten wird der endgültige Befund des Kardiologen erstellt. Danach wird ein neuer Nachsorgetermin festgelegt und der Befund generiert. Dieser wird in der H.ELGA abgelegt und dem Hausarzt zugesandt. Bei einem unsicheren Befund, erkannten Problemen oder einer Batterie-Erschöpfung kann der Kardiologe den Patienten in die HSM-Ambulanz bestellen.

1.6.4 Technologie

Übersicht der technischen Realisierung der H.ELGA:

1. Applicationserver. Der Applicationserver basiert auf Zope 2.7.4² (Zope Corporation, Fredericksburg, USA). Zope ist ein objektorientiertes open

¹www.mathworks.com

²www.zope.org

source Produkt welches in der Programmiersprache Python³ (Python Software Foundation, Hampton, USA) implementiert ist.

2. Datenbankserver. Die Funktion des Datenbankservers ist das Speichern und Verwalten der Daten. Hierfür wird PostgreSQL⁴ (PostgreSQL Global Development Group, California, USA) verwendet. Dies ist ein objektrelationales Datenbankmanagementsystem.
3. Webserver. Als Webserver wird ein Apache⁵ Server (Apache Software Foundation, Delaware, USA) verwendet.
4. Fileserver. Der Fileserver speichert Dateien und Dokumente außerhalb der Datenbank.

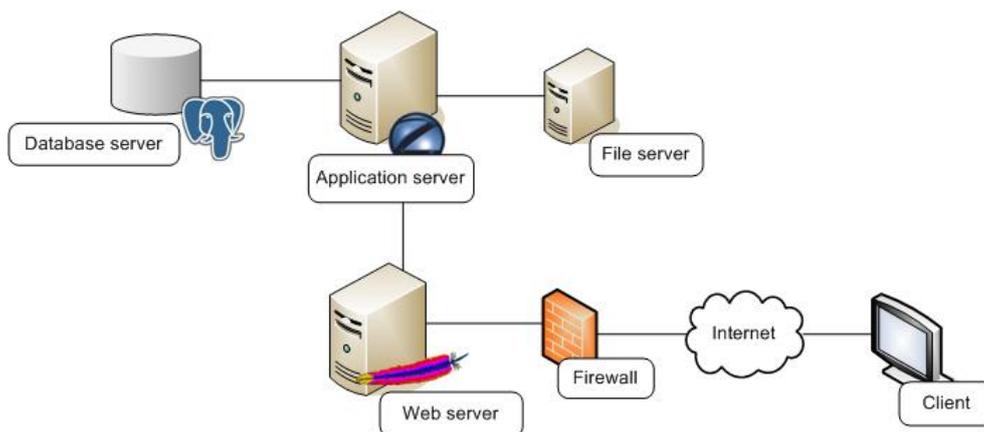


Abbildung 1.6: Technische Übersicht der H.ELGA

1.6.5 Aufgabenstellung

Vor Beginn der Diplomarbeit gab es in der H.ELGA nur die Möglichkeit, das EKG als statisches Bild anzuzeigen. In dieser Darstellung (siehe Abbildung 1.7) war nur das EKG von kurz vor bis kurz nach dem Magneteffekt

³www.python.org

⁴www.postgresql.org

⁵<http://httpd.apache.org/>

dargestellt. Die Darstellung des gesamten EKGs und verschiedene Marker war nicht möglich. Als Erweiterung für die H.ELGA sollte ein EKG Viewer entwickelt werden, um den Kardiologen das Review zu erleichtern. Zur Datenübertragung von der Monitoringzentrale zum Viewer sollte ein geeignetes Datenformat gefunden und eine Schnittstelle implementiert werden. Mit dem Viewer sollte das EKG Signal und verschiedene Marker dargestellt werden können. Weiters sollte Zoomen und Scrollen innerhalb des Signals möglich sein.

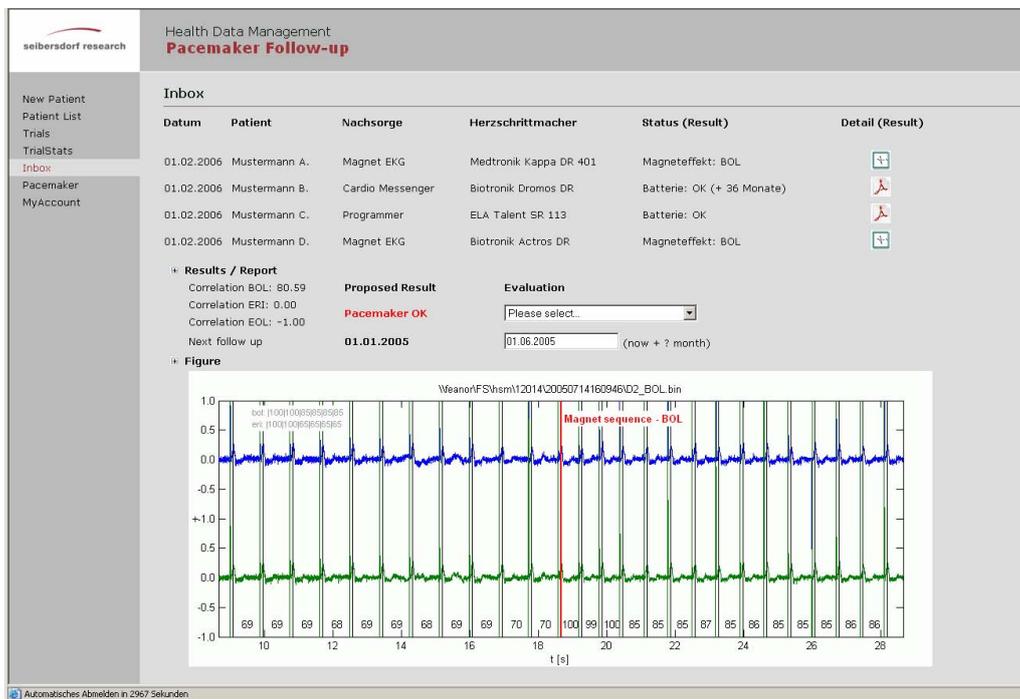


Abbildung 1.7: EKG Bild in der „alten“ H.ELGA

1.6.6 Recherche

Im ersten Schritt wurde eine Internet-Recherche nach bestehenden EKG-Viewern durchgeführt. Es konnten zahlreiche Programme zur Betrachtung von EKGs gefunden werden, die meisten davon waren allerdings Demoversionen. Der andere Teil war kommerziell erhältlich und in seiner Funktionalität

nicht veränderbar.

Am ehesten schienen noch die folgenden Viewer für die H.ELGA zu passen: Ein frei erhältlicher Viewer mit sehr guter Dokumentation ist der Viewer „Wave“ von Physionet.org [16]. Dieser besitzt die Funktion scrollen, kann aber nur Signale aus der Physionet Datenbank darstellen. Alternativ dazu gibt es den Viewer „Pixelmed“ von openecg.net [7]. Dieser ist in Java programmiert, scrollen ist nicht möglich aber es können die Anfangs- und Endwerte der Achsen verändert werden, was einem Zoomen nahe kommt. Beide Viewer können EKG Signale mit bis zu zwölf Kanälen mit Annotationen darstellen. Die Tabelle 1.4 zeigt eine Gegenüberstellung der beiden Viewer mit den Anforderungen für den EKG-Viewer für die H.ELGA. Wie aus der Tabelle ersichtlich erfüllen auch diese beiden Viewer nicht die gewünschten Anforderungen, weshalb in der gegenständlichen Arbeit ein eigener Viewer implementiert wurde.

Funktion	OpenECG	Physio Net	Anforderungen
Dateiformat	SCP	nur Signale der Physionet Datenbank	beliebiges
scrollen	nein	ja	ja
zoomen	Achsen können geändert werden	nein	ja
messen	nein	nein	ja
Raster	ja	ja	ja
Annotationen	ja	ja	ja
Kanäle	bis 12	bis 12	beliebig
Link	www.openecg.net	www.physionet.org	

Tabelle 1.4: Übersicht von Eigenschaften derzeit erhältlicher open-source Viewern und den Anforderungen für den Viewer für die H.ELGA

Kapitel 2

Methoden

In diesem Kapitel werden die im Rahmen der Diplomarbeit verwendeten und näher betrachteten Programme und Dateiformate beschrieben. Zu Beginn wird auf die Schnittstelle PyMat eingegangen, danach werden verschiedene Dateiformate die in die engere Auswahl gekommen sind erläutert. Im Anschluss gibt es einen Überblick der Entwicklungsumgebung und zum Schluss wird die Schnittstelle zu Zope erklärt.

2.1 Schnittstelle Matlab-Python

Die Signalanalyse in der H.ELGA erfolgt mit Matlab. Zu Beginn der vorliegenden Arbeit gab es bereits ein fertiges GUI zur Signaldarstellung in Matlab. Das in der H.ELGA implementierte Backend hingegen basiert auf Python. Um daher das Matlab-GUI in der H.ELGA verwenden zu können, muss eine Schnittstelle zwischen Matlab und Python verwendet werden. Eine der möglichen Schnittstellen ist PyMat.

2.1.1 Python

Python ist eine objektorientierte und plattformunabhängige Programmiersprache. Python besitzt eine sehr große Standardbibliothek, die vor allem für Internet Anwendungen geschrieben wurde. Es ist eine leicht zu erlernende Sprache, da die Syntax übersichtlich (z.B.: zwingende Einrückung)

und einfach ist. Python verwendet eine dynamische Typverwaltung [14], das bedeutet, dass der Typ zur Laufzeit und abhängig vom Einsatz der Variablen bestimmt wird. Dadurch erspart sich der Entwickler einiges an Arbeit und der Quelltext ist übersichtlicher und kürzer. Um die Module der Standardbibliothek zu erweitern, können sie mit C (Borland, Austin, USA) oder Python ergänzt werden.

Beispiel der Berechnung der Fakultät der Zahl n :

```
def faculty(n):
    if n == 0:
        return 1
    else:
        return n * faculty(n - 1)
```

2.1.2 Matlab

Matlab ist ein kommerzielles Programm zur Bearbeitung von mathematischen und technischen Aufgaben und deren grafischen Darstellung. Es ist in der Programmiersprache C geschrieben. Der Name Matlab kommt von *matrix laboratory*.

Eigenschaften von Matlab [13]:

- arbeitet mit Matrizen und Vektoren
- mathematische Probleme werden numerisch gelöst
- keine Compilersprache wie Java, Pascal oder C
- benutzerfreundliches System mit einfacher Syntax
- Grafiksystem zur Visualisierung von 2D und 3D Datenfeldern
- die Syntax lehnt sich stark an die mathematische Schreibweise an
- Matlab-Programme und -Toolboxen sind vollständig portierbar
- sehr umfangreiche Bibliothek von mathematischen Funktionen

2.1.3 PyMat

PyMat [26] ist eine Schnittstelle zwischen NumPy und Matlab, es ist für UNIX und Windows einsetzbar. NumPy ist eine Bibliothek für Python zum Arbeiten mit Arrays und linearer Algebra. Es sind folgende Funktionen mit PyMat möglich: `open`, `close`, `eval`, `get` und `put`. Beim Aufrufen einer Funktion mittels PyMat wird Matlab gestartet und die gewünschte Rechenoperation durchgeführt. Dieser Vorgang ist sehr langsam, da der Start von Matlab einige Zeit in Anspruch nimmt.

PyMat ist ein Open Source Projekt, wurde aber in den letzten Jahren nicht weiterentwickelt und ist deshalb auch nicht für die neueren Versionen von Matlab einsetzbar. Die aktuellste Download Version von sourceforge.net ist vom Jahr 2003 und getestet mit Python 2.2. und Matlab 6.5. PyMat ist in der Programmiersprache Python geschrieben.

Um die Schnittstelle zu starten, muss zuerst mit dem Befehl `H=pymat.open()` Matlab geöffnet werden. Danach können verschiedene mathematische Funktionen durchgeführt werden, auch eine grafische Darstellung von Kurven ist möglich. Mit `pymat.close(H)` wird Matlab dann wieder beendet.

Als Beispiel wird hier ein einfaches Python-Programm gezeigt, welches zum Testen der Schnittstelle PyMat dient. Dabei wird eine Matrix mit 100 Werten erstellt, von diesen Werten der Sinus berechnet und ausgegeben.

```
import pymat
i = 0
n = []
while i < 100:
    n.append(i)
    i=i+1
x = array(n)
H = pymat.open
pymat.put(H, 'x', x)
pymat.eval(H, 'y = sin(x)')
print pymat.get(H, 'y')
pymat.close(H)
```

2.2 Dateiformate

Biomedizinische Signale werden in den verschiedensten Datenformaten gespeichert. Es gibt zurzeit weit mehr als 1000 verschiedene Formate. Teilweise wurden sie von Firmen, von Forschungsgruppen oder von Organisationen der Standardisierung entwickelt. Meistens wurde für eine bestimmte Anwendung ein eigenes Datenformat entwickelt, das den Erfordernissen der Anwendung entspricht. Daher gibt es auch im Bereich der EKG Signale kein einheitliches standardisiertes Datenformat. Im Folgenden wird auf ein paar wichtige und häufig verwendete Formate näher eingegangen.

2.2.1 Dateiformat SCP(Standard Communication Protocol)

Dieses Format wurde von der Europäischen Initiative OpenECG entwickelt [17] und ist ein Open Source Format mit mehreren Tools. Die Basis für das SCP-ECG Format wurde im Jahre 1989 im Rahmen eines von der EU unterstützten Projektes gesetzt. Im Jahr 1993 wurde SCP bei der CEN (Europäisches Komitee für Normung) als Standard ENV 1064 genehmigt. Mit diesem Format ist es möglich EKG Daten zu speichern, konvertieren und komprimieren.

2.2.2 Dateiformat XML FDA

Dieses Format benutzt HL7 und XML als Standard und ist von der FDA (amerikanische Arzneimittelzulassungsbehörde) für Studien zur Arzneimittelzulassung genehmigt [6]. Die Basis für dieses Format ist der HL7 Standard (www.hl7.org). HL7 ist die Abkürzung von Health Level 7, wobei 7 für die Schicht des OSI-Referenzmodells steht. Eine HL7 Nachricht besteht aus einzelnen Segmenten, die wiederum in Felder aufgeteilt sind. Es gibt nun verschiedene Segmente, die wie in einem Baukastensystem beliebig zusammengesetzt werden können.

Die *Extensible Markup Language* XML wird für den Datenaustausch vor allem im Internet verwendet und hat sich in letzter Zeit als Standard etabliert.

Die Spezifikation zu XML wurde vom World Wide Web Consortium (W3C) im Jahre 1998 zum ersten Mal ausgegeben und ist im September 2006 in der vierten Version aktualisiert worden.

2.2.3 Dateiformat MAT

Die Software Matlab besitzt auch ein eigenes Datenformat. Die Signalanalyse der H.ELGA erfolgt mit Matlab und deshalb kommt auch das Format MAT zur näheren Betrachtung. MAT-Files [28] haben einen 128 Byte großen Header mit nachfolgenden Datenelementen. Jedes dieser Datenelemente besteht aus einem 8 Byte großen Tag und den eigentlichen Datenelementen. Dieser Tag spezifiziert die Anzahl der Bytes im Datenelement und gibt an, wie diese gelesen werden sollen, als 16-Bit Werte, als 32-Bit Werte, als Gleitkommazahl oder als ein anderer Typ. Das MAT Format besitzt auch die Möglichkeit der Datenkompression mittels gepuffertem gzip. Diese Technik benötigt weniger Speicherplatz als Systeme, die eine Datei als Gesamtes komprimieren.

2.2.4 Dateiformat WAV

Das Dateiformat WAV ist ein Containerformat zur Speicherung von Audiodaten, das auf dem Resource Interchange File Format (RIFF) aufsetzt, das von Microsoft für das Betriebssystem Windows definiert wurde. Bei diesem Format besteht die Struktur aus einer Sequenz von Datenpaketen. Der erste Teil ist der RIFF-Chunk, der die Datei als WAV-Datei identifiziert. Der zweite Teil ist der Format-Chunk, der gewisse Parameter beinhaltet. Der dritte Chunk enthält die eigentlichen Daten.

2.2.5 Dateiformat GDF

Das General Dataformat for Biosignals (GDF) [22] ist ein Format, welches für alle Biosignale eingesetzt werden kann und aktuell in der Version 2.0 erhältlich ist. Die Daten werden im Little Endian Format gespeichert. GDF ist ein Open Source Produkt, und viele Dateiformate können ins Format GDF ohne einen Datenverlust konvertiert werden. Der Header besteht aus

einem fixen Teil mit 256 Bytes und einem variablen Teil mit 256 Bytes, den Nummern der Kanäle. Bei diesem Dateiformat ist eine Datenkompression nur mit Hilfe eines anderen Programmes möglich.

2.2.6 Dateiformat EDF

Das European Data Format (EDF) ist ein einfaches Format zum Austausch und zur Speicherung von Biosignalen. Die Idee für dieses Format ist im Jahre 1987 im Rahmen des Schlaf-Kongresses in Kopenhagen entstanden, 1992 wurde es erstmals publiziert [10]. Zu diesem Zeitpunkt war EDF für EEG Signale konzipiert, im Jahre 2002 entstand EDF+, ein Dateiformat welches für alle Biosignale verwendet werden kann.

Ein großer Vorteil dieses Formates ist es, dass von mehreren Biosignalen die Werte, Notierung und Events in einem File gespeichert werden können. Es können aber auch nur die Notierung und Events ohne Signalwerten gespeichert werden. Eine EDF Datei besteht aus einem Header und einem Datensatz. Im Header befinden sich Informationen über den Patienten, Aufnahme datum, Aufnahmegerät, Wertebereich und eine Beschreibung der Signalkanäle. Im nachfolgenden Datensatz befinden sich die verschiedenen Signale nacheinander.

2.3 Anforderungen an das Dateiformat

Die Anforderungen an ein Dateiformat sind je nach Einsetzbarkeit sehr unterschiedlich. Für den EKG-Viewer ergeben sich folgende Anforderungen an das Dateiformat:

1. minimaler Header
2. möglichst kleine Datei
3. schnelle Übertragbarkeit
4. Metainformationen (Marker, Annotationen)
5. einfaches, universelles Format

Diese Eigenschaften erfüllt aber keines der vorher diskutierten Dateiformate. Deshalb wurde vorerst ein eigenes - sehr einfaches - Format für die H.ELGA entwickelt, der Viewer jedoch so programmiert, dass ohne großen Aufwand auch jedes andere Dateiformat eingelesen werden kann.

2.4 Entwicklungsumgebung

2.4.1 Java

Java¹ ist eine Programmiersprache, die von der Firma Sun Microsystems (Santa Clara, USA) entwickelt wurde. Die Java-Laufzeitumgebung Java Runtime Environment (JRE) beinhaltet unter anderem die Java Virtual Machine (JVM) die zur Ausführung von Java-Programmen benötigt wird. Die aktuellste Version zur Zeit ist JRE 1.6.0.

Die wichtigsten Eigenschaften von Java sind [19]:

- Plattformunabhängigkeit
- objektorientiert
- Syntax ähnlich wie bei C und C++
- umfangreiche Klassenbibliothek
- Sicherheit von Internet Anwendungen

Erklärung der wichtigsten Eigenschaften:

Plattformunabhängigkeit: Die Plattformunabhängigkeit entsteht daraus, dass die Java Programme in neutralen Byte Code kompiliert werden. Der Byte Code ist Maschinenbefehlen sehr ähnlich, aber ist nicht für eine Maschine spezifisch.

objektorientiert: Programme, die in Java geschrieben werden, benutzen Objekte. Ein Objekt kann in Datenelemente und die dazugehörigen

¹<http://java.sun.com/>

Funktionen (=Methoden) unterteilt werden. Der Inhalt von Datenelementen gibt den Zustand an, in dem sich das Objekt befindet. Funktionen sind Operationen, die ein Datenobjekt von einem Zustand in einen anderen versetzen.

Die größten Vorteile von Java bestehen darin, dass es sehr weit verbreitet und plattformunabhängig ist. Unterschied zwischen Applikation und Applets bei Java:

- Applikationen sind selbständig laufende Programme
- Applets können nur mit Hilfe einer Web-Page laufen

Sicherheit in Java

Bei Client/Server Anwendungen ist der Bereich Sicherheit sehr wichtig, da es hier eine Menge Angriffspunkte gibt. Vor allem bei der Programmiersprache Java, die primär für die Entwicklung von verteilten Anwendungen im Internet zum Einsatz kommt, sollte es einen ganzheitlichen Schutz gegen Angriffe auf allen Ebenen geben. Das Sicherheitskonzept von Java [20] besteht aus den folgenden Hauptkomponenten, eine Übersicht dazu ist in Abbildung 2.1 ersichtlich:

- Entwicklungsumgebung mit einem Compiler, der Java-Quellcode in den systemunabhängigen Bytecode übersetzt.
- Java Core Classes für die Entwicklung von Anwendungen und Applets.
- Laufzeitumgebung mit der JVM zur Ausführung des Bytecodes.
- Werkzeuge für die Konfiguration der Sicherheitsmechanismen (keytool, jarsigner).

Im Weiteren werden die Teile des Sicherheitskonzeptes von Java ausführlicher erklärt.

Compiler

Der Compiler übersetzt den Quellcode in einen Bytecode und überwacht

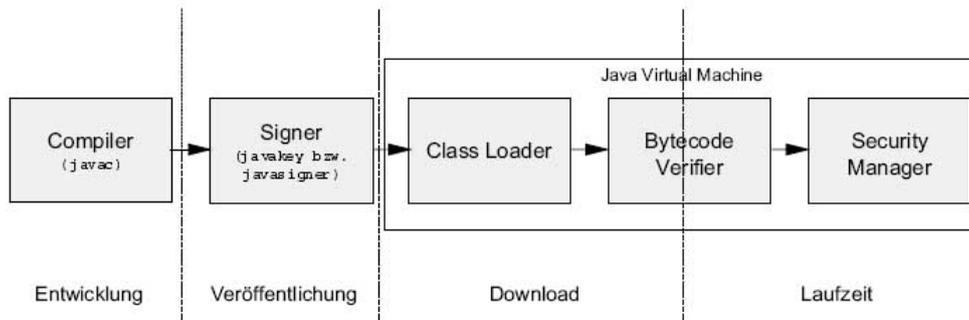


Abbildung 2.1: Übersicht des Sicherheitskonzepts bei Java [20]

dabei, ob sich der Quellcode an die von Java festgelegten Regeln der Sprachdefinition hält. Ein paar wichtige Regeln:

- Jeder Methode wird eine eigene Zugriffsebene zugeordnet. Möglichkeiten dafür sind `public`, `protected` und `private`. Dadurch ist der Zugriff auf bestimmte Daten begrenzt.
- Ein sicherer und verifizierter Code kann mit `final` vor einer weiteren Vererbung und einer Methodenüberschreibung geschützt werden.
- Alle Variablen müssen vor der Verwendung mit einem gültigen Wert initialisiert werden.

Sandbox

Die Sandbox, zu Deutsch Sandkasten, ist ein Hauptbestandteil des Java Sicherheitskonzeptes. Jedes Applet wird in einer eigenen Sandbox ausgeführt. In dieser kann das Applet keinen Schaden anrichten und hat keinen Zugriff auf lokale Ressourcen. Die Sandbox besteht aus den drei nachfolgend erklärten Teilen.

Class Loader

Der Class Loader unterscheidet bei der Standardeinstellung zwischen vertrauenswürdigen und nicht vertrauenswürdigen Klassen. Beim Laden einer Klasse wird grundsätzlich auf den Code einer lokalen Datei zugegriffen, bevor ein Code vom Internet geladen wird. Durch diese Vorgehensweise wird

sichergestellt, dass die Basisklassen der Laufzeitumgebung nicht von einer gleichlautenden Klasse aus einem Applet überschrieben und ausgeführt werden. Wenn mehrere Java Anwendungen auf einem Rechner laufen, werden ihnen vom Class Loader eigene Adressräume zugewiesen. Dadurch können verschiedene Applets nicht miteinander kommunizieren und es kann zu keinen Problemen bei Klassen mit gleicher Namensgebung kommen. Alle Basisklassen werden nicht mehr vom Verifier geprüft, da sie als vertrauenswürdig eingestuft werden.

Verifier

Hier erfolgt die Überprüfung der Binärdaten in vier Schritten. Im ersten Schritt wird die Einhaltung des im Sprachstandard vorgeschriebenen Formats für Java Klassen überprüft. Im zweiten Schritt wird die Integrität der Klasse geprüft. Der Verifier untersucht ob das Object von einer gültigen, nicht finalen Superklasse abgeleitet wurde. Bei Klassen des Typs Object gilt diese Einschränkung nicht. Im dritten Schritt erfolgt der Aufruf des Bytecode Verifiers, er überprüft den Code auf unzulässige Instruktionen. Der vierte Schritt prüft nacheinander für jede Instruktion im Byte Code, ob auf dem Stack der JVM genügend Elemente vom richtigen Datentyp zur Verfügung stehen und bei Zugriffen auf lokale Variablen keine falschen Typen verwendet werden.

Security Manager

Die Aufgabe des Security Managers besteht darin, alle gefährlichen Operationen und nicht vertrauenswürdig eingestuften Klassen zu überwachen und nötigenfalls einzuschreiten. Ein Einschreiten wird erforderlich beim Lesen und Schreiben von lokalen Dateien, unerlaubten Aufbau von Netzwerkverbindungen, Verwendung des Druckers.

2.4.2 Applet

Ein Java-Applet ist in der Programmiersprache Java verfasst und wird über einen Webbrowser ausgeführt. Applets haben wie jedes Programm, das lokal auf dem Rechner ausgeführt wird, ein Sicherheitsrisiko. Ein Nachteil von Applets ist die Ladezeit, die ein Applet zum Herunterladen und zum Initialisieren braucht. Das Applet wird vom Webserver zum lokalen Rechner

übertragen und dort mit Hilfe der JVM im Browser ausgeführt.

Sicherheit bei Applets

Da Applets in einer Sandbox laufen, können sie normalerweise nicht auf lokale Dateien, andere Programme oder Systemkomponenten zugreifen. Weiters kann ein Applet keine Befehle oder andere Programme am Client starten oder eine Internetverbindung aufbauen, mit Ausnahme zu dem Server, von welchem das Applet gestartet wurde. Bei signierten Applets kann der Anwender Ausnahmen der Sicherheitsregeln zulassen, z.B. ist ein Zugriff auf verschiedene Dateien oder Ordner auf der Festplatte möglich.

Programmaufbau eines Applets

Ein Applet kann nicht wie eine Application alleine gestartet werden, denn es besitzt keine *main()* Methode und kann nur von einem Browser aus aufgerufen werden. Wird ein Applet gestartet, werden verschiedene Methoden vom Browser automatisch aufgerufen. Dies beginnt mit dem Aufruf von *init()*. Diese Methode wird nur einmal aufgerufen und in ihr erfolgen die Initialisierungen. Danach folgen Aufrufe der Methoden *start()* und *stop()* immer dann, wenn ein Applet im Browser sichtbar ist oder von der Seite wieder verschwindet.

Applet in eine HTML Seite einbinden

Applets werden über einen `<applet>` Tag in eine HTML Seite eingebunden, die Ausführung erfolgt durch die JVM. Im Tag sind alle notwendigen Informationen enthalten. Er dient der Schnittstelle zwischen dem Applet und der HTML Seite. Weiters können einem Applet beliebig viele Parameter der HTML Seite übergeben werden.

```
<applet code = 'helloworld.class'  
codebase = 'U:/hello/world'  
archive = 'helloworld.jar'  
name = 'HelloAppler'  
width = '200'
```

```
height = "100" >
<param name= "text"
value="Hallo!">
</applet>
```

code Gibt die Datei an, in der sich die Hauptklasse des Applets befindet, mit der Erweiterung `.class`.

width Breite des Applets in Pixel.

height Höhe des Applets in Pixel.

codebase Dient dazu, den Pfad zum Applet anzugeben, falls dieses sich nicht im Verzeichnis der HTML Datei befindet.

archive Zur Angabe von Archiven, die Java Klassen enthalten. Es können verschiedene Archive durch Beistriche getrennt angegeben werden.

name Damit kann dem Applet ein Name zugewiesen werden, über den es von anderen Applets auf derselben Seite aufgerufen werden kann.

param name Parametername, muss der gleiche wie im Applet sein.

value Wert für den Parameter, der an das Applet übergeben wird. Wird immer als String übergeben.

Signieren von Applets

Java Applets laufen normalerweise in einer Sandbox und besitzen nur eingeschränkte Möglichkeiten, um auf Ressourcen des Systems zugreifen zu können. Damit ein Applet nun auch auf geschützte Ressourcen zugreifen kann, muss es signiert werden. Es ist nur möglich, Jar Files zu signieren. Ein Jar-File ist ein Java Archiv, in dem sich alle Java Klassen und eine Manifest Datei befinden. In dieser Manifest Datei befinden sich die wichtigsten Meta-Informationen, wie z.B. Name der Hauptklasse, Informationen über Java Beans und andere Informationen. Jar Dateien können auch durch einen Doppelklick gestartet werden, wenn eine JRE installiert ist. Ein Browser fragt immer nach, ob ein signiertes Applet gestartet werden darf. Seit JDK 1.2

ist es nicht nur möglich zwischen signierten und nicht signierten Applets zu unterscheiden, sondern es wird der Code in verschiedene Sicherheitsstufen eingeteilt. Jeder Java Klasse wird einer Protection Domain zugeordnet. In dieser Domain werden die Rechte verwaltet. Die einzelnen Rechte werden in der Datei .policy gespeichert. In Abbildung 2.2 ist der Umgang mit Applets im Java Sicherheitskonzept ersichtlich.

Vorgang beim Signieren eines Applets

Die folgenden Befehle werden in der Kommandozeile eingegeben. Zuerst muss ein Schlüssel erzeugt werden. Dies erfolgt mit Hilfe des Befehls

```
keytool -genkey.
```

Es müssen nun verschiedene Parameter (Name, Name der Firma, Ort) und ein Passwort eingegeben werden. Mit dem Befehl

```
jarsigner testfile.jar testkey
```

wird die Jar Datei testfile.jar mit dem Schlüssel testkey signiert.

2.4.3 JFreeChart

Das JFreeChart Projekt wurde im Jahre 2000 von David Gilbert begonnen, mittlerweile wird es von ca. 50.000 Entwicklern verwendet. Diese Java Bibliothek ist vor allem zur Erstellung beliebiger Diagramme (Kreisdiagramm, Balkendiagramm, Liniendiagramm...) in 2D und 3D und beliebiger Kurven entstanden. Es können mehrere Plots untereinander, nebeneinander oder in verschiedenen Tabs dargestellt werden. Beispiele können auf der Homepage² betrachtet werden. Im Jahr 2006 wurde JFreeChart 388.348 mal von der Seite sourceforge.net heruntergeladen. Diese Downloadzahl spricht für eine weite Verbreitung dieses Packages.

JFreeChart ist eine Bibliothek für Java mit folgenden wichtigen Eigenschaften:

²<http://www.jfree.org/jfreechart/samples.html>

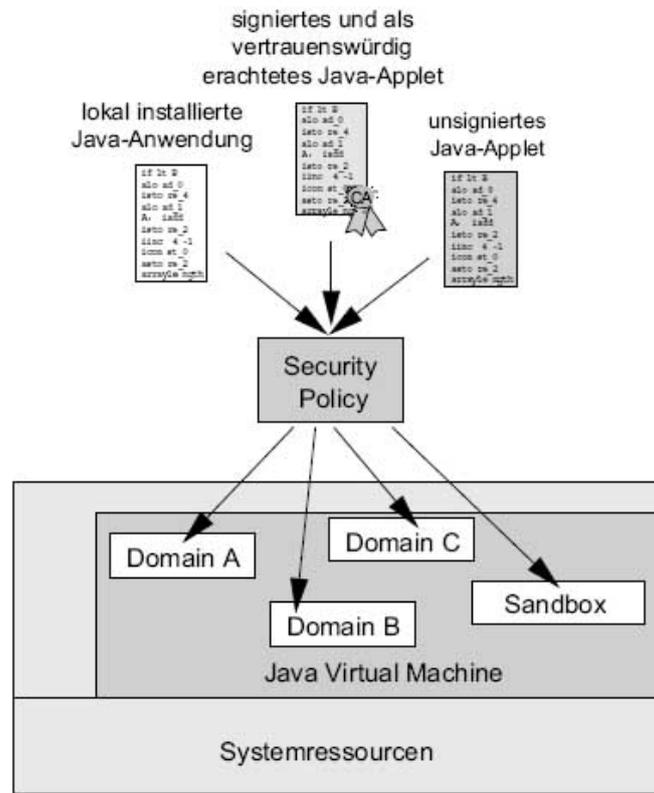


Abbildung 2.2: Applets in der Java Sicherheitsumgebung, aus [20]

- Programmierschnittstelle
- sehr gute Dokumentation
- open source Produkt
- flexibles Design
- Ausgabe in PNG, JPEG oder PDF möglich
- mehrere Kurven in einem Plot darstellbar
- mehrere Plots auf einer Seite
- zeitabhängige Kurven

2.4.4 andere Chart Packages

Chart FX ChartFX (SoftwareFx, Boca Raton, USA) ist ein kostenpflichtiges Programm. Es ist nur für Diagramme und nicht für zeitabhängige Kurven entwickelt worden.

(www.chartfx.com)

easychart easychart (ObjectPlanet, Oslo, Norwegen) ist ebenfalls ein kostenpflichtiges Programm mit der Hauptverwendung zur Darstellung von Diagrammen.

(www.objectplanet.com/easycharts)

jcharts jcharts ist ein Open Source Projekt mit vielen Möglichkeiten zur Darstellung von Diagrammen und Kurven, vor allem zur Darstellung von Diagrammen mit Servlets. Es ist leider sehr wenig Dokumentation vorhanden, die letzte Version ist von 2003 und seit September 2007 wird dieses Projekt nicht mehr weiterbetreut.

(<http://jcharts.sourceforge.net/index.html>)

chart2D chart2D ist ebenfalls ein Open Source Projekt, das viele Möglichkeiten der Darstellung bietet. Auch eine Dokumentation ist vorhanden. Seit September 2002 gibt es aber kein Update dieses Projektes mehr.

<http://chart2d.sourceforge.net/index.php>

jchart2D jchart2D ist auch ein Open Source Projekt zur Darstellung von Kurven in einem zweidimensionalen Koordinatensystem, das vor allem für laufzeitabhängige Kurven entwickelt wurde. Weiters sind eine gute Dokumentation und Javadoc vorhanden. Es gibt regelmäßige Updates, aber leider nicht viele Möglichkeiten zur Darstellung.

(<http://jchart2d.sourceforge.net/index.shtml>)

2.4.5 Eclipse

Eclipse (www.eclipse.org, Eclipse Foundation, Portland, USA) ist ein Open-Source-Framework zur Entwicklung von Software. Hauptsächlich wird es mit der Programmiersprache Java verwendet. Eclipse ist der Nachfolger von IBM

Visual Age for Java 1.4.0. Im Jahr 2004 wurde die rechtlich eigenständige „Eclipse Foundation“ gegründet. Das Programm Eclipse besteht nur aus einer kleinen Kernapplikation, der Plattform-Runtime, und ist für die Verwaltung und Ausführung der Plugins zuständig. Beim Start wird festgestellt, welche Plugins lokal installiert sind, dazu wird das Unterverzeichnis `plugins` durchsucht. Dabei werden die Manifest-Dateien `plugin.xml` eingelesen. Diese Informationen speichert Eclipse in einer Plugin-Registry als Baumstruktur. Nach diesem Vorgang überträgt das Laufzeitmodul die Verantwortlichkeit für das Starten der Umgebung an ein Anwendungs-Plugin. Standardmäßig wird diese Aufgabe vom Workbench-Plugin übernommen, welches das Hauptfenster öffnet und Menüs und Symbolleisten erstellt. Die gesamte Eclipse Funktionalität inklusive des Workbenches wird durch Plugins verfügbar gemacht.

Plugin

Ein Plugin ist der kleinste Teil einer Eclipse Funktion, die extra entwickelt und installiert werden kann. Dies ist ein wichtiger Mechanismus, mit dem sich neue Funktionen definieren lassen. Kleine Werkzeuge können als einzelne Plugins entwickelt werden, größere können auf mehrere Plugins aufgeteilt sein. Plugins sind in Java kodiert und enthalten neben der Manifest Datei [5] `plugin.xml` meistens eine JAR-Bibliothek und weitere Ressourcen (z.B. Bilder). Es gibt aber auch Plugins, die keinen Java Code enthalten. Diese dienen zum Beispiel der Erweiterung der Online Hilfe im HTML Format. In der XML-Datei werden Verbindungen zu anderen Plugins festgelegt. Jedes Plugin besitzt eine beliebige Anzahl von benannten Erweiterungspunkten (*extension points*) und eine beliebige Anzahl an Erweiterungen (*extensions*) zu anderen Erweiterungspunkten. Mit Hilfe dieser Erweiterungspunkte kann man neue Funktionen bereitstellen oder hinzufügen.

Hier ein paar Gründe, warum die Softwareentwicklung mit Eclipse erfolgte:

Open Source Jeder kann zur Verbesserung dieses Programmes beitragen und einen Teil dafür entwickeln. Änderungen des Codes werden vom „Projekt Management Comitee“ und den „Projekt Leaders“ begutachtet, bevor sie allen zur Verfügung gestellt werden.

Benutzerorientiert Eclipse wird nach den Wünschen der Benutzer weiterentwickelt.

Modularer Aufbau mit Plugins Neben der Entwicklung der Plattform kann jeder seine eigenen Plugins schreiben und anderen Benutzern zur Verfügung stellen.

Plugins Mittlerweile gibt es riesige Bibliotheken mit Plugins zum Download für jeden Benutzer unter <http://eclipse-plugins.2y.net>. Mit dem Datum 26.09.2007 gibt es 1514 verschiedene Plugins.

2.5 Schnittstelle Zope

Im Projekt H.ELGA wird als Application Server ein Zope Server [14] verwendet. Ein Zope Server ermöglicht es Protokolle schnell einzubinden, web-basierte Applikationen zu entwickeln und den Konnex zwischen anderen internetbasierten Services herzustellen. Er besitzt eine integrierte Objekt Datenbank, die Zope Object Database (ZODB) . In dieser können Applikationen und Daten als Objekte gespeichert werden. Relationale Datenbanken können mit Hilfe eines Datenbank Adapters eingebunden werden.

2.5.1 Datenspeicherung

Werden Dateien im System H.ELGA registriert, wird entsprechend ihrer Patientenummer ein Storage URL zusammengesetzt. Unter diesem URL werden die Dateien im File Server abgelegt. Werden die Dateien wieder benötigt, baut Zope aufgrund derselben Datenbankinformation wieder den Storage URL zusammen. Danach schaltet Zope den Storage URL für einen Client zum Download frei und erzeugt einen http URL, der auf den Freigeschalteten Storage Bereich verweist.

2.5.2 File laden

Das Modul BiomedicalStreamConveyor (BMSC) wurde im Vorfeld von der Austrian Research Centers GmbH - ARC entwickelt. Dieses Modul besitzt

drei Funktionen: Authentifikation, Datenkodierung und Datentransfer. Für die Übertragung der EKG Daten kommt die letzte Funktion dieses Moduls zum tragen, der Datentransfer. Es ist eine Server/Client Schnittstelle zur sicheren Datenübertragung mit HTTP/HTTPS. Die Grundidee dieses Modules war es Datentransfers für große Daten zu ermöglichen, ähnlich wie bei FTP. Es werden der Transferstatus, die Verbindungsgeschwindigkeit und die wiederaufgenommenen Transfers angezeigt.

Der Viewer fordert beim Starten zwei URLs (Signal und Event) von der Datenbank an. Dies erfolgt mit Hilfe der Befehle

```
getParameter(APPLETPARAM_ECGRESOURCEURL) und  
getParameter(APPLETPARAM_ECGMARKERURL).
```

Kann keine Datei geladen werden, erscheint eine Fehlermeldung.

Mögliche Fehlermeldungen:

invalid ecg data url Diese Meldung wird ausgegeben, wenn eine URL in einem falschen Format oder wenn ein unbekanntes Übertragungsprotokoll angegeben wurde.

fail to load ecg data Diese Meldung wird ausgegeben, wenn es sich um einen allgemeinen Fehler beim Laden der Datei handelt.

Kapitel 3

Ergebnisse

3.1 Lösungsansätze

Die ursprünglich angestrebte Lösung, das in Matlab bereits vorhandene GUI mit PyMat als Schnittstelle zu Zope zu verwenden erwies sich als nicht erfolgreich. Folgende Probleme sind dabei aufgetreten:

1. Funktionen laufen sehr langsam mit der Schnittstelle PyMat.
2. Die Schnittstelle PyMat wurde nicht weiterentwickelt. Sie hätte neu komiliert werden müssen, was sehr zeitaufwendig gewesen wäre.
3. Matlab und Zope müssen auf dem selben Server laufen, derzeit gibt es keine Unix Lizenz für Matlab.

Da sich dieser Lösungsansatz als langsam, teuer und sehr zeitaufwändig erwies, fiel die Entscheidung einen eigenen Viewer als Java Applet zu entwickeln.

3.2 EKG-Viewer

3.2.1 Dateiformat

Da für die Anforderungen in der H.ELGA kein optimal geeignetes Dateiformat gefunden wurde, ist ein eigenes, sehr einfaches Format erstellt worden.

Es handelt sich dabei um eine Datei mit binären Daten, wobei in der ersten Zeile ein kurzer Header aus Text steht und danach die Werte binär kodiert kommen.

Im Rahmen der H.ELGA gibt es zwei Dateien, die an den Viewer übertragen werden. Die erste entspricht dem Signal und die zweite Datei den Markern von gefundenen Schrittmacher Spikes.

Beispiel der ersten Zeile einer Signal-Datei:

easyG signal - nrow=33536 ncol=12 fs=256

easyG Ist der Name der Signalanalyse Software (electrocardiogram analysis system **Graz**).

signal Es handelt sich um eine Datei mit den Werten des Signals.

nrow=33536 Gibt die Anzahl der Zeilen (Samples) an.

ncol=12 Gibt die Anzahl der Spalten (Kanäle) an.

fs=256 Ist der Wert der Abtastfrequenz.

Beispiel der ersten Zeile einer Marker-Datei:

easyG pointlist nrow=133 ncol=2 types: 1=S (c=[0 0 0] s=10 m=.) 2=bol (c=[0 0 0] s=10 m=.) samples types

easyG Ist der Name der Signalanalyse Software.

pointlist Es handelt sich um eine Datei mit den Werten der Marker.

nrow=133 Gibt die Anzahl der Zeilen an.

ncol=2 Gibt die Anzahl der Spalten an.

types Typen der verschiedenen Marker.

1=S Die Zahl 1 steht für einen Spike, wird in der Darstellung mit S gekennzeichnet.

c=[0 0 0] Definiert die Farbe des Markers im Format Rot-Grün-Blau.

s=10 Gibt die Strichstärke des Markers an.

m=. Gibt die Darstellungsart des Markers an (Punkt, Linie...).

samples types Gibt den Aufbau der nachfolgenden Tabelle mit den Daten an, in der ersten Spalte stehen die Samplewerte, in der zweiten Spalte die Typen der Marker.

Die Zahlenwerte des Signals sind in der Datei als ASCII/ANSI Zeichen gespeichert. Immer acht Zeichen ergeben einen Zahlenwert im Format Gleitkommazahl mit doppelter Genauigkeit, dies entspricht einer Zahl mit acht Byte. Die Zeichen werden in einer Schleife mit acht Durchläufen als Integer Werte eingelesen.

Die Integer-Zahl wird in eine Bit-Zahl umgewandelt, danach erfolgt eine Linksverschiebung (es werden die Bits nach links verschoben und auf der rechten Seite Nullen nachgeschoben) um $8 \cdot \text{Anzahl des Schleifendurchlaufes}$. Dann kommt es zu einer bitweisen Oder-Verknüpfung der letzten Zahl und der gerade eingelesenen Zahl. Nach acht Schleifendurchläufen wird der Wert in ein Double umgewandelt und in einem Array gespeichert.

Beispiel für die Umwandlung in eine double Zahl:

in der Datei steht: \textcircled{R} , $1/2$. h \AA \AA \AA \AA \AA \AA \AA \AA

Integerwerte der Zeichen: 174 44 189 46 104 198 175 191

erster Schleifendurchlauf ergibt: 1010 1110

zweiter Schleifendurchlauf ergibt: 1000 0010 1010 1110

Von den 64 Bit steht das Erste für das Vorzeichen (0= positiv, 1=negative), die nächsten elf Bits stellen den Exponenten dar und die restlichen 52 Bits die Mantisse. Die Berechnung der Gleitkommazahl erfolgt mit folgender Formel:

$$Z = (-1)^{VZ} * (1,0 + M/2^{52}) * 2^{E-1023}$$

Erklärung der Abkürzungen:

Z Zahl im Format double

VZ Vorzeichen, erstes Bit

M Mantisse, 52 Bits

E Exponent, elf Bits

3.2.2 Dateigröße

Die Größe der Datei ist wichtig, da von ihr die Ladezeit abhängt. Je größer die Datei ist, desto länger dauert das Laden und die Darstellung des Signals. Beim selbst definierten Dateiformat für das Projekt H.ELGA wurde die Dateigröße nicht optimiert, denn es handelt sich nur um kleine Dateien. Tabelle 3.1 gibt einen Überblick der Dateigröße von EKGs mit verschiedenen Kanalanzahl und Aufzeichnungsdauer.

Dateiname	Spalten	Zeilen	Dateigröße	Aufzeichnungsdauer
Datei 1	15.360	2	240 kB	60 s
Datei 2	17.152	2	268 kB	67 s
Datei 3	2.500	15	293 kB	9,77 s
Datei 4	21.248	2	332 kB	83 s
Datei 5	25.000	15	3,06 MB	97,65 s

Tabelle 3.1: Dateiengröße

Wie in der Tabelle 3.1 ersichtlich, steigt die Dateigröße mit der Anzahl der Kanäle und der Aufnahmedauer sehr stark an. Die für die H.ELGA aufgezeichneten EKGs haben immer nur zwei Kanäle und eine Aufnahmedauer von einer Minute. Daraus ergibt sich eine Dateigröße von ungefähr 250 kB für die Signal-Datei. Die Marker-Datei ist ungefähr 1 kB groß und spielt im Vergleich keine Rolle.

3.2.3 Viewer

In der H.ELGA werden einem Patienten zugeordnete EKGs im File Server gespeichert. Wird ein EKG neu in die H.ELGA geladen, wird das Programm easyG gestartet und automatisch eine Analyse des EKGs durchgeführt. Während der Analyse werden auch die zwei für den Viewer benötigten Dateien (Signal- und Marker Datei) erzeugt und im File Server gespeichert. Das EKG

wird mit den Markern als Bild dargestellt. Ebenfalls ist die Magnetfrequenz rechts oben im Bild ersichtlich. In diesem Bild werden nur 20 Sekunden des gesamten Signals auf den Magneteffekt getriggert dargestellt.

Durch Klicken auf das Bild des EKGs wird in einem neuen Fenster ein Java Applet, der Viewer (siehe Abbildung 3.1) geöffnet. In den Starteinstellungen werden alle Marker und Frequenzwerte angezeigt, diese sind mit Hilfe der Kästchen ein- und ausschaltbar. Weiters ist es möglich, innerhalb dieses Signalverlaufs horizontal und vertikal zu scrollen und zoomen. Des Weiteren ist es möglich, beliebige Zeitwerte aus dem Signal herauszumessen.

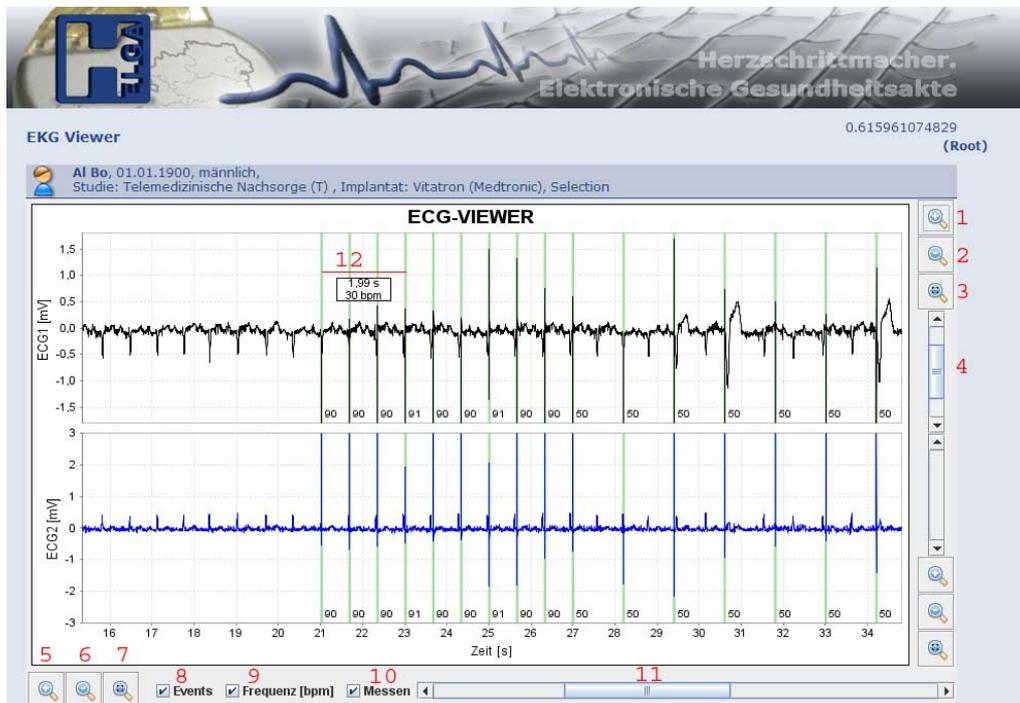


Abbildung 3.1: Beispiel eines EKGs in der H.ELGA

1. Button zum vertikalen Hineinzoomen für Kanal 1
2. Button zum vertikalen Herauszoomen für Kanal 1

3. Button um die gesamte Signalamplitude für den Kanal 1 darzustellen
4. Scrollbalken für die horizontale Achse für den Kanal 1
5. Button zum horizontalen Hineinzoomen für alle Kanäle
6. Button zum horizontalen Herauszoomen für alle Kanäle
7. Button um das gesamte Signallänge darzustellen
8. Kästchen zum Ein- und Ausschalten der Events
9. Kästchen zum Ein- und Ausschalten der Frequenzwerte
10. Kästchen zum Ein- und Ausschalten der Messfunktion
11. Scrollbalken für die horizontale Achse
12. Beispiel für einen Messwert

Die Buttons zum Zoomen und die Scrollleiste sind für den Kanal 2 ident. Die Frequenzwerte werden immer zwischen zwei Events berechnet und dargestellt. Der Benutzer hat auch die Möglichkeit den zeitlichen Abstand zwischen zwei Punkten im EKG zu bestimmen. Der Messvorgang läuft folgendermaßen ab:

Mit dem ersten Klick der linken Maustaste wird der Startpunkt in Form eines roten Kreuzes markiert, mit dem zweiten Klick der Endpunkt. Zwischen diesen beiden Punkten wird nun eine Linie gezeichnet und die Zeit berechnet und in Sekunden und Schlägen pro Minute (bpm) angegeben. Es können nur Zeiten und keine Amplituden gemessen werden. Die Messung ist für jeden Kanal unabhängig, d.h. es können gleichzeitig zwei verschiedene Messungen durchgeführt werden. Wird nach einer Messung wieder die linke Maustaste auf dem Signalbild gedrückt, wird damit ein neuer Startpunkt gekennzeichnet. Mit dem Kästchen *Messen* kann die Funktion ein- und ausgeschaltet werden. Beim Ausschalten werden alle Punkte gelöscht.

3.2.4 Voraussetzungen

Zur Verwendung des Viewers muss ein Internetzugang vorhanden sein. Ebenfalls muss am Computer Java ab der Version 1.4.0 installiert sein, sonst kann der Viewer nicht geladen und gestartet werden.

3.2.5 allgemeiner Programmaufbau

Einlesen

Derzeit kann der Viewer nur das selbst definierte Dateiformat einlesen. Um ein anderes Dateiformat verwenden zu können, muss die Funktion *Einlesen* entsprechend dem neuen Format adaptiert werden.

Bei Aufruf des Applets werden diesem zwei Urls mit den Pfaden zu den beiden Dateien übergeben. Beim Einlesen wird zuerst die Datei mit den Signal Werten eingelesen und danach die Datei mit den Markern. Zuerst wird die erste Zeile eingelesen und die wichtigen Parameter wie Anzahl der Zeilen, Anzahl der Spalten, Abtastfrequenz und die Kanalbezeichnungen gespeichert. Im zweiten Schritt werden die binär kodierte Signalwerte eingelesen, dekodiert und gespeichert. Die Kanäle werden nacheinander eingelesen und in jeweils einer eigenen Liste als Double Wert gespeichert. Weiters wird hierbei das Minimum und Maximum der Signalamplituden festgestellt und als Parameter Minimum und Maximum gespeichert. Diese sind zur Erstellung der Grafik für die Grenzen notwendig. Bei der Marker Datei wird ebenfalls zuerst die Textzeile eingelesen und folgende Parameter gespeichert:

- Anzahl der Spalten
- Anzahl der Zeilen
- Marker Name, Bezeichnung des Markers
- Marker Color, wird in Form von drei Werten (Rot, Gelb, Blau) angegeben
- Marker Text, dieser Text wird in der Grafik zum jeweiligen Marker geschrieben

Die binären Daten werden eingelesen, dekodiert und in einer Liste als Double Werte gespeichert. Es existieren zwei Listen, eine Eventliste (Zeitpunkte des

Events) und eine Typenliste (Art des Events).

Signaldarstellung

Zur Darstellung des EKG Signals wird die Klasse `CombinedDomainXYPlot` vom Package `JFreeChart` verwendet. Bei dieser Klasse sind mehrere Subplots (entspricht den Kanälen) mit einer gemeinsamen Achse (entspricht der Zeitachse) möglich. Dabei wird die X-Achse als Zeitachse verwendet und die Y-Achse stellt die Amplitude des EKGs dar. Die Amplitudenwerte werden in einer Schleife eingelesen und zu den dazugehörigen Zeitwerten in einer XY-Serie gespeichert. Die Zeitwerte werden mit Hilfe der Zeilenanzahl und der Abtastrate berechnet:

$$\text{Zeitwerte} = \frac{\frac{\text{Zeilen}}{\text{Abtastrate}}}{\text{Zeilen}}$$

Diese XYSerie wird dann in einem Subplot dargestellt.

Verwendete Klassen und Funktionen von `JFreeChart`

In diesem Teil werden die wichtigsten der verwendeten Klassen des Package `JFreeChart` kurz erklärt.

`CombinedXYPlot` Ist eine Möglichkeit mehrere Kurven mit einer gemeinsamen Achse darzustellen.

`XYSeries` Stellt eine Serie mit Datenelementen in der Form (x,y) dar. Die Werte innerhalb dieser Serie werden nach X-Werten aufsteigend sortiert und doppelte X-Werte sind erlaubt.

`XYPlot` Ist eine generelle Klasse um Daten in der Form (x,y) darzustellen.

`XYItemRenderer` Ist ein Interface zum Rendern von Punkten der Klasse `XYPlot`.

JPanel Ist eine grafische Benutzeroberfläche zur Darstellung von Objekten aus JFreeChart.

JToolBar Diese Klasse bietet die Möglichkeit Buttons, Checkboxes und Scrollleisten in einer Menüleiste darzustellen.

JButton Diese Klasse definiert einen Button der durch Mausklick gedrückt werden kann. Es kann ein beliebiges Bild als Hintergrund verwendet werden.

JCheckBox Diese Klasse definiert eine Checkbox, diese kann aktiviert und deaktiviert werden.

JScrollBar Damit wird eine Scrollbar implementiert. Mit Hilfe der Parameter können die Eigenschaften der Scrollbar geändert werden.

ItemListener Wird der Status eines Buttons oder einer Checkbox geändert, wird dies vom ItemListener registriert.

ChartChangeEvent Ändert sich die Grafik (z.B. durch scrollen oder zoomen), kommt es zu einer Rückmeldung an das Applet.

In der Abbildung 3.2 ist das Klassendiagramm des Viewers ersichtlich. Wegen der besseren Übersicht wurden nicht alle Zoom Buttons und Scroll Leisten einzeln dargestellt.

3.2.6 EKG-Viewer für mehrere Kanäle

Bei dieser Version des Viewers kann eine beliebige Anzahl an Kanälen eingelesen und dargestellt werden. In Abbildung 3.3 ist ein EKG mit 12 Kanälen und zusätzlich noch drei Kanälen mit Herzparameter dargestellt. Zur einfacheren Übersicht werden immer drei Kanäle in einer Registerkarte dargestellt. Ein Umschalten zwischen den Registerkarten erfolgt durch einen Mausklick auf diese. Die Zeitachse kann für jede Registerkarte extra gesteuert werden. Die Funktionen Scrollen, Zoomen und Messen sind für jeden Kanal einzeln möglich. Bei der Funktion *Einlesen* wird die Anzahl der Kanäle eruiert und

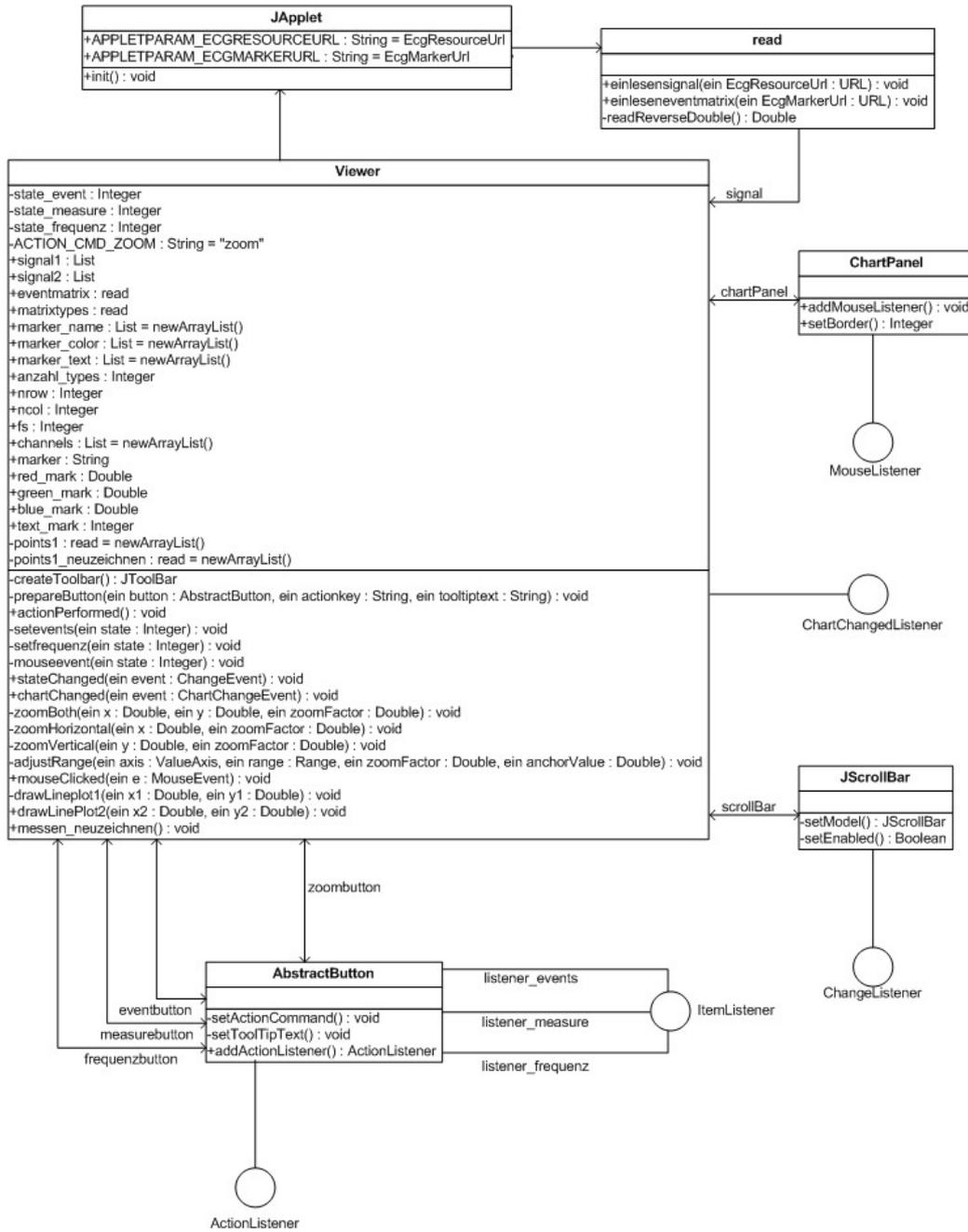


Abbildung 3.2: Klassendiagramm

die Anzahl der Registerkarten berechnet. Danach werden die Registerkarten nacheinander dargestellt.



Abbildung 3.3: Viewer für Mehrkanal EKGs

3.2.7 EKG Darstellungszeiten

Die folgenden Tabellen 3.2 und 3.3 geben einen Überblick der Darstellungszeiten für Zwölf- bzw. Zwei-Kanal EKGs mit unterschiedlichen Größen. Unter der Darstellungszeit versteht man die Zeit, die das Applet benötigt das Signal zu laden und darzustellen. Diese Messungen wurden mit Signalen der Abtastrate von 256 Hz und Signalwerten von 8 Byte durchgeführt.

3.2.8 EKG-Viewer für Langzeit EKGs

Es wurde eine Version des Viewers für Langzeit EKGs entwickelt. Das große Problem dabei ist, das Signal in einer verhältnismäßig schnellen Zeit zu la-

Name	Spalten	Zeilen	Dateigröße	EKG Dauer	Darstellungszeit
Datei 1	5.000	12	468 kB	19,5 s	3,7 s
Datei 2	10.000	12	937 kB	39,1 s	4,1 s
Datei 3	20.000	12	1,83 MB	78,1 s	5,2 s
Datei 4	50.000	12	4,57 MB	195,3 s	9,1 s

Tabelle 3.2: Zeiten der EKG Darstellung von Dateien mit verschiedener Größe

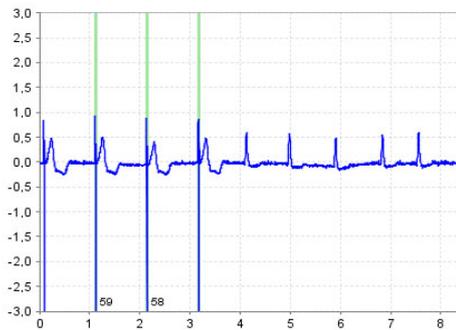
Name	Spalten	Zeilen	Dateigröße	EKG Dauer	Darstellungszeit
Datei 1	5.000	2	78,1 kB	19,5 s	2,9 s
Datei 2	10.000	2	156 kB	39,1 s	3,5 s
Datei 3	20.000	2	312 kB	78,1 s	3,7 s
Datei 4	50.000	2	781 kB	195,3 s	4,1 s
Datei 5	100.000	2	1,52 MB	390,6 s	5,7 s

Tabelle 3.3: Zeiten der EKG Darstellung für Zweikanal EKGs mit verschiedener Größe

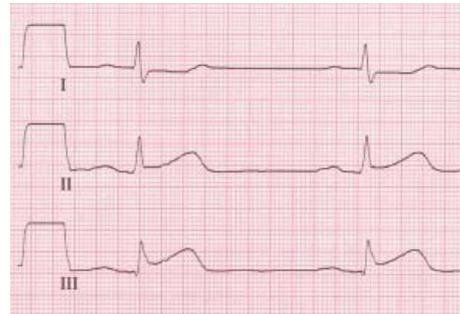
den. Die Signal-Dateien werden sehr groß und das Laden benötigt sehr viel Zeit. Deshalb werden bei dieser Version immer nur 20 Sekunden geladen und dargestellt. Mit den beiden Buttons *weiter* und *zurück* können die nächsten oder letzten 20 Sekunden dargestellt werden. Die Tabelle 3.4 gibt einen kurzen Überblick über die Ladezeiten des Viewers mit verschiedenen Dateien. Bei diesen Dateien handelt es sich um lokal gespeicherte Daten.

Dateiname	Spalten	Zeilen	Dateigröße	Signaldauer	Ladezeit
Datei 1	15.360	2	240 kB	60 s	3,5 s
Datei 2	17.152	2	268 kB	67 s	3,5 s
Datei 3	2.500	15	293 kB	9,77 s	3,5 s
Datei 4	21.248	2	332 kB	83 s	4,9 s
Datei 5	25.000	15	3,06 MB	97,65 s	21,5 s

Tabelle 3.4: Ladezeiten lokal gespeicherter Dateien



(a) EKG mit automatischem Raster von JFreeChart



(b) EKG auf EKG Papier mit Raster

Abbildung 3.4: Vergleich des automatisch erzeugten Rasters und dem Raster auf einem EKG Papier

3.2.9 Raster

Der Raster wird in JFreeChart automatisch erzeugt (Abbildung 3.4a). Es wird bei jeder horizontalen und vertikalen Beschriftung eine Rasterlinie gezeichnet. Da dies aber nicht dem Raster eines EKGs entspricht (Abbildung 3.4b), wurde auf Wunsch der Ärzte eine Methode Raster im Applet implementiert. Damit ist es möglich den Raster beliebig einzustellen. Es können auch dünne und dicke Linien gezeichnet werden. Das Zeichnen des Rasters erfolgt mit der Funktion

```
XYLineAnnotation lineplot = new XYLineAnnotation (x1, y1, x2, y2,
                                                    Strichstärke, Farbe);
```

Dabei werden innerhalb einer Schleife einzelne Linien gezeichnet. Die x- und y-Werte müssen dabei als double angegeben werden. Die Strichstärke und Farbe werden vorher definiert.

Diese Methode ist aber nur in der Version des Mehrkanalviewers implementiert.

3.3 Geschwindigkeitstest

Wegen der teilweise langen Ladezeiten des Applets wurden die beiden benötigten Packages von JFreeChart verkleinert. Nicht benötigte Funktionen der Packages wurden weggelassen. Somit hat sich die Größe der Packages verringert, was sich auf die Ladezeiten des Applets auswirkt. In Tabelle 3.5 ist die Reduzierung ersichtlich.

Package Name	Originalgröße	verkleinert	Verkleinerung
jcommon-1.0.5.jar	318 kB	86,3 kB	72%
jfreechart-1.0.2.jar	1,08 MB	343 kB	68%

Tabelle 3.5: Größe der Packages

In der Tabelle 3.6 ist der Unterschied zwischen den Darstellungszeiten mit den originalen und den verkleinerten Packages ersichtlich. Die Darstellungszeit ist die Zeit, die das Applet braucht um die Signalwerte zu laden, einzulesen und das Signal grafisch darzustellen. Es ist ersichtlich, dass es keine signifikante Verringerung der Zeiten mit den verkleinerten Packages gibt. Der Versuch wurde mit einer Übertragungsrate von 12 Mbit/s durchgeführt. Den größten Teil der Darstellungszeit benötigt das Applet zum Zeichnen des Signals. Dabei spielt die Packagegröße keine Rolle.

Datei	Spalten	Zeilen	Darstellungszeit mit originalen Packages	Darstellungszeit mit verkleinerten Packages
Datei 1	15369	2	1,9 s	1,9 s
Datei 2	17152	2	2,2 s	2,1 s
Datei 3	2500	15	2,3 s	2,3 s

Tabelle 3.6: Darstellungszeiten mit originalen und verkleinerten Packages

3.4 Einbindung in H.ELGA

Der Workflow einer EKG Darstellung mit Hilfe des Viewers im Rahmen der H.ELGA ist in Abbildung 3.5 dargestellt. Um ein EKG mit Hilfe des Viewers zu betrachten muss der Benutzer in der H.ELGA angemeldet sein und die notwendigen Rechte besitzen.

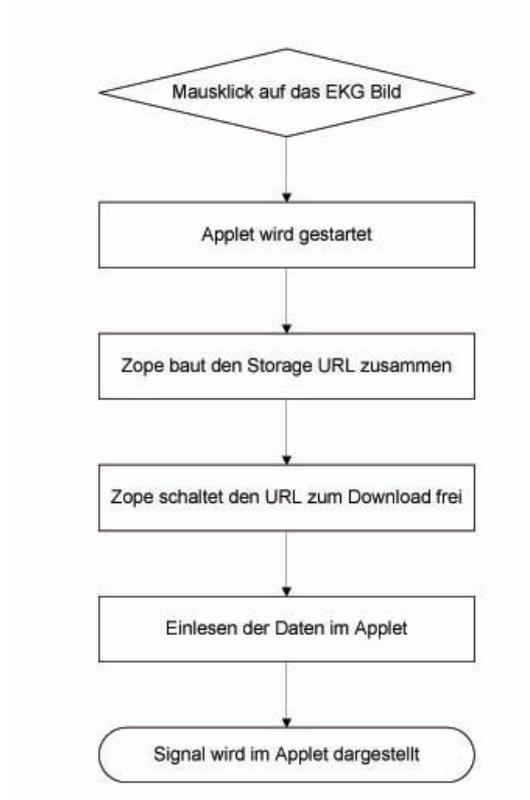


Abbildung 3.5: Workflow des Viewers in der H.ELGA

3.5 Sicherheit

Sicherheit spielt im Bereich der medizinischen Datenübertragung eine große Rolle. Im Rahmen der H.ELGA erfolgt die Übertragung der Signaldaten mit Hilfe des Moduls BMSC. Dieses Modul benutzt das Hypertext Transfer Pro-

protocol Secure (HTTPS) , welches Secure Sockets Layer (SSL) zur Verschlüsselung verwendet.

Das Java Applet läuft in einer geschützten Umgebung (Sandbox) ab. Es kann nicht von aussen auf dieses Applet zugegriffen werden und das Applet selbst hat auch keinen Zugriff auf andere Programme. Der Url zum EKG wird mit einem eigenen Schlüssel des Moduls BMSC erstellt. Die EKG Daten werden am Rechner weder zwischengespeichert, noch sind sie danach vom Rechner abrufbar. Es werden keine personenbezogen Daten verwendet. Das bedeutet, dass mit dem EKG Signal kein Zusammenhang zu einem Patienten erstellt werden kann.

3.6 Anforderungen

Im Kapitel 1.6.6 wurden die Anforderungen an den EKG Viewer aufgelistet. Tabelle 3.7 zeigt, dass alle mit dem neuen Viewer erfüllt sind.

Funktion	OpenECG	Physio Net	EKG Viewer
Dateiformat	SCP	nur Signale der Physionet Datenbank	Derzeit nur das im Zuge dieser Arbeit entwickelte Format.1)
scrollen	nein	ja	✓
zoomen	Achsen können geändert werden	nein	✓
messen	nein	nein	✓
Raster	ja	ja	✓
Annotationen	ja	ja	✓
Kanäle	bis 12	bis 12	beliebig
Link	www.openecg.net	www.physionet.org	

Tabelle 3.7: Vergleich bestehende EKG Viewer mit dem programmierten Viewer; 1) Einfache Möglichkeit zur Erweiterung auf beliebige andere Formate

In Abbildung 3.6 ist das Web-Portal H.ELGA mit einem geöffneten EKG Viewer dargestellt.

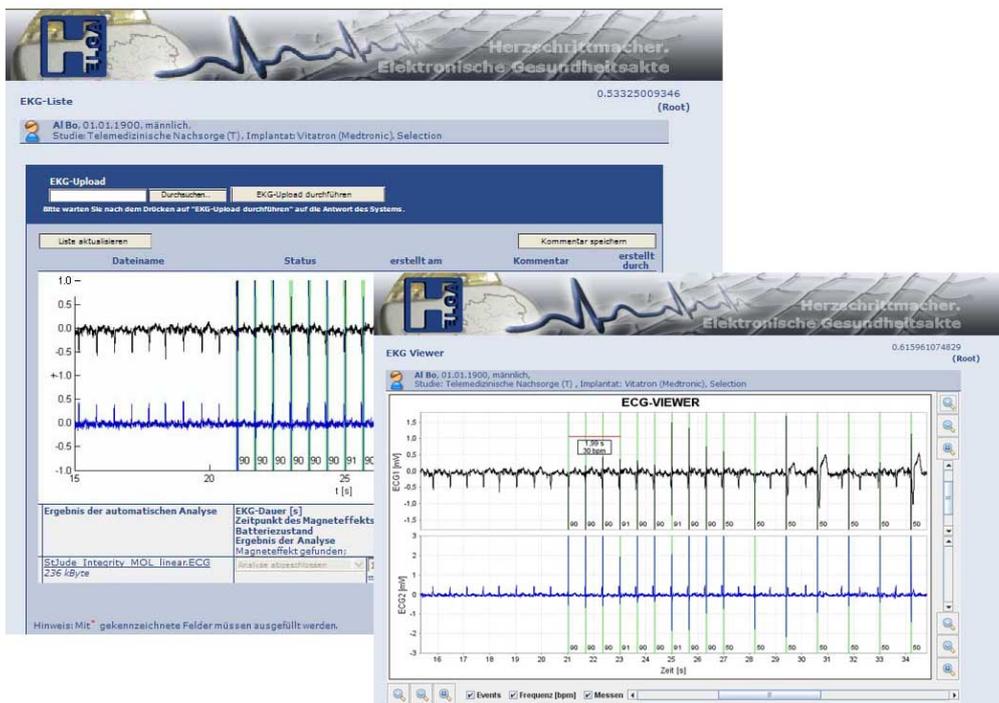


Abbildung 3.6: H.ELGA und EKG Viewer

Kapitel 4

Diskussion

4.1 Viewer

Mit dem entwickelten EKG Viewer lassen sich die Zweikanal EKGs von HSM Patienten im Rahmen ihrer Nachsorgeuntersuchungen problemlos und einfach betrachten. Die gewünschten Funktionen

- Zoomen
- Scrollen
- Marker darstellen
- Frequenz darstellen

wurden alle implementiert. Weiters wurde der Viewer auch erweitert um EKGs mit einer beliebigen Kanalanzahl zu betrachten. Auch verschiedenen Rastereinstellungen sind möglich. Somit sind die Anforderungen laut Tabelle 1.4 erfüllt worden.

Ein Problem war es, dass auf dem Computer in der Schrittmacher Ambulanz der Uni Klink Graz die Software Java ursprünglich nicht installiert war. Mit Hilfe eines System Administrators konnte die Installation jedoch durchgeführt werden.

Bei der Anwendung des Viewers ist es bisher zu keinen Problemen gekommen. Eine Einschulung der Benutzer war nicht notwendig, da der Viewer leicht zu bedienen ist. Im Rahmen der Diplomarbeit wurde auch eine kurze

Bedienungsanleitung erstellt. Diese kann als PDF auf der Projektseite von H.ELGA heruntergeladen werden.

4.2 Dateiformat

Obwohl es viele verschiedene Dateiformate im Bereich der Biomedizinischen Signale gibt, konnte kein passendes Format für die Anforderungen des Viewers gefunden werden. Es gibt derzeit noch kein Standard Format im Bereich der EKG Aufzeichnung. Für den Fall, dass es in nächster Zeit ein standardisiertes Format gibt, wurde vorgesorgt. Der Viewer lässt sich leicht ändern, es muss nur die Methode *einlesen* neu programmiert werden.

Da bei Aufruf eines Applets immer alle Packages mitgeladen werden, wurde versucht durch Verkleinerung der beiden Packages von JFreeChart die Darstellungszeit zu verkleinern. Es wurde dabei eine Reduktion um ca. 70% erreicht (siehe Tabelle 3.5), indem die nicht benötigten Funktionen der Packages entfernt wurden. Jedoch hat die Verkleinerung nur eine Zeitreduktion von maximal einer Sekunde bei der Darstellung des Signals gebracht. Bei den derzeitigen Internetverbindungen macht sich der Unterschied der Verkleinerung nicht bemerkbar. Der Großteil der Darstellungszeit ist die Zeit bis das Signal gezeichnet wird.

Derzeit werden die Signalwerte des EKGs als *double* Werte (8 Byte) gespeichert, es wäre aber genauso möglich Werte vom Datentyp *float* (4 Byte) zu verwenden. Dadurch hätte die Signal Datei nur die halbe Größe.

Versuche haben gezeigt, dass damit die Zeit in der das EKG in das Applet geladen wird sinkt. Aber die Zeit in der das Signal dargestellt wird sich nur minimal verkleinert. Daraus ist ersichtlich, dass sich für Langzeit EKGs ein Problem ergibt. Bei Signalen die länger als fünf Minuten sind, ist das Laden der beiden Dateien möglich, aber bei der Darstellung des Signals gibt es Probleme. Es dauert auch mindestens fünf Sekunden bis das Signal dargestellt wird. Tabelle 3.2 und Tabelle 3.3 zeigen eine Übersicht der Darstellungszeiten für Zwölf- bzw. Zwei- Kanal EKGs mit verschiedener Größe. Die Abtastfrequenz beträgt bei allen Signalen 256 Hz. Eine weitere Möglichkeit wäre es, die Abtastrate von 256 Hz herabzusetzen. Dadurch wird die Datei kleiner

und es sind weniger Werte für den Viewer zu bearbeiten. Für Langzeit EKGs ist dieser Viewer in der derzeitigen Version noch nicht einsetzbar.

In einer weiteren Version werden im Viewer immer nur 20 Sekunden dargestellt. Durch zwei zusätzliche Buttons *next* und *back* kann ein Weiter- und Zurückblättern in der Zeitachse realisiert werden. Damit kann das Problem der Darstellung von langen EKG Signalen behoben werden.

4.3 Einsatzmöglichkeiten des EKG-Viewers

Der Viewer kann für weitere Projekte verwendet werden, da das Einlesen an kein bestimmtes Format gebunden ist, sondern beliebig geändert werden kann.

Ein möglicher Anwendungsfall in Zukunft ist, den Viewer im Rahmen von klinischen Studien von Arzneimittelzulassungen zu verwenden. Bei diesen Studien werden EKGs der Probanden aufgezeichnet und eine mögliche Veränderung registriert.

4.4 Ausblick

Es gibt noch viele weitere Möglichkeiten den Viewer zu erweitern, mögliche nächste Schritte:

- Kompression der Signal- und Marker Datei
- Verbesserung der Darstellungszeit bei Langzeit EKGs
- Einlesen von anderen Dateiformaten implementieren
- Ausdrucken von EKG Signalen
- EKG Ausschnitt zum Arztbrief hinzufügen
- Verwendung von anderen Biosignalen (z.B. Elektroenzephalogramm - EEG, Elektromyogramm - EMG)

4.5 Fazit

Der Viewer läuft seit einigen Monaten problemlos im realen Betrieb und bei über 100 telemedizinischen Nachsorgeuntersuchungen gab es keine Probleme. Laut den Ärzten, die den Viewer verwenden, ist er sehr benutzerfreundlich. Durch die Darstellung des ganzen Signals mit den Markern erfolgt eine effizientere Befundung der EKGs durch den Kardiologen. Ein Ausdrucken der EKGs ist nicht mehr notwendig, da mit Hilfe von Mausclicks Zeiten aus dem EKG gemessen werden können.

Literaturverzeichnis

- [1] ARBEITSGRUPPE HERZSCHRITTMACHER UND ELEKTROPHYSIOLOGIE DER SGK: *Richtlinien 2005 zur Nachkontrolle von Patienten mit implantierten Defibrillatoren*, 2005.
- [2] ARBEITSGRUPPE HERZSCHRITTMACHER UND ELEKTROPHYSIOLOGIE DER SGK: *Herzschrittmacher Statistik*. Arbeitsgruppe Herzschrittmacher und Elektrophysiologie der SGK, 2006.
- [3] BACHINGER E.: *Lebenserwartung und Mortalität in Wien und Österreich - Internationaler Vergleich*. Stadt Wien, 1:8, 2003.
- [4] BERNSTEIN A., DAUBERT J.: *The revised NASPE/BPEG generic code for antibradycardia, adaptive-rate, and multisite pacing*. Pacing and Clinical Electrophysiology, 25:260, 2002.
- [5] BOLOUR A.: *Notes on the Eclipse Plug-in Architecture*. Eclipse Corner Articles, 1:1, 2003. <http://eclipse.org/articles>.
- [6] BROWN B., KOHLS M., STOCKBRIDGE N.: *FDA XML Data Format Design Specification*, 2002.
- [7] CLUNIE D.: *Extension of an open source DICOM toolkit to support SCP-ECG Waveforms*. 2nd OpenECG Workshop 2004, Berlin, Germany, 2004.
- [8] FACHGRUPPE HERZSCHRITTMACHER UND BQS BUNDESGESCHÄFTSSTELLE QUALITÄTSSICHERUNG GGBH: *Jahresbericht 2005 des Deutschen Herzschrittmacher-Registers*. Fachgruppe Herzschrittmacher und BQS Bundesgeschäftsstelle Qualitätssicherung gGmbH, 2005.

- [9] FRASER JD., GILLIS AM., IRVINE ME., ET AL.: *Guidelines for Pacemaker follow-up in Canada: A consensus statement of the Canadian Working Group on Cardiac Pacing*. Can J Cardiol, 16(3):355–363, March 2000.
- [10] KEMP B., VÄRRI A.: *A simple format for exchange of digitized polygraphic recordings*. Electroencephalography and Clinical Neurophysiology, 1:391–393, 1992.
- [11] KOLLMANN A.: *Patientennahe Herzschrittmacher-Nachsorge*. Diplomarbeit, Technische Universität Graz, 2002.
- [12] KOLLMANN A., HAYN D., KASTNER P., ET AL.: *Telemedizin in der Herzschrittmacher-Nachsorge*. Tagungsband der Telmed 2006 - Telematik im Gesundheitswesen, 1:203–211, 2006.
- [13] KUZMIN D.: *Matlab-Kurs*. Universität Dortmund, 2007.
- [14] LEDERER D.: *Python und Zope als Unterrichtswerkzeuge*. Diplomarbeit, Fachhochschule Technikum Wien, 2007.
- [15] LEMKE B., NOWAK B., PFEIFFER D.: *Leitlinien zur Herzschrittmachertherapie*. Zeitschrift für Kardiologie, 94(10):1–17, 2005.
- [16] MOODY G.: *WFDB Programmer's Guide*. Harvard-MIT Division of Health Sciences and Technology, 2008.
- [17] OPENECG: *Dateiformat SCP*. Internet, 2007. <http://www.openecg.net>.
- [18] OPPENHEIM A., SCHAFFER R., BUCK J.: *Zeitdiskrete Signalverarbeitung*. Pearson Studium, 2004.
- [19] PRATL H.: *Java Einführung*. Zentraler Informatikdienst ZID Universität für Bodenkultur Wien, 2007. <http://www.boku.ac.at/javaeinf>.
- [20] RAEPPEL M.: *Sicherheitskonzepte für das Internet*. Dpunkt Verlag, Mai 2001.

- [21] ROTMAN B., KOLLMANN A., SCHREIER G., ET AL.: *Integriertes Therapiemanagement auf Basis der H.ELGA*. Tagungsband der eHealth 2008, 1:17–22, 2008.
- [22] SCHLÖGL A.: *GDF - A General Dataformat For Biosignals*. Internet, 2006.
- [23] SCHREIER G., HAYN D., KOLLMANN A., ET AL.: *Automated and manufacturer independent assessment of the battery status of implanted cardiac pacemakers by electrocardiogram analysis*. Conf Proc IEEE Eng Med Biol Soc, 1:76–79, 2004.
- [24] SCHWEGLER J.: *Der Mensch - Anatomie und Physiologie*. Georg Thieme Verlag Stuttgart, 1998.
- [25] STATISTIK AUSTRIA: *Jahrbuch der Gesundheitsstatistik*. Bundeszentralamt Statistik Austria, 2006.
- [26] STERIAN A.: *PyMat - An interface between Python and MATLAB*. Internet, 1999.
- [27] ÖSTERREICHISCHE GESELLSCHAFT FÜR MEDIZINRECHT: *Telemedizin und Recht*, 2002.
- [28] THE MATHWORKS: *MAT-File Format*. The MathWorks, September 2005.