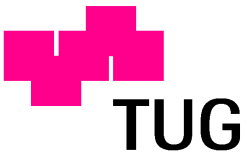
Diplomarbeit

Entwicklung einer Validierungsumgebung für ein EKG-Analysesystem

Jörg Teubl, Bakk. techn.

Institut für Genomik und Bioinformatik
Technische Universität Graz
Leiter: Univ.-Prof. Dipl.-Ing. Dr. techn. Zlatko Trajanoski



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

Betreuer: Dipl.-Ing. Dr. techn. Günter Schreier,
ARC Seibersdorf research GmbH

Graz, März 2004

Diese Arbeit wurde bei der

ARC Seibersdorf research GmbH - Standort Graz Biosignalverarbeitung und Telemedizin

Leiter: Dipl.-Ing. Dr. techn. Günter Schreier, MSc

durchgeführt.

Kurzfassung

Die automatische Analyse von Elektrokardiogrammen (EKGs) bekam in den letzten Jahren einen immer höheren Stellenwert in der klinischen EKG-Diagnostik. In dieser Arbeit wurde eine Validierungsumgebung für ein automatisches EKG-Analysesystem implementiert. Mit dieser Validierungsumgebung soll die Entwicklung neuer EKG-Analysealgorithmen beschleunigt und deren Qualität verbessert werden. Es wurde eine Datenbank entwickelt, die dem EKG-Algorithmusentwickler die Verwaltung aller Daten abnimmt, die bei einem Validierungsprozess anfallen. Um detaillierte Endberichte der Validierung anzeigen zu können, wurde eine graphische Oberfläche implementiert, die dem Entwickler ein flexibles Filtern der Validierungsergebnisse ohne Kenntnisse der Structured Query Lanuage (SQL) ermöglicht. Zahlreiche Verfahren zur statistischen Bewertung und graphischen Darstellung der Validierungsergebnisse wurden implementiert.

Schlüsselwörter: Validierung, Biosignalverarbeitung, Automatische EKG-Analyse, Java, Matlab, Datenbank

Abstract

Over the past few years automatic analysis of electrocardiograms (ECGs) has gained more and more significance in the field of clinical ECG-diagnosis. With this thesis a validation tool for an automatic ECG analysis system was developed in order to accelerate the development process of new ECG analysis algorithms and improve their quality. A database was developed to help the ECG algorithm developer to manage all the data that are produced in a validation process. A graphical user interface was implemented to help the developer to query the validation results without knowing the Structured Query Language (SQL). Various statistical methods to evaluate and illustrate the validation results were developed.

Keywords: biosignal processing, automatic ECG analysis, validation, java, matlab, database

Danksagung

Besonderer Dank gilt meinen Eltern, die mir mein Studium ermöglicht haben und mich bei allen Vorhaben während dieser Zeit tatkräftig unterstützt und motiviert haben.

Weiters möchte ich mich bei allen Mitarbeitern der ARC Seibersdorf research GmbH am Standort Graz bedanken, die mich bei der Erstellung meiner Diplomarbeit unterstützt haben. Besonders möchte ich mich bei meinem Betreuer DI Dr. techn. Günter Schreier bedanken, da er mir bei der Bearbeitung meines Themas sehr viel Freiheit ließ. Großer Dank gilt auch DI Dieter Hayn, der sich immer Zeit genommen hat, wenn ich Fragen bezüglich des easyG - electrocardiogram analysis system Graz - hatte. Weiters möchte ich mich bei beiden für die nützlichen Anregungen bedanken, die sie mir bei unseren zahlreichen Diskussionen gegeben haben.

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Alle Personenbezeichnungen in dieser Arbeit sind geschlechtsneutral zu verstehen. Zur Erhöhung der Lesbarkeit wird durchgehend die männliche Form verwendet.

Graz, März 2004

Inhaltsverzeichnis

1 Einleitung

Zu Beginn dieser Arbeit erhält der Leser Hintergrundinformationen zum Elektrokardiogramm (EKG). Danach werden einige Begriffe, die für das Verständnis dieser Diplomarbeit wichtig sind, anhand des EKG-Signalverlaufes erklärt und definiert. In weiterer Folge wird auf die Aufgabenstellung eingegangen und die Umsetzung kurz erläutert.

1.1 Das Elektrokardiogramm

1.1.1 Historische Entwicklung

Seit der Aufzeichnung des ersten Elektrokardiogramms (EKG) durch den Herzspezialisten Augustus Desire Waller an seiner Bulldogge Jimmie (Abb. 1.1) sind 117 Jahre vergangen.

Wenige Jahre später wurde die Elektrokardiographie durch Willem Einthoven (1860-1927), der Professor für Physiologie an der Universität Leiden in den Niederlanden war, in die Klinik eingeführt. 1924 bekam er für die Entdeckung des Mechanismus des Elektrokardiogramms den Nobelpreis für Medizin [1].

Heute ist das EKG ein fixer Bestandteil der modernen Medizin, da es bei der Diagnostik zahlreicher Erkrankungen, wie beispielsweise Herzinfarkten, Herzrhythmusstörungen oder anderer Erkrankungen des Herzkreislauf-Systems eine, zentrale Rolle spielt [2]. Krankheiten des Herz-Kreislaufsystems verursachten im Jahr 2002 48,5 % aller Sterbefälle in Österreich [3]. Daher ist es unbedingt notwendig, die EKG-Diagnostik weiterzuentwickeln.

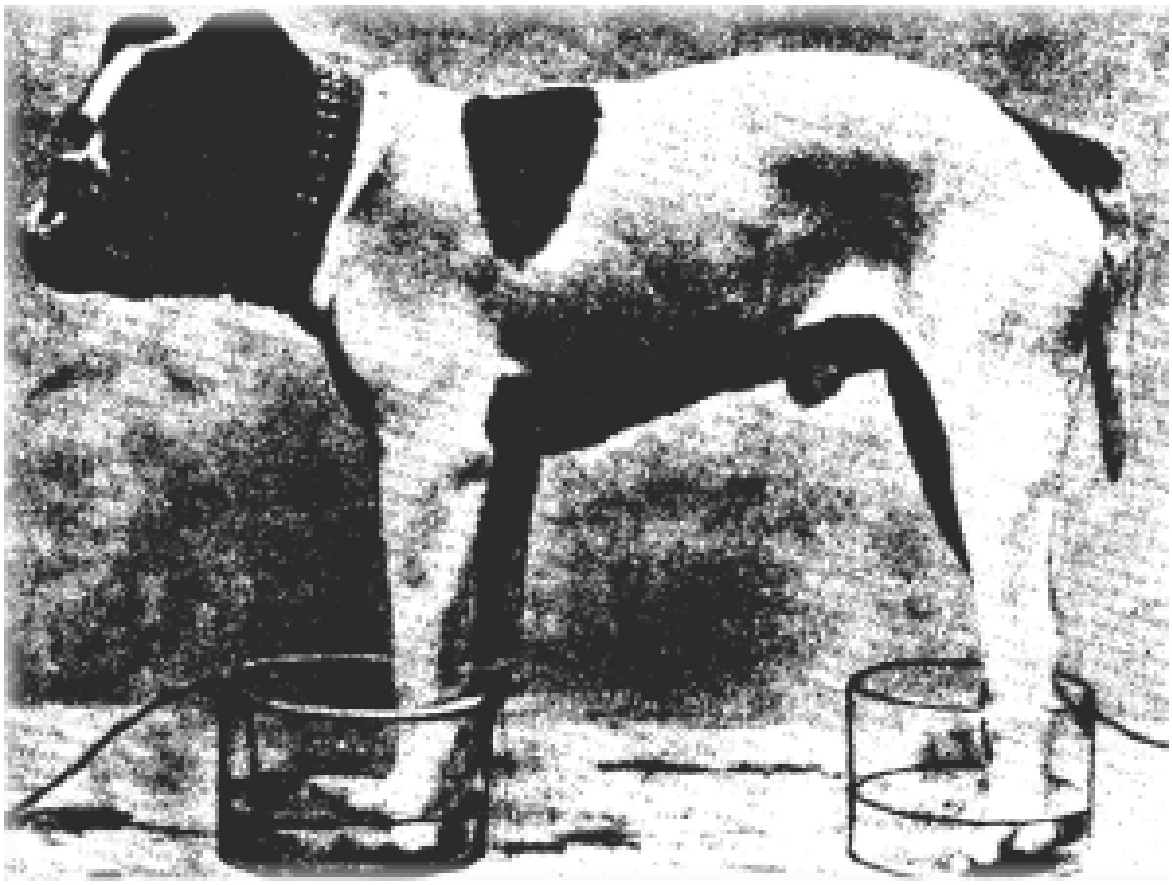


Abbildung 1.1: Die elektrische Potenzialdifferenz zwischen den beiden Wannen, in denen die Bulldogge steht, wird von der Herzaktivität verursacht. Augustus Desire Waller führte diesen Versuch 1887 aus. Das war die Geburtsstunde des Elektrokardiogramms [4].

Bedeutung der automatischen EKG-Analyse

In den letzten Jahren ist es in zahlreichen Bereichen gelungen, die Kardiologen bei der EKG-Diagnostik durch automatische EKG-Analysen zu unterstützen. EKG-Signalanalysealgorithmen werden zum Beispiel verwendet, um die riesigen Datenmengen, die bei Langzeit-EKGs gewonnen werden, zu filtern. Eine kardiologische Intensivstation ist heutzutage ohne eine automatische EKG-Überwachung nicht mehr vorstellbar. In den Defibrillatoren, die in der Notfallmedizin benötigt werden, helfen EKG-Signalanalysealgorithmen dem medizinischen Personal bei der Entscheidung, ob ein Patient einen Elektroschock bekommen soll.

1.1.2 Signalverlauf

Die Erregunsausbreitung im Herzen verursacht an der Körperoberfläche ein elektrisches Feld, das sich zeitlich verändert. Das EKG stellt die Aufzeichnung von Potentialdifferenzen, die durch dieses elektrische Feld entstehen, zwischen definierten Messpunkten in Abhängigkeit von der Zeit dar (Abb. 1.2) [5].

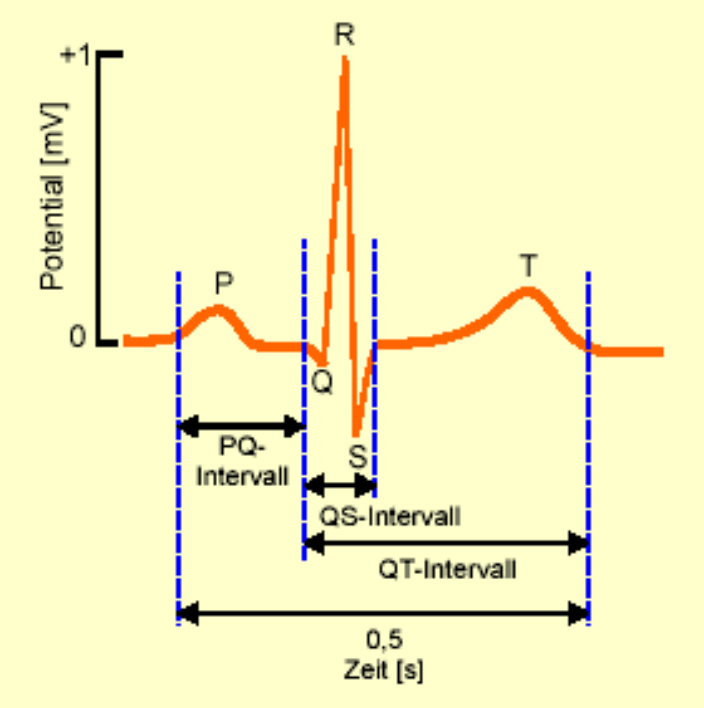


Abbildung 1.2: Standard EKG-Aufzeichnung und Definition wichtiger charakteristischer Punkte und Intervalle [6].

Die einzelnen EKG-Abschnitte sind einer bestimmten Phase der Herzerregung zugeordnet. Die P-Welle repräsentiert die Erregung der Vorhofmuskulatur. Das PQ-Intervall bezeichnet die Zeit, die die Erregung braucht, um vom Sinusknoten bis zur Kammer zu gelangen. Der QRS-Komplex entspricht der Erregungsausbreitung im Kammermyokard. Während der ST-Strecke ist die Kammer vollständig und gleichmäßig erregt. Die T-Welle ist der elektrische Ausdruck der Kammer-Repolarisation [7].

1.1.3 Klassifizierung von Merkmalen eines EKG-Signals

Jedem EKG-Merkmal kann ein Name, der in weiterer Folge als marker bezeichnet wird, zugeordnet werden. Die marker benennen zwei Hauptklassen von Merkmalen:

- Art eines Herzschlags
  Man unterscheidet beispielsweise Normalschläge, vorzeitige Herzschläge und Herzschläge mit einem Schenkelblock. Der Name einer Herzschlagart wird in dieser Arbeit im Folgenden als beat marker bezeichnet. Zusätzlich zu einem Namen wird jedem Herzschlag ein Zeitpunkt innerhalb des Signals zugeordnet, zu dem er detektiert wurde. Dieser Zeitpunkt heißt in weiterer Folge beat sample.

- Charakteristischer Punkt eines Herschlags
  Ein Herzschlag hat unterschiedliche charakteristische Punkte. Es gibt unter anderem den Beginn der P-Welle, die R-Zacke oder das Ende der T-Welle. Der Begriff waveform marker steht im Folgenden für den Namen eines charakteristischen Punktes eines Herzschlages. Ähnlich wie bei den Herzschlägen ist auch jedem charakteristischen Punkt ein Zeitpunkt innerhalb des Signals (waveform sample) zugeordnet, zu dem er auftritt.

Aus Gründen der Kürze wird nachfolgend der Begriff "charakteristischer Punkt eines Herzschlages" waveform genannt. Analog ersetzt beat den Begriff "Herzschlag".

1.1.4 Validierung eines EKG-Analysealgorithmus

In weiterer Folge wird zur Erhöhung der Lesbarkeit der Begriff "Entwickler" statt der genauen Bezeichnung "Entwickler eines EKG-Analysealgorithmus" verwendet. Analog wird "EKG-Analysealgorithmus" durch "Algorithmus" ersetzt. Die Gesamtheit aller marker zuzüglich der Zeitpunkte ihres Auftretens, die einem EKG-Signal entweder durch einen Algorithmus oder einen Experten zugeordnet werden kann, wird Algorithmusannotation oder Expertenannotation genannt.

Die Validierung eines Algorithmus soll dem Entwickler anhand des Vergleichs der Expertenannotationen mit den Algorithmusannotationen Aufschluss über die Qualität des Algorithmus geben. Wenn der Algorithmus in der Praxis eingesetzt werden soll, muss er genau definierte Prüfverfahren bestehen, da die Ergebnisse eines Algorithmus

von hoher medizinischer Relevanz sind. Der Standard (ANSI/AMMI EC38:1998) [8] beschreibt solche Prüfverfahren.

Ablauf einer Validierung

Um einen Algorithmus zu validieren, sind vier Schritte notwendig:

1. Aufzeichnung von Testsignalen
   Es ist notwendig Testsignale aufzuzeichnen, die die marker, die der Algorithmus detektieren soll, enthalten.

2. Expertenannotationen hinzufügen
   Ein Experte muss bei den Testsignalen alle marker annotieren. Diese Annotationen werden später als Referenz verwendet. Das easyG beinhaltet ein Annotationswerkzeug, das den Experten bei dieser Arbeit unterstützt (1.3).

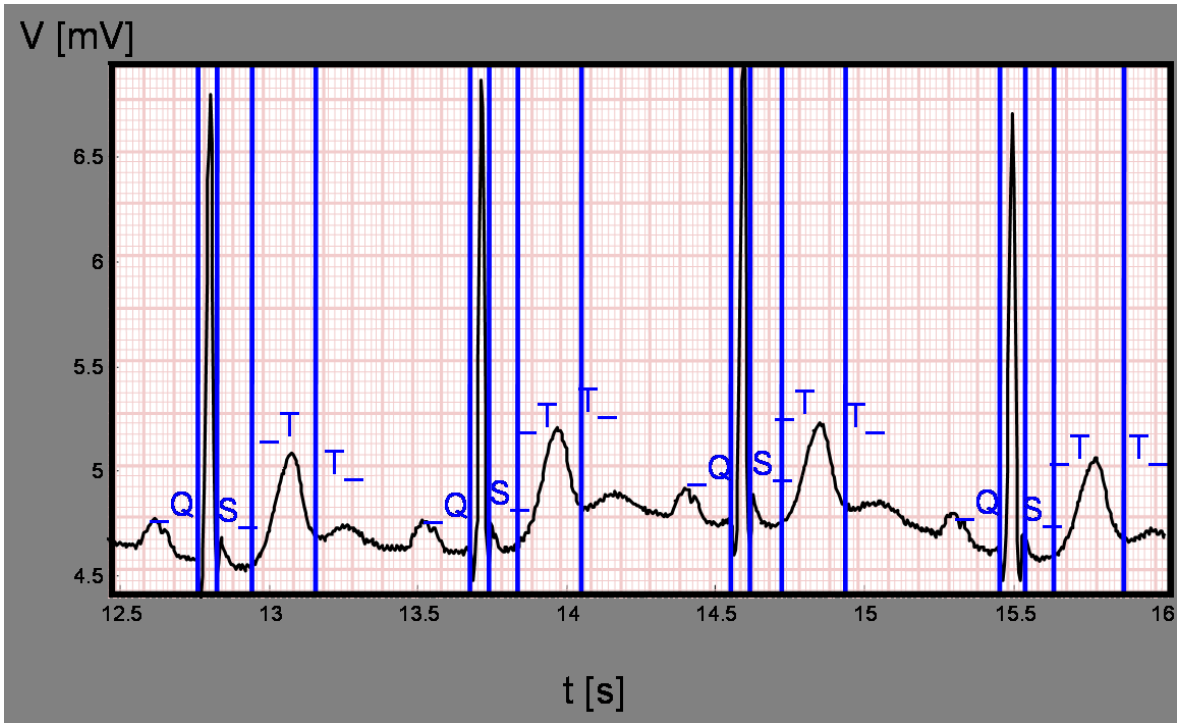


Abbildung 1.3: Das Annotationwerkzeug ermöglicht dem Experten ein Setzen von neuen markern und ein Löschen bzw. Verschieben von gesetzten markern. Diese Abbildung zeigt ein EKG-Signal mit Expertenannotationen für den Beginn des QRS-Komplexes (_Q), das Ende des QRS-Komplexes (S_), den Beginn der T-Welle(_T) und das Ende der T-Welle(T_).

3. Automatische Analyse der Testsignale
   Die Testsignale werden vom Algorithmus analysiert. Dabei wird eine Algorithmusannotation erzeugt.

4. Vergleich
   Die Expertenannotation wird mit der Algorithmusannotation verglichen, um die Qualität des Algorithmus zu bestimmen.

Nachfolgend wird auf die Vorgaben des ANSI/AMMI EC38:1998 [8] eingegangen.

- Auswahl der Testsignale mit zugehörigen Expertenannotationen
  Im ANSI/AMMI EC38:1998 [8] sind fünf Standarddatenbanken für die Validierung vorgesehen:
  - **AHA-DB** - The American Heart Association Database for Evaluation of Ventricular Arrhythmia Detectors (80 Signale, 35 Minuten)
  - **MIT-DB** - The Massachusetts Institute of Technology Beth Israel Hospital Arrhythmia Database (48 Signale, 30 Minuten) [9]
  - **ESC-DB** - The European Society of Cardiology ST-T Database (90 Signale, zwei Stunden) [10]
  - **NST-DB** - The Noise Stress Test Database (12 Signale, 30 Minuten) [11]
  - **CU-DB** - The Creighton University Sustained Ventricular Arrhythmia Database (35 Signale, 8 Minuten) [12]

- Methode der Zuordnung von beat markern
  Die Zuordnung der beat marker des Algorithmus zu denen der Expertenannotation ist im ANSI/AMMI EC38:1998 [8] genau definiert. Der Zuordnungsprozess wird auf Seite 18 ausführlich beschrieben.

- Statistische Methoden
  Die statistischen Methoden zur Erzeugung des Abschlussberichts der Validierung sind im ANSI/AMMI EC38:1998 [8] festgelegt. Einige dieser Methoden, wie zum Beispiel die Sensitivität oder der positive prädiktive Wert werden ab Seite 21, beschrieben.

1.2 easyG - electrocardiogram analysis system Graz

Das **e**lectrocardiogram **a**nalysis **sy**stem **G**raz (easyG) wird seit einigen Jahren vom Team der ARC Seibersdorf research GmbH am Standort Graz, Biosignalverarbeitung und Telemedizin, entwickelt [13] [14] [15]. Es ermöglicht die automatische und interaktive Analyse von EKGs sowie das schnelle Entwickeln neuer EKG-Signalanalysealgorithmen. Abb. 1.4 zeigt überblicksmäßig die verschiedenen Module des easyG. Es besteht aus einer modular aufgebauten Software, die in Matlab 6.5 (MathWorks Inc., Natick, Massachusetts, USA) entwickelt wurde. Von dem System wird remote processing über das Internet unterstützt [16]. Theoretische Untersuchungen über verschiedene Eigenschaften der Erregungsausbreitung können auch mit Hilfe eines Modells zur Simulation der Erregungsausbreitung und der resultierenden EKGs untersucht werden [17]. Im Rahmen dieser Diplomarbeit wurde eine Validierungsumgebung für dieses EKG-Analysesystem entwickelt, die bereits unter dem Namen validation tool in der Systemübersicht (Abb. 1.4) eingezeichnet ist.

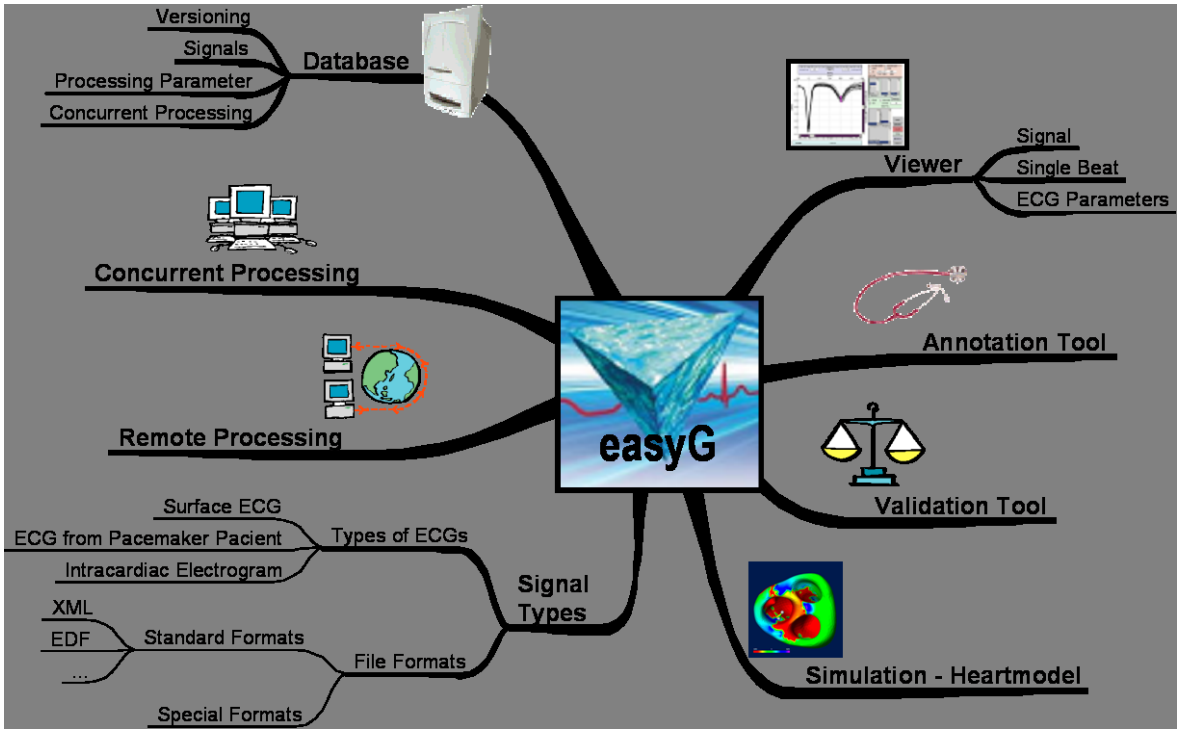


Abbildung 1.4: Systemüberblick zum easyG - electrocardiogram analysis system Graz

1.3 Aufgabenstellung

Das easyG sollte um ein Validierungsumgebung erweitert werden, wobei die folgenden Teil-aufgaben zu lösen sind:

- Es sollte eine komfortable Verwaltung der bei einem Validierungsprozess anfallenden Daten entwickelt werden.
- Eine Methode zum Vergleich von Expertenannotationen mit Algorithmusannotationen sollte implementiert werden.
- Aufbauend auf den Vorgaben des ANSI/AMMI EC38:1998 [8] sollte ein Abschlussbericht der Validierung erstellt werden.
- Der gesamte Validierungsablauf sollte über eine graphische Benutzerschnittstelle gesteuert werden können.

1.4 Umsetzung

Die Validierungsumgebung wurde in die Submodule Annotationsarchiv, Kernmodul und Statistkmodul gegliedert. In Abb. 1.5 wird das Zusammenspiel der einzelnen Teile beschrieben. Der Entwickler kann neue Annotationen zum Annotationsarchiv hinzufügen, ausgewählte Annotationen validieren oder eine statistische Methode auf die Rohdaten der Validierung anwenden. Das Kernmodul benötigt Zugriff auf das Annotationsarchiv und stellt die zugeordneten marker zur Verfügung. Das Statistikmodul beinhaltet Methoden, um die Validierungsergebnisse zu visualisieren bzw. in Berichten zusammenzufassen.

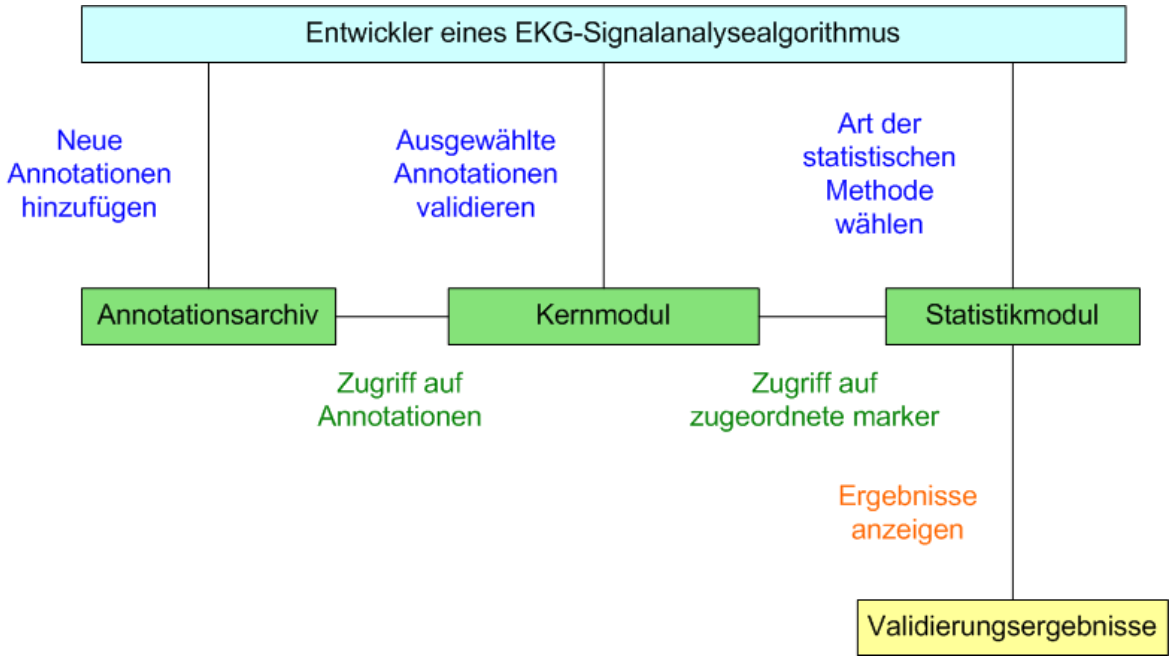


Abbildung 1.5: Gesamtüberblick über die Validierungsumgebung und die Interaktionen zwischen den einzelnen Submodulen.

Im Folgenden wird kurz die Funktion der drei zentralen Module erläutert.

1.4.1 Annotationsarchiv

Zum Zeitpunkt der Niederschrift dieser Arbeit enthält das easyG einige hundert EKG-Signale und etwa ein Dutzend Annotationsverfahren, und es kann davon ausgegangen werden, dass die Anzahl in Zukunft weiter steigen wird. Da jedes Annotationsverfahren auf jedes Signal angewandt werden kann und zu jedem Signal mehrere Expertenannotationen existieren können, muss dem Entwickler eines EKG-Signalanalysealgorithmus

ein komfortables Werkzeug zur Verfügung gestellt werden, mit dem er die anfallenden Annotationen effizient verwalten kann. Aus diesem Grund wurde eine Annotationsdatenbank entwickelt. Zur komfortablen Verwaltung dieser Datenbank wurde eine graphische Benutzerschnittstelle implementiert, die unter anderem Methoden zur Verwaltung der Namen, zum Zusammenfassen ausgewählter Signale zu Signalsammlungen und zum Importieren von Annotationen zur Verfügung stellt.

1.4.2 Kernmodul

Um einen Validierungsvorgang starten zu können, muss der Entwickler mit Hilfe der graphischen Benutzerschnittstelle des Kernmoduls Signale, einen Algorithmus, der validiert werden soll, und einen Experten als Referenz auswählen. Danach werden die Annotationen für die ausgewählten Signale in die Datenbank des Kernmoduls importiert und die Zuordnung der marker wird durchgeführt. Die zugeordneten marker stehen dann für die Auswertung durch das Statistikmodul zur Verfügung.

1.4.3 Statistikmodul

Aus der enormen Datenmenge der zugeordneten marker in der Datenbank des Kernmoduls kann mit Hilfe des Statistikmoduls ein Abschlussbericht für den beat-to-beat-Vergleich erzeugt werden, wie er im ANSI/AMMI EC38:1998 [8] definiert ist. Außerdem wurden Möglichkeiten implementiert, um die zugeordneten marker flexibel filtern zu können.

1.5 Softwareauswahl

Aus der Aufgabenstellung lässt sich der Bedarf an folgenden Softwareprodukten ableiten:

- Datenbank
- Werkzeug zum Erstellen einer graphischen Benutzerschnittstelle
- Statistikprogramm

Datenbank

Da im easyG bereits die relationale Datenbank Interbase (Borland Software Corporation, Scotts Valley, CA, USA) verwendet wird, wurde die Entscheidung getroffen, diese Datenbank auch für das Validierungsmodul zu verwenden.

Werkzeug zum Erstellen einer graphischen Benutzerschnittstelle

Die graphische Benutzerschnittstelle von easyG wurde in Matlab 6.5 (MathWorks, Inc., Natick, MA, USA) realisiert. Dennoch wurde für die Erstellung der graphischen Benutzerschnittstelle beim Validierungsmodul aus folgenden Gründen der Java Technologie (Sun Microsystems Inc., Santa Clara, CA, USA) der Vorzug gegeben:

- Objektorientierte Programmiersprache
- Verfügbarkeit sehr vieler, freier Lösungen für Standardprobleme
- Automatische Dokumentation des Quellcodes mittels javadoc

Da Matlab die Möglichkeit zur Verfügung stellt, Java Programme einzubinden, ist eine Integration der graphischen Benutzerschnittstelle in das easyG möglich.

Statistikprogramm

Für diese Aufgabe wurde Matlab ausgewählt, da es sich hervorragend für statistische Berechnungen und Visualisierungen eignet [18].

2 Methoden

Zu Beginn dieses Kapitels wird ein kurzer Überblick über die verwendete Software gegeben. Danach werden die Methoden beschrieben, die zum Erstellen der relationalen Datenbank des Annotationsarchivs verwendet wurden. Anschließend werden die Verfahren behandelt, die in der Validierungsdatenbank Verwendung finden, um marker zuzuordnen. Weiters werden die statistischen Methoden zum Analysieren der Validierungsergebnisse erläutert. Zuletzt wird beschrieben, wie das Validierungsmodul getestet wurde.

2.1 Verwendete Software

2.1.1 Matlab 6.5

Matlab (Matrix Laboratory) entstand Ende 1970 an den Universitäten von New Mexico und Stanford. Im Laufe der Jahre wurde Matlab aufgrund des positiven Feedbacks vieler Anwender ständig weiterentwickelt. Die aktuellen Versionen beinhalten alle Funktionen eines modernen Mathematik-Softwaresystems. Wegen der leichten Erlernbarkeit und Vielseitigkeit wird Matlab nicht nur im akademischen Bereich, sondern auch kommerziell verwendet. Das Basispaket ist mittels sogenannter Toolboxen auf verschiedene Anwendungsbereiche erweiterbar. Die folgenden beiden Toolboxen wurden bei der Erstellung des Validierungsmoduls verwendet [19]:

- **Database Toolbox**
  Es wird eine Schnittstelle zu ODBC/JDBC-kompatiblen Datenbanken zur Verfügung gestellt. Das Ausführen von SQL-Anweisungen aus Matlab wird ermöglicht. Das Importieren von großen Datenmengen für anschließende Berechnungen in Matlab wird unterstützt [20].

- **Statistics Toolbox**

Es werden unter anderem zahlreiche Methoden für die beschreibende Statistik angeboten. Außerdem werden zahlreiche Visualisierungsmöglichkeiten für die Ergebnisse der statistischen Berechungen zur Verfügung gestellt [18].

2.1.2 Java Technologie

Am 23. Mai 1995 wurde die objektorientierte Programmiersprache Java zum ersten Mal offiziell vorgestellt. In der ersten Hälfte des Jahres 1996 wurde Java 4325-mal in der Presse erwähnt. Bill Gates kam zum Vergleich 5096-mal im selben Zeitraum in der Presse vor. Das enorme öffentliche Interesse an Java ist auf die Methode zurückzuführen, die Portabilität für Java Programme garantiert. Das Java Quellprogramm wird in eine Zwischensprache (bytecode) übersetzt. Der bytecode kann anschließend von einem Interpreter (virtuelle Maschine) auf einem beliebigen Rechner ausgeführt werden. Die virtuelle Maschine ist ein Programm, das in verschiedenen Versionen für viele unterschiedliche Plattformen erhältlich ist. Die folgenden Javapakete wurden bei der Umsetzung der Diplomarbeit verwendet [21] [22] [23]:

- **Apache Ant**
  Im Java-Umfeld ersetzt Apache Ant das make-Werkzeug. Es wird verwendet, um Projektquellcode zusammenzustellen, zu bearbeiten, zu übersetzen und zu starten. Vor der Entwicklung von Ant war man bei der Steuerung eines umfangreichen Java Build-Vorganges auf die Unterstützung einer Entwicklungsumgebung angewiesen. Da Apache Ant in Java programmiert ist, ist dieses Werkzeug plattformunabhängig [24].

- **Log4j**
  Dieses Paket erlaubt ein logging mit verschiedenen Prioritäten (Debug-Meldungen, Informationen, Warnungen und Fehler). Die Debug-Meldungen sind bei der Entwicklung von Programmen eine große Hilfe. Bei der Verwendung des fertigen Programms können dem Benutzer mit Hilfe von Log4j strukturierte Informationsmeldungen zur Verfügung gestellt werden. Sollte es zu einer Fehlfunktion kommen, kann das Problem anhand des loggings der Fehlermeldungen diagnostiziert werden [25].

- **JMI**
  Das Java Matlab Interface (JMI) ist ein Paket, das von Matlab zur Verfügung

gestellt wird [26]. Mit diesem Paket ist es möglich, Matlab Funktionen aus einem Java Programm aufzurufen. Eine Einführung in die Verwendung des Pakets ist auf der Homepage von Kamin Whitehouse zu finden [27].

- **FindAccessory**
  Das FindAccessory Paket wurde von Ken Klinner entwickelt. Es bietet unter anderem die Möglichkeit, in einem Verzeichnis rekursiv nach Dateinamen zu suchen [28].

- **SWING**
  Dieses Paket stellt Standardkomponenten zur Verfügung, die bei einer graphischen Benutzerschnittstelle verwendet werden [29].

2.1.3 Interbase

Das Datenbankmanagementsystem Interbase wurde 1985 entwickelt. 1992 wurde das Produkt von Borland übernommen. Nachfolgend werden die wichtigsten Merkmale dieser relationalen Datenbank aufgelistet [30] [31].

- **Plattformunterstützung**
  Von Interbase werden alle wichtigen Windows und Unix Plattformen unterstützt.

- **Unterstützung von Standards**
  Die folgenden Industriestandards werden von Interbase unterstützt: ANSI SQL 92, JDBC bzw. ODBC Datenbankverbindungen.

- **Entwicklung**
  Für die Datenbankentwicklung werden unter anderem die folgenden Methoden zur Verfügung gestellt: Generatoren, Trigger und stored procedures.

2.2 Annotationsarchiv

In weiterer Folge werden die Methoden beschrieben, die bei der Implementierung des Annotationsarchivs verwendet wurden. Zuerst wird das Erstellen einer relationalen Datenbank besprochen. Danach werden die Verfahren erläutert, die beim Entwickeln der graphischen Benutzerschnittstelle verwendet wurden.

2.2.1 Annotationsdatenbank

Bei der Entwicklung der Annotationsdatenbank wurden die Methoden in folgender chronologischer Reihenfolge verwendet: Am Anfang wurde eine Workflow-Analyse durchgeführt. Mit den dabei gewonnenen Ergebnissen wurde ein logisches Datenmodell entwickelt. Dieses diente als Grundlage für das relationale Datenbankmodell. Das entstandene Modell wurde normalisiert und mittels SQL in ein Format übertragen, das vom gewählten Datenbankmanagementsystem verwendet werden kann [32] [33] [34] [35].

- **Workflow Analyse**
  Die Workflow-Analyse verfolgt das Ziel, alle für die Annotationsdatenbank wichtigen Entitäten zu finden.

- **Logisches Datenbankmodell**
  Entitäten, Beziehungen und Attribute sind die Grundbausteine des logischen Datenbankmodells. Zuerst werden die Entitäten identifiziert, die Personen, Dingen oder Konzepten der realen Welt entsprechen. In weiterer Folge werden den Entitäten Attribute zugeordnet. Diese entsprechen wiederum den Eigenschaften der Personen, Dinge oder Konzepte in der realen Welt. Zuletzt werden die Beziehungen zwischen den Entitäten erfasst.

- **Relationales Datenbankmodell**
  Die Entität des logischen Datenbankmodells entspricht der Relation im relationalen Datenbankmodell. Ein Attribut entspricht einer Spalte der Tabelle und kann Werte aus einer bestimmten Domäne annehmen. Die Beziehungen zwischen den Entitäten des logischen Modells werden im relationalen Datenbankmodell mit Schlüsseln aus zwei Tabellen hergestellt.

- **Relationale Datenbank**
  Eine relationale Datenbank organisiert die Daten in einer Sammlung von Tabellen (Relationen), die eindeutige Namen besitzen. Jede Tabelle hat eine oder mehrere Spalten, die in einer speziellen Reihenfolge von links nach rechts angeordnet sind. In jeder Tabelle gibt es keine, eine oder mehrere Zeilen, die für jede Spalte einen einzigen Wert beinhalten. Die Zeilen haben keine bestimmte Ordnung. Alle Datenwerte in einer bestimmten Spalte haben denselben Datentyp und entstammen dem Definitionsbereich der Spalte. In einer relationalen Datenbank

sind die Tabellen durch die Werte verknüpft, die sie enthalten. Ein relationales Datenbankmodell verwendet primary keys und foreign keys, um die Verbindungen zwischen den Tabellen zu repräsentieren. Ein primary key ist eine Spalte oder eine Kombination von Spalten, deren Werte eine Zeile eindeutig bestimmen. Jede Tabelle hat nur einen primary key. Ein foreign key ist eine Spalte oder eine Spaltenkombination in einer Tabelle, deren Werte einen primary key einer anderen Tabelle darstellen. Eine primary key/foreign key-Kombination erzeugt eine Vater/Kind-Beziehung zwischen den Tabellen, die diese Kombination enthalten.

- **Normalformlehre**
  Die Normalisierung beinhaltet ein Regelwerk, das dem Datenbank-Designer hilft, Anomalien innerhalb eines Datenbankschemas zu beseitigen. Es werden Einfüge-, Lösch- und Updateanomalien unterschieden.

- **SQL**
  Um das erhaltene relationale Datenbankmodell in das Datenbankmanagementsystem zu übertragen, wurde die Structured Query Lanuguage (SQL) verwendet.

2.2.2 GUI der Annotationsdatenbank

Bei der Implementierung der graphischen Benutzerschnittstelle der Annotationsdatenbank wurde die Technologie SWING von Java verwendet. Für die Kommunikation mit der Datenbank wurde auf die JDBC-Technologie zurückgegriffen. Das Tool find accessory wurde verwendet, um Signale auf eine komfortable Art einbinden zu können.

- **Java Database Connectivity (JDBC)**
  Der Industriestandard für eine datenbankunabhängige Verbindung zwischen Java und einer Vielzahl von Datenbanken ist JDBC. Es werden Methoden zur Verfügung gestellt, um eine Verbindung mit einer Datenbank aufzubauen, SQL-Anweisungen zu übermitteln und die Resultate zu verarbeiten [36].

- **Threads**
  Da das Importieren von Annotationen viel Zeit in Anspruch nehmen kann, wurde für diese Aufgabe ein eigener thread verwendet. Threads erlauben die parallele Ausführung von Programmteilen. Daraus ergibt sich der Vorteil, dass die Anzeige des Importfortschritts regelmäßig aktualisiert werden kann [37].

2.3 Kernmodul

Das Kernmodul erlaubt dem Benutzer, Annotationen aus dem Annotationsarchiv auszuwählen und diese in die Datenbank des Kernmoduls zu importieren. In dieser Datenbank findet eine Zuordnung der korrespondierenden marker statt.

2.3.1 GUI des Kernmoduls

Die graphische Benutzerschnittstelle wurde in SWING realisiert und der Zugriff auf die Annotationsdatenbank mittels JDBC ermöglicht.

2.3.2 Datenbank des Kernmoduls

Zusätzlich zu den Methoden, die bereits bei der Annotationsdatenbank beschrieben wurden, kamen bei dieser Datenbank external tables und stored procedures zum Einsatz.

- **External tables**
  Interbase bietet mit den external tables einen effizienten Mechanismus an, um große Datenmengen, die in Textdateien abgespeichert sind, rasch in eine Datenbank zu importieren [35]. Diese Methode wurde verwendet, um einen schnellen Zugriff auf das Annotationarchiv zu ermöglichen.

- **Stored procedures**
  Stored procedures sind kleine Programme, die auf einem Interbase Server gestartet werden können. Zum Programmieren von stored procedures verwendet man eine Sprache, die SQL-Anweisungen und nützliche Erweiterungen wie Schleifen oder Bedingungen enthält [35]. In der Datenbank des Kernmoduls wurden sie angewandt, um eine Zuordnung von markern vornehmen zu können.

2.3.3 Zuordnung korrespondierender beat marker

Anhand der Abb. 2.1 wird die Methode des Auffindens korrespondierender beat marker illustriert. Der Leser wird ausdrücklich darauf hin gewiesen, dass die folgende Beschreibung die Zuordnung der beat marker stark vereinfacht darstellt. Sie soll dem Leser ermöglichen, das Prinzip dieses Vorgangs rasch zu verstehen. Entwickler, die diesen Vorgang implementieren wollen, seien auf den ANSI/AAMI EC38:1998 Standard [8] verwiesen, der eine exakte Definition für die Zuordnung der beat marker enthält.

Um den ersten beat marker der Expertenannotation bzw. der Algorithmusannotation wurde ein Zeitfenster von +/- 150 ms gelegt. Lag innerhalb dieses Zeitfensters zusätzlich ein beat marker der anderen Annotation, kam es zu einer Zuordnung der beiden korrespondierenden beat marker. Lag kein korrespondierender beat marker innerhalb des Zeitfensters, spricht man bei einer Expertenannotation von einer fehlenden Detektion (der Algorithmus hat den beat nicht detektiert) und bei einer Algorithmusannotation von einer extra Detektion (der Algorithmus hat den beat fälschlicherweise annotiert). Danach wurde das Zeitfenster um den nächsten beat marker gelegt. Das Zeitfenster iterierte solange durch die Annotationen, bis bei beiden der letzte beat marker erreicht wurde.

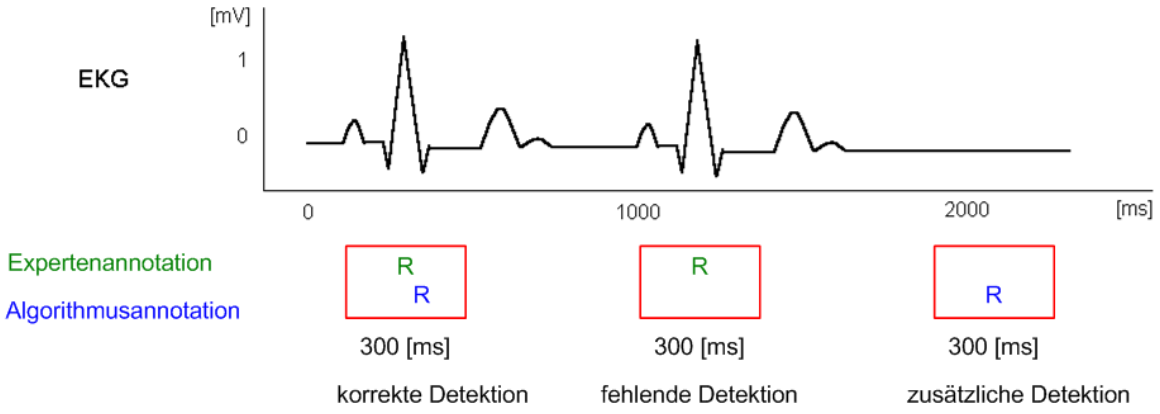


Abbildung 2.1: Links sieht man eine korrekte Detektion, weil die R-Zacke der Expertenannotation und die R-Zacke der Algorithmusannotation innerhalb eines Zeitfensters von +/- 150 ms liegen. In der Mitte hat der Algorithmus die R-Zacke nicht detektiert. Rechts erkennt man eine fälschliche R-Zacken-Detektion des Algorithmus.

Die drei möglichen Fälle bei der beat marker-Zuordnung

- **Korrekte Detektion**
  Der beat marker des Experten und der beat marker des Algorithmus befinden sich innerhalb eines Zeitfensters von +/- 150 ms. Die beiden beat marker bilden ein korrespondierendes Paar.

- **Fehlende Detektion**
  Innerhalb des Zeitfensters um einen beat marker des Experten befindet sich kein beat marker des Algorithmus. Der beat wurde vom Algorithmus nicht detektiert.

- **Extra Detektion**
  Innerhalb des Zeitfensters um einen beat marker des Algorithmus kann kein beat marker des Experten gefunden werden. Der Algorithmus hat fehlerhaft einen extra beat erkannt.

2.3.4 Zuordnung der waveform marker zu den beat markern

Um eine getrennte Validierung der Herzschlag-Detektion (beat marker) und der Detektion der waveform marker vornehmen zu können, wurden die waveform marker zuerst den entsprechenden beat markern zugeordnet. Für die Validierung wurden danach nur diejenigen waveform marker herangezogen, die einen korrespondierenden beat marker besaßen. Fehler in der waveform marker-Detektion, die auf falsch detektierten beat markern basieren, konnten so erkannt und bei der Validierung der waveform marker vernachlässigt werden.

Auch der Vorgang der Zuordnung der waveform marker zu den beat markern wird nachfolgend mit der Intention, dem Leser das grundsätzliche Prinzip zu erklären, vereinfacht dargestellt. In der Praxis ist es sinnvoll die Größe des Zeitfensters an die waveform marker-Art anzupassen. In Abb. 2.2 wurde für den waveform marker, der den Beginn der Q-Welle annotiert, ein Zeitfenster mit 200 ms gewählt. Für den waveform marker, der das Ende der T-Welle annotiert, wurde ein Zeitfenster mit 500 ms festgelegt.

Die Abb. 2.2 zeigt den Vorgang bei der Zuordnug der waveform marker zu den entsprechenden beat markern. Ähnlich wie bei der Zuordnung der beat marker wurde ein Zeitfenster verwendet, um die waveform marker an die beat marker anzuhängen. Grundsätzlich wurden zwei Arten von waveform markern unterschieden: waveform marker, die vor einem beat marker auftreten müssen und solche, die nach einem beat marker liegen müssen. Bei der Zuordnung gab es drei Möglichkeiten:

- korrespondierendes Paar
  Innerhalb eines Zeitfensters kamen ein beat marker und ein waveform marker vor.
- fehlender waveform marker
  Es existierte ein beat marker ohne entsprechenden waveform marker.
- fehlender waveform marker
  ein waveform marker hatte keinen beat marker.

Die zuzuordnenden waveform und beat marker entstammen immer derselben Annotation (entweder beide Annotationen desselben Experten oder beide Annotationen desselben Algorithmus). Es wurde nicht (wie bei den beat markern) kontrolliert, ob es zu einem waveform marker der Algorithmusannotation einen entsprechenden marker der Referenzannotation gab. Vielmehr wurde versucht, jeden waveform marker einer konkreten Herzaktion, also einem beat marker, zuzuordnen.

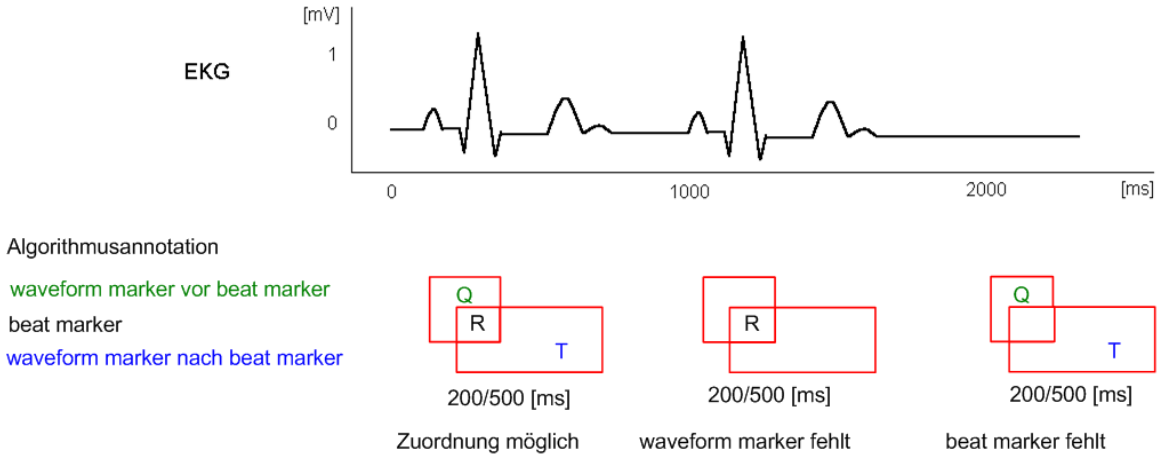


Abbildung 2.2: Diese Abbildung zeigt die Zuordnung der waveform marker zu den beat markern einer Algorithmusannotation. Links sieht man eine Zuordnung der beiden waveform marker Q und T zu dem beat marker R. In der Mitte erkennt man eine R-Zacke mit fehlenden waveform markern. Rechts sind waveform marker ohne korrespondierenden beat marker abgebildet.

2.4 Statistikmodul

Das Statistikmodul beinhaltet Methoden für die Analyse von beat markern und waveform markern. Zunächst wird auf die Analyse der beat marker eingegangen. Danach werden die statistischen Methoden erläutert, die für die Validierung der waveform marker verwendet wurden.

2.4.1 Beat marker

Die Abb. 2.3 zeigt das Endergebnis des beat-to-beat Vergleichs, wie er im ANSI/AMMI EC38:1998 [8] definiert ist. Jedes Element der Matrix entspricht einem Paar von je einem beat marker eines Experten und eines Algorithmus. Die Anzahl des Auftretens eines bestimmten Paares wird in die Matrix eingetragen. Drei verschiedene Bereiche werden unterschieden:

Expertenannotationen	EKG-Signalanalysealgorithmusnannotationen				
	n	s	v	x	
N	Nn	Ns	Nv	Nx	
S	Sn	Ss	Sv	Sx	
V	Vn	Vs	Vv	Vx	
X	Xn	Xs	Xv		

Abbildung 2.3: Diese Tabelle ist das Endergebnis des beat-to-beat Vergleichs. Jedes Element der Matrix entspricht einem Paar eines beat markers von einem Experten und einem Algorithmus. Die grünen Elemente entsprechen den korrekt detektierten beat markern. Die roten Elemente sind fehlende Detektionen des Algorithmus und die blauen Elemente extra Detektionen.

- **Korrespondierende beat marker**
  Die grünen Elemente sind korrespondierende Paare, also beats, die sowohl vom Experten als auch vom Algorithmus detektiert wurden.

- **Fehlende Detektionen**
  Die roten Elemente sind beats, die vom EKG-Signalanalysealgorithmus fälschlicherweise nicht detektiert wurden.

- **Extra Detektionen**
  Die blauen Elemente sind beats, die vom Algorithmus fälschlicherweise zusätzlich detektiert wurden.

Der ANSI/AAMI EC38:1998 Standard sieht weiters die Angabe der Sensitivität und des positiv prädiktiven Wertes vor. In weiterer Folge werden die beiden Begriffe definiert. Anschließend wird die Berechung der beiden Werte für den beat-to-beat Vergleich beschrieben.

Ein Annotationsverfahren kann einen beat erkennen (positiv) oder nicht erkennen (negativ). Die Tabelle 2.4 stellt die Möglichkeiten dar, die beim Vergleich von zwei Annotationsverfahren entstehen können:

	Expertenannotation: positiv	Expertenannotation: negativ
EKG-Algorithmus: positiv	richtig positiv (RP)	falsch positiv (FP)
EKG-Algorithmus: negativ	falsch negativ (FN)	richtig negativ (RN)

Abbildung 2.4: Beim Vergleich von zwei Diagnosemethoden gibt es vier mögliche Ausgänge. RP, FP, RN und FN stellen die Anzahl der Ausgänge in den entsprechenden Klassen dar.

Sensitivität:

Definition allgemein: $Sens = \frac{RP}{RP+FN}$
Definition für den beat-to-beat Vergleich: $Sens = \frac{Korrespondierende\ beats}{Korrespondiernde\ beats\ +\ Fehlende\ Detektionen}$

Positiv prädiktiver Wert:

Definition allgemein: $PPW = \frac{RP}{RP+FP}$
Definition für den beat-to-beat Vergleich: $PPW = \frac{Korrespondierende\ beats}{Korrespondiernde\ beats\ +\ Extra\ Detektionen}$

2.4.2 Waveform marker

Zur Analyse der waveform marker wurden die folgenden statistischen Methoden verwendet. Es wird darauf hingewiesen, dass es sich bei den angegebenen Methoden nur um eine kleine Auswahl der möglichen statistischen Analysemethoden handelt. Die angegebenen Methoden sind sehr allgemein und können für die Analyse aller waveform marker herangezogen werden. Das Statistikmodul wurde so konzipiert, dass spezielle hochentwickelte Analysemethoden einfach integriert werden können [38] [39].

- **Histogramm**

  Eine gebräuchliche Form der Häufigkeitsverteilungsdarstellung ist das Histogramm oder Balkendiagramm. Unter Häufigkeitsverteilung versteht man die Zuordnung von Häufigkeiten zu den Merkmalsausprägungen. Dabei wird die Höhe jedes Balkens von der Anzahl an Merkmalsausprägungen bestimmt, die in dem Intervall vorkommen, das dem Balken zugeordnet ist.

- **Empirische Verteilung**

  Stellt die kumulierten relativen Häufigkeiten dar. Jedem Wert auf der X-Achse ist ein Wert zugeordnet, der angibt, wieviele Prozent der Merkmalswerte kleiner sind als der Wert auf der X-Achse.

- **Arithmetischer Mittelwert**

  Der arithmetische Mittelwert ist eine charakteristische Größe einer Normalverteilung. Er gibt Auskunft über die durchschnittliche Lage eines Messpunktes.
  Definition: $\overline{x} = \frac{\sum x}{n}$ $(x \text{ ... } Messwert, n \text{ ... } Anzahl\ der\ Messwerte)$

- **Standardabweichung**

  Die Standardabweichung gibt bei einer Normalverteilung die Streuung um den arithmetischen Mittelwert an.
  Definition: $s = \sqrt[2]{\frac{\sum x^2 - (\sum x)^2}{n(n-1)}}$ $(x \text{ ... } Messwert, n \text{ ... } Anzahl\ der\ Messwerte)$

- **Median**

  Der Median ist derjenige Wert in der nach Größe der Einzelwerte geordneten Reihe, der die Reihe halbiert. Bei Verdacht auf Ausreißer, bei zuwenig Meßwerten oder bei einer asymmetrischen Verteilung ist der Median dem arithmetischen Mittel vorzuziehen, da er für derartige Verteilungen ein robusteres Maß bietet.
  Definition: $\tilde{x} = \begin{cases} x_{m+1}, & falls\ n = 2m+1 \\ \frac{x_{m+1}+x_m}{2}, & falls\ n = 2m. \end{cases}$
  $(x \text{ ... } Messwert, n \text{ ... } Anzahl\ der\ Messwerte,\ m \text{ ... } Ganzzahl)$

- **Interquartil Bereich**

  Eine nach der Größe der Einzelwerte sortierte Reihe wird von drei Werten in vier gleiche Teile geteilt. Der zentrale dieser drei Werte ist der Median, die beiden anderen bezeichnet man als unteres Quartil (Q_1) und oberes Quartil (Q_3).
  Definition: $I_{50} = Q_3 - Q_1$

Boxplot

Abb. 2.5 beschreibt einen Boxplot, der mit Matlab erzeugt wurde. Der Boxplot dient zur graphischen Darstellung von Verteilungen. Seine wichtigsten Merkmale werden in der folgenden Liste definiert [18]:

- **Unteres und oberes Quartil**
  Die obere blaue Linie der Box ist auf der Höhe des oberen Quartils. Die untere blaue Linie der Box steht für das untere Quartil.
- **Median**
  Die rote Linie in der Mitte der Box gibt den Median an.
- **Einkerbungen**
  Die blauen Einkerbungen sind graphische Konfidenzintervalle des Medians.
- **Ausreißer**
  Ausreißer sind Werte, die weiter als das Eineinhalbfache des Interquartilbereichs von der Oberkante der Box oder der Unterkante der Box entfernt sind. Sie werden als rote Pluszeichen dargestellt.
- **Maximum und Minimum**
  Der schwarze Strich über der Box gibt das Maximum der Verteilung ohne Berücksichtigung der Ausreißer an. Der untere Strich steht für das Minimum der Verteilung ohne Berücksichtigung der Ausreißer.

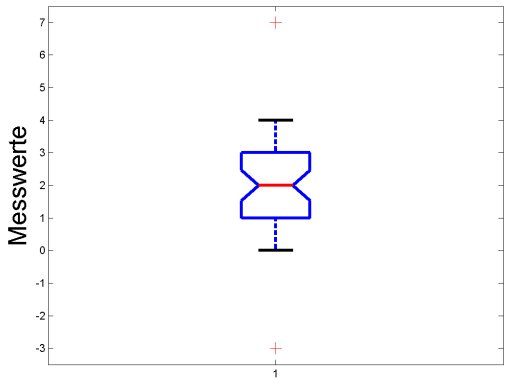


Abbildung 2.5: Boxplot

2.5 Testen des Validierungsmoduls

Um die korrekte Funktionsweise des Validierungsmoduls testen zu können, wurden die Validierungsergebnisse, die mit dem Validierungsmodul berechnet wurden, mit Referenzvalidierungsergebnissen verglichen.

Die Ergebnisse der beat marker-Detektion der ARCS-Rohanalyse wurden anhand der frei verfügbaren Signale der MIT Arrhythmia Datenbank [9] auf zwei unterschiedliche Arten validiert: Einmal mit der im Rahmen dieser Diplomarbeit neu entwickelten Validierungsumgebung, und einmal mit einer bereits existierenden Funktion, die speziell für die Validierung der ARCS-Rohanalyse mit diesen Signalen geschrieben wurde. Die beiden Validierungsergebnisse wurden anschließend verglichen und die Unterschiede analysiert.

Analog wurde die waveform marker-Detektion des ARCS-QT-Algorithmus [15] anhand der ARCS-ICE-Datenbank[1] auf zwei unterschiedliche Arten validiert und die Unterschiede der Validierungsergebnisse wurden analysiert.

[1] Austrian Research Centers Seibersdorf - Intra Cardiac Electrogram Database: 10 Signale, jeweils ca. 15-25 Minuten lang, aufgezeichnet mit einer Abtastrate von 4000Hz. Zur Aufzeichnung der Signale wurden fünf Oberflächenelektroden (I, II, III, V4, V5) und Elektroden innerhalb des Atriums und des Ventrikels verwendet. Für jedes Signal existieren Referenzannotationen für alle atrialen und ventrikulären beat marker bzw. waveform marker für den Beginn des QRS-Komplexes und das Ende der T-Welle

3 Ergebnisse

Am Beginn dieses Kapitels wird die Umsetzung des Validierungsmoduls beschrieben. Auf der nächsten Seite (Abb. 3.1) soll dem Leser ein Gesamtüberblick über das System vermittelt werden. Danach wird die Implementierung der Submodule einzeln erklärt. Zuerst wird die Realisierung des Annotationsmoduls behandelt. Danach werden die Validierungsdatenbank und das Statistikmodul vorgestellt. Für alle Submodule gilt, dass die graphische Benutzerschnittstelle in Java verwirklicht wurde. Die Datenbanken wurden auf einem Interbase Server entwickelt. Für die statistischen Methoden und die Visualisierung der Endergebnisse wurde Matlab verwendet.

Außerdem werden die Validierungsergebnisse, die mit dieser Validierungsumgebung berechnet wurden, erläutert. Danach werden die Ergebnisse die beim Testen der Validierungsumgebung ermittelt wurden, erklärt. Am Ende des Kapitels wird auf das Laufzeitverhalten eingegangen.

3.1 Systemüberblick

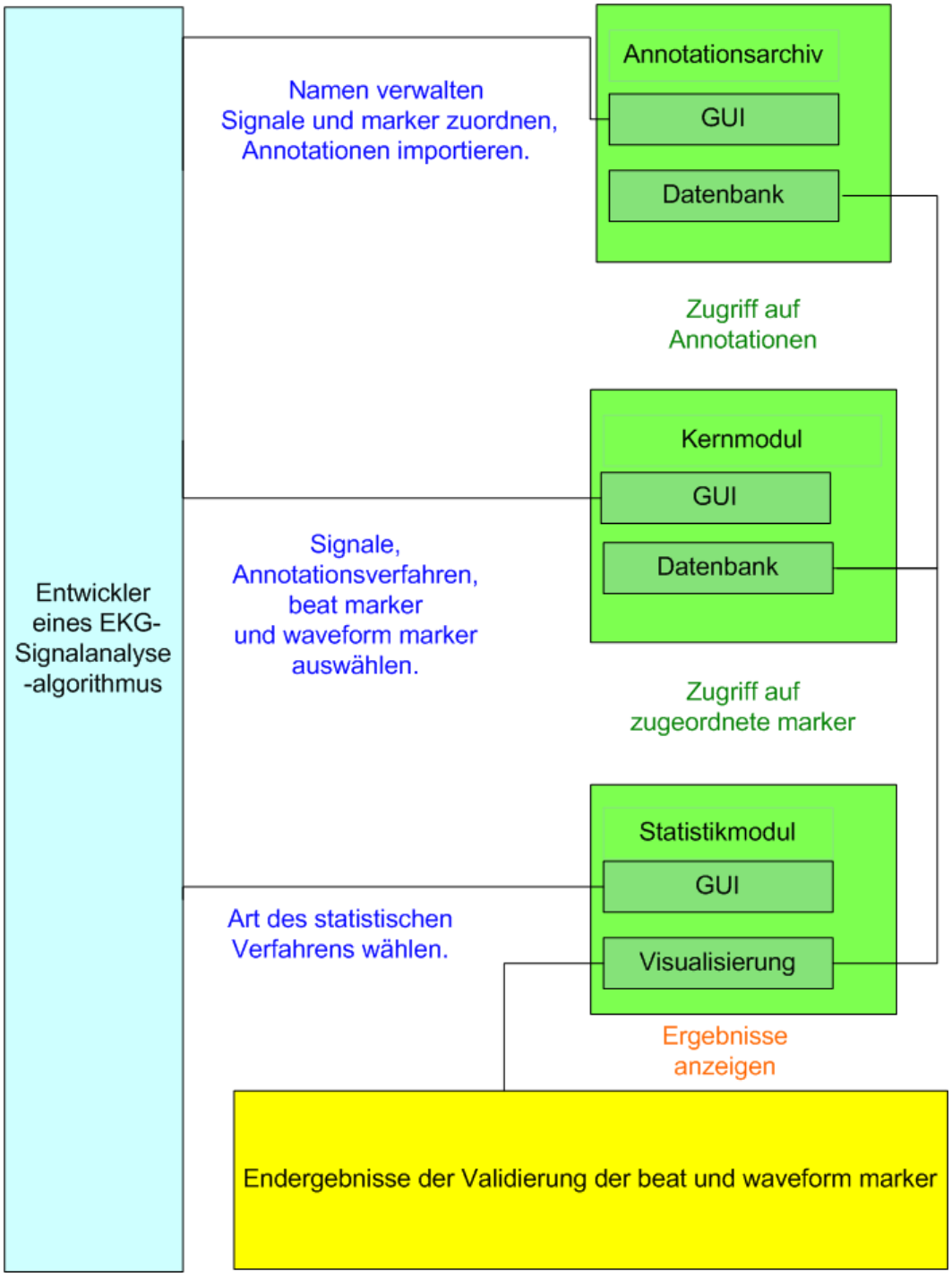


Abbildung 3.1: Gesamtüberblick zur Validierungsumgebung. Die Verteilung der Teilaufgaben auf die Subsysteme wird illustriert.

3.2 Annotationsarchiv

Die Resultate des Datenbankentwurfs werden in folgender Reihenfolge behandelt: Zuerst werden die Ergebnisse der Workflow-Analyse beschrieben. Danach wird das logische Datenbankmodell erklärt. Zum Schluss wird als Endergebnis das relationale Datenbankmodell präsentiert.

3.2.1 Annotationsdatenbank

Workflow-Analyse

Um für die Annotationsdatenbank alle wesentlichen Entitäten zu erhalten, wurde der Entstehungsprozeß von Annotationen mittels Workflow-Analyse (Abb. 3.2) untersucht.

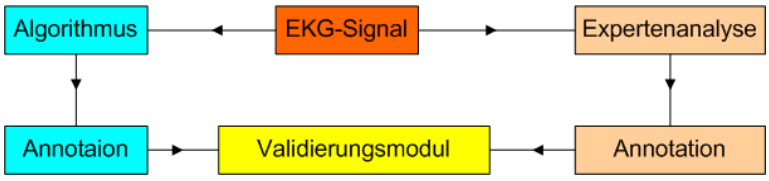


Abbildung 3.2: Die Abbildung zeigt, welche Entitäten beim Erstellen von Annotationen beteiligt sind.

Die folgenden Entitäten kann man direkt aus der Workflow-Analyse ableiten: EKG-Signale, Annotationsverfahren und Annotationen. Weiters ergibt sich die Notwendigkeit, die Signale in Signalsammlungen zu gruppieren, um eine übersichtliche Struktur zu erhalten. Wie in der Einleitung bereits definiert, setzt sich eine Annotation aus markern und den Zeitpunkten ihrer Detektion zusammen. Dadurch kann ein Annotationsmarker als zusätzliche Entität abgeleitet werden. Es folgt eine Zusammenstellung aller Entitäten:

- Annotation
- Annotationsverfahren
- Signal
- Signalsammlung
- Annotationsmarker

Logisches Datenbankmodell

Beim Entwerfen des logischen Datenbankmodells wurden den Entitäten Attribute zugeordnet. Danach wurden die Beziehungen zwischen den Entitäten identifiziert. Das Endergebnis sieht man in Abb. 3.3.

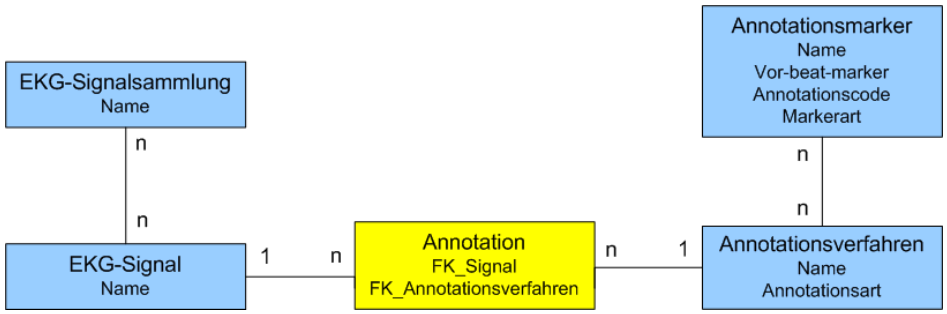


Abbildung 3.3: Das logische Datenbankmodell zeigt die Entitäten mit ihren Attributen. Die Verbindungslinien zwischen den Entitäten geben Auskunft über ihre Beziehung.

Begründungen der Zuordnung von Attributen:

- Annotation

  Jede Annotation wurde mit einem bestimmten Annotationsverfahren erzeugt und bezieht sich auf ein bestimmtes Signal.

- Annotationsverfahren

  Jedes Annotationsverfahren hat einen eindeutigen Namen. Ein Annotationsverfahren kann entweder eine Experten-Analyse oder ein Algorithmus sein.

- Signal

  Jedes Signal hat einen eindeutigen Namen.

- Signalsammlung

  Jede Signalsammlung hat einen eindeutigen Namen.

- Annotationsmarker

  Jeder Annotationsmarker hat einen eindeutigen Namen. Weiters wird jedem marker ein Annotationscode zugeordnet. Dadurch erreicht man, dass der Name eines markers von der Repräsentation in der Annotation unabhängig bleibt. Zum Beispiel ist es sinnvoll, einen normalen Herzschlag immer gleich zu benennen, obwohl

es unterschiedliche Codierungen in den spezifischen Annotationsformaten gibt. Ein waveform marker kann vor einem beat marker oder danach auftreten. Man unterscheidet zwei Markerarten: waveform marker und beat marker.

Begründungen für die identifizierten Beziehungen:

- Annotation n $\Longleftrightarrow$ 1 Annotationsverfahren
  Jede Annotation wurde mit einem bestimmten Annotationsverfahren erzeugt. Mit einem Annotationsverfahren können beliebig viele Annotationen erzeugt werden.

- Signalsammlung n $\Longleftrightarrow$ n Signal
  Jede Signalsammlung setzt sich aus mehreren Signalen zusammen und jedes Signal kann mehreren Signalsammlungen zugeordnet werden.

- Signal 1 $\Longleftrightarrow$ n Annotation
  Jedes Signal kann beliebig oft annotiert werden. Jede Annotation ist eindeutig einem Signal zugeordnet.

- Annotationsverfahren n $\Longleftrightarrow$ n Annotationsmarker
  Jedes Annotationsverfahren analysiert eine beliebige Anzahl von Annotationsmarkern. Jeder Annotationsmarker kann bei beliebig vielen Annotationsverfahren verwendet werden.

Relationales Datenbankmodell

Aus dem logischen Datenbankmodell wurde mit Hilfe der Normalisierungsmethoden das relationale Datenbankmodell (Abb. 3.4) gewonnen.

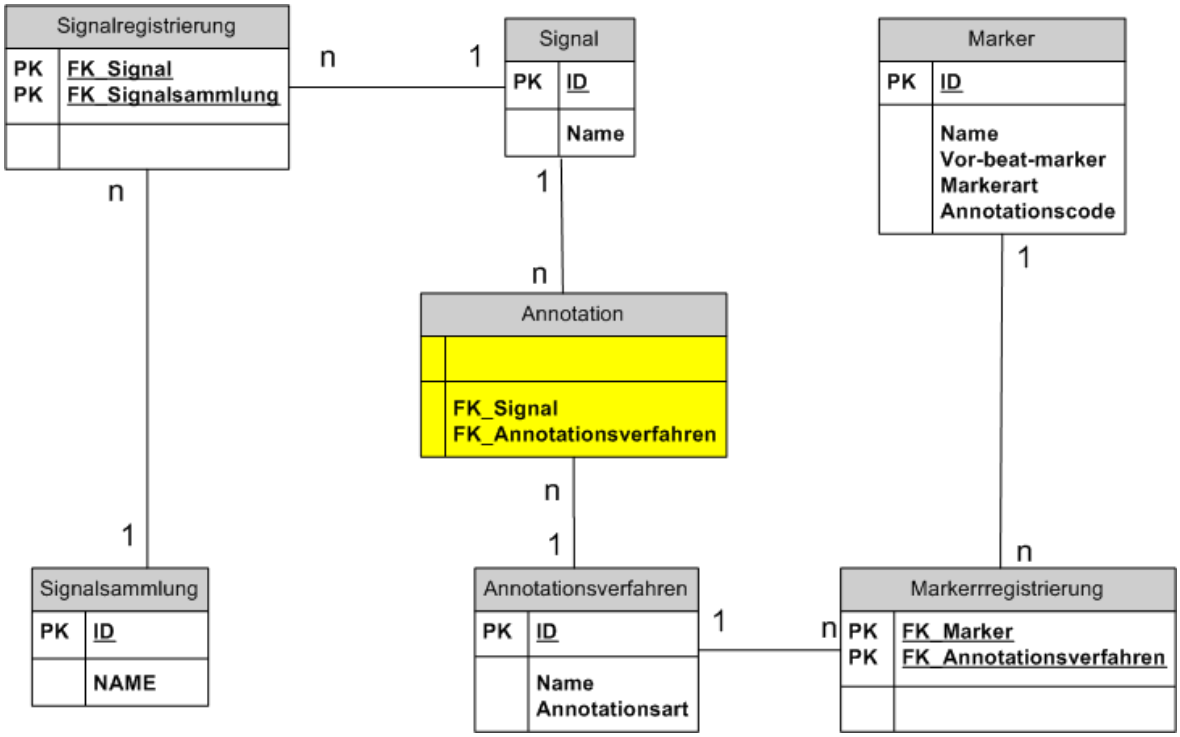


Abbildung 3.4: Die Abbildung zeigt die wichtigsten Tabellen des relationalen Datenbankmodells der Annotationsdatenbank als Endergebnis des Datenbankentwurfs.

3.2.2 GUI der Annotationsdatenbank

Die graphische Benutzerschnittstelle der Annotationsdatenbank besteht aus einem Namenseditor, einem Werkzeug zum Festlegen von Beziehungen zwischen Entitäten und einer Oberfläche, die es dem Entwickler erlaubt, neue Annotationen in das Annotationsarchiv zu integrieren.

Namenseditor

Der Entwickler kann dem Annotationsarchiv neue Signalsammlungen, Annotationsverfahren und waveform bzw. beat marker hinzufügen. Alle Entitäten, die dem Annotationsarchiv hinzugefügt wurden, können nachträglich entfernt oder umbenannt werden (Abb. 3.5).

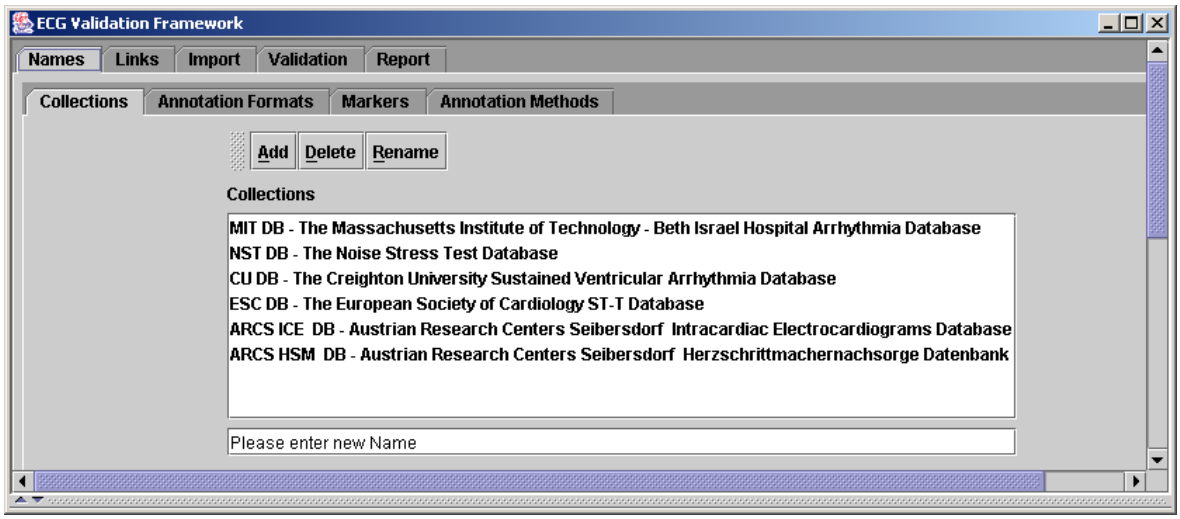


Abbildung 3.5: Diese Abbildung zeigt die Signalsammlungen (collections), die momentan im Annotationsarchiv enthalten sind. Es sind einige Standarddatenbanken (MIT, NST, CU, ESC) zu sehen, die im ANSI/AMMI EC38:1998 [8] für die Validierung vorgesehen sind. Die Datenbanken am Ende der Liste wurden von den Austrian Research Centers zur Herzschrittmachernachsorge [16] und zur Analyse von intrakardialen Elektrogrammen angelegt.

Herstellung von Beziehungen zwischen Entitäten

Zu den mit dem Namenseditor angelegten Signalsammlungen können vom Entwickler Signale hinzugefügt werden. Dazu steht ein Werkzeug zur Verfügung, das es erlaubt, in Verzeichnissen nach mehreren Kriterien rekursiv nach neuen Signalen zu suchen [28]. Weiters kann der Entwickler einem bestimmten Annotationsverfahren marker zuordnen. Beim Zuordnen von markern muss angegeben werden, ob es sich um einen beat marker oder einen waveform marker handelt. Bei waveform markern muss zusätzlich angegeben werden, ob sie sich vor oder nach einem beat marker befinden. Jedem marker muss ein matlab code zugewiesen werden, der angibt, wie der marker im entsprechenden Annotationsformat codiert ist (Abb. 3.6).

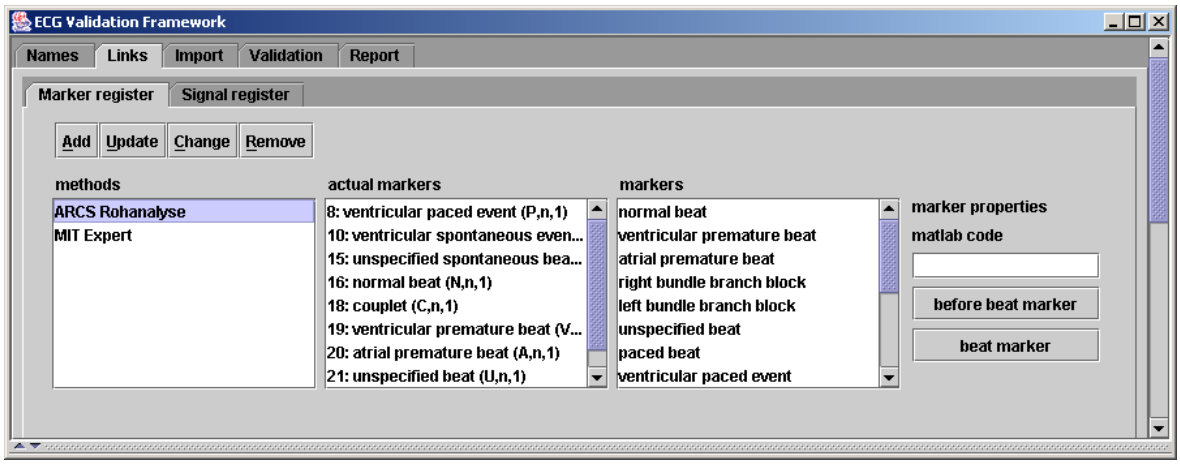


Abbildung 3.6: In der linken Liste sieht man zwei Annotationsverfahren. In der Mitte sind alle marker aufgelistet, die der ARCS-Rohanalyse zugeordnet wurden. Auf der rechten Seite ist die Tabelle mit allen zur Verfügung stehenden markern abgebildet.

Importieren von Annotationen

Wenn der Entwickler eine Signalsammlung auswählt, werden alle Signale angezeigt, die bei dieser Signalsammlung registriert sind. Nachdem der Entwickler ein Annotationsverfahren ausgewählt hat, werden für alle Signale der selektierten Signalsammlung die entsprechenden Annotationen importiert. Die Annotationsdaten werden danach strukturiert im Dateisystem abgelegt und in der Annotationsdatei wird ein Verweis auf die Daten gespeichert. Um dem Entwickler den aktuellen Importstatus visualisieren zu können, wurde ein Fortschrittsbalken implementiert.

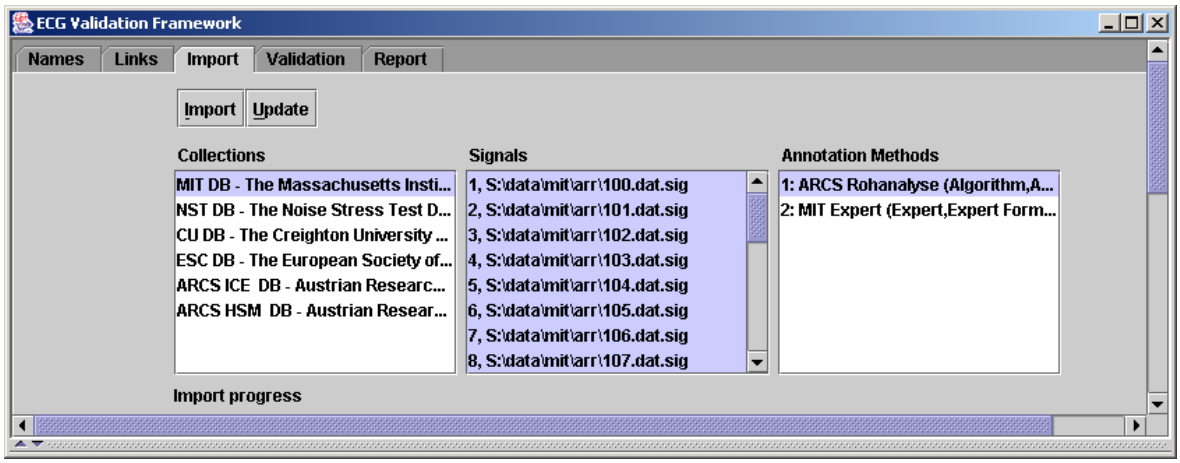


Abbildung 3.7: Die Abbildung zeigt, dass für alle Signale, die der MIT DB zugeordnet sind, die Annotationen, die durch die ARCS-Rohanalyse erzeugt wurden, importiert werden.

3.3 Kernmodul

Das Kernmodul umfasst eine graphische Benutzerschnittstelle, die es dem Entwickler ermöglicht, Annotationen auszuwählen. Die Zuordnung der korrespondierenden marker dieser Annotationen wird anschließend in der Datenbank des Kernmoduls vorgenommen.

3.3.1 GUI des Kernmoduls

Der Entwickler gibt an, welche Signalsammlung er validieren möchte. Danach werden die Signale und die Annotationsverfahren angezeigt, für die Annotationen im Annotationsarchiv zur Verfügung stehen. Sobald der Entwickler ein Annotationsverfahren bestimmt hat, bekommt er alle marker angezeigt, die diesem Annotationsverfahren zugeordnet wurden. Nachdem der Entwickler alle marker, die er in den Validierungsprozess einbeziehen will, ausgewählt hat, kann er die Validierung starten. Sollte der Entwickler für die Zuordnung der beat marker abweichend vom ANSI/AMMI EC38:1998 [8] ein spezielles Zeitfenster benötigen, kann er es mit einem Schieberegler einstellen.

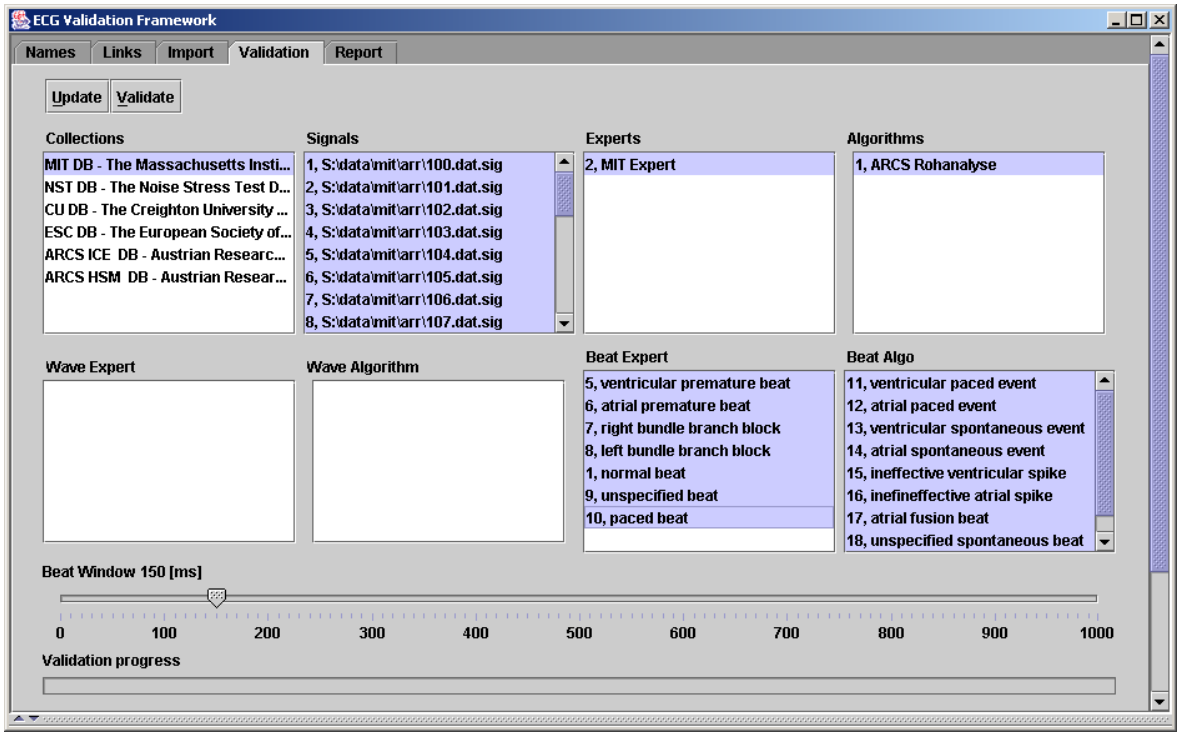


Abbildung 3.8: Die Signale der MIT-DB wurden für einen Vergleich der Annotationen, die mit der ARCS-Rohanalyse bzw. vom MIT-Experten erzeugt wurden, ausgewählt.

3.3.2 Datenbank

In der Datenbank des Kernmoduls erfolgt die Zuordnung korrespondierender beat marker und die Zuordnung von waveform markern zu beat markern. Die zugeordneten marker werden anschließend strukturiert in der Datenbank abgelegt.

Zuordnung der korrespondierenden beat marker

Die Zuordnung der korrespondierenden beat marker erfolgt für jedes EKG-Signal getrennt. Zuerst müssen die Expertenannotationen und die Algorithmusannotationen aus dem Annotationsarchiv importiert werden. Um einen raschen Import zu ermöglichen, werden externe Tabellen verwendet, um die entsprechenden Annotationsdaten aus dem Dateisystem zu importieren. Danach erfolgt die Zuordnug der korrespondierenden beats durch eine stored procedure. Die Struktur der Datenbank wird in Abb. 3.9 dargestellt.

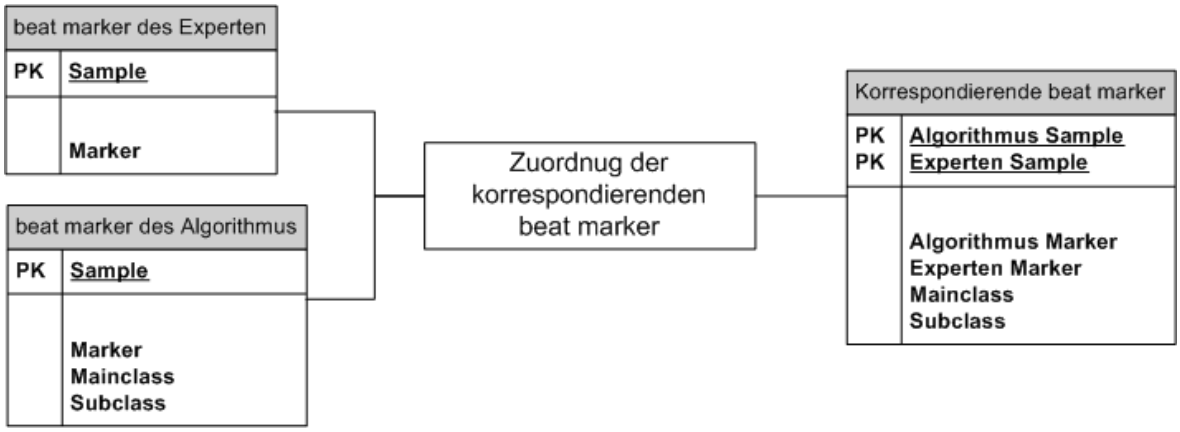


Abbildung 3.9: Die korrespondierenden beat marker der Expertenannotation und der Algorithmusannotation werden mittels stored procedure ermittelt und die dabei gewonnenen Ergebnisse in einer eigenen Tabelle abgespeichert.

Zuordnung der waveform marker zu den beat markern

Die Zuordnung der waveform marker zu den beat markern erfolgt separat für die Expertenannotationen und die Algorithmusannotationen. Abb. 3.10 zeigt den Vorgang für die Expertenannotationen. Zuerst werden die Annotationsdaten mit Hilfe externer Tabellen aus dem Annotationsarchiv importiert. Danach erfolgt die Zuordnung der waveform marker zu den beat markern. Da der Vorgang bei den Algorithmusannotationen analog verläuft, wurde auf eine eigene Darstellung verzichtet.

Abbildung 3.10: Diese Abbildung zeigt den Prozess der Zuordnung von waveform markern zu den beat markern einer Expertenannotation mittels stored procedure in der Validierungsdatenbank.

Archivierung der zugeordneten marker

Die zugeordneten marker eines Signals werden anschließend in der Datenbank des Kernmoduls archiviert. Dabei wird den zugeordneten markern der entsprechende Algorithmus, der entsprechende Experte und das entsprechende Signal zugeordnet. Auf diese Daten greift anschließend das Statistikmodul zu, um die Endergebnisse der Validierung für die waveform marker bzw. die beat marker zu berechnen.

3.4 Statistikmodul

Die Möglichkeiten der statistischen Analyse für die beat marker und waveform marker werden in diesem Kapitel vorgestellt.

3.4.1 Beat marker-Statistik

Dem Entwickler werden die Endergebnisse des beat-to-beat-Vergleichs für jedes EKG-Signal, das in der Validierungsdatenbank enthalten ist, getrennt in einer Tabelle zur Verfügung gestellt. Mit einem Schieberegler hat der Entwickler die Möglichkeit das Signal, das ihn interessiert, auszuwählen. (Abb.3.11). Zusammenfassend wird für jedes Signal die Sensitivität und der positive prädiktive Wert angezeigt.

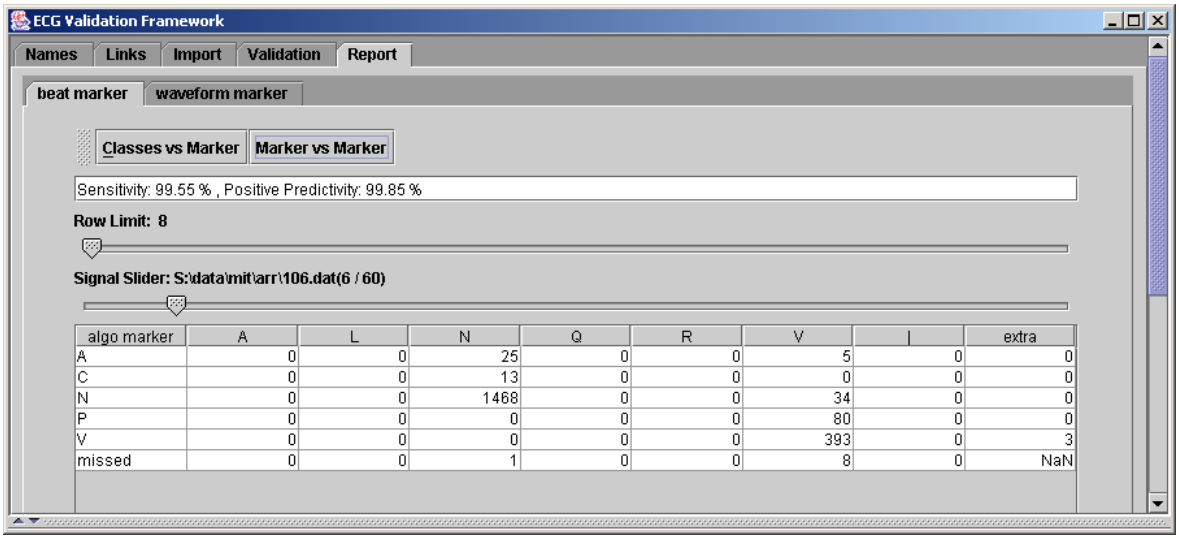


Abbildung 3.11: Die Abbildung zeigt die Endergebnisse des beat-to-beat-Vergleichs von einem Signal, das von einem MIT-Experten und der ARCS-Rohanalyse annotiert wurde. In der ersten Spalte sieht man die beat marker der Rohanalyse und in der ersten Zeile stehen die beat marker, die der MIT-Experte in seiner Annotation verwendet hat. Jede Zahl steht für die Anzahl der korrespondierenden Paare der beat marker in der entsprechenden Zeile bzw. Spalte

3.4.2 Waveform marker-Statistik

Die graphische Schnittstelle für die waveform marker Statistik ermöglicht dem Entwickler folgende Interaktionsmöglichkeiten:

Filterung der Rohdaten der Validierungsergebnisse

Filtern bedeutet, dass der Entwickler eine Untermenge der Rohdaten der Validierungsergebnisse auswählen kann. Es gibt acht Kriterien, nach denen gefiltert werden kann Abb. 3.12:

- Hauptklassen
  Die ARCS-Rohanalyse klassifiziert jeden beat auf Grund seiner Morphologie und ordnet ihm eine Hauptklasse zu.
- Signale
- Experten
- Algorithmen
- beat marker des Experten
- waveform marker des Experten
- beat marker des Algorithmus
- waveform marker des Algorithmus

Es gibt acht Listen (Abb. 3.12), die den acht Kriterien entsprechen. Der Entwickler kann in jeder Liste beliebig viele Elemente auswählen, nach denen er filtern möchte.

Der Entwickler kann zum Beispiel die Filtereinstellungen so wählen, dass er die T-Wellen-Detektion des ARCS-QT-Algorithmus bei allen verfügbaren Signalen der ARCS-ICE-Datenbank mit den jeweiligen Expertenannotationen bei allen normalen Herzschlägen vergleichen kann.

Gruppierung der Rohdaten der Validierungsergebnisse

Wenn in einer Liste mehr als ein Element ausgewählt wurde, kann der Entwickler nach diesen Elementen gruppieren. Gruppieren bedeutet, dass die statistische Auswertung der Validierungsergebnisse für jedes der markierten Elemente der entsprechenden Liste getrennt erfolgt. Daraus entstehen so viele Untermengen der Rohdaten wie Elemente aus der Liste ausgewählt wurden.

Wenn die Einstellungen, die beim obigen Beispiel des Filterns gewählt wurden, so abgeändert werden, dass zusätzlich nach Signalen gruppiert wird, erhält der Entwickler für jedes ausgewählte Signal eine eigene Menge von Ergebnissen. Mit dieser Gruppierungseinstellung kann er analysieren, ob die T-Wellen-Detektion des ARCS-QT-Algorithmus [15] bei allen Signalen der ARCS-ICE-Datenbank gleich gut funktioniert bzw. bei welchen Signalen Probleme aufgetreten sind.

Auswahl eines statistischen Verfahrens

Nachdem der Entwickler seine Filter- und Gruppierungseinstellungen vorgenommen hat, kann er derzeit die folgenden statistischen Methoden auf die entstandenen Gruppen anwenden.

- Boxplots (Abb. 3.13)
- Arithmetischer Mittelwert +/- Standardabweichung
- Median +/- Interquartilbereich
- Empirische Verteilung (nur für eine Gruppe möglich) (Abb. 3.14)
- Histogramm (nur für eine Gruppe möglich) (Abb. 3.15)

Das Statistikmodul wurde so konzipiert, dass eine Erweiterung um weitere statistische Verfahren mit sehr geringem Aufwand jederzeit möglich ist.

Abb. 3.12 zeigt die graphische Benutzerschnittstelle, die es dem Entwickler ermöglicht zu filtern, zu gruppieren und statistische Methoden auszuwählen. Mit den dargestellten Einstellungen werden lediglich beats der Hauptklassen 1-4 (links oben), die vom Algorithmus als normal beat klassifiziert wurden (rechts unten), validiert. Weiters werden nur die Ergebnisse der vier markierten Signale untersucht (2. Feld von links, oben). Der Vergleich erfolgt zwischen dem ICE-Experten und dem ARCS-QT-Algorithmus (rechts oben). Verglichen werden nur waveform marker des Typs T wave onset (links unten). Die Gruppierung erfolgt nach den vier ausgewählten Signalen. Mit diesen Einstellungen kann untersucht werden, bei welchen dieser Signale der Algorithmus besser bzw. schlechter mit den Expertenannotationen übereinstimmt.

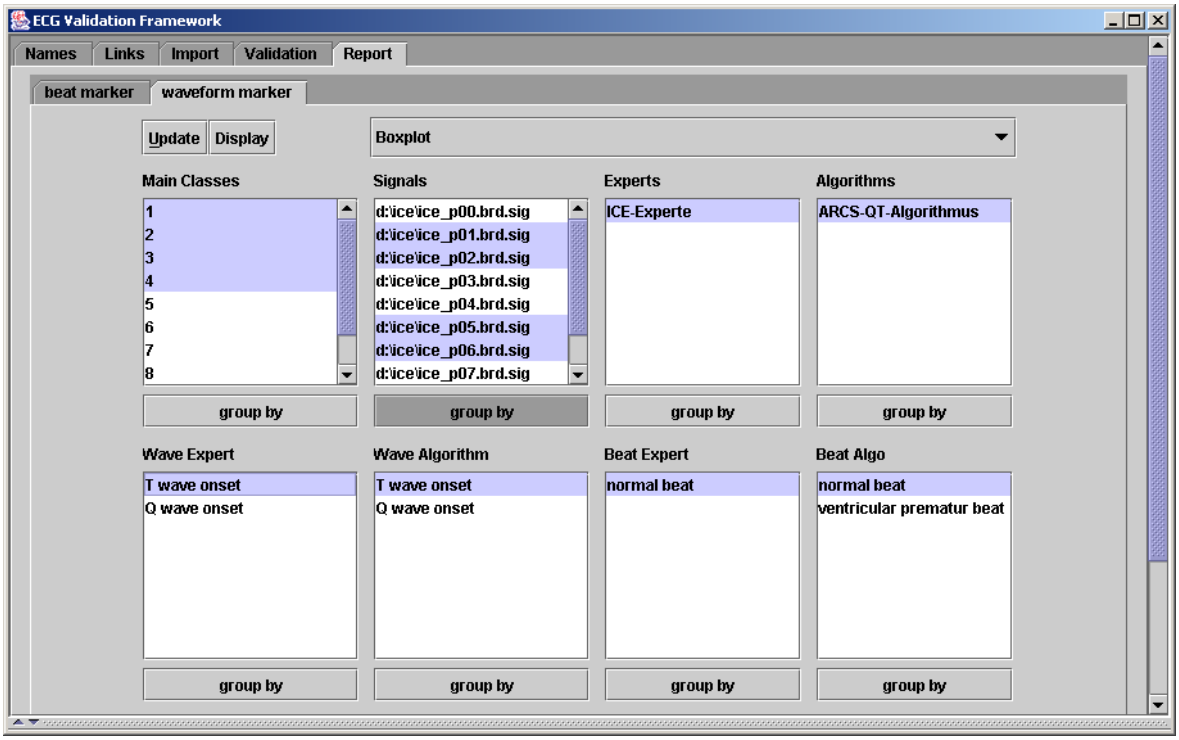


Abbildung 3.12: Filterung, Gruppierung und Auswahl der Statistikmethode für die waveform maker-Statistik.

Boxplots

Abb. 3.13 wurde mit den Einstellungen generiert, die laut Abb. 3.12 vorgenommen wurden. Es wurden vier Boxplots entsprechend den vier Gruppen erzeugt, die jeweils einem Signal entsprechen. Jeder Boxplot illustriert für je eines der ausgewählten Signale die Abweichungen zwischen dem marker für das Ende der T-Welle, wie er vom ARCS-QT-Algorithmus detektiert wurde, und dem korrespondierenden marker des Experten für jeden einzelnen Herzschlag. Der erste Boxplot illustriert für das zweite Signal der ARCS-ICE-Datenbank die Abweichungen zwischen den waveform markern des ARCS-QT-Algorithmus von jenen des ICE-Experten, wobei lediglich waveform marker des Typs 'Ende der T-Welle' und beat marker des Typs 'normal beat' mit einer Hauptklasse von eins bis vier berücksichtigt wurden. Die zweite Gruppe unterscheidet sich von der ersten dadurch, dass nach dem dritten Signal gefiltert wird. Analog gilt für die Gruppe drei und vier, dass nach dem sechsten und siebten Signal gefiltert wurde.

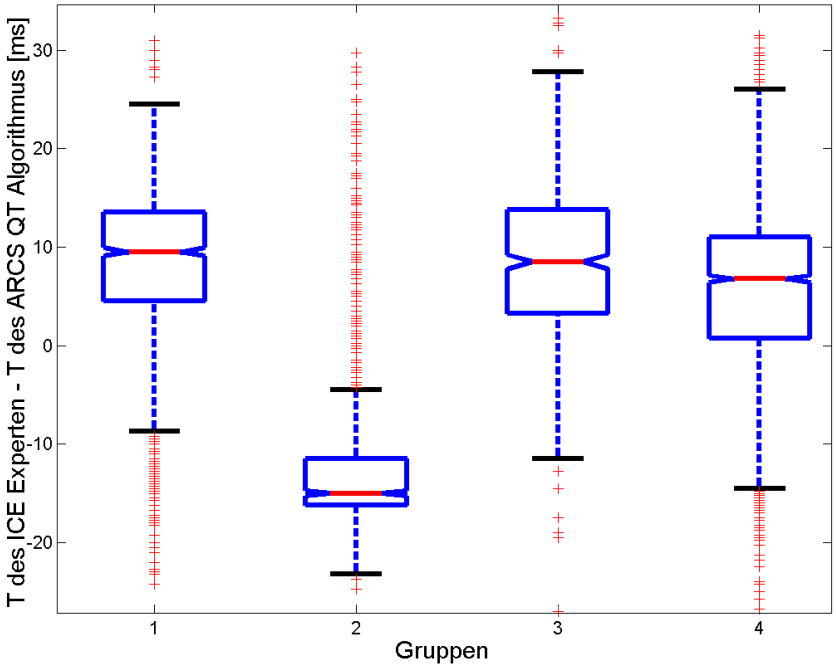


Abbildung 3.13: Die Abbildung stellt die Verteilung der zeitlichen Differenz zwischen jenem Zeitpunkt, zu dem der ICE-Experte das Ende der T-Welle detektiert hat und jenem, zu dem der ARCS-QT-Algorithmus das Ende der T-Welle detektiert hat, für alle waveform marker der vier Gruppen durch Boxplots dar.

Abb. 3.13 zeigt, dass die Detektion des Endes der T-Welle beim zweiten Signal sehr stabil war (geringe Streuung). Allerdings waren die Algorithmusannotationen des Endes der T-Welle absolut gesehen weiter von der Expertenannotation entfernt als bei den anderen Signalen (Median ca. -17ms im Vergleich zu unter 10ms bei den anderen Signalen). Die Signale 1, 3 und 4 haben etwa die gleiche Streuung. Beim zweiten Signal wurde das Ende der T-Welle durchschnittlich etwas später detektiert, als es vom Experten annotiert wurde, während die Detektion bei den übrigen Signalen etwas zu früh war.

Empirische Verteilung

Abb. 3.14 wurde - bis auf die Signalauswahl - mit den gleichen Einstellungen generiert, wie in Abb. 3.12. Statt nach vier Signalen zu gruppieren, wurde nur das zweite Signal aus der Signalliste in Abb. 3.12 ausgewählt.

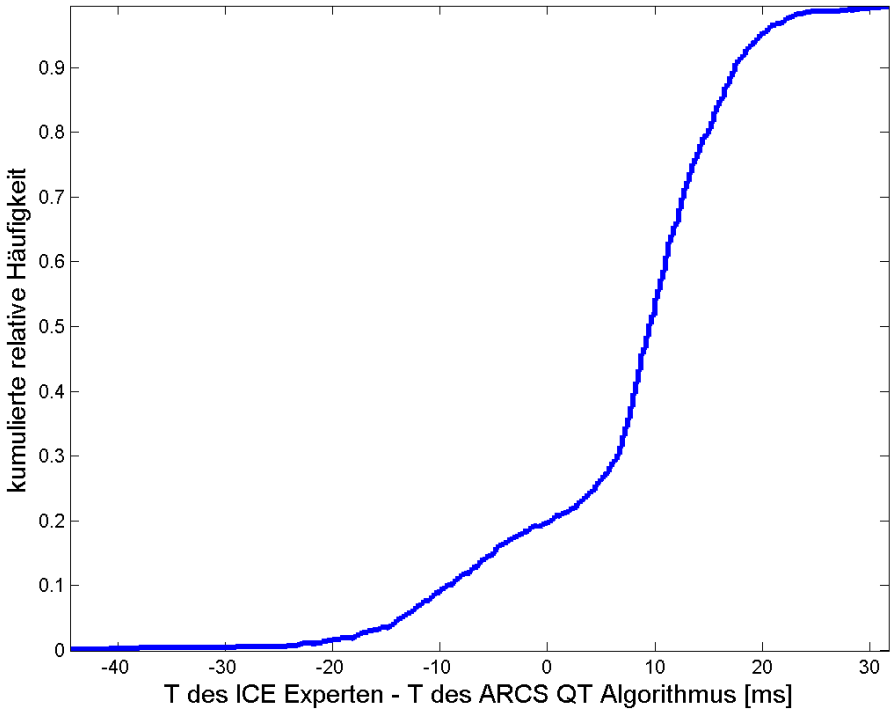


Abbildung 3.14: Die Abbildung zeigt die kumulierte Häufigkeitsverteilung der Differenz zwischen jenem Zeitpunkt, zu dem der ICE-Experte das Ende der T-Welle detektiert hat und jenem, zu dem der ARCS-QT-Algorithmus das Ende der T-Welle detektiert hat, für alle waveform marker.

Histogramm

Auch Abb. 3.15 wurde - bis auf die Signalauswahl - mit den gleichen Einstellungen generiert, wie in Abb. 3.12. Statt nach vier Signalen zu gruppieren, wurde nur das zweite Signal aus der Signalliste in Abb. 3.12 ausgewählt.

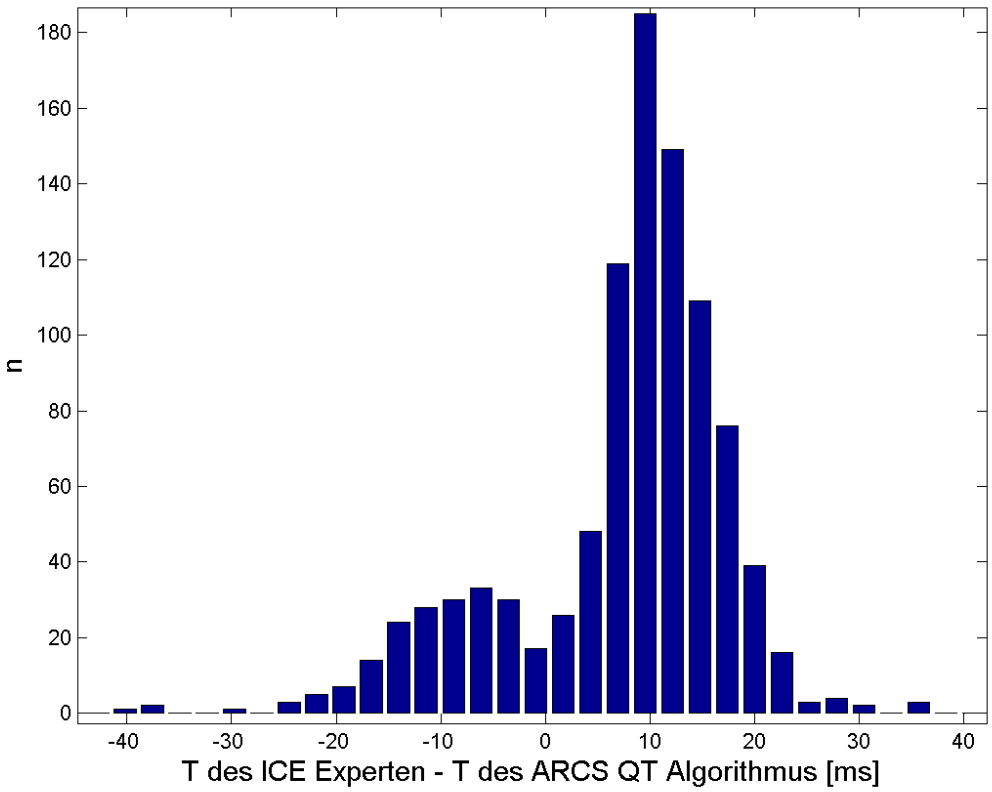


Abbildung 3.15: Die Abbildung zeigt das Histogramm der Differenzen zwischen jenen Zeitpunkten, zu denen der ICE-Experte das Ende der T-Wellen detektiert hat und jenen, zu denen der ARCS-QT-Algorithmus das Ende der T-Wellen detektiert hat, für alle waveform marker.

3.5 Validierungsergebnisse

Um die Praxistauglichkeit der Validierungsumgebung zu demonstrieren, wurde die ARCS-Rohanalyse (beat marker-Detektion) anhand folgender vier Datenbanken validiert.

- **MIT-DB** - The Massachusetts Institute of Technology Beth Israel Hospital Arrhythmia Database [9]
- **ESC-DB** - The European Society of Cardiology ST-T Database [10]
- **NST-DB** - The Noise Stress Test Database [11]
- **CU-DB** - The Creighton University Sustained Ventricular Arrhythmia DB [12]

Die Validierungsergebnisse sind in der Abb. 3.16 zusammengefasst.

Datenbankbezeichnung	Signale	Signaldauer	Sensitivität	PPW
MIT-Datenbank	25	30 Minuten	99,24%	99,22%
ESC-Datenbank	48	2 Stunden	99,44%	99,83%
NST-Datenbank	12	30 Minuten	94,31%	89,23%
CU-Datenbank	35	8 Minuten	92,78%	81,21%

Abbildung 3.16: Endergebnisse der Validierung der ARCS-Rohanalyse.

Laut ANSI/AMMI EC38:1998 [8] müssen die Sensitivität (Sens) und der positive prädiktive Wert (PPW) bei der Validierung eines QRS-Detektors für die MIT-Datenbank, die NST-Datenbank und die American Heart Association Database for Evaluation of Ventricular Arrhythmia Detectors (AHA) angegeben werden. Die AHA-Datenbank war nicht verfügbar und so konnte mit dieser Datenbank keine Validierung durchgeführt werden. Die Ergebnisse der MIT-Datenbank zeigen, dass die ARCS-Rohanalyse bei diesen Signalen sehr gut funktioniert. Die etwas schlechteren Ergebnisse der NST-Datenbank sind auf das geringe Signal-Rausch-Verhältnis dieser Signale zurückzuführen. Im ANSI/AMMI EC38:1998 [8] ist eine optionale Validierung anhand der ESC-Datenbank vorgesehen. Die Signale dieser Datenbank wurden von der ARCS-Rohanalyse ausgezeichnet detektiert. Die Validierung anhand der CU-Datenbank ist laut ANSI/AMMI EC38:1998 [8] nur für QRS-Detektoren vorgesehen, die Flattersequenzen erkennen können, was mit der ARCS-Rohanalyse derzeit nicht möglich ist.

Daher sind diese Signale nicht geeignet, um die Qualität der ARCS-Rohanalyse zu beurteilen.

3.6 Testergebnisse

Die Ergebnisse der beat marker-Detektion der ARCS-Rohanalyse wurden anhand der frei verfügbaren Signale der MIT-Arrhythmia-Datenbank [9] auf zwei unterschiedliche Arten validiert: Einmal mit der im Rahmen dieser Diplomarbeit neu entwickelten Validierungsumgebung, und einmal mit einer bereits existierenden Funktion, die speziell für die Validierung der ARCS-Rohanalyse mit diesen Signalen geschrieben wurde. Die Datenbank besteht aus 25 Signalen, die jeweils 30 Minuten lang sind. Die Ergebnisse der beiden Validierungsverfahren stimmten zwar großteils miteinander überein, bei manchen Werten gab es aber Unstimmigkeiten. Bei genauerer Analyse der Funktionen und Signale wurde festgestellt, dass die Unstimmigkeiten auf einen Fehler in der alten Matlab-Validierungsfunktion zurückzuführen waren, während die Ergebnisse des Validierungsmoduls korrekt waren.

Auch die Berechnung der Ergebnisse der Validierung des ARCS-QT-Algorithmus wurde anhand der ARCS-ICE-Datenbank auf zwei unterschiedliche Arten durchgeführt: Einmal mit der Validierungsumgebung und einmal mit einer bereits existierenden Funktion, die speziell für die Validierung des ARCS-QT-Algorithmus anhand der ARCS-ICE-Datenbank entwickelt wurde. Bei Signalen, bei denen die ARCS-Rohanalyse gut funktioniert hat, stimmten die Validierungsergebnisse der beiden Verfahren exakt überein. Bei Signalen, bei denen die ARCS-Rohanalyse nicht alle beats detektieren konnte, kam es zu Abweichungen der Validierungsergebnisse. Wiederum wurden die Funktionen und Signale genau analysiert. Dabei wurde festgestellt, dass bei der alten Matlab-Validierungsfunktion alle waveform marker zur Validierung herangezogen wurden und die Validierung mittels der Validierungsumgebung so funktioniert, dass nur jene waveform marker berücksichtigt wurden, denen beat marker zugeordnet werden konnten. Prinzipiell sind beide Verfahren korrekt. Es muss anhand der konkreten Fragestellung ermittelt werden, ob die Auswertung der waveform marker unabhängig von der Rohanalyse untersucht werden soll oder nicht.

3.7 Laufzeitanalyse

Das Laufzeitverhalten der Zuordnung der korrespondierenden beat marker und das Laufzeitverhalten, wenn zusätzlich noch eine Zuordnung der waveform marker zu den beat markern durchgeführt wird, wurde analysiert. Diese beiden Prozesse wurden ausgewählt, weil sie den größten Anteil an der Gesamtlaufzeit einer Validierung haben. Die Laufzeitanalyse wurde auf einem AMD XP Athlon 2400+ mit 512 MB Arbeitsspeicher durchgeführt.

Bei beiden Zuordnungsprozessen setzt sich die Gesamtlaufzeit aus dem Importieren der entsprechenden Annotationen aus dem Annotationsarchiv, der eigentlichen Zuordnung mittels stored procedure und dem anschließenden strukturierten Abspeichern der gewonnenen Ergebnisse in der Datenbank zusammen. In Abb. 3.17 werden die verschiedenen Laufzeiten für Signale mit einer unterschiedlichen Anzahl an beat markern bzw. waveform markern dargestellt. Die Höhe der blauen Balken gibt die Laufzeit für die Zuordnung von beat markern an. Analog steht die Höhe der grünen Balken für die Laufzeit, die benötigt wird, wenn zusätzlich noch waveform marker zu den beat markern zugeordnet werden müssen.

Der Zuordnungsprozess der beat marker benötigte für die MIT-Arrhythmia-Datenbank [9] 92 Sekunden. Diese Datenbank besteht aus 25 Signalen mit durchschnittlich 2400 beats pro Signal.

Der Zuordnungsprozess der beat marker und der Zuordnungsprozess der waveform marker zu den beat markern dauerten bei der ARCS-ICE-Datenbank 72 Sekunden. Diese Datenbank besteht aus 10 Signalen mit durchschnittlich 2000 beats und 4000 waveform markern (pro beat ist je der Beginn der Beginn des QRS-Komplexes und das Ende der T-Welle annotiert).

Auf eine detaillierte Laufzeitanalyse wurde in dieser Arbeit aus zwei Gründen verzichtet: Erstens ist die Geschwindigkeit der Validierung für die Praxis ausreichend, da der Zeitbedarf für die Analyse von Signalen um eine Größenordnung höher ist als für die Validierung. Zweitens ist die Laufzeit primär von der verwendeten Datenbank abhängig und es ist nicht Ziel dieser Arbeit eine genaue Performanceanalyse von Interbase vorzunehmen. Abb. 3.17 zeigt das Laufzeitverhalten für den in der Praxis am häufigsten vorkommenden Bereich von 500 bis 2000 beats.

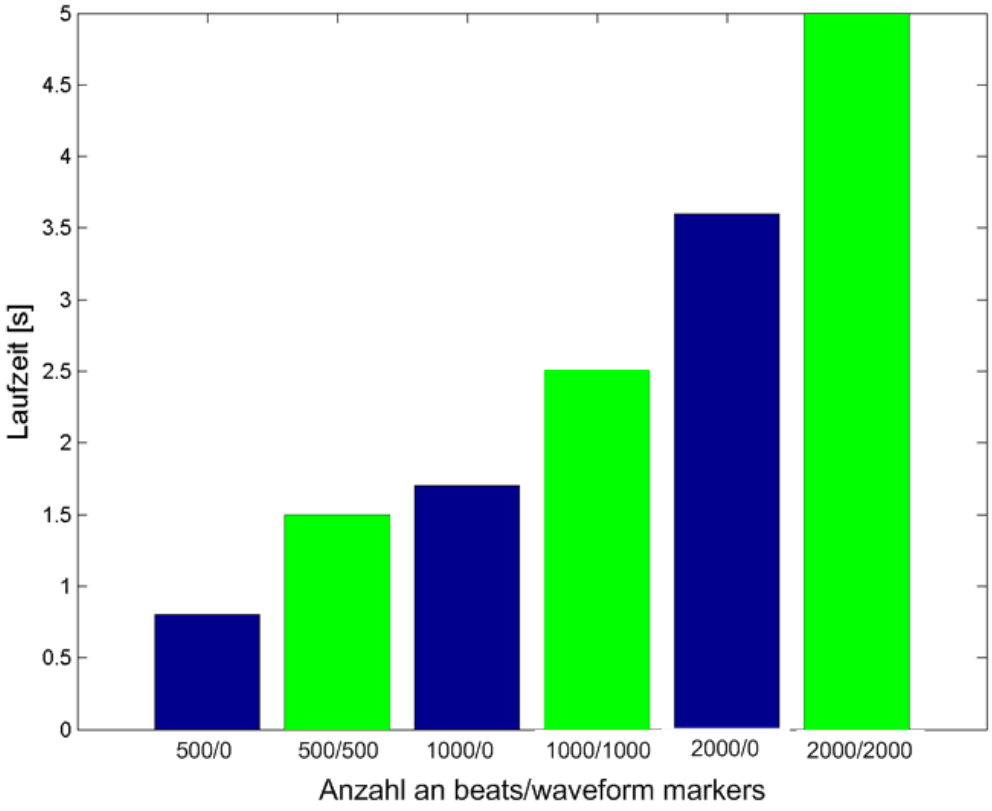


Abbildung 3.17: Das Balkendiagramm zeigt die unterschiedlichen Laufzeiten (blaue Balken), die für das Importieren der Annotationen, die Zuordnung der korrespondierenden beat marker und das Abspeichern der gewonnenen Ergebnisse in der Datenbank für Signale mit je 500, 1000 und 2000 beats benötigt wurden. Analog sieht man die unterschiedlichen Laufzeiten (grüne Balken), wenn zusätzlich noch eine Zuordnung der waveform marker zu den beat markern erfolgt, für Signale mit je 500, 1000 und 2000 waveform markern und beat markern.

4 Diskussion

Zu Beginn dieses Kapitels werden die Konsequenzen analysiert, die sich durch die Entscheidung für Java für die Implementierung der graphischen Benutzerschnittstelle ergeben haben. Danach werden die Stärken und Schwächen des Annotationsarchivs, der Validierungsdatenbank und des Statistikmoduls beschrieben.

4.1 Java versus Matlab

Ein wichtiger Unterschied zwischen Matlab und Java besteht darin, dass Matlab eine imperative und Java eine objektorientierte Programmiersprache ist. Bei großen Softwareprojekten können durch die Verwendung von objektorientierten Entwicklungsmethoden wichtige Qualitätsmerkmale wie Korrektheit, Robustheit, Erweiterbarkeit, Wiederverwendbarkeit und Kompatibilität leichter erreicht werden.

Ein Vorteil von Java ist die freie Verfügbarkeit einer großen Anzahl von Komponenten. Dadurch konnte die Entwicklung der graphischen Benutzerschnittstelle beschleunigt werden. Zum Beispiel war eine Komponente, die eine flexible Auswahl von Dateien ermöglicht [28], für Java verfügbar. Dadurch, dass Java weltweit von sehr vielen Entwicklern verwendet wird, können Fehler in Java-Komponenten rasch erkannt und ausgebessert werden.

Da es zur Aufgabenstellung gehörte, das Validierungsmodul in das easyG, das großteils in Matlab implementiert ist, zu integrieren, musste ein Weg gefunden werden, Java aus Matlab bzw. Matlabfunktionen aus einem Java-Programm zu starten. Matlab bietet das Java Matlab Interface (JMI) für diese Aufgabe an. Leider wird dieses Interface von Matlab nicht ausführlich dokumentiert. Daher war es eine schwierige und langwierige Aufgabe herauszufinden, wie man Matlab Funktionen aus Java startet. Eine

große Hilfe bei der Lösung dieses Problems war die Hompage von Kamin Whitehouse [27]. Ein weiterer Nachteil bei der täglichen Arbeit ist das Fehlen einer Möglichkeit, Java-Programme dynamisch in Matlab zu laden. Daher muss bei jeder Änderung des Java-Programms Matlab neu gestartet werden.

Der anfängliche Aufwand für das Integrieren von Java in Matlab hat sich allerdings rasch amortisiert, da die Entwicklung in Java schneller möglich ist. Außerdem bietet Java die Möglichkeit, mit Hilfe von javadoc das entwickelte Programm zu dokumentieren. Diese Dokumentation wird sicherlich eine große Hilfe sein, wenn in Zukunft Veränderungen an dem Validierungsmodul vorgenommen werden sollen.

4.2 Annotationsarchiv

Der Entwickler kann durch die Verwendung des Annotationsarchivs viel Zeit sparen, weil es ihm die aufwendige Verwaltung aller Daten abnimmt, die für die Validierung eines Algorithmus benötigt werden. Außerdem hilft die automatische Verwaltung, Fehler zu vermeiden, die bei der händischen Verwaltung von einigen hundert Signalen und einem Dutzend Algorithmen wahrscheinlich sind.

Da die Annotationen im Dateisystem gespeichert werden und in der Datenbank nur ein Verweis auf die Daten gespeichert wird, kann eine von der Anzahl der im Annotationsarchiv gespeicherten Annotationen nahezu unabhängige Zugriffsgeschwindigkeit auf die Annotationen garantiert werden. Außerdem wird der Speicherplatz, der für die Annotationen zur Verfügung steht, nicht von der Datenbank sondern nur vom Betriebssystem limitiert.

Um den Validierungsprozess für den Entwickler weiter zu vereinfachen, wäre es in Zukunft möglich, das Annotationsarchiv stärker in das easyG zu integrieren. So könnte man nach einer Analyse von Signalen durch das easyG die erhaltenen Annotationen direkt im Annotationsarchiv abspeichern. Um diese Funktionalität zu implementieren, müssten allerdings größere Veränderungen am easyG vorgenommen werden, die den Rahmen dieser Diplomarbeit gesprengt hätten.

4.3 Kernmodul

Zu Beginn der Diplomarbeit wurde entschieden, die Zuordnung der marker nicht in Matlab sondern in Interbase vorzunehmen. Daraus ergibt sich der Vorteil, dass zugeordneten marker in einer relationalen Datenbank vorliegen und mit Hilfe von SQL abgefragt werden können. Im Gegensatz zu Matlab, das zwar bei der Erstellung spezifischer Validierungsergebnisse schneller ist, können die Rohdaten der Validierung, die in Interbase durchgeführt wurde, beliebig oft nach unterschiedlichen Gesichtspunkten gefiltert werden, ohne den gesamten Validierungsprozess neu starten zu müssen. Die gewonnene Flexibilität musste allerdings mit einem höheren Entwicklungsaufwand bezahlt werden. Der Entwicklungszeitaufwand, um die Annotationsdaten rasch in die Datenbank zu importieren, wurde stark unterschätzt.

Da die Validierungsdauer um eine Größenordnung kleiner ist als die Analyse von Signalen, ist die Geschwindigkeit der Validierung für die Praxis ausreichend.

4.4 Statistikmodul

Das Statistikmodul bietet dem Entwickler ein mächtiges Werkzeug, um die Validierungsdatenbank bequem abfragen zu können. Es werden vom Entwickler keine SQL-Kenntnisse benötigt, um die Rohergebnisse der Validierungsdatenbank nach beliebigen Gesichtspunkten filtern, gruppieren und graphisch darstellen zu können.

5 Schlussfolgerungen

Das Ziel dieser Diplomarbeit wurde mit der Entwicklung der Validierungsumgebung erreicht. Es bietet dem Entwickler eine komfortable Möglichkeit, um die Annotationen, die sein Algorithmus ausgibt, mit den Referenzannotationen eines Experten zu vergleichen. Mit Hilfe dieser Validierungsumgebung kann die Entwicklung von Algorithmen beschleunigt und die Qualität verbessert werden.

Die Umsetzung der Validierungsmethoden, die im American National Standard/AMMI EC38:1998 für den beat-to-beat Vergleich vorgesehen sind, soll dem Entwickler helfen, seine entwickelten Algorithmen nach diesem Standard zertifizieren zu lassen.

Literaturverzeichnis

[1] Barold, S. *Willem Einthoven and the birth of clinical electrocardiography a hundred years ago.* http://makeashorterlink.com/œT2D2129D7

[2] So, C.: *Praktische Elektrokardiographie.* Selecta Verlag, 1980

[3] Statistik Austria. *Gesundheit im Überblick.* http://www.statistik.at/fachbereich_03/gesundheit_txt.shtml

[4] EKG-Grundkurs. *Hund des Herzspezialisten Augustus Desire Waller.* http://www.laborkurs.de/uni-ss03/EKG-Grundkurs-TEIL%20C.pdf

[5] Schmidt, R.F. ; Thews, G.: *Physiologie des Menschen.* Springer, 1995

[6] Pessenhofer, H. *Vorlesungsunterlagen Physiologie.* http://www.kfunigraz.ac.at/phywww/biomedtechnik/herz_med.pdf

[7] Keidel, W.D.: *Kurzgefaßtes Lehrbuch der Physiologie.* Georg Thieme Verlag, 1985

[8] ANSI: *Ambulatory electrocardiograph (ANSI/AAMI EC38:1998).* 1998

[9] Moody, G.B. ; Mark, R.G.: *The impact of the MIT-BIH Arrhythmia Database.* IEEE Engineering in Medicine and Biology Magazine 20(3): 45-50, 2001

[10] Taddei, A. ; Biagini, A. ; a. Distante ; Marchesi, G. ; Mazzei, C. ; M.G. ; Pisani ; Roggero, P. ; Zeelenberg, N. ; Moody, C.G.B. ; Mark, R.G.: *An annotated database aimed at performance evaluation of algorithms for ST-T change analysis.* Comp Cardiol 16:117-120 (1989), 1989

[11] Moody, G.B. ; Muldrow, W.E. ; Mark, R.G.: *A noise stress test for arrhythmia detectors.* Comp Cardiol 1984; 11:381-384, 1984

[12] Nolle, F. M. ; Badura, F. K. ; Bowser, J. M. Catlett R. W. ; Sketch, M. H.: *a new concept in computerized arrhythmia monitoring systems.* Comp Cardiol 13:515-518 (1986)., 1986

[13] Peer, C. ; Schreier, G. ; Kastner, P. ; Marko, W. ; Messmer, J. ; Rotman, B. ; Lercher, P. ; Klein, W.: *EASYG - Electrocardiogram Analysis System Graz.* IEEE Proceedings 2003: 207-210, 2003

[14] Schreier, G. ; Kastner, P. ; Marko, W.: *An automatic ECG processing algorithm to identify patients prone to paroxysmal atrial fibrillation.* IEEE Computers in Cardiology 2001, 28:133-135, 2001

[15] Schreier, G. ; Hayn, D. ; Lobodzinski, S.: *Development of a new QT algorithm with heterogenous ECG databases.* J Electrocardiol 2003, 36:145-150, 2003

[16] Kollmann, A.: *Patientennahe Herzschrittmacher-Nachsorge*, TU Graz, Diplomarbeit, 2002

[17] Hayn, D.: *Ein Finite Elemente Herzmodell*, TU Graz, Diplomarbeit, 2002

[18] Matlab. *Statistics Toolbox 4.* `http://www.mathworks.com/products/statistics/description1.shtml`

[19] Überhuber, C. ; Katzenbeisser, S.: *Matlab 6.5 Eine Einführung.* Springer, 2002

[20] Matlab. *Database Toolbox 2.2.1.* `http://www.mathworks.com/products/database/description1.jsp`

[21] Meyer, B.: *Object-Oriented Software Construction.* Prentice-Hall, 1997

[22] Arnold, K. ; Gosling, J. ; Holmes, D.: *The Java Programming Language.* Addison-Wesley, 2000

[23] J.Byous. *Java Technology: The early years.* `http://java.sun.com/features/1998/05/birthday.html`

[24] Edlich, S.: *Ant kurz und gut.* Addison-Wesley, 2002

[25] Log4j Project. *Introduction.* `http://logging.apache.org/log4j/docs/`

[26] Webb, P. *Integrating Java Components into MATLAB.*
http://www.mathworks.com/company/newsletter/win02/patterns.shtml

[27] Whitehouse, K. *Calling Matlab from Java.*
http://www.cs.berkeley.edu/~kamin/matlab/JavaMatlab.html

[28] Klinner, K. ; Guy, R. *FindAccessory.*
http://makeashorterlink.com/œA3F2129D7

[29] The Swing Tutorial. *Intro.*
http://java.sun.com/docs/books/tutorial/uiswing/start/swingIntro.html

[30] Interbase. *History.*
http://bdn.borland.com/article/0,1410,25302,00.html

[31] Interbase. *Features of Interbase 6.5.*
http://www.borland.com/interbase/pdf/ib65_feaben.pdf

[32] Scerbakov, N. *relationale Datenbanken.* coronet.iicm.edu/is/is_res.htm

[33] Loomis, M. E. S.: *The database book.* Macmillan Publishers, 1987

[34] Grofft, J.R. ; Weinberg, P.N.: *Using SQL.* McGraw-Hill, 1990

[35] Borland: *Interbase Data Definition Guide*

[36] JDBC. *Overview.* http://java.sun.com/products/jdbc/overview.html

[37] Java Tutorial. *Threads.*
http://java.sun.com/docs/books/tutorial/essential/threads/

[38] Sachs, L.: *Angewandte Statistik.* Springer, Berlin, 2003

[39] Bronstein, I.N. ; Semendjajew, K.A. ; Musiol, G. ; Mühlig, H.:
Taschenbuch der Mathematik. Harri Deutsch, 1999