



Universitetet
i Stavanger

DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

MASTEROPPGAVE

Studieprogram/spesialisering: Master i Informasjonsteknologi	Vårsemesteret, 2010 Åpen
Forfatter: Ørjan Horpestad (signatur forfatter)
Faglig ansvarlig: Arne Rettedal Veiledere: Arne Rettedal, Erlend Tøssebro, Ståle Freyer.	
Tittel på masteroppgaven: Grafisk grensesnitt for håndtering av lagrede hjertedata (EKG) Engelsk tittel: Graphical interface for handling offline electrocardiogram	
Studiepoeng: 30	
Emneord: EKG, aktiveringsnivå, Alive Heart Monitor, XML, GUI, Java, EDF, ATS, annotering, Stressmåling ved simulatortrening, PatSim.	Sidetall: 59 + vedlegg/annet: 3 Stavanger, dato/år

Grafisk grensesnitt for håndtering av lagrede hjertedata (EKG)

Masteroppgave

Av

Ørjan Horpestad

Våren 2010

Sammendrag

Det antydes av læringspsykologer og pedagoger at læring øker hos en person ved et noe økt stressnivå. Det viser seg at denne læringen avtar straks stressnivået overskrider en viss grense, som er individuell for hver person.

Institutt for data og elektroteknikk ved UiS har et simulatorlaboratorium hvor det gjøres simuleringer innen intensiv behandling og anestesi. Denne simulatoren er bygget tett opp mot realistiske scenario for å gi mest mulig reelle observasjoner.

PatSim er en pasientsimulator utviklet med hensikt å la studenter trene seg i intensivbehandling. Ved å tilby studenter en simulator å øve seg på denne type behandling, unngås det at pasienter blir utsatt for unødig risiko. I PatSim er det lagt stor vekt på realisme. Studenter kan her observere og gjøre målinger direkte på en dukke. De fysiologiske signalene dukken gir ut, inkluderer blant annet blodtrykk, elektrokardiogram (EKG), temperaturer, utslett, tårer, blå lepper, bevegelser og lyder.

Målet om å oppnå en viss grad av realisme i simulatorer, begrenses av studentens mentale innstilling til situasjonen. Ved likegyldig innstilling hos en student, vil han eller hun ha vanskelig for å få et pedagogisk utbytte av opplevelsen. Dersom studenten engasjeres så mye at det kommer til et punkt hvor han eller hun føler seg for stresset, kan læringsprosessen også svinne hen.

Prosjektet *Stressmåling ved simulatorentrening (SVS)* har som mål å se på den enkeltes stressnivå og kunne oppdage når det vil være hensiktsmessig å begrense stressaktiveringen under simuleringer. Dette for å unngå irreversible og uheldige effekter på deltakeren som er i simulatoren. Elektrokardiogramregistrering gjøres med en portabel hjerteropptaker knyttet sammen med en datamaskin via Bluetooth. I prosjektet brukes EKG-registrering og analyse av hjerterate og hjerteratevarabilitet som en del av stressmålingen.

Denne masteroppgaven har som mål å utvikle og konstruere et grensesnitt som er tilrettelagt for å enklere kunne jobbe med disse lagrede data. Grensesnittet skal muliggjøre annotering på disse opptakene. Dette innebærer markering av episodestart/-slutt, identifikasjonsdata (kandidatens ID, type scenario, og lignende) for den aktuelle episoden, samt støy og ekstrasystoler. Grensesnittet skal gi muligheter for å kunne lese inn opptak gjort med Alive Technologies' Bluetooth-baserte hjertemonitor.

Den utviklede løsningen skal på grunnlag av annotasjonene som er gjort, kunne generere en annoteringsfil til senere bruk i MATLABs batch-analyse. Ved å markere (annotere) områder inneholdende støy og ekstrasystoler, legges det til rette for å kunne få tilgang til tidligere utilgjengelig informasjon i de lagrede hjertedataene.

Innholdsfortegnelse

1	INNLEDNING	9
1.1	BAKGRUNN	9
1.1.1	<i>Pasientsimulatoren PatSim</i>	9
1.2	OPPGAVEN	10
1.3	HVA SKAL UTFØRES?.....	10
1.3.1	<i>Flytdiagram</i>	10
1.3.2	<i>Nødvendig funksjonalitet i detalj</i>	11
2	TEORI	12
2.1	AKTIVERINGSNIVÅ.....	12
2.1.1	<i>Aktiveringsnivå i kombinasjon med PatSim</i>	12
2.2	ELEKTROKARDIOGRAM.....	12
2.2.1	<i>Bakgrunn</i>	13
2.2.2	<i>Ekstrasystoli</i>	13
2.2.3	<i>Det normale EKG</i>	13
2.2.4	<i>Støy i EKG-opptak</i>	14
2.2.5	<i>EKG-utstyr</i>	14
2.3	HJERTERYTMEVARABILITET.....	16
2.4	HJERTEDATAPROGRAMVARE.....	16
2.5	HJERTEDATAFORMAT.....	17
2.5.1	<i>ATS</i>	17
2.5.2	<i>EDF</i>	18
3	UTVIKLING.....	20
3.1	ALTERNATIVER FOR LAGREDE HJERTEDATAFORMAT	20
3.1.1	<i>Implementasjon av ATS-format</i>	20
3.1.2	<i>Relatert arbeid med EDF</i>	20
3.1.3	<i>Implementasjon av European Data Format</i>	21
3.2	ALTERNATIVER FOR PLOTTLØSNING	21
3.2.1	<i>Kriterier</i>	21
3.2.2	<i>Alternativer</i>	21
3.3	ALTERNATIVER FOR ANNOTASJONSFORMAT	22
3.3.1	<i>Kriterier</i>	22
3.3.2	<i>Alternativer</i>	22
4	EKSPERIMENTELT	24
4.1	VALGT FORMAT FOR LAGREDE HJERTEDATA	24
4.2	VALGT PLOTTLØSNING.....	24
4.3	VALGT ANNOTASJONSFORMAT.....	24
4.3.1	<i>Annotasjonsstruktur</i>	25
4.4	PROGRAMDESIGN	27
4.4.1	<i>Brukervennlighet</i>	27
4.4.2	<i>Målgruppe</i>	28
4.4.3	<i>Fleksibilitet</i>	29
4.4.4	<i>Pålitelighet</i>	29
4.4.5	<i>Ytelse</i>	30
5	RESULTAT.....	35
5.1	KLASSEDIAGRAM	35
5.2	FLYTDIAGRAM OVER IMPLEMENTERT LØSNING.....	36
5.3	VIKEMÅTE AV UTVIKLET PROGRAMLØSNING	36
5.4	BRUKERVEILEDNING.....	37
5.4.1	<i>Systemkrav</i>	37
5.4.2	<i>Oppsett og innstillinger</i>	38
5.4.3	<i>Funksjonalitet i mus</i>	39
5.4.4	<i>Åpne hjertedata (ATS) for annotering</i>	41

5.4.5	Åpne annotasjonsfil (XML) for visning eller videre annotering.....	48
5.5	TILLEGGSINFORMASJON.....	51
5.5.1	Forsinkelse i deteksjon av R-takk og HR.....	51
5.5.2	Regler for filnavn.....	51
5.5.3	Panorering for tidsvindu på 1 og 5 sekund.....	51
6	DISKUSJON.....	52
6.1	MULIGE FEILKILDER.....	52
6.2	OPPDATERT ATSCONVERT UNDER UTVIKLING.....	53
6.3	MULIGE FORBEDRINGER.....	53
6.4	FREMTIDIG ARBEID.....	53
6.4.1	Analyse av lagrede hjertedata.....	54
6.4.2	Sanntidsanalyse av hjertedata.....	55
7	REFERANSE- OG KILDEHENVISNING.....	56
7.1	FIGURER, BILDER, IKONER.....	57
7.2	JAVA BIBLIOTEKER.....	58
7.3	PROGRAMVARE BRUKT TIL UTVIKLING OG IMPLEMENTERING.....	58
8	VEDLEGG.....	59

Forord

Jeg ønsker å takke mine veiledere Arne Rettedal, Ståle Freyer og Erlend Tøssebro for sin veiledningsinnsats gjennom arbeidet med denne masteroppgaven.

Jeg ønsker også å rette en takk til Kim Barnett ved Alive Technologies Pty. Ltd. (AUS) for tilbakemeldinger vedrørende selskapets hjertemonitor- og applikasjonsportefølje.

Stavanger, 15. juni 2010.

Ørjan Horpestad

Oversikt

Rapporten er oppdelt i sju kapitler. I forkant av første kapittel listes tabell med forkortelse- og begrepsforklaring.

Kapittel 1: Innledning

Inneholder bakgrunn for denne masteroppgaven. Videre utdypes de funksjonalitetskrav som er lagt til grunn. Avslutningsvis for kapitlet redegjøres en mer detaljert oppgavebeskrivelse etterfulgt av et flytdiagram over en prinsipiell implementering.

Kapittel 2: Teori

Omfatter hovedemnene aktiveringsnivå, elektrokardiogram, hjerterytmearabilitet, samt hjertedataprogramvare og filformat for lagrede hjertedata.

Kapittel 3: Utvikling

Datatekniske alternativer redegjøres og vurderes for implementering i det grafiske grensesnittet. Dette inkluderer alternativer for henholdsvis plott av hjertedata, filformat for lagrede hjertedata, samt annotasjonsformat.

Kapittel 4: Eksperimentelt

På bakgrunn av vurderte alternativer i kapittel 3, beskrives og begrunnes de ulike valg som er tatt. Kapitlet omfatter i tillegg momenter for programdesign.

Kapittel 5: Resultat

Klassediagram gir en oversikt over endelig programstruktur. En teknisk gjennomgang forklarer prinsipiell virkemåte. En brukerveiledning er inkludert for å demonstrere og forklare virkemåten av den utviklede prototypen. I tillegg listes tilleggsinformasjon relevant for prototypen.

Kapittel 6: Diskusjon

Diskusjonen inneholder drøfting av mulige feilkilder, mulige forbedringer, fremtidige løsninger, videre håndtering av lagrede annotasjonsfiler, fremtidig arbeid, samt en oppsummering.

Kapittel 7: Referanse- og kildehenvisning

I siste kapittel listes litteratur-, kode- og figurreferanser.

Forkortelser og begrep

Forkortelse, begrep	Betydning, forklaring
AHM	Alive Heart Monitor
Alive Technologies	Australsk selskap som spesialiserer seg på utvikling og produksjon av medisinsk utstyr. Dette inkluderer utvikling av trådløse hjertemonitorer som muliggjør visning, diagnose og overvåking av kroniske sykdommer, eller for bruk av vanlige forbrukere med fokus på helse og trening. <i>Alive Heart and Activity Monitor</i> er hjertemonitoren lagt som basis for denne oppgaven.
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ATS	Alive Technologies' egenutviklede dataformat for lagring og overføring av hjertedata.
AtsConvert	Programvareverktøy for konvertering fra ATS-format til andre hjertedataformat.
CSV	Comma Separated Value Format
DOM	Document Object Model
EDF	European Data Format
EKG / ECG	Elektrokardiograf, elektrokardiografi
Ekstrasystol	Betegnelse på ekstra hjerteslag som kommer mellom to normale hjerteslag.
HR	Hjerterate
HRV	Hjerterytmearabilitet
JFC	JFreeChart. Java-bibliotek for plotting av grafer.
JRE	Java Runtime Environment
OS	Operativsystem
PatSim	Pasientsimulator utviklet ved Universitetet i Stavanger, med hensikt å la studenter trene seg i intensivbehandling.
QRS-kompleks	Komponentene Q, R og S utgjør sammen med P, T og U et elektrokardiogram.
R-takk	Den største, raskeste elektriske ladningen som forekommer i et hjerteslag.
RR	Avstanden mellom to R-takk.
SAX	Simple API for XML
SFP	Sammenfallende punkter (ytelsesfremmende metode implementert).
SVS	Stressmåling ved simulatortrening. Forskningsprosjekt ved UiS.
Tooltip	Grafisk grensesnittelement med informasjon som vises når musepekeren hviler over et punkt eller et annet grafisk element.
XML	Extensible Markup Language

1 Innledning

Innledningsvis beskrives bakgrunn for masteroppgaven. Videre utdypes de funksjonalitetskrav som er lagt til grunn. Avslutningsvis for kapitlet redegjøres en mer detaljert oppgavebeskrivelse etterfulgt av et flytdiagram over en prinsipiell implementering.

1.1 Bakgrunn

Institutt for data og elektroteknikk ved Universitetet i Stavanger har et simulatorlaboratorium hvor det gjøres simuleringer innen intensiv behandling og anestesi. Denne simulatoren er bygget tett opp mot realistiske scenario for å gi mest mulig reelle observasjoner. Det antydes av læringspsykologer og pedagoger at læring øker hos en person ved et noe økt stressnivå. Det viser seg at denne læringen avtar straks stressnivået overskrider en viss grense, som er individuell for hver person.

Å ha et visst stressnivå i situasjoner der det kreves litt ekstra, vil en person dra nytte av ved at personen er mer åpen for læring. Blir derimot stressnivået for høyt, kan personen oppleve konsekvenser i form av handlingslammelser og/eller lavt læringsutbytte. Med andre ord, tilstander som ikke er ideelle for læring.

Prosjektet *Stressmåling ved simulatorentrening (SVS)* har som mål å se på den enkeltes stressnivå og å kunne oppdage når det vil være hensiktsmessig å begrense stressaktiveringen under simuleringer. Dette for å unngå irreversible og uheldige effekter på deltakeren som er i simulatoren.

Elektrokardiogramregistrering gjøres med en portabel hjerteopptaker knyttet sammen med en datamaskin via Bluetooth. I prosjektet brukes EKG-registrering og analyse av hjerterate og hjerteratevarabilitet som en del av stressmålingen. Dette muliggjør diagnosestilling av ulike sykdommer og/eller vurdering av fysisk og mental belastning hos friske mennesker. I prosjektets innledende del vil analysen av opptakene foregå retrospektivt. Videre i prosjektet er det satt som mål å gjøre denne analysen i sanntid. Sanntidsanalyse beskrives nærmere i kapittel 6.4.2.

1.1.1 Pasientsimulatoren PatSim

PatSim er en pasientsimulator utviklet ved Universitetet i Stavanger med hensikt å la studenter trene seg i intensivbehandling. Simulatoren er bygget rundt en torso produsert av Laerdal Medical. Ved å tilby studenter en simulator for å øve seg på denne type behandling, elimineres risikoelementer i forhold til pasienter. Simulatoren tilbyr også studenter å øve seg på bedøvende behandling.

I utviklingen av PatSim er det lagt stor vekt på realisme. Det faktum at pasientdukken selv gir ut fysiologiske signaler i ulike former, skiller simulatoren fra andre. Studenter kan gjøre målinger og observasjoner direkte på dukken. Dette til forskjell fra andre simulatorer fra samme tidsperiode, hvor det var mer vanlig at fysiologiske tegn ble generert direkte på en monitor. De fysiologiske signalene dukken gir ut, inkluderer blant annet blodtrykk, EKG, temperaturer, utslett, tårer, blå lepper, bevegelser og lyder. For å gjøre illusjonen av en ordinær operasjonsstue ytterligere reel, kan det også settes spesielle krav til bekledning og omgivelser [1].

1.2 Oppgaven

Denne masteroppgaven har som mål å utvikle og konstruere et grensesnitt som er tilrettelagt for å enklere kunne jobbe med disse lagrede stressmålingene. Et opptak gjort i løpet av en simuleringssesjon lagres som en datafil. Et opptak kan bestå av forskjellige episoder, studenter (herav: *kandidater*) og i tillegg støy.

Grensesnittet har som mål å kunne gjøre ulike annoteringer på disse opptakene. Dette innebærer episodestart og -slutt og identifikasjonsdata for den aktuelle episoden (kandidatens ID, type scenario, og lignende). I tillegg skal støy på opptakene markeres i forkant av opptakets analyse. Dermed er det også et mål å kunne gjøre annoteringer for tidsperioder inneholdende støy.

1.3 Hva skal utføres?

Det skal konstrueres et grafisk brukergrensesnitt for å kunne håndtere de lagrede hjertedataene. Grensesnittet skal gi muligheter for å kunne lese inn opptak gjort med Alive Technologies' Bluetooth-baserte hjertemonitor. Dette opptaket kan bestå av data målt over en tidsperiode som involverer flere kandidater. Det er derfor et krav å kunne markere hvor på tidsaksen målingen for hver kandidat starter og slutter.

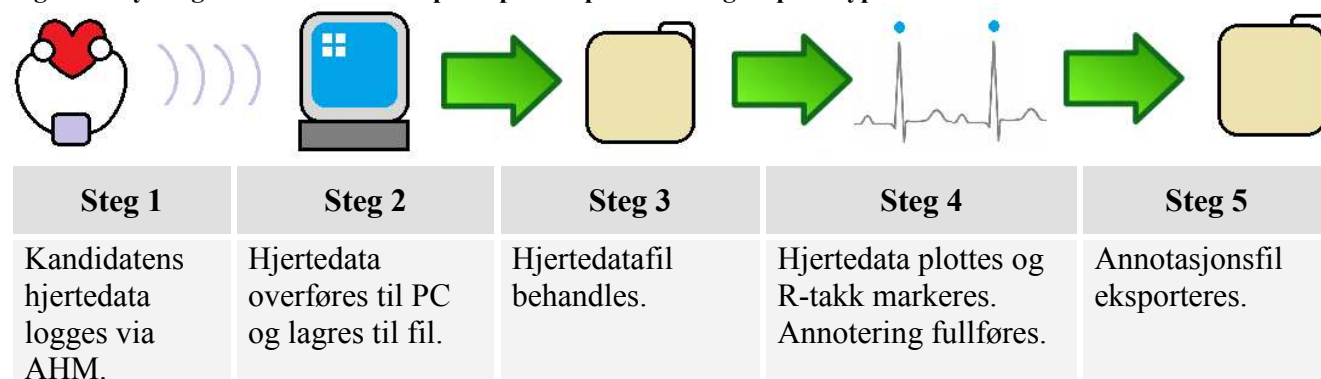
Opptaket kan også inneholde perioder med støy av ulike opphav. Et annet krav er derfor at det også skal være mulig å markere hvor eventuelle støyområder starter og slutter. Det er i tillegg nyttig å kunne markere forekomster av ekstrasystoler (ekstra hjerteslag). Det er ønskelig med felter som kan brukes for å angi tilleggsinformasjon for hver kandidat (scenario, kjønn, alder, etc.).

Systemet skal på grunnlag av annotasjonene som er gjort, kunne generere en annoteringsfil til senere bruk i MATLABs batch-analyse. Ved å markere (annotere) områder inneholdende støy og ekstrasystoler legges det til rette for å kunne få tilgang til tidligere utilgjengelig informasjon i de lagrede hjertedataene.

1.3.1 Flytdiagram

Hjertedata registreres med bærbar opptaker og lagres på en PC. Hjertedataene konverteres til passende format og plottes med tilhørende RR-intervall. Annotasjoner gjøres på graf, og eksporteres til et eget annotasjonsformat. Annotasjonene kan på et senere tidspunkt leses inn i MATLAB for videre analyse.

Figur 1: Flytdiagram som illustrerer prinsipiell implementering av prototypen:



Denne masteroppgaven er sentrert rundt steg 3 - 5. Avslutningsvis foreslås i tillegg en videre håndtering av annotasjonsfilene i forbindelse med MATLAB, som vil være et sjette steg.

1.3.2 Nødvendig funksjonalitet i detalj

Følgende (fritt sitert) funksjonalitet listet i opprinnelig oppgavebeskrivelse for masteroppgaven.

- Import og visning av ATS (og indirekte EDF)
 - Innbærer korrekt lesing/tolking
 - Invertere kurve ved behov
- Algoritme for å detektere hjerteslag
- Annoteringsmuligheter for opptaket
- Opprinnelige hjertedatafiler skal ikke altereres. I stedet skal annotasjoner lagres eksternt.
- Eksportering av annotasjoner kompatible med MATLAB
- Eksportering av liste med tidsintervall mellom hjerteslag¹

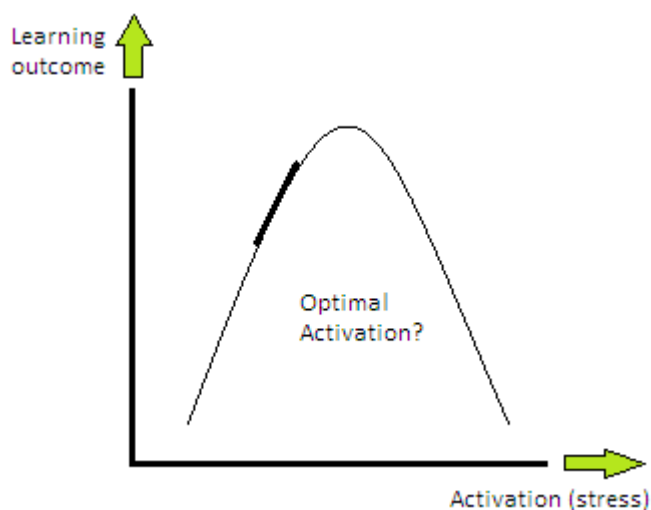
¹ Dette ligger allerede tilgjengelig i CSV-fil konvertert med AtsConvert-programmet.

2 Teori

I dette kapittelet redegjøres eksisterende teknikker, begreper, programvare, filformat og andre emner som er relevante for arbeidet med denne masteroppgaven.

2.1 Aktiveringsnivå

Målet om å oppnå realisme av en viss grad i simulatorer, begrenses i varierende grad av kandidatens mentale innstilling til situasjonen. Denne begrensningen kan spenne fra liten, noe, delvis, stor, til svært stor grad. Ved likegyldig innstilling hos en kandidat, vil han eller hun ha vanskelig for å få et pedagogisk utbytte av situasjonen. Dersom kandidaten engasjeres så mye at det kommer til et punkt hvor han eller hun føler seg for stresset, kan læringsprosessen også svinne hen [1]. Dette gjelder også for flere opplærings situasjoner i fullskala simulatorer.



Figur 2: Læring og aktiveringsnivå [35]

2.1.1 Aktiveringsnivå i kombinasjon med PatSim

I SVS brukes trådløse hjertermontorer levert av Alive Technologies for måling av blant annet hjerterate (HR) og elektrokardiogram (EKG). I tillegg kan måleparametere som bioimpedans og kandidatens tonenivå på stemmebåndet gi et mer korrekt syn på personens aktivasjonsnivå.

For å prøve å holde et aktiveringsnivå hvor kandidaten har utbytte av læringen, kan en instruktør manipulere scenarioet direkte under simuleringen [2]. For å kjenne til kandidatens aktiveringsnivå, må personen utstyres med måleelektroder fra *Alive Heart Monitor* (AHM) opptakeren, slik at hans eller hennes EKG-data kan registreres og analyseres.

2.2 Elektrokardiogram

Det er elektrokardiogramdata som er de mest essensielle for forskningsprosjektet. En persons EKG-data kan gi indikasjoner på den enkeltes stressnivå. I lengden er det i SVS et mål om å bruke denne informasjonen til å finne studentens optimale aktiveringsnivå.

2.2.1 Bakgrunn

Hjertet består av totalt fire kammer. Disse kamrene er fordelt på høyre og venstre side av hjertet. Hver side har et forkammer (atrium) og et hovedkammer (ventrikkel). Kamrene på høyre side av hjertet pumper blod ut i lungekretsløpet, mens kamrene på den venstre siden av hjertet pumper blod ut i det store kretsløpet.

Under et ordinært hjerteslag vil først forkamrene trekke seg sammen, noe som medfører at de avslappede hovedkamrene fylles med blod. Deretter trekkes hovedkamrene seg sammen, lukker klaffene inn til forkamrene og presser blodet ut i kretsløpet. Når klaffene mellom forkamrene og hovedkamrene er lukket, vil forkamrene igjen fylles med blod. Slik gjentas denne prosessen, med variabelt tempo, avhengig av situasjonen.

Hjertet drives av signaler fra sinusknuten til å pumpe blod rundt i kroppen. Sinusknuten er den delen av hjertet som har som oppgave å holde takten i hjerteslagene (derav det engelske uttrykket *pacemaker*). Når musklene i hjertet trekker seg sammen vil det kunne detekteres ulike elektriske impulser. Disse impulsene kan fanges opp av en hjertemonitor.

EKG står for elektrokardiogram (også referert til som elektrokardiografi), og er en måling som gjøres av den elektriske aktiviteten i hjertemuskelen. En EKG-graf består av flere bølger med ulike utslag. Størrelsen på utslaget til en bølge avhenger av hvilket hjertekammer som avgir den elektriske impulsen [8, 13, 14].

2.2.2 Ekstrasystoli

Ekstrasystoli er definert som en unormal hjerterytm. Denne hjerterytmen er karakterisert ved at to kontraksjoner i hjertet skjer i løpet av et mindre tidsrom enn vanlig. Etter siste slag tas det en pause før det påfølgende tredje hjerteslaget. Denne pausen har like lang varighet som den tid det normalt sett ville forløpt mellom det første og det tredje hjerteslaget.

Ekstrasystoler kan opptre individuelt eller i serier, og er et alminnelig fenomen. Personer som opplever ekstrasystoler kan føle at hjertet hopper over et slag, ettersom de to første kontraksjonene kommer såpass tett etter hverandre at de kan føles som én. Det tredje slaget vil føles kraftigere enn normalt ved at hjertet i dette slaget pumper ekstra mye blod [15, 16].

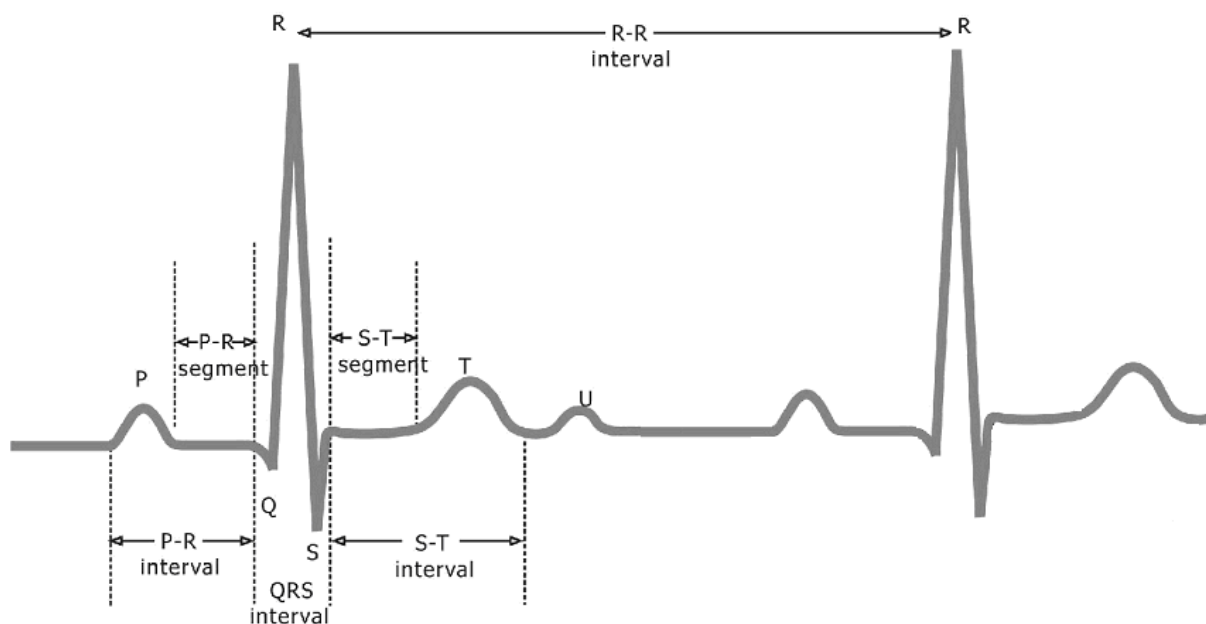
2.2.3 Det normale EKG

De ulike komponentene i et normalt EKG (se Figur 3) betegnes med bokstavene P, Q, R, S, T, og U. P-bølgen kommer av sammentrekningen i forkamrene. Hovedkamrene vil som oftest være representert med tre komponenter; Q, R og S. Disse har samlebetegnelsen QRS-kompleks, og forekommer når hovedkamrene trekker seg sammen. Når muskelcellene henter seg inn etter sammentrekningen (repolariseres) vil det kunne registreres en elektrisk impuls representert med en T-bølge.

Den påfølgende U-bølgen er vanligvis ikke regnet som en del av det normale EKG. Den skyldes vanligvis av en sen, unormal repolarisering av enkelte deler av hjertemuskelen. U-bølgen regnes som sjelden, og er derfor heller ikke alltid synlig i et EKG.

Ved å se på avstanden mellom de ulike bølgene i en EKG-bølge er det, innen fysiologi, mulig å indikere ulike symptomer og hjertesykdommer. En EKG-måling gjør det også mulig å påvise avvik i hjerterytmen.

Et reelt EKG hentet fra et tilfeldig menneske vil kunne variere fra det normale EKG. Dette kommer av at grafen avhenger av flere faktorer; blant annet alder, fysisk tilstand, stress, post-hjerteanfall, og andre faktorer [21].



Figur 3: "Det normale EKG". Dette er hva som er ansett som et normalt elektrokardiogram. EKG-grafen vil variere fra person til person. Et normalt hjerteslag genererer et EKG som består av en liten P-bølge først. Deretter følger QRS-komplekset, etterfulgt av en T- og U-bølge på slutten. Det største utslaget skjer i QRS-komplekset. Dette kommer av at det er større muskelmasse i hjertets hovedkamre enn i forkamrene [21, 34].

Ut ifra grafen i Figur 3 er det mulig å se tidsintervallet mellom hvert R-takk. Dette kalles RR-intervall. RR-intervall er et kritisk element i aktiveringsanalysen som gjøres i forskningsprosjektet som denne masteroppgaven er basert på.

2.2.4 Støy i EKG-opptak

Begrepet støy vil i et EKG-opptak kunne være et resultat av følgende:

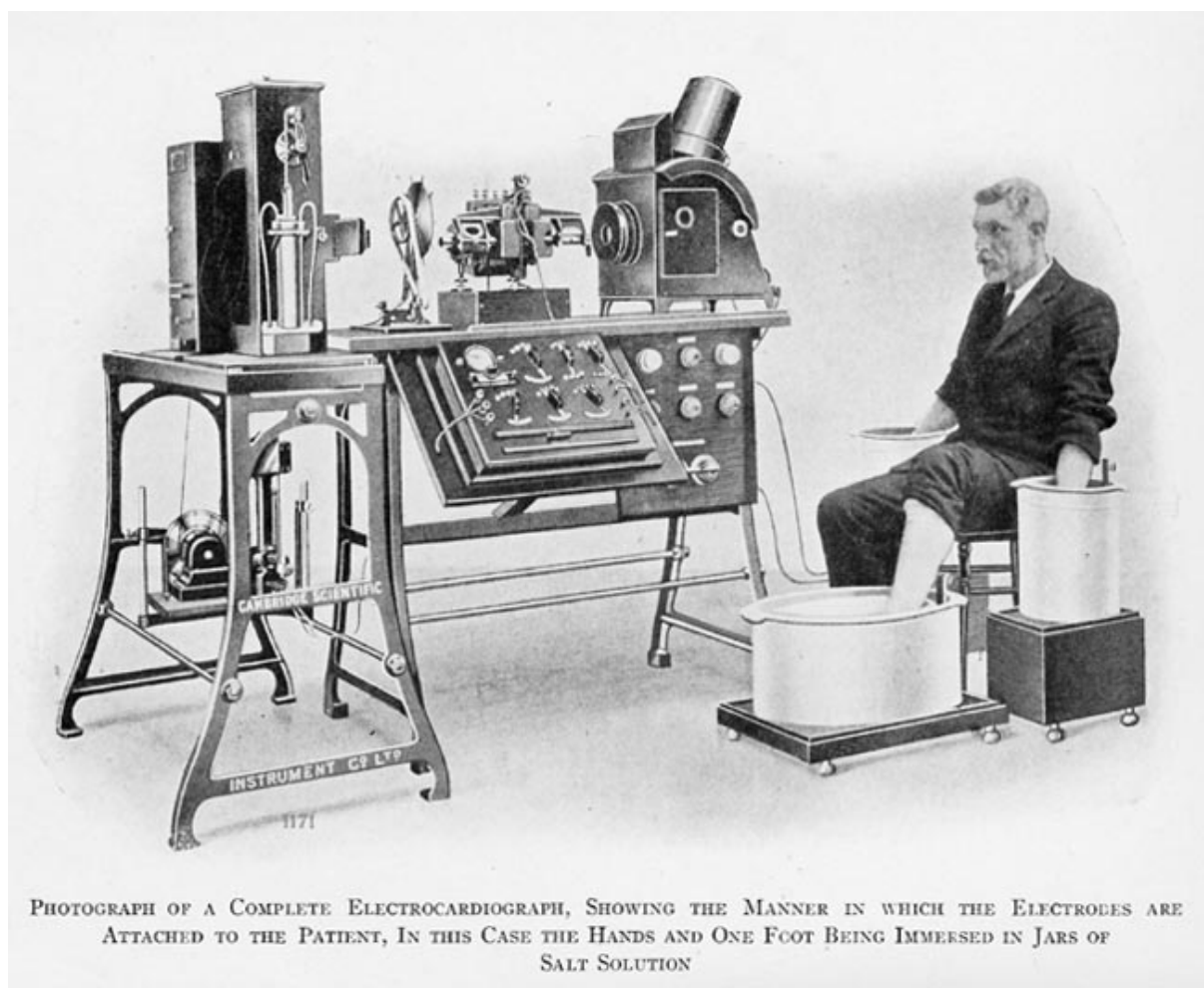
1. at en (eller flere) måleelektroder har blitt flyttet på
2. at en (eller flere) måleelektroder har falt av
3. dårlig kontakt i ledninger
4. dårlig kontakt i en (eller flere) måleelektroder
5. bevegelsesartefakter
6. ekstern indusert støy (forårsaket av f. eks. mobiltelefon, nettstøy fra jordfeil - sjelden på bærbare opptakere, etc.).

Ved å kunne fjerne og/eller markere områder med støy og ekstrasystoler, er det mulig å kunne få tilgang til informasjon i EKG-signalet som tidligere har vært utilgjengelig.

2.2.5 EKG-utstyr

Verdens første fungerende elektrokardiograf ble oppfunnet av Willem Einthoven i 1903. Einthoven klarte å måle endringer i hjertets potensielle elektriske energi ved hjelp av en av hans tidligere oppfinnelser, kjent som Einthoven-Galvanometeret. Ved å videreutvikle galvanometeret gjorde han det mulig å gjøre en grafisk representasjon av denne elektriske

energien, og i tillegg lagre dette opptaket. Elektrokardiografen medførte at han i 1924 ble tildelt Nobelspris innen fysiologi/medisin [28].



Figur 4: En tidlig variant av Einthovens elektrokardiograf [28]. Elektrodene kobles på pasienten ved å nedsenke hendene og den ene foten i tre ulike beholdere, med hver beholder fylt med saltløsning.

En hjertemonitor eksisterer i dag i forskjellige utforminger; fra tradisjonelle monitorbokser med elektrodepar, personlige pulsklokker, til bærbare opptakere (gjerne trådløse). En variant av sistnevnte er brukt som basis for denne masteroppgaven (se Figur 7).



Figur 5: Tradisjonell hjertemonitor [32]



Figur 6: Personlig pulsklokke[31]



Figur 7: Bærbar opptaker [33]

2.3 Hjerterytmearabilitet

Hjerterytmearabilitet (HRV) har flere definisjoner. Begrepet hjerterytmearabilitet karakteriserer svingninger i både RR-intervaller og momentan hjerterate. Det finnes definisjoner som tar utgangspunkt i avstanden mellom P-bølgene i et EKG. Denne definisjonen begrenses av at P-bølger er vanskelige å detektere samt å finne maksimumsverdier for.

R-takket i et QRS-kompleks er distinktivt, og karakteriseres ved at det har kort tidsvarighet samtidig som det har en skarp topp. R-takket er derfor et bedre utgangspunkt for arbeid med HRV. Med utgangspunkt i R-takket defineres HRV som variasjon i lengde mellom etterfølgende RR-intervaller. Et RR-intervall defineres som et tidsforløp mellom to etterfølgende R-takk i QRS-komplekser [3, 4].

2.4 Hjertedataprogramvare

Her gis en gjennomgang av et eksisterende hjertedataprogramvare og hvilken funksjonalitet disse programvareløsningene støtter.

Det eksisterer per dags dato ingen kjente analyseverktøy som støtter Alive Technologies' eget ATS-format. Dette medfører at konverteringsverktøyet AtsConvert [48] må brukes for å konvertere ATS-filer til EDF-filer. AtsConvert inneholder i tillegg en funksjon for å konvertere ATS til CSV (Comma Separated Value format). Sistnevnte filformat lagres i rentekst, og kan behandles i de fleste teksteditorer og regneark. I tillegg kan CSV enkelt importeres og behandles i de fleste programmeringsspråk uten videre komplikasjoner.

Til forskjell fra EDF er det ved CSV-konvertering mulig å trekke ut detekterte RR-intervall. Data lagres i kommaoppdelt fil og kan etter preferanse også inneholde både RR-intervall, forløpt tid og hjerteslag.

Programvare	Grafisk plott	Annotasjonsstøtte	Eksportering av annotasjoner	RR visning	ATS-støtte
Polyman	Ja	Kun lesing	Nei	Nei	Nei
jEDF	Ja	Nei	Nei	Nei	Nei
<i>Utviklet prototype</i>	<i>Ja</i>	<i>Ja</i>	<i>Ja</i>	<i>Ja</i>	<i>Ja</i>

Tabell 1: Oversikt over støttet funksjonalitet i hjertedataprogramvare.

For visning av lagrede hjertedata eksisterer blant annet programvareløsningene Polyman [47] og jEDF [44]. Polyman har sitt opphav fra samme gruppe personer som i sin tid formet EDF-standarden, og er derfor et av de mest utbredte hjertedataverktøyene – også ved UiS. Verktøyet gjør det mulig å lese og vise EDF-filer, men mangler funksjonalitet for å gjøre annoteringer for start/slutt, samt identifisering av en episode i en hjertedatafil. Polyman kan vise annotasjoner² som allerede ligger lagret i EDF-filer. Det er derimot ikke mulig å endre

² Annotasjoner som kan leses av Polyman er av eget spesifisert format og må ikke forveksles med annotasjonsfiler generert av utviklet prototype.

eksisterende annoteringer eller gjøre nye annoteringer. Dette analyseverktøyet mangler også funksjoner for markering av perioder med støy.

På lik linje med Polyman er jEDF-verktøyet også kun beregnet for visning av EDF-filer, og mulighet for å gjøre annoteringer mangler også her. Ingen kjente, eksisterende programvareløsninger støtter visning av RR-intervall.

Målet med verktøyet som utvikles i forbindelse med denne masteroppgaven er å kombinere liknende funksjonalitet av hva overnevnte analyseverktøy kan tilby, og i tillegg skreddersy det grafiske grensesnittet for bruk i SVS-prosjektet. Dette innebærer utstrakt bruk av annoteringer og fortrinnsvis muligheten til å jobbe direkte med ATS-filer uten å være avhengig av å måtte gjøre manuelle konverteringer i forkant.

2.5 Hjertedataformat

I dette delkapittelet gis en gjennomgang av dataformatene ATS (som henholdsvis datastrømformat og filformat), EDF og EDF+.

2.5.1 ATS

ATS er Alive Technologies' eget dataformat for lagring og overføring av hjertedata. Det er et forhåndssatt standardformat som brukes sammen med *Alive Heart Monitor*. Selv om ATS ikke følger noen kjent åpen standard for lagring og overføring av hjertedata (som f. eks. EDF), er formatet åpent dokumentert slik at de som ønsker kan utvikle egne bibliotek for håndtering av formatet. Den tre sider lange formatdokumentasjonen [26] er kortfattet, men viser en beskrivelse over de ulike dataheadere og -felter som brukes i dataformatet. Sammenlignet med EDF-formatet så er det lite tredjepartsarbeid som er gjort på ATS.

2.5.1.1 ATS som trådløst datastrøm format

ATS formatet består av et datastrømformat for bruk i selve Bluetooth-overføringen fra opptaker til PC og et filformat for lagringen av datastrømmene til disk. Datastrømformatet brukes ved overføring av EKG og akselerometerdata over en trådløs Bluetooth-forbindelse. Bluetooth-forbindelsen er av type Serial Port Profile. Dette innebærer at det simuleres en tradisjonell, kablet seriell port (RS-232) for bruk i overføringen.

Data overføres i pakker som avsluttes i én byte sjekksum, og pakkestrukturen kan sees slik;

Packet Header	ECG Header	ECG Data	Acc Header	Acc Data	Checksum
6 Bytes	5 Bytes	n Bytes	5 Bytes	n Bytes	1 Byte

Tabell 2: Tabell hentet fra AHM Data Format [26, side 1]

- Packet-header inneholder en oversikt over sekvensnummer for pakker, batterinivå for den Bluetooth-baserte opptakeren, status og antall av påfølgende datablokker som skal sendes.
- ECG-header viser id³ av datastrøm, dataformat (datatype, samplingsrate og rekkevidde) og lengde av ECG-header og -data.

³ Eks. 0xAA = ECG

- Accelerometer-data representerer registrerte bevegelser hos personen som er tilkoblet AHM-opptaker. Acc-header feltene (2- og 3-axis) er bygget opp likt som ECG-header og inneholder i tillegg innfellede [29] (*interleaved*) sampler [26].

2.5.1.2 ATS som filformat

Ettersom den trådløse hjertemonitoren genererer datafiler av et AHM-spesifikt format, vil det foretrekkes å kunne bruke filer av ATS-formatet direkte i analyseverktøyet som utvikles. Dette er det to grunner til:

1. Å slippe mellomledd, som å kjøre AtsConvert i tillegg til analyseverktøyet, er den første, og minst viktige grunnen.
2. ATS-formatet inneholder ferdig detekterte R-takk fra hjertesignalet. Dette er til forskjyll fra EDF-formatet, hvor R-takk ikke inngår og må implementeres på andre måter.

Filformatet består av tre deler. Main-header, kanalbeskrivelse for hver kanal og data for de ulike kanalene [26].

1. I main-header beskrives filformatet med en id ("ATSF"), antall kanaler, dato og tidspunkt for dataregistrering, antall datablokker og lengden av disse.
2. I kanalbeskrivelsen listes de ulike datatypene; Status, ECG, 2-axis accelerometer, 3-axis accelerometer. I tillegg listes formatet til datatypene og bytelengde for hver kanals respektive datapakke.
3. I siste del lagres selve hjertedataene. Hver datatype beskrives med samplingsrate og rekkevidde. Innfellede sampler listes i tillegg for datatypene 2- og 3-axis accelerometer. Basis for denne masteroppgaven er EKG-data; accelerometer-data er ikke av relevans for SVS-prosjektet og vil derfor ikke bli videre omtalt.

En komplett, detaljert struktur er tilgjengelig fra [26].

2.5.2 EDF

EDF (European Data Format) er et standardisert filformat for bruk sammen med medisinske data. EDF er utviklet for lagring, overføring og analyse av medisinske tidsserier. Filformatet er et åpent, ikke-proprietært format og kan derfor brukes av både kommersielle og ikke-kommersielle parter [9]. Dette muliggjør f. eks. at uavhengige programvareløsninger kan lese og analysere målinger registrert av kommersielt lisensiert medisinsk utstyr.

Første utgave av EDF ble utgitt i 1992. Denne utgaven støtter ulike samplingsrater for hvert signal gjennom å tilby multikanals data. Formatet er bygget opp med en header og ett (eller flere) dataregister. Header består av generell informasjon; dataformatets versjonsnummer, pasientidentifikasjon, tidsinformasjon, antall dataregistre og antall signal i hvert dataregister. Dette er de første 256 byte av header-en. Deretter følger ytterligere 256 byte for hvert signal. Header inneholder:

$$256 + (\#\text{signaler} * 256) \text{ byte}$$

Eksempelvis, fire signaler:

$$\underline{256 + 4 * 256 = 1280 \text{ byte}}$$

EDF-formatet er kompatibelt med samme type datasignaler som ATS [9]. Et signal kan blant annet være av type EKG, kroppstemperatur og status. For hvert av disse signalene lagres signaltipe, informasjon for kalibrering av amplitude og antall samples i hvert dataregister. Samplingsfrekvens kan beregnes ut ifra sistnevnte ettersom dataregisterets tidsperiode er kjent. Denne oppbygningen gjør formatet fleksibelt med hensyn på signalers ulike samplingsfrekvenser og ut/inn-forhold (*gain*).

EDF+ standarden ble offentliggjort i 2003. EDF+ er en utvidelse av EDF og har to hovedforskjeller fra den opprinnelige standarden. EDF+ har utvidede spesifikasjoner i forhold til tilleggsinformasjon for datafeltene i formatet. De 12 ekstra spesifikasjonene i EDF+ er listet i sin helhet på [30] og omfatter blant annet påkrevd syntaks for å angi tid og dato (med hensyn til ”år 2000”-problematikk) samt anbefalt navnsetting i forhold til pasientidentifikasjon. I tillegg støtter den oppdaterte standarden lagring av flere ikke-sammenhengende opptak til en og samme fil. Av mindre forskjeller nevnes UTF-8 koding av lagrede opptak, i forhold til ASCII (som brukes i EDF) og muligheten til å lagre opptak bestående av kun annotasjoner og event-er (hendelse utført av pasient, f. eks. trykket en knapp), uten at det kreves andre signaler.

Standarden kan brukes sammen med blant annet elektroencefalogram (EEG), elektromyografi (EMG), polysomnografi (PSG) og til slutt elektrokardiografi (EKG) [9], som denne oppgaven baserer seg på.

3 Utvikling

På grunnlag av teori redegjort for i forrige kapittel, beskrives de ulike alternativene for henholdsvis hjertedataformat, valg av plott og type annotasjonslagring.

3.1 Alternativer for lagrede hjertedataformat

Alive Heart Monitor lagrer som standard hjertedata i Alive Technologies eget ATS-format. En mer utbredt variant for lagring av hjertedata er European Data Format (EDF). Ettersom det er mulig å konvertere til EDF fra standardformatet ATS, vil begge formatene i dette kapitlet vurderes som mulige løsninger.

3.1.1 Implementasjon av ATS-format

Det eksisterer allerede tredjeparts rutiner for innlesing av ATS for bruk sammen med MATLAB [20]. Fordelen med å bruke MATLAB som basis for filformat av denne typen, er at det finnes egne, ferdigdefinerte funksjoner for manipulasjon av matriser tilgjengelig i programmet (listet i Tabell 3).

MATLAB- funksjon	Forklaring
<code>zeros(m, n)</code>	Genererer en nullmatrise (alle verdier er 0) med dimensjon $m * n$.
<code>reshape(A, m, n)</code>	Genererer ny matrise ($m * n$) bestående av verdier hentet kolonnevis fra matrise A .

Tabell 3: MATLAB- funksjoner for transformasjon av matriser

Ved å bruke overnevnt eksisterende MATLAB-rutine som basis for en tilsvarende Java-parser ble det skrevet et Java-bibliotek fra grunnen som kan lese Alive Technologies' egne ATS-formaterte filer. Java-biblioteket er ferdigstilt til en viss grad, og kan brukes til å lese ut dataheadere fra ATS-filer. Uten å gå veier rundt separate matematiske biblioteker, er det i Java en nødvendighet å skrive egne metoder for funksjonene listet i Tabell 3. Som en konsekvens er implementeringen av datakanal-parsing ikke ferdigstilt, på grunnlag av tidsbruk og den viten om at det for EDF allerede eksisterer Java-bibliotek som innehar ønsket funksjonalitet.

På grunnlag av dette velges direkte ATS-parsing bort til fordel for en løsning som innebærer sømløs konvertering fra ATS til EDF via `AtsConvert`-verktøyet.

3.1.2 Relatert arbeid med EDF

Den største fordelen med å basere analyseverktøyet på EDF er at formatet har fått godt fotfeste og allerede eksisterer som en mer eller mindre offisiell standard for lagring av hjertedata.

Tidligere omtalte `jEDF` er et fritt tilgjengelig analyseverktøy for bruk sammen med EDF-filer. Programmets kildekode er lisensiert under GNU General Public License, og er kategorisert som åpen kildekode. Programmet er kryssplattformkompatibelt og kan kjøres på både Windows og Linux. `jEDF` er skrevet i Java og utviklet av Nizar Kerkeni.

Verktøyet utviklet i denne oppgaven bruker jEDFs Java-bibliotek for innlesing av EDF-formaterte filer.

3.1.3 Implementasjon av European Data Format

De importerte klassefilene *EDFFile* og *EDFSignal* [44] har ansvar for henholdsvis innlesing av EDF-fil (header- og datafelter) og håndtering av datasignalene i EDF-filen. Det mest sentrale datasignalet for denne masteroppgaven er EKG. Ved å hente ut EKG-signalet kan graf(er) genereres på grunnlag av data tolket av *EDFSignal*-klassen.

RR-intervallene detektert av hjertemonitoren inkluderes ikke EDF-formatet, men kan trekkes ut som egen CSV-fil. Hjertedataverktøyet er utviklet på en slik måte at hjertedata lagret i ATS-format automatisk konverteres til både EDF- og CSV-fil. På denne måten kombineres de aktuelle hjertedataene med ferdig detekterte RR-intervall.

3.2 Alternativer for plottløsning

I dette delkapittelet drøftes mulige løsninger for å plote hjertedata. Hver løsning vurderes ut i fra nøkkeltrekk lagt til grunn for å oppnå best mulig resultat for utviklet prototype.

3.2.1 Kriterier

Når løsning for grafisk representasjon av hjertedataene skal velges, er det noen kriterier som bør vurderes.

- Fleksibilitet
- Praktisk bruk
- Prosesseringstid
- Eventuell utviklingstid

For å gi en grafisk representasjon av de lagrede hjertedataene er det i hovedsak to muligheter; å skrive eget bibliotek for plott, eller ta i bruk eksisterende plottbibliotek.

3.2.2 Alternativer

JFreeChart

JFreeChart (JFC) er blant de største fritt tilgjengelige plottbibliotekene som eksisterer for Java. Biblioteket er veldokumentert og har en konsistent API. Det gjør det enklere å sette seg inn i forståelsen av biblioteket. I tillegg har JFC et stort og aktivt supportmiljø. Det er fleksibelt med tanke på utseende, og har flere diagramtyper tilgjengelig (12 grunntyper). JFC bruker i tillegg akseptabel prosesseringstid ved plottning (avhengig av maskinvare). Ytelse beskrives i detalj i 4.4.5.

Andre praktikaliteter:

- X- og Y-akser skaleres automatisk ut i fra størrelsen på data som legges til grunn.
- Har en innebygget mulighet for å eksportere graf til bildefil (PNG og JPEG). Det er også mulig å lagre direkte til vektorformat (SVG, PDF og EPS).

- Finnes mulighet for tooltip-funksjon som viser koordinaten for punktet musepekeren hviler på.

Skrive eget Java2D- bibliotek

Fordelene med å skrive egne metoder for tegning av grafene er muligheten til å skreddersy en løsning for bruk sammen med hjertedataene, og optimalisere med hensyn på EKG og HR. I tillegg gir denne fremgangsmåten full oversikt over virkemåten til plottbiblioteket. Å skrive et eget bibliotek er dog en affære som kan bli tidkrevende; utviklingstiden er derfor usikker, og vanskelig å anslå. Det er heller ingen garanti for å oppnå gode resultater med tanke på prosesseringsstid.

3.3 Alternativer for annotasjonsformat

Ett hjertedataopptak er ikke synonymt med å ha én kandidat. På dette grunnlag er det ønskelig å markere tidsmessig hvor i opptaket en ny kandidat blir koblet til hjertemonitoren.

En tidsperiode for én kandidat i opptaket består ikke alltid av fullt brukbare data, ettersom den kan være påvirket av ulike typer støy. Det er derfor ønskelig å markere hvor i opptaket støyområder forekommer, slik at det på et senere tidspunkt kan gjøres en vurdering om hvorvidt støyområdene må forkastes eller kan brukes i videre analyse.

I tillegg kan andre parametere endres i løpet av et opptak som er ønskelig å markere og kommentere individuelt for hver kandidat.

Opprinnelig hjertedatafil vil forbli uendret og vil aldri altereres i løpet av en annotasjonsprosess. Med utgangspunkt i dette og overnevnte beskrivelse fremtrer derfor *forløpt tid* som essensielt for oppbygningen av annotasjonsformatet. Forløpt tid vil knytte ferdigstilte annotasjoner direkte opp mot behandlede hjertedatafiler. Det er derfor viktig å finne en allsidig løsning hvor tidsdata er lett tilgjengelig for videre bruk i MATLAB.

Tradisjonelle EDF- lesere/plottere har ikke gitt brukeren mulighet til å gjøre annoteringer som nevnt over. Ved å kunne lagre annotasjonene kan markerte områder leses inn i MATLAB for videre analyse, som kan gi tilgang på informasjon som ellers ville vært utilgjengelig.

3.3.1 Kriterier

Når selve formatet for annoteringene skal velges er det noen kriterier som bør vurderes.

- Kryss-kompabilitet og portabilitet
- Praktisk bruk
- Fremtidsutsikter
- Eventuell kostnad

3.3.2 Alternativer

Excel-fil (spreadsheet)

Skriving/lesing av Excel-filer krevde tidligere spesielle, og gjerne kostnadsforbundede Java-bibliotek. Per dags dato eksisterer det flere fritt tilgjengelige tredjepartsbibliotek som gjør samme nytte. En Excel-fil for annotering er ikke videre praktisk når det kommer til å lese inn lagrede annoteringer. I tillegg er det ofte forbundet lisenskostnader med selve Microsoft Excel-programvaren.

Rentekst (komma/tabulator- delte verdier)

Rentekst-filer er i Java enkelt å både lese og skrive til, og krever ingen tilleggsbibliotek. Det finnes derimot ingen praktisk måte å jobbe med rentekstfiler, og det kreves spesielle (og omfattende) parsere hvis innholdet f. eks. skal organiseres hierarkisk. I tillegg er det ikke mulig å lese ut enkelt-noder uten å lese inn hele filen først.

Database (SQL)

Lagring av annotasjonene i en database er et av de bedre alternativene som vurderes. Dette kommer av at det i en database er enkelt å plukke ut, samt sette inn informasjon. Disse operasjonene går i tillegg hurtig. Det finnes flere moduler for bruk av database sammen med MATLAB. For MySQL-varianten er det heller ingen kostnader forbundet med bruk. MySQL er en konkurransedyktig og utbredt standard. Ettersom den er lisensiert under GNU General Public License (open source), har den også gode fremtidsutsikter med tanke på oppdateringer. Den største, avgjørende ulempen med en database er at den ikke kan medbringes "on-the-go". En database kjører som regel på én maskin og en eventuell flytting til en annen maskin fremtvinger ytterligere konfigurasjonsarbeid. Dette hemmer portabilitetskriteriet.

XML

Ved å gi annoteringsfilene en hierarkisk struktur ved bruk av XML, er det i Java enkelt å både skrive og lese data. Det samme gjelder for tolking i MATLAB. XML er en universal standard, og er per dags dato godt implementert i Java. Det finnes flere moduler som er skreddersydd for bruk av XML sammen med MATLAB. Flere av disse modulene er fritt tilgjengelige. For Java finnes to standarder for håndtering av XML; SAX (Simple API for XML) og DOM (Document Object Modell). En XML fil krever ingen støtteressurser og er dermed svært portabel. XML-filer kan i tillegg leses direkte i flere nettlesere (bl.a. Internet Explorer 7 og nyere), noe som er praktisk ved manuell sjekk av lagrede annotasjoner.

4 Eksperimentelt

I dette kapitlet beskrives og begrunnes valg gjort ut fra de ulike alternativene listet i forrige kapittel.

Valgt utviklingspråk for det grafiske grensesnittet er Java. Dette er et naturlig valg ettersom det er et programmeringsspråk som forfatter har mest erfaring med. I tillegg er det i denne sammenheng ingen andre programmeringsspråk som har andre klare fordeler i forhold til Java. Valgte løsninger for hjertedataformat har i tillegg relatert arbeid utviklet i Java og plottbiblioteket JFC er Java-basert.

4.1 Valgt format for lagrede hjertedata

Som beskrevet i kapittel 3.1 ble det på et tidlig tidspunkt besluttet å ta utgangspunkt i EDF som format for de lagrede hjertedataene. Dette valget ble tatt på grunnlag av mer dokumentert og relatert arbeid gjort med dette formatet, og i tillegg tidsbruk.

EDF inneholder ikke data for RR-intervallene i et opptak. Dette er det største ankepunktet som veier mot EDF. Dette er derimot løst ved å trekke ut RR-intervaller i en separat CSV-fil, for deretter å kombinere disse dataene sammen med signaldata lagret i EDF-fil. Denne prosessen er utviklet slik at den ikke krever noen samhandling fra brukeren av programmet, og foregår sømløst i bakgrunnen.

4.2 Valgt plottløsning

Hovedfokus i denne masteroppgaven er å kunne gjøre annotasjoner på hjertedataene. JFreeChart er en løsning som gjør det mulig å fokusere på nettopp dette, ettersom det ikke er nødvendig å bruke tid og ressurser på å utvikle egne plottløsninger, når fullgode alternativer allerede eksisterer.

På grunnlag av vurderte kriterier i delkapittel 3.2 velges JFC som basis for plotting av de lagrede hjertedataene. Av bibliotekets forskjellige diagramtyper brukes *XYLineChart*. Denne diagramtypen trekker automatisk linjer mellom datapunktene som leses inn fra hjertedatafilen. I tillegg støttes et egendefinert tidsformat⁴ for X-aksen. Dette er en nyttig egenskap som gjør det lettere å vite tidsmessig hvilken del av opptaket som plottvinduet viser.

4.3 Valgt annotasjonsformat

På grunnlag av vurderte kriterier i delkapittel 3.3 velges XML som annoteringsformat. For håndtering av XML filer vil DOM-metoden brukes. Grunnlaget for valget av DOM fremfor SAX er at SAX er mest passende til mindre, enkle oppgaver (som f. eks. søk etter én verdi i en fil). Ulempen med SAX er derimot at den ikke kommer med en ferdig implementert løsning for å bygge trestruktur av XML-filer. DOM tilbyr denne funksjonaliteten fra start. En umodifisert SAX-parser vil kun kjenne den sist brukte verdien, og bare denne vil være disponibel i minnet. Dette bygger også opp under valget av DOM.

⁴ SimpleDateFormat. Her satt til å vise tid som *H:MM:SS.sss* (eks: 1:23:45.678).

4.3.1 Annotasjonsstruktur

Strukturen til annotasjonsfilene er konstruert slik det kan lagres informasjon om flere kandidater, og hver kandidat kan ha flere områder merket som støy eller ekstrasystoler.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ekg_annotasjoner KildeFil="">
  <Kandidat>
    <KandidatID></KandidatID>
    <TidsromStart></TidsromStart>
    <TidsromSlutt></TidsromSlutt>
    <Scenarie></Scenarie>
    <Tilleggsopplysninger></Tilleggsopplysninger>
    <MarkertStøy>
      <Tidsrom>
        <Start></Start>
        <Slutt></Slutt>
      </Tidsrom>
    </MarkertStøy>
  </Kandidat>
</ekg_annotasjoner>
```

Figur 8: Annotasjonsfilens struktur (uten data)

Foreldernoden `<ekg_annotasjoner>` har `<Kandidat>` som barn. `<Kandidat>` har flere felter som barn (kandidatens navn eller ID, tidsrom for kandidaten, scenario for opptaket, og et ekstrarfelt for tilleggsinformasjon). `<Kandidat>` har også `<MarkertStøy>` som barn.

`<MarkertStøy>` er foreldernode til barnnoden `<Tidsrom>`. Hver `<Tidsrom>` node er forelder til barnnodene `<Start>` og `<Slutt>`, som sammen representerer ett markert støyområde.

Ved lagring av flere støyområder vil den aktuelle `<Kandidat>` noden inneholde en barnnode `<MarkertStøy>` med flere `<Tidsrom>` noder. Hvert av disse `<Tidsrom>` -nodene vil inneholde ett støytidsrom- start/slutt par for hvert støyområde. Dersom `<Start>` og `<Slutt>` -nodene har samme tidsverdi betyr det at seksjonen er en ekstrasystol.

Verdiene som lagres i nodene `<TidsromStart>`, `<TidsromSlutt>`, `<Start>` og `<Slutt>` (for markert støy) er antall forløpte millisekunder med utgangspunkt i at hjertedatafil starter på 0 millisekunder.

Et eksempel på annotasjonsfil bestående av to kandidater med henholdsvis tre og to støyområder vises under:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ekg_annotasjoner KildeFil="D:\hjertedata.edf">
  <Kandidat>
    <KandidatID>Kandidat1</KandidatID>
    <TidsromStart>46.67</TidsromStart>
    <TidsromSlutt>1185796.67</TidsromSlutt>
    <Scenarie>Stresset</Scenarie>
    <Tilleggsopplysninger>M24</Tilleggsopplysninger>
    <MarkertStøy>
      <Tidsrom>
        <Start>193546.67</Start>
        <Slutt>206956.67</Slutt>
      </Tidsrom>
    </MarkertStøy>
    <MarkertStøy>
      <Tidsrom>
        <Start>650626.67</Start>
        <Slutt>672286.67</Slutt>
      </Tidsrom>
    </MarkertStøy>
    <MarkertStøy>
      <Tidsrom>
        <Start>1098916.67</Start>
        <Slutt>1109986.67</Slutt>
      </Tidsrom>
    </MarkertStøy>
  </Kandidat>
  <Kandidat>
    <KandidatID>Kandidat2</KandidatID>
    <TidsromStart>1298866.67</TidsromStart>
    <TidsromSlutt>2160013.33</TidsromSlutt>
    <Scenarie>Avslappet</Scenarie>
    <Tilleggsopplysninger>K20</Tilleggsopplysninger>
    <MarkertStøy>
      <Tidsrom>
        <Start>1509916.67</Start>
        <Slutt>1543066.67</Slutt>
      </Tidsrom>
    </MarkertStøy>
    <MarkertStøy>
      <Tidsrom>
        <Start>2076676.67</Start>
        <Slutt>2089966.67</Slutt>
      </Tidsrom>
    </MarkertStøy>
    <MarkertStøy>
      <Tidsrom>
        <Start>2127286.67</Start>
        <Slutt>2127286.67</Slutt>
      </Tidsrom>
    </MarkertStøy>
  </Kandidat>
</ekg_annotasjoner>
```

Figur 9: Ferdig eksportert annotasjonsfil med eksempeldata.

Ettersom én annotasjonsfil alltid vil basere seg på én hjertedatafil (med tilhørende RR-intervall fra CSV), refereres denne hjertedatafilen til som attributt for rotnoden `<ekg_annotasjoner>`. I tillegg til å fungere som praktisk informasjon for brukeren, med å beskrive hvilken hjertedatafil som ble brukt under annoteringen, så muliggjør dette attributtet også at utviklet programløsning kan gjenoppta en annotasjonssesjon på senere tidspunkt. Rent praktisk fungerer dette som en lagre/last-funksjon (*save/load*).

4.4 Programdesign

Momenter som tas stilling til ved design av programløsningen er brukervennlighet, målgruppe, fleksibilitet og pålitelighet.

4.4.1 Brukervennlighet

Verktøyets brukervennlighet avhenger av hvor enkelt og intuitivt det er å bruke. Sammen avgjør disse faktorene hvor effektivt verktøyet faktisk er i bruk. Det er hensiktsmessig å fokusere på implementeringer som gjør annoteringsprosessen enklere og raskere.

For manøvrering i et opptak implementeres derfor panorering ved hjelp av musens scrollhjul. Panorering frem i tid kan gjøres ved å rulle scrollhjulet fremover (opp), og motsatt for panorering tilbake i tid. Dette vil gjøre panorering mer tilgjengelig i forhold til å måtte bruke dedikerte navigasjonsknapper. For datamaskiner der mus med scrollhjul ikke er tilgjengelig (f. eks. bærbar PC med touchpad), inkluderes likevel et sett med navigasjonsknapper som kan nyttes.

Lagring av start- og sluttverdier for annoteringer vil stå sentralt i bruken av programmet. Dette implementeres slik at verdiene lagres ved henholdsvis dobbelklikk og enkelklikk på EKG-grafen. Dette medfører at start- og slutfeltene vil autofylles med de respektive tidsverdiene.

En programvareløsnings brukervennlighet begrenses også av hvor mange valg brukeren må ta stilling til ved bruk. Mulige brukerdefinerte parametere er i denne masteroppgaven mange. Et viktig mål er derfor å finne en balanse mellom et omfattende verktøy og et enkelt verktøy uten å miste muligheten til å påvirke essensielle funksjoner.

4.4.1.1 Egendefinerte parametere

På basis av samtaler med målgruppe (SVS) er følgende parametere prioritert med hensyn på muligheten til å egendefinere:

- størrelse på tidsvindu for grafene
- skalapreferanser for HR
- invertering av EKG-graf
- angi panoreringsfaktor fra et tidsvindu til neste/forrige
- lagringskatalog
- eventuell korrigerende av grafenes x-verdier (offset)
- justering (eller deaktivering) av ytelsesfremmende funksjoner

I tillegg er det gitt mulighet til å endre tidssone for opptak, samt spesifisering av katalogsti til AtsConvert. Førstnevnte kan brukes for justering av opptak som gir feil på tidsaksen (eksempelvis dersom tidsakse starter på 01:00:00 i stedet for 00:00:00).

4.4.1.2 Forhåndsdefinerte parametere

- **Samplingsrate:** AHM-opptakeren lagrer alltid ATS- data med en samplingsrate på 300 for EKG og 1 for HR [27]. Ettersom prototypen kun skal brukes sammen med ATS-filer er disse verdiene ferdigdefinert i programmets kildekode.
- **GUI-parametere** er mindre viktige å kunne egendefinere. Det viktigste er her at det er enkelt å skille grafene fra hverandre.
- **Signalplott:** EKG og HR er i dette prosjektet de eneste relevante datasignalene å representere med plott.
- **Signalakser:**
Y-akse for HR. Y-verdier (BPM) til HR-signalet har nytteverdi i en annotasjonssesjon og en dedikert y-akse inkluderes for HR på bakgrunn av dette. Ettersom det i SVS ikke er mulig å trekke konklusjoner ut ifra antall millivolt EKG-signalet består av til en hver tid, er EKG-signalets y-akse uinteressant. Det er derimot EKG-grafens faktiske form (plott) som er av interesse. EKG-signalets y-akse inkluderes derfor ikke i utviklet prototype. Grafens y-verdier kan dog fortsatt leses via tooltip. Til gjengjeld frigjøres piksel plass til et større plottområde i forhold til å ha flere y-akser.
- **Felles x-akse over forløpt tid.** Denne x-aksen er relevant for både HR- og EKG-signalet.
- **Tidsformat** for x-akse er satt til *time:minutt:sekund.millisecond* ("H:mm:ss.S").

4.4.2 Målgruppe

I tillegg til overnevnte parametere er det nødvendig å ta stilling til forkunnskaper hos sluttbruker. Det er markant forskjell på programvare utviklet for ordinære brukere og en spesifikk målgruppe, ettersom logikkandelen vil deles mellom henholdsvis programvaren og brukeren selv. Dette kan deles i to scenario, A og B:

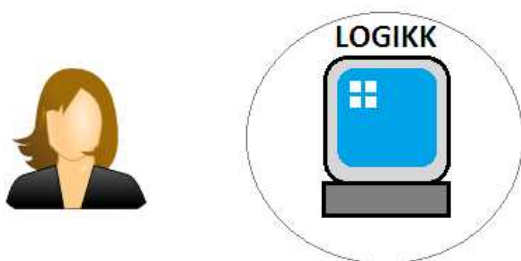


Fig. 10: A) Program sørger for all logikk

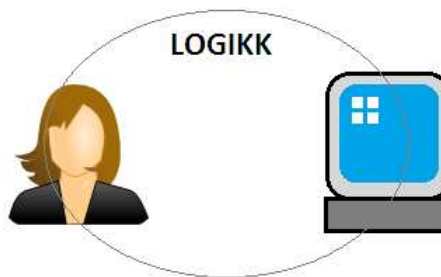


Fig. 11: B) Logikk delt mellom bruker og programvare

I A-scenariet (vist i Fig. 10) er det vanlig å bruke større andel utviklingsressurser på å sørge for at brukeren opererer programvaren riktig, i forhold til B hvor brukerterskelen settes høyere. Til gjengjeld er det i B-scenariet en mulighet å nytte sparte utviklingsressurser på andre deler av implementeringen. I noen tilfeller vil det være helt nødvendig å implementere programvare etter B. Det kan komme av at logikken som skal utføres er for kompleks for programvaren alene å utføre.

Ettersom verktøyet i denne masteroppgaven er ment for bruk av personell involvert i forskningsprosjektet SVS, antas fysiologisk kompetanse hos sluttbruker. På grunnlag av dette kan grensesnittets brukerterskel settes ved et fysiologisk nivå (B-scenariet). Følgende antakelser kan derfor gjøres:

Sluttbruker besitter kunnskap nok til å vite at...

1. ... det ikke er hensiktsmessig å annotere ekstrasystoler og/eller støyseksjoner utenfor kandidatens tidsintervall. Dersom ingen kandidat er koblet til vil det heller ikke eksistere hjertedata verdt å analysere.
2. ... det ikke er hensiktsmessig å sette sluttpunktet for et tidsintervall (kandidat og/eller støyseksjon) før startpunktet for det samme tidsintervallet (og omvendt). Dette vil være en tidsmessig umulighet.
3. ... det ikke er hensiktsmessig å overlape tidsintervallet mellom to eller flere kandidater. Det kan kun være en kandidat koblet til hjertemonitoren om gangen.

4.4.3 Fleksibilitet

Et hjertedataopptak kan variere i lengde. Flere kandidater vil ofte medføre lengre opptak. AHM har en maksimumsgrense på fem dager sammenhengende opptak via internminnet i enheten [27]. Denne maksimumsgrensen kan økes opp til åtte ganger ved bruk av Secure Digital-minnekort (opp til 2 GB i lagringskapasitet). Den påfølgende annotasjonssesjonen vil være direkte avhengig av varigheten på opptaket. For opptak som spenner over lengre tidsperioder er det derfor viktig å ha mulighet til å lagre annotasjonsarbeidet før annoteringen er ferdigstilt.

I løpet av en annotasjonssesjon er det også viktig å operere uavhengig av et tidsvindu etter påbegynte kandidat- eller støyannoteringer. Det skal være mulighet for å påbegynne en annotasjon ved å sette startverdi i et tidsvindu, for så å kunne panorere til et annet tidsvindu (og/eller endre størrelsen på tidsvinduet) hvor sluttverdien kan settes.

4.4.4 Pålitelighet

En lagringsfunksjon som muliggjør lagring av annotasjonsarbeid vil også bidra til å gjøre programvaren mer pålitelig, ved at det er mulig å mellomlagre for så å innhente lagrede data i etterkant av et eventuelt systemkræsje.

For å unngå uforutsette tap av data ved lagring vil det være viktig å få en indikasjon dersom en fil allerede eksisterer. En lagringsprosess vil avvente tilbakemelding fra bruker om å overskrive eller bruke et alternativt ikke-eksisterende filnavn.

Ved annotering ønskes konsistens i bakomliggende konstruksjon av XML-fil. Dette sikres ved at tekstfelter beregnet for annoteringsdata aktiveres/deaktiveres i løpet av en annotasjonssesjon. Disse tekstfeltene skifter tilstand flere ganger avhengig av hvilke deler av annoteringen som er fullført. Dersom et felt er aktivert, må dette fylles ut før det er mulig å gå videre. Dersom det forsøkes å gå videre uten å fylle inn et tekstfelt, vil feltet bli merket med rød farge frem til det er korrekt utfyllt. Dersom et felt er deaktivert, er det ikke mulig å fylle inn informasjon i det. For at feltet skal aktiveres må bruker gjøre seg ferdig med påbegynt

annotasjon. Eksempelvis er det ikke mulig å kunne starte støyannotering uten først å ha opprettet en kandidat.

Dersom egendefinerte parametere blir definert feil og medfører at programmet ikke fungerer korrekt, vil det være hensiktsmessig å tilbakestille parametrene til sine respektive standardverdier. På grunnlag av dette inkluderes et tilbakestill-valg i innstillingspanelet.

4.4.5 Ytelse

Ytelsesmomentet ved programdesign er i denne oppgaven sentralt. Det er ønskelig å legge best mulig til rette for et effektivt arbeidsverktøy. Det er viktig at panorering frem og tilbake i opptak, samt inn- og utskalering ikke påvirkes negativt av for lange prosesseringstider.

4.4.5.1 Visning av RR-intervall

Grunnet antall datapunkter som må tegnes i tidsvindu større eller lik 5 minutter, så deaktiveres opptegning av RR-intervaller for tidsvindu av denne størrelsen. Dette gjøres på grunnlag av at større tidsvindu medfører at RR-intervallene vil stå tettere og tettere, og til slutt tegnes som en lang, tilsynelatende sammenhengende linje. Når dette skjer er det ikke lenger noen nytteverdi i visning av RR-intervaller.

Denne restriksjonen medfører i tillegg forbedret overall ytelse ved plotting for tidsvindu av denne størrelsen. Etersom prosesseringskraft ment for plott av RR frigjøres vil total prosesseringstid reduseres.

4.4.5.2 Utnytte sammenfallende punkter

PC-ens skjermoppløsning vil bli en begrensning ved visning av større tidsvindu. Nærliggende datapunkter ved større tidsvindu vil, avhengig av oppløsning, tegnes på tilsynelatende samme sted. Sammenfallende punkter (SFP) kan derfor utnyttes til å bidra positivt til ytelsesmomentet.

Utviklet prototype får dermed implementert en utviklet SFP-funksjon som aktiveres for tidsvindu fra og med 10 minutter i størrelse. Denne funksjonen sørger for at punkter som fremstår som samme punkt utelates fra plott. Størrelsen på tidsvinduet tilsier hvor mange punkter som faktisk blir synlige, ergo hvor mange punkter som kan kuttes ut i plottingen.

Eksempelvis medfører dette at fire nærliggende punkter i grafen tegnes som ett punkt, for et tidsvindu på 10 minutter. For et tidsvindu på 20 minutter dobles SFP-faktoren til åtte punkter, og så videre.

Testing

For å illustrere effekt av overnevnt endring ved plotting ble prototypen satt til å tegne opp tidsvindu på henholdsvis 10 og 20 minutter med og uten plott av sammenfallende punkter. Prosesseringstid for hver metode ble registrert etter fullført plott. Identisk hjertedatafil ble brukt som utgangspunkt for begge målingene, og PC med Intel Centrino 1,7 GHz med 1 GB systemminne ble brukt under testing.

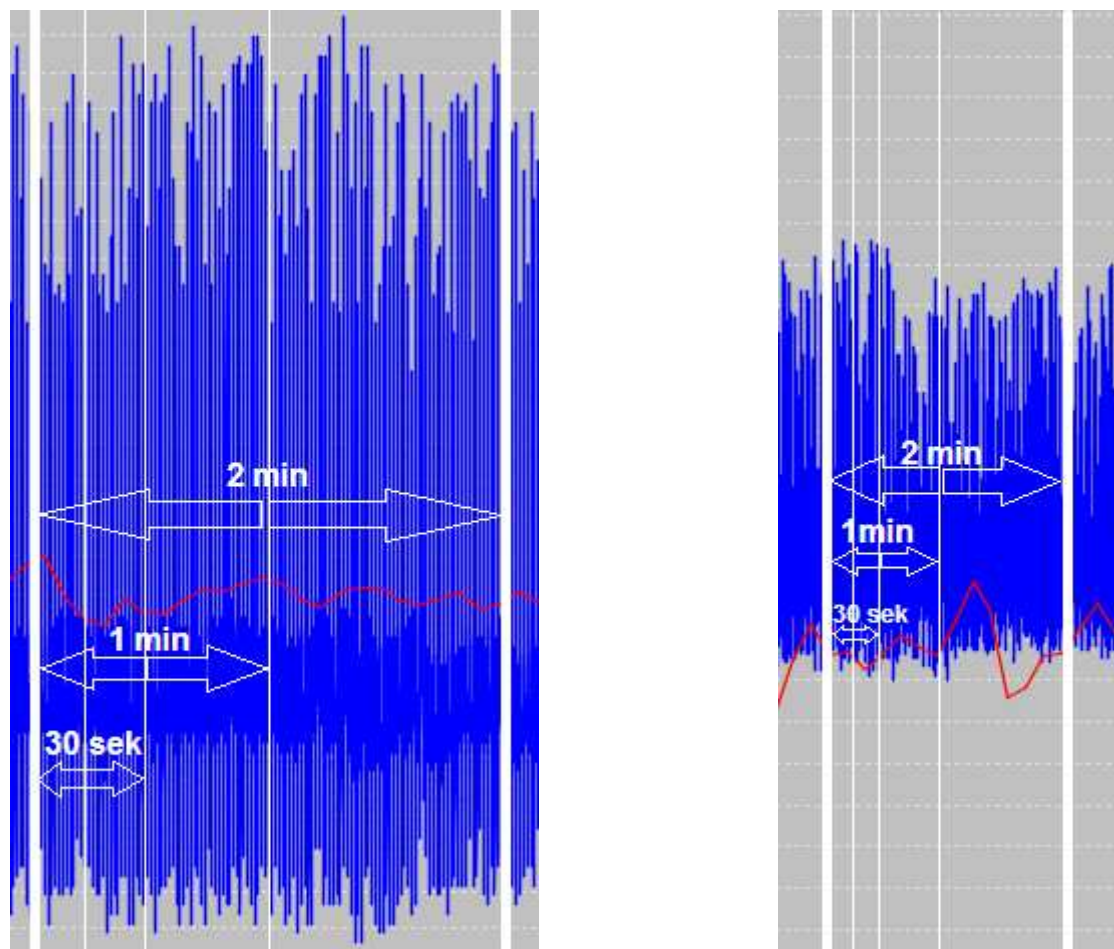
Tidsvindu	Prosesseringstid [sekund]		% forbedring
	SFP-funksjon deaktivert	SFP-funksjon med faktor = 4	
10 minutter	35,5	7,7	78,3
20 minutter	84,0	10,2	87,8

Tabell 4: Prosesseringstid for plott med og uten sammenfallende punkter

Plotting uten sammenfallende punkter medfører en registrert ytelsesforbedring under plotting på opp til 87,8 %, og er helt essensielt for prosesseringstiden til prototypen.

Denne type ytelsesforbedring kan medføre en form for klipping [25] av signalet, avhengig av SFP-faktor. En økning av SFP-faktor medfører økning av antall punkter som ikke vil bli plottet. Gjennom testing ble en SFP-faktor på 4 (per 10 minutters intervall) funnet akseptabel og dermed satt som standardverdi. Som nevnt i 4.4.1.1 kan denne faktoren justeres eller deaktiveres i innstillingspanelet.

En deaktivering av SFP-funksjonen medfører opptegning av alle punkter på skjermen, uavhengig av størrelse på tidsvindu. Men ettersom det vil tegnes et såpass stort antall punkter på skjermen samtidig, er tidsvindu for større intervaller (> 10 min) likevel ikke optimale for å gjøre nøyaktige annoteringer. En figur som illustrerer denne situasjonen vises under:



Figur 12: Figurene viser ½, 1 og 2 min. utsnitt fra tidsvindu på henholdsvis 10 minutter (venstre) og 20 minutter (høyre).

Som illustrert i Figur 12, vil det være vanskelig å etterstrebe nøyaktig annotering ved visning av større tidsintervall. På grunnlag av dette vil en aktivering av implementert SFP-funksjon for disse tidsintervallene ikke medføre nevneverdig forringing av annotasjonsresultatet.

En innskalering til f. eks. 5 minutters tidsvindu vil ikke trigge SFP-funksjonen. I tillegg vil dette tidsvinduet tilsi mindre antall datapunkter på samme skjermbredde. Dermed muliggjøres mer nøyaktig annotering.

JFCs zoom-funksjon (via musen) opererer uavhengig av utviklet programlogikk. Bruk av denne når SFP-funksjon er aktivert vil hverken gi best eller korrekt representasjon av opprinnelige data. Derfor anbefales ikke JFC-zoom i kombinasjon med annotering for tidsvinduer fra og med 10 minutter i størrelse. Ved annotering anbefales i stedet skalering på normalt vis via kombinasjonsboksen nederst til venstre i annotasjonspanelet (beskrives i detalj i 5.4.3.2).

4.4.5.3 Ytelse ved punktinnsetting

Standardinnstillinger for JFC plott tilsier at hvert punkt som tegnes på skjermen skal gi tilbakemelding (callback av datatype *boolean*) til JFC-API om punktet kan tegnes innenfor det aktuelle dataområdet. Dette gjøres ved lineær iterering gjennom alle tidligere opptegnede punkter. Dette er en enkel metode å implementere ettersom alle punkter har lik innsettingsmetode. Den er dog kun egnet for mindre mengder datapunkter.


```
dataserie.add(hr_x, hr_y);
```

Figur 13: Standard metode for innsetting av punkter

For hvert punkt som settes inn (med *add()*-metoden) økes innsettingstiden, noe som vil medføre en betraktelig økning i kjøretid ved større antall datapunkter. Dette betyr at for opptegning av et punkt N vil medføre lineær iterering gjennom alle $N-1$ foregående punkter som allerede er tegnet opp. Et punktantall på eksempelvis 400 000 punkter innebærer at innsetting av det siste punktet vil medføre lineær iterering gjennom de foregående 399 999 punktene. Med overnevnt mengde av datapunkter vil plotting være en svært tidkrevende affære.

I stedet for å gi tilbakemelding til JFC-API for hvert opptegnede punkt, implementeres prototypen til å kun gi tilbakemelding på det siste tegnede punktet. Dermed sjekkes foregående punkter kun én gang [23].

```
dataserie.add(hr_x, hr_y, false);
(...)
dataserie.add(hr_x, hr_y, true);
```

Figur 14: Tilbakemelding kun på siste opptegnet punkt

Testing

For å illustrere effekten av overnevnt endring i punktinnsetting ble prototypen ble satt til å tegne opp tidsvindu på henholdsvis 1 og 10 minutter med overnevnte innsettingsmetoder. Prosesseringstid for hver metode ble registrert etter hvert fullførte plott. Identisk hjertedatafil ble brukt som utgangspunkt for begge målingene, og PC med Intel Centrino 1,7 GHz med 1 GB systemminne ble brukt under testing.

Signal	Prosesseringstid [sekund]		% forbedring
	Callback på alle punkter	Callback på siste punkt	
EKG	48,227	0,057	99 %
HR	0,350	0,021	94 %
Total	48,577	0,078	99 %

Tabell 5: Punktinnsetting i 1 minutt tidsvindu

Signal	Prosesseringstid [sekund]		% forbedring
	Callback på alle punkter	Callback på siste punkt	
EKG	182,499	0,115	99 %
HR	1,210	0,034	97 %
Total	183,709	0,149	99 %

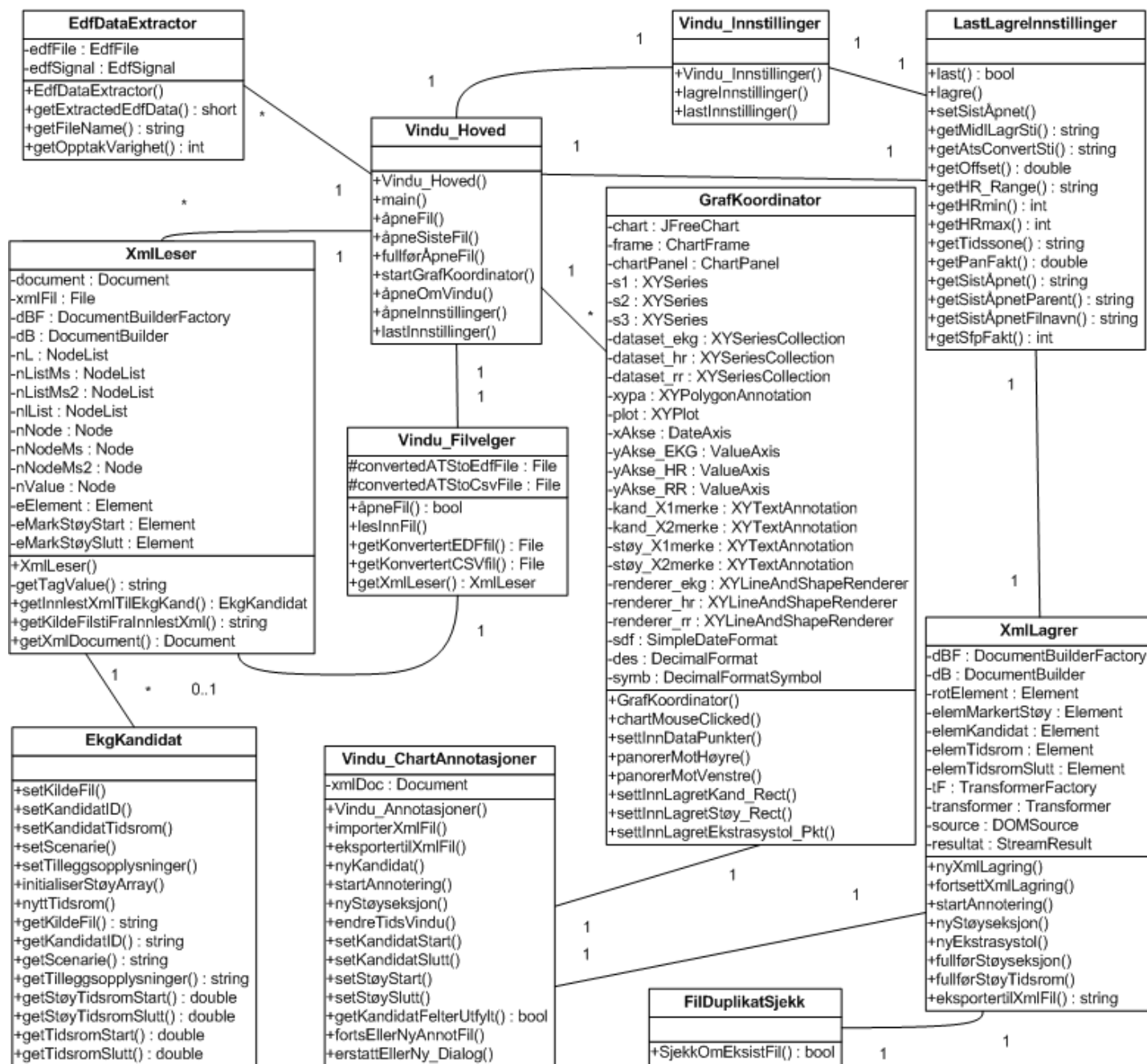
Tabell 6: Punktinnsetting i 10 minutt tidsvindu

Tester som er utført viser nedslående utfall i form av forbedret prosesseringstid. Dette rettferdiggjør fjerning av callback for nesten samtlige datapunkter (bortsett fra det siste). Videre testing med kombinasjon av ulike tidsvindu og åtte andre hjertedatafiler viser ingen antydning til irregulær oppførsel ved bruk av denne metoden.

5 Resultat

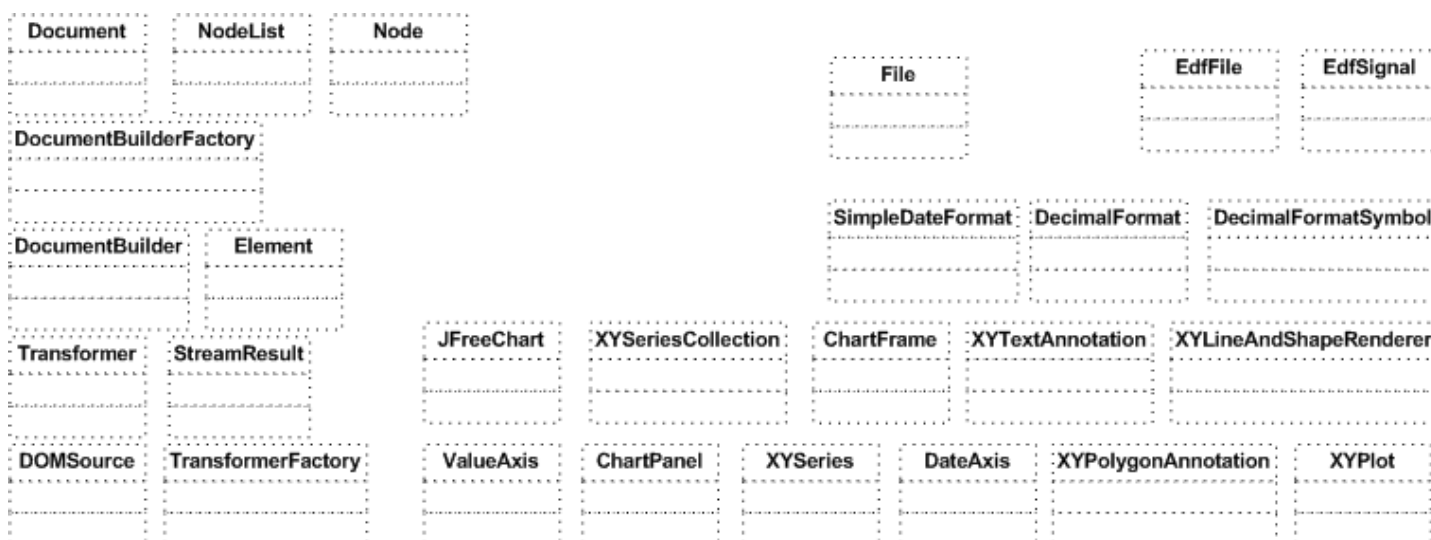
Dette kapittelet inneholder en oversikt over programstruktur i form av klassediagram i første delkapittel. Videre vises flytdiagram over implementert løsning. En teknisk gjennomgang forklarer prinsipiell virkemåte i 5.3. En brukerveiledning er inkludert for å demonstrere og forklare virkemåte til utviklet prototype. Til slutt listes tilleggsinformasjon relevant for prototypen.

5.1 Klassediagram



Figur 15: Klassediagram⁵ over implementert prototype.

⁵ Java Swing/AWT-spesifikke klasser inkluderes ikke i klassediagram.

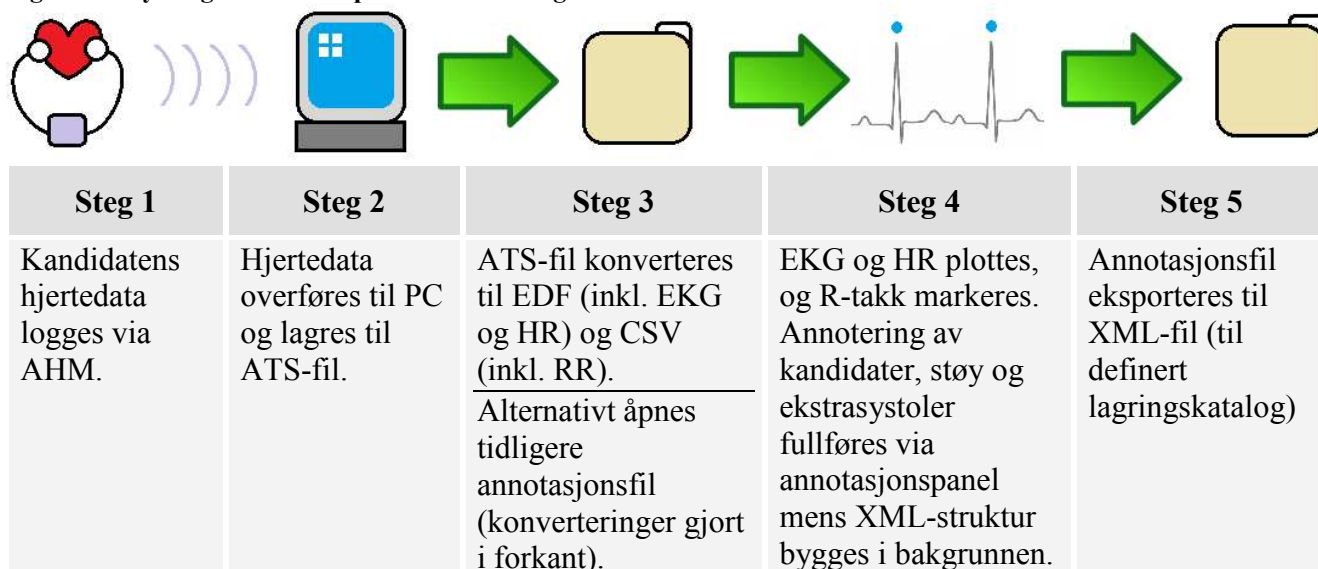


Figur 16: Klassediagram forts. Importerte klasser brukt i implementert prototype.

5.2 Flytdiagram over implementert løsning

Endelig resultat av flytdiagram som ble vist innledningsvis i rapporten (Figur 1) kan sees i Figur 17, vist under:

Figur 17: Flytdiagram over implementert løsning:



5.3 Virkemåte av utviklet programløsning

1. Lagrede innstillinger lastes via *LastLagreInnstillinger*-klassen.
 - Innstillinger kan egendefineres via *Vindu_Innstillinger*-klassen.
2. Ønsket hjertedatafil åpnes i programmet
 - *Vindu_Hoved*-klassen kaller *Vindu_FilVelger*-klassen
 - Etter ønsket ATS/XML-fil er valgt kaller *Vindu_Hoved*-klassen *EdfDataExtractor*- og *CsvDataExtractor*-klassene.

3. Dersom XML-format → kaller opp *XmlLeser*-klassen
4. Dersom ATS-format → kjører *AtsConvert* opp i bakgrunn og konverterer til EDF og CSV (for RR-intervall).
5. Data fra EDF-fil leses inn ved at *EdfDataExtractor*-klassen kaller opp klassene *EDFFile* og deretter *EDFSignal*.
6. Brukeren kan så trykke ”Tegn graf” for å få generert grafene ut ifra innlest EDF- og CSV-data. *GrafKoordinator*-klassen vil da kalles.
7. *Vindu_Anotasjoner*-klassen viser et annotasjonspanel på nedre del av skjermen. I tillegg til annoteringshåndtering, administreres også panorering og skalering av tidsvinduet, samt invertering av grafen fra annotasjonspanelet.
8. Annoteringer for aktuell kandidat og eventuelle støyseksjoner (og ekstrasystoler) lagres via musepekeren:
 - Dobbelklikk = Startverdi
 - Enkelklikk = Sluttverdi
 - Brukeren kan markere om støyseksjonen skal gjelde kun ett punkt (for å markere ekstra hjerteslag; ekstrasystol).
 - Annoteringene bygges automatisk hierarkisk i bakgrunnen av annotasjonssesjonen.
9. Annoteringer kan eksporteres til MATLAB-kompatibelt format.
 - *FilDuplikatSjekk*-klassen brukes for å detektere eventuelle eksisterende filnavn.
 - *XmlLagrer*-klassen oppretter en XML-fil inneholdende annotasjonene.
10. Eksporterte annotasjonsfiler kan leses inn for senere visning og/eller fortsetting av annotasjonene via *XmlLeser*-klassen.
 - Rotnodens attributt brukes til å finne sti til opprinnelig EDF-fil.
 - Innleste annoteringer lagres i *EkgKandidat*-klassen, og inneholder metoder for å lagre/hente data (kandidat-id, tidsseksjoner for kandidat, samt støy, ekstrarfelter og scenarie). Hver annotert kandidat representeres med et eget *EkgKandidat*-objekt.

5.4 Brukerveiledning

Dette delkapittelet inneholder en detaljert brukerveiledning for prototypen bestående av systemkrav, skjermbilder og kommentarer.

5.4.1 Systemkrav

- **Anbefalt plattform: Windows.**
Ettersom konverteringsverktøyet *AtsConvert* kun finnes kompilert for Windows, derfor kreves Windows-plattform for å åpne ATS-filer i utviklet prototype.
- **Minimumskrav plattform: Alle OS med JRE.**
I situasjoner der ATS-kompabilitet ikke er en nødvendighet, vil ikke prototypen begrenses av *AtsConvert*, og kan kjøres på alle plattformer som støtter standard JRE.
- Minimum skjermopløsning: 1024x768 piksler
- Anbefalt skjermopløsning: 1280x768 piksler (gir større pikselbredde til annotasjonspanel)
- Datamus med scrollhjul anbefales. Ordinær 2-knapps mus er et minimumskrav.

5.4.2 Oppsett og innstillinger

Prototypen kan bruke standard- eller egendefinerte innstillinger.

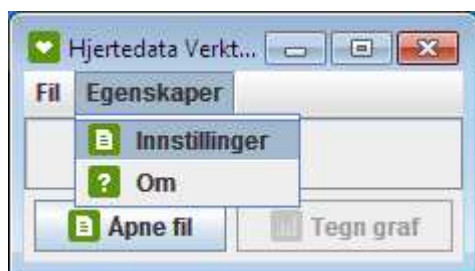


Figur 18: Ved førstegangskjøring møtes brukeren av følgende dialogboks.

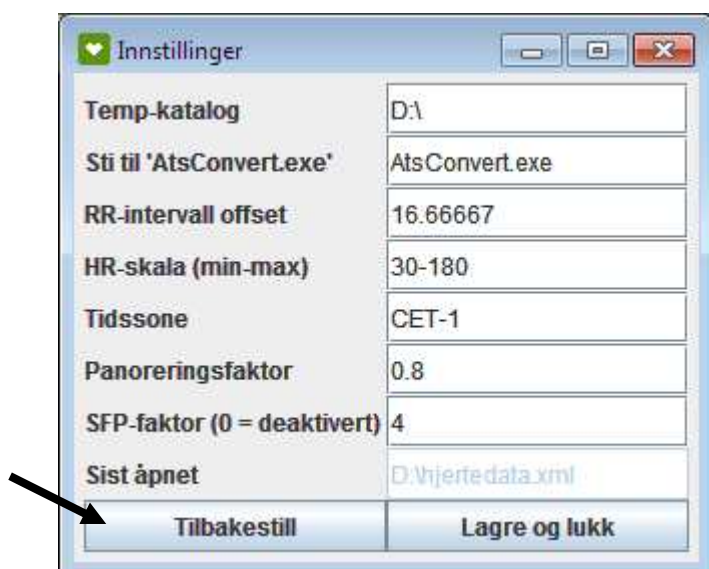


Figur 19: Programinnstillinger kan endres i innstillingspanelet som vises.

1. Temp-katalog angir katalog hvor genererte filer vil bli lagret. Katalognavn må avsluttes med "\" tegn.
2. Sti til AtsConvert-programmet kan egendefineres.
3. For å få korrekt oppmerking av RR-intervallene er det lagt inn en offsetverdi. Standardverdien for denne er i de aller fleste tilfeller akseptabel. Skilletegn for offset er satt med punktum. Bruk av andre skilletegn vil gi feilmelding.
4. Minimums- og maksimumsverdier for hjerteratens y-akse kan settes i det nederste feltet. Skilletegn for hjerterate er satt med bindestrek.
5. Tidssone kan endres for hjertedatafiler der tidsaksens timeparameter ikke vises korrekt.
6. Panoreringsfaktor angir prosentandel (fra 0.0 til 1.0) panorering fra et tidsvindu til det neste.
7. Sammenfallende punkter (SFP) -faktor angir antall punkter som skal utelates (per 10 minutt tidsintervall). Denne er kun aktivert for tidsvindu fra og med 10 minutter i størrelse. SFP kan deaktiveres ved å bruke verdi 0.



Figur 20: Programinnstillingene kan endres på et senere tidspunkt via egenskaper-menyen.



Figur 21: Innstillinger kan resettes via tilbakestill-valget.

5.4.3 Funksjonalitet i mus

I bruk av verktøyet kan musen og dens implementerte funksjonalitet være et effektivt hjelpemiddel.

5.4.3.1 Panorering



Figur 22: For å panorere frem og tilbake i opptaket brukes scrollhjulet på musen. Panorering kan også styres via navigasjonsknappene markert nederst til venstre i annotasjonspanelet i Figur 29. Dette er en nødvendighet for mus uten scrollhjul.

Panoreringsfaktor er som standardverdi satt til 80 %. I praksis for en høyrepanorering, betyr dette at de *siste* 20 % av et panoreringsbilde vil tegnes opp i neste panorering som de *første* 20 %. Dette er implementert for å lettere kunne holde oversikt og koordinasjon fra en panorering til den neste.

5.4.3.2 Zoom



Figur 23: Det er mulig å forstørre/forminske en valgt seksjon av grafen (zoom) ved å holde inne en vilkårlig museknapp og dra pekeren ned mot høyre i ønsket størrelse (1). For å gå tilbake til normal aspekt holdes en vilkårlig museknapp inne mens musepeker flyttes opp mot venstre (2).

5.4.3.3 Tooltip



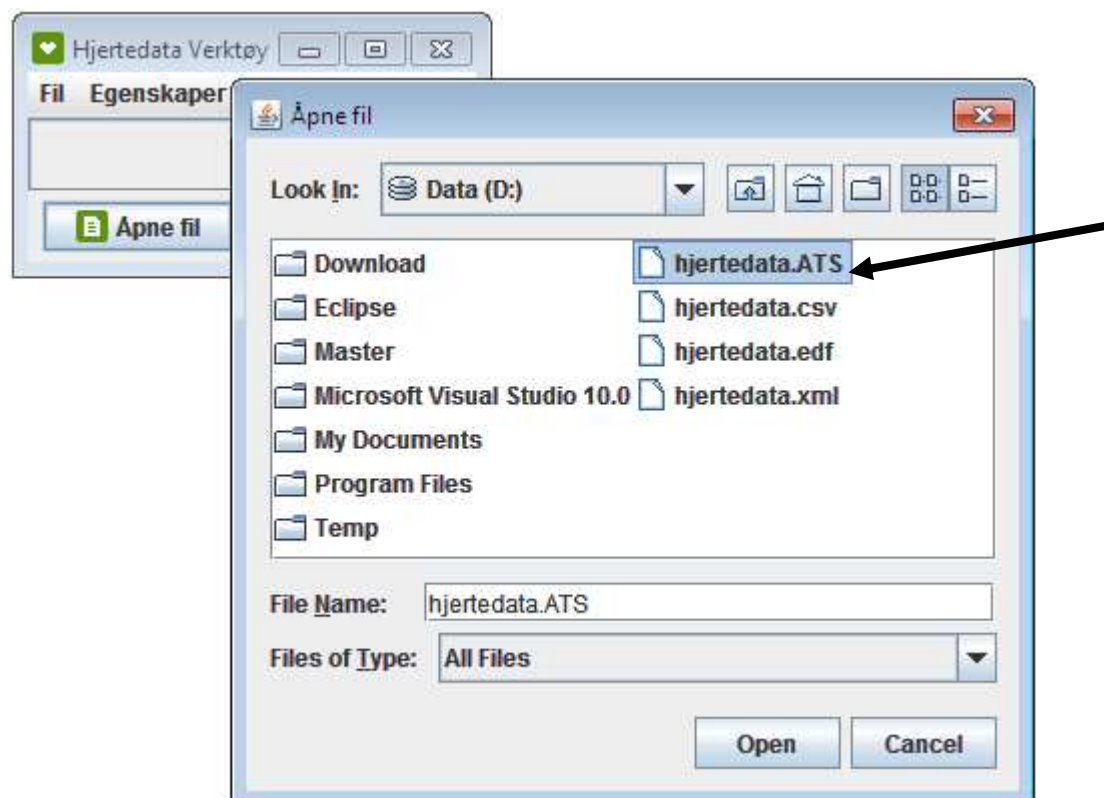
Figur 24: Det er mulig å lese koordinatens data ved å holde musepekeren over punktet. En tooltip-verdi er delt opp i forløpt tid (i *Timer:Minutter:Sekunder.Millisekunder*-format) og (X,Y)-data.

5.4.4 Åpne hjertedata (ATS) for annotering

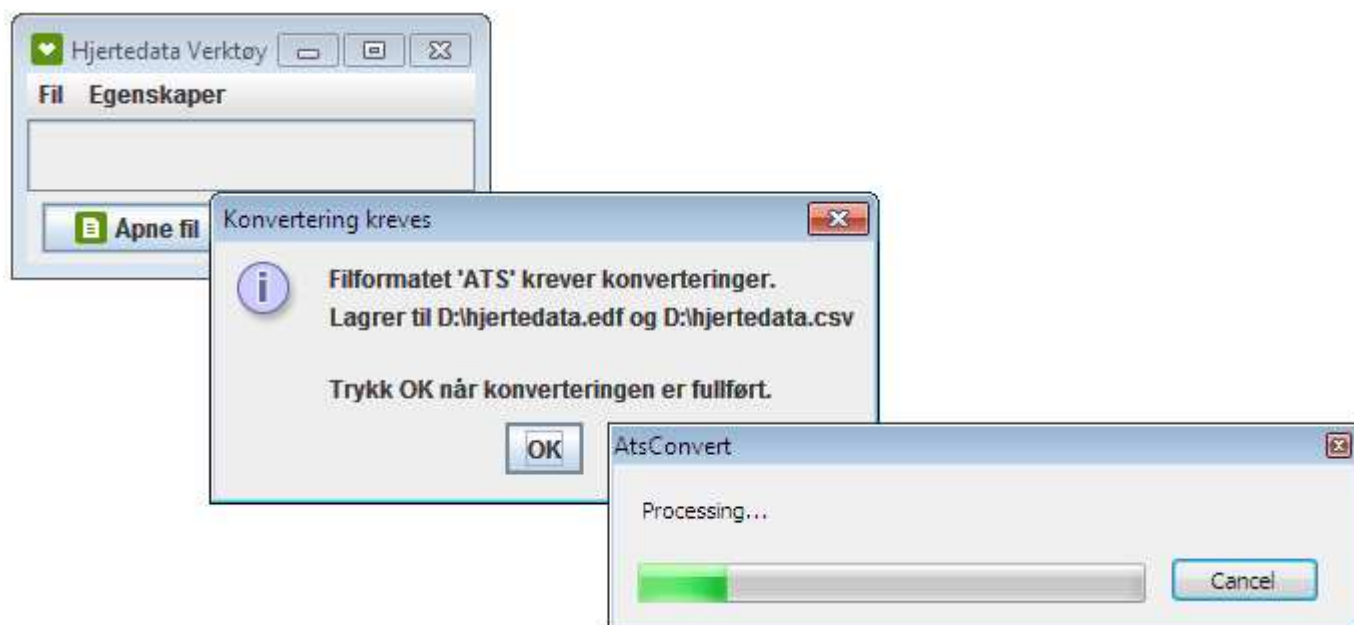
Ved førstegangsannotering vil hjertedataopptaket være av ATS-format.



Figur 25: Hovedvindu ved oppstart.



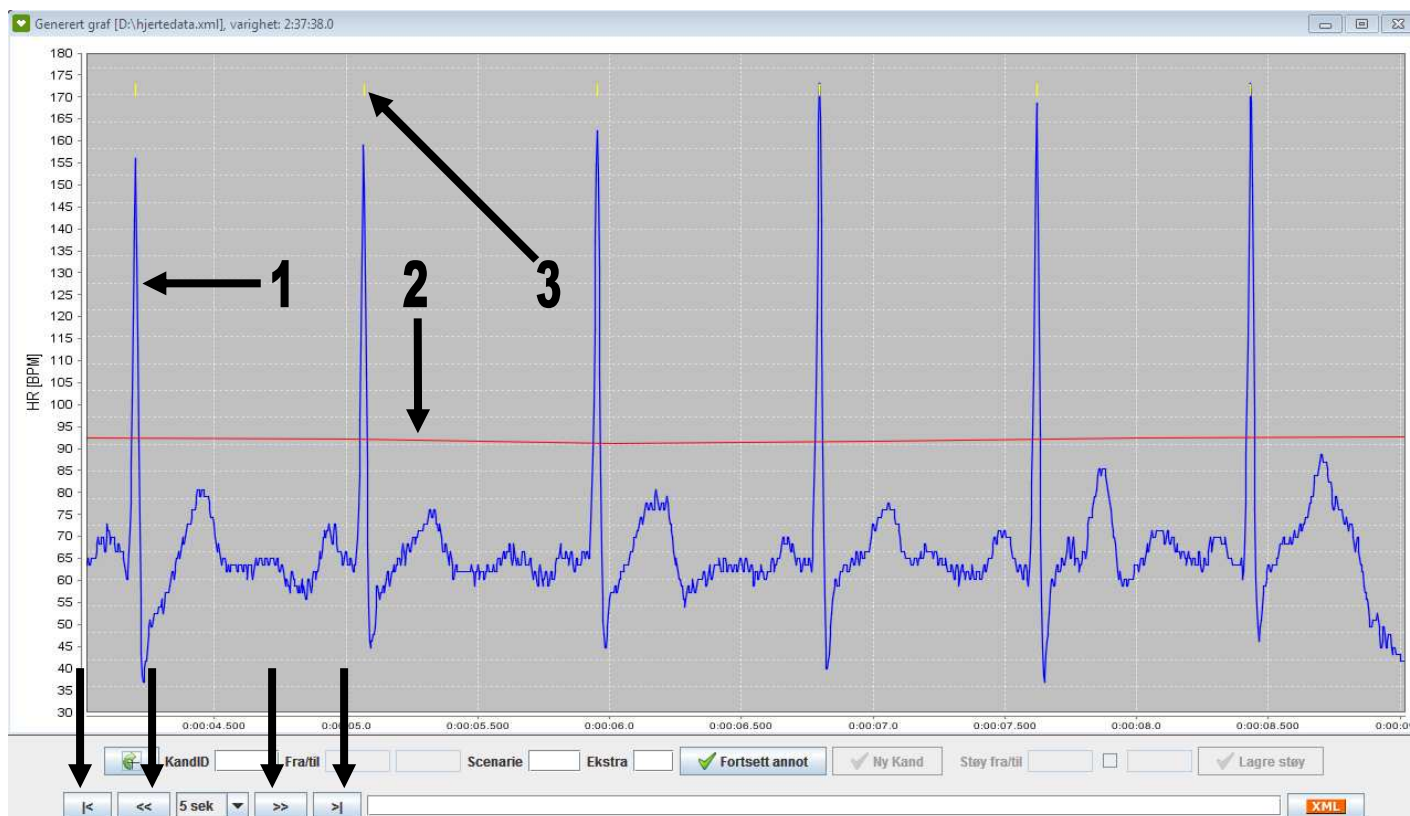
Figur 26: Filvelgeren startes. Det er mulig å velge hjertedatafiler av type ATS eller XML (som er ferdig lagret annotasjonsfil) for åpning.



Figur 27: Ved åpning av ATS vil programverktøyet automatisk konvertere til EDF og CSV i forhåndsdefinert temp-katalog.

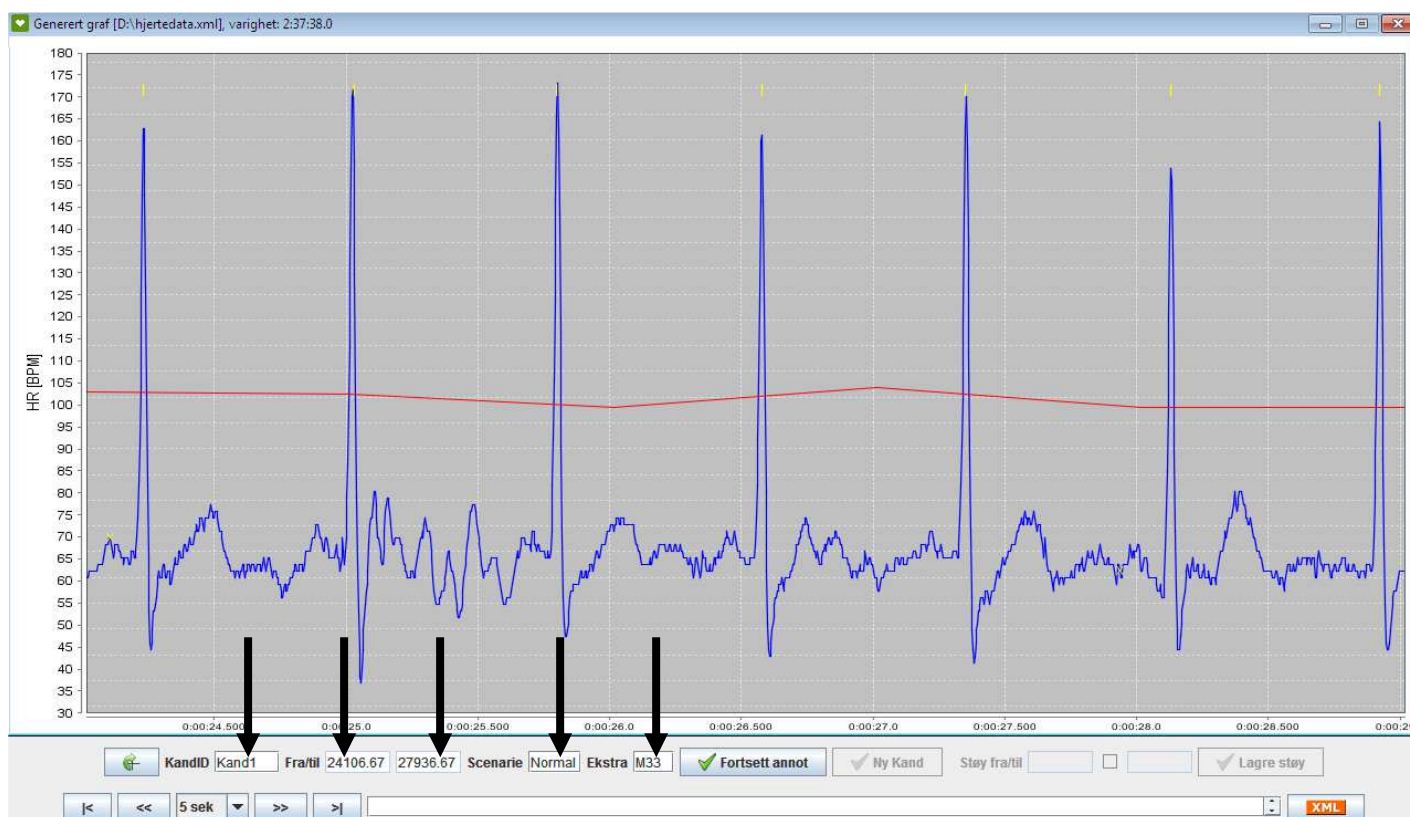


Figur 28: En prosessindikator vises mens EKG-, HR-, og RR-data leses inn. Etter ferdig innlesing aktiveres "Tegn graf"-valget som, når klikket, åpner et nytt vindu inneholdende plott.



Figur 29: Opptaket filnavn samt totale varighet vises i tittellinje, øverst på skjermen. På nedre del av skjermen vises annotasjonspanelet. Knapper for navigering i opptaket (start, tilbake, frem og slutt) finnes nederst til venstre i dette panelet. Y-akse viser HR-verdier, mens x-akse viser forløpt tid.

1. Blå graf markerer EKG (målt i millivolt).
2. Rød graf markerer HR målt i beats per minute [BPM] (her representert med en tilnærmet horisontal linje).
3. RR-intervaller er markert med korte gule streker.



Figur 30: En annotasjonssesjon startes med å angi data for den første kandidaten som skal annoteres på. Først fylles data inn for Kandid, Scenario og Ekstra informasjon.

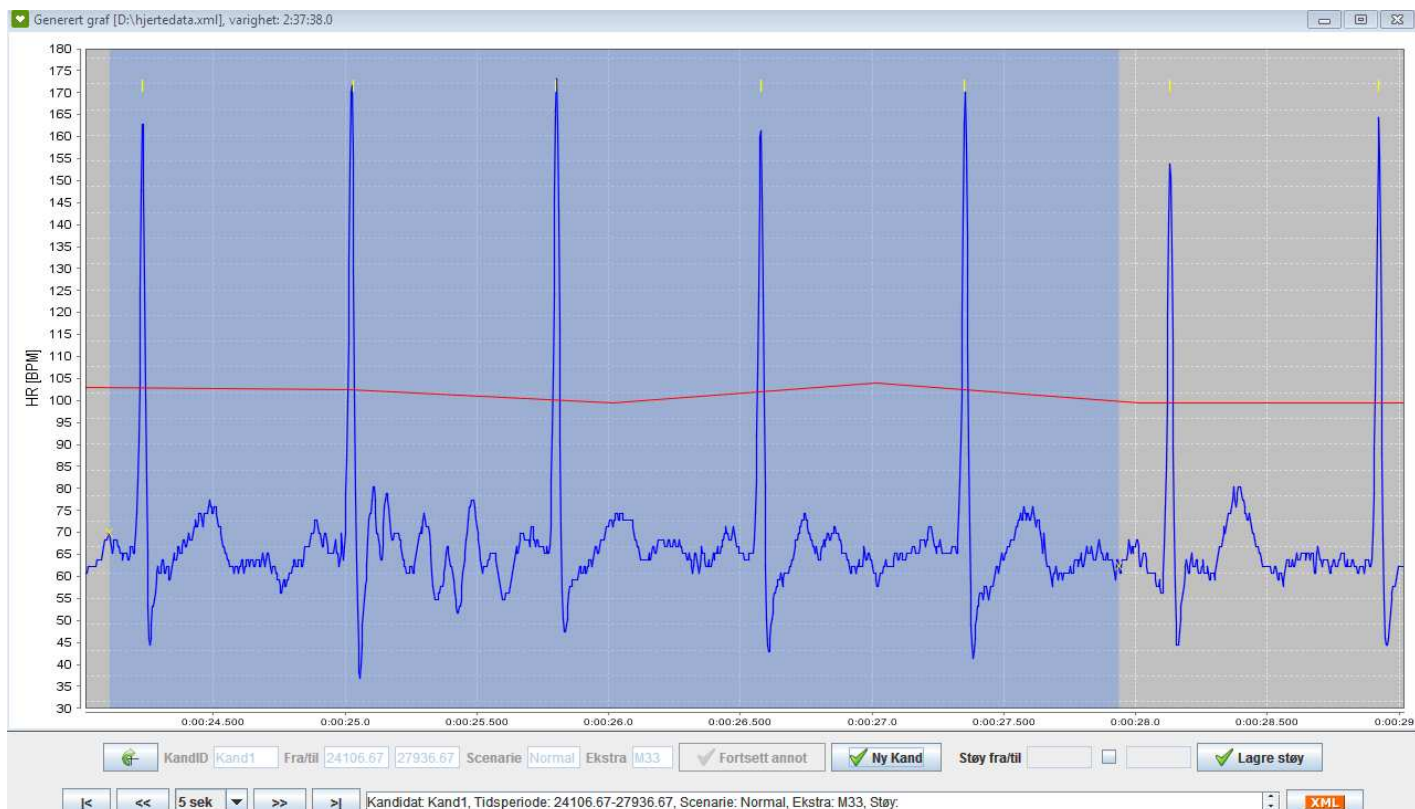
Typiske verdier for Scenario-feltet er "Normal", "Stresst", "Avslappet", etc. Ekstra-feltet kan brukes til å beskrive kandidaten ytterligere med kjønn og alder; eksempelvis "M24", som kan leses som mannlig kandidat på 24 år.



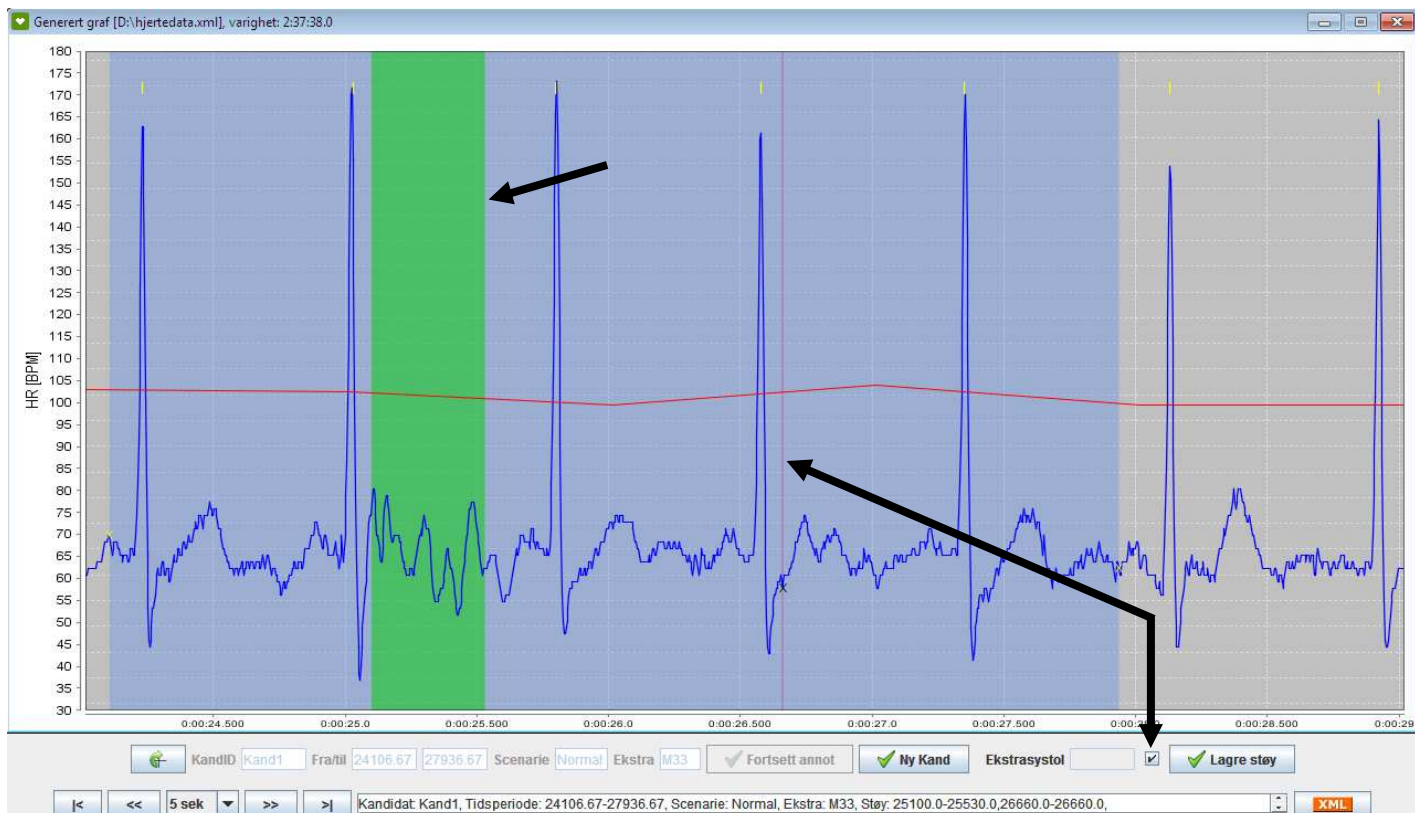
(1)

(2)

Figur 31: Start- og slutt punkt for kandidaten angis ved henholdsvis dobbelklikk (1) og enkelklikk (2) med venstre museknapp på EKG-grafen. "Fra/til" feltene markert i Figur 30 vil da automatisk fylles med tidsverdien for det respektive punktet som ble klikket på.



Figur 32: Når kandidaten er ferdig markert trykkes "Fortsett annot". Området på grafen som inneholder kandidaten vil markeres med blå bakgrunn.

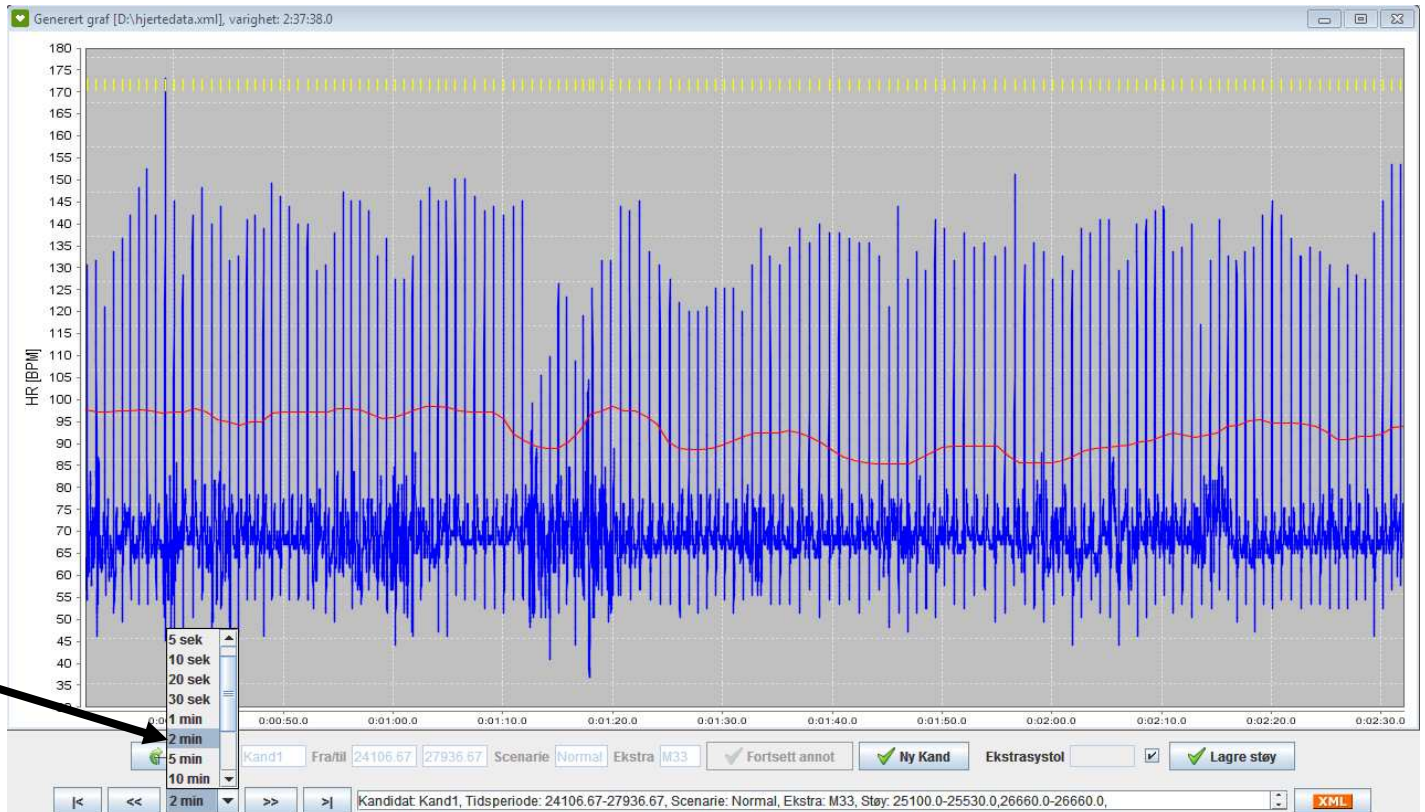


Figur 33: Når kandidaten er ferdig markert er det mulig å gjøre annotasjoner for støy. Støyseksjoner kan annoteres på to måter:

1) Start- og sluttverdi (angis på samme måte som med kandidat, med henholdsvis dobbel- og enkelklikk).

2) Ekstrasystol. En ekstrasystol er kun ett punkt. Dermed brukes kun startverdien; altså er det kun nødvendig med dobbelklikk. Et punkt angis som ekstrasystol ved å markere sjekkboksen plassert mellom ”Støy start/slutt” feltene.

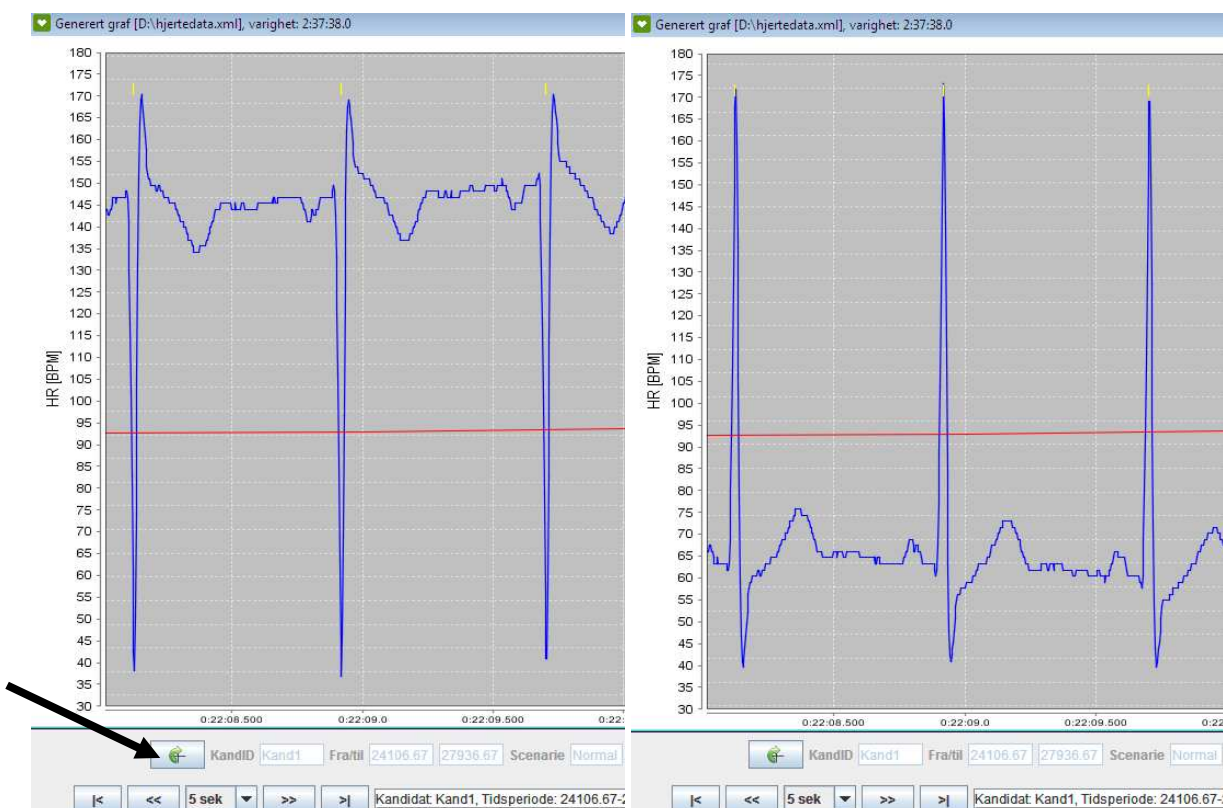
En ferdig støyannotering bekreftes med valget ”Lagre støy”. Dette gjøres helt til all støy er ferdig annotert for den respektive kandidaten. Støyområder markeres med grønn bakgrunn mens ekstrasystoler markeres med en vertikal, rød linje.



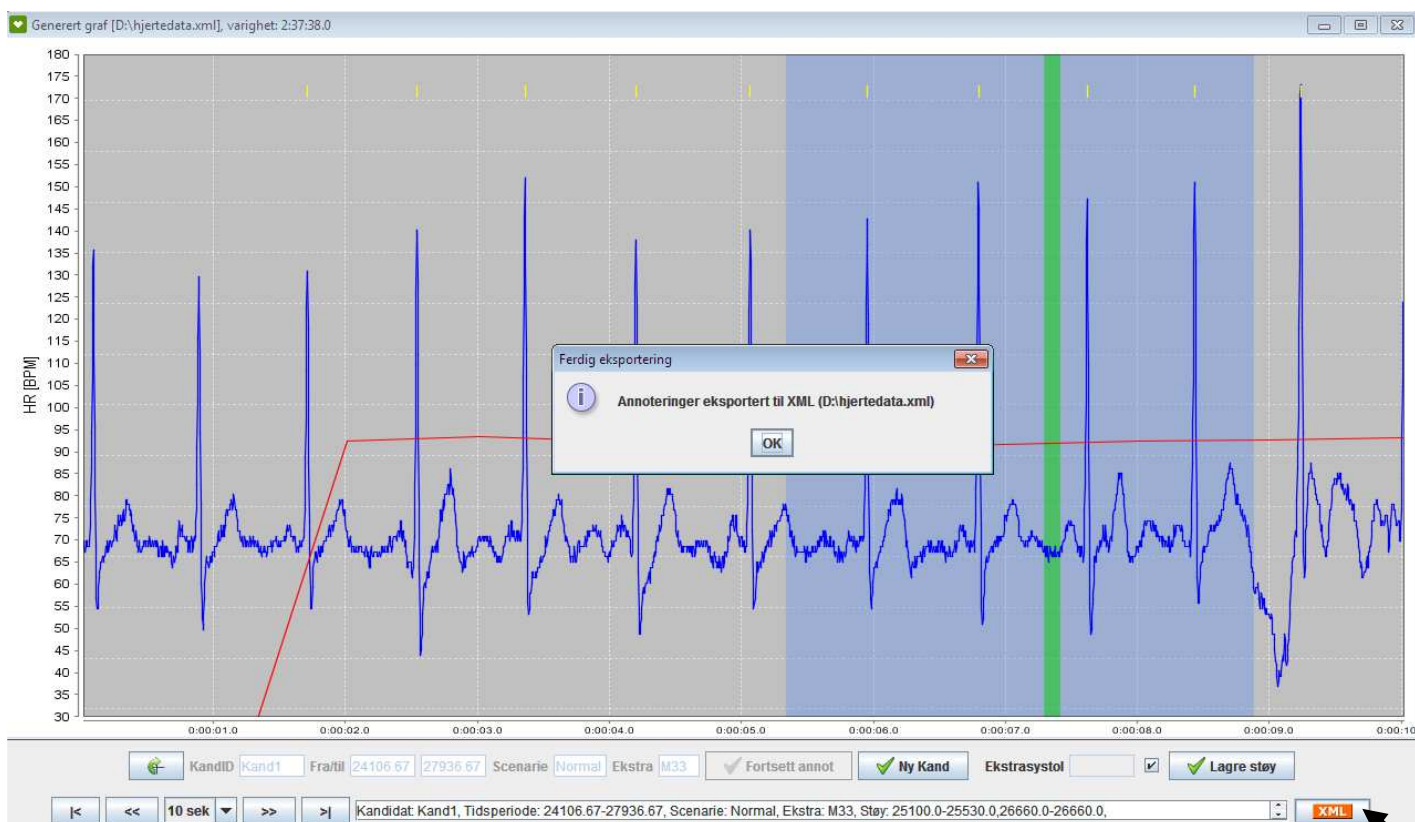
Figur 34: Tidsvinduet kan endres via kombinasjonsboksen. Tidsvinduet kan være 1 sek, 5 sek, 10 sek, 20 sek, 30 sek, 1 min, 2 min, 5 min, 10 min, 20 min, 30 min, 1 time eller hele opptaket.

Kandidat: Ola Nord, Tidsperiode: 223.0-2243.0, Scenario: Stress, Ekstra: M33, Støy: 980.0-1403.0, 1976.0-1976.0

Figur 35: Etter hvert som flere kandidatannotasjoner lagres er det mulig å bla gjennom kandidatlisten via pilknappene til høyre i tekstfeltet, nederst i annotasjonspanelet. Pilknapper blir først synlige når mer enn én kandidat er lagret. Lagrede tidsdata angis i millisekunder.



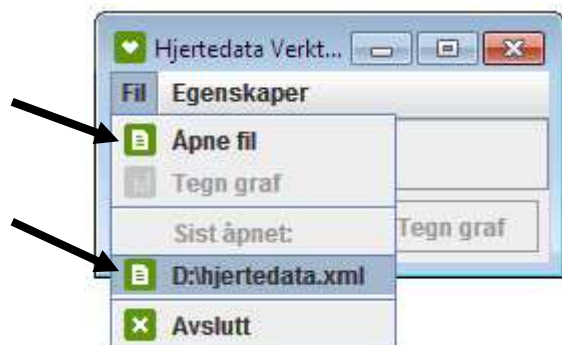
Figur 36: I løpet av en simulatorsesjon hender det at kandidaten får påsatt målelektrode i motsatt rekkefølge av forrige kandidat. Det er mulig å invertere grafen via invertér-valget øverst til venstre i annotasjonspanelet.



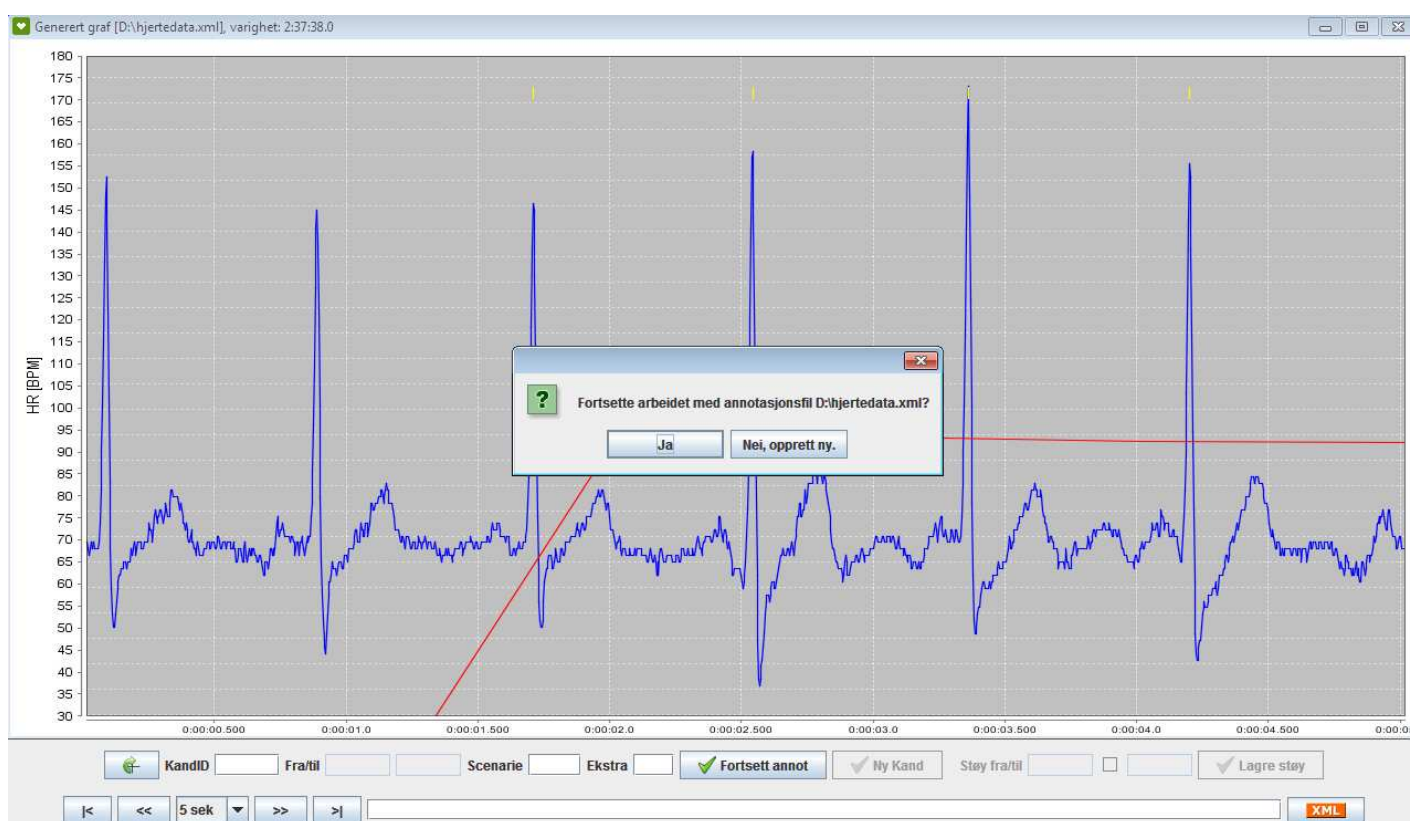
Figur 37: Når hjerteroptaket er ferdig annotert for alle kandidater og kandidatenes støyområder, kan annotasjonene eksporteres via XML-valget nederst til høyre i annotasjonspanelet. En dialogboks vil vises og opplyse hvor på harddisken filen ble lagret.

5.4.5 Åpne annotasjonsfil (XML) for visning eller videre annotering

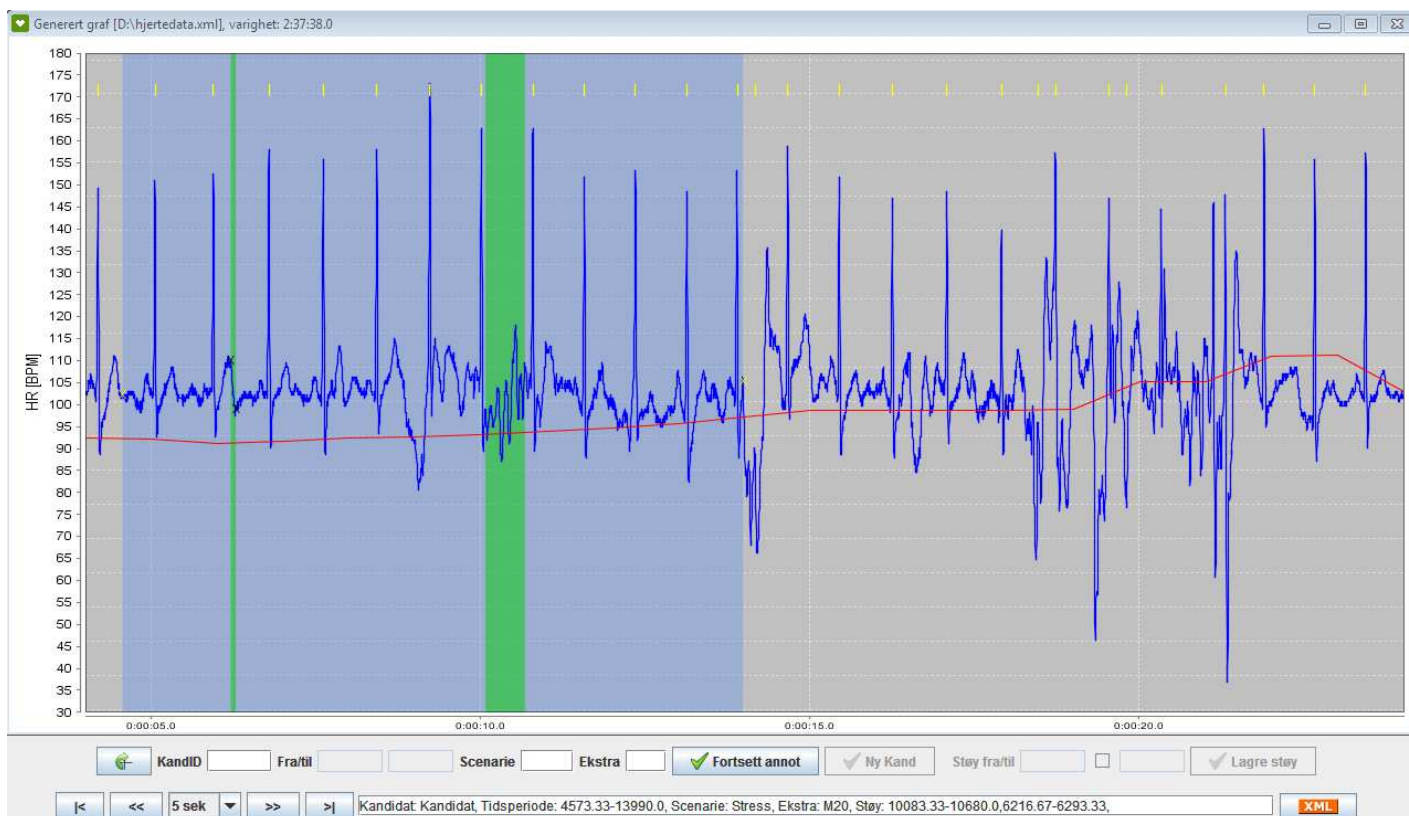
Annotasjonssesjoner kan gjenopptas ved åpning av lagrede annotasjonsfiler. Ved gjenåpning av en XML-fil vil verktøyet automatisk fortsette på den allerede påbegynte annotasjonssesjonen.



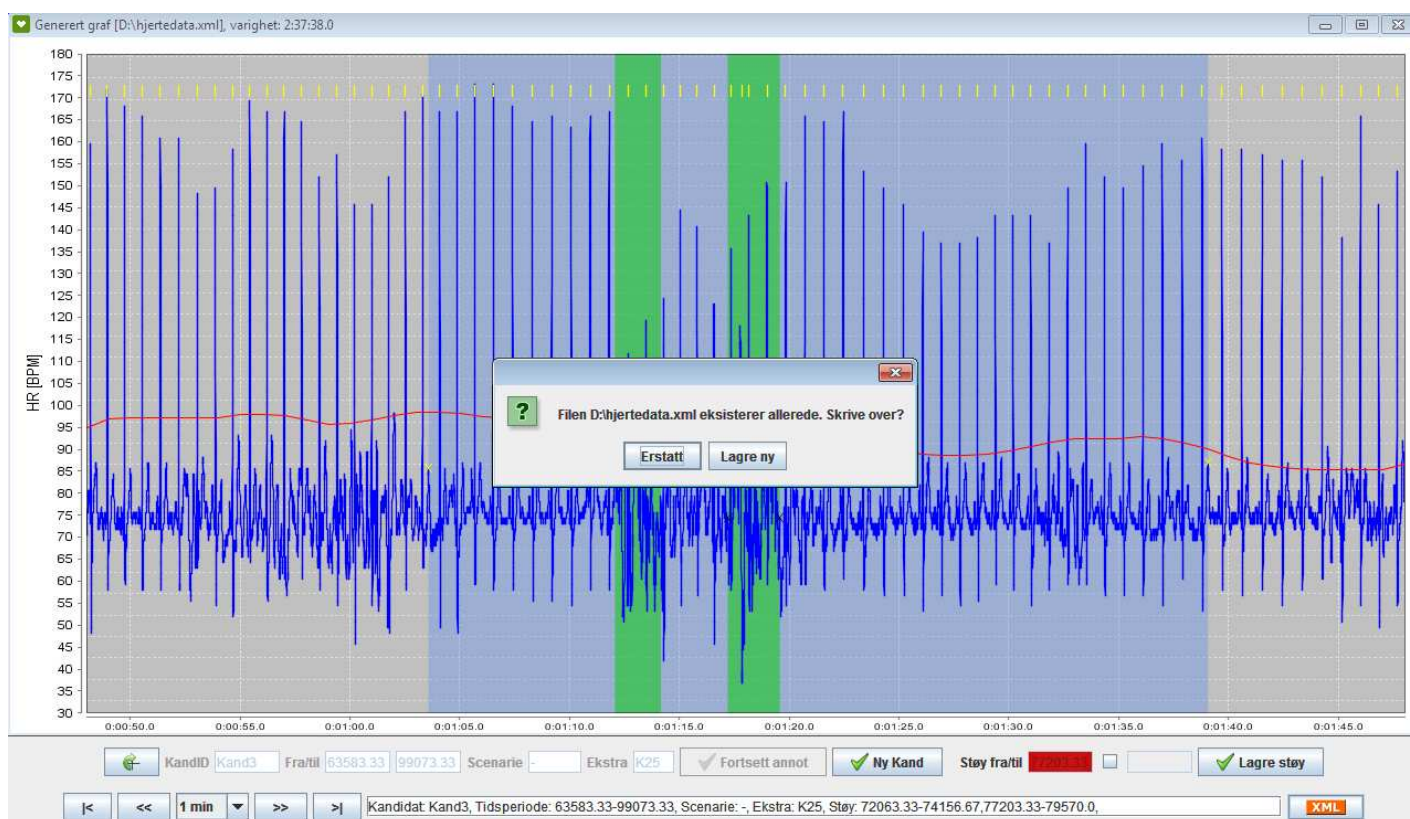
Figur 38: En annotasjonssesjon kan gjenopptas på et senere tidspunkt via "Sist åpnet" valget i Fil-menyen, eller ved å velge "Åpne fil" og lete fram en tidligere lagret annotasjonsfil (XML).



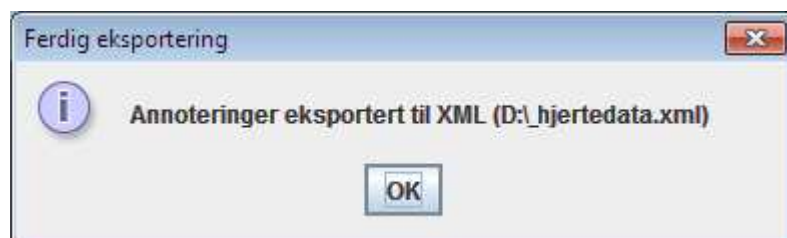
Figur 39: Ved lasting av en innhentet annotasjonsfil er det mulig å fortsette på lastet annotasjonssesjon eller opprette ny.



Figur 40: Ved fortsettelse på innhentet annotasjonssesjon vil lagrede annotasjoner tegnes opp og nye annoteringer kan gjøres.



Figur 41: Ved ny eksportering av annotasjonsfil sjekkes destinasjonen for eksisterende filnavn for å unngå tap av data. Detekterte navneduplikater resulterer i en dialogboks⁶ med spørsmål om overskriving eller lagring til ny fil.



Figur 42: Ved valg om å lagre ny fil vil opprinnelig filnavn suppleres med et ekstra understreketegn i filnavnet.

⁶ Dialogboksen vil ikke vises dersom valg om å fortsette på innehentet annotasjonssesjon er bekreftet på et tidligere tidspunkt (Figur 39).

5.5 Tilleggsinformasjon

5.5.1 Forsinkelse i deteksjon av R-takk og HR

Alive Heart Monitor (AHM) vil normalt sett starte deteksjon av R-takk et par sekunder ut i opptaket. Dette kan medføre at det i prototypen ikke alltid er mulig å se uthevede RR-intervaller uten først å panorere mot høyre (frem i tid). Dette gjelder også for HR, og er en begrensning i AHM-opptakeren.

5.5.2 Regler for filnavn

Filnavn uten punktum

Hjertedatafiler som skal åpnes i verktøyet kan ikke inneholde punktum i filnavnet (med unntak av ordinært punktum før filendelse). Dette er normal navnpolicy for de fleste operativsystemer, og gjelder også for utviklet prototype.

Omdøping og flytting av filer

Det anbefales ikke å omdøpe og/eller flytte hjertedatafiler (EDF) og RR-intervallfiler (CSV) generert av prototypen. Ved en eventuell omdøping og/eller flytting er det sjans for at lagrede annotasjonsfiler ikke lenger vil inneholde korrekt sti til disse filene. Denne regelen berører ikke filer av ATS-format eller genererte annotasjonsfiler (XML).

5.5.3 Panorering for tidsvindu på 1 og 5 sekund

JFC har problemer med å plote hjertedata med mindre panoreringsfaktorer enn *1.0* for tidsvindu på 1 og 5 sekunds størrelse. Prototypens panoreringsfaktor er derfor fastsatt til *1.0* (100 %) for tidsvindu av disse størrelsene.

6 Diskusjon

I dette kapittelet drøftes først mulige feilkilder i utviklet prototype. Videre beskrives en alternativ metode for å inkludere RR-intervall i EDF som er under arbeid. En oversikt over mulige forbedringer, fremtidig arbeid og videre håndtering av lagrede annotasjonsfiler drøftes i siste del av kapittelet.

I denne masteroppgaven er det utviklet et verktøy som er ment å være et nyttig og tidsbesparende hjelpemiddel ved analyse av EKG-data i SVS-prosjektet. I prosjektets innledende og inneværende fase (våren 2010) gjøres annoteringer og analyse på allerede lagrede hjertedata ("offline").

Den største utfordringen i arbeidet er å opprettholde en akseptabel prosesseringstid ved bruk av verktøyet. For å begrense prosesseringstid er flere tiltak gjort (som beskrevet i 4.4.5 Ytelse). Et av disse tiltakene er SFP, og innebærer en endring som medfører direkte påvirkning av plott. Plott påvirkes ved at ikke alle punkter vil tegnes når SFP er aktivert. En vurdering på hvorvidt denne påvirkningen er avgjørende for resulterende plott er tatt av undertegnede og er ansett som akseptabel. Dersom, i framtiden, denne funksjonen ut i fra et fysiologisk ståsted viser seg å påvirke resultat i negativ forstand, er det gitt mulighet til å justere eller deaktivere denne funksjonen.

6.1 Mulige feilkilder

Utviklet verktøy er bygget over to eksisterende Java-bibliotek [43, 44]. En vurdering av samtlige, navngitte verktøy og bibliotek er gjort gjennom utviklingsløpet og alle er funnet både stabile og robuste nok for det aktuelle formålet. Over tid, eller i andre testmiljø, kan disse dog være potensielle feilkilder.

ATS-formatet og AtsConvert. Det er lite tredjepartsarbeid som er gjort rundt Alive Technologies' eget dataformat. Den europeiske standarden for hjertedata EDF er langt mer utbredt, og er brukt som utgangspunkt for hjertedataverktøyet nettopp på grunn av dette. Etersom ATS-filer konverteres til EDF via Alive Technologies' AtsConvert-applikasjon, gis det ingen garanti for at konvertert data blir korrekt representert. Dette kan gi hypotetisk sett gi utslag i form av forskjøvet tidslinje med tanke på plottingen av EKG, HR og RR-intervallene.

AtsConvert har i tidligere versjoner (før versjon 1.5.0.8) inneholdt en feil som medførte at QRS-deteksjonene var unøyaktige. Dette kunne medføre at RR-intervallene som ble detektert ikke samsvarte med EKG-grafens R-takk.

JFreeChart. Dette Java-biblioteket er basert på åpen kildekode. Åpen kildekode kan medføre variable resultater. Positivt med JFC er at det er et svært praktisk hjelpemiddel med tanke på plott. Det er fritt tilgjengelig og gir mange muligheter og varianter av type plott. Negativt med JFC er at ettersom biblioteket er basert på åpen kildekode, garanteres ikke feilfrie grafer. Åpen kildekode kan inneholde kode fra forskjellige parter, og det stilles ikke nødvendigvis like strenge krav til åpen programvare som kommersielle alternativer. På samme måte som med ATS-formatet kan dette rent hypotetisk sett gi utslag i form av forskjøvet tidslinje med tanke på plottingen av EKG, HR og RR-intervaller.

jEDF. På lik linje med JFC er jEDF-biblioteket også basert på åpen kildekode. I utviklet verktøy brukes to klasser fra jEDF som omhandler innlesing av EDF-filer, og det gis derfor ingen garanti for korrekt innlesing av hjertedata.

Dersom overnevnte feilkilder blir et faktum, er det lagt inn en mulighet for å korrigere RR-intervallenes x-verdier. Dette kan gjøres med valget *RR-intervall offset*, som kan justeres i innstillingspanelet (vist i 5.4.2, Figur 19).

6.2 Oppdatert AtsConvert under utvikling

Jeg har vært i dialog med Kim Barnett som jobber med programvareutvikling i Alive Technologies. Han meddeler at de per mars måned 2010 ser på muligheten for å i framtiden også kunne inkludere RR-intervaller i EDF-formatet. Som en mulig framgangsmåte for å implementere dette peker Barnett på muligheten til å tilpasse EDF-formatets event-signal til å i stedet inneholde ferdig detekterte RR-intervall.

For verktøyet utviklet i denne masteroppgaven vil dette gjøre innlesing av ATS-filer mindre tidkrevende, ettersom dagens bruk av CSV for importering av RR-verdier vil bli overflødig.

Både utviklet prototype og et eventuelt oppdatert AtsConvert-verktøy kan være aktuelt ved ønske om å knytte annotasjonsformatet direkte sammen med hjertedata i fremtidig analyse.

6.3 Mulige forbedringer

- Redusere nødvendige hjertedatafiler og konverteringer. Her er det to muligheter;
 1. Implementere direkte ATS-parsing for å slippe mellomledet som AtsConvert-verktøyet utgjør.
 2. Dersom Alive Technologies slipper en oppdatert versjon av AtsConvert (omtalt i 6.2) som tillater lagring av RR-intervall i EDF-formatets event-signal, vil en mulig forbedring være å tilpasse prototypen til å hente alle nødvendige data fra denne ene hjertedatafilen.
- Utviklet prototype inneholder en standardverdi for RR-offset, som i ulike testede miljø og scenario viser seg å være akseptabel. En mulig forbedring vil være å implementere en funksjon som sørger for en helautomatisk offset-korreksjon.
- Ytterligere utbedring av prosesseringstid ved plotting.
- Det er i denne masteroppgaven ikke ytret noe krav til funksjonalitet som muliggjør redigering og/eller sletting av lagrede annotasjoner. Dette er funksjonalitet som i lengden kan vise seg å være nyttig, og foreslås derfor som en mulig forbedring i en videreutvikling av programmet.

6.4 Fremtidig arbeid

Videre arbeid i *Stressmåling ved simulatortrening* – prosjektet vil fortsette i nåværende ("offline") fase med hjertedataanalyse av annotasjonsfiler i kombinasjon med hjertedatafilene. Neste fase vil påbegynnes over tid, da i form av sanntidsanalyse av kandidater i simulatoren.

6.4.1 Analyse av lagrede hjertedata

Denne masteroppgaven omhandler ikke videre analyse av eksporterte annotasjonsfiler. I dette delkapitlet beskrives likevel en kort, prinsipiell innføring i hvordan de XML-baserte annoteringene *kan* importeres i MATLAB.

Etter at ønskede annoteringer er gjort på lagrede hjertedata er det hensiktsmessig å kunne fortsette analysen i MATLAB.

Lagrede annotasjonsfiler er representert ved hjelp av XML. En annotering består av det aktuelle feltet og en x-koordinat (plasseringen på tidslinjen angitt i millisekund).

Stien til den aktuelle EDF-filen som annoteringen ble gjort i forhold til lagres i rotelementets attributfelt. Denne stien brukes av hjertedataverktøyet for å kunne gjenoppta arbeidet med påbegynt annotasjonsfil. Stien er også tenkt brukt av fremtidige MATLAB-rutiner for innlesing av annoteringsfilen, ettersom det er ønskelig å operere direkte på EDF/ATS-filen som inneholder det opprinnelige opptaket.

Det eksisterer allerede komplette MATLAB-rutiner for parsing av hjertedata lagret i ATS-format [20]. Denne parseren muliggjør plott av hjertedata til en viss grad. Et annet alternativ er åpen kildekode-prosjektet *BioSig* [50]. BioSig leverer en MATLAB-modul for bruk sammen med EDF-filer. I tillegg finnes også en tilsvarende modul for *Octave* [51], som er et gratis alternativ til MATLAB.

6.4.1.1 Foreslått løsning

For innlesing av annotasjonsfiler i XML-format finnes ulike typer XML-moduler tilgjengelige for MATLAB. Her tas utgangspunktet i den fritt tilgjengelige modulen *XMLTree Toolbox*. Modulen er beregnet for både skriving og lesing av data i XML-filer.

En eventuell fremtidig løsning basert på XMLTree Toolbox kombinert med ATS-parser referert til i [20] bør muliggjøre å koble innhentede annotasjoner direkte sammen med hjertedata for aktuelle tidsseksjoner. Teoretisk sett bør det være mulig å eksempelvis generere plottutklipp av et EKG-opptak i form av en støyseksjon på X antall millisekunder. Matematisk analyse av støyseksjonen kan da resultere i tilgang på ny, tidligere utilgjengelig informasjon.

Prinsipiell installasjon og konfigurering:

1. XMLTreeToolbox er tilgjengelig fra [19].
2. Følg instruksjoner gitt i "Install".
3. *Xmldemo1.m*, *Xmldemo2.m* og *Xmldemo3.m* kan kjøres (i MATLAB) for å gjøre seg kjent med virkemåten til modulen.
4. Videre kan *Xmldemo1* (eventuelt -2 eller -3) redigeres og tilpasses for å lese XML-data fra eksporterte annotasjonsfiler (eksempelvis *hjertedata.xml*).
5. ATS-parser kan hentes på [20] og trenger lite eller ingen tilpasning for å lese ATS-filer.
6. Et naturlig siste steg vil være å bruke tilpasset *Xmldemo* til å kalle opp ATS-parser med relevante tidsparametere for den aktuelle annotasjonsfilen.

6.4.2 Sanntidsanalyse av hjertedata

I neste fase av SVS-prosjektet er det ønskelig å kunne konstruere en løsning som tillater automatisk analyse av hjertedata direkte (i sanntid) mens kandidaten er i simulatoren.

I denne fasen av prosjektet vil nye problemstillinger ta form. Markering av når en kandidat kobles fra/til EKG- monitor er en mindre implementasjonsmessig utfordring. Algoritmer for deteksjon av støy i opptaket vil være en nødvendighet. Simulerings situasjoner preget av ekstrasystoli vil etter all sannsynlighet skape utfordring i en sanntidsanalyse, og vil kreve en algoritme som klarer å skille et normalt hjerteslag fra et ekstra hjerteslag.

7 Referanse- og kildehenvisning

1. *'Flight simulators' getting into the operation theatre. A new dimension in clinical engineering and patient safety.* Arne Rettedal. University of Stavanger, Norway.
2. *PatSim – simulator for practising anaesthesia and intensive care.* Arne Rettedal, Ståle Freyer, Ragna Kleppa, Petter Larsen. University of Stavanger, Norway. 1995-1996.
3. *Stress og HRV.* HRV, s. 11. Hovedoppgave av Julian Zec, 2006
4. *Biomedical signal processing on cardiac and neurological applications.* Sornmo L, Laguna P, Elsevier Academic Press, (s.567), 2005
5. *Java Software Solutions: Foundations of Program Design. 4th Edition.* John Lewis, William Loftus. Addison-Wesley, USA. 2005.
6. *Introduction to Scientific and Technical: JFileChooser Dialog.* JavaTech. Tilgjengelig fra:
[<http://www.particle.kth.se/~lindsey/JavaCourse/Book/Part1/Java/Chapter09/chooser.html>] (Hentet 12.01.10)
7. *Computing with Java.* Clark S. Lindsey, Johnny S. Tolliver, Thomas Lindblad. Cambridge University Press, October 2005.
8. *EKG - verdt å vite.* Carl J. Brandt. Tilgjengelig fra:
[<http://www.nettdoktor.no/helseraad/undersokelser/ekg.php>] (Hentet 28.04.10)
9. *Full Specification of EDF.* Bob Kemp, Alpo Värri, Agostinho C. Rosa, Kim D. Nielsen & John Gade. Tilgjengelig fra: [<http://www.edfplus.info/specs/edf.html>] (Hentet 20.02.10)
10. *DOM og SAX.* W3C, Sax Project. Tilgjengelig fra:
[<http://www.it.hiof.no/~borres/ml/dom/p-dom.html>] (Hentet 04.02.10)
11. *ECG history.* Open ECG Project. Tilgjengelig fra: [<http://www.open-ecg-project.org/tiki-index.php?page=ECG+history>] (Hentet 23.03.10)
12. *Ekstrasystoler.* Norsk Elektronisk Legehåndbok (via NHI.no). Tilgjengelig fra:
[<http://nhi.no/sykdommer/hjerte-kar/hjertearytmer/ekstraslag-systoler-12834.html>] (Hentet 28.04.10)
13. *Hurtig forkammerrytme.* Norsk Elektronisk Legehåndbok via (via NHI.no). Tilgjengelig fra: [<http://nhi.no/sykdommer/hjerte-kar/hjertearytmer/hurtig-forkammerrytme-14898.html>] (Hentet 28.04.10)
14. *Hjerterytmer.* Hentet fra kompendium av Anne Siri Rokvam. Tilgjengelig fra:
[<http://www.hjerteovervakningen.com/hjerterytmer.htm>] (Hentet 28.04.10)
15. *Ekstrasystoli.* Det Store Norske Leksikon. Tilgjengelig fra:
[<http://www.snl.no/ekstrasystoli>] (Hentet 03.05.10)
16. *Ekstrasystoli.* JournalWiki, Universitetet i Oslo. Tilgjengelig fra: [<http://www.journalwiki.no/Ekstrasystoli>] (Hentet 03.05.10)
17. *SimpleDateFormat (tutorial).* Java2s.com. Tilgjengelig fra:
[http://www.java2s.com/Tutorial/Java/0040__Data-Type/SimpleDateFormat.htm] (Hentet 22.04.10)
18. *DOM Parser.* Mkyong.com, 2008. Tilgjengelig fra:
[<http://www.mkyong.com/java/how-to-read-xml-file-in-java-dom-parser/>] (Hentet 18.03.10)
19. *XMLTree: an XML toolbox for MATLAB.* Tilgjengelig fra: [<http://www.artefact.tk/software/matlab/xml/>] (Hentet 15.03.10)
20. *MATLAB ATS-parser.* Jussi Virkkala, 2007 Tilgjengelig fra:
[<http://www.neuroupdate.com/alivetec/ats-07.08.15.m>] (Hentet 22.01.10)
21. *Characteristics of the Normal ECG.* Frank G. Yanowitz, University of Utah School of Medicine. Tilgjengelig fra:

- [http://library.med.utah.edu/kw/ecg/ecg_outline/Lesson3/index.html] (Hentet 02.04.10)
22. *Programmeringsspråk, filbehandling*. Ulf Uttersrud, 2005. Tilgjengelig fra: [<http://www.iu.hio.no/~ulfu/Programmeringssprak/filbehandling1.html>] (Hentet 02.02.10)
23. *To Those Who Use JFreeChart for Dynamic Plotting/Large Sets*. (diskusjonstråd; side 1, punkt 1). Tilgjengelig fra: [<http://www.jfree.org/phpBB2/viewtopic.php?f=3&t=18592>] (Hentet 09.03.10)
24. *Export to XML*. Tilgjengelig fra: [http://www.experts-exchange.com/Programming/Languages/Java/J2SE/JAX_SAXP/Q_24274642.html] (Hentet 15.03.10)
25. *Clipping*. Tilgjengelig fra: [http://en.wikipedia.org/wiki/Clipping_%28audio%29] (Hentet 28.05.10).
26. *Bluetooth Heart Monitor Data Format*. Alive Technologies, August 2006.
27. *Alive Wireless Heart Monitor Product Specification*. Alive Technologies.
28. *ECG History*. Open ECG Project. Tilgjengelig fra: [<http://www.open-ecg-project.org/tiki-index.php?page=ECG+history>] (Hentet 11.03.10)
29. *Explaining Interleaving*. TechGeeks-Online. Tilgjengelig fra: [<http://techgeeks-online.com/2010/explaining-interleaving/>] (Hentet 01.06.10)
30. *Additional specifications in EDF+*. Bob Kemp, Jesus Olivan, 2003. Tilgjengelig fra: [<http://www.edfplus.info/specs/edfplus.html#additionalspecs>] (Hentet 20.02.10)

7.1 Figurer, bilder, ikoner

31. *Pulsklokke (figur)* Tilgjengelig fra: [http://www.mrgadget.com.au/catalog/images/Polar_S625X_Running_Heart_Rate_Monitor_Watch_1.jpg]
32. *Tradisjonell hjertemonitor (figur)* Tilgjengelig fra: [http://www.timexheartratemonitor.com/images/heart_rate_monitor/heart_rate_monitor_250x251.jpg]
33. *Alive Heart Monitor (figur)*. Alive Technologies. Tilgjengelig fra: [<http://www.alivetec.com/images/heartmonitorsm.jpg>]
34. *Det normale EKG" (figur)*. National Instruments Corporation. Tilgjengelig fra: [<http://zone.ni.com/devzone/cda/tut/p/id/6349>] (figur utvidet med RR- intervall av Ørjan Horpestad).
35. *Aktiveringsnivå (figur)*. Hentet fra 'Flight simulators' getting into the operation theatre. A new dimension in clinical engineering and patient safety. Arne Rettedal. University of Stavanger, Norway. (Figur 5, s. 61))
36. I prototype: *Twotiny Icons Free icon-set*. Paul Jarvis. Tilgjengelig fra: [<http://code.google.com/p/twotiny/>]
37. I prototype: *Cw, Rotate icon*. Marco Martin. Tilgjengelig fra: [<http://www.iconfinder.com/icondetails/8037/64/>]
38. I prototype: *OK icon*. Deleket. Tilgjengelig fra: [<http://www.veryicon.com/icons/system/sleek-xp-basic/ok-1.html>]
39. I prototype: *XML icon*. Tilgjengelig fra: [http://commons.wikimedia.org/wiki/File:Small_Orange_XML_Button_for_RSS_Page.gif]
40. *Logitech v470*. Tilgjengelig fra: [<http://workaholic.files.wordpress.com/2009/07/logitech-v470-white-mighty-mouse.jpg>]

41. *Green Arrows Set Left*. OCAL. Tilgjengelig fra: [<http://www.clker.com/clipart-green-arrows-set-left.html>]
42. *Female User Icon*. OCAL. Tilgjengelig fra: [<http://www.clker.com/clipart-12291.html>]

7.2 Java biblioteker

43. *JFreeChart (Java library)*. Object Refinery Limited, 2005-2007. Tilgjengelig fra: [<http://www.jfree.org/jfreechart/>]
44. *jEDF (Java EDF library)*. Nizar Kerkeni. Tilgjengelig fra: [<http://freshmeat.net/projects/jedf>]

7.3 Programvare brukt til utvikling og implementering

45. *Eclipse (Galileo) SDK v. 3.5.0* (2009)
46. *Microsoft Visio Office 2007*
47. *Polyman v. 1.3.4.631* av Roessen & Kemp. Tilgjengelig fra: [<http://www.edfplus.info/downloads/software/polyman.zip>]
48. *AtsConvert v. 1.5.0.8* (2010)
49. *Launch4j v3.0.1 EXE-wrapper*. Tilgjengelig fra: [<http://launch4j.sourceforge.net>]
50. *BioSig*. Tilgjengelig fra: [<http://biosig.sourceforge.net>]
51. *Octave*. Tilgjengelig fra: [<http://www.gnu.org/software/octave/>]

8 Vedlegg

Vedlegg 1 (1 side): Nedskalert utgave av plakat brukt i plakatpresentasjon 19.05.2010.

Vedlegg 2 (2 sider): *Alive Heart and Activity Monitor Specifications*.

Praktisk informasjon

Kildekode til utviklet prototype er vedlagt på CD i komprimert Zip-fil. Kildekoden er kommentert og det er generert dokumentasjon i form av JavaDoc. JavaDoc-dokumentasjonen finnes i doc-katalogen i rotkatalogen til kildekode (doc\index.html). GNU GPL (General Public License) er inkludert på CD (.txt-fil) av respekt til JFC og JEDF.

Prototype kan kjøres via vedlagt EXE-fil. Detaljert informasjon om systemkrav og praktisk bruk beskrives i brukerveiledningen. Det anbefales å kopiere *Hjertedata Verktøy*-katalog fra CD til harddisk før kjøring for å muliggjøre lagring av innstillinger.

Grafisk grensesnitt for håndtering av lagrede hjertedata (EKG)

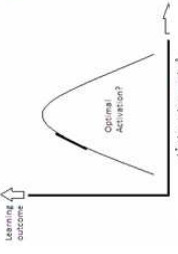
Masteroppgave vår 2010 -- Ørjan Horpestad

Sammandrag

- UI5 prosjektet "Stressmåling ved simulering" får studenter trene seg i intensivbehandling på en pasientsimulator (Patsim).
- Det gjøres EKG opptak av studentene under simuleringen.
- Via prototypen utviklet i denne masteroppgaven er det mulig å annotere på lagrede opptak.
- Annoterte opptak lagres og kan leses inn i MATLAB for videre analyse.

Bakgrunn

- Det antydes av læringspsykologer og pedagoger at læring øker hos en person ved etnoe økt stressnivå.
- Det viser seg at denne læringen avtar straks stressnivået overskrider en viss grense – individuelt for hver person.
- Å ha et visst stressnivå i situasjoner der det kreves litt ekstra vil en person dra nytte av, ved at man er mer åpen for læring. Blir stressnivået for høyt kan personen oppleve konsekvenser i form av f. eks. stressanfäll. Slike tilstander er ikke ideelle for læring.
- Patsim er en simulator utviklet med hensikt å la studenter trene seg i intensivbehandling. Patsim er bygget rundt en torso produsert av Laerdal Medical. Ved å tilby studenter en simulator å øve seg på slik type behandling unngår man å usette pasienter for unødig risiko.
- Elektrokardiogram registrering av studenten gjøres med en portabel hjerteoppfatter knyttet sammen med en datamaskin via Bluetooth.



datamaskin via Bluetooth.

- UI5 prosjektet *Stressmåling ved simulering* har som mål å se på den enkelte studentens stressnivå og kunne oppdage når det vil være hensiktsmessig å begrense stressaktiviteten under simuleringer.

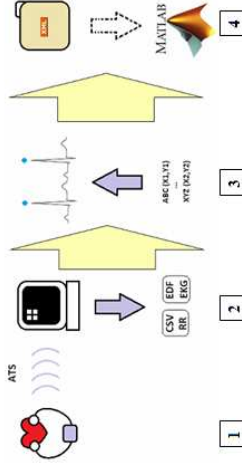
Oppgave- og problembeskrivelse

- Hensikt med denne masteroppgaven er å lage et grensesnitt for å enklere kjernejobbe med disse lagrede hjertedataene.
- Et elektrokardiogram opptak gjort i løpet av en simuleringssesjon lagres som en datafil.
- Et opptak kan bestå av flere studenter.
- Hver student kan ha forekomster av ekstra hjerteslag (ekstrasystoler)
- Hver student kan ha flere perioder med støy. Støy forekommer av:



Alive Heart Monitor

Virkemåte



1. Registrert hjertedata overføres fra Bluetooth-oppfater og lagres i Alive Technologies' dataformat ATS, som kan leses av utviklet prototype.
2. Hjertedata inneholder elektrokardiogram (EKG), hjerterate (HR) og RR-intervaller, og trekkes ut av ATS-formatet i henholdsvis EDF og CSV.
3. EDF (inneholdende EKG og HR) og CSV (inneholdende RR-intervaller) leses inn og plottes av prototypen.
 - RR-intervaller plottes som en egen dataserie
 - Plot kan panoreres frem og tilbake i tid
 - Plot kan skaleres etter ulike tidsvinduer (sekund, minutt, timer).
4. Ved ferdigstilt annotasjonssesjon lagres annoteringer med input- og tidsdata som en MATLAB-kompatibel XML-fil.

Skjermbilder

Opptak med to kandidater
Hver kandidat har fått annotert to støysesjoner hver.

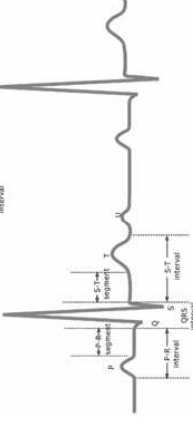
Zoomet inn på del av opptak markert og annotert som støy

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<seg_annotasjoner_ukleffile-Ds.Wledf ?>
<kandidatID>Orjan</kandidatID>
<TidromStart>56.0</TidromStart>
<TidromSlutt>14700.0</TidromSlutt>
<Tidrom>
<TidromOpplysnings>Net</TidromOpplysnings>
</Tidrom>
<MarkertStoy>
<Start>6066.0</Start>
<Slutt>6033.0</Slutt>
</MarkertStoy>
<Tidrom>
<TidromOpplysnings>M25</TidromOpplysnings>
</Tidrom>
<MarkertStoy>
<Start>16220.0</Start>
<Slutt>16973.0</Slutt>
</MarkertStoy>
<Tidrom>
<TidromOpplysnings>M25</TidromOpplysnings>
</Tidrom>
<MarkertStoy>
<Start>19876.0</Start>
<Slutt>20500.0</Slutt>
</MarkertStoy>
<KandidatID>Student</KandidatID>
<TidromStart>15093.0</TidromStart>
<TidromSlutt>15093.0</TidromSlutt>
<Tidrom>
<TidromOpplysnings>M25</TidromOpplysnings>
</Tidrom>
<MarkertStoy>
<Start>10040.0</Start>
<Slutt>10040.0</Slutt>
</MarkertStoy>
<KandidatID>Student</KandidatID>
<TidromStart>15093.0</TidromStart>
<TidromSlutt>15093.0</TidromSlutt>
<Tidrom>
<TidromOpplysnings>M25</TidromOpplysnings>
</Tidrom>
<MarkertStoy>
<Start>16220.0</Start>
<Slutt>16973.0</Slutt>
</MarkertStoy>
<Tidrom>
<TidromOpplysnings>M25</TidromOpplysnings>
</Tidrom>
<MarkertStoy>
<Start>19876.0</Start>
<Slutt>20500.0</Slutt>
</MarkertStoy>
</KandidatID>
</seg_annotasjoner_ukleffile-Ds.Wledf ?>
```

Eksempel på annotasjonsfil

- at en (eller flere) måleelektroder har blitt flyttet på og/eller falt av
- dårlig kontakt i ledninger eller måleelektroder
- bevegelsesartefakter
- ekstrem indusert støy (mobilttelefon, netstøy fra joutfeil - men lite på berørte oppfater).

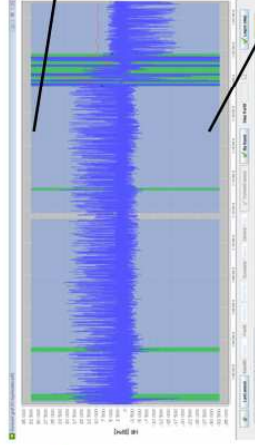
- I forkant av en hjertedatanalyse er det nødvendig å ha markert disse annoteringene.
- En fremtidig analyse vil kreve at annoteringene kan leses av MATLAB.



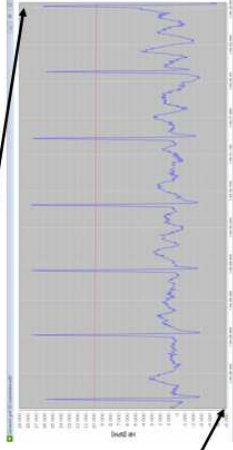
Det normale elektrokardiogram

Annotationsstruktur

- Foreldemoden <seg_annotasjoner> har <KandidatID> som barnmode.
- <KandidatID> har flere barnmoder (kandidatens ID, tidsrom, scenario, og et ekstratref for tilleggssinformasjon), i tillegg til <MarkertStoy>.
- <MarkertStoy> er foreldemoden til barnmoden <Tidrom>.
- Hver <Tidrom> node er forelder til barnmodene <Start> og <Slutt> som sammen representerer et markert støyområde.
- Ved lagring av flere støyområder vil den aktuelle <KandidatID> inneholde en barnmode <MarkertStoy> med flere <Tidrom>. Hvert av disse <Tidrom> nodene vil inneholde ett støy-tidsrom-start-slutt-par for hvert støyområde.



Opptak med to kandidater
Hver kandidat har fått annotert to støysesjoner hver.



Zoomet inn på del av opptak som ikke karakteriseres som støy

Resultat

Utviklet verktøy har følgende funksjonalitet

- Store for Alive Technologies' hjertedataformat ATS
- Automatisk markering av P- bølger i et hjerteslag.
- Plott og visning av elektrokardiogram (EKG)
- Plott og visning av hjerterate (HR)
- Mulighet for annotering av kandidater, støy og ekstrasystoler.
- Ferdige annoteringer bygges automatisk hierarkisk og kan eksporteres til XML-format.
- Lagrede annoteringer kan gjenleses og utvides.
- Lagrede annoteringer kan åpnes i MATLAB for videre analyse.



University of Stavanger



Alive Wireless Heart Monitor



Product Highlights

- ✓ **Wireless** Real-time transmission to PC, Smartphone or PDA for storage and analysis
- ✓ **Bluetooth® Technology** Widely accepted wireless communication standard
- ✓ **Powerful** 48 hours continuous wireless transmission, and internal SD memory card storage option
- ✓ **Versatile** Records ECG, heart rate, activity, body position, and falls
- ✓ **Small and Mobile** Can be worn comfortably during normal daily activities with remote wireless monitoring



Specifications

General Features

ECG:	Single channel, two electrodes
Accelerometer:	3 axes (monitoring activity, body position and falls)
Patient events:	Event button
LEDs:	Bluetooth, Battery, Heart Beat
Internal Storage Type:	Secure Digital (SD) Card up to 2GB
Internal Storage Capacity:	5 days using a 256 MB SD Card

ECG

Channels	Single channel
Recording Resolution:	8 bit
Recording Sampling Rate:	300 samples/sec
Dynamic Range:	5.3mV Peak to Peak
Bandwidth	0.5Hz – 90Hz

Accelerometer

Axis:	3 axes
Recording Resolution:	8 bit
Recording Sampling Rate:	75 samples/sec
Dynamic Range:	+/-2.7g
Bandwidth	0 – 20Hz

Physical

Dimensions:	90mm (length) x 40mm (width) x 16mm (depth)
Weight:	55g with battery

Power Source

Source:	3.7 volt removable and rechargeable lithium-ion battery
Operating:	48 hrs - continuous wireless transmission
Low Battery Indication:	< 2 hours remaining

Bluetooth Features

Compliance:	Version 1.2 compliant
Classification:	Class 1 (up to 100 meter range)
Profile:	Serial Port Profile (SPP)
Operation:	Slave Point-to-Point
Antenna Type:	Internal

