

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Ljiljana Babič

**Razvoj interaktivnega grafičnega
urejevalnika za označevanje posnetkov
elektrohisterograma**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Franc Jager

Ljubljana, marec 2015

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Razvoj interaktivnega grafičnega urejevalnika za označevanje posnetkov elektrohistograma

Tematika naloge:

Razvijte interaktivni grafični urejevalnik za pregledovanje in ročno ter polavtomatsko označevanje posnetkov elektrohistograma (elektromiogram maternice). Razviti urejevalnik naj omogoča postavljanje, brisanje in pregledovanje oznak ter polno urejevanje spremljajoče klinične informacije. Tipi oznak naj se nanašajo na prave popadke, druge električne aktivnosti maternice, premike ploda in motnje v signalih. Urejevalnik razvijte z orodjem Netbeans. Temelji naj na paradigmi objekt–akcija. Pri načrtovanju urejevalnika upoštevajte splošno sprejete principe in navodila načrtovanja uporabniških vmesnikov.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisana Ljiljana Babič, z vpisno številko 63090235, sem avtorica diplomskega dela z naslovom:

Razvoj interaktivnega grafičnega urejevalnika za označevanje posnetkov elektrohistograma

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom prof. dr. Franca Jagra,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 14. marca 2015

Podpis avtorice:

Zahvala

Zahvaljujem se mentorju prof. dr. Francu Jagru za strokovno pomoč, svetovanje in usmerjanje pri izdelavi diplomske naloge. Zahvala tudi moji družini za vso podporo tekom celotnega študija.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Namen dela	1
1.2	Elektrohisterogram	2
1.3	Delovanje popadkov in maternice	3
1.4	Prezgodnji porod	5
1.5	Physionet	5
2	Metode	7
2.1	Orodja in knjižnjice	7
2.1.1	Interaktivno grafično okolje za razvoj interaktivnih grafičnih vmesnikov Netbeans	7
2.1.2	Semia	8
2.1.3	WFDB	10
2.1.4	TPEHG DB	12
2.1.5	Ponazoritev paradigm	13
2.2	Principi načrtovanja uporabniških vmesnikov	17
2.2.1	Mandelovi principi	17
2.2.2	Nielsenovi principi	17
2.3	Razvoj grafičnega orodja LANNOT	19
2.3.1	Glavno okno	20

KAZALO

2.3.2	Vrstični menu	20
2.3.3	Delovno področje orodnih vrstic	24
2.3.4	Delovno področje risalnega polja	36
2.3.5	Drsna površina	40
2.3.6	Statusna vrstica	40
3	Rezultati	43
3.1	Grafično orodje LANNOT	43
4	Razprava in zaključki	53
	Literatura	55

Slike

1.1	Posnetek vsebuje zapis električne aktivnosti (slika zgoraj) in zapis tokograma (slika spodaj) [8]	3
1.2	EHG posnetek pacientke s terminskimi popadki (primer a) in prezgodnjimi popadki (primer b) [8]	4
2.1	Glavno okno razvojnega okolja Netbeans	8
2.2	Interaktivno grafično orodje Semia 3.0 [3]	9
2.3	Zaglavna datoteka posnetka tpehg1007	12
2.4	Stanje 1: Akcija [8]	14
2.5	Stanje 2: Cancel [8]	14
2.6	Stanje 1: Objekt [8]	15
2.7	Objekt	16
2.8	Akcija	16
2.9	Glavno delovno polje vmesnika	20
2.10	Izgled vrstičnega menuja, sledimo FEVH modelu	21
2.11	Izgled orodnih vrstic	24
2.12	Delovno področje risalnega polja ob uvozu posnetka	37
2.13	Teorija upoštevanja vzorca najbližjega piksla [8]	38
2.14	Primer pravilnega izrisovanja signalov	39
2.15	Izgled drsne površine ob izbiri Examine	40
2.16	Izpis statusa ob uspešnem uvozu datoteke tpehg1007.txt	41
3.1	Prvi papirnati prototip	43
3.2	Računalniški prototip v zgodnjih fazah razvoja	44

3.3	Računalniški prototip v kasnejših fazah razvoja	44
3.4	Dialog za izbiro datoteke	45
3.5	Prikaz markerja in izpis informacij signalov	46
3.6	Označevanje popadka v desetem signalu	47
3.7	Brisanje oznake EA	48
3.8	Izpis podrobnosti oznake EN	49
3.9	Dialog za shranitev lokalnega intervala v tekstovno datoteko .	50
3.10	Izgled datoteke tpehg1007.tru. Za prikaz oznak se uporabita indeks vzorca in ime oznake	51

Seznam uporabljenih kratic

kratica	angleško	slovensko
EHG	ElectroHysteroGram	Elektrohisterogram
GUI	Graphical User Interface	Grafični uporabniški vmesnik
IDE	Integrated Development Environment	Integrirano razvojno okolje
JDK	Java Development Kit	Java razvojno okolje
JRE	Java Runtime Environment	Java izvajalno okolje
LTST DB	Long-Term ST DataBase	Podatkovna baza dolgih ambulantnih posnetkov LTST DB
TPEHG DB	Term-Preterm ElectroHysteroGram DataBase	Podatkovna baza terminskih in prezgodnjih porodov elektrohisterograma TPEHG DB
WFDB	WaveForm DataBase	Podatkovni format WFDB

Povzetek

Cilj diplomske naloge je bil razvoj interaktivnega grafičnega urejevalnika za grafično predstavitev, pregledovanje in označevanje posnetkov elektrohisterograma (elektromiogram maternice). Razvili smo grafični uporabniški vmesnik LANNOT, ki prikazuje signale posnetkov, omogoča ročno in polavtomatsko postavljanje, brisanje in pregledovanje oznak, polno urejevanje spremljajoče klinične informacije, generacijo ukaznih datotek ter generacijo novih procedur. Tipi oznak se nanašajo na prave popadke, druge električne aktivnosti maternice, premike ploda in motnje v signalih. Vmesnik je bil razvit z orodjem Netbeans IDE za razvoj interaktivnih grafičnih uporabniških vmesnikov. Razviti urejevalnik temelji na paradigmi objekt–akcija. Pri načrtovanju urejevalnika smo sledili Nielsenovim principom načrtovanja vmesnikov in splošno sprejetim navodilom načrtovanja uporabniških vmesnikov, najpomembnejša dimenzija uporabnosti pa je bila učinkovitost. Razviti urejevalnik bodo pri svojem kliničnem delu uporabljali eksperti porodničarji. Prav tako se bo urejevalnik uporabljal za označevanje posnetkov mednarodne, prosto dostopne, referenčne podatkovne baze TPEHG DB (Term Pre-term ElectroHysteroGram DataBase). Oznake postavljene s strani ekspertov porodničarjev, ki se bodo nanašale na popadke maternice in druge električne aktivnosti maternice, bodo omogočile različne vrste avtomatske analize posnetkov in vrednotenje zmogljivosti algoritmov.

Ključne besede: interaktivni grafični urejevalnik, uporabniški vmesnik, paradigma objekt-akcija, ekspertne oznake, podatkovna baza TPEHG DB.

Abstract

Goal of this thesis was development of interactive graphic editor for graphic representation, examination and annotating of records of electrohystero-gram (electromyogram of uterus). We developed a graphic user interface LANNOT, which displays signals of records, allows manual and semi-automatic setting, deleting and examining of annotations, full editing of accompanied clinical information, generation of scripts and generation of new procedures. The types of the annotations should regard true contractions, other electrical activities of uterus, baby movements, and noise of signals. The interface was developed using Netbeans IDE tool to develop interactive graphic user interfaces. The developed editor is based on object-action paradigm. During designing of the editor, we followed Nielsen's principles of interface design and commonly accepted guidelines of user interface design, while the most important usability dimension was efficiency. The developed editor will be used by expert obstetricians during their clinical praxis. It will also be used for annotating of international, freely available, reference TPEHG DB (Term Pre-term ElectroHysteroGram DataBase) database. The annotations set by expert obstetricians regarding contractions of uterus and other electrical activities of uterus, will allow different kinds of automatic analysis of records and evaluation of their performance.

Keywords: interactive graphic editor, user interface, object - action paradigm, expert annotations, TPEHG DB database.

Poglavje 1

Uvod

1.1 Namen dela

Referenčne oznake v referenčnih podatkovnih bazah bioloških signalov so potrebne, da avtorji lahko razvijajo svoje razpoznavalne algoritme, vrednotijo svoje algoritme v smislu zmogljivosti ter za primerjavo z drugimi algoritmi. Potrebne so tudi za karakterizacijo posnetkov v smislu študije fizioloških mehanizmov.

Namen dela je bilo razviti grafični urejevalnik LANNOT za pomoč pri delu porodničarjem s posnetki elektromiograma maternice in za označevanje elektromiogramskih posnetkov referenčne podatkovne baze TPEHG DB v smislu postavljanja, brisanja in pregledovanja oznak popadkov in drugih električnih aktivnosti maternice.

Posnetki elektrohistograma s terminskim ali prezgodnim porodom te podatkovne baze so na voljo na prosto dostopnem spletnem strežniku Physionet (<http://www.physionet.org/pn6/tpehgdb>).

Razviti urejevalnik LANNOT je uporabna različica interaktivnega grafičnega polavtomatskega urejevalnika Semia. Za razliko od razvitega urejevalnika, Semia temelji na paradigmi akcija-objekt. Prav tako kot je bilo naše orodje razvito za uporabo posnetkov TPEHG DB baze, je bila tudi Semia razvita z namenom uporabe označevanja posnetkov podatkovne baze LTST DB(Long-

Term ST DataBase). Označevanje z orodjem Semia se je izkazalo za težko in zamudno, zato smo se odločili za razvoj novega orodja, ki bo temeljil na paradigmi objekt-akcija.

V vmesniku je definiranih 20 tipov oznak. Razvrstili smo jih v tri klinične kategorije oznak, ki se nanašajo na prave popadke, premike ploda, druge električne aktivnosti in motnje v signalih. Za manipulacijo s podatki vmesnik vsebuje še druge nadzorne gradnike.

Z dimenzijami vmesnika smo se želeli približati tipičnim resolucijam današnjih prikazovalnikov, hkrati pa smo omogočili tudi poljubno spreminjanje dimenzij na uporabnikovo željo. Za razvoj vmesnika smo želeli uporabiti orodje Netbeans.

S podporo WFDB knjižnice in možnostjo ustvarjanja novih ukaznih datotek, smo dosegli učinkovito pretvarjanje med formati podatkovnih, zaglavnih in tekstovnih datotek.

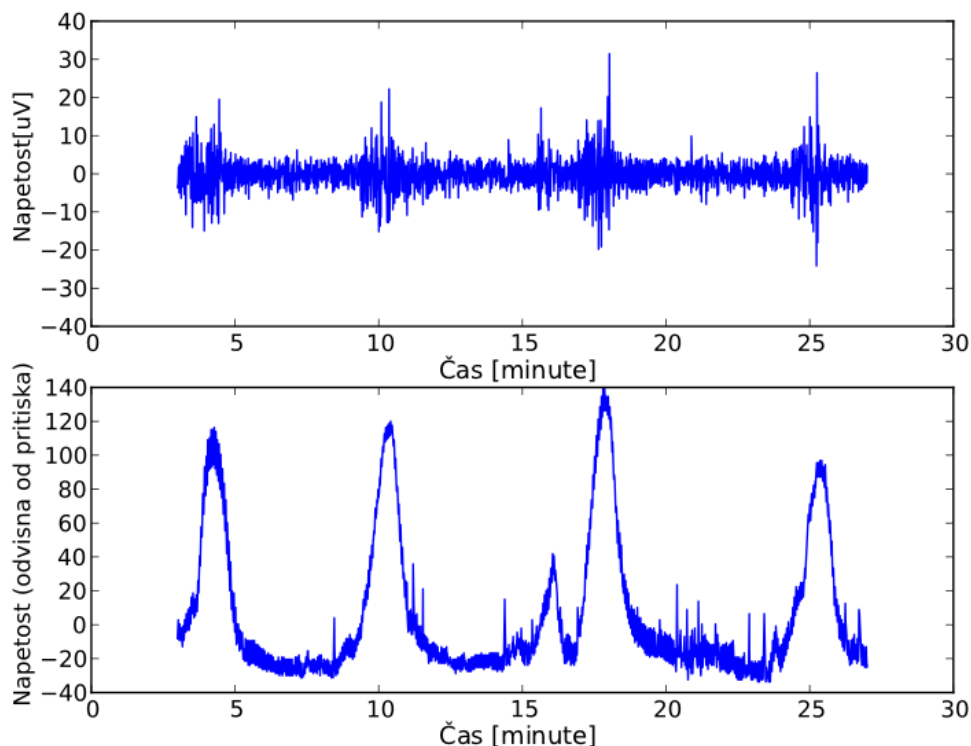
1.2 Elektrohisterogram

Elektrohisterogram je tehnika za spremljanje električne aktivnosti mišic maternice. Uvrščajo jo med boljše metode za napovedovanje prezgodnjega poroda na osnovi prezgodnjih popadkov [16].

EHG signali omogočajo zaznavanje aktivnosti maternice tako med nosečnostjo, kot tudi med porodom. Porodničarjem so v veliko pomoč pri karakterizaciji popadkov med nosečnostjo, predvsem pri populaciji, ki je najbolj izpostavljena dejavnikom predčasnega poroda [4, 10].

EHG signali so pridobljeni z merjenjem električnih aktivnosti mišic na osnovi elektrod, ki se postavijo na območje trebušne površine nosečnice. Elektrode zaznajo električno aktivnost v obliki signala elektromiograma, ki je sprožen ob krčenju mišic. Najboljši položaj elektrod je ob sredinski navpični osi trebuha [7, 12]. Nekateri posnetki poleg zapisa električne aktivnosti vsebujejo tudi zapis tokograma, ki predstavlja merilo pritiska v maternici (Slika 1.1). EHG zagotavlja še veliko dodatnih informacij o učinkovitosti popadkov, ki

lahko pripomorejo k zgodnji diagnostiki prezgodnjega poroda [17].



Slika 1.1: Posnetek vsebuje zapis električne aktivnosti (slika zgoraj) in zapis tokograma (slika spodaj) [8]

1.3 Delovanje popadkov in maternice

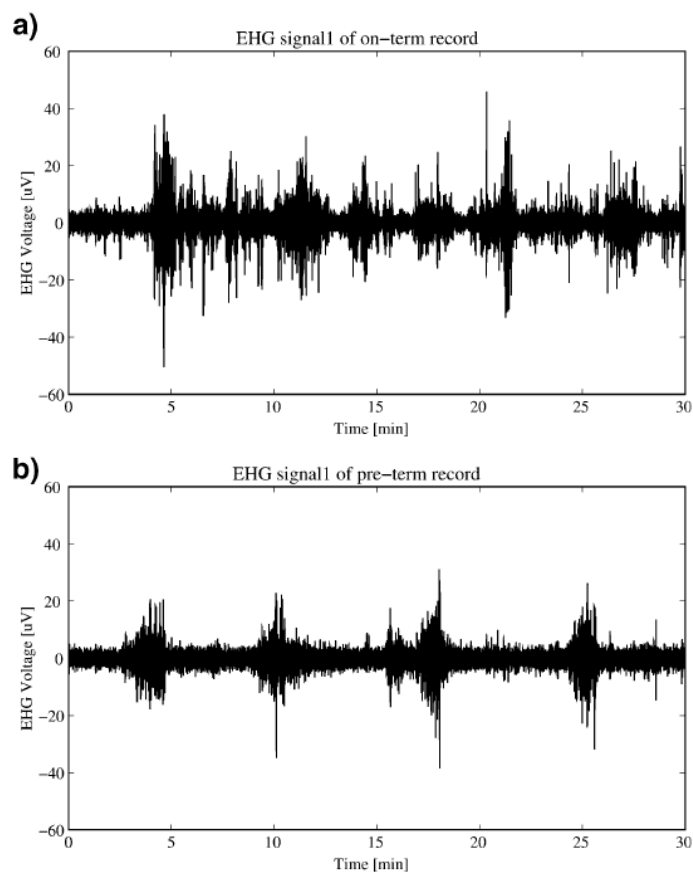
Popadek ima glavno funkcijo naprežanja mišičnega tkiva v maternici [19]. Obdobje popadkov predstavlja najdaljše obdobje poroda.

Večne polemike zdravnikov se pojavljajo pri razlikovanju normalnih popadkov ter prezgodnjih popadkov z visokim tveganjem, ki lahko vodijo k prezgodnjemu porodu. Raziskave so pokazale, da prvi znaki prezgodnjega poroda lahko nastopijo že v 27. tednu nosečnosti [12, 16].

Glavna razlika med prezgodnjimi in praviimi popadki je, da pri pravih popadkih nastopi postopno odpiranje materničnega vratu, kar je tudi povod za

pričetek poroda (Slika 1.2).

Električna aktivnost postaja sočasna sorazmerna z bližajočim se časom poroda. Vzrok za nastajanje in širitev električne aktivnosti je predvsem krčenje maternice, ki poteka z vrha maternice navzdol. Krčenje maternice se sproži ob širjenju električne aktivnosti mišičnih celic. S tem se telo nosečnice posledično pripravlja na porod.



Slika 1.2: EHG posnetek pacientke s terminskimi popadki (primer a) in prezgodnjimi popadki (primer b) [8]

1.4 Prezgodnji porod

Prezgodnji porod je glavni vzrok obolevnosti in perinatalne umrljivosti nedonošenčkov. Zaradi naraščanja odstotka prezgodnjih porodov, na področju perinatologije kljub številnim raziskavam ostaja nedonošenost še vedno velika dilema [14].

Za novorojenčka so najbolj kritični prezgodnji porodi pred obdobjem 32. in 34. tedna nosečnosti. Prezgodnji porod je povečan pri večplodnih nosečnostih, pri ženskah, ki imajo kratek materični vrat, razvojno nepravilnost maternice, so pogosto v stresu in imajo težko delo.

Kljub velikim prizadevanjem in uspešnosti neonatologije za odpravljanje teh težav, se vedno znova pojavljajo nove. Raziskovalci k problemu pristopajo na različne načine. Neprestana pogostost prezgodnjih porodov je vzrok pomanjkanja naprednih in produktivnih metod za preprečevanje [9].

Zaradi varnosti, nezadostne učinkovitosti ter stranskih učinkov tako za mater kot za plod, se pri zdravnikih pojavlja dvom pripisovanja različnih zdravil. Zavedajo se, da se da marsikaj ukreniti, da ne pride do slabih posledic, zato upoštevajo vsa pomanjkanja, prednosti, nezaželene stranske učinke in ceno prezgodnjega poroda. Pri zdravljenju in posledicah le teh, se uporabljajo antibiotiki, progesteron, tokoliki, pogosto pa je tudi izvajanje metod cerklaže.

1.5 Physionet

Spletna stran Physionet [15] izvira iz Massachusetts Institute of Technology v Cambridge-u, ZDA. Je prostodostopen vir, ki vsebuje zbirko posnetkov biomedicinskih signalov ter z njimi povezano programsko opremo. Za vzdrževanje ter dostop do spletnega strežnika Physionet na evropskem področju je zadolžen Laboratorij za biomedicinske računalniške sisteme in oslikave, s Fakultete za računalniško in informatiko [18]. Ukvarjajo se z razvojem in raziskavami biomedicinskih signalov in slik, med glavnimi raziskovalnimi temami pa je tudi razvoj interaktivnih grafičnih uporabniških urejevalnih orodij.

Stran je bila ustanovljena z namenom spodbujanja trenutnih raziskav in novih preiskav biomedicinskih in fizioloških signalov. Sestavljena je iz različnih sklopov:

1. **PhysioBank**

Arhiv, ki vsebuje digitalne posnetke fizioloških signalov in je namenjen za uporabo v biomedicinskih raziskovalnih skupnostih.

2. **PhysioToolkit**

Knjižnjica programske opreme za obdelavo in analizo signalov, s pomočjo katere odkrivajo pomembne dogodke z uporabo različnih tehnik in metod.

3. **PhysioNetWorks**

Delovni prostor dostopen samo članom, kateri imajo pravico do ogleda in sodelovanja pri aktivnih projektih.

Poglavje 2

Metode

2.1 Orodja in knjižnjice

2.1.1 Interaktivno grafično okolje za razvoj interaktivnih grafičnih vmesnikov Netbeans

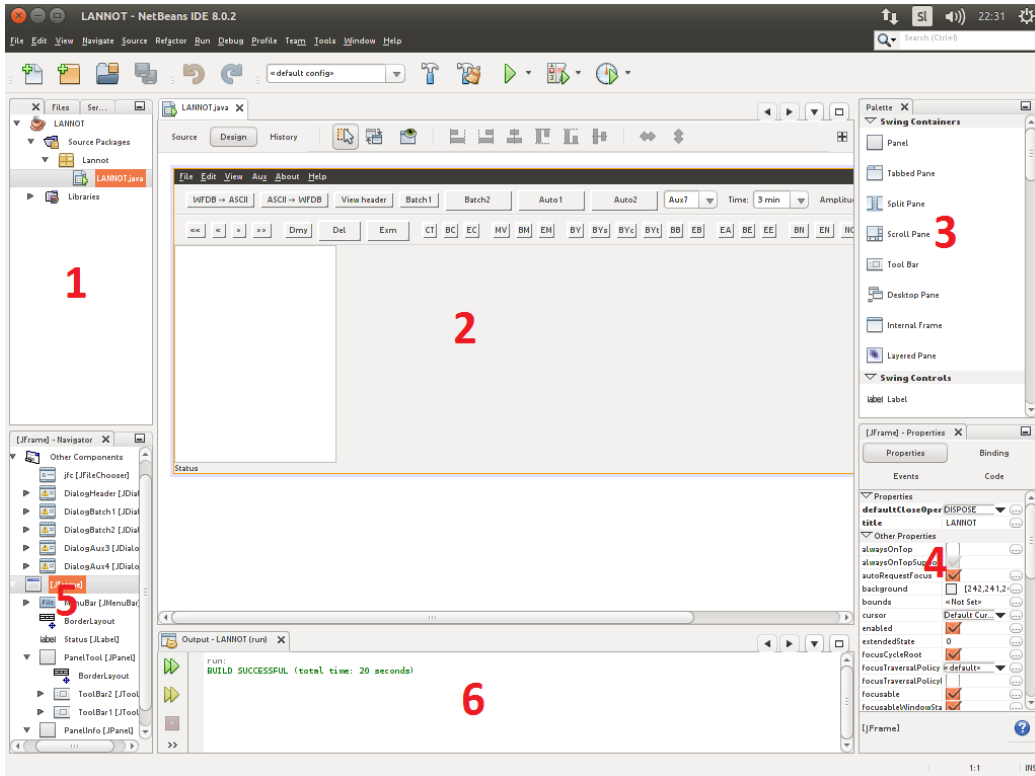
Orodje je bilo izdelano na platformi Netbeans IDE, ki je odprtokodno ter prostodostopno razvojno okolje. Razvoj je potekal na zadnji izdani verziji 8.0.2. Kodo vmesnika smo pisali s programskim jezikom Java, ki je tudi primaren jezik za izvajanje v tem programskem okolju.

Za delo z Java je potrebna namestitev programske knjižnjice JRE in JDK, ki omogočata zagon in delo z Java programi. Netbeans omogoča podporo platformam Windows, Linux, MAC OS X, Solaris in različico omejene podpore za neodvisne operacijske sisteme. Okna, ki tvorijo celotno podobo Netbeans:

1. Okno s projekti
2. Polje za načrtovanje
3. Paleta z gradniki Swing
4. Urejevalnik lastnosti

5. Struktura aplikacije

6. Izhodno okno



Slika 2.1: Glavno okno razvojnega okolja Netbeans

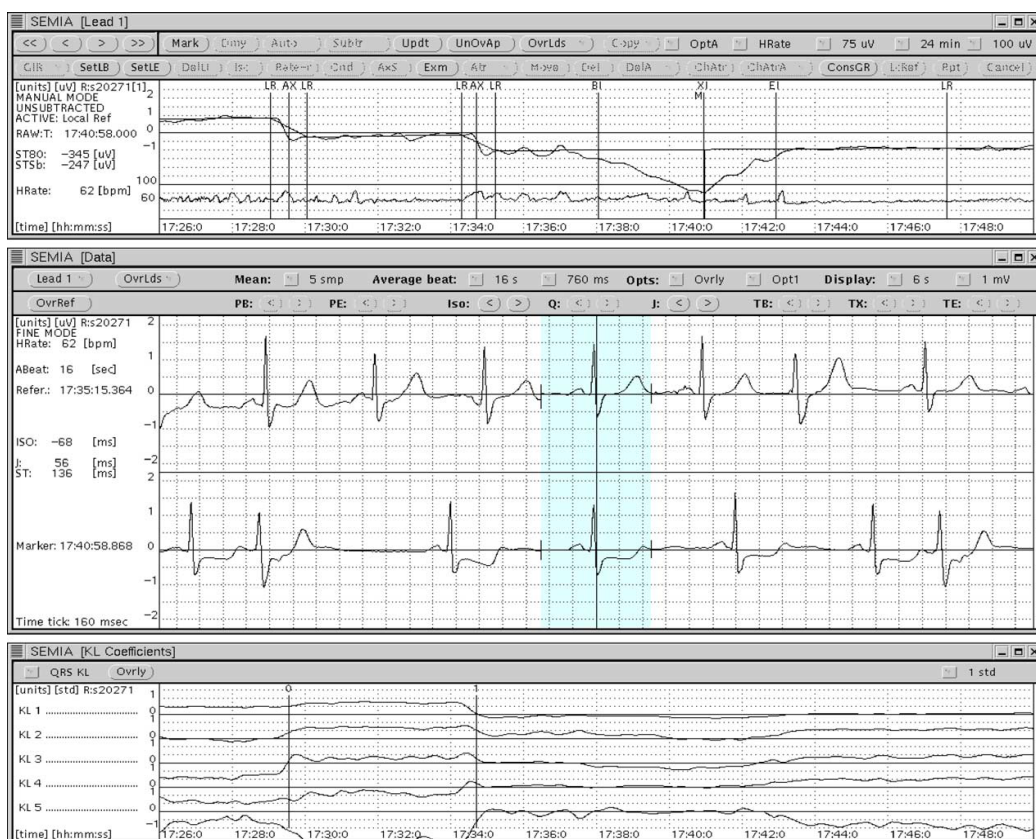
2.1.2 Semia

Semia je interaktivno grafično polavtomatsko orodje za učinkovito analiziranje ambulantnih elektrokardiogramskih posnetkov in pregledovanje časovnih vrst.

Ustanovljena je bilo v času razvoja podatkovne baze dolgih ambulantnih posnetkov LTST DB, ki je za uspešno merjenje algoritmov in analiziranje posnetkov potrebovala ustrezno orodje [3]. Razvoj je potekal v Laboratoriju za biomedicinske računalniške sisteme in oslikave na Fakulteti za računalništvo

in informatiko. Semia omogoča hiter prikaz ambulantnih posnetkov, podroben vpogled v podatke na več resolucijah, označevanje na katerikoli točki ter ročno in avtomatsko zaznavanje ST epizod [8]. Izkazalo se je, da je označevanje s Semio težko, zato smo se odločili za razvoj novega grafičnega vmesnika, ki bo temeljil na paradigmi objekt-akcija.

Semia je nastala pod okriljem operacijskega sistema UNIX na okenskem sistemu X(X Window System) s pomočjo knjižnjice Xlib in grafičnega okolja XView.



Slika 2.2: Interaktivno grafično orodje Semia 3.0 [3]

2.1.3 WFDB

Za učinkovito delo pri branju in pisanju datotek v formatih PhysioBank baz, je razvita programska oprema WFDB [13]. Uporaba je možna na vseh operacijskih sistemih, ki podpirajo prevajalnike za uporabo jezikov C, C++ in Fortran. WFDB vključuje tudi aplikacije za obdelavo signalov in interaktivno grafično okolje WAVE, ki omogoča obdelavo signalov vseh formatov, ki jih knjižnjica WFDB podpira.

Vsak posnetek formata WFDB je sestavljen iz zaglavne datoteke (tip .hea), podatkovne datoteke (tip .dat) in datoteke z oznakami (tip .atr). Za pretvorbo med datotekami nam služijo programi, ki jih poganjamo preko ukazne vrstice. V našem primeru smo uporabljali štiri najpomembnejše operacije, ki jih bo orodje upoštevalo za pretvorbo med formati:

- **rdsamp**

Naziv: *rdsamp* – *read WFDB signal files*

Sinopsis: *rdsamp -r record [options ...]*

Primer: *rdsamp -r tpehg1007 > tpehg1007.txt*

Rdsamp prebere signalno in zaglavno datoteko za dani posnetek in v izhodno datoteko zapiše vzorce v obliki decimalnih števil, ločenimi s presledki. Število decimalnih zapisov v vsaki vrstici izhodne datoteke je enako številu danih signalov v zaglavni datoteki, če zanemarimo upoštevanje prvega števila v vrstici, ki predstavlja zaporedno številko signala. V zgornjem primeru smo signalno datoteko tpehg1007.dat in zaglavno datoteko tpehg1007.hea zapisali v tekstovno datoteko tpehg1007.txt.

- **wrsamp**

Naziv: *wrsamp* – *write WFDB signal files*

Sinopsis: *wrsamp [options ...] [column ...]*

Primer: *wrsamp -o tpehg1007 < tpehg1007.txt*

Wrsamp prebere tekstovno datoteko, jo pretvori in zapiše vsebino decimalnih števil v ustvarjeno podatkovno in zaglavno datoteko. Vsebinska zaglavna datoteka v tem primeru ne vsebuje podatkov o pacientu, temveč le podatke o signalih. V zgornjem primeru smo tekstovno datoteko tpehg1007.txt pretvorili v podatkovno datoteko tpehg1007.dat in zaglavno datoteko tpehg1007.heg.

- **rdann**

Naziv: *rdann* – *read a WFDB annotation file*

Sinopsis: *rdann -r record -a annotator [options ...]*

Primer: *rdann -r s20011 -a atr > s20011.txt*

Rdann prebere datoteko z oznakami in podatkovno datoteko in zapiše pretvorbo v izhodno tekstovno datoteko. V izhodno datoteko so v vsako vrstico zapisane podrobnosti posamezne oznake. Podrobnosti sestojijo iz šestih vrednosti, ki predstavljajo čas oznake, indeks vzorca, tip oznake, podtip oznake, kanal lokacije in dodatna številka k oznaki. V zgornjem primeru smo datoteko z oznakami s20011.atr in podatkovno datoteko s20011.dat pretvorili v tekstovno datoteko s20011.txt.

- **wrann**

Naziv: *wrann* – *write a WFDB annotation file*

Sinopsis: *wrann -r record -a annotator*

Primer: *wrann -r s20011 -a atr < s20011.txt*

Wrann zapiše oznake za posnetek v datoteko z oznakami. V zgornjem primeru smo iz tekstovne datoteke s20011.txt po pretvorbi dobili podatkovno datoteko s20011.dat in datoteko z oznakami s20011.atr.

2.1.4 TPEHG DB

TPEHG DB(Term-Preterm ElectroHysteroGram DataBase) je podatkovna baza posnetkov terminskih in prezgodnjih porodov elektrohisterograma. Posnetki so nastajali pri populaciji nosečnic v 23. in 32. tednu nosečnosti, jemanje posnetkov pa je potekalo v obdobju med letoma 1997 in 2005 na Univerzitetnem kliničnem centru v Ljubljani na oddelku za ginekologijo in porodništvo [1]. Posnetki so v standardnem formatu WFDB in so sestavljeni iz dveh tipov datotek:

- **zaglavna datoteka (header file, .hea)**

Datoteko sestavljajo podatki o posnetku, kot so število signalov, frekvenca vzorčenja, kalibracija in število vzorcev in klinične informacije o pacientu (Slika 2.3).

```
tpehg1007 12 20.000000 35180
tpehg1007.dat 16 13107/mV 16 0 0 0 0 1
tpehg1007.dat 16 13107/mV 16 0 0 0 0 1_DOCFILT-4-0.08-4
tpehg1007.dat 16 13107/mV 16 0 0 0 0 1_DOCFILT-4-0.3-3
tpehg1007.dat 16 13107/mV 16 0 0 0 0 1_DOCFILT-4-0.3-4
tpehg1007.dat 16 13107/mV 16 0 0 0 0 2
tpehg1007.dat 16 13107/mV 16 0 0 0 0 2_DOCFILT-4-0.08-4
tpehg1007.dat 16 13107/mV 16 0 0 0 0 2_DOCFILT-4-0.3-3
tpehg1007.dat 16 13107/mV 16 0 0 0 0 2_DOCFILT-4-0.3-4
tpehg1007.dat 16 13107/mV 16 0 0 0 0 3
tpehg1007.dat 16 13107/mV 16 0 0 0 0 3_DOCFILT-4-0.08-4
tpehg1007.dat 16 13107/mV 16 0 0 0 0 3_DOCFILT-4-0.3-3
tpehg1007.dat 16 13107/mV 16 0 0 0 0 3_DOCFILT-4-0.3-4
#Comments:
#   RecID 1007
#   Gestation 35
#   Rectime 31.3
#   Age 30
#   Parity 0
#   Abortions 0
#   Weight 58
#   Hypertension no
#   Diabetes no
#   Placental_position front
#   Bleeding_first_trimester no
#   Bleeding_second_trimester no
#   Funneling negative
#   Smoker no
```

Slika 2.3: Zaglavna datoteka posnetka tpehg1007

- **podatkovna datoteka (data file, .dat)**

Datoteko sestavljajo vrednosti signalov.

Posnetki ne vsebujejo datotek z oznakami(annotation file, .atr).

2.1.5 Ponazoritev paradig

Paradigma akcija-objekt

Paradigma je bila uporabljena v orodju Semia. Za to paradigmo je najprej značilna zahteva akcije z izbiro nadzornega gradnika in nato objekt, nad katerim bo akcija izvršena. Dinamičen nadzor vmesnika daje na razpolago izbiro dinamičnih nadzornih gradnikov, le tistih ki so za predvideni dogodek potrebni. Ob aktiviranju določenega nadzornega gradnika, se možnost izbire aktiviranega nadzornega gradnika onemogoči za ponovno uporabo.

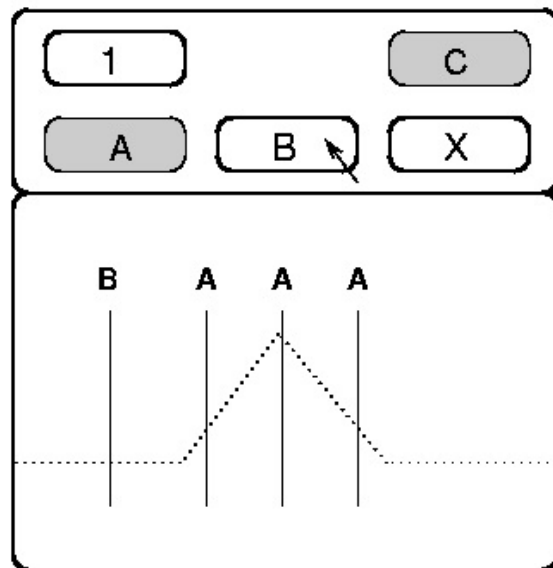
V spodnjih primerih sivi nadzorni gradniki predstavljajo onemogočene nadzorne gradnike, beli pa omogočene.

Postopek:

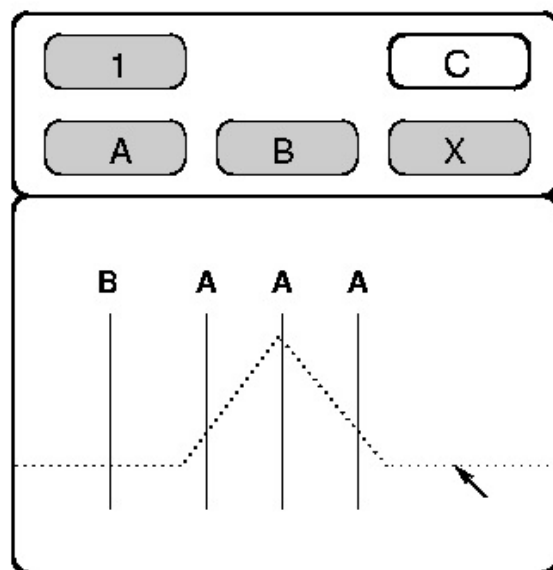
1. Uporabnik najprej aktivira nadzorni gradnik oznake B (Slika 2.4).
2. Vmesna operacija je nadzorni gradnik C (gumb Cancel) za prekinitev akcije (Slika 2.5).
3. Uporabnik izbere objekt nad katerim bo izvršena akcija (Slika 2.6).

Med akcijo in objektom obstaja vmesna operacija za prekinitev trenutnega stanja. Operacija je koristna v primeru, če se uporabnik zmede in želi spremeniti izbiro akcije. V našem primeru so po izbiri oznake ostali nadzorni gradniki onemogočeni, na voljo je le izbira Cancel za prekinitev akcije.

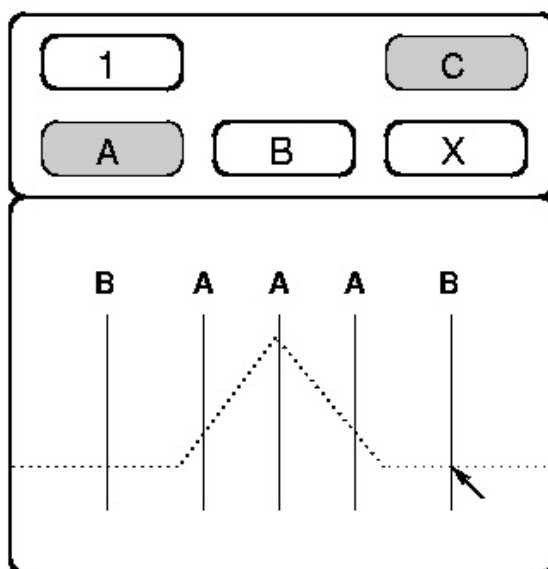
Akcija in objekt sta v istem stanju, kar je moč videti v dialogih za izbiro nadzornih gradnikov.



Slika 2.4: Stanje 1: Akcija [8]



Slika 2.5: Stanje 2: Cancel [8]



Slika 2.6: Stanje 1: Objekt [8]

Paradigma objekt-akcija

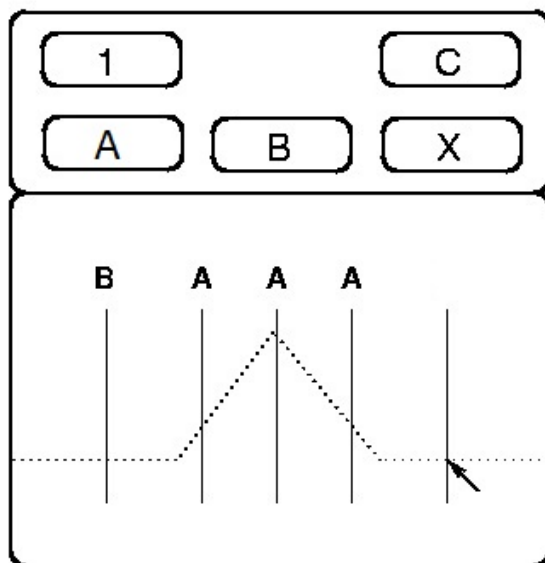
To paradigmo smo uporabili v razvitem orodju LANNOT. Vmesnik ne vsebuje dinamičnih nadzornih gradnikov, pač pa daje uporabniku na razpolago prosto izbiro vseh nadzornih gradnikov, hkrati pa ob izbiri določenega nadzornega gradnika ponuja možnost izbire istega nadzornega gradnika v vseh scenarijih.

Z vidika uporabnika je paradigma bolj preprosta in prijazna, v današnjih časih pa je to tudi splošno sprejeta paradigma modernih orodij.

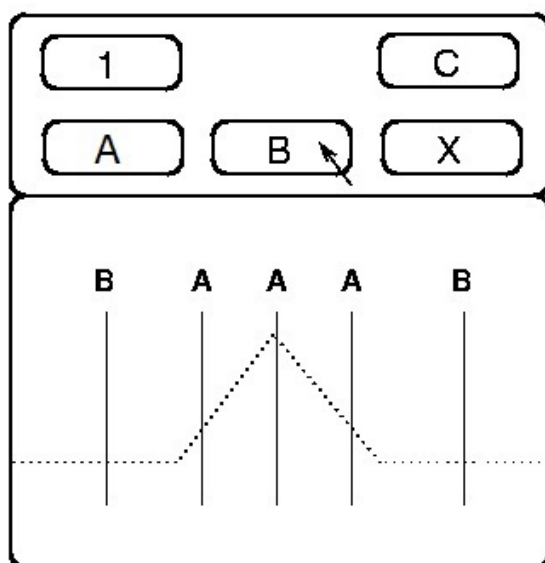
V spodnjih primerih imamo tako na voljo vse bele nadzorne gradnike.

Postopek:

1. Uporabnik najprej izbere objekt, nad katerim bo izvršena akcija in premakne marker na ustrezno mesto v polju za risanje (Slike 2.7).
2. Uporabnik izbere nadzorni gradnik in s tem aktivira akcijo na izbranem objektu (Slika 2.8).



Slika 2.7: Objekt



Slika 2.8: Akcija

2.2 Principi načrtovanja uporabniških vmesnikov

Pri načrtovanju grafičnega orodja LANNOT smo upoštevali splošno sprejete Mandelove [11] in Nielsenove [5] principe načrtovanja uporabniških vmesnikov ter splošno sprejeta navodila načrtovanja [6].

2.2.1 Mandelovi principi

1. Zagotovi nadzor uporabnika
2. Reduciraj obremenitev uporabnikovega spomina
3. Zagotovi konsistentnost

2.2.2 Nielsenovi principi

1. Prilagodi se realnemu svetu
Poimenovanje gradnikov je definirano v skladu z željami porodničarjev. V oznakah so uporabljene kratice, jezik je enostaven in prilagojen porodničarjem.
2. Konsistentnost in standardi
Vrsti red gradnikov je grupiran po kategorijah. Gradniki istih kategorij so enakega videza in so poravnani v enaki liniji. Uporabniku je omogočena dobra orientacija za izbiro gradnikov. Oznake so v sklopu s terminologijo porodničarjev.
3. Pomoč in dokumentacija
Dokumentacija za uporabo orodja je ta diplomska naloga.
4. Uporabnikov nadzor in svoboda
Glavno okno in vsi uporabljeni dialogi imajo jasno označno izbiro za izhod. Nadzorni gradnik za razveljavitev ni potreben, saj dialog lahko nemudoma zapustimo brez shranjevanja, v glavnem oknu pa vsaka nova

akcija povozi prejšnjo. Uporabnik lahko ob vsakem trenutku zapusti neželjeno stanje ali povzročeno napako. Uporabnik lahko prosto pregleduje podatke z uporabo nadzornih gradnikov za premik ekrana. Polna manipulacija s podatki je omogočena z izbiro časovne in amplitudne skale.

5. Vidljivost statusa sistema

Na dnu vmesnika je statusna vrstica, ki izpisuje trenutni status. Status se spreminja sočasno z uporabnikovimi akcijami. Poseben pregled nad dogajanjem po risalnem polju nam omogoča drsna površina, ki izpisuje informacije signalov. S pomočjo kurzorja dobimo trenutno informacijo o signalu iz kurzorja, ki smo ga definirali v markerju.

6. Fleksibilnost in učinkovitost

Vsi menuji in končne izbire imajo definirane bližnjice in mnemonike za hitrejše delo z uporabo kombinacij tipk preko tipkovnice. Na voljo imamo delo z ukaznimi datotekami, ki omogočajo izvajanje procesov v ozadju orodja, ne da bi pri tem potrebovali zaganjati temu pripisanih knjižnic.

7. Izogibanje napakam

Izbiro časovne in amplitudne skale smo definirali s kombiniranimi izvlečnimi seznamami. Za ugoden izris signalov smo za skale definirali skrbno izbrana števila, s čimer smo uporabniku onemogočili vnos skale, ki ga na risalnem polju nebi bilo moč izrisati.

8. Raje prepoznaj kot si zapomni

V vmesniku ni nobenega polja za vnos (vnos podatkov preko tipkovnice). Sama zasnova vmesnika ne zahteva takšnih gradnikov, hkrati pa je uporabnik manj podvržen napakam. Navigacija večinoma poteka preko nadzornih gradnikov in kombiniranih izvlečnih seznamov.

9. Javljanje napak, diagnoza, reševanje

Ob napačni izbiri formata uvoženega posnetka, nam vmesnik izpiše da

izbrana datoteka ni v zahtevanem formatu. V primeru uporabnikovega klika izven mej risalnega polja, sledi izpis obvestila o položaju klika izven predpisanih mej. V konkretnem primeru predlagana rešitev ni potrebna.

10. Estetika in minimalistično načrtovanje

Estetski izgled vmesnika je preprost, načrtan na osnovi principa manj je več. Povezujejo ga preproste linije z osnovnimi gradniki, kjer ni nobene detaljne informacije. Barve smo dodali konzervativno (vmesnik je skoraj enoličen), skrbno smo le definirali rdečo barvo markerja in modro barvo oznake, ki omogočata dober pregled nad risalnim poljem. Povsod je uporabljen osnoven font *Ubuntu 9 Plain*. Nadzorni gradniki so obogateni z okvirjem tipa *Soft bevel border*, s katerim smo dosegli boljšo vidljivost. Dialogi so minimalistično načrtani, uporabili smo standardne modalne in nemodalne.

V smislu uporabnosti je grafično orodje naučljivo, učinkovito in varno. Najpomembnejši kriterij uporabnosti je bila učinkovitost, v smislu učinkovitega upravljanja z orodjem.

2.3 Razvoj grafičnega orodja LANNOT

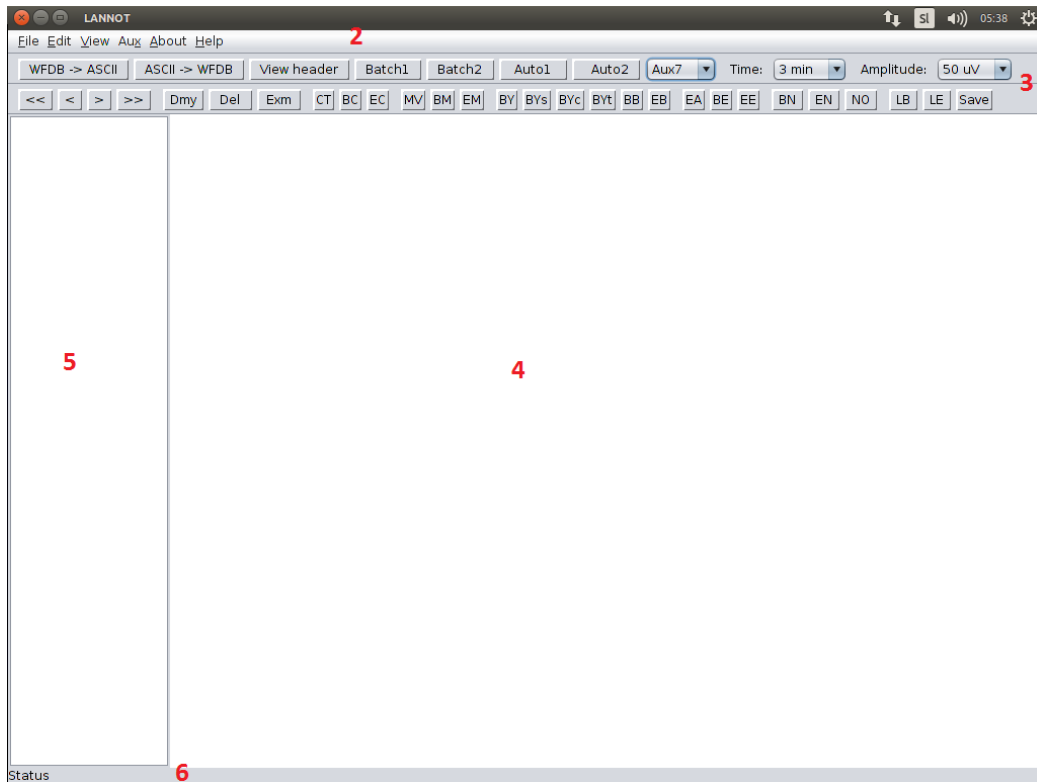
Grafično orodje LANNOT smo razvili v okolju Netbeans IDE. Orodje temelji na glavnem samostojnem oknu (JFrame Form), ki spada v kategorijo vsebovalnikov Swing GUI Forms in deluje na osnovi projekta tipa Java Application. Glavne komponente grafičnega uporabniškega vmesnika so (Slika 2.9):

1. Glavno okno (JFrame)
2. Vrstični menu (JMenuBar)
3. Delovno področje orodnih vrstic (JPanel)
4. Delovno področje risalnega polja (JPanel)

5. Drsna površina (JScrollPane)
6. Statusna vrstica (JLabel)

2.3.1 Glavno okno

Je osnovna komponenta, v katero smo postavili ostale glavne gradnike. Lokacijo gradnikov smo v glavnem oknu razporedili s privzetim razporejevalnikom Border Layout.



Slika 2.9: Glavno delovno polje vmesnika

2.3.2 Vrstični menu

Vrstični menu sledi FEVH (File, Edit, View, Help) modelu. Definiran je z opcijami File, Edit, View, Aux, About, Help. Vsi so kaskadni nadzorni

gradniki, katerim sledijo izvlečni menuji (JMenuItem). Omogočeni so ekvivalenti preko tipkovnice, opcije so definirane z ustreznimi mnemoniki in bližnjicami (Slika 2.10).



Slika 2.10: Izgled vrstičnega menuja, sledimo FEVH modelu

File

Menu (JMenu) File nam omogoča odpiranje, shranjevanje in izhod iz vmesnika. Vsebuje opcije:

- **New**
Omogoča enako delovanje kot opcija Open.
- **Open**
Povzroči klic dialoga za izbiro datoteke (JFileChooser). Dialog zahteva izbiro zaglavne datoteke s končnico `.hea`. Če se zmotimo in izberemo datoteko drugega tipa, nas vmesnik opozori, da tip izbire ni pravi.
- **Save**
Omogoča zapis informacij dodanih oznak risalnega polja v tekstovno datoteko. Ustvarjena datoteka za osnovo dobi ime posnetka, za končnico pa format `.tru`. Vsaka oznaka v datoteko doda novo vrstico, ki vsebuje čas, indeks vzorca, oznako in tri vrednosti ki predstavljajo podtip, kanal lokacije in dodatno število k oznaki, ki smo jih v našem primeru definirali kot nične vrednosti.
- **Save As**
Omogoča enako funkcionalnost kot Save, le da se pri tej opciji sproži dialog za izbiro datoteke, v katerem izberemo poljubno izhodno datoteko za zapis dodanih oznak risalnega polja.

- **Print**

Namenjen razširitvi vmesnika v prihodnosti.

- **Aux1**

Namenjen razširitvi vmesnika v prihodnosti.

- **Dump**

Sproži pojavitev kurzorja, s katerim se usmerimo po ekranu in ob kliku naredimo sliko ekrana. Slika se shrani v posebno datoteko delovnega direktorija.

- **Quit**

Izhod iz vmesnika.

Edit

Menu za urejanje, ki v vsaki končni izbiri omogoča prikaz dialoga za delo z datotekami in besedilom. Vsak dialog vsebuje menu File, z opcijami:

- **Open**

Prikaz dialoga za izbiro tekstovne datoteke, katera se nato prikaže v besedilnem območju dialoga.

- **Save**

Besedilo lahko shranimo v poljubno tekstovno datoteko.

- **Quit**

Izhod iz dialoga.

Menu Edit vsebuje opcije:

- **Header**

Ob zagonu dialog že vsebuje celotno vsebino trenutno uporabljane zaglavne datoteke posnetka. Vsebino dialoga lahko poljubno spreminjamo, ji dodajamo besedilo in zapisujemo v novo izbrano datoteko.

- **Batch1**

Prazen dialog, v katerega vpišemo ukaz, ki bi ga želeli kasneje izvajati.

- **Batch2**

Prazen dialog, v katerega vpišemo ukaz, ki bi ga želeli kasneje izvajati.

- **Aux3**

Namenjen razširitvi vmesnika v prihodnosti.

- **Aux4**

Namenjen razširitvi vmesnika v prihodnosti.

View

Menu View omogoča pregled in izbiro različnih signalov. Maksimalno število signalov je 20. Ob uvozu datoteke, vmesnik iz zaglavne datoteke prebere osnovne lastnosti posnetka, med katerimi je tudi število signalov. Signali, ki presegajo število teh signalov, so avtomatsko onemogočeni. Izbira signalov, ki jih v zaglavni datoteki ni, tako ni mogoča. Izbiramo lahko poljubne signale, največje število prikazanih je štiri, najmanjše pa ena. V primeru izbora več kot štirih signalov, so izrisani prvi štirje po vrstnem redu.

Aux

Vsebuje opciji:

- **Aux5**

Namenjen razširitvi vmesnika v prihodnosti.

- **Aux6**

Namenjen razširitvi vmesnika v prihodnosti.

About

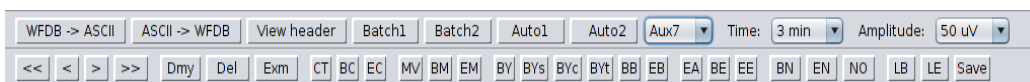
Diploma thesis Ljiljana Babič.

Help

See diploma thesis Ljiljana Babič.

2.3.3 Delovno področje orodnih vrstic

Delovno področje privzetega okna (JPanel) je namenjeno prikazu dveh orodnih vrstic (JToolBar) (Slika 2.11). Pozicionirano je na vrhu vmesnika pod glavnim menujem in v neposredni bližini polja za risanje.



Slika 2.11: Izgled orodnih vrstic

Prva orodna vrstica je namenjena pretvarjanju datotek ob zagonu ukazov preko ukazne vrstice in izbiri časovne in amplitudne skale.

Leva stran vsebuje nadzorne gradnike:

- **WFDB→ASCII**

Preusmeritev celotne vsebine iz podatkovne in zaglavne datoteke v tekstovno datoteko, s pomočjo ukaznega programa *rdsamp*.

- **ASCII→WFDB**

Obratna preusmeritev celotne vsebine, iz tekstovne datoteke v zaglavno in podatkovno datoteko, s pomočjo ukaznega programa *wrsamp*.

- **View header**

Vpogled v vsebino zaglavne datoteke. Za razliko od izbire Header v meniju Edit nam ta opcija omogoča samo vpogled v vsebino brez možnosti urejanja.

- **Batch1**

Sproži zagon ukazne datoteke Batch1, ki je definirana v končni izbiri Batch1 menuja Edit.

- **Batch2**

Sproži zagon ukazne datoteke Batch2, ki je definirana v končni izbiri Batch2 menuja Edit.

- **Auto1**

Namenjen avtomatski analizi lokalnih intervalov. Ni končno definiran.

- **Auto2**

Namenjen avtomatski analizi lokalnih intervalov. Ni končno definiran.

Desna stran prve orodne vrstice je sestavljena iz treh kombiniranih izvlečnih seznamov (JComboBox):

- **Aux7**

Namenjen razširitvi vmesnika v prihodnosti.

- **Time**

Izbor časovne skale. Prvotni čas ob uvozu posnetka je 3 min. Izbor časovnih skal: 1 s, 3 s, 5 s, 8 s, 15 s, 45 s, 3 min, 6 min, 12 min, 15 min, 18 min, 21 min, 24 min, 27 min, 30 min, 60 min.

Privzeta izpisujoča časovna skala je v sekundah, ne glede na enoto uporabnikove izbire. Če je izbrana časovna skala v minutah, funkcija to ustrezno pretvori.

- **Amplitude**

Izbor amplitudne skale. Prvotna amplituda ob uvozu posnetka je 50 μV . Izbor amplitudnih skal: 1 μV , 2 μV , 5 μV , 10 μV , 20 μV , 50 μV , 100 μV , 200 μV , 500 μV , 1 mV , 2 mV , 4 mV , 5 mV , 10 mV , 20 mV . Privzeta amplitudna skala je milivolt (mV). Ob izboru enote μV se vrednost izbora ustrezno pretvori v privzeto.

Med delom lahko kadarkoli spremenimo izbor časovne ali amplitudne skale.

Druga orodna vrstica je namenjena premikanju po risalnemu polju in delu z oznakami.

Skrajno levo so oznake, ki se najpogosteje uporabljajo. Definirani so štirje nadzorni gradniki za premik površine za pol ekrana ali cel ekran. Te so:

- <<
Premik površine v levo za cel ekran.
- <
Premik površine v levo za pol ekrana.
- >
Premik površine v desno za pol ekrana.
- >>
Premik površine v desno za cel ekran.

Sledi kategorija treh nadzornih gradnikov za delo z oznakami:

- **Dmy (Dummy)**
Prazna oznaka brez vsebine.
- **Del (Delete annotation)**
Brisanje postavljenih oznak. Funkcionalnost zahteva postavitev markerja v neposredno bližino oznake. Če je marker dovolj blizu, se ob pritisku na nadzorni gradnik Del oznaka izbriše. Uspešna akcija je vidna tudi v statusni vrstici, ki izpiše katera oznaka na kateri poziciji je bila izbrisana. V primeru da marker ni postavljen dovolj blizu oznake, nas statusna vrstica obvesti, da v bližini markerja ni bila najdena nobena oznaka.

Primer kode za brisanje oznak z glavnimi spremenljivkami:

- * markInd - indeks markerja

- * line - število vzorcev
- * distance - največja dovoljena razdalja med markerjem in oznako
- * annNr - zaporedna številka oznake
- * disMa, annDif - razdalja med markerjem in oznako
- * annInd - indeks oznake
- * lowT - začetni čas prikazanega risalnega polja
- * hghT - končni čas prikazanega risalnega polja
- * annot - seznam oznak (vsebuje indeks in tip)
- * status - spremenljivka statusne vrstice

```
public void deleteAnnotation () {
    if (markInd > 0) {
        if (line > 0) {
            double xScale = (getSize().width - 30 - (xStart))
                / (double) (hghT - lowT);
            int distance = (int) Math.ceil(5 / xScale);
            int annNr = -1;
            int annDif = -1;
            for (int i = 0; i < annot.size(); i++) {
                int annInd = (int) annot.get(i).getIndex();
                if (annInd >= lowT && annInd <= hghT) {
                    int disMa = (int) Math.abs(markInd - annInd);
                    if ((annNr < 0 || (annDif > disMa)) && (disMa
                        < distance)) {
                        annNr = i;
                        annDif = disMa;
                    }
                }
            }
        }
    }
}
```

```
    }  
    if (annNr >= 0) {  
        status.setText("Removed_annotation_" + annot.  
            get(annNr).getType() + "_at_index_" + annot.  
            get(annNr).getIndex());  
        annot.remove(annNr);  
    } else {  
        status.setText("Annotation_not_found");  
    }  
    drawAnnotationsToPixmap();  
}  
}  
repaint();  
}
```

- **Exm (Examine)**

Pregledovanje oznak. Deluje po enakem principu kot funkcija Delete. Ob pritisku na nadzorni gradnik Exm, marker skoči na želeno oznako in izpiše podrobnosti oznake v drsni površini. Uspešen pregled oznake na njenem indeksu nam izpiše tudi statusna vrstica. Za razliko od funkcije Delete, je v tej funkciji še definirana postavitev markerja, ki skoči na oznako, kadar je marker dovolj blizu. Kadar marker ni dovolj blizu, pregled ni možen.

Primer kode za pregledovanje oznak z glavnimi spremenljivkami:

- * markInd - indeks markerja
- * line - število vzorcev
- * distance - največja dovoljena razdalja med markerjem in oznako
- * annNr - zaporedna številka oznake

- * disMa, annDif - razdalja med markerjem in oznako
- * annInd - indeks oznake
- * lowT - začetni čas prikazanega risalnega polja
- * hghT - končni čas prikazanega risalnega polja
- * annot - seznam oznak (vsebuje indeks in tip)
- * g2d - razred za risanje
- * signals - struktura ki vsebuje vrednosti signalov
- * status - spremenljivka statusne vrstice
- * paneThird - spremenljivka drsne površine ob opciji examine

```
public void examine() {
    if (markInd > 0) {
        if (line > 0) {
            double xScale = (getSize().width - 30 - (xStart))
                / (double) (hghT - lowT);
            int distance = (int) Math.ceil(5 / xScale);
            int annNr = -1;
            int annDif = -1;
            for (int i = 0; i < annot.size(); i++) {
                int annInd = (int) annot.get(i).getIndex();
                if (annInd >= lowT && annInd <= hghT) {
                    int disMa = (int) Math.abs(markInd - annIdx);
                    if ((annNr < 0 || (annDif > disMa)) && (disMa
                        < distance)) {
                        annNr = i;
                        annDif = disMa;
                        markInd = annInd;
                    }
                }
            }
        }
    }
}
```

```
    }  
  }  
  if (annNr >= 0) {  
    paneThird = "";  
    if (markInd > -1) {  
      too = new BufferedImage(getSize().width - 30,  
        getSize().height, BufferedImage.  
        TYPE_INT_ARGB);  
      Graphics2D g2d = (Graphics2D) too.getGraphics  
        ();  
      int xPos = (int) (Math.ceil((markInd - lowT) *  
        xScale + xStart));  
      if (xPos < xStart || xPos > pixmap.getWidth(  
        null)) {  
        return;  
      }  
      g2d.setColor(Color.red);  
      g2d.drawLine(xPos, yStart, xPos, pixmap.  
        getHeight(null) - yStart - 4);  
      xPosX = annot.get(annNr).getIndex();  
      status.setText("Examined annotation " + annot.  
        get(annNr).getType() + " at index " +  
        annot.get(annNr).getIndex());  
      typeExm = annot.get(annNr).getType();  
  
      TreeMap<Integer, Double> signals = new TreeMap  
        <Integer, Double>();  
      signals.put(first, value1);  
      signals.put(second, value2);  
      signals.put(third, value3);  
      signals.put(fourth, value4);
```

```
paneThird += "Record:␣"+getFile;
for (Integer signal : signals.keySet()) {
    if (signal != 0) {
        Double value = signals.get(signal);
        value = (data[col + (signal - 1)][((int)
            xPosX)]);
        String time = convert(xPosX);
        paneThird += "\n\n\n\n\n" + signal + ".␣
            Signal\nFile:␣" + name + "\nValue:␣" +
            value + "␣[mV]" + "\nPosX:" + time + "[s]" +
            "\nType:␣" + typeExm;
    }
}
drawAnnotationsToPixmap();
setPane();
}
} else {
    status.setText("Annotation␣not␣found");
}
}
}
repaint();
}
```

Preostale nadzorne gradnike smo razvrstili v tri kategorije kliničnih oznak. Te so:

1. Pravi popadki

Oznake ki spadajo v to kategorijo:

- **CT (Contraction)**
Popadek.
- **BC (Begin contraction)**
Začetek popadka.
- **EC (End contraction)**
Konec popadka.

2. Otrok v povezavi z mamo

Oznake opisujejo dogodke, ki jih je povzročila mama. Porodničarji takim dogodkom pripisujejo oznako Baby, čeprav so povezani z mamo. Najbolj pogosta dejanja med nosečnostjo smo pripisali maminemu smehu, kašlju in govoru. Oznake ki spadajo v to kategorijo:

- **MV (Movement)**
Premik otroka.
- **BM (Begin movement)**
Začetek premikanja (npr. brcanje).
- **EM (End movement)**
Konec premika.
- **BY (Baby)**
Otrok povzroči popadek.
- **BYs (Baby smile)**
Smeh.
- **BYc (Baby cough)**
Kašelj.

- **BYt (Baby talk)**
Govor.
- **BB (Baby begining)**
- **EB (End baby)**

3. Električna aktivnost maternice

Aktivnost, kjer niso prisotni pravi popadki. Oznake ki spadajo v to kategorijo:

- **EA (Electrical activity)**
Električna aktivnost.
- **BE (Begin electrical activity)**
Začetek električne aktivnosti.
- **EE (End electrical activity)**
Konec električne aktivnosti.
- **BN (Begin noise)**
Začetek šuma.
- **EN (End noise)**
Konec šuma.
- **NO (Noise)**
Šum.
- **LB (Local interval begining)**
Začetek lokalnega intervala.
- **LE (Local interval end)**
Konec lokalnega intervala.
- **Save**
Shrani območje lokalnega intervala. Ob želji uporabnika za pregled nad signali na določenem mestu, se postavi oznaka LB za začetek in LE za konec označevanja. Ob kliku na nadzorni gradnik Save, uporabnik preko dialoga za izbiro datoteke, izbere datoteko,


```
        continue;
    }
}
if (br != 1) {
    annot.add(new DrawingArea.Annotation1(markInd
        , type));
    markIndTable[tab] = markInd;
    tab++;
    Collections.sort(annot);
    drawAnnotationsToPixmap();
    status.setText("Set new annotation at index: "
        + markInd);
}
} else {
    annot.add(new DrawingArea.Annotation1(markInd,
        type));
    markIndTable[tab] = markInd;
    tab++;
    Collections.sort(annot);
    drawAnnotationsToPixmap();
    status.setText("Set new annotation at index: "
        + markInd);
}
}
repaint();
} else {
    JOptionPane.showMessageDialog(this, "Trying to set
        annotation outside the editing field!");
}
}
```

2.3.4 Delovno področje risalnega polja

Najbolj pomembna struktura glavnega okna je risalno polje. Površina nam omogoča izris izbranih signalov (Slika 2.12). Izris celotne površine s signali smo dosegli s strukturo *DrawingArea*. Celotni razred z definiranimi funkcijami smo poimenovali *DrawingArea*, katerega razširja vsebovalnik plošča (JPanel). Med gradniki smo definirali spremenljivko *drawingArea*, ki predstavlja ploščo s signali.

Povezavo z miško smo definirali v razredu *MyMouseMoveListener* z implementacijo dogodkov *MouseListener* in *MouseEvent*, ki sta zadolžena za sledenje miškinim akcijam. Z implementacijo teh dogodkov se nam avtomatsko prikažejo metode, ki služijo dejanjem miške. Za potrebe našega vmesnika smo uporabili metodi *mouseDragged(MouseEvent e)* za vleko miške in *mouseClicked(MouseEvent e)* za klik miške.

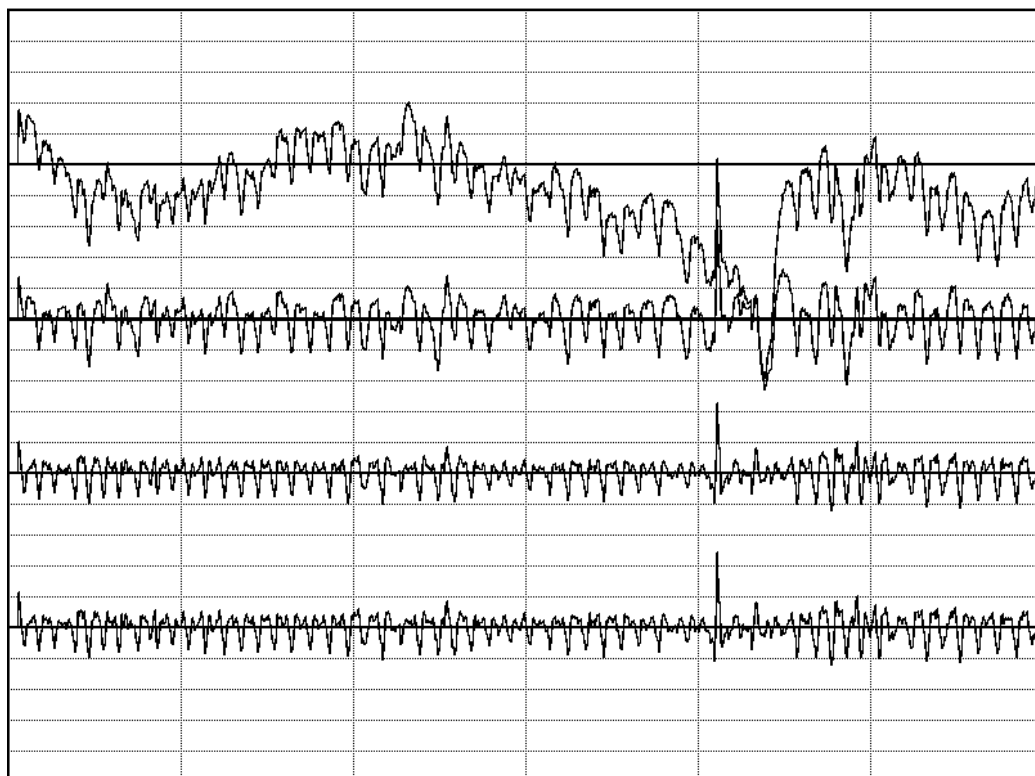
Primer kode:

```
class MyMouseMoveListener implements MouseInputListener
{
    @Override
    public void mouseClicked(MouseEvent e) {
        drawPositionLine(e);
        repaint();
    }

    @Override
    public void mouseDragged(MouseEvent e) {
        drawPositionLine(e);
        repaint();
    }
}
```


Povezavo s komponentami smo definirali v razredu *MyComponentListener* z implementacijo dogodkov *ComponentListener* in *ComponentEvent*, ki se odzivata na dogodke povezane s komponentami. Metoda, ki smo jo potrebovali za implementacijo premika vmesnika je *componentResized(ComponentEvent e)*.

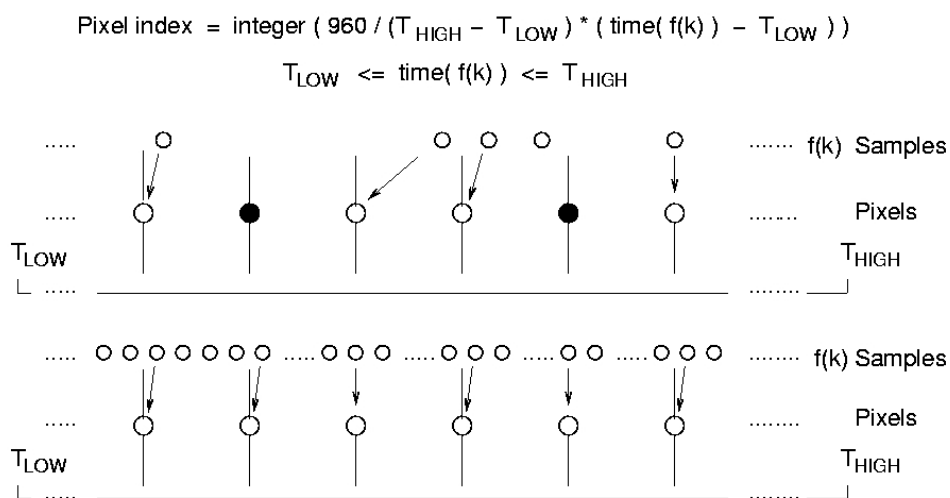
```
class MyComponentListener implements ComponentListener {
    @Override
    public void componentResized(ComponentEvent e) {
        drawAllToPixmaps();
        repaint();
    }
}
```



Slika 2.12: Delovno področje risalnega polja ob uvozu posnetka

Izrisovanje signalov

Risalno polje omogoča prikaz največ štirih signalov. Izris signalov smo dosegli s prikazovanjem pikselov posameznih vzorcev, katere smo medseboj povezovali z vektorskim risanjem črt na osnovi koordinatnega sistema. Prikaz vzorcev temelji na teoriji upoštevanja vzorca najbližjega piksla (Slika 2.13). Na ta način je onemogočen prikaz vseh vzorcev.



Slika 2.13: Teorija upoštevanja vzorca najbližjega piksla [8]

Potrebno je bilo poskrbeti za skladno prikazovanje signalov na različnih časovnih skalah. Tu igrata pomembno vlogo izračun relativnega faktorja glede na izbiro časovne skale in odmika.

$$f_1 = \frac{K_v}{1 \text{ mV}} T_s , \quad (2.1)$$

kjer je f_1 relativni faktor, K_v vrednost kalibracije in T_s izbrana časovna skala. Vrednost kalibracije dobimo iz zaglavne datoteke. Izbrana časovna skala ima ne glede na enoto izbora vedno končno vrednost v milivoltih (mV). Izbrana skala se pokrajša z 1 mV in končni rezultat enačbe je breznoten. Dobljena vrednost enačbe je relativni faktor glede na izbrano časovno skalo.

$$O = \frac{x[i]}{f_1} N_{pt} , \quad (2.2)$$

kjer je O odmik, $x[i]$ vrednost podatka, f_1 relativni faktor in N_{pt} število pikslov na enoto. Slednja enačba je enačba odmika. $x[i]$ je vrednost podatka v mV , ki ga dobimo ob branju tekstovne datoteke. Relativni faktor dobimo iz prejšnje enačbe, število izbranih pikslov na enoto pa smo v vmesniku definirali na 30 pikslov/enoto. Tolikšna je tudi razlika med posameznimi števci ob y osi (2, 1, 0, -1, -2). Funkcija odmika se izvaja neprestano.

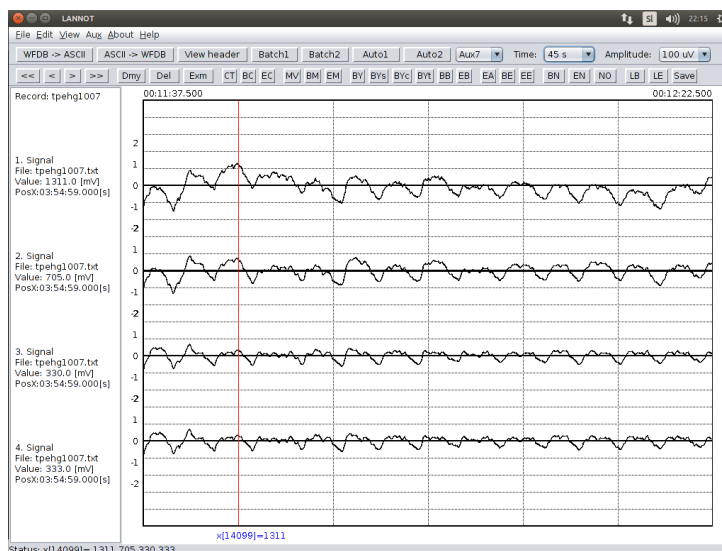
Primer izračuna:

$$T_s = 100 \mu V$$

$$K_v = 13107$$

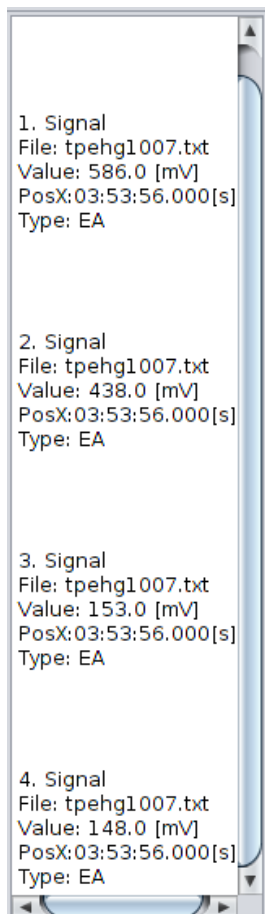
$$\begin{aligned} f_1 &= \frac{13107}{1 \text{ mV}} 100 \mu V \\ &= 1310.7 \text{ mV} \end{aligned}$$

Če primerjamo dobljeno vrednost in vrednost prvega signala, ki ga prikazuje skala 1 na y časovni osi (Value 1311.0 [mV]), lahko sklepamo da enačba pravilno funkcioniira (Slika 2.14).



Slika 2.14: Primer pravilnega izrisovanja signalov

2.3.5 Drсна površina



Slika 2.15: Izgled drsne površine ob izbiri Examine

Namenjena je prikazu urejevalnika teksta. Zaseda levo mesto vmesnika in prikazuje podrobnosti signalov (Slika 2.15). Te so:

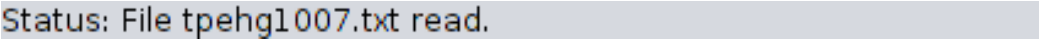
- Oznaka signala
- Ime posnetka
- Vrednost funkcije
- Indeks vzorca

Ob kliku na risalno površino dobimo indeks vzorca na osnovi koordinatnega sistema, kjer se vmesnik sprehodi po koordinatah x in poišče najbližji podatek.

Vsebina urejevalnika se osvežuje sočasno s premikanjem markerja. Ob izbiri nadzornega gradnika Examine, se vsebina urejevalnika osveži sočasno z izbrano oznako, katera poleg osnovnih informacij nosi tudi informacijo o tipu oznake.

2.3.6 Statusna vrstica

Izpisuje status ob spremenjenih akcijah. Ob uspešni izbiri datoteke nam status izpiše naziv datoteke. Ob dodajanju in brisanju oznak nam izpiše izbrano oznako z njenim indeksom. Med sprehajanjem po signalih imamo tudi pregled nad vrednostmi izrisanih signalov. Obvesti nas tudi, kadar z



```
Status: File tpehg1007.txt read.
```

Slika 2.16: Izpis statusa ob uspešnem uvozu datoteke tpehg1007.txt

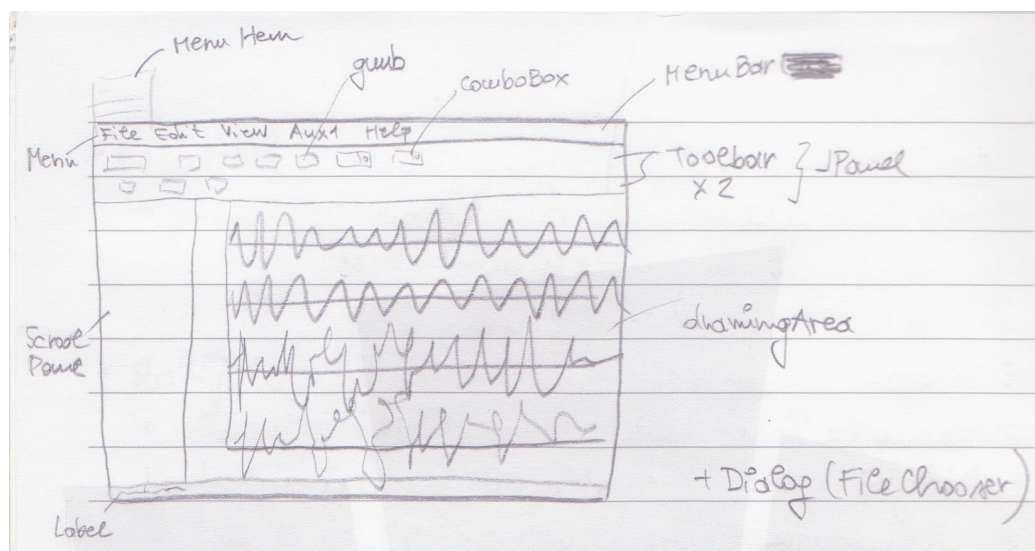
miško zaidemo s površine. Statusna vrstica se pojavlja v mnogih metodah vmesnika. Ob začetku je definirana na besedilo *Status*, ki pa se z dogajanjem na vmesniku hitro začne spreminjati. Definirana je s spremenljivko *status*, ki jo je moč videti v metodah za dodanje, brisanje in pregledovanje oznak, metodah za branje tekstovne in zaglavne datotke in v metodi premika markerja.

Poglavje 3

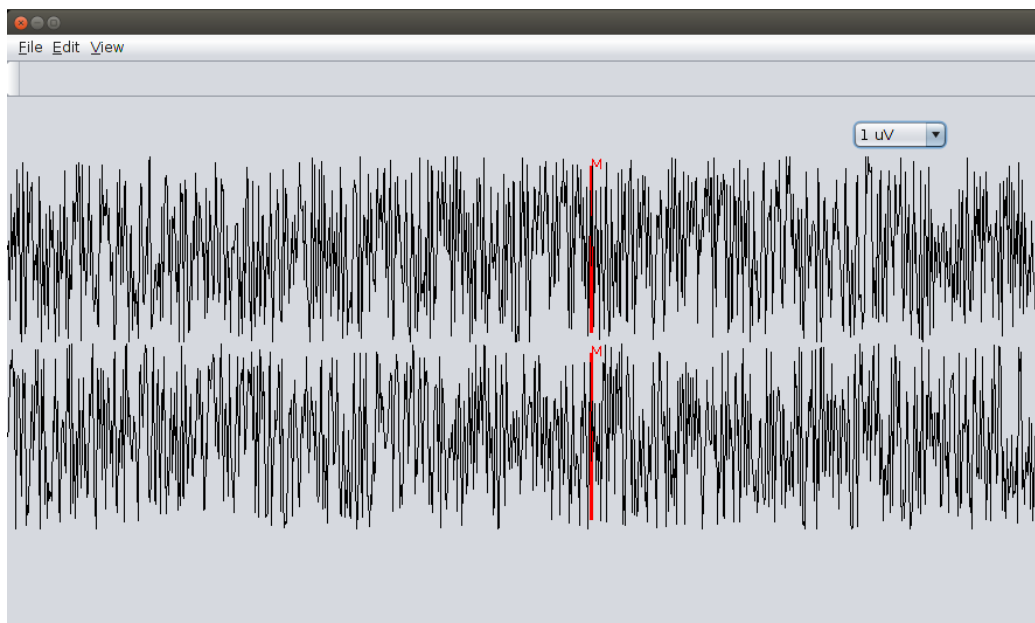
Rezultati

3.1 Grafično orodje LANNOT

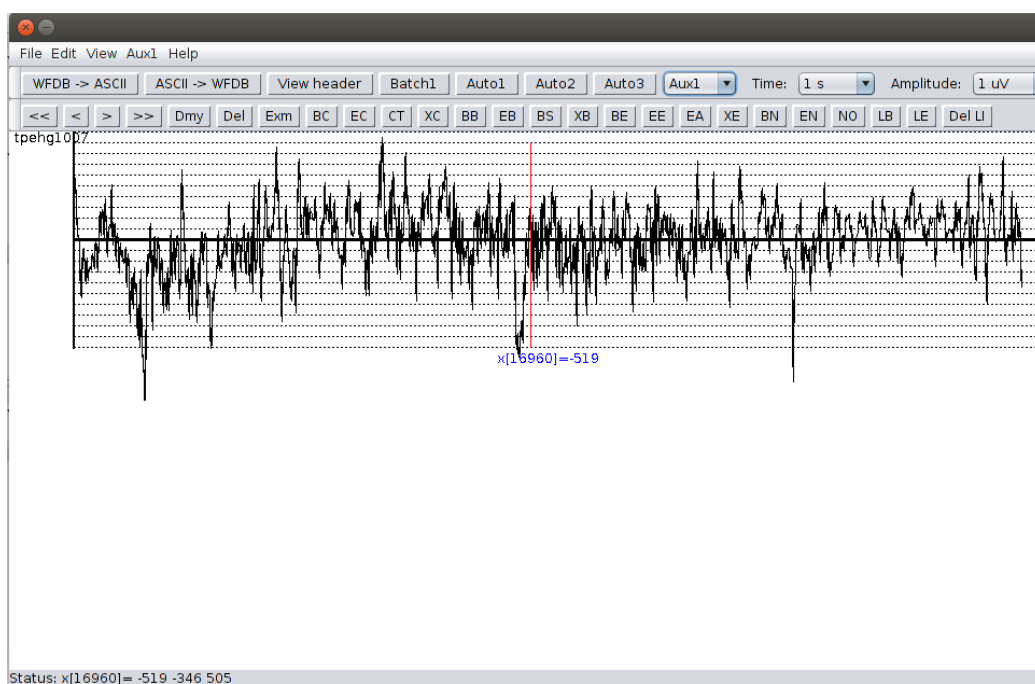
Iterativno načrtovanje orodja se je začelo s ponazoritvijo papirnatega prototipa za lažjo predstavitev (Slika 3.1), ki se je kasneje izkazalo za zelo koristno. Faze razvoja so vidne na računalniških prototipih vmesnika (Slika 3.2) (Slika 3.3).



Slika 3.1: Prvi papirnati prototip



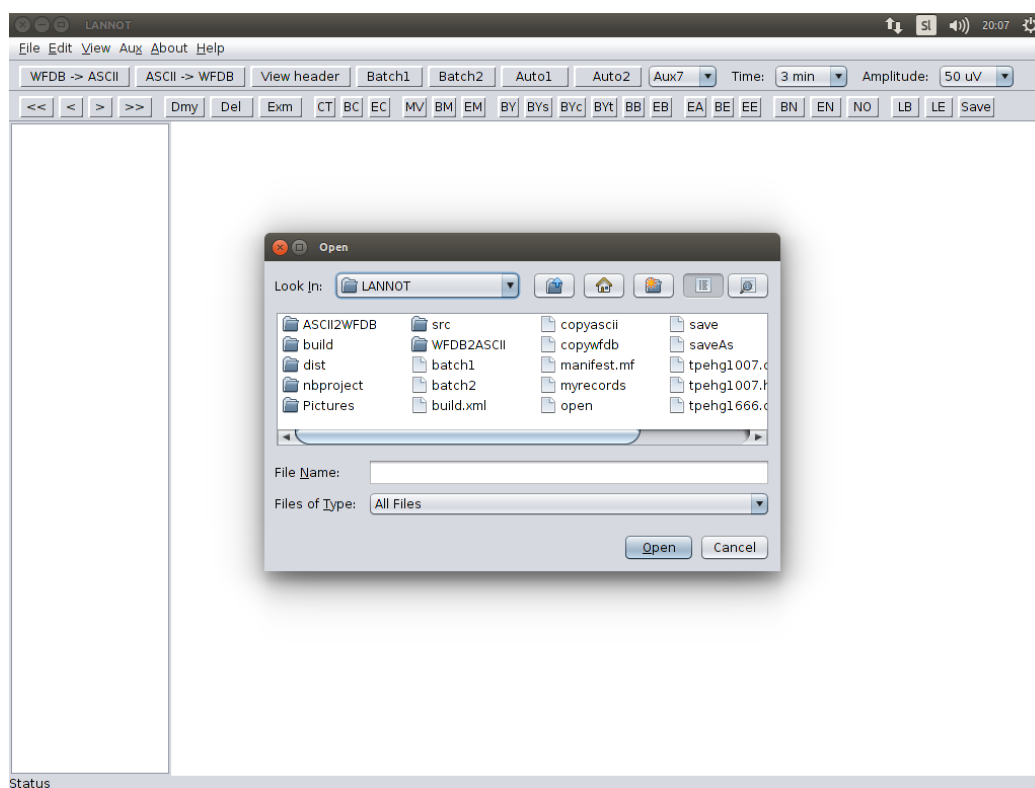
Slika 3.2: Računalniški prototip v zgodnjih fazah razvoja



Slika 3.3: Računalniški prototip v kasnejših fazah razvoja

V poglavju sledi predstavitev celotnega postopka uporabe orodja s slikovnimi ponazoritvami. Za testiranje delovanja vmesnika smo uporabljali posnetke podatkovne baze TPEHG DB. Za primer ponazoritve smo uporabili posnetek tpehg1007.

Najprej je potrebno uvoziti posnetek. Opcija New ali Open povzroči klic dialoga za izbiro datoteke (Slika 3.4). Dialog zahteva branje zaglavne dato-

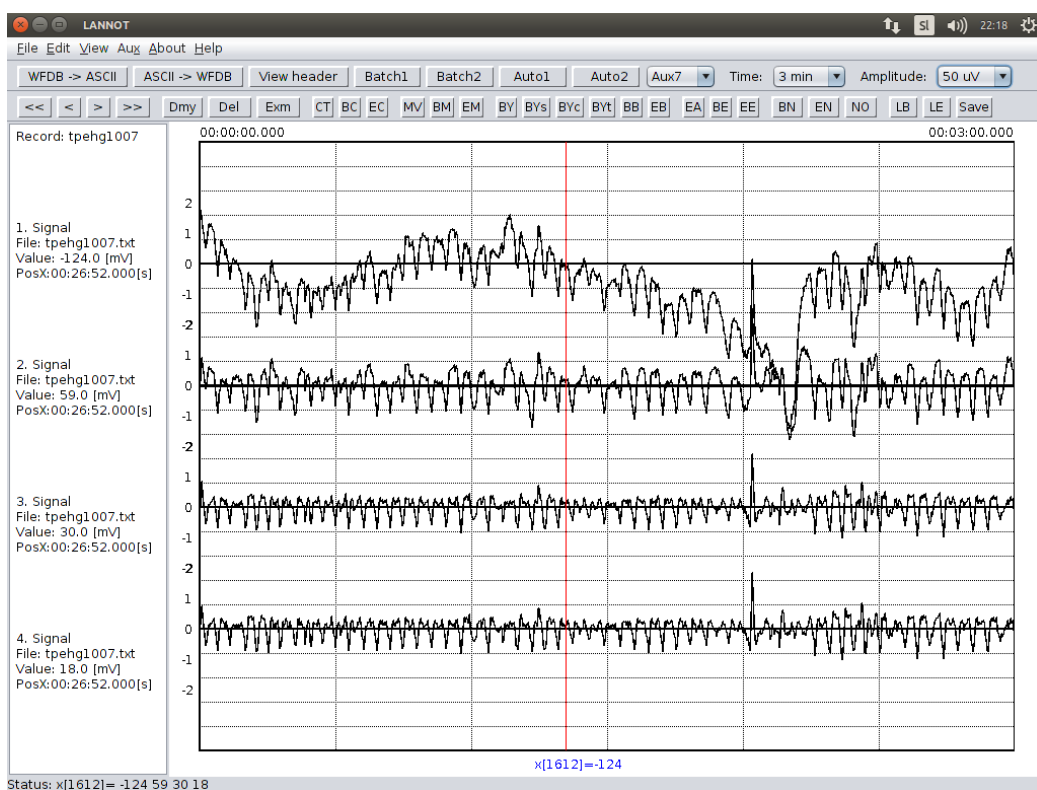


Slika 3.4: Dialog za izbiro datoteke

teke tipa .hea. V primeru izbora datoteke drugega tipa, nas vmesnik na to opozori. Poleg zaglavne datoteke se v delovnem direktoriju mora nahajati tudi podatkovna datoteka posnetka tipa .dat. Ob izbiri zaglavne datoteke, vmesnik v ozadju na osnovi podatkovne datoteke, s pomočjo ukaznega programa *rdsamp*, ustvari novo tekstovno datoteko z istim imenom kot je ime posnetka. Uporabljen ukaz:

```
rdsamp -r tpehg1007 > tpehg1007.txt
```

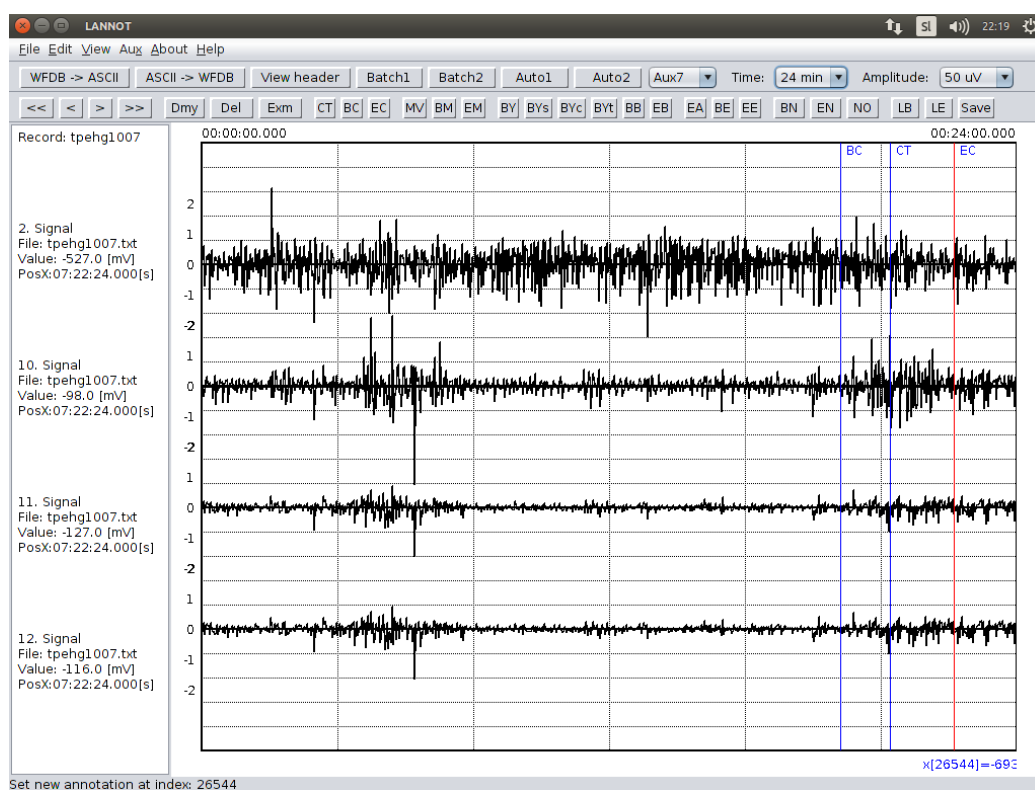
Ustvarjena tekstovna datoteka vsebuje vzorce signalov, ki jih vmesnik prebere in izriše na risalno površino. Vzorci so izrisani po metodi vzorca najbližjega piksla. Prvotni izrisani signali so prvi štirje signali iz zaglavne datoteke, ki jih kasneje lahko poljubno spreminjamo. Ob prvem kliku na risalno površino se nam prikaže marker za vodenje po signalih. Pod markerjem lahko vidimo vrednost prvega signala, informacije vseh štirih pa lahko spremljamo v drsni površini in statusni vrstici (Slika 3.5).



Slika 3.5: Prikaz markerja in izpis informacij signalov

V orodni vrstici imamo na voljo nabor dvajsetih različnih oznak. Vsako oznako lahko večkrat izberemo, vendar na različnih pozicijah. Na osnovi paradigme objekt-akcija je za dodajanje oznake potrebno ustvariti klik na poziciji risalne površine in akcijo potrditi s klikom na zeleno oznako. Oznaka

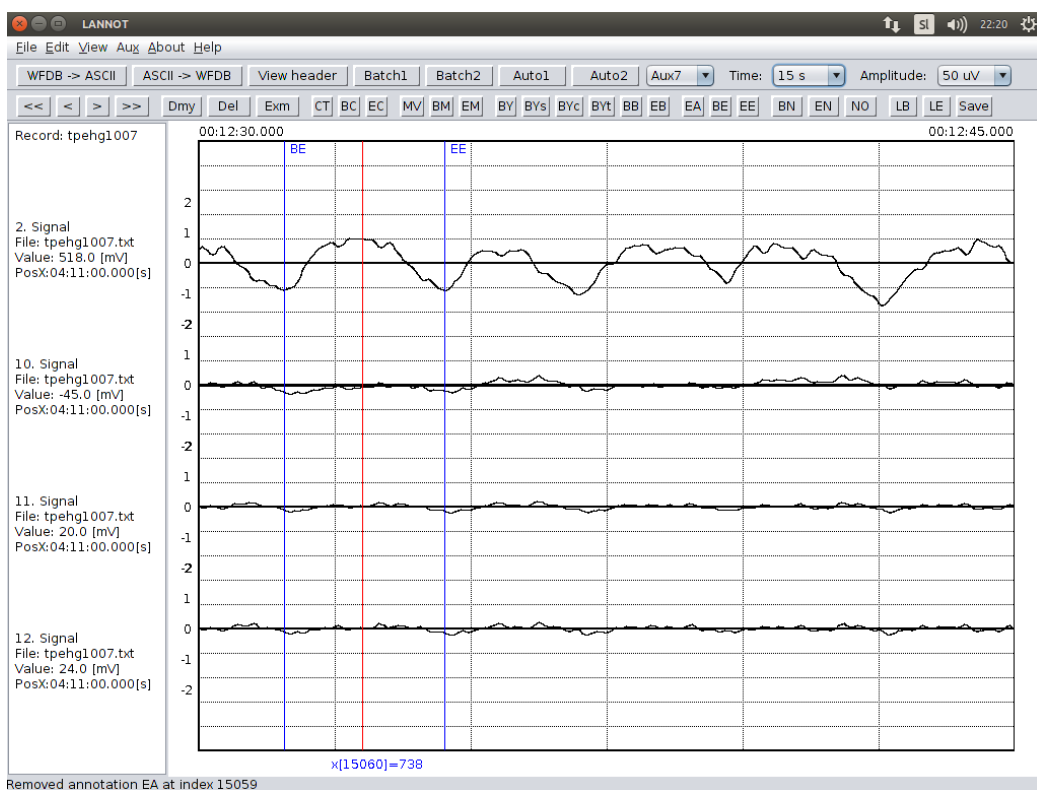
je prikazna v rdeči barvi, saj ob prvotnem izrisu še vedno leži pod markerjem. Ob premiku markerja na drug položaj je vidno obarvana modro. Kratica oznake se izpiše na vrhu izrisane črte (Slika 3.6).



Slika 3.6: Označevanje popadka v desetem signalu

Brisanje nam omogoča popolno eliminacijo oznak iz risalne površine. Za uspešno brisanje je najprej potrebna postavitvev markerja v neposredno bližino oznake in klik na nadzorni gradnik Del. Če je marker dovolj blizu območju oznake, jo uspešno izbriše, kar nam potrди tudi status v statusni vrstici. Ob sproženi akciji za brisanje, marker primerja razdaljo do oznake, ki mora biti dovolj kratka, da bi bilo brisanje uspešno. V primeru, da je v bližini markerja več oznak, se izbriše markerju najbližja (Slika 3.7).

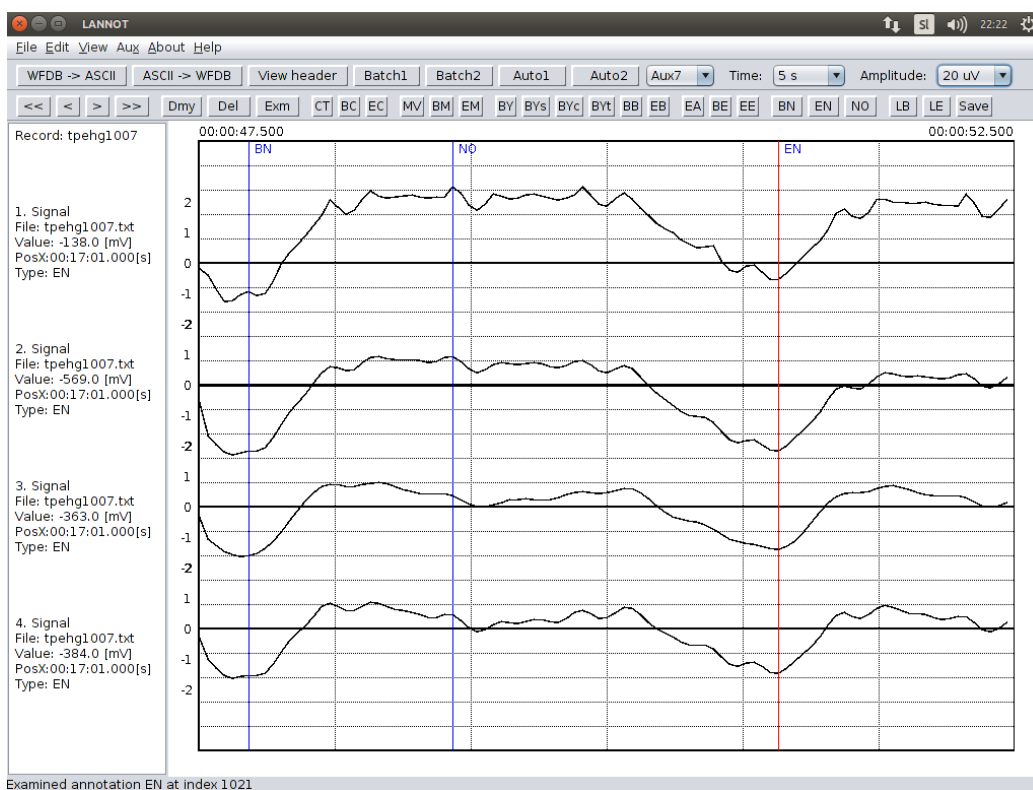
Poleg dodajanja in brisanja imamo tudi možnost pregledovanja oznak, ki deluje po enakem principu kot odstranjevanje oznak. Kadar želimo izvedeti



Slika 3.7: Brisanje oznake EA

informacije o določeni oznaki, v njeno neposredno bližino postavimo marker in ustvarimo klik na nadzorni gradnik Exm. Marker skoči na območje oznake in v drsni površini osveži urejevalnik teksta z informacijami o oznaki. Te informacije za razliko od informacij o markerju vsebujejo še tip oznake. Če je v območju markerja več oznak, urejevalnik teksta izpiše informacije o najbližji oznaki (Slika 3.8).

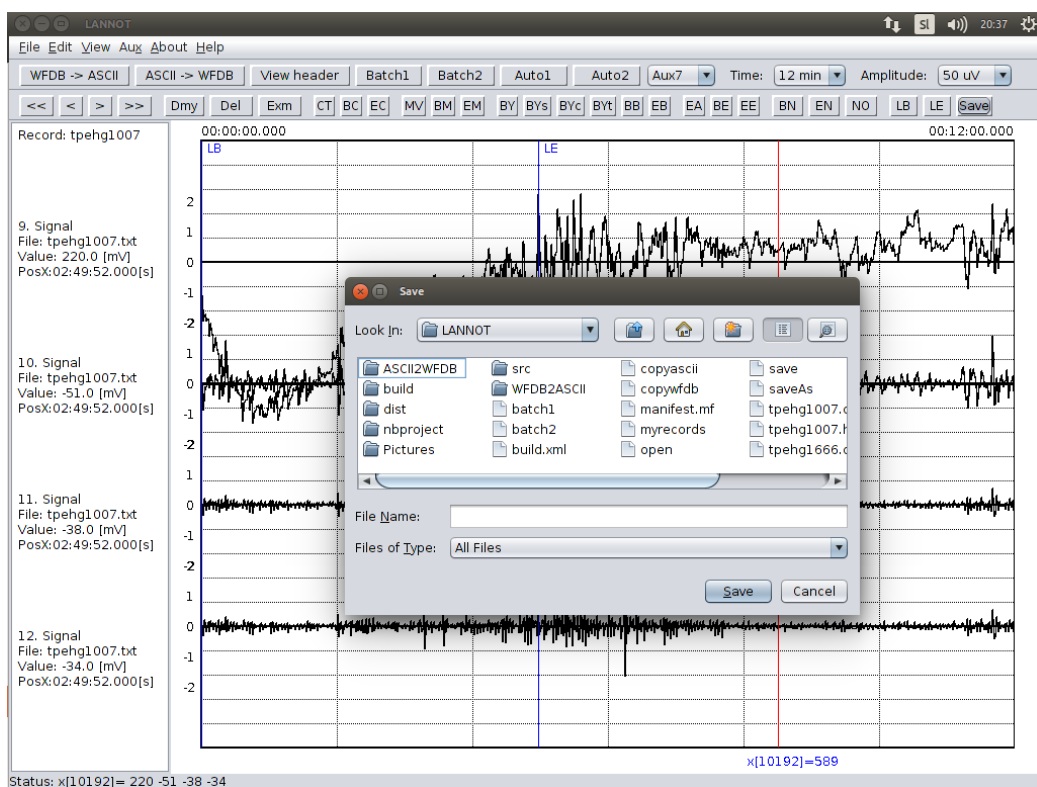
Signali so lahko medseboj prikazani zelo različno. V primeru odstopanja vzorcev signala na določenem območju, nam je omogočena funkcija za shranjevanje lokalnih intervalov. Območje intervala označimo z nadzornima gradnikoma LB in LE. In sicer LB za začetek in LE za konec intervala. Vmesnik si ob tem zapomni vzorce začetnega in končnega intervala. Ob kliku na nadzorni gradnik Save, vmesnik v tekstovni datoteki poišče vzorce signala, ki



Slika 3.8: Izpis podrobnosti oznake EN

ustrezajo vzorcem začetnega in končnega intervala ter vso vmesno vsebino vzorcev preslika v novo tekstovno datoteko, ki jo uporabnik izbere v dialogu za izbiro datoteke. V novo ustvarjeni datoteki je prikaz vzorcev v enakem zaporedju kot v tekstovni datoteki celotnega posnetka (Slika 3.9).

Ob zelenem pregledu signalov na različnih časovnih in amplitudnih skalah, ima uporabnik veliko možnosti. Prvotno risalno polje sprva prikazuje začetni interval časovne skale posnetka od nič do tri minute. Za sprehod po skalah celotnega posnetka do njegove dolžine imamo na voljo nadzorne gradnike za premikanje po ekranih. Lahko se premaknemo za pol ekrana naprej ali nazaj ali za cel ekran naprej in nazaj, dokler ne dosežemo začetek ali konec posnetka.



Slika 3.9: Dialog za shranitev lokalnega intervala v tekstovno datoteko

Vmesnik nam omogoča tudi vmesno shranjevanje oznak. Ob izbiri opcije Save menuja File, vmesnik vse postavljene oznake shrani v tekstovno datoteko, ki dobi privzeto ime posnetka in končnico `.tru`, v našem primeru datoteka `tpehg1007.tru` (Slika 3.10). Omenjena funkcionalnost je zelo uporabna ob odpiranju posnetka. V primeru da za dani posnetek že obstaja istoimenska datoteka končnice `.tru`, se nam na risalnem polju poleg izrisanih signalov pojavijo tudi oznake, ki so bile za ta posnetek ob nekem dogodku shranjene. V želji, da uporabnik želi oznake shraniti v drugo datoteko, izbere opcijo Save As.

Za uporabo nadzornega gradnika WFDB→ASCII smo v delovnem direktoriju ustvarili direktorij WFDB2ASCII, ki vsebuje datoteko `wfdb2ascii`. V omenjeni datoteki smo definirali kodo, ki izvede ukaz `rdsamp` za vse datoteke

00:00:00.000	0	LB	0	0	0
01:39:07.000	5947	LE	0	0	0
04:50:09.000	17409	BC	0	0	0
05:23:35.000	19415	CT	0	0	0
05:59:43.000	21583	EC	0	0	0

Slika 3.10: Izgled datoteke tpehg1007.tru. Za prikaz oznak se uporabita indeks vzorca in ime oznake

zahtevanega tipa. Ob uvozu posnetka se ime posnetka shrani v datoteko *myrecords*, če ga ta še ne vsebuje. V direktorij WFDB2ASCII se hkrati skopirata zaglavna in podatkovna datoteka posnetka, ki sta potrebni za uspešno izvedbo ukaza *rdsamp*. Ob kliku na nadzorni gradnik WFDB→ASCII se v direktoriju pojavijo tekstovne datoteke vseh posnetkov, ki so posledica klica ukaza *rdsamp* iz datoteke *wfdb2ascii*.

Za izvedbo nadzornega gradnika ASCII→WFDB smo definirali enak postopek, le da v tem primeru poteka obratna pretvorba. Direktorij ASCII2WFDB s pomočjo datoteke *myrecords* pridobi le tekstovne datoteke posnetkov, ki so zahtevane za uspešno izvedbo ukaza *wrsamp* datoteke *ascii2wfdb*. Ob izbiri nadzornega gradnika ASCII→WFDB se pojavijo nove zaglavne in podatkovne datoteke posnetkov.

Obe funkcionalnosti omogočata učinkovit pregled nad zaglavno, podatkovno in tekstovno datoteko določenega posnetka v skupnem direktoriju.

Poglavje 4

Razprava in zaključki

Razvili smo interaktivno grafično orodje za vizualizacijo in polno označevanje elektromiogramov maternice. Glavni namen je bila grafična predstavitev signalov in označevanje posnetkov.

Pokazali smo različne načine za delo s signali danega posnetka v smislu postavljanja, brisanja in pregledovanja oznak. Oznake bodo omogočale različne vrste avtomatske analize, kot so karakterizacija posnetkov in razvoj razpoznavalnih algoritmov.

Orodje bo omogočalo postavljanje oznak v novo nastajajoči podatkovni bazi TPEHG DB. Uporabljali ga bomo za obravnavanje bioloških signalov na osnovi programske opreme WFDB.

Ideja vmesnika je bila razviti urejevalnik, ki bi lahko bil lahka (Light) različica orodja Semia.

Vmesnik omogoča delo tudi s tekstovnimi datotekami in formate za zapisovanje podatkov WFDB, medtem ko v ozadju živi knjižnjica WFDB. Med delom imamo možnost klicev ukaznih datotek, ki uporabljajo ukaze okolja WFDB, ne da bi pri tem morali razvijati nove programe v samem okolju. Dejstvo pa je, da vmesnik ne zahteva poglobljenega znanja knjižnjice WFDB.

S pretvarjanjem formatov iz WFDB v ASCII in obratno smo zadovoljili zahtevam interaktivne pretvorbe med različnimi formati, s čimer je omogočena manipulacija s podatki.

V vmesniku imamo definirana tudi nadzorna gradnika Auto1 in Auto2, ki služita avtomatski analizi lokalnih intervalov. V vmesniku nista končno definirana, sta pa odprta za fleksibilno dodajanje rutin uporabnika za potrebe razvoja v prihodnosti.

LANNOT je fleksibilna različica grafičnega orodja Semia. Pri Semii obstoja težava s kompleksnostjo sistema, saj deluje na principu paradigme akcija–objekt. Razvito orodje predstavlja alternativo Semii z nasprotno paradigmo, s katero smo dosegli visoko uporabnost, to je učinkovitost in naučljivost.

Z razvitim urejevalnikom smo zadovoljili glavnim načelom načrtovanja uporabniških vmesnikov.

V smislu prvega principa (Zagotovi nadzor uporabnika) vmesnik omogoča direktno manipulacijo in nedvoumno in takojšnjo povratno informacijo. Tip interakcije je objekt-akcija.

V smislu drugega principa (Reduciraj obremenitev uporabnikovega spomina) zagotavlja pravilen izgled, permanentno izpostavlja gradnike glede na relativno zahtevan protokol označevanja in odpušča uporabnikove napake.

V smislu tretjega principa (Zagotovi konsistentnost) smo uporabili iste gradnike, stile, barvo in font pisave.

Orodje omogoča predstavitev in manipulacijo signalov ter interakcijo s podatki globalno in lokalno. Prav tako zagotavlja preiskovanje podatkov v vsaki točki, urejevanje oznak in brezpapirno delo. Zagotavlja lokalno in globalno pregledovanje podatkov, hiter prikaz podatkov in hitro delo.

Orodje je učinkovito, uporabniku prijazno in uporabno.

Literatura

- [1] The Term-Preterm EHG Database (TPEHG DB). Website, 2010. <http://www.physionet.org/physiobank/database/tpehgdb/>.
- [2] H Alvarez and R Caldeyro. Contractility of the human uterus recorded by new methods. *Surgery Gynecology and Obstetrics*, 91(1):1–13, 1950.
- [3] Roman Dorn and Franc Jager. Semia: semi-automatic interactive graphic editing tool to annotate ambulatory ECG records. *Computer methods and programs in biomedicine*, 75(3):235–249, 2004.
- [4] Gašper Fele-Žorž, Gorazd Kavšek, Živa Novak-Antolič, and Franc Jager. A comparison of various linear and non-linear signal processing techniques to separate uterine EMG records of term and pre-term delivery groups. *Medical & biological engineering & computing*, 46(9):911–922, 2008.
- [5] Jager Franc. Uporabniški vmesniki - predloge za predavanja. , Šolsko leto 2014/2015.
- [6] Wilbert O Galitz. *It's time to clean your windows: Designing GUIs that work*. John Wiley & Sons, Inc., 1994.
- [7] Marko Gregorc and Luka Knez. Elektromiografija. Website, 2008. http://medfiz.fmf.uni-lj.si/Leto08/Files/Sem_Elektromiografija.pdf.

- [8] Franc JAGER. Semi-automated interactive graphic tool to annotate 24-hour ambulatory ECG records. Vabljeno predavanje. *Cambridge (Massachusetts): The Harvard - MIT, Division of Health Sciences and Technology*, 10, 2003.
- [9] Znaki ki kažejo na prezgodnji porod. Website. <http://www.bambino.si/metode-za-preprecevanje-prezgodnjega-poroda.html>.
- [10] H elene Leman, Catherine Marque, and Jean Gondry. Use of the electrohysterogram signal for characterization of contractions during pregnancy. *Biomedical Engineering, IEEE Transactions on*, 46(10):1222–1229, 1999.
- [11] Theo Mandel. *The elements of user interface design*, volume 20. Wiley New York, 1997.
- [12] Catherine K Marque, J er emy Terrien, Sandy Rihana, and Guy Germain. Preterm labour detection by use of a biophysical marker: the uterine electrical activity. *BMC pregnancy and childbirth*, 7(Suppl 1):S5, 2007.
- [13] George B Moody. WFDB programmer’s guide. *Harvard-MIT Division of Health Sciences and Technology, Cambridge, MA, USA*, 10, 2006.
- [14]  ziva Novak-Antoli c, Ivan Verdenik, Vladimira Assejev, and Marija Avanzo-Velkavrh. Ocenjevanje tveganja za prezgodnji porod. *Zdrav Vestn*, 70:347–9, 2001.
- [15] Spletna stran Physionet. Website, 2015. <http://physionet.org/>.
- [16] Maartje PGC Vinken, Chiara Rabotti, Massimo Mischi, and S Guid Oei. Accuracy of frequency-related parameters of the electrohysterogram for predicting preterm delivery: a review of the literature. *Obstetrical & gynecological survey*, 64(8):529–541, 2009.
- [17] Yiyao Ye-Lin, Javier Garcia-Casado, Gema Prats-Boluda, Jos e Alberola-Rubio, and Alfredo Perales. Automatic identification of motion

artifacts in EHG recording for robust analysis of uterine contractions. *Computational and mathematical methods in medicine*, 2014, 2014.

- [18] Laboratorij za biomedicinske računalniške sisteme in oslikave. Website, 2015. <http://www.fri.uni-lj.si/si/laboratoriji/lbrso/>.
- [19] V Zahn. Uterine contractions during pregnancy. *J Perinat Med*, 12:107–113, 1984.