

TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduste instituut
Keeletehnoloogia õppetool
Informaatika eriala

Kadri Pirn

Graafiline kasutajaliides korpuse käsitsi
süntaktiliseks märgendamiseks
Bakalaureusetöö

Juhendaja : Kaili Müürisep

Autor: “.....” mai 2002

Juhendaja: “.....” mai 2002

Õppetooli juhataja: “.....” mai 2002

Sissejuhatus	3
1 Kasutajaliidese disain	5
1.1 Graafiline kasutajaliides	5
1.2 GUI disain erinevate autorite silmade läbi	6
2 Keeletehnoloogia Eestis	12
2.1 Keeletehnoloogia olemus ja eesmärgid	12
2.2 Millises seisus on hetkel Eestis keeletehnoloogia?	13
2.3 Süntaktiline märgendamine	14
2.4 Korpuse süntaktilised märgendid	17
3 GUI süntaktiliseks ühestamiseks	20
3.1 Ülesande püstitus.....	20
3.2 Realisatsioon	22
3.3 Nõuetele vastavus.....	24
3.4 Programmi installeerimine.....	26
3.5 Kasutusjuhend	27
3.6 Kasutajaliidese esimene avalik versioon	34
Kokkuvõte	35
Kirjandus	36
Abstract	37
Lisa 1. Kasutajaküsitlus	38
Lisa 2. Kasutajaliides käsitsi märgendamiseks.....	41

Sissejuhatus

Käesolev bakalaureusetöö käsitleb käsitsi süntaktilise märgendamise graafilise kasutajaliidese loomise protsessi. Süntaktiline analüüs on üks suur keeletehnoloogia valdkond, mida on vaja uurida paljude keeletehnoloogiliste rakenduste loomiseks. Käesoleva töö valmimise hetkeks on olemas süntaksianalüsaator [1], mille töö parandamiseks ning uute eesti keele kitsenduste grammatika reeglite välja töötamiseks on vaja suurendada süntaktiliselt ühestatud tekstide korpust. Seda saab teha aga vaid käsitsi. Seega on vaja mugavat vahendit, mis võimaldaks teatud formaadis teksti süntaktiliselt ühestada. Siiani on toimunud antud protsess suvalise tekstiredaktoriga, mis aga soodustas inimlike vigade tekkimist – tehti sisestamisel trükkivigu, muudeti väljundfaili formaati ning esines ka teisi vigu.

Käesoleva töö käsitleb ka graafilise kasutajaliidese disaini protsessi ning eesti keele keeletehnoloogia olemust ning arengut.

Samal teemal kaitses töö autor detsembris 2001 ka oma semestritöö [2]. Võrreldes semestritööga on arendatud kasutajaliidese võimalused, uurimus graafiliste kasutajaliidestele esitatavatest nõuetest ning juurde on kirjutatud keeletehnoloogiat puudutav peatükk.

Töö eesmärk

Töö esmaseks eesmärgiks on luua kasutajaliides, mis võimaldaks kindla formaadiga tekste süntaktiliselt ühestada. Kuna märgendajateks on tihti inimesed, kelle arvutikasutus oskus ning võimalused on erinevad, kuid märgendama peavad saama kõik ühte moodi, peab loodav kasutajaliides olema graafiline, lihtne ning see peab töötama kõigi enimlevinud graafiliste platvormide peal. Kasutajaliides ei tohi nõuda interneti püsiühendust. Märgendamine peab olema intuitiivne ning tuttavlik ka algajale arvutikasutajale ning aitama vältida inimlikke vigu.

Töö teiseks eesmärgiks on uurida, mida nõutakse üldiselt graafiliselt kasutajaliideselt. Millistele tingimustele peab vastama üks kaasaegne programm, mis võimaldab mugavalt ja intuitiivselt kasutajal rakendusega suhelda.

Ülevaade tööst

Töö esimene peatükk käsitleb graafilise kasutajaliidese disaini. Peatüki esimene punkt tutvustab üldiselt kasutajaliidest ning ka graafilist kasutajaliidest eraldi. Teises punktis uuritakse viie autori töid antud valdkonnast ning tehakse järeldusi, millistele tingimustele peab kaasaegne mugav graafiline kasutajaliides vastama.

Töö teine peatükk on keeletehnoloogiat puudutav. Peatüki esimene punkt kirjeldab keeletehnoloogia olemust ning eesmäärke, teine punkt tutvustab hetke keeletehnoloogia arengu olukorda Eestis. Kolmandas punktis uuritakse lähemalt süntaktilist analüüsi. Viimane neljas punkt sisaldab endas ülevaadet eesti keele kitsenduste grammatika (edaspidi ESTKG) süntaktilistest märgenditest.

Töö kolmas peatükk tutvustab töö peamist eesmärki – süntaktilise ühestamise kasutajaliidest. Välja on toodud ülesande püstitus, programmi realisatsioon, installeerimis- ning kasutusjuhend. Erilist tähelepanu on pööratud graafilise kasutajaliidesele esitatud nõuetele vastavuse kirjeldamisel. Tutvustatakse ka kasutajaliidese esimese versiooni kasutusele võtmist ning selle vastukaja.

Töö juurde kuulub ka kaks lisa – esimene nendest on kasutajaküsitluse küsimustik ning teine, mis asetseb kaasasoleva CD plaadi peal, töö käigus valminud kasutajaliidese programm. Teise lisa juurde kuuluvad ka näidistekstide failid, elektroonne kasutusjuhend, installeerimisjuhend ning teised abistavad failid.

1 Kasutajaliidese disain

Käesolevas peatükis tutvustatakse graafilise kasutajaliidese disaini põhimõtteid.

1.1 Graafiline kasutajaliides

Tarkvararakenduse kasutajaliides on meetodite kogu, mis võimaldab vahendada kasutaja ja rakenduse vahelist informatsiooni [3]. Selleks kasutatakse kaasajal kuvarit, kõlareid, printerit ja muid väljundseadmeid ning klaviatuuri, hiirt, mikrofoni jt sisendseadmeid.

Graafiliseks kasutajaliideseks nimetatakse kasutajaliidest, mis esitab informatsiooni graafiliselt, kasutades selleks tüüpiliselt aknaid, nuppe ja ikoone, mille suurus on muudetav ja mis on liigutatav, vastandina tekstipõhisele kasutajaliidesele, kus informatsioon on esitatud tekstipõhises ekraanis ja käske sisestab klaviatuurilt kasutaja.

Graafiline kasutajaliides (edaspidi GUI) on tänapäeval enim levinud kasutajaliidese liik. Graafilist ja mittegraafilist kasutajaliidest eristab eelkõige väljanägemine, kiirus ning “hiire” kasutamise võimalus. Ühesuguse jõudlusega ning funktsionaalsusega arvutite korral on tekstipõhine kasutajaliides kiirem, kuid tänapäevase arvutusjõudluse juures ei suuda enamasti inimene tajuda nende kiiruste erinevust. Samas kergendab graafiline kasutajaliides tööd programmiga, seda on algajal arvutikasutajal enamasti lihtsam õppida, selle töö on tavainimesele intuiitselt arusaadavam.

Tihti eeldatakse, et graafilise kasutajaliidese rakendused on kergemini kasutatavad, aga see ei pruugi olla tõsi, kui GUI pole hästi disainitud või ei sobi olemasoleva riistvaraga [4].

Selleks aga, et GUI head omadused ebamugavuseks ei muutuks, peab riistvara seda toetama – võib ju suure resolutsiooniga värvilisel kuvaril väga esteetiline ja efektiivne kasutajaliides olla täiesti mõistetamatu pisikesel must-valgel ekraanil. Samuti peab graafilist kasutajaliidest disainides oluliselt läbi mõtlema menüüde hierarhia ning ikoonpiltide mõistetavuse. GUI üks väga suuri eeliseid on ka võimalus erinevate värvidega rõhutada erinevaid objekte. Ka seda tuleb aga tagasihoidlikult kasutada, sest inimese nägemismeelel on piirid ning väga kirevaid pilte on raske jälgida. Seetõttu võib muidu väga efektiivse kasutajaliidese kasutus jääda väikeseks.

Kuna erinevad inimesed näevad värve erinevalt, peab kasutajal jääma võimalus muuta värve.

Kasutajaliidese loomise juures on vaja väga tihedat koostööd teha tulevaste kasutajatega. Tarkvara loojal tuleb luua prototüüpe nii paberil kui elektroonselt. Igal arendusetapil tuleb lõpptulemus kooskõlastada kasutajatega. Samuti tuleb arvestada nii algajate kui ka tavakasutajate vajadustega.

Graafiliste kasutajaliideste loomise juures ei tasu siiski unustada ka otse klaviatuurilt sisestatavaid käske (klahvikombinatsioone). Igapäevastel kasutajatel on süsteemiga juba tuttav olles lihtsam sisestada teadaolevat klahvikombinatsiooni, kui liigutada käsi hiire juurde, otsida sobivat menüüpunkti ning seejärel sooritada soovitud tegevust.

Kui kasutajaliides on juba valmis ning testkasutamises, tuleks kasutajatelt saada võimalikult palju tagasisidet. Näiteks anda neile võimalus saata e-maile oma probleemide ja ettepanekutega, salvestada veateateid või jälgida nende tööd uue tarkvaraga. Üks võimalus on ka testperioodi lõpul läbi viia küsitlus, kus tuleb hinnata erinevaid tarkvara aspekte näiteks viie palli süsteemis. See on kasulik ka tarkvara tellinud organisatsiooni juhtkonnale, kes saavad nii teada, kui õige otsuse nad tegid soetades just selle tarkvara. Ainult tagasisidega saavad tarkvara loojad adekvaatselt öelda, et tarkvaraga on lihtne harjuda, ei nõua kasutajatelt liigset koormust kätele, peale ega silmadele.

1.2 GUI disain erinevate autorite silmade läbi

Käesolevas punktis tutvustatakse lühidalt viie autori, Chris Brown'i ([3]), Jonathan Hodgson'i ([5]), James Hobart'i ([6]), Ian Sommerville'i ([7]) ning Laura Arlov'i ([8]) tähelepanekuid GUI loomisest.

1.2.1 Kasutajaliidese tunnetuslikud küljed

Üheks olulisimaks kasutajaliidese omaduseks on selle intuiitiivsus [5]. Kasutajaliidese funktsioonid peavad olema tunnetuslikult arusaadavad. Selleks peab aga esmalt analüüsima, kes on süsteemi tulevased kasutajad, millised on nende oskused, arusaamad ja tõekspidamised. Erinevate kultuuriliste taustadega inimesed mõistavad erinevaid värve ja motiive tihti erinevalt ning seega võib kasutajatel tekkida raskusi funktsioonide mõistmisega.

Kasutajate liigitamisel tuleb arvestada kasutajate füsioloogiliste, sensorsete, tunnetuslike ja kultuuriliste omadustega.

Intuiitsuse olulisust on rõhutanud ka teised autorid. Ian Sommerville'i põhiteesid kasutajaliidese disainil on [7]:

- Süsteem peab jääma analoogseks reaalse eluga;
- Kasutajaliidese kujundust, vormi jms ei tohi kasutusse võtta ainult sellepärast, et selle tegemine on lihtne;
- Käsud ja objektid peavad olema tuttavad;
- Õppimisaeg peab olema lühike;

GUI peab disainima nii, et võimalikult väheste füüsiliste liigutustega (kaasa arvatud silma liigutamine) oleks kõike vajalikku võimalik teha [8].

1.2.2 Menüü

GUI üheks kindlaks osaks on menüüd. Jonathan Hodgson ([5]) on välja toonud järgmised menüüliigid: lihtne, lineaarne, puu struktuuriga, atsükliline ja tsükliline. Mida liik edasi, seda raskem on kasutajal intuiitselt leida õiget menüüpunkti. Puukujulise menüü disainimisel on vaja hoolikalt läbi mõelda, kuidas on õige grupeerida erinevaid funktsioone. Liigne või vale grupeerimine võib teha süsteemi kasutamise ülikeeruliseks ning viia kasutajate programmist ja tootjafirma teenustest loobumiseni. Samas peab arvestama ka ekraani piiratud suurusega ning alati pole võimalik ega otstarbekas teha lihtsaid või lineaarseid menüüsid. Menüü disainimisel tuleb kindlasti hoolikalt kavandada ka menüüpunktide nimetused. Kasutajatele peab olema kergesti arusaadav, mis on ühe või teise menüüpunkti funktsionaalsus. Vastasel korral võivad kasutajad segadusse sattuda ning süsteemi valesti kasutada. Kindlasti tuleks jälgida, et erinevates süsteemi alamprogrammides oleksid ühtedel funktsioonidel samad menüünimed ning need asuksid menüüdes analoogsetes kohtades.

Inimese silm ei suuda korraga haarata väga suurt ala. Seega peab ka GUI disainimisel arvestama, et menüüribal on omad piirid ning kui valikuid tekib liiga palju, on kasutajal väga raske õigeid menüüpunkte või valikuid leida. Näiteks menüüribas soovib James Hobart [6] näidata maksimaalselt kümmet võimalust, ühes menüüs aga võib olla kuni 12 võimalust. "*Check Box*"-e (ehk märkeruute) võib ühes grupeeringus olla kuni 12 jne.

Samuti tuleb menüüide kavandamisel arvestada väljakujunenud standardeid. Näiteks paiknevad funktsioonid “Salvesta”, “Ava”, “Välju” ja muu selline menüüpunkti “Fail” all [3].

1.2.3 Ekraani kujundus

Graafilise kasutajaliidese juures peab kindlasti silmas pidama, et ekraan oleks lihtsalt silmadega haaratav [5]. Kui kasutada ikoone, peavad need olema intuitiivselt arusaadavad, üksteisest selgelt eristatavad ning mitte liiga segavad. Ikoone ei tohi olla liiga palju ning analoogsetel funktsioonidel peavad olema sarnased ikoonid. Kui kasutaja vajutab ikoonile, peab see olema selgelt arusaadav. Kasulikud on ka vihjed, mis tekivad ekraanile, kui hiirega ikoonile osutatakse.

Samas ei tohi kalduda teise äärmusesse. Suur viga, mida tehakse, on ka kõikide funktsioonide kutsumise võimalus esipaneelilt. Tegelikult tuleks harvem kasutatavad ning rohkem andmeid nõudvad funktsioonid peita paneelide taha [6].

Ikoonidel kasutatavad pildid peavad olema koheselt arusaadavad. Need ei tohi olla liiga keerulised, pigem lakoonilised. Palju aitavad üldtuntud logod. Paraku pole enamasti ikoonide valimine nii lihtne ning vähegi erandlike funktsioonide ikoonide kujundamine on aega ning tugevat läbimõtlemist nõudev ülesanne.

Pidevalt muutuva info jaoks on kasulik kasutada ka graafilisi pilte või jooniseid (näiteks nn olekuriba ehk *statusbar*, mis näitab mingi protsessi valmimise järku) [7].

Ka ekraani värvide kasutusele peab suurt rõhku asetama. Ideaalne kasutajaliides on selge, lihtne ja maitsekas. Sellest on võimalik aru saada esimesel silmapilgul. Kasutaja saab aru, kust peab ta infot lugema, kuhu sisestama, kus parandama ja milliseid välju saab vajutada, milliseid tõmmata. Aken annab ise edasi oma mõtte [8].

1.2.4 Tekstiline info

Dialogid, mille abil programm saab kasutajalt infot, peavad olema selged ning üheselt mõistetavad. Täpselt tuleb läbi mõelda vigade töötlus ning veateated. Valed või segased teated võivad viia programmi valekasutuseseni või kutsuda esile kasutajate õigustatud pahameele.

Veateated peavad olema kontekstipõhised ning selgitama täpsemalt, mida kasutaja valesti tegi ning soovitama edaspidiseid samme. Samuti peab veateatest olema võimalik saada ka täpsemat abiinfot [7].

Oluline on ka terminite kasutus, mis peaks olema järjepidev. Kui programmi ühes osas mainitakse “Ese”, teises “Toode” ning kolmandas midagi muud, kuid need kõik tähendavad ühte mõistet, satub kasutaja kindlasti segadusse. Sellise olukorra vältimiseks soovib James Hobart koostada reserveeritud sõnade nimekirja ning kontrollida, et need oleksid intuiitselt üheselt arusaadavad ning ei esineks sünonüüme. On olemas ka üldtunnustatud reserveeritud sõnad nagu inglise keeles “OK”, “Cancel”, “Close”, “Exit” jne [6].

Kuna igal GUI disaineril ning ka programmeerijal on oma sõnavara, kuid eelneva põhjal selgus, et ühe tootja tooted võiksid olla üheselt mõistetavad, tuleb disaineritele ning programmeerijatele selgeks teha antud tootja sõnavara ning jälgida, et seda kõikjal ka kasutataks. Seega tuleb firmasiseselt enne vastava sõnavara leidmiseks tööd teha ning ilmselt ka spetsialistidelt abi küsida. Loomulikult tuleb jälgida ka juba kasutusel olevaid kasutajaliideste sõnavara ning sealt võimaluse korral eeskuju võtta – nii kujunevad standardid ning kasutajatel on lihtsam harjuda erinevate keskkondadega [6].

Abiinfo saamise võimalus on ülimalt oluline ning selle saamiseks ei pea kasutaja väga suuri pingutusi tegema. Abiinfo peab olema struktureeritud – ühe probleemi selgituse juures peavad olema viited ka teiste analoogsete või antud probleemiga seotud funktsioonide abiinfole [7].

1.2.5 Kasutajaliidese reageerimisaeg

Jonathan Hodgson leiab ([5]), et programmid, mille reageeringu aeg on üle 2 sekundi, on kasutajate jaoks liiga aeglased. Kui pole võimalik, et programm väljastab tulemuse kiiremini, peab GUI kuidagi näitama kasutajale, et töö käib. Näiteks kuvab kasutajale liivakella.

Samas, ainult liivakella näitamine või teade “Oodake veidi...” ei aita juhtudel, kui töö kestab üle 7-10 sekundi. Kasutaja soovib teada, kui kaua ta veel ootama peab. Selleks võib lisada näiteks ekraanile teate, mitu protsenti tööst on tehtud.

1.2.6 Suhtlus tulevase kasutajaga

Enamik kasutajaliidese disaini õpikuid peavad väga oluliseks suhtlust tulevaste kasutajatega ning prototüüpide kasutajatega testimist. Analüüsi etapis tuleb teha kindlasti tulevaste kasutajatega intervjuusid ning nende tulemusi ka kasutajatele endile näidata. Kasutajatele tuleb anda võimalus valida mitme erineva prototüübi

vahel. Neile tuleb anda aega kohanemiseks ning mõtlemiseks, mis võiks teisiti olla. Kasutajatele peab olema juba eelnevalt selga, milline programm hakkab välja nägema, milline on selle funktsionaalsus. Kindlasti tuleb nendega “läbi mängida” kogu töö protsess prototüüpidega. Enamasti pole ju kasutajatel kogemusi, et mõista, mida nad tulevase programmiga teha saavad ning mida mitte. Samas läbi käies kogu programmi kasutajale nähtava poole, võib kasutajale meenuda, et mõni funktsioon peab käituma teisiti või mõnda vajalikku tegevust ei saa üldse teostada. Kui programm juba reaalselt kasutuses, on muudatusi märksa raskem teha [5].

James Hobart rõhutab, et kõige suurem viga, mida GUI disainerid teevad, on kasutajate unustamine. Arvatakse, et teatakse täpselt, mida kasutaja soovib, kuid tihti see nii pole. Kasutajatele tuleb anda kontroll programmi üle, vastasel korral tundub kasutajatele, et neid kontrollitakse. Disaineritel võivad olla arvutitelt hoopis teised ootused ning ka teise harjumused võrreldes loodava GUI kasutajatega.

Ian Sommerville [7] on pakkunud välja ka võimaluse testperioodil kasutajate jälgimise – seda näiteks videokaamerate abil. Videokaamerad jälgivad iga kasutajaliigutust (ka silmade liikumist) erinevatest külgedest. Selle analüüsimine on küll väga aeganõudev, kuid tulemuseks on kasutajate töö optimeerimine. Samuti on kasulik salvestada testperioodidel kasutajatel tekkinud veateateid – nii leiab üles tüüpvead ning saab analüüsida, kas nende vältimiseks ei saaks midagi kasutajaliideses muuta.

1.2.7 Järjepidevus

Hea oleks jälgida ka, et ühe tootja erinevatel produktidel oleks ühesugune stiil. Nii teavad kasutajad juba ette, milline on ühe või teise tootja GUI. Nii ei pea kasutajad iga uuenduse korral kõike uuesti õppima, vaid neile on üldine keskkond juba tuttav ning harjumine võtab vähem aega [6]. Samuti on oluline terminite järjepidevus [6]. Klahvikombinatsioonid või ühesugused menüüpunktid ei tohi ühes alamprogrammis tähendada ühte ja teises teist.

1.2.8 Duaalsus

Andmete sisestamisel võiks kasutajal olla võimalus valida vahendeid, kuidas programmis edasi liikuda või erinevaid funktsioone välja kutsuda. Selleks sobib nii hiire abil osutamine kui ka klahvikombinatsioonid.

Sagedasti kasutatavaid funktsioone on mugav kutsuda välja klaviatuuri abil, eriti kui eelnevalt on kasutaja tegelenud klaviatuurilt informatsiooni sisestamisega. Samas on kasutajaid, kes oskavad ja tahavad kasutada vaid hiirt. Seega peab jälgima, et kõik funktsioonid oleksid kätte saadavad vähemalt hiirega. Võib olla funktsioone, mida funktsiooniklahvidega valida ei saa, kuid hiirega peab olema võimalik kõike teha. Funktsiooniklahvide valikul peab jälgima samuti üldtunnustatud lühendeid – nii jäävad need kasutajatele paremini meelde ning nendest on rohkem kasu. Loomulikult oleks kasulik, kui ka kogu tööd saaks teha ainult klaviatuuri abil [6].

1.2.9 Vigade vältimine

Kasutajaliidese kavandamisel tuleb teha põhjalik võimalike inimlike vigade analüüs. Programmis peab toimima elementaarne inimlike vigade ennetamine – enne jäädavate muudatuste tegemist küsib tarkvara kinnitust [7]. Samuti peab kasutajaliides ennetama kasutaja sisestusvigu, seda näiteks salvestades pakkuma failile nime või lahkudes hoiatama kasutajat muudatuste mitte salvestamisest. Sellised teated peavad olema intuiitselt arusaadavad ning kasutajal ei tohi tekkida küsimust, mida tuleb edasi teha [8].

Näide: Kasutaja soovib programmist lahkuda, kuid pole oma viimaseid muudatusi salvestanud. Kui talle esitatakse küsimus: “Dokumenti on tehtud muudatusi. Kas jätkata ilma salvestamata?”, ning antakse vastusevariandid “Katkesta” ja “OK”, siis peab enne kasutaja juurdlema, kumba varianti valida. Samas kui esitatakse küsimus “Dokumenti on tehtud muudatusi” ning vastusevariandid “Salvestada” ja “Jätta salvestamata”, on kasutajal palju lihtsam mõista antud situatsiooni.

2 Keeletehnoloogia Eestis

Käesolev peatükk tutvustab lühidalt keeletehnoloogia hetkeseisu. ning kirjeldab täpsemalt süntaktilise märgendamise olemust.

2.1 Keeletehnoloogia olemus ja eesmärgid

Keeletehnoloogia on infotehnoloogia osa, mis tegeleb loomuliku keele töötlustega. Keeletehnoloogia haarab nii kirjutatud kui ka suulise keele töötlust infotehnoloogilises keskkonnas. Infotehnoloogia areng on kaasajal ülimalt kiire ning üha enam inimesi tegeleb informatsiooni tootmise ja haldamisega. Oleks loomulik, et igatüks, sõltumata tema emakeelest, omaks võrdseid tingimusi osa saamiseks infoühiskonna võimalustest.

Keeletehnoloogia on keelelaste teadmiste rakendamine paremate arvutisüsteemide loomiseks:

- *inimese ja arvuti vahelise suhtluse parandamiseks,*
- *informatsiooni paremaks esitamiseks, kasutamiseks, otsimiseks ja analüüsimiseks,*
- *inimkeele paremaks mõistmiseks ja töötlemiseks.*

[9].

Keeletehnoloogia rakendusi on palju. Siia alla kuuluvad poolitus, õigekirja kontroll, stiilkorrektor jne. Dokumenditöötlusteks loetakse info-otsimist, dokumendi keele tuvastamist, dokumendi liigitamist, automaatset kokkuvõtte tegemist, hüperteksti ja viitade automaatset genereerimist, terminikogude loomist ning terminoloogide abivahendeid. Info-otsimist on kõige enam arendatud inglise keelt kõnelevates maades. Rikka morfoloogiaga keeled nagu eesti, soome, türgi jms. peavad aga info-otsimist põhjalikult muutma võrreldes inglise keelega, kuna nendes keeltes esinevad käänded, pöörded, liitsõnad ning sõnatuletused. Selleks kasutatakse morfoloogilist analüüsi (leitakse sõna algvorm(id) ning liitsõnade osasõnad ning otsitakse nende abil). Keeletehnoloogia rakenduse valdkonda kuulub ka ükskeelsete ja mitmekeelsete sõnastike kasutamine päringute tegemisel. Seejuures ei pea kasutaja kõigepealt leidma sõna algvormi, sest seda on võimalik automaatselt leida.

Tõlkijate abivahendeid on samuti mitmeid liike. Näiteks terminoloogia haldamise vahendid. Masintõlge on esialgu vägagi keeruline ülesanne, kuna parimal juhul

suudetakse automaatselt tõlkida ühest keelest teise sõna-sõnalt, kuid enamasti kannatab selle tõttu teksti kontekst või stiil. Keeletehnoloogiat oleks võimalik kasutada ka keeleõppes. Nimelt häälduse õppimisel, kasutades selleks kõnesignaali analüüsisivahendeid.

Dialogsüsteemid, mis kasutavad kõnetuvastust, on tänapäeval juba reaalses kasutuses kindlalt piiritletud valdkondades, näiteks infootsing telefonikataloogist, sõiduplaanidest ning piletite reserveerimisest. Need vahendid on kasutusel näiteks Inglismaal, Prantsusmaal jne.

Keeletehnoloogia suur kasutajate ring on mitmesuguste puuetega inimesed, peamiselt pimedad ja kurdid. Kasutusel on kõnesünteesi programmid, mis võimaldavad ette lugeda arvutis leiduvad tekstikujulist informatsiooni. Kõnesüntesaator ja skänner koos optilise tekstituvastuse programmiga moodustavad lugemismasina, mille abil saab ette lugeda paberkandja trükitud teksti. Kõnesüntesaator on vajalik ka kõnepuuetega inimeste suhtlemisel tavainimesega. Kurdid saavad keeltehnoloogia rakenduste abil õppida paremini rääkima – arvuti analüüsib kurdi poolt sisse loetud teksti ning näitab, mis moodi võiks teisiti oma häält ja suud sättida, et hääldus oleks korrektsem.

Keeletehnoloogial on kindlasti veel palju rakendusi. Neid tekib kogu aeg juurde ning kuna keel on arenev, siis ei saa keeltehnoloogia kunagi valmis ning sellega peab pidevalt tegelema.

2.2 Millises seisus on hetkel Eestis keeltehnoloogia?

Võrreldes enamike keeltega on eesti keele kõnelejate ring väga väike. Tänu sellele puudub tarkvara tootjatel majanduslik huvi finantseerida keeltehnoloogiat ning luua eesti keelt toetavaid rakendusi. Samas on eesti keele säilimiseks vajalik eesti keelsete programmide olemasolu. Seega on vajalik riigi toetus keeltehnoloogia arenguks. Tänu erafirmade investeerimise puudumisele pole realiseerunud paljud võimalikest rakendustest [9].

Eestis on kolm suuremat keeltehnoloogia uurimiskeskust: Tartu Ülikool, Eesti Keele Instituut ja TTÜ Küberneetika Instituut. Lisaks tegelevad arendustööga ka mõned erafirmad nagu näiteks FiloSoft, Nektom ja Festart.

Keeletehnoloogiat kasutatakse Eestis praegu äärmiselt vähe, sest kasutajaid pole teavitatud selle võimalustest. Kõige aktiivsem kasutaja on Riigikantselei, kes kasutab algvormide leidjat valitsuse otsuste ja määruste täistekst-andmebaasis. Laiatarbe-

programmidest kasutatakse keeletehnoloogiat kirjutaja abivahendites: speller, poolitaja, tesaaurus. Kasutatakse ka teksti sisselugemist skänneriga ning kõige laiemas kasutusringis on kindlasti elektroonilised sõnastikud.

Eesti keeletehnoloogia peamiseks uurimissuundadeks on eesti keele morfoloogia arvutusmudelid, eesti keele süntaktilised arvutusmudelid, eesti keele semantika formaliseerimine ning eesti keelse dialoogi modelleerimine [10].

Reaalselt on valmis inglise-eesti sõnastik, teksti lausestaja, morfoloogiaanalüsaator ESTMORF (leiab tekstis esinevate sõnavormide võimaliku(d) tüve(d) ning määrab, millises käändes ja pöördes sõna on ning võimaldab ka genereerida algvormis sõnale soovitud vorme; EKI morfoloogiamoodulid (Windows dll-id silbituseks, morfoloogiliseks analüüsiks ja sünteesiks, tüvemoodustuseks ja tüübituvastuseks); ESTMORFi tulemuste ühestaja; nimisõnafraaside filtreerija, mis võimaldab nimisõnafraase leida jooksvast tekstist (tekstist terminite ja tervikmõistete eraldamiseks); kitsenduste grammatikal põhinevad morfoloogiline ühestaja [11] ja süntaksianalüsaator [1] ning eesti keele kõnesüntesaator. Täpsemalt võib antud rakenduste kohta leida infot <http://www.eki.ee/keetechnologia/tarkvara.html>.

Lisaks on valminud ka morfoloogilise ühestaja ja süntaksianalüsaatori rakendusena sisukokkuvõtte genereerija ning eesti keele semantiline ühestaja.

Keeletehnoloogia edasiarenduseks on vaja enne kõike teadvustada kasutajaid, mida võimaldab keeletehnoloogia, kui palju muudab see kasutajate tegevust lihtsamaks.

Samuti on vaja luua juurde nii morfoloogilisi kui ka süntaktiliselt märgendatud korpusid, paralleelkorpusi ning ka eri allkeelte korpusi. Puudu on ka kõnekorpusi.

2.3 Süntaktiline märgendamine

Süntaktiliseks analüüsiks nimetatakse suvalise pikkuse ja keerukusega sisendteksti morfoloogilise ja süntaktilise kirjelduse automaatset kindlaksmääramist, kuid mitte semantilist interpretatsiooni. Süntaktiline analüüs koosneb kolmest etapist: morfoloogilisest ühestamisest, osalausepiiride määramisest ja sõnade süntaktiliste funktsioonide määramisest. Süntaktiliseks märgendamiseks nimetatakse iga sõnavormi juurde õige süntaktilise märgendi märkimist [12].

Süntaktilist analüüsi kasutatakse mitmetes keeletehnoloogia rakendustes. Hetkel on peamine rõhk asetatud nimisõnafraaside leidjal ning dokumentide automaatse sisukokkuvõtte tegemisel. Samas läheb süntaktilist analüüsi tarvis pea igas rakenduses alates kirjutaja abivahenditest ja lõpetades tekst-kõne sünteesiga.

Valminud on eesti keele süntaksianalüsaator ([1]), mille töö korrektsus on üle 97 % ja täpsus ligikaudu 85%. Eesti keele süntaksianalüsaator põhineb kitsenduste grammatika formalismil.

Loomuliku keele automaatne süntaktiline analüüs koosneb kuuest etapist [1]:

1. Eeltöötlus ja lausestamine, mille käigus viiakse tekstid ühtsele kujule ja märgendatakse lausete algused ja lõpud.
2. Morfoloogiline analüüs, mille käigus leitakse sõnavormi tüvi ning lõpud ja neile vastav sõnaliik, kääne või pööre. Kui sõnavorm on mitmeti tõlgendatav, antakse selle kõik tõlgendused.
3. Morfoloogiline ühestamine – konteksti info põhjal leitakse sõnavormi paljude tõlgenduste seast korrektne tõlgendus.
4. Osalause piiride määramine – konteksti info, kirjavahemärkide ja morfoloogilise info põhjal leitakse liitlausetes osalause piirid.
5. Süntaktiliste märgendite lisamine – morfoloogilise info ja konteksti põhjal lisatakse sõnavormile kõik võimalikud süntaktilised märgendid.
6. Süntaktiline ühestamine – konteksti põhjal eemaldatakse sõnavormilt kõik lubamatud süntaktilised märgendid.

Süntaksianalüsaatorile grammatika kirjutamiseks on vaja koostada süntaktiliselt käsitsi märgendatud treeningkorpus, millel saab testida uusi reegleid. Hetkel on treeningkorpuses süntaktiliselt märgendatud sõnu umbes 50 000. Selleks, et täiendada automaatsele süntaksianalüsaatorile vajalikku treening- ja testkorpust, on vaja teha eespool nimetatud süntaktilise analüüsi etapid käsitsi või poolautomaatselt. Käsitsi märgendatud mahuka korpuse olemasolu võimaldab hõlpsasti leida süntaksianalüsaatori vigu ja aitab luua uusi reegleid analüsaatori grammatikasse.

Käsitsi süntaktiliseks märgendamiseks võetakse ette morfoloogilise ühestaja treening- ja testkorpuse tekstid (need on automaatselt morfoloogiliselt analüüsitud ja ühestatud ning seejärel hoolikalt erialase väljaõppega inimese poolt üle kontrollitud ja parandatud), analüüsitakse neid automaatselt kasutades olemasolevat süntaksianalüsaatorit ja selle grammatikat [1]. Seejärel antakse need tekstid kahele eksperdile, kes käsitsi parandavad vead märgenduses ning eemaldavad üleliigsed märgendid. Neid kahte teksti võrreldakse automaatselt, et leida üles erimeelsused. Tekkinud probleeme arutatakse suuremas ekspertide ringis ning seejärel viiakse lõplikud muudatused korpusesse sisse [12].

Märgendajad saavad ette süntaktilisest analüsaatori poolt väljastatava teksti, mis on näiteks selline :

```
Lennart
    Lennart+0 //_S_ prop sg nom #cap //    @SUBJ
on
    ole+0 //_V_ main indic pres ps3 sg ps af #FinV #Intr //
@+FMV
nagu
    nagu+0 //_J_ sub //    @J
prints
    prints+0 //_S_ com sg nom //    @SUBJ @NN> @ADVL
valgel
    valge+1 //_A_ pos sg ad //    @AN>
hobusel
    hobune+1 //_S_ com sg ad //    @ADVL
```

Märgendajad peavad lisama sisendtekstis õigete märgendite ette “+” märke, kui õiget märgendit ei leidu, siis lisama juurde märgendi koos sümbolitega “+=” ning kahtluse korral kirjutama kommentaare, mis algavad sümbolitega “??”. Töö väljund antud näite korral peaks olema alljärgnev:

```
Lennart
    Lennart+0 //_S_ prop sg nom #cap //    @SUBJ
on
    ole+0 //_V_ main indic pres ps3 sg ps af #FinV #Intr //
@+FMV
nagu
    nagu+0 //_J_ sub //    @J
prints
    prints+0 //_S_ com sg nom //    @SUBJ @NN> @ADVL +=@PRD
valgel
    valge+1 //_A_ pos sg ad //    @AN>
hobusel
    hobune+1 //_S_ com sg ad //    @ADVL +=@<NN
```

Käsitsi märgendamine ja korpuste suurendamine on ülimalt aeganõudev töö. Valede andmete sattumisel korpustesse võidakse teha valesid järeldusi ning sellega kaasnevaid probleeme on hiljem ülikeeruline lahendada.

2.4 Korpuse süntaktilised märgendid

Käesolev punkt on lühendatud väljavõtte Kaili Müürisepa väitekirjast [1].

Eesti keele kitsenduste grammatikas (ESTKG) märgendatavad süntaktilised funktsioonid vastavad enam-vähem standardses eesti keele grammatikas (EKG) eristatavatele süntaktilistele funktsioonidele.

1. Õeldise märgendid

@+FMV - finiidne öeldis

@-FMV - infiniitne öeldis

@+FCV - *olema* liitaegades ning modaalverbid ahelverbides, finiidne vorm

@-FCV - *olema* liitaegades ning modaalverbid ahelverbides, infiniitne vorm

@NEG - verbi eitus

Näide: *Sellest ei (@NEG) oleks (@+FCV) pidanud(@-FCV) rääkima(@-FMV).*

2. Põhja märgendid

@SUBJ - alus ehk subjekt

Näited: *Hommik (@SUBJ) oli eriliselt kaunis. Külalisi (@SUBJ) ei tulnudki. Meil on siia jääda (@SUBJ) kohe päris võ imatu.*

@OBJ - sihitis ehk objekt

Näited: *Kõik teooriad (@OBJ) arutati läbi. Öine vihm muutis õhu (@OBJ) värseks. Võhik oleks selle peale õlgu (@OBJ) kehitanud. Siin kavatses ta paremale pöörata (@OBJ).*

@PRD - öeldistäide ehk predikatiiv

Näited: *Hommik oli eriliselt kaunis (@PRD). Tema ainus töö oli vaatamas käia (@PRD).*

@ADVL - määrus ehk adverbiaal, ka fraasiadverbiaal

Näited: *Loomulikult (@ADVL) ei saanud ma oma oletust millegagi (@ADVL) tõestada.*

3. Laiendite märgendid

Fraasi laiendite märgendid näitavad lisaks süntaktilisele funktsioonile ka põhja leidumise suunda – kas paremas või vasakus kontekstis.

@AN> - omadus- ja järgarvsõna eestäiendina

@<AN - omadus- ja järgarvsõna järeltäiendina

@AD> - määrsõna eestäiendina

@<AD - määrsõna järeltäiendina

@PN> - kaassõna eestäiendina

@<PN - kaassõna järeltäiendina

@NN> - nimi-, ase- ja põhiarvsõna eestäiendina

@<NN - nimi-, ase- ja põhiarvsõna järeltäiendina

@VN> - partitsiip eestäiendina

@<VN - partitsiip järeltäiendina

@INF_N> - verbi infinitiitne vorm eestäiendina

@<INF_N - verbi infinitiitne vorm järeltäiendina

Traditsiooniline eesti keele grammatika loeb kaassõnafraasi ühtseks tervikuks. Kuna kitsenduste grammatika formalism eeldab lause analüüsi ühe sõna kaupa ning fraasistruktuuri ei moodustata, märgendatakse kaassõnafraasi kuuluva nimisõnafraasi põhi ühega allpool toodud märgenditest ning kaassõna märgendatakse kogu kaassõnafraasi süntaktilist funktsiooni näitava märgendiga.

@<P - eessõna laiend,

@P> - tagasõna laiend

Näited: *Isegi kaebama polnud kellegi (@P>) juurde minna. See juhtus pärast mingi dokumentaalfilmi vaatamist (@<P).*

EKG järgi on noomenist kvantori laiend atribuut, adverbi laiend sõltuvusmäärus. ESTKG-s tähistatakse mõlemat liiki kvantori laiendeid allpool toodud märgenditega, sest kvantori laiendi kohta kehtivad tavalistest täienditest-määrustest erinevad reeglid.

@<Q - kvantori järellaiend (*kaks meest @<Q*),

@Q> - kvantori eeslaiend (*inimesi (@Q>) tulvil*)

Näide: Vastu tuli palju (@SUBJ) energiat (@Q>) tulvil (@AD>) inimesi (@<Q).

4. Muud

@J - sidend

@I – hüüatus

3 GUI süntaktiliseks ühestamiseks

Käesolev peatükk tutvustab antud bakalaureusetöö peamist eesmärki – ühe konkreetse graafilise kasutajaliidese loomist, mis on mõeldud käsitsi süntaktiliseks ühestamiseks.

3.1 Ülesande püstitus

Keeletehnoloogia arenguks on vaja hetkel süntaktilise analüüsi arengut. Kaili Müürisepa loodud süntaksianalüsaator töötab praegu olemasolevate korpuste peal juba 85 % täpsusega [1]. Selliste tulemuste pealt ei ole võimalik enam uusi reegleid tuletada ning seetõttu ka analüsaatori tööd parandada. Seega on vaja suurendada treeningkorpust, mida saab teha vaid käsitsi ühestamise teel.

Kuni antud kasutajaliidese valmimiseni tuli filoloogidel, kes märgendasid, kasutada neile meeldivat tekstiredaktorit, kuid selline lähenemine tekitab põhjendamatult vigu - eksitakse märgendite formaadis; jäetakse mõned sõnad märgendamata või lisatakse kogemata juhuslikke sümboleid faili. Loomulikult pole selline märgendamise viis kuigi mugav ka filoloogile. Kuna kasutajatel on väga erinevad nii riist- kui ka tarkvaralised võimalused, siis kasutas mõni väga algelisi tekstiredaktoreid, teised jälle mugavaid, kuid ehk mitte nii optimaalset tekstiredaktorit. Ka väljundfaili formaadiga oli probleeme – on juhtunud, et väljundfail on *Microsoft Word* formaadis. Selle teisendamine ASCII formaati on aga lisatöö. Kuna enamasti on käsitsi märgendajad filoloogid, kelle arvutikäsitusala oskusteave on nõrgavõitu, on juhtunud õnnetusi ka kooditabelite vaheliste teisenduste käigus, mistõttu on näiteks kogu tekstis asendatud õ-tähed ö-dega. Kõigi nende probleemide tõttu otsustati luua kasutajaliides, mis võimaldaks märgendada juba osaliselt analüüsitud teksti.

Vaja oli seega luua programm, mis abistaks kasutajatel käsitsi süntaktilisi märgendeid märkida, lisada ning eemaldada. Samas on vaja, et programm võimaldaks kasutajatel ka viia läbi lihtsamaid kontroll- ning otsimistoiminguid.

Programm saab ette teksti kujul:

Aknas

```
aken+s //_S_ com sg in **CLB // @ADVL @<NN @NN>
```

kustus

```
kustu+s //_V_ main indic impf ps3 sg ps af #Intr // @+FMV
```

tuli

```
tuli+0 // _S_ com sg nom // @SUBJ @OBJ @ADVL @NN> @<NN  
$.  
. // _Z_ Fst //
```

Programm peab väljastama teksti kujul:

Aknas

```
aken+s // _S_ com sg in **CLB // +@ADVL @<NN @NN>  
kustus
```

```
kustu+s // _V_ main indic impf ps3 sg ps af #Intr // @+FMV
```

tuli

```
tuli+0 // _S_ com sg nom // +@SUBJ @OBJ @ADVL @NN> @<NN  
$.  
. // _Z_ Fst //
```

Programm peab väljastama logifaili, kus kirjas märgendaja nimi, viimati muudetud märgend ning selle asukoht failis. Selle abil peab kasutaja saama tööd jätkata pooleli jäänud kohast.

Funktsionaalsus, mida programm peab täitma:

- Peab töötama iga tuntud graafilise operatsioonisüsteemi all;
- Märgendamine peab olema võimalikult mugav ja tehtav optimaalsete operatsioonidega, näiteks hiire abil;
- Kasutajaliides peab olema eestikeelne ja mõistetav ka algajale arvutikasutajale;
- Väljund peab olema ASCII formaadis;
- Salvestuma peavad ka sellised tehnilised andmed nagu märgendaja nimi, viimati märgendatud rida ning lisatud kommentaarid;
- Peavad olema realiseeritud elementaarsed otsingu võimalused (otsida viimati märgendatud rida, kommentaari jne);
- Märkida peab olema võimalik ainult süntaktilisi märgendeid ning alati peavad sümbolid “+”, “+=” ja “??” jääma vahetult vastava komponendi ette;
- Peab olema võimalus lisada kommentaare ja uusi süntaktilisi tõlgendusi;
- Kontrollima märgitud märgendite ühesust

3.2 Realisatsioon

Programm on realiseeritud *Java jdk1.3* all. Kasutatud on Java Swingi komponente, teksti põhiosa on *DefaultStyledDocument* objekt.

Teksti sisse lugedes luuakse lihtahel, mille elementideks on tekstis esinevad sõnad. Iga sõna juurde kuulub õiges formaadis teksti puhul tema morfoloogiline teisendus ning lubatud märgendid, mille hulgast peab kasutaja hiljem valima.

Iga sõna juurde võib kuuluda mitu morfoloogilist teisendust. Seega luuakse iga sõna juurde omakorda lihtahel morfoloogilistest tõlgendustest. Kuna igal morfoloogilisel tõlgendusel võib olla (võib ka hiljem lisanduda) üle ühe märgendi, luuakse iga tõlgenduse juurde veel üks lihtahel – märgendid.

Iga sõna juurde võib veel kuuluda kantsulgudes olev märgend, kommentaar ning morfoloogiliste tõlgenduste ahel; morfoloogiline tõlgendus sisaldab endast sõna tüve, morfoloogilist teisendust, lauseosa lõpu tunnuseid ja märgendite ahelat. Märgend koosneb märgendist endast ning selle tüübist (tavaline, valitud, lisatud, üksi reas).

Tabel 3.1 illustreerib sisseloetud teksti struktuuri.

Sõna	→	Sõna
Sõnavorm		Sõnavorm
Kommentaar		Kommentaar
Süntaktiline märgend		Süntaktiline märgend
Morfoloogiline analüüs		Morfoloogiline analüüs
Sõnavomi algus tekstis		Sõnavomi algus tekstis
Sõnavormi lõpp tekstis		Sõnavormi lõpp tekstis
Morfoloogiline analüüs	→	Morfoloogiline analüüs
Sõna tüvi+ lõpp		Sõna tüvi+ lõpp
Morfoloogiline tõlgendus		Morfoloogiline tõlgendus
Lisa (osalause lõpu märk)		Lisa (osalause lõpu märk)
Süntaktilised märgendid		Süntaktilised märgendid
Süntaktilised märgendid:	→	Süntaktilised märgendid:
Süntaktiline märgend		Süntaktiline märgend
Tüüp		Tüüp

Tabel 3.1. Teksti struktuur

Kui kasutaja muudab ekraanil mõne sõna juurde kuuluvaid komponente, leitakse ahelas üles õige sõna ning viiakse sisse muudatused. Ahela kiiremaks läbimiseks luuakse faili sisse lugedes kümne elemendiline massiiv, kus kirjas iga teksti kümnendiku kohta, mitmes ahela element on antud sõna ning mitmes sümbol on esimese sõna esimene sümbol kogu tekstis. Iga muudatuse korral muudetakse ka antud massiivi teist komponenti.

Otsimise, kontrollimise ja ekraanil liikumise funktsioonide korral läbitakse moodustatud massiiv, otsitakse sobiv koht. Seejärel otsitakse sõnade ahelast sobiv sõna. Selleks kasutatakse sõna asukohta *DefaultStyledDocument*'is. Midagi lisades või eemaldades muudetakse ka antud sõna pikkust ning abimassiivi.

3.3 Nõuetele vastavus

Käesolev kasutajaliides hõlbustab kindlasti käsitsi süntaktilise märgendamise tööd. Autor loodab, et programmi areng ei piirdu antud töö kaitsmisega. Koostöö kasutajatega võiks olla suurem ning nende soovidega tahetakse arvestada.

Kasutajaliidese tegemisel on arvestada püütud üldiste nõuetega, mis esitatakse graafilistele kasutajaliidestele ning loomulikult ülesande püstituses esitatud nõuetega.

3.3.1 GUI-le esitatavad nõuded

Kasutajaliides on üritatud luua võimalikult intuitiivne. Kasutajal on selge, mis on vaja teha juhul, kui ta on tutvunud käsitsi märgendamise juhendiga. Samas on palju kasutatud üldisi GUI võimalusi – kerimisriba, olekuriba, menüüpunktid ning ikoonid. Üldine programmi kasutamine ei tohiks olla raske kasutajatele, kes on enne kokku puutunud graafiliste kasutajaliidestega. Samas on ilmselt veidi juhendamist vaja märgendite ja kommentaaride lisamisel ning eemaldamisel, kuna selleks tuleb klõpsata hiire paremal nupul ning see ei ole esmasel kasutamisel väga intuitiivne.

Menüüd on realiseeritud puukujulisena. On selgelt grupeeritud kõige olulisemad tegevused, lisavõimalused ning abiinfo. Kõige tuntumad menüüpunktid – faili avamine, salvestamine ja lahkumine on samas kohas, nagu on enamasti – kõige vasakpoolsem menüü ning nimega “Töö failidega”. Menüüde keskmine punkt sisaldab endas mugavusvahendeid ning kõige parempoolsem menüü on “Abiinfo”. Nii nagu on välja kujunenud enamike GUIde standard.

Ekraani kujundusel on mõeldud värvide valikul tagasihoidlikkusele. Enamus tekstist on musta värvi, olulisem info on välja toodud erinevate värvidega. Ikoonid on üritatud valida analoogsed teiste GUIde ikoonidega. Nii on “Ava uus fail” ikoonil kujutatud tühja paberilehte, salvestamise juures disketi kujutist ning abiinfo ikooniks on suur punane küsimärk. Kokku on ekraanil 4 ikooni, mis võimaldavad sooritada ainult kõige olulisemaid tegevusi, lisaks eelnevalt mainitutele saab avada ka pooleliolevat faili. Ekraani kõige suurem osa on ette nähtud teksti näitamiseks. Olemas on ka olekuriba, kus näidatakse infot parajasti pooleli oleva protsessi kohta või töö tegemise käigus aktiivse faili nime. Ekraani päis on standardne – seal asetsevad kasutajatele tuttavad võimalused akent minimeerida, maksimeerida ning sulgeda.

Tekstiline info, mida väljastatakse, võiks olla ilmselt veidi rohkem läbimõeldud. Esineb nii ainsust kui mitmust. Samas on üritatud dialoogides vältida mitmeti mõistmist. Küsimused on jah/ei vastused ning ei kasutata mitmekordset eitust.

Kasutust on leidnud ka üldtuntud lühendid nagu “OK” ja “Cancel”. Üldiselt on dialoogid, teated ning menüüpunktid kõik eesti keelsed. Abiinfot on võimalik saada üldisemat ning täpsemat. Kasutajal on võimalik lihtsalt hiire paremal nupul klõpsates teada saada teatud märgendi täpset tähendust. Samas pole abiinfo viidetega ega linkidega – kogu abiinfo on tavalises ASCII tekstis ning pole väga hästi struktureeritud.

Kasutajaliidese disaini juures on arvestatud ka reageerimisajaga. Pikemate protsesside juures kuvatakse olekuribal protsenti, kui kaugel faili või teksti lugemine on. Olekuribal püütakse kasutajale pidevalt adekvaatset infot jagada, mida programm parajasti teeb. Ka on realiseeritud “liivakell” – niipea, kui kasutaja mingi protsessi käima paneb, muutub kursor liivakella kujutiseks.

Suhtlus tulevaste kasutajatega on jäänud napiks. Programmi autor on suhelnud programmi ühe potentsiaalse kasutajaga – oma juhendajaga. Programmi valmimise käigus oli üks kokkupuude ka teise kasutajaga, kellelt saadi samuti palju kasulikku infot. Olemas on ka kasutajaküsitluse küsimustik, mida palutakse täita kõigil uutel kasutajatel. Kahjuks pole veel programmi nii laialdaselt kasutatud, et neid tulemusi saaks analüüsida. Antud teemast räägib lähemalt punkt “Programmi esimene avalik versioon”.

Kasutajaliideses on realiseeritud ka kõikide menüüpunktide funktsiooniklahvid. Samas on kõike võimalik teha ka hiire abil.

Vigade vältimiseks on realiseeritud muudatuste salvestamise kontroll. Kui kasutaja valib menüüpunkti “Lahku”, kuid pole viimaseid muudatusi salvestanud, pakutakse talle võimalus neid siiski salvestada. Loomulikult toimib sama funktsioon ka juhul, kui soovitakse faili avada. Samuti on võimalik kasutajal kontrollida, kas kõikidel sõnadel on täpselt üks märgend valitud. Selleks on olemas menüüpunkt “Kontrolli märgendite ühesust”. Aeg-ajalt on lubatud, et sõnavormile jääks üle ühe valitud märgendi, kuid üldjuhul peaks siiski kõigil sõnadel olema töö lõpetamise hetkeks täpselt üks märgend. Kasutajal on võimalik ka vaadata järjest enda lisatud kommenteeritud sõnu ning kommentaare endid. See võimaldab leida kergemini sõnu, mis esimesel läbivaatlusel on jäänud kasutajale segaseks või on tekkinud mõni teine probleem. Kogemata pole võimalik teksti lisada juhuslikke sümboleid, kuna teksti saab muuta vaid ette antud õige struktuuriga märgendeid lisades või eemaldades.

3.3.2 Vastavus ülesande püstitusele

Käesoleva töö juurde kuuluvas kasutajaliideses on realiseeritud kõik nõudmised, mis on ette nähtud punktis 3.1.

Märkida, lisada ning eemaldada on võimalik hiirega. Programm on suhteliselt intuitiivne ning kasutatav ka algajale arvutikasutajale. Realiseeritud on hulk lisafunktsioone. Olemas on ka kogenuma kasutaja jaoks funktsiooniklahvid. Kasutajale peaks olema iga hetk selge, millega kasutajaliides parajasti tegeleb ning mida kasutajalt oodatakse.

Kasutajaliidest on testitud Windowsi ja Linuxi graafiliste operatsioonisüsteemide peal ning probleeme ei ole tekkinud.

Rohkem võiks realiseerida otsimisfunktsioone – on avaldatud soovi ka lihtsa sõnavormi otsimise funktsioonile. Arendada võiks koostöös kasutajatega keele kasutust.

3.4 Programmi installeerimine

Eeldused riistvarale on samad, mida nõuab *Java jdk.1.3*.

Eeldused tarkvarale:

1. Graafilise kasutajaliidesega operatsioonisüsteem (*Windows, CDE, KDE, Gnome jt*);
2. *Java jdk.1.3*. Kui antud tarkvara puudub, on seda võimalik installeerida. Kaasasoleval CD-l on olemas vastav installeerimise fail. Kui kasutaja arvutis pole veel vastavaid *Java* teeke, tuleb installeerimise fail CD-lt kasutaja arvuti kõvakettale salvestada ning installeerida see endale.

Kasutajaliidese installeerimiseks on operatsioonisüsteemide Windows ja Linux jaoks olemas installeerimise failid. Windows keskkonnas tuleb käivitada fail *install.bat*. Kasutaja saab ise ette anda, kuhu soovib rakendust salvestada. Vaikimisi on selleks kataloogiks *c:*. Rakenduse käivitamiseks tuleb käivitada fail *Margendaja.bat*.

Unixi laadsete operatsioonisüsteemi kasutajatel tuleb installeerimiseks käivitada fail *install.sh*. Kasutaja arvutisse salvestatakse rakenduse tööks vajalikud failid kasutaja poolt sisestatud kataloogi. Vaikeväärtus on kasutaja kodukataloog. Kasutajaliidese käivitamiseks tuleb käivitada fail *Margendaja.sh*. Juhul, kui installeerimine ei õnnestu, tuleb kasutajal lahti pakkida fail *Margendaja.zip* endale meeldivasse kataloogi, jättes paika failistruktuuri.

Kõik vajalikud failid on olemas antud töö Lisas 2 oleva CD plaadil. Seal asub ka täpsem juhend, kust vajalikke faile igal juhul leida võib.

3.5 Kasutusjuhend

3.5.1 Programmi käivitamine

Kasutajaliidese käivitamiseks tuleb Windowsi keskkonnas käivitada fail nimega *Margendaja.bat*. Linuxi keskkonnas tuleb käivitada *Margendaja.sh*.

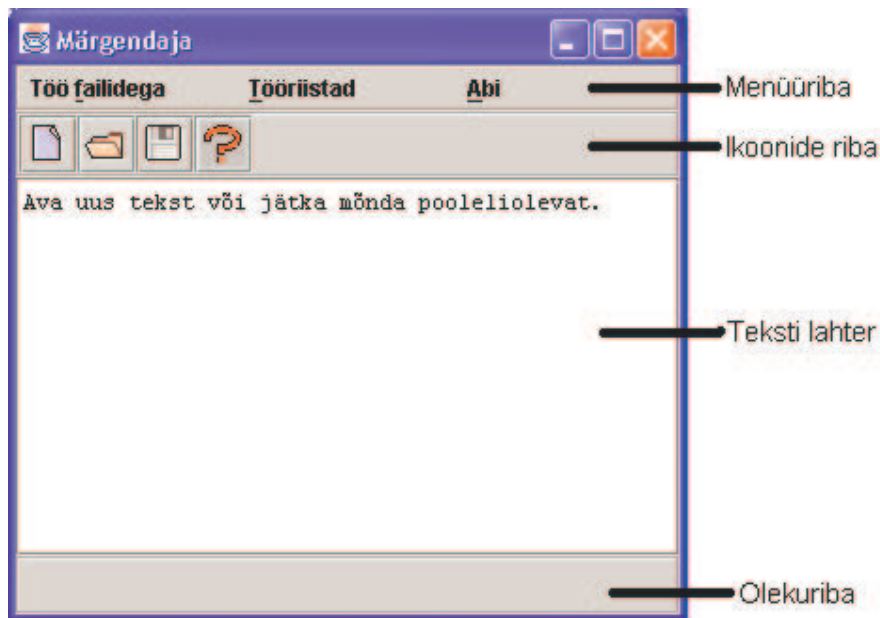
Käesolevat kasutajaliidest võib vajadusel käivitada ka käsurealt käsuga
java SyntaxAnnot

Kasutajaliides nõuab kasutaja arvutilt *java jdk1.3* toetust (vt. Installeerimise juhendit) ning graafilise kasutajaliidesega operatsioonisüsteemi.

Käsurida on võimalik käivitada *Windowsi* keskkonnas valides “*Start*” menüüst “*Run*” ning sisestades sinna “*Command*”. Unixi keskkonnas tuleb käivitada konsool.

3.5.2 Ekraani seletus

Käivitades programmi, avaneb kasutajale joonisel 3.1 näidatud ekraan.



Joonis 3.1 Programmi avapilt

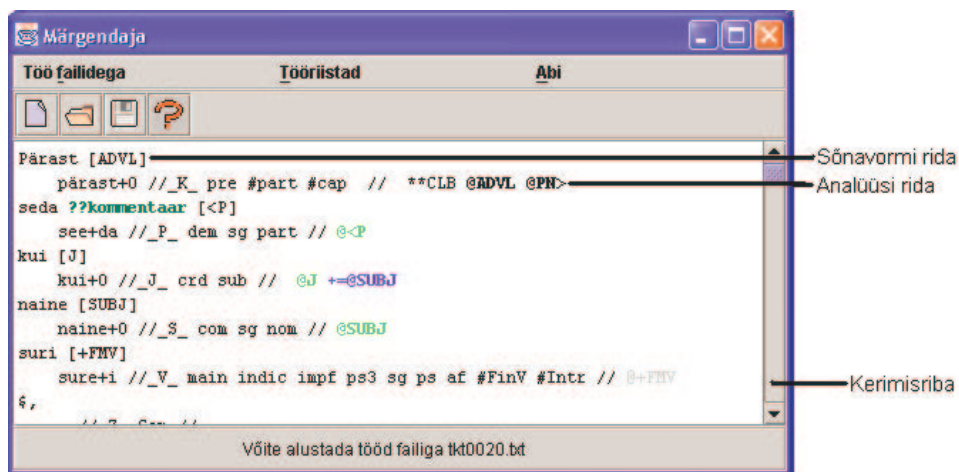
Menüüriba abil on võimalik kasutajal valida sobilik tegevus. Näiteks avada uut faili, salvestada seda, lahkuda vms.

Ikoonide riba on tähtsamate menüüde otse välja kutsumiseks. Esimene ikoon on uue faili avamine, teine pooleli jäänud faili avamine, kolmas teksti salvestamine ning neljas kasutusjuhend.

Teksti lahtrisse ilmub peale faili avamist soovitud faili tekst. Teksti äärde ilmub ka “Scrollbar” ehk kerimisriba, mille abil on võimalik kiirelt mööda faili liikuda.

Ekraani allservas on olekuriba. Sinna ilmub abistav info.

Pärast faili avamist, millest tuleb juttu edaspidi, näeb ekraan välja analoogne joonisega 3.2.



Joonis 3.2 Programmiaken näitetekstiga

Teksti lahtris võib näha erinevat värvi tekste. Tavalise musta värviga on kujutatud kõige üldisem info – sõnavorm, morfoloogiline analüüs ja igasugune muu info, mida muuta ei saa. Tumerohelise värviga on ekraanil kommentaarid, mida kasutaja saab lisada, eemaldada ning otsida. Märgendeid on nelja värvi:

- Paksu mustaga on märgendid ridades, kus on üle ühe märgendi – seega märgendid, mille vahel peab ühe märgendi valima.
- Helerohelisega on märgendid, mis on üks reas.
- Lilla värviga on kujutatud kasutaja poolt lisatud märgendid
- Halli värvi on märgendid, mis on eemaldatud.

3.5.3 Menüüpunktide seletused

3.5.3.1 Töö failidega

Menüüpunkti töö failidega alla on koondatud kõige olulisemad tegevused kasutaja jaoks. Edaspidi igast punktist lähemalt.

3.5.3.1.1 Uus fail

Menüüpunkt uue faili avamiseks. Mõeldud on kasutamiseks juhul, kui kasutaja ei ole veel antud faili märgendanud ning seejärel salvestanud. Programm annab võimaluse otsida sobiv fail. Seejärel võib minna veidi aega, kuna failist tuleb mällu lugeda enamasti üsna suur fail. Olekuribal on näha, kui palju failist on loetud. Tööd saab alustada, kui tekst on ekraanile kuvatud.

3.5.3.1.2 Ava pooleliolev fail

Kui kasutaja on mingit teksti märgendanud ning soovib oma tööd jätkata, on selle Menüüpunkti valik igati õigustatud. Kasutajalt küsitakse nime, et avada ka faili juurde kuuluv logifail. Nii saab kasutaja sobivat Menüüpunkti kasutades liikuda tekstis kohta, kus eelmine kord pooleli jäi. Nii nagu eelmises punktis, näidatakse olekuribal, kui mitu protsenti on failist loetud. Tööd saab alustada pärast teksti ilmumist ekraanile.

3.5.3.1.3 Salvesta fail

Tehtud töö salvestamiseks vajalik Menüüpunkt. Kasutajalt küsitakse salvestatavale failile nime. Kui kasutaja ei kirjuta failile laiendit, lisatakse automaatselt selleks “.txt”, vastasel korral salvestatakse täpselt kasutaja poolt antud nimega failiks. Kui kasutaja pole märkinud oma nime, küsitakse ka märgendaja nime. Faili ja kasutaja nimest luuakse kokku logifaili nimi. Logifaili salvestatakse märgendaja nimi ning viimati muudetud, lisatud, eemaldatud märgendi nimetus ning selle asukoht failis. Selle abil on võimalik hiljem kasutajal oma tööd pooleli jäänud kohast jätkata.

3.5.3.1.4 Märgendaja info

Selle Menüüpunkti abil on kasutajal võimalik muuta märgendaja nime. Seda võib vaja minna, kui soovitakse salvestada logifail teise kasutajanimega, kui tekst avati. Näiteks kui soovitakse säilitada erinevaid logifaile.

3.5.3.1.5 Lahku

Menüüpunkt programmist lahkumiseks. Kui tekstis on tehtud muudatusi peale viimast salvestamist, küsitakse kasutajalt, kas tõesti soovitakse väljuda viimaseid muudatusi salvestamata. Valida on kolme variandi vahel – salvestada ning lahkuda seejärel, lahkuda koheselt või loobuda (seega ei lahkuta programmist ning on võimalik tööd jätkata).

3.5.3.2 Tööriistad

Menüüpunkti “Tööriistad” alla on koondatud tegevused, mida kasutajal läheb vaja märgendamise käigus. Need pole hädavajalikud, kuid lihtsustavad kindlasti kasutaja tööd.

3.5.3.2.1 Näita algteksti

Kasutajale avatakse teine aken, kus näidatakse teksti ilma igasuguse analüüsita. See võimaldab kasutajal lihtsamalt taibata mõne sõna tähendust. Näiteks võib kasutaja lugeda enne töö alustamist läbi teksti, mida ta peab märgendama. Algteksti ekraan võib lahti olla kogu töö vältel. Seal ei ole võimalik siiski otsida või liikuda automaatselt kohta, kuhu ollakse jõutud märgendamisega.

3.5.3.2.2 Näita ühestatud teksti

Menüüpunkt, mille tulemusena näidatakse teksti lahtris ainult nende sõnade analüüsi ridu, mis on jäänud üheselt märgendamata. Ka nüüd on võimalik kasutajal märgendada ning kasutada kõiki tööriistu. Sel teel on võimalik märgendades näha ainult neid sõnu, mida on veel vaja märgendada. Samas tuleb rõhutada, et ka nendelt sõnadelt, millele süntaksianalüsaator on jätnud ainult ühe märgendi, eemaldatakse analüüs. Seega võivad sel teel märgendades jääda sisse süntaksianalüsaatori vead. Kindlasti tuleb enne töö lõpetamist kontrollida kogu teksti koos kõikide märgenditega. Ainult nii näeb, kas kõik sõnad on ikka õieti üheselt märgendatud.

3.5.3.2.3 Näita kõike

Kogu tekst kuvatakse kõigi oma komponentidega teksti lahtrisse. Vajalik eelkõige juhul, kui on eelnevalt kasutatud eelmist Menüüpunkti ning soovitakse uuesti näha kõikide sõnade analüüsi read. Kuna enamasti on tekstid pikad, on selle Menüüpunkti tegevus veidi aeganõudev. Olekuribal näidatakse protsentides, kui kaugel protsess on.

3.5.3.2.4 Kontrolli märgendite ühesust

Antud tööriist käib üle terve teksti ning otsib välja sellised sõnad, millel on üle ühe märgendi valitud. Üldjuhul on selline seis lubamatu, kuid vahel võib kasutaja seda just soovida. Kasutajale näidatakse dialoogiaknas, mitu sellist sõna on tekstis, ning antakse võimalus liikuda ekraanil esimese sellise sõna juurde. Seejärel on kasutajal võimalus liikuda kas järgmise taolise sõna juurde või katkestada tegevus. Kui kasutaja näeb sõna, millel on kaks või rohkem märgendit valitud eksklikult, peab kasutaja

valima “Katkesta” ning seejärel sõna muutma. Kui kasutaja soovib kontrollida ka viimaseid liigselt märgendatud sõnu, peab ta valima uuesti antud tööriista.

3.5.3.2.5 Näita kommentaare

Kasutajale näidatakse, mitu kommenteeritud sõna on terves tekstis ning võimaldatakse kasutajal liikuda kommenteeritud sõnade juurde. Kui kasutaja on liikunud kommentaari juurde, mida soovib parandada, peab vajutama “Katkesta”. Ekraani ülemisse serva jääb kommenteeritud sõna.

3.5.3.2.6 Mine viimati märgendatud sõna juurde

Valides antud tööriista, liigutatakse ekraani kohani, kus kasutaja viimase muudatuse tegi. Pärast faili avamist ning enne esimese sõna märgendamist või kommenteerimist, on selleks kohaks logifailist saadud positsioon. Logifailist saadakse info ainult sel juhul, kui märgendajaks on märgitud sama nimi, mis eelmisel salvestamisel.

3.5.3.3 Abi

“Abiinfo” koondab enda alla võimalused saada üldist abi – infot programmi kasutamisest, süntaktilisest märgendamisest ning programmi autorist.

3.5.3.3.1 Abiinfo

Uude aknasse ilmub kasutusjuhend, mis on analoogne käesolevaga. Abiinfo faili tee peab olema õige, vastasel korral ei suudeta faili leida ning abiinfo ekraani ei ilmu. Õige tee on /Margendaja/texts/Abi.txt

Kui järgitakse installeerimise juhendit ning ei kustutata Margendaja kataloogist tekste, peab vajalik fail olema õiges kohas.

3.5.3.3.2 Käsitsi märgendamise reeglid

Uude aknasse ilmub ESTKG süntaktilise märgendamise reeglid, mis on pärit Kaili Müürisepa doktoriväitekirjast ([1]). Kasutajal võib see ekraan olla avatud kogu töö vältel. Nii õnnestub loodetavasti vältida kasutajate teadmatusest ilmnevaid vigu. Reeglid võetakse failist, mis peab asuma kataloogis texts/Reeglid.txt. Kataloog texts/ peab asuma kataloogis, kust käivitatakse programm. Vastasel korral ei suudeta abiinfot jagada.

3.5.3.3 Programmi autoriõigustest

Ilmub ekraan, kus kirjas programmi autor ning versiooni number. Kui programmi kasutamisega probleeme, tasub edastada ka versiooni number.

3.5.4 Kuidas kasutada programmi

Programmi kasutamiseks on vajalik sobivas formaadis teksti olemasolu kasutaja arvutis. Näitetekstid on olemas kohe peale programmi installeerimist. Nende asukohaks on õige installeerimise korral /Märgendaja/texts/. Seal on 2 faili – tkt0020.txt ning tkt0031.txt.

Töö alustamiseks tuleb avada uus fail nii, nagu on seletatud punktis “Uus fail”.

Teksti lahtrisse ilmub peale veidikest ootamist soovitud fail kogu oma sisuga. Ekraani seletuse leiab käesoleva juhendi punktis “Ekraani seletus”. Süntaktilise märgendamise reeglite kogumiku leiab menüüpunktis Abi -> Käsitsi märgendamise reeglid.

Märgendite olekud muutuvad järgmises järjekorras: Valimata -> Valitud -> Eemaldatud. Valitud märgendil on “+” märk ees ning ta on heleroheheline. Heleroheheline, kuid ilma eesliiteta “+” on märgendid, mille on üheselt leidnud antud morfoloogilisele analüüsile süntaksianalüsaator.

3.5.4.1 Märgendi valimine

Märgendi valimiseks tuleb liikuda hiire kursoriga valitava märgendi peale ning klõpsata sellega märgendil hiire vasakul nupul. Märgend muutub heleroheliseks ning selle ette ilmub “+” märk.

3.5.4.2 Märgendi lisamine

Märgendi lisamiseks tuleb vastaval analüüsi real klõpsata hiire paremat nuppu. Ilmub menüü, kust tuleb valida “Lisa uus”. Avaneb ekraan, kus võimalik valida kõikide märgendite vahel. Tehes aktiivseks sobiva märgendi ning vajutades seejärel “OK” nupule, ilmub valitud märgend ekraanile lilla värvi ja “+=” eesliitega.

3.5.4.3 Märgendi eemaldamine

Märgendi eemaldamiseks tuleb liikuda hiire kursoriga eemaldatavale märgendile. Süntaksianalüsaatori poolt teksti lisatud märgend muutub halliks ning selle eest kaovad eesliited. Kasutaja poolt lisatud märgendid kaovad ekraanilt.

3.5.4.4 Kommentaari lisamine

Kommentaare saab lisada sõnavormi rea peale. Selleks tuleb hiirega liikuda soovitud sõnavormi juurde, vajutada hiire paremal nupul ning valida menüüst “Lisa kommentaar”. Avanevasse ekraani tuleb lisada soovitud kommentaar ning seejärel vajutada “OK” nupul. Kommentaar ilmub ekraanile sõnavormi järele tumerohelisena ning eesliitega “??”.

3.5.4.5 Kommentaari eemaldamine

Kommentaari eemaldamiseks tuleb kasutajal liikuda eemaldatava kommentaari peale, klõpsata hiire paremat nuppu ning valida “Eemalda kommentaar”. Kommentaar kaob ekraanilt.

3.5.4.6 Abiinfo töö käigus

Klõpsates paremal hiire nupul sõnavormi või analüüsi real, ilmub võimalus valida punkt “Abiinfo”. Selle valimisel ilmub väike seletus asjasse puutuva infoga. Märgeendi peal klõpsates ilmub ka valik “Seletus”. Selle alt ilmub aktiivse märgeendi tähendus. (Näiteks @<VN - partitsiip järeltäiendina). Alati soovitatakse lugeda ka kasutusjuhendit.

3.5.5 Funktsiooniklahvid

Realiseeritud on ka funktsiooniklahvid kõigile menüüpunktidele. Funktsiooniklahvid on mõeldud kogenumale kasutajale, kes soovivad kiiremalt ligi pääseda menüüpunktidele. Järgneb loetelu vastavate klahvikombinatsioonidega:

- Alt+f – “Töö failidega”
- Alt+f & u – “Uus fail”
- Alt+f & a – “Ava pooleliolev fail”
- Alt+f & s – “Salvesta fail”
- Alt+f & i – “Märgeandaja info”
- Alt+f & l – “Lahku”
- Alt+t – “Tööriistad”
- Alt+t & a – “Näita algteksti”
- Alt+t & ü – “Näita ühestatud teksti”
- Alt+t & n – “Näita kõike”
- Alt+t & o – “Kontrolli märgeandite ühesust”
- Alt+t & k – “Näita kommentaare”

- Alt+t & v – “Mine viimati märgendatud sõna juurde”
- Alt+a – “Abi”
- Alt+a & a – “Abiinfo”
- Alt+a & m – “Käsitsi märgendamise reeglid”
- Alt+a & p – “Programmi autoriõigustest”

(“+” märgiga ühendatud klahvid tuleb koos all hoida ning seejärel vajutada klahvil, mis märgitud pärast “&” märki.)

Märgendite ja kommentaaride muutmiseks funktsiooniklahve pole.

3.6 Kasutajaliidese esimene avalik versioon

Aprillis 2002 jõudis programmi loomine järku, kus oli vajalik esimene kasutajate testperiood. Esimese esitluse tarvis loodi interneti lehekülg, mille võib leida aadressilt <http://kadri.pirn.net>. Antud lehel on toodud tutvustav jutt süntaktilisest märgendamisest, installeerimis- ning kasutusjuhend, programmi kasutamiseks vajalikud failid ning kasutajaküsitlus. Kasutajaküsitlus on toodud ka antud töö esimeses lisas.

Paraku on töö kirjutamise hetkeks tagasiside kasutajatelt puudulik. Autoril puudub ka info, kui palju on programmi üldse kasutatud. Sellegi poolest loodab autor, et programmi hakatakse aktiivselt kasutama ning kasutajaküsitlusele saabub hulgaliselt vastuseid ning programmi areng jätkub.

Kokkuvõte

Töö eesmärgiks oleva graafilise kasutajaliidese loomist alustati jaanuaris 2001. Töö esimene testversioon ei olnud piisavalt kiire ega mugav ning seetõttu alustati programmi loomist teistkordselt peaaegu algusest suvel 2001. Töö esmane funktsionaalsus (märgendite märkimine, lisamine ning eemaldamine) oli täidetud realisatsioonis, mida käsitles käesoleva töö autori semestritöö [2]. Käesoleva töö juurde kuuluvat programmi on muudetud kasutajasõbralikumaks, on loodud palju lisafunktsioone. Programmi esimene kasutamiseks sobilik versioon valmis aprillis 2002 ning sellega koos publikatsiooni internetilehekül¹, kust on võimalik saada kõik vajalik programmi tööks.

Programmi valmimisest on möödunud liialt vähe aega, et saada infot programmi eelistest ja puudustest. Programmi levik on alles algusjärgus ning kasutajaküsitlusele, mis valmis samuti antud töö käigus, pole veel piisavalt vastuseid. Senini pole autorini jõudnud infot, mis raskendab programmiga töötamist. Seega on põhjust arvata, et programm on töökindel.

Töö esimese peatüki kirjutamine on aidanud autoril kindlasti palju mõista graafilistele kasutajaliidestele pandavaid nõudeid.

Käitsi süntaktilise märgendamise kasutajaliides on praeguseks küll kasutuskõlblik ning üsna mugav, kuid autor soovib kindlasti selles suunas tööd jätkata. Selleks peab programm olema mõnda aega kasutuses. Ainult nii tulevad välja programmi vead. Kindlasti saab kasutajaküsitluse põhjal teha ka üldisemaid järeldusi, mida oodatakse reaalselt ühelt kasutajaliideselt kaasajal.

¹ <http://kadri.pirn.net>

Kirjandus

1. Kaili Müürisep, "Eesti keele arvutigrammatika: süntaks".Dissertationes Mathematicae Universitatis Tartuensis 22. Tartu 2000.
2. Kadri Pirn, "Kasutajaliides käsitsi süntaktiliseks märgendamiseks", semestritöö, Tartu 2001.
3. Chris Brown, User Interface Design. Cornell University, 1995
<http://sma.cit.cornell.edu/cfg/design/contents.htm>
4. <http://www.interface-design.net/>
20. aprill 2002.
5. Jonathan Hodgson, The Design of Graphic User Interfaces (kursuse materjal)
<http://www.sju.edu/~jhodgson/gui/guihome.html>
6. James Hobart, Principles of good GUI Design
http://axp16.iie.org.mx/Monitor/v01n03/ar_ihc2.htm
7. Ian Sommerville, "Software engineering" , Wokingham [etc.] : Addison-Wesley,1993
8. Laura Arlov, How to design a good User Interface,1997
<http://www.wordfixers.no/process.html> (pole hetkel kättesaadav)
9. Haldur Õim, Haiki-Jaan Kaalep, Einar Meister, "Eesti keeletehnoloogia arenduskava", Tartu 2000.
10. Mare Koit, Haldur Õim. "Keeletehnoloogia Tartu Ülikoolis: õpe ja uurimistöö",
<http://www.cs.ut.ee/~koit/KT/Keeletehnoloogia11042002>
13. mai 2002
11. Tiina Puolakainen, "Eesti keele arvutigrammatika: morfoloogiline ühestamine". Dissertationes Mathematicae Universitatis Tartuensis 27. Tartu 2001.
12. Kadri Muichnek, Kaili Müürisep, Heili Orav, Andriela Rääbis, Heli Uiibo, "Süntaktiline märgendamine – arvutiga ja käsitsi". Arvutuslingvistikalt inimesele. Tartu Ülikool üldkeeleteaduse õppetooli toimetised 1. Tartu 2001, lk 219-244.

GUI FOR SYNTACTIC ANNOTATION OF ESTONIAN

Kadri Pirn

Abstract

The aim of this work is to create a graphical user interface for manual syntactic annotation. This work contains three chapters, the first of them discussing general requirements for a GUI, the second one giving an overview of Estonian language technology, and the third chapter describing the GUI for manual syntactic annotation. Syntactically marked language corpora form a basis for the vast majority of language processing tools. Until now, such corpora have been conducted with regular text editors that don't have any support for that particular task. However, the people who are creating such corpora have very different computer usage skills and habits, and many of them would like to see a more comfortable and supportive environment for their daily work. Therefore, having a GUI for the corpora development tasks could be highly beneficial for many linguists.

This GUI is based on the Java platform. It allows one to select, remove, and add syntactic labels easily with computer's mouse; there are also several functions which make marking process more comfortable.

The first version of the GUI was released in April 2002. Each user is encouraged to fill the user survey, in order to provide the feedback for the author.

This work also contains two appendixes. The first one of them is the aforementioned user survey, while the second has the form of CD and contains the GUI's source code, its documentation, and some example texts.

Lisa 1. Kasutajaküsitlus

Üldandmed:

Nimi:.....

E-mail:.....

Sugu: M

N

Vanus: ..-18

18-25

26-35

36-45

45-..

Haridus: IT-alane kõrg

IT-alane lõpetamata kõrg

filoloogia kõrg

filoloogia lõpetamata kõrg

muu

Kui heaks peate enda arvutikasutuse oskust?

Algaja

Keskmisest halvem

Keskmine

Keskmisest parem

Spetsialist

Millist tüüpi operatsioonisüsteemi Te kasutate?

Windows

Linux

Solaris

Muu

Kas olete ka enne käsitsi süntaktiliselt märgendanud?

Jah – tavalise tekstiredaktoriga

Ei

Kas peate sellise kasutajaliidese olemasolu oluliseks?

Jah

Ei

Mitu korda olete antud programmi kasutanud?

Mitte ühtegi

1

2-5

6-10

üle 10

Kuidas hindaksite kasutajaliides üldiselt?

Väga hea

Hea

Keskmine

Halb

Kas kasutajaliidesega töö tegemist oli raske õppida?

Jah, kuna programm pole intuitiivne

Jah, kuna kasutusjuhend oli nõrk

Ei, kasutusjuhendi abil oli lihtne alustada

Ei, programm oli nii intuitiivne, et kasutusjuhendit polnud vaja

Kas antud kasutajaliides lihtsustas Teie tööd?

Jah, lihtsustas väga

Jah, natuke lihtsustas

Ei, antud kasutajaliides pole mugav

Ei, antud kasutajaliidesega on tööd raskem teha kui tekstiredaktoriga

Milliseks peate antud kasutajaliidese mugavust?

Väga mugav

Suhteliselt mugav - arenguruumi veel on

Ei ole praegu eriti mugav, kuid ehk saab asja

Ei ole mugav ning sellest on väga palju puudu

Mis meeldis/ei meeldinud nende programmi osade juures?

Programmi käivitamine:.....

Teksti avamine:.....

Märgendamine:.....

Teksti salvestamine:.....

Tööriistad:.....

Palun kirjeldage, mis Teile antud kasutajaliidese juures kõige rohkem meeldis:

.....

Palun kirjeldage, mis Teile antud kasutajaliidese juures üldse ei meeldinud:

.....

Kas jäite rahule kasutajaliidese kiirusega?

Jah

Mitte eriti

Üldse mitte

Millise konfiguratsiooniga arvutit Te kasutate (mälu, protsessor jne)?

.....

Kuidas hindaksite kasutajaliidese keelelist korrektsust?

Väga hea

Korrektne

Leidub vigu

Palju keelelist ebakõla

Milliseid keelelisi korrekture teeksite?

.....

Kui kasulikud on Teile jaoks Tööriistade menüüs olevad funktsioonid?

.....

Millised muudatused programmis võiksid Teile tööd lihtsamaks teha?

.....

Millised on soovitud programmi autorile?

.....

Lisa 2. Kasutajaliides käsitsi märgendamiseks