

Tallinna Pedagoogikaülikool
Matemaatika-loodusteaduskond
Informaatika osakond

Hans Põldoja

**VEEBIPÕHISE ÕPIHALDUSSÜSTEEMI
KASUTAJALIIDESE DISAIN**

Magistritöö

Juhendaja:
Mart Laanpere

Autor: “.....” 2003

Juhendaja: “.....” 2003

Osakonna juhataja: “.....” 2003

Tallinn 2003

SISUKORD

1	Sissejuhatus.....	4
2	Veebirakenduste kasutajaliidese disaini põhimõtted	7
2.1	Kasutatavus	8
2.1.1	Kriteeriumid kasutatavuse hindamiseks	8
2.2	Veebirakenduste interoperabluse standardid.....	10
2.2.1	Hüpertexti standardid	11
2.2.2	Astmelised laadilehed	13
2.2.3	Veebisisu käideldavusjuhised	15
3	Metafooridel põhinev kasutajaliidese disain.....	17
4	IVA projekt õpihaldussüsteemi loomiseks TPÜ-le.....	22
5	Õpihaldussüsteemi IVA kasutajaliidese disain	25
5.1	Nõuete analüüs.....	26
5.1.1	Nõuete spetsifikatsioon.....	27
5.1.2	Kasutajastsenaariumid	30
5.1.3	Võrdlev analüüs	32
5.2	Kontseptuaalne disain	32
5.3	Visandid ja prototüübid.....	38
5.4	Tootmine	44
5.5	Kasutuselevõtt.....	46
6	Õpihaldussüsteemi IVA kasutajaliidese disaini evalvatsioon.....	48
6.1	Üldine arvamus IVA kohta	49
6.2	Hinnang IVA ekraanikujundusele.....	51
6.3	Hinnang IVA kasutajaliidese terminitele ja metafooridele.....	52
6.4	Hinnang IVA kasutajaliidese õpitavusele	53
6.5	Hinnang IVA õpihaldussüsteemi võimekusele	54
6.6	Hinnangud kasutatavuse kriteeriumide lõikes	55

6.7	Kokkuvõte ja järeldused	56
7	Kokkuvõte.....	58
	Kasutatud kirjandus	59
	Lisa 1 Volere malli raamistik nõuete spetsifikatsiooni koostamiseks	62
	Summary	64

1 SISSEJUHATUS

Viimasel aastakümnel on kõrghariduses ning eriti kaugkoolituses toimunud rida olulisi muudatusi, mille vedavaks jõuks on veebipõhiste õpihaldussüsteemide kasutuselevõtt. Õpihaldussüsteem (*Learning Management System*, LMS) on tarkvara, mis automatiseerib õpisündmuste haldamist. Enamus õpihaldussüsteeme sisaldab kasutajate haldust, õppematerjalide ning kasutaja andmete haldust, suhtlemis- ja koostöövahendeid ning toimingute monitooringut (Hall, 2001).

Hetkeolukorda võib pidada õpihaldussüsteemide buumiks – 90ndate keskel turuliidreiks tõusnud kommertssüsteemide kõrvale on tekkinud arvukalt vabavaralahendusi, mis sageli edestavad kommertskeskkondi paindlikkuse ning innovaatilise pedagoogilise lähenemise poolest. Selles situatsioonis on ka Tallinna Pedagoogikaülikool otsustanud loobuda Eestis enimkasutatava WebCT litsentsist ning katsetanud erinevaid vabavaralahendusi. Käesoleva töö autor on veebipõhiste õpikeskkondade ja õpihaldussüsteemide temaatikat käsitlenud juba diplomitöö raames (Põldoja, 2001), mille tulemusena töötas välja üldhariduskoolidele suunatud õpikeskkonna VIKO. TPÜ Haridustehnoloogia keskuses läbi viidud võrdlusuuringud näitavad, et olemasolevad õpihaldussüsteemid ei vasta hästi Eesti õppejõudude ja ülikoolide vajadustele eelkõige süsteemi kasutatavuse osas.

2002. aasta kevadel loodi TPÜ-s interdistsiplinaarne töörühm ülikooli oma e-õppekeskkonna väljatöötamiseks. Autori ülesandeks töörühmas oli loodava õpihaldussüsteemi kasutajaliidese disain. Käesolevas töös on palju kasutatud meie vormi, sest suur osa otsustest sündis IVA arendusmeeskonnas ühiselt.

Magistritöö eesmärgiks oli disainida kasutajaliides, mida iseloomustab

- intuiitiivsus läbi **metafooride** kasutamise;
- interoperaablus läbi W3C **standardite** järgimise;
- järjekindlus läbi **disainiprotsessi** süstemaatilise;
- köitvus läbi innovatiivse pedagoogilise **kontseptsiooni** ja uudsete töövahendite.

Käesolev töö tuginebki TPÜ veebipõhise õpihaldussüsteemi IVA arendamise käigus saadud praktilisele kogemusele ning sellele eelnenud teoreetilisele uurimistööle (Põldoja, Laanpere, 2002; Kikkas, 2002). Magistritööd ei saa võtta tagasivaatena IVA

arendusprotsessile, kuna see on valminud paralleelselt IVA arendusega, mis hetkel jätkub. Siiski on see terviklik käsitus kasutajaliidese disaini protsessist, kuna selle olulisem osa jääb arendusprotsessi esimestele etappidele.

Õpihaldussüsteemi IVA sihtgrupp on laiem kui TPÜ õppejõud ja tudengid – tahame pakkuda sobivat tasuta õpihaldussüsteemi ka Eesti E-ülikoolile ning teistele huvitatud õppeasutustele.

Kuna alustasime hiljem kui konkurendid, oli meil võimalik nende kogemustest ja ka vigadest õppida. IVA keskkonna muudab unikaalseks eelkõige see, et:

- keskkonna kontseptsioon tugineb innovatiivsele pedagoogilisele teooriale;
- keskkonna ülesehitus ja kasutajaliides tugineb väga tugevalt metafooridele;
- keskkond loodi olemasolevate veebirakenduste põhjal.

Käesoleva magistritöö eesmärgid on järgmised:

- analüüsida kirjanduse põhjal veebirakenduste disainile esitatavaid nõudeid ning metafooride rakendamist kasutajaliidese disainis;
- anda terviklik ülevaade kasutajaliidese disaini protsessist TPÜ õpihaldussüsteemi näitel;
- luua TPÜ veebipõhise õpihaldussüsteemi kasutajaliidese disain;
- viia läbi loodud õpisüsteemi kasutajaliidese evalvatsioon.

Peatükis 2 analüüsitakse kirjanduse ja veebiallikate põhjal veebirakenduste kasutajaliidese disaini üldiseid põhimõtteid nagu kasutatavus ja interoperabluse standardid.

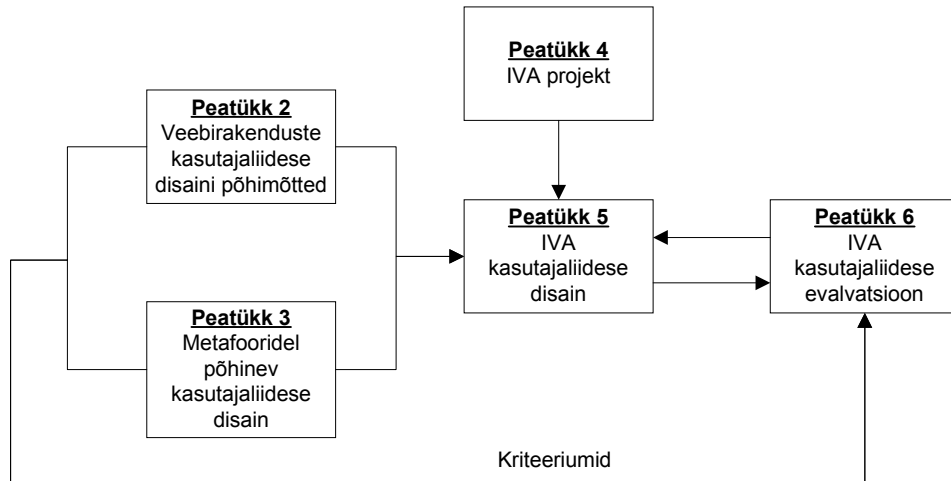
Peatükis 3 käsitletakse metafooride rakendusvõimalusi kasutajaliidese disainis ning tutvustatakse meetodeid, mille abil saab leida sobivaid metafooride komplekte.

Peatükk 4 keskendub TPÜ õpihaldussüsteemi loomise projekti üldisele tutvustusele, taustinfole, arendusprotsessi lühiülevaatele, meie ees seisnud olulistele valikutele ning nende põhjendustele.

Peatükis 5 antakse ülevaade veebirakenduse disainiprotsessi erinevatest etappidest ning nende käigus rakendatavatest kasutatavuse meetoditest IVA projekti näitel, tuuakse välja iga meetodi kasutamise tulemused ning antakse sealjuures soovitusi, mida tuleks vastavate meetodite rakendamisel silmas pidada.

Peatükis 6 tutvustatakse kasutajaliidese evalvatsiooni põhimõtteid ning viiakse läbi peatükkides 2 ja 3 saadud kriteeriumide põhjal IVA kasutajaliidese evalvatsioon.

Töö struktuur on esitatud joonisel 1.1.



Joonis 1.1. Magistritöö struktuur

2 VEEBIRAKENDUSTE KASUTAJALIIDESE DISAINI PÕHIMÕTTED

Veebirakenduste kasutajaliidese disaini põhimõtete juures liituvad mitmed aspektid. Kasutajate jaoks on väga oluline esmamulje ja süsteemi subjektiivne meeldivus, kuid see on iga inimese puhul siiski individuaalne. Tunduvalt lihtsam on tuua välja vigu, milline ei tohi veebilehe kasutajaliides olla kui määratleda, milline see olema peab. Tuntud veebidisaini ja kasutatavuse ekspert Jakob Nielsen on kirjutanud järgmist:

"Kasutajate veebis käitumise uuringud näitavad, et kasutajatele ei meeldi keerukad või aeglased lehed. Inimesed ei taha oodata. Inimesed ei taha õppida, kuidas kasutada kodulehekülge. Ei saa olla kursust või kasutusjuhendit kodulehekülje kasutamiseks. Inimesed peavad olema võimelised mõistma veebirakenduse kasutamispõhimõtet paari sekundiga pärast lehekülje vaatamist." (Nielsen, 2000)

Esmajoones peame silmas pidama neid, kellele me õpihaldussüsteemi loome: üliõpilased, õppejõud ning kindlasti ei tohi unustada ka süsteemi haldajaid. Seega peab meie õpihaldussüsteem olema võimalikult kasutajasõbralik. Alapeatükis 2.1 toome välja kriteeriumid, mille põhjal saab hinnata veebirakenduse kasutatavust ning tutvustame lühidalt tüüpilisi kasutatavusvigu.

Interneti toimimise aluseks on kindlate standardite järgimine, millest olulisim on kahtlemata Interneti alustala TCP/IP võrguprotokoll. Üheks selliseks standardiks on World Wide Webi alusepanija Tim Berners Lee poolt juhitava World Wide Web Consortiumi (edaspidi W3C) poolt välja antav HTML keele standard, mille viimane versioon on hetkel XHTML 1.1. Kuna nimetatud standard on soovitusliku iseloomuga, on veebi leviku laienedes probleemiks selle standardi vähenev järgimine.

Veebirakenduste loomisel tuleb kahtlemata arvestada ka puuetega inimestega. Tihti on Internet nende jaoks peamiseks infoallikaks ning veebipõhine õppevorm annab neilegi ligipääsu kõrgharidusele. W3C juures töötab spetsiaalne töörühm veebilehtede käideldavusstandardi WCAG kallal, mille järgimine võimaldab kõigile võrdset ligipääsu veebirakendusele.

Alapeatükk 2.2 tutvustabki lühidalt neid standardeid. Õpihaldussüsteemidega on tihedalt seotud ka õpiobjektide metaandmestiku standardid, kuid nende käsitlemine ei mahu antud magistritöö teema raamidesse.

2.1 Kasutatavus

Ärimaailmas võib pidada tehingut õnnestunuks, kui müüja ja ostja mõlemad saavad sellest kasu. Sama kehtib ka veebi kohta: nii informatsiooni pakkuja kui vastuvõtja peavad tundma ennast võitjatena. Tänapäeval ei saa ülikooli võrdlemist äriga pidada enam pühaduseetuseks – nii üliõpilane, õppejõud kui ka ülikool institutsioonina peavad veebipõhise õppemeetodi rakendamisest kasu saama.

Siinjuures on väga oluline tegur veebilahenduse kasutatavus (ingl k. *usability*) – see on süsteemi kasutajaliidese atribuut, mis näitab kui kerge on õppida süsteemi või selle komponente käsitlema, süsteemi sisendeid ette valmistama ja väljundeid tõlgendama. (Preece, 1996) Veebiliidese disainiprotsessis on kasutatavus kesksel kohal.

Kuigi veeb baseerub suhteliselt lihtsal kasutajaliidesel, mis koosneb hüperlinkidest, nuppudest, menüüdest, tekstiväljadest, tekstist ja graafikakomponentidest, on kasutatavuse probleemid väga sagedased. Alljärgnevalt on välja toodud neist olulisemad (Brinck et al, 2002, lk 4-11):

- inimese taju probleemid (näiteks raskused pikast nimekirjast vajaliku valimisel);
- navigeerumisprobleemid (näiteks kasutajal puudub teave oma täpsest asukohast keskkonnas või on keskkonnas kasutatud ebastandardseid navigatsioonilahendusi);
- kasutaja mälu (kui liiga palju asju või pikaks ajaks tuleb meeles pidada on tõenäoline, et midagi ununeb);
- andmebaasi kasutamisega seotud probleemid (näiteks topelt postitamine).

2.1.1 Kriteeriumid kasutatavuse hindamiseks

Brinck, Gergle ja Wood on kasutatavuse defineerimisel toonud välja 6 disaini eesmärki, millele vastavus määrab veebirakenduse kasutatavuse taseme (2002, lk 2-3):

- funktsionaalne korrektsus;
- kasutusefektiivsus;
- õppimislihtsus;
- meeldejätmislihtsus;
- veakindlus;

- subjektiivne meeldivus.

Samuti on veebirakenduste kasutatavust põhjalikult käsitlenud Jakob Nielsen, kes esitab 10 kasutatavuse heuristikut koos lühiseletusega (Nielsen, 1994a):

- ülevaade süsteemi staatusest;
- seos veebirakenduse ja reaalse maailma vahel;
- kasutajapoolne kontroll ja vabadus;
- järjekindlus ja väljakujunenud standardite järgmine;
- vigade ennetamine;
- pigem äratundmine kui meenutamine;
- paindlikkus ja kasutusefektiivsus;
- esteetiline ja minimalistlik kujundus;
- aita kasutajatel ära tunda, ennetada ja hoiduda vigadest;
- abiinfo ja dokumentatsioon.

Ülaltoodud heuristikute põhjal on Nielsen ja Molich koostanud heuristilise evalvatsiooni meetodi (Nielsen, 1994b).

Põhjaliku loetelu soovitusi, mida tuleks kasutajasõbraliku veebirakenduse loomisel arvesse võtta, on välja toonud ka Bruce Tognazzini (Tognazzini). Evalvatsiooni aluseks võtmiseks on see liiga suuremahuline, kuid see-eest pakub see palju kasulikke soovitusi veebirakenduste kasutajaliidese kujundajatele.

Kitsamalt õpihaldussüsteeme silmas pidades võime ülaltoodu põhjal kokku panna 5 kriteeriumide gruppi, millega tuleb õpihaldussüsteemide kasutajaliidese loomisel arvestada.

- **Õpihaldussüsteemi kasutama õppimise lihtsus.** Kuna sihtgrupi siseselt võib IKT pädevuste tase olla väga erinev, on õpihaldussüsteemi puhul selle kasutamise lihtne omandamine väga oluline. Selleks peab süsteem suhtlema kasutajaga kasutaja keeles, olema kooliõpilasele ja õpetajale täielikult mõistetav. Informatsioon peab olema esitatud loomulikus ja loogilises järjekorras, järgides tavalises õppeprotsessis väljakujunenud tavasid. Tuleb järgida väljakujunenud standardeid, et kasutajad ei peaks mõtlema, kas sõnad, situatsioonid ja tegevused omavad sama tähendust, mis teistes keskkondades. Menüü lingid peavad olema nimetatud üheselt mõistetavalt.

- **Õpiahaldussüsteemi kasutamise efektiivsus.** Süsteemis liikumine peab olema võimalikult kiire ja lihtne. Eelistatud on tekstimenüüd ja mõõdukas graafika, et tagada lehekülgede kiire allalaadimine. Elementide paigutus peab olema kõikidel lehekülgedel sama, et muuta süsteemis liikumine võimalikult lihtsaks ja sujuvaks. Kogenud kasutajale tuleb pakkuda linke, mis võimaldavad tal jõuda otse vajaliku infoni. Leheküljed peavad olema lühikesed ning olulisem informatsioon peab olema lehe ülaosas ilma alla kerimata nähtav.
- **Piisav tagasiside kasutajale.** Arvestades õpiahaldussüsteemi sihtgruppi, peab olema pidevalt tagatud asjakohane tagasiside. Kasutajal peab olema ülevaade oma asukohast süsteemis. Süsteemi abiinfo peab olema kasutajale igal hetkel kättesaadav ja loetlema konkreetseid sammud eesmärgi saavutamiseks. Abiinfo tekst ei tohi olla liialt mahukas, sest see raskendab vajaliku osa üles leidmist.
- **Vigadekindlus.** Süsteemi disaini etapil tuleb teha kõik võimalik vigade ennetamiseks ja vältimiseks. Kasutajatel peab alati olema vigade vältimiseks varuväljapääs, et pöörduda keerukast ja soovimatust olukorrast tagasi pealehele. Veateated peavad olema kasutajale arusaadavad, viitama täpselt probleemile ning pakkuma konstruktiivse lahenduse. Tuleb vältida weakoodide näitamist kasutajale.
- **Esteetiline ja minimalistlik kujundus.** Õpiahaldussüsteemi kujundus peab olema hoolikalt läbi mõeldud ning esteetiliselt nauditav. Pigem tuleb jääda konservatiivseks, sest süsteemil on püsikasutajad, keda pole vaja efektse kujundusega üllatada. Tuleks kasutada läbivat visuaalset elementi, mis aitaks tugevdada ühtekuuluvust keskkonna erinevate osade vahel. Teksti fondi, värvi ja suuruse valikul tuleks arvestada, et see oleks võimalikult lihtsalt nii ekraanilt loetav kui ka prinditav. Taustavärv ei tohi raskendada teksti lugemist, tuleb vältida taustamustrite kasutamist.

2.2 Veebirakenduste interoperabluse standardid

1994. aastal asutas Tim Berners-Lee Massachusettsi Tehnoloogiainstituudi juures World Wide Web konsortsiumi, eesmärgiga arendada välja veebi interoperablust tagavad standardid. Tänapäevaks ühendab W3C konsortsium üle 450 liikmesorganisatsiooni akadeemilisest ning ärimaailmast.

W3C tegevuse kaugemateks eesmärkideks on (W3C, 2000):

1. **Universaalne ligipääs** – muuta veeb ligipääsetavaks kõigile inimestele propageerides tehnoloogiad, mis arvestavad kõikidel kontinentidel elavate inimeste laiade erinevustega kultuuris, keeles, hariduses, võimetes, materiaalsetes ja tehnilistes võimalustes ning füüsilistes piirangutes.
2. **Semantiline veeb** – luua tarkvarakeskkond, mis võimaldab igal kasutajal rakendada parimal viisil veebis kättesaadavaid ressursse.
3. **Usaldatav veeb** – suunata veebi arengut, pidades hoolikalt silmas selle tehnoloogia õiguslikke, majanduslikke ning sotsiaalseid aspekte.

W3C poolt välja töötatud interoperabluse standarditest vaataksime lähemalt hüperteksti standardeid HTML ja XHTML, astmeliste laadilehtede standardit CSS ning veebilehtede käideldavusstandardit WCAG. Interoperaablus tähendab kahe või enama süsteemi võimet vahetada informatsiooni ning kasutada vahetatud informatsiooni (IEEE, 1990).

2.2.1 Hüperteksti standardid

Esimesed nägemused omavahel lingitud dokumentide võrgustikust pärinevad juba 1940'ndatest aastatest, mil Vannevar Bush esitas oma visiooni süsteemist Memex (Bush, 1945). 1965 aastal sõnastas Ted Nelson hüperteksti mõiste ning 1981 aastal kirjeldas süsteemi Xanadu, mis erinevalt Memexist pidi töötama juba personaalarvutil. 1987 avaldati Bill Atkinsoni loodud lihtne hüpertekstisüsteem HyperCard, mis levis laialdaselt, kuna seda levitati 1987-1992 tasuta kõikide Apple Macintoshi arvutitega. HyperCardi analoogiks PC platvormil võib pidada Asymetrix Toolbooki, mis on tänaseks üks võimalusterohkemaid multimeedia autorsüsteeme.

HTML keelele (hüperteksti märgendikeel, ingl k *HyperText Markup Language*) pani 1989. aastal aluse Tim Berners-Lee, kes otsis teadustekstide koostamiseks lihtsamat võimalust, kui TeX või Postscript. Uus süsteem kasutas HTTP protokollit ning oli tarkvaraplatvormist sõltumatu. Laiemale avalikkusele esitati veebi idee 1991. Esimene HTML keele standard pärineb aastast 1992. Peagi möödus veeb kasutajate arvu poolest WAIS'ist, Gopherist ja UseNetist.

HTML keele peamiseks eesmärgiks on dokumendi struktuuri kirjeldamine ning dokumentide omavaheline linkimine, kuid selle abil on võimalik ka lehekülgede

kujundust paika panna. Laiemalt levisid HTML keele versioonid 2.0, 3.2 ja 4.0. HTML keele viimaseks versiooniks jäi HTML 4.01, mis sai W3C ametlikuks soovitusel 24. detsembril 1999. Antud soovitus on suunatud nii veebilehtede koostajatele kui ka brauserite ja teiste HTML vormingus dokumente kasutavate programmide tootjatele.

Interneti levikuga laiadesse massidesse on tunduvalt halvenenud standardite järgimine. Brauserite vanemad versioonid (näiteks 4. versiooni Netscape brauserid, mida veel mitmepool kasutatakse) ei saa hakkama kõigi HTML võimaluste näitamisega. Lisaks sellele on nii Microsoft kui Netscape lisanud oma brauseripõhiseid HTML'i elemente, mida ainult nende brauser oskab näidata. Internet Exploreri suur ülekaal brauserite turul tähendab seda, et osa veebirakenduste loojaid kasutab võimalusi, mis nõuavad ilmtingimata veebilehe külastajalt uuemat Microsofti veebibrauserit. Viimasel paaril aastal on Internet Exploreri ja Netscape'i kõrvale kerkinud rida teisi esialgu küll väiksema kasutajaskonnaga brausereid nagu Mozilla, Opera jmt.

Jeffrey Zeldman toob välja järgmised tagajärjed, mis on tingitud standardite mittejärgimisest (Zeldman, 2001):

- leheküljed pole ligipääsetavad puuetega kasutajatele;
- leheküljed paistavad erinevates brauserites erinevalt ning vanema versiooni jaoks loodud leheküljed ei pruugi töötada uuema brauseriga (näiteks Netscape 4 jaoks loodud lehekülg võib paista erinevalt Netscape 6 peal);
- veebirakenduste loomine nõuab rohkem aega ja raha, kui luuakse erinevatele brauseritele oma versioonid;
- kuna kujundus ja sisu ei ole lahus, on problemaatiline lehekülgede portimine õhukestele klientidele kasutatavale kujule.

Nendele probleemidele on lahenduseks W3C soovitude järgimine. HTML keele soovitude edasiarendamine on lõpetatud ning uueks soovitude perekonnaks on XHTML. See on paljuki sarnane HTML'iga, kuid on disainitud töötamiseks koos laiendatud märgendikeelega XML, mis peaks kujunema kõigi veebirakenduste põhikeeleks. XHTML'i loomisel on arvestatud, et ta töötaks koos erinevate dokumenditüüpide ja hüpermeediumi programmeerimiskeeltega. Tänu sellele saab

pakkuda rikkama sisuga veebirakendusi laiemale hulgale klientidele: mobiiltelefonid, televiisorid, autod, pihuarvutid, multimeediakioskid ja tavalised personaalarvutid.

XHTML'i dokumentitüübid on järgmised:

- **XHTML 1.0 Strict.** Dokumendi struktuuri kirjeldamine. Kõikide kujundusega seotud märgiste asemel tuleb kasutada CSS'i.
- **XHTML 1.0 Transitional.** Sobivaim dokumentitüüp olukorras, kui tahetakse, et lehekülj vastaks XHTML'i standarditele kuid kujundus töötaks ka CSS'i mittetoetavate brauseritega. Võimalik kasutada kujunduseks mõeldud HTML'i märgiseid.
- **XHTML 1.0 Frameset.** Kasutatakse juhul, kui soovitakse jaotada brauseriakent paneelideks.
- **XHTML 1.0 Basic.** See dokumentitüüp on mõeldud klientidele, mis ei toeta kõiki XHTML'i võimalusi: mobiiltelefonid, pihuarvutid jne. Dokumentitüüp võimaldab kasutada pilte, ekraanivorme, lihtsamaid tabeleid ja objekte.
- **XHTML 1.1.** Modulaarse ülesehitusega XHTML, millest on kõrvaldatud kõik hüljatud HTML märgised, sarnaneb XHTML 1.0 Strict dokumentitüübiga, kuid sobib paremini XHTML edasiarendamise aluseks ning teiste keeltega integreerimiseks.

Et kontrollida lehekülje vastamist standarditele on W3C leheküljel validaator, mille sisendiks võib anda kontrollitava lehekülje veebiaadressi või laadida faili arvuti kõvakettalt. Veebirakenduste valideerimine on keerulisem, sest nende puhul genereeritakse hüpertekst iga päringu peale dünaamiliselt ning valideerimiseks tuleb see staatiliseks HTML failiks salvestada. Siinjuures tuleb arvestada sellega, et ühel veebirakenduse lehel võib olla mitu erinevat esitusviisi, mida tuleb eraldi valideerida.

2.2.2 Astmelised laadilehed

Astmeliste laadilehtede ehk CSS'i (*Cascading Style Sheets*) tehnoloogiat tutvustati esmakordselt 1996. aastal. Tol ajal oli valdavaks brauseriks Netscape ning HTML keele standardite alal käis veel võitlus W3C ja konkurentide vahel. Lehekülje tüpograafia kujundamiseks kasutati HTML'i piiratud võimalusi ning paigutust hakati määrama tabelite abil. Tänapäevaks on Internet Explorer veelgi suurema ülekaaluga, kui Netscape aastal 1996, samuti on välja arenenud alapeatükis 2.2.1 kirjeldatud

hüpertexti standardid. Veebirakenduste kasutajaliidese kujundamisel pole aga siiani täielikult loobunud 1996 kasutatud võtetest.

CSS keel pakub suurepäraseid võimalusi lehekülje tüpograafia ja paigutuse määramiseks. Paljud kujundusvõtted, mis puhta HTML'i abil olid võimatud teostada, on CSS'iga lihtsalt lahendatavad, näiteks teksti reavahe määramine. Sellest veelgi olulisem on võimalus veebidokumendi struktuur ja kujundus omavahel täiesti lahutada. CSS'i abil on võimalik defineerida veebirakenduse kujundus keskselt, nii et üks koodimuudatus mõjutab kogu rakenduse väljanägemist. W3C soovitab kasutada HTML'i märgistest ainult seda osa, mis on mõeldud dokumendi struktuuri kirjeldamiseks (peakirjad, tekstilõigud, loetelud) ja objektide lisamiseks (tabelid, pildid, ekraanivormid). Kogu kujundus ja paigutus on tuleks määrata CSS'i abil.

Praegusel hetkel on kasutusel W3C soovitud CSS1 (1996) ja CSS2 (1998) ning käib töö CSS3 kallal. Lisaks sellele on eraldi CSS soovitus mobiilseadmete jaoks. Laadilehtede kasutamisel on peamiseks probleemiks see, et enamiku brauserite toetus CSS'le on hetkel ainult osaline – peamised CSS käsud töötavad, kuid vähemkasutatavaid käskke brauser ei toeta või tõlgendab W3C soovitustest erinevalt. Üldiselt toetavad levinumad brauserid CSS'i alates järgmistest versioonidest:

- Netscape Navigator 4.0 ja uuemad;
- Internet Explorer 3.0 ja uuemad;
- Opera 3.5 ja uuemad;
- Mozilla kõik versioonid.

Tüpograafia ja värvilahenduse määramine CSS'i abil töötab piisaval tasemel kõigi enamlevinud brauseritega, kuid rohkem probleeme põhjustab lehekülgede paigutuse määramine. Levinud viisiks on määrata lehekülgede paigutus tabelite abil, kuid need on HTML'is ette nähtud siiski andmete esitamiseks tabeli kujul (näiteks erinevad brausrid ja nende vead), mitte teksti ja objektide paigutuse määramiseks lehekülgedel. Praegu on tabelid siiski ainuke võimalus keerulisema paigutuse määramiseks nii, et see ka vanematel brauseritel (nagu Netscape 4.x) töötaks, kuid tulevikku arvestades soovitatakse juba praegu kasutada paigutuse määramiseks ainult CSS'i võimalusi. Rachel Andrew (Andrew, 2001) toob välja järgmised soovitud:

- **Järgi standardeid.** Oluline on, et lehekülg oleks valideeruv HTML 4.01 või XHTML. Uuemad brauserid oskavad standarditele vastavat koodi küllaltki

hästi interpreteerida ning vanemate brauserite puhul on kujundusprobleeme lihtsam lahendada, kui on teada, et viga on brauseri renderdusmootoris ja mitte HTML koodis endas.

- **Paiguta sisu loogilises järjekorras.** Tekstid leheküljel peavad olema loogilises järjekorras ka ilma laadileheta. Sellises järjekorras, nagu dokumendi sisu on HTML failis, näitavad seda mittegraafilised brauserid (Lynx) ja loevad ekraanilugejad.
- **Anna 4. versiooni brauseritele erinev laadileht.** JavaScripti abil on võimalik kontrollida vaataja brauseri versiooni ning määrata 4. põlvkonna brauseritele erinev CSS laadileht, milles kasutatakse vaid Netscape 4-ga ühilduvat alamhulka CSS'i käskudest.
- **Ära muretse, kui lehekülg ei paista kõikjal ühtemoodi.** 19" tollise monitori ja pihuarvuti ekraanil ei saagi lehekülg samasugune välja näha. CSS2 pakub võimalusi määrata erinev laadileht näiteks ekraanil vaatamiseks, printimiseks, pihuarvuti jaoks ja ekraanilugejale.

Aadressil <http://jigsaw.w3.org/css-validator/> paikneb ka CSS'i validaator, mis võimaldab kontrollida laadilehe vastavust standarditele.

2.2.3 Veebisisu käideldavusjuhised

Erinevatel hinnangutel on puuetega inimeste osakaal ühiskonnas sõltuvalt aluseks võetud metoodikast 10...20 %. USA's on puuetega inimeste osakaal 16-aastaste ja vanemate hulgas U.S. Census Bureau 1999. a. andmetel isegi 22 % (Brinck et al, 2002, lk 48). E-õpe pakub liikumis-, nägemis või kuulmispuuetega inimestele võimaluse hariduse omandamiseks koduarvuti tagant. Seetõttu tuleb selle sihtgrupiga veebirakenduste disainil kindlasti arvestada.

Hüperteksti ja laadilehtede standardite järgi edukalt valideeruv veebirakendus ei pruugi veel olla piisavalt kasutajasõbralik. W3C juures on spetsiaalne töögrupp WAI (*Web Accessibility Initiative*), mille eesmärgiks on töötada välja reeglistik kasutajasõbralike ning erinevatel platvormidel käideldavate veebilehtede loomiseks ning testid olemasolevate lehtede käideldavuse kontrollimiseks.

WAI poolt on välja töötatud veebisisu käideldavusjuhised (*Web Content Accessibility Guidelines*, WCAG), mis on esitatud kontrollküsimuste vormis. Kehtiv soovitus

WCAG 1.0 pärineb 1999. aastast, kuid arvestades WWW kiiret arengut tasub veebirakenduste loomisel aluseks võtta hetkel WCAG 2.0, mille mustand on samuti avaldatud ning täiesti kasutatav (Praust, 2003).

WCAG 2.0 on jagatud viieks kategooriaks (W3C, 2003):

- **Tajutav** (*perceivable*). Kindlusta, et kõik funktsionaalsused ja informatsioon on esitatud kõigile kasutajatele tajutavas vormis, va need aspektid, mida pole võimalik sõnades edasi anda.
- **Rakendatav** (*operable*). Kindlusta, et kõik kasutajaliidese elemendid on igale kasutajale rakendatavad.
- **Navigeeritav** (*navigable*). Aita kaasa leheküljel orineteerumisele ja liikumisele.
- **Arusaadav** (*understandable*). Tee sisust ja juhtimisseadmetest arusaamine nii võimalikult lihtsaks.
- **Viimistletud** (*robust*). Kasuta veebitehnoloogiaid, mis tagavad sisu maksimaalse töövõime praeguste ja tulevaste käideldavustehnoloogiate ning brauseritega.

Kokku on 5 kategooria all 21 kontrollküsimumst, millele saab veebilehe kohta üldjuhul anda vastuse kolmel tasemel: miinimumtase, 2. tase või 3. tase. Kui kontrollküsimumses mainitud tehnikat veebirakendus juures ei kasutata, märgitakse see eraldi ära ning vastus ei lähe lõpphindamisel arvesse. Tervikuna võimaldab selline klassifikatsioon jagada veebilehed viide kohandustasemesse (*conformance level*):

- miinimumtase – kõik kontrollküsimumsed rahuldatud miinimumtasemel;
- tase 1+ – paljud kuid mitte kõik kontrollküsimumsed lahendatud tasemel 2;
- tase 2 – kõik kontrollküsimumsed lahendatud tasemel 2;
- tase 2+ – paljud, kuid mitte kõik kontrollküsimumsed lahendatud tasemel 3;
- tase 3 – kõik kontrollküsimumsed lahendatud tasemel 3.

W3C ei paku veel hetkel automaatseid vahendeid WCAG kontrollküsimumstele vastavuse testimiseks, kuid WCAG 1.0 testi on võimalik sooritada aadressil <http://bobby.watchfire.com/>. Aadressil <http://colorfilter.wickline.org/> on testimismootor, mis konverteerib lehekülje sellisele kujule, nagu seda näevad värvipimedad.

3 METAFOORIDEL PÕHINEV KASUTAJALIIDESE DISAIN

Mõiste metafoor tuleb kreekakeelsest sõnast *metaphora*, mis tähendab ülekannet. Võõrsõnade leksikon defineerib metafoori mõiste järgmiselt: "ühe sõna või väljendi tähenduse teisele ülekandmine sarnasuse alusel". Metafooride teemat on põhjalikult käsitlenud Lakoff ja Johnson, kes sõnastavad metafoori lahti kui "ühe objekti mõistmise ja kogemise teise terminites" (Lakoff, Johnson, 1980, lk 5).

Kui inimesed puutuvad esimest korda kokku nende jaoks võõra tehnoloogiaga, näiteks personaalarvutiga, siis püüavad inimesed võrrelda seda nende jaoks juba tuttavate seadmetega, leida metafoore. Näiteks esmakordselt tekstitöötlusprogrammi kasutamisel leiavad inimesed selles palju sarnasusi (ja ka erinevusi) trükimasina kasutamisega. Metafoorid, juhul kui neid rakendatakse sobivalt, võimaldavad kiiremat arusaamist ning vähendavad sellega uue vahendi õppimiseks kuluvat aega (Preece, 1996).

Cates (2002) toob välja Milleri põhjal välja metafoori mõistmise kolm faasi:

- äratundmine (*recognition*) – me tajume objekti suhte või võrdlusena, mitte otseselt;
- taasloomine (*reconstruction*) – loome enda jaoks pildi tajutavast objektist võrdlusena kasutatud objekti abil;
- tõlgendamine (*interpretation*) – mõistame, kuidas tajutava objekti omadused on muutunud.

Siinjuures ei saa lahutada taasloomist ja tõlgendamist, sest esineb palju erinevaid taasloomise võimalusi, millest osa sobib antud kontekstis ja osa mitte.

Tarkvara loojad räägivad sageli "õige metafoori leidmisest", millele võiks tugineda kasutajaliides. Metafoor on oma olemuselt eelkõige keeleline vahend ning seda on rakendatud juba mittegraafilise kasutajaliidese perioodil, näiteks failipuu. Laiemalt hakati metafoore kasutama graafilise kasutajaliidese tulekul, kusjuures paljud metafoorid esitati visuaalselt nuppude, ikoonide jms näol. Graafilise kasutajaliidese ideed pärinevad juba 1950-ndatest aastatest, kuid esmakordselt rakendati neid Xeroxi poolt tööjaamades Alto (1972) ja Star (1981). Xeroxi idee baasil tõi Apple 1983. aastal turule esimese graafilise kasutajaliidese personaalarvuti Lisa, mida saatis

suur edu. Sellele järgnes 1984. aastal Apple Macintosh (Sanford, 2003). Microsoft Windowsi esimene versioon ilmus müügile aastal 1985 (Lessard, 2002).

Kõiki eelnevalt mainitud graafilise kasutajaliidesega rakendusi iseloomustab töölaua metafoori kasutamine. Töölaual paiknevaid faile kujutati ikoonidena. Graafiline kasutajaliides võimaldab dokumentide avamist, sulgemist, kopeerimist ja kustutamist hiirega klikkimise, viitamise, märkimise, liigutamise ja lohistamise abil.

Virtuaalse kasutajaliides metafooride ja verbaalsete metafooride erinevuseks on see, et esimene on osaks kasutajaliideses. Verbaalne metafoor kutsub kasutajat nägema sarnasusi ja erinevusi süsteemi ja kasutajale tuttava ala vahel, kasutajaliides metafoor ühendab süsteemi ja kasutajale tuttavad mõisted üheks tervikuks. Töölaua metafoor on nagu kontori töölaud, kuid see on ka süsteemi kasutajaliides. Selle tulemusena arendavad kasutajad süsteemi kohta oma mõistemudeli, mis toetub pigem metafooridele, kui täpsele teadmisele, kuidas süsteem töötab.

Enamasti rakendatakse veebirakenduste kasutajaliideses liitmetafoore (*composite metaphors*). Näiteks töölaud on liitmetafoor – töölaud on alusmetafooriks ning sellele lisanduvad objektid on abimetafoorideks. Täiesti iseseisvateks töölauaga mitteseotud alusmetafoorideks võib pidada aknaid ja menüüsid. Kognitiivsest vaatevinklist võib näida, et inimestel tekkivad raskused liitmetafooride interpreteerimisel. Sellegipoolest ühendavad inimesed enamikul juhtudel erinevad mõisted ning arendavad välja mitmekordsed mõttemudelid (*multiple mental models*).

Ebaõnnestunult disainitud liitmetafoorid võivad põhjustada kasutajatele kontseptuaalseid probleeme. Levinumaks probleemiks on see, et kasutaja eelnevate kogemuste põhjal väljakujunenud ootused kasutajaliides objekti funktsionaalsuse suhtes ei lange kokku objekti tegeliku funktsionaalsusega (Preece, 1996, lk 147). Üheks näiteks võib siinjuures tuua disketi väljastamise Macintoshi keskkonnas.

Sobiva metafoori valimisel soovitab Cates (2002) rakendada POPITS mudelit. POPITS on akronüüm sõnadest *properties* (omadused), *operations* (toimingud), *phrases* (fraasid), *images* (pildid), *types* (liigid) ja *sounds* (helid). Antud mudel aitab leida alusmetafooriga sobivad abimetafoorid. Juhul kui võtame alusmetafooriks raamatu, märgime üles sellega seonduvad omadused (õhuke, paks, kõvade kaantega, pealkiri, autor, ...), toimingud (riiulist võtmine, avamine, lehitsemine, lugemine, järjehoidja kasutamine, märkuste tegemine, ...), fraasid (loe häälega, loe vaikselt,

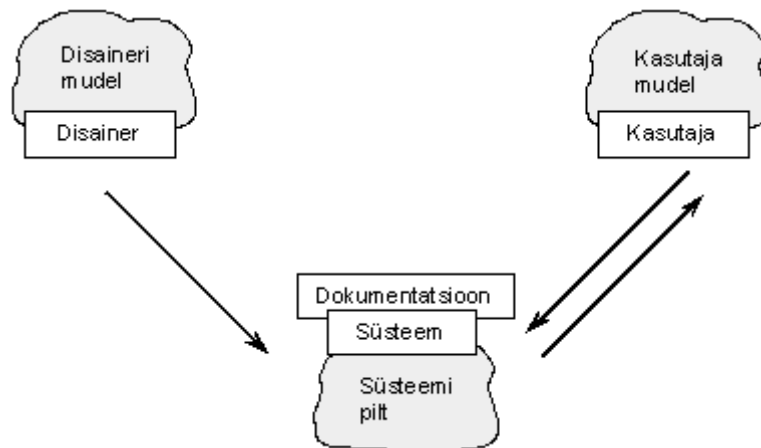
keera lehte, tee märkusi, ...), kujutluspildid (kõvade kaantega paks raamat, õhuke kulunud raamat, raamatud riulis), raamatute liigid (sõnaraamatud, entsüklopeediad, atlased, käsiraamatud, õpikud, ilukirjandus...) ja raamatu kasutamisega seonduvad helid (lehekülje keeramine, kiire lehitsemine, raamatu sulgemine, ...). Need moodustavad sobiva materjali abimetafooride valikuks.

Töölaua kõrval on ka teisi sagelikasutatavaid metafoore, mis on esitatud tabelis 3.1.

Tabel 3.1. Sagedamini rakendatavad kasutajaliidese metafoorid (Preece, 1996, lk 149)

Rakendusvaldkond	Metafoor	Lähedane seos
Operatsioonisüsteem	Töölaud	Kontoritöö ülesanded
Hüpertekst	Märkmekaardid	Paindlikult korrastatud struktureeritud tekst
Õpikeskkonnad	Reisimine	Reisid, giidid, navigeerumine
Failihoidlad	Virnad	Erinevate atribuutide järgi kategooriatesse jaotatud objektid
Multimeediumipõhised keskkonnad	Ruumid	Ehitiste struktuur

Seda, kuidas süsteemi tööd mõistavad erinevad inimesed, kirjeldatakse kontseptuaalsete mudelite abil. Enamasti võib need jaotada kaheks – kuidas kasutajad näevad ja mõistavad süsteemi ning kuidas süsteemi disainerid näevad ja mõistavad süsteemi. Kuna reeglina töötavad disainerid rühmades, ühendab disaineri mudel endas erinevate disainerite nägemuse süsteemi tööst. Ideaaljuhul peaksid disaineri ja kasutaja mudelid kattuma - siis saavad kasutajad rakendada süsteemi loojate poolt kavandatud funktsionaalsusi täies mahus. Kasutajad peavad mõistma disaineri mudelit läbi kasutajaliidese, selle toimimise ning süsteemi dokumentatsiooni. Sellest moodustub kasutaja jaoks süsteemi pilt (*system image*) (vt joonis 3.1).



Joonis 3.1. Disaineri mudel, kasutaja mudel ja süsteemi pilt (Preece, 1996, lk 152)

Kasutajaliidese visuaalsete metafooride disaini mõjutavaid faktoreid on käsitlenud Huang, Shieh ja Chi (2002). Nad toovad välja järgmised ikoonide laialdase kasutamise põhjused:

- ikoone on lihtne ära tunda ja meelde jätta;
- pildid on universaalsemalt äratuntavad kui tekst, sest puuduvad keelebarjäärid;
- ikoonid pakuvad visuaalselt tajutavaid vihjeid objekti funktsionaalsuste kohta ning hõlbustavad sellega inimese ja masina vahelist interaktsiooni;
- kasutajad eelistavad ikoone tekstile, kuigi nende töö tulemuslikkus ei pruugi selle läbi paraneda ega ka halveneda.

Kasutajaliideste liigset tuginemist metafooridele kritiseerib Alan Cooper (1995). Cooperi järgi hõlbustavad metafoorid küll natuke keskkonna esmakordset kasutama õppimist, kuid piiravad tarkvara hilisemaid laiendusvõimalusi: erinevaid metafoore pole lihtsalt piisavalt, samuti on probleemiks nende skaleeritavus (tuhat faili tähendab tuhandet ikooni) ning kasutajate võime metafoore õigesti ära tunda. Ta väidab, et Macintoshi kasutajaliidese edu taga ei olnud mitte metafoorid, vaid piiratud hulk kasutajaliidese termineid ning hiirega teostatavaid operatsioone. *"Metafoorid olid kõigest ilusateks maalideks hästidisainitud maja seintel."*

Cooper toob välja kolm kasutajaliidese disaini paradigmat, millest tema järgi tuleks selgelt eelistada idiomaatilist paradigmat metafoorilisele:

- **Tehnoloogiline paradigma.** Kasutajaliides väljendab üks-ühele programmi ülesehitust, iga nupu taga on nähtav konkreetne funktsionaalsus. Programmi kasutamiseks tuleb mõista, kuidas see töötab.

- **Metafooriline paradigma.** Kasutajaliides baseerub metafooridel. See on samm edasi tehnoloogilisest paradigmast, kuid siiski liialt piiratud.
- **Idiomaatiline paradigma.** Idioomideks on kasutajaliideses näiteks märkeruudud, kerimisribad jms, mis ei oma enamasti mingit metafoorilist tähendust. Inimene õpib idioomi tundma ja jätab meelde väga lihtsalt, sest ta ei pea seejuures läbima metafoori mõistmise faase ega teadma, kuidas programm töötab. Reaalse maailma idioomide näitena võib tuua paljud kaubamärgid, mis ei oma keeles otsest tähendust ning visuaalse idioomina näiteks viis olümpiarõngast. Tegelikult võib isegi akent pidada enam idioomiks kui metafooriks – reaalses maailmas ei saa me ju akna suurust muuta.

Sedalaadi kriitika on tekitanud uurijates küsimusi, kas metafooride rakendamine annab soovitud efekti või mitte. Elissa Smilowitz tõestas oma uurimuses, et metafooridel põhinevas kasutajaliideses tegid katsealused vähem vigu, kui metafoorideta liideses (Smilowitz, 1996). Sama küsimus vajab siiski uurimist ka IVA kasutajaliidese puhul.

4 IVA PROJEKT ÕPIHALDUSSÜSTEEMI LOOMISEKS TPÜ-LE

TPÜ-s on veebipõhiseid õpihaldussüsteeme rakendatud alates 1998. aastast, mil võeti kasutusele WebCT. See on laialdaselt kasutatav Kanada/USA päritolu kommertstarkvara, mida Eestis kasutavad veel TÜ, TTÜ, IT Kolledž ja EBS. Algselt Briti-Kolumbia ülikoolis välja arendatud WebCT tootmine läks hiljem edasi ühele USA tarkvarafirmale ning seoses sellega hakkas litsentsi hind aina tõusma (hetkel odavaim litsents 7000 USD aastas asutuse kohta). Samaaegselt kommertssüsteemide hinnatõusuga on tekkinud hulk väikeseid tasuta ja avatud lähtekoodiga õpihaldussüsteeme, mis ei paku küll kõiki suurte kommertssüsteemide funktsionaalsusi (integratsioon ülikooli õpiinfosüsteemiga jne), kuid on sageli pedagoogiliselt hulga innovatiivsemad.

Viimastel aastatel on TPÜ haridustehnoloogia keskuse juurde moodustunud e-õppe rakendustest huvitatud üliõpilaste, õppejõudude ja teadurite ring, kes on katsetanud tervet hulka tasuta õpihaldussüsteeme. Selle tulemusena on leitud arvestatavaid alternatiive kasutamiseks WebCT-laadsete kommertssüsteemide kõrval või asemel. 2001/02 õppeaastast võeti peamise e-õppe platvormina kasutusele LearnLoop, selle kõrval leidsid rakendamist ka FLE3, VIKO ja mitmed rühmatöökeskkonnad.

29. oktoobril 2001 kinnitati TPÜ nõukogus E-ülikooli projekt, mille eesmärgiks on välja arendada ja rakendada terviklik ülikooli põhitegevusi toetav elektrooniline tugisüsteem. Projekti alameesmärgiks oli tõsta ülikooli õppejõudude ja teadurite haridustehnoloogilisi oskusi ja teadmisi ning saavutada seeläbi haridustehnoloogia meetodite laiaulatuslik rakendamine õppetöös. Uute õppemeetodite rakendamise eelduseks on sobiva e-õppe platvormi olemasolu - seetõttu otsustati loobuda pedagoogiliselt aegunud ning pidevalt tõusva litsentsitasuga WebCT-st ja luua ülikooli oma veebipõhise õpihaldussüsteemi loomine juuniks 2003 (TPÜ, 2001).

2002. aasta jaanuaris loodi haridustehnoloogia keskuse, informaatika osakonna ja infoteaduste osakonna vaheline suure potentsiaaliga interdistsiplinaarne töörühm, mille eesmärgiks seati ülikooli oma veebipõhise õpihaldussüsteemi arendus ja sellega seonduv uurimistegevus. Loodav õpihaldussüsteem sai nimeks IVA (Interaktiivne VirtuaalAkadeemia) ning selle arendust rahastas Haridusministeerium.

Lähtudes soovist luua nii pedagoogilises kui tehnilises mõttes innovaatiline lahendus võeti projekti algfaasis vastu kolm strateegilist otsust, mis tagasid projekti õigeaegse ja eduka täitmise:

- IVA peab jääma avatud lähtekoodiga vabavaraks, võimaldamaks asjahuvilistel väljastpoolt TPÜ-d ja Eestit ühineda arendustegevusega;
- Kuna meid huvitab uute ideede katsetamine ja mitte tootearendus, siis pole mõttekas programmeerida kõike ise. Selle asemel tuleks kasutada rakendusserverit Zope ning luua vähemasti IVA esimene versioon olemasolevate Zope-põhiste vabavara-moodulite baasil – neid siis vastavalt meie kontseptsioonile modifitseerides ja täiendades;
- Tarkvaraarenduse kiirendamiseks tuleks kasutada ekstreemprogrammeerimise meetodeid (sprindid, paarisprogrammeerimine, prototüübid).

Siinjuures põhjustas enim vaidlusi arendusplatvormi valik. Levinuimaks veebirakenduse loomise platvormiks on Eestis kahtlemata PHP/MySQL ning ka meie arendusrühma liikmetel oli seljataga antud platvormil koostatud tarkvaraprojekte. PHP ja Zope võrdlemisel tuleb arvestada, et tegemist on erineva taseme vahenditega - üks on serveripoolne skriptimiskeel, teine täismahus rakendusserver. Zope oli kõigi jaoks suhteliselt uus platvorm, kuid sel olid mitmed eelised PHP ees:

- sissehitatud korralik ja paindlik kasutajahaldus;
- äriloogika, andmed ja kujundus selgelt ja lahus, seetõttu on igaüks eraldi kerge vaevaga uuendatav;
- sissehitatud tugi kaasaegsetele metadata-standarditele, XML eksport ja import;
- tugi paindlikule lokaliseerimisele.

Lõplikult otsustasime Zope kasuks pärast augustis 2003 toimunud ühisseminari FLE3 (*Future Learning Environment*) loojatega, kui olime saanud põhjaliku ülevaate FLE3 aluseks olevatest pedagoogilistest ideedest, süsteemi ülesehitusest, funktsionaalsustest ja eripäradest. See on üks innovatiivsema pedagoogilise lähenemisega õpihaldussüsteeme, mille ülesehitus on algusest peale rangelt lähtunud progressiivse uuringu (*progressive inquiry*) teooriast (Leinonen et al, 2002) ning on sobiv meie pedagoogilise kontseptsiooniga (käsitletakse peatükis 5.2). Ühest küljest muudab see FLE3 unikaalseks ja pedagoogilises mõttes huvitavaks, kuid teisest küljest takistab

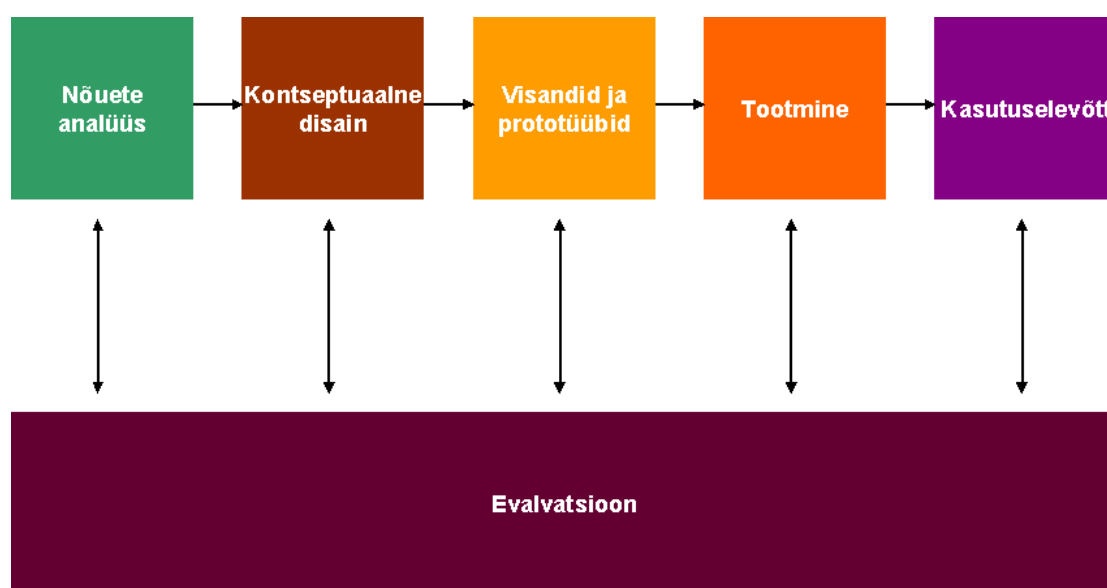
jäik seotus Eestis veel suhteliselt vähelevinud õppimiskäsitlusega selle rakendamist. Leidsime, et GPL litsentsi alusel jagatava FLE3 koodi aluseks võttes on võimalik arendada meile sobiv süsteem. Zope võimaldas meil ehitada FLE3 peale oma kasutajaliides, muuta keskkonna struktuuri, kuid säilitada suur osa algsest koodist. Praeguseks on siiski erinevate muudatuste ja täienduste tulemusena meiepoolse koodi osakaal juba umbes 50%.

IVA esimest kasutuskõlblikku prototüüpi (versioon 0.1) esitleti avalikkusele 24. jaanuaril 2003. Teine, juba töökindlam ja kakskeelne versioon valmis Eesti e-ülikooli avakonverentsiks 21. veebruaril. 2003. aasta kevadsemestril käib IVA piloottestimine, käivitunud on õppejõudude koolitus ning loodud üle 40 kursuse, millest kümmekond on reaalselt toimuvad e-kursused ning ülejäänud õppejõudude poolsed IVA-katsetused. Lisaks TPÜ-le testivad IVA ka Linzi pedagoogikaakadeemia Austriast, Sheffield Hallami ülikool ning Politseikool ja IT Kolledž Eestist. Samuti toimuvad IVA keskkonnas kooliõpetajatele suunatud ainealased DigiDidaktika kursused. IVA paikneb aadressil <http://iva.tpu.ee/IVA/>.

Praegust FLE3 baasil arendatud IVA versiooni vaatleme me esimese omapoolse katsetusena. Lisaks FLE3-le kasutame ära ka mitmeid teisi Zope platvormil tarkvaratooteid nagu näiteks Plone ja ZWiki, seetõttu kutsume arendusmeeskonnas praegust versiooni Frankensteiniks. Plaanis on pärast stabiilse ja vigadeta IVA versiooni valmimist ning laiema vabatahtliku arendusmeeskonna tekkimist alustada nullist päris oma koodil baseeruva õpihaldussüsteemi loomist. Selleks oleme liitunud vaba tarkvara arendusprojektide kommuuniga Savannah, mille all asub IVA installifailid, CVS lähtekoodiga, veakoguja, aruteluforum jpm (vt <http://savannah.nongnu.org/projects/iva/>).

5 ÕPIHALDUSSÜSTEEMI IVA KASUTAJALIIDESE DISAIN

Kasutajaliidese disainiprotsessi aluseks võtame Brinck, Gergle ja Wood (2002) poolt välja pakutud kõikehõlmava kasutatavuse protsessi mudeli (*Pervasive Usability Process*). Antud mudelis on kasutatavus integreeritud kõikidesse tarkvaraarenduse etappidesse. Disainiprotsess jaguneb viide etappi, millega paralleelselt toimub evalvatsioon (vt joonis 5.1). Antud protsessi puhul kaasatakse disainiprotsessi süsteemi tulevasi kasutajaid võimalikult varajastest etappidest alates. Tagasiside reaalsetelt kasutajatelt on väga väärtuslik informatsioon.



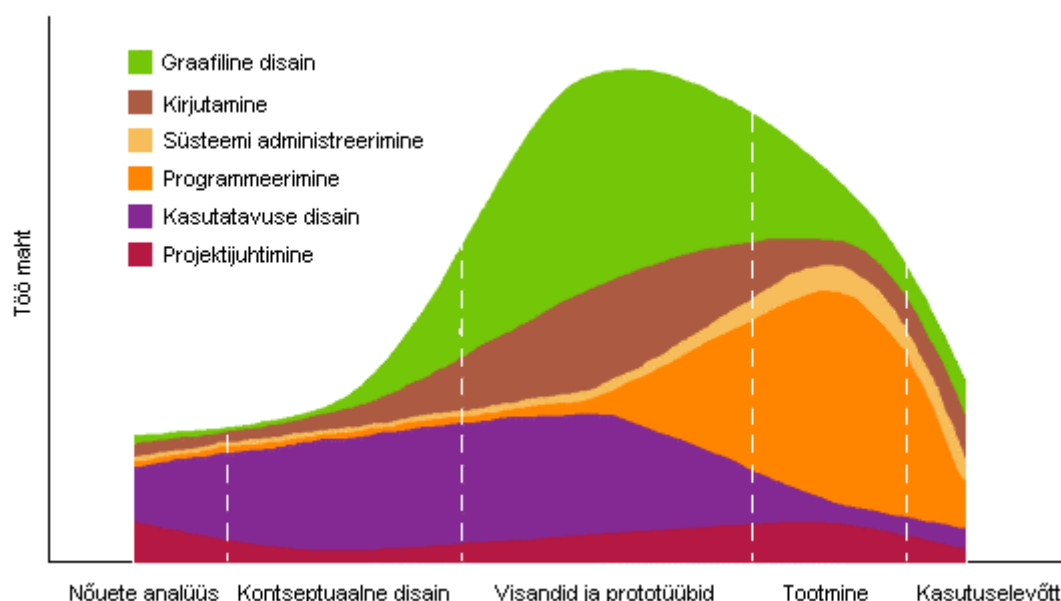
Joonis 5.1. Kasutajaliidese disainiprotsess (Brinck et al, 2002, lk 16)

Seda mudelit võib vajadusel vastavalt projektile kohandada. Näiteks on küllalt tavaline, et eiratakse mudeli lineaarset kulgu nõuete analüüsist avaldamiseni ning pöörduakse visandite ja prototüüpide juurest tagasi nõuete analüüsi faasi, et seal koostatud nõuete spetsifikatsiooni parandada ja täiendada. Disainiprotsess on iteratiivne – seda võib vaadata spiraalse liikumisena, mille igas faasis läbitakse ka evalvatsioon.

Igal disainiprotsessi etapil rakendatakse kindlaid kasutatavuse meetodeid, millest osa on vabatahtlikud, teine osa aga kohustuslikud. Projekti algfaasis otsustasime viia IVA arendusprotsess läbi vastavalt joonisel 5.1 toodud mudelile. Tagantjärele hinnates see meil mõningaste täienduste ja muudatustega ka õnnestus. Käesoleva peatüki

alampeatükkides tutvustamegi IVA disainiprotsessi erinevaid etappe ning rakendatud kasutatavuse meetodeid.

Kasutajaliidese disaini seisukohalt on olulisemateks etappideks nõuete analüüs, kontseptuaalne disain ning visandite ja prototüüpide etapp. Hiljem on tähtsamal kohal graafiline disain ja programmeerimine. Brinck, Gergle ja Wood esitavad tööjõuvajaduse graafiku veebisaitide loomisel (vt joonis 5.2). Põhimõtteliselt võib selle üle kanda ka IVA-laadsete veebirakenduste loomisele, ainult kirjutamise osakaal on IVA puhul tunduvalt väiksem.



Joonis 5.2. Tööjõuvajadus erinevatel disainiprotsessi etappidel (Brinck et al, 2002, lk 27)

5.1 Nõuete analüüs

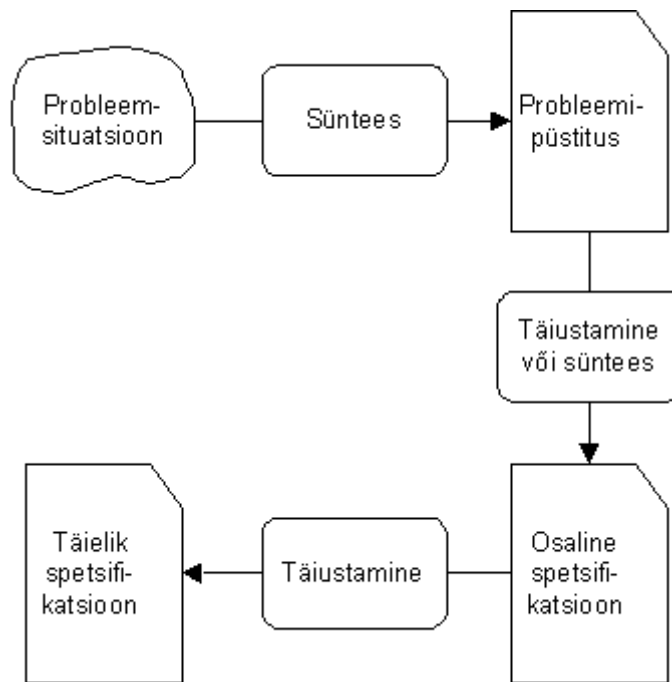
Nõuete analüüsi etapil püstitatakse disainiülesanded ning määratletakse sihtgrupp, tarkvaraplatvorm, tehnilised nõuded jne. Evalvatsiooni osa hõlmab siin olemasolevate analoogiliste rakenduste võrdlevat analüüsi, kasutajaintervjuusid ja -küsitlusi. Antud etappi tuleb pidada disainiprotsessi kõige olulisemaks, sest siin langetatud otsused määravad kogu projekti edukuse. Nõuete analüüsi etapil rakendatavad kasutatavuse meetodid on esitatud tabelis 5.1. IVA projekti käigus koostati nõuete spetsifikatsioon ja kasutajastsenaariumid ning viidi läbi olemasolevate õpihaldussüsteemide võrdlev analüüs.

Tabel 5.1. Nõuete analüüsi etapil rakendatavad kasutatavuse meetodid (Brinck et al, 2002, lk 32)

Meetod	Nõutav	Kestvus	Õppimiseks kuluv aeg	Usaldusväärsuse tase	Mõju lõplikule disainile
Nõuete spetsifikatsioon	jah	lühike	lühike	madal	kõrge
Kasutajaintervjuud	ei	keskmine	keskmine	keskmine	kõrge
Kasutajate küsitlus	ei	pikk	keskmine-pikk	kõrge	keskmine
Võrdlev analüüs	ei	lühike	lühike	madal	madal

5.1.1 Nõuete spetsifikatsioon

Nõuete spetsifikatsioon on iga tarkvaraprojekti kohustuslik osa. Selle koostamise eesmärgiks on saada võimalikult täielik ülevaade tarkvarale esitavatest nõuetest enne kodeerimise algust. Nõuete spetsifikatsiooni loomisesse on oluline kaasata kõik projektirühma liikmed. Nõuete spetsifikatsiooni väljatöötamine on järkjärguline protsess, mille käigus probleemi tajumisest mitmekordse sünteesi ja laiendamise käigus valmib täielik spetsifikatsioon. Newman ja Lamming (1995) toovad välja järgmised etapid, mis spetsifikatsiooni väljatöötamisel läbitakse (vt joonis 5.3). Ellen Gottesdiener soovib nõuete spetsifikatsiooni väljatöötamiseks spetsiaalse seminaride seeria läbiviimist (Gottesdiener, 2001). Ka IVA projekti algfaasis rakendati peamise töömeetodina seminare.



Joonis 5.3. Sammud nõuete spetsifikatsiooni koostamiseks (Newman, Lamming, 1995, lk 79)

James ja Suzanne Robertson esitavad *Volere* malli, mille põhjal töötada välja nõuete spetsifikatsioon. Selle järgi jagunevad nõuded 27 erinevasse tüüpi, mis omakorda liigitatakse viide kategooriasse (Robertson, Robertson, 2002):

- projekti suunajad (toote eesmärk, kliendid, kasutajad);
- projekti piirajad (piirangud seoses tarkvaraplatvormiga, ajaga jne);
- funktsionaalsed nõuded (mõistekaardid, kasutusjuhud);
- mittefunktsionaalsed nõuded (väljanägemine, kasutatavus, hallatavus, turvalisus, kultuurilised, poliitilised ja õiguslikud nõuded);
- projekti nõuded (maksumus, riskid, dokumenteerimine, kasutajate koolitus).

Monumentaalmetoodikate puhul peeti oluliseks võimalikult täpset ja detailset nõuete spetsifikatsiooni, ning hinnati hilisemaid ebatäpsest spetsifikatsioonist tingitud koodimuudatusi kulukateks. IVA-suuruste projektide puhul on sobivamateks agiilmetoodikad nagu ekstreemprogrammeerimine. Nende puhul ei ületähtsustata nõuete spetsifikatsiooni ning peetakse selle hilisemat korduvat muutmist tavaliseks. Samuti on tänapäevaste tarkvaraarendusplatvormide nagu Zope puhul hilisem koodimuudatuste sisseviimine suhteliselt valutult teostatav.

Esialgse IVA spetsifikatsiooni koostamisel ei ole järgitud *Volere* malli, kuid spetsifikatsiooni on hiljem vastavalt süsteemi arengule muudetud ja täiendatud ning ühtlasi viidud vastavusse *Volere* malliga. Tabelis 5.2 esitame IVA nõuete spetsifikatsiooni. Kõik nõuded on nummerdatud ning iga nõude juures on ka *Volere* malli tüüp, millele antud nõue vastab (vt lisa 1).

Tabel 5.2. IVA nõuete spetsifikatsioon

#1	(1)	Meie eesmärgiks on luua innovaatilise pedagoogilise kontseptsiooniga vabavarana levitavad õpihaldussüsteem Tallinna Pedagoogikaülikoolile, Eesti E-ülikoolile ja teistele süsteemist huvitatud institutsioonidele;
#2	(4)	Süsteem arendatakse Zope rakendusserveril olemasolevate vabavaramoodulite põhjal;
#3	(4)	Süsteemi esimese avaliku versiooni valmimistähtaeg on jaanuar 2003;
#4	(9)	Süsteem toetab kursuste eksporti ja importi vastavalt õpiobjektide metadata standarditele;
#5	(10)	Süsteemi kasutajaliides tugineb verbaalsetele ja visuaalsetele metafooridele;
#6	(10)	Süsteemi kasutajaliides peab nägema välja võimalikult identne alates brauseritest Internet Explorer 5.0, Netscape Navigator 6.1, Opera 5.02 ja Mozilla 1.0;
#7	(10)	Tagamaks püsiv paigutus ka nõrga CSS'i toetusega brauseritel tuleb objektide paigutamiseks ekraanil kasutada tabelleid, mitte CSS'i;
#8	(10)	216 värvi standardit järgida paindlikult: (a) tekstid ja lehekülje taust kasutagu 216 veebi värvipaleti värvi, (b) väiksemad taustad kasutagu soovituslikult laiendatud veebi värvipaletti (1536 tooni), (c) graafika puhul võib standardsete värvide kasutamisest vajadusel loobuda;
#9	(11)	Süsteemi kasutajaliidese disainimisel tuleb arvestada W3C soovitusi puutega inimestele disainimisel (WCAG);
#10	(11)	Õpihaldussüsteemi kasutamine peab olema tudengitele intuitiivselt õpitav;
#11	(11)	Õpihaldussüsteemi peab olema integreeritud piisav abiinfo;
#12	(12)	Süsteem peab olema piisavalt skaleeritav kasutamiseks TPÜ ulatuses;

- #13 (13) Õpiahaldussüsteemi poolt genereeritud leheküljed peavad valideeruma W3C soovitusel XHTML 1.0 Transitional järgi;
- #14 (13) Lehekülgede kujunduse kirjeldamiseks kasutatud CSS peab valideeruma vastavalt W3C soovistele;
- #15 (13) Süsteem peab olema kasutatav kõigi veebibrauseritega (ka Lynx) Windowsi, Maci ja Linuxi platvormidel;
- #16 (13) Kliendipoolsete skriptimiskeelte kasutamist tuleb võimaluse korral vältida. Vajadusel rakendatakse JavaScripti;
- #17 (16) Süsteem on tõlgitava kasutajaliidesega, eesti ja inglise keel on lisatud keskkonna loojate poolt. Osa lokaliseeritavatest komponentidest võivad olla graafilised;
- #18 (16) Süsteem peab arvestama erinevatest maadest kasutajatega (kuupäeva- ja kellaajavorming, värvide ja sümbolite kasutamine).

5.1.2 Kasutajastsenaariumid

Veebirakenduste loomisel on soovitatav koostada tulevaste näidiskasutajate kirjeldused realselt eksisteerivate inimeste põhjal. See võimaldab disainiprotsessi käigus otsuste langetamisel arvestada konkreetsete kasutajatega. Brinck, Gergle ja Wood soovivad, et kasutajastsenaarium peaks sisaldama Kasutajatüübi kirjeldus võib sisaldada kasutajaprofiili (nimi, vanus, sugu, haridus, perekonnaseis, hobid, töö, tööaeg, puuded, arvuti, monitor, ühenduskiirus, tehnilised oskused, ...), tüüpilist päevakava, ootuseid keskkonnale jms (Brinck et al, 2002). Hiljem võib kuid ei pea rakendama stsenaariumides kirjeldatud kasutajaid evalvatsioonis.

IVA loomise algfaasis kirjeldasime 9 tüüpkasutajat, seda küll mitte nii põhjalikult nagu ülalpool toodud autorid soovivad (vt tabel 5.3). Osa allpool toodud kasutajatest on kaasatud evalvatsiooni läbiviimisel. Siinjuures jätsime ekslikult ühe olulise kasutaja välja - iga suurema veebirakenduse puhul tuleb kindlasti süsteemi kasutajate hulka arvestada süsteemiadministraator. Kuigi ta ei pruugi palju kokku puutuda keskkonna kasutajaliidesega, tuleb teda silmas pidada süsteemi dokumentatsiooni (installeerimisjuhend, uuendamisjuhend, varundus- ja taastejuhend) koostamisel.

Tabel 5.3. IVA kasutajatüübid

Heli – otsese ütlemisega, nõudlik ja kohusetundlik asjaajaja, osakonna "klassijuhataja". Arvutikasutamisoskusega probleeme pole, kuid virtuaalsete töökeskkondade kasutamise kogemus on napivõitu.

Aili – tagasihoidlik ja hajameelsevõitu asjaajaja, arvutiga tuleb toime täpselt igapäevase rutiinse töö raames. Väiksemategi arvutiprobleemide puhul otsib maja pealt meesabilisi. Veebipõhise õpihaldussüsteemi kasutamise vajadus kohutab teda pisut, väga võõras asi.

Sirje K – informaatikuharidusega õppejõud, kes annab peamiselt õpetajakoolituses arvutikasutamise metoodika kursusi mitte-informaatikutele ja on selle jaoks lausa kohustatud rakendama veebipõhist õpet. Virtuaalsete töökeskkondade kasutamise kogemused (WebCT, LearnLoop, YahooGroups) on olemas, kuid mitte kuigi põhjalikud – rohkem on ta neid kasutanud õppijana kui õpetajana.

Sirje V – pikaajalise WebCT kasutamise kogemusega õppejõud ja kaugkoolituse teooria/praktika parimaid eksperte ülikoolis. Samas humanitaaria- ja mitte informaatikataustaga.

Katrin – kasvatusteaduse taustaga noor õppejõud, kes on huvitatud kaugkoolitusest. Senised kokkupuuted e-õppega pigem kirjanduslikud, vähesed virtuaalse õpikeskkonna kasutamise kogemused saadud õppijana.

Andrus – multimeediaõppejõud, kel veebiõppe kogemused piirduvad siiski peamiselt omatehtud õppematerjalide ülesriputamisega veebi tavaliste HTML-lehtede ja PDF failidena.

Kerli – andragoogikatudeng, kes on kangesti huvitatud kõigest uuest pedagoogika valdkonnas ja ei karda hüpata pea ees vette kui asi puudutab talle täiesti võõraste tehniliste vahendite kasutamist õppetöös.

Karin – klassiõpetajaks õppiv tudeng, kel on (ilmselt lastetoa mõjul) parajalt negatiivne eelhäälestus kõigesse tehnoloogilisse. Kui kõik ülejäänud rühmaliikmed esitavad essee õppejõule meelsasti Interneti vahendusel, palub tema tungivalt luba esitada see käsitsi paberile panduna (käekiri on tal muidugi ilus).

Christian – informaatikatudeng, kes on harjunud "elama Internetis".

5.1.3 Võrdlev analüüs

Erinevate õpihaldussüsteemide testimisega on TPÜ Haridustehnoloogia keskuse juures tegeletud juba mitmed aastad. Käesoleva töö autor alustas erinevate õpihaldussüsteemide testimist 2 aasta eest diplomitöö koostamise eel. Kuna antud teemat on korduvalt kirjalikult käsitletud (Põldoja, 2001; Põldoja, Laanpere, 2002; Kikkas 2002) ei esitata käesoleva magistritöö raames sellest põhjalikumalt ülevaadet.

5.2 Kontseptuaalne disain

Kui nõuete analüüsi etapis panime me paika esialgsed nõuded õpisüsteemile, siis kontseptuaalse disaini etapi eesmärgiks on analüüsida ja optimeerida protseduure, mida kasutajad järgivad veebirakendust kasutades. IVA projekti käigus olid kontseptuaalse disaini ja nõuete analüüsi etapid omavahel väga tihedalt seotud ning korduvalt pöörduti ühest etapist teise ja tagasi.

Sel perioodil oli meil mitmeid huvitavaid arutlusi inimestega, kel juba olemas kogemus ja ettekujutus veebipõhisest õpikeskkonnast. Arvamuste paljusus ja erinevus oli üllatavalt suur: kui ühe jaoks olid õpihaldussüsteemi olulisimateks funktsionaalsusteks õppematerjalide lineaarne esitamine koolitaja poolt määratud tempos (aegviitega) ja valikvastustega testide tegemise võimalus, siis teised tähtsustasid pigem koostöö- ja suhtlusvahendeid või hoopis interaktiivseid-meelelahutuslikke multimeedia-esitlusi (*edutainment*).

Selle disainietapi juurde kuuluvad kasutatavuse meetodid koos lühiiseloostusega on toodud tabelis 5.4.

Tabel 5.4. Kontseptuaalse disaini etapil rakendatavad kasutatavuse meetodid (Brinck et al, 2002)

Meetod	Nõutav	Kestvus	Õppimiseks kuluv aeg	Usaldusväärsuse tase	Mõju lõplikule disainile
Idee genereerimine	jah	lühike-keskmine	lühike	madal	kõrge
Informatsiooni arhitektuur	jah	lühike-keskmine	keskmine	madal	kõrge
Kaartide sorteerimine	ei	lühike	lühike	kõrge	keskmine
Ülesannete analüüs	ei	olenevalt projekti mahust	keskmine	keskmine	kõrge

Ajalooliselt on reaalteadustes ja tehnoloogia valdkonnas olnud suur mõju positivistlikul teadmiskäsitusel ja sellele tugineval biheivioristlikul õppimiskäsitusel. Selle formaliseeritud peegelduseks on Skinneri operantse tingituse teooria, mille kohaselt õppimine on indiviidi käitumise muutumine sündmustele (stimuli, nt. õpetajapoolne esitus) reageerimise tulemusena (Krull, 2000). Õpetaja rolliks on siin (lisaks stimuli pakkumisele) soovitatavate reaktsioonide õhutamise ja ebasoovitavate summutamine läbi relevantse (kas siis positiivse või negatiivse) tagasiside. Sedalaadi õppimise eesmärgiks on nähtu/kuuldu/loetu reprodutseerimine.

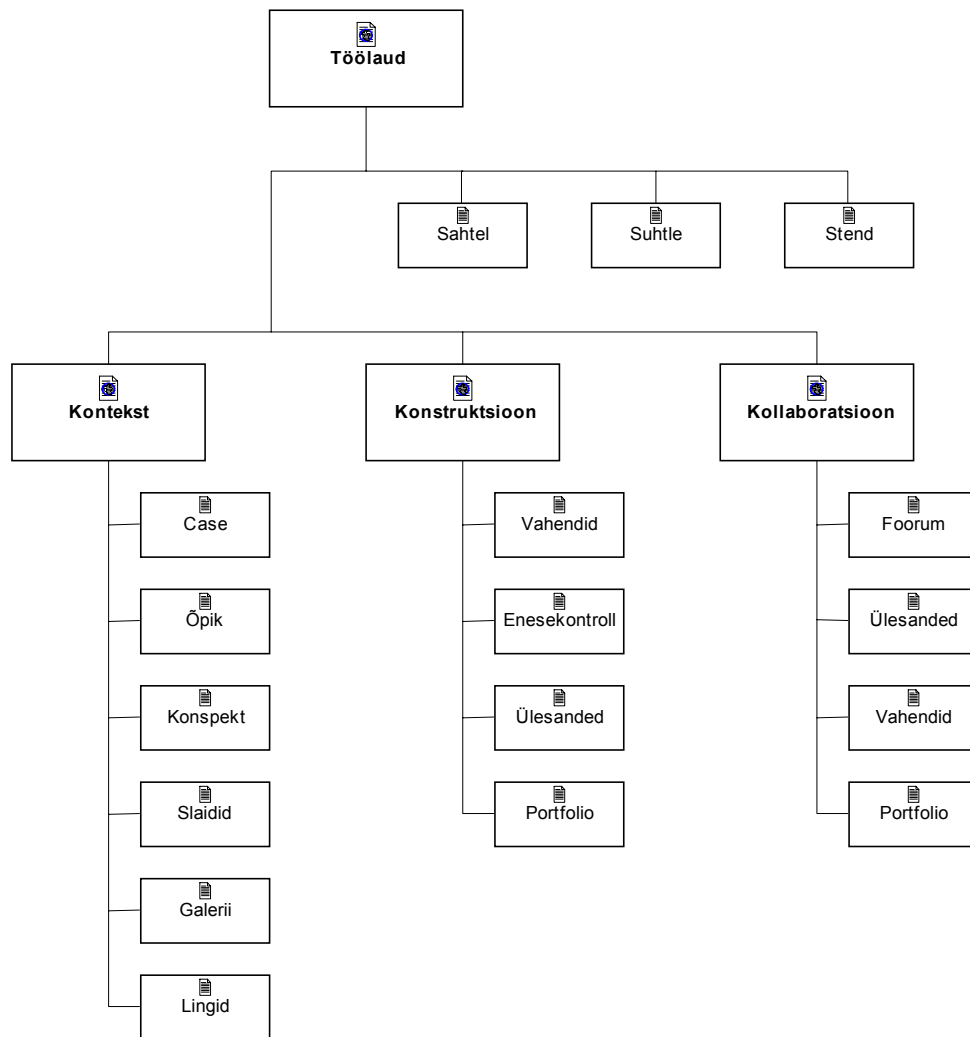
Võib julgelt väita, et enamus õpisüsteeme, mis on loodud pigem tehnilise kui pedagoogilise taustaga inimeste poolt, järgib just biheivioristlikku õppimiskäsitlust. Samas on biheivioristlik õppimiskäsitlus ja vastavad teooriad olnud juba mitu aastakümnet üha ägedama kriitikatule all. Kui 20. sajandi alguse industriaalühiskonda veel sobis sedalaadi õppimis- ja õpetamisviis, siis 21. sajandil infoajastu künnisel on ühiskonna vajadused muutunud. Üha suurem osa töötajaskonnast on nn. "teadmustöötajad" (*knowledge workers*), kellel pole tarvis mitte valmisteadmisi, vaid oskusi lahendada loovalt meeskonnatöös uudseid pretsedentideta probleeme. Reprodutseeriv õppimine ei anna õpilastele/tudengeile selliseid valmisolekuid, mistõttu on nii pedagoogikateadlaste kui praktikute tähelepanu pöördunud alternatiivsetele õppimiskäsitustele ja -meetoditele.

Biheiviorismi vastukaaluks on esile kerkinud sotsiaalkonstruktivistlik õppimisteooria. Haridustehnoloogias on üheks häälekamaks "pehme" (s.t. mitteradikaalse) konstruktivismi propageerijaks olnud D.H. Jonassen, kes oma artiklis "Thinking Technology: Toward a Constructivist Design Model" pakub välja "kolm vaala", millele ehitada konstruktivistliku õpikeskkonna kontseptuaalsed mudelit (Jonassen, 1994):

- autentne ja tähendusrikas kontekst (*Context*), mis sisaldab juhtumikirjeldusi ja probleeme elust enesest koos sellest tuleneva ebatäiuslikkuse ja keerukusega;
- õppijapoolne aktiivne ja reflektiivne tähendusloome (*Construction*), millega kaasneb seletavate, järeldavate ja ennustavate mõttemudelite (*mental models*) loomine ning nende "läbirääkimine" kaasõppuritega;
- aktiivne koostöö (*Collaboration*) ja arutelu nii õpilaste vahel kui õpetajaga, kellel on pigem nõustaja/mentori kui teadmistemoonopoli valitseja roll.

Jonassen näeb nendes kolmes C-s infoajastu pedagoogilise kompetentsi tuuma, tõmmates paralleele kolme R-iga (*Reading, wRiting, aRithmetics*), mis olid kirjaoskamatus vastu võitlemise lipukirjaks 19. sajandi lõpul. Kui toona piisas õpetaja ametis töötamiseks lugemise, kirjutamise ja rehkendamise oskustest, siis tänapäeval seisneb õpetaja professionaalsus ehk just neis Jonasseni poolt kirjeldatud "kolmes vaalas" (kirjaoskust ja oma aine tundmist tuleks võtta juba vaikimisi eeldusena).

Soovisime, et IVA ei oleks mitte komplekt vahendeid vaid kindlale pedagoogilise põhjaga kontseptsioonile toetub innovaatiline õpihaldussüsteem. Seetõttu katsetasime ehitada IVA struktuur Jonasseni 3C mudelile. Kursusest kõrgemale tasemele otsustasime paigutada iga kasutaja isikliku töölaua kasutajaprofiili, faili- ja lingihalduse, kalendri, ajaveebipõhise uudistestendi ja foorumiga. Nimetatud funktsionaalsuste realiseerimiseks kavatsesime rakendada Zope-põhist portaalitarkvara Plone. Selle põhjal valmis esimene nägemus IVA struktuurist (vt joonis 5.4).



Joonis 5.4 IVA esialgne Jonasseni 3C mudelist inspireeritud struktuur

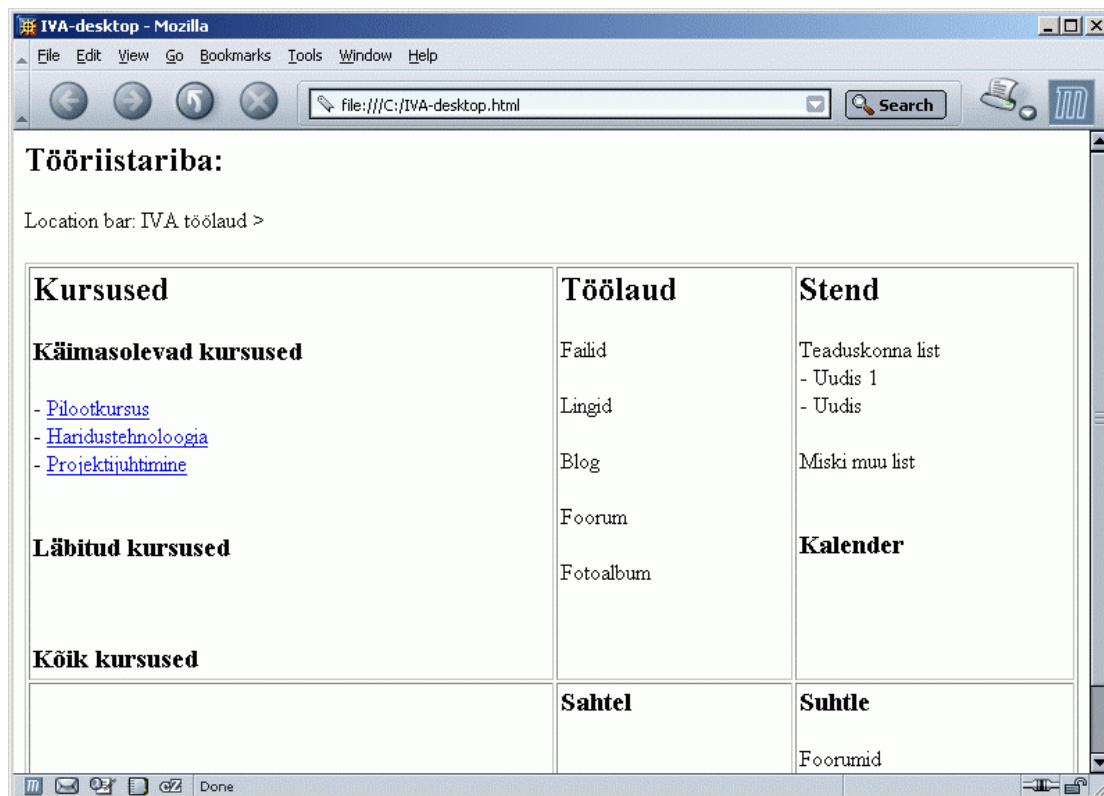
Antud joonisel puuduvad veel IVA sektsioonide õiged nimetused. Otsustasime seade IVA kasutajaliidese kujundamisel põhitähtsuseks kasutamisihtsuse ja intuiivsuse, selle saavutamiseks rakendame mitmeid kasutajale tuttavaid metafoore (nii verbaalseid kui visuaalseid). Sobivate metafooride leidmisel oli meil kaks erinevat lähenemist: kas toetuda moodsatele uudissõnadele või püüda taasleida ja -elustada vanu eesti keele sõnu. Selle tulemusena valmis meil ajurünnaku käigus kaks erinevat metafooride komplekti, millest valik on esitatud tabelis 5.5.

Tabel 5.5. IVA kasutajaliidese metafooride komplektid

Eesti 1	Eesti 2	Inglise	Semantika
Raamaturiiul	Sisu	Context	Vastavalt Jonasseni 3C mudelile
Veebilaud	Seedimine	Construction	Vastavalt Jonasseni 3C mudelile
Rühmatöö	Talgud	Collaboration	Vastavalt Jonasseni 3C mudelile
Töölaud	Pink	Webtop	Kasutaja töölaud kursustest kõrgemal tasemel
Sahtel	Laegas	Drawer	Kasutaja suletud kaust kursusel
Õpimapp	Tegemised	Portfolio	Kasutaja avalik kaust kursusel
Arvustused	Irisemine	Peer review	Kursusekaaslaste tööde arvustamine (LearnLoopi eeskujul)
Enesekonktroll	Endakatsumine	Self test	Enesehindamise testid
Kasutajainfo	Mina ise	User profile	Kasutajainfo lehekülg
Märkmik	Vihk	Notepad	Lihtne vormindamisvõimalusteta tekstitoimeti
Teated	Kõlakad	Announcements	Ülikooli ametlikud teated stendil
Töövahendid	Riistad	Tools	Erinevate vahendite üldnimetus
Veebimärkmik	Võrguvihk	Webpad	HTML'i võimaldav tekstitoimeti
Õpik	Koolitükid	Textbook	Põhjalik struktureeritud õppematerjal
Üleslaadimine	Upitamine	Upload	Faili üleslaadimine

Ülaltoodud tabelis esitatud metafoore ei ole täies ulatuses rakendatud, kuna kõik nimetatud funktsionaalsused ei ole keskkonnas realiseeritud. IVA kasutajaliidese jäi siiski peale kaasaegsem sõnavara. Osa metafoore vahetati kontseptsiooni täpsustumise käigus välja, näiteks rühmatöö asemel on töötoad ja õpimapi asemel portfoolio.

Puustruktuuri kõrval koostati arendusmeeskonna poolt ka lihtne linkidega ühendatud ilma kujunduseta HTML lehekülgedest koosnev prototüüp, nn *wireframe*, mille eesmärgiks on kavandada objektide üldist paigutust ekraanil ning navigeerumist (vt joonis 5.5).

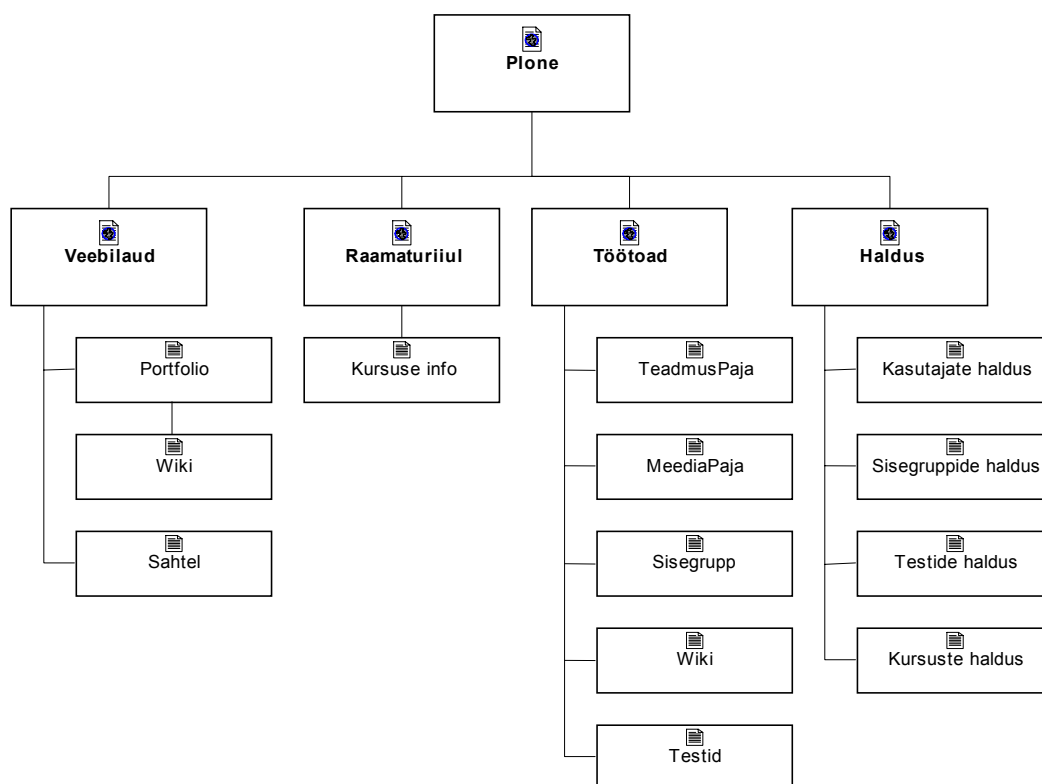


Joonis 5.5. IVA wireframe

Idee rakendada IVA struktuuri alusena Jonasseni 3C mudelit omandas täiesti uue dimensiooni pärast ühisseminari FLE3 loojatega. Saime nendelt nõusoleku kasutada oma lähteplatvormina FLE3 koodi. Selle kasutajaliides jaguneb kolmeks õpilasele nähtavaks sektsiooniks: *Webtop* (teistele kursusekaaslastele nähtav piirkond failide, linkide ja memode hoidmiseks), *Knowledge building* (teadmusloomel põhinev struktureeritud foorum) ning *Jamming* (koostööpiirkond meediafailide modifitseerimiseks). Meie projektmeeskonna poolt valmis ka FLE3 eestikeelne tõlge, milles võtsime kasutusele metafoorid Veebilaud (*Webtop*) TeadmusPaja (*Knowledge building*) ja MeediaPaja (*Jamming*). Sõna "paja" (ÕS: van. töökoda, sepapaja) sissetoomine on tekitanud vastakaid suhtumisi - kui osale kasutajatest seostub see sõna eelkõige toidukeetmise nõuga (nimetav: pada), siis teistele meeldib vana ja ununemakippuva eestikeelse sõna taaselusdamine kaasaegses kontekstis. Analoogia on olemas üha enam kasutamist leidva uudissõnaga "tööpaja" (ingl. k. *workshop*, soome k. *työpaja*).

Leidsime, et veebilaua ja raamaturiuli aluseks sobib FLE3 *Webtop* ning TeadmusPaja ja MeediaPaja on rakendatavad vahenditena töötubade sektsioonis. Sellisel puhul on võimalik FLE3 ümberkujundamise teel saada omanäoline õpihaldussüsteem. FLE3

kõrval tundusid IVA-s rakendamiseks sobivat lihtsa süntaksiga tekstipõhine sisuhaldusvahend Zwiki ning portaalitarkvara Plone. FLE3, Plone ja Zwiki liitmisel saadud struktuur on lähedane IVA tegelikule struktuurile (vt joonis 5.6).



Joonis 5.6. Õpihaldussüsteemi IVA struktuur

IVA struktuuri peamised erinevused võrreldes FLE3-ga on järgmised:

- Iga kursuse jaoks on kasutajal eraldi veebilaud;
- Veebilaud on jaotatud privaatseks Sahtliks ja avalikuks Portfoolioks;
- Veebilauale ja Töötubade sektsiooni on lisatud Wiki;
- Töötubade all on võimalik luua väiksemaid alamrühmi (sisegruppe);
- Lisatud on algupärane testimissüsteem.

5.3 Visandid ja prototüübid

Kontseptuaalse disaini etapi tulemusena valmis loodava õpihaldussüsteemi struktuur. Kolmas etapp liigub madalamale tasemele ning selle eesmärgiks on paika panna õpihaldussüsteemi lehekülgede kujundus, et need võiks programmeerimisega alustada. Sellel etapil rakendatavad kasutatavuse meetodid on esitatud tabelis 5.6.

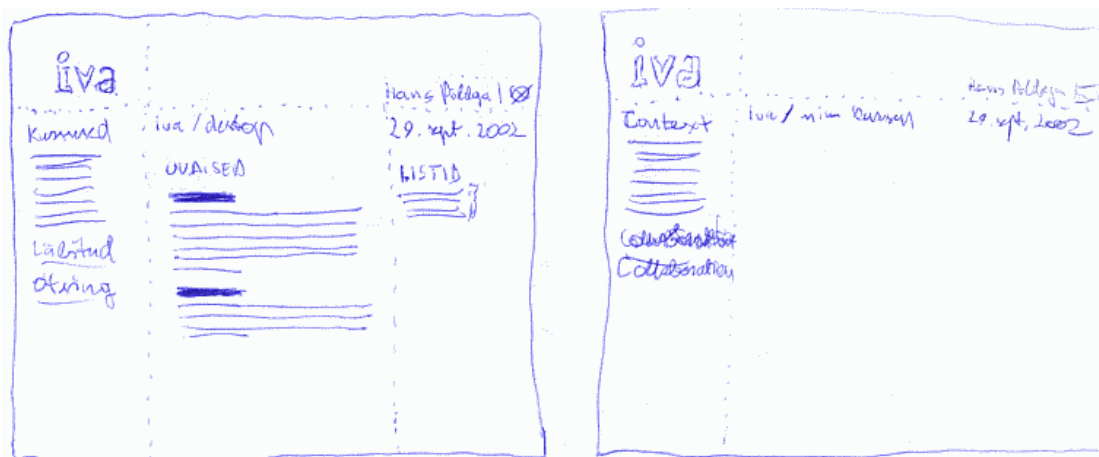
Tabel 5.6. Visandite ja prototüüpide etapil rakendatavad kasutatavuse meetodid
(Brinck et al, 2002, lk 32)

Meetod	Nõutav	Kestvus	Õppimiseks kuluv aeg	Usaldusväärsuse tase	Mõju lõplikule disainile
Visandite genereerimine	Jah	keskmine-pikk	pikk	madal	Kõrge
Grupi tagasiside	Ei	lühike	keskmine	kõrge	keskmine
Kasutajatega testimine	Ei	lühike	keskmine	keskmine	keskmine

Visandamine algab lihtsate lehekülje struktuuri kujutavate pisipiltide skitseerimisest. Ühe pisipildi peale kulub kõigest 15...20 sekundit, eesmärgiks pole detailsus vaid erinevate paigutusiideede kiire genereerimine ja talletamine. Kui poole tunni jooksul tuleb igas minutis üks idee, on saadud piisavalt algmaterjali, millest edasi minna.

Sobivatest pisipiltidest arendatakse edasi paberprototüübid. Need on juba samas mõõtkavas kui lõppkujundus arvutiekraanil ning nende joonistamisel kasutatakse värve. Paberprototüüpide puhul keskendub tähelepanu üldmuljele, mitte detailidele, nende valmistamine on kiire ning neid on võimalik igale poole kaasa võtta. Samas on keeruline mitmete analoogilise kujundusega kuid detailide tasemel mõnevõrra erinevate (näiteks erinev värvilahendus) paberprototüüpide valmistamine.

IVA puhul detailseid paberprototüüpe ei valmistatud, kuid õnnestunud paigutusega visandid joonistati uuesti ja detailsemalt ümber. Joonisel 5.7 on toodud näide kasutajaliidese visandist, millel on kujutatud Veebilauda. Hiljem valmis selle visandi põhjal ka teine HTML prototüüp.

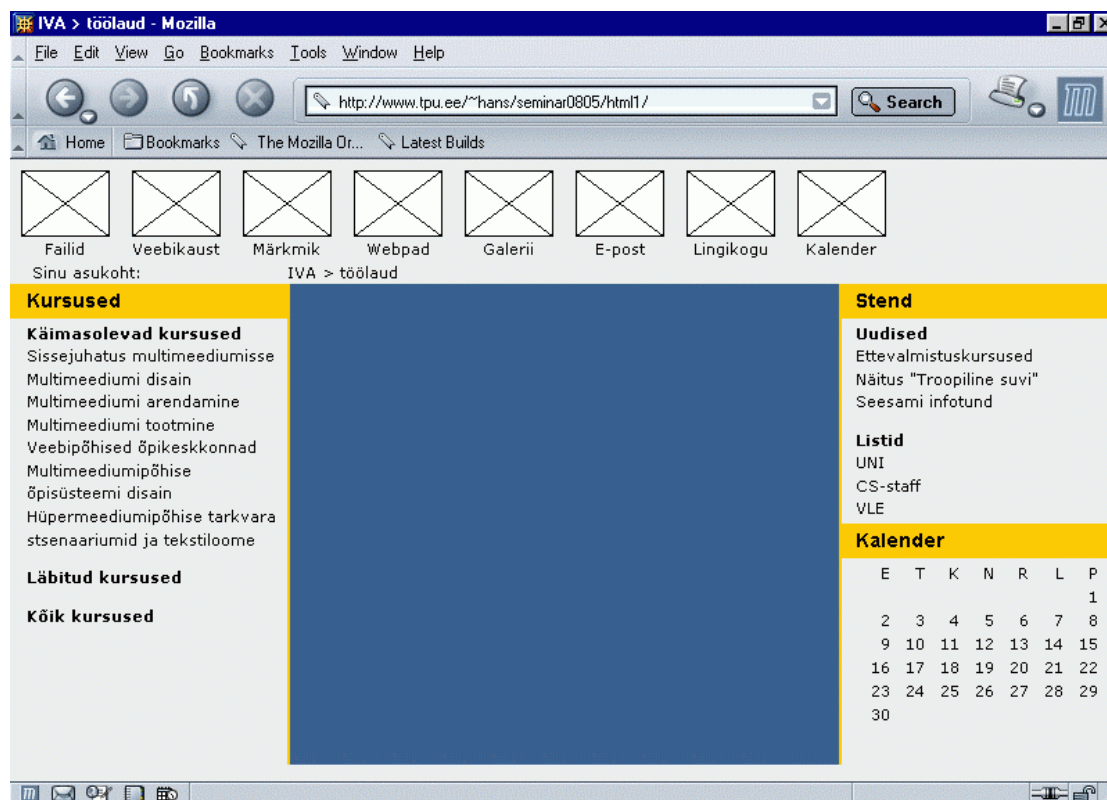


Joonis 5.7. IVA kasutajaliidese visandid

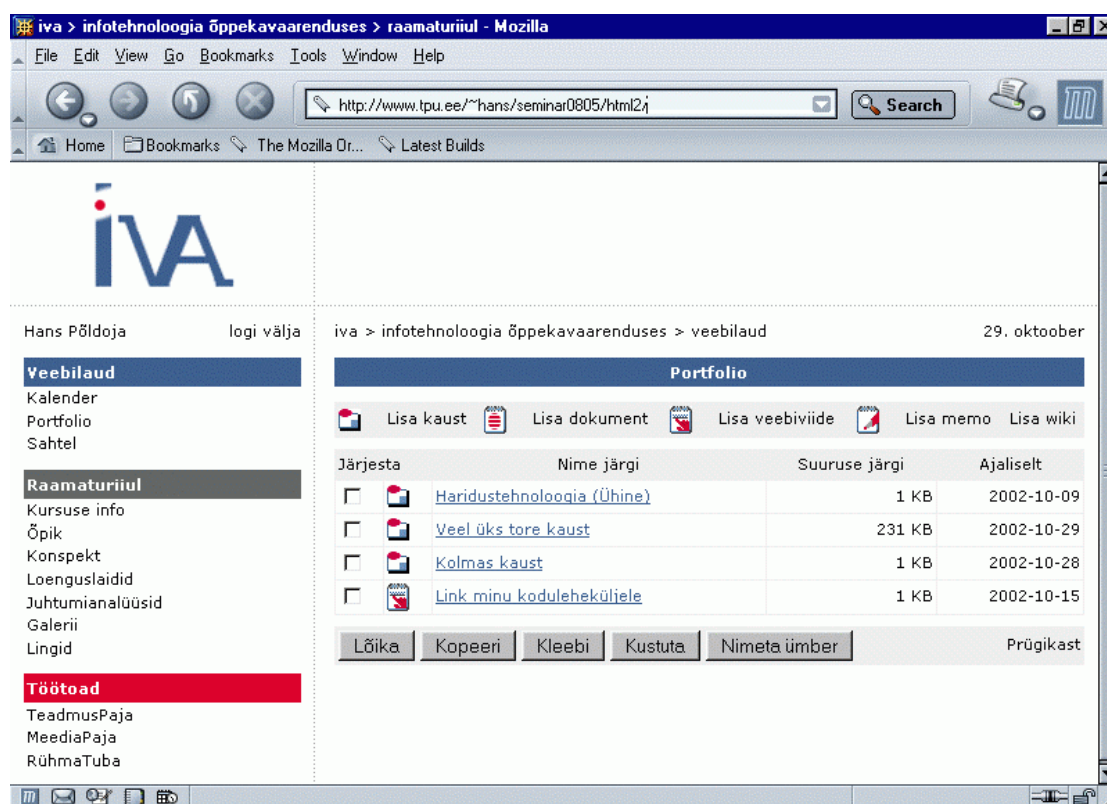
Digitaalsete prototüüpide valmistamiseks kasutatakse kas graafikaprogramme või HTML märgendikeelt. Nende valmistamine on aeganõudev, kuid samas on need lihtsalt muudetavad ning neid on võimalik panna veebi, et saada kasutajate hinnangut. Digitaalsete prototüüpide puuduseks on see, et kasutajad kipuvad neid võtma lõplike kujundusversioonidena ning norima seetõttu üksikasjade kallal, mitte keskenduma üldmuljele ja paigutusele. Samuti võivad disainerid oma idee küljes liiga tugevalt kinni olla, sest digitaalse (eriti HTML) prototüübi loomisele kulub teistest prototüüpimismeetoditest rohkem tööd ja aega.

IVA puhul valmis kolm HTML prototüüpi, millest viimase põhjal on toodetud IVA praegune versioon. HTML prototüüpe eelistati graafikaprogrammide abil tehtud prototüüpidele seetõttu, et eesmärgiks oli HTML poole pealt suhteliselt lihtsate, kuid CSS'i abil pilkupüüdvalt kujutatud lehtedega kasutajaliides. CSS'i abil on selliste prototüüpide muutmine tunduvalt lihtsam kui graafikaprogrammis.

Esimene HTML prototüüp tugines operatsioonisüsteemi töölaua (kõik objektid ikoonidena töölaual näha) ning brauseri metafooridele (vahendid lehe ülaservas nuppude reana). Seda prototüüpi lõpuni ei arendatud, sest selle CSS ja tabelite struktuur läks liiga keeruliseks. Samuti oleks võinud kasutajates segadust tekitada topelt nupurida ülaservas (brauseri nupud ning IVA nupud). Viimatimainitud probleemi oleks saanud vältida IVA avamisega eraldi nupureata brauseriaknas, kuid see oleks eeldanud JavaScripti kasutamist ning tekitanud ühildusprobleeme vanemate brauseritega. Kõnealune prototüüp on esitatud joonisel 5.8.



Joonis 5.8. HTML prototüüp nr 1



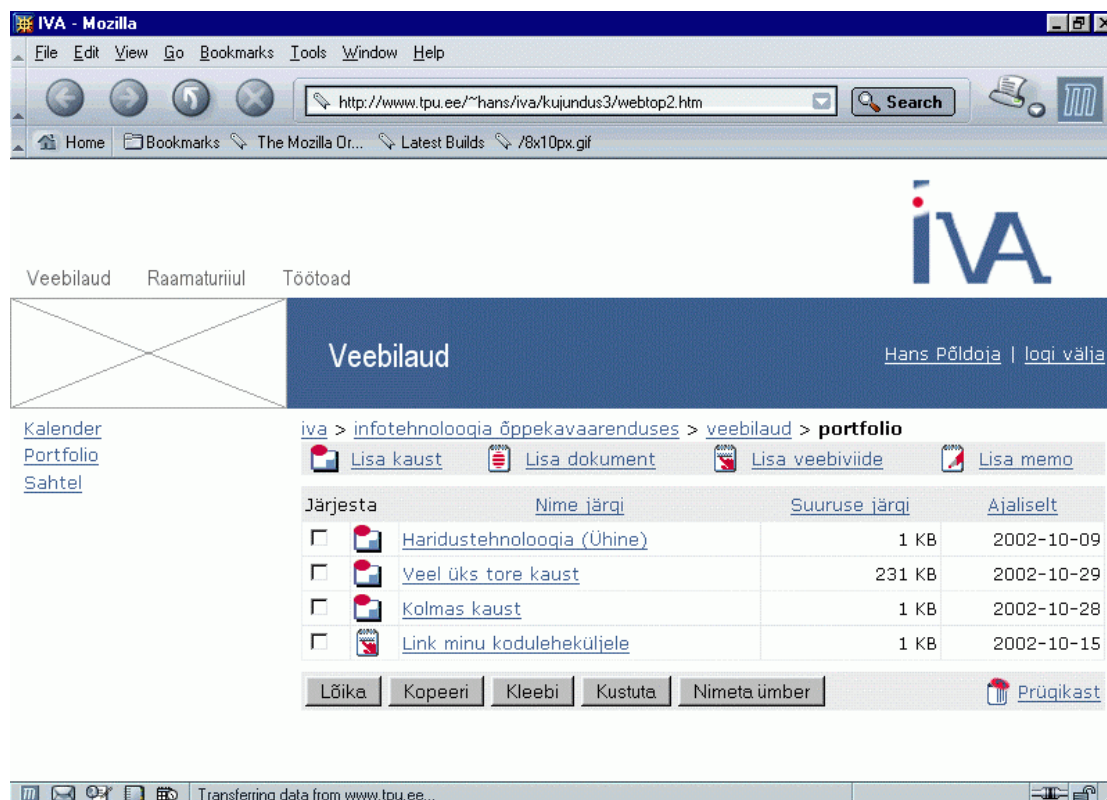
Joonis 5.9. HTML prototüüp nr 2.

Teine HTML prototüüp on kujutatud joonisel 5.9. Inimestelt, kes seda kujundust nägid, kuulsime positiivseid arvamusi, kuid mitmetel põhjustel ei läinud seegi prototüüp kasutusse. Peamiseks probleemiks oli, et kasutaja jaoks ei eristu piisavalt selgelt see, millises IVA sektsioonis ta parajasti viibib. Seda on võimalik kindlaks teha parempoolse ekraaniosa peakirja taustariba värvi ning selle kohal oleva asukoharibaga (nn “leivapuru”), kuid see ei pruugi algajate kasutajate jaoks piisav olla. Teiseks probleemiks oli see, et vasakpoolne menüü võib sõltuvalt raamaturiulil olevate kaustade arvust liiga pikaks venida. Kaalusime dünaamilist lahendust, kus korraga oleks nähtav kahe sektsiooni nimed ning aktiivne sektsioon koos selle all olevate linkidega, kuid sellise esitusviisi korral oleks ühest sektsioonist teise mineku korral menüüpunktide paigutus muutunud. Selle prototüübi puudusteks toodi arutelu käigus veel see, et väljalogimise link peaks paiknema traditsiooniliselt üleval paremal servas. Arendusmeeskonnas tekitas ka lahkkelisid hulk tühja pinda paremal akna ülaserbas. Kujunduslikult on see vastukaaluks ülejäänud pinnale, mis on suhteliselt tihedalt täidetud, kuid samas on nii kasutamata jäetud lehekülje kõige väärtuslikum pind, mida vaataja esmapilgul märkab.

Kolmanda HTML põhiliseks erinevuseks võrreldes eelkäijaga oli see, et Veebilaud, Raamaturiul, Töötoad ja õpetajale ligipääsetav Haldus olid teineteisest selgelt eristatavad. Selleks võeti kasutusele 80 pikseli kõrgune horisontaalriba koos vastavat IVA sektsiooni iseloomustava taustavärvi ja fotoga (vt joonis 5.10, fotod valmisid tootmise etapis ning pole seetõttu prototüübil esitatud).

IVA ekraani ülaosas on tabel kolme (õpetajal nelja) peamise jaotuse nupu ning IVA logoga. Sellest allpool teine tabel vastava jaotuse värviriba koos jaotuse nime, kasutaja nime (viitab kasutajainfole) ning väljalogimise lingiga. Lehekülje põhiosa paikneb kolmandas tabelis. Vasakul paikneb iga sektsiooni menüü: Veebilaua puhul on seal Portfoolio ja Sahtel; Raamaturiuli puhul kursuse info ja kataloogide loetelu, Töötubade puhul Teadmuspaja, MeediaPaja, Wiki ning sisegruppide ja testide loetelu. Paremal paikneb asukohariba ning selle all konkreetse lehekülje sisu.

Veebilaua puhul koosneb sisu lisamisribast, järjestamisribast, objektide loetelust ning operatsioonide ribast. Ribade taustavärv on helehall (#EEEEEE), et need valgel taustal paremini esile tuua, kuid samas mitte liiga intensiivseks muuta.



Joonis 5.10. HTML prototüüp nr 3

Kõigi kolme HTML prototüübi puhul määrati lehekülgede paigutus siiski tabelite, mitte CSS'i, abil. Sedasi saab olla kindel, et lehekülje paigutus ja üldmulje ei erine oluliselt ka nõrga CSS'i toetusega brauseritel, nagu Netscape 4.x, mis TPÜ-s veel mitmel pool kasutusel on. Samal põhjusel on pindade taustavärvid määratud HTML'is. Prototüübid valideeruvad vastavalt XHTML 1.0 Transitional soovitusetele ning ka CSS vastab standarditele.

Et näha erinevate ekraanikujunduste koosmõju kasutatakse *storyboard* meetodit. Paberprototüübi puhul kinnitatakse lehed läbimise järjekorras stendile, digitaalse prototüübi puhul koostatakse näiteks presentatsioonitarkvara abil slaidiseeria ekraanipiltidest läbimise järjekorras (hea ülevaate annab selline režiim, kus korraga kuvatakse mitme slaidi eelvaadet).

Visandamine ja prototüüpimine on iteratiivne protsess – selle käigus on korduvalt vaja saada tulevaste kasutajate poolt tagasisidet, mille põhjal viiakse sisse muudatused ja parandused. IVA projektis hinnati prototüüpe arendusmeeskonna ning tulevaste kasutajate poolt. Saadud tagasiside põhjal parandati ja täiendati prototüüpe. Tagantjärele hinnates oleks tulnud koostada sama üldkujundusega rohkem erinevate lehtede kohta prototüüpe ning neid laiemale kasutajate ringile tagasiside saamiseks

tutvustada. Piiratud ajaressursi tõttu ei olnud see kahjuks võimalik, ning kolmanda prototüübi valmimise järel läks keskkond tootmisesse.

5.4 Tootmine

Tootmise etapil on kujunduse muutmine juba keerulisem ja kulukam, kui varasematel etappidel. Sellel etapil on kõige enam hõivatud graafikatöötluste, HTML'i kirjutamise ning programmeerimisega seotud projektirühma liikmed. Reeglina tegelevad sellega erinevad inimesed. Seetõttu on siinjuures väga oluline projekti koordineerimine ning tööjaotus. Sõltuvalt kasutatavast arendusplatvormist on võimalik kujundajate ja koodikirjutajate tööd suhteliselt teineteisest sõltumatuna hoida. Tootmise etapis rakendatavad kasutatavuse meetodid on esitatud tabelis 5.7.

Tabel 5.7. Tootmise etapil rakendatavad kasutatavuse meetodid (Brinck et al, 2002, lk 32)

Meetod	Nõutav	Kestvus	Õppimiseks kuluv aeg	Usaldusväärsus tase	Mõju lõplikule disainile
Kasutatavuse kontroll-loendid	Ei	Lühike	lühike	Madal	keskmine
Kasutajatega testimine	Ei	lühike-keskmine	keskmine	Kõrge	keskmine

IVA puhul erines tootmise etapp tavalistest tarkvaraprojektidest selle poolest, et me ei alustanud mitte tühja koha pealt, vaid hakkasime valmis süsteemi kasutajaliidest oma vajadustele ümber kujundama.

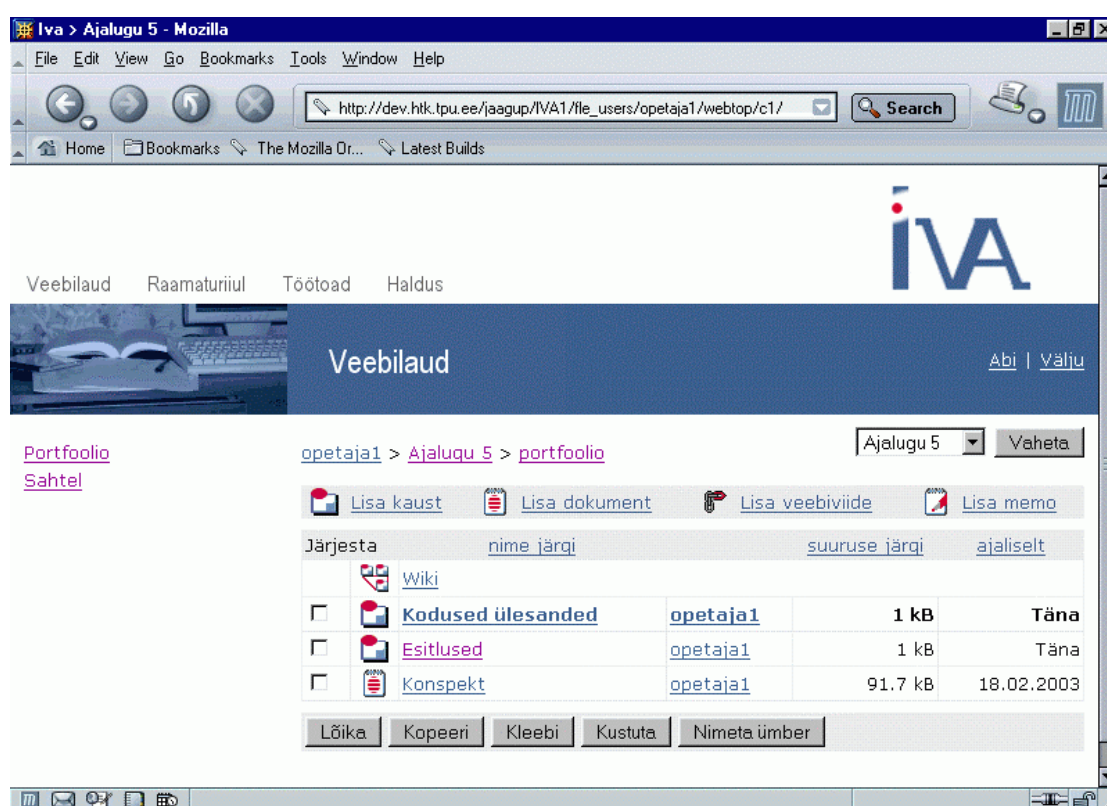
Zope puhul on lehekülgede kujundamiseks kaks tehnoloogiat – FLE3 kasutajaliides on loodud DTML märgendikeele (*Document Template Markup Language*) abil, kuid Plone kasutab ZPT (*Zope Page Templates*) tehnoloogiat.

DTML märgendikeele kasutamine võimaldab laadida lehekülje kokku erinevatest failidest umbes samamoodi nagu include() funktsioon PHP-s. Nii paikneb näiteks lehekülje päis ühes failis, kusjuures sõltuvalt kasutaja asukohast süsteemis kuvatakse teine rida vajaliku foto ja sektsiooni nimetusega. Samamoodi tuleb olenevalt leheküljest erinevatest failidest ka vasak menüü, lehekülje põhiosa, jalus jne.

FLE3 poolt genereeritud leheküljed vastasid HTML 4.01 Transitional standardile, kuid olid liiga keeruliste tabelite peal üles ehitatud. Nimelt kasutatakse FLE3

kujunduses tabelitel kumeraid nurki, mille tegemiseks on vaja igale tabelile juurde kahe rida ning kahte veergu (üles, alla, vasakule, paremale). Nendest loobumine võimaldas IVA lehekülgi võrreldes FLE3-ga lihtsamateks ja kiiremini laadivateks muuta.

Kasutatavuse meetoditest rakendati sel etapil kasutajatega testimist. Esimesed kursused algasid IVA keskkonnas juba versiooni 0.1 valmimisele järgneval nädalal. Kasutajapoolne tagasiside on veel tootmise faasis olevat süsteemi võimaldanud mitmest küljest kasutajasõbralikumaks muuta. Praegusel hetkel on 3. prototüübiga võrreldes kujundust mõnevõrra muudetud ja täiendatud (vt joonis 5.11).



Joonis 5.11. IVA Veebilaud

Lisandunud on rippmenüü kursuse vahetamiseks. Kasutaja nime asemel näidatakse väljalogimise lingi kõrval viidet IVA abiinfo lehekülgedele. Kasutaja nimi ja ligipääs kasutajaprofiili muutmisele on seoses sellega liikunud “leivaterakeste” algusesse. Välja on vahetatud veebiviite ikoon, kuna see oli liialt sarnane dokumendi ja memo ikooniga. Siiski on ka praegu ikoonide kohta tulnud kriitikat, mis sunnib otsima paremini eristuvaid ikoone dokumendile ja memole. Objektide loetelu ülemisel real on Portfoolios viide Wiki lehekülgedele, Sahtlis pääseb kasutaja samast kohast ligi oma kasutusstatistikale. Objekti kõrval on näha selle lisaja nimi (eelkõige on sellest

kasu sisegruppides) ning sellel klõpsides statistika objekti vaatamistest. Kasutaja poolt vaatamata objektide tekst on esitatud paksus kirjas, analoogiliselt elektronpostiprogrammidega, kus lugemata kirjade päised on paksus kirjas. Eemaldatud on viide prügikastile, kuna FLE3 Veebilaua lahutamisel eraldi Portfoolioks ja Sahtliks ei funktsioneerinud prügikast enam.

Tootmise etapis täieneb IVA lisaks kasutajaliidese parandamisele ja ühtlustamisele ka uute funktsionaalsustega, nagu näiteks testide osa. Töö käigus järgitakse W3C standardeid ning testitakse lehekülgede valideeruvust. Eesmärgiks on saavutada täielik vastavus XHTML 1.0 Transitional soovitusel.

5.5 Kasutuselevõtt

Kasutuselevõtu etapp hõlmab teatavat osa õpikeskkonna kasutamiseks avamisele eelnevast ja järgnevast ajast. IVA ametlik kasutuselevõtt on augustis 2003, seetõttu ei saa me siinjuures veel kirjeldada, mida täpselt antud etapis teinud oleme. Kasutuselevõtu etapil rakendatavad kasutatavuse meetodid on toodud järgnevas tabelis.

Tabel 5.8. Kasutuselevõtu etapil rakendatavad kasutatavuse meetodid (Brinck et al, 2002, lk 32)

Meetod	Nõutav	Kestvus	Õppimiseks kuluv aeg	Usaldusväärsus tase	Mõju lõplikule disainile
Kvaliteedi tagamine	jah	lühike-keskmine	keskmine	Kõrge	keskmine
Kasutaja probleemide haldamine	jah	lühike	lühike	Madal	madal

Kvaliteedi tagamine peab eelnema õpikeskkonna kasutamiseks avamisele. Selle eesmärkideks on:

- kõrvaldada vead võimalikult varases staadiumis;
- pakkuda juhtnööre muudatusteks ja täiendusteks;
- jälgida esialgse nõuete spetsifikatsiooni täitmist.

Kvaliteedi tagamise üheks osaks on peatükis 6 kirjeldatud evalvatsiooni küsimustiku läbiviimine. Küsimustiku tulemusena esiletoodud vigade parandamise järel on tarvis

täiendavat testimist kasutajatega. Selleks sobib näiteks vestlusel baseeruv testimisviis *think aloud* protokoll.

Kasutajate probleemide registreerimiseks ja haldamiseks on loomisel IVA abiinfo lehele lisaks ka *Helpdesk*. Põhimõtteliselt on kasutajatel võimalik vigadest teada anda ka Savannah'i veakogujasse, kuid enamiku jaoks on see liiga keeruline.

6 ÕPIHALDUSSÜSTEEMI IVA KASUTAJALIIDESE DISAINI EVALVATSIOON

Evalvatsioonil on täita väga oluline roll kasutajaliidese disainis. Ideaaljuhul võiks evalvatsioon kulgeda disainiprotsessiga paralleelselt alates esimesest etapist. Preece defineerib evalvatsiooni mõiste kui "protsess, mille käigus kogutakse informatsiooni süsteemi kasutatavuse kohta eesmärgiga täiustada süsteemi või hinnata valmis liidest" (Preece, 1996, lk 713).

IVA kasutajaliidese evalvatsioon koosnes järgmistest osadest:

- testimine reaalsete veebikursuste käigus alates esimesest avalikust versioonist;
- ekspertide hulgas läbi viidud ankeetküsitlus kasutajaliidese hindamiseks.

IVA keskkonnas toimunud veebikursused on andnud vajalikku tagasisidet kasutajaliidese probleemide ja puuduste kohta. Kuna me pole veel jõudnud rakendada tagasiside haldamise süsteemi on enamasti suulisel teel meieni jõudnud tagasiside salvestatud lihtsalt tekstifaili. Suur osa laekunud probleemidest olid arendusrühmale endalegi teada, kuna lasime teadlikult veel lõplikult viimistlemata keskkonna testkasutusse. Samas on välja toodud ka palju soovitusi, mis aitab meil IVA paremaks muuta ning avastatud pisivigu.

Evalvatsiooni teiseks komponendiks on ekspertide hulgas läbi viidud ankeetküsitlus eesmärgiga saada hinnang kasutajaliideses rakendatud kujunduslahenduste ja metafooride kohta ning soovitusel kasutajaliidese parendamiseks.

Enne evalvatsiooniküsimustiku koostamist tutvuti olemasolevate standardsete evalvatsiooniküsimustikega. Sellistest küsimustikest kõige laiemalt kasutatav on Marylandi ülikoolis välja töötatud QUIS (*Questionnaire for User Interaction Satisfaction*). Selle eesmärgiks on uurida kasutajate rahulolu interaktsioonidega. Küsimustik koosneb 12 erinevast sektsioonist sellistel teemadel nagu näiteks ekraanipilt, terminoloogia ja multimeedia kasutamine. QUIS on koostatud selliselt, et sõltuvalt projektist ja evalvatsiooni eesmärkidest oleks võimalik erinevaid sektsioone välja jätta (Lazar, 2001, lk 280). QUIS uusim versioon on 7.0 ja seda levitatakse kommertsalustel (QUIS, 2003). Selle kõrval on levinud kasutatavuse evalvatsiooni testideks veel IBM'i poolt koostatud PUEU (*Perceived Usefulness and Ease of Use*), CSUQ (*Computer System Usability Questionnaire*) ja ASQ (*After Scenario Questionnaire*), Jakob Nielsen'i koostatud NAU (*Nielsen's Attributes of Usability*) ja

NHE (*Nielsen's Heuristic Evaluation*), PHUE (*Practical Heuristics for Usability Evaluation*), SUS (*System Usability Scale*) ning põhjalik 100-küsimuseline PUTQ (*Purdue Usability Testing Questionnaire*). Kõigi ülaltoodud küsimustike põhiosa on hinnatav Likerti skaalal (Perlman, 2001; Brooke, 1994).

Otsustasime võtta meiepoolse evalvatsiooniküsimustiku aluseks QUIS küsimustiku versiooni 5.5 avaliku variandi, mis koosneb 27 küsimusest. Igale küsimusele sai vastata 10-jaotuselisel Likerti skaalal (0...9). QUIS testis oli lisaks ka väli, mis tuli märkida juhul, kui vastav küsimus pole antud rakenduse puhul hinnatav. Asendasime sellised küsimused (näiteks helide kohta) IVA kasutajaliidese juures olulisemate küsimustega ning kõrvaldasime välja “mitte rakendatav”. Iga küsimuse kõrval oli tekstiväli kommentaari sisestamiseks, lisaks eraldi kommentaariväljad küsitluse lõpus. Küsimustik jagunes viieks sektsiooniks, mille tulemusi järgnevatel alapeatükkides käsitleme:

- üldine arvamus tarkvarast;
- ekraanipilt;
- kasutajaliidese terminid ja metafoorid;
- õppimine;
- süsteemi võimekus.

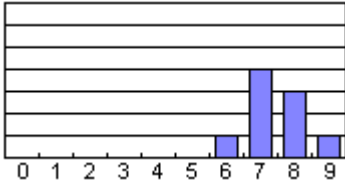
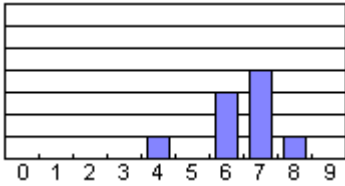
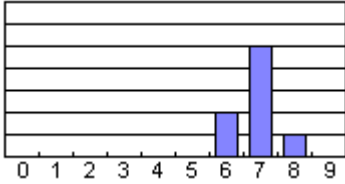
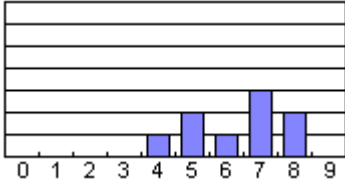
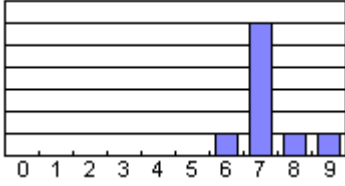
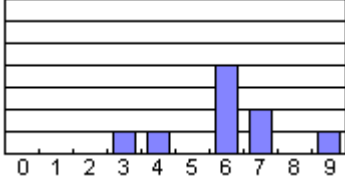
Alapeatükis 6.6 on välja toodud küsitluse tulemused peatükis 2.1 koostatud kasutatavuse kriteeriumide lõikes.

Küsimustiku täitjaks valiti 10 eksperti, kes olid IVA-ga tuttavad õpilase kui ka õpetaja rollis ning omasid lisaks ka kogemusi teiste õpihaldussüsteemide kasutamisel. Ekspertidelt laekus tagasi 9 täidetud ankeeti.

6.1 Üldine arvamus IVA kohta

Selle sektsiooni eesmärgiks oli välja selgitada kasutajate subjektiivne üldine arvamus IVA keskkonnast. Kokkuvõtte vastuste jaotusest ning aritmeetilisest keskmisest on esitatud tabelis 6.1.

Tabel 6.1. Ekspertide üldine arvamus tarkvarast

Küsimus	Vastuste jaotus	Keskmine
1. Üldine arvamus tarkvarast (0 hirmus ... 9 suurepärase)		7,4
2. Üldine arvamus tarkvarast (0 keeruline ... 9 lihtne)		6,4
3. Üldine arvamus tarkvarast (0 frustreriv ... 9 rahulolu valmistav)		6,9
4. Üldine arvamus tarkvarast (0 piiratud võimalustega ... 9 võimalusterohke)		6,3
5. Üldine arvamus tarkvarast (0 vastumeelsust põhjustav ... 9 stimuleeriv)		7,2
6. Üldine arvamus tarkvarast (0 jäik ... 9 paindlik)		6,0

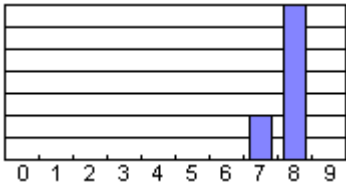
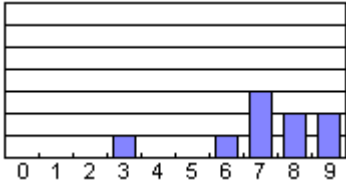
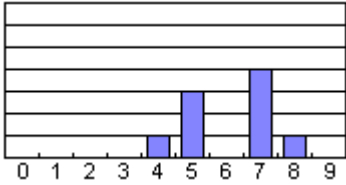
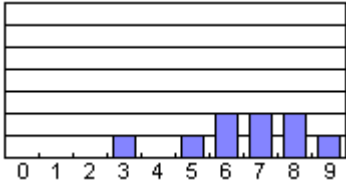
Ekspertide kommentaarides domineeris positiivne suhtumine IVA kasutajaliidesesse. Puudustest leidis mainimist TeadmusPaja esitusviis ja algajale liialt keeruline kasutamine. IVA peeti pedagoogilisest aspektist võimalusterohkeks, kuid WebCT-st vähem paindlikuks. Puudusena märgiti seda, et mitmed õpihaldussüsteemide tavapärased võimalused nagu kalender ja eksport on puudu. Lisaks sooviti uute funktsionaalsustena näha omavahel seotud HTML lehtede keskkonda üleslaadimise võimalust. Kasutatavuse tõstmiseks soovitati rohkem integreerida abiinfo süsteemiga

nagu on integreeritud näiteks teadmüstüüpide info TeadmusPajas. Samuti tehti ettepanek võtta kasutusele helilised märguanded.

6.2 Hinnang IVA ekraanikujundusele

Küsimused 7...10 keskendusid IVA ekraanipildile ja visuaalsele kujundusele. Vastuste jaotus ning aritmeetilised keskmised on esitatud tabelis 6.2.

Tabel 6.2. Ekspertide hinnang IVA ekraanikujundusele

Küsimus	Vastuste jaotus	Keskmine
7. Teksti loetavus ekraanilt (0 raske ... 9 kerge)		7,8
8. Värvivalik lihtsustab kasutamist (0 üldse mitte ... 9 väga palju)		7,1
9. Ekraanikujunduse liigendatus (0 segadusse ajav ... 9 väga selge)		6,1
10. Icoonide intuitiivne mõistetavus (0 segadusse ajav ... 9 väga selge)		6,6

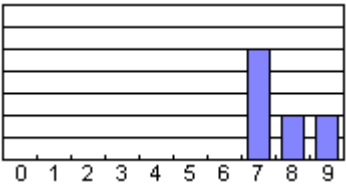
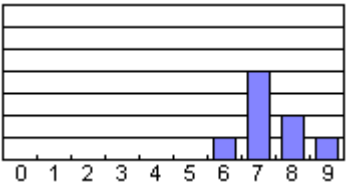
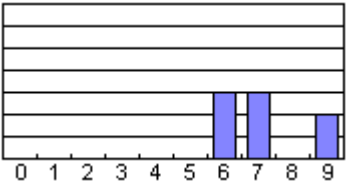
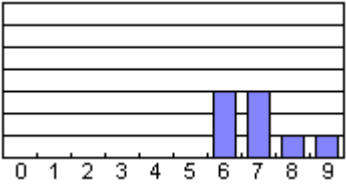
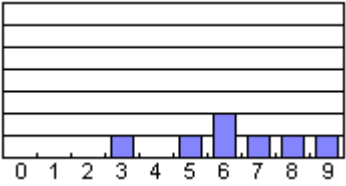
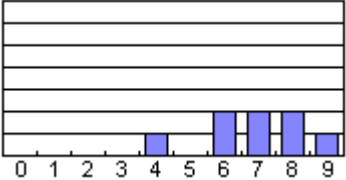
Positiivsena mainiti head loetavust. Üks ekspert tegi ettepaneku muuta Memo praegust püsilaiusega kirjatüübiga esitusviisi. Sellele saab vastu väita, et Memo kui vormindamata tekstidokument peaks olema esitatud püsilaiusega kirjatüübiga, et võimaldada selle kujundamist teksti paigutuse abil. Samuti tehti ettepanek esitada vasakpoolse menüü ees puustruktuur, mis lihtsustaks menüü mõistmist. Kaalusime antud võimalust IVA kujunduse loomisel, kuid kuna seal menüüs sai lõpuks esitatud ainult üks tase viiteid, ei ole puu lisamine viidete ette mõttekas. Ka siin jaotuses soovitati TeadmusPaja esituse selgemaks muutmist. Neli vastajat tõi välja selle, et

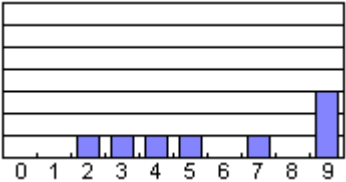
ikoonid on harjumatud. Soovitati muuta kausta ikoon harjumuspäraseks ning leiti, et dokumendi ja memo ikoonid on liiga sarnased.

6.3 Hinnang IVA kasutajaliidese terminitele ja metafooridele

Küsimused 11...17 käsitlevad IVA kasutajaliidese termineid ja metafoore. Algses QUIS küsimustikus oli terminoloogia kõrval süsteemipoolset tagasiside puudutavad küsimused, mis olid veebirakenduste puhul raskesti rakendatavad. Seetõttu on nende asemel IVA disainis olulisel kohal olev metafooride teema. Vastuste jaotus on esitatud tabelis 6.3.

Tabel 6.3. Ekspertide hinnang kasutajaliidese terminite ja metafooride rakendamisele

Küsimus	Vastuste jaotus	Keskmine
11. Mõistete kasutamine läbi süsteemi (0 ebajärjekindel ... 9 järjekindel)		7,7
12. Metafooride intuiitiivne õpitavus (0 keeruline ... 9 lihtne)		7,4
13. Metafoorid lihtsustavad IVA kasutamist (0 mitte kunagi ... 9 alati)		7,1
14. Funktsionaalsus vastab metafoori poolt kujundatud ootustele (0 mitte kunagi ... 9 alati)		7,0
15. TeadmusPaja ja MeediaPaja mõisted (0 segadusse ajavad ... 9 selged)		6,3
16. Sisestusvormid (0 segadusse ajavad ... 9 selged)		6,9

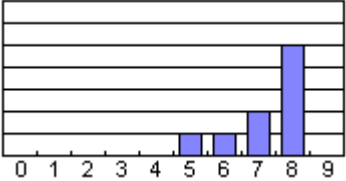
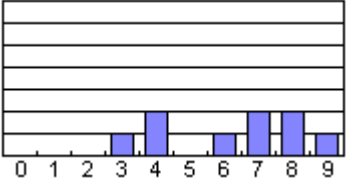
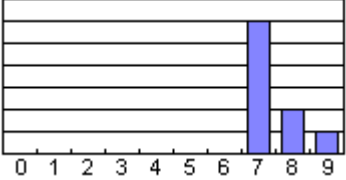
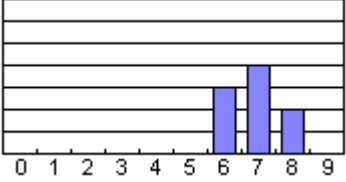
17. Veateated (0 kasutud ... 9 abistavad)		6,0
--	--	-----

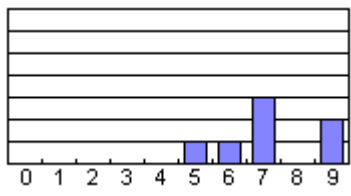
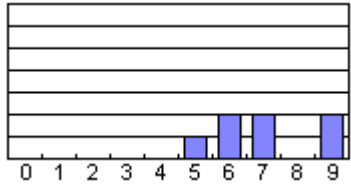
Ekspertid tõid välja tõsiasja, et IVA metafooride paremaks mõistmiseks peaks kasutajad olema kursis IVA pedagoogilise kontseptsiooniga. Üks ekspert pani kahtluse alla kursuse info paiknemise Raamaturiiulil. Sisestusvormide juures võib tekitada segadust kasutaja asukoht süsteemis vormi saatmise järel. Tõsise puudusena toodi välja puuduvad või ebatäpsed veateated.

6.4 Hinnang IVA kasutajaliidese õpitavusele

Küsimused 18...23 käsitlevad IVA kasutajaliidese õpitavust. Vastuste jaotus ja keskmised on esitatud tabelis 6.4.

Tabel 6.4. Ekspertide hinnang IVA kasutajaliidese õpitavusele

Küsimus	Vastuste jaotus	Keskmine
18. Süsteemi kasutama õppimine (0 keeruline ... 9 lihtne)		7,2
19. Keskkonnaga tutvumine katse-eksituse meetodil (0 keeruline ... 9 lihtne)		6,2
20. Käskude nimede ja kasutuse meelespidamine (0 keeruline ... 9 lihtne)		7,4
21. Operatsioonide teostamine on lihtne (0 mitte kunagi ... 9 alati)		6,9

22. Abiteated ekraanil (0 kasutud ... 9 abistavad)		7,1
23. Muud abimaterjalid (0 segadusse ajavad ... 9 selged)		7,0

Kommentaari IVA kasutama õppimise kohta (küsimus 18): „Õppejõud, keda olen nõustanud, leiavad ehk, et on raskem; samas – võrreldes teiste uuritud keskkondadega, on IVA lihtsus ja selgus võrreldamatu – paljud keskkonnad on ülepaistatud ja segadusse ajavad“. Teine ekspert lisab, et elementaarne kasutamine suhteliselt lihtne, kuid osata saada täpselt seda tulemust, nagu tahad, on keeruline.

Katse-eksituse meetodil õppimise lihtsustamiseks soovitati võimalust lülitada õpetajal õpilase režiimi. Selle meetodi takistusena toodi välja, et töötava kursuse peal ei saa katsetada, kuna igal pool puudub lisatu kustutamise võimalus.

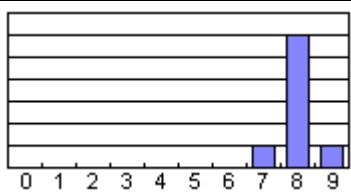
Taas mainiti, et TeadmusPaja kasutamine on keeruline. Üks ekspert tõi välja probleemi, et vormi sisestamiseks ei piisa Enter-klahvile vajutamisest, vaid tuleb ilmtingimata hiirega sisestusnuppu vajutada.

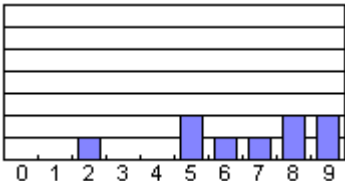
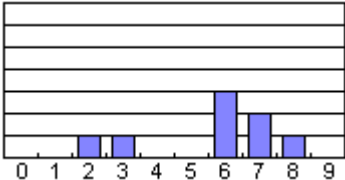
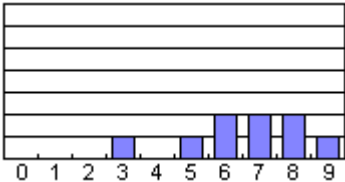
Üksmeelselt leiti, et abiteateid ekraanil võiks enam olla. Kuna üks vastaja soovis spetsiaalset linki „abi“ võib sellest järeldada, et praegune abi link pole piisavalt hästi nähtav (kuigi see paikneb harjumuspäraselt paremal ülaseravas).

6.5 Hinnang IVA õpihaldussüsteemi võimekusele

Küsimused 24...27 käsitlevad õpihaldussüsteemi võimekust. Kokkuvõtte vastuste jaotumisest on esitatud tabelis 6.5.

Tabel 6.5. Ekspertide hinnang süsteemi võimekusele.

Küsimus	Vastuste jaotus	Keskmine
24. Süsteemi kiirus (0 liiga aeglane ... 9 piisavalt kiire)		8,0

25. Süsteemi usaldusväärsus (0 ebausaldusväärne ... 9 usaldusväärne)	 <table><caption>Data for Item 25</caption><thead><tr><th>Rating</th><th>Frequency</th></tr></thead><tbody><tr><td>2</td><td>1</td></tr><tr><td>5</td><td>2</td></tr><tr><td>6</td><td>1</td></tr><tr><td>7</td><td>1</td></tr><tr><td>8</td><td>2</td></tr><tr><td>9</td><td>2</td></tr></tbody></table>	Rating	Frequency	2	1	5	2	6	1	7	1	8	2	9	2	6,6
Rating	Frequency															
2	1															
5	2															
6	1															
7	1															
8	2															
9	2															
26. Vigade parandamine (0 keeruline ... 9 lihtne)	 <table><caption>Data for Item 26</caption><thead><tr><th>Rating</th><th>Frequency</th></tr></thead><tbody><tr><td>2</td><td>1</td></tr><tr><td>3</td><td>1</td></tr><tr><td>6</td><td>3</td></tr><tr><td>7</td><td>2</td></tr><tr><td>8</td><td>1</td></tr></tbody></table>	Rating	Frequency	2	1	3	1	6	3	7	2	8	1	5,6		
Rating	Frequency															
2	1															
3	1															
6	3															
7	2															
8	1															
27. Sobiv erinevate oskustega kasutajatele (0 mitte kunagi ... 9 alati)	 <table><caption>Data for Item 27</caption><thead><tr><th>Rating</th><th>Frequency</th></tr></thead><tbody><tr><td>3</td><td>1</td></tr><tr><td>5</td><td>1</td></tr><tr><td>6</td><td>2</td></tr><tr><td>7</td><td>2</td></tr><tr><td>8</td><td>2</td></tr><tr><td>9</td><td>1</td></tr></tbody></table>	Rating	Frequency	3	1	5	1	6	2	7	2	8	2	9	1	6,6
Rating	Frequency															
3	1															
5	1															
6	2															
7	2															
8	2															
9	1															

Süsteemi usaldusväärsuse probleemidena mainiti aeg-ajalt esinevaid seletamatuid vigu. Vigade parandamise juures toodi välja see probleem, et kasutajal puudub paljudes kohtades võimalus sisestatud teksti hiljem muuta. Märgit, et päris algajale kasutajale IVA siiski ei sobi.

6.6 Hinnangud kasutatavuse kriteeriumide lõikes

Peatükis 2.1 on koostatud kasutatavuse hindamise kriteeriumid. QUIS testi väljavalimisel ja kohandamisel peeti silmas seda, et küsitluse tulemused võimaldaks anda hinnangu käesolevas töös koostatud kasutatavuse kriteeriumidele. Ankeetküsimustiku küsimuste seosed kriteeriumidega on esitatud tabelis 6.6. Küsimused, mis iseloomustavad mitut kriteeriumi, on pandud mitme kriteeriumi küsimuste komplekti. Küsimused 1...6, mis käsitlesid üldist arvamust tarkvarast, ei kuulu otseselt ühegi kriteeriumi juurde.

Hindamaks IVA kasutajaliidese kriteeriumile vastavust on võetud kõigi antud kriteeriumiga seotud küsimuste vastuste aritmeetiline keskmine ning teisendatud see parema ülevaatlikkuse saamiseks 0...1 skaalale.

Tabel 6.6. Kasutatavuse kriteeriumidele vastavad ankeetküsitluse küsimused

Kriteerium	Seotud küsimused	Keskmine
Õpiahaldussüsteemi kasutama õppimise lihtsus	10, 12, 14, 15, 18, 19, 20, 21, 22, 23, 27	0,76
Õpiahaldussüsteemi kasutamise efektiivsus	8, 11, 13, 21, 24, 27	0,80
Piisav tagasiside kasutajale	16, 17, 22	0,74
Vigadekindlus	16, 22, 25, 26	0,73
Esteetiline ja minimalistlik kujundus	7, 8, 9, 10	0,77

6.7 Kokkuvõte ja järeldused

Küsitluse lõpus pidid vastajad sisestama kolm IVA kasutajaliidese juures positiivset ning kolm negatiivset aspekti. Kokkuvõte nendest on esitatud tabelis 6.7. Järjestusse seadmiseks andis esikoht 3, teine koht 2 ja kolmas koht 1 punkti. Erineva sõnastusega, kuid samasisulised aspektid on töö autori poolt kokku võetud.

Tabel 6.7. Kokkuvõtte IVA kasutajaliidese positiivsetest ja negatiivsetest aspektidest

Positiivne	Punkte	Kordi
1. Visuaalne kujundus	12	5
2. IVA kontseptsioon ja metafoorid	9	4
3. Eestikeelne kasutajaliides	6	2
Negatiivne	Punkte	Kordi
1. Teadmuspaja kasutatavus	9	4
2. Kasutajate haldus	9	3
3. Wiki kasutamine ebaselge	6	3

4 hindajat tõid positiivsena välja IVA olemasolu ja pideva arengu, kuid kokkuvõtte tegemisel jäeti see kui kasutajaliidesega mitteseonduv aspekt välja.

Kasutajatega testimisel osutuks olulisimaks puuduseks IVA TeadmusPaja keeruline esitusviis. Selle probleemi lahendamiseks töötatakse välja erineva TeadmusPaja kujundusega HTML prototüübid ning lastakse kasutajatel neist sobivaim valida. Lisaks sellele toodi välja vajadus sõnumite kustutamise järele.

Kasutajate haldus põhjustab probleeme eelkõige pika kasutajate nimekirja puhul – kursusele kasutajate lisamisel näidatakse ühel leheküljel kõiki süsteemi kasutajaid. See pärineb FLE3-st, mida ei kasutatud suuremahuliseks veebipõhiseks õppeks ning kus see seetõttu probleemiks ei kujunenud. Nüüd on FLE3 meeskond siiski jaotanud kasutajate nimekirja dünaamiliselt 20-kaupa lehekülgedele. Samalaadne lahendus tuleb sisse viia ka IVA-s.

Wiki puhul põhjustab probleeme ilmselt selle vahendi tundmatus kasutajate jaoks. Kasutajatele pole harjumuspärane see, et loodud alamlehti ei saa kustutada. Samuti eeldas üks kasutaja, et Veebilaua Wiki sisu on nähtav ka Töötubade Wikis. Selle vahendi kasutamisega seotud probleeme aitab vältida kasutajate koolitus ning kasutusjuhendi täpsustamine. Samuti aitab kaasa Wiki laienev levik (on ju teine uue kontseptsiooniga veebivahend – *Blog* – muutunud juba Eestis tuntuks).

7 KOKKUVÕTE

IVA projekti üheks eesmärgiks oli viia arendusprotsess läbi nii, et sellest saadav teadmuspagas leiaks rakendust TPÜ-s ja teistes Eesti ülikoolides e-õppe massiliseks muutmisel. IVA arendus oli mitmes mõttes omanäoline, sest piiratud aja- ja inimressursiga valmis õpihaldussüsteem mitmete olemasolevate tarkvaraproduktide põhjal. Käesolev magistritöö on esimene IVA arendusprojekti tulemuste põhjal kaitsmisele esitatud teadustöö.

Arendusprotsess tugines tunnustatud arendusmudelile (Brinck et al, 2002) ning metoodikatele, seades keskele kohale kasutatavuse ja evalvatsiooni. Töö tulemusena valmis:

- kasutajaliidese disainiprotsessi käsitus, mis ühendab nii traditsioonilisi tarkvaraarenduse meetodeid kui ka humanitaarseid aspekte nagu metafooride rakendamine;
- ainulaadne innovatiivsel pedagoogilisel mudelil põhinev õpihaldussüsteemi kontseptsioon ja struktuur;
- intuitiivselt õpitav, kasutajasõbralik ja esteetiline kasutajaliidese kujundus TPÜ õpihaldussüsteemile IVA.

Paralleelselt arendusprotsessiga viidi töö autori poolt läbi mitmeosaline evalvatsioon, mis koosnes arendusjärgus keskkonna kasutamisest reaalses õppetöös, heuristilist laadi ankeetküsimustikust ekspertidele ning W3C testidest tootmise käigus. Evalvatsioon aitas juhtida tähelepanu mitmetele IVA kasutajaliidese nõrkadele kohtadele. Evalvatsiooni tulemusi arvestades viib arendusmeeskond IVA kasutajaliidese läbi vajalikud parandused ja täiendused. Alates 2003. aasta sügisest võetakse IVA kasutusele TPÜ ametliku e-õppe keskkonnana. Saadud kogemuse põhjal on kavas alustada IVA uut versiooni, mis baseerub täielikult meie oma koodil.

Magistritöö käigus kerkisid üles uued uurimisküsimused, millega tegelemine pakuks käesoleva töö autorile edaspidises uurimistöös suurt huvi, näiteks: kuidas lahendada kasutatavusprobleeme, mida toovad kaasa tänapäevased trendid õpihaldussüsteemide arengus (õpiobjektide metaandmete paisumine, tsentraliseeritud õpiobjektide ladude kasutamine erineva disainiga õpihaldussüsteemide poolt).

KASUTATUD KIRJANDUS

- Andrew, R. (2001). Real World Table Free Site Development. Loetud Internetis 28. aprill 2003 aadressil <http://www.womendesignersgroup.com/articlerachel.shtml>
- Brinck, T., Gergle, D., Wood, S.D. (2002). Usability for the web: designing web sites that work. Morgan Kaufmann Publishers.
- Brooke, J. (1994). SUS – A quick and dirty usability scale. Loetud Internetis 14. mai 2003 aadressil <http://www.usability.serco.com/trump/documents/Suschart.doc>
- Bush, V. (1945). As We May Think. –The Atlantic Monthly. 1945, juuli. Loetud Internetis 21. aprillil 2003 aadressil <http://www.ps.uni-sb.de/~duchier/pub/vbush/vbush-all.shtml>
- Cates, W.M. (2002) Systematic selection and implementation of graphical user interface metaphors. – Computers & Education 38 (2002), lk 385-397.
- Cooper, A. (1995). The Myth of Metaphor. – Visual Basic Programmer's Journal. 1995, juuni. Loetud Internetis 30. aprill 2003 aadressil http://www.cooper.com/articles/art_myth_of_metaphor.htm
- Gottesdiener, E. (2001). Collaborate for Quality: Using workshops to determine your project's requirements. – Software Testing and Quality Engineering Magazine. Märts/aprill 2001. Loetud Internetis 9. mail 2003 aadressil <http://www.volere.co.uk/collaborate.pdf>
- Hall, B. (2001). New Technology Definitions. Loetud Internetis 1. mai 2003 aadressil <http://www.brandonhall.com/public/glossary/glossary.html>
- Huang, S-M., Shieh, K-K., Chi, C-F. (2002). Factors affecting the design of computer icons. – International Journal of Industrial Ergonomics 29 (2002), lk 211-218.
- IEEE. (1990). IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York.
- Jonassen, D.H. (1994). Thinking Technology: Toward a Constructivist Design Model. – Educational Technology. Aprill, 1994, lk 34-37.
- Kikkas, K. (2002). E-koolitus mõõdukate kulutustega: vaba tarkvara e-õppes. – A&A. 2002 nr 6, lk 24-29.

- Krull, E. (2000). Pedagoogilise psühholoogia käsiraamat. Tartu: Tartu Ülikooli Kirjastus.
- Lakoff, G., Johnson, M. (1980). Metaphors We Live By. Chicago: The University of Chicago Press.
- Lazar, J. (2001). User-Centered Web Development. Jones and Bartlett Publishers.
- Leinonen, T., Virtanen, O., Hakkarainen, K., Kligyte, G. (2002). Collaborative Discovering of Key Ideas in Knowledge Buiding. Proceedings of the Computer Support for Collaborative Learning 2002 Conference. Boulder, Colorado, USA.
- Loetud Internetis 12. mai 2003 aadressil
http://www2.uiah.fi/~tleinone/codi/codi_cscl.pdf
- Lessard, D. (2002). PCmuseum – Personal computer history and collection of pc's. Loetud Internetis 1. mai 2003 aadressil <http://members.fortunecity.com/pcmuseum/>
- Newman, W.M., Lamming, M.G. (1995). Interactive System Design. Cambridge: Affison-Wesley.
- Nielsen, J., Norman, D.A. (2000). Web-Site Usability: Usability On The Web Isn't A Luxury. Loetud Internetis 19. aprill 2003 aadressil
<http://www.informationweek.com/773/web.htm>
- Nielsen, J. (1994a). Ten Usability Heuristics. Loetud Internetis 16. aprill 2003 aadressil http://www.useit.com/papers/heuristic/heuristic_list.html
- Nielsen, J. (1994b). How to Conduct a Heuristic Evaluation. Loetud Internetis 16. aprill 2003 aadressil http://www.useit.com/papers/heuristic/heuristic_evaluation.html
- Perlman, G. (2001). Web-Based User Interface Evaluation with Questionnaires. Loetud Internetis 14. mai 2003 aadressil <http://www.acm.org/~perlman/question.html>
- Praust, V. (2003). Veebilehtede käideldavustest WCAG ehk kuidas teha probleemideta vaadatavat veebi. – Arvutimaailm. 2003 nr 1, lk 26-29.
- Preece, J. (1996). Human computer interaction. Harlow: Addison-Wesley.
- Põldoja, H. (2001). Lihtne veebipõhine õpikeskkond. Prototüübi väljatöötamine ja evalvatsioon. Tallinna Pedagoogikaülikooli informaatika õppetool. Tallinn. [Diplomitöö].

Põldoja, H., Laanpere, M. (2002). Veebipõhiste kursuste autorsüsteemidest. – A&A. 2002 nr 1, lk 43-49.

QUIS. (2003). Questionnaire for User Interaction Satisfaction. Loetud Internetis 14. aprill 2003 aadressil <http://www.lap.umd.edu/QUIS/>

Robertson, J., Robertson, S. (2002). Volere Requirements Specification Template. Loetud Internetis 9. mai 2003 aadressil <http://www.systemsguild.com/GuildSite/Robs/Template.html>

Sanford, G. (2003) apple-history.com. Loetud Internetis 1. mai 2003 aadressil <http://www.apple-history.com/>

Smilowitz, E. (1996). Do Metaphors Make Web Browsers Easier to Use? Loetud Internetis 2. mai 2003 aadressil <http://www.baddesigns.com/mswebcnf.htm>

Tognazzini, B. First Principles. Loetud Internetis 16. aprill 2003 aadressil <http://www.asktog.com/basics/firstPrinciples.html>

TPÜ. (2001). E-ülikooli projekti rakendusjuhis. Loetud Internetis 6. mai 2003 aadressil <http://www.tpu.ee/editmode/e-ylikool/dokumendid/rakendusjuhis.html>

Zeldman, J. (2001) Why Don't You Code For Netscape. Loetud Internetis 24. aprill 2003 aadressil <http://www.alistapart.com/stories/netscape/>

W3C. (2000) About the World Wide Web Consortium (W3C). Loetud Internetis 21. aprill 2003 aadressil <http://www.w3.org/Consortium/>

W3C. (2003) Web Content Accessibility Guidelines 2.0. W3C Working Draft 24 April 2003. Loetud Internetis 28. aprill 2003 aadressil <http://www.w3.org/WAI/GL/WCAG20/>

LISA 1 VOLERE MALLI RAAMISTIK NÕUETE SPETSIFIKATSIOONI KOOSTAMISEKS

IVA nõuete spetsifikatsiooni koostamisel aluseks olnud Volere malli raamistik (Robertson, Robertson, 2002). Tabelis 5.2 on esitatud nõuete juures sulgudes Volere malli tüübi number, mille alla nõue kuulub.

Volere Requirements Specification Template

PROJECT DRIVERS:

1. The Purpose of the Product
2. Client, Customer, Stakeholders
3. Users of the Product

PROJECT CONSTRAINTS:

4. Mandated Constraints
5. Naming Conventions and Definitions
6. Relevant Facts and Assumptions

FUNCTIONAL REQUIREMENTS:

7. The Scope of the Work
8. The Scope of the Product
9. Functional and Data Requirements

NON-FUNCTIONAL REQUIREMENTS:

10. Look and Feel
11. Usability
12. Performance
13. Operational
14. Maintainability and Portability
15. Security
16. Cultural and Political
17. Legal

PROJECT ISSUES:

18. Open Issues
19. Off-the-shelf Solutions

- 20. New Problems
- 21. Tasks
- 22. Cutover
- 23. Risks
- 24. Costs
- 25. User Documentation
- 26. Waiting Room
- 27. Ideas for Solutions

SUMMARY

Keywords: metaphors, usability design, learning management systems

This MA thesis focuses on the issues of metaphor-driven user interface design for a Web-based learning management system called IVA. IVA is an open-source product developed in Tallinn Pedagogical University. It is built upon the leading free Web application server Zope, part of the source code is inherited from another open-source Zope product called FLE3. The goal of this thesis project was to reach better usability of learning management system by systematically increasing:

- Intuitivity through the use of metaphors;
- Interoperability through following W3C standards;
- Consistency through systemic design process;
- Engagement through innovative pedagogical foundation and variety of tools.

In order to reach these goals the literature review was conducted, prototype of IVA user interface was designed and evaluated against a set of usability heuristics.

IVA is currently available in Estonian, English and German, but it can be easily translated into other languages.

The length of the thesis is 61 pages. 13 figures and 16 tables are included. 35 resources of literature and Web links are referenced. The thesis is written in Estonian.