

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Informaatika instituut

Teadmussüsteemide õppetool

**Klienditoe töö automatiseerimine
Directo OÜ näitel**

Magistritöö

Üliõpilane: Irina Roost

Üliõpilaskood: 111830IABM

Juhendaja: Jaak Tepandi

Tallinn
2015

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

(kuupäev)

(allkiri)

Annotatsioon

Käesoleva magistritöö eesmärgiks on arendada klienditoe töö automatiseerimise lahendus äritarkvara arendamise ja hooldamisega tegeleva ettevõtte jaoks. Töö lähtepunktiks on vajadus vähendada igapäevaselt klienditoele esitatavate rutiinsete küsimuste arvu.

Töö käigus teostatakse ettevõtte probleemide analüüs ning tuginedes läbitöötatud teoreetilisele raamistikule keskendutakse juhtumipõhise süsteemi projekteerimisele, mis suudab võtta vastu kliendi sisestatud päringut ning väljastada sellele sobilikku lahendust. Süsteemi realiseerimist teostatakse närvivõrgu ja kNN algoritmide abil.

Peamise tulemusena on saadud intelligentne süsteem, mis leiab sisestatud päringule sobiliku lahenduse umbes 93% juhtumitest. Saadud lahendus vastab ettevõtte ootustele ning pärast kasutajaliidese loomist ning ettevõtte infosüsteemiga integreerimist võib see olla juurutatud ettevõtte igapäevasse töösse.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 58 leheküljel, 3 peatükki, 9 joonist, 9 tabelit.

Abstract

The purpose of the current work is to develop a solution for automating customer support in business software developing company. The starting point is the need to reduce the number of routine matters, which are presented on a daily basis to customer support.

Based on the theoretical framework and the company's needs the main focus is on the design of the case-based reasoning expert system, which is able to receive the customer's request and to return a suitable solution. The realization of the system is performed by using neural network and kNN algorithms.

The main result of this work is an intelligent system, which is able to find suitable solution for the request entered with the accuracy of 93%. The result meets company's expectations and system can be used in company's everyday work after user interface is created and integration with the company's information system is done.

The thesis is in Estonian and contains 58 pages of text, 3 chapters, 9 figures, 9 tables., etc.

Lühendite ja mõistete sõnastik

Bot	<i>Bot</i> tarkvaraline rakendus, mis automaatselt teostab etteantud ülesandeid läbi Internetti
CRM	<i>Customer Relationship Management</i> ettevõtte kliendisuhete planeerimise ja juhtimise süsteem
Ekspertsüsteem	<i>Expert system</i> arvutisüsteem, mis põhineb etteantud valdkonna ekspertide teadmistel ning üritab emuleerida inimese tegevusi, mis nõuavad arutamist ja teatud kogemust
ERP	<i>Enterprise Resource Planning</i> ettevõtte ressursside planeerimise süsteem
FAQ	<i>Frequently Asked Questions</i> korduma kippuvate küsimuste loetelu
IDF	<i>Inverse Document Frequency</i> statistiline näitaja, mida kasutatakse termini tähtsuse hindamiseks dokumentide kollektsiooni osana oleva dokumendi kontekstis
Informatsiooni otsing	<i>Information retrieval</i> informatsiooni vajadusega relevantsete infoallikate hankimise protsess infoallikate kollektsiooni hulgast
Kuulutustahvel	<i>Bulletin Board System</i> kasutajaliidesega server, mis võimaldas kasutajate suhtlust ning failide vahetust

Lekseem	<i>Token</i> morfoloogilise analüüsi põhiüksus, mis ühendab endas ühe sõna erinevad vormid
Lekseemide eraldamine	<i>Tokenization</i> teksti töötlusel tähendab teksti lahti löömist eraldi lauseteks, sõnadeks või muudeks tähendust omavateks üksusteks (lekseemideks)
Lemma	<i>Lemma</i> sõna morfoloogiline algvorm
Stopp-sõnad	<i>Stop words</i> loend tihti esinevatest sõnadest, mis ei mõjuta teksti tähendust, ning selle tõttu neid eemaldatakse teksti töötlemise optimeerimiseks
Sõnade koti mudel	<i>Bag-of-words model</i> teksti esitamise mudel, kus teksti esitatakse üksiksõnade loendina, sõnade grammatiliste seoste ning järjekorraga ei arvestata, kuid säilitatakse sõnade korduvuste arvu
Sündmus	<i>Ticket</i> probleemide jälgimise süsteemi osa, näitab antud probleemi staatust ning muu relevantset informatsiooni
TF	<i>Term Frequency</i> termini esinemissagedus antud dokumendis
Väljasttellimine	<i>Outsourcing</i> teatud äriprotsesside edastamine teisele organisatsioonile

Jooniste nimekiri

Joonis 1: Directo OÜ kasutajatoe ülesehitus	14
Joonis 2: Juhtumipõhise süsteemi töötsükkel (Aamodt ja Plaza, 1994: 46).....	28
Joonis 3: Juhtumi kirjelduses muutujate kasutamise näide	32
Joonis 4: Salvestatud päringu ja juhtumi seos	33
Joonis 5: Teksti lemmatiseerimise sammud	35
Joonis 6: Kasutaja päringu töötlemine närvivõrgu jaoks.....	38
Joonis 7: Päringu töötlemine närvivõrgu sisendiks	39
Joonis 8: Esimese ja teise teostuse võrdlus päringute klasside lõikes.....	46
Joonis 9: Klienditoe automatiseerimise juhtumipõhise ekspertsüsteemi töövoogu kirjeldus	50

Tabelite nimekiri

Tabel 1: Juhtumi kirjeldamise näide.....	32
Tabel 2: Sõnad ja nende IDF	37
Tabel 3: Närvivõrkude tulemused	40
Tabel 4: Sõnade grupid.....	41
Tabel 5: Sõna gruppide tähtsuse vektorid	42
Tabel 6: kNN tulemused (esimene teostus).....	43
Tabel 7: kNN tulemused päringute klasside lõikes	44
Tabel 8: kNN tulemused ilma päringu klassifitseerimise etapita	45
Tabel 9: kNN tulemused (teine teostus)	45

Sisukord

Sissejuhatus	11
1. Olukorra kirjeldus.....	14
1.1 Directo OÜ olemasolevad praktikad	14
1.2 Olemasoleva olukorra eelised ja puudused	15
1.3 Nõuded lahendusele.....	16
2. Lahenduse valik.....	18
2.1 Kliendi iseteenindamisvõimalused veebis.....	18
2.2 Tehisintellektil põhinevate algoritmide ülevaade.....	20
2.2.1 Ekspertsüsteemid.....	20
2.2.1.1 Reeglipõhised ekspertsüsteemid (RBES).....	20
2.2.1.2 Juhtumipõhised ekspertsüsteemid (CBR).....	20
2.2.2 Klassifitseerijad	21
2.2.2.1 Naiivsed Bayesi klassifitseerijad.....	21
2.2.2.2 K-lähima naabri algoritm (k-Nearest Neighbor – kNN)	22
2.2.3 Närvivõrgud.....	22
2.3 Parema alternatiivi valik.....	23
3. Lahenduse realiseerimine	27
3.1 Süsteemi projekteerimine	27
3.2 Juhtumite baasi projekteerimine	28
3.2.1 Algandmete ettevalmistamine	29
3.2.1.1 Algandmete korjamine	29
3.2.1.2 Algandmete klassifitseerimine	31
3.2.2 Juhtumite kirjeldamine	31
3.3 Juhtumite leidmine	33
3.3.1 Lemmatiseerimine	34
3.3.2 Stopp-sõnade eemaldamine	36
3.3.3 Klassifitseerimine	38
3.3.3.1 Andmete ettevalmistamine	38
3.3.3.2 Närvivõrgu esimene teostus.....	39
3.3.3.3 Närvivõrgu teine teostus.....	41
3.3.4 Juhtumite leidmine	42
3.3.4.1 Esimene teostus	42
3.3.4.2 Teine teostus	45

3.4 Juhtumite taaskasutamine ja hindamine	47
3.5 Juhtumite salvestamine.....	48
3.6 Süsteemi hindamine.....	51
Kokkuvõte	52
Summary.....	54
Kasutatud kirjandus	55

Sissejuhatus

Tänapäeva majandus on suuresti orienteeritud teenuste osutamisele ja konkurentide seast välja paistmiseks tihti juba ei piisa ainult toote kvaliteedist. Kliendikeskses kultuuris pööratakse erilist tähelepanu heale teenindusele ja ettevõtte üldisele miljööle.¹ Suured ettevõtted panustavad klientide rahulolu uuringute läbiviimisesse ja klienditeeninduse strateegiate väljatöötamisse, väiksemad ettevõtted lihtsalt keskenduvad klienditeeninduse taseme tõstmisele. Lisandväärtust toodavad klienditoe osakonnad – klientide küsimuste kiire ja kvaliteetne lahendamine avaldab positiivset mõju klientide rahulolule ja tervikuna ettevõtte mainele (Gonzalez *et al.*, 2005: 390).

Klienditoe spetsialistid omavad teadmisi ettevõtte toodete/teenuste kohta ja vahendavad neid lõppkasutajatele. Levinumateks kommunikatsiooni kanaliteks on telefonikõne, e-kiri, foorum või jututuba, harva võivad esineda isiklikud kohtumised. Kõiki klientide pöördumisi saab laias laastus olla kahte sorti – probleemile lahenduse otsimine ja infopäring. Esimese puhul klient kontakteerub klienditoeiga siis, kui tal on tekkinud probleem toote või teenuse kasutamisel: süsteem ei tööta, väljund ei vasta nõuetele. Teise variandi puhul probleemi veel ei pruugi olla ja kasutaja lihtsalt vajab informatsiooni tulevaste toimingute jaoks (Heckman ja Guskey, 1998: 66). Mõlemat laadi pöördumiste hulgast 60-70% moodustuvad rutiinsed, korduvad päringud (Gonzalez *et al.*, 2005: 390), millest tuleneb üks olulisematest klienditoe osakondade probleemidest – rutiinsete päringutega tegelemine võtab suurema osa klienditoe spetsialisti tööajast, erakordsete probleemide käsitlemisel võib esineda ressursside nappus.

Inimressurssidega seotud kitsendusi saab lahendada mitmel erineval moel (Bulchand-Gidumal ja Melian-Gonzalez, 2009: 207,208; Mohapatra, 2009: 145):

- klienditoe osakonnas töötavate spetsialistide arvu suurendamine;
- klienditoe spetsialistide hankimine organisatsioonist väljaspool (*outsourcing*);
- klienditoe töö automatiseerimine.

¹ Kaja Tampere (2002). Pisasjad mõjutavad kliendi valikuid. –*Äripäev*. [WWW] <http://www.aripaev.ee/default.aspx?publicationid=31503ED6-39D4-4163-9D98-74AA1E3959CE&code=88380> (04.08.2014)

Esimesed kaks varianti ei lahenda klientide päringute korduvuse probleemi, pigem kompenseeritakse seda suurema spetsialistide arvuga. Pikaajalises perspektiivis see ei pruugi olla parim lahendus, sest tuleb arvestada ettevõtte ja selle klientide arvu võimaliku kasvuga ja järelkult ka inimressursi nappuse probleemi uuesti esile kerkimisega. Samuti tuleb arvestada uute inimeste väljaõpetamis- ja ettevalmistuskuludega. Spetsialistide väljast tellimine üldjuhul on ettevõtte jaoks odavam, kuid mida keerulisem ja unikaalsem on toetatav toode või teenus, seda keerulisem on leida sobilikke spetsialiste väljastpoolt organisatsiooni (Bulchand-Gidumal ja Melian-Gonzalez, 2009: 207,208). Veel üheks kitsenduseks on ärisaladuste olemasolu. Kolmanda variandi peamiseks ideeks on rutiinsete küsimuste ennustamine ja iseteeninduse võimaldamine, mille tulemusena klient saab ise oma probleemiga toime tulla ja ei pea pöörduma klienditoe poole. Selline lähenemine lahendab lisaks ka klienditoe spetsialistide informeerituse probleemi – toetatavate toodete/teenuste arendamisel on inimesel raske olla kõikide uuendustega kursis (Göker ja Roth-Berghofer, 1999: 134), süsteem on aga võimeline seda tegema.

Käesoleva töö eesmärgiks on arendada klienditoe automatiseerimise lahendus äritarkvara arendamisega tegeleva ettevõtte jaoks.

Püstitatud eesmärgi saavutamiseks on vaja lahendada kolm ülesannet:

1. Luua teadmusbaas probleemide ja nende lahendustega;
2. Luua süsteemi esimene prototüüp, milleks on vaja arendada algoritm kliendi päringust probleemi tuvastamiseks ja tuvastatud probleemile lahenduse pakkumiseks.
3. Hinnata saadud süsteemi efektiivsust ning sobilikkust ettevõtte eesmärkidega.

Esimese ülesande juures on vaja teha kaks sammu. Esimeseks sammuks on klientide tihti esinevate küsimuste ja nende lahenduste korjamine alamsüsteemi. Kuna küsimused kipuvad korduma ja nende sõnastus ei ole identne, siis järgmiseks sammuks on topelt küsimustest vabanemine.

Teise ülesande juures kliendilt saadud päringut töödeldakse ja võrreldakse probleemide teadmusbaasi kirjetega. Vastavuse leidmisel, kliendile pakutakse tuvastatud probleemile sobilikku lahendust, mida võetakse teadmusbaasist.

Kolmandaks ülesandeks on saadud süsteemi hindamine – tähelepanu pööratakse nii süsteemi otsestele tulemustele kui ka sellele, kas saadud süsteem vastab ettevõtte eesmärkidele.

Käesolevat tööd tehakse Directo äritarkvara näitel, mis kujutab endast *Enterprise Resource Planning* (ERP) ja *Customer Relationship Management* (CRM) valmislahendust. Tegemist on 14 aastat Eesti turul tegutseva kodumaise ettevõttega, mille toodet hetkel kasutab ligi 1600 erineva suurusega firmat. Directo klienditoe osakonnas töötab hetkel kaheksa inimest, kelle peale igapäevaselt tuleb keskmiselt 10 tundi telefonikõnesid ja ligi 50 e-kirja (üldinfo aadressile). Uute kasutajate arvu märkimisväärse kasvuga näitajad suurenevad pidevalt. Klienditoe tööprotsesside automatiseerimine vähendaks kasutajatoe koormust, tõstaks teenindamise kvaliteeti. Edukalt loodud lahendust saab tulevikus kasutada uute klienditoe spetsialistide väljaõppel.

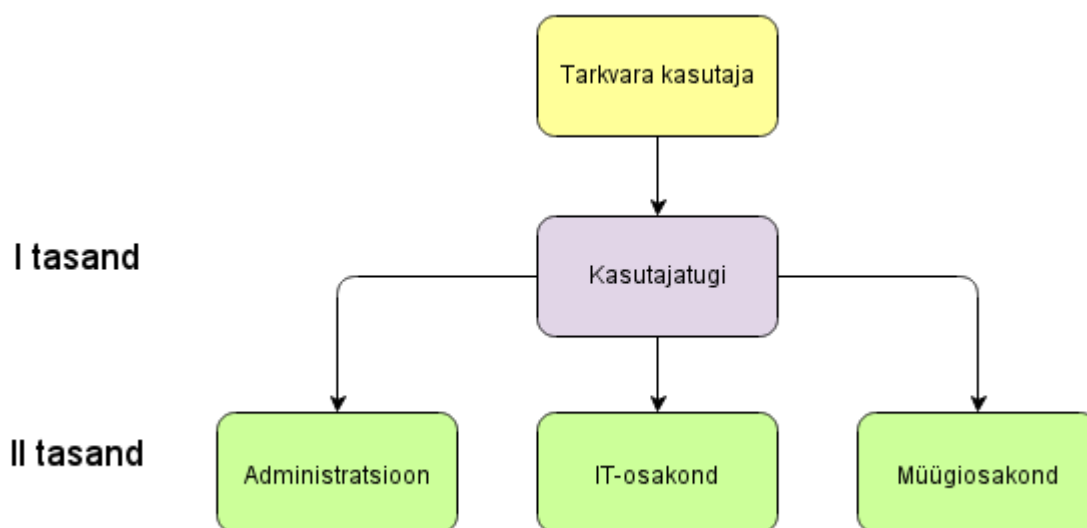
Autor tänab oma juhendajat ja töökaaslast, kes olid suureks toeks töö ettevalmistamisel.

1. Olukorra kirjeldus

1.1 Directo OÜ olemasolevad praktikad

Directo OÜ on äritarkvara arenduse ja hooldamisega tegelev väikeettevõtte, kus käesoleva töö kirjutamise ajal oli 24 töötajat.² Ettevõttes ei ole ametlikku liigendust osakondadeks, kuid loogiliselt saab välja tuua järgmised: administratsioon, IT-osakond, müügiosakond ning kasutajatoe osakond.

Kasutajatugi on kahetasandiline (Joonis 1).



Joonis 1: Directo OÜ kasutajatoe ülesehitus

Esimese tasandi moodustab kasutajatoe osakond ise, mis koosneb 8 konsultandist, kes vastavad Directo äritarkvara kasutajate telefonikõnedele ning ettevõtte infoaadressile saadetud e-kirjadele. Kasutajatoes töötavad inimesed on läbinud organisatsioonisisese koolituse ning omavad teadmisi nii tarkvara ülesehitusest ja funktsionaalsusest kui ka raamatupidamisest. Juhul kui tarkvara kasutaja poolt esitatud küsimus ei ole kasutajatoe pädevuses, pöördub kasutajatugi kas IT-osakonna (tegemist on tarkvara vea või uue funktsionaalsuse loomisega), müügiinimese või juurutaja (esitatud küsimus puudutab väga spetsiifilist tööprotsesside korraldust, mis on välja ehitatud konkreetse ettevõtte jaoks või kasutaja nõuab tarkvaraalast

² Firmast. *Directo OÜ*. [WWW] <http://www.directo.ee/et/firmast.html> (24.06.2014)

koolitust) või administratsiooni poole (näiteks arveldusega seotud küsimused), kes moodustavadki teise tasandi toe.

Esimese ja teise tasandi vaheline suhtlemine käib enamasti Skype'i vahendusel. Avatud ruumidega kontor soodustab suulisi vestlusi – nii saavad vastatud lihtsamad küsimused, mis ei nõua dokumenteerimist. Uusi arendussoove ja avastatud vigu dokumenteeritakse – nende kohta sisestatakse süsteemi sündmus (*ticket*).

Tänapäeval ei ole põhjalikku juhendit Directo kohta, kuid programmi paljudest moodulitest saab F1 klahvi abil kutsuda spikreid koos selle mooduli kohta käiva seletusega. Siinkohal on suuremaks probleemiks tarkvara pidev arendamine, mille tõttu olemasolev abi jääb täiendamata ja ei pruugi vastata hetkeolukorrale. Üheks põhjuseks on tööressurside nappus – konsultandid on pidevalt hõivatud hetkeküsimuste lahendamise ja puudub aeg programmi täiendamiseks.

1.2 Olemasoleva olukorra eelised ja puudused

Praeguse olukorra suuremaks eeliseks on pidev suhtlemine Directo kasutajate ja Directo töötajate vahel. Directo meeskond saab kiiret tagasisidet enda töö kohta otse kasutajatelt; suulise informatsioonikanali kasutamine (infotelefon) soodustab sõbralikke suhete tekkimist, probleemide korral kasutajaga saab nendest rääkida ja leevendada negatiivset tunnet. Paljud Directo kasutajad kiidavad just kasutajatoe sõbralikkust ja abivalmidust, mõnikord see mängib rolli tarkvara kasutusse võtmisel (Intervjuu Martin Rinnega, 21.04.2014).

Samas on ka päris mitu puudust. Esiteks, tarkvara kasutajal ei ole põhimõtteliselt mingit võimalust ise leida probleemile lahendus – suurt ja kõikehõlmavat kasutusjuhendit ei ole, olemasolevad juhendid võivad olla puudulikud ning, mis on tähtsam, juhendite saamiseks tuleb pöörduda kasutajatoe poole (nad ei ole avalikus veebis kättesaadavad). Samas, eriti uutel tarkvara kasutajatel, tekkib palju küsimusi erinevate tarkvara moodulite ja väljade kohta: tavalisteks küsimusteks on “Mida see nupp teeb” või “Milleks on see lahter mõeldud”. Kasutajatoe koormus oleks märkimisväärselt alandatud, kui sellistele küsimustele saaks kasutaja ise vastuseid leida. Eraldi tuleb mainida küsimusi, mis puudutavad tarkvara käitumist seoses kehtiva seadusandlusega (näiteks puhkusetasu arvestamise ja väljamaksmise kord). Tihti kasutajatoe konsultandid teavad, kuidas tarkvara peab käituma, kuid selleks, et seda käitumist tõestada, tuleb otsida vastavat seadust. Sellised olulised viited võiksid olla otse tarkvarast kättesaadavad, et tarkvara käitumine oleks alati lõppkasutajal selge ja arusaadav.

Teiseks, kasutajatoel on kindel töögraafik – tööpäeviti 9.00 – 17.00, nädalavahetusel saab abi e-posti kaudu, kuid see tähendab, et probleemide lahendamine võtab palju rohkem aega kui telefoni teel. Suurem osa Directo kasutajatest on kontoritöötajad, kelle töögraafik kattub Directo kasutajatoe omaga. Keerulisem on olukord poekettidega, kes kasutavad Directo kassa süsteemi ning töötavad nii õhtuti kui ka nädalavahetustel. Näiteks kaardimakseterminalist tulnud tšekk veateatega “XML süntaksi viga” võib katkestada töövoo. Kuigi lahendus võib olla lihtne – terminalis sai lihtsalt paber otsa, saab kasutajatoelt abi alles järgmisel hommikul. Lahenduseks võiks olla uue konsultandi palgale võtmine, kes oleks nädalavahetustel tööl, kuid see tähendab rahalisi kulusid Directo OÜ jaoks ning kuna talle ei saa tagada täistööaega, siis see ei ole hetkel otstarbekas.

Lisaks esineb ka psühholoogilisi probleeme. Osad inimesed üritavad erinevatel põhjustel vältida telefonikõnesid: keegi arvab, et liiga lihtsa küsimuse esitamisel näitab ta enda ebakompetentsust, osade jaoks on barjääriks võõrkeelne aktsent või ebakindlus enda keeleoskuses. E-posti teel küsimuste esitamine võib mõnikord olla komplitseeritud – näiteks kui on tegemist eriti tundlike andmete või firmasaladusega ning kasutaja ei soovi saata olukorra kirjeldust üldisele infoadressile. Selliste olukordade puhul aitaks süstematiseeritud küsimuste ja nende lahenduste olemasolu, et tarkvara kasutaja saaks ise otsida oma küsimusele vastust ning ainult siis, kui variandid on proovitud ja selgub, et tegemist ei ole primitiivse veaga, kasutaja julgeb pöörduda kasutajatoe poole.

Selge, et hetkel kasutusel olev lahendus on lihtne ja toimiv, kuid tarkvara kasutajate arvu pideva kasvu tõttu tekkib vajadus olemasoleva süsteemi parendamiseks.

1.3 Nõuded lahendusele

Peamiseks ideeks on optimeerida kasutajate päringute käsitlemist. Lihtsamate probleemide lahendamine võiks olla automaatne ja nõuda minimaalset kasutajatoe spetsialistide sekkumist. Oodatavaks tulemuseks on rutiinsete küsimuste arvu vähenemine.

Uuest lahendusest on huvitatud järgmised osapooled:

- tarkvara kasutaja:
 - tekkib võimalus ise leida enda probleemile lahendus
 - ööpäevaringne tarkvaraalne abi

- Directo kasutajatugi
 - väheneb rutiinsete küsimuste arv
- Directo OÜ juhtkond
 - parem tööressurside jaotamine
 - eduka lahenduse puhul pareneb tarkvara maine

Lahendus peab olema veebilehitseja põhine ja asuma Directo tarkvara veebilehel ja seega oleks lihtsasti kättesaadav kõikidele tarkvara kasutajatele ning ei nõuaks kasutaja arvutisse installimist.

2. Lahenduse valik

2.1 Kliendi iseteenindamisvõimalused veebis

Klienditoe töö automatiseerimise võimaluseks on klientide sellise käitumise soodustamine, kus ta kõigepealt üritab ise oma küsimusele vastuse leida. Selleks võimaldatakse kasutajale ligipääs informatsioonile ja kõige aluseks on ettevalmistatud tekstide olemasolu (Gibson-Odgers, 2007: 69).

Informatsiooni vajaduse rahuldamise võimalused on erinevad. Üheks varasemaks *online*-klienditoe vormiks olid kuulutustahvlid (*Bulleting Board Systems*), mis suunasid lahendust otsivat kliendi **FAQ veebilehtedele** (*Frequently Asked Questions*) (Foo *et al.*, 2000: 130). FAQ veebilehtede ideeks on kasutajate tüüpiliste küsimuste ennustamine: toote kohta tihti esinevad küsimused pannakse kirja ja sobitakse kokku ekspertide vastustega.

Eeliseks on küsimuste kategoriseerimine, mis annab hea ülevaate olukorrast nii toote arendajatele kui ka kasutajatele. Suuremaks ohuks on aga artiklite mitteasjakohane uuendamine (Lockhart, 2005:142) ja uute artiklite lisamisel kategooriate ignoreerimine või valede kategooriate omistamine. Aegunud informatsioon ei soodusta klienditoe poole pöördumiste arvu vähenemist, kategoriseerimata artiklid aga võivad esile kutsuda informatsiooni ülevoolu – tekib informatsiooni dubleerimise oht³ ja kannatab veebilehe kasutusmugavus. Mitteasjakohase informatsiooni kasutamine võib põhjustada kliendi valet käitumist, mis omakorda võib väljenduda kliendi tööprotsesside aeglustumises ja järelikult saab olla rahaliste kaotuste põhjuseks. Lisaks inimesed kipuvad vältima mahukate artiklite lugemist ja paremal juhul lähevad oma küsimustega foorumitesse⁴ või pöörduvad kasutajatoe poole, halvemal juhul võib pikk FAQ kasutajal isegi tekitada muljet, et tegemist on probleemirikka tootega. FAQ loetelude lühendamine aga ei ole mõeldav, sest sellega piiratakse kättesaadava informatsiooni hulka (Moreo *et al.*, 2012: 9). Lahenduseks on tõhusa otsingu võimaldamine.

³ Moon, G. 10 Tips For Creating a Killer FAQ Page. [WWW] <http://todaymade.com/blog/10-tips-for-creating-a-killer-faq-page/> (31.07.2014)

⁴ Shapcott, T. 10 Reasons why FAQ's are bad for Intranets & Websites. [WWW] <http://www.tealeshapcott.com/intranet/10-reasons-faqs-bad-intranets-websites/> (31.07.2014)

Lisaks FAQ'ile võib informatsioon toote kohta esineda ka **artiklite ja juhendite kujul**. Informatsiooni hulgas orienteerumiseks kasutatakse standardseid otsingumootoreid: klient sisestab oma päringu ja leiab sobilikke lahendusi pakutud tulemuste hulgast. Loomuliku keele töötlemise algoritmid teevad otsingu veelgi intuitiivsemaks (Moreo *et al.*, 2012: 9). Otsingu tulemuste relevantsust saab parendada otsingumootorite optimeerimise tehnikate abil (Yalcin ja Köse, 2010: 488). Samas populaarsete otsingumootorite algoritmid ei pruugi just väga hästi sobida kliendi iseteenindamise põhimõtetega. Põhjus on lihtne – toote/teenuse veebileht ei pruugi sisaldada kõiki võimalikke märksõnu, mida klient võib kasutada oma päringu formuleerimisel. Näiteks klient võib küsida “uue telefoni installimist”, vastava artikli pealkiri aga võib olla “Üldkasutatavate telefonivõrkude varustamine” (Cui *et al.*, 2006: 96).

Selline lähenemine on kasutaja jaoks lihtne ja ei nõua spetsiaalseid oskusi; oht seisneb pigem suure informatsiooni hulga inimesele esitamises, kus vajalik informatsioon võib jääda märkamata. Samuti teeb viimase otsuse klient ise ja võib juhtuda, et ta valib pakutud artiklitest sobimatu lahenduse.

Tehisintellektil põhinevad veebirakendused valivad ja esitavad ise lõppkasutajale sobilikuma vastuse või lahenduse. Üldjoontes sellised süsteemid kujutavad endast teadmusbaasi ja analüütikamootori integratsiooni. Kasutajaliides on ühe juurdepääsupunktiga, kliendi küsimuse vastuvõtja rollis võib olla *bot*, kes vajadusel esitab täpsustavaid küsimusi olukorra selgitamiseks, millele järgneb sobilikuma lahenduse esitamine (Cui *et al.*, 2006: 97). Süsteemide intelligentsuse tase võib varieeruda väga lihtsast, nagu soovitus õigeaegselt käivitada teatud protseduur, väga keerukate ja paindlike otsustussüsteemideni (Sandhu, 2008: 221).

Lähenemise eeliseks on intuitiivse ja kasutajasõbraliku keskkonna loomine, kus kliendil pole vaja süveneda probleemi lahenduse metoodikasse, intelligentne süsteem teeb kõike automaatselt ja pakub efektiivsemat lahendust (Zemke ja Woods, 1999: 325). Efektiivsuse tagamiseks tuleb vältida olukordi, kus kliendi päringu sisestamise ja tulemuse saamise vahele tekib ebaloosulik paus (Knapp ja Woch, 2002: 97) – kliendile tuleb tagada probleemi lahendus reaajas, vastasel juhul langeb kasutusmugavuse tase ja klient ei pruugi kasutada süsteemi tulevikus uuesti.

Klienditoe töö automatiseerimise ja inim-konsultandi töökoormuse vähendamise kontekstis pakuvad erilist huvi tehisintellektil põhinevad süsteemid, mis üritavad minimeerida kliendi

otsustamise vajadust sobiliku vastuse või lahenduse valimisel. Käesolevas töös keskendutakse selle lähenemise vaatlemisele.

2.2 Tehisintellektil põhinevate algoritmide ülevaade

2.2.1 Ekspertsüsteemid

Ekspertsüsteemiks nimetatakse arvutisüsteemi, mis põhineb etteantud valdkonna ekspertide teadmistel ning üritab emuleerida inimese tegevusi, mis nõuavad arutamist ja teatud kogemust. Ekspertsüsteemid on levinud diagnostika, andmete interpreteerimise, finantsprognoside valdkondades.⁵

2.2.1.1 Reeglipõhised ekspertsüsteemid (RBES)

Reeglipõhiseid süsteeme luuakse probleemide lahendamiseks etteantud valdkonnas; probleemide lahendamine toimub inim-ekspertide poolt eelnevalt kirjeldatud reeglite alusel. Kuna informatsioon on tihti esitatud hierarhilise otsustuspuu kujul, siis lahenduse leidmine on loogiline ja suhteliselt kiire (Kriegsman *et al.*, 1993: 19).

Reeglipõhiseid ekspertsüsteeme iseloomustab nende püsivus, reeglite loomine mitme inim-eksperti poolt suurendab süsteemi usaldusväärsust (Engin *et al.*, 2014: 22-23). Lähenemise puudusteks on teadmusbaasi loomise keerukus – protsess on aeganõudev, uute reeglite lisamine ja olemasolevate reeglite muutmine on komplitseeritud. Lisandub võimatus kasutada süsteemi olukordades, milleks ta ei ole disainitud (Kriegsman *et al.*, 1993: 19).

2.2.1.2 Juhtumipõhised ekspertsüsteemid (CBR)

Juhtumipõhiste süsteemide ideeks on üritada õppida eelnevate kogemuste põhjal ja pakkuda probleemile analoogne või assotsiatiivne lahendus (Fong ja Hui, 2001: 218). Üks juhtum (*case*) sisaldab probleemi kirjeldust, vastuseid täpsustavatele küsimustele, ja assotsieerub konkreetse probleemi lahendamise tegevusega. CBR lõhub etteantud probleemi märksõnade hulkadeks, mida on võimalik võrrelda eelnevalt sisestatud juhtumitega ja kontrollida nende sarnasust. Sellist lihtsustatud lähenemist kasutavad otsingumootorid: otsingu päring töödeldakse

⁵ Expert System. [WWW] https://en.wikipedia.org/wiki/Expert_system (10.09.2014)

märksõnadeks, eemaldatakse tihti esinevad sõnad ehk stopp-sõnad (*stop words*), seejärel võrdlevad spetsiaalsed algoritmid päringus sisalduvaid märksõnu veebilehtede märksõnadega. Sarnast protsessi teostab ka CBR (Wood ja Howlett, 2008: 352).

Meetodi eeliseks on suhteliselt lihtne juhtumite baasi täiendamine. Reeglipõhise otsustamismeetodiga võrreldes sobib CBR paremini struktureerimata teadmiste jaoks (Chan *et al.*, 1999: 1062, 1064; Wood ja Howlett, 2008: 351). See on üks levinumatest klienditoe automatiseerimise lähenemistest.

2.2.2 Klassifitseerijad

Klassifitseerimisalgoritmide ülesandeks on omistada etteantud klass vaadeldavale objektile. See on oluline komponent infootsingu (*information retrieval*) teostamiseks – klasside abil sobitatakse kokku kasutaja sisestatud päringut sellele sobiliku vastusega. Klassifitseerijate kasutusvaldkondade loetelu hõlmab loomuliku keele töötlemist, otsingumootoreid, mallide tuvastamist jne.⁶

2.2.2.1 Naiivsed Bayesi klassifitseerijad

Naiivsed Bayesi klassifitseerijad kujutavad endast juhitud õppega algoritmide kogumit, mis põhineb lihtsustatud oletusel, et atribuutide väärtused on tinglikult sõltumatud antud sihtväärtuse puhul. Meetodit kasutatakse masinõppes ning andme- ja tekstikaevandamises, see on levinud meditsiiniliste diagnooside automaatse püstitamisega tegelevates süsteemides (Islam *et al.*, 2007: 1542; Farid *et al.*, 2014: 1938).

Tekstikaevandamise kontekstis seisneb naiivse Bayesi põhimõtte selles, et erinevad sõnad ilmuvad sama teemaga tekstis üksteisest sõltumata. Selline oletus on väga naiivne, kuid vaatamata sellele meetod on efektiivne – sõnade grammatilised ja semantilised sõltuvused on samad ükskõik millise teemaga tekstide puhul.⁷

Meetodi eelisteks nimetatakse (Farid *et al.*, 2014: 1940;):

⁶ Statistical classification. [WWW] https://en.wikipedia.org/wiki/Statistical_classification#Application_domains (10.09.2014)

⁷ Рекомендательные системы: теорема Байеса и наивный байесовский классификатор. (2012). [WWW] <http://habrahabr.ru/company/surfingbird/blog/150207/> (04.09.2014)

- kasutamise lihtsust;
- õpetamise kiirust ja lihtsust: ei nõuta suuri õppehulkasid, tõenäosuse väljaarvutamiseks piisab ühest õppeandmete skaneerimise korrast;
- puuduvate väärtuste käsitlemise oskust: neid jäetakse vahele atribuutide klassi kuuluvuse tõenäosuse arvutamisel.

Puuduseks on vajadus taastada klasside tõenäosuste funktsioone lõplike andmete valimite pealt (Воронцов, 2008: 9).

2.2.2.2 K-lähima naabri algoritm (k-Nearest Neighbor – kNN)

KNN algoritm on üks levinumatest tekstide kategoriseerimise ja tekstikaevandamise meetoditest. Selle algoritmi põhimõte seisneb selles, et etteantud objektile omistatakse see klass, mis on kõige rohkem levinud selle objekti naabrite seas. Iga näidis peab olema seda ümbritsevate näidistega sarnaselt klassifitseeritud, seetõttu kui näidise klass ei ole teada, saab seda ennustada vaadates näidise naabrite klasse. Juhul kui on antud teadmata klassiga näidis ja õppeandmed, arvutatakse kõik nendevahelised kaugused. Õppeandmetest väikseima kaugusega näidis määrab etteantud objekti klassi (Imandoust ja Bolandraftar, 2013: 606).

Tekstikaevandamises kasutatakse natuke modifitseeritud lähenemist – erinevate klasside jaoks kasutatakse erinevat lähimate naabrite arvu (standardsel kujul k lähimate naabrite arv peab olema fikseeritud kõikide klasside jaoks). Sellisel juhul k -väärtus vastab vaadeldava klassi suurusele, mis tõstab meetodi efektiivsust ja langetab selle tundlikkust (*ibid.*, 609).

KNN algoritmi tugevateks külgedeks on selle lihtsus, intuiitsus ja head klassifitseerimise tulemused erinevates valdkondades. Nõrkusteks nimetatakse algoritmi aeglust siis, kui on tegemist suure õppeandmete kogumiga – see on seotud vajadusega arvutada iga päringu kaugust igalt õppenäidiselt. Lisaks on kNN väga tundlik asjakohatutele ja liigsetele atribuutidele, sest kõik atribuudid lähevad sarnasuse võrdlemisele ja klassifitseerimisele (*ibid.*, 608).

2.2.3 Närvivõrgud

Närvivõrgud töötlevad informatsiooni, imiteerides inimese aju struktuuri ning nende kasutamise eeliseks on võimekus töödelda ükskõik millist tüüpi sisendandmeid (Knapp ja

Woch, 2002: 98). Loomuliku keele töötlemise kontekstis räägitakse närvivõrkudest tihti kui mallide tuvastamise (*pattern recognition*) mehhanismidest.

Majevvki ja Zuraba (2008: 633) kirjeldavad järgmist meetodit lausete tuvastamiseks. Kõigepealt lüüakse sisendtekst eraldi sõnadeks, mida töödeldakse kodeerimise algoritmide abil – sõnu teisendatakse binaarseks koodiks. Kodeeritud sõnad on sõnade tuvastamise närvivõrgu sisendid. Õppeandmete hulk koosneb samuti kodeeritud sõnadest ja saab olla õpetatud sõnade kui lause osa tuvastamiseks. Teiseks komponendiks on lausete tuvastamise närvivõrk, mis käsitleb sisendeid vektoritena ja kasutab treenimiseks faili koos võimalike lausetega.

Närvivõrkude tugevuseks on nende madal tundlikkus õppimisandmetes sisalduvate vigade suhtes. Puuduseks saab nimetada vajadust luua väga suur baas õppimisandmetega (Knapp ja Woch, 2002: 98). Veel üheks kitsenduseks on närvivõrkude tulemuste ennustamatus ja võimalus neid “üle õpetada” – sellisel juhul tuleb õppimisprotsessi nullist korrata (Tepandi, 2012: 44).

2.3 Parema alternatiivi valik

Käesolevas töös püstitatud ülesandeks on klienditoe töö automatiseerimine läbi rutiinsete probleemide süstematiseerimise ja nendele vastuste leidmise veebikeskkonna loomise. Sobilikuma meetodika valimisel tuleb lähtuda nii võimalike tehnoloogiate eelistest kui ka arvestada ettevõtte valdkonna ja ärieesmärkidega.

Oluliste kriteeriumite hulgas on:

- **Päringu töötlemise kiirus:** süsteem peab olema võimeline leidma kliendi probleemile vastuse ja esitama seda viivitamatult, pikad ja ebaloomulikud pausid ei ole lubatud. Näiteks, tuleb välistada olukorda, kus süsteem ei väljasta lahendust koheselt vaid saadab seda ekspordile ning süsteemi kasutaja saab vastuse oma probleemile alles tundide pärast. Vastasel juhul kliendil on lihtsam ja kiirem pöörduda kasutajatoe poole, mis on vastuolus kasutajatoe automatiseerimise eesmärgiga.
- **Laiendatavus:** küsimuste ja vastuste baasi peab olema võimalik kiiresti täiendada.
- **Lahenduste kvaliteedi tagamine:** ettevõtte ekspertidel peab säilima kontroll esitatavate lahenduste üle. Ettevõtte tegutseb valdkonnas, kus sobimatu lahendus võib väljenduda

valesti väljamakstud palkades, valesti deklareeritud maksudes, valesti arvatud toodete omahindades jne. Hetkel vastutavad lahenduste õigsuse eest ettevõtte klienditoe spetsialistid, sama taseme kontroll peab säilima ka automaatse klienditoe süsteemi puhul.

- **Andmed peavad paiknema enda serveritel:** ettevõtte kliendid jagavad enda tundlikke andmeid, seega ei tohi juhtuda andmete leket. Reeglina välised rakendused ei ava oma koodi, järelikult ei saa kontrollida, et andmete leket ei toimu.
- **Veebipõhisus:** süsteem peab olema Directo tarkvaraga integreeritav. Ettevõtte poliitika on suunatud veebipõhise äritarkvara arendamisele, mis ei nõua kasutaja arvutisse eraldi paigaldamist, klienditoe automatiseerimise süsteem peab toetama neid põhimõtteid.
- **Hind:** süsteemi arendamine ei tohi olla ettevõtte jaoks rahaliselt ja ajaliselt kulukas.

Kõik vaadeldavad alternatiivid toetavad ettevõtte enda serverile andmete paiknemist ning veebipõhisuse põhimõtet, mille tõttu edasi pööratakse tähelepanu teistele valiku kriteeriumitele.

Teostamise poolest kõige lihtsamaks variandiks on koostada **juhendid** tarkvara moodulite kohta ning panna neid avalikku veebi. Lähenemise peamine puudus on tingitud kasutajate iseloomust – enamus inimesi tahab saada oma küsimusele kohese vastuse. Directo klienditoe osakonna kogemus näitab, et mõnikord keelduvad kasutajad juhendite saamisest, sest nad ei taha kulutada aega nende läbitöötamisele ning eelistavad helistada klienditoe telefonile. Samuti, kuigi juhendite koostamisega hakkavad tegelema klienditoe eksperdid, ei saa tagada seda, et kasutaja valib enda probleemi lahendamiseks sobilikumat juhendit. Seega juhendite avalikustamine on kindlasti abiks klienditoe töö automatiseerimisel, kuid seda ei saa vaadelda ainsa variandina.

Üheks olulisemaks kriteeriumiks on ekspertide kontroll kliendile pakutavate lahenduste üle. Kõige paremini tagavad seda ekspertsüsteemid, mille loogika panevad paika inimesed-ekspersedid, ning tänu sellele on süsteemi tulemused selgelt jälgitavad. **Reeglipõhised ekspertsüsteemid** ei ole aga piisavalt paindlikud ning uute küsimuste ja lahenduste tekkimisel tuleb teha muudatusi ka reeglites. See teeb reeglipõhiste ekspertsüsteemide arendamise ja haldamise keerukaks ja aeganõudvaks protsessiks, ning järelikult ka rahaliselt kalliks, mis ei ole antud juhul sobilik.

Juhtumipõhine ekspertsüsteem on piisavalt paindlik, et kasutusjuhtude (ehk probleemide ja lahenduste) teadmusbaasi saaks jooksvalt täiendada ja selleks ei peaks muutma terve süsteemi loogikat. Samas kuna on tegemist ekspertsüsteemiga, siis inim-ekspertidel säilib kontroll olukorra üle. Kõik probleemid ja lahendused saavad olla ekspertide poolt valideeritud, ning kui kasutusjuhtude tekkimist saab vähegi automatiseerida, siis on võimalik minimeerida andmete sisestamist ning ekspertide ülesandeks jääb ainult andmete ülevaatamine ja vajadusel korrigeerimine. Seega sellise süsteemi haldamine ei tohi olla ettevõtte jaoks kulukas.

Probleemide ja lahenduste sobitamine klassifitseerijate või närvivõrkude abil on võimalik ja võimaldab õpetada süsteemi uute küsimustega ise toime tulema. **Klassifitseerijad** on tõhusad teatud klasside ja kategooriate olemasolul ehk nende abil saab tuvastada kliendi probleemi tüüpi (näiteks kas on tegemist arvelduse või pigem laoarvestuse kohta käiva küsimusega). See on aga ainult osa ülesandest, sest pärast probleemi tüübi identifitseerimist on veel vaja valida leitud probleemi tüübi kohta sisalduvatest lahendustest see ainuke, mis sobib esitatud probleemi lahendamiseks. Seega klassifitseerijad ei ole antud ülesande juures kõige paremaks alternatiiviks, kuid neid saab kasutada alamülesannete teostamisel.

Närvivõrk on võimeline õppima probleemide lahendamist õppeandmete põhjal ja selle suuremaks tugevuseks on võimekus lahendada probleeme, mis ei sisaldu õpiandmetes. Antud meetod kasutab „musta kasti“ loogikat, mis tähendab seda, et ei ole võimalik täpselt ennustada järeldamise protsessi käiku ja tulemust. Sisuliselt see tähendab, et kaob ära ekspertide kontroll klientidele pakutavate lahenduste üle. Kuigi seda aspekti saab leevendada tõstes vastuste usaldusväärsust läbi võimalikult suure õpiandmete hulga kasutamise, jääb ikkagi oht et küsimusele antakse sobimatu lahendus. Samuti kui eelnevalt kehtinud lahendus muutub seadusandluse muudatusega vääraks ja mitteasjakohaseks, ei piisa ainult uue lahenduse sisestamisest. Vana lahendus peab olema näidisandmetest eemaldatud ja närvivõrk peab olema uuesti ümber õpetatud, teste tuleb samuti korrata. Arvestades sellega, närvivõrkude kasutamine terve süsteemi raames ei vasta püstitatud nõuetele, kuid neid saab kasutada alamülesannete jaoks.

Klienditoe automatiseerimise süsteemi jaoks sobib kõige paremini juhtumipõhine ekspertsüsteem, kuna see vastab olulisematele valimiskriteeriumitele. Teiste alternatiivide kasutamine on ressursikulukam ettevõtmine, sest on otstarbekam kohe kirjeldada probleemide ja nendele vastavate lahenduste paarid kasutusjuhtude kujul, kui hakata neid sobitama erinevate

metoodikate abil. Samal ajal klienditoe osakond hakkab tegelema tarkvara kasutusjuhendite koostamisega, mis on toeks klientide iseseisvuse tõstmisel.

3. Lahenduse realiseerimine

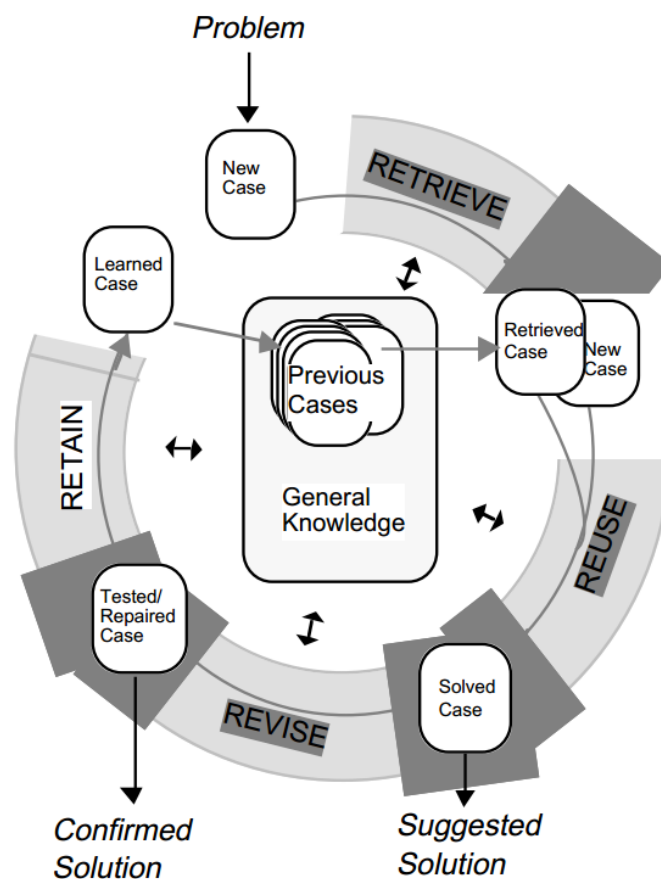
3.1 Süsteemi projekteerimine

Klienditoe töö automatiseerimist hakati teostama arendades juhtumipõhist ekspertsüsteemi, mille kasutajaliides näeb välja kui küsimusele vastamine. Idee seisneb selles, et kasutaja esitab oma küsimuse läbi veebikeskkonna, küsimus töödeldakse kohe süsteemi poolt, tulemusena pakutakse kasutajale kõige sobilikum lahendus.

Tüüpilise juhtumipõhise ekspertsüsteemi töötsükkel koosneb neljast etapist (Aamodt ja Plaza, 1994: 7-8; Aha, 1998: 1-2):

- Leidmine (*retrieve*) – süsteemi sisestatud probleemile otsitakse paremini sobiv juhtum või juhtumid.
- Taaskasutamine (*reuse*) – eelmises etapis saadud juhtumid üritatakse kasutada sisestatud probleemi lahenduseks. Juhul kui leitud juhtum ei ole probleemile täpne vastavus, üritatakse juhtumis sisalduvat lahendust adapteerida uuele probleemile.
- Hindamine (*revise*) – pakutud adapteeritud lahendust testitakse ja hinnatakse selle asjakohasust. Testimine võib esineda enne lõppkasutajale lahenduse esitamist või pärast seda.
- Salvestamine (*retain*) – kui pakutud lahendus on tunnistatud õigeks, salvestatakse see koos probleemi kirjeldusega uue juhtumi kujul.

Joonis 2 illustreerib juhtumipõhise ekspertsüsteemi töötsükli.



Joonis 2: Juhtumipõhise süsteemi töötsükkel (Aamodt ja Plaza, 1994: 46)

3.2 Juhtumite baasi projekteerimine

Juhtumipõhise süsteemi toimimise eelduseks on juhtumite baasi olemasolu. Juhtumi kirjeldus peab minimaalselt koosnema kahest komponendist: üheks on probleemi kirjeldus ja teiseks lahenduse kirjeldus.

Lahenduse osa sisaldab ekspertide poolt sisestatud ja üle vaadatud juhiseid probleemse olukorra lahendamiseks. Probleemi kirjelduse alusel aga teostatakse sobilikke juhtumite otsingut – juhtumipõhised ekspertsüsteemid kasutavad analoogilist järeldamist (Aamodt ja Plaza, 1994: 41). See tähendab seda, et kasutaja sisestatud päringut hakatakse võrdlema juhtumite baasis olevate probleemide kirjeldustega. Kui sarnane juhtum on leitud, siis on suur tõenäosus, et kasutaja probleemi saab lahendada analoogselt juhtumis kirjeldatud probleemiga ning järelikult juhtumis sisalduv lahendus peab olema sobilik.

Esimese prototüübi projekteerimisel otsustati piirduda ainult ühe tarkvara mooduli kohta käivate küsimustega. Ühe mooduli katmine võimaldab esiteks selle detailsemat läbitöötamist ning teiseks – kõikide moodulite kohta käivad tüüpilisemad küsimused on enamasti analoogsed, muutuvad vaid väljade ja nuppude nimetused. Arusaadav, et igas tarkvara moodulis on oma spetsiifika (näiteks laosissetulekute loogika erineb kassa-arvete omast) ja nad kajastavad erinevaid tööprotsesse, kuid käesoleva töö ülesanne seisnes rutiinsete küsimuste katmises, mille tõttu selline piiritlemine oli mõttekas ja ei mõjuta lõpptulemusi. Otsustati, et keskendutakse müügiarvete moodulile, sest seda moodulit kasutab põhimõtteliselt iga ettevõtte, vaatamata sellele, mis tegevusvaldkonda ta kuulub. Müügiarvete kohta esitatakse igapäevaselt küsimusi ning selle mooduli tugi on peamine ettevõtmine.

Seega esimeseks ülesandeks oli aru saada Directo klientide küsimustest, mis on seotud müügiarvete mooduliga. Saadud informatsioon näitab, mis kujul ja milliseid küsimusi kliendid hakkavad süsteemile esitama. Saadud andmete põhjal on võimalik projekteerida juhtumite baasi (sh töötada välja juhtumi kirjeldamise malli) ning saada aru kasutaja sisestatud päringu töötlemise printsiipidest.

3.2.1 Algandmete ettevalmistamine

3.2.1.1 Algandmete korjamine

Küsimuste analüüsil otsustati lähtuda Directo tarkvara kasutajate poolt klienditoele saadetud e-kirjadest. Kuigi päeva jooksul esitatakse Directo klienditoele rohkem küsimusi telefoni kui e-posti teel, telefonikõnede salvestust ega kirjalikult dokumenteerimist ei teostata. Klienditoe spetsialisti poolne küsimuse sõnastus võib aga suuresti erineda sellest, kuidas tavakasutaja formuleerib oma päringut. Samuti võib inimene väljendada ennast erinevalt sõltuvalt sellest, kas on tegemist suulise või kirjaliku küsimusega – kirjaliku küsimuse puhul jääb inimesel võimalus oma sõnastuse üle vaatamiseks ja korrigeerimiseks, selline tekst on telefonikõnega võrreldes kompaktsem. E-kirjade kaasamine võimaldab paremani aru saada just sellest, kuidas kasutajad formuleerivad kirjalikke pöördumisi ning säilib kasutajapoolne sõnastus.

E-kirjade analüüsimise eeliseks paistis ka võimalus proovida automatiseerida probleemide ja vastuste paaride väljavõtmist, nagu oli kirjeldatud Marom ja Zukermani töös (2007: 1665). Idee seisnes selles, et kasutada klientide poolt saadetud e-kirju küsimuste näidisteks ning Directo ekspertide poolt saadetud vastuseid lahenduste kirjeldusteks. Selleks olid Microsoft Exchange e-kirjade serverist eksporditud aastatel 2011–2014 Directo infopostkastile saabunud e-kirjad.

E-kirjade lähimal vaatlemisel ja analüüsimisel selgus, et nende töötlemisega kaasnevad järgmised kitsendused:

- enamuse kirjadest on formuleeritud väga isiklikult, eeldatakse, et tegemist on juba tuntud ettevõtte tausta, probleemi või inimesega;
- kirjades sisalduvad tundlikud andmed (nimed, firmade rekvisiidid, jne);
- e-kirjades sisalduv info on tihti puudulik ja eeldab kliendiga uuesti ühenduse võtmist või probleemi vaatamist tema kasutajakonto alt.

Arvestades nende kitsendustega selgus, et e-kirjade töötlemise algoritmi kirjeldamine on ajaliselt kulukam kui kirjade sisust küsimuste käsitsi väljavõtmine. See ei olnud käesoleva töö primaarne eesmärk ning otsustati töödelda e-kirjade baasi käsitsi. Igal juhul automaatselt töödeldud kirjade sisu nõuaks spetsialistide poolt üle vaatamist.

Sobilikke e-kirjade andmebaasi loomine koosnes mitmest etapist. Kõigepealt kõikidest e-kirjadest valiti ainult need, mille sisus leidis sõna „arve“. Saadud valimist jäeti alles ainult need kirjad, mis moodustasid lühikese, küsimusest ja vastusest koosneva kirjavahetuse. Kuna saadud valim oli ikkagi töötlemiseks liiga suur, siis otsustati jätta ainult 1000 värskeimat kirjavahetust. Valitud e-kirjad vaadati üle, mitterelevantset e-kirjad kustutati. Kirjade sisust jäeti ainult küsimused ise, kõik muu, s.h ka tundlik informatsioon, kaotati. Tulemusena saadi rutiinsete küsimuste valim, mis koosneb kokku 300 probleemi kirjeldusest.

E-kirjade analüüsimisel kerkis esile veel üks probleem – üldiselt oli tegemist mittestruktureeritud tekstiga, leidis juhtumeid, kus ühes lauses võis sisalduda mitu küsimust korraga. Veel keerulisemaks tegi olukorra see, et need küsimused ei pruukinud olla omavahel seotud ning isegi võisid puudutada tarkvara erinevaid mooduleid. Juhul kui sellise ebamäärase probleemi kirjeldusega puutub kokku inimene-konsultant, siis tema kõigepealt üritab probleemi klassifitseerida – üritab saada aru, mis tarkvara mooduliga on tegemist, mis on probleemi iseloom (näiteks kas on tegemist veateate või pigem informatsiooni päringuga). Seetõttu otsustati, et kasutaja päringu süsteemipoolne töötlemine peab sisaldama ka klassifitseerimise etappi. Lisaks sellele, et sellel etapil toimuks kasutaja päringu klassi tuvastamine, mängiks see filtri rolli, et kitsendada kasutusjuhtude kandidaatide arvu.

3.2.1.2 Alandmete klassifitseerimine

Küsimuste klasside määramine teostati juba ettevalmistatud e-kirjade põhjal, algandmete klassifitseerimise protsess teostati ekspertide poolt. Kõik probleemide kirjeldused vaadati uuesti üle, igale küsimusele omistati silt, mis kajastab probleemi iseloomu (veateade, välja tähenduse täpsustus jne). Seejärel analüüsiti juba silte: kontrolliti, et ei oleks mitut sama tähendusega, kuid erineva sõnastusega silti, veenduti, et kõik võimalikud küsimused on saanud siltidega kaetud. Lõpuks saadi neli silti ehk kasutajate rutiinsete küsimuste klassi:

- **Andmevahetus:** sellesse klassi kuulusid küsimused, mis on seotud arvete liikumisega ühest süsteemist teise (näiteks Directo tarkvarast muusse tarkvarasse või e-postiga saatmine).

Näide: Kuidas on võimalik saata arve eArvekeskusesse?

- **Nupp:** siia kuulusid küsimused, mis on seotud arve mooduli nuppude tähenduse või kasutamisega.

Näide: Kuidas arvele saab lisada nuppu „Mail“?

- **Veateade:** küsimused, mis olid seotud programmist saadud veateatega.

Näide: Kuidas on võimalik kinnitada arve, kui süsteemist tuleb veateade „Ladu on vale või puudu“?

- **Väljatrükk:** arve väljatrükkimisega seotud küsimused.

Näide: Kuidas on võimalik vahetada arve väljatrükkil olevat logot?

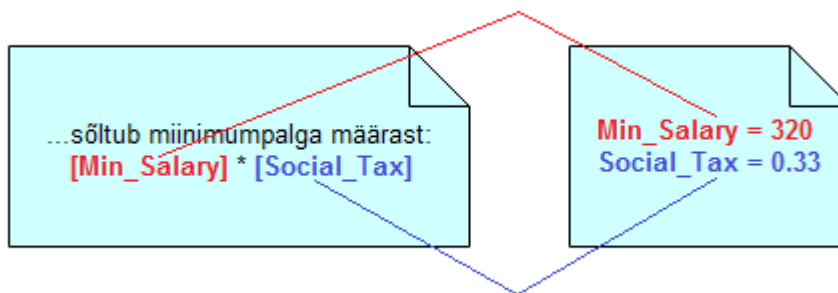
Siis kui klasside loetelu oli selgunud, omistati kõikidele valimis olevatele küsimustele nende õige klass ning seega oli ette valmistatud probleemide alg- ehk õppeandmete kogum.

3.2.2 Juhtumite kirjeldamine

Juhtumi peamised osad on probleemi kirjeldus ja sellele sobilik lahendus. Nende sõnastuste eest hakkavad vastutama klienditoe eksperdid.

Probleemi kirjelduse osa pidi olema ühtlasi arusaadav, informatiivne ja selgesti sõnastatud, sest just selle alusel hakatakse hindama kas käesolev juhtum on kasutaja probleemiga analoogne. Selleks, et juhtumite baasi oleks mugav hallata, pidi juhtumitel olema võimalik määrata kehtivuse alguse ja lõpu kuupäeva. Esiteks andis see võimaluse juhtumi asjakohasuse kaotamisel säilitada juhtumi kirjeldus ajaloo jaoks. Teiseks saavad eksperdid ette valmistada juhtumeid tulevikuks ja lülitada neid sisse või välja alates teatud kuupäevast.

Lahenduse kirjeldus kujutas endast struktureeritud teksti, mis annab selgeid juhiseid juhtumis kirjeldatud probleemi lahendamiseks. Muutuvad andmed, nagu näiteks arvud, valemid ja viited seadustele, esitati muutujate kujul (Joonis 3). See andis süsteemile juurde paindlikkust ja võimaldas tagada juhtumite baasi asjakohasuse.



Joonis 3: Juhtumi kirjelduses muutujate kasutamise näide

Tabel 1 illustreerib juhtumi kirjeldamise malli.

Tabel 1: Juhtumi kirjeldamise näide

Parameeter	Näide
Probleem (<i>problem</i>)	Arve kinnitamisel tuleb veeteade „Ladu on vale või puudu“
Kehtivuse algus (<i>starts_at</i>)	01.01.2000
Kehtivuse lõpp (<i>ends_by</i>)	31.12.2100
Lahendus (<i>solution</i>)	Veenduge, et dokumendi päise lahtrisse „Ladu“ on märgitud lao kood ning et

Parameeter	Näide
	<p>sisestatud ladu on eelnevalt süsteemi lisatud (seda saate kontrollida vajutades Alt-V lahtris „Ladu“).</p> <p>Kui ilmub teade, et sisestatud lao kood ei ole leitud, saate seda lisada tehes topelt kliki lahtris „Ladu“ ning vajutades nuppu „Lisa uus“.</p>

Juhtumite dubleerimise vältimiseks – juhul kui muutub probleemi sõnastus, kuid mitte probleemi sisu – hakatakse kasutama otseviiteid. Probleemi sõnastus salvestatakse eraldi tabelisse koos viitega relevantsele juhtumile, uut juhtumit sellisel juhul ei looda (Joonis 4). See lihtsustab juhtumite baasi haldamist ning teeb probleemi lahenduse otsingu kiiremaks.



Joonis 4: Salvestatud päringu ja juhtumi seos

3.3 Juhtumite leidmine

Juhtumite leidmise etapi eesmärgiks on tuletada kasutaja sisestatud päringule sobilik lahendus juhtumite andmebaasist. Kuigi juhtumipõhised süsteemid on laialt levinud kasutajate probleemide lahendamisel, ei ole ühte standardset algoritmi sobilikke juhtumite leidmiseks ja nende relevantsuse hindamiseks. Üldiselt koosneb juhtumite leidmise etapp järgmistest sammudest (Barbera *et al.* 2006: 666-667; Knapp ja Woch, 2002: 97-98; Wood ja Howlett, 2008: 352):

- Kasutaja päringu töötlemine – kasutaja päringut lemmatiseeritakse ehk viiakse algsele vormile, tihti soovitatakse eemaldada stopp-sõnu, kuigi mõnedel juhtudel

stopp-sõnade eemaldamine põhjustab vajaliku informatsiooni kadu ning seega viib halvemale tulemusele (Moreo *et al.*, 2012: 11; Yelati ja Sangal, 2011: 371).

- Töödeldud päringu klassifitseerimine – päringu klassi kasutatakse filtrina juhtumite tuletamisel.
- Sobilike juhtumite leidmine – juhtumite baasist otsitakse vastava klassiga juhtumeid, tuletatud juhtumid sorteeritakse algsele päringule relevantsuse järgi.

3.3.1 Lemmatiseerimine

Kasutaja päringu töötlemise esimeseks sammuks on selles sisalduvate sõnade lemmatiseerimine. Lemmatiseerimise protsessi käigus leitakse sõnale tema süntaktiline tüvi, üldjuhul see eeldab käände, arvulise ja ajalise vormi eemaldamist, sõna morfoloogiline tuletus jääb puutumata (Monz, 2003: 574). Kuna kasutaja päringule hakatakse edaspidi otsima vastavusi juhtumite baasist, siis juhtumites sisalduvad probleemide kirjeldused peavad ka olema lemmadena esitatud. Lemmade leidmiseks otsustati kasutada välisserverit⁸ ja salvestada tulemused andmebaasi. See andis võimaluse oma sõnaraamatu koostamiseks ning leitud lemmade taaskasutamiseks. Raamatupidamise valdkonnas on levinud oma sõnavara kasutamine, seetõttu sõnastiku koostamise etapil oli mõistlik pöörata tähelepanu võimalike terminite ja lühendite kajastamisele. Vajaliku leksikoni allikana kasutati firma e-kirjade baasi.

Teksti lemmatiseerimise eelduseks on teksti lekseemideks lahti löömine (*tokenization*). Käesolevas töös kasutati sõnade koti mudelit (*bag-of-words*), mille järgi tekstid on esitatud üksiksõnadena, sõnade grammatiliste seoste ja sõnade järjekorraga ei arvestata, sõnade korduvuste arv on säilitatud.⁹ Sõnade eraldajaks saavad olla tühikud ja punktuatsiooni märgid. Seega teksti lemmatiseerimine teostati kolme sammuga:

1. Töötlemiseks kasutati 300 probleemi kirjeldust, mis saadi klientide e-kirjadest;
2. Probleemide kirjeldused lõhuti lahti üksiksõnadeks, kokku saadi 3023 sõna, millest unikaalseid oli 976 sõna;
3. Unikaalsetele sõnadele leiti nende lemmad, millest unikaalseid lemmasid oli 511.

⁸ Eesti keele lemmatiseerija, FiloSoft. [WWW] http://www.filosoft.ee/lemma_et/ (24.10.2014)

⁹ Bag-of-words model. [WWW] http://en.wikipedia.org/wiki/Bag-of-words_model (10.11.2014)

Kõik sõnade vormid ja nende lemmad salvestati eraldi tabelitena ning kirjeldati sõnade lemmadeks teisendamise algoritm. Välisserveri poole pöördumist teostati vaid juhul, kui sõnaraamatus puudus vaadeldava sõna lemma. Uue lemma leidmisel toimus sõnaraamatu täiendamine.

Joonisel 5 on toodud näide teksti lemmatiseerimise sammudest. Tulemusena saadud lemmade loetelu on sobilik edasiseks töötlemiseks.

Algne tekst	Kuidas saan kinnitada arvet, kui tuleb veateade, et ladu on vale või puudu?
Sõnad eraldatud	kuidas, saan kinnitada, arvet, kui, tuleb, veateade, et, ladu, on, vale, või, puudu
Lemmad	kuidas, saama, kinnitama, arve, kui, tulema, veateade, et, ladu, olema, vale, või, puuduma

Joonis 5: Teksti lemmatiseerimise sammud

Lemmade leidmisel tuli välja päringute õigekirjaga seotud probleem. Grammatilised vead raskendavad tekstide töötlemist, süsteem ei leia sõnale sobilikku lemmat ei sõnaraamatus ega välisserveri rakenduse andmebaasis. Probleemiga võideldakse mitmel moel, näiteks:

- Kasutaja päringut saab saata otsingumootorisse (näiteks Google Inc. oma), juhul kui päringus sisaldub viga, tagastab otsingumootor kõige tõenäolisema sõnastuse, mida süsteem saab oma töös edasi kasutada.
- Saab mõõta kasutaja poolt sisestatud sõnade korrelatsiooni süsteemis sisalduvate sõnade lemmadega. Otsingumootorile pöördumist teostatakse ainult juhul, kui süsteemi poolt leitud vastavuse kandidaatide arv on null või mitu.
- Forsseerida päringu väljade automaatset täitmist süsteemis sisalduvate variantidega. Selleks saab kasutada n-gram¹⁰ lekseeme, näiteks, kui kasutaja sisestab sõnast vähemalt kolm sümbolit, siis automaattäitja pakub talle kõik võimalikud variandid, kus on need sisestatud sümbolid järjest olemas.

Käesolevas töös otsustati kasutada otsingumootorile päringut juhul, kui tekstis sisalduvale sõnale ei olnud võimalik leida tema lemmat.

¹⁰ N-gram, [WWW] <http://en.wikipedia.org/wiki/N-gram> (10.11.2014)

3.3.2 Stopp-sõnade eemaldamine

Tihti soovitakse teksti eeltöötlemise protsessi kaasata stopp-sõnade eemaldamist. Need sõnad ei mõjuta päringu tähendust ja nende eemaldamine kiirendab päringu edaspidist töötlemisprotsessi. Levinud keelte puhul on kättesaadavad stopp-sõnade loendid, eesti keele stopp-sõnade loendit avalikust veebist ei leitud, mistõttu koostati oma stopp-sõnade loend.

Puudub ühtne arvamus, mis hetkel tuleb teostada stopp-sõnade eemaldamine – kas enne või pärast lemmatiseerimist. Stopp-sõnade eemaldamine enne lemmade leidmist vähendab lemmatiseerimist vajavate sõnade arvu, mis võib suurte tekstide puhul teha töötlemise oluliselt kiiremaks. Samas see tähendab, et stopp-sõnade loendis peavad sisalduma kõik võimalikud stopp-sõnade vormid ning kui mingi sõna vorm on puudu, siis seda ei tuvastata stopp-sõnana ning see pääseb edasistele töötlemise etappidele. Kui esialgu teostada teksti lemmatiseerimist, siis seda probleemi saab leevendada, ka stopp-sõnade loend saab oluliselt lühemaks. Käesolevas töös otsustati kasutada teist varianti ning teha stopp-sõnade eemaldamine pärast päringu lemmatiseerimist.

Stopp-sõnade leidmiseks kasutati IDF (*Inverse Document Frequency*) ehk dokumendi pööratud esinemissagedust. Idee seisneb selles, et harvemini esinevad sõnad hakkavad suurema tõenäosusega esinema relevantsetes dokumentides. Seega antud dokumentide kontekstis nende sõnade tähtsust ja informatiivsust hinnatakse kõrgemateks kui teistel sõnadel. IDF arvutamine toimub järgmise valemi abil (Lo *et al.*, 2005: 4):

$$idf_k = \log\left(\frac{N_{Doc}}{D_k}\right), \text{ kus}$$

N_{Doc} tähendab dokumentide arvu ja D_k kajastab dokumentide arvu, mis sisaldavad terminit k . Mida väiksem on sõna IDF näitaja, seda väiksem on selle sõna tähtsus antud dokumentide kollektsiooni kontekstis.

Tekstide allikana kasutati klientide e-kirjadest saadud probleemide kirjeldusi:

- iga sõna jaoks arvutati tema IDF;
- sõnade loend järjestati IDF järgi ja esitati kasvavas järjekorras;

- sõnad, mille IDF on alla 2, salvestati stopp-sõnade loendis. Piirmäära väärtus valiti konkreetselt vaadeldava dokumentide kollektiooni jaoks – IDF näitaja suuruste analüüsil selgus, et enamus sõnadest, mille IDF on suurem kui 2, võivad juba omada semantilist koormust. Standardset piirmäära väärtust ei ole, sest IDF sõltub suuresti kontekstist;
- saadud stopp-sõnade loend valideeriti ekspertide poolt (et loendisse ei oleks sattunud üksik sõna, mille tähendus võib olla oluline informatsiooni leidmisel). Näiteks stopp-sõnade loendist eemaldati sõna „arve“, sest kuigi see esines iga probleemi kirjelduses, omab see tähendust päringust arusaamisel. Seda enam, et probleemide loend oligi valitud selle võtmesõna järgi. Teisteks stopp-sõnade loendist eemaldatud sõnadeks olid näiteks klient, kinnitama, muutma.

Tabelis 2 on toodud mõned näited sõnadest ja nende IDF väärtustest. Punasega on märgitud stopp-sõnad, sõna „arve“ on erand, mis oli loetelust eemaldatud.

Tabel 2: Sõnad ja nende IDF

Sõna	IDF
kuidas	0,394336
arve	0,460088
saama	0,532639
olema	0,537602
kas	0,69897
selline	1,875061
tekkima	1,875061
allahindlus	2
finantskanne	2,176091
hoiatus	2,176091

3.3.3 Klassifitseerimine

3.3.3.1 Andmete ettevalmistamine

Päringu töötlemise meetodika on mõnes mõttes sarnane eksperdi arutuskäiguga – kõigepealt üritatakse aru saada, mis tüüpi küsimusega on tegemist ehk klassifitseerida saadud probleem. Levinud klassifitseerimise meetodid nagu tugivektorite masin või naiivsed Bayesi klassifitseerijad, on tõhusad kahe klassiga süsteemides, näiteks spämmi/mittespämmi tuvastamisel. Käesolevas töös vaadeldava ettevõtte puhul toodi välja neli probleemide klassi – andmevahetus, nupp, veateade ja väljatrükk. Selliste mitme klassiga süsteemide juures soovitavad Knapp ja Woch (2002: 97-98) teostada klassifitseerimist närvivõrgu abil. See võimaldab väga erinevate sisendandmete töötlemist ning oskab toime tulla selliste olukordadega, kus on rohkem kui kaks klassi.

Närvivõrgu sisend peab olema esitatud numbriliste väärtustega, mille tõttu kõik sõnastikus olevad sõnade lemmad nummerdati, päringus olevad lemmad teisendati nende numbriliste vastetega (Joonis 6). Juhtumite kirjeldused peavad olema klassidega täiendatud.



Joonis 6: Kasutaja päringu töötlemine närvivõrgu jaoks

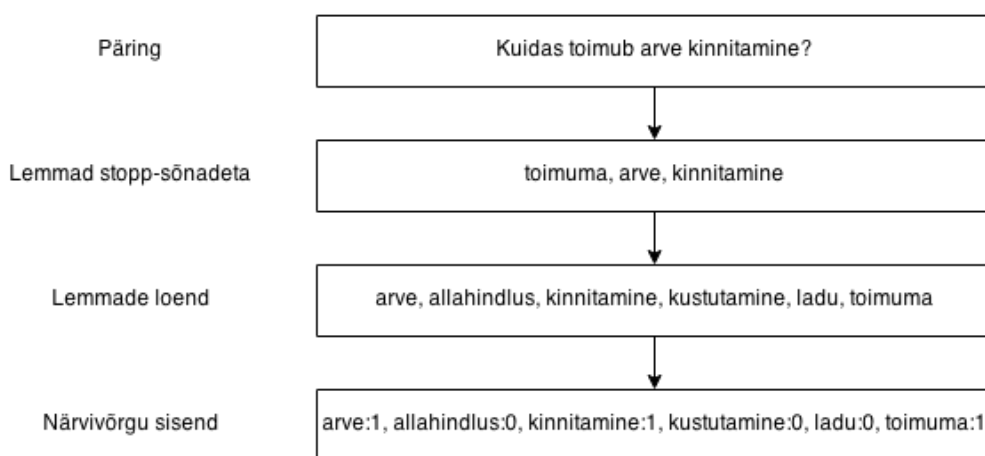
Projekti prototüüpimist teostati *Ruby* keeles, närvivõrgu jaoks valiti projekti *ai4r* (*Artificial Intelligence for Ruby*)¹¹, mis kasutab vea pöördlevimise meetodil (*backpropagation*)¹² põhinevat õppimisalgoritmi. Õppeandmeteks valiti käsitsi klassifitseeritud probleemide

¹¹ Ai4r, SergioFierens. [WWW] <https://github.com/SergioFierens/ai4r> (03.11.2014)

¹² Backpropagation. [WWW] <https://en.wikipedia.org/wiki/Backpropagation> (28.11.2014)

kirjeldused, 60 probleemi iga klassi kohta. Kuigi kokku oli saadaval 300 probleemide kirjeldust, ei olnud nende klassidesse jagunemine võrdne. Selleks, et vältida probleeme närvivõrgu õpetamisel, õppeandmetes sisalduv erinevate klassidega probleemide arv tehti võrdseks. Närvivõrgu testimiseks kasutati probleemide sõnastusi, mis ei sisaldanud õppeandmete hulgas. Seda tehti veendumaks, et närvivõrk on õppinud klasside tuvastamist ning ei jätnud lihtsalt õppeandmeid meelde. Testimiseks kasutati kokku 40 probleemi, 10 iga klassi kohta.

Sisendandmete atribuutide rollis oli unikaalsete lemmade loend. Probleemide kirjeldused esitati sõnade esinemise statistikaga. Joonis 7 illustreerib sisendandmete ettevalmistamist.



Joonis 7: Päringu töötlemine närvivõrgu sisendiks

3.3.3.2 Närvivõrgu esimene teostus

Parema tulemuse saavutamiseks viidi läbi mitu katsetust erinevalt seadistatud närvivõrguga. Üldiselt oli närvivõrgu struktuur järgmine:

- Sisendkiht: neuronite arv on dünaamiline ja sõltub unikaalsete lemmade arvust süsteemis (katsetuste hetkel – 511 neuronit).
- Õppimiskiht: kaks korda suurem kui väljundikiht, sest närvivõrgu tulemuslikkus sõltub tekkinud seoste arvust. Mida rohkem neuroneid on kaasatud, seda rohkem seoseid saab tekkida.¹³
- Väljundkiht

¹³ Искусственная нейронная сеть. [WWW] https://ru.wikipedia.org/wiki/Искусственная_нейронная_сеть (28.11.2014)

Õppimis- ja väljundkihi suurusega oli võimalik eksperimenteerida, iga katsetuse järel kontrolliti saadud närvivõrgu tulemuslikkust. Selleks kasutati testimisandmeid, kõik vastavused ja mittevastavused loeti üle ning nendest kujunes närvivõrgu klasside äraarvamise määr (Tabel 3).

Tabel 3: Närvivõrkude tulemused

Nr	Stopp-sõnad	Sisendkiht	Õppimiskiht	Väljundkiht	Tulemus
1	eemaldatud	511	8	4	55%
1a	jäetud	511	8	4	47%
2	eemaldatud	511	9	5	67%
2a	jäetud	511	9	5	59%
3	eemaldatud	511	18	9	82,5%
3a	jäetud	511	18	9	75%

Esimese katsena saadud närvivõrk valis õige probleemi klassi 55% juhtumites. Saadud tulemus ei olnud rahuldav, kuna probleemide klassifitseerimine pidi olema esimeseks sammuks kasutaja probleemi tuvastamisel ning probleemile vale klassi omistamine väljendub suure tõenäosusega mitteasjakohase lahenduse pakkumisel.

Krõlov (2012: 33) oma töös leiab, et väljundkihi suurendamisel lisaneuroni abil saab oluliselt parandada närvivõrgu tulemuslikkust. Seetõttu teise närvivõrgu õppimis- ja väljundkihti suurendati ühe neuroni võrra, probleemi klassi äraarvamise määr kasvas 67%-ni.

Kolmanda katsena suurendati veelkord õppimis- ja väljundkihi neuronite arvu. Neuronite arvu suurendamisel lähtuti ideest, et kuna närvivõrk ei ole kindel, mis klassiga on tegemist ja tihtipeale kahe erineva klassi tõenäosuse vahe on suhteliselt väike, siis neuronite arvu suurendamine annab vajalikku ruumi juurde ning ebatõenäolised seosed kaovad ära. Selline lähenemine parandas oluliselt närvivõrgu tulemust, õige klass tuvastati 82,5% juhtumitest.

Iga katse juures prooviti ka stopp-sõnade alles jätmist, kuid sellise sisendi puhul saadud tulemus oli alati halvem kui stopp-sõnade eemaldamisel (need katsed on tabelis 3 märgitud „a“ tähega).

Õppimiskihtide arvu suurendamine ei andnud samuti positiivset tulemust, selline närvivõrk määras kõikide testimisandmete klassideks ühe klassi, mistõttu õppimiskihtide arvuga katsetustest loobuti.

3.3.3.3 Närvivõrgu teine teostus

Närvivõrgu tulemuste parendamisel pakkus erilist huvi Knapp ja Woch (2002: 98-99) kirjeldatud tähtsuse vektoreid kasutatav närvivõrk. Sisuliselt on tegemist kaalude lisamisega, mis aitavad kaasa klassi valiku teostamisel. Kaalude leidmiseks analüüsiti kasutajate probleemide kirjeldusi ning määravaid sõnu, mis aitavad inim-eksperdil saada aru probleemi sisust.

Probleemide defineerimisel tihti kasutatavatest sõnadest moodustati sõnade grupid, kus ühte gruppi kuuluvatel sõnadel on lähedane tähendus. Kuigi tegemist ei ole sünonüümidega, vaadeldi neid antud juhul kui lähedase tähendusega sõnu. Vastavalt sellele toimus sõnade asendamine sõna grupi pealkirjaga. Näiteks ühte gruppi kuulusid sõnad, mis tähistavad erinevaid arvete edastamise tehnoloogiaid. Kuna kasutajad võivad oma probleemi sõnastamisel kasutada inglisekeelseid väljendeid, siis lisati ka need sõnade loendisse. Kokku saadi 23 sõna gruppi, mõned näited on toodud tabelis 4.

Tabel 4: Sõnade grupid

Sõna grupp	Sõnad
edastuskanal	edastuskanal, earvekeskus, telema, edisoft
importima	importima, eksportima, tõmbama, laadima
konto	konto, finantskonto, deebetkonto, krediidkonto, deebet, krediid
väljatrükk	väljatrükk, vorm, blankett, plank, layout, kujundus, xsl

Sõnade gruppide defineerimisele järgnes nende kaalude leidmise etapp. Iga gruppi kuuluva sõna jaoks arvutati tema ühes klassis esinemise sagedus ehk TF (*term frequency*). TF arvutus toimus järgmise valemi järgi:

$$TF(t, d) = \frac{n_i}{\sum_k n_k}, \text{ kus}$$

n_i tähistab sõna esinemissagedust dokumendis ning nimetajas on dokumendi sõnade arv kokku.¹⁴

¹⁴ TF-IDF. [WWW] <https://ru.wikipedia.org/wiki/TF-IDF> (28.11.2014)

Üksikute sõnade esinemissagedused kokku andsid sõna grupi esinemissageduse konkreetse klassis. Nendest näitajatest koostati koondtabel (tabel 5), mis oli aluseks närvivõrgu tähtsuse vektorite lisamisel.

Tabel 5: Sõna gruppide tähtsuse vektorid

Sõna grupp	Klass 1 (Väljatrükk)	Klass 2 (Andmevahetus)	Klass 3 (Veateade)	Klass 4 (Nupp)
email	0	0,614707	0,046706	0,338587
kinnitama	0	0	0,750633	0,249367
nupp	0,033064	0	0	0,966936
veateade	0,305853	0	0,694147	0
väljatrükk	0,913795	0	0	0,086205

Tabelist on näha, et mõned sõnade grupid ei ole teatud klasside juures üldse kasutatavad, näiteks „email“ võib olla suurema tõenäosusega kasutatud arvete andmevahetusega seotud küsimustes, mõnikord seda mainitakse nuppude kohta käivates küsimustes, kuid väljatrüki ja veateatega seotud küsimustes valitud grupi sõnu põhimõtteliselt ei kasutata.

Esimeses teostuses kirjeldatud närvivõrgu struktuur jäi samaks (sisendkiht: 511, õppimiskiht: 18, väljundkiht: 9 neuronit), leitud kaalud lisati närvivõrgu sisendile. Tulemusena saadud klasside äraarvamise määr moodustas 97,5%, mis on väga rahuldav ning lubab kasutada saadud närvivõrku kasutajate probleemide klassifitseerimiseks. Samas nii hea tulemuse saavutamine oli võimalik seepärast, et kasutati antud valdkonnas levinud leksikoni, tavalise kõne analüüsimine ja klassifitseerimine ei pruugi anda sellist tulemust.

3.3.4 Juhtumite leidmine

3.3.4.1 Esimene teostus

Juhtumipõhised süsteemid teostavad päringule vastavuste otsimist põhinedes analoogiatel. Sisuliselt see tähendab kasutaja päringu atribuutide võrdlemist koos juhtumi probleemi kirjelduse atribuutidega. Barbera *et al.* (2006: 667) nimetab suurimaks probleemiks selliste päringute ja probleemide sarnasuste leidmist, millel ühtib vaid osa atribuutidest või atribuudid kuuluvad erinevatesse klassidesse. Mainitud probleem on eriti aktuaalne selliste süsteemide puhul, kus juhtumid ja kasutaja päring on esitatud parameetrite kogumina (ravimite koostisosad jne). Käesoleva töö juures olid atribuutide rollis sõnade lemmad, seega antud probleem ei ole

nii terav. Erinevatesse klassidesse kuulumist lahendatakse kasutaja päringu klassifitseerimisega.

Juhtumipõhiste süsteemide üheks levinuks juhtumite leidmise algoritmiks on kNN. Idee seisneb selles, et arvutada päringu ja juhtumite kaugust, väiksema kaugusega päringu ja juhtumi paari peetakse sarnasteks. Käesolevas töös otsustati samuti kasutada kNN algoritmi teostust¹⁵, mis on sobiv prototüübi jaoks valitud Ruby keelega.

Testimise andmeteks kasutati 180 päringu sõnastust, nende struktuur jäi samaks nagu klassifitseerimise närvivõrgu puhul ehk tegemist on lemmatiseeritud ja stopp-sõnadest puhastatud päringute variantidega. Juhtumite baas koosnes 30 probleemi kirjeldusest, mis valmistati ette klienditoe ekspertide poolt ning mis erinesid testimisandmete sõnastustest. Päringu ja juhtumi probleemikirjelduse kauguse arvutamiseks kasutati Eukleidese kaugust (*Euclidean distance*) ja Jaccardi indeksit (*Jaccard index*).

Iga testimise rea jaoks tagastati kaks lähemat juhtumit. Saadud tulemusi võrreldi eksperdi poolt määratud tulemusega. KNN algoritmi tulemuslikkuse hindamiseks kasutati Aunimo *et al.* (2003: 200) pakutud varianti – kui esimene tagastatud tulemus on õige, siis teda hinnatakse 1 punktiga, kui õige on teine tulemus – saab ta ½ punkti jne. Kui ükski vastus ei ole õige, siis saadakse 0 punkti. Terve süsteemi hinnangu saamiseks arvutatakse keskmine hinne. Klienditoe küsimustele juhtumite sobitamise tulemused on näidatud tabelites 6 ja 7.

Tabel 6: kNN tulemused (esimene teostus)

kNN kauguse mõõdik	1. valik - õige	2. valik - õige	Süsteemi hinnang
Eukleidese kaugus	79,1%	5,81%	0,82
Jaccardi indeks	81,4%	10,5%	0,866

¹⁵ kNN, Red Davis. [WWW]
<https://github.com/reddavis/knn/tree/4f5a154dbec0de07ec435c1d2bb2be556b7d43e6> (30.11.2014)

Tabel 7: kNN tulemused päringute klasside lõikes

Päringu klass	kNN kauguse mõõdik	1. valik - õige	2. valik - õige	Süsteemi hinnang
Väljatrükk	Eukleidese kaugus	95,2%	0%	0,952
	Jaccardi indeks	95,2%	4,8%	0,976
Andmevahetus	Eukleidese kaugus	81%	9,5%	0,857
	Jaccardi indeks	76,2%	9,5%	0,81
Veateade	Eukleidese kaugus	60%	10%	0,65
	Jaccardi indeks	70%	20%	0,8
Nupp	Eukleidese kaugus	92,9%	0%	0,929
	Jaccardi indeks	92,9%	0%	0,929

Selgus, et Jaccardi indeksi kasutamine päringu ja juhtumi vahelise kauguse leidmiseks andis üldiselt paremaid tulemusi. Samuti ligi 81% juhtudel osutus õigeks esimene pakutud juhtum, mis on üsna hea tulemus. Kui vaadata tulemusi päringu klasside lõikes, siis halvemaid tulemusi andis andmevahetusega seotud probleemide leidmine, kõige paremini aga leiti vastavusi väljatrükkidega seotud päringutele – 95,2% juhtudel süsteemi esmane valik osutus õigeks.

Ebaõnnestunud katsetuste analüüs pööras tähelepanu järgmisele probleemile – kuna päringu ja juhtumite võrdlus käib nendes sisalduvate sõnade alusel, siis on oodatav ja loogiline, et süsteem tuleb hästi toime nende päringutega, mille sõnastus on sarnane juhtumi omaga. Probleem tekib siis, kui päringus kasutatakse täiesti teistsuguseid sõnu ja väljendeid ning juhtumi kirjeldusega ühtib vaid päringu väike osa. Näiteks on see juhtunud andmevahetuse klassiga seotud päringutega. Probleemi leevendamiseks on kaks lahendust:

1. Lisada juhtumi kirjeldusse veel üks osa, mis koosneks just otsingu jaoks sobilikest sõnadest. Olemasolev juhtumi probleemi kirjeldus jääks juhtumi pealkirjaks, mis on inim-konsultantide juhtumites orienteerimise aluseks.

2. Kaasata juhtumite otsingusse kasutajate päringud, mis on seotud juhtumite kirjeldustega otseviidete abil. Mida rohkem kasutajate päringuid on salvestatud süsteemi, seda suurem on probleemide sõnastuse hulk, mis on saadav kNN algoritmi jaoks, ning seda täpsem on sarnaste juhtumite otsing.

Eraldi viidi läbi eksperiment päringute eelneva klassifitseerimise vajaduse hindamiseks. Selleks päringule sobiliku probleemi kirjelduse leidmiseks käivitati ainult kNN algoritm, päringu klassifitseerimise etapp jäeti vahele. Eksperimendi tulemused on toodud tabelis 8, ning on näha, et juhtumite leidmise täpsus langes, seega kinnitatakse päringu klassifitseerimise otstarbekust.

Tabel 8: kNN tulemused ilma päringu klassifitseerimise etapita

kNN kauguse mõõdik	1. valik - õige	2. valik - õige	Süsteemi hinnang
Eukleidese kaugus	69,8%	10,5%	0,75
Jaccardi indeks	75,6%	9%	0,802

3.3.4.2 Teine teostus

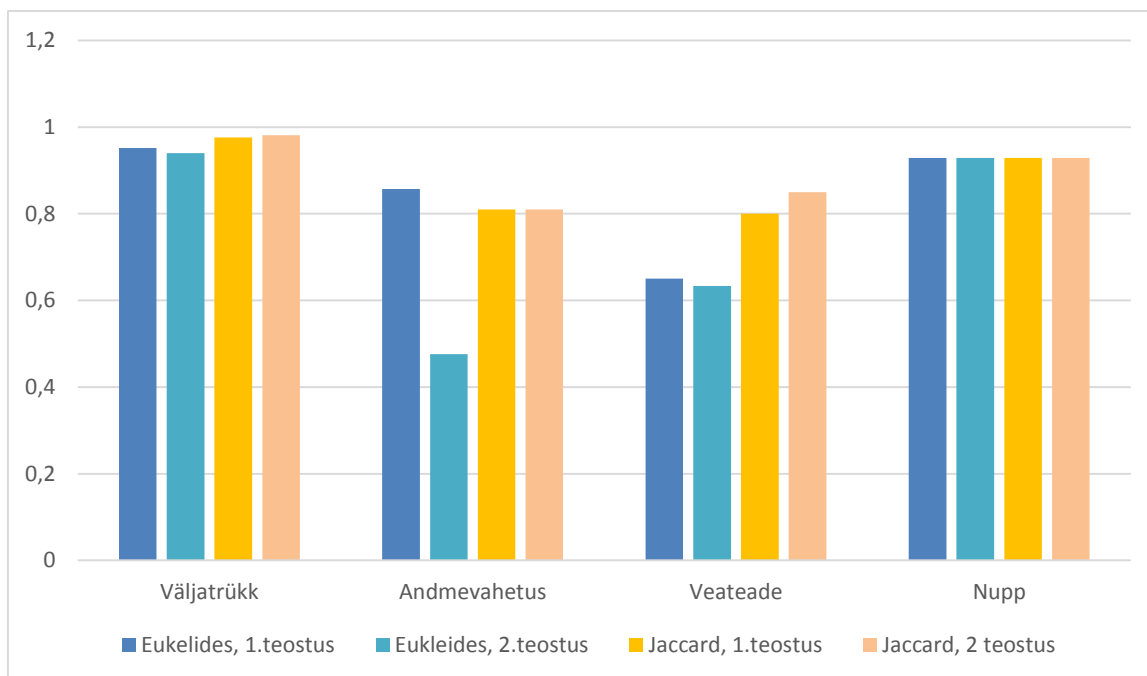
Juhtumite leidmise etapi teises teostuses üritati tõsta juhtumite leidmise täpsust. Kõigepealt analüüsiti esimeses teostuses ebaõnnestunud päringuid ning nendele sobilikke probleemide kirjeldusi. Juhtumites sisalduvad probleemide kirjeldused muudeti nii, et nende sõnastus oleks universaalsem ja sobiks suuremale päringute arvule. Saadud tulemused on näidatud tabelis 9.

Tabel 9: kNN tulemused (teine teostus)

kNN kauguse mõõdik	1. valik - õige	2. valik - õige	Süsteemi hinnang (vs 1.teostus)
Eukleidese kaugus	68,6%	7%	0,721 (-0,099)
Jaccardi indeks	83,7%	9,3%	0,884 (+0,018)

Nagu ka esimese teostuse puhul andis parema tulemuse Jaccardi indeksi kasutamine, kuid esimese ja teise teostuse tulemuste vahe ei olnud suur, süsteemi hinne paranes vaid 0,018 punkti võrra.

Kuna esimeses teostuses näitas kõige halvemat tulemust andmevahetuse ja veateate klassidega seotud päringute juhtumitega sobitamine, siis kõige rohkem tähelepanu pöörati just nende kahe klassiga seotud juhtumite probleemide kirjeldustele. Süsteemi tulemused klasside lõikes on näidatud joonisel 8.



Joonis 8: Esimese ja teise teostuse võrdlus päringute klasside lõikes

Otsingutulemuste parendamiseks juhtumite kirjelduste täiendamine probleemi uute sõnastustega avaldas positiivset mõju veateate klassiga seotud päringute ja probleemide sobitamisega. Kuid samas esimese ja teise teostuse tulemuste vahe ei olnud suur. Testimisandmete analüüs näitas, et süsteem leidis õige vastavuse mõnede esimeses teostuses ebaõnnestunud päringutele, kuid samaaegselt osad seni õnnestunud katsed said negatiivse tulemuse. Asi on selles, et juhtumite jaoks probleemide formuleerimine tehti vastavalt sellele, kuidas klienditoe ekspert sõnastaks tekkinud probleemi ehk oli tegemist selgesti formuleeritud ja üheselt mõistetava sõnastusega. Tarkvara kasutajad aga sõnastavad vahel oma probleemi nii, et isegi klienditoe konsultandil tuleb esitada täpsustatavaid küsimusi. Ebaõnnestunud päringud kujutavad endast just selliseid olukordi – päringud on väga pikad, tuuakse palju ebaolulisi detaile, mis takistavad päringule vastavuste leidmist.

Seega antud probleemi leevendati eelmises peatükis toodud teise võimaluse abil: süsteemi käivitamisel kasutaja päringute andmeid hakati salvestama süsteemi, päringute ja sobilikke juhtumite vahel hakkasid tekkima otseviited (sisuliselt iga juhtumiga hakkas olema seotud mitu

päringu sõnastust). Mida rohkem päringute sõnastusi on süsteemis olemas, seda rohkem on informatsiooni uute päringute sobitamiseks ning seda paremad on süsteemi tulemused.

3.4 Juhtumite taaskasutamine ja hindamine

Juhtumipõhiste süsteemide võlu on seotud **juhtumite taaskasutamisega** – kui kasutaja sisestatud probleemile ei leita täpset vastavust juhtumite baasis, siis süsteem, tuginedes kirjeldatud reeglitele, on suuteline pakkuma võimaliku lahenduse. Juhul kui pakutud lahendus osutub sobilikuks, see salvestatakse süsteemis koos kasutaja küsimusega uue juhtumi kujul.

Juhtumi lahenduse adapteerimisele saab läheneda kolmel moel (Adeyanju, 2011: 27):

- **Sõnasõnaline esitus.** Kasutajale esitatakse kõige sobilikuma juhtumi lahendus muutumatul kujul. Esitatud lahenduse adapteerimisega tegeleb inimene ise – see võib olla kas ekspert, kes muudab lahenduse sisu ning salvestab selle uue juhtumi kujul, või süsteemi lõppkasutaja, kes kohandab süsteemi poolt pakutud juhiseid oma olukorrale.
- **Transformeerimine.** Uuele probleemile sobilik lahendus koostatakse erinevate lahenduste osade kombineerimise abil. Selline lähenemine on võimalik pikemate protseduurude kirjelduste puhul. Näiteks printeri installimise juhendit, mis võib koosneda erinevatest sammudest sõltuvalt lõppkasutaja operatsioonisüsteemist, printeri versioonist jne, on võimalik jagada väiksemateks osadeks ning vajadusel neid kombineerida (*ibid.*, 29).
- **Asendamine.** Selle lähenemise puhul süsteem muudab lahenduses sisalduvaid parameetreid, et adapteerida lahendus kasutaja sisestatud päringule. Sellisel juhul juhtumid tavaliselt koosnevad teatud numbrilistest parameetritest või kategooriatest, näiteks ravimi kirjelduses sisalduvad koostisosad ja nende osakaalud on toodud juhtumi kirjelduses ja kui tekib vajadus asendada üks komponent teisega, siis lähtudes etteantud reeglitest pakub süsteem ravimi uue retseptuuri (*ibid.*, 29-30).

Käesoleva töö juures selline transformeerimist või asendamist eeldatav adapteerimine ei olnud võimalik, sest juhtumites sisalduvad lahendused on lühikesed, koosnevad tihti ühest-kahest lausest ja neid ei olnud võimalik lüüa lahku eraldiseivateks komponentideks. Lahenduste sisu süsteemipoolne modifitseerimine on keeruline, sest tuleb täpselt kirjeldada, milliseid sõnu tohib muuta ja millised mitte. Kuna tegemist oli loomulikus keeles kirjutatud tekstiga, mitte

atribuutide loendiga (nagu näiteks ravimi retseptuur), siis otsustati jätta adapteerimise funktsioon inimesele.

Samal ajal hakkab toimima ka pakutud lahenduste **hindamine**. Sisuliselt kasutaja valib kahest pakutud lahendusest sobilikuma ning märgib kas valitud lahendus sobis täielikult või osaliselt. Sobilikuma lahenduse valiku mugandamiseks hakatakse kasutajale näitama nii juhtumi lahendust kui ka juhtumis oleva probleemi kirjeldust. See on vajalik selleks, et süsteemi kasutaja saaks paremini aru, mis probleemi jaoks oli mõeldud pakutud lahendus. Kõigepealt selline täpsustus on mõeldud nende olukordade jaoks, millal kasutaja probleemi sõnastus suuresti erineb ekspertide poolt ettenähtud sõnastusest.

Töövoog näeks järgmiselt välja:

1. Kasutaja sisestab päringu.
2. Süsteem töötleb päringut ning pakub kaks võimalikku lahendust.
3. Kasutaja vaatab pakutud lahendused üle ning annab hinnangu, kas lahendus on täielikult sobilik, osaliselt sobilik või sobimatu. Kui mõlemad lahendused osutuvad sobimatuks, pakutakse kasutajale saata esitatud päring klienditoe e-postile.
4. Saadud tagasiside saadetakse klienditoe ekspertidele, kes analüüsivad kasutaja päringu ja lahenduse paari, teevad vajalikke muudatusi lahenduse kirjelduses ning salvestavad need uue juhtumi kujul.

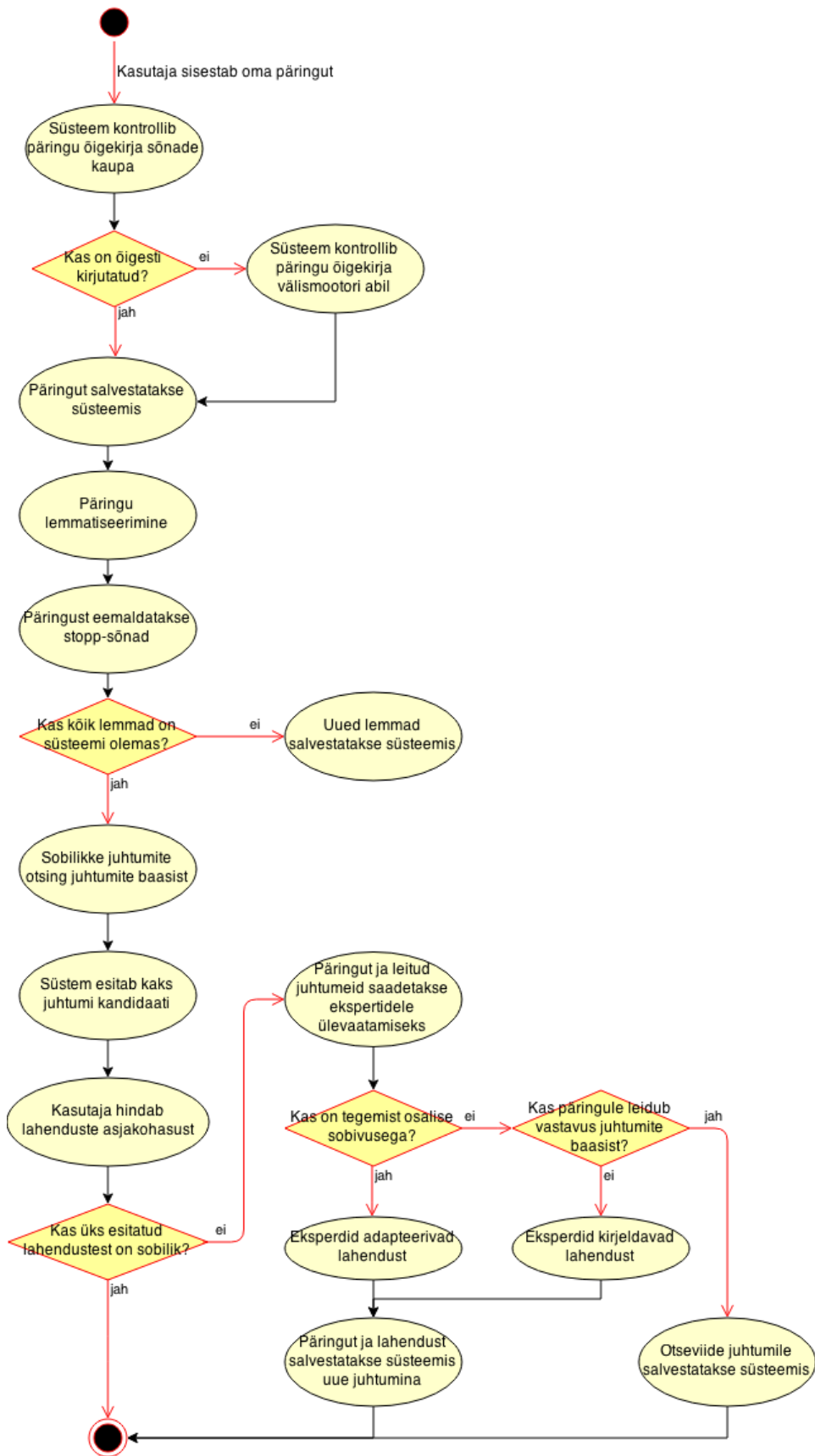
3.5 Juhtumite salvestamine

Käesoleva etapi eesmärgiks on salvestada kasutaja päring koos sobiliku vastusega juhtumite baasi. Juhtumi salvestamise eelduseks on süsteemi kasutajalt saadud hinnang päringu ja süsteemi leitud lahenduste sobivuse kohta. Süsteemi esimesel realisatsioonil salvestamise funktsioon on delegeeritud klienditoe ekspertidele, see tagab ekspertide kontrolli juhtumite baasis sisalduvate lahenduste üle. Kuigi esialgu salvestamisele kuuluvate juhtumite kandidaatide üle vaatamine võtab aega, siis tulevikus, kui juhtumite baas on kasvanud ning sisaldab palju erinevaid probleemide sõnastuste ja lahenduste variatsioone, uute juhtumite kandidaatide arv väheneb ning süsteemi haldamine saab olema ekspertide jaoks vähem ajakulukas.

Salvestamisele kuuluvad:

- Päringu ja lahenduse paar, mis on saanud osalise sobivuse hinnangu. Lahenduse adapteerimisega tegelevad klienditoe eksperdid ning kui vajalikud muudatused on tehtud, salvestatakse päringu ja uue lahenduse paar uue juhtumi kujul.
- Päringu ja lahenduse paar juhul, kui süsteem ei leidnud päringule sobilikku lahendust ning sellist probleemi ei ole enne juhtumina kirjeldatud. Klienditoe eksperdid analüüsivad kasutaja päringut. Kui on tegemist reaalse probleemiga (tuleb välistada olukordi, kus kasutaja sisestab mõttetu päringu, mis ei saa olla probleemi kirjelduseks) ning sellele ei ole juhtumite baasis ühtegi vastavust, siis kirjeldatakse päringule lahendus ning need salvestatakse uue juhtumi kujul.
- Päring ning otseviide lahendusele juhul, kui süsteem ei leidnud sellele sobilikku lahendust, sest päringu sõnastus on väga erinev olemasolevatest probleemide kirjeldustest ning vastav lahendus on juhtumite baasis juba olemas.

Terve süsteemi töövoogu illustreerib joonis 9.



Joonis 9: Klienditoe automatiseerimise juhtumipõhise ekspertsüsteemi töövoog kirjeldus

3.6 Süsteemi hindamine

Süsteemi hindamise protsessi käigus pööratakse tähelepanu klienditoe töö automatiseerimise lahenduse nõudmistele ning sellele, kuivõrd leitud lahendus vastab neile. Käesolevas töös püstitatud eesmärkidele vastas kõige paremini küsimustele vastamise intelligentse süsteemi loomine.

Süsteemi esimene prototüüp analüüsib kasutaja sisestatud päringut ning väljastab kaks lahenduse kandidaati. Lahenduste esitamine toimub kohe, ilma põhjendamatu viivitusega. Kasutajasõbralikkust toetab see, et juhul kui süsteem ei leia sobilikku lahendust, on kasutaja küsimus võimalik edastada klienditoele otse süsteemist.

Projekteeritud süsteem on lihtsasti laiendatav, uute juhtumite lisamist ja täiendamist on võimalik teostada vastavalt vajadusele – nii juhul, kui on tegemist täiesti uue probleemiga kui ka siis, kui süsteem ei saanud tuvastada kasutaja sisestatud päringut.

Väga oluliseks aspektiks on ekspertide kontrolli säilitamine süsteemi poolt pakutavate lahenduste üle. See nõue on täidetud, kasutajatele esitatakse lahendused, mis on eelnevalt ekspertide poolt valideeritud, ka uute lahenduste lisamine eeldab ekspertide poolset kinnitamist.

Süsteem kogu andmetega paikneb ettevõtte serveritel, mis minimeerib andmete lekke võimalust. Samuti on süsteemi võimalik integreerida Directo tarkvaraga.

Süsteemi esimene prototüüp leiab õige lahenduse kasutaja sisestatud probleemile umbes 93% juhtumitel, millest süsteemi esmane valik on õige umbes 84% juhtudel. Arvestades sellega, et testimiseks kasutatud andmed erinesid õppeandmetest, on saadud tulemus päris hea ning vastab ettevõtte ootustele ja ärieesmärkidele. Tänu sellele, et süsteem hakkab korjama statistikat selle kasutajatest ning nende sisestatud küsimustest, saab saadud andmeid kasutada süsteemi ja üldise klienditoe osakonna töö tulemuslikkuse parendamiseks. Täpsema hinnangu käsitletavale lahendusele saab anda ainult pärast süsteemi realiseerimist ja juurutamist.

Kokkuvõte

Klienditoe osakonnad mängivad olulist rolli klientide rahulolu loomisel. Kiire ja kvaliteetne tugi jätab kliendile positiivse mulje ning võib olla tähtsaks teguriks kauba või teenuse pakkuja edasisel valikul. Uuringud näitavad, et suurema osa tööajast on klienditoe spetsialistid hõivatud korduvatele küsimustele vastamisega, erakordsete probleemide lahendamisele jääb vähem aega. Tekkinud olukorda saab parendada läbi klientide kaasamise rutiinsete probleemide lahendamise protsessi ehk automatiseerida klienditoe tööd. Sõltuvalt konkreetse ettevõtte kontekstist, võib sobilikuks alternatiiviks osutada nii kasutusjuhendite ning korduma kippuvate küsimuste avalikustamine kui ka intelligentseid süsteemid.

Magistritöö põhieesmärgiks oli arendada klienditoe automatiseerimise lahendus äritarkvara arendamisega tegeleva Directo OÜ näitel. Asjakohase lahenduse valimisel vaadeldi erinevaid klienditoe töö automatiseerimise võimalusi ning põhisuunaks valiti küsimustele vastamise intelligentse süsteemi projekteerimine. Idee seisnes selles, et kasutaja saaks sisestada oma küsimuse või probleemi loomulikus keeles ning süsteem saaks viivitamatult pakkuda sellele sobilikku lahenduse. Teised võimalikud lahendused, nagu näiteks kasutusjuhendite avalikustamine, nõuavad kasutajalt suuremat süvenemist ning nagu näitab Directo OÜ klienditoe osakonna kogemus, tarkvara kasutajad ei taha ise tegeleda vajaliku informatsiooni otsinguga ja eelistavad kohe pöörduda klienditoe poole. Automaatne küsimustele vastamise süsteem on rohkem sarnane klienditoe konsultandi poole pöördumisega, mille tõttu see lahendus sobib käesoleva olukorra jaoks paremini.

Intelligentse süsteemi raamistikuks valiti juhtumipõhine ekspertsüsteem, mis üritab leida probleemile lahenduse põhinedes varasemal kogemusel. Kogemuse allikaks on eelkirjeldatud juhtumid, mis koosnevad probleemi ja lahenduse kirjeldusest, ning juhul kui on tegemist küsimusega, mille lahendamise kogemus on puudu, üritab süsteem leida analoogne lahendus ning kasutada seda käesoleva probleemi juures. Seega intelligentse süsteemi projekteerimine koosnes kahest etapist – esimeseks oli küsimuste ja vastuste baasi loomine, teiseks – kasutajate päringute tuvastamine ning nendele sobilike lahenduste leidmine. Lisaks hinnati saadud süsteemi.

Esimese ülesande lähtepunktiks olid Directo OÜ klientide e-kirjad, mis sisaldasid küsimusi arvete mooduli kohta. Saadud andmed olid aluseks küsimustest arusaamisel, nende analüüsi tulemusena määrati klientide tüüpiliste küsimuste klassid, mida sai kasutada kasutaja päringute

tuvastamisel. Samuti loodi juhtumite baas, mis suuresti koosneb läbitöötatud klientide küsimuste sõnastustest.

Teiseks ülesandeks oli kasutaja päringu tuvastamine ja sobiliku lahenduse leidmine. Kuna tegemist on loomulikus keeles esitatavate päringutega, siis selle ülesande teostus oli jagatud alamülesanneteks. Kõigepealt teostatakse päringu eeltöötlemine, mille käigus kontrollitakse päringu õigekirja, päringus sisalduvad sõnad viiakse algsele vormile, toimub stopp-sõnade eemaldamine. Järgmisena teostatakse päringu klassifitseerimine, tuvastatud klassi kasutatakse filtrina relevantsete juhtumite leidmisel. Eksperimentide käigus leiti, et parima tulemuse annab kolmest kihist koosnev närvivõrk, mis lisaks kasutab atribuutide tähtsuse vektoreid. Need vektorid kujutavad endast terminite kaalusid, mis näitavad termini ilmumise kalduvust ühes või teises klassis. Tulemusena saadi närvivõrk, mis oskab õigesti tuvastada sisestatud päringu klassi 97,5% juhtumitel. Kolmandaks etapiks on päringule vastava juhtumi otsing, mida teostatakse kNN algoritmi abil.

Katsetuste tulemusena saadi süsteem, mis oskab leida päringule sobiliku vastuse 93% juhtumitest, millest süsteemi esmane valik osutub õigeks umbes 84% juhtumitel ning teine valik – umbes 9%. Nagu oli nõutud, säilib klienditoe ekspertidel kontroll süsteemis salvestatud juhtumite üle ning seega tagatakse süsteemi poolt väljastatavate lahenduste kvaliteedi tase.

Lisaks põhitulemusele saab projekteeritud süsteemi abil koguda statistilisi andmeid kasutajate ja nende poolt esitatavate küsimuste kohta. See informatsioon on kasulik antud intelligentse süsteemi tulemuste ning terve ettevõtte teenuste kvaliteedi parendamisel.

Töö jaoks läbi viidud klienditoe osakonna sisene uuring tõstis ette seni märkamata jäänud probleemide olemasolu, sealhulgas ka neid, mis puudutavad klienditoe töökorraldust. Töö kirjutamise ajal vaadati üle, täiendati ning lisati uued tarkvarateemalised kasutusjuhendid, algatati tähtsamate spikrite avalikustamise praktika Directo tarkvara foorumis.

Tulevikus võiks süsteemi paremini integreerida Directo tarkvaraga, et kasutaja saaks ainult sisestada probleemse dokumendi numbri ning süsteem oskaks kontrollida, mis väljad on valesti täidetud või täitmata jäänud. Teiseks suunaks on süsteemi laiendamine. Kõigepealt puudutab see tarkvara teiste moodulite katmist. Siis kui suurem kasutajate grupp on kaetud, saaks süsteemi laiendada teistes keeltes kasutamiseks. Ettevõtte kasvuga kasutatakse tarkvara üha rohkem välismaal ning inglise keele tugi oleks suureks eeliseks.

Summary

The main purpose of this master's thesis was to develop a solution for customer support automation by the example of Directo OÜ. While selecting an appropriate solution, a variety of different customer support automation options were considered and designing a question answering system was chosen. The idea was that user can enter their question or problem in natural language and the system is able to immediately provide a suitable solution. Other possible solutions, such as publication of user manuals, assume a greater user effort. As Directo OÜ customer support experience shows, the users do not usually want to deal with the necessary information retrieval and prefer to immediately contact customer support. Automatic question answering system is more akin to a customer consultant referral, which makes it more suitable for the situation observed.

Case-based expert system was chosen as a framework for the intelligent system. It tries to find a suitable solution to the user problem based on its past experience. Predefined cases, consisting of the problem and its solution description, are the source of the system's experience and if there is no earlier experience of handling the entered user problem, the system tries to find and use a similar solution to this problem. Thus, the intelligent system design consisted of three sub-tasks. The first was building a question and answer base, which was performed on the basis of the company's e-mails. The second task was to identify users' queries and to find suitable solutions for them. This task was carried out using a neural network for queries' classification, and kNN algorithm for finding suitable solution candidates. Finally, the system assessment was done.

As a result, an intelligent system was designed that is able to find a suitable answer to the query in 93% of cases in which the primary systems' choice is correct in about 84% of the cases, and the second choice - in about 9%. As required, customer support experts retained the control over the recorded cases, which is essential for ensuring a quality level of the solutions issued.

In addition to main result a system gives a possibility to collect statistical data about users and their issues. This information is useful for the intelligent system performance and the whole enterprise service quality improvement.

Kasutatud kirjandus

Aamondt, A., Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches. – *AI Communications*, Vol.7: 1, 39-59. [WWW] <http://www.iiia.csic.es/~enric/papers/AICom.pdf> (6.12.2014)

Aha, D.W. (1998). The omnipresence of case-based reasoning in science and application. – *Knowledge-Based Systems*, Vol. 11: 5-6, 261-273. [Online] ScienceDirect (6.12.2014)

Adeyanju, I.A. (2011). Case Reuse in Textual Case-Based Reasoning: doktoritöö. Aberdeen, Robert Gordon University. [WWW] <https://openair.rgu.ac.uk/bitstream/10059/706/4/Adeyanju%20PhD%20thesis.pdf> (6.12.2014)

Aunimo, L. , Heinonen, O., Kuuskoski, R., Makkonen, J., Petit, R., Virtanen, O. (2003). Question Answering System for Incomplete and Noisy Data: Methods and Measures for Its Evaluation– *Advances in Information Retrieval: 25th European Conference on IR Research, ECIR 2003, Pisa, Italy, April 14–16, 2003. Proceedings.* – 2633, 193-206 [Online] SpringerLink (28.11.2014)

Barbera, S.H., Garcia-Pardo, J.A., Ramos-Garijo, R., Palomares, A., Julian, V., Rebollo, M., Botti, V. (2006). CBR Model for the Intelligent Management of Customer Support Centers. – *Intelligent Data Engineering and Automated Learning : Lecture Notes in Computer Science*, Vol. 4224, 663-670. [Online] SpringerLink (20.11.2014)

Bulchand-Gidumal, J., Melian-Gonzalez, S. (2009). Redesign of the IS/ICT help desk at Spanish public university. - *Higher Education*, Vol. 60, Issue 2, 205-216. [Online] SpringerLink (04.08.2014)

Cui, Z., Ducatel, G., Thint, M., Assadian, B., Azvine, B. (2006). Towards automated customer self-help. - *BT Technology Journal*, Vol. 24, 96-106. [Online] SpringerLink (06.08.2014)

Engin, G., Aksoyer, B., Avdagik, M., Bozanli, D., Hanay, U., Maden, D., Ertek, G. (2014). Rule-based expert systems for supporting university students. - *Procedia Computer Science*, Vol. 31, 22-31. [Online] Science Direct (10.09.2014)

- Farid, D. M., Zhang, L., Rahman, C. M., Hossain, M. A., Strachan, R. (2014). Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. - *Expert Systems with Applications*, Vol. 41, 1937-1946. [Online] Science Direct (03.09.2014)
- Fong, A. C. M., Hui, S.C. (2001). An intelligent online machine fault diagnosis system. - *Computing and Control Engineering Journal*, Vol. 12 (5), 217-223. [Online] IEEEXplore (07.08.2014)
- Foo, S., Hui, S. C, Leong, P. C., Liu, S. (2000). An integrated help desk support for customer services over the World Wide Web – a case study. - *Computers in Industry*, Vol. 41, 129-145. [Online] Science Direct (05.08.2014)
- Gibson-Odgers, P. (2007). *The World of Customer Service*, 2nd edition. USA: Thomson Higher Education. [Online] Google Books (01.08.2014)
- Gonzalez, L.M., Giachetti, R.E., Ramirez, G. (2005). Knowledge management-centric help desk: specification and performance evaluation. - *Decision Support Systems*, Vol. 40, 389-405. [Online] Science Direct (04.08.2014)
- Göker, M., Roth-Berghofer, T. (1999). Development and Itilization of a Case-Based Help-Desk Support System in a Corporate Environment. - *Case-Based Reasoning Research and Development: Lecture Notes in Computer Science*, Vol. 1650, 132-146. [Online] SpringerLink (04.08.2014)
- Heckman, R., Guskey, A. (1998). Sources of Customer Satisfaction and Dissatisfaction with Information Technology Help Desks. - *Journal of Market Focused Management*, 3, 59-89. [Online] SpringerLink (04.08.2014)
- Imandoust, S.B., Bolandraftar, M. (2013). Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events. - *Int. Journal of Engineering Research and Application*, Vol. 3, 605-610. [Online] Academia.edu (10.09.2014)
- Islam, M. J., Wu, Q. M. J., Ahmadi, M., Sid-Ahmed, M. A. (2007). Investigating the Performance of Naive-Bayes Classifiers and K-Nearest Neighbor Classifiers. - *International Conference on Convergence Information Technology*, 1541 – 1546. [Online] IEEEXplore (04.09.2014)

Knapp, M., Woch, J. (2002). Towards a Natural Language Driven Automated Help Desk. - *Computational Linguistics and Intelligent Text Processing: Lecture Notes in Computer Science*, Vol. 2276, 96-105. [Online] SpringerLink (06.08.2014)

Krõlov, V. (2012). Kelmuskõnede tuvastamise intelligentne süsteem: magistritöö. Tallinn, Tallinna Tehnikaülikool.

Lo, R.T-W., He, B., Ounis, I. (2005). Automatically Building a Stopword List for and Information Retrieval System. – *Journal of Digital Information Management*, Vol. 3 (1), 3-8. [WWW] <http://www.dirf.org/jdim/v3n102.pdf> (27.10.2014)

Lockhart, J. (2005). How to Market Your School: A guide to marketing, public relations, and communication for school administrators. USA: Universe. [Online] Google Books (01.08.2014)

Marom, Y., Zukerman, I. (2007). A predictive approach to help-desk response generation. - *20th International Joint Conference on Artificial Intelligence (IJCAI'07), Hyderabad, India, January 6-12, 2007. Proceedings*, 1665-1670. [WWW] <http://ijcai.org/papers07/Papers/IJCAI07-269.pdf> (14.09.2014)

Mohapatra, S. (2009). Business Process Automation. New Delhi: PHI Learning Private Limited. [Online] Google Books (01.08.2014)

Monz, C. (2003). Document Retrieval in the Context of Question Answering. – *Advances in Information Retrieval: 25th European Conference on IR Research, ECIR 2003, Pisa, Italy, April 14–16, 2003. Proceedings*. – 2633, 571-579 [Online] SpringerLink (24.10.2014)

Moreo, A., Navarro, M., Castro, J.L., Zurita, J.M. (2012). A high-performance FAQ retrieval method using minimal differentiator expressions. - *Knowledge Based Systems*, Vol. 36, 9-20. [Online] Science Direct (30.07.2014)

Rinne, Martin. Roost, Irina. Üleskirjutus. Tallinn: 21.04.2014

Sandhu, K. (2008). Theoretical Perspectives for E-Services Acceptance Model. - *Advances in Computer and Information Sciences and Engineering*, 218-223. [Online] SpringerLink (06.08.2014)

Tepandi, J. (2012). *Intelligentsed süsteemid*. Tallinn: Tallinna Tehnikaülikool.

Yalcin, N., Köse, U. (2010). What is search engine optimization: SEO? - *Procedia Social and Behavioral Sciences*, 9, 487-493. [Online] Science Direct (30.07.2014)

Yelati, S., Sangal, R. (2011). Novel approach for tagging of discourse segments in help-desk emails. – *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference, Lyon, August 22-27, 2011. Proceedings.* - 3, 369-372. [Online] IEEEXplore (01.11.2014)

Zemke, R., Woods, J. (1999). *Best Practices in Customer Service*. USA: AMACOM. [Online] Google Books (06.08.2014)

Wood, S., Howlett, R.J. (2008). *A Web-Based Customer Support Knowledge Base System*. [Online] Google Books (06.08.2014)

Воронцов, К. В. (2008). Лекции по статистическим (байесовским) алгоритмам классификации. [Online] <http://www.ccas.ru/voron/download/Bayes.pdf> (04.09.2014)