

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Tarkvarateaduse instituut

Tanel Õunas 177982IABM

**ROLLIKAEVANDAMISE MEETODITE
VÕRDLUS KOLME ORGANISATSIOONI
ANDMETE PÕHJAL**

Magistritöö

Juhendaja: Ants Torim, PhD

Tallinn 2021

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.

Autor: Tanel Õunas

04.01.2021

Annotatsioon

Käesoleva lõputöö eesmärgiks on potentsiaalsete meetodite leidmine, mida varasemalt ei ole uuritud rollikaevandamise valdkonnas ja mis sobiksid rollide kaevandamiseks. Selleks toetutakse formaalsele kontseptianalüüsile ja andmetabelite ümberjärjestamise tehnikale.

Eesmärkide saavutamiseks analüüsitakse teooriat ja nendele toetuvaid meetodeid. Potentsiaalsete meetodite hindamise ja võrdlemise jaoks otsitakse sobivad traditsioonilised meetodid rollikaevandamise valdkonnast. Peale seda rakendatakse kõikide meetodite algoritme kolme organisatsiooni andmestikel. Meetodite tulemuste põhjendatud hinnangute andmiseks toetutakse kvantitatiivsetele tulemustele ja visuaalseid esitusi hinnatakse Geštalt visuaalse taju printsiipidele ning andmetindi printsiipidele.

Töö käsitlevaks probleemiks on uute andmeanalüüsi tehnikate meetodite rakendamise uurimine rollikaevandamise valdkonnas ning kas saadud tulemused näite andmestikest on rollitehnikas arvestatavad.

Lõputöö tulemusteks on rollikaevandamise tulemused meetodite rakendamisega kolme organisatsiooni andmestikel. Seejärel võrreldakse omavahel potentsiaalsete ja traditsiooniliste rollikaevandamise meetodite tulemusi lõputöö eesmärkide saavutamiseks.

Töö uurimise ja tulemuste põhjal järeldus, et formaalsele kontseptianalüüsile ja monotoonsetele süsteemidele toetuvaid meetodeid on võimalik tulevikus kasutada rollikaevandamise valdkonnas.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 94 leheküljel, 8 peatükki, 24 joonist, 37 tabelit.

Abstract

Comparison of role mining methods on the basis of the data from three organizations

The purpose of this thesis is to find potential methods that are not previously explored in the field of role mining and are suitable for mining roles. For this purpose, we rely upon the field of Formal Concept Analysis and data table reordering technique.

In order to achieve this goal, theoretical analysis and introduction is made along with methods that are based on those data analysis techniques. Suitable traditional role mining methods from the field of role mining are researched and then selected in need of evaluation and comparison with potential methods. After that, algorithms for all the methods are implemented on the basis of the data from three organizations. For well-reasoned valuation of methods, we rely upon quantitative results. Visual representations are assessed based on Gestalt principles of visual perception and data-ink principles.

The problem addressed in the work is to research suitability of new data analysis methods in the field of role mining and assess the results obtained from test data sets.

The result of the thesis is role mining outcome on the basis of the data from three organizations. The purpose of the thesis is achieved by comparing prior potential and traditional role mining methods results.

As a result of this work, it was found out that methods which are based on Formal Concept Analysis and data table reordering, give interesting and considerable results compare to the traditional role mining methods. Future work can be expanded by broadening the data test sets for more detailed results comparison and usage of those methods in the field of role mining.

The thesis is in Estonian and contains 94 pages of text, 8 chapters, 24 figures, 37 tables.

Lühendite ja mõistete sõnastik

RBAC	<i>Role-based Access Control</i> , rollipõhine juurdepääsuõiguste juhtimine
FCA	<i>Formal Concept Analysis</i> , Formaalne kontseptianalüüs
GSH	<i>Galois Sub-hierarchy</i> , Galois alam-hierarhia
QC	QualityCover algoritm
GAIN	Kontsepti olulisuse hindamise funktsioon
BD	<i>Best decision</i> , parim otsustus
ORCA	Rollikaevandamise algoritm
GO	Graafi Optimeerimise algoritm
WSC	<i>Weighted Structural Complexity</i> , kaalutud struktuurne keerukus
UA	<i>User to role assignment</i> , kasutaja-rolli seos
PA	<i>Permission to role assignment</i> , õiguse-rolli seos
RH	<i>Role hierarchy</i> , rollihierarhia
DUPA	<i>Direct user-permission assignment</i> , otsene kasutajaõiguste seos
UAT	<i>User-attribute information</i> , kasutaja atribuudi informatsioon
FM	FastMiner algoritm
CM	CompleteMiner algoritm
RMiner	Rollikaevandamise rakendus
WEKA	Masinõppele toetuv andmekaeve tööriist
Jaccard	Koefitsient näitamaks elementide omavahelist sarnasust
CMS	Organisatsiooni andmestiku nimetus
HM	HierarchicalMiner algoritm

Sisukord

1 Sissejuhatus	11
1.1 Taust ja probleemi kirjeldus	11
1.2 Lõputöö eesmärgid	11
1.3 Metoodika.....	12
1.4 Töö struktuur	12
2 Teoreetilised alused	14
2.1 RBAC ja rollitehnika	14
2.2 Formaalne kontseptianalüüs	15
2.2.1 Formaalne kontekst ja kontsept	15
2.2.2 FCA rakendamine rollikaevandamises	16
2.3 Geštal visuaalse taju printsiibid.....	17
2.4 Andmetint	19
2.4.1 Andmetindi suhtearvu ja osakaalu maksimeerimise printsiip	19
2.4.1 Kustutamise printsiibid.....	19
3 Formaalne kontseptianalüüsi meetodid rollide kaevandamiseks	21
3.1 GSH	21
3.1.1 GSH diagrammid rollikaevandamisel	23
3.2 Kontseptiahelad	24
3.2.1 Algoritmid kontseptiahelate leidmiseks	25
3.2.2 Mitmete kontseptiahelate leidmise meta-algoritm	25
3.3 QualityCover - asjakohane katvus binaarmaatriksist	26
3.3.1 Algoritmi kirjeldus ja näide	27
4 Serieerimise meetodid	29
4.1 Monotoonsed süsteemid	29
4.1.1 Miinustehnika algoritm	30
5 Rollikaevandamise kogukonna meetodid	31
5.1 Graafi optimeerimine.....	31
5.2 Rollide kaevandamine vähendatud kontseptivõre kärpimisega.....	33
5.2.1 Rollide kaevandamine kasutades semantikat	35

5.3 Rollide identifitseerimine alamhulkade loendamise meetodil.....	36
5.3.1 CompleteMiner.....	37
5.3.2 FastMiner.....	37
5.4 Visuaalne rollikaevandamine	38
5.4.1 Rollide visualiseerimise probleem	38
5.4.2 Adviser – maatriksi sorteerimise algoritm	40
5.4.3 Extract algoritm - Rollide visuaalne esilekutsumine kasutades pseudorolle.	40
5. Lähteandmestikel genereeritud tulemused vaadeldavate meetoditega.....	42
6.1 Lähteandmestikud.....	43
6.2 GSH diagrammid.....	44
6.3 Kontseptiahelate tulemused.....	47
6.4 QualityCover tulemused.....	51
6.4.1 Tulemuste võrdlus kontseptiahelatega	56
6.5 Miinustehnika tulemused.....	58
6.6 Graafi optimeerimise tulemused.....	60
6.7 HierarchicalMiner tulemused	63
6.8 CompleteMiner ja FastMiner tulemused	68
7 Tulemuste võrdlev analüüs.....	72
7.1 FCA-le tuginevate meetodite tulemuste võrdlus rollikaeve meetoditega.....	72
7.1.1 Algoritmide tööaegade tulemused.....	73
7.2 Visuaalseid esitusi leidvate rollikaeve meetodite võrdlus	74
7.2.1 GSH diagramm ja visualiseerimise printsiibid.....	75
7.2.2 GSH ja Graafi Optimeerimise meetodite võrdlus andmestiku näitel	76
7.3 Monotoonsete süsteemide ümberjärjestamise tehnika ja Visual Role Mining.....	79
7.4 Järeldused	80
7.5 Võimalikud edasiarendused.....	81
8 Kokkuvõte	82
Kasutatud kirjandus	84
LISA 1 - Algandmestike kasutajaõiguste seoste tabelid	86
LISA 2 – CMS ja Intranet kontseptiahelate tulemused	90
LISA 3 – HierarchicalMiner minimaalse rollide arvu seadistuse tulemused	93

Jooniste loetelu

Joonis 1 Geštalt printsiibid	18
Joonis 2 Andmetindi suhtearvu näide.....	20
Joonis 3 Näiteandmestiku tulemused GSH diagrammil	23
Joonis 4 Serieerimise struktuuride näited	24
Joonis 5 Formaalse konteksti näide	27
Joonis 6 Rollide ühendamise olukorrad	32
Joonis 7 Binaarmaatriksi visuaalse esituse loomine.....	38
Joonis 8 Pseudorollide näide	41
Joonis 9 RMiner rakenduses kasutatavad vaated rollide kaevandamiseks.....	43
Joonis 10 Spordikeskuse GSH diagramm	45
Joonis 11 CMS GSH diagramm	46
Joonis 12 Intranet GSH diagramm	46
Joonis 13 Spordikeskuse kontseptiahelate järjestuse visuaalne esitus	48
Joonis 14 Spordikeskuse andmestiku katmised kontseptidega.....	52
Joonis 15 Spordikeskus katvus üle kontseptide.....	57
Joonis 16 CMS katvus üle kontseptide.....	57
Joonis 17 Intranet katvus üle kontseptide.....	57
Joonis 18 Miinustehnika tulemused	59
Joonis 19 GO rollihierarhia spordikeskuse andmestikul	61
Joonis 20 HM nelja rolli visuaalne esitus spordikeskuse näitel	66
Joonis 21 GSH diagramm.....	75
Joonis 22 GSH diagramm spordikeskuse tulemuse näitel.....	76
Joonis 23 Graafi optimeerimise hierarhia diagramm spordikeskuse tulemuse näitel	77
Joonis 24 Miinustehnika ja Adviser tulemused vastavalt serieerimise mustritele	80

Tabelite loetelu

Tabel 1 Formaalse konteksti lihtne näide	16
Tabel 2. Näiteandmestik GSH diagrammide illustreerimiseks	22
Tabel 3 Kasutajaõiguste seosed.....	39
Tabel 4 Teksti kujul kandidaatrollide esitus.....	39
Tabel 5 Algoritmide rakendusi käsitlevate seadmete tehnilised andmed.....	42
Tabel 6 Kontseptiahelate tulemused osalisel ja täieliku katvusel.....	47
Tabel 7 Kontseptiahela tulemused spordikeskuse andmestikul	49
Tabel 8 QualityCover tulemused osalisel ja täielikule katvusel.....	51
Tabel 9 QualityCover spordikeskus kontseptide tulemused.....	53
Tabel 10 QualityCover CMS kontseptide tulemused	54
Tabel 11 QualityCover Intranet kontseptide tulemused	55
Tabel 12 Miinustehnika algoritmi tööajad.....	59
Tabel 13 Graafi optimeerimise rollikaevandamise tulemused	60
Tabel 14 Graafi optimeerimise spordikeskuse tulemused.....	61
Tabel 15 Graafi optimeerimise CMS tulemused	62
Tabel 16 Graafi optimeerimise Intranet tulemused	63
Tabel 17 HierarchicalMiner rollikaevandamise tulemused.....	64
Tabel 18 HierarchicalMiner algoritmi tööajad	65
Tabel 19 HM tulemused minimaalse keerukuse seadistusega Spordikeskus andmestikul	66
Tabel 20 HM tulemused minimaalse keerukuse seadistusega CMS andmestikul	67
Tabel 21 HM tulemused minimaalse keerukuse seadistusega Intranet andmestikul	67
Tabel 22 CM ja FM algoritmide tulemused lähteandmestikel	68
Tabel 23 CM ja FM tulemused spordikeskuse andmestikul.....	69
Tabel 24 CM tulemused CMS andmestikul	70
Tabel 25 FM tulemused CMS andmestikul.....	70
Tabel 26 CM ja FM tulemused Intranet andmestikul.....	71
Tabel 27 QualityCover ja rollikaeve algoritmide rollide võrdlus.....	73
Tabel 28 Algoritmide tööajad lähteandmestike rollikaevandamisel	73

Tabel 29 Algoritmid wsc tulemused	74
Tabel 30 Spordikeskus kasutajaõiguste seoste tabel	86
Tabel 31 CMS kasutajaõiguste seoste tabel	86
Tabel 32 Intranet kasutajaõiguste seoste tabel	89
Tabel 33 Kontseptiahelate tulemused CMS andmestikul	90
Tabel 34 Kontseptiahelate tulemused Intranet andmestikul	91
Tabel 35 HM tulemused minimaalse rollide arvu seadistusega Spordikeskuse andmestikul	93
Tabel 36 HM tulemused minimaalse rollide arvu seadistusega CMS andmestikul	93
Tabel 37 HM tulemused minimaalse rollide arvu seadistusega Intranet andmestikul ...	94

1 Sissejuhatus

1.1 Taust ja probleemi kirjeldus

Tänapäeva laienevates IT-lahendustes on tekkinud uueks probleemiks süsteemide juurdepääsuõiguste juhtimise keerukus ja nende haldamine ühest kesksest halduspunktist. Organisatsioonides, kus kasutajate ja õiguste arv on väga suur (tuhandetes) on muutunud süsteemi õiguste juhtimine suurte andmete töötluks ja on tekkinud vajadus nende korrektseks määratlemiseks. Puuduseks ei ole sellistel juhtudel mitte ainult õiguste juhtimine, vaid ka organisatsiooni halduses olevate üksuste ja süsteemi turvalisuse üle kontrolli saavutamine. Üheks selliseks õiguste haldamise lähenemiseks on rollipõhine juurdepääsuõiguste juhtimine (RBAC), millega piiratakse kasutajate toimingud organisatsiooni haldusalas rollide kaudu.

Selliste rollihulkade leidmist nimetatakse rollitehnikaks (ingl k *role engineering*), mis on üks peamine tegevus RBAC juurutamisel ettevõtetes [2]. Rollitehnika puhul saab toetuda mitmele viisile, millest üks on äriprotsesside abil tööülesannete leidmine ja nendele sobivate rollide määratlemine. Teisel juhul toetutakse rollide leidmisel olemasoleva süsteemi õigustele. Esimesel juhul on tegemist manuaalse meetodiga rollide leidmisel, mis küll korrektse protsessi läbi määratleb rolle täpsemini, aga ei ole suurtes süsteemides tohtu ajakulu tõttu mõistlikult ja tasuvalt teostatav. Sellest tulenevalt on keerukamate süsteemide tekkimisest välja kujunenud automatiseeritud rollide leidmine andmete kaevandamise ehk rollikaevandamisega.

1.2 Lõputöö eesmärgid

Käesoleva lõputöö peamine eesmärk on leida potentsiaalsed meetodid, mis sobiksid rollide kaevandamiseks ja mida varasemalt ei ole uuritud rollikaevandamise valdkonnas. Selle jaoks vaadeldakse lähemalt andmeanalüüsi tehnikaid nagu formaalne kontseptianalüüs (FCA) ja monotoonsed süsteemid, mille meetodite hulgast saaks leida potentsiaalsed meetodid.

Soovitud tulemusteni jõudmiseks on paika pandud täpsemad eesmärgid:

1. Uuritakse FCA teoreetilisi aluseid rakendamaks rollikaevandamise valdkonnas ja otsitakse meetodeid, mille lähenemised sobiksid rollide kaevandamiseks.
2. Analüüsitakse monotoonsete süsteemide andmeanalüüsi tehnikat ja proovitakse leida sobilikud meetodid rollide kaevandamiseks.
3. Potentsiaalsete meetodite sobivuse hindamiseks kvantitatiivsete ja visuaalsete tulemuste alusel, leitakse rollikaevandamise valdkonnast mitmed traditsioonilised rollikaevandamise meetodid.
4. Viimaks rakendada meetodite algoritme kolme organisatsiooni andmestikel, et leida rollikaevandamise tulemused.

Lõputöö oodatav tulemus on kolme organisatsiooni andmestikest saadud rollikaevandamise tulemuste võrdlus erinevate potentsiaalsete ja traditsiooniliste rollikaevandamise meetodite vahel.

1.3 Metoodika

Sobivate potentsiaalsete meetodite leidmiseks uuritakse kahte andmeanalüüsi tehnikat, mida varasemalt on andmeanalüüsi valdkondades rakendatud ja mis sobiksid kaevandama juurdepääsuõiguse andmeid. Toetutakse erinevatele kirjanduslikele allikatele, et analüüsida teoreetilisi aluseid ja leida sobivad meetodid.

Traditsiooniliste rollikaevandamise meetodite valikul toetutakse kirjanduslikele allikatele, milles on võrreldud erinevaid meetodeid ja nende rollikaevandamise sooritusi. Lisaks on sobivate meetodite valikul arvestatud väljundite ja rakendusliku kättesaadavusega. Eelneva alusel saab garanteerida potentsiaalsete ja traditsiooniliste meetodite tulemuste võrreldavuse kolme organisatsiooni andmestikel.

Meetodite võrdlus toetub kvantitatiivsetele tulemustele nagu kontseptide ehk rollide arvud ja tööaegade tulemused. Visuaalsete tulemuste põhjendatud hinnangute andmiseks toetutakse laialt tunnustatud Geštalt visuaalse taju printsiipidele ja E.Tufte andmetindi printsiipidele.

1.4 Töö struktuur

Lõputöö on jagatud eesmärgi saavutamiseks nelja etappi. Esmalt vaatame teoreetilisi aluseid, kus tutvustame rollikaevandamise eesmärki, FCA-d ja laialt tunnustatud

visuaalsete esituste printsiipe. Seejärel tutvustame kolme peatüki all välja valitud võimalikke potentsiaalseid ning samuti rollikaevandamise kogukonnas tuntud meetodeid. Järgnevas etapis leiame meetodite algoritmidega rollikaevandamise tulemused kolme organisatsiooni juurdepääsuõiguste andmetest. Viimases etapis teostame tulemuste võrdleva analüüsi, järelduse ja võtame kokku lõputöö eesmärgid.

2 Teoreetilised alused

Käesolevas peatükis vaatame lähemalt, mille jaoks on rollikaevandamine vajalik ja kuidas seda rakendatakse. Lisaks anname tutvustava ülevaate FCA-le, millele toetavad mitmed lõputöös vaadeldavad meetodid ning tutvustame printsiipe, mille alusel saab anda visuaalsetele esitustele põhjendatud hinnanguid.

2.1 RBAC ja rollitehnika

Rollitehnika all mõeldakse rollide leidmist ja defineerimist selliselt, et need oleksid vajalikud ja korrektselt määratletud organisatsiooni seisukohast. Rollitehnika on eelduseks rollipõhise juurdepääsuõiguste juhtimise (RBAC) kasutusele võtmisel. Hästi teostatud RBAC realisatsioon sisaldab organisatsiooni ärireegleid, mis on seotud juurdepääsuõiguste juhtimisega ja peegeldab need reeglid rollide defineerimisel ja struktureerimisel [1]. Õigesti identifitseeritud rollid on eelduseks suure ja keeruka organisatsiooni heale infosüsteemi juhtimisele [1].

Rollitehnikal on mitmeid lähenemisi, mis on jagatud suuresti kaheks: ülalt-alla (ingl k *top-down*) ja alt-üles (ingl k *bottom-up*). Ülalt-alla on äriprotsesside põhjalik analüüs, kus kaardistatakse juurdepääsuõigused, mis on vajalikud spetsiifiliste ülesannete täitmisel. Teisisõnu on tegemist „puhtalt lehelt“ rollide identifitseerimise ja õiguste määramisega organisatsiooni võtmeisikute ja valdkonna ekspertide käe läbi [2]. Antud meetod nõuab manuaalse protsessina RBAC juurutamisel suurt aja- ja inimressurssi kulu. Alt-üles lähenemisel kaardistatakse tegelikud rollid, mis on juurutatud juba olemasolevasse juurdepääsuõiguste juhtimise infosse [3], mistõttu kutsutakse seda ühtlasi rollikaevandamiseks. Üheks keerukuseks on konkreetse süsteemi tulemuste normaliseerimine üldlevinud raamistikule, mistõttu kasutakse just hübriid lahendust, kus on kasutatud mõlemat lähenemist, sest organisatsioonid ei ole nõus juurutama rolle, millel ei ole ärilist tähendust. Käesolevas lõputöös uurime ja võrdleme erinevaid rollikaevandamise meetodeid, mis toetuvad alt-üles lähenemisele. Kui varasemalt toetuti rollide leidmisel andmekaeve meetoditele, siis käesolevas lõputöös uurime traditsioonilisi meetodeid, mis on edasiarendus andmekaeve meetodite kitsaskohti lahendatavatest rollikaeve meetoditest. Erinevate meetodite algoritmid võtavad arvesse rollitehnikas kasutatavat binaarmaatriksi sisendeid ja pakuvad hästi genereeritud rollidele lisaks rollihierarhiaid ning kandidaatreole.

2.2 Formaalne kontseptianalüüs

Formaalne kontseptianalüüs (FCA) on andmeanalüüsi meetod, mille eesmärgiks on tuletada läbi atribuutide hulga objektide vahelisi seoseid ja struktureerida neid sarnaselt klasterdamise tehnikale, kus ühiste omadustega objektid viiakse ühtekokku. FCA oli välja töötatud 1980.a alguses, millele järgnes ajastule kohane vaevaline rakenduslik kasutusele võtmine kuniks aastatuhande alguseni, kui suurte andmete analüüsi vajaduse osakaal andmeinformatsiooni plahvatusliku kasvuga suurenes. Nüüdseks kasutatakse FCA-d paljudes valdkondades nagu tarkvaraarendus, masinõppe, psühholoogia [4] ja teistes, kus tegeletakse suurte andmetega ja kehtib vajadus nende analüüsile.

Analüüsimeetodi nimetus tuleneb andmete struktureerimisest sellisteks üksusteks, mille formaalsete kontseptide üldistused on vastuvõetavad tõlgendused inim mõistusele ehk loomulik arusaam [5]. Täendus piirneb ümber kahe üksuse tüübi ja nende vahelise rakendusliku tarviduse ehk seose olemasolul [4]. Kontseptid kirjeldavad meile objekte või objektide hulka ja nendele kuuluvaid omadusi. Abstraktselt saame välja tuua näiteid nagu: voolu kasutavad seadmed või pallimängud, mis omavad mingeid ühiseid omadusi.

2.2.1 Formaalne kontekst ja kontsept

FCA eristab objektide- ja atribuutide hulka, mis võivad vastavalt reaalses maailmas olla kaubad ja materjalid, inimesed ja nende omadused või lõputöö mõistes kasutajad ja õigused. Viimase näitena toodud andmehulkade elementide vahel eksisteerivad seosed tõeväärtusega, kus ühe andmehulga objekt on teise andmehulga atribuudiga seotud või mitte. Objektid ja atribuudid koos seostega loovad mingisuguse vaadeldava ruumi ehk formaalse konteksti. Definitsiooni kohaselt **formaalne kontekst** formaalses kontseptianalüüsis koosneb kolmikust $K = (G, M, I)$, kus G on hulk formaalseid objekte, M hulk formaalseid atribuute ja $I \subseteq G \times M$ on binaarne seos objekti ja atribuudi vahel [6]. Kõige lihtsam näide formaalse konteksti esitamiseks on binaarne maatrikstabel ehk kahedimensiooniline tabel, kus objektid (veerus) ja atribuudid (reas) on seostatud mingi kindla sümboliga nagu näha allolevas tabelis.

Formaalsest kontekstist saame edasiselt vaadata formaalse kontsepti, mille jaoks defineerime hulgad $A \subseteq G$ ja $B \subseteq M$:

$$A' := \{m \in M \mid \forall g \in A : gIm\}$$

$$B' := \{g \in G \mid \forall m \in B : gIm\}$$

Leitakse vastavalt kõik atribuudid, mis on ühised kõikidele objektidele hulgas A. Teisiti B' tähistab hulka kõikidest objektidest, millel on ühised atribuudid. Seega A' näitab ära hulga atribuute, mis on ühised kõigile objektidele selles objektide hulgas A ja B' näitab ära hulga objekte, mis on ühised kõigile atribuutidele selles atribuutide hulgas B [6].

Selle abil saame näidata, et formaalse konteksti (G, M, I) **formaalne kontsept** on iga paar (A, B), kus $A \subseteq G$, $B \subseteq M$ ja $A' = B$, $B' = A$. Formaalse kontsepti formaalsete objektide hulk A on ekstensioon ja formaalsete atribuutide hulk B on intentsioon. Esimene sisaldab endas kõiki objekte, mis on kaasatud kontsepti piiridesse ja viimane kõiki objekti atribuute (omadusi), mis on tõesed nende objektide kohta.

Allolevas maatrikstabeli näites on tegemist väljamõeldud lihtsa süsteemi kasutajate ja õiguste andmetega, mis määravad kasutajate tööülesannete õigusi ja ligipääse. Võttes aluseks ühe paari (A,B), kus A on objektide hulk ehk ekstensioon {Arved} ja B on atribuutide hulk ehk intentsioon {Juhataja, Raamatupidaja}, saame nende binaarsete seoste ($I \subseteq G \times M$) abil kätte ühe kontsepti. Laiendades ekstensiooni hulka kasutajaga {Töötaja}, saame uue alam-kontsepti, kuna selle ekstensioon on sisaldunud eelmises kontseptis.

Tabel 1 Formaalse konteksti lihtne näide

	Juhataja	Raamatupidaja	Töötaja	Kassapidaja
Turundus		X		X
Ladu		X	X	
Arved	X	X	X	
Seadistus	X		X	

Näitest saab järeldada, et isegi mõistliku suurusega formaalsest kontekstist on võimalik leida väga palju formaalseid kontsepte, mis on üheks puuduseks FCA andmeanalüüsis.

2.2.2 FCA rakendamine rollikaevandamises

Formaalsete kontseptide hulka kontekstist tähistakse $B(G,M,I)$. Kontsept (A_1, B_1) on kontsepti (A_2, B_2) alam-kontsept tähistatuna $(A_1, B_1) \leq (A_2, B_2)$, siis ja ainult siis, kui $A_1 \subseteq A_2$ ja samaväärselt $B_1 \supseteq B_2$. Selline kontseptide omavaheline järjestatud seos defineerib kontseptivõre ehk Galois võre ning iga kontsept võres pärib õigused alam-kontseptidelt ja teistpidi kasutajad ülem-kontseptidelt. See tähendab, et võres eksisteerib kasutajaid ja

õigusi üle mitmete kontseptide, mis on otseselt üleliigsed. Nende eemaldamisel on võimalik vaadelda kontseptivõret kui hierarhiat ja käsitledes kontsepte kui rolle saame kogu FCA teoreetilistele alustele toetudes rollihierarhia tulemuse. Rollihierarhia on rollihulkade osaline järjestus.

Vaadeldes formaalsest kontekstist hoopis objektide ja atribuutide kontsepte, vastab objektile $g \in G$ objekti kontsepti $\gamma g = (\{g\}'', \{g\}')$, mille intentsioon on selle objekti kõigi atribuutide hulk ja ekstensioon objektide hulk, mis on seotud nende atribuutidega [6]. Samamoodi vastab atribuudile $m \in M$ atribuudi kontsept $\mu m = (\{m\}'', \{m\}')$. Juhul, kui eesmärgiks on leida kõikide kontseptide asemel teatud alamhulka kontseptivõres, saame kõikide kontseptide hulga $B(G,M,I)$ alamhulga defineerida järgnevalt:

$$\beta(G,M,I) = \{\gamma g \mid g \in G\} \cup \{\mu m \mid m \in M\},$$

mille hulk eelnevalt välja toodud järjestuse seosega nimetatakse Galois alam-hierarhiaks ja mida saab rakendada kontseptide ehk rollide visuaalse esituse ühe lahendusena rollikaevandamisel. Seda pakub GSH diagrammide joonis, mille lahendust ja tulemusi vaatleme lõputöö järgmistes osades.

On näha, et FCA-l on väga mitmeid rakendusi rollikaevandamise teostamisel, olgu selleks rollide või hierarhiate leidmine. FCA-le toetuvate meetodite arvestatavaid tulemusi saame vaadata hilisemalt meetodite võrdlemisel. Sealt tulenevalt võtame hilisemates peatükkides eesmärgiks FCA-le toetuvate meetodite uurimise ja tulemuste leidmise.

2.3 Gešalt visuaalse taju printsiibid

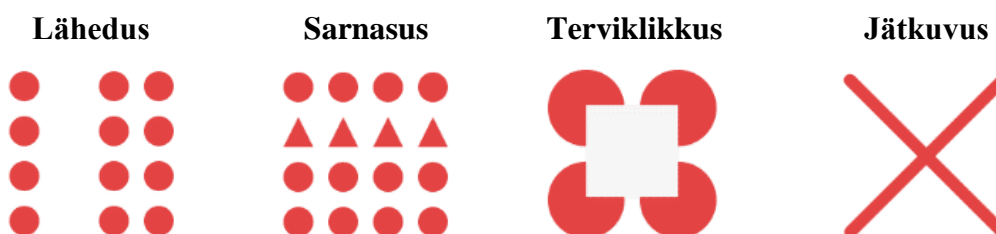
Kvantitatiivsete andmete visuaalseks esituseks on mitmeid erinevaid võimalusi. Käesolevas lõputöös kasutatakse informatsiooni esitamiseks mitmeid erinevaid viise nagu näiteks tabelid ning joonised, aga peamiseks osaks on vaadeldavad rollikaevandamise meetodid, mis annavad tekstina esitatud rollide asemel veel visuaalseid esitusi hierarhia graafikute ja ümberjärjestatud tabelitena. Nende omaduste ja headuste võrdlemiseks peame toetuma aktsepteeritud ja tunnustatud printsiipidele [7]. Käesoleva psühholoogia haru põhjal on selgeks tehtud, kuidas inimesed tajuvad maailma enda ümber ja sealt tulenevalt on tekkinud mitmed visuaalse taju printsiibid. Järgnevalt esitame printsiibid ja selgitame lähemalt nende põhimõtteid.

Esimese printsiibina vaatame mõistet lähedus (ingl k *proximity*), mille puhul tunneme, et objektid, mis on lähestikku asetsevad kuuluvad ühtselt gruppi [7]. Selle alusel saab muuta

visuaalset taju täielikult tabeli kujul asetsevate punktide omavahelise kauguse muutmisel, kus ühel kujul me näeme veergude ja teisel ridade gruppi. Kõige lähem näide sellele on klasterdamine, mis kehtib erinevate kaevandamise meetodite puhul andmetabelite ümberjärjestamisel. Printsipiide mõisteid ja selgitusi toetab allolev joonis.

Sarnasuse (ingl k *similarity*) korral on objektid tähistatud sama sümboli, kuju, värvi või orientatsiooniga luues taju, et need tekitavad ühise grupi, isegi korrapäratu struktuuri kujul [7]. Selle abil saab vaatlejat suunata märkama või tajuma objekte sobival viisil. Vaatleja antud kontekstis on osapool, kes üritab visuaalset objekti mõista. Näiteks ühe tähendusega või ühte gruppi kuuluvad elemendid tähistatakse ühesuguse värviga, kirjatüübi või kujuga.

Olukorras, kus objekt terviklikuna ei ole täielikult siduv ja koosneb mitmetest eraldiseisvatest elementidest, suudame tajuda seda objekti ühtselt loogilise ja käsitletavana. Seda nimetatakse terviklikkuseks (ingl k *closure*). Elementide vahel täidetakse lüngad ja tekitatakse selline konstruktsioon, mis tundub mõistetav ja vastuvõetav. Sama kehtib graafikute kohta, mis ei vaja konkreetset eraldatust ning tähistamist tekstist või objektidest dokumendis, sest me suudame ühtse eristatava üksuse luua. Täpselt nagu terve allolev joonis, millele ei ole eraldi joontega tähistatud kasti külge ehitada.



Joonis 1 Geštalt printsipiidid

Jätkuvus (ingl k *continuity*) on võime objektidest tuletada mingit teekonda või naturaalselt tulemit, isegi kui seda ei pruugi eksisteerida. Inimese taju täidab puuduva osa, tekitades vaatlusest kõige loogilisema järelduse. Võtame näiteks objektid, mis on üksteise taha peidetud, aga nähtavad on suurem osa struktuurist. Me oletame, et need osaliselt peidetud struktuurid on täidetud ja mitte-katkendlikud, kuigi realselt ei pruugi see selliselt olla.

Lõputöös käsitleme rollikaevandamise visuaalsete esituste võrdlusi väljatoodud printsipiidele. Printsipiide eesmärk on näidata, et missuguste omaduste alusel

visualiseerimise esitused on head ning kuidas neid saab hinnata. Sealt tulenevalt on võimalik printsiipide alusel tulemusi argumenteerida ja põhjendatult hinnata.

2.4 Andmetint

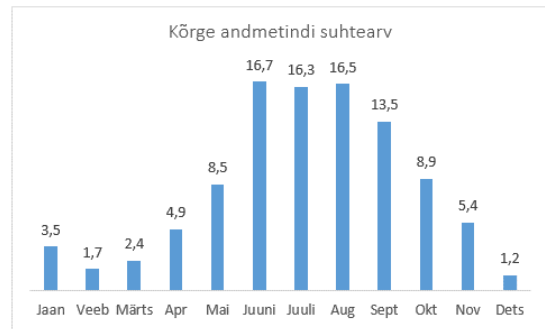
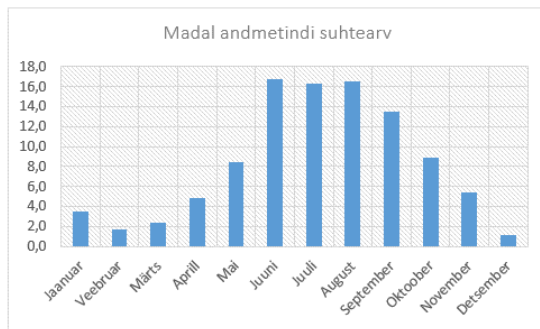
Vähesed, kui mitte öelda üksikud graafikud edastavad visuaalselt lugejale selliselt andmeid, kus sisu kustutades ei kaotata vähematki vajalikku informatsiooni [8]. Selline hulk graafilist sisu on formuleeritud kui andmetint (ingl k *data-ink*). Hea praktika puhul on alati suurem osa tindist graafikutel täidetud andmetega ja muutused graafikutel toimuvad ainult andmetindi eemaldamisel. Järgnevalt vaatame andmetindiga seotud printsiipe, mille eesmärgiks on suunata lugeja tähelepanu tähtsate andmete tajumisele.

2.4.1 Andmetindi suhtearvu ja osakaalu maksimeerimise printsiip

Andmetindi suhtearv (ingl k *data-ink ratio*) on võrdeline andmetindi ja kogu graafilise info kujutamiseks kasutatud tindi suhtega. Sealt tulenevalt, mida kõrgem on andmetindi osa, seda suurem on suhtearvu väärtus. Üheks suhtearvu tõstmise meetodiks on andmetega seotud tindi osakaalu suurendamine (maksimeerimine) mõistlike otsuste tegemisel. Mõistlikkusest lähtuvalt tähendab, et alati peab leiduma põhjus tindi lisamise vajalikkusest ning ainuüksi ebaolulise andmetindi lisamise vajadusest selleks ei piisa [8]. Lisandunud andmetint peab alati edastama graafiliselt uut vajalikku informatsiooni.

2.4.1 Kustutamise printsiibid

Üks suhtearvu suurendamise võimalusi on eemaldada mitte-andmetindi (ingl k *non-data-ink*) mahtu. Mõningatel juhtudel kasutatakse suurel hulgal tinti visuaalsete kujutiste loomisel. Teatud osas on sellel mõjuv põhjus, mille eesmärk on tekitada lõuend andmeinformatsiooni kujutamisele, aga enamuse juhtudel põhjus ei ole vajalik. Järgnevalt on esitatud joonis illustreerides madala ja kõrge andmetindi suhtearvu graafikuid, millele on rakendatud printsiipe. Eemaldatud on graafiku ruudustik (nimetatud lõuend), lühendades kuu nimetusi ja täpsemalt näidates andmepulkadel väärtusi.



Joonis 2 Andmetindi suhtarvu näide

Teiseks printsiibiks on üleliigse andmetindi eemaldamine, mõistlikkuse piires. Enamus andmeesitused on ühesed ja vähese graafilise liiasusega, aga juhul, kui eristatavalt on märgata andmetindi infos liiasust, on võimalik rakendada printsiipi. Teatud tingimuste korral on liiasuse ehk korduva andmetindi olemasolu aktsepteeritav, kui see aitab kaasa võrdlus, täielikkuses või keerukuse hindamises. Rollikaevandamise tulemustes võrdleme visuaalseid esitusi eelnevalt vaadeldud printsiipidele.

3 Formaalse kontseptianalüüsi meetodid rollide kaevandamiseks

Alljärgnevas peatükikes uurime lähemalt mitmeid FCA meetodeid, mis sobiksid potentsiaalselt rollikaevandamise meetoditeks.

3.1 GSH

Antud lõputöös kasutatakse ühe FCA meetodina Galois alam-hierarhia (GSH) diagramme. Meetod on loodud esialgse eesmärgiga tarkvaraarenduse tööriistana klassi hierarhia ümberehituseks [9], aga võimaldab lahendada veel meie jaoks FCA-st tuntud probleemi, kus kontseptivõrest tekitatud kontseptide arv on kordades või eksponentsiaalselt suurem seoste arvust. GSH diagrammid võimaldavad piirata kontseptide arvu kontseptivõres ja visualiseerida seoseid elementide (objektid ja atribuudid) ehk rollikaevandamise mõistes kasutajate ja õiguste vahel konkreetselt ja lühidalt. See on saavutatud lahendusega, kus diagrammidel on kuvatud mitmed elemendid ühe sõlme all, tähistades need nimetatud siltidega ja kujutades ühe seosega mitmeid suhteid objektide ja atribuutide vahel. Lisaks on ühe seose kaudu võimalik liikuda mitmeid kordi, et tuvastada õiguste ja kasutajate vahelist seost ning mitte ainult otsese sõlmega, vaid pikenedes ka järgmistele sõlmedele. GSH diagrammidel on kolm unikaalset elementide kogumiku, mis määravad diagrammi omadused:

1. sõlmed (kontseptid)
2. Seosed nende kontseptide vahel
3. Sildid atribuutide ja objektide hulgast, mis tähistavad sõlmede nimetusi.

GSH põhineb Hasse diagrammidel, mida kasutatakse osaliselt järjestatud hulkade kujutamiseks. Iga osaliselt järjestatud hulga elemendi jaoks joonistatakse välja sõlm ehk praeguses mõistes kontsept $C_i \in \beta(G, M, I)$ ja seosed joonistatakse välja nende sõlmede vahel järgnevatel tingimustel [6]:

1. Kui $C_1 < C_2$ hulgas, siis sõlm, mis tähistab C_1 asub joonisel madalamal kui vastav sõlm C_2 .
2. Kujutatakse tee alla sõlmest C_1 kuni C_2 , kui $C_1 > C_2$ ja C_1 katab C_2 ehk ei ole kontsepti $C_3 \in \beta(G, M, I)$, kus $C_1 > C_3 > C_2$.

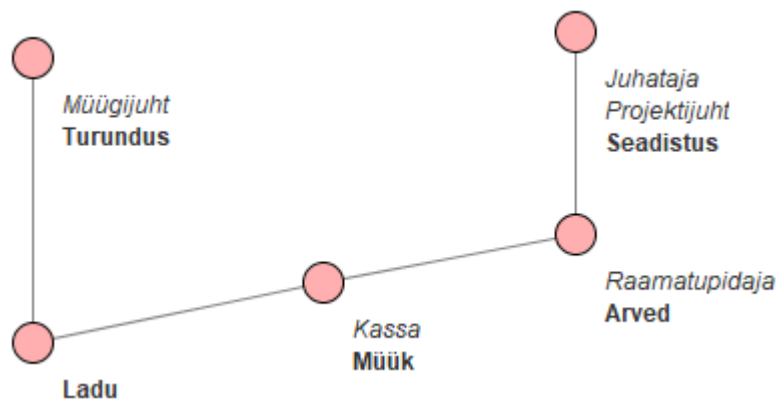
Kontseptivõre diagramm ehk Hasse diagramm esitab väga head järjestatud ja struktureeritud kontseptide ja nende seoste visuaalset esitust. Käesolevas lõputöös on kasutatud Hasse diagrammide loomiseks Galois alam-hierarhiates, Maarja Raua bakalaureuse lõputööna loodud programmi. Tegemist on tasuta vabatarkvaraga, mis täidab lähteandmestike analüüsiks ja võrdlemiseks vajalikud omadused. Leidub veel teisi kontseptivõrede andmete töötluste tarkvarasid nagu näiteks: ConceptExplorer ja Galicia, kuid nendes esineb nõrkusi, milleks on GSH meetodi puudus või ebapiisav ning nõrk siltide lisamine [9].

Rollide visuaalseks esitamiseks on GSH diagrammid hästi sobivad, sest kõige naturaalsemal kujul kasutab meetod andmeid, mis kirjeldavad kahte tüüpi elemente ja seoseid nende vahel. Kogu käesoleva töö raames on vaatlemises andmestike kasutajad, õigused ja nendevahelist seost tähistavad kasutajaõigused ehk juurdepääsuõigused. Järgnev lihtne tabeli näide demonstreerib formaalset konteksti GSH diagrammi abil. Määratletud on õigused $G = \{Seadistus, Turundus, Ladu, Arved, Müük\}$ ja kasutajad $M = \{Juhataja, Raamatupidaja, Kassa, Müügijuht, Projektijuht\}$

Tabel 2. Näiteandmestik GSH diagrammide illustreerimiseks

	Juhataja	Raamatupidaja	Kassa	Müügijuht	Projektijuht
Seadistus	X				X
Turundus				X	
Ladu	X	X	X		X
Arved	X	X			X
Müük	X	X	X	X	X

Alloleva joonise diagrammil on sõlmedes tähistatud rasvases kirjas atribuudid (õigused), kaldkirjas objektid (kasutajad) ja joontega seosed nende vahel. Iga sõlm tähistab kontsepti. Järgnevalt vaatame, kuidas lugeda antud diagrammilt välja seoseid objektide (m) ja atribuutide (g) vahel. Eksisteerib seos süsteemi kasutajate $m \in M$ ja õiguse $g \in G$ vahel, siis ja ainult siis, kui on tee alla sildist m, sildini g või sildid asuvad ühes sõlmes. Näitena - kasutajale Kassa on antud õigused kasutada õigusi Müük ja Ladu. Kui eksisteerib tee alla kasutajalt m1 kasutajani m2, siis kasutaja m2 kasutab sama süsteemi õiguste alamhulka nagu kasutaja m1. Antud näite puhul, kasutaja Kassa omab kasutajate *Juhataja*, *Projektijuht* ja *Raamatupidaja* õiguste alamhulka. Ehk ei lisata kasutaja Kassa õiguste hulka diagrammi ülemistesse sõlmedesse paigutatud õigused.



Joonis 3 Näiteandmestiku tulemused GSH diagrammil

GSH diagramm annab andmetabelite kompaktsemat esitust, mida oli eelnevast lihtsast näitest näha, kus diagrammil on oluliselt vähem sõlmesid (5tk) ja seoseid (4tk) võrreldes andmetabeli vastavate 10 üksuse (kasutajad ja õigused) ja 15 seosega (kasutajaõigused). Selline GSH diagrammi kompaktsem esitus aitab visuaalselt paremini avastada vastavusi ja sarnasusi objektide ning atribuutide vahel. Väga suurte andmetabelite korral on sarnasusi selliselt oluliselt lihtsam seoste ja hea sildistamise korral ära tunda. Väga konkreetsetel juhtudel saab olla GSH diagrammides rohkem seoseid sõlmede vahel, kui algandmestikus, tekitades meetodi esituste kompaktsusele tõestamise vajaduse. Õnneks on uurimused kinnitanud, et GSH diagrammid on keerukamate andmestike puhul alati oluliselt lihtsamad ja kokkuvõtlikumad [9], mida proovime kinnitada samuti lõputöö lähteandmestike ehk kolme organisatsiooni juurdepääsuõiguste rollikaevandamise tulemustega.

3.1.1 GSH diagrammid rollikaevandamisel

GSH diagrammid on rakendatavad olukordades, kus eksisteerib seos kahte tüüpi elementide vahel (binaarmatriksid) nagu on praeguses lõputöös käsitletav andmestikud. Teadaolevalt ei ole praeguse hetkeni leitav infot, et GSH diagramme oleks kunagiselt uuritud või käsitletud rollikaeve seisukohast. GSH diagrammidel on võimalik käsitleda kõiki sõlmesid vaadeldavas süsteemis rollide või rollikandidaatidena nagu peatükis 2.2.2 olime välja toonud, kus tutvustasime formaalse konteksti kõikide kontseptide hulga $B(G,M,I)$ alamhulka, mis sisaldas ainult objektide või atribuutide kontsepte.

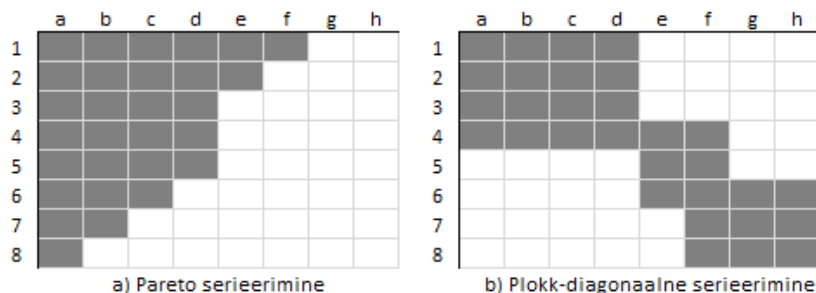
Üldjoontes on diagrammi ülemisele poolele koondatud suuremate õiguste hulgaga kasutajad ja lisaks ka teatud juhtudel õigused, millel on unikaalne seos ühe kasutajaga.

GSH omadus on visuaalselt grupeerida sõlme ehk rolli alla ühe või mitu silti. Kasutajate aspektist saame leida nende kasutajatega seotud õigused ehk õiguste kogumiku, nimetades seda rolliks. Eelneva alusel on võimalik teoreetiliselt käsitleda GSH diagramme visuaalse rollikaevandamise ja rollihierarhia loomise eesmärgil.

3.2 Kontseptiahelad

Kontseptiahelad pakuvad uut meetodit lahendamaks FCA liiga suure arvu kontseptide probleemile kontseptivõres, proovides leida kontseptide asemel huvitavaid kontseptide ahelaid. Alamhulkade leidmiseks kõige huvitavamatest kontseptidest leidub mitmeid meetodeid, kuid kontseptiahelate eelis on säilitada kontseptivõre struktuuraalsus, mistõttu on see peamine valik teiste meetodite hulgast [10]. Lihtsamalt öeldes on kontseptid järjestatud lineaarselt, kus igal kontseptil on eelneva ja järgneva kontseptiga seos täpselt samamoodi nagu kontseptivõres liikudes.

Kontseptiahelate meetod on seotud serieerimisega, mis on visuaalne uurimuslik andmeanalüüsi tehnika, kus üksused järjestatakse selliselt, et avaldub parim korrapärane ja mustriline esitus [11]. Kontseptiahelate tulemusel asetsevad kontseptid andmestikus selliselt, et sarnaste üksuste asetus on kõrvuti või lähestikku. Sealt tulenevalt saame vaadelda järgnevalt kahte struktuuri järgmisel joonisel, mis on seotud kontseptide leidmise ja kontseptiahelatega. [10].



Joonis 4 Serieerimise struktuuride näited

Vasakpoolse tabeli puhul on esitatud Pareto serieerimine, kus andmetabeli järjestus on selliselt, et ühed moodustavad kolmnurkse kujundi tabeli ühte nurka ja on omane kontseptiahelale. Plokk-diagonaalne serieerimine on esitatud parempoolses andmetabelis, kus ühed on diagonaalselt järjestatud kontseptidena. Viimane on vastupidine kontseptiahelale ja nimetatakse anti-ahelaks, sest kontseptidel puudub järjestuses suhe üksteise vahel.

3.2.1 Algoritmid kontseptiahelate leidmiseks

Kontseptiahelate katvuse leidmiseks on kasutatavad mitmed heuristilised algoritmid: monotoonse süsteemi ahela algoritm (MS-chain), sagedus-leksikograafiline algoritm (FL-sort) ja klasterdamise algoritm (KM-chain) [10]. Nimetatud algoritmide hulgast kasutame FL-sort algoritmi täiendatud versiooni (CL-sort), mis põhineb sageduse asemel konformismiskaalal ning mille korral määratakse rea kaaluks ühtedega veergude veerusageduste summa. Konformismiskaala on miinustehnika peatükis 3.1.1 detailsemalt defineeritud. CL-sort algoritm töötab võrreldes MS-chain algoritmiga kiiremini ja pakub kontseptiahelate paremaid tulemusi. KM-chain algoritm võib pakkuda teoreetilisel tasemel ühtemoodi huvipakkuvaid kontseptiahelaid, aga kahjuks puudub praegusel hetkel tulemusi genereeriv rakendus.

CL-sort algoritmi omadus on järjestada andmestiku kasutajad ja õigused kahes eraldiseisvas etapis, kus algselt sorteeritakse read ja veerud vastavalt konformismiskaalale ja seejärel iteratiivselt paigutades leksikograafilisse ehk tähestikulisse järjestusse. Sisendiks on binaarne andmetabel ja väljundiks on ümberjärjestatud tabel, kus tekitatud on objektide järjestus S , millest genereeritakse kontseptiahel.

Kolmnurga ehk Pareto struktuuri tekitamisel paigutatakse tabeli ühte nurka elemendid selliselt, et kõige suurema ühtede sagedusega (kus on kõige rohkem ühtesid) elemendid asuksid seal. Leksikograafilise sorteerimisega leitakse ühtne ja minimaalsete katkestustega ühtede jada, millest saab lõplikult kätte kontseptiahela. Käesoleva lõputöö teema valdkonna puhul on leksikograafilise tähestiku all mõeldud ainult kasutajaõiguste seose ehk 0/1 elemente.

3.2.2 Mitmete kontseptiahelate leidmise meta-algoritm

Ühe kontseptiahelaga on väga harvadel juhtudel võimalik leida täielik andmestiku katvus, sest paratamatult jäävad tulemustesse seosed (kasutajaõigused), mida ei ole suudetud ära katta. Terve konteksti katvuse probleemi lahendamiseks võtame kasutusele ahne meta-algoritmi, mis leiab formaalsest kontekstist $K = (G, M, I)$ funktsiooni $f(K)$ põhjal parima kontseptiahela C , arvestades maksimaalset vigade arvu lävendit δ (delta) [10]. Väljundiks on minimaalne kontseptiahelate järjestus $C_{seq} = (C_1, C_2, \dots, C_n)$ [10]. Eesmärk on leida kontseptiahel ning visata välja leitud ahelas kaetud seosed. Seejärel leida järgmised

ahelad, et katta järgmised seosed ja nii edasi kuni jõutakse selleni, et lävend delta on saavutatud ehk katvus on piisav. See tähendab, et algoritm on seadistatav leidmaks täieliku katmist või mitte ehk töö jätkub kuni katmata seoste (vigade) on väiksem või võrdne delta. Algoritm on heuristiline, mis tähendab, et ei garanteerita kõige optimaalsemat lahendust.

Hilisemalt leiame lõputöös CL-sort ja meta-algoritmiga tulemused näite andmestikel ja võrdleme tulemusi teiste FCA-le toetuvate meetodite ja rollikaeve meetoditega. Tulemustele toetudes analüüsime, kas käesoleva peatüki algoritmi tulemus on sobiv meetod rollide kaevandamiseks.

3.3 QualityCover - asjakohane katvus binaarmatriksist

QualityCover (edaspidi QC) lahendab formaalses kontseptianalüüsi juba teadaolevat probleemi, et tekitatakse liialt suur arv formaalseid kontsepte formaalsetest kontekstidest. Käesoleva meetodiga leitakse minimaalne katvus formaalsete kontseptidega, mis toetub ahnele (ingl k *greedy*) meetodile [12]. Viimane on optimeerimises väga laialdaselt kasutatav lihtne ja intuiitiivne lähenemine, kus igal sammul tehakse lokaalselt optimaalne valik. Käesoleva meetodi puhul valitakse igal etapil kontsept, mis katab kõige suurema arvu katmata elemente [12]. Lisaks on minimaalse katvuse leidmise eesmärk väga sarnane klassikalise hulga katvuse probleemile erinevatest valdkondadest, üritades leida väikseimat alam-hulkade kogumiku, mis katab vaadeldava süsteemi. Minimaalse katmise jaoks on vajalik teha igal sammul kindel otsustus ja selleks toetutakse GAIN funktsioonile hindamiseks kontseptide katvuse headust. Vaatleme järgnevalt, kuidas leitakse minimaalne katvus formaalsest kontekstist ja kuidas on see seotud GAIN funktsiooniga.

Teoreetilistes alustes defineeritud formaalsest kontekstist ja formaalsest kontseptist on võimalik tuletada pseudokontsept, mis on binaarne seos paaridele (a, b) kontekstist tähistatult PC_{ab} , kus leitakse maksimaalne hulk atribuute, mis täidavad objekti a ning vastupidisel maksimaalse hulga objekte, mis täidavad atribuudi b. Formaalselt tähistades

$$PC_{ab} = \{(g, m) \mid (g, m) \in G \times M \subseteq R \mid g \in G \wedge m \in M\}$$

Lihtsamalt öeldes on PC_{ab} seos kõikidest formaalsetest kontseptidest, mis sisaldavad paari (a, b) [12]. Edasiselt saame vaadata, mis on asjakohane katvus (ingl k *pertinent*) ning millest see koosneb.

GAIN funktsiooni saab defineerida formaalsest kontseptist $\langle A, B \rangle$ järgnevalt: $GAIN(\langle A, B \rangle) = (|A| \times |B|) - (|A| + |B|)$, kus $| \cdot |$ on hulga liikmete arvu operaator [12]. Asjakohase formaalse kontsepti defineerime, kasutades kaardistamiseks GAIN funktsiooni $P(M) \times P(G) \rightarrow [-1, 1]$, kus $P(M)$ ja $P(G)$ tähistavad objektide G ja atribuutide M lõplikke hulkasid ning formaalsete kontseptide hulka $F_k = \{ \langle A_i, B_i \rangle \mid i = 1, \dots, n \}$, mis leitakse formaalsest kontekstist K . Seega formaalne kontsept $\langle A_k, B_k \rangle \in F_k$ on asjakohane siis ja ainult siis, kui $\langle A_k, B_k \rangle = \text{argmax}_{i=1, \dots, n} \{ GAIN(\langle A_i, B_i \rangle) \}$. Sealt tulenevalt on lõplik asjakohane katvus formaalsest kontekstist K , katvus asjakohalistest formaalsetest kontseptidest, mis katavad kõiki paare binaarses suhtes R [12]. Teisisõnu on asjakohane katvus asjakohaste formaalsete kontseptide hulk, mis täidavad andmestiku täielikult. Vaatleme lihtsa näite põhjal, kuidas leitakse asjakohane formaalne kontsept [12]. Võttes arvesse PC_{1c} pseudokontsepti, seotuna paariga $(1, c)$ ehk $(\{12345\}, \{c\})$ ja nendest leitud nelja formaalset kontsepti $(\{12345\}, \{c\})$, $(\{123\}, \{ce\})$, $(\{1245\}, \{cf\})$ ja $(\{12\}, \{cef\})$, saame funktsiooniga kontseptidele väärtused: $GAIN(\{12345\}, \{c\}) = -1$; $GAIN(\{123\}, \{ce\}) = 1$; $GAIN(\{1245\}, \{cf\}) = 2$; ja $GAIN(\{12\}, \{cef\}) = 1$;

	a	b	c	d	e	f	g		c		c	e		c	f		c	e	f
1	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	1	0	1	1	1	2	1	2	1	1	2	1	1	2	1	1	1
3	0	0	1	0	1	0	1	3	1	3	1	1	4	1	1	3	1	1	0
4	0	0	1	0	0	1	0	4	1	4	1	0	5	1	1	4	1	0	1
5	0	0	1	1	0	1	0	5	1	5	1	0	3	1	0	5	1	0	1
6	0	1	0	0	0	1	1												
7	1	1	0	0	0	1	1												

Joonis 5 Formaalse konteksti näide

Tulemustest näeme, et kõige suurema väärtusega on kontsept $(\{1245\}, \{cf\})$, millest saab asjakohane formaalne kontsept. Edasiselt leitakse järgmistele paaridele $\langle A, B \rangle$ pseudokontseptid ja sealt juba GAIN funktsiooniga asjakohased formaalsed kontseptid, mis lõplikult tekitavad formaalsele kontekstile asjakohase katvuse.

3.3.1 Algoritmi kirjeldus ja näide

QualityCover eesmärk on leida parim võimalik katvus formaalsest kontekstist valides ahne meetodiga iteratiivselt igal etapil hulk, mis katab kõige suurema arvu katmata kasutajaõigusi. Katmata kasutajaõiguste leidmisel tekitatakse igast elemendist

pseudokontseptid ja kui see on koheselt täielik formaalne kontsept, siis peetakse seda asjakohaseks ning lisatakse kätte. Teisel juhul algoritmi töö jätkub, otsides pseudokontseptist kõik selle formaalsed kontseptid ja leides GAIN funktsiooniga nende väärtused. Suurima väärtusega formaalsest kontseptist saab asjakohane formaalne kontsept ja lisatakse kätte [12]. Eelnevate kirjeldatud sammude läbi kaetakse formaalne kontekst minimaalse arvu kontseptidega. QC on heuristiline algoritm ja leiab ligilähedaselt optimaalseima tulemuse.

Algoritmi töö illustratiivse kirjelduse toetamiseks kasutame algset tabelit (eelnev joonis) ja vaatame, kuidas leitakse asjakohane katvus. Kõik formaalse konteksti paarid (o, i) on esialgu järjestatud vähenevalt, vastavalt pseudokontsepti suuruse väärtusest. Seejärel, iga paari puhul arvutatakse selle pseudokontsepti $PC_{(o, i)}$, millest leitakse formaalsete kontseptide hulgad. Algoritmi esimeseks iteratsiooniks on läbi käia paar $(7, a)$, kus leitakse pseudokontsept, milleks on $PC_{7a} = \langle \{7\}, \{abfg\} \rangle$. Käesolevast näitest on lihtne, sest PC_{7a} on juba formaalne kontsept ja lisatakse kontsepti hulka F_k . Iteratsiooni lõpetamiseks üleliigsed paarid $(7, a)$, $(7, b)$, $(7, f)$, $(7, g)$ eemaldatakse. Protsess käib seni, kuni jõutakse viimase ehk antud näites 7-nda iteratsioonini, milleks on paar $(2, g)$. Pseudokontsept on arvutatud, mis on võrdeline $PC_{2g} = \langle \{2367\}, \{cefg\} \rangle$. Antud näitest on aga näha, et 4 formaalset kontsepti on peidetud $PC_{2g} - \langle \{23\}, \{ceg\} \rangle$, $\langle \{267\}, \{fg\} \rangle$, $\langle \{2\}, \{cefg\} \rangle$ ja $\langle \{2367\}, \{g\} \rangle$. Viimane on asjakohane, sest annab maksimaalse väärtuse ja formaalne kontsept $\langle \{2367\}, \{g\} \rangle$ on lisatud F_k -le. Pärast kõikide formaalse konteksti paaride katmist lõpetab algoritm töö.

4 Serieerimise meetodid

Serieerimine ehk järjestamine on visuaalne uurimuslik kombinatoorne andmeanalüüsi tehnika, kus järjestatakse objektid selliselt, et avaldub parim korrapärane ja mustiline esitus [11]. Üheks struktuurseks mustriliseks esituseks on Pareto serieerimine, mida vaatatakse juba lähemalt kontseptiahelate meetodi peatükis 3.2 ning mille samasugust esitust pakub monotoonsete süsteemide ümberjärestamise tehnika. Käesolevas peatükis laiendame konformismiskaala meetodit ja serieerimise tehnikat monotoonsetele süsteemidele.

4.1 Monotoonsed süsteemid

Käesolevas peatükis vaatame monotoonsete süsteemide algoritmide lähenemist, mille eesmärgiks on andmetabeli lõplik kuvamine parima otsustus (BD) [13] objektiga, mis esitab kaalufunktsiooni maksimaalse väärtuse. Valime välja ühe konkreetse andmetabeli järjestamise tehnikale toetuva algoritmi ja uurime seda lähemalt.

Selleks, et jõuda algoritmideni on vajalik lähemalt vaadata, kuidas BD objekt on kasulik ja rakendatav. Parima otsustuse puhul on leitav iga andmetabeli puhul „keskmine“ objekt, mis esitab diskreetseid andmeid [13]. Käesolevalt kasutame selleks konformismiskaala meetodit sageduse põhised. Mõõdetakse sarnasust ehk teisisõnu tüüpilisust teiste objektidega ja leitakse kõige sagedasem objekt, mis on kõige suurema kaaluga. Andmetabeli rea konformismiskaala väärtuse saamiseks leitakse kõikide selle atribuudi väärtuste sageduste summa. Kõige suurema väärtusega andmetabeli rida saab olema *Parim otsustus* [13]. Eelneva meetodi ja monotoonsete süsteemide teooriale toetuvalt on võimalik andmetabelite ümberjärestamiseks kasutada mitmeid algoritme: miinus-, pluss- ja segatehnikat.

Erinevate algoritmide ehitamiseks on vajalik eelnevalt andmetabel monotoonse süsteemi põhjal paika panna. Oletame, et eksisteerib selline andmetabel $X(A,B)$, kus iga element X_{ij} omab mingisugust diskreetset väärtust vahemikus $h_j = 0, 1, \dots, K_j-1$ [13]. Valides kaalufunktsiooni, tegevuse (näiteks „+“ ja „-“) ja kaalude ümberarvutamise reegli [13], tagame sellega süsteemi monotoonsuse. Ühe tegevuse rakendamine elemendile hulgas vastavalt suurendab või vähendab kõikide teiste elementide kaalusid, mis kuuluvad samasse hulka ja on seotud algse elemendiga. Kui elemendid ei ole seotud, siis kaal ei ole

muutuv. Monotoonsuse mõte seisneb selles, et elemendi eemaldamisel süsteemist muutuvad teiste elementide kaalud monotoonselt.

4.1.1 Miinustehnika algoritm

Miinustehnika on andmetabeli ümberjärjestamise tehnika, mis kasutab konformismiskaala meetodit [13]. Algoritmi nimetusest on mõistetav, et kasutatakse eelnevalt kirjeldatud „-“ tegevust ehk kaalude vähendamist. Segatehnika keerulisema lahenduse ja puuduva implementatsiooni põhjustel kasutame tulemuste leidmiseks käesolevas lõputöös miinustehnika algoritmi. Plusstehnika toetub samale loogikale ja annaks sarnaselt identsed tulemused, mistõttu ei ole selle algoritmiga tulemuste leidmine vajalik.

Miinustehnikas teostatakse ridade ümberjärjestamist alustades sageduste leidmist igale atribuudile veerus, mille abil saab arvutada ridade kaalude summad. Minimaalse kaaluga rida eemaldatakse ja alustatakse uuesti sageduste leidmist andmetabelile, milles puudub eelmises tsüklis eemaldatud minimaalse kaaluga rida. Tsükliliselt eemaldatakse ridu kuni andmetabelis ei ole enam ridasid. Lõpuks järjestatakse maatriksi read eemaldamise järjekorras ehk viimasena eemaldatud rida lisatakse tagasi andmetabelisse samuti viimasena. Veergude ümberjärjestamiseks rakendatakse sama algoritmi pööratud maatrikstabelile (st read ja veerud vahetatakse peale ridade ümberjärjestamist ära). Lõpuks pannakse ümberjärjestatud tabeli read ja veerud ühtekokku tabelisse koos seostega. Tulemuseks on algse andmestiku ümberjärjestatud Pareto serieerimise struktuur, mille illustreeriv näite esitus on kontseptiahelate peatüki 3.2 joonisel 4. Ümberjärjestatud andmetabelid ei anna meile otseselt rollide tulemusi, vaid pakuvad visuaalset esitust.

5 Rollikaevandamise kogukonna meetodid

Organisatsioonides juurutatavad RBAC mudelid toetuvad rollidele, milles sisalduvad kasutajate ja õiguste hulgad. See tähendab, et rollide hulgad peavad olema täielikult ja korrektselt määratletud ning sellekohast eesmärki täidavad mitmed rollikaevandamise algoritmid [14].

Järgnevalt vaatame nelja rollikaevandamise algoritmi, mis toetuvad kolmele erinevale lähenemisele. Algoritmide valik toetus kahele kriteeriumile. Esmane vajadus oli algoritmide implementatsioonide kättesaadavus rollide leidmiseks ja teiseks, et algoritmid oleksid võrreldavad teoreetiliste aluste ja tulemuste osas. Valitud algoritm HierarchicalMiner tugineb FCA teoriale kärpides vähendatud kontseptivõret ja Graafi optimeerimine rolli seoste optimeerimisele. Mõlema puhul on väljundiks rollide hulgad ja rollihierarhiad, mis annavad võimaluse visuaalseid esitusi võrrelda GSH meetodiga. Viimased kaks algoritmi CompleteMiner ja FastMiner toetuvad alamhulkade loendamisele ning annavad võimaluse neid kahte algoritmi võrrelda omavahel ning teiste rollikaeve meetoditega. Väljundiks on kandidaatrollide hulk [15]. Rollihierarhiat genereerivate valikutel seas oli kaalul ka ORCA algoritm, mis toetus õiguste hierarhilisele klasterdamisele, aga ei osutunud valituks seoses oluliselt ebaefektiivsete ja keerulisemate rollihierarhiate tulemuste tõttu [16].

5.1 Graafi optimeerimine

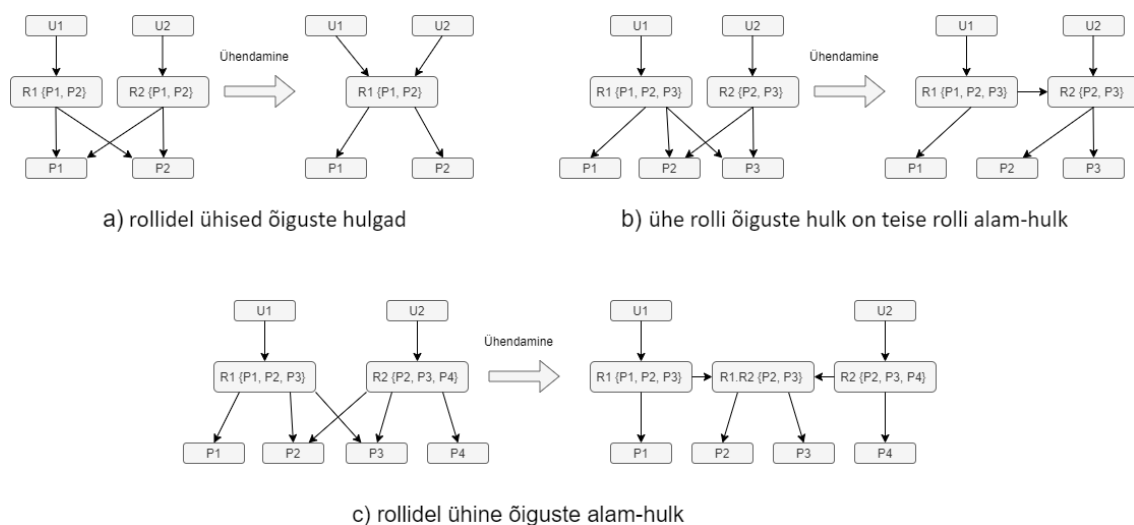
Graafi optimeerimisel kasutatakse *bottom-up* lähenemist rollide leidmiseks, mis on automatiseeritud rollitehnika peamine lähenemine, kaardistamaks juba kasutusel olevat juurdepääsuõiguste andmeid ja mille abil leitakse sobiv rollihierarhia [2]. Varasemalt on rollikaevandamise puhul kasutatud andmekaeve meetodeid, mille kitsendused rollide ja hierarhiate leidmiseks tekivad vastavalt valitud andmekaeve meetodist. Käesoleva meetodiga sarnaseid kitsaskohti ei leidu ning kasutajate ja nende õiguste kujutamiseks kasutatakse graafe ja matrikseid [2]. Lühidalt öeldes tuletatakse parima rollihierarhia leidmiseks kasutajate-rollide ja rollide-õiguste omavahelisi seoseid.

Graafi Optimeerimise algoritmi (edaspidi GO) eesmärk on parima tulemuse jaoks optimeerida ehk minimeerida rollide ja seoste koguarvu [2]. Terviklikult on igasuguse rollitehnikas esineva rollide leidmise eesmärk vähendada rollide ja nende seoste arvu

juurdepääsuõiguste juhtimise lihtsustamiseks. Sellekohast eesmärki näeme ka edasiste meetodite puhul. Ühtlasi on see graafi optimeerimise meetodi probleemi püstitus. Järgnevalt vaatame, kuidas see probleem meetodiga lahendatakse ja mis lahenduskaikude abil.

Esmalt koostatakse iga kasutaja õiguste hulgast üks roll, millest tekitatakse hierarhia graaf ja alustatakse edasist optimeerimist. Tuleb täpsustada, et koostatakse eraldi sõlmed isegi identsete õiguste hulkadega kasutajatele ning esialgse sammuga neid ei koondata. Seejärel leitakse optimeerimise mõõdik servade ja rollide arvult. Mõõdiku alusel on võimalik tsükliliselt algoritmil alustada rollipaaride võrdlust ja vastavalt optimeerimise väärtusele teostada ühendamise või tükeldamise tegevust [2]. Kahe rolli õiguste hulga kattuvuse korral on neli võimaliku olukorda, mida saab lühidalt kirjeldada järgnevalt.

Identsete õiguste hulkade korral liidetakse kaks rolli kokku üheks ning nende seosed uuendatakse (joonis 6a). Joonisel 6b on näidatud, et kui ühe rolli õiguste hulk on teise rolli alam-hulk, siis luuakse seos ülem-hulgast alam-hulka ja ülem-hulga seosed eemaldatakse, sest hulkade vahelise seose tõttu oleks see juba liiasus. Kolmandal juhul oleks rollide hulkade vahel ainult üksikud jagatud õigused ehk kattuvus, mille korral tekitatakse uus kolmas roll, mis sisaldab eelnevaid ühiseid kaetud õigusi ja uus roll seotakse esialgsete rollidega. Seda illustreerib joonis 6c. Olukordade muustrit vaadates on võimalik juba mõista, et neljas olukord tähendab võrdlust, mille korral ei ole õiguste hulgad kuidagi kattuvad ning ühtegi tegevust ei teostata. Vastupidiselt graafi optimeerimise eesmärgile ei ole selles olukorras võimalik tekitada uut rolli uue õiguste hulgaga, mis suurendaks optimeerimise mõõdikut [2].



Joonis 6 Rollide ühendamise olukorrad

Välja oleme jätnud toomata iga olukorra optimeerimise mõõdiku muutused, milleks esimesel juhul on alati väärtuse täienemine, kaotades ühe rolli ja servade arvusid vastavalt hulgas olevatele õigustele. Teisel juhul omakorda lahkneb optimeerimise mõõdik muutumatusele või vähenemisele, mis sõltub sellest, kas alam-hulgaga rollil on vastavalt üks või mitu sarnast õigust seotud teise rolliga. Kolmandal juhul on optimeerimise kasv sõltuv uuele rollile omistatud ühiste kattuvate õiguste arvuga, sest sõltuvalt sellest väheneb servade arv. Mõõdiku muutus täieneb alates kolmandast õigusest ehk kuni kahe ühise kattuvusega on mõõdik muutumatu. Nagu eelnevalt välja toodud ei ole neljandas olukorras võimalik optimeerimise mõõdikut vähendada, mistõttu seda tegevust ei teostata.

Lühidalt saab kirjeldada, et algoritm esmalt genereerib algse binaarse andmetabeli põhjal rolle ühe kasutaja kaupa, kus igal kasutajal on mingisugune hulk õigusi ning seejärel optimeerib servade arvu vastavalt võrdsusele, alam-hulkadele ja kattuvustele (eelnevalt vaadeldud olukorrad) kuniks rohkem täiendamisi ei ole enam võimalik teostada. Tulemuseks on rollihierarhia graaf, mida saab kasutada juurdepääsuõiguste juhtimises [2]. Lisaks väljastatakse rollid ning nendega seotud kasutajate ja õiguste hulgad.

5.2 Rollide kaevandamine vähendatud kontseptivõre kärpimisega

Käesolev meetod pakub FCA teooriale tuginedes rollikaevandamist kasutajaõiguste seoste lähenemise põhiselt ja on suutnud näidata eksperimentaalselt rollide ja rollihierarhiate loomist arvestataval tasandil teiste traditsiooniliste rollikaevandamiste algoritmidega. Antud lähenemine on edasiarendus kontseptivõre ehk Galois võre meetodist kontseptide (A_1, B_1) ja (A_2, B_2) vahel, mida saab defineerida kui

$$(A_1, B_1) \leq (A_2, B_2) \leftrightarrow A_1 \subseteq A_2 \leftrightarrow B_1 \supseteq B_2, \text{ kus}$$

teadupärast iga kontsept võres pärib õigused oma alam-kontseptidelt ja vastupidiselt kasutajad ülem-kontseptidelt. See tähendab, et kontseptides ehk sõlmedes sisalduvad õigused ja kasutajad on mitmekordselt esindatud ja võimalik on need eemaldada. Selle tulemusena saame kätte vähendatud kontseptivõre, mida võib nimetada juba täielikuks RBAC seisundiks (ingl k *state*) [16] ning mille väärtus koosneb rollide, rolli hierarhia, kasutaja-rolli seose ja õiguse-rolli seose arvust, mida vaatame lähemalt tulemuste osas. Vähendatud kontseptivõres on iga kasutaja seotud ühe kindla rolliga ja mitte rohkemaga ehk kontsept on rolliks ja võre on käsitletav rollihierarhiana [16], mis on väga sarnane

lähenemine peatükis 3.1 vaadeldud GSH diagrammidele. Puuduseks on vähendatud kontseptivõrede puhul kontseptid, mis ei sisalda kasutajate või õiguste silte ning jättes rollihierarhia ebavajalikult suureks. Seega on võimalik edasiselt vähendatud kontseptivõrede kärpimist teostada, aga selleks peab olema mingisugune konkreetne mõõdik tegevuse headuse hindamiseks.

Selleks tutvustame kaalutud struktuurse keerukuse (wsc) mõõdikut, mida kasutatakse RBAC seisundi ehk rollihierarhia optimaalsuse hindamiseks [16]. Definiitsiooni kohaselt on $W = \langle w_r, w_u, w_p, w_h, w_d \rangle$, kus kaalud on positiivsete ratsionaalarvude hulk. Selle põhjal on kaalutud struktuurse keerukuse RBAC seisund γ võrdeline:

$$wsc(\gamma, W) = w_r * |R| + w_u * |UA| + w_p * |PA| + w_h * |t_reduce(RH)| + w_d * |DUPA|,$$

kus $|\cdot|$ tähistab hulga või seose suurust ning $t_reduce(RH)$ on rollihierarhia transitiivne vähenemine, mis sisuliselt tähistab hierarhia minimaalset seoste hulka [16]. DUPA on otsene kasutajaõiguse seos ($U \times P$) [16].

Kui eelnevalt vaadatud GO puhul oli optimeerimise mõõdikuks rollide ja servade arv, siis praeguse mõõdiku väärtuse keerukus tuleneb hierarhia seoste arvust ja nendele seostele kehtivate kaalude määrast. Seega on eesmärgipõhiselt võimalik algoritmi väljundit muuta, seadistades kaaluparameetreid. Tavapäraselt on eesmärgiks rollihierarhia kõige väiksem wsc väärtus ehk rolli seisundi kadu (w_d), hierarhia (w_h), õiguste (w_p), kasutajate (w_u) ja rollide (w_r) kaalude väärtused on ühtlustatud (määratakse väärtus „1“). Rollide arvu minimeerimise eesmärgil seadistatakse kaalud järgnevalt:

$$w_r = 1, w_u = w_p = 0 \text{ ja } w_h = w_d = \infty. [16]$$

Tulemusi vaatame hilisemas peatükis, kus rollikaevandamise tööriistale anname sisendiks erinevad parameetrite seadistused.

HierarchicalMiner algoritm leiab optimeeritud rollihierarhia, kärpides juba vähendatud kontseptivõret, mis omakorda leitud lihtsast konteksti kontseptivõrest. Vähendatud kontseptivõre on esmase rollihierarhia põhjaks ja heuristiliselt optimeeritakse seda kaalutud struktuurse keerukuse (wsc) mõõdiku põhiselt kuniks leitakse parim tulemus ja selle tulemus sisalduvad rollid [16]. Algoritm käib iteratiivselt üle kõikide esmaste rollide ja teostab kärpimist vastavalt rollihierarhia väärtuse muutustele ehk peatub alles siis, kui enam tegevusi vähenemisele ei ole võimalik teha. Kärpimist teostatakse järgnevatele kolmele reeglile alluvalt [16]:

1. Eemaldatakse rollid, milles puuduvad kasutajate või õiguste sildid, aga ainult tingimustel, et eemaldamine on kasulik ja pärimise seosed jäävad korrektseks. Viimase puhul on vajalik uued seosed sõlmede vahel kokku viia, et hierarhia oleks jätkuvalt eksisteeriv.
2. Puuduvate õigustega roll eemaldatakse, lisades rollis olevad kasutajad igasse alam-rolli $Jun(r)$ ja $Thr(r)$ lisatakse rollihierarhiasse RH, et mitte kaotada seosed rollide vaheliselt. $Thr(r)$ on selline rollihulkade paar (r_i, r_j) , kus ilma rollita r , ei oleks r_i rolli r_j vanem [16].
3. Eemaldatakse roll, millel on mõned õigused, aga mitte ühtegi kasutajat. Siis jagatakse rolli õigused igale senior rollile $Sen(r)$.

Lihtsamalt öeldes optimeeritakse vähendatud kontseptivõret, viies hierarhias olevaid sõlmesid kokku üles-alla liikumiste seoste alusel. Algoritmi töö tulemuseks on rollide hulgad ning naturaalselt genereeritud rollihierarhia, mis on näidanud üldiselt paremaid tulemusi, kui teised rollikaevandamise algoritmid [16].

5.2.1 Rollide kaevandamine kasutades semantikat

Lisaks on võimalik rolle kaevandada lisainformatsiooni põhjal, mis ei ole ainuüksi kasutajaõigused. Peaaegu kõik organisatsioonid hoiavad mingisugust infot kasutajate kohta, sest see kirjeldab nende tööülesandeid, seoseid ja staatuseid. Näiteks võib IT ala programmeerijal olla seos projekti või projektidega, arenduspositsioon või isegi oskuslitsentsid. Seda saab rollikaevandamisel määratleda kasutaja-atribuudiga (UAT) ning tähistada 0/1 väärtusega, mis määrab semantilise tähenduse ning seeläbi rollide tähendus rohkem seotud reaalse maailma olukorraga [16]. Semantika on teisiti öeldes inimeste asjatundlik teadmine täiendamaks rollide kaevandamist [17]. Algse rollide kaevandamise algoritmile on peale ehitatud AttributeMiner algoritm, mis võimaldab võtta arvesse semantilisi tähendusi ja genereerida optimeeritud RBAC seisundeid ning uusi rolle. Selline lahendus pikendab veelgi automatiseeritud rollitehnikat, lihtsustades RBAC juurutamist. Käesolevas lõputöös ei puudutata seda osa, sest antud rollide kaevandamise laiendamine vajab sarnaste ja alternatiivsete meetodite uurimist ning võrdlemist detailsemalt.

5.3 Rollide identifitseerimine alamhulkade loendamise meetodil

Antud lõputöös käsitleme veel kahte algoritmi FastMiner (FM) ja CompleteMiner (CM), mis võimaldavad leida rolle tuginedes alamhulkade loendamisele. Alamhulkade loendamise eesmärk on avastada rollid õiguste andmehulkadest, mis on omatud kõikide kasutajate poolt [18]. Lihtsamalt öeldes käsitletakse rolle kui teatud õiguste grupe ehk hulkasid.

Õiguste grupeerimise kirjeldamise järgi on meetod väga sarnane klasterdamisele ja teistele sarnaselt üles ehitatud rollikaeve algoritmidele, kus eesmärgiks on sarnaste õiguste klasterdamine rollidesse. Üheks konkreetseks puudujäägiks klasterdamise puhul on rakenduslik sobivus rohkem arvandmetele, mis aga ei sobi rollikaevandamises kasutajaid ja õigusi määratlema, sest õigused on jaotatud individuaalselt rollide vahel [18]. Sealt tulevalt tekib vajadus uue rollide kaevandamisele sobiva meetodi leidmiseks. Vaatleme lähemalt kahte printsiipi, millele toetub rollide leidmine:

1. Vaadeldakse ainult neid õiguste alamhulkasid, millele on omistatud vähemalt üks kasutaja. Seda põhjusel, et kui mingisugusel õiguste hulgal $\{p_a, p_b\}$ puuduvad kasutajad, siis eeldatavasti see õiguste hulk ei tekita rolli [18]
2. Leitakse rollid, mis sisaldavad kõige enam õigusi, olles sarnane ahnele meetodile, et vähendada niigi suurte tekkitavate rollide arvu. Kahe erineva õiguse p_a ja p_b korral tekitatakse õiguste hulk $\{p_a, p_b\}$, kui needsamad õigused on seotud kõikide kasutajatega.

Õiguste hulkade puhul, kus üks hulk on teise hulga alam-hulk, tekitab meetodi puhul kitsaskoha lõplike rollide leidmise kohapealt, sest teatud tingimustel peame arvestama kõikide potentsiaalsete rollidega tekkivas andmestikus. See tähendab, et lähenemine peab jagunema kaheks sammuks. Esimeseks sammuks on rollide identifitseerimine ja teiseks tähtsuslikkuse väärtuse ehk prioriteetide määramine rollidele. Järgnevalt ongi välja toodud kaks algoritmi, mis võimaldavad seda teostada, võttes sisendiks kasutajaõiguse informatsiooni andmeid ja väljastades kõik leitud kandidaatroolid prioriteetide järgi. Esimeses faasis leitakse kandidaatroolid hulk, mis on üldjuhul suur number rolle ja teises faasis määratakse igale kandidaatroolile prioriteedi väärtus, kus väiksema väärtusega rollid on tähtsamad ja kasulikumad [15].

5.3.1 CompleteMiner

Kahest vaadeldavast algoritmist keerukam on CompleteMiner, millel on eksponentsiaalne keerukus $O(n^3)$. Autorite poolt esitatud eksperimentide põhjal on väidetud, et algoritmi töö on ebapraktiline või ebamõistlik väga suurte andmestike puhul ning sobib kuni keskmiselt väikeste andmehulkade rollikaevandamiseks [18]. Algoritm esmalt genereerib esialgsed rollid iga kasutaja õiguste hulgale ja seejärel ristab need esialgsed rollid ja uued genereeritud rollid, et kätte saada veelgi rohkem kandidaatrolle, kuniks enam ei ole võimalik rolle genereerida. Algoritm sisaldab kahte tähtsat etappi rollide leidmisel:

1. Algsete rollide hulkade identifitseerimine – grupeeritakse kasutajad, kellel on täpselt samad õiguste hulgad, mis vähendab andmehulga suurust. Moodustatakse esialgne rollide hulk [18].
2. Alamhulkade loendamine – leitakse kõik potentsiaalsed huvitavad rollid, otsides kõik võimalikud rollidest tekkinud lõikepunktide hulgad. Tulemuseks on protsessi käigus genereeritud maksimaalne hulk huvitavaid rolle.

Nagu näha tekitatakse andmestiku kasvades eksponentsiaalselt potentsiaalseid rolle lõikepunktidest ja leitud rollide hulk võib tulemustena olla ebamäärane ja mitte sobiv.

5.3.2 FastMiner

FastMiner on CompleteMiner algoritmi efektiivsem ja lihtsam implementatsioon, kus ainukesed lõikepunktid leitakse esialgsete rollide vahel (etapi esimene punkt) ja mitte kõikide potentsiaalsete huvitavate rollide vahel. Algoritm on suures osas identne eelnevalt vaadeldud algoritmiga, kuid võtmepunktides on keerukus vähenenud ja sealt tulenevalt on see $O(n^2*m)$. See tähendab, et algoritm on suurte andmehulkade jaoks sobilikum, aga eelneva info alusel võib pidada puuduseks kandidaatrollide alam-hulkade identifitseerimist. Õnneks on praktiliste eksperimentide põhjal kinnitatud, et algoritmi poolt leitud rollide hulgad on küllaldased selle kasutamiseks [18].

Küsimus seisneb selles, kas rohkemate kandidaatrollide arvestamisel lõikepunktides on tulemused täpsemad. Andmekaeve probleemide korral keerukuse lisamine võib pakkuda üldjuhul paremaid tulemusi, kuid rollikaevandamises tähendab keerukuse lisamine rohkemate kandidaatrollide arvu ja pärsib oluliselt tähtsate ehk kasulike rollide identifitseerimisel [18]. Hilisemalt näeme missugused tulemused leitakse mõlema algoritmi poolt lõputöö kolmel lähteandmestikel.

5.4 Visuaalne rollikaevandamine

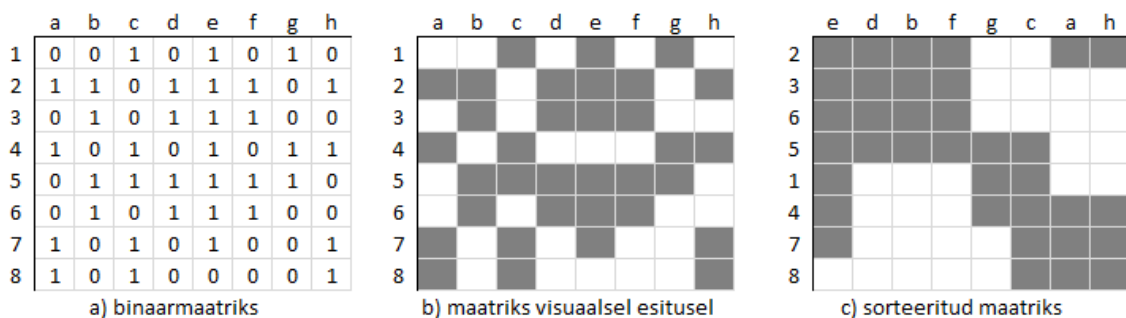
Visuaalne rollikaevandamine on uus lähenemine terves rollikaevandamise valdkonnas, mille eesmärgiks on lihtsustada rollide leidmist visuaalse esituse teel. Erinevus varasematest rollitehnikas kasutatavatest meetoditest on visuaalse rollikaevandamise omadus leida tähtsad rollid visuaalselt, ilma traditsiooniliste rollikaeve meetodite rakendamiseta [19]. Teisisõnu on eesmärgiks analüüsida juurdepääsuõiguste andmeid tuues esile kogum olulisi rolle, mis lihtsustavad RBAC haldamist.

Visuaalse rollikaevandamise meetod toetub kahele algoritmile - Adviser ja Extract [19], millest viimane on rakendatav esimese algoritmi tulemuste eeldusel. Adviser eesmärk on rollide visuaalse esituse loomine kasutajaõiguste andmetest ja algoritmi lühend tähendab juurdepääsuõiguste andmete visualiseerimist. Extract eesmärk on ilma tavapäraste andmekaeve meetodite rakendamiseta toetuda Adviser tulemustele realiseerides rollide visuaalset esilekutsumist. Lõplikult saavutatakse tulemused, mis pakuvad lisaks hästi toetatud rollidele veel ka visuaalseid tulemusi.

5.4.1 Rollide visualiseerimise probleem

Selles peatükis uurime lähemalt rollide visualiseerimise probleemi. Miks seda lahendada ja kuidas visuaalsed esitused mõjutavad andmete analüüsi võrreldes statistiliste meetodite ja tavapäraste rollikaeve algoritmidega. Käesolevaga saame vastused visuaalse rollikaevandamise eesmärgile. Proovime näidata, et rolle on lihtsam tuvastada visuaalselt binaarse andmematriksi abil, kui seda on teksti kujul.

Nagu teame on juurdepääsuõiguste andmed rollikaevandamises kõige standardsemal kujul binaarmatriksi esitusel [19]. Alloleval joonisel on esitatud lihtne näide sellest, kus veergudes ja ridades asuvad vastavalt kasutajad või õigused ning tabeli sisus avalduvad seosed nende vahel.



Joonis 7 Binaarmatriksi visuaalse esituse loomine

Binaarmaatriksi esitust on võimalik veel abstraktselt kirjeldada iga kasutaja ja õiguse vahekorra nagu on näidatud allolevas tabelis. Sellisel kujul juurdepääsuõiguste andmed ei ole kergesti analüüsitavad ega edasiselt rollidesse kantavad.

Tabel 3 Kasutajaõiguste seosed

Kasutajaõiguste seosed
{⟨1,c⟩, ⟨1,e⟩, ⟨1,g⟩, ⟨2,a⟩, ⟨2,b⟩, ⟨2,d⟩, ⟨2,e⟩, ⟨2,f⟩, ⟨2,h⟩, ⟨3,b⟩, ⟨3,d⟩, ⟨3,e⟩, ⟨3,f⟩, ⟨4,a⟩, ⟨4,c⟩, ⟨4,e⟩, ⟨4,g⟩, ⟨4,h⟩, ⟨5,b⟩, ⟨5,c⟩, ⟨5,d⟩, ⟨5,e⟩, ⟨5,f⟩, ⟨5,g⟩, ⟨6,b⟩, ⟨6,d⟩, ⟨6,e⟩, ⟨6,f⟩, ⟨7,a⟩, ⟨7,c⟩, ⟨7,e⟩, ⟨7,h⟩, ⟨8,a⟩, ⟨8,c⟩, ⟨8,h⟩}

Tulles tagasi joonisele on võimalik numbritest koosneva binaarmaatriksi esitust rohkem loetavale kujule konverteerida (joonis 7b) ja järjestada andmetabeli ridasid ja veergusid, tekitades hulga paremini analüüsitava esituse (joonis 7c). Viimasest esitusest võib juba leida rolle ning need kasutajate ja õiguste hulkadega märkida allolevas tabelis. Sellest tulenevalt saame järeldada, et rollide ekstraheerimist saab teostada juba ainuüksi visuaalsetele esitustele toetudes ning ilma rollikaeve algoritmide rakendamiseta [19]. Kui praeguse näite puhul piisas andmetabeli ümberjärjestamisel käsitsi tehtud tööst, siis väga suurte andmestike töötlemiseks on vajalik kasutada rollikaeve algoritme.

Tabel 4 Teksti kujul kandidaatrollide esitus

Roll	Õigused	Kasutajad
R1	{1,2,3,4,5,6,7}	{e}
R2	{2,3,5,6}	{b,d,e,f}
R3	{1,4,5}	{c,g}
R4	{4,7,8}	{a,c,h}
R5	{2}	{a,h}

Antud lihtsast näitest on võimalik veel lisaks mitmeid järeldusi teha. Esmalt aitavad visuaalsed esitused efektiivselt näha andmetes avalduvaid erijuhtumeid, kus näiteks ühel kasutajal on üksikisikuliselt antud unikaalne kasutajaõigus, avaldades vale õiguste määratlemise. Lisaks on paremini tuvastatav rollide kattumine visuaalsest esitusest, mistõttu ei ole vajalik väljastada infot kasutajaõiguste kohta, mis erinevates rollides kattuvad.

5.4.2 Adviser – maatriksi sorteerimise algoritm

Järgnevalt vaatame visuaalse rollikaevandamise esimese osa algoritmi Adviser, mille eesmärgiks on järjestada ümber read ja veerud kasutajaõiguste maatriksis selliselt, et oleks minimeeritud iga rolli eraldatus nagu oli esitatud eelnevas peatüki näites. Algoritm on üles ehitatud kolmele põhimõttele, mida järgitakse ümberjärjestamisel, pakkudes kiiret ja head, kuid mitte alati kõige optimaalsemat tulemust käsitletavale probleemile [19]. Esiteks ja kõige tähtsamalt peaksid olema suuremad rollid andmetabelis kõige paremini esitatud. Teiseks tuleneb rolli visualiseerimise kulu eraldatusest ehk selle keerukusest [19].

Esitame järgnevalt algoritmi tööprotsessi kirjelduse, mille kaudu jõutakse sorteeritud andmetabeli parima esituseni:

1. Enne andmetabeli sorteerimise alustamist pannakse paika, et read ja veerud järjestatakse eraldi ja sõltumatult ehk tekitatakse eraldi sorteerimise hulgad. Kuna protsess toimub kahel korral, siis nimetame edaspidi neid hulkasid ühtse mõistega „elemendid“.
2. Üksikute elementide asemel sorteerib algoritm elementide gruppi, sest ühte rollide hulka kuuluvad elemendid koondatakse kokku.
3. Lähtudes algoritmi põhimõtte esimesest punktist, siis suuremate rollide paremaks visuaalseks esituseks analüüsitakse mahukamate pindaladega rollidesse kuuluvaid elementide hulka esimesena. Järgnevad kaks punkti töötlevad elemente käesoleva punkti tsüklis.
4. Suurte lünkade vältimiseks tulemustes kasutatakse *Jaccard* koefitsienti, et paigutada lähestikku elementide hulgad, mis jagavad ühiseid suuremaid rolle.
5. Eelistatult paigutatakse iga elementide hulk juba ümberjärjestatud elementide hulga ette või taha sarnasuse väärtuse alusel, et mitte rikkuda juba leitud elementide hulga vahelisi sarnasusi.

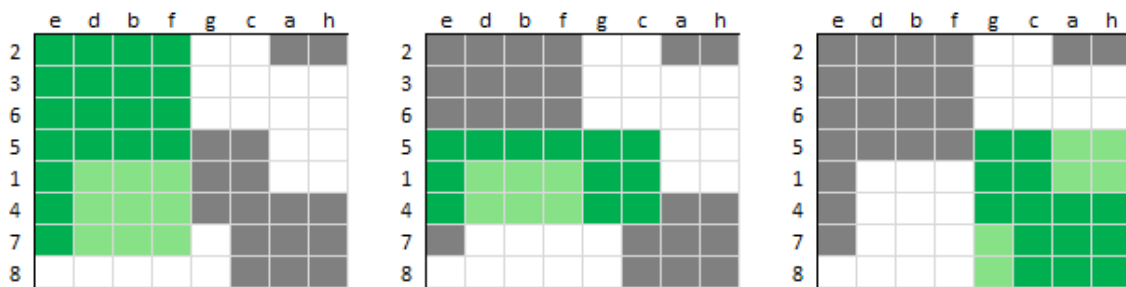
Toetudes nendele punktidele grupeeritakse kasutajate ja õiguste hulgad rollidesse ning leitakse nendele sobiv järjestus, pannes näiteks kaks esimest kasutajate hulka esmalt üksteise järgi ja selle peale ehitades leitakse kõikide hulkade sobiv paigutus andmetabelis.

5.4.3 Extract algoritm - Rollide visuaalne esilekutsumine kasutades pseudorolle

Rollide visualiseerimise probleemi peatükis käsitlesime, et hästi loetava kasutajaõiguste maatriksi puhul on võimalik rollitehnikas leida ainuüksi andmetabeli visualiseerimisega

rollid. Samuti järeldasime, et enamus juhtudel see ei ole piisav tulem ja eelnevalt leitud visuaalne esitus peab jätkuma lahendusega, mis suudab sellest esitusest leida rollide hulga. Suurte andmestikega töötamisel on vajalik sellised tegevused algoritmide abil lahendada, sest käsitsi järjestamine hinnanguliselt optimaalse tulemuse saamiseks on väga raske tegevus.

Rollide leidmiseks kasutatakse algoritmi nimega *Extract*, et leida pseudorollid [19]. Pseudorollide peamine eesmärk on esile tõsta mustrid, mis võivad olla huvi pakkuvad ja pidada kinni põhimõttest kuvada kõige suurema katvusega pinnad (kasutajaõiguste hulga). Nimetus „pseudo“ tuleneb antud juhul nendest kasutajaõigustest, mis ei ole otseselt nähtavad vaadeldavas andmehulgas, kuid on analüüsitavas alam-maatriksis. Eesmärgiks on, et pseudorollide poolt esile toodud alad kataksid kõikvõimalikud rollid, mida on üldse võimalik kujutada [19]. Allolev joonis on illustreerib pseudorollide kasutust algse andmestiku näitel. Rohelised seosed on kasutajaõigused, mis kaetud pseudorollide poolt ja hele-rohelised on mitte-eksisteerivad kasutajaõigused pseudorollis.



Joonis 8 Pseudorollide näide

Extract algoritm võtab sisendiks Adviser algoritmi poolt leitud tulemused ja kujutab kõik kasutajaõiguste seosed ning samuti mitte-eksisteerivad kasutajaõiguste seosed kogu pseudorollide hulkades. Tulemused järjestatakse sagedusele vähenemise põhiselt ehk eespool on sellised rollid, mis katavad võimalikult suured alad ja võimalikult vähese mitte-eksisteerivate kasutajaõiguste seostega. Algoritm sarnaneb QualityCover algoritmis vaadeldud GAIN funktsioonile, et katta kasutajaõigused minimaalse ja kõige suurema väärtusega kontseptide/rollidega.

5. Lähteandmestikel genereeritud tulemused vaadeldavate meetoditega

Lõputöös vaadeldavate algoritmide erinevate rakenduste ja implementatsioonidega leitud eksperimentaalsed tulemused olid genereeritud üle mitme erineva seadme. Seda põhjusel, et lahendused olid kättesaadavad ainult teatud seadmetes. Allolevas tabelis on välja toodud vastavad algoritmid ja seadmete näitajad, millega need rollide kaevandamiseks käivitati.

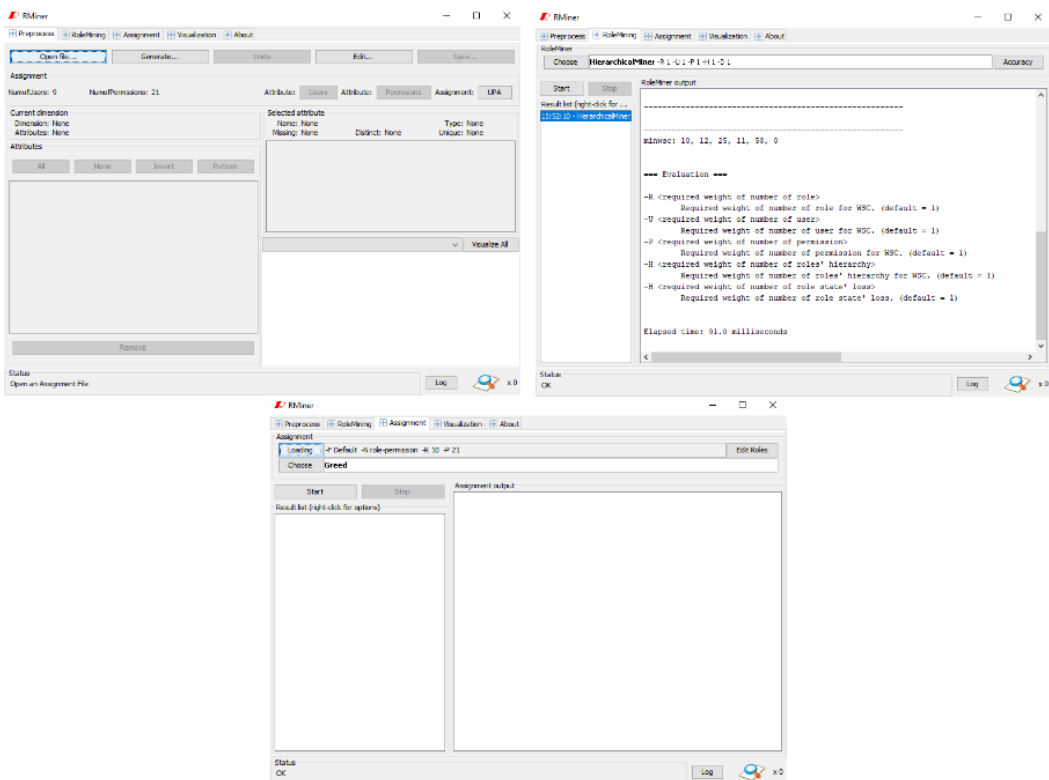
Tabel 5 Algoritmide rakendusi käsitlevate seadmete tehnilised andmed

Algoritm	CPU	RAM
QualityCover	i7-8665U @ 1.90GHz	32GB
GSH	i7-2620M @ 2.7GHz	4GB
Graafi Optimeerimine		
HierarchicalMiner		
CompleteMiner		
FastMiner		
Miinustehnika	i5-6600K @ 3.50GHz	8GB
CL-sort (Kontseptiahelad)		

Eelnevas peatükis vaadeldud traditsiooniliste rollikaeve meetodite tulemused olid leitud läbi RMiner rakenduse. Tegemist on interaktiivse rollikaevandamise tööriistaga, mis koondab erinevaid rollikaeve meetodeid ning võimaldab erinevate parameetrite seadistusega algoritme käivitada [20]. RMiner põhineb avatud lähtekoodiga masinõppele toetaval andmekaeve rakendusel nimega WEKA.

Rollitehnika lõplikuks eesmärgiks on leida sobiv rollipõhine juurdepääsuõiguste juhtimine (RBAC), mis eeldab rollide genereerimist, nende määratlemist ja staatuste muutmist, et olla vastav organisatsiooni ja turvalisuse nõuetele. Käesolev rakendus seda võimaldabki, millest antud lõputöös kasutame ainult vajalikku funktsionaalsust rollide genereerimisel. Selleks kasutame kasutajaõiguste informatsiooni importimiseks andmete eeltöötuse osa, rollide leidmiseks algoritmide ja nende parameetritega rollide kaevandamise funktsionaalsust ja rollihierarhiat sisaldavate algoritmide diagrammide visuaalseks esituseks rollide määratlemise funktsionaalsust (Joonis 9). Importimise jaoks on vajalik lähteandmestik konverteerida käsitsi andmeformaadi kujule, mis sisaldab

formaadi tüüpi, kasutajate ja õiguste nimetusi ja binaarmatriksit. Lõputöös vajalikud tulemused avalduvad rollidena *RoleMining* vahelehel ja *RoleMiner Output* plokis (joonis 9), kus on kirjas algoritmile määratud seadistused, failis sisalduvad kasutajad ja õigused ning rollide hulgad nendes sisalduvate kasutajate ja õiguste hulkadega. Teatud algoritmide puhul on kuvatud lisaks kaalutud struktuurne keerukus.



Joonis 9 RMiner rakenduses kasutatavad vaated rollide kaevandamiseks

6.1 Lähteandmestikud

Antud lõputöös on meetodite tulemuste võrdlemiseks kasutusel kolme organisatsiooni veebikeskkonna süsteemide juurdepääsuõiguste andmed. Kaitsmaks organisatsioonide ärisaladusi, ei avaldata nende nimesid ega detailsemat infot.

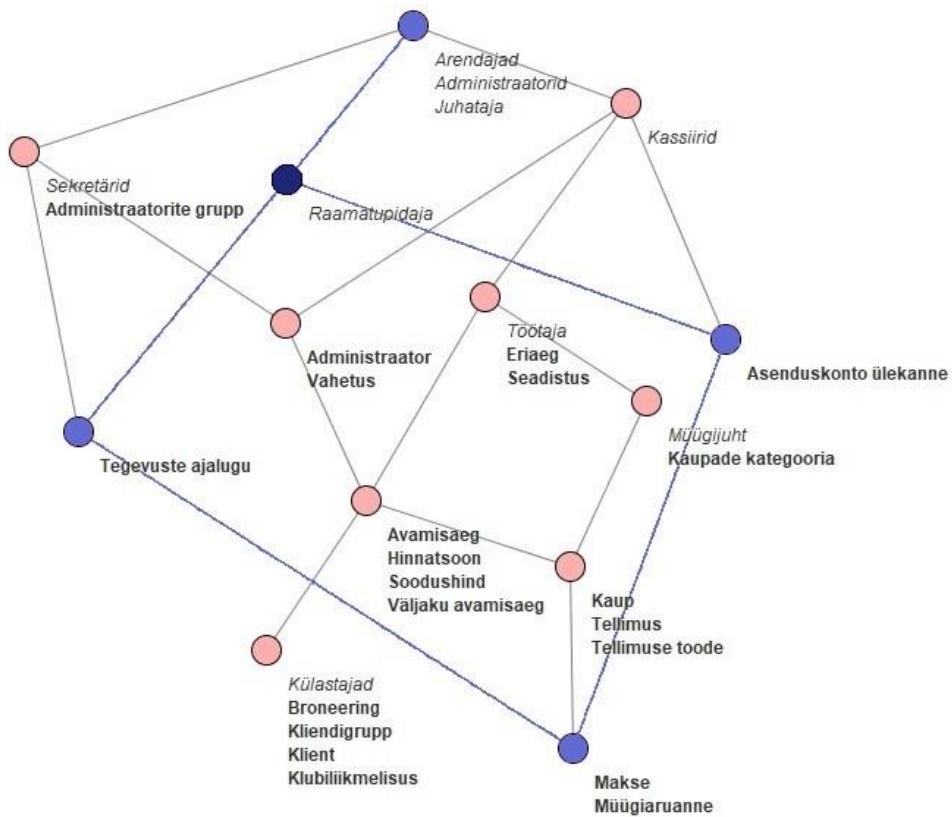
Süsteemide juurdepääsuõiguste andmed on eksporditud või käsitsi ümber kirjutatud vastavalt käesoleva töö analüüsi jaoks sobivale formaadile (binaartabelile). Esitatud lähteandmed olid enne valikut teiste juurdepääsuõiguste andmetega võrreldud ning keerukust hinnatud, osutudes valituks seoste struktuuri ja sobiva koguse hulga järgi. Kahjuks puudusid valikute hulgast väga suurte kasutajaõiguste seoste arvuga andmestikud.

Esimeseks näiteks on veebirakenduse kasutusõiguste süsteem, kus kasutajad ja õigused on omavahelise andmebaasitabeliga seotud. Süsteemis on 9 kasutajagruppi ja 21 moodulit, millel on omavaheliselt 128 kasutajaõigust. Kutsume seda lähteandmestiku näidet järgnevalt – spordikeskus. Teiseks süsteemiks on eelmisele sarnane veebirakenduse platvorm, kus on 16 kasutajat, 63 moodulit koos 373 kasutajaõigusega. Nimetame selle näite – CMS. Kolmandaks näiteks on 20 kasutaja, 16 õigusega ja 91 kasutajaõigusega süsteem, mida nimetame – Intranet. Andmestike kasutajate ja õiguste seosed on tabeli kujul välja toodud LISA 1.

6.2 GSH diagrammid

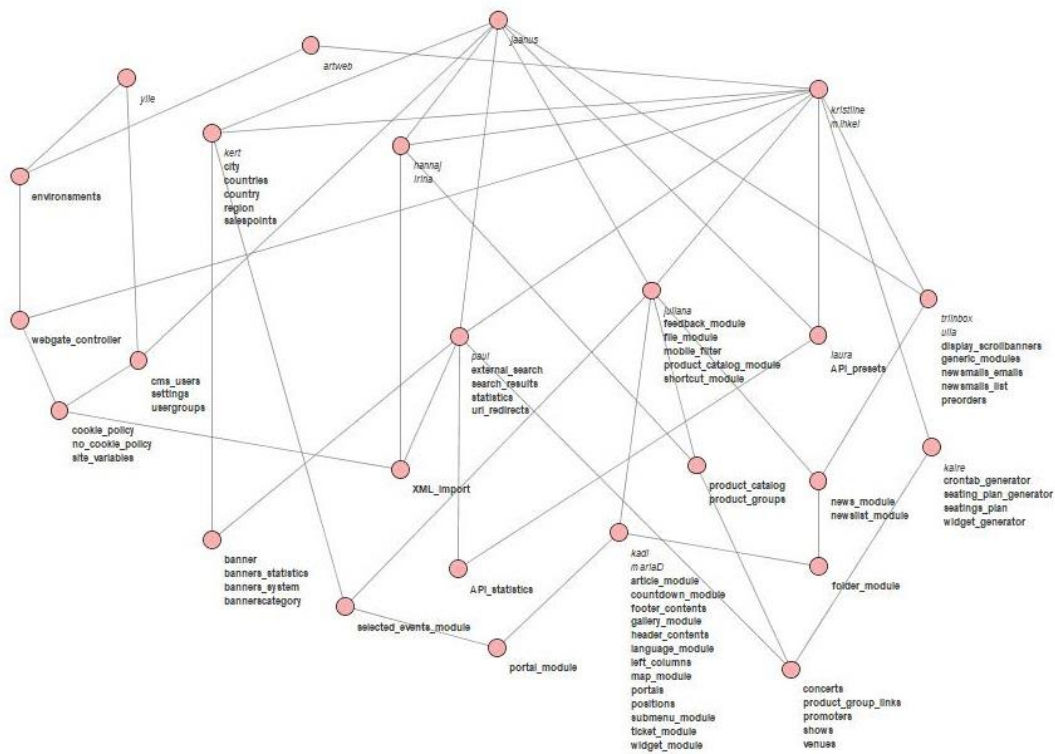
Käesolevas peatükis vaatame GSH rakenduse tulemusi, mis on leitud kolmest andmestikust. Tulenevalt meetodi tingimustest visuaalsele esitusele on diagrammid käsitsi ümber paigutatud lõputöö autori poolt. Rakenduse eesmärk ei ole sõlmede ja nende seoste visuaalne paigutus. Diagrammi sõlmede paigutused on korrektselt teostatud kontrollituna lähteandmestikest, aga ei määratle kindlasti kõige paremat paigutust. Allpool on esitatud kolme joonisega tulemused, kus kõik sõlmed on rollid või rollikandidaadid ning seosed tähistavad rollihierarhiat. Liikumine ülevalt-alla on liikumine üldisemast rollist konkreetsema rollini. Kui on mingi sõlm, siis sellest sõlmest on võimalik esituse põhjal leida, missugused kasutajad sinna kuuluvad ja vastupidiselt õigustega.

Alustades mingisugusest konkreetsest sõlmest, milles sisaldub vähemalt üks kasutaja, saame alates sellest sõlmest diagrammil allapoole liikudes tuvastada sellele sõlmele kuuluvate õiguste hulga. Võtame näitena spordikeskuse tulemuste diagrammi alloleval joonisel, mille hulgast valides sõlme, kus on ainult üks silt ehk Raamatupidaja. Liikudes diagrammil allapoole, leiame esimese hierarhia kaks sellesse kontsepti kuuluvad õigused Asenduskonto ülekanne ja Tegevuste ajalugu. Edasiselt saame liikuda ainult nende kahe sõlme ühisesse punkti milleks on sõlm siltidega Makse ja Müügiaruanne. See tähendab, et esimesest sõlmest liikudes saame ühe kontsepti ehk rolli välja leida, milleks oleks {{Raamatupidaja} ; {Asenduskonto ülekanne, Tegevuste ajalugu, Makse, Müügiaruanne}}. Joonisel siniste sõlmedega ja joontega tähistatu illustreerib liikumist diagrammil. Teisipidi on algsest sõlmest liikumine ülesse, leides seosega kolm uut kasutajat, mis tekitab kasutajate hulga rollile {Raamatupidaja, Arendajad, Administraatorid, Juhataja}.

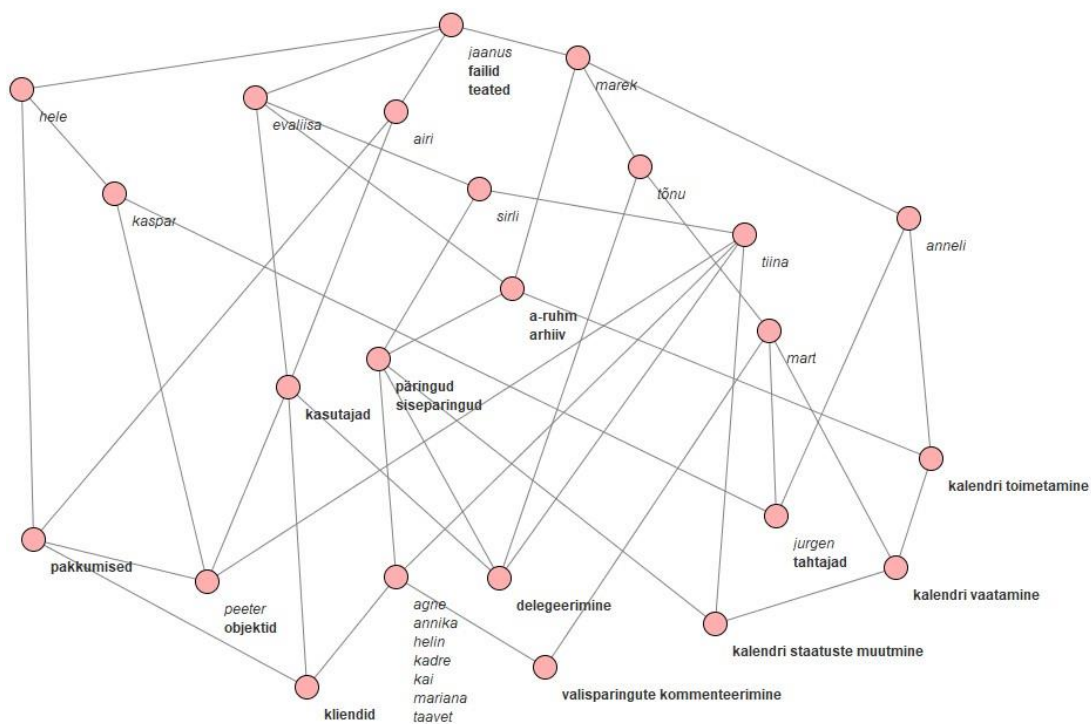


Joonis 10 Spordikeskuse GSH diagramm

Kandidaattrollid saavad tekkida, kui vaadata sõlme, milleks on ainult õigused nagu näiteks {Makse, Müügiaruanne}. Kuna rollide hulka ei ole väljastatud on diagrammilt võimalik genereerida väga palju kandidaattrollide kombinatsioone. Rollide määratlemise puhul on see suureks miinuseks, aga antud meetodi peamine eesmärk ei olegi minimaalsete rolli hulkade leidmine, vaid nende esitamine visuaalselt.



Joonis 11 CMS GSH diagramm



Joonis 12 Intranet GSH diagramm

Konkreetsete andmestike põhjal saab väita, et GSH diagrammid pakuvad kompaktsed visuaalseid esitisi binaarmatriksitest, kuid ei paku meile konkreetseid rollide hulksid,

millest saaks ekstraheerida tähtsad rollid. Seoste visuaalse esituse optimaalsust ja sobivust vaatleme hiljemalt võrdluses teiste algoritmidega, mis pakuvad rollihierarhia tulemusi.

6.3 Kontseptiahelate tulemused

Käesolevaga vaatame, missugused tulemused on andnud kontseptiahelate CL-sort meetod lõputöö andmestikel. Järgnevalt esitame kokkuvõtvad tulemused, vaatame tulemuste katvuse määrasid ja kontseptide arvu. Seejärel genereeritud kontsepte teksti kujul, visuaalselt ja analüüsimise tulemusi.

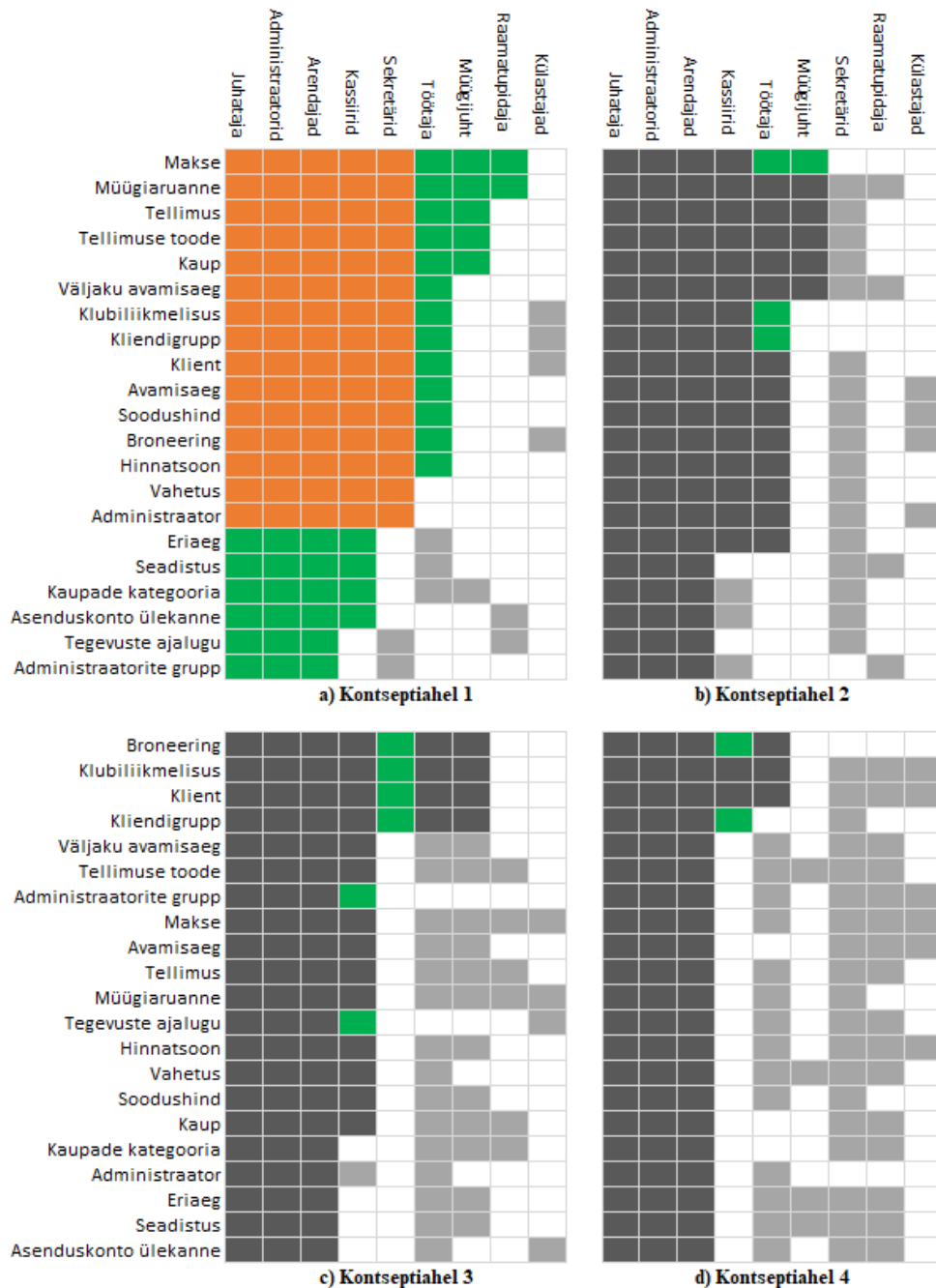
Alloleva tabeli tulemustest saame lihtsalt järeldada, et kontseptiahelad katavad küllaltki väikese kontseptide arvuga suurema osa andmestikust, kuid viimase katmata osa jaoks kasvab kontseptide arv väga kõrgeks. See tähendab, et esimeste ahelatega katmata kasutajaõiguseid on järgnevate ahelatega keeruline katta. Täieliku katvuse puhul jätkatakse kontseptiahelate leidmist osalise katvuse kontseptiahelatest.

Tabel 6 Kontseptiahelate tulemused osalisel ja täieliku katvusel

Tulemus	Spordikeskus	CMS	Intranet
Osalise katvuse määr	90,69%	87,13%	90,11%
Osalise katvuse kontseptide (ahelate) arv	6 (1)	11 (2)	18 (3)
Täieliku katvuse kontseptide (ahelate) arv	15 (4)	29 (8)	22 (4)

Iga genereeritud ahela järgi on võimalik kasutajate ja õiguste hulkadest järjestada vastavalt andmetabeli veerud ja read ning vajadusel tekitada visuaalne esitus kaetud andmestikust. Üks võimalik kontseptiahel on terve tabeli üks võimalik järjestus. Selleks oleme näitena välja toonud spordikeskuse kontseptiahelate katvuse alloleval joonisel, mis on saadud tabeli 7 tulemustest ja järjestatud selliselt, kus kasutajad on veergudena vasakult paremale järjestatud iga ahela viimasest kontseptist kuni esimeseni ning õigustega ülevalt alla vastupidiselt, esimesest kontseptist kuni viimaseni. Joonisel on oranži/rohelistega märgitud igas ahelas uued kaetud kasutajaõigused. Tumehalliga on ahelas kaetud kasutajaõigused, kuid uusi kasutajaõigusi mitte katvad ning helehalliga kasutajaõigused, mida ei kaetud leitud ahelas. Oranž värv tähistab ühe kontsepti loetavuse selgitust, mida vaadatakse hiljemalt tabeli põhjal.

Jooniselt on näha, et kontseptiahelatele on omane Pareto serieerimisega korrastatud andmetabel, mida vaatasime ka töö esimeses osas. Esimese ahela puhul on üritatud katta võimalikult palju kasutajaõigusi ja seejärel leitakse järgmised ahelad, katmaks ülejäänud kasutajaõigused. Esimese ahelaga kaetakse veidi enam kui 90% kasutajaõigustest ja ülejäänud kolme ahelaga ainult sealt puuduv osa.



Joonis 13 Spordikeskuse kontseptiahelate järjestuse visuaalne esitus

Allpool esitatud tabelis on Spordikeskuse andmestiku neli kontseptiahelat osalise ja täieliku katvuse puhul. Kuna tekkinud kontseptiahelate tulemused olid teistel andmestikel väga mahukad, oleme need välja toonud LISA 2. Ahelates sisalduvad kontseptid on kompaktsel kujul esitatud ehk igast kontseptist on eemaldatud mitmekordselt üle erinevate kontseptide sisalduvad kasutajad ja õigused. Kui algoritmi väljundis sisaldub ülevalt alla igas järgnevas kontseptis eelmises eksisteeriv õiguste alam-hulk, siis käesolevalt oleme kompaktse esitusviisi jaoks need õigused eemaldanud. Samasuguselt on vastupidisel liikumisel kasutajate kuvamised eemaldatud. Järgnevalt selgitame ühe näitega, kuidas sellise kompaktse esitusega andmetabelist aru saada.

Tabelis on oranži värviga tähistatud kontsepti nr.4 kuuluvad kasutajate ja õiguste hulk. Kontsepti kuuluvad kõik õigused, mis on alates neljandast kontseptist tabeli allapoole kuni kontseptini 6 ja õigused vastavalt ülesse poole kuni kontseptini 1. Näidet illustreerib eelnev joonis 13c kontseptiahel 1, kus oranžiga on tähistatud vastav kontsept.

Tabel 7 Kontseptiahela tulemused spordikeskuse andmestikul

K	Kasutajad	Õigused
1	Raamatupidaja	Makse, Müügiaruanne
2	Müügijuht	Tellimus, Tellimuse toode, Kaup
3	Töötaja	Väljaku avamisaeg, Klubiliikmelisus, Kliendigrupp, Klient, Avamisaeg, Soodushind, Broneering, Hinnatsoon
4	Sekretärid	Vahetus, Administraator
5	Kassiirid	Eriaeg, Seadistus, Kaupade kategooria, Asenduskonto ülekanne
6	Arendajad, Juhataja, Administraatorid	Tegevuste ajalugu, Administraatorite grupp
Kasutajaõiguste katvuse määr - 90,69%		
Täieliku katvuse leidmise jätkumine		
1	Müügijuht	Kaupade kategooria, Makse, Tellimus, Tellimuse toode, Kaup, Müügiaruanne
2	Kassiirid, Töötaja	Eriaeg, Seadistus, Väljaku avamisaeg, Klubiliikmelisus, Kliendigrupp, Klient, Avamisaeg, Soodushind, Broneering, Hinnatsoon

3	Arendajad, Juhataja, Administraatorid	Tegevuste ajalugu, Vaetus, Administraator, Administraatorite grupp, Asenduskonto ülekanne
Kasutajaõiguste katvuse määr – 93,79%		
1	Külastajad, Kassiirid, Töötaja	Broneering, Klubiliikmelisus, Klient, Kliendigrupp
2	Sekretärid	Väljaku avamisaeg, Tellimuse toode, Administraatorite grupp, Makse, Avamisaeg, Tellimus, Müügiaruanne, Tegevuste ajalugu, Hinnatsoon, Vaetus, Soodushind, Kaup
3	Arendajad, Juhataja, Administraatorid	Kaupade kategooria, Administraator, Eriaeg, Seadistus, Asenduskonto ülekanne
Kasutajaõiguste katvuse määr – 98,45%		
1	Sekretärid	Tegevuste ajalugu, Makse, Müügiaruanne
2	Raamatupidaja	Asenduskonto ülekanne
3	Arendajad, Juhataja, Administraatorid	Väljaku avamisaeg, Kliendigrupp, Tellimus, Tellimuse toode, Kaupade kategooria, Avamisaeg, Vaetus, Soodushind, Kaup, Klient, Administraator, Eriaeg, Seadistus, Administraatorite grupp, Klubiliikmelisus, Broneering, Hinnatsoon
Kasutajaõiguste katvuse määr – 100%		

Tulemused näitavad, et kui käsitleme kontseptiahelates sisalduvaid kontsepte kandidaattrollidena, siis on need väga suure arvu ja mitmekordse kasutajaõiguste katvusega. Tabelist ja joonisest avaldub, et ühte ja sama kontsepti on täieliku katvuse puhul välja toodud mitmeid kordi. Isegi kui korduvad rollid käsitsi eemaldame, saame tulemustena suurema rollide arvu, kui andmestiku kasutajate koguarv, mis ei taga arvestatavat rollide leidmist. Praeguste andmestike põhjal ei täida kontseptiahelad rollitehnika rollikaevandamise eesmärki kõige tulemuslikumalt meie jaoks.

6.4 QualityCover tulemused

Varasemalt vaadeldud meetodi uurimise näitest avaldus, et QualityCover algoritmi puhul leitakse ligilähedaselt optimaalseim tulemus, mis katab osaliselt või täielikult minimaalse kontseptide arvuga formaalse konteksti.

Vaatame järgnevalt algoritmi poolt osalise ja täieliku katvuse kontseptide tulemusi rollide vaatepunktist. Üritame leida analüüsi ja keerukuse hindamisega osalise ja täieliku katvuse tulemuste sobivuse.

Allpool asetseva tabeliga on esitatud kolme lähteandmestike pealt genereeritud tulemused osalisele ja täielikule katvusele. Täieliku katvuse puhul ei ole vajalik eraldi tulemusi välja tuua, sest algoritmi eduka töö korral on alati väärtus 1.

Tabel 8 QualityCover tulemused osalisel ja täielikule katvusel

Tulemus	Spordikeskus	CMS	Intranet
Osalise katvuse määr	84,50%	78,82%	74,72%
Osalise katvuse kontseptide arv	2	4	5
Täieliku katvuse kontseptide arv	6	12	13

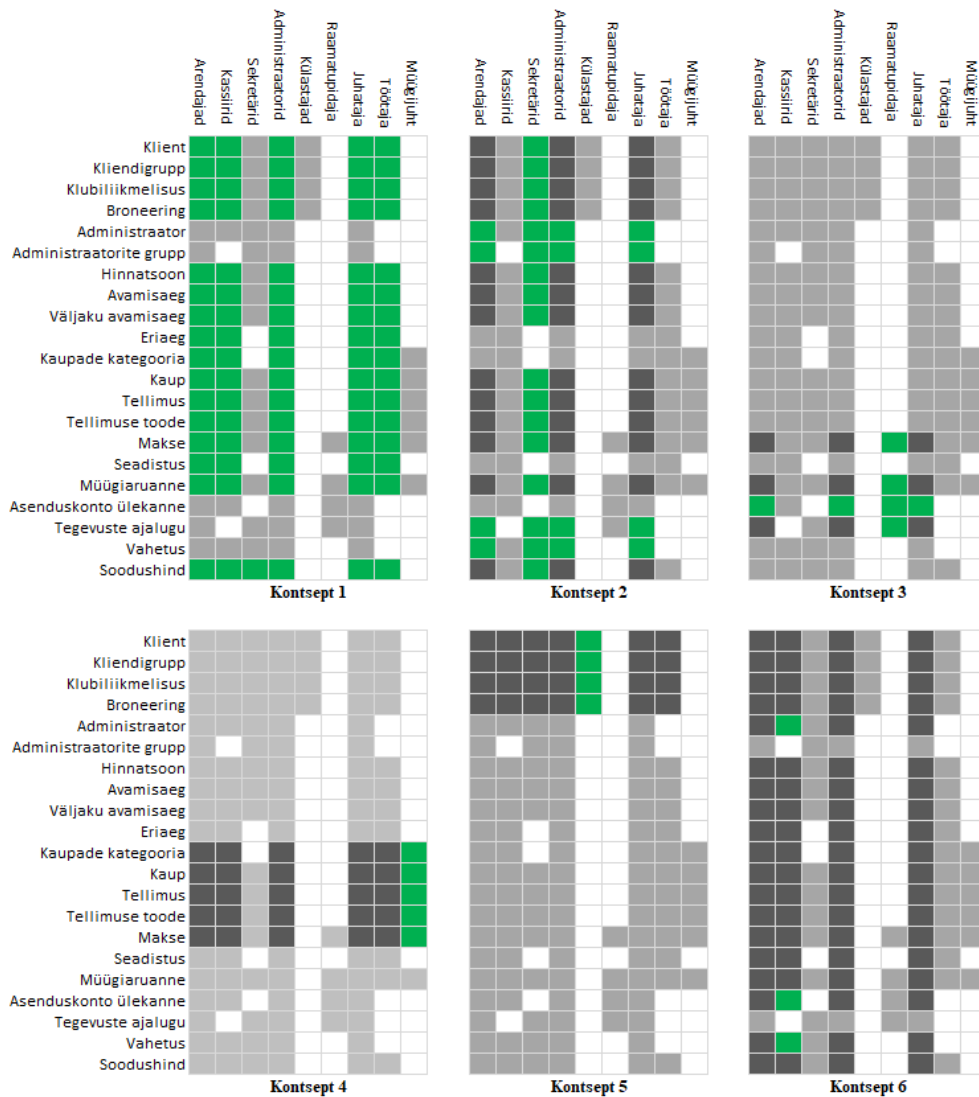
Tulemused avaldavad, et osalise katvuse puhul on kontseptide arv väga väike, sest eesmärk on katta minimaalsete kontseptide arvuga võimalikult palju kasutajaõigusi. Täieliku katvuse puhul on kontseptide ja katvuse suhtearvu määr peale igat osalise katvuse kontsepti oluliselt väiksem.

Üldiselt on kontseptide genereerimine QC algoritmiga andnud kõigi kolme andmestiku osalise katvuse puhul küllaltki väiksete kontseptide arvu. Ainult kolmanda andmestiku puhul on peaaegu viiendik kasutajaõigustest katmata kontseptide arvuga, mida näiteks esimese andmestiku puhul on leitud juba täielik kattega. See näitab, et kolmanda andmestiku puhul on seoste arvu arvestades tegemist keskmiselt keerukama andmestikuga algoritmide jaoks.

Kontseptid on järjestatud tulemustes uute kaetud kasutajaõiguste arvu põhjal suuremast väiksemani. Järjestikused kontseptid võivad katta elementide arvus olla identsed ning eelmise või eelmiste kontseptide ülekate ei ole määravaks faktoriks. Täieliku katvuse kontseptid jätkuvad sealt, kus osalise leidmisel algoritmi töö pooleli jäi ehk täielikus on esimesed kontseptid samad, mis osalises. Sealt jätkab algoritm tööd täieliku

kasutajaõiguste katmisele ning proovib leida paarid, mille puhul katmata seosed kontseptides oleks kaetud. Algoritm lõpetab oma töö ehk tulemuseks jääb osaline katvus, kui teatud lävendi väärtus hakkab suurenema. Lävendi väärtuse kättesaamiseks kasutatakse hinnafunktsiooni valemit. Väljundiks on selline saadud kontseptide jada, kus hinnafunktsiooni väärtus on minimaalne.

Vaatleme spordikeskuse täieliku katvuse andmestiku pealt täpsemalt, kuidas leitakse kaetud seoste (kasutajaõiguste) arv kontseptis. Toetamaks kirjeldust on joonisel välja toodud spordikeskuse algne binaartabel, kus igale kontseptile on esitatud tabel ning roheline on kuvatud selles kontseptis kaetud uued kasutajaõigused, tumehalliga kontseptis sisaldunud kasutajaõigused (mis ei olnud uued kaetud kasutajaõigused) ja halliga kontseptist välja jäänud konteksti kasutajaõigused.



Joonis 14 Spordikeskuse andmestiku katmised kontseptidega

Osalise katvuse esimese kontsepti puhul on katvus $80 = 16 \text{ õigust} \times 5 \text{ kasutajat}$. Tuleb meeles pidada, et parima suurima katmise leidmisel peavad üle mitme kasutaja ja vastupidiselt õiguse, olema kõik ühised seosed ehk igal 5 kasutajal on 16 õigust ja mitte vähem ega rohkem selles kontseptis. Teise kontsepti puhul on katvus $17 \times 4 = 68$, aga 39/68-st on juba esimese kontsepti poolt kaetud, seega uue katvuse tulemus on 29. Liikudes edasi täieliku katvuse tulemustele on kolmanda puhul uute seoste arv 7 ja lõpetades kuuendas kontseptis 3 viimase katmata seosega.

Käesolevate lähteandmestike puhul on küllaltki lihtne järeldusele jõuda, et valida tuleb täieliku katvuse variandid, sest rollide arvud on ei ole ilmatult suured ja rollikaevandamise jaoks ongi esmatähtis katta kõik kasutajaõiguste informatsioon.

Tulemusi vaadates võib pidada ahnele meetodile ja GAIN funktsiooni tuginevat minimaalse katvuse meetodit potentsiaalseks rollikaevandamise lahenduseks. Koondatakse suurimate õiguste hulkadega kasutajad, mis tähistavad sobilike rolle. Ahne meetodi puudujäägiks antud korral on see, et kasutajaõiguste katmine üle mitmete kasutajate ja õiguste on eelistatum, kui näiteks veidi väiksema kattega kasutajate või õiguste katmine täielikult. Spordikeskuse näitel ei ole QC algoritm rollina tekitanud {arendaja, juhataja, administraator} kasutajate hulka ühtse rollina, kuigi neil on ühised õigused ja tegemist oleks suure kaaluga rolliga.

Järgnevates tabelites on esitatud QC täieliku katvuse kontseptide ehk rollide tulemused, kus esimeses veerus on tähistatud kontsepti ehk rolli number (R), teises ja kolmandas vastavalt kasutajate ja õiguste hulgad ning neljandas uute kaetud kasutajaõiguste arv igas kontseptis (C).

Kasutajatele ühised õigused on tuletatavad LISA 1 tabelitest

Tabel 9 QualityCover spordikeskus kontseptide tulemused

R	Kasutajad	Õigused	C
1	Arendajad, Kassiirid, Administraatorid, Juhataja, Töötaja	16 erinevat kasutajatele ühist õigust	80
2	Arendajad, Sekretärid, Administraatorid, Juhataja,	17 erinevat kasutajatele ühist õigust	29
3	Arendajad, Administraatorid, Raamatupidaja, Juhataja	4 erinevat kasutajatele ühist õigust	7

4	Arendajad, Kassiirid, Administraatorid, Juhataja, Töötaja, Müügijuht	6 erinevat kasutajatele ühist õigust	6
5	Arendajad, Kassiirid, Sekretärid, Administraatorid, Külastajad, Juhataja, Töötaja	Klient, Kliendigrupp, Klubiliikmelisus, Broneering	4
6	Arendajad, Kassiirid, Administraatorid, Juhataja	19 erinevat kasutajatele ühist õigust	3

Tabel 10 QualityCover CMS kontseptide tulemused

R	Kasutajad	Õigused	C
1	artweb, kristiine, mihkel	59 erinevat kasutajatele ühist õigust	177
2	jaanus, artweb juliana, kristiine, mihkel	30 erinevat kasutajatele ühist õigust	60
3	jaanus, kadi, mariaD, artweb, juliana, kristiine, mihkel	15 erinevat kasutajatele ühist õigust	30
4	jaanus	58 erinevat õigust	27
5	jaanus, triinbox, ulla, artweb, kristiine, mihkel	8 erinevat kasutajatele ühist õigust	16
6	jaanus, artweb, hannaj, irina, kristiine, mihkel	8 erinevat kasutajatele ühist õigust	16
7	jaanus, paul, artweb, kristiine, mihkel	15 erinevat kasutajatele ühist õigust	15
8	jaanus, artweb, kert, kristiine, mihkel	11 erinevat kasutajatele ühist õigust	11
9	artweb, kaire, kristiine, mihkel	9 erinevat kasutajatele ühist õigust	9
10	ülle	9 erinevat kasutajatele ühist õigust	9
11	jaanus, artweb, kristiine, laura, mihkel	API statistics, API presets	2
12	artweb	60 erinevat õigust	1

Tabel 11 QualityCover Intranet kontseptide tulemused

R	Kasutajad	Õigused	C
1	annika, kadre, sirli, mariana, helin, marek, agne, tiina, kai, evaliisa, taavet, jaanus	kliendid, välispäringute kommenteerimine	24
2	evaliisa, jaanus	12 erinevat kasutajatele ühist õigust	24
3	marek, mart, tõnu, jaanus	tahtajad, välispäringute kommenteerimine, kalendri vaatamine, kalendri staatuste muutmine	12
4	marek, jaanus	10 erinevat kasutajatele ühist õigust	6
5	sirli, tiina, evaliisa, jaanus	5 erinevat kasutajatele ühist õigust	6
6	airi, jaanus	5 erinevat kasutajatele ühist õigust	6
7	hele, jaanus	tahtajad, objektid, kliendid, pakkumised	4
8	marek, anneli, jaanus	tahtajad, kalendri vaatamine, kalendri staatuste muutmine, kalendri toimetamine	4
9	kaspar, marek, anneli, hele, mart, jürgen, tõnu, jaanus	tähtajad	2
10	sirli, kaspar, peeter, tiina, hele, evaliisa, airi, jaanus	objektid	2
11	jaanus	16 erinevat kasutajatele ühist õigust	2
12	sirli, evaliisa, jaanus	7 erinevat kasutajatele ühist õigust	2
13	marek, tõnu, jaanus	5 erinevat kasutajatele ühist õigust	1

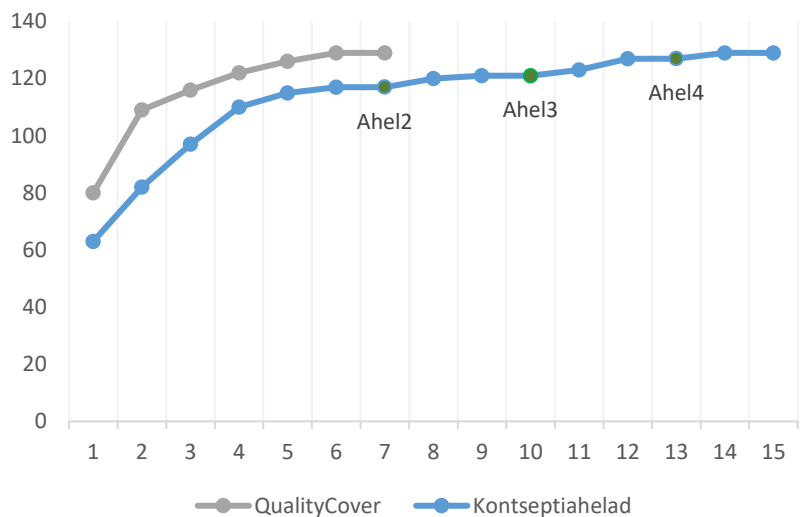
QualityCover meetodi tulemused annavad väga kompaktseid ja potentsiaalseid rollide tulemusi, kuid nõrgaks kohaks on rollide leidmine just õiguste või kasutajate hulkade seisukohast, sest üritatakse katta puhtalt kasutajaõigusi. See tähendab, et meetod vajab rollikaevandamiseks laiendamist meetodi täiendamisega või pärast tulemusi käsitsi kasutajate või õiguste hulkade kärpimist, et katta ühel või teise rolli puhul täielikult kasutajaid või õigusi.

6.4.1 Tulemuste võrdlus kontseptiahelatega

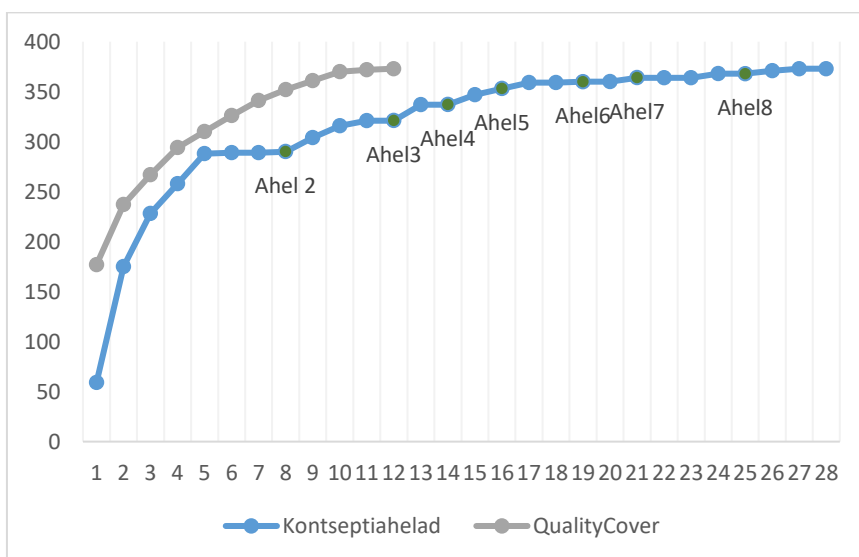
QualityCover ja kontseptiahelate meetod toetuvad mõlemad kontseptidele ja algoritmide tulemustes sisalduvaid kontsepte saab käsitleda rollidena. Kontseptiahelate eesmärgiks ei ole andmetabelite täielik katmine võrreldes QualityCover algoritmiga, mistõttu oli vaja meetodite tulemuste võrdluse jaoks laiendada kontseptiahelate leidmist. Eesmärkidest hoolimata saame kahe meetodi puhul võrrelda rollide arvulisi tulemusi ja osalist katmist.

Kontseptiahela eesmärgiks on säilitada kontseptivõre struktuuraalsus, mis tähendab, et objektidel on järjestuses sarnasused ja iga kontsepti puhul lisandub uus kasutaja kasutajate hulka. Sealt tulenevalt tekib oluliselt paremini välja just nimetatud rollide tasemed. Tekib rollihierarhia, alustades baas-õigusest, mis kehtivad kõikidele järgnevatele rollidele ehk nendes sisalduvatele kasutajatele. Antud rollide infot saab kasutada rollihierarhia ülesehitamiseks ja võrdluseks.

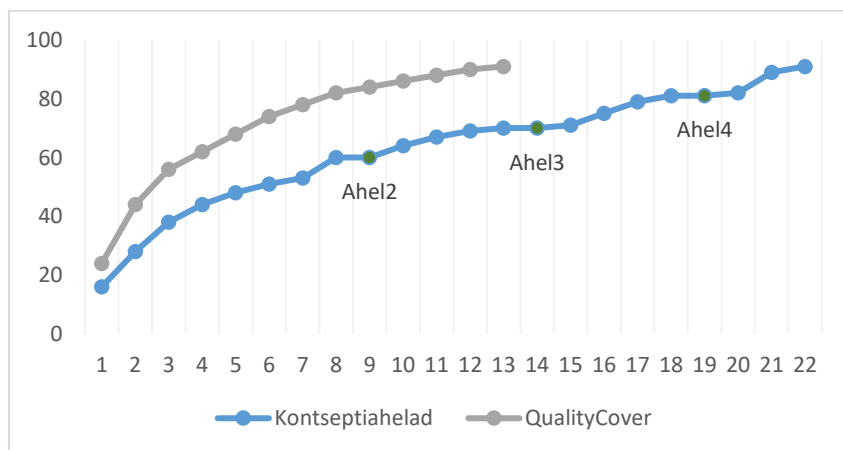
Allpool on esitatud kolm joonist kasutajaõiguste katmisele üle kõikide kontseptide andmestikes. Diagrammidel on x-teljel tähistatud kasutajaõiguste arv ja y-teljel rollide arv. Kontseptiahelate diagrammi joontele on lisatud ahelate punktid tähistamaks, mis kontseptidel algab uus ahel. QC joone lõpp tähistab antud algoritmi tulemuste leidmist täieliku katvusega. Joonistelt näeme, et esimese kahe andmestiku puhul kaetakse suurem osa kasutajaõigustest esimeste kontseptidega ja seejärel täieliku katvuse leidmine kulgeb väga vaevaliselt üle paljude kontseptide. Ainult kolmanda andmestiku puhul on graafik lineaarsem mõlema meetodi puhul. Selgelt on näha, et QC pakub oluliselt väiksemate kontseptide arvuga andmestiku katmist ning uute kasutajaõiguste katmine on palju ühtlasema jätkuga. Kontseptiahela meetodi kontseptide arv on kolmandiku või isegi teatud juhtudel poole võrra suurem. Peaaegu kõikidel andmestikel on QualityCover leidnud täieliku katvuse kontseptiahelate osalise katvuse kontsepti arvul.



Joonis 15 Spordikeskus katvus üle kontseptide



Joonis 16 CMS katvus üle kontseptide



Joonis 17 Intranet katvus üle kontseptide

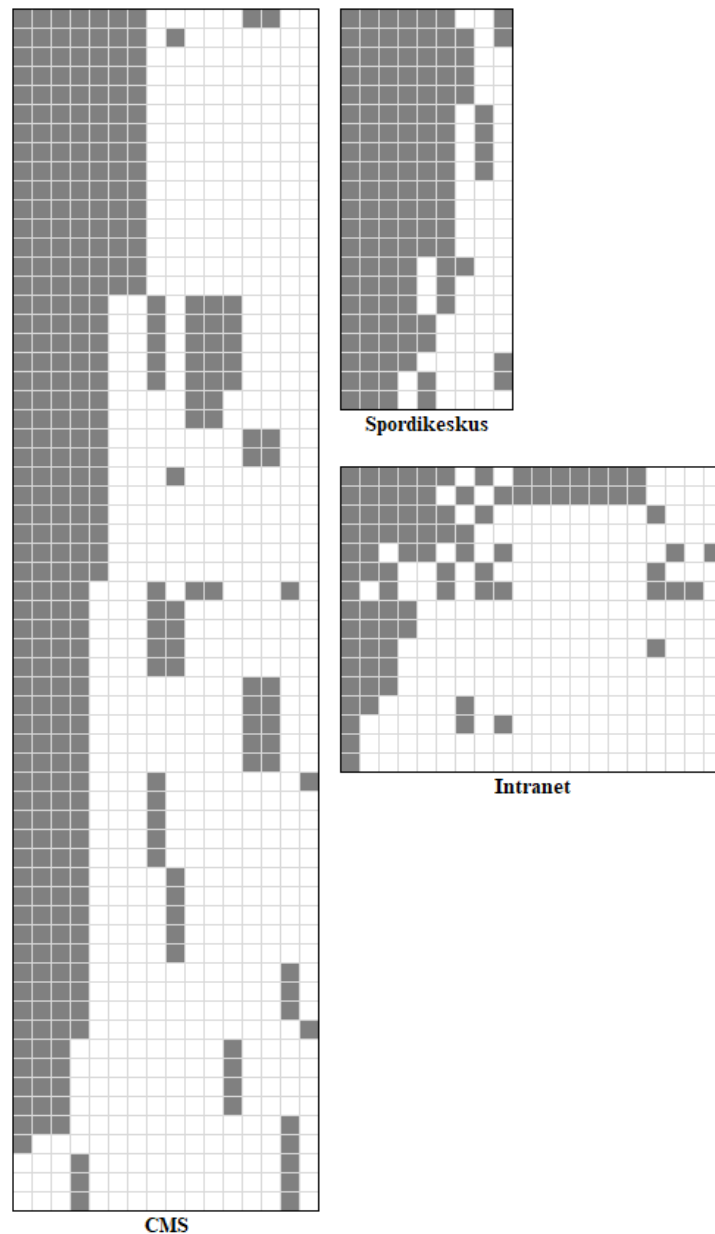
Kontseptiahelates on kontseptivõre struktuuraalsuse säilitamisel kontseptid ahelas seotud eelnevate ja järgmiste kontseptidega. Nagu peatükis 6.4 kontseptiahelate tulemustes nägime on kasutajaõigused üle mitmete ahelate paratamatult korduvad, et katta andmestik täielikult. Seetõttu ei ole täielikult korrektne võrrelda kontseptiahelate kontsepte QC kontseptide arvuga. Teisipidi saaksime omistada ühe ahela kontseptina ehk rollina kuna tegemist on ühe järjestusega. See tähendaks jällegi, et ühe rolli alla koonduks väga suured kasutajate ja õiguste hulgad, mis ei ole soovitud rollikaevandamise tulemus.

Kõigest sellest saame järeldada, et QC tulemused langevad paremini kokku klassikaliste rollikaeve meetodite tulemustega ja kontseptiahelate kasutamine rollikaevandamises vajaks meetodi laiendamist või rolli mõiste ümberdefineerimist. Näiteks rolliahelatele lisaks tekitatakse peamised rollid, millel on erinevad tasemed ja läbi selle tuletatakse tähtsad rollid. Praegusel juhul pakuvad kontseptiahelad ainult teoreetilisel tasemel rollikaevandamist, aga mitte huvi pakkuvaid tulemusi.

6.5 Miinustehnika tulemused

Käesolevas peatükis analüüsime miinustehnikaga genereeritud andmestike tulemusi. Meetodis on ainult arvestatud ühtesid ehk kaaluna ainult need kasutajaõigused, kus on ühed ning puudumisi (nullisagedusi) arvesse ei ole võetud ehk tulemuste leidmisel on vajalikuks osaks andmestikus õigused ning mitte õiguste puudumised. Nagu teoorias välja tooduna on andmestik mõlemat dimensiooni pidi sorteeritud. Üldjuhul on esialgsed binaarsed andmetabelid väga segased, kuid ridade ja veergude ümberjärjestamine miinustehnikaga annab välja korrastatud ja edasiselt analüüsitava andmetabeli struktuuri.

Alloleva joonise tulemustest näeme, et suurem osa andmestikust on ühe kolmnurkse ehk Pareto serieerimise struktuuriga esitatud. Üle kõikide andmestike on järjestatud ülevalt alla igale veerule kõige rohkemate kasutajaõiguste seosega katkematu ühtede järjestus. Need tulemused annab ridade järjestamine kaaludega ja vasakult-paremale võimalikult suur katkematu jada veergude ümberjärjestamisega. Tulemustest näeme, et suurem katkematu jada on veergudel, mida teostatakse algoritmi viimase osana, kui juba read on ümber järjestatud. Miinustehnika väljundiks on andmetabeli fail sarnaselt sisendina, aga ümberjärjestatud on read ja veerud, mis tekitavad parimal võimalikul viisil katkematu ühtede jada.



Joonis 18 Miinustehnika tulemused

Allolevas tabelis on esitatud miinustehnika algoritmi tööajad. Ümberjärjestatud tabelite tulemused on leitud väga kiire ajaga ning väga suurte andmestike puhul ei tohiks antud tulemuste põhjal miinustehnika algoritmile keerukaks osutada. See tähendab, et rollikaevandamise valdkonnas, kus juurdepääsuõiguste arv võib olla väga suured, suudab miinustehnika need arvestava ajaga leida.

Tabel 12 Miinustehnika algoritmi tööajad

Parameeter	Spordikeskus	CMS	Intranet
Tööaeg	5.59ms	10.71ms	4.21ms

Kuna miinustehnika eesmärgiks ei ole väljastada sarnaselt FCA-le kontseptide tulemusi, siis ei ole praktiline välja tuua visuaalsest tulemusest rollide hulka kasutajate või õiguste põhisel, mida saaksime täpselt samasuguselt algele andmestikule käsitsi leida. Sellisel juhul ei oleks tegemist rollikaevandamisega ega ühegi meetodi rakendamisega, sest saaksime iga kasutaja defineerida ühe rollina. Nagu eelnevast joonisest näeme, katab miinustehnika suurema osa andmestikest struktuuriga, kuid kahjuks ei saa käsitleda neid tulemusi otseselt rollide hulkade määramisel.

Eelnevatest tulemustest saame järeldada seda, et miinustehnika ei sobi iseenesest rollide hulkade leidmiseks, vaid saame pidada meetodit alternatiivselt visuaalseks rollide esituse viisiks, kus edasiselt saaks laiendada rollide hulkade leidmist.

6.6 Graafi optimeerimise tulemused

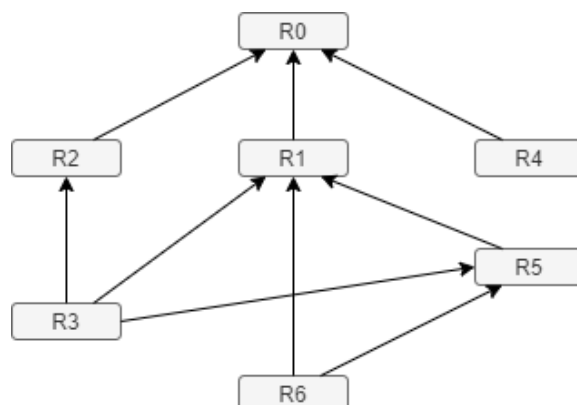
Selles peatükis vaatleme lähemalt GO tulemusi ja uurime neid toetudes teoorias tutvustatud algoritmi tööpõhimõttele ja optimeerimise olukordadele. Järgnevas tabelis on välja toodud optimeerimise tulemused ehk rollide arv ja teiste parameetrite tulemused, mis näitavad lõpliku RBAC seisundit ehk keerukust. Optimeerimise naturaalne eesmärk on vähendada rollide ja servade arvu, mistõttu need tulemused on avaldatud. Algoritmi parameetrite sisendiks oli kaalutud struktuurse keerukuse minimeerimine: $(w_d) = (w_h) = (w_p) = (w_u) = (w_r) = 1$

Tabel 13 Graafi optimeerimise rollikaevandamise tulemused

Parameeter	Spordikeskus	CMS	Intranet
Rollide arv	7	13	14
UA	33	56	58
PA	84	288	79
RH	9	18	22
WSC	133	375	173

Järgnevalt on esitatud spordikeskuse rollihierarhia struktuur, mis esitab rolle ja nendevahelisi seoseid. Tegemist on lihtsama rollihierarhia joonisega, mille detailsemat tulemit vaatame võrdleva analüüsi peatüki käigus. Praegusest näitest saame välja tuua ainult õiguste alam-hulkade seosed rollide vahel ning täiendavaks struktuuriks on vaja joonisele lisada kasutajad, õigused ja mainitud operatsioonid rollide vahel. Vaadates andmetabelit, saame öelda, et olukordasid rollide vahel tekib küllaltki palju, sest iga

rollide omavaheline seos on vähemalt üks operatsioon ning lõplik rollihierarhia joonis võib väga keeruliseks osutuda.



Joonis 19 GO rollihierarhia spordikeskuse andmestikul

Tulemuste leidmiseks RMiner rakenduse abil vajasis muudatust andmestikud, kus eemaldada tuli samade õiguste hulkadega ehk duplikaat kasutajad. Leidmata põhjusel tekkis lõpmatu programmi tööaeg. Sellest olenemata ei muutu teiste algoritmidega võrreldes tulemused, sest sisuliselt on iga meetodi eesmärk sarnaste õiguste hulkadega kasutajate koondamine, mida praeguse algoritmi jaoks teostasime käsitsi koheselt algandmetes. Allolevas tabelites on esitatud tulemused, kus on kirjas rolli number (R), rolli kasutajate ja õiguste hulgad ja ülemise sõlme (ingl k *parent node*) ehk rolli number (RH). Lõplikest rollide tulemustest on näha, et graafi optimeerimise meetod alustab RBAC seisundist, milleks on õiguste hulk igale kasutajale ja seejärel alustatakse ühendamise ja tükeldamise meetoditega selle optimeerimist.

Kasutajatele ühised õigused on tuletatavad LISA 1 tabelitest

Tabel 14 Graafi optimeerimise spordikeskuse tulemused

R	Kasutajad	Õigused	RH
1	Arendajad_Administraatorid_ Juhataja	20 erinevat kasutajatele ühist õigust	-
2	Arendajad_Administraatorid_ Juhataja, Kassiirid	19 erinevat kasutajatele ühist õigust	0
3	Arendajad_Administraatorid_ Juhataja, Sekretärid	17 erinevat kasutajatele ühist õigust	0

4	Arendajad_Administraatorid_ Juhataja, Kassiirid, Sekretärid, Külastajad, Töötaja	klient, kliendigrupp, klubiliikmelisus, broneering	5
5	Arendajad_Administraatorid_ Juhataja, Raamatupidaja	makse, müügiaruanne, asenduskonto ülekanne, tegevuste ajalugu	0
6	Arendajad_Administraatorid_ Juhataja, Kassiirid, Töötaja	15 erinevat kasutajatele ühist õigust	1
7	Arendajad_Administraatorid_ Juhataja, Kassiirid, Töötaja, Müügijuht	5 erinevat kasutajatele ühist õigust	5

Tabel 15 Graafi optimeerimise CMS tulemused

R	Kasutajad	Õigused	RH
1	jaanus	57 erinevat kasutajatele ühist õigust	-
2	jaanus, kadi_mariaD, artweb, juliana, kristiine, mihkel	15 erinevat kasutajatele ühist õigust	9
3	jaanus, paul, artweb, kristiine, mihkel	15 erinevat kasutajatele ühist õigust	9
4	jaanus, triinbox_ulla, artweb, kristiine, mihkel	8 erinevat kasutajatele ühist õigust	9
5	artweb	60 erinevat kasutajatele ühist õigust	-
6	jaanus, artweb, hannaj, irina, kristiine, mihkel	8 erinevat kasutajatele ühist õigust	9
7	jaanus, artweb, juliana, kristiine, mihkel	30 erinevat kasutajatele ühist õigust	
8	artweb, kaire, kristiine, mihkel	9 erinevat kasutajatele ühist õigust	9
9	jaanus, artweb, kert, kristiine, mihkel	11 erinevat kasutajatele ühist õigust	9
10	artweb, kristiine, mihkel	58 erinevat kasutajatele ühist õigust	4
11	jaanus, artweb, kristiine, laura, mihkel	API statistics, API presets	9
12	ylle	9 erinevat kasutajatele ühist õigust	-

13	jaanus, paul, artweb, hannaj, irina, kristiine, mihkel	6 erinevat kasutajatele ühist õigust	2,5
----	---	--------------------------------------	-----

Tabel 16 Graafi optimeerimise Intranet tulemused

R	Kasutajad	Õigused	RH
1	annika_kadre_mariana_ helin_agne_kai_taavet, sirli, marek, tiina, evaliisa, jaanus	kliendid, välispäringute kommenteerimine	13
2	sirli, evaliisa, jaanus	7 erinevat kasutajatele ühist õigust	13
3	kaspar, hele, jaanus	tähtajad, objektid	13
4	sirli, kaspar, peeter, tiina, hele, evaliisa, airi, jaanus	objektid	13
5	marek, jaanus	11 erinevat kasutajatele ühist õigust	13
6	marek, anneli, jaanus	tahtajad, kalendri vaatamine, kalendri staatuste muutmine, kalendri toimetamine	13
7	sirli, tiina, evaliisa, jaanus	5 erinevat kasutajatele ühist õigust	13
8	hele, jaanus	tähtajad, objektid, kliendid, pakkumised	13
9	marek, mart, tõnu, jaanus	tahtajad, välispäringute kommenteerimine, kalendri vaatamine, kalendri staatuste muutmine	13
10	kaspar, marek, anneli, hele, mart, jürgen, tõnu, jaanus	tahtajad	13
11	evaliisa, jaanus	12 erinevat kasutajatele ühist õigust	13
12	airi, jaanus	5 erinevat kasutajatele ühist õigust	13
13	marek, tõnu, jaanus	5 erinevat kasutajatele ühist õigust	13
14	jaanus	16 erinevat kasutajatele ühist õigust	-

6.7 HierarchicalMiner tulemused

HierarchicalMiner tulemused RMiner rakenduse abil olid genereeritud kahe erineva parameetrite kombinatsiooniga. Täpselt nagu lahenduse tutvustuse peatükis 4.2 välja

tooduna, kasutame selleks kahte seadistust, mis esmalt minimeerib kaalutud struktuurse keerukuse (*minwsc*) ja teisalt rollide arvu (*minR*):

$$1. \text{ minwsc: } (w_d) = (w_h) = (w_p) = (w_u) = (w_r) = 1$$

$$2. \text{ minR: } w_r = 1 ; w_u = w_p = 0 ; w_h = w_d = \infty$$

Mõlema parameetrite seadistusega on kolm andmestiku rakendusega läbi töötatud, milleks oli kokku 6 erinevat rollide tulemust. Allolevas tabelis on esitatud kokkuvõtvad tulemused. Tabelis on kirjas mitmed lühenditega parameetrid, millest UA tähendab kasutaja-rolli seost (ingl k *user to role assignment*), PA õiguse-rolli seost (ingl k *permission to role assignment*) ja RH rollihierarhiat (ingl k *role hierarchy*).

Rakendus väljastab lisaks rollidele servade arvud, rollihierarhia väärtuse ning kaalutud struktuurse keerukuse väärtuse. Servad on kokku kasutaja-rolli seosed ja õiguse-rolli seosed, rollihierarhia on rollide omavaheline seoste koguarv ning kaalutud struktuurne keerukus eelmiste väärtuste kogusumma. Näiteks vastavalt spordikeskuse esimese seadistusele ($10 + 12 + 25 + 11 = 58$). Tabelist on näha, et rollide minimeerimise seadistusega suureneb servade arv kuna rollides sisaldunud kasutajate ja õiguste arv on kõrgem. Vähemate rollide arvuga üritatakse terve andmestik ära katta. Sealt tulenevalt on lõplik keerukus (WSC) suurem. Rollihierarhia number on vähenenud, sest rollide arv on jäänud väiksemaks ja sealt tulenevalt seosed rollide vahel.

Tabel 17 HierarchicalMiner rollikaevandamise tulemused

Parameeter	Spordikeskus		CMS		Intranet	
	minwsc	minR	minwsc	minR	minwsc	minR
Rollide arv	10	7	15	12	13	13
UA	12	24	27	45	33	51
PA	25	34	72	88	24	16
RH	11	5	18	6	17	7
WSC	58	70	132	151	87	87

Üllatuslikul kombel ei ole aga Intranet tulemuste keerukuse väärtus muutunud arvestades, et tulemuste väärtused on erinevad. Lisaks on näha, et rollide minimeerimise seadistus tõesti suudab vähendada rollide arvu isegi väga väikeste andmestike puhul.

Kahe parameetri tulemuste leidmise korral on vajalik lisaks hinnata algoritmide tööaegasid, et ei tekiks juhust, kus ühe seadistuse tööajad on teisest märkimisväärselt

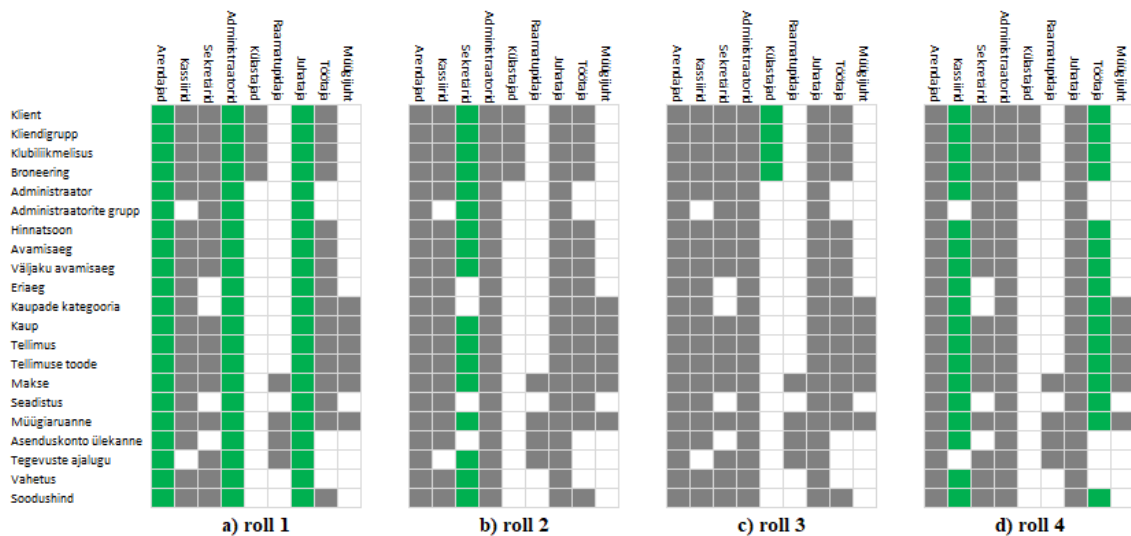
paremad. Tööaegadest näeme, et minimaalse rollide arvu leidmise puhul on algoritmi töö kiirem, sest teisel juhul üritatakse leida kõige optimaalsem RBAC seisund ning proovitakse lahendada eelnevalt välja toodud hulga katte probleem, mis on NP-keerukas [20]. Täpsema selgituse kohaselt toimub pärast kontseptivõrest leitud kandidaatrollide alam-hulkade valimi kinnitamist rollide ja õiguste määratlemine kasutajatele.

Tabel 18 HierarchicalMiner algoritmi tööajad

Seadistus	Spordikeskus	CMS	Intranet
Minimaalne kaalutud struktuurne keerukus	100.0	123.0	89.0
Minimaalne rollide arv	75.0	93.0	80.0

Praeguste andmestike rollikaevandamisel ei ole tulemuste tööaegade vahel suured erinevused, aga väga suurte andmestike puhul tuleb arvestada *minwsc* tulemuste leidmise suurema ajakuluga.

Allpool on esitatud HM algoritmi rollikaevandamise tulemused kolme organisatsiooni andmestikel kaalutud struktuurse keerukuse minimeerimise seadistustele. Kui HM algoritm suudab minimeerida RBAC seisundi kahe vaadeldud keerukuse mõõdiku järgi, siis praegusel juhul arvestame *minwsc* tulemusi, sest sama seadistuse ja mõõdiku järgi leitakse ka teiste traditsiooniliste meetodite tulemused. Minimaalse rollide arvu seadistuste tulemused on välja toodud LISA 3. Tulemustes on välja toodud, missugune kasutaja või õigus selles rollis on kõige suuremas või täielikus osas andmestikus kaetud. Spordikeskuse rollide visuaalse esituse tulemuste joonisel on esimeses rollis on kaetud kasutajad, kes omavad kõiki õiguseid andmestikus, aga puuduvad õigused, mida kaetaks täielikult. Järgmises rollis on kaetud Sekretärid kasutaja, mille tulemusena on lõpuks kaetud Administraatorite grupp õigus. Nagu näeme, sisaldub rollis oluliselt rohkem kasutajaõiguste seoseid, kui rolli sisu meile avaldab.



Joonis 20 HM nelja rolli visuaalne esitus spordikeskuse näitel

HM alustab optimeerimist sellisest RBAC seisundist, kus esialgu on iga kasutaja ja õigus määratud ainult ühte rolli. Vähendatud kontseptivõre heuristilise kärpimisega saadakse tulemused, kus see enam paika ei pea. Seda näeme ka allolevatest tulemustest, kus on kirjas rolli number (R) ja iga rolli kasutajate ja õiguste hulgad.

Tabel 19 HM tulemused minimaalse keerukuse seadistusega Spordikeskus andmestikul

R	Kasutajad	Õigused
1	Arendajad, Administraatorid, Juhataja	-
2	Sekretärid	Administraatorite grupp
3	Külastajad	Klient, Kliendigrupp, Klubiikmelisus, Broneering
4	Kassiid, Tootaja	Eriaeg, Seadistus
5	Muugijuht	Kaupade kategoria
6	Kassiid	Administraator, Vahetus
7	-	Hinnatsoon, Avamisaeg, Vajaku avamisaeg, Soodushind
8	-	Kaup, Tellimus, Tellimuse toode, Makse, Muegiaruanne
9	Kassiid, Raamatupidaja	Makse, Muegiaruanne, Asenduskonto uelekanne

10	Raamatupidaja	Makse, Müügiaruanne, Tegevuste ajalugu
----	---------------	--

Alloleva tabeli kasutajatele ühised õigused on tuletatavad LISA 1 tabelitest.

Tabel 20 HM tulemused minimaalse keerukuse seadistusega CMS andmestikul

R	Kasutajad	Õigused
1	kadi, mariaD	15 erinevat kasutajatele ühist õigust
2	jaanus, paul	5 erinevat kasutajatele ühist õigust
3	jaanus, triinbox, ulla	8 erinevat kasutajatele ühist õigust
4	jaanus, juliana	10 erinevat kasutajatele ühist õigust
5	kaire	widget generator, seatings plan, crontab generator, seating plan generator
6	jaanus, kert	7 erinevat kasutajatele ühist õigust
7	artweb, kristiine, mihkel	webgate controller
8	jaanus, laura	API statistics, API presets
9	-	cookie policy, no cookie policy, site variables
10	jaanus, ylle	settings, cms users, usergroups
11	artweb, ylle	environments, webgate controller
12	jaanus, hannaj, irina	product catalog, product groups
13	-	shows, concerts, venues, promoters, product group links
14	-	banners system, bannerscategory, banner statistics, banner
15	jaanus, hannaj, irina	XML import

Tabel 21 HM tulemused minimaalse keerukuse seadistusega Intranet andmestikul

R	Kasutajad	Õigused
1	annika, kadre, mariana, helin, marek, agne, kai, taavet	kliendid
2	kaspar, peeter	objektid
3	sirli, tiina, evaliisa	delegeerimine, kalendri staatuste muutmine
4	kaspar, marek, anneli, hele, mart, jurgen	tahtajad

5	marek, tõnu	delegeerimine
6	jaanus	failid, teated
7	evaliisa, airi	delegeerimine, kliendid, kasutajad
8	hele, airi	kliendid, pakkumised
9	mart	välispäringute kommenteerimine
10	mart	kalendri vaatamine, kalendri staatuste muutmine
11	marek, anneli	kalendri toimetamine
12	sirli	delegeerimine, kliendid, kalendri staatuste muutmine, päringud, sisepäringud
13	marek, evaliisa	a-rühm, arhiiv

6.8 CompleteMiner ja FastMiner tulemused

Käesolevaga vaatame alamhulkade loendamise algoritmide tulemusi üle ühe parameetrite seadistuse. Proovime leida tulemuste põhjal teoreetilises osas välja toodud CM keerukuse kitsaskoha ja FM tulemuste sobivuse andmestike rollikaevandamisega.

Allolevas tabelis on rollikaevandamise tulemused, kus rollide arvu reas on esimese numbrina kirjas kasutajaid sisaldavad rollid, mis katavad andmestiku minimaalse rollide arvuga ja sulgudes kõik algoritmi poolt leitud potentsiaalsed rollid. Nagu näeme, leiab FM algoritm suuremate ja keerukamate andmestike puhul vähem kandidaatrolle. Minimaalsete rollide arvu tulemuste puhul on rollide hulk väga sarnane kõikide teiste eelnevalt vaadatud meetodite tulemustele. Tabelis esitatud parameetrite lühendite tähendused on kirjeldatud eelmises peatükis.

Tabel 22 CM ja FM algoritmide tulemused lähteandmestikel

Parameeter	Spordikeskus		CMS		Intranet	
	CM	FM	CM	FM	CM	FM
Rollide arv	7 (13)	7 (13)	27 (12)	26 (14)	34 (14)	33 (14)
UA	9	9	16	21	20	20
PA	21	21	63	72	16	16
RH	53	53	150	129	420	396
WSC	96	96	256	248	490	465

Kaalitud struktuurse keerukuse tulemustest saab järeldada, et rakendatud näiteandmestikel töötavad algoritmid väga sarnaselt. Struktuurne keerukus on osaliselt väiksem FM puhul ja alles suuremates andmestikes on väljendunud algoritmide erinevus.

Algoritmide statistilistele tulemustele on allpool väljastatud rollid koos kasutajate ja õiguste hulkadega. Iga andmestiku ja mõlema algoritmi puhul oli programmile antud ette minimaalse keerukuse seadistuste sisend: $(w_d) = (w_h) = (w_p) = (w_u) = (w_r) = 1$

Nagu juba eelnevalt esitatud tulemuste tabelist oli märgata, siis mõlemad algoritmid on tekitanud täielikult identsed tulemused ühele andmestikule, mistõttu esitame spordikeskuse andmestikul rollide väljavõtte ühekordselt. Samasuguselt tegutseme kolmanda Intranet andmestikuga, kus esitame tulemused üheselt, sest CM algoritm on leidnud ainult ühe lisa kandidaatrolli, millel täpsemalt välja toodud kasutajaid ega õigusi ei ole. Sellisel kujul kasutajate ja õiguste hulkadega rollid on esialgsete rollide lõikepunktidest leitud kandidaatrollid, mis ei täida täielikult ühegi kasutaja kõiki õigusi ega vastupidiselt õiguste kõiki kasutajaid. Algoritm on rollide hulgas esialgselt selliselt väljastanud ning praegusel juhul ei ole selliste rollide tulemused vajalikud ja vaatleme ainult rolle, millel on kasutajate hulk. Sellisel kujul väljastatud tulemusi saab pidada meetodi nõrkuseks, sest täielikult tühjade hulkadega rollid ei oma rollikaevandamisel absoluutselt mingisugust infot. Seepärast esitame rollihulgas ainult kasutajahulkadega rollid.

Tabel 23 CM ja FM tulemused spordikeskuse andmestikul

R	Kasutajad	Õigused	Kaalud
1	arendajad, administraatorid, juhataja	-	1.5
2	kassiirid	-	3.5
3	sekretärid	administraatorite grupp	3.5
4	raamatupidaja	-	3.5
5	töötaja	eriaeg, seadistus	4.5
6	müügijuht	kaupade kategooria	5.5
7	külastajad	klient, kliendigrupp, klubi liikmelisus, broneering	6.5

Kasutajatele ühised õigused on tuletatavad LISA 1 tabelitest

Tabel 24 CM tulemused CMS andmestikul

R	Kasutajad	Õigused	Kaalud
1	jaanus	-	0.5
2	artweb	-	0.5
3	ylle	-	0.5
4	kristiine, mihkel	-	2.0
5	kaire	widget generator, seatings plan, crontab generator, seating plan generator	3.5
6	laura	API presets	4.5
7	kert	salespoints, countries, country, city, region	4.5
8	paul	url redirects, search results, statistics, external search	4.5
9	juliana	5 erinevat kasutajatele ühist õigust	4.5
10	triinbox, ulla	5 erinevat kasutajatele ühist õigust	5.0
11	hannaj, irina	-	5.0
12	kadi, mariaD	13 erinevat kasutajatele ühist õigust	6.0

Tabel 25 FM tulemused CMS andmestikul

R	Kasutajad	Õigused	Kaalud
1	jaanus	-	0.5
2	ylle	settings, cms users, usergroups	0.5
3	artweb	-	2.0
4	kristiine, mihkel	-	2.0
5	ylle	6 erinevat kasutajatele ühist õigust	2.0
6	kaire	widget generator, seatings plan, crontab generator, seating plan generator	3.5
7	jaanus, artweb, kristiine, mihkel	cookie policy, no cookie policy, site variables	4.0
8	laura	API presets	4.5
9	kert	salespoints, countries, country, city, region	4.5
10	paul	url redirects, search results, statistics, external search	4.5
11	juliana	5 erinevat kasutajatele ühist õigust	4.5

12	triinbox, ulla	5 erinevat kasutajatele ühist õigust	5.0
13	hannaj, irina	-	5.0
14	kadi, mariaD	13 erinevat kasutajatele ühist õigust	6.0

Tabel 26 CM ja FM tulemused Intranet andmestikul

R	Kasutajad	Õigused	Kaalud
1	jaanus	failid, teated	0.5
2	marek	-	1.5
3	evaliisa	-	1.5
4	hele	-	1.5
5	airi	-	1.5
6	tõnu	-	2.5
7	kaspar	-	2.5
8	anneli	-	2.5
9	sirli	-	2.5
10	tiina	-	3.5
11	mart	-	3.5
12	peeter	objektid	7.5
13	jurgen	tahtajad	7.5
14	annika, kadre, mariana, helin, agne, kai, taavet	-	8.5

Andmestikust CMS avaldub CM algoritmi soorituslikult parem rollihulga genereerimine suuremate andmestike puhul. FM algoritmi rollide arvu tulemus on isegi suurem kui unikaalsete õigustega kasutajahulkade arv andmestikus. Seetõttu on tõendatav teoreetilises osas välja toodud FM algoritmi puudus kandidaatorollide alam-hulkade identifitseerimisel. Käesolevate lihtsate andmestike puhul CM algoritmi eksponentsiaalne keerukus ei ole puudusi tekitanud.

7 Tulemuste võrdlev analüüs

Järgnevas alam-peatükkides võrdleme potentsiaalsete ehk FCA-le toetuvate meetodite tulemusi traditsiooniliste rollikaeve meetoditega, toome võrdluseks GSH ja graafi optimeerimise diagrammid ja viimasena vaatame visuaalse rollide kaevandamise meetodeid. Lõplikult teeme vaadeldud tulemustele järeldused.

7.1 FCA-le tuginevate meetodite tulemuste võrdlus rollikaeve meetoditega

Rollikaeve meetodite uurimisest ja tulemustest avaldus, et rollitehnikas rollide määratlemist ja nendele õiguste sidumist on võimalik väga erinevate meetodite abil teostada. Olgu selleks rollide leidmine kasutajate aspektist, kus õiguste hulkade ühendamise protseduuridega optimeeritud lahendus leitakse (GO) või hoopiski vastupidiselt õiguste aspektist, kus toimub õiguste alamhulkade loendamine ning selle alusel rollidele kasutajate ja õiguste hulkade leidmine (CM ja FM). Lisaks leidsime, et eksisteerib ka FCA teooriale toetuvaid traditsioonilisi rollikaevandamise meetodeid (HM), mis kärbivad Galois kontseptivõret, saades sellest kontseptide tulemused ehk rollid.

Käesolevas peatükis teostame QualityCover ja rollikaeve algoritmide tulemuste võrdlused. Vaatame tulemusi leitud rollide arvu põhiselt ja algoritmide kiiruste osas. QC ja traditsiooniliste rollikaeve algoritmide puhul on võimalik paralleelselt võrrelda tekkinud rolle samade andmestike põhjal ainult täieliku katmise puhul, sest rollikaeve algoritmide eesmärgiks on katta andmestiku kasutajaõigused täielikult. Ainuke erinevus rollikaeve valdkonna meetodite puhul on edasised rollitehnikat laiendavad tegevused nagu kõige optimaalsema RBAC seisundi leidmine (rollihierarhia) ja rollide laiendamine semantikaga ehk kasutaja-atribuutide arvestamine.

Allolevas tabelis on esitatud kokkuvõtvalt rollikaevandamise tulemused kolme organisatsiooni andmetele. HM algoritmi tulemused on esitatud kaalutud struktuurse keerukuse minimeerimisele (*minwsc*) ja sulgudes minimaalse rollide arvule (*minR*). CM ja FM algoritmide tulemused rollidele, mis sisaldasid kasutajate hulkasid.

Tabel 27 QualityCover ja rollikaeve algoritmide rollide võrdlus

Algoritm	Spordikeskus	CMS	Intranet
HM rollide tulemused	10 (7)	15 (12)	13 (13)
GO rollide tulemused	7	13	14
CM rollide tulemused	7	12	14
FM rollide tulemused	7	14	14
QC rollide tulemused	6	12	13

Nagu näha on oluliseks tulemuseks QC suutelisus katta andmestikud identsete või isegi väiksemate rollide arvuga, kui seda suudavad teha traditsioonilised rollikaeve algoritmid. Ainuüksi rollihulkade tulemusele toetudes saame järeldada, et QC suudab täita rollikaeve meetodite tööd sama või isegi enam efektiivsemalt. Samuti on varasemalt meetodi autorite poolt ulatuslik eksperimentaalne võrdlus tehtud, kus võrreldi sarnaseid ahne algoritmi meetodeid toetudes erinevatele kvaliteedi mõõdikutele: kompaktsus, stabiilsus, eraldatus ja sidestus [12]. QualityCover pakkus parimaid tulemusi suurema osa andmestikel, mille formaalse konteksti kontseptide arv ulatus paarikümnest kuni miljoni kontseptini. Sellest tulenevalt saab järeldada, et QC algoritm on rakendatav väga suurtele andmestikele, kus võimalik leida heuristiliselt minimaalne arv rolle koos kasutajate ja õiguste hulkadega.

7.1.1 Algoritmide tööaegade tulemused

Algoritmide tööaegade tulemused tabelis näitavad, et kõige kiiremaid tulemusi leiab üle kõikide andmestike GO. Kui alamhulkade loendamise algoritmid (CM ja FM) näitasid esimese andmestiku rollide leidmise puhul häid tööaegasid, siis viimase andmestiku keerukuse tõttu tekkis tulemuste leidmisel teadmata põhjusega probleem. Probleemi lahendamiseks oli katsetatud sarnaste õiguste hulgaga kasutajate ühendamist üheks kasutajaks, mis kahjuks algoritmide sooritust ei parandanud. Sealt tulenevalt näitasid järgmisi kiiruse aegasid QC ja HM. Viimase tulemused olid leitud minimaalse keerukuse seadistustega.

Tabel 28 Algoritmide tööajad lähteandmestike rollikaevandamisel

Algoritm	Spordikeskus	CMS	Intranet
HierarchicalMiner (HM)	100.0ms	123.0ms	89.0ms
Graafi Optimeerimine (GO)	32.0ms	46.0ms	28.0ms

CompleteMiner (CM)	44.0ms	165.0ms	207238.0ms
FastMiner (FM)	34.0ms	76.0ms	162813.0ms
QualityCover (QC)	38.0ms	226.0ms	68.0ms

Aegade kõrval on vajalik arvestada ka kaalutud struktuurse keerukuse tulemusi, mis on kokkuvõtvalt allolevas tabelis. Tööaegade tulemustele on keerukuse tulemused vastupidised, mistõttu ei saa käesolevate andmete põhjal konkreetselt määratleda missugune algoritm tulemuste ja kiiruste osas välja paistab. Selleks on vaja laiendada võrdlust test-andmetega (andmestikega), mille mahud oleksid palju suurema amplituudiga. Seejärel eksperimentaalselt tulemused genereerida ja sobivate mõõdikute (RBAC staatus, tööajad ja rollihulgad) alusel tulemusi analüüsida. Käesolevas lõputöös saame järeldada, et vaadeldavatel andmestikel leitakse tulemused küllaltki sarnaste aegadeaga.

Tabel 29 Algoritmide wsc tulemused

Algoritm	Spordikeskus	CMS	Intranet
HierarchicalMiner (HM)	58	132	87
Graafi Optimeerimine (GO)	133	375	173
CompleteMiner (CM)	96	256	490
FastMiner (FM)	96	248	465

QC algoritmi puhul on väiksemate andmestike puhul tööaja tulemused konkureerivad, kuid CMS tulemuste puhul on näha, et suuremate andmestike korral hakkavad tööajad kasvama. Kitsaskoht peitub algoritmi kõige ajakulukamas tegevuses, milleks on pseudokontseptide seoste arvutamine [12]. Sellegipoolest võib tulemuste põhjal järeldada, et QC pakub häid rollikaevandamise tulemusi ning on edasiselt lähemalt uuritav meetodi laiendamine rollikaevandamise algoritmiks.

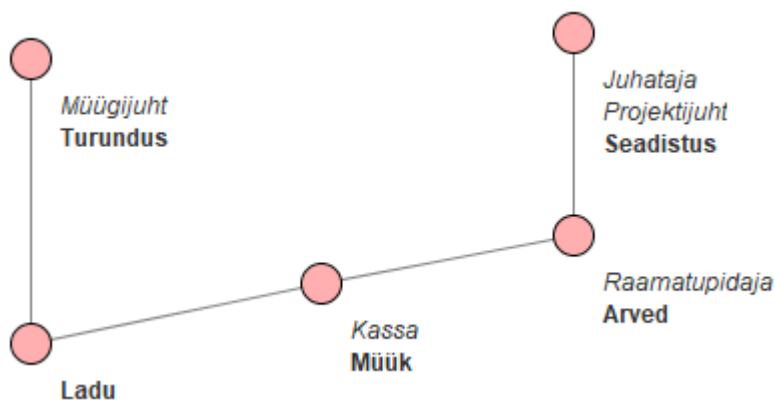
7.2 Visuaalseid esitusi leidvate rollikaeve meetodite võrdlus

Varasemas peatükis 6.2 vaatasime GSH poolt leitud tulemusi ja analüüsisime neid andmestike ja rollide leidmise keerukuse kohapealt. Käesolevas peatükis vaatame kui kompaktselt suudab GSH visuaalse esituse lihtsuse saavutada, informatsiooni lugejale edastada ning kuidas allub teoreetilistes alustes väljatoodud printsiipidele. Lisaks võrdleme teiste rollihierarhiaid esitavate meetoditega.

7.2.1 GSH diagramm ja visualiseerimise printsiibid

Nagu kindlaks tegime, suudab GSH kujutada sama informatsiooni kokkuvõtlikumalt ja konkreetsemalt kui algsed andmetabelid optimeerides sõlmede ja seoste arvu. Mitme elemendi kuvamise omadus GSH-s määrab tindi hulga seoste omapära ja kui seoste kuvamise efektiivsuse ehk mitu funktsiooni või tähendust on ühel sõlmel või joonel.

Kasutame visualiseerimise printsiipidele alluvuse vaatlemiseks varasemalt kasutatud lihtsa andmestiku näidet, mis asub allpool. Kõige esimesena jääb silma andmetindi suhtarv, kus andmete visualiseerimiseks ei ole tarvilik külge lisada võrestike, raamistike ega teisi otseseid andmeid mitte tähistavaid silte või kirjeldusi. Võib öelda, et GSH diagrammid kasutavad peaaegu igat tindi tilka (ingl k *drop of ink*) edastamaks vaatleja informatsiooni. Diagrammidelt ei ole võimalik eemaldada midagi, mis ei kaota käesoleva näite puhul andmeinformatsiooni. Ainuke võimalus on joonist ennast kompaktsemalt tähistada, vähendades seoste pikkust ja tuues sõlmed üksteisele lähemale.



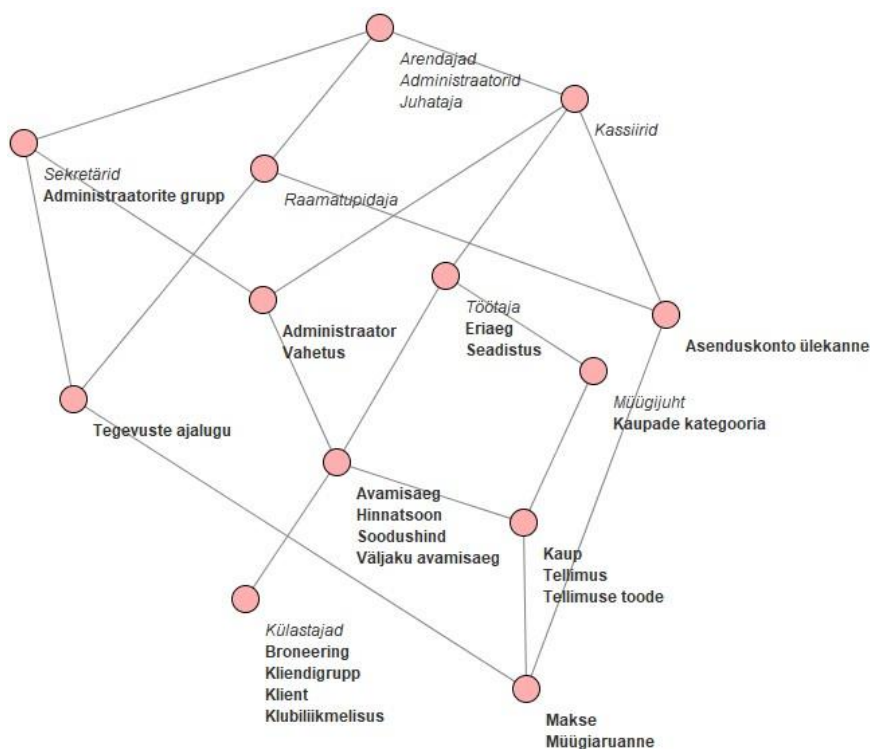
Joonis 21 GSH diagramm

Lõputöös kasutatud rakenduse puhul on erineva tähendusega sildid diagrammis tähistatud suuruse, kirjatüübi ja fondiga arusaadavalt. Kasutajad ning õigused on ühe sõlme all tähistatud erinevalt, mis paneb vaatleja jaoks elemente tajuma sobival viisil (sarnasuse printsiip) ja aitab sõlmede vahelisi seoseid paremini käsitleda. Sõlmede vahelise seoste tähistustes puuduvad GSH diagrammidel nooleotsad ja seda sellepärast, et seostel on mitmed tähendused, mida saab välja lugeda. Üles/alla ja alt/üles liikumised tähistavad ühekordselt mitut erinevat seost sõlmede vahel. Seepärast ei ole mõlemat pidi lisatud liiasuse lisamiseks nooleotsasid joonte mõlematele pooltele. Liikumine ehk seoste loomine erinevate sõlmede ja nendele kuuluvate siltide vahel on tähistatud joontega. Väga

hästi selgitab liikumise taju mitme erineva sõlme vahel jätkuvuse printsiip, kus sõlmede vahelised seosed on otseselt katkenud, kuid vaatlejale tekitab see mulje, et sõlmes lõppenud üks seos on jätkuv sõlme teisest või isegi kolmandast sõlmest.

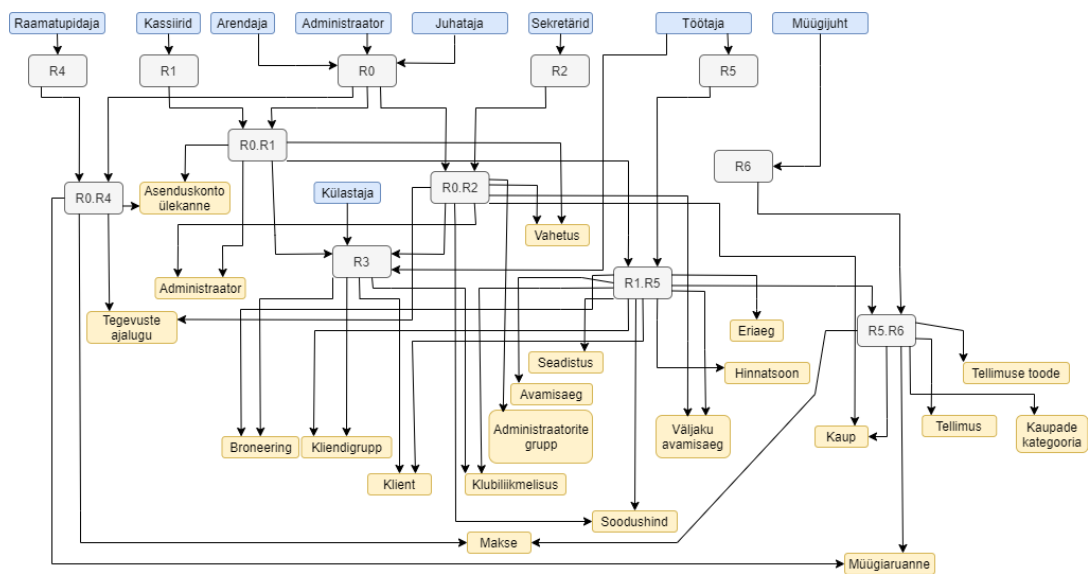
7.2.2 GSH ja Graafi Optimeerimise meetodite võrdlus andmestiku näitel

GSH ja Graafi optimeerimise meetodite eesmärk on visualiseerida seoseid antud lõputöö kontekstis kasutajate ja õiguste vahel kompaktselt. Mõlemas visuaalses esituses on kasutatud sõlmesid, seoseid nende sõlmede vahel ja silte, mis tähistavad vastavalt teatud objekte. Mõlemad meetodid kasutavad sisendiks andmematriksi tabelit, luues lõplikult visuaalselt samaväärse info rollide kohta. Kahe meetodi visuaalsete esituste võrdluseks kasutame spordikeskuse andmestiku näidet. Allolevalt on esitatud mõlema meetodi tulemused joonistena.



Joonis 22 GSH diagramm spordikeskuse tulemuse näitel

Kuna GO hierarhilist tulemust ei olnud RMiner rakendusega võimalik genereerida, on lõputöö autor ise käsitsi visualiseerimise vahendis loonud vastavalt meetodi artikli kirjeldusele, artikli praktilisele näidetele [2] ja tulemustele tuginevalt hierarhia diagrammi. Seda tüüpi joonis on üks võimalike RBAC infrastruktuuri kavandeid väikse rollide hulga tulemustele.



Joonis 23 Graafi optimeerimise hierarhia diagramm spordikeskuse tulemuse näitel

Toetudes teoreetilistes alustes väljatoodud printsiipidele saab kõige lihtsamalt alustada lugemist sõlmede arvu põhjal. Esimesel joonisel on neid 13 ja teisel 42. Graafi optimeerimise joonisel on iga kasutaja ja õiguse jaoks tähistatud eraldi sõlmed. Sealt tulenevalt saab välja tuua andmetindi suhtearvu tulemused, kus GSH suudab täpselt samasuguse lõplike tulemuste korral kordades kompaktsemalt esitust pakkuda. Samamoodi on seoste (joonte) arvud numbrites väljendatuna 19 ja 57, mis kinnitab GSH tulemuste paremust.

Vaadeldes tulemusi Geštalt printsiipide põhiselt, saame viidata läheduse printsiibile. Praegusel juhul on mõlemal meetodil oma eelised ja puudused. GSH diagrammid tähistavad sõlmedele kehtivaid silte väljaspool sõlmesid, tekitades suuremate ja tihedamate diagrammide puhul keerukust iga hulgas asetseva sildi omavahelise seose loomisele. Samas võimaldatakse sellega kompaktsemat diagrammi esitust, sest siltide paigutus sõlmedesse suurendaks oluliselt tindi hulka. Seda näeme GO jooniselt, kus sõlmede tähistused on sõlme sees. Samas ei teki viimasel juhul probleeme sõlme ja sildi seose mõistmisel. Läheduse printsiibil siltide tähistamine kõrvuti ja alati all paremas positsioonis sõlmest vaadatuna räägib GSH diagrammide kasuks andmetindi osakaalu maksimeerimises.

Sarnasuse printsiibil on GSH diagrammides kasutaja puhul määratud fondi tüüp (kalkkiri) ja õiguste puhul rasvane tekst. GO esituses on tähistatud erineva taustavärviga kasutajad, õigused ja rollid. Mõlemal joonisel on suudetud tekitada ühised grupid, mistõttu ei saa

otseselt ühte teisele eelistada ainult sarnasuse printsiibi põhjal. Käesolevalt saame järjekordselt välja tuua andmetindi suhtearvu printsiibi, kus GSH puhul on suurendatud nimetatud suhtearvu väärtust.

Vaadates GSH sõlmesid ja silte ühe komplektina laiendame läheduse printsiipi terviklikkuse printsiibiga, mis paneb tajuma neid objekte ühtse loogilise grupina. Sealt tulenevalt ei ole vajalik iga komplektis eksisteerivat silti siduda sõlmega, sest see on juba arusaadavalt vaadeldes seotud.

GSH joonistelt on näha, et teatud juhtudel asuvad sildid joonte peal, mis oluliselt pärsivad sõlmede omavahelise seoste mõistmist. Näiteks Raamatupidaja silt katab joone ja katkestab selle osaliselt. Kui me sooviks teha joonist laiuse ja kõrguse osas veelgi kompaktsemaks võib seoste katmine suurenedas sellises mahus, et diagramm ei pruugi olla enam üldse loetav. Õnneks toetab seda jätkuvuse printsiip, mis näitab, et objekti naturaalselt tulemit saab tuletada ka juhul, kui see ei ole täielikult nähtav. Sama kehtib ka GO seostest, kus need üksteist katavad, aga ei ole sellises mahus probleemiks. Kõigest sellest tulenevalt on siltide tähistamise omaduse puhul GSH diagrammidel teatav puudus.

Nagu näeme on ühesuguse andmestiku puhul visuaalsed esitused vägagi erineva andmetindi suhtearvu ja kompaktsusega. GSH esitab GO joonisega võrreldes palju kompaktsemat tulemust, aga mõlema puhul on joonistele toetudes välja loetav ühesugune lõplik info kasutajate, õiguste ja rollide osas. GSH ei kuva otseselt kaevandatud rolle, kuid esitab kandidaatorolle ja selles sisalduvad täpselt samasugused rollide hulgad kasutajate ja õigustega nagu GO joonisel. Samuti tuleb mainida seda, et käesolevalt oli tegemist väga väikese andmestikuga (128 kasutajaõigusega). Rollikaevandamisel on tegemist andmestikega, mille kasutajaõiguste arvud ulatuvad tuhandetesse või kümnetesse tuhandetesse. See tähendab, et selline arhitektuur ja keerukus RBAC infrastruktuuri esitamisel sellistest algoritmidest nagu GO võib tekitada joonised, mida on kas raske või isegi võimatu käsitleda. Käesoleva peatükki võrdluse eesmärk oli kuvada GSH diagrammide omadusi, mida oleks võimalik rakendada rollikaevandamises kasutatavate RBAC infrastruktuuri diagrammide loomisel. Kui varasemalt on käsitletud GSH diagramme kasutajalugude kompaktseks kirjeldamiseks [9], siis samasuguste omaduste alusel saab laiendada meetodit rollikaevandamise kogukondade algoritmide tulemuste kuvamiseks.

7.3 Monotoonsete süsteemide ümberjärjestamise tehnika ja Visual

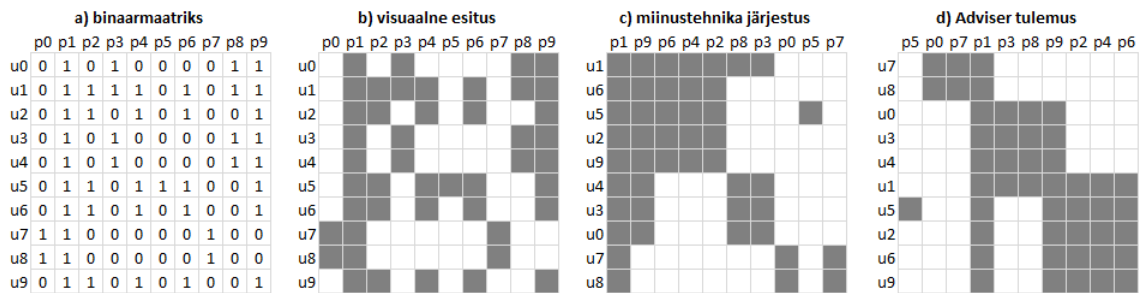
Role Mining

Käesolevas peatükis võtame vaatluse alla monotoonsete süsteemide miinustehnika meetodi, mille eelnevad teoreetiline osa ja andmestike tulemused näitasid, et ei pakuta otseselt alternatiivse rollikaevandamise meetodina välja rollide hulka, vaid seda kuidas võiks saada leida rolle visuaalselt. Sealt tulenevalt saame pidada seda alternatiivseks visuaalse rollide kaevandamise meetodiks. Vastukaaluks toome võrdluseks visuaalse rollikaevandamise meetodi, mille põhiline eesmärk on teatavasti rollide leidmine binaarmaatriksi ümberjärjestatud tabelitest visuaalselt samal põhimõttel. Sealt tulenevalt saame vaadelda, kuidas sarnanevad antud kaks meetodit binaarmaatriksi ümberjärjestamisel ja kuidas need on eelduseks tähtsate rollide leidmisele.

Miinustehnikas tulevad välja rollide hulgad veergude järgi. Nagu nägime on tulemused väga sarnased kontseptiahelatele, kus loomulikult teel tuleb välja ahelate järjestus Pareto serieerimise viisil. Teiseks struktuuralseks mustriks andmetabelite ümberjärjestamisel on plokk-diagonaal serieerimine, mille mustri struktuuri järgi visuaalse rollikaevandamise meetod. Seda sellepärast, et maatriksi sorteerimise algoritm Adviser lähtub rollide maksimaalsest eraldatusest (rollid peavad võimalikult vähe ühiseid kasutajaid ja õigusi jagama), leitakse maksimaalselt suurte pindadega rollid ja sorteerimine toimub gruppide kaupa. Alloleval joonisel on esitatud ühine näide miinustehnika ja Adviser algoritmi tulemustele, toetudes visuaalse rollikaevandamise artikli lihtsale algandmestikule. Lõputöö kirjutamise vältel oli proovitud visuaalse rollikaevandamise artikli autoritega korduvalt ühendust võtta, kuid kahjuks ei olnud võimaldatud lähtekoodi jagamine. Ühe artikli kaasautori vastusele põhinedes oli rakendus loodud peaautori poolt ning praeguse hetkeni ei ole teada selle seisund. Kuna lõputöö eesmärgiks ei ole visuaalse rollikaevandamise kahe algoritmi implementeerimine rakendusena, toetume artiklis väljatoodud lihtsale andmestiku näitele, mis võimaldab esitada meetodi tulemusi ja korrektset struktuurset ülesehitust.

Miinustehnika tulemustest on rollide leidmine ahela põhine, olgu selleks kasutajate või õiguste järgi. Seejärel tekib ahela lõppedes andmemüra, mida on võimalik sobivalt rollidesse jaotada, kuid ei järgi enam ühtset struktuuri rollide leidmisel. Käesoleva näite puhul oleks isegi võimalik veergusid p5 ja p7 joonisel 24c ümber tõsta, et tekiks

andmemüra ühised alad, mida võimalik visuaalsemalt paremini üheks rolliks tekitada. Praeguses näites ei tekitaks selline muudatus suurema tihedusega aladel konflikte.



Joonis 24 Miinustehnika ja Adviser tulemused vastavalt serieerimise mustritele

Algoritmi autorite poolt artiklis esitatud Adviser tulemus [19] on eraldatud plokkidena ning anti-ahela järjestuses. Käesolevas näites on andmemüra osakaal väiksem kui miinustehnika puhul. Plokid on küll osaliselt kattuvad, kuid aitavad seeläbi minimeerida andmemüra ja tekitada visuaalselt parema esituse rollide ekstraheerimiseks.

Miinustehnika suureks puuduseks on ainult osalise katvuse leidmine, mistõttu oleks vajalik algoritmi laiendada, andmaks juurde lisa õigusi andmestiku täielikuks katmiseks rollide kaudu. Teine võimalus on andmemürale eraldi rollid tekitada, aga nagu eelnevalt vaadeldud, ei pruugi andmemüra olla kõige paremini tulemustes esitatud.

7.4 Järeldused

Potentsiaalsete ja traditsiooniliste rollikaevandamise algoritmide rakendamine kolme organisatsiooni andmestikel näitas, et leidub veel selliseid FCA andmeanalüüsi tehnikale toetuvaid meetodeid, mida saab leida rollide kaevandamisel. QualityCover rollihulkade tulemused olid kõigi kolme andmestiku korral võrdsed või paremad traditsiooniliste rollikaevandamise algoritmide tulemustega. HierarchicalMiner tulemused, mille kaaluparameetrid oli määratud leidmaks minimaalsete rollide arvu jäid ühel juhul suuremaks ja teistel juhtudel identseks QC algoritmi tulemustega. Ainsaks QC kitsaskohaks võib pidada tööaegade tulemusi andmestike suuruse ja keerukuse kasvades, mille kontrollimiseks oleks vajalik tulevikus laiendada eksperimentaalseid võrdlusi suuremate juurdepääsuõiguste arvuga test-andmetel.

Rollikaevandamise tulemustest tekkivad RBAC infrastruktuuride diagrammid võivad olla mahult suurtest andmestikest tingituna raskesti käsitletavad või kulukalt realiseeritud.

Käesolevas lõputöös vaatasime GSH diagrammide omadusi visuaalsete printsiipide abil rollikaevandamise valdkonnas ja nägime võimekust esitada rollihierarhia diagramme oluliselt kompaktsemalt ja konkreetsemalt. Ühe andmestiku näite põhjal oli GSH seoste ja sõlmede arv diagrammil kordades väiksem, mis otseselt viitab andmetindi suhtearvu paranemisele ja keerukuse vähenemisele. Edasine samm oleks uurida RBAC infrastruktuuride esitamist GSH diagrammide visuaalse esituse abil ja selle võimekust rollikaevandamise valdkonnas läbi FCA kontseptivõre kärpimise meetodi.

Visuaalne rollikaevandamine on RBAC rollitehnika valdkonnas uus rollikaevandamise viis, millel puuduvad laiemalt uuritud alternatiivsed meetodid. Lihtsustades rollitehnika protsessi läbi visuaalse esituse on tegemist potentsiaalse edasiarendusega valdkonnas, mida saab laiendada erinevate lähenemistega. Üheks selliseks leitud meetodiks on andmetabelite ümberjärjestamise tehnikad, mis võimaldavad kasutajaõiguste andmestike visualiseerida ja seeläbi pakuvad teoreetilist võimekust visuaalseks rollikaevandamiseks.

7.5 Võimalikud edasiarendused

Tulemused ja nende võrdlus näitasid, et FCA andmeanalüüsi tehnikas leidub veel meetodeid, mis on alternatiiviks tavapärasele rollikaeve meetoditele valdkonnas. Edasiselt oleks võimalik käesolevas lõputöös leitud QualityCover meetodi võrdlemine traditsiooniliste meetoditega laiendades tulemuste genereerimist mahukamate test-andmetega ja hindamist paika pandud mõõdikute alusel. Uurimise eesmärgiks oleks kinnitada QualityCover kitsaskohta tööaegade puhul, kui ülesandeks on rollide kaevandamine väga suurtest andmestikest. Teadaolevalt kinnitasid lõputöö andmestike tulemused osaliselt tööaegade kiiremat kasvumist võrreldes traditsiooniliste meetoditega.

Teine võimalik edasine uurimistöö eesmärk saaks olla GSH diagrammide võrdlemine edasiste rollihierarhiaid esitavate rollikaevandamise meetoditega ja võimalusel kasutusele võtmine rollikaevandamise valdkonnas RBAC infrastruktuuride esitamisel.

8 Kokkuvõte

Käesolev lõputöö uuris erinevaid rollikaevandamise meetodeid toetudes FCA-le, monotoonsetele süsteemidele ja rollikaevandamise valdkonnale. Eesmärgiks oli meetoditega rollide kaevandamine kolme organisatsiooni andmestikel ja nende võrdlus tulemuste põhjal. Lisaks uuriti uudset lähenemist, mis toetub rollide visuaalsele esitusele.

Formaalse kontseptianalüüsi tehnikast vaadeldi GSH, kontseptiahelate ja QualityCover meetodeid. Monotoonsete süsteemide tehnikast uuriti lähemalt miinustehnika algoritmi. Rollikaevandamise kogukonna meetoditest vaatlesime nelja algoritmi, mis toetusid kolmele erinevale lähenemisele. Nendeks olid Graafi Optimeerimine, HierarchicalMiner ja ühisele lähenemisele toetuvad CompleteMiner ning FastMiner.

Lõputöö esmasteks tulemusteks olid rollikaevandamise tulemused meetodite rakendustega kolme organisatsiooni andmestikel. GSH tulemused andsid kompaktsed visuaalsed esitused kontseptivõre diagrammide kaudu ning kontseptiahelad ja QualityCover kontseptide ehk rollihulkade tulemused andmestikest. Kontseptiahelate puhul leidsime võrdluses QualityCover-ga, et viimane pakub paremaid tulemusi rollikaevandamise valdkonnas ja kontseptiahelate meetod ei paku praegusel juhul võrreldavaid tulemusi. Miinustehnika algoritmiga andmetabelite ümberjärjestatud esitused Pareto serieerimise mustriga ei andnud meile sarnaselt eelnevate meetoditega rollihulkade tulemusi, vaid pakkus alternatiivset valikut lähenemaks rollide visuaalseks kaevandamiseks. Samuti genereeriti andmestike tulemused traditsiooniliste meetoditega Rminer tööriista abil.

Töö peamiseks tulemuseks oli potentsiaalsete meetodite võrdlus traditsiooniliste meetoditega. Olulise tulemusena saab välja tuua, et FCA-le toetuv QualityCover algoritm, mis pole otseselt rollikaevandamise jaoks mõeldud ja mida ei ole senini kasutatud rollikaevandamise valdkonnas, näitas väga häid tulemusi võrreldes traditsiooniliste rollikaevandamiste meetoditega kolme organisatsiooni andmestikel. QualityCover algoritmiga olid rollide arvud andmestike katmisel identsed või teatud juhtudel isegi väiksemad. Sealt tulenevalt on võimalik tulevikus QualityCover rollikaevandamise valdkonnas edasiselt uurida ja kasutusele võtta.

GSH diagrammide ja Graafi Optimeerimise rollihierarhia tulemuste analüüs avaldas GSH visuaalse esituse kompaktsuse sõlmede ja seoste mitmekordse vähenemisega. Sealt

tulenevalt nägime, et GSH diagrammid, mida samuti ei ole varasemalt rollitehnika eesmärkide saavutamiseks uuritud, suudaks pakkuda RBAC infrastruktuuride visuaalseid esitusi palju paremini, kui käesolevalt võrreldud Graafi Optimeerimise meetod.

Toetudes lihtsale andmestiku näitele vaatlesime miinustehnikat alternatiivse visuaalse rollikaevandamise meetodina. Kahjuks ei saanud käesolevas lõputöös kolme organisatsiooni andmestikel rohkemaid võrreldavaid visuaalseid tulemusi genereerida. Teoreetilisel tasemel on andmetabelite ümberjärjestamise tehnika üks arvestatav meetod edasiseks uurimiseks visuaalse rollikaevandamise lähenemisel.

Lõputöö autori poolne huvi oli veel praegu avalikkusele küllaltki tundmatu RBAC ja rollikaevandamise valdkonna uurimine ning eesmärgiks potentsiaalsete meetodite leidmine rollide kaevandamiseks. Nagu tulemustest selgus leidub FCA-le ja monotoonsetele süsteemidele toetuvaid meetodeid, mida saaks edasiselt uurida ja kasutada rollikaevandamise valdkonnas.

Kasutatud kirjandus

- [1] Coyne, E. 1996. Role Engineering [WWW] <https://dl.acm.org/doi/abs/10.1145/270152.270159> (18.08.2020)
- [2] Zhang, D., Ramamohanarao, K., Ebringer, T. Juuni 2007. Role Engineering using Graph Optimisation [WWW] <https://dl.acm.org/doi/abs/10.1145/1266840.1266862> (14.09.2020)
- [3] Colantonio, A., Pietro, R., Ocello, A. & Verde. 2014. Mining business-related RBAC states through Decomposition [WWW] https://hal.inria.fr/hal-01054506/PDF/2-Paper-155-Mining_Business-Relevant_RBAC_States_Through_Decomposition-Collantonio_Alessandro.pdf (14.09.2020)
- [4] Priss, U. 2004. Formal Concept Analysis in Information Science (15.11.2020)
- [5] Wille, R., Stumme, G., Ganter, B. 2005. Formal Concept Analysis: Foundations and Applications (15.11.2020)
- [6] Torim, A. 2012. A Visual Model of the CRUD Matrix (09.09.2020)
- [7] Knafllic, C. 2015. Storytelling with Data: A Data Visualization Guide for Business Professionals (13.09.2020)
- [8] Tufte, E. R. 1983. The Visual Display of Quantitative Information (15.10.2020)
- [9] Torim, A. 2013. Galois Sub-Hierarchies Used for Use Case Modeling (09.09.2020)
- [10] Torim, A., Mets, M., Raun, K. 2019 Covering Concept Lattices with Concept Chains (09.09.2020)
- [11] Liiv, I. 2010. Seriation and matrix reordering methods: An historical overview (10.09.2020)
- [12] Mouakher, A., Yahia, S. 2016. QualityCover: Efficient binary relation coverage guided by induced knowledge quality [WWW] <https://www.researchgate.net/publication/299375039> (18.09.2020)
- [13] Võhandu, L., Kuusik, R., Torim, A., Aab, E., Lind, G. 2006. Some algorithms for data table (re)ordering using Monotone Systems (09.09.2020)

- [14] Hamid, N., Alarood, A., Manaf, A., Ahmad, R. 2018. Assessing Role Mining Algorithms using RMiner tool in identifying the best model for Access Control [WWW] <http://www.jatit.org/volumes/Vol96No20/23Vol96No20.pdf> (24.10.2020)
- [15] Molloy, I., Li, N., Li, T., Mao, Z. & Wang, Q. 2009. Evaluating Role Mining Algorithms [WWW] <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=4EC050B068A2E3F42A563B12E70C469B?doi=10.1.1.567.4536&rep=rep1&type=pdf> (14.09.2020)
- [16] Molloy, I et al, Calo, S., Lobo, J. Juuni 2008. Mining Roles with Semantic Meanings [WWW] <https://dl.acm.org/doi/abs/10.1145/1377836.1377840> (14.09.2020)
- [17] Vaidya, J., Atluri, V., Guo, Q. Juuli 2010. The role mining problem: A formal perspective [WWW] <https://dl.acm.org/doi/pdf/10.1145/1805974.1805983> (14.09.2020)
- [18] Vadyia, J., Atluri, V., Warner, J. 2006. RoleMiner: Mining Roles using Subset Enumeration [WWW] <https://dl.acm.org/doi/abs/10.1145/1180405.1180424> (14.09.2020)
- [19] Colantonio, A., Pietro, R., Ocello, A. & Verde, N. 2012. Visual Role Mining: A picture Is Worth a Thousand Roles [WWW] <https://ieeexplore.ieee.org/abstract/document/5710922> (27.09.2020)
- [20] Li, R., Li, H., Wang, W., Ma, X., Gu, X. 2013. RMiner: A Tool Set for Role Mining [WWW] <https://dl.acm.org/doi/abs/10.1145/2462410.2462431> (25.10.2020)

LISA 1 - Algandmestike kasutajaõiguste seoste tabelid

Tabel 30 Spordikeskus kasutajaõiguste seoste tabel

Kasutaja	Õigused
arendajad, administraatorid, juhataja	klient, kliendigrupp, klubiliikmelisus, broneering, administraator, administraatorite grupp, hinnatsoon, avamisaeg, väljaku avamisaeg, eriaeg, kaupade kategooria, kaup, tellimus, tellimuse toode, makse, seadistus, müügiaruanne, asenduskonto ülekanne, tegevuste ajalugu, vahetus, soodushind
kassiid	klient, kliendigrupp, klubiliikmelisus, broneering, administraator, hinnatsoon, avamisaeg, väljaku avamisaeg, eriaeg, kaupade kategooria, kaup, tellimus, tellimuse toode, makse, seadistus, müügiaruanne, asenduskonto ülekanne, vahetus, soodushind
sekretärid	klient, kliendigrupp, klubiliikmelisus, broneering, administraator, administraatorite grupp, hinnatsoon, avamisaeg, väljaku avamisaeg, kaup, tellimus, tellimuse toode, makse, müügiaruanne, tegevuste ajalugu, vahetus, soodushind
külastaja	klient, kliendigrupp, klubiliikmelisus, broneering
raamatupidaja	makse, müügiaruanne, asenduskonto ülekanne, tegevuste ajalugu
töötaja	klient, kliendigrupp, klubiliikmelisus, broneering, hinnatsoon, avamisaeg, väljaku avamisaeg, eriaeg, kaupade kategooria, Kaup, tellimus, tellimuse toode, makse, seadistus, müügiaruanne, soodushind
müügijuht	kaupade kategooria, kaup, tellimus, tellimuse toode, makse, müügiaruanne

Tabel 31 CMS kasutajaõiguste seoste tabel

Kasutaja	Õigused
jaanus	portals, portal module, positions, language module, folder module, article module, shortcut module, file module, mobile filter, newslit module, news module, cookie policy, no cookie policy, widget module, map module, feedback module, gallery module, product

	catalog module, submenu module, selected events module, ticket module, countdown module, header contents, left columns, footer contents, settings, CMS users, Usergroups, site variables, URL redirects, search results, statistics, external search, API statistics, generic modules, preorders, display scrollbanners, newsmails list, newsmails emails, salespoints, countries, country, city, region, API presets, product catalog, product groups, shows, concerts, venues, promoters, product group links, banners system, bannerscategory, banners statistics, banner, XML import
kadi, mariaD	portals, portal module, positions, language module, folder module, article module, widget module, map module, gallery module, submenu module, ticket module, countdown module, header contents, left columns, footer contents
paul	URL redirects, search results, statistics, external search, API statistics, shows, concerts, venues, promoters, product group links, banners system, bannerscategory, banners statistics, banner, XML import
triinbox, Ulla	folder module, newslit module, news module, generic modules, preorders, display scrollbanner, newsmails list, newsmails emails
artweb	portals, portal module, positions, language module, folder module, article module, shortcut module, file module, mobile filter, newslit module, news module, cookie policy, no cookie policy, widget module, map module, feedback module, gallery module, product catalog module, submenu module, selected events module, ticket module, countdown module, header contents, left columns, footer contents, site variables, environments, URL redirects, search results, statistics, external search, API statistics, generic modules, preorders, display scrollbanners, newsmails list newsmails emails, salespoints, countries, country, city, region, API presets, product catalog, product groups, shows, concerts, venues, promoters, product group links, banners system, bannerscategory, banners statistics, banner, webgate controller, widget generator,

	seatings plan, crontab generator, seating plan generator, XML import
hannaj, irina	product catalog, product groups, shows, concerts, venues, promoters, product group links, XML import
juliana	portals, portal module, positions, language module, folder module, article module, shortcut module, file module. mobile filter, newlist module, news module, widget module, map module, feedback module, gallery module, product catalog module, submenu module, selected events module, ticket module, countdown module, header contents, left columns, footer contents, product catalog, product groups, shows, concerts, venues, promoters, product group links
kaire	shows, concerts, venues, promoters, product group links, widget generator, seatings plan, crontab generator, seating plan generator
kert	portal module, selected events module, salespoints, countries, country, city, region, banners system, bannerscategory, banners statistics, banner
kristiine, mihkel	portals, portal module, positions, language module, folder module, article module, shortcut module, file module, mobile filter, newlist module, news module, cookie policy, no cookie policy, widget module, map module, feedback module, gallery module, product catalog module, submenu module, selected events module, ticket module, countdown module, header contents, left columns, footer contents, site variables, URL redirects, search results, statistics, external search, API statistics, generic modules, preorders, display scrollbanners, newsmails list, newsmails emails, salespoints, countries, country, city, region, API presets, product catalog, product groups, shows, concerts, venues, promoters, product group links, banners system, bannerscategory, banners statistics, banner, webgate controller, widget generator, seatings plan, crontab generator, seating plan generator, XML import
laura	API statistics, API presets
ylle	cookie policy, no cookie policy, settings, CMS users, Usergroups, site variables, environments, webgate controller, XML import

Tabel 32 Intranet kasutajaõiguste seoste tabel

Kasutaja	Õigused
annika, kadre, mariana, helin, agne, kai, taavet	kliendid, välispäringute kommenteerimine
sirli	delegeerimine, objektid, kliendid, välispäringute kommenteerimine, kalendri staatuste muutmine, päringud, sisepäringud
kaspar	tähtajad, objektid
peeter	objektid
marek	tähtajad, delegeerimine, kliendid, välispäringute kommenteerimine, kalendri vaatamine, kalendri staatuste muutmine, kalendri toimetamine, päringud, sisepäringud, a-rühm, arhiiv
anneli	tähtajad, kalendri vaatamine, kalendri staatuste muutmine, kalendri toimetamine
tiina	delegeerimine, objektid, kliendid, välispäringute kommenteerimine, kalendri staatuste muutmine
hele	tähtajad, objektid, kliendid, pakkumised
mart	tähtajad, välispäringute kommenteerimine, kalendri vaatamine, kalendri staatuste muutmine
jürgen	tähtajad
eva-liisa	delegeerimine, objektid, kliendid, kasutajad, välispäringute kommenteerimine, kalendri vaatamine, kalendri staatuste muutmine, kalendri toimetamine, päringud, sisepäringud, a-rühm, arhiiv
airi	delegeerimine, objektid, kliendid, kasutajad, pakkumised
tõnu	tähtajad, delegeerimine
jaanus	tähtajad, delegeerimine, objektid, kliendid, kasutajad, pakkumised, välispäringute kommenteerimine, kalendri vaatamine, kalendri staatuste muutmine, kalendri toimetamine, failid, teated, päringud, sisepäringud, a-rühm, arhiiv

LISA 2 – CMS ja Intranet kontseptiahelate tulemused

Kasutajatele ühised õigused on tuletatavad LISA 1 tabelitest

Tabel 33 Kontseptiahelate tulemused CMS andmestikul

K	Kasutajad	Õigused
1	kert	portal module
2	mariaD, kadi	14 erinevat kasutajatele ühist õigust
3	juliana	15 erinevat kasutajatele ühist õigust
4	jaanus	24 erinevat kasutajatele ühist õigust
5	mihkel, kristiine	5 erinevat kasutajatele ühist õigust
6	artweb	environments
Kasutajaõiguste katvuse määr – 78,55%		
1	kaire, juliana	shows, concerts, venues, promoters, product group links
2	irina, hannaj	XML import
3	paul	9 erinevat kasutajatele ühist õigust
4	jaanus	39 erinevat kasutajatele ühist õigust
5	mihkel, artweb, kristiine	10 erinevat kasutajatele ühist õigust
Kasutajaõiguste katvuse määr - 87,13%		
Täieliku katvuse leidmise jätkumine		
1	ulla, artweb, triinbox, kristiine, mihkel	8 erinevat kasutajatele ühist õigust
2	jaanus	49 erinevat kasutajatele ühist õigust
Kasutajaõiguste katvuse määr – 92,23%		
1	kert	11 erinevat kasutajatele ühist õigust
2	mihkel, jaanus, kristiine	43 erinevat kasutajatele ühist õigust
3	artweb	8 erinevat kasutajatele ühist õigust
Kasutajaõiguste katvuse määr - 94,91%		
1	paul, irina, hannaj	XML import

2	artweb, kristiine, mihkel	site variables, no cookie policy, cookie policy
3	jaanus	settings, usergroups, cms users
4	ylle	webgate controller, environments
Kasutajaõiguste katvuse määr – 97,32%		
1	kaire	9 erinevat kasutajatele ühist õigust
2	mihkel, artweb, kristiine	50 erinevat kasutajatele ühist õigust
Kasutajaõiguste katvuse määr – 98,39%		
1	irina, hannaj, juliana	7 erinevat kasutajatele ühist õigust
2	mihkel, artweb, kristiine	47 erinevat kasutajatele ühist õigust
3	jaanus	usergroups, settings, cms users
Kasutajaõiguste katvuse määr – 99,46%		
1	paul	API statistics
2	laura	API presets
3	mihkel, jaanus, kristiine	52 erinevat kasutajatele ühist õigust
4	artweb	6 erinevat kasutajatele ühist õigust
Kasutajaõiguste katvuse määr – 100%		

Tabel 34 Kontseptiahelate tulemused Intranet andmestikul

K	Kasutajad	Õigused
1	annika, taavet, kai, agne, kadre, mariana, helin	välispäringute kommenteerimine
2	mart	kalendri staatuste muutmine
3	tõnu	delegeerimine
4	tiina	kliendid
5	sirli	sisepäringud, päringud
6	marek	a-rühm, kalendri toimetamine, arhiiv, kalendri vaatamine

7	evaliisa	objektid, kasutajad
8	jaanus	tahtajad, pakkumised, failid, teated
Kasutajaõiguste katvuse määr – 65,93%		
1	kaspar, peeter	objektid
2	evaliisa, tiina, sirli	kliendid
3	airi	pakkumised
4	hele	tahtajad
5	jaanus	12 erinevat kasutajatele ühist õigust
Kasutajaõiguste katvuse määr - 78,02%		
1	kaspar, jürgen, hele	tahtajad
2	mart, tõnu	kalendri staatuste muutmine, kalendri vaatamine
3	anneli	kalendri toimetamine
4	marek	6 erinevat kasutajatele ühist õigust
5	jaanus	6 erinevat kasutajatele ühist õigust
Kasutajaõiguste katvuse määr – 90,11%		
Täieliku katvuse leidmise jätkumine		
1	annika, taavet, tiina, kai, hele, sirli, agne, marek, kadre, mariana, helin	kliendid
2	airi	delegeerimine, objektid, kasutajad
3	evaliisa	8 erinevat kasutajatele ühist õigust
4	jaanus	tahtajad, pakkumised, failid, teated
Kasutajaõiguste katvuse määr – 100%		

LISA 3 – Hierarchical Miner minimaalse rollide arvu seadistuse tulemused

Tabel 35 HM tulemused minimaalse rollide arvu seadistusega Spordikeskuse andmestikul

R	Kasutajad	Õigused
1	arendajad, sekretärid, administraatorid, juhataja	administraatorite grupp
2	külastajad	klient, kliendigrupp, klubiikmelisus, broneering
3	arendajad, kassiirid, administraatorid, juhataja, Töötaja	hinnatsoon, avamisaeg, väljaku avamisaeg, eriaeg, seadistus, soodushind
4	Muugijuht	kaupade kategooria, kaup, tellimus, tellimuse toode, makse, müügiaruanne
5	arendajad, kassiirid, administraatorid, juhataja	administraator, hinnatsoon, avamisaeg, väljaku avamisaeg, kaup, tellimus, tellimuse toode, makse, müügiaruanne, vahetus, soodushind
6	arendajad, kassiirid, administraatorid, raamatupidaja, juhataja	makse, müügiaruanne, asenduskonto ülekanne
7	arendajad, administraatorid, raamatupidaja, juhataja	makse, müügiaruanne, tegevuste ajalugu

Kasutajatele ühised õigused on tuletatavad LISA 1 tabelitest

Tabel 36 HM tulemused minimaalse rollide arvu seadistusega CMS andmestikul

R	Kasutajad	Õigused
1	kadi, mariaD	15 erinevat kasutajatele ühist õigust
2	jaanus, paul, artweb, kristiine, mihkel	14 erinevat kasutajatele ühist õigust
3	jaanus, trinibox, ulla, artweb, kristiine, mihkel	8 erinevat kasutajatele ühist õigust
4	jaanus, artweb, juliana, kristiine, mihkel	10 erinevat kasutajatele ühist õigust
5	artweb, kaire, kristiine, mihkel	9 erinevat kasutajatele ühist õigust
6	jaanus, artweb, kert, kristiine, mihkel	11 erinevat kasutajatele ühist õigust

7	jaanus, artweb, kristiine, laura, mihkel	API statistics, API presets
8	jaanus, ylle	6 erinevat kasutajatele ühist õigust
9	artweb, ylle	environments
10	jaanus, hannaj, irina	7 erinevat kasutajatele ühist õigust
11	artweb, kristiine, mihkel	cookie policy, no cookie policy, site variables, webgate controller
12	jaanus, hannaj, irina	XML import

Tabel 37 HM tulemused minimaalse rollide arvu seadistusega Intranet andmestikul

R	Kasutajad	Õigused
1	sirli, kaspar, peeter, tiina	objektid
2	kaspar, marek, anneli, hele, mart, jürgen, tõnu	tahtajad
3	jaanus	failid, teated
4	sirli, marek, tiina, tõnu	delegeerimine
5	annika, kadre, sirli, mariana, helin, agne, tiina, kai, taavet	kliendid
6	evaliisa, airi	kasutajad
7	hele, airi	pakkumised
8	annika, kadre, sirli, mariana, helin, marek, agne, tiina, mart, kai, tõnu, taavet	välispäringute kommenteerimine
9	marek, mart, tõnu	kalendri vaatamine
10	sirli, tiina	kalendri staatuste muutmine
11	marek, anneli	kalendri toimetamine
12	sirli	päringud, sisepäringud
13	marek, evaliisa	a-rühm, arhiiv