

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Informaatika instituut

Infosüsteemide õppetool

# **GSH meetodi kasutamine infosüsteemide analüüsimisel**

bakalaureusetöö

Üliõpilane: Laura Erro  
Üliõpilaskood: 104349IABB  
Juhendaja: Ants Torim

Tallinn  
2015

## **Autorideklaratsioon**

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

-----  
*(kuupäev)*

-----  
*(allkiri)*

## **Annotatsioon**

Antud töö eesmärkideks on tõestada, et GSH meetodi kasutamine infosüsteemi analüüsis võimaldab ennetada probleeme tulevikus infosüsteemis; et selle meetodi kasutamine on hea viis analüüsimiseks ja et seda on lihtne kasutada ja saab rakkendada ükskõik millise infosüsteemi jaoks.

Sellised eesmärgid olid püstitud sellepärast, et tulevad infosüsteemid saavad rikkuda alates algstaadiumist, kuna koosneb mitte loogiliselt sobivamate osadest, või probleemid ilmuvad siis, kui infosüsteemi tuleb rohkem arendada.

Pärast GSH meetodi rakkendamise tudengite projektide infosüsteemide analüüsile me saime tulemuseks seda, et antud analüüsi meetod aitab organiseerida infosüsteemi tarkvara koodi struktuuri, aitab leida vead teistes mudelites, mis aitavad analüüsi teha, ning aitab leida kriitilisi funktsioone. Kõik need probleemid ilmuvad siis, kui infosüsteemi analüüs on tehtud valesti või analüüsides polnud kogutud vajalik informatsioon.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 43 leheküljel, 6 peatükki, 18 joonist, 7 tabelit, 1 lisa.

## **Abstract**

The aim of the work is to prove: that using GSH method in infosystem analysis makes possible to forecast problems of a future infosystem, that using this method is a good way of analysing, it is easy to use and applyable for any infosystem.

Such targets have been set, because future infosystems might break, consist of non logically matching parts in the beginning or problems might occure when there will be a need to develop this infosystem. All those problems might be faced if analysis has been made in a wrong way or it was not full enough to get all the needed information.

After we have applied GSH method to students' projects' infosystem analysis we got next results: a given method of analysis helps to organize the structure of a software code, help to find mistakes in other models which help to analyze and also to find critical parts of an infosystem.

The thesis is in Estonian and contains 43 pages of text, 6 chapters, 18 figures, 7 tables, 1 appendix.

## **Lühendite ja mõistete sõnastik**

<b>SPOF</b>	<i>Single Point Of Failure</i> Üksikrikke
<b>GSH</b>	<i>Galois' Sub-hierarchy</i> Galois' Alam-hierarhia
<b>FCA</b>	<i>Formal Concept Analysis</i> Formaalne Kontseptuaal Analüüs

## Jooniste nimekiri

Joonis 2. 1 konteksti K binaarne seos (Berry, 2005).....	11
Joonis 2. 2 kontsepti B(G,M,I) võre (Berry, 2005) .....	11
Joonis 2. 3 Galois' alam-hierarhia (Berry, 2005).....	12
Joonis 2. 4 Kontsepti võre .....	14
Joonis 2. 5 Galois' Alam-hierarhia .....	14
Joonis 3.1. 1 Tele2 Eesti AS. GSH diagramm (põhiobjekt/kasutusjuht) .....	19
Joonis 3.1. 2 Tele2 Eesti AS. GSH diagramm (register/kasutusjuht).....	20
Joonis 3.1. 3 Tele2 Eesti AS. GSH diagramm (põhiobjekt/funktsionaal-allsüsteem).....	21
Joonis 3.1. 4 Tele2 Eesti AS. GSH diagramm (register/funktsionaal-allsüsteem).....	22
Joonis 3.2. 1 ARK. GSH diagramm (põhiobjekt/kasutusjuht) .....	24
Joonis 3.2. 2 ARK. GSH diagramm (põhiobjekt/register/kasutusjuht).....	25
Joonis 3.3. 1 POOD.ELION.EE . GSH diagramm (põhiobjekt/kasutusjuht).....	27
Joonis 3.3. 2 POOD.ELION.EE . GSH diagramm 1 (põhiobjekt/register/kasutusjuht).....	28
Joonis 3.3. 3 POOD.ELION.EE . GSH diagramm 2 (põhiobjekt/register/kasutusjuht).....	29
Joonis 3.4. 1 Coursera. GSH diagramm (põhiobjekt/kasutusjuht) .....	31
Joonis 3.4. 2 Coursera. GSH diagramm (põhiobjekt/register/kasutusjuht).....	32
Joonis 3.5. 1 Solaris Kino. GSH diagramm (põhiobjekt/kasutusjuht).....	35
Joonis 3.5. 2 Solaris Kino. Andmemudel .....	35

## **Tabelite nimekiri**

Tabel 2. 1 binaarsete suhete maatriks .....	12
Tabel 3.1. 1 Tele2 Eesti AS. CRUD – maatriks .....	18
Tabel 3.2. 1 ARK. CRUD - maatriks	24
Tabel 3.3. 1 POOD.ELION.EE . CRUD – maatriks .....	27
Tabel 3.4. 1 Coursera. CRUD – maatriks.....	31
Tabel 3.5. 1 Solaris Kino. CRUD – maatriks .....	34
Tabel 4.1 GSH mudelite hinnang .....	36

## Sisukord

1. Sissejuhatus.....	9
2. Probleemi püstitus. Teooria.....	10
3. Projektide analüüs.....	16
3.1 Tele2 Eesti AS .....	16
3.2 ARK.....	23
3.3 POOD.ELION.EE .....	25
3.4 Coursera .....	30
3.5 Solaris Kino.....	32
4. GSH mudelite hinnang .....	36
5. Tulemused ja järeldused .....	38
6. Kokkuvõte .....	40
Summary .....	41
Kasutatud kirjandus.....	42
Lisa 1. "Tele2 Eesti AS" projekti andmebaas.....	43



# 1. Sissejuhatus

Infotehnoloogia on selline teadusharu, mis pidevalt areneb. Ning tarkvara loomise protsessist võtab osa mitte üks spetsialist vaid palju rohkem. Seega võivad tarkvara koostamisel tekkida olukorrad, mil tarkvara osad ei tööta koos. Uus funktsionaalsus võib muuta kogu tarkvara struktuuri ja sellel juhul programm võib lisada süsteemi vigu. Selleks, et leida kõike vea saamise riske on üheks võimaluseks analüüsida infosüsteemi ja koguda nii palju kasulikku informatsiooni selle loomiseks kui võimalik.

Eesmärgiks on tõestada, et GSH meetodi kasutamine infosüsteemide analüüsis võimaldab luua täielikkuma infosüsteemi, millel on vähem vigu ka tulevikus.

Antud töös me uurime kontsept võrede kanoonilise esituse kohaldamist ( mida nimetatakse Galois' Alam-hierarhiaks ("Galois' Sub-hierarchy", GSH)) erinevate infosüsteemide analüüsiks.

Suurem osa tööst on pühendatud hüpoteetiliste, kuid tegelikuste andmete analüüsile. Andmed tulevad Tallinna Tehnikaülikooli tudengite projektidest ainest "Kontseptuaalne süsteemi analüüs", milles analüüsitakse infosüsteemi. Me hakkame kasutama CRUD-maatriksi andmeid ning uurima kuidas otseselt kohaldada GSH, et genereerida algandmetest head visualiseerimist ja kuidas GSH faktoriseerimine ja formaliseerimine võib anda kasu ekspertidele, tarkvara arhitektidele ja kõikidele osapooltele, kes võtavad osa infosüsteemi arendamisest ja hooldamisest. Analüüsis me asetame diagrammides objekte ja kasutusjuhte registritega või funktsionaal all-infosüsteemidega ja leiame eeliseid ja puudusi GSH meetodi rakkendamises.

Esimene osa tööst on teoreetiline osa FCA'st, GSH'st ja CRUD-maatriksist. Pärast me teeme viie projektide analüüsi, kasutades GSH meetodit, võrdleme saadud GSH diagramme ning viimases osas me määrame GSH meetodi eeldusi ja puudusi.

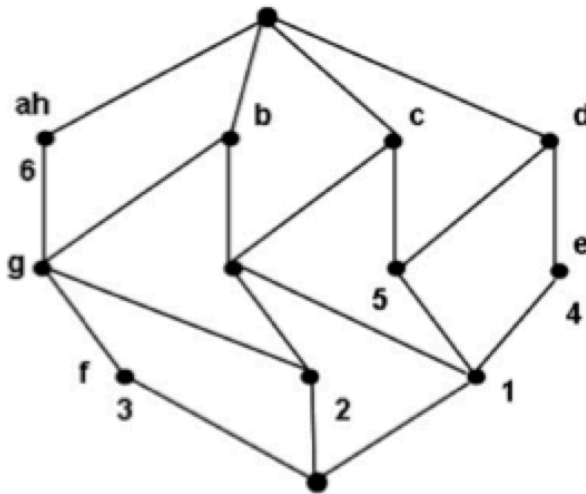
## 2. Probleemi püstitus. Teooria

Tänapäeval on väga oluline, et infosüsteem oleks täielik ja õigesti koostatud, kuna selle arendamine tulevikus viib sellele, et süsteem võib kokku joosta ning lihtsalt ei toimi nii nagu ta peab. Kui infosüsteem toimib valesti, siis kõik osapooled saavad kahju, mida saab mõõta olenevalt sellest kui kriitiline oli viga. Et sellist olukorda vältida on vaja algusest peale põhjalikult analüüsida tuleviku infosüsteemi ja koguda sellest võimalikult palju infot. Infosüsteemi analüüs hõlmab kontseptuaalset mudelit, andmemudelit, CRUD maatriksit ja teisi abimudeleid ning diagramme. Need küll lihtsustavad korrektset ja täieliku infosüsteemi loomist kuid aga analüüsimisel võivad tekkida olukorrad millal mudelite info ei klappi kokku, puudub loogiline side. Tulemuseks on valesti koostatud andmebaas ja selle põhjal ka tarkvara. Seda olukorda oluliselt saaks parandada formaalse kontseptuaalse analüüsi (Formal Conceptual Analysis - FCA) Galois' alam-hierarhia (Galois' Sub-hierarchy - GSH) meetod.

Rudolf Wille ja Bernhard Ganter tutvustasid formaalse kontseptuaalse analüüsi aastal 1990. See põhineb kontsepti filosoofilise mõiste matematiseerimisel ja kontseptide hierarhial (Wille, 1999). Formaalses kontseptuaalses analüüsis formaalseks kontekstiks on kolmik  $K = (G, M, I)$  kus  $G$  ja  $M$  on hulgad (objektid ja atribuutid vastavalt) ja  $I$  on binaarne seos, st  $I \subseteq G \times M$ . Joonis 2.1 näitab konteksti  $K = (\{1, 2, 3, 4, 5, 6\}, \{a, b, c, d, e, f, g, h\}, I)$ . Hulga  $A \subseteq G$  jaoks defineerime hulga  $A' := \{m \in M \mid \forall g \in A, gIm\}$  (atribuutide hulk ühine objektidele hulgast  $A$ ). Vastavalt hulga  $B \subseteq M$  jaoks defineerime hulga  $B' := \{g \in G \mid \forall m \in B, gIm\}$  (objektide hulk mille sees on kõik hulga  $B$  atribuutid) (Berry, 2005). Seega konteksti  $(G, M, I)$  formaalne kontsept on paar  $(A, B)$ , kus  $A \subseteq G, B \subseteq M, A' = B$  ja  $B' = A$ . Hulga  $A$  nimetatakse ekstendiks ja hulga  $B$  kontsepti  $(A, B)$  intendiks. Hulk  $B$   $(G, M, I)$  tähistab konteksti  $(G, M, I)$  kõigi kontseptide hulga. Joonis 2.2 näitab kuidas kontsepti võre vastab meie näidele.

	a	b	c	d	e	f	g	h
1		x	x	x	x			
2	x	x	x				x	x
3	x	x				x	x	x
4				x	x			
5			x	x				
6	x							x

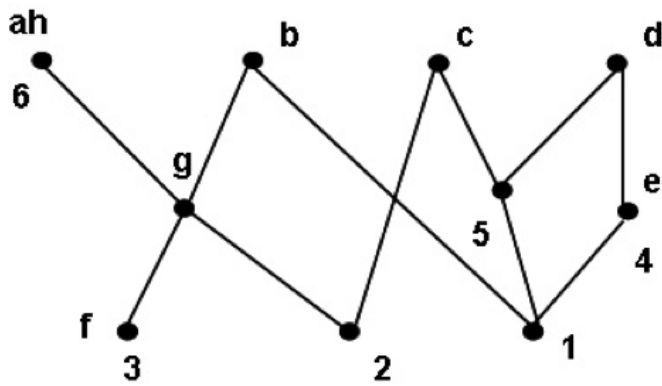
Joonis 2. 1 konteksti K binaarne seos (Berry, 2005)



Joonis 2. 2 kontsepti  $B(G,M,I)$  võre (Berry, 2005)

Kontsepte  $CO = \{\gamma o = (o'', o') | o \in G\}$  nimetatakse  $o$  objekt-kontseptideks ja kontsepte  $CA = \{\mu a = (a', a'') | a \in A\}$  nimetatakse atribuut-kontseptideks. (Berry, 2005) Objekt-kontsept mis vastab objektile  $o$ ,  $\gamma o$  on kõige väiksem kontsept mille ekstendis on olemas  $o$ . Duaalselt atribuut-kontsept mis vastab atribuudile  $a$ ,  $\mu a$  on kõige suurem kontsept mille intendis on olemas  $a$ . Galois' alam-hierarhia on alamjärjestuseks võrele moodustatud hulgast

$CO \cup CA$  ja võre järjestuse kitsendusest sellele hulga. Joonis 3.2 näitab võre vastavust kontekstiga  $K$ . Joonis 2.3 näitab Galois' alam-hierarhia kontekstile mis on tutvustatud Joonisel 2.1. Galois' alam-hierarhia Hasse diagrammist on näha, et tühjad kontseptid on sealt välja jäetud.



Joonis 2. 3 Galois' alam-hierarhia (Berry, 2005)

Üks võimalustest saada GSH mudeli on järgmine: on olemas binaarsete suhete maatriks:

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

Tabel 2. 1 binaarsete suhete maatriks

Esimene samm: leiame kontsepte, millel on sama atribuudid (ehk numbrid) ja nende arv on kõige suurem:

236ah	123b	125c	145d
-------	------	------	------

Teine samm: iga kontsepti jaoks kirjutame alla kõik võimalikud kahenumbrilised variatsioonid atribuutidest ja vaatame kas nende jaoks on sidet objektidega. Tabelist 3.2. saame:

23abgh	36ah	26ah		12bc	23abgh	13b		12bc	25c	15cd		14de	45d	15cd
--------	------	------	--	------	--------	-----	--	------	-----	------	--	------	-----	------

Siin iga tühi väli jagab kontseptide variatsioonide gruppe eelleitud kontseptide jaoks.

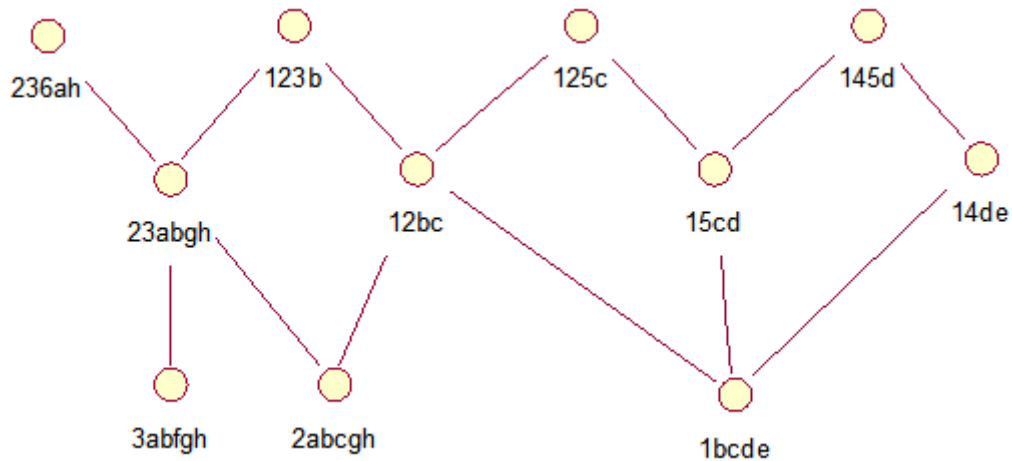
Kolmas samm: leiame ühesuguste kontseptide gruppe ja jätame igast grupist ühe kontsepti, nimelt: “23abgh”, “12bc”, “15cd”. Siis me eemaldame neid kontsepte, mille objektide grupid on teiste kontseptide objektide gruppi seas, ehk me eemaldame “36ah” ja “26ah” kuna kontsept “23abgh” sisaldab objektide gruppi “ah”. Eemaldame ka kontsepti “13b”, sest objekt “b” on kontseptis “12bc”. Ja viimaseks kontseptiks millest me lahti saame on “45d”, sest objekt “d” on kontseptis “45de”. Meil jäid järgmised kontseptid:

23abgh			12bc			15cd			14de
--------	--	--	------	--	--	------	--	--	------

Neljandas sammus me kordame loogikat sammus kolm ja saame järgmisi kontsepte:

2abcgh	3abfgh	1bcde
--------	--------	-------

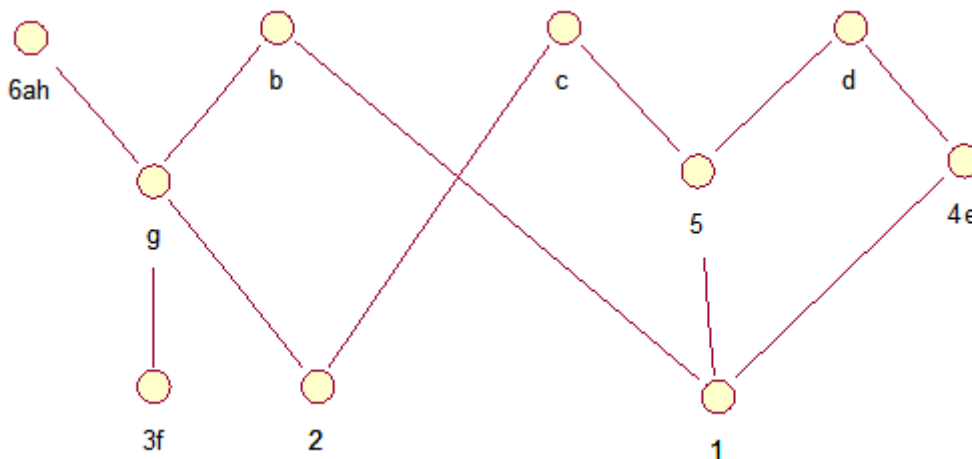
Kui me võtame saadud tulemusi igast sammust ja seome neid loogiliselt siis me saame järgmist mudelit:



Joonis 2. 4 Kontsepti võre

Nüüd selleks, et saada GSH mudelit on vaja lihtsustada eelnevat diagrammi.

Esialgu, liigume allpoolt ülespoole eemaldades atribuute ehk numbreid mis hakkavad korduma kontseptis, mis asub ühe rea üleval antud kontseptist. Siis liigume ülespoolt allapoole ja eemaldame objekte ehk tähti, mis hakkavad korduma kontseptides, mis asuvad ühe rea allapoole vaadatuimast kontseptist. Nii me saame Galois' mudelit:



Joonis 2. 5 Galois' Alam-hierarhia

GSH meetodi kasutamine infosüsteemide analüüsimisel võimaldab analüüsida produktiivsemalt.

Aga selleks, et genereerida GSH diagrammi me kasutame CRUD - maatriksit.

CRUD – maatriks on tabel, mille read on protseduurid ja veergudeks on tabelid. Iga protseduur võib kohaldada loomise (Create), lugemise (Read), uuendamise (Update) või kustutamise (Delete) operatsioone ühe või rohkem tabelite jaoks.

### **3. Projektide analüüs**

Hüpoteesi tõestamiseks oli otsustatud analüüsida GSH meetodiga 5 Tallinna Tehnikaülikooli tudengite kirjutatud projekti. Projektide infosüsteeme analüüsimisel kasutatakse CRUD maatriksit, mis aitab GSH diagrammi saada. Juhul, kui CRUD maatriksis objekti ja kasutusjuhu vahel on ükskõik milline side, siis GSH mudeli loomiseks võtame arvesse seda sidet, vastasel juhul GSH diagrammis me seda sidet ei näe. Oli otsustatud kasutada tudengi Maarja Raud tarkvara (Raud, 2012), mis genereerib vajalikku GSH diagrammi.

Antud töö erineb Julia Visnapu tööst (Visnapu, 2013) sellega, et analüüsis kasutatakse ka GSH-diagramme, mis sisaldavad ka registre. Registrid olid lisatud projektidesse, et teha analüüsi efektiivsemaks ja terviklikumaks. Lisaks analüüsis kasutatakse teiste tudengite projekte ning pakutakse uut visualiseerimist ja vaateviisi infosüsteemi probleemile.

#### **3.1 Tele2 Eesti AS**

Projekti koostasid tudengid Daniela Beljakova ja Aljona Skubi (Skubi, 2013).

Antud projekt on Tele2 Eesti AS süsteemist, mis vastutab klientide teenindamise eest ning on seotud teenuspakettide, kaupa müümise, arve koostamise, kliendi ja töötaja saamise, teenuspaketti, kaupa, arve, kliendi ja töötaja arvestamisega. Vaatluse alla ei kuulu palgaarvestuse ega maksimisega seotud rahalisi operatsioonid.

Kirjeldatav infosüsteem peab toetama Tele2 Eesti AS toimuvat protsessi, võimaldama selles protsessis juhataja, süsteemi administraatori, raamatupidaja, klienditeenindaja ning kliendi oluliste tegevuste jälgimist, registreerimist ja lihtsustamist ning juhataja, süsteemi administraatori, raamatupidaja, klienditeenindaja ja kliendi varustamist neile vajaliku infoga.

Tegemist on firmaga, mis pakub oma klientidele kõrge kvaliteediga mobiilside- ja internetiteenust ja mugavat ning kiiret kauba otsimist ning ostmisviisi nii Eestis, kui ka välismaal. Igas riigis on oma peakontor, kus töötab juhataja ning muu personal, näiteks raamatupidaja. Juhataja koos muu personaliga loovad uue paketti, tellivad kaupa. Juhataja kinnitab uut paketti ja kauba kasutusele. Raamatupidaja tegeleb arve koostamisega. Klient saab ise kauba/teenuspaketti valida ning osta.

Tudengite projekti CRUD maatriksil (Skubi, 2013) on päris palju kasutusjuhte ning objekte:



Objekt/ Kasutusjuht	a	b	c	d	e	f	g	h	i	j	k	l	m
1	C	U											
2	RU	U											
3	R	R											
4	R		R		R	R	R	R	R			R	
5	D	U											
6										C			
7										U			
8										RU			
9										UD			
10	R		R		C	U	C	U	CU	R		C	
11			C	U									R
12			R	R									R
13			RU	RU									R
14			D	U									R
15	R	R	R	R	R	R	R	R	R			R	R
16	R		R		R	R	C	U	C	R	R	R	
17	R		R		R	R	U	U	U	R	R	R	
18	R		R		R	R	R	R	R	R	R	R	
19	R							R	U				
20							D	U	D				
21	R		R		R	R	R	U	R	R	R	R	
22	R		R		R	R	R	U	R	R	R	R	
23	R		R		R	R	R	R	R	R	R	R	
24											C		
25											U		

26											UD		
27											RU		
28											R		
29	R		R		C	U	C	U	CU		R	C	

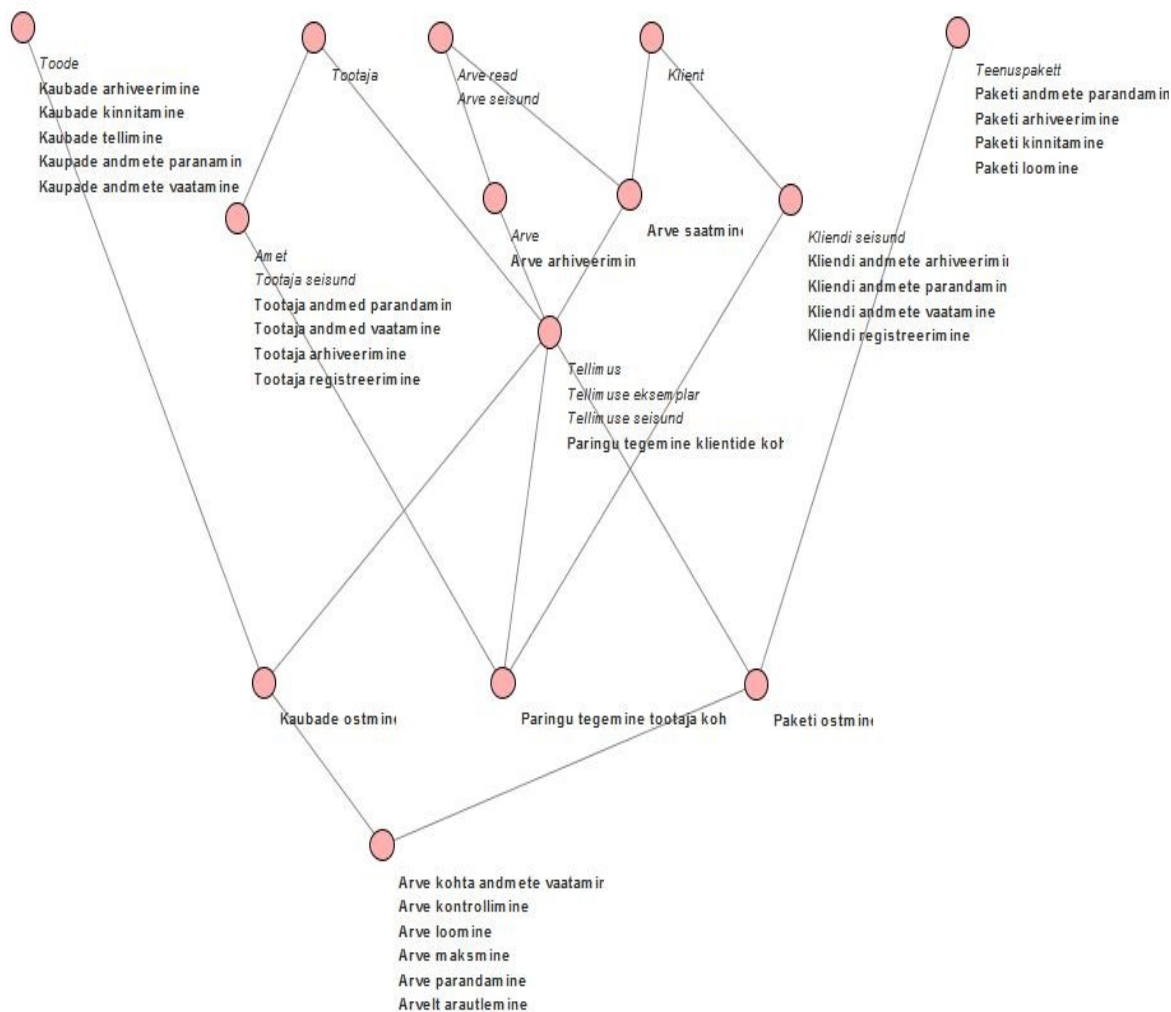
Tabel 3.1. 1 Tele2 Eesti AS. CRUD – maatriks

Kasutusjuhud:

1 - kliendi registreerimine 2 - kliendi andmete parandamine, 3 - kliendi andmete vaatamine, 4 - päringu tegemine klientide kohta, 5 - kliendi andmete arhiveerimine, 6 - paketi loomine, 7 - paketi kinnitamine, 8 - paketi andmete parandamine, 9 - paketi arhiveerimine, 10 - paketi ostmine, 11 - töötaja registreerimine, 12 - töötaja andmete vaatamine, 13 - töötaja andmete parandamine, 14 - töötaja arhiveerimine, 15 - päringu tegemine töötaja kohta, 16 - arve loomine, 17 - arve parandamine, 18 - arve kontrollimine, 19 - arve saatmine, 20 - arve arhiveerimine, 21 - arve maksmine, 22 - arvelt äraütlemine, 23 - arve kohta andmete vaatamine, 24 - kaupade tellimine, 25 - kaupade kinnitamine, 26 - kaupade arhiveerimine, 27 - kaupade andmete parandamine, 28 – kaupade andmete vaatamine, 29 - kaupade ostmine.

Objektid: a – klient, b - kliendi seisund, c – töötaja, d - töötaja seisund, e – tellimus, f - tellimuse seisund, g – arve, h - arve seisund, i - arve read, j – teenuse pakett, k – toode, l - tellimuse eksemplar, m – amet.

Antud CRUD maatriksi põhjal genereerime GSH diagrammi:



Joonis 3.1. 1 Tele2 Eesti AS. GSH diagramm (põhiobjekt/kasutusjuht)

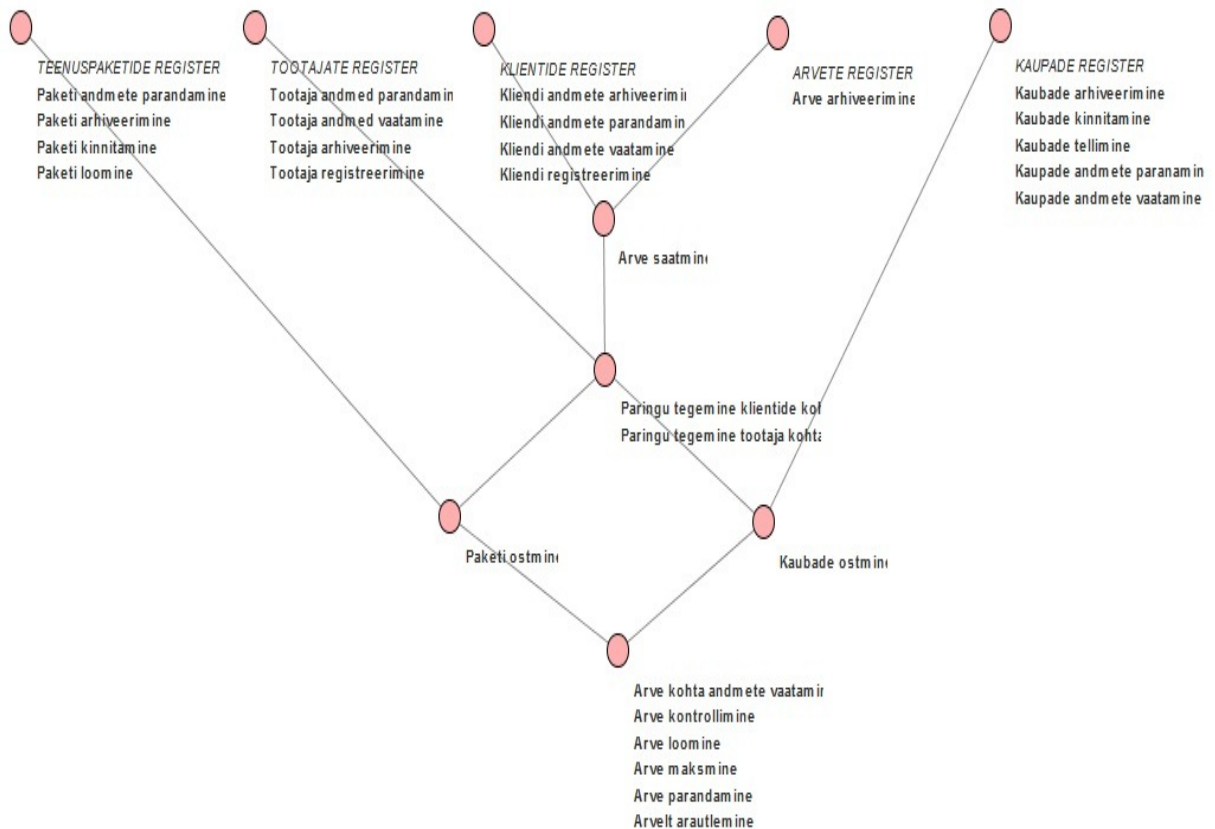
Saadud graafist on näha milliseid kasutusjuhte objektid jagavad. Need on kasutusjuhud: arve kohta andmete vaatamine, arve kontrollimine, arve loomine, arve maksmine, arve parandamine, arvelt äraütlemine. Jagatakse kõige rohkem järgmisi objekte: töötaja, toode, arve, arve read, arve seisund, klient, teenuse pakett, tellimus, tellimuse eksemplar, tellimuse seisund.

Antud graafil on sarnasusi projekti andmemudeliga (Lisa 1. “Tele2 Eesti AS” projekti andmebaas): igale objektile graafilt vastab üks tabel andmemudelist. Selle sarnasusega on lihtsam luua andmemudelit ja leida vigu või ebatäpsusi.

Antud GSH mudelit saab vaadelda nagu diagrammi, milles on olemas protsessid, andmed või tabelid ning seosed nende kahe vahel. Sellel juhul graaf annab informatsiooni sellest millistes infosüsteemi protsessides on vaja kasutada andmeid ühest või teisest tabelist. Näiteks, et

registreerida töötajat infosüsteemis tuleb võtta andmeid tabelitest “Töötaja”, “Amet” ja “Töötaja seisund”.

Järgmiseks vahetame põhiobjekte registrite peale, et saada suuremat vaadet infosüsteemist:



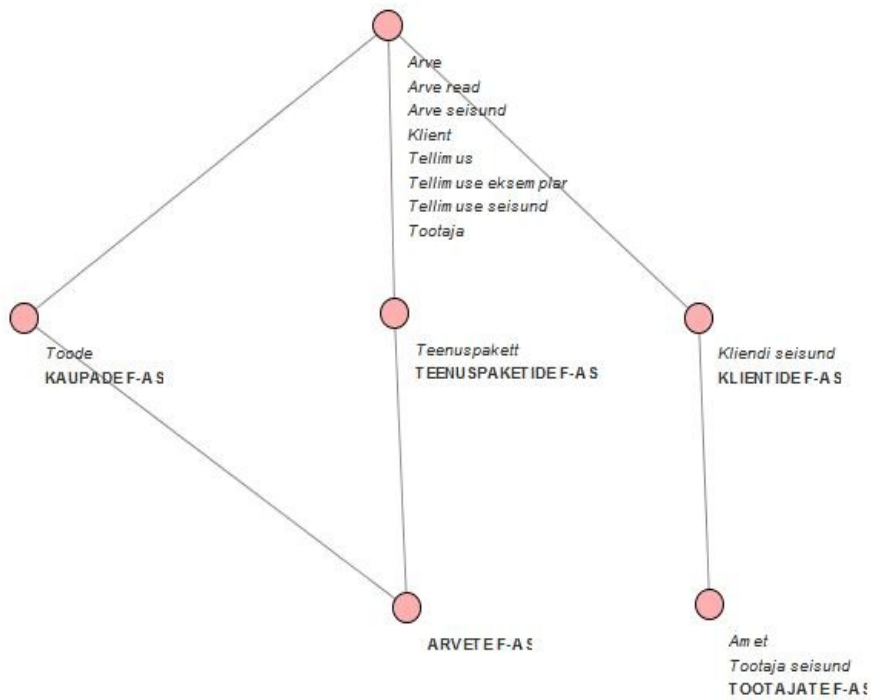
Joonis 3.1. 2 Tele2 Eesti AS. GSH diagramm (register/kasutusjuht)

Siit saab näha mis registre kasutavad antud kasutusjuhud. Näiteks, kasutusjuht “Paketi ostmine” kasutab teenuspakete registrit, töötajate registrit, klientide ning arvete registre.

Lisaks, graaf aitab leida funktsionaal-allüsteeme antud projektis, kuna need kasutavad registrite andmeid ja iga allüsteemi kuuluvad kasutusjuhud ehk protsessid. Selleks, et selgitada funktsionaalse allüsteemi, piisab võtta ühe kontsepti kuhu kuuluvad register ja kasutusjuhud: klientide register, kliendi andmete arhiveerimine, parandamine, vaatamine, kliendi registreerimine - antud kontseptist saab öelda, et antud projekti üheks funktsionaal allüsteemiks on klientide funktsionaal-allüsteem, mis teenindab klientide registrit.

Juhul, kui midagi juhtub mõne registriga (ei tööta), siis on teada mis protsessile see hakkab mõjuma. Ning vastupidi: kui mõned protsessid keelduvad töötada, siis saab kiiremini leida milles registris on viga.

Nüüd proovime vahetada kasutusjuhete funktsionaalsete allsüsteemide peale:



Joonis 3.1. 3 Tele2 Eesti AS. GSH diagramm (põhiobjekt/funktsionaal-allsüsteem)

Nagu me saame näha, mida vähem seoseid ja kontsepte on graafi peal, seda lihtsam on “lugeda”, aru saada: ei pea otsima mis kontseptile viib mõni seos ning ei pea meeles pidama atribuute või objekte.

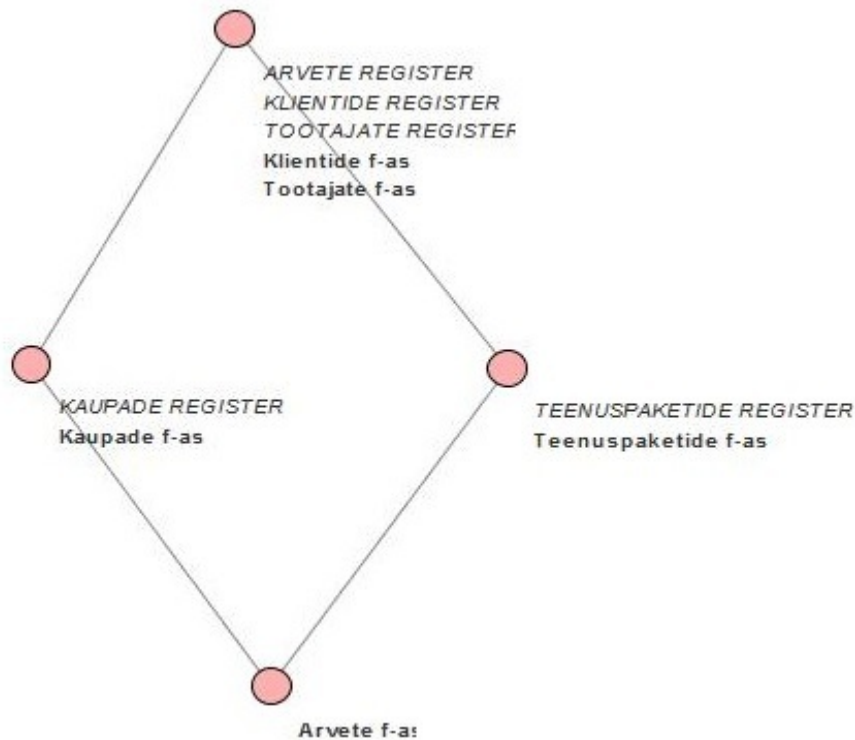
Genereeritud diagramm annab informatsiooni sellest - mis põhiobjekte kasutavad erinevad funktsionaal-allsüsteemid, ja muidugi milliseid nendest need jagavad, ehk, antud juhul, kõik allsüsteemid jagavad järgmisi põhiobjekte: arve, arve read, arve seisund, klient, tellimus, tellimuse eksemplar, tellimuse seisund, töötaja.

Lisaks sellele, juhul kui andmete sisestamisel tabelisse, mille põhjal genereeritakse GSH mudeli, funktsionaal-allsüsteemi nime sisestatakse rohkem kui ühe korra, siis seda oleks hästi

näha, juhul kui meie diagrammis pole palju seoseid. Sellel juhul saab vigu kiiresti parandada ja GSH meetodi kasutamine on kasulik analüüsis.

Ei saa tähelepanuta jätta ka seda, et antud mudel toob mitte vähem vajalikku informatsiooni tarkvaratehnikas. Täpsemalt: diagramm aitab nimetada ja organiseerida koodi struktuuri, kõige rohkem aga, objekt orienteeritud programmeerimises, kus klasside ierarhia on oluline. Antud juhul, igast põhiobjektist saab olla üks eraldi klass, nimelt kui on vaja kirjutada tarkvara klientide funktsionaal-allüsteemile, siis oleks vaja jagada koodi järgmiste klasside vahel: kliendi seisund, arve, arve read, arve seisund, klient, tellimus, tellimuse eksemplar, tellimuse seisund, töötaja.

Proovime saada diagrammi milles on registrid ja funktsionaal-allüsteemid:



Joonis 3.1. 4 Tele2 Eesti AS. GSH diagramm (register/funktsionaal-allüsteem)

Väga arusaadavalt on näidatud, mis registre jagavad funktsionaal-allüsteemid.

Juhul, kui tehakse muudatusi tabelite struktuuris või nende väljade omadustes või tüüpides, oleks vaja teada, millistes teistes funktsionaal-allüsteemides kasutatakse neid registre ja kuidas ühed või teised muudatused mõjutavad allüsteeme.

### 3.2 ARK

Koostas: Madis Peetersoo.

Järgmiseks projektiks on võetud ARK (autoregistrikeskuse) infosüsteem (Peetersoo, 2013). Projektis on vaatluse all see osa autoregistrist, mis tegeleb autodokumentide koostamise, haldamise ja väljastamisega ning juhilubade taotlemise ja väljastamisega. Kirjeldatav infosüsteem peab võimaldama jälgida autodega seonduvaid tehinguid ning autoomanike ja ARK töötajate varustamist neile olulise informatsiooniga.

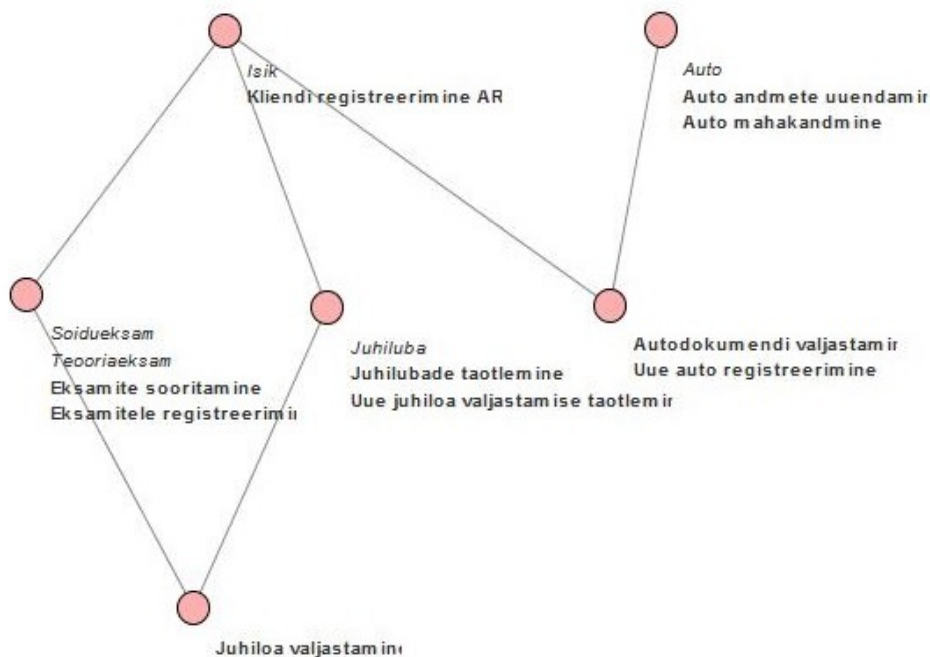
ARK CRUD – maatriks (Peetersoo, 2013) näeb järgmiselt:

<b>Objekt/ Kasutusjuht</b>	<b>Auto</b>	<b>Isik</b>	<b>Juhiluba</b>	<b>Sõidueksa m</b>	<b>Teooriaeks am</b>
<b>Autodokumendi väljastamine</b>	R	R			
<b>Uue auto regist- reerimine</b>	C	R			
<b>Auto andmete uuendamine</b>	U				
<b>Auto mahakandmine</b>	U				
<b>Kliendi registreerimine ARKi</b>		C			
<b>Juhilubade taotlemine</b>		R	C		
<b>Eksamitele registreerimine</b>		R		C	C,U
<b>Eksamite sooritamine</b>		R		U	U

<b>Uue juhiloa väljastamise taotlemine</b>		R	R		
<b>Juhiloa väljastamine</b>		R	U	R	R

Tabel 3.2. 1 ARK. CRUD – maatriks (Peetersoo, 2013)

ARK genereeritud GSH mudel:



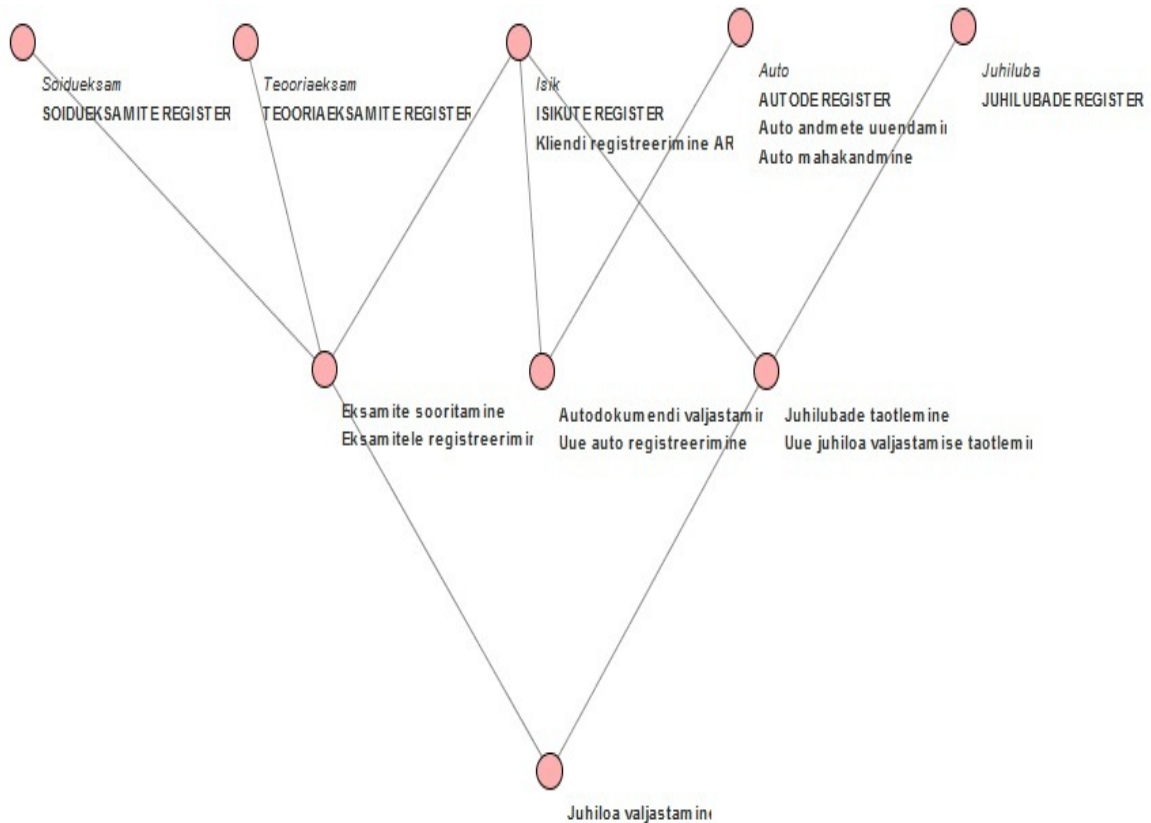
Joonis 3.2. 1 ARK. GSH diagramm (põhiobjekt/kasutusjuht)

Antud graaf annab võimalust valmistada (korjata) vajalikku informatsiooni andmemudeli ning pärast luua andmebaasi. Kui me võtame mis iganes kaks seotud teineteisega kontsepti ja kujutame nende põhiobjekte nagu tabelleid, siis sellest tuleb, et tabelid on seotud välisvõtmega (FK), mis tähendab, et ühes tabelis on väli primaarvõtmega ja teises – välisvõtmega, mis viidab vastavale primaarvõtmele, et unikaalselt identifitseerida seost nende tabelite vahel. Sellega on lihtsam leida nimesid väljadele ja planeerida sidet kõikide tabelite vahel. Näiteks, on olemas tabel “Isik”, ja sellega seotud kolm tabelit: “Juhiluba”, “Teooriaeksam”, “Sõidueksam”, siis võib pakkuda, et tabeli “Isik” primaarvõtmeks võib võtta välja “isik\_id” ja selle seotud tabelitel välisvõtmeteks vastavalt sama nimega väli. Muidugi, kui tabelleid on



palju rohkem, siis seda teha on keerulisem. Antud juhul GSH meetod sobib nende infosüsteemide jaoks mille andmebaas on väiksem keskmisest andmebaasist.

Selleks, et mitte genereerida kaks graafi, mis oleksid sarnased ainult kasutusjuhtude olekuga - proovime lisada eelnevalt loodud diagrammidele registrid:



Joonis 3.2. 2 ARK. GSH diagramm (põhiobjekt/register/kasutusjuht)

Antud näidest saab võtta vajalikku informatsiooni tarkvara kirjutamisel: nagu enne oli pakutud – võtame põhiobjekti klassiks, kasutusjuhu nagu meetodit klassis ja register saab olla üks andmebaasi osa, kus asub tabel meile vajalikke andmetega. Ja selle tabeli nimi on sama nagu objekti nimi. Diagrammi kasutamine lihtsustaks programmeerijal alustada tarkvara arendamist, kuna see mudel annab n.o. malli või baasi, mida saab arendada ja korrigeerida.

### 3.3 POOD.ELION.EE

Järgmiseks projektiks on pood.elion.ee infosüsteem, mille koostasid Ingrid Puusemp, Karli Oruste (Oruste, 2013).

Elion Ettevõtted AS on suurim ning kõige kauem tegutsenud telekommunikatsiooni ettevõtte Eestis. Elion on tegutsenud alates aastast 1993. Ettevõtte tegeleb kõnelahendustega eraisikutele kui ka ettevõtetele. Lisaks sellele see tegeleb andmesideteenustega.

E-pood jaguneb kahte peamisse ossa: kliendi- ja töötajaliideseks. Internetipood on klientidele toodetega tutvumiseks ja toodete tellimiseks läbi interneti. Admin kasutaja haldab tootevalikut. Klienditeenindaja tegeleb esitatud tellimustega.

Projektis vaadeldakse põhiliselt toote elutsüklit E-poes alates selle lisamisest kuni eemaldamiseni ja tellimuse täitmise protsessi, kuni selleni, mil klient on kauba kätte saanud ja tellimus täidetud.

Antud projekti CRUD maatriksi (Oruste, 2013) tabel:

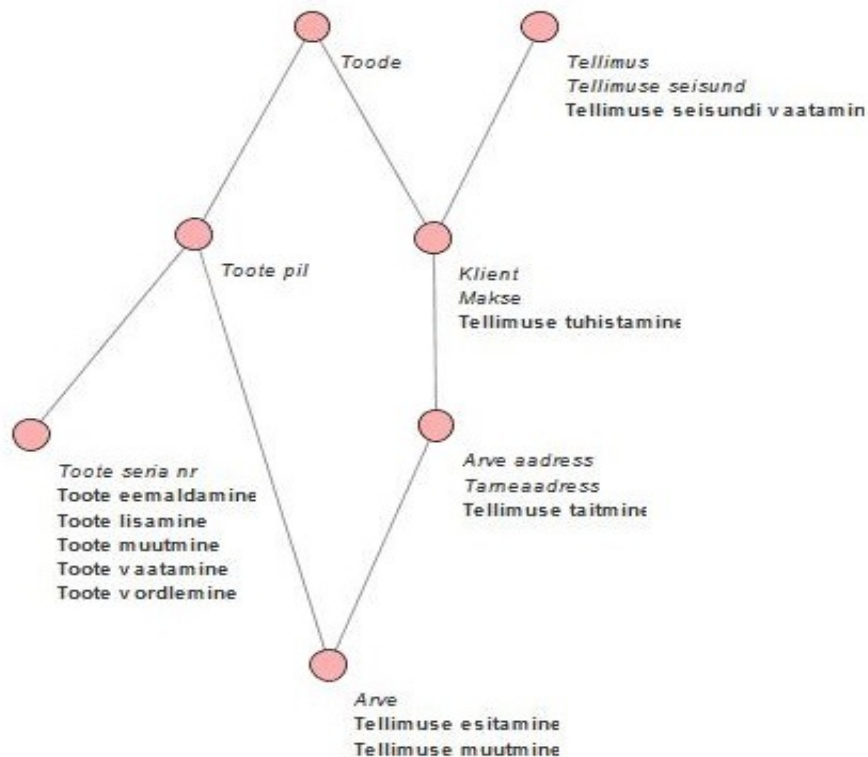
<b>Objekt / Kasutusjuht</b>	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	<b>h</b>	<b>i</b>	<b>j</b>
<b>Toote lisamine</b>			C	C	C					
<b>Toote eemaldamine</b>			D	D	D					
<b>Toote muutmine</b>			U	U	U					
<b>Toote vaatamine</b>			R	R	R					
<b>Toote võrdlemine</b>			R	R	R					
<b>Tellimuse täitmine</b>	R	RU	R			U	R		R	R
<b>Tellimuse tühistamine</b>	R	R	R			U	RU			
<b>Tellimuse esitamine</b>	CR	C	R	R		C	C	C	R	C

<b>Tellimuse seisundi vaatamine</b>		R				R				
<b>Tellimuse muutmine</b>	R	RU	R	R		U	R	U	R	U

Tabel 3.3. 1 POOD.ELION.EE . CRUD – maatriks

Objektid: a – klient, b – tellimus, c – toode, d - toote pilt, e - toote seeria nr, f - tellimuse seisund, g – makse, h – arve, i - arve aadress, j – tarneaadress.

Genereeritud GSH mudelit CRUD maatriksi baasil:

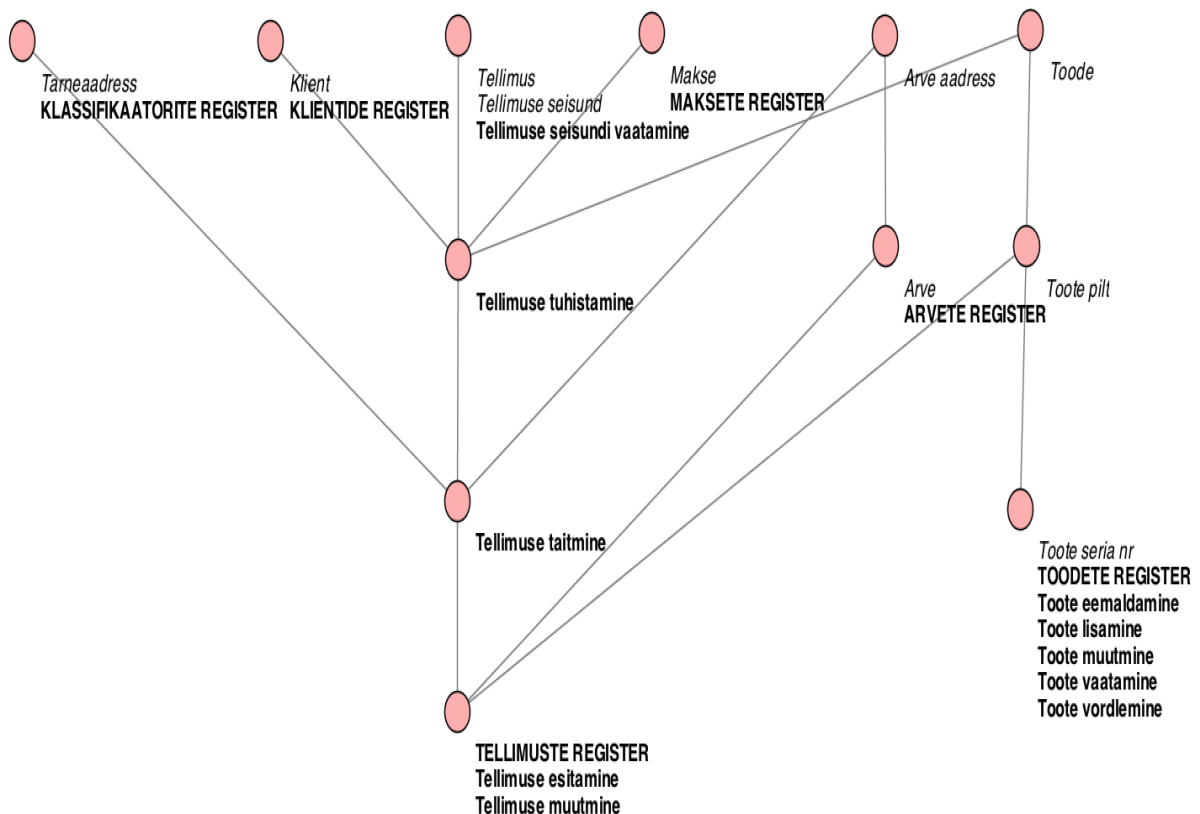


Joonis 3.3. 1 POOD.ELION.EE . GSH diagramm (põhiobjekt/kasutusjuht)

Diagramm tuli välja väga kompaktne ja arusaadav aga ei saa mainimata jätta seda, et see fakt, et meil puudub digrammis informatsioon sellest kuidas iga kasutusjuht saab andmeid kasutada: kas luua, lugeda, muuta või kustutada – on üheks puuduseks GSH mudeli jaoks.

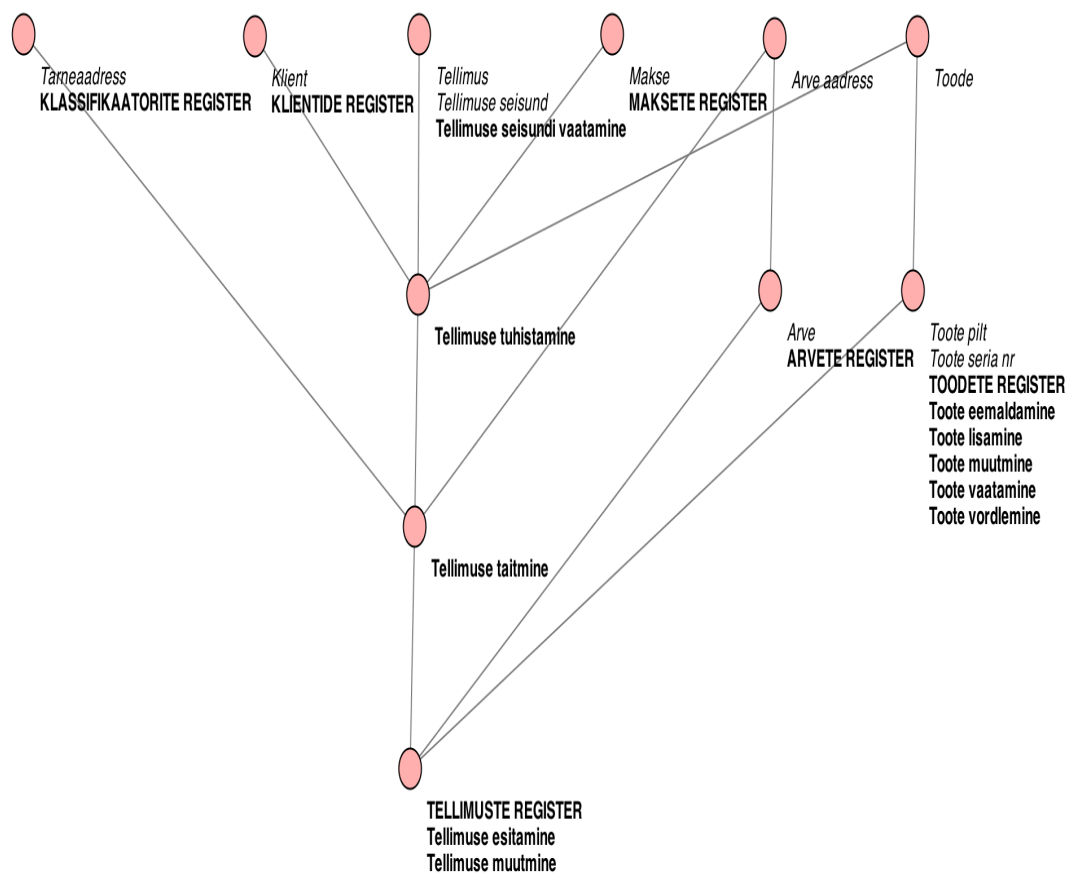
Lisaks, diagramm näitab ka millisele olulistele funktsionaalile, tabelile või lihtsalt infosüsteemi osale rohkem tähelepanu pöörata, sest juhul, kui süsteemis tuleb viga selles samas olulises ettevõtte jaoks kohas, siis see võib mõjuda büdžeedi peale, ressursidele mida tuleb kasutada, et lahendada probleemi. Üks viga funktsionaalis või tabelis võib ka peatada kogu infosüsteemi funktsionaalsust, sellist funktsiooni või tabelit nimetatakse üksikrikkeks (“single point of failure” , SPOF). Meie allsüsteemi jaoks kõige olulisemad protsessid on järgmised: Tellimuse tühistamine, Tellimuse täitmine, Tellimuse esitamine, Tellimuse muutmise – mis tähendab, et tuleks rohkem tähelepanu pöörata nende funktsioonide arendamisel, nende funktsioonidega seotud tabelite struktuuri peale ja muidugi turva küsimustele.

Lisame nüüd registre, mis pole kirjeldatud antud projektis aga nad annavad suuremat pilti :



Joonis 3.3. 2 POOD.ELION.EE . GSN diagramm 1 (põhiobjekt/register/kasutusjuht)

Saadud graafist on näha, et “Tellimuste register” asub diagrammi keskel ja sellel on sidet kõikide tabelitega peale üht “Toote serial nr”, mis on viga. Tabel “Toote serial nr” peab olema seotud “Tellimuste registriga” ja siis me saame õiget mudelit:



Joonis 3.3. 3 POOD.ELION.EE . GSH diagramm 2 (põhiobjekt/register/kasutusjuht)

Nüüd “Tellimuste register” on seotud kõikide tabelitega. See tähendab, et juhul, kui juhtub midagi seotud registritega, tabelitega või funktsioonidega – siis funktsioonid “Tellimuse esitamine” ja “Tellimuse muutmine” ei tööta ka. Ja teispidi, kui ei tööta “Tellimuste register”, siis kogu infosüsteem on kriitilises olukorras.

Kasutades antud mudelit saab kontrollida vigu esinemist infosüsteemis kahel viisil: esiteks kontrollida seost registri ja põhiobjekti vahel ning teine viis - kontrollida seost põhiobjekti ja kasutusjuhude vahel.

### 3.4 Coursera

Antud projekt on veebipõhise õppekeskkonna Coursera infosüsteemist, mille kirjutas Sarah Marion Mikk (Mikk, 2013).

Projektis vaadeldakse õppekeskkonna osa, mis on seotud kursuse loomise ning õppimisega. Infosüsteem peab lihtsustama hariduse andmist ning parandama selle kättesaadavust.

Õppekeskkonnaga teevad koostööd ülikoolid, mille õppejõud annavad Courseras kättesaadavaid kursusi. Ühe kursuse loomisega võib olla seotud mitu õppejõudu. Üks õppejõud võib anda mitut kursust. Kui kursus lõpeb eksamiga, kinnitab õppejõud tulemused. Infosüsteem võimaldab kasutajatel võtta osa soovitud kursustest. Õppijad võivad läbida korraga mitut kursust. Üht kursust võib õppida mitu korda.

CRUD – maatriksis (Mikk, 2013) suurem osa andmeid loetakse:

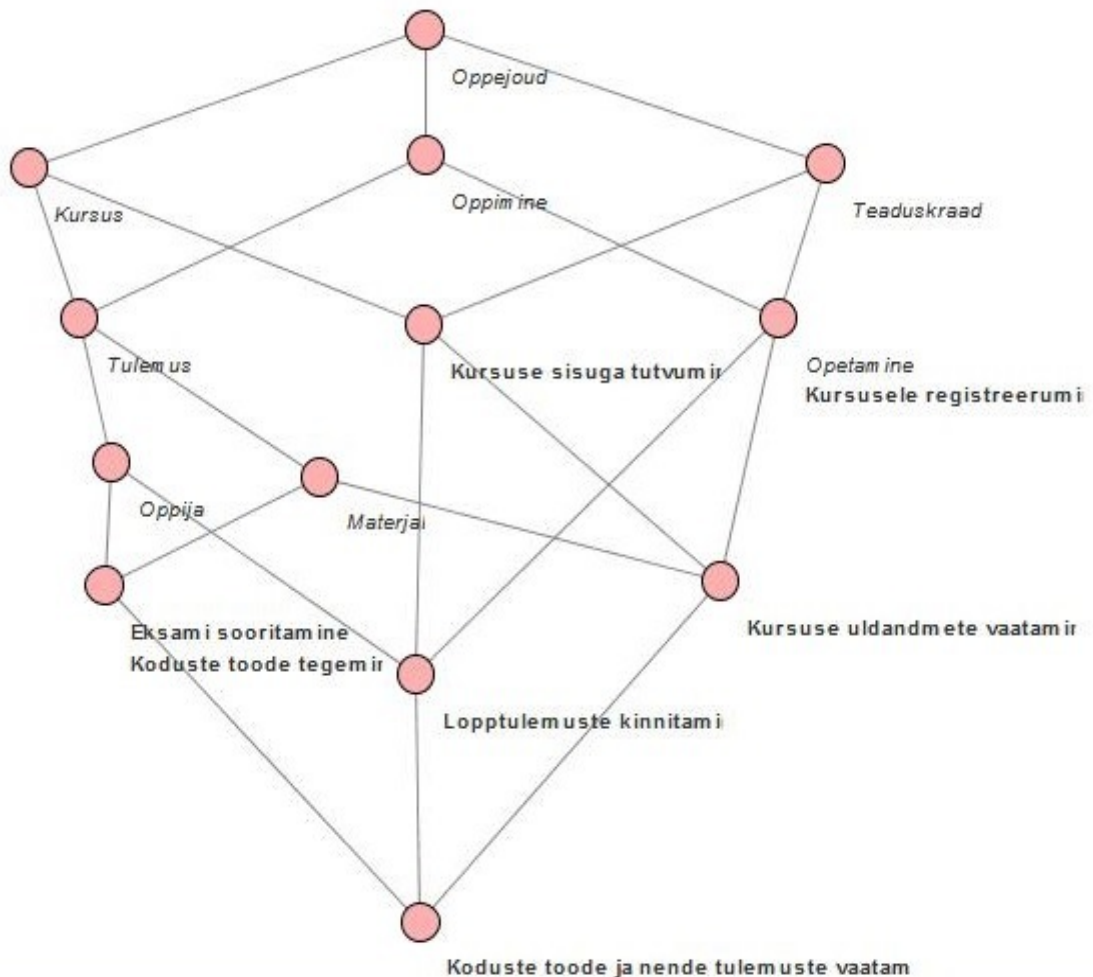
<b>Objekt/ Kasutusjuht</b>	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	<b>h</b>
<b>Kursuse sisuga tutvumine</b>	R		R					R
<b>Kursusele registreerumine</b>	R			C	R			R
<b>Koduste tööde tegemine</b>	R	R	R	R		R	C	
<b>Eksami sooritamine</b>	R	R	R	R		R	C	
<b>Koduste tööde ja nende tulemuste vaatamine</b>	R	R	R	R	R	R	R	R
<b>Läbitud kursuste tulemuste vaatamine</b>	R	R	R	R	R	R	R	R
<b>Kursuse üldandmete vaatamine</b>	R		R	R	R	R	R	R

<b>Lõpptulemuse kinnitamine</b>	R	R	R	R	R		U	R
---------------------------------	---	---	---	---	---	--	---	---

Tabel 3.4. 1 Coursera. CRUD – matriks

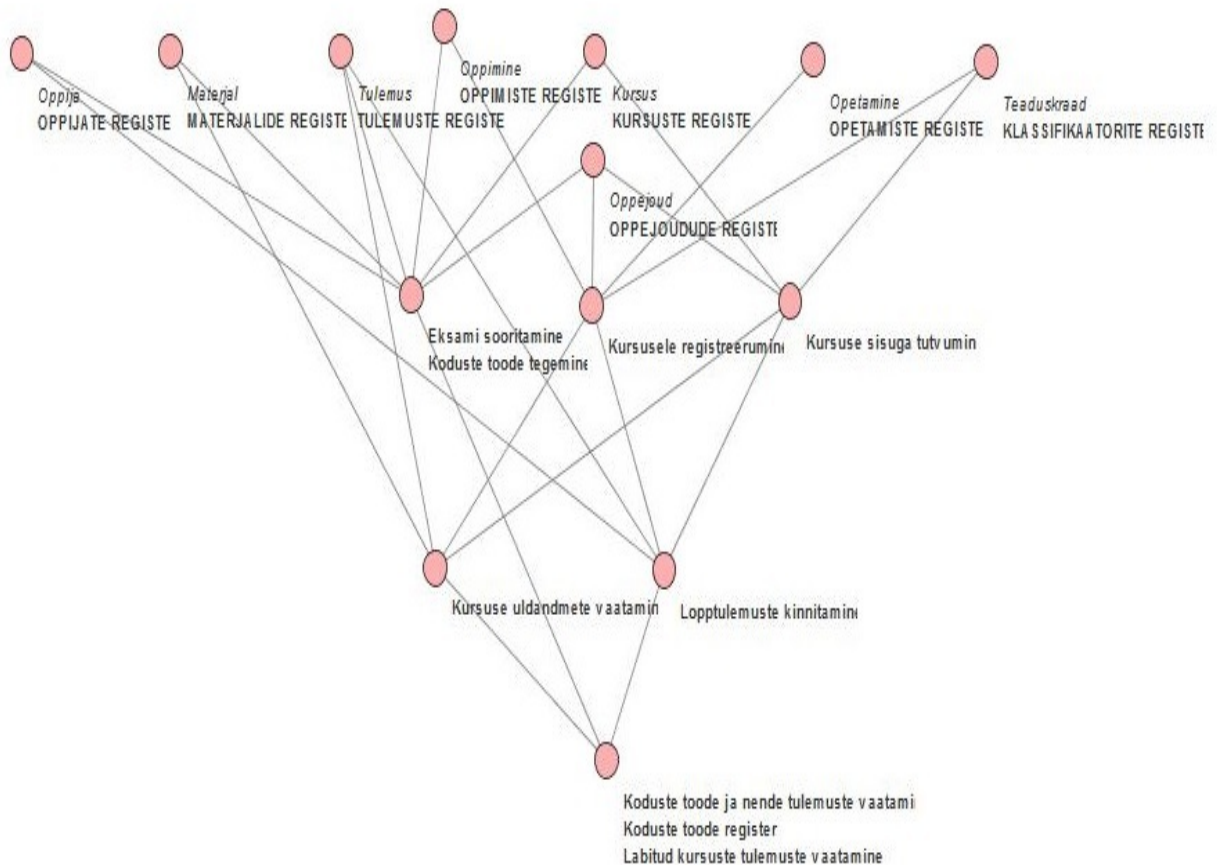
Objektid: a – õppejõud, b – õppija, c – kursus, d – õppimine, e – õpetamine, f – materjal, g - tulemus, h – teaduskraad.

On raske ühest pilgust aru saada antud graafist, kuna igal kontseptil on vähemalt kolm sidet teiste kontseptidega ja need seosed lähevad risti teine teisega, mis teeb antud mudelit keeruliseks navigeerimiseks ning kontsentreerimiseks ühele kontseptide ahelale. Puudub ka visuaalne grupeerimine.



Joonis 3.4. 1 Coursera. GSH diagramm (põhiobjekt/kasutusjuht)

Ja kui me lisame registreid eelnevale mudelile (Joonis 3.4.1 Coursera. GSH diagramm (põhiobjekt/kasutusjuht)), siis graaf läheb veel keerulisemaks (Joonis 3.4.2 Coursera. GSH diagramm (põhiobjekt/register/kasutusjuht)). Nüüd paljud kontseptid on seotud nelja kuni kuue kontseptidega. Nii võib ütelda, et ühe meetodi eelisest, milleks on visualiseerimine, me saame puudust. See fakt aga, et Galois' mudelis pole korduvaid atribuute ja objekte, ei tee antud mudeli vaadet liiga arusaamatuks ja mitte võimalikuks, et teha analüüsi selle põhjal.



Joonis 3.4. 2 Coursera. GSH diagramm (põhiobjekt/register/kasutusjuht)

### 3.5 Solaris Kino

Ja viimaseks vaatleme Solarisi Kino infosüsteemi, mille koostajad on Vladimir Potašenkov ja Stanislav Nazmutdinov.

Solarisi infosüsteem toetab kinos toimuvat äriprotsessi. Külalised saavad registreeruda et saata igasuguseid soodustusi sõltuvalt oma vanust. Kasutajad ja külalised ostavad piletit selle



infosüsteemi kaudu. Töötajad lisavad filmi, nad ka täiendavad info filmidest: kirjeldus, seansid, treilerid, reklaam. Töötajad liigitavad filmi kahte kategoriasse: “hetkel kinos”, “varsti kinos”, “peidetud”. Nad ka võivad filmi eemaldada. Saitil ka oma uudiste leht, kus avaldatakse igasugused uudiset seoted Solars kinoga. (Nazmutdinov, 2013)

Projekti CRUD maatriks (Nazmutdinov, 2013) :

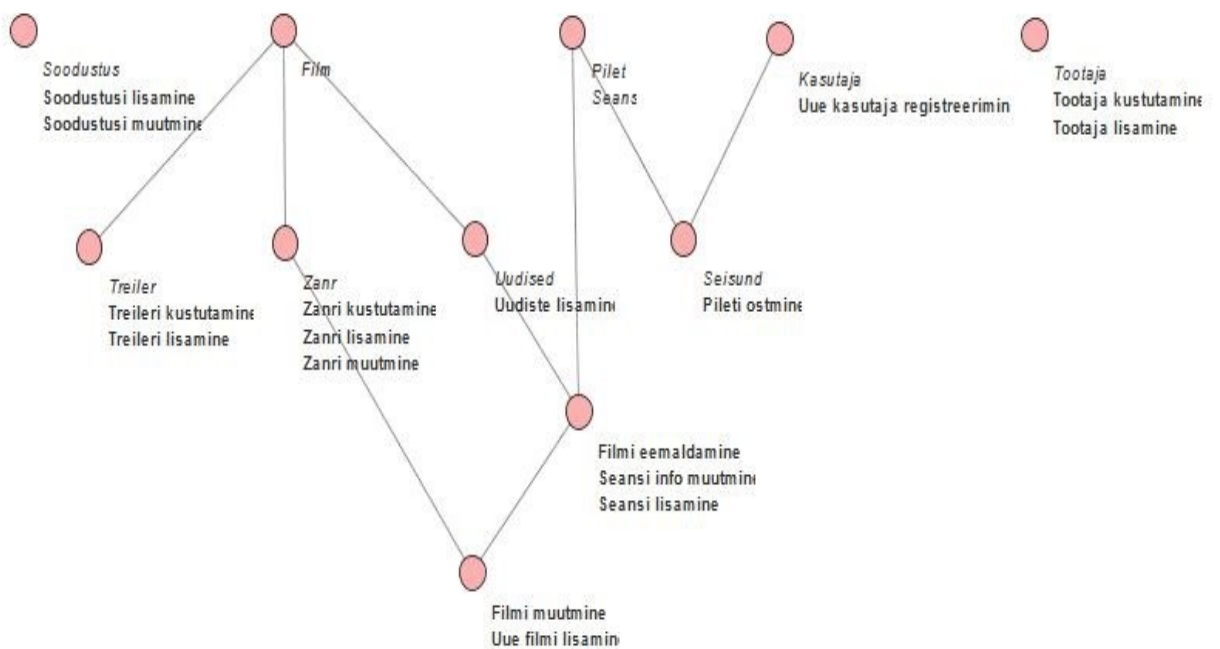
<b>Objekt/ Kasutusjuht</b>	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	<b>h</b>	<b>i</b>	<b>j</b>
<b>Uue filmi lisamine</b>			C	C	C	C				R
<b>Filmi eemaldamine</b>			D	D	D	C				
<b>Filmi muutmine</b>			U	U	U	C				R
<b>Uudiste lisamine</b>			R			C				
<b>Seansi info muutmine</b>			R	U	U	C				
<b>Seansi lisamine</b>			U	C	C	C				
<b>Uue kasutaja registreerimine</b>		C								
<b>Pileti ostmine</b>		CU		U	U				R	
<b>Treileri lisamine</b>			R				C			
<b>Treileri kustutamine</b>			R				D			
<b>Soodustusi lisamine</b>								C		

<b>Soodustusi muutmine</b>								U		
<b>Töötaja lisamine</b>	C									
<b>Töötaja kustutamine</b>	D									
<b>Zanri lisamine</b>			U							C
<b>Zanri muutmine</b>			U							U
<b>Zanri kustutamine</b>			U							D

Tabel 3.5. 1 Solaris Kino. CRUD – maatriks

Objektid: a – töötaja, b – kasutaja, c – film, d – pilet, e – seans, f – uudised, g – trailer, h – soodustus, i – seisund, j – žanr.

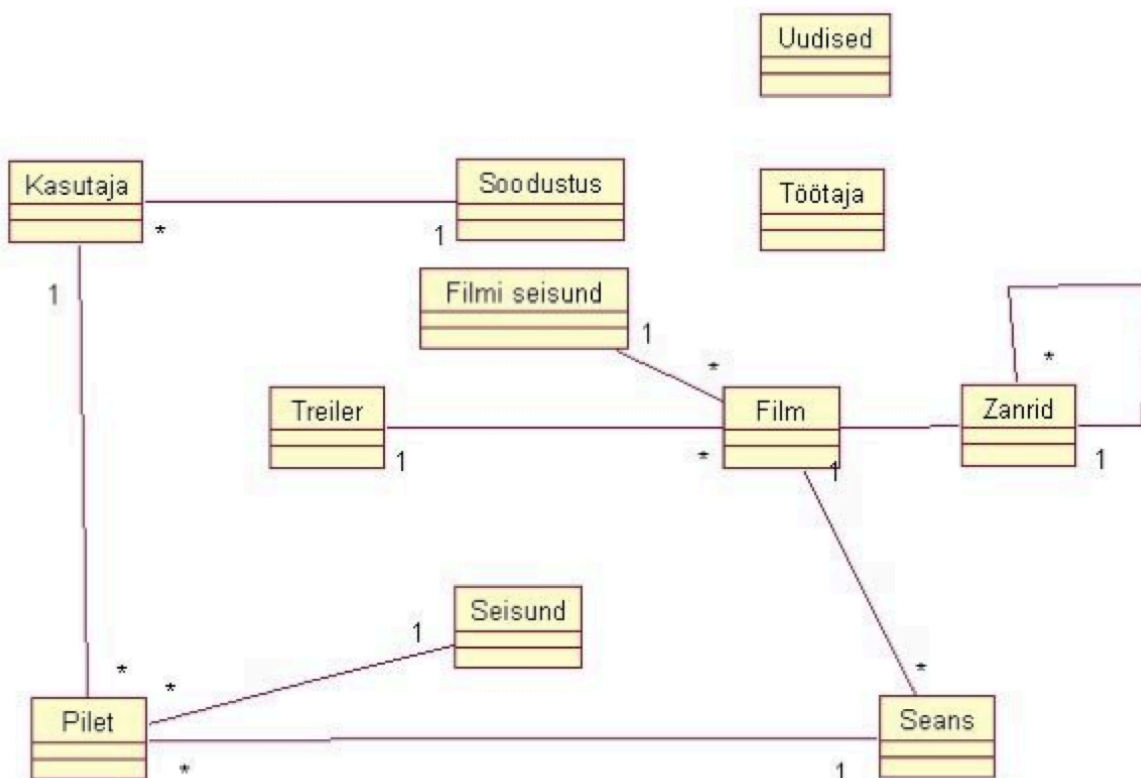
Genereeritud GSH mudel:



Joonis 3.5. 1 Solaris Kino. GSH diagramm (põhiobjekt/kasutusjuht)

Antud programmi väljund toob järgmist: on kohe näha, et tudengite töös esinevad vead. Üks viga on selles, et “Töötaja” objekt pole seotud mingi teise objektiga. “Töötaja” objekti on vaja seostada järgmiste kasutusjuhtudega: soodustusi lisamine, soodustusi muutmise, treileri lisamine, treileri kustutamine, žanri lisamine, žanri kustutamine, žanri muutmise, uudiste lisamine, seansi info muutmise, seansi lisamine, uue filmi lisamine.

Nüüd kui me teame millised vead olid CRUD maatriksis siis võib vaadata, mis toimub andmemudelil (Nazmutdinov, 2013):



Joonis 3.5. 2 Solaris Kino. Andmemudel

Esiteks, tabel “Uudised” Joonisel 3.5.2 (ehk. Joonis 3.5. 2 ) pole seotud mingi teise tabeliga aga Joonisel 3.5.1 sellel objektil on üks seos objektiga “Film” .

Ja teiseks, Joonisel 3.5.2 (ehk Joonis 3.5. 3 ) tabelil “Töötaja” pole mingit sidet teiste tabelitega kuigi sidet peavad olema.

## 4. GSH mudelite hinnang

Analüüsid infosüsteeme GSH meetodiga me kasutasime erinevaid objekte ja atribuute. Antud tabelis on toodud andmed iga projekti mudeli kohta: CRUD maatriksi seoste arv, GSH mudeli seoste arv, ühe GSH meetodi seose kirjelduse arv CRUD maatriksis, GSH mudeli lihtsus.

	Seosed CRUD maatriksis (arv)	Seosed GSH mudelis (arv)	Kui palju seoseid üks GSH seos kirjeldab CRUD maatriksis (arv)	GSH mudel on lihtsam (jah/ei)
<b>Tele2 Eesti AS</b>	132			
põhiobjekt/kasutusjuht		17	8	✓
register/kasutusjuht		10	13	✓
põhiobjekt/funktsionaal- allsüsteem		6	22	✓
register/funktsionaal- allsüsteem		4	33	✓
<b>ARK</b>	21			
põhiobjekt/kasutusjuht		6	22	✓
põhiobjekt/register/kasutusjuht		9	15	✓
<b><u>POOD.ELION.EE</u></b>	<b>47</b>			
põhiobjekt/kasutusjuht		7	19	✓
põhiobjekt/register/kasutusjuht		11	12	✓
põhiobjekt/register/kasutusjuht (parandatud)		10	13	✓
<b>Coursera</b>	49			
põhiobjekt/kasutusjuht		14	9	✓
põhiobjekt/register/kasutusjuht		22	6	✓
<b>Solaris Kino</b>	43			
põhiobjekt/kasutusjuht		9 (+ 2 eraldi kontsepti)	5	✓

Tabel 4. 1 GSH mudelite hindamine

Kui me võrdleme seoste arv CRUD maatriksis ja GSH mudelis, siis erinevus on suur. Võtame näiteks projekti “Tele2 Eesti AS” : CRUD maatriksi seoseid on 132, aga GSH mudeli

põhiobjekt/kasutusjuht – 17. Ja üks GSH seos kirjeldab 8 seost CRUD maatriksist, mis on hea tulemus. Aga kui me võtame GSH mudelit register/funktsionaal allsüsteem, siis seal on ainult 4 seost., mis tähendab, et üks GSH seos kirjeldab 33 seost CRUD maatriksist. Muidugi GSH mudel on lihtsam nii lugemiseks kui ka navigeerimiseks.

Võtame ARK projekti: CRUD maatriksis on 21 seos ja GSH mudel põhiobjekt/kasutusjuht omab 6 seost, ehk ühe GSH seose kohta 22 seost. Ja kui me vaatame GSH diagrammi põhiobjekt/register/kasutusjuht, siis seoste arv pole märkamalt suurem, kui mudelis ilma registrita.

Kõikidel projektidel GSH mudelites seoste arv on palju vähem, kui CRUD maatriksis, ja muidugi seoses sellega ka ühe GSH meetodi seosele kuulub päris palju seoseid CRUD maatriksist. GSH mudeli kasutamine on lihtne igal mudeli analüüsimise korral.

## 5. Tulemused ja järeldused

Rakkendades GSH meetodit infosüsteemide analüüsi jaoks olid leitud selle eelised ja üheks nende eelistest on muidugi visualiseerimine ja navigeerimine. Visualiseerimisel on lihtsam informatsiooni tajuda, kuna on näha tervikliku vaadet süsteemist. Navigeerimisel probleeme ka pole, sest informatsioon on gruppeeritud kontseptidega.

Atribuudid ja objektid ei kordu mudelis, mis on hea, kuna kordumine võib viia sellele, et diagramm läheb suuremaks ja pärast meil on liiga palju mitte vajalikku informatsiooni.

Teiseks eeliseks on see, et saab leida vigu teistes mudelites, mis aitavad teha analüüsi, näiteks andmemudelid, kontseptuaal ja kasutusjuhtude mudelites. See tähendab, et GSH diagramm võib siduda informatsiooni teiste mudelite vahel, mis viib sellele, et infosüsteem on loogiliselt kokku pandud, loodud.

On ka lihtsam luua andmemudelit, kuna iga objektile GSH diagrammist vastab tabel andmemudelidest. Ning on kiirem leida selles ebatäpsusi ja vigu. Tahaks lisada ka seda, et on lihtsam leida välisvõtit ja selle nime iga tabeli jaoks, kuna kõik tabelid on seotud välisvõtme kaudu ja muidugi me leiame primaarvõtme ja selle nimetust. Leides seda me saame loogiliselt kindlaks teha tabeli väljade tüüpe ning nende omadusi.

GSH diagramm aitab organiseerida koodi struktuuri. Me saame jagada tarkvara koodi klassideks, mille nimed me saame võtta diagrammist. Eriti see aitab objekt orienteeritud programmeerimises, kus objektide jagamine klassideks on oluline. Saame leida funktsioone, mis tuleb hoida igas klassis. Lisaks sellele, iga klassi funktsioonis me teeme päringut tabelile, mille nimi võrdub klassi nimele. Juhul, kui diagrammis me kasutame ka registreid (GSH mudelis, milles on seoseid põhiobjektide, registreid ja kasutusjuhtude vahel), siis me saame teha päringut mitte lihtsalt tabelile vaid tabelile, mis asub kindlas andmebaasi osas ja mille nimi võrdub registri nimele.

Igas infosüsteemis on kriitiline funktsioon või tabel ning juhul, kui tuleb viga sellega seotud tabeliga või funktsiooniga, siis see kriitiline funktsioon hakkab toimima ka valesti või lihtsalt lõpetab funktsioneerimist ja ettevõtte raiskab ressursse probleemi lahendusele ja võib olla

kaotab osa raha büdzeedist. GSH mudel näitab neid seoseid kriitiliste osadega infosüsteemis, mis aitab ennetada suurte probleemide tekimist.

GSH mudelite hindamise tabel näitas, et GSH mudelil on palju vähem seoseid võrreldes CRUD maatriksiga, mis tähendab, et GSH diagramm on palju kompaktsem.

Tasub tähelepanu pöörata GSH mudelitele, milles on nii põhiobjektid, registrid kui ka kasutusjuhud. Sellised diagrammid annavad täielikuma pildi infosüsteemist, võib ka öelda, et registrite lisamisel GSH mudelisse seosega põhiobjekt/kasutusjuht mudeli objektid ja atribuudid näevad rohkem gruppeerituna. Nagu oli varem mainitud, koodi struktuur on põhjalikum. Vigu infosüsteemis on lihtsam leida, kuna kontrollida seoseid saab kahel viisil: esiteks, saab kontrollida seoste õgust põhiobjektide ja kasutusjuhtude vahel; teiseks – põhiobjektide ja registride vahel.

Aga analüüsides tudengite projekte oli leitud ka puudusi selle meetodi kasutamises. Üheks miinuseks on see, et vaatamata sellele, et GSH meetodi kasutamine näitab meile suurt ja täis pildi infosüsteemist – visualiseerimine võib ka mitte anda head vaadet kontseptide seostele juhul, kui kontseptide ja nende seoste arv on suur ning kui infosüsteemi andmebaas on keskmisest andmebaasist suurem.

Ja teiseks puuduseks on see, et GSH diagramm ei anna meile seda informatsiooni, mis annab CRUD maatriks.

Sellest tuleb järeldus, et GSH meetodist on suurt kasu infosüsteemi analüüsimisel ja seda on lihtne kasutada ja leida vajalikku infot aga see tõttu, et GSH diagramm ei anna päris seda informatsiooni (mitte vähem olulist), mida annavad teised mudelid ja tabelid, mis aitavad analüüsida – antud meetod tuleb kasutada lisaks nende mudelitele ja tabelitele.

Selleks, et kasutada GSH meetodit on parem jagada suurt infosüsteemi allinfosüsteemideks selleks, et saada rohkem infot analüüsist.

## 6. Kokkuvõte

Eesmärgiks oli tõestada, et GSH meetod on hea viis infosüsteemi analüüsiks ja kasutades GSH diagrammi infosüsteemis on vähem loogilisi vigu, ebatäpsusi. Eesmärgiks oli ka näidata, et GSH meetodi kasutamine, ehk analüüsimine selle abil, on lihtne. Lisaks, GSH meetod aitab ennetada loogilisi vigu tulenevas infosüsteemis.

Projekti GSH meetodi analüüsimisel diagrammides vahetati põhiobjektid registriteks, kasutusjuhud funktsionaalsete allinfosüsteemidega. Pärast analüüsi selgus, et GSH diagrammi olulisemaks eeliseks on visualiseerimine ja navigeerimine. Lisaks, see aitab koodi struktuuri organiseerimisel ning aitab leida vigu teistes mudelites nagu andmemudel või kasutusjuhtude mudel. Aitab ka leida kriitilisi funktsioone selle põhjal kui palju on kontseptil seoseid ja millele viib antud kontsepti objektides või atribuutides vea tekkimine. GSH mudelid, milles on näidatud side põhiobjektide, registride ja kasutusjuhtude vahel, annavad kõige suuremat pilti infosüsteemist, aitavad leida vigu infosüsteemis kahel viisil ning kirjeldavad detailsemalt koodi struktuuri. Hinnates GSH meetodeid leidsime, et GSH diagramm on palju kompaktsem, kui CRUD maatriks ja sellel on suhteliselt vähem seoseid. Aga suur puudus on selles, et GSH meetod ei anna sama informatsiooni nagu CRUD maatriks, mis võiks anda rohkem olulist informatsiooni.

Pärast analüüsi tegemist oli kindlaks tehtud, et eesmärgi saavutati: GSH meetodi kasutamine on tõeliselt lihtne ja parem seda kasutada lisaks teiste analüüsiks mõeldud mudelitele ja tabelitele, sest üks meetod ei saa asendada teisi mudeleid; Oleks mugavam kasutada GSH diagrammi asendades kasutusjuhud funktsionaal-allüsteemideks, kuna sellel olukorral meil on vähem kontsepte ja nende seoseid teineteisega.

Tulevikus võiks edasi arendada tarkvara, mis genereerib GSH mudelit, et see näitaks ka CRUD maatriksi informatsiooni, kuna see on üheks puuduseks Galois' diagrammis. Ja võiks lisaks kasutada mitme mõõtmelist massiivi, et hoida rohkem seotud infot infosüsteemist.



## Summary

The aim of this work is to prove that GSH method is a good way to analyze information systems and using GSH diagrams in information systems will lead to less amount of logical issues and inaccuracies. The aim is also to show that using GSH method for analysis is easy. And also this method helps to forecast future infosystem crash because of logical mismatch between parts of that infosystem.

In students' projects' analysis using GSH method we tried using different data in the diagram so we could get as much information as possible about the information system. After analysis it was clear, that the main advantage of GSH diagram is visualization and navigation. In addition to that, it will help to organize the structure of a software code and also to find mistakes in other models like data model or use-case model. The method also helps to find a single point of failure, basing on the amount of concept's relations and on the greatness of a problem to which the failure will lead. GSH models, which have relations between main objects, parts of the database and use-case, give the biggest picture of information system, help to find mistakes in it in two ways and also to provide with more detailed structure of the software code. When we evaluated GSH methods then we have found out that GSH diagram is more compact than CRUD matrix and it has less connections. But the big disadvantage is that GSH diagram do not give the same information as CRUD – matrix, which could give more useful information.

After finishing with analysis we could say that using GSH is really easy and it is better to use it in addition to other models and tables which help to analyze information system, because one GSH method can not replace other models. It would be more convenient to use GSH diagram when we replace use-cases with functional sub-info systems, because in that case we will have less concepts and relations between them.

## Kasutatud kirjandus

Berry, H. M. (February 2005. a.). Efficiently Computing a Linear Extension of the Sub-hierarchy of a Concept Lattice. Lens, France: Springer-Verlag. Allikas: <http://sigayret.fr/rch/publis/05icfca.pdf>

Mikk, S. M. (2013). *COURSERA*. Tallinn: Tallinn University of Technology.

Nazmutdinov, V. P. (2013). *Solaris Kino*. Tallinn: Tallinn University of Technology.

Oruste, I. P. (2013). *POOD.ELION.EE*. Tallinn: Tallinn University of Technology. (Mikk, 2013)

Peetersoo, M. (2013). *ARK*. Tallinn: Tallinn University of Technology.

Raud, M. (2012). *Galois' alamhierarhia graafi genereerimise rakendus*. Tallinn University of Technology.

Skubi, D. B. (2013). *Tele2 Eesti AS*. Tallinn: Tallinn University of Technology.

Visnapu, J. (2013). *Analysis of using Galois' sub-hierarchies*. Tallinn: Tallinn University of Technology.

Wille, B. G. (1999). *Formal concept analysis — mathematical foundations*. Springer-Verlag .

