

**VILNIAUS UNIVERSITETAS  
KAUNO HUMANITARINIS FAKULTETAS**

**INFORMATIKOS KATEDRA**

VERSLO INFORMACIJOS SISTEMŲ STUDIJŲ PROGRAMA

Kodas 62103S138

SIMONA BENETYTĖ

MAGISTRO BAIGIAMASIS DARBAS

**TIKSLAIS GRINDŽIAMAS SISTEMOS PROJEKTAVIMAS**

Kaunas 2008

**VILNIAUS UNIVERSITETAS  
KAUNO HUMANITARINIS FAKULTETAS**

**INFORMATIKOS KATEDRA**

VERSLO INFORMACIJOS SISTEMŲ STUDIJŲ PROGRAMA

Kodas 62103S138

SIMONA BENETYTĖ

MAGISTRO BAIGIAMASIS DARBAS

**TIKSLAIS GRINDŽIAMAS SISTEMOS PROJEKTAVIMAS**

Leidžiama ginti \_\_\_\_\_

Magistrantas \_\_\_\_\_  
(parašas)

Darbo vadovas \_\_\_\_\_  
(parašas)

Prof. Saulius Gudas

\_\_\_\_\_  
(darbo vadovo mokslinis laipsnis, mokslo  
pedagoginis vardas, vardas ir pavardė)

Darbo įteikimo data \_\_\_\_\_

Registracijos Nr. \_\_\_\_\_

Kaunas 2008

# TURINYS

SANTRUMPŲ SĄRAŠAS.....	5
LENTELIŲ SĄRAŠAS.....	6
PAVEIKSLŲ SĄRAŠAS.....	7
SANTRAUKA .....	9
ĮVADAS.....	10
1. TEORINIS/ANALITINIS SKYRIUS.....	11
1.1. Tikslais orientuota reikalavimų inžinerija.....	11
1.1.1. EKD metodas .....	13
1.1.2. GQM metodas .....	15
1.1.3. GRL metodas.....	17
1.1.4. KAOS metodas.....	19
1.2. Tikslais grindžiamų požiūrių lyginamoji analizė .....	20
1.3. KAOS metodologija.....	23
1.3.1. KAOS metodologijos principai.....	23
1.3.2. KAOS modeliai .....	24
1.3.3. Tikslų modelis .....	26
1.3.4. Atsakomybių modelis.....	26
1.3.5. Objektų modelis .....	27
1.3.6. Operacijų modelis .....	28
1.3.7. KAOS pritaikymai.....	29
1.3.8. KAOS metodologijos privalumai ir trūkumai .....	30
1.4. DoDAF metodologija.....	31
1.4.1. DoDAF struktūra.....	31
1.4.2. DoDaf operacijų modelių architektūra.....	35
1.4.3. DoDaf sisteminių modelių architektūra .....	37
1.4.4. DoDaf techninių standartų sudarymo architektūra.....	39
1.4.5. DoDaf pagrindinių modelių sudarymo architektūra.....	39
1.4.6. DODAF modelių išrinkimas .....	40
1.5. Analizės išvados .....	46
2. SIŪLOMO SPRENDIMO METODIKA .....	47
2.1. Modelių sudarymo eiga.....	48
2.2. Tikslais grindžiamo metodo profiliai .....	50
2.2.1. Tikslų modelio profilis .....	50
2.2.2. Atsakomybių modelio profilis.....	51
2.2.3. Objektų modelio profilis .....	51
2.2.4. Operacijų modelio profilis .....	52
2.2.5. Operacijų mazgų modelio profilis.....	53
2.2.6. Sistemos sąsajų modelio profilis .....	53
2.3. Tikslais grindžiamo metodo profilių naudojimas.....	54
3. EKSPERIMENTINIS SKYRIUS .....	55
3.1. Objektams orientuotas gamybos planavimo sistemos projektavimas.....	55
3.1.1. Gamybos planavimo panaudojimo atvejų diagrama .....	56
3.1.2. Gamybos planavimo sistemos panaudojimo atvejų diagrama .....	57
3.1.3. Gamybos planavimo sistemos veiklos diagrama .....	57
3.1.4. Gamybos planavimo sistemos sekų diagrama.....	59

3.1.5. Gamybos planavimo sistemos klasių diagrama .....	60
3.1.6. Gamybos planavimo sistemos išdėstymo diagrama.....	61
3.2. Tikslais orientuotas gamybos planavimo sistemos projektavimas.....	62
3.2.1. Gamybos planavimo sistemos tikslų modelis .....	62
3.2.2. Gamybos planavimo sistemos atsakomybių modelis.....	63
3.2.3. Gamybos planavimo sistemos objektų modelis .....	64
3.2.4. Gamybos planavimo sistemos operacijų modelis .....	66
3.2.5. Gamybos planavimo sistemos operacijų mazgų modelis.....	67
3.2.6. Gamybos planavimo sistemos sąsajų modelis .....	68
3.3. Tikslais orientuoto ir objektais orientuoto metodų palyginimas .....	69
3.4. Papildyto KAOS metodo palyginimas .....	71
IŠVADOS.....	72
LITERATŪRA.....	73
PRIEDAS KAOS notacija .....	76

## SANTRUMPŲ SĄRAŠAS

DB – duomenų bazė

IS – informacinė sistema

IT – informacinės technologijos

KAOS - Knowledge Acquisition In Automated Specification

RE – Requirement Engineering

EKD - Enterprise Knowledge Development

GQM - Goal/Question/Metric

GRL - Goal Oriented Requirement Language

DoDaf - The Department of Defense Architecture Framework

GORE – Goal Oriented Requirements Engineering

## LENTELIŲ SĄRAŠAS

1 lentelė EKD modelių apžvalga.....	13
2 lentelė Tikslais grindžiamų požiūrių palyginimas.....	20
3 lentelė Tikslų požiūrių palyginimas .....	21
4 lentelė Tikslais grindžiamų požiūrių komponentų palyginimas .....	22
5 lentelė KAOS odontologija.....	25
6 lentelė KAOS projektų pavyzdžiai.....	29
7 lentelė Sumodeliuotų KAOS sąvokų skaičius sistemoms A-G.....	29
8 lentelė Darbo krūvio dalijimas pagal GORE.....	30
9 lentelė DoDaf architektūros produktai .....	33
10 lentelė Pagrindiniai DoDaf modeliai.....	40
11 lentelė DoDaf modelių įtraukimas .....	41
12 lentelė Modelių atitikmenys.....	48
13 lentelė Panaudojimo atvejų diagramos ir atsakomybių modelio palyginimas.....	69
14 lentelė Sekų diagramos ir operacijų modelio palyginimas.....	70
15 lentelė Klasių diagramos ir objektų modelio palyginimas .....	70
16 lentelė KAOS metodo ir papildyto KAOS metodo palyginimas.....	71

## PAVEIKSLŲ SĄRAŠAS

1 pav. Tikslais orientuoto požiūrio architektūra.....	11
2 pav. EKD architektūra.....	14
3 pav. GQM architektūra .....	16
4 pav. GQM metodo fazės .....	17
5 pav. GRL notacija .....	18
6 pav. GRL metodo architektūra.....	18
7 pav. KAOS modeliai pagal KAOS notaciją.....	19
9 pav. Tikslų modelis pagal KAOS notaciją.....	26
10 pav. Atsakomybių modelis pagal KAOS notaciją.....	27
11 pav. Objektų modelis pagal KAOS notaciją .....	27
12 pav. Operacijų modelis pagal KAOS notaciją.....	28
13 pav. KAOS aspektai .....	30
14 pav. DoDaf modelių sudarymo schema .....	35
15 pav. Operacijų modelių tarpusavio ryšiai.....	36
16 pav. Sisteminių modelių tarpusavio ryšiai .....	37
17 pav. Modeliai, kurie siejasi su standartais.....	39
18 pav. Pagrindiniai DoDaf modeliai.....	40
19 pav. Sistemos sąsajų modelis .....	43
20 pav. Operacijų mazgų modelis.....	43
21 pav. SV-1 ir OV-2 modeliai .....	44
22 pav. Papildytas KAOS metamodelis .....	44
23 pav. Papildyto KAOS metodo sudėtis.....	47
24 pav. Papildyto tikslais grindžiamo metodo modelių seka .....	49
25 pav. Tikslų modelio profilis .....	50
26 pav. Atsakomybių modelio profilis.....	51
27 pav. Objektų modelio profilis .....	51
28 pav. Operacijų modelio profilis.....	52
29 pav. Operacijų mazgų profilis .....	53
30 pav. Sistemos sąsajų modelio profilis .....	53
31 pav. Gamybos planavimo panaudojimo atvejų diagrama.....	56
32 pav. Gamybos planavimo sistemos panaudojimo atvejų diagrama.....	57
33 pav. Gamybos planavimo sistemos veiklos diagrama.....	58
34 pav. Gamybos planavimo sistemos sekų diagrama .....	59
35 pav. Gamybos planavimo sistemos klasių diagrama.....	60
36 pav. Gamybos planavimo sistemos išdėstymo diagrama .....	61
37 pav. Gamybos planavimo sistemos tikslų modelis.....	63
38 pav. Vadybininko atsakomybių modelis .....	63
39 pav. Laboranto atsakomybių modelis.....	64
40 pav. Gamybos vadovo atsakomybių modelis .....	64
41 pav. Gamybos planavimo sistemos objektų modelis be atributų.....	65

42 pav. Gamybos planavimo sistems objektų modelis su atributais .....	65
43 pav. Gamybos planavimo sistemos operacijų modelis .....	66
44 pav. Gamybos planavimo sistemos operacijų mazgų modelis .....	67
45 pav. Gamybos planavimo sistemos sąsajų modelis .....	68



## **SANTRAUKA**

BENETYTĖ, Simona. (2008) *Goal-Driven System Development*. MBA Graduation Paper. Kaunas: Vilnius University, Kaunas Faculty of Humanities, Department of Informatics. 72 p.

## **S U M M A R Y**

Goal-oriented requirement engineering has received much attention in recent years both by researchers and practitioners alike. In this paper I identify the uses of these approaches in analysis phases. The main object of the paper is to design goal-oriented method in order to model systems. That allowed to title Goal-Driven System Development. I was challenged to solve tasks and come to following outcomes.

Analysis of the main goal-driven methodologies was represented in the first part as well as their advantages and disadvantages were described by benchmarking four goal-driven methods. The best methodology was chosen for the further analysis. According to the distinguished disadvantages of DoDaf methodology was chosen in order to reduce the problems by modeling systems.

A new goal-driven method was created in the second part. was created a. In order to increase its usefulness it was represented in UML. The profile was introduced by Magicdraw UML tool.

The experiment was made in the third part in order to try the effectiveness of the new goal-driven method. Production plan system was modeled by using Magicdraw profiles. The example was compared by modeling according to the object-oriented method.

The results of the project. It is important to note that goal-oriented requirement engineering takes a wider system engineering perspective compared to the traditional requirement engineering methods.

The new goal-driven method make universal requirement engineering realization for creating information system.

The project consists of 72 pages, 45 pictures, 16 tables.

## IVADAS

Informacinės sistemos kūrimas yra pagrįstas reikalavimų analize ir apdorojimu. Be šio proceso sistemos kūrimas tampa sunkiai įmanomas. Šiuo metu yra įvairių reikalavimų inžinerijos metodologijų. Dauguma iš jų pagrįstos grafine tiriamų objektų ir jų elgsenos vizualizacija. Darbe nagrinėjamos tikslais grindžiamos reikalavimų inžinerijos. Jos yra pagrįstos sistemos tikslais ir skirtos kompiuterinės programinės įrangos reikalavimams apdoroti.

Pasirinkta magistrinio darbo tema – tikslais grindžiamas sistemos projektavimas. Ši tema aktuali tiems, kurie atlieka sistemos projektavimą pagal tikslais grindžiamus metodus.

Darbo objektas: tikslais grindžiamos metodologijos.

Darbo tikslas: papildyti tikslais grindžiamą metodą, skirtą sistemų projektavimui.

Siekiant įgyvendinti darbo tikslą, išskirti šie uždaviniai:

- Išanalizuoti magistrinio darbo objektą;
- Atlikti pasirinktų tikslais grindžiamų metodologijų analizę;
- Parinkti tinkamiausią metodą, skirtą tikslais grindžiamai reikalavimų inžinerijai;
- Išskirti pasirinkto tikslais grindžiamo metodo trūkumus;
- Atlikti DoDAF modelių analizę;
- KAOS metodą papildyti reikalingais DoDaf modeliais;
- Tikslais grindžiamo metodo naudojimui sukurti profilius su Magicdraw UML įrankiu;
- Patikrinti metodo efektyvumą, projektuojant UAB „Ventos polimeriniai gaminiai“ gamybos planavimo sistemą.

Darbe buvo naudoti tyrimo metodai: duomenų analizės metodas – analizuojami lentelių duomenys, kiti įvairių šaltinių duomenys; lyginamosios analizės metodas - lyginami ir vertinami produktai, panašūs bruožai, savybės; sintezės ir apibendrinimo metodai - apibendrinant informacijos šaltinius, skyrius ir straipsnius, pateikiamos išvados, sprendimai. Šių metodų pagalba buvo siekiama užsibrėžtų tikslų ir uždavinių įgyvendinimo.

Darbas susideda iš šių dalių: analitiniame skyriuje aprašyta, kas yra tikslais grindžiama inžinerija. Toliau darbe pateikta pasirinktų tikslais grindžiamų metodų analizė. Palyginus nagrinėjamas metodologijas, pasirenkama analizuoti tinkamiausia metodologija, kuri yra KAOS. Pateikta detali jos analizė, taip pat išskirti privalumai ir trūkumai. Remiantis trūkumais analizuojama DoDaf metodologija, kurios reikalingi modeliai įtraukiami į KAOS metodą. Antroje darbo dalyje pagal papildytą metodą sudaryti modelių profiliai su Magicdraw UML 12 įrankiu. Eksperimentinėje dalyje patikrinamas metodo efektyvumas, modeliuojant situaciją tradiciniu objektiškai orientuotu metodu ir naujuoju tikslais grindžiamu metodu.

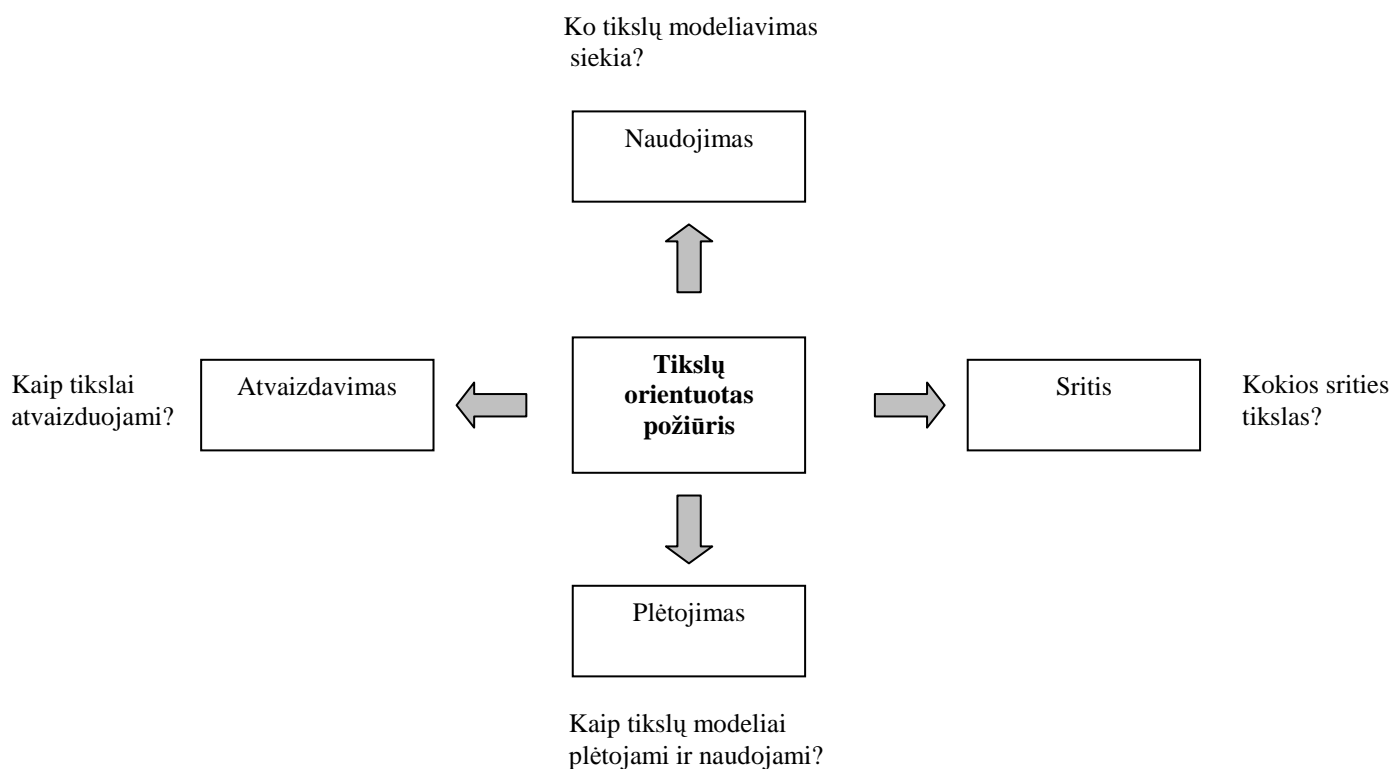
Darbą sudaro: 72 puslapiai, 16 lentelių, 45 paveikslai ir 1 priedas.

## 1. TEORINIS/ANALITINIS SKYRIUS

Tikslais grindžiamų požiūrių yra įvairių. Pirmiausia analitiniame skyriuje apžvelgsime, kas yra tikslais orientuota reikalavimų inžinerija, iš ko ji susideda. Sekantis etapas – pasirinktus tikslais grindžiamus požiūrius išanalizuosime detaliau, atliksime palyginamąją analizę. Pasirinktas tinkamiausias metodas bus analizuojamas ir taikomas mano darbe.

### 1.1. Tikslais orientuota reikalavimų inžinerija

Tikslais orientuotas projektavimas tapo svarbia technologija pirmaisiais IS vystymo etapais. Modeliais paremtas IS vystymas leidžia manyti, kad tikslų modeliavimas ateityje plėsis. Tikslais orientuotas požiūris, naudojamas reikalavimų inžinerijoje susideda iš kelių dalių: naudojimo požiūrio, atvaizdavimo požiūrio, srities požiūrio ir plėtojimo požiūrio (1 pav.).



1 pav. Tikslais orientuoto požiūrio architektūra

Šaltinis: Kavakli, E. (2003) <http://www.emmsad.org/2003/Final%20Copy/05.pdf>

Naudojimo požiūris apibūdina tikslų modeliavimą RE: esamos organizacijos situacijos suvokimą, pokyčių naudą suvokimą, atsargumo konteksto palaikymą RE, verslo tikslų ir sistemos komponentų surišimą, sistemos specifikacijos palaikymą prieš tarpininkus.

Srities požiūris žiūri į tikslą ir jo srities supratimą. Pradžioje, atrodo sudėtinga atskirti vienodą supratimą tikslų RE. Iš tikrųjų, tikslas buvo panaudotas skirtinguose požiūriuose, kad perduotų kelias reikšmes, įtraukiant užduotis, problemą, rezultatus. Tačiau, jei mes žiūrime į skirtingas tikslo sąvokas, galima režiūnuoti tris tipus: įmonės tikslai, proceso tikslai ir kokybės tikslai.

Smulkiau, kai tikslų modeliavimas yra naudojamas egzistuojančios organizacijos situacijos suvokimui, įmonės tikslai apibūdina tai, ką organizacija šiuo metu nori pasiekti. Panašus būdas yra, kuomet tikslų modeliavimas naudojamas apibūdinti, kaip verslo tikslai susieja funkcinius ir nefunkcinius sistemos komponentus, susitelkia ties būsimais įmonės tikslais, kurie apibrėžia pageidaujamą situaciją, kurią kiekvienas nori pasiekti ateityje.

Metodai, kurie naudoja tikslo analizę, kad suprastų pakeitimų poreikį, daugiausia naudojami proceso tikslų apibrėžimu. Proceso tikslai pažymi bet koki reikalavimą, svarstomą problemą, kuri bus aptarta, apskritai kažkas, kas gali iškilti per RE procesą, kažkas, kas gali veikti kaip RE proceso tikslas.

Kokybės tikslai reiškia tarpininkų kriterijus, prieš kuriuos sistemos specifikacija gali būti įvertinta. Kokybės tikslų gali būti dviejų tipų: RE rezultato įvertinimas (organizacinė sistema į apsvartymą) taip pat kaip RE proceso įvertinimas.

Atvaizdavimo metodas reiškia būdą, kuriuo tikslai yra išreikšti. Tikslai gali būti išreikšti įvairioje formatu, naudodami daugiau ar mažiau formaliai apibrėžto žymėjimo ženklus. Atskiriame tarp neoficialų, pusiau oficialų ir oficialų metodus

Neoficialūs metodai apskritai naudoja gimtosios kalbos tekstą, kad išreikštų tikslus; pusiau oficialus naudojimas daugiausia naudoja diagramas; pagaliau, oficialiuose metodų tiksluose yra išreikšti kaip logiški tvirtinimai kažkokia forma specifikacijos kalba.

Plėtojimo metodas domina būdą, kad tikslo modeliai yra sukurti ir plėtojasi (Kavakli, E. (2003) Goal Driven Requirements Engineering: Evaluation of Current Methods).

Tikslų modeliai yra geras būdas vartotojams valdyti reikalavimus. Tikslai siūlo tinkamą abstrakcijų lygmenį, leidžiantį sprendimo priėmėjams patvirtinti pasirinkimus, padarytus tarp alternatyvų ir siūlant kitas alternatyvas.

Tikslų modeliavimo privalumai:

- Tikslais orientuota reikalavimų inžinerija apima platesnę sistemos projektavimo perspektyvą lyginant su tradiciniais RE metodais: srities ypatybės nustatomos iš reikalavimų detalizavimo proceso.
- Tikslai pateikia tikslus kriterijus reikalavimų specifikacijos išbaigtumui. Specifikacija yra užbaigta, kuomet visi tikslai gali būti pasiekti.

- Tikslai pateikia tikslus kriterijus reikalavimų tinkamumui. Reikalavimas yra tinkamas, jei jo specifikacija yra panaudojama siekiant tam tikro tikslo.
- Tikslų medis jungia aukšto lygio strateginius tikslus su žemais techniniais reikalavimais.
- Tikslų modeliavimas pateikia paprastą mechanizmą, kaip suformuoti sudėtingus reikalavimų dokumentus.
- Viena iš RE problemų yra daugialypių požiūrių konfliktų valdymas. Tikslai gali būti naudojami, kad valdytų konfliktus tarp reikalavimų.

Tyrinėjimai parodė, kad tikslai yra daug stabilesni negu žemesnio lygio sąvokos, tokios kaip reikalavimai ar operacijos. (Lapouchnian, A. (2005) From Requirements to Architectural Design – Using Goals and Scenarios).

### 1.1.1. EKD metodas

EKD (Enterprise knowledge development) – tai tikslais grindžiama modeliavimo kalba, kuri susideda iš šešių modelių 1 lentelė). Kiekvienas jų turi savo specifinį organizacijos akcentą. Modeliavimo komponentai siejasi atitinkamais ryšiais.

1 lentelė

**EKD modelių apžvalga**

	<b>Tikslų modelis</b>	<b>Verslo taisyklių modelis</b>	<b>Sąvokų modelis</b>	<b>Verslo procesų modelis</b>	<b>Dalyvių ir šaltinių modelis</b>	<b>Techninių komponentų ir reikalavimų modelis</b>
<b>Akcentas</b>	Vizija ir strategija	Elgsena ir taisyklės	Verslo ontologija	Verslo operacijos	Organizacinė struktūra	Reikalavimai informacinei sistemai
<b>Problema</b>	Ką organizacija nori pasiekti ar išvengti ir kaip?	Kokios yra verslo taisyklės, kaip jos padeda organizacijos tikslams?	Kokios sąvokos yra naudojamos kituose modeliuose?	Kokie yra verslo procesai? Kaip jie valdo informaciją ar medžiagas?	Kas yra atsakingi už tikslus ir procesus? Kaip yra įtraukti dalyviai?	Kokie yra reikalavimai IS? Kai jie siejasi su kitais modeliais?
<b>Komponentai</b>	Tikslas, problema, išoriniai apribojimai, galimybės	Verslo taisyklė	Sąvoka, atributas	Procesas, išorinis procesas, informacijos rinkinys, medžiagų rinkinys	Dalyvis, rolė, organizacijos vienetas	IS tikslas, IS problema, IS reikalavimas, IS komponentas

Šaltinis: Stirna, J. (2007) <http://www.via-nova-architectura.org/files/EMMSAD2007/Stirna.pdf>

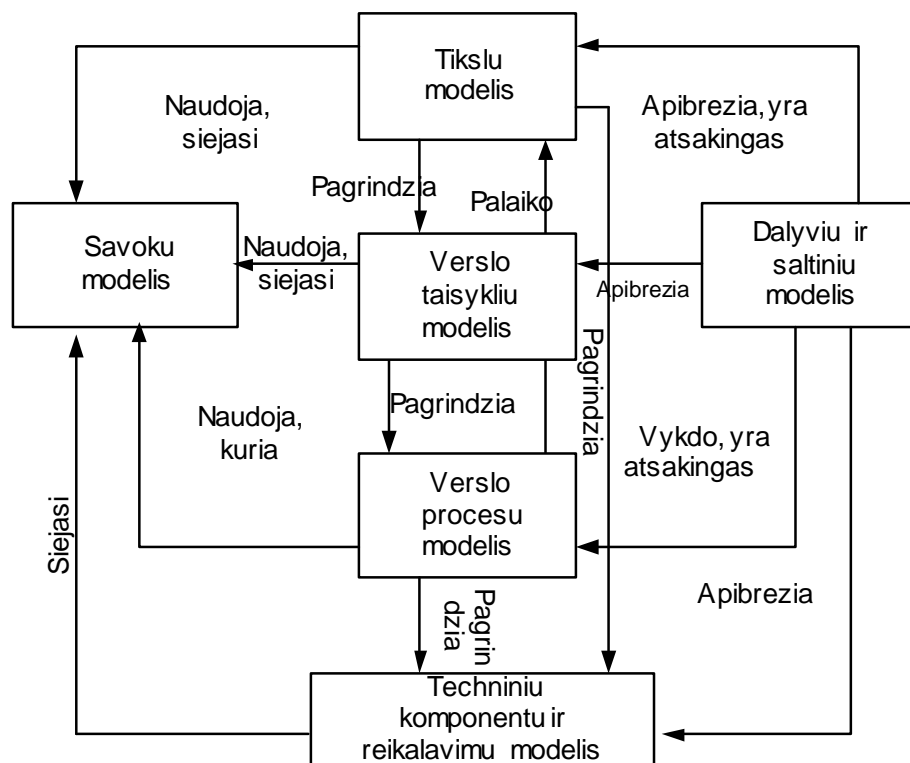
Gebėjimas surasti sprendimus, komponentus ir kitus aspektus priklauso ryšių supratimo. Pavyzdžiui, tikslų modelio komponentai turi būti daug aiškiau apibrėžti nei sąvokų modelyje. Tomet yra apbrėžiama sąsaja tarp tikslų modelių komponentų ir sąvokų modelio komponentų.

Tokiu pat būdu tikslai motyvuoja tam tikrus procesus verslo procesų modelyje. Procesai yra būtini, kad pasiektų išdėstytus tikslus. Tam ir reikalinga sąsaja tarp tikslo ir proceso. Bendras modelių vaizdas rodo problemos sprendimą (Stirna, J. (2007) Ten Years Plus with EKD: Reflections from Using an Enterprise Modeling Method in Practice).

Yra keletas požiūrių į EKD notaciją. Yra paprasta notacija, naudojama, kuomet nereikalaujama aukšto lygio formalumo, ir semantiškai turtingesnė notacija, kuomet reikalaujama aukšto lygio formalumo. Priklausomai nuo modeliavimo situacijos yra parenkama tinkama notacija.

Norint pasiekti aukštų rezultatų, modeliavimo procesas yra toks pat svarbus kaip ir modeliavimo kalba. Yra du proceso aspektai: dalyvavimo požiūris ir modelio vystymo procesas.

Visi modeliai atsako į klausimus: kodėl, kaip, ką ir kas. Nepaisant to, kad EKD turi savo modeliavimo kalbą, galima pakeisti kurio nors vieno modelio modeliavimą kalbą. EKD taip pat leidžia pridėti modelius. Šis bruožas yra naudingas, kuomet situacija yra tinkama EKD panaudojimui, bet turi specifinį modeliavimo poreikį, ko metodas negali pateikti. Modelių naudojimo supratimui, pateikiame EKD architektūrą (2 pav).



2 pav. EKD architektūra

Šaltinis: Stirna, J. (2007) [http://people.dsv.su.se/~js/ekp/ekd\\_method.html](http://people.dsv.su.se/~js/ekp/ekd_method.html)

1. Tikslų modelis apibūdina įmonės tikslus. Apibūdinama tai, ko įmonė ir jos darbuotojai nori pasiekti ar išvengti. Tikslų modeliai dažniausiai aiškina tokius klausimus: kur orgnazacija turi judėti, koks yra tikslų svarbumas, šių tikslų prioritetai, kaip tikslai yra susieti vieni su kitais, kokios problemos trukdo siekti tikslų.

2. Verslo taisyklių modelis naudojamas apibrėžti aiškiai suformuluotas verslo taisykles, kurios yra suderintos su tikslų modeliu. Verslo taisyklėmis galima vadinti tikslų ribas. Verslo taisyklių modelis paaiškina tokius klausimus: kurios taisyklės paveikia organizacijos tikslus, kaip verslo taisyklė siejasi su tikslu, kaip tikslai gali būti palaikomi tikslų.
3. Sąvokų modelis yra naudojamas griežtai apibrėžti daiktus ir reiškinius, naudojamus kituose modeliuose (įmonės sąvokos, požymiai ir ryšiai). Sąvokos yra naudojamos apibrėžti tikslų modelyje esančius posakius. Sąvokų modelis paprastai paaiškina tokius klausimus: kokios sąvokos yra pripažintos įmonėje (apima ryšius su tikslais, įvykius, procesus, dalyvius), kaip jie apibrėžia verslo taisykles ir apriboja objektus.
4. Verslo procesų modelis reikalingas apibrėžti įmonės procesams. Verslo procesų modelis yra panašus į tradicinius duomenų struktūrų modelius. Šis modelis paaiškina tokius klausimus: kurios veiklos ir procesai yra pripažinti įmonėje, ar jie turi būti ten. Kaip verslo procesai ir t.t. turi būti vykdomi?
5. Dalyvių ir šaltinių modelis paaiškina, kaip aktoriai ir šaltiniai yra susieti vieni su kitais, ir kaip jie yra susieti su tikslų modelio komponentais, pvz.: ar aktorius gali būti atsakingas už tam tikrą procesą verslo procesų modelyje, ar aktorius gali įvykdyti tam tikrą tikslą tikslų modelyje. Aktorių ir šaltinių modelis paaiškina tokius klausimus: kas turi valdyti ir vykdyti tam tikrus procesus ir užduotis? Kokia yra atsakomybių struktūra tarp aktorių?
6. Techninių komponentų ir reikalavimų modelis. Dėmesys yra sutelktas ties technine sistema, kuri yra būtina, kad palaikytų tikslus, procesus ir įmonės aktorius. Pradžioje kiekvienas turi išvystyti aukšto lygio reikalavimus ir tikslus informacijos sistemai. Jais grindžiant, yra kuriama informacinė sistema iš posistemų ar techninių komponentų. Šis modelis paaiškina tokius klausimus: kokie yra reikalavimai informacijos sistemai? Kurie reikalavimai yra sukurti verslo procesų? (Stirna, J. (2007) Ten Years Plus with EKD: Reflections from Using an Enterprise Modeling Method in Practice)

EKD metodas gali būti naudojamas dėl daugelio priežasčių: dėl verslo vystymo, dėl informacijos pasidalijimo apie verslą, jo viziją, kaip jis valdomas, dėl verslo procesų užtikrinimo. Verslo plėtra yra vienas svarbiausių tikslų. Verslo plėtra dažnai apima pakeitimų vadybą – kaip pasiekti vizijas ir tikslus esamai organizacijai. Verslo proceso orientacija yra konkretus verslo plėtros atvejis – organizacija nori pertvarkyti/perprojektuoti savo verslo operacijas.

### 1.1.2. GQM metodas

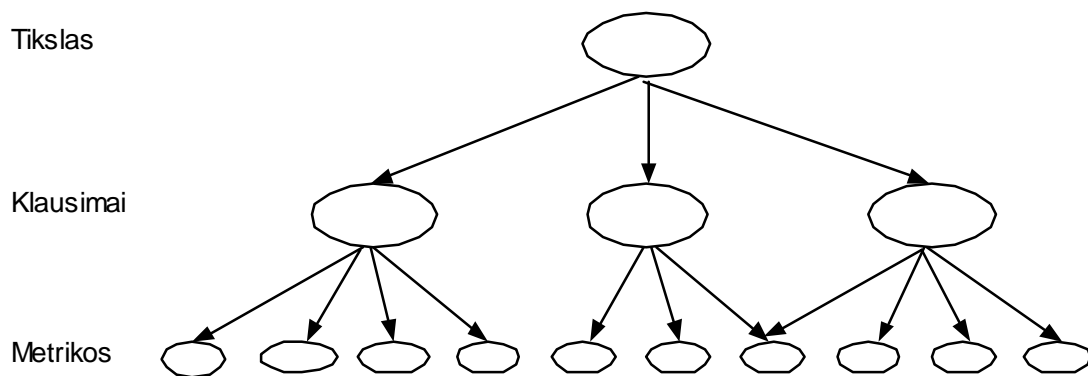
Sėkmingas programinės įrangos vystymas yra neįmanomas, nežinant, kokie yra gerinimo tikslai ir kaip jie siejasi su verslo tikslais. Pats sudėtingiausias klausimas yra, kaip apibrėžti tam

tikrą gerinimo tikslą, išreiškiant jį verslo tikslais. Pagerinimo tikslai turi būti identifikuoti. Šiuo metu programinės įrangos proceso pagerinimas yra vienas svarbiausių tikslų.

GQM metodas – tai tikslo/klausimo/metrikos metodas (Goal/Question/Metric/), kuris yra taikomas praktiškai. (Software Measurement Services (2004) The GQM Method)

Kad pagerinti tam tikrą procesą, reikia apibrėžti tikslus, kuris išskaidomas į klausimus ir į metrikas, kurios talpina savyje visą reikalingą informaciją, atsakant į tuos klausimus. GQM metodas pateikia planą, turintį problemų ir taisyklių rinkinius duomenų interpretacijai. Interpretacija duoda atsakymą, jeigu projekto tikslai yra pasiekti. GQM metodas turi struktūrą iš trijų žingsnių (3 pav.).

Tam, kad nustatytume, ar sutikome būtent tą tikslą, yra užduodami klausimai, kurių atsakymai pasakys, ar tikslai yra pasiekti. Tačiau yra daug situacijų, kuomet veiksmai gali būti svarbūs, nepaisant to, kad tikslai nėra aiškiai apibrėžti. Jeigu mažas skaičius metrikų adresuoja į skirtingus tikslus, tuomet reikia pasirinkti tinkamiausią.



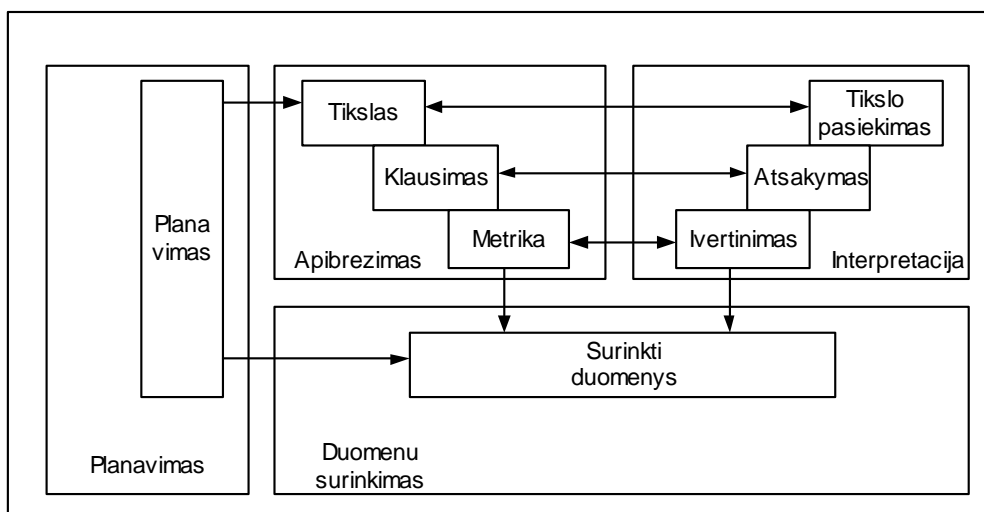
**3 pav. GQM architektūra**

Šaltinis: Solingen, R. (1999) <http://ivs.cs.uni-magdeburg.de/sw-eng/us/java/GQM/link1.shtml>

GQM metodas turi keturias fazes (4 pav.):

1. Planavimo fazė. Jo metu projektas yra išrenkamas, apibrėžiamas, charakterizuojamas, suplanuojamas;
2. Apibrėžimo fazė. Jos metu yra apibrėžiama programa (tikslas, klausimai, metrika ir hipotezės) ir dokumentuojama;
3. Duomenų surinkimo fazė. Jos metu vyksta duomenų surinkimas.
4. Interpretacijos fazė. Jos metu surinkti duomenys yra apdirbami, kurie aprūpina atsakymus apibrėžtiems klausimams, po kurio gali būti įvertinatas tikslo siekimas.





**4 pav. GQM metodo fazės**

Šaltinis: Solingen, R. (1999) <http://ivs.cs.uni-magdeburg.de/sw-eng/us/java/GQM/link1.shtml>

Tikslų nustatymui yra naudojami klausimai, kurie palengvina tikslų sudarymo procesą. Pateikiame klausimus, į kuriuos reikia atsižvelgti.

1. Kokie yra patys svarbiausi tikslai?
2. Kas gali padaryti įtaką šiems tikslams?
3. Pagalvokite apie būdus, kaip galima pagerinti jūsų vaidmenį ir apie problemas, kurios gali atsirasti.
4. Pagalvokite apie tikslų pagerinimą.
5. Pagalvokite apie tinkamus tikslus ir pasirinkite juos aukščiausiu prioritetu.

Abstraktesniam vaizdui galima paaudoti lentelę, kuri leis labiau suvokti GQM metodą:

- Ką? - Vystymo objektas
- Kodėl? - Supratimas, kontrolė, objekto gerinimas
- Kokių aspektų? - Objekto dėmesio centras yra ten, kur yra nustatymas
- Kas? - Žmonės, kurie nustato objektą
- Kontekstas - Aplinka, kurioje vyksta nustatymas

GQM metodas apima kokybės tikslus, o jų atvaizdavimas yra neformalus. Šis metodas naudojamų atributų turi nedaug. Pagrindinis yra tikslas. Šis metodas apima reikalavimų tvirtinimą.

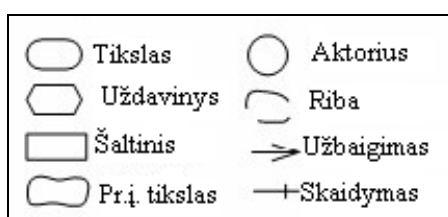
### 1.1.3. GRL metodas

GRL – tai tikslais grindžiama reikalavimų kalba (Goal Oriented Requirement Language). Ši metodologija palaiko tikslų modeliavimą ir reikalavimų apibūdinimą, ypač nefunkcinius reikalavimus. Apima tris konceptų kategorijas: iš anksto apgalvotus elementus, nuorodas ir aktorius. Elementai yra šie: tikslas, sisteminis tikslas, šaltinis, uždavinys. Jie yra iš anksto apgalvoti, nes jie

yra naudojami modeliams, kurie reikalauja atsakymų į tokius klausimus: kodėl tam tikri struktūriniai ir informaciniai aspektai yra įtraukiami į sistemos reikalavimus, kokios alternatyvos buvo diskutuotos, kokie kriterijai buvo naudojami alternatyvoms parinkti, dėl kokių priežasčių buvo parinktos alternatyvos prieš kitas (Liu, L. (2002) From Requirements to Architectural Design – Using Goals and Scenarios).

GRL metodologijos esmė – suvokti esamą sistemos situaciją. GRL turi grafinę ir tekstinę notaciją. Aptarsime tikrai grafinę notaciją bendram vaizdui susidaryti.

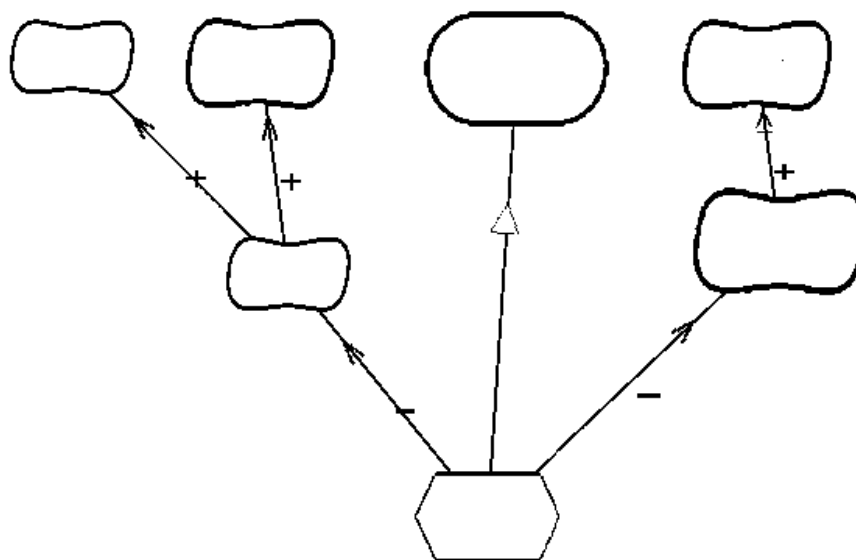
GRL metodo notacija nėra didelė (5 pav.):



5 pav. GRL notacija

Šaltinis: sudaryta autoriaus pagal Cares, C. (2005) <http://www.lsi.upc.edu/~cayala/Papers/2005-AOSDM-camera-ready.pdf>

GRL notaciją sudaro elementai: tikslas, uždavinys, šaltinis, programinės įrangos tikslas, aktorius. Ryšiai: užbaigimas, skaidymas, kontribucija, koreliacija, priklausomybė.



6 pav. GRL metodo architektūra

Šaltinis: sudaryta autoriaus pagal Liu, L. (2002) [http://www.cs.toronto.edu/~liu/publications/From\\_R2A-revised.pdf](http://www.cs.toronto.edu/~liu/publications/From_R2A-revised.pdf)

GRL metodo principas nėra sudėtingas (6 pav.). GRL metodas apima nefunkcijius reikalavimus. Šis metodas apima tikslais ir agentais grindžiamą modeliavimą. GRL apima reikalavimų išgavimą.

### 1.1.4. KAOS metodas

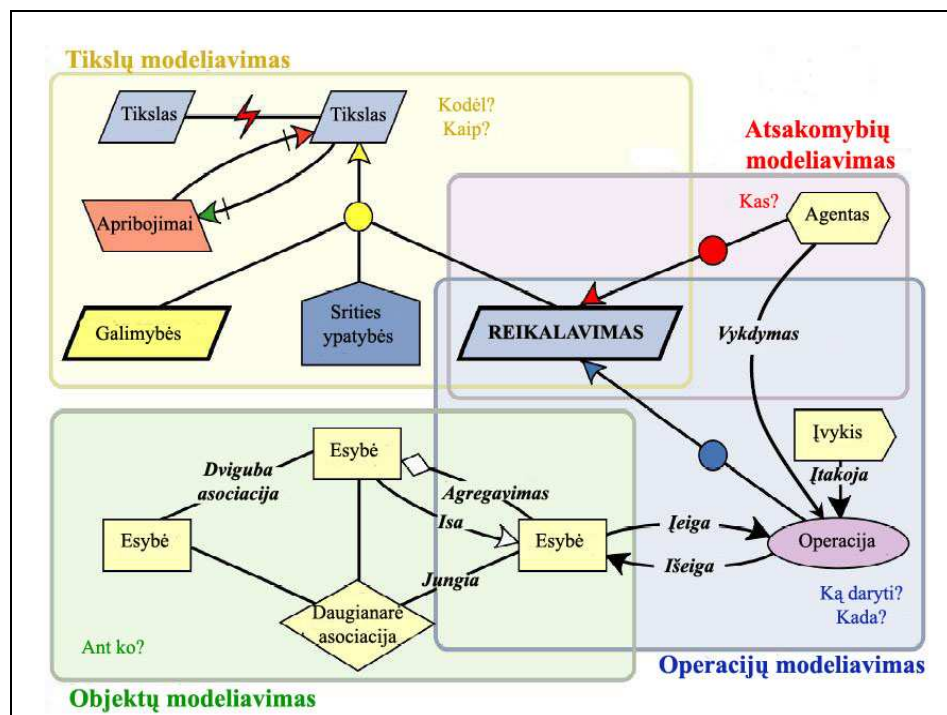
KAOS (Knowledge Acquisition In Automated Specification) - tai informacinių sistemų kūrimo reikalavimų inžinerijos metodologija, pagrįsta kuriamos sistemos tikslais.

KAOS modelio pagrindinė idėja ir tikslas – užtikrinti visapusišką reikalavimų inžinerijos realizaciją kuriamai informacinei sistemai.

KAOS reikalavimai sudėliojami pagal keturis modelius:

1. Tikslų modelis. Tikslai gali būti sistemos ar aplinkos. Tikslas apibūdina problemą, kurią norimą išspręsti. Tikslai yra dėliojami nuo aukšto lygio strateginių tikslų iki techninių tikslų, žemo lygio reikalavimų. Konfliktai tarp tikslų, neleidžiančių juos pasiekti, taip pat atvaizduojami modelyje.
2. Objektų modelis. Šis modelis naudojamas terminologiją, leidžiančią spręsti problemą.
3. Atsakomybių modelis. Parodoma, kas yra atsakingi už tikslus.
4. Operacijų modelis. Jis apibūdina, kaip agentai turi elgtis, siekiant tikslų.

KAOS yra sėkmingai naudojamas pramonės ir paslaugų sferoje. Jis yra naudojamas reikalavimų dokumentų sudarymui, apibrėžti strategijas ir įtraukti jas į IT planus, atlikti egzistuojančios sistemos reikalavimų reinžineriją. KAOS modeliai ir jų tarpusavio ryšiai yra pavaizduoti 7 paveiksle.



7 pav. KAOS modeliai pagal KAOS notaciją

Šaltinis: Objectiver (2004) <http://www.objectiver.com/download/documents/leaflet.pdf>

KAOS modeliai sudaromi pateikta seka ir talpinami į KAOS reikalavimų dokumentą, kartu su probleminiu sakiniu, reikalavimų surinkimo metodų aprašymu bei kiekvieno modelio trumpu aprašymu, pateikiant informaciją, kuri negali būti išskaityta iš grafinių modelių.

## 1.2. Tikslais grindžiamų požiūrių lyginamoji analizė

Yra įvairių tikslais grindžiamų požiūrių, kurių kiekvienas turi savo išskirtines savybes (2 lentelė). Požiūriai yra lyginami bendru požiūriu. Detaliau keletą iš jų jau išanalizavome darbe.

2 lentelė

**Tikslais grindžiamų požiūrių palyginimas**

Tikslais orientuoti požiūriai									
Architektūros komponentai		EKD	ISAC	SIBYL	REMAP	KAOS	GBRAM	GSN	GQM
Naudojimas	Supranta esamos organizacijos situaciją	√							
	Supranta pokyčių naudą		√						
	Palaiko atsargumo kontekstą RE viduje			√	√				
	Suriša verslo tikslus su sistemos komponentais					√	√		
	Įvertina sistemos specifikaciją prieš tarpininkų tikslus							√	√
Sritis	Įmonės tikslai	√	√		√	√	√		
	Procesų tikslai		√	√					
	Kokybės tikslai							√	√
Atvaizdavimas	Formalus	√			√	√			
	Pusiau-formalus			√			√		
	Neformalus		√					√	√

Šaltinis: sudaryta autoriaus pagal Kavakli, E. (2003) <http://www.emmsad.org/2003/Final%20Copy/05.pdf>

Tikslo analizės naudojimas pasiūlo naudą ir pranašumus beveik kiekvienoje stadijoje, įtrauktoje į RE procesą (2 lentelė). Šis platus pritaikymas tikslo sąvokų RE mano, kad išaiškinimas ir tikslų manipuliavimas yra natūrali ir neatskiriama dalis kurti RE, nepaisant to, kad ankstyvesni RE metodai nepadarė to aiškaus jungtinio palaikymo. Ši interpretacija yra tikėtina, kadangi reikalavimai atstovauja tikslui, kuris bus pasiektas, noras, kuris bus įvykdytas, vizija, kuri bus realizuota. Vis dėlto skirtingos struktūros papildo viena kitą, tokiu būdu sudedant įvairius metodus, galima gauti stipresnę struktūrą. Mažiau metodų pabrėžia poreikį modeliuoti proceso tikslus, kurie

valdo RE procesą, kol du metodai (GSN ir GQM) palaiko sistemos įvertinimą. Beveik niekas iš metodų (išskyrus ISAC, kuris kreipiasi ir į organizaciją ir į proceso tikslus) nesiekia suprasti tarpusavio priklausomybės tarp įvairaus supratimo tikslo lauko viduje (Kavakli, E. (2003) *Goal Driven Requirements Engineering: Evaluation of Current Methods*).

Tolesniai analizei buvo pasirinkti šie metodai: EKD metodas, GQM metodas, GRL metodas, KAOS metodas.

Pagrindinių svarbiausių tikslais grindžiamų metodų apžvalga padaryta. Trumpai išanalizuoti EKD, GQM, GRL, KAOS požiūriai. Kiekvienas metodas savaip pateikia modeliavimą (3 lentelė).

3 lentelė

### Tikslų požiūrių palyginimas

RE veikla	Tikslų modeliavimo apibūdinimas	Tikslų modeliavimo požiūris
Reikalavimų išgavimas	1. Supranta esamą organizacijos situaciją 2. Supranta poreikį keisti	GRL, EKD
Reikalavimų detalizavimas	3. Susieja verslo tikslus su funkciniais ir nefunkciniais komponentais	KAOS
Reikalavimų patvirtinimas	4. Pagrindžia sistemos specifikaciją prieš tarpininkų tikslus	GQM

Šaltinis: sudaryta autoriaus pagal Kavakli, E. (2003) <http://www.emmsad.org/2003/Final%20Copy/05.pdf>

Nagrinėti požiūriai pateikia skirtingas RE veiklas. GRL ir EKD metodai gilinaisi į reikalavimų išgavimą. Abu požiūriai analizuoja esamą organizacijos situaciją, akcentuojasi į įmonės tikslus. EKD apima aukšto lygio tikslus, o detalizacijos nedaro. Todėl EKD nėra efektyvus modeliavimas, lyginant su kitais nagrinėtais požiūriais. EKD labiau yra naudojamas verslo plėtrai. GRL metodas neturi reikalavimų detalizavimo, tik jų išgavimą. GRL metodas puikiai nusako esamą organizacijos situaciją. Modeliavimo atžvilgiu turi tikslų ir agentų požiūrių modeliavimą, tačiau trūksta objektų modeliavimo, architektūrinio dizaino. GQM metodas nėra formalus modeliavimas. Jo esmė yra išsiaiškinti kurie yra tinkami tikslai. Jis neturi modelių, tiesiog vyksta neformalus tikslų, klausimų ir metrikų nubraižymas. KAOS turi daugiausia modelių, kurie yra konkretesni, aprašo funkcinius ir nefunkcinius reikalavimus.

Palyginimui surašysime, kokius komponentus turi nagrinėjami metodai (4 lentelė).

## Tikslais grindžiamų požiūrių komponentų palyginimas

GRL		EKD		KAOS		GQM	
Reikalavimų išgavimas				Reikalavimų detalizavimas		Reikalavimų patvirtinimas	
Aktorius	+	Aktorius	+	Agentas	+		
Tikslas	+	Tikslas	+	Tikslas	+	Tikslas	+
Uždavinys	+	Procesas	+	Operacija	+		
Šaltinis	+	Šaltinis	+	Įvykis	+		
		Sąvoka	+				

Šaltinis: sudaryta autoriaus

Visus nagrinėjamus metodus jungia bendras komponentas – tikslas (4 lentelė). Aišku, tie tikslai kai kuriuose požiūriuose gali būti detalizuojami, tačiau šiuo atveju parodėme bendrą metodų sudėtį. EKD metodas turi daugiausia komponentų, nes jam priklauso penki modeliai. GQM metodas nėra formalus, todėl jį lyginti gana sudėtinga.

Tikslais grindžiamam sistemos projektavimui tinkamiausias yra KAOS metodas. Jis turi reikalavimų detalizavimą. Tolesniam darbui pasirenkame šį požiūrį. Toliau pateiksime detalią KAOS analizę, išsiaiškinsime trūkumus ir privalumus, papildysime metodą trūkstamais modeliais.

### 1.3. KAOS metodologija

KAOS (Knowledge Acquisition In Automated Specification) - tai informacinių sistemų kūrimo reikalavimų inžinerijos metodologija, pagrįsta kuriamos sistemos tikslais.

Gali būti ir kitoks paaiškinimas pavadinimo: KAOS = Keep All Objectives Satisfied

#### 1.3.1. KAOS metodologijos principai

KAOS modelio pagrindinė idėja ir tikslas – užtikrinti visapusišką reikalavimų inžinerijos realizaciją kuriamai informacinei sistemai. Ši metodologija buvo sukurta šiems tikslams:

- Išspręsti problemos aprašymo ir manipuliavimo šiuo aprašymu problemą;
- Pagerinti problemos analizės procesą naudojant sisteminių požiūrį surenkant ir struktūrizuojant reikalavimus;
- Atskleisti kliento atsakomybes projekte;
- Leisti klientui lengvai suprasti sistemos reikalavimus ir komunikuoti tarpusavyje bei su sistemų analitikais vertinant pateiktus reikalavimų modelius.

KAOS reikalavimų inžinerijos metodologija pagrįsta tikslais. Pirmiausiai nustatomi sistemos biznio tikslai, kuriuos turėtų patenkinti kuriama informacinė sistema, o nuo šių tikslų pereinama iki informacinės sistemos tikslų. Taip sudaromas hierarchinis tikslų modelis, kuriame vienas biznio tikslas išskiriamas kaip pagrindinis. Ši hierarchija dažniausiai būna itin didelė, kurią sudaro šimtas ir daugiau tikslų, tad modeliavimo eigoje tikslai dažnai skaidomi į atskirus modelius pagal tam tikrus tikslų tipus:

- Patenkinamieji tikslai arba kitaip sakant tikslai, kurie patenkina objektų norus;
- Informaciniai tikslai;
- Atsparumo klaidoms tikslai;
- Sistemos palaikomumo tikslai;
- Saugumo tikslai.

Pagal nustatytus sistemos tikslus identifikuojami sistemos agentai, t.y. objektai, kurie atsakingi už tam tikrų tikslų įgyvendinimą (sudaromas atsakomybių modelis). Nustačius objektus identifikuojamos jų atliekamos operacijos ir operacijų sąlygojami įvykiai. Taip pat sudaromi kiti rekomenduotini ir savo nuožiūra parinkti modeliai pagal KAOS notaciją. Visi šie modeliai pateikiami ir aprašomi KAOS reikalavimų dokumente (BOCK Konrad (2000) Goal-driven Modeling).

KAOS metodologija taip pat įdomi tuo, kad kuriami reikalavimų modeliai nebūtinai turi būti vizualizuojami. KAOS pateikia ir savitą reikalavimų kalbą, kuri gali būti naudojama vietoje

vizualių modelių, tačiau pastaruoju metu ši praktika taikoma ne itin dažnai dėl ganėtinai sudėtingo jos kūrimo ir juo labiau skaitymo.

Daugialypis KAOS metodologijos požiūris per KODĖL, KA, KAS ašis:

- Iš anksto apgalvotas požiūris: funkcinių ir nefunkcinių tikslų modeliavimas, panaudojant IR/ARBA tikslų diagramas;
- Struktūrinis požiūris: srities objektų modeliavimas;
- Atsakomybių požiūris: sistemos agentų modeliavimas;
- Funkcinis požiūris: S2B paslaugų modeliavimas;
- Elgsenos požiūris: sistemos dinamikų modeliavimas;

KAOS supratimo procesas:

- Identifikuoti tikslus ir su jais susietus objektus
- Identifikuoti potencialius agentus ir jų galimybes
- Sukurti tikslų apribojimus
- Tobulinti objektus ir įvykius
- Identifikuoti objektų ir įvykių apribojimus
- Identifikuoti alternatyvias galimybes
- Nustatyti įvykius atsakingiems agentams

### **1.3.2. KAOS modeliai**

KAOS odontologija susideda iš pagrindinių sąvokų (5 lentelė). KAOS metodologija susideda iš keturių pagrindinių modelių:

- Tikslų modelio;
- Atsakomybių modelio;
- Objektų modelio;
- Operacijų modelio.



**KAOS odontologija**

<b>Sąvoka</b>	<b>Apibūdinimas</b>
<b>Objektas</b>	Objektas yra sistemos daiktas. Objektai yra specifikuojami: esybė, sąryšis, įvykis, priklausomai nuo to, koks objektas – savarankiškas, momentinis ar antraeilis.
<b>Įvykis</b>	Įvykis – tai vidinis ar išorinis ryšys tarp objektų. Įvykis charakterizuoja būsenų perėjimus.
<b>Agentas</b>	Agentas – tai kita objekto sąvoka, kuris valdo tam tikrus veiksmus. Agentai turi prieigą, kontrolę, jeigu objekto būsenos yra jo kontrolėje. Agentai gali būti žmonės, programos, įrenginiai.
<b>Tikslas</b>	Tikslas – tai objektas, kurio sistema siekia. IR-tobulimo ryšys sieja tikslą su potiksliais. Tai reiškia, kad jeigu visi potiksliai yra tenkinami, pakankama sąlyga tikslui tenkinti. ARBA-tobulinimo ryšys sieja tikslą su alternatyviais potiksliais. Reiškia, jeigu vienas iš potikšlių yra tenkinamas, pakankama sąlyga tenkinti tikslui. Tikslų struktūra gali būti pavaizduota IR/ARBA cikliniu grafiku.
<b>Apribojimas</b>	Apribojimas – tai įgyvendinantis tikslas, kuris nusakomas valdomų būsenų, kurias valdo individualūs agentai. Tikslai turi būti aprašyti apribojimuose.
<b>Prielaida</b>	Prielaida – tai faktas, atsižvelgiant į agentus aplinkoje. Prielaidos dažnai būna pagalbinėmis, leidžiančiomis įrodyti teisingą tikslo panaudojimą. Jos yra preliminariai teisingos, tačiau gali būti lengvai pakeičiamos. .

Šaltinis. Sudaryta autoriaus pagal Brandozzi, M. (2001) <http://users.ece.utexas.edu/~perry/work/papers/R2A-MB-thesis.pdf>

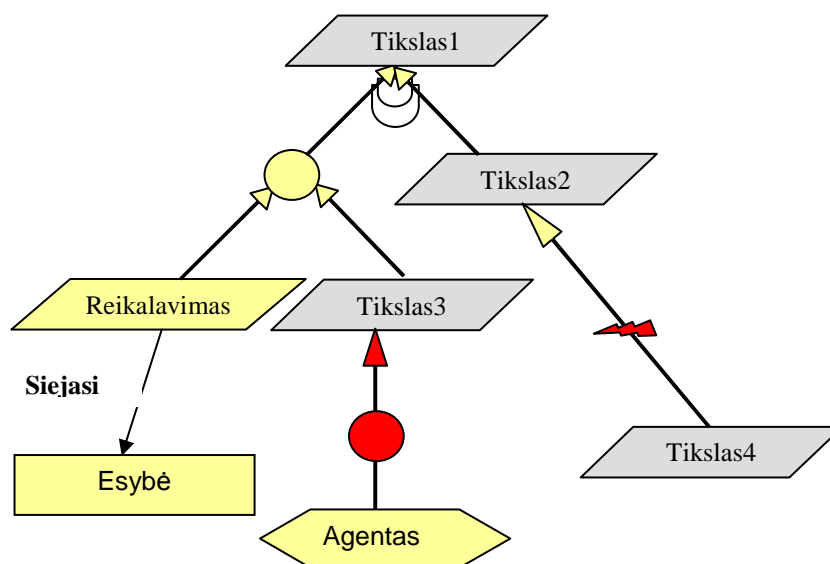
Šie modeliai sudaromi pateikta seka ir talpinami į KAOS reikalavimų dokumentą, kartu su probleminiu sakiniu, reikalavimų surinkimo metodų aprašymu bei kiekvieno modelio trumpu aprašymu, pateikiant informaciją, kuri negali būti išskaityta iš grafinių modelių.

### 1.3.3. Tikslų modelis

Tikslų modelis - tai modelis, kuriame apibūdinami kuriamos sistemos tikslai. Tikslai modelyje sudaro hierarchiją. Viršutiniai hierarchinio medžio tikslai dažniausiai apibūdina verslo tikslus, o apatiniai informacinės sistemos tikslus.

Tikslų modelis sudaromas iš tikslų, galimybių, agentų, esybių, tobulinimo ryšių, „ir“ ir „arba“ ryšių, konflikto tarp tikslų ir atsakomybės ryšių. Šio modelio sudarymas nėra sudėtingas. Didžiausias sunkumas sudarant modelį, tiksliai identifikuoti kuriamos sistemos tikslus bei sujungti juos hierarchiniais ryšiais. Dar viena nemaža problema – teisingai identifikuoti galimybes. Kartais surasti skirtumus tarp tikslų ir galimybių yra ganėtinai sunku, tačiau šiuo atveju notacijos klaida nėra stipriai įtakojanti projektuojamą reikalavimų modelį (WENTZ, A. (2004) Goal-Driven Requirements Engineering: the KAOS Approach)

Tikslų modelio sudarymo taisyklės ir naudojami notacijos elementai pateikti 8 paveiksle.



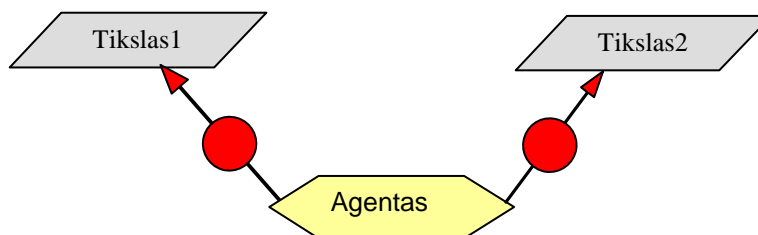
9 pav. Tikslų modelis pagal KAOS notaciją

Šaltinis: sudaryta autorius pagal project Tacos (2005), Key ideas underlying KAOS

### 1.3.4. Atsakomybių modelis

Atsakomybių modelis skirtas atvaizduoti tikslams, kurių įgyvendinimą įtakoja tam tikras agentas, kurio atsakomybės atskleidžiamos. Tikslai, agentas ir atsakomybės ryšiai yra trys elementai sudarantys šio modelio notaciją.

10 paveiksle pateiktas atsakomybių modelis.

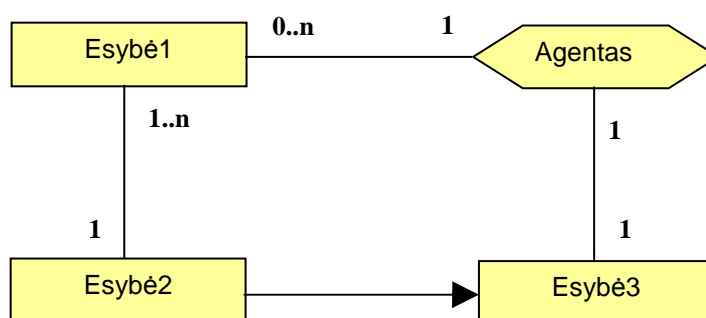


**10 pav. Atsakomybių modelis pagal KAOS notaciją**

Šaltinis: sudaryta autorias project Tacos (2005), Key ideas underlying KAOS

### 1.3.5. Objektų modelis

Objektų modelis apibūdina tikro pasaulio objektus susijusius su tam tikrais tikslais ar inicijuojančius tam tikras operacijas. Modelyje gali būti atvaizduojami dviejų tipų objektai: esybės ir agentai. Modelio esmė – parodyti objektų tarpusavio sąsajas (asociacijas), paveldėjimą ir atributus. Objektų tarpusavio ryšiai apibūdinami pateikiant sąsajos daugybiškumą. Šis modelis pateikiamas 11 paveiksle.



**11 pav. Objektų modelis pagal KAOS notaciją**

Šaltinis: sudaryta autorias project Tacos (2005), Key ideas underlying KAOS

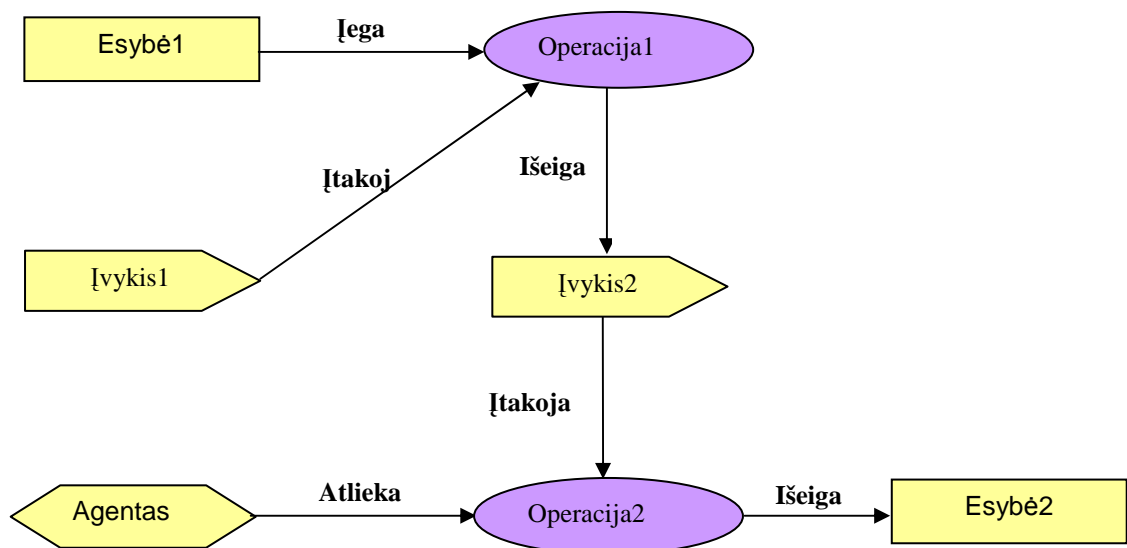
Objektų modelyje tiek esybė, tiek agentas sąveikauja su kitomis esybėmis arba agentais tokiais pačiais ryšiais. Meta-modelyje esybė yra lygi agentui pagal savo atvaizdavimo galimybes. Kaip matome, kiekvienoje asociacijoje žymimas daugybiškumas. Jei daugybiškumas pažymėtas tik vienoje traktuojama, kad daugybiškumas yra vienodas abiejuose asociacijos galuose. Galimi įvairūs daugybiškumo variantai: 1 su 1, m..n (reiškia nuo nulio iki begalybės) su 1, 1 su m..n, kur m – bet koks sveikas skaičius nuo 0, o n – begalybė. Paveldėjimas modelyje parodo, kad Esybė3 turi visus Esybė2 atributus ir dar gali turėti papildomus, tik jai būdingus atributus.

### 1.3.6. Operacijų modelis

Šis modelis apibūdina agentų atliekamus veiksmus siekiant įgyvendinti tikslus, už kuriuos agentas yra atsakingas (atsakomybių modelis). Šiame modelyje sutinkame naujus notacijos elementus, tokius kaip operacija, įėjimo ir išėjimo ryšiai, priežasties ryšys ir įvykis.

Operacijos nustatomos apklausiant dalykinės srities objektus. Dali operacijų gali būti identifiukuota ir analizuojant surinktus reikalavimus bei kitais gerai žinomais standartiniais duomenų surinkimo metodais. Įsidėmėtina tai, kad kiekviena operacija turi turėti priežastį, t.y. turi būti atliekama agento arba inicijuojama tam tikro įvykio, kuris gali būti išorinis arba sukurtas kitos operacijos. Iš šio teiginio seka, kad kiekviena operacija turi turėti tiek įėjimo arba priežasties ryšį, tiek išėjimo ryšį (WENTZ, A. (2004) Requirements Engineering: the KAOS Approach).

Operacijų modelio pagrindinius notacinius elementus ir galimas sąsajas tarp jų geriausiai atskleidžia vizualizuotas operacijų modelis (12 pav.).



12 pav. Operacijų modelis pagal KAOS notaciją

Šaltinis: sudaryta autorias project Tacos (2005), Key ideas underlying KAOS

Operacijų modelio notacija ir kūrimas vienas sudėtingiausių iš KAOS modelių. Kaip matome 5 paveikslėlyje šiame modelyje operacija gali būti sukelta agento arba įtakota įvykio. Galimas ir suminis variantas, t.y. operacija įvyksta tik atliekama agento, jei įvykęs tam tikras įvykis. Statinis objektas – esybė, šiame modelyje yra operacijos įėjimas arba išėjimas. Operacijos išėjimas taip pat gali būti įvykis.

Be jau minėtų KAOS rekomenduotinių modelių, dažnai naudojamas modelis jungiantis operacijas su tikslais, taip pat įvykių hierarchijos modelis. Šie modeliai sudaromi pagal operacijų modelį.

### 1.3.7. KAOS pritaikymai

KAOS metodas buvo pritaikytas 20 pramoninių projektų. Metodas buvo panaudotas tikslų orientuotiems modeliams kurti įvairiuose projektuose (strateginio planavimo, biznio proceso reinžinerijos projektuose). Lentelė iliustruoja įvairovę sričių ir sistemų, kuriose KAOS buvo panaudota.

6 lentelė

**KAOS projektų pavyzdžiai**

Sritis	Sistema
Telekomunikacija	Telefonų paslaugos per TV kabelinę televiziją
Oro linijų kontrolė	Skrydžių vadovo komunikavimo pagalba
Oro linijų kontrolė	Konfliktų valdymo tarp žemės ir lėktuve susidūrimams išvengti sistema
Kosmoso aviacija	Testų rinkinių konstravimas raketoms paleisti
Plieno pramonė	Integruotas produkcijos valdymas
Automobilių pramonė	Produkcijos tvarkaraščiai; užsakymų apdorojimas
Leidyba	Autorinių teisių sekimas ir valdymas
Spauda	Laikraščių redakcijų sistema
Maisto pramonė	Prekybos centrų nuolaidų valdymas
Farmacija	Nuotolinio mokymo paslauga jūreiviams
Sveikata	Ligoninių kortelių sistema
Sveikata	Ligoninės nenumatytų atvejų paslaugos palaikymas
Natūralios kalbos apdorojimas	Tinklalapių vertėjai

Šaltinis: Mylopoulos, J. (2006) <http://www-users.cs.umn.edu/~dliang/CSci5801/Readings/GORE.pdf>

Kita lentelė (7 lentelė) demonstruoja tikslais pagrįstų reikalavimų modelių dydį, kurie buvo konstruoti kai kuriuose iš šių projektų.

7 lentelė

**Sumodeliuotų KAOS sąvokų skaičius sistemoms A-G**

Sąvoka	A	B	C	D	E	F	G
Tikslas	370	56	141	341	640	171	151
Reikalavimas	160	50	164	256	440	311	108
Agentas	80	24	32	8	315	116	21
Esybė	240	123	106	102	215	166	127
Asociacija	90	71	48	13	77	126	5
Operacija	60	59	42	70	86	36	80

Šaltinis: Mylopoulos, J. (2006) <http://www-users.cs.umn.edu/~dliang/CSci5801/Readings/GORE.pdf>

GORE (tikslais grindžiami) projektai turėjo du svarbiausius dalykus:

- reikalavimų modelis, sudarytas iš tikslo, agento, objekto ir operacijos modelių, HTML kalbos formate,
- modelių grindžiamas reikalavimų dokumentas, nurodytame formate (pavyzdžiui, IEEE Std-830 standartas).

Reikalavimų dokumento dydis būdavo tipiška nuo 100 iki 300 puslapių.

8 lentelė

### Darbo krūvio dalijimas pagal GORE

Interviu	16 %
Iššifruotos stenogramos, reziumė, ir informacijos išgavimas iš papildomų šaltinių	27 %
Modeliavimas tikslų, objektų, atstovų ir operacijų	33 %
Modelių patvirtinimas su tarpininkais, alternatyvų aptarimas, peržiūrėjimas ir dokumentacijos	9 %
Kiti	15 %

Šaltinis: Mylopoulos, J. (2006) <http://www-users.cs.umn.edu/~dliang/CSci5801/Readings/GORE.pdf>

Vidutinio dydžio projektai tipiška svyruoja nuo 3 iki 8 darbuotojų, su 1-2 konsultantais, dirbančiais per 2-5 mėnesius. 3 lentelė rodo pastangų dalijimą tokiems projektams prie RE veiklos tipo. Tų skaičių pakitimai priklauso nuo srities ekspertizės, analitiko profilio, ir kaip gerai apibrėžta problema, kuri būtų išspręsta. 3 lentelę naudojame, kad įvertintume pastangas.

### 1.3.8. KAOS metodologijos privalumai ir trūkumai

KAOS metodologija iš tikro gerai identifikuoja reikalavimus. Privalumas yra toks, kad tikslų modelis leidžia detalizuoti tikslus nuo įmonės tikslų iki informacinės sistemos tikslų. Kiti modeliai taip pat yra naudingi. KAOS apima dalį išankstinių reikalavimų, vėlyvus reikalavimus ir dalį architektūrinio dizaino (13 pav.).



13 pav. KAOS aspektai

Šaltinis: sudaryta autoriaus pagal Matulevičius, R.  
<http://www.via-novaarchitectura.org/files/EMMSAD2006/Matulevicius2.pdf>

Kitas privalumas yra tas, kad KAOS identifikuoja funkcinius ir nefunkcinius reikalavimus. KAOS modeliai pateikia atsakomybių ryšius, objektų tarpusavio ryšius, operacijų ryšius su agentais, įvykiais. KAOS turi savo specifinę notaciją, kurią sunku suprasti iš pirmo žvilgsnio. Todėl tai sumažina vartotojų skaičių. KAOS notacijai naudoti yra skirta speciali programa "Objectiver", kurios pagalba galima braižyti modelius. Tačiau, jos neturint, išskyla sunkumų dėl KAOS naudojimo.

Kadangi KAOS pateikia ir savitą reikalavimų kalbą, kuri gali būti naudojama vietoje vizualių modelių, tačiau pastaruoju tai ganėtinai sudėtinga ją naudoti ir juo labiau skaityti. Todėl būtų naudinga KAOS metodologiją pateikti UML (objektiškai orientuota) kalba, kuri yra labiau suprantama, ir leis vartotojams lengviau projektuoti sistemą KAOS modelių pagaba.

Atlikus tikslais grindžiamų požiūrių analizę, nutarta tikslais grindžiamam sistemos projektavimui naudoti KAOS metodologiją. Lyginant nagrinėtus metodus, KAOS detalizuoja reikalavimus, o tai yra svarbu atliekant tikslais grindžiamą reikalavimų inžineriją. Remiantis analizės išvadomis, KAOS papildysime modeliais, kurie detalizuoja sistemos architektūrą ir tarpusavio ryšius. Taip pat KAOS operacijų modelis yra detalus, todėl sistemos projektavimui yra naudinga sugrupuoti operacijas.

Kadangi yra reikalingi sistemos architektūrą detalizuojantys modeliais ir operacijų modeliai, tam panaudosime DoDAF metodą, kuris turi tris požiūrius: sisteminių, operacijų, techninių.

## **1.4. DoDAF metodologija**

DoDAF (The Department of Defense Architecture Framework) - tai karinės pramonės standartas. Tai neabejotinai pritaikyta kariniams tikslams sistema, taip pat plačiai taikoma privačiame, viešame ir visuomeniniame sektoriuose visame pasaulyje. DoDAF ypač tinkama didelėms sistemoms, susiduriančioms su sudėtingo integravimo ir veikimo sunkumais, ir yra išskirtinai unikalus savo naudojamu „operaciniu požiūriu“ detalizuojant išorinį vartotojo valdymo domeną, kuriame kuriama sistema funkcionuos ir veiks. Visapusiškas architektūros apibūdinimas reikalauja daugialypio požiūrio, kurio kiekviena sudėtinė dalis perteikia skirtingus architektūrinių modelių aspektus (Modeling DoDAF (2004) DoD Architecture Framework Version 1.5)

### **1.4.1. DoDAF struktūra**

DoDAF struktūra apima šiuos požiūrius (9 lentelė):

- **Operacinis požiūris** pateikia informaciją, kaip nagrinėjama sistema veikia realiame pasaulyje. DoD misijų atliekama veikla ir su misijomis susijusios informacijos apsikeitimas tarp personalo arba organizacijos padalinių yra esminiai aspektai, kurios modeliuoja

valdymo arba veiklos požiūris. Šis požiūris atskleidžia pajėgumų ir tarpusavio integracijos reikalavimus.

- **Sisteminis požiūris** apibūdina egzistuojančias ir būsimas sistemas, jų tarpusavio ryšius, kurie palaiko DoD nurodytus poreikius valdymo požiūryje.
- **Techninių standartų požiūris** kataloguoja standartų sistemų dalis bei jų tarpusavio ryšius. Šis požiūris papildo sisteminį požiūrį techninėmis detalėmis ir standartinių technologijų vystymosi prognozėmis.
- **Visi požiūriai** papildo vienas kitą, perduodant informaciją bei naudojantis vienodo pobūdžio žodynu.



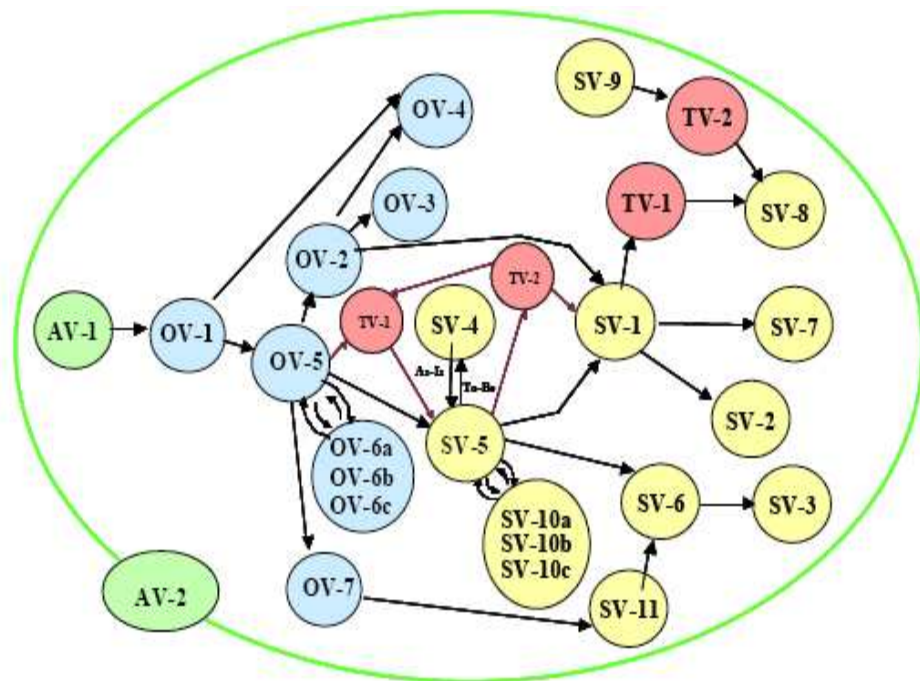
## DoDaf architektūros produktai

Požiūris	Produktas	Produkto pavadinimas	Apibūdinimas	UML diagramos
1	2	3	4	5
Visi požiūriai	AV-1	Bendro supratimo ir santraukos informacija	Apimtis, tikslas, numatomi vartotojai, aplinkos aprašymas, analitinės išvados	Tekstiniai dokumentai
Visi požiūriai	AV-2	Integruotas žodynas	Duomenų saugykla su visais produktų apibrėžimais ir išsireiškimais	UML modelio užklauskos+ataskaitos generatorius
Operacijų	OV-1	Aukščiausio lygio operacijų konceptų grafikas	Aukščiausio lygmens grafinis/tekstinis operacijų konceptų apibūdinimas	Klasių arba vartotojo reikalavimų diagrama (Class, Use Case diagram)
Operacijų	OV-2	Operacijų viršūnių sujungimų apibūdinimas	Operacijų viršūnės, operacijų veiklos tose viršūnėse, sujungimai ir informacijos pasikeitimai tarp viršūnių	Kombinuota struktūros diagrama (Composit structure diagram)
Operacijų	OV-3	Operacijų informacijos pasikeitimų matrica	Informacijos pasikeitimai tarp viršūnių, svarbūs pasikeitimų atributai	UML modelio užklauskos+ataskaitos generatorius
Operacijų	OV-4	Organizacijos ryšių diagrama	Organizaciniai ryšiai, rolės ir kt. ryšiai organizacijoje	Klasių diagrama (Class Diagram)
Operacijų	OV-5	Operacijų veiklos modelis	Veiklos, ryšiai tarp veiklų,	Veiklos diagrama (Activity diagram with Object Flows)
Operacijų	OV-6a	Operacijų taisyklių modelis	Vienas iš trijų produktų, naudojamas apibūdinti operacijų veiklos seką ir laiką, identifikuoti verslo taisykles	Tekstinis dokumentas, susietas su veiklomis
Operacijų	OV-6b	Operacijų būsenų perėjimų apibūdinimas	Vienas iš trijų produktų, naudojamų apibūdinti operacijų veiklos seką ir laiką, identifikuoti verslo procesus	Būsenų diagrama (State Machine Diagram)
Operacijų	OV-6c	Operacijų įvykių apibūdinimas	Vienas iš trijų produktų, naudojamų apibūdinti operacijų veiklos seką ir laiką, identifikuoja įvykių seką ir įvykių laiką	Darbų sekos diagrama (Sequence Diagram)
Operacijų	OV-7	Loginis duomenų modelis	Reikalavimų duomenis dokumentas ir struktūrinės veiklos procesų taisyklės	Klasių diagram (Class Diagram)
Sisteminis	SV-1	Sisteminis interfeiso apibūdinimas	Visos sistemos ir sistemos dalių, tarpusavio ryšių identifikavimas	Kombinuota struktūros diagrama (Composit structure diagram)
Sisteminis	SV-2	Sistemos sujungimo apibūdinimas	Sistemos sujungimo taškai	Kombinuota struktūros diagrama (Composit structure diagram)
Sisteminis	SV-3	Sistemos-sistemos matrica	Sistemų ryšiai pateiktoje architektūroje	DOORS dokumentas
Sisteminis	SV-4	Sistemų funkcionalumo apibūdinimas	Sistemoje veikiančios funkcijos ir informacija, einanti per šias funkcijas	Veiklos diagrama (Activity diagram with Object Flows)

1	2	3	4	5
<b>Sisteminis</b>	SV-5	Operacijų veiklos, susietos su funkcijomis, matrica	Apibrėžiama kaip operacijos siejasi su funkcijomis ir atvirkščiai, sistemų funkcijos su veiklomis	DOORS dokumentas
<b>Sisteminis</b>	SV-6	Sistemos duomenų pasikeitimo matrica	Palaiko sistemos duomenų apsikeitimus tarp sistemų	UML modelio užklauskos+ataskaitos generatorius
<b>Sisteminis</b>	SV-7	Sistemos eksploatacijos parametrų matrica	Techninės ir programinės įrangos eksploataavimo charakteristikos	UML modelio užklauskos+ataskaitos generatorius, DOORS dokumentas
<b>Sisteminis</b>	SV-8	Sistemos vystymo apibūdinimas	Planuojami žingsniai kaip vienoks sistemos vaizdas pereina palaipsniui į kitokį, esamos sistemos plėtojimas ateityje	Projekto palnavimo dokumentas, susijęs su modelio elementais
<b>Sisteminis</b>	SV-9	Sistemos technologinės prognozės	Nurodomi techniniai ir programiniai produktai, kurie jau bus sukurti ateityje ir kurie įtakos architektūros vystymą ateityje	Tekstinis arba DOORS dokumentas
<b>Sisteminis</b>	SV-10a	Sistemos taisyklių modelis	Vienas iš trijų produktų, naudojamų apibūdinti operacijų veiklos seką ir laiką, sistemos funkcionalumo apribojimai, veikiantys sistemos dizainą ir integravimą	Tekstinis arba DOORS dokumentas, surištas su sistemos funkcijomis
<b>Sisteminis</b>	SV-10b	Sistemos būsenų perėjimų apibūdinimas	Vienas iš trijų produktų, naudojamų apibūdinti operacijų veiklos seką ir laiką, reakcijos į sistemos įvykius	Būsenų diagrama (State Machine Diagram)
<b>Sisteminis</b>	SV-10c	Sistemos įvykių-įrašų apibūdinimas	Vienas iš trijų produktų, naudojamų apibūdinti operacijų veiklos seką ir laiką, specifiniai kritinių įvykių sekos tobulinimai, ir tų įvykių laikas	Darbų sekos diagrama (Sequence Diagram)
<b>Sisteminis</b>	SV-11	Fizinė schema	Fizinis loginių duomenų modelio informacijos integravimas (žinučių formatai, failų struktūros, fizinė schema)	Klasių diagram (Class Diagram)
<b>Techninis</b>	TV-1	Techninių standartų trumpas apibūdinimas	Standartų aprašymas, kurie reikalingi pateiktai architektūrai	Tekstinis arba DOORS dokumentas, surištas su sistemomis
<b>Techninis</b>	TV-2	Techninių standartų prognozės	Standartų apibūdinimas, kuriuos tikimasi panaudoti architektūroje ateityje	Tekstinis arba DOORS dokumentas, surištas su sistemomis

Šaltinis: Kobryn, C. (2004) [http://www.uml-forum.com/docs/papers/White\\_Paper\\_Modeling\\_DoDAF\\_UML2.pdf](http://www.uml-forum.com/docs/papers/White_Paper_Modeling_DoDAF_UML2.pdf)

DoDaf architektūrą sudaro daug modelių. Iš pirmo žvilgsnio ši metodologija atrodo labai sudėtinga vien dėl gausybės modelių. Todėl jie turi savo kodus, kurių pagalba galima lengviau pavaizduoti DoDaf architektūrą.



14 pav. DoDaf modelių sudarymo schema

Šaltinis: Kobryn, C. (2004) [http://www.uml-forum.com/docs/papers/White\\_Paper\\_Modeling\\_DoDAF\\_UML2.pdf](http://www.uml-forum.com/docs/papers/White_Paper_Modeling_DoDAF_UML2.pdf)

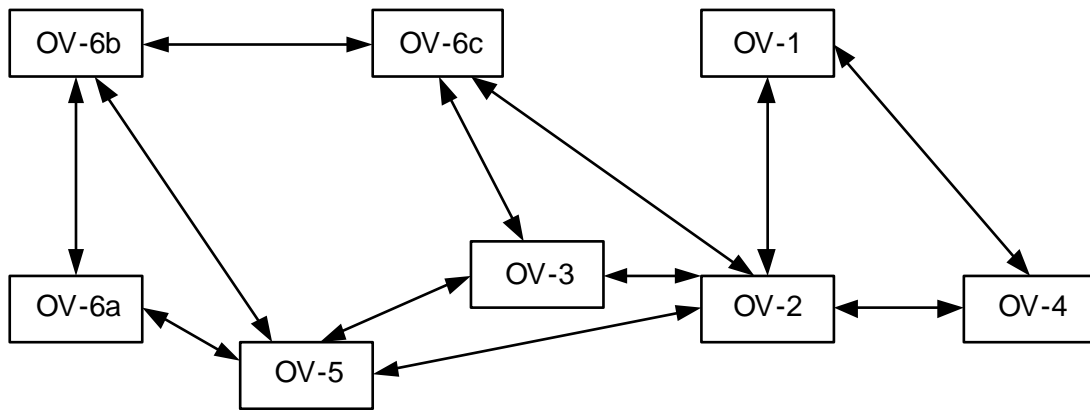
Kaip matome iš 14 paveikslo, kiekvienas modelis turi savo vietą.

Tarpusavio sąveika yra tipinis architektūros akcentas, kuris demonstruoja kritišką šių tarp-požiūrinių ryšių vystymą. OV apibūdina ir identifikuoja kas turi būti padaryta ir kaip tai turi būti padaryta. OV valdo SV kad identifikuoti sisteminius poreikius. SV identifikuoja kuri sistema palaiko operacinius reikalavimus, ir palygina esamas/pradines priemones su reikalingomis operaciniais pajėgumais. Tokiu būdu, 3 požiūriai ir jų tarpusavio ryšiai teikia pagrindą (bazę) gautų priemonių.

DoDAF 3-jų požiūrių integravimas bet kurioje pateiktoje architektūroje yra lemiamas, kai sistemos architektūros aprašymas yra naudingas kaip analitinis įrankis.

#### 1.4.2. DoDaf operacijų modelių architektūra

Labiau detalizuosime kiekvieną požiūrį, jų modelių sudarymo eiliškumą. Pirmiausia parodysime operacijų modelių eiliškumo sudarymą (15 pav.).



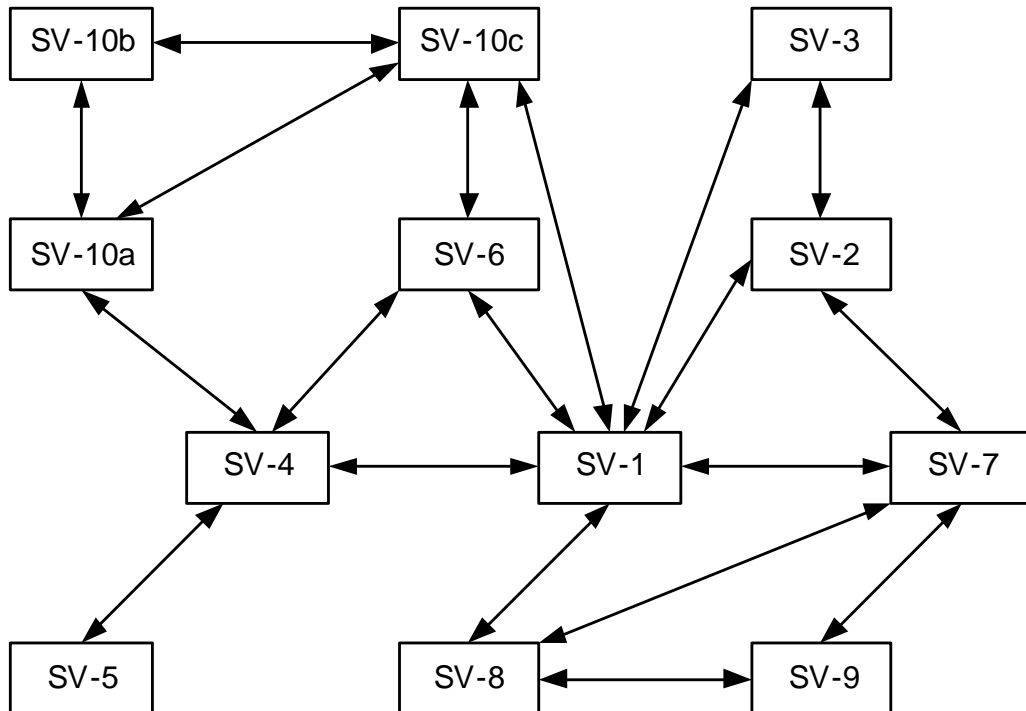
**15 pav. Operacijų modelių tarpusavio ryšiai**

Šaltinis: Kobryn, C. (2004) [http://www.uml-forum.com/docs/papers/White\\_Paper\\_Modeling\\_DoDAF\\_UML2.pdf](http://www.uml-forum.com/docs/papers/White_Paper_Modeling_DoDAF_UML2.pdf)

1. OV-1: operacijų apibūdinimas abstrakčiame lygyje;
2. OV-5: biznio proceso dokumentavimas (arba veiklos modelio sudarymas) (įvesties, valdymo, išvesties ir mechanizmų, susijusių su tomis veiklomis identifikavimas);
3. OV-6: biznio taisyklių, susijusių su biznio procesais dokumentavimas (naudokite scenarijus ir remkites OV-5 sudarant biznio taisykles ir būsenų diagramas);
4. OV-2: veiklų apjungimas į operacinius mazgus.(operaciniai mazgai – tai grupės panašių veiklų, atliekamų kartu vykdant operacija);
5. OV-7: remiantis informacijos srautais, identifikuotais veiklos modelyje, sudaryti loginį duomenų modelį.
6. OV-3: remiantis OV-5 ir OV-2 galima automatizuotai sudaryti operacinę informacijos mainų matricą;
7. OV-4: identifikuojami tipai organizacijų, kurios atliks veiklas, susijusias su operaciniais mazgais;
8. Operaciniams mazgams priskirti organizacijas ir fizines koordinates.

### 1.4.3. DoDaf sisteminių modelių architektūra

Sistemų vaizdas atvaizduoja sistemų elementus, programinę įrangą, duomenis ir sujungimus tarp jų (16 pav.).



16 pav. Sisteminių modelių tarpusavio ryšiai

Šaltinis: Kobryn, C. (2004) [http://www.uml-forum.com/docs/papers/White\\_Paper\\_Modeling\\_DoDAF\\_UML2.pdf](http://www.uml-forum.com/docs/papers/White_Paper_Modeling_DoDAF_UML2.pdf)

Žingsnių eiliškumas apibrėžiant sistemos vaizdus yra toks:

1. Identifikuoti fizinių mazgų fizines koordinatas;
2. Identifikuoti ir apibūdinti turimas sistemas (apibrėžti kas yra jų savininkai, kokias funkcijas atlieka tos sistemos ir kokios jų techninės charakteristikos);

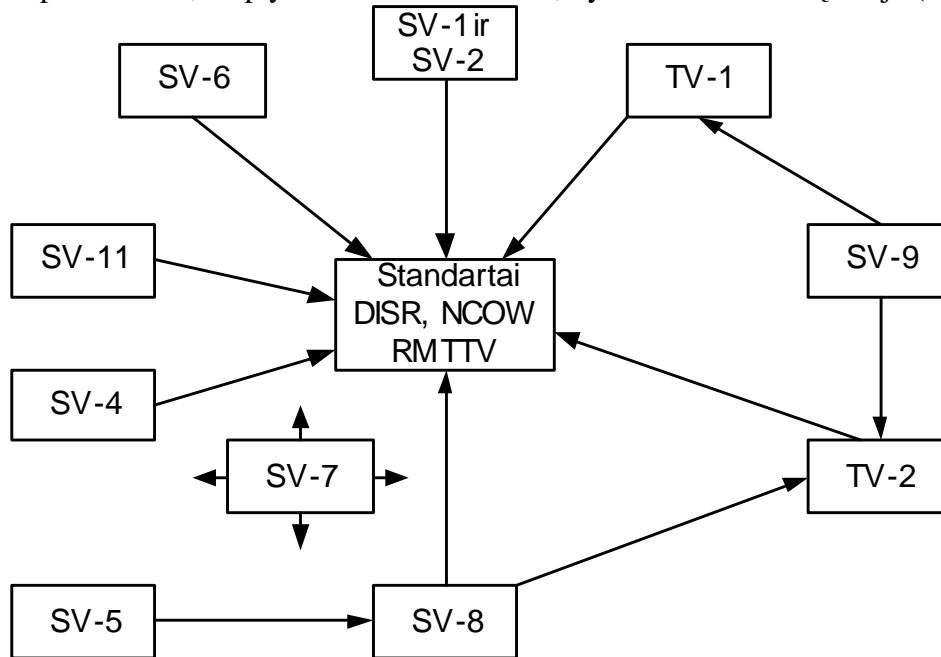
Jeigu egzistuoja sistemos, kurių funkcijos atitinka identifikuotas veiklas, tuomet pereiti prie sekančio žingsnio. Jeigu ne, tuomet pereiti prie 6-o žingsnio:

3. SV-4: Identifikuoti turimų sistemų sisteminės funkcijas. Nustatyti loginius sąryšius tarp funkcijų ir pavaldžių funkcijų. Aprašyti egzistuojančių sistemų funkcionalumą (SV-4)
4. SV-5: Remiantis SV-4 ir OV-5 susieti egzistuojančias sistemos funkcijas su veiklomis, kurias jos palaiko ir sudaryti operacinės veiklos sąryšio su sistemų funkcijomis atsekamumo matricą (SV-5);
5. Pereiti prie 8-o žingsnio;

6. SV-5: Remiantis identifikuotomis veiklomis (OV-5 vaizdu), nustatyti reikalingas sistemos funkcijas ir sudaryti operacinės veiklos sąryšio su sistemų funkcijomis atsekamumo matricą (SV-5);
7. SV-4: apibrėžti sistemos funkcijų sąryšius (SV-5 vaizde aprašytas funkcijas dekomponuoti ir sudaryti SV-4 vaizdą);
8. SV-11: Remiantis loginiu duomenų modeliu (OV-7), apibrėžti kaip šie duomenys bus fiziškai realizuojami sistemose. Ši informacija – tai fizinė schema (SV-11). Ši schema gali būti sudaroma bet kuriuo metu po to, kai jau yra turima OV-7, tačiau prieš tai kol bus apibrėžta sistemos informacinių srautų matrica (SV-6);
9. SV-10b, SV-10c, SV-10a: specifikuoti sistemų elgseną sistemų būsenų diagramomis, sistemų įvykių trajektorijų diagramomis ir sudarant sistemų taisyklių modelį. Šis etapas pasižymi dideliu iteratyvumu. Šio etapo metu gali būti identifikuojamos naujos sistemų funkcijos ir atitinkamai tikslinami SV-4 ir SV-5 vaizdai;
10. SV-1: apibrėžti sistemų interfeisus. Susieti sistemas ir jų interfeisus su OPFAC (operaciniais mazgais ir jų fizinėmis koordinatėmis);
11. SV-6: Remiantis OV-5, SV-5, SV-11 apibrėžti reikalavimus sisteminiams duomenų srautams, kurie atitinka IER OV-3 vaizde. Sudaryti sistemų informacijos srautų matrica (SV-6);
12. SV-2: sistemų sujungimų aprašymas (reikalavimų tinkliniams sujungimams (tarpmazginiams ir mazgo viduje) apibrėžimas);
13. SV-7: identifikuoti aparatūros ir programinės įrangos techninių charakteristikų parametrus ir sudaryti sistemų techninių parametrų matricą.

#### 1.4.4. DoDaf techninių standartų sudarymo architektūra

Galime pavaizduoti, kaip yra sudaromi standartai, t.y. kokie modeliai įtakoja (17 pav.)



17 pav. Modeliai, kurie siejasi su standartais

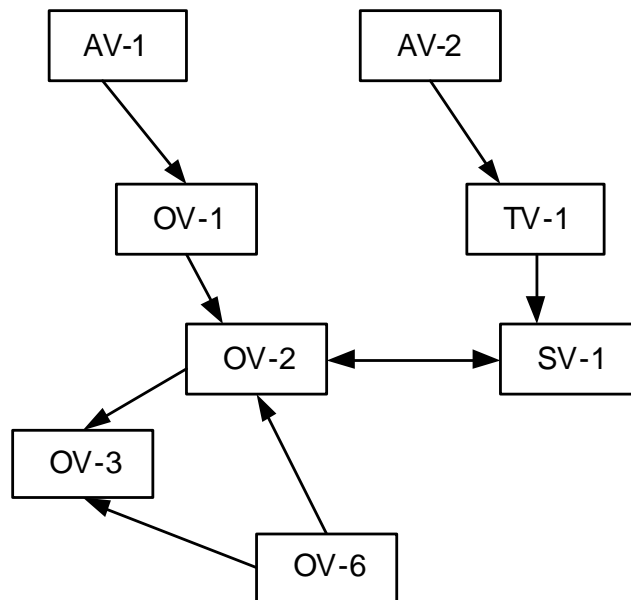
Šaltinis: Kobryn, C. (2004) [http://www.uml-forum.com/docs/papers/White\\_Paper\\_Modeling\\_DoDAF\\_UML2.pdf](http://www.uml-forum.com/docs/papers/White_Paper_Modeling_DoDAF_UML2.pdf)

#### Techninių standartų vaizdų sudarymas

1. TV-1: Remiantis OV-5 nustatyti tinkamas paslaugų teikimo sritis ir susieti jas su egzistuojančiais standartais (pvz: ISO, IEEE, ITU ir kt.);
2. TV-2: remiantis SV-4/SV-5 ir TV-1, identifikuoti sritis, kurioms negalima pritaikyti jokio žinomo standarto. Dokumentuoti šias sritis techninių standartų prognozės dokumente (TV-2) Pirmiausia išskirkime pagrindinius DoDAF modelius, kurie yra DoDAF metodologijos pagrindas.

#### 1.4.5. DoDaf pagrindinių modelių sudarymo architektūra

Pavaizdavome detalios DoDaf modelių sudarymo architektūrą. Dabar patikslinsime pagrindinius DoDaf modelius. Turint tokią didžiulią architektūrą, naudinga yra žinoti, kurie gi yra pagrindiniai modeliai. Sistemos projektavimas gali būti atliekamas iš pagrindinių modelių (18 pav.).



**18 pav. Pagrindiniai DoDaf modeliai**

Šaltinis: Kobryn, C. (2004) [http://www.uml-forum.com/docs/papers/White\\_Paper\\_Modeling\\_DoDAF\\_UML2.pdf](http://www.uml-forum.com/docs/papers/White_Paper_Modeling_DoDAF_UML2.pdf)

Kad aiškiau suprastume, kurie modeliai yra pagrindiniai, sudarome lentelę (10 lentelė).

10 lentelė

**Pagrindiniai DoDaf modeliai**

Kodas	Modelio pavadinimas	Modelio apibūdinimas
AV-1	Bendro supratimo ir suvestinė informacija	Apimtis, tikslas, numatomi vartotojai, aplinkos aprašymas, analitinės išvados
AV-2	Integruotas žodynas	Duomenų saugykla su visais produktų apibrėžimais ir išsireiškimais
OV-1	Aukšto lygmens funkcinė schema (aprašymas)	Aukščiausio lygmens grafinis/tekstinis operacijų konceptų apibūdinimas
OV-2	Operacinis tinklo mazgų sujungimo aprašymas	Operacijų viršūnės, operacijų veiklos tose viršūnėse, sujungimai ir informacijos pasikeitimai tarp viršūnių
OV-3	Operacinės informacijos apsikeitimų matrica	Informacijos pasikeitimai tarp viršūnių, svarbūs pasikeitimų atributai
OV-6	Veiklos modelis	Darbų, įvykių, veiklos sekos
SV-1	Sistemų sąsajos (interfeiso) aprašymas	Visos sistemos ir sistemos dalių, tarpusavio ryšių identifikavimas
TV-1	Techninės architektūros specifikacija (profailas)	Standartų aprašymas, kurie reikalingi pateiktai architektūrai

Šaltinis: sudaryta autoriaus

Kiekvienas modelis turi savo atitikmenį UML metodologijoje. Kadangi DoDaf neturi specialios notacijos, modelius galima vaizduoti UML diagramomis.

#### 1.4.6. DODAF modelių išrinkimas

Tam, kad išskirtume reikalingus modelius, kurie bus panaudoti KAOS metodui, sudarome lentelę (11 lentelė).



## DoDaf modelių įtraukimas

Požiūris	Produktas	Produkto pavadinimas	Modelių įtraukimas	Pagrindimas
1	2	3	4	5
Visi požiūriai	AV-1	Bendro supratimo ir santraukos informacija	-	Tai yra papildoma informacija, tiesiogiai neįtakojanti projektuojant sistemą, todėl neįtraukiama
Visi požiūriai	AV-2	Integruotas žodynas	-	Tai yra papildoma informacija, tiesiogiai neįtakojanti projektuojant sistemą, todėl neįtraukiama
Operacijų	OV-1	Aukščiausio lygio operacijų konceptų grafikas	-	KAOS operacijų modelis apibūdina operacijas, todėl nėra įtraukiamas, nes persidengia
Operacijų	OV-2	<b>Operacijų mazgų apibūdinimas</b>	+	<b>Šis modelis yra įtraukiamas į KAOS, nes reikalinga aprašyti kaip yra valdomos operacijos</b>
Operacijų	OV-3	Operacijų informacijos pasikeitimų matrica	-	Nereikalinga
Operacijų	OV-4	Organizacijos ryšių diagrama	-	Šis modelis turi atitikmenį KAOS metode, tai yra atsakomybių modelis, todėl neįtraukiamas
Operacijų	OV-5	Operacijų veiklos modelis	-	KAOS operacijų modelis aprašo detalai operacijų veiklą, todėl persidengia nėra įtraukiamas
Operacijų	OV-6a	Operacijų taisyklių modelis	-	Tai yra papildoma informacija, todėl KAOS metode nereikalingas, neįtraukiamas
Operacijų	OV-6b	Operacijų būsenų perėjimų apibūdinimas	-	Tikslų grindžiamame požiūryje nereikia detalaus būsenų apibūdinimo, todėl nėra įtraukiamas
Operacijų	OV-6c	Operacijų įvykių apibūdinimas	-	Įvykių seką apibūdina KAOS operacijų modelis iš dalies, tikslų grindžiamame požiūryje nereikia detalaus įvykių apibūdinimo, todėl nėra įtraukiamas
Operacijų	OV-7	Loginis duomenų modelis	-	Šio modelio atitikmuo yra KAOS objektų modelis, persidengia, todėl nėra įtraukiamas
Sisteminis	SV-1	<b>Sisteminis interfeiso apibūdinimas</b>	+	<b>Šis modelis yra būtinas KAOS metode, nes reikalinga sistemos dalių sąsajas aprašyti, todėl yra įtraukiamas</b>
Sisteminis	SV-2	Sistemos sujungimo apibūdinimas	-	Šis modelis reikalingas retai, specifinėms sistemoms projektuoti, todėl nėra įtraukiamas
Sisteminis	SV-3	Sistemos-sistemos matrica	-	Šis modelis reikalingas retai, specifinėms sistemoms projektuoti, todėl nėra įtraukiamas
Sisteminis	SV-4	Sistemų funkcionalumo apibūdinimas	-	Šis modelis reikalingas retai, specifinėms sistemoms projektuoti, todėl nėra įtraukiamas
Sisteminis	SV-5	Operacijų veiklos, susietos su funkcijomis, matrica	-	Iš dalies tai aprašo KAOS operacijų modelis, todėl nėra įtraukiamas
Sisteminis	SV-6	Sistemos duomenų pasikeitimo matrica	-	Tikslais grindžiamame metode modelis nėra reikalingas, tad neįtraukiamas
Sisteminis	SV-7	Sistemos eksploatacijos parametrų matrica	-	Tai yra papildoma informacija, tiesiogiai neįtakojanti projektuojant sistemą

1	2	3	4	5
<b>Sisteminis</b>	SV-8	Sistemos vystymo apibūdinimas	-	Tai yra papildoma informacija, tiesiogiai neįtakojanti projektuojant sistemą, todėl neįtraukiama
<b>Sisteminis</b>	SV-9	Sistemos technologinės prognozės	-	Tai yra papildoma informacija, tiesiogiai neįtakojanti projektuojant sistemą, todėl neįtraukiama
<b>Sisteminis</b>	SV-10a	Sistemos taisyklių modelis	-	Tai yra papildoma informacija, tiesiogiai neįtakojanti projektuojant sistemą, todėl neįtraukiama
<b>Sisteminis</b>	SV-10b	Sistemos būsenų perėjimų apibūdinimas	-	Šis modelis reikalingas retai, specifinėms sistemoms projektuoti, todėl nėra įtraukiamas
<b>Sisteminis</b>	SV-10c	Sistemos įvykių-įrašų apibūdinimas	-	Šis modelis reikalingas retai, specifinėms sistemoms projektuoti, todėl nėra įtraukiamas
<b>Sisteminis</b>	SV-11	Fizinė schema	-	Šio modelio atitikmuo yra KAOS objektų modelis, persidengia, todėl nėra įtraukiamas
<b>Techninis</b>	TV-1	Techninių standartų trumpas apibūdinimas	-	Tai yra papildoma informacija, tiesiogiai neįtakojanti projektuojant sistemą, todėl neįtraukiama
<b>Techninis</b>	TV-2	Techninių standartų prognozės	-	Tai yra papildoma informacija, tiesiogiai neįtakojanti projektuojant sistemą, todėl neįtraukiama

Šaltinis: sudaryta autoriaus pagal Kobryn, C. (2004) [http://www.uml-forum.com/docs/papers/White\\_Paper\\_Modeling\\_DoDAF\\_UML2.pdf](http://www.uml-forum.com/docs/papers/White_Paper_Modeling_DoDAF_UML2.pdf)

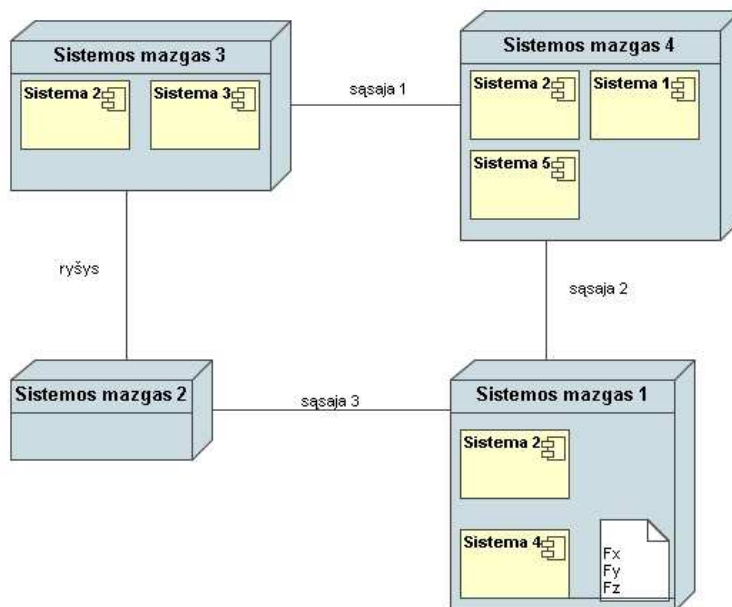
Atsižvelgiant į KAOS trūkumus, buvo išrinkti šie modeliai:

SV-1 - Sistemų sąsajos (interfeiso) aprašymas;

OV-2 - Operacijų mazgų aprašymas.

SV-1 modelis – tai sistemos interfeiso aprašymas. Jau anksčiau minėjau, kad KAOS metodologijai trūksta sistemos dalių pavaizdavimo, tad patalpinus šį modelį, bus išspręsta ši problema. OV-2 – tai operacijų mazgų apibūdinimas.

Sistemų sąsajos (interfeiso) aprašymo modelis pavaizduojamas UML diagrama (19 pav.).

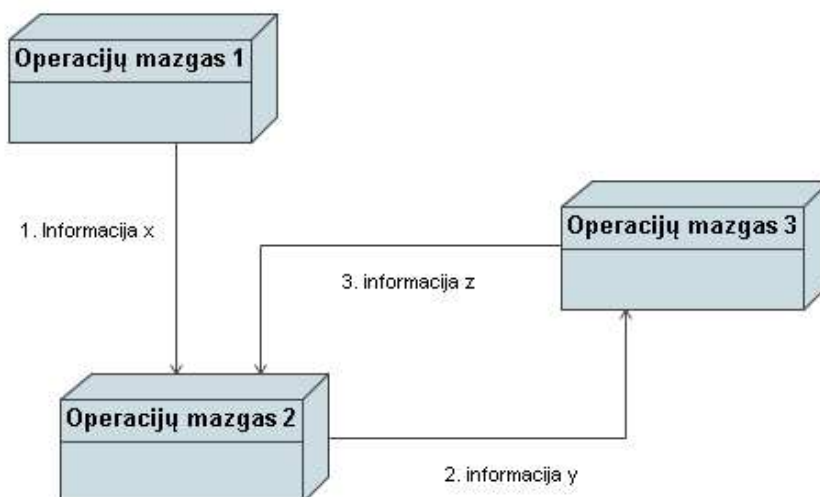


**19 pav. Sistemos sąsajų modelis**

Šaltinis: sudaryta autoriaus

Sistemos sąsajų modelis aprašo sistemos topologiją, ryšius ir programinės įrangos dislokacijos vietą (kuriame kompiuteryje koks komponentas veikia). Taip pat aprašo sprendimus, tenkinančius nefunkcinius sistemos reikalavimus. Suteikia informaciją sistemų administratoriams, įrangų tiekėjams. Gali atvaizduoti ir sistemos procesų informaciją (parodo, kuriame kompiuteryje koks procesas veikia). Šį modelį jungsime prie KAOS metodo, tokiu būdu praplėsdami tikslais grindžiamą metodą.

Operacijų mazgų modelis pavaizduojamas UML diagrama (20 pav.).

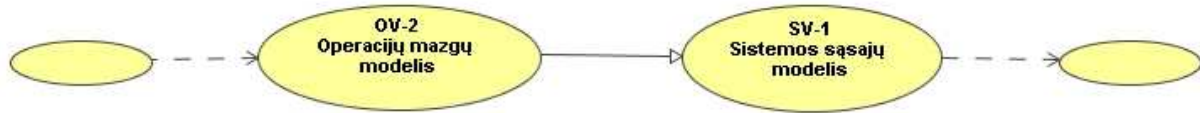


**20 pav. Operacijų mazgų modelis**

Šaltinis: sudaryta autoriaus

Operacijų mazgų modelis pavazduoja, kaip operacijų grupės siejasi tarpusavyje. KAOS metode esantis operacijų modelis vaizduoja detaliai visas operacijas, jų eigą. Todėl būtų naudinga praplėsti šią vietą, prijungiant operacijų mazgų diagramą.

Išanalizavus visą DoDaf architektūrą, pastebėta, kad SV-1 modelis yra sudaromas po OV-2 modelio.

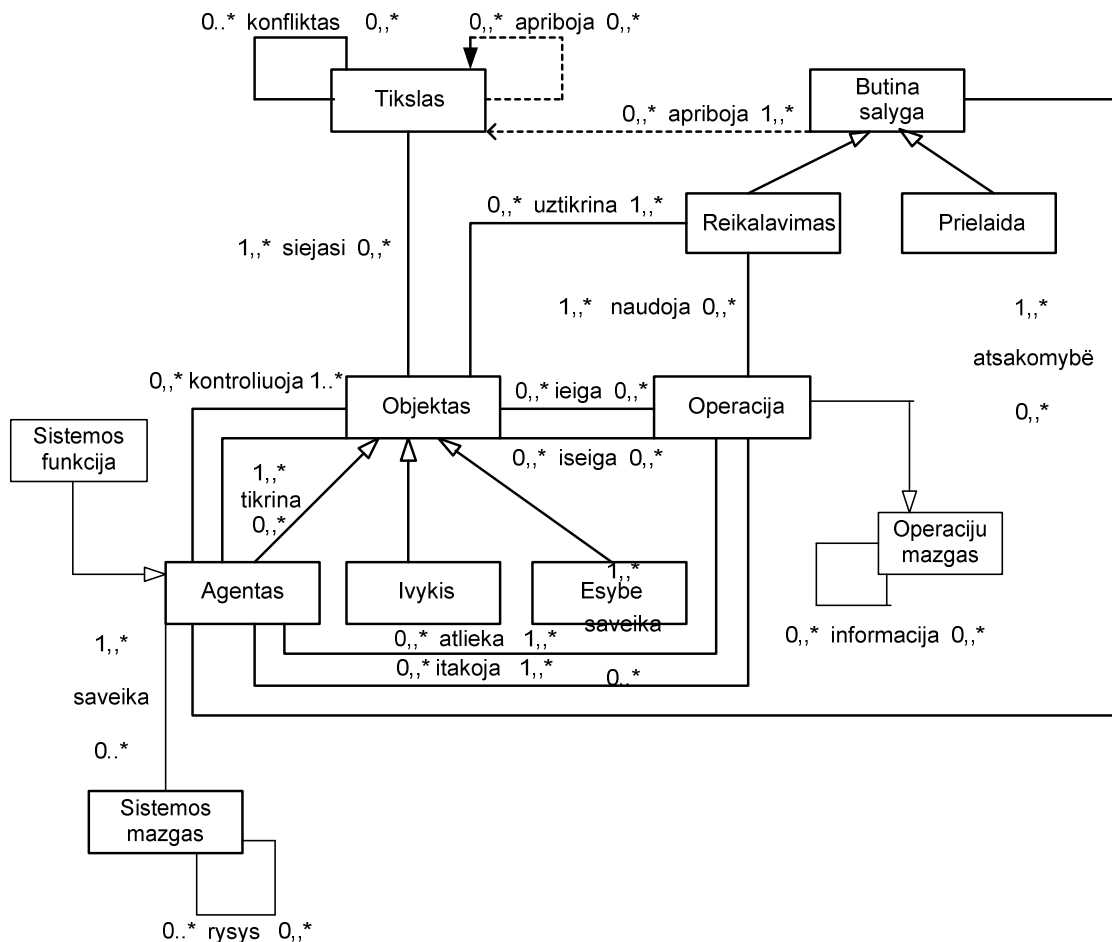


21 pav. SV-1 ir OV-2 modeliai

Šaltinis: sudaryta autoriaus

Išrinktus modelius reikia prijungti prie KAOS modelių atitinkamais ryšiais. Pirmiausia sudarysime KAOS metamodelį klasių diagrama, o po to prijungsime išrinktus DoDaf modelius.

Naudojant klases, buvo sudarytas bendras modelių metamodelis, kuris bus naudojamas kuriant profilius (23 pav.).



22 pav. Papildytas KAOS metamodelis

Šaltinis: sudaryta autoriaus pagal Heaven, W. (2003)

<http://www.cs.ucl.ac.uk/staff/A.Finkelstein/papers/umlreprofile.pdf>

Sistemos sąsajų modelis buvo prijungtas per agentą. Agentas KAOS metode vaizduoja žmogų, sistemą. Todėl logiška, kad per šį komponentą buvo prijungtas sistemos sąsajų modelis. Šio modelio sistema reiškia agentą KAOS modeliuose. Operacijų mazgų modelis buvo prijungtas per komponentą – operacija. Operacijos jungiasi pagal sritį į operacijų mazgus, tokiu būdu sukuriama operacijų mazgų modelis, kuris vaizduoja apibendrintą operacijų grupavimą ir informacijos dalijimąsi tarpusavyje.

Sistemos sąsajų modelis aprašo ryšius tarp programinės įrangos, o to KAOS metodas nespécifikavo. Sistemos projektavimui tai yra naudinga diagrama, leidžianti suprojektuoti sistemos topologiją.

Operacijų mazgų modelis papildo KAOS metodą tuo, kad visas operacijas leidžia grupuoti ir pateikti tarpusavio ryšius, t.y. informacijos pasidalijimą identifikuoja. Esant dideliame operacijų skaičiui, tai būtų naudinga pateikti struktūrizuotą diagramą.

Pasiūlyta KAOS metodą papildyti dviem DoDaf modeliais. Jais papildytas metodas bus išbandytas eksperimentinėje dalyje.

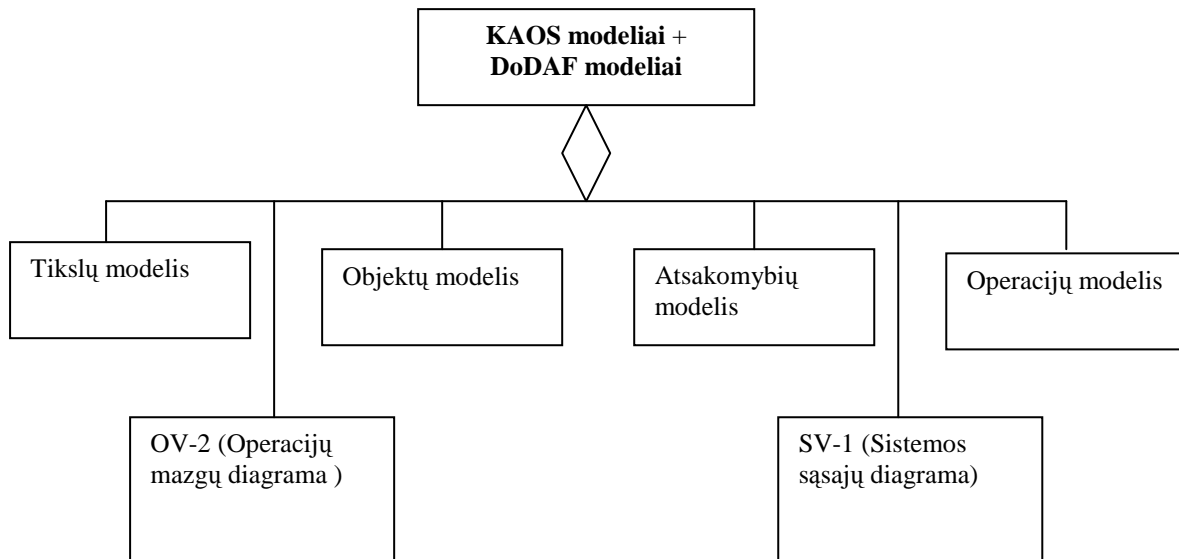
## 1.5. Analizės išvados

1. GRL metodas apima nefunkcinius reikalavimus, o funkcinių neapima. Šis metodas apima tikslais ir agentais grindžiamą modeliavimą, todėl sistemos projektavimui nėra tinkamas.
2. GQM metodas apima kokybės tikslus, tačiau tik įvertina reikalavimus prieš tarpininkų tikslus, o reikalavimų detalizavimo nevykdo.
3. EKD metodas naudojamas verslo vystymui, verslo procesų užtikrinimui. Šis metodas dažnai apima tai, kaip pasiekti vizijas ir tikslus esamai organizacijai.
4. Sistemos projektavimui pasirinkta KAOS metodologija, nes atlieka reikalavimų detalizavimą, susiedama tikslus su funkciniais ir nefunkciniais komponentais.
5. KAOS metodologiją nuspręsta papildyti šiais DoDAF modeliais: sistemos sąsajų modelis, operacijų mazgų modelis. Šie modeliai užtikrina visapusišką sistemos sąsajų projektavimą.

## 2. SIŪLOMO SPRENDIMO METODIKA

Analizės dalyje pateiktas papildytas KAOS metodas, kurio sudarymą ir panaudojimą aprašysime šioje dalyje.

Parodomi KAOS ir DoDAF modeliai, kurių yra šeši: (23 pav.). KAOS modeliai: tikslų modelis, objektų modelis, atsakomybių modelis, operacijų modelis. DoDAF modeliai: operacijų mazgų modelis, sistemos sąsajų modelis.



23 pav. Papildyto KAOS metodo sudėtis

Šaltinis: sudaryta autoriaus pagal Jureta, I. (2005)

<http://www.isys.ucl.ac.be/staff/stephane/GETI2100Slide/KAOS.pdf>

KAOS yra vienas iš svarbiausių tikslais grindžiamų metodologijų. KAOS turi savo specifinę notaciją, todėl būtų naudinga KAOS modelius pavaizduoti UML notacija. KAOS projektavimui buvo sukurtas specialus komercinis įrankis Objectiver, bet jis naudoja nestandartinę grafinę notaciją. Ši notacija nėra gerai žinoma. Modeliuojant KAOS UML notacijoje, mums nereikia naujos programinės įrangos, kuri modeliuoja specifinę KAOS notaciją.

Tikslesniam šių modelių suvokimui pateikiame šių modelių atitikmenis: koks modelis koki UML modelį atitinka (12 lentelė).

**Modelių atitikmenys**

<b>Modelis</b>	<b>Atitikmuo (UML diagrama)</b>
Tikslų modelis	IR/ARBA tikslų diagramos
Objektų modelis	Klasių diagram (Class Diagram)
Atsakomybių modelis	Kontekstinės diagramos, vartotojo reikalavimų diagramos (Use Case diagram)
Operacijų modelis	Būsenų diagrama (State Machine Diagram), sekos diagrama (Sequence Diagram)
SV-1 (Sistemos sąsajų diagrama)	Kombinuota struktūros diagrama (Composit structure diagram)
OV-2 (Operacijų sujungimų diagrama)	Kombinuota struktūros diagrama (Composit structure diagram)

Šaltinis: sudaryta autoriaus

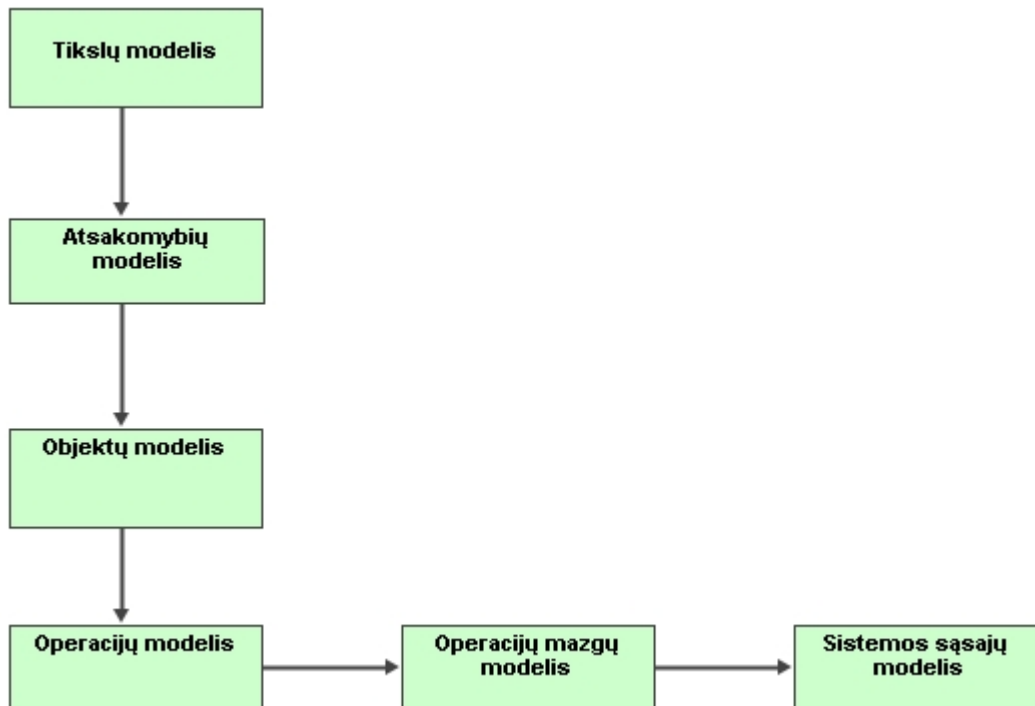
Papildyto KAOS metodo modelius visus galima pavaizduoti UML diagramomis. Aišku, tai nėra tikslūs jų atitikmenys, o panašūs. Tam, kad galėtume naudoti šį metodą, sudarysime profilius su Magicdraw 12 įrankiu.

**2.1. Modelių sudarymo eiga**

Papildyto KAOS metodo modelius siūloma sudaryti pagal pateiktą eigą (24 pav.). Pateikto metodo modelių seka yra tokia: pirmiausia sudaromi KAOS modeliai, o po to sudaromi pridėti DoDaf modeliai:

1. Tikslų modelis;
2. Agentų modelis;
3. Objektų modelis;
4. Operacijų modelis;
5. Operacijų viršūnių sujungimų modelis;
6. Sistemos sąsajų modelis.





**24 pav. Papildyto tikslais grindžiamo metodo modelių seka**

Šaltinis: sudaryta autoriaus

KAOS metodo modelių sudarymo eiga nepakeista. Jos net nebūtų tikslinga pakeisti. Pirmiausia sudaromas tikslų modelis, po jo seka atsakomybių modelis, objektų modelis ir operacijų modelis.

Prijungtas operacijų mazgų modelis sudaromas po to, kai suprojektuojamas operacijų modelis. Tik identifikavus operacijas, galima sudaryti operacijų mazgų modelį.

Prijungtas sistemos sąsajų modelis sukuriamas paskutinis. Čia yra vaizduojama programinės įrangos topologija, todėl tik suprojektavus veiklą, galima šį modelį sudaryti. Jis dažniausia būna vienas iš paskutinių.

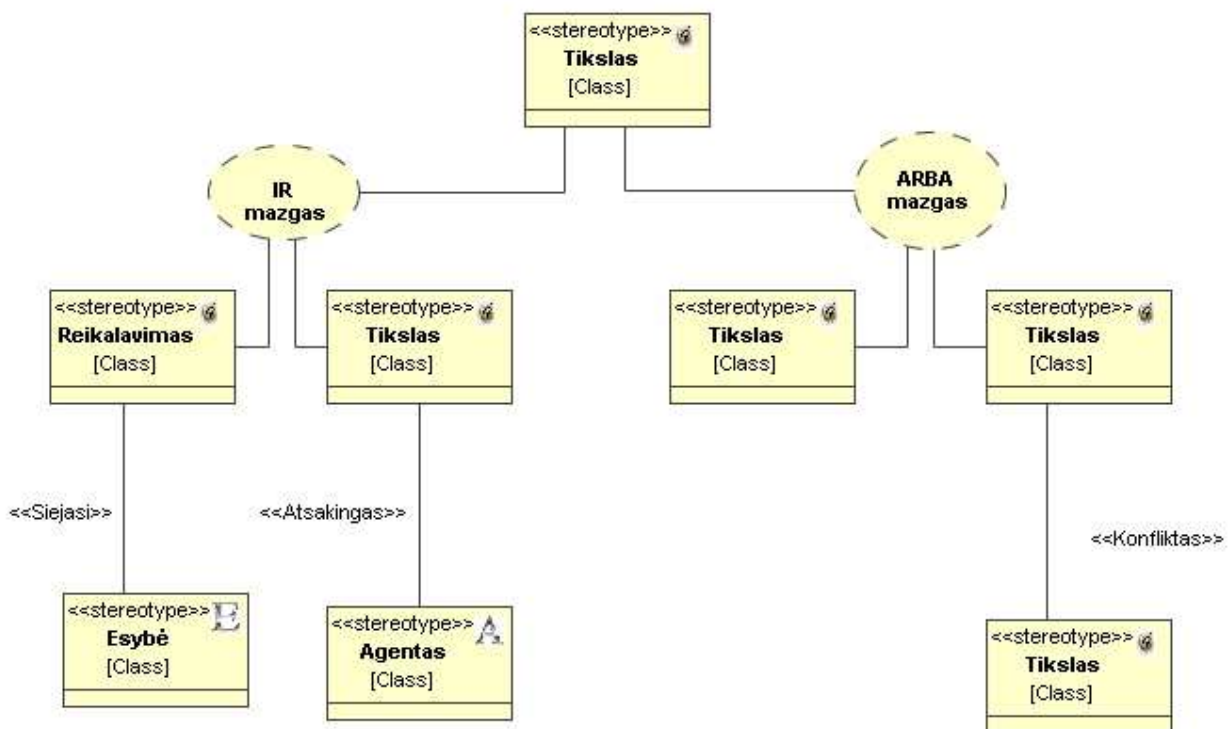
## 2.2. Tikslais grindžiamo metodo profiliai

Magicdraw paketas turi savo profilių, kuriuos galima naudoti, kuriant modelius. Profiliai turi specifinius stereotipus, apribojimus, galimas reikšmes. Profiliai taip pat dažnai turi specifinius elementus, kuriuos galima panaudoti savo profiliams. Nuosavus profilius galima naudoti kituose projektuose. Kiti vartotojai gali panaudoti profilį kaip modulį. Norint susikurti savo profilį, reikia:

- Susikurti profilio elementus;
- Pridėti stereotipus, apribojimus, kitus modelio elementus.

Mano darbe pateiktas metodas bus aprašytas Magicdraw profiliais, kurie bus naudojami tikslais grindžiamam sistemos projektavimui. Profiliai sukurti su Magicdraw 12.0 paketu.

### 2.2.1. Tikslų modelio profilis



25 pav. Tikslų modelio profilis

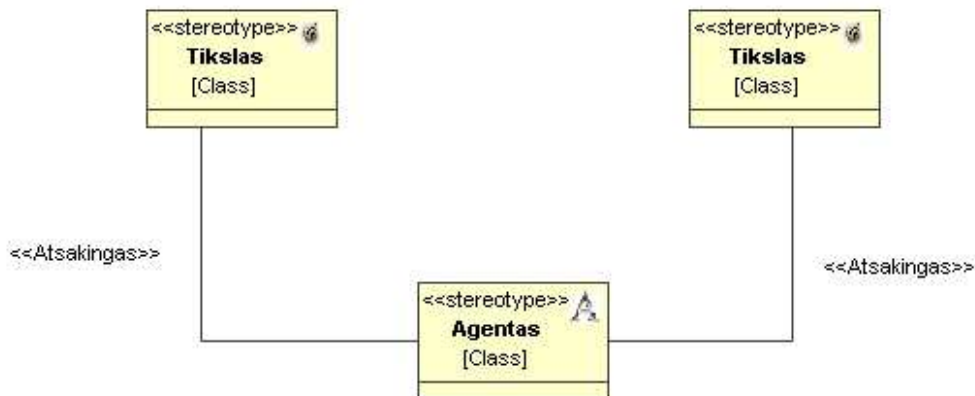
Šaltinis: sudaryta autoriaus

Tikslai tikslų modelyje turi tokias logikas: AND, OR, XOR, šie ryšiai sujungia tikslus su potiksliais. Tai reiškia, kad jei yra ryšys AND, tiklas bus patenkinamas, jei abu potiksliai bus patenkinti. Šį ryšį UML notacijoje pažymėjome IR mazgu. Sekanti logika – OR, kuri reiškia, jeigu vienas iš potikslu pas įvykdytas, tikslas bus tenkinamas. UML notacijoje mes pavaizdavome jį ARBA mazgu. Paskutinis ryšys yra XOR – jis reiškia konfliktą: jei abu potiksliai bus įvykdyti, tuomet tikslas nebus tenkinamas, jeigu vienas iš potikslių bus įvykdytas, tikslas bus tenkinamas, o jeigu abu potiksliai nebus įvykdyti, tikslas nebus tenkinamas. Taip pat tikslų modelyje yra

reikalavimas, kuris panašus į tikslą, o jis siejasi su esybe. Agentas turi savo vietą modelyje, yra atsakingas už tam tikrus tikslus. Konfliktą pavaizduojame <<konfliktas>> asociacija tarp konfliktuojančių tikslų. Ryšys tarp tikslo ir agento vaizduojamas asociacija „atsakingas“. Na o ryšys tarp tikslo ir esybės vaizduojamas asociacija „siejasi“.

### 2.2.2. Atsakomybių modelio profilis

Sekantis modelis – tai atsakomybių modelis (26 pav.)



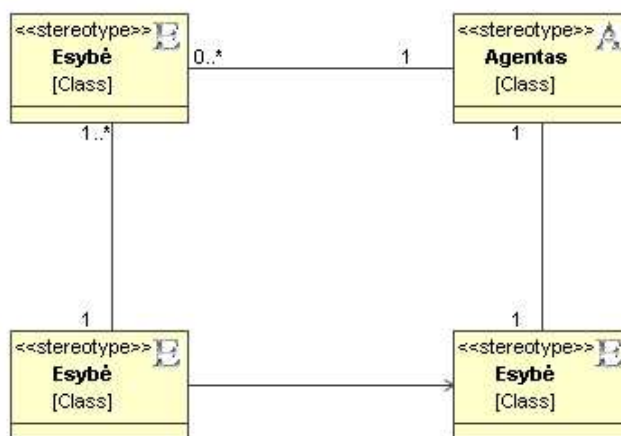
26 pav. Atsakomybių modelio profilis

Šaltinis: sudaryta autoriaus

Agentas negaliu būti pavaizduotas kaip aktorius UML, nes aktorius sąveikauja su sistema, bet nėra jos dalis. Todėl negalime jų sutapatinti. Taigi, agentą pavaizduojame klase. Sukuriame <<atsakingas>> asociaciją, kuri bus naudojama jungiant agentą su tikslu.

### 2.2.3. Objektų modelio profilis

Kitas modelis – objektų modelis (27 pav.).



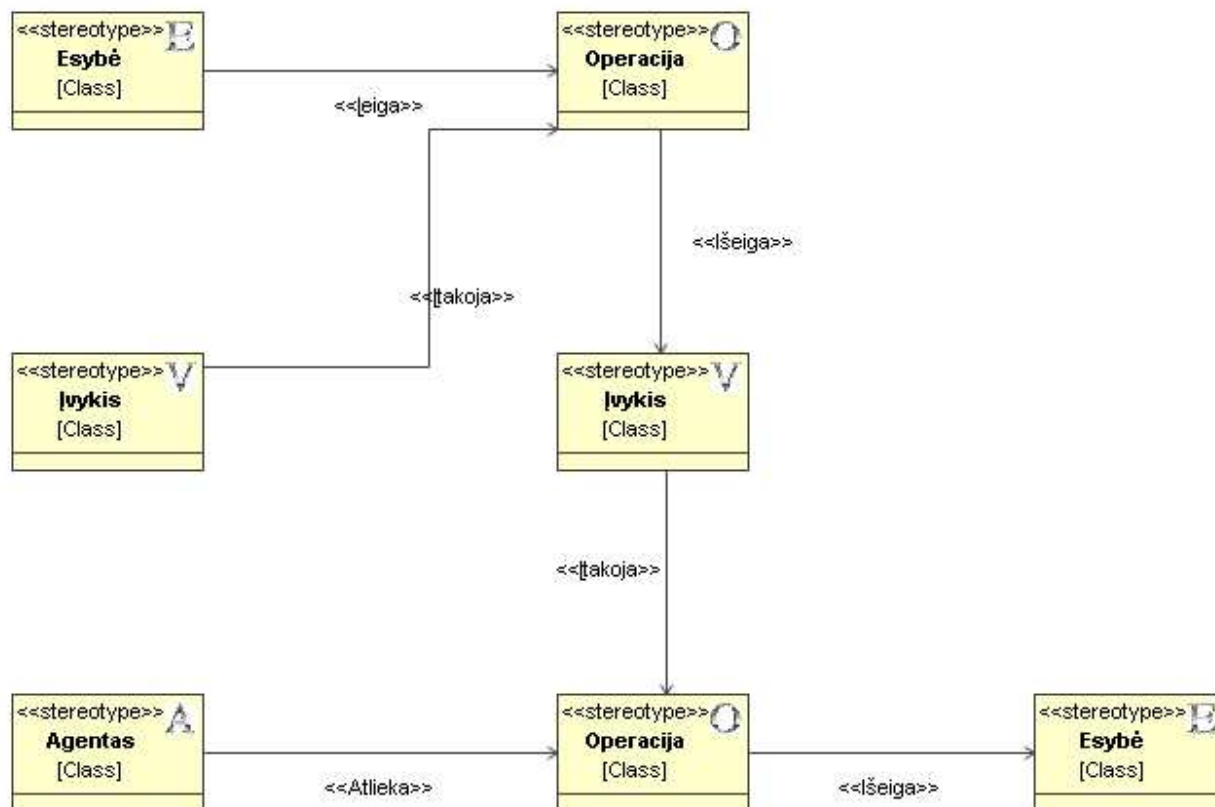
27 pav. Objektų modelio profilis

Šaltinis: sudaryta autoriaus

Objektų modelis atitinka UML klasių modelį. Agento ir esybės stereotipus jau turėjome. O ryšiams nauji stereotipai nėra kūrimai, nes jie atitinka klasių diagramos ryšius. Klasių diagrama pavaizduojamas objektų modelis.

## 2.2.4. Operacijų modelio profilis

Sekantis modelis – operacijų modelis (28 pav.). Jis pagal prasmę atitinka UseCase diagramą. Tačiau aktoriai nėra pavaizduojami kaip UseCase diagramoje.



28 pav. Operacijų modelio profilis

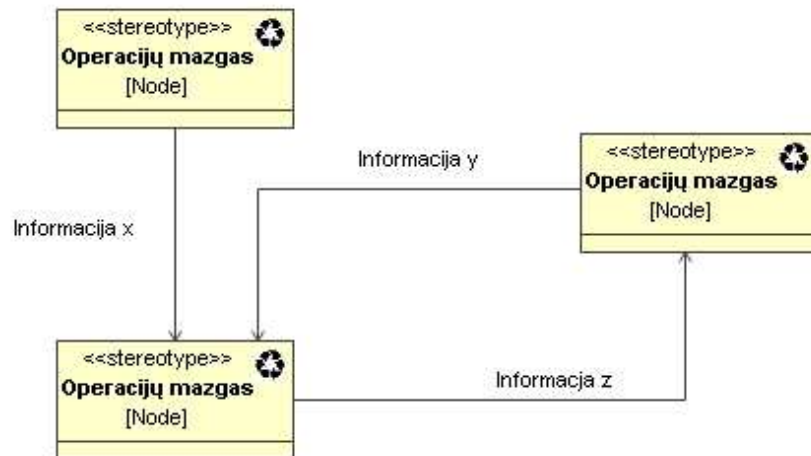
Šaltinis: sudaryta autoriaus

Įvykis pavaizduojamas kaip klasė. Sukuriamas jam stereotipas. Taip pat operacija irgi pavaizduojama klase. Jų ryšiams sukurtos asociacijos <<įeiga>>, <<išeiga>>, <<atlieka>>. Su šiais komponentais galima kurti operacijų diagramas.

Operacijos nustatomos apklausiant dalykinės srities objektus. Dalis operacijų gali būti identifikuota ir analizuojant surinktus reikalavimus bei kitais gerai žinomais standartiniais duomenų surinkimo metodais. Kiekviena operacija turi turėti priežastį, t.y. turi būti atliekama agento arba inicijuojama tam tikro įvykio, kuris gali būti išorinis arba sukurtas kitos operacijos. Iš šio teiginio seka, kad kiekviena operacija turi turėti tiek įėjimo arba priežasties ryšį, tiek išėjimo ryšį.

## 2.2.5. Operacijų mazgų modelio profilis

UML notacijoje pavaizduotiems DoDaf modeliams taip pat sukūriami profiliai. Operacijų viršūnių sujungimų modelio profilis (29 pav.).



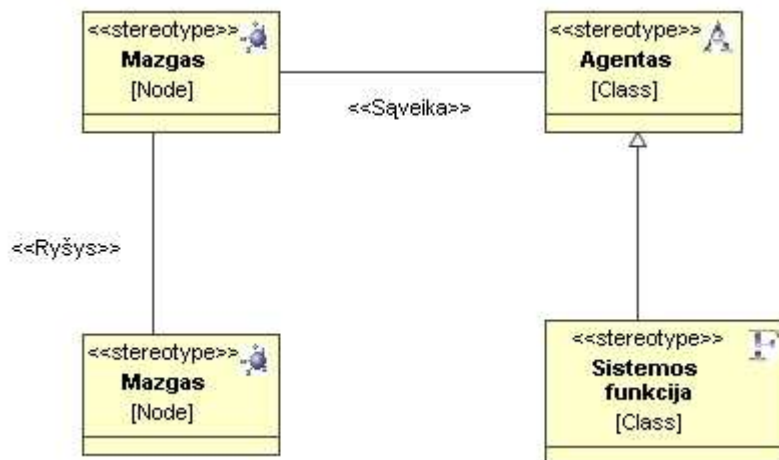
29 pav. Operacijų mazgų profilis

Šaltinis: sudaryta autoriaus

Buvo sukurtas naujas stereotipas <<operacijų mazgas>>. Ryšiai yra paprasti. Jie identifikuoja informaciją. Svarbu tai, kad ryšys turi būti su rodykle tam, kad parodytų, kas gauna informaciją.

## 2.2.6. Sistemos sąsajų modelio profilis

Paskutis modelis – sistemos sąsajų modelio profilis (30 pav.).



30 pav. Sistemos sąsajų modelio profilis

Šaltinis: sudaryta autoriaus

Šio modelio atvaizdavimui buvo sukurtos naujos klasės: <<mazgas>> ir <<sistemos funkcija>>. Sistemą atitinka <<agentas>> KAOS metodologijoje. Sistemos funkcijoms pavaizduoti sukurta nauja klasė <<sistemos funkcija>>.

Sukurti visi tikslais grindžiamos reikalavimų inžinerijos modeliai, kurių efektyvumą patikrinsime eksperimente, projektuodami konkretų pavyzdį.

### **2.3. Tikslais grindžiamo metodo profilių naudojimas**

Norint naudotis sukurtais profiliais, reikia susinstaliuoti kompiuteryje Magicdraw įrankį. Atsidaryti profilį pavadinimu „Tikslais grindžiamas modeliavimas“. Jame yra sukurti visi stereotipai, kuriuos reikia naudoti, kuriant savo modelius. Šiame kataloge yra ir teoriniai modeliai, kuriuos galima naudoti kaip pavyzdį. Pats modelių kūrimas nėra sudėtingas, nes visi reikalingi stereotipai ir ryšiai yra profilyje.

### 3. EKSPERIMENTINIS SKYRIUS

Modelių veiksmingumui patikrinti paimsime konkrečią situaciją. Pirmiausia atliksime sistemos projektavimą tradiciniu objektiškai orientuotu metodu, o po to naujuoju sukurtu metodu.

Mūsų įmonė - UAB "Ventos polimeriniai gaminiai" užsiima polimerinių gaminių gamyba ir realizavimu. Gamybos produktai yra šie: elektros izoliacinė juostelė, sodų tepalas, polimerinė bituminė plėvelė, hermetizuojanti mastika „Gelanas“, polimeriniai mastikiniai intarpai, karščiui atspari mastika, mišinys medeliams baltinti. Produkcija yra parduodama didmenininkams Lietuvoje ir užsienyje. Taigi, gamybos planavimui palengvinti, projektuosime gamybos planavimo sistemą.

#### 3.1. Objektams orientuotas gamybos planavimo sistemos projektavimas

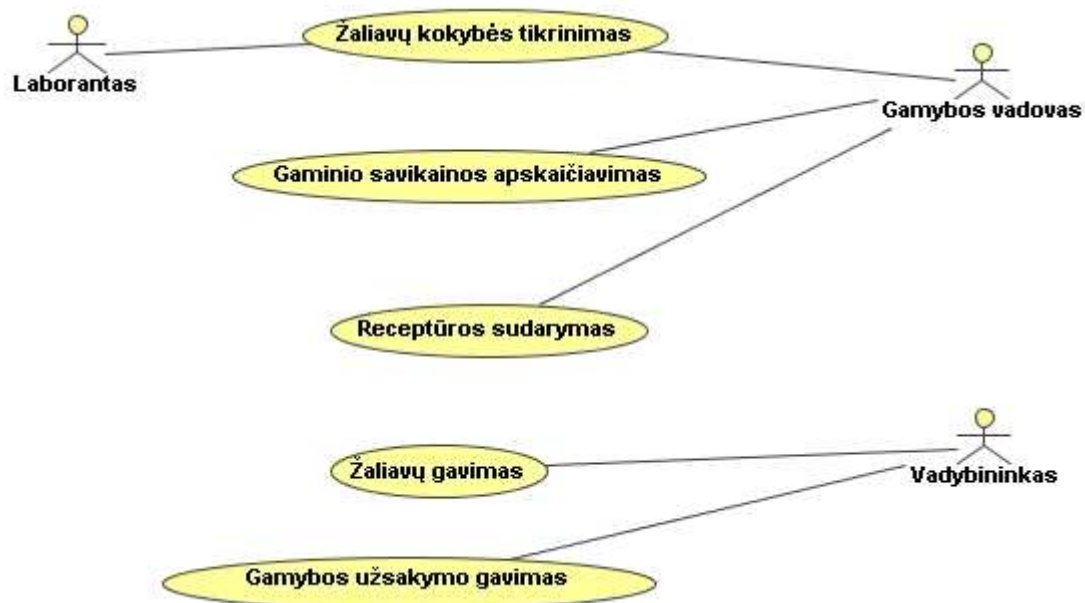
UML 2.0 turi savyje 13 diagramų: Veiklos, klasių, komunikavimo, komponentų, sudėtinės, išdėstymo, sąveikos peržiūros, objektų, paketų, sekų, būsenų, laiko ir elgsenos diagramos. Eksperimentui panaudosime elgsenos – panaudos atvejų, veiklos -, klasių, sekų, įrangų:

1. Panaudos atvejų diagrama: specifikuoja sistemos funkcijų sąveiką su vartotojais; įtraukia ryšius tarp sistemos ir jos aplinkos;
2. Veiklos diagrama: tai yra procesų modeliavimo priemonė. Joje vaizduojama, kurie procesai po kokių įvyksta, kas juos įvykdo.
3. Klasių diagrama: specifikuoja statinę objektų struktūrą, įtraukdama atributus ir operacijas, ryšius;
4. Sekų diagrama: specifikuoja dinaminę elgseną tarp objektų, reprezentuojant chronologine žinučių sekų tvarka tarp objektų;
5. Komponentų diagrama: modeliuoja programą vykdančias komponentes ir programinę įrangą, kuri dirba su šiomis komponentėmis.

Klasių diagrama modeliuoja statinius sistemos elementus. Sekų, panaudojimo atvejų ir veiklos diagramos modeliuoja dimaninį sistemos vaizdą. Komponentų diagrama modeliuoja sistemos realizaciją

Eksperimentui naudosime šias diagramas, nes jos pagal savo funkcionalumą leis atlikti sistemos projektavimą ir tiks palyginti papildytam KAOS metodui.

### 3.1.1. Gamybos planavimo panaudojimo atvejų diagrama



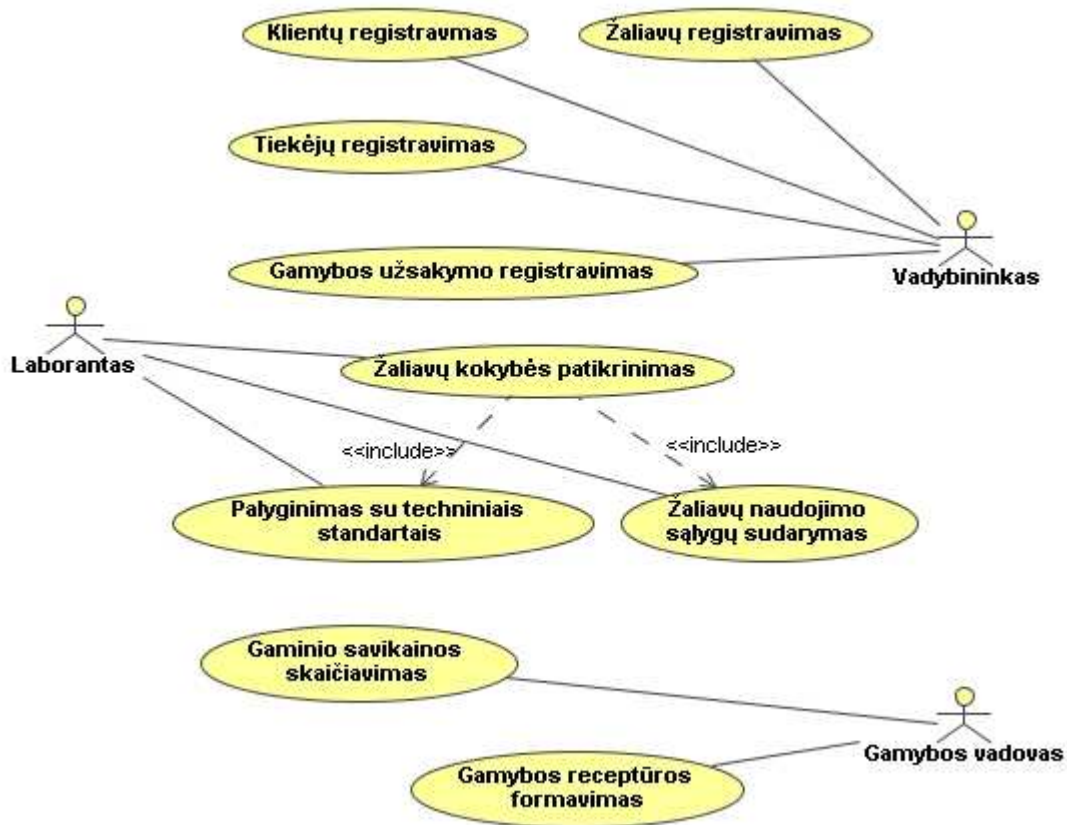
31 pav. Gamybos planavimo panaudojimo atvejų diagrama

Šaltinis: sudaryta autoriaus

Aprašoma elgsena iš vartotojo pozicijų. Dalyvių yra trys: laborantas, gamybos vadovas, vadybininkas. Laborantas yra atsakingas už vieną procesą – žaliavų kokybės. Gamybos vadovas yra atsakingas už šiuos procesus: gaminio savikainos apskaičiavimą, receptūros sudarymą, žaliavų kokybės tikrinimą. Vadybininkas yra atsakingas už žaliavų registravimą ir gamybos užsakymo registravimą.



### 3.1.2. Gamybos planavimo sistemos panaudojimo atvejų diagrama

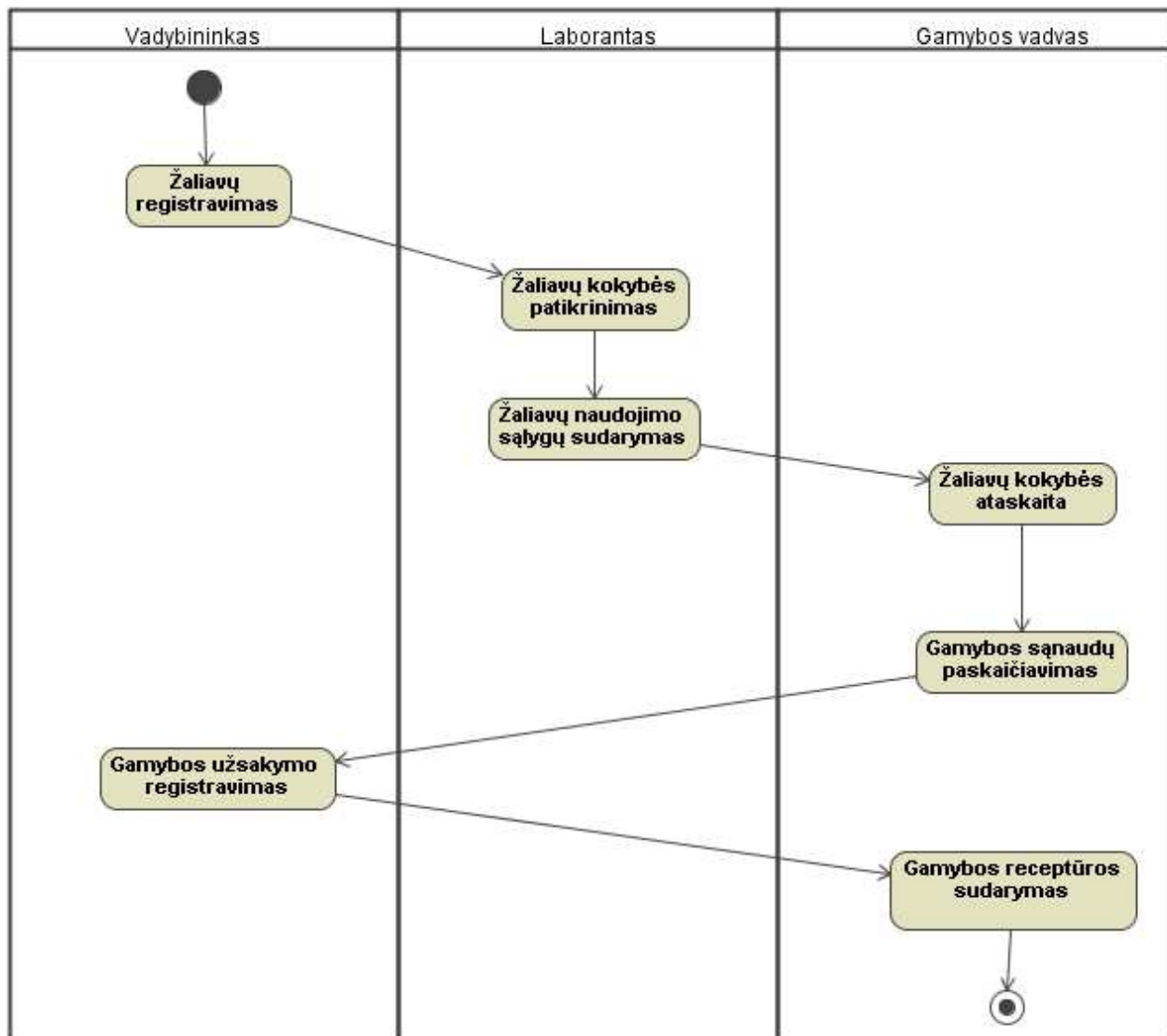


32 pav. Gamybos planavimo sistemos panaudojimo atvejų diagrama

Šioje diagramoje jau yra parašomas kompiuterizuotas procesas (32 pav.). Pirmoje panaudojimo atvejų diagramoje (31 pav.) buvo aprašytas nekompiuterizuotas procesas. Čia lygiai taip aprašoma elgsena iš vartotojo pozicijų. Pavaizduojama, kas už kokius procesus yra atsakingi. Laborantas valdo žaliavų kokybės patikrinimą, kurio metu jas palygina su techniniais standartais ir sudaro žaliavų naudojimo sąlygas. Kadangi žaliavas gali tiekti skirtingi tiekėjai, ir tų pačių tiekėjų žaliavos kaskart šiek tiek gali skirtis, reikia atlikti tikrinimą. Po jo tampa aiškios naudojimo ribos, gainant produkciją. Gamybos vadovas atsakingas už du procesus: apskaičiuoja gaminio savikainą, kuri kaskart keičiasi, priklausomai nuo užsakymo dydžio, sudaro gamybos receptūrą pagal tai, kiek ir ko ruošiasi gaminti. Vadybininkas yra atsakingas už klientų, žaliavų, užsakymų ir tiekėjų registravimą.

### 3.1.3. Gamybos planavimo sistemos veiklos diagrama

Sekančioje veiklos diagramoje aprašysime, kokie procesai po kokių vyksta, kas už juos yra atsakingi. Detalizuosime gamybos planavimo veiklą (33 pav.).



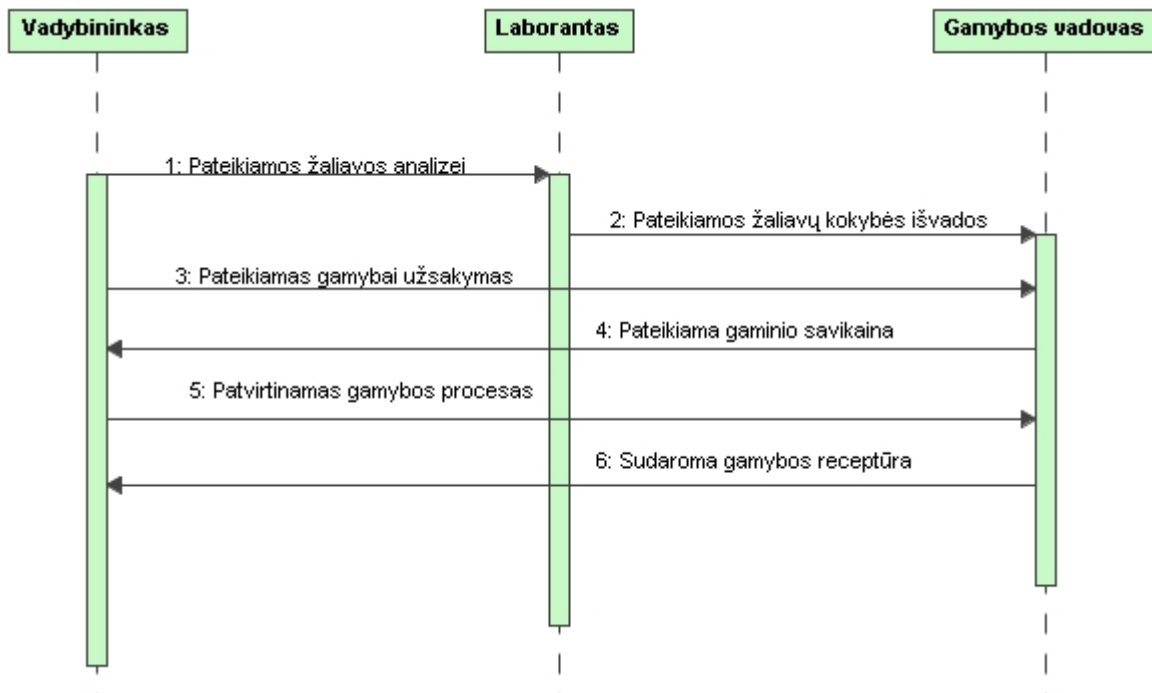
**33 pav. Gamybos planavimo sistemos veiklos diagrama**

Šaltinis: sudaryta autoriaus

Veiklos diagrama yra procesų modeliavimo priemonė. Joje vaizduojama (33 pav.), kurie procesai po kokių įvyksta, kas juos įvykdo. Čia vėl dalyvauja 3 dalyviai – vadybininkas, laborantas ir gamybos vadovas, procesai išdėstyti eiliškumo tvarka. Pirmiausia vadybininkas registruoja gautas žaliavas, kurių kokybę patikrina laborantas. Laborantas sudaro žaliavų naudojimo sąlyga ir žaliavų kokybės ataskaitą pateikia gamybos vadovas. Savo ruožtu gamybos vadovas paskaičiuoja gamybos sąnaudas, rezultata pateikia vadybininkui. Jeigu tenkina sąnaudų kiekis ir kaina, tuomet registruoja gamybos užsakymą ir gamybos vadovas gavęs užsakymą sudaro gamybos receptūrą.

### 3.1.4. Gamybos planavimo sistemos sekų diagrama

Sekanti diagrama – sekų diagrama. Sekų diagrama vaizduoja įvykių scenarijus ir objektus, kurie dalyvauja. Ši diagrama nurodo įvykių eilės tvarką, nenurodo laiko reikšmių (34 pav.).



34 pav. Gamybos planavimo sistemos sekų diagrama

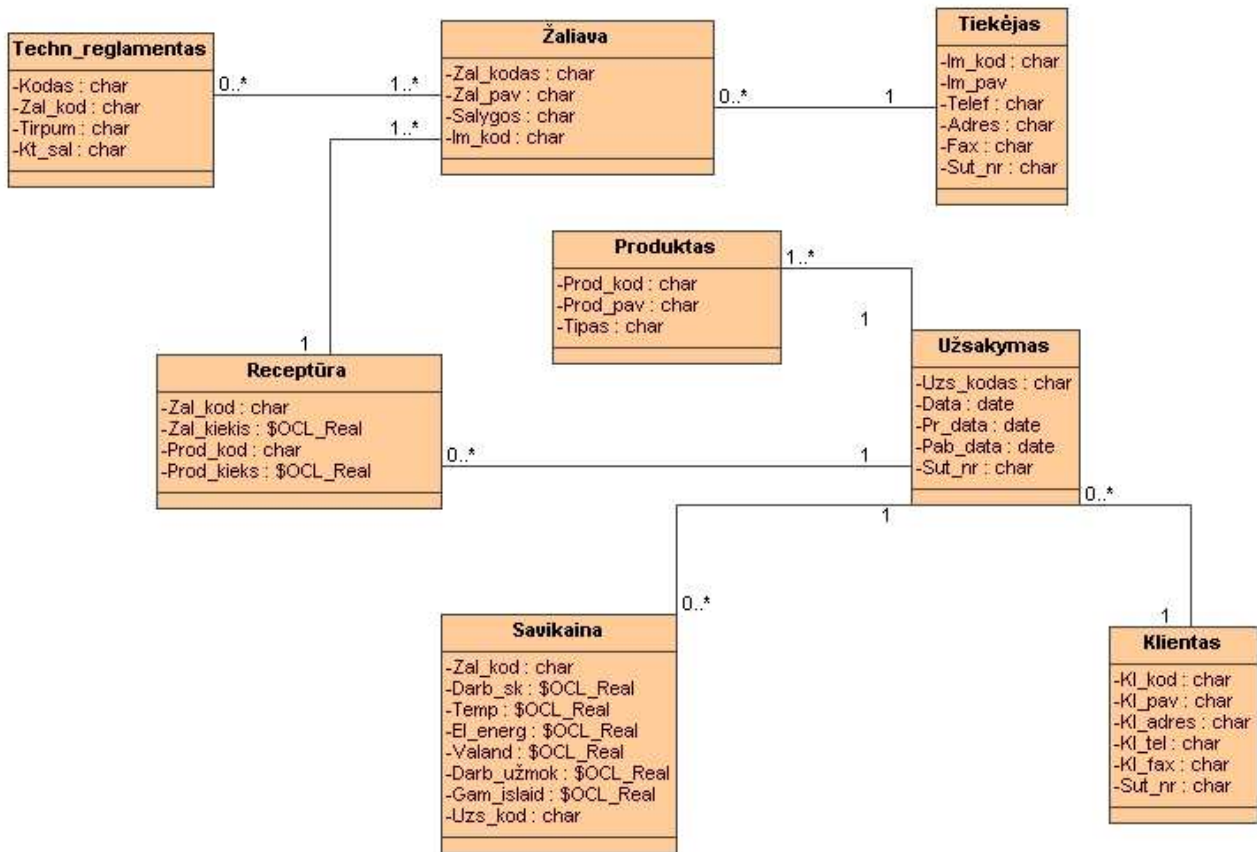
Šaltinis: sudaryta autoriaus

Sekų diagrama galima atvaizduoti pranešimus, kurie yra siunčiami tarp biznio objektų. Modelis parodo įvykių ir pranešimų eilės tvarką.

Pagrindiniai elementai šiame modelyje yra objektai, kurie dalyvauja sąveikose, objektų atliekami metodai, kurių dėka tarpusavyje ir sąveikauja objektai. Tarp objektų metodų nurodomas pranešimas, ką reikia atlikti objektui gaunančiam pranešimą. Sekos diagramos nurodo, kokie objektai dalyvauja, kokia eilės tvarka ir kokius pranešimus (bei atsakymus) jie siunčia vienas kitam.

Sekų diagramoje pavaizduojama įvykių seka. Pirmasis įvykis – tai žaliavų pateikimas analizei. Gamybos vadovas gauna išvadas iš laboranto apie patikrintas žaliavas. Toliau seka įvykis – pateikiamas gamybai užsakymas. Gamybos vadovas pateikia gaminio savikainą vadybininkui. Jis savo ruožtu patvirtina gamybos procesą. Gamybos vadovas sudaro gamybos receptūrą.

### 3.1.5. Gamybos planavimo sistemos klasių diagrama



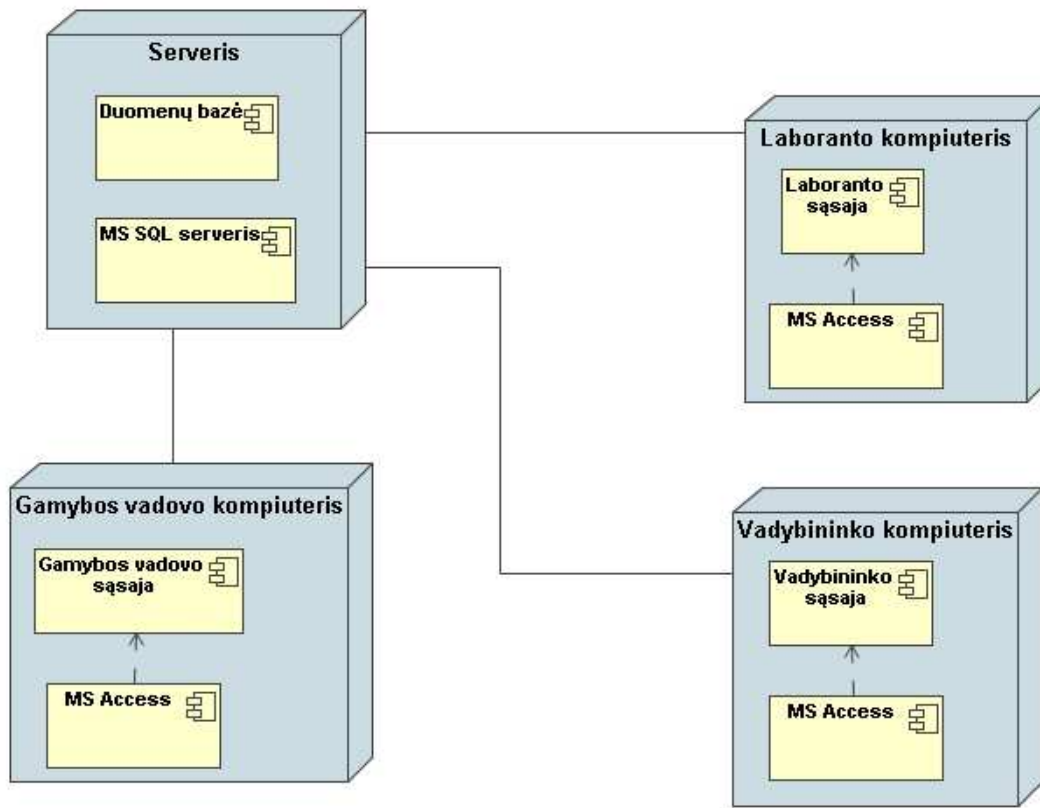
35 pav. Gamybos planavimo sistemos klasių diagrama

Šaltinis: sudaryta autoriaus

Klasių diagrama nurodo analizės rezultatus: kam informacijos sistema yra skirta, ką IS gali vykdyti (atlikti), ir projektavimo rezultatus: iš ko informacijos sistema bus sudaryta (iš kokių dalių suprojektuota). Informacija, esanti klasių diagramose, tiesiogiai projektuojasi į programos išėties tekstą, kuris realizuoja taikomąją IS, todėl klasių diagramas yra būtina sudaryti. Klasės yra aprašomos nurodant objektų (patenkančių į aprašomą klasę) atributus (parametrus), atliekamus veiksmus (metodus) ir ryšius (asociacijas) su kitomis klasėmis. Klasių modelį sudaro esybės.

Esybių iš viso yra aštuonios: žaliava, tiekėjas, produktas, užsakymas, klientas, receptūra, savikaina, tech\_reglamentas. Šios esybės susietos atitinkamais ryšiais. Kiekviena jų turi savo atributus.

### 3.1.6. Gamybos planavimo sistemos išdėstymo diagrama



**36 pav. Gamybos planavimo sistemos išdėstymo diagrama**

Šaltinis: sudaryta autoriaus

Išdėstymo diagrama modeliuoja programą vykdančias komponentes ir programinę įrangą, kuri dirba su šiomis komponentėmis (36 pav.). Išdėstymo diagrama yra paprasti modeliai, kurie parodo, kaip tarpusavyje siejamos techninės ir programines įrangos komponentės, reikalingos IS realizavimui. Šias diagramas sudaro techninės įrangos projektavimo grupė. Siekiama paskirstyti programinę įrangą taip, kad būtų tenkinami reikalavimai, keliami IS.

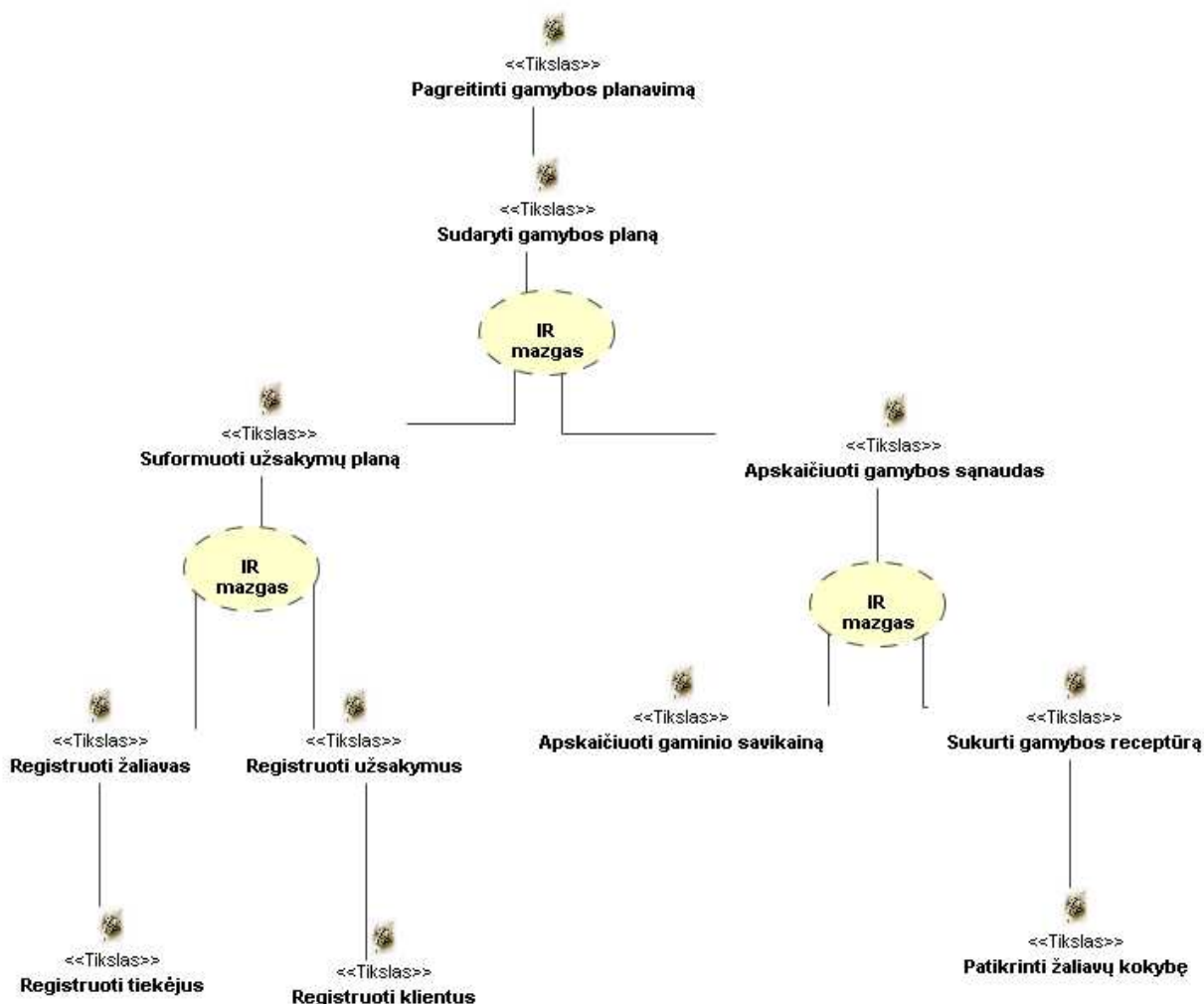
Šis modelis parodo sąveikaujančias sistemas. Sistema veikia per serverį, kuriama saugoma duomenų bazė. Serveris sąveikauja su vartotojais. Gamybos vadovas turi savo sąsają per MS Access, vadybininkas taip pat turi savo vartotojo sąsają per MS Access, laborantas lygiai taip pat. Jų visų kompiuteriai naudojami MS SQL serveriu, kuriame saugoma duomenų bazė.

### 3.2. Tikslais orientuotas gamybos planavimo sistemos projektavimas

Sukuriami visi modeliai, naudojant MagicDraw profilius. Eksperimentui panaudosime papildyto KAOS metodo modelius:

1. Tikslų diagrama;
2. Atsakomybių diagrama;
3. Objektų diagrama;
4. Operacijų diagrama;
5. Operacijų mazgų diagrama;
6. Sistemos sąsajų diagrama.

#### 3.2.1. Gamybos planavimo sistemos tikslų modelis



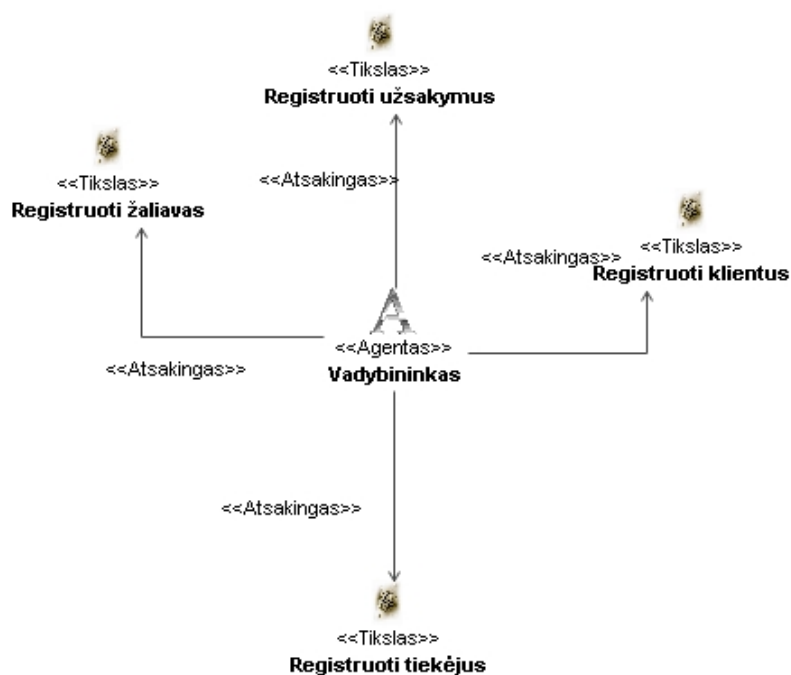
### 37 pav. Gamybos planavimo sistemos tikslų modelis

Šaltinis: sudaryta autoriaus

Pirmiausia sudaromas tikslų modelis. Viršutinis hierarchinio medžio tikslas apibūdina verslo tikslą – pagreitinti gamybos planavimą. Šiam tikslui pasiekti, reikalinga tikslų detalizacija. Tam, kad būtų pasiektas šis tikslas, reikia sudaryti gamybos planą. Sekantys tikslai jau yra informacinės sistemos tikslai. Tikslas – suformuoti užsakymų planą yra sudarytas iš dviejų tikslų: registruoti žaliavas ir registruoti užsakymus. Tam, kad šie tikslai švyktų, reikia registruoti tiekėjus ir klientus. Kitas tikslas – apskaičiuoti gamybos sąnaudas – susideda iš potikslių: apskaičiuoti gaminio savikainą ir sukurti gamybos receptūrą. Potikslis – sukurti gamybos receptūrą įvyks tuomet, kai bus įvykdytas jo potikslis – patikrinti žaliavų kokybę.

#### 3.2.2. Gamybos planavimo sistemos atsakomybių modelis

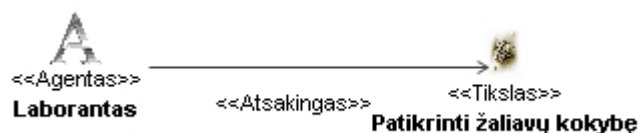
Kitas modelis yra atsakomybių modelis. Tačiau jį dažnai jungia su tikslų modeliu. Mes šuo atveju vaizduojame atskirai.



#### 38 pav. Vadybininko atsakomybių modelis

Šaltinis: sudaryta autoriaus

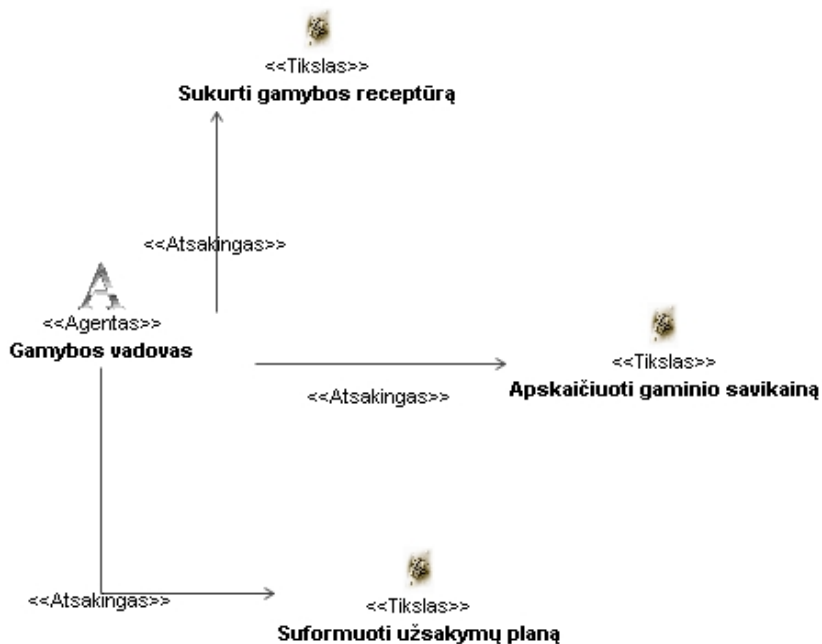
Vadybininkas yra atsakingas už šiuos tikslus: registruoti užsakymą, registruoti žaliavas, registruoti tiekėjus, registruoti klientus.



**39 pav. Laboranto atsakomybių modelis**

Šaltinis: sudaryta autoriaus

Laborantas yra atsakingas už tikslą – patikrinti žaliavų kokybę.



**40 pav. Gamybos vadovo atsakomybių modelis**

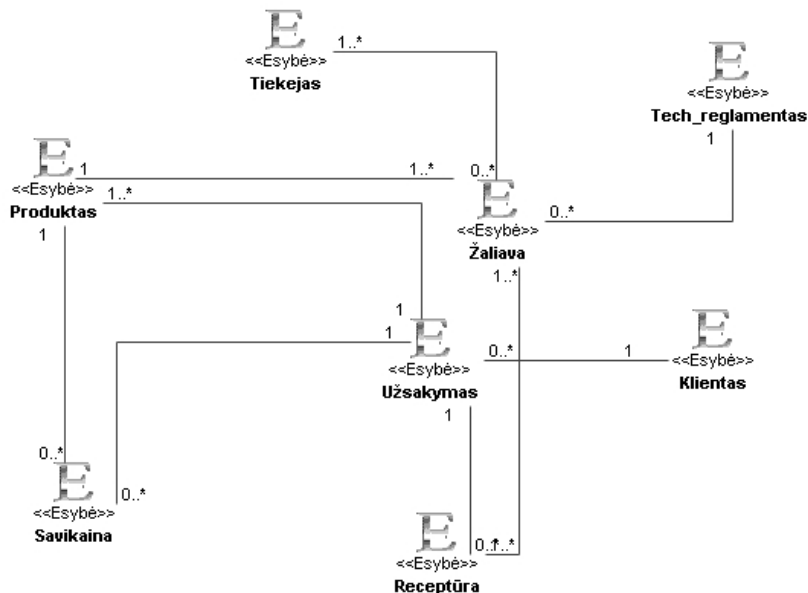
Šaltinis: sudaryta autoriaus

Gamybos vadovas yra atsakingas už keletą tikslų: sukurti gamybos receptūrą, apskaičiuoti gaminio savikainą, suformuoti užsakymų planą.

### 3.2.3. Gamybos planavimo sistemos objektų modelis

Sekantį sudarome objektų modelį.

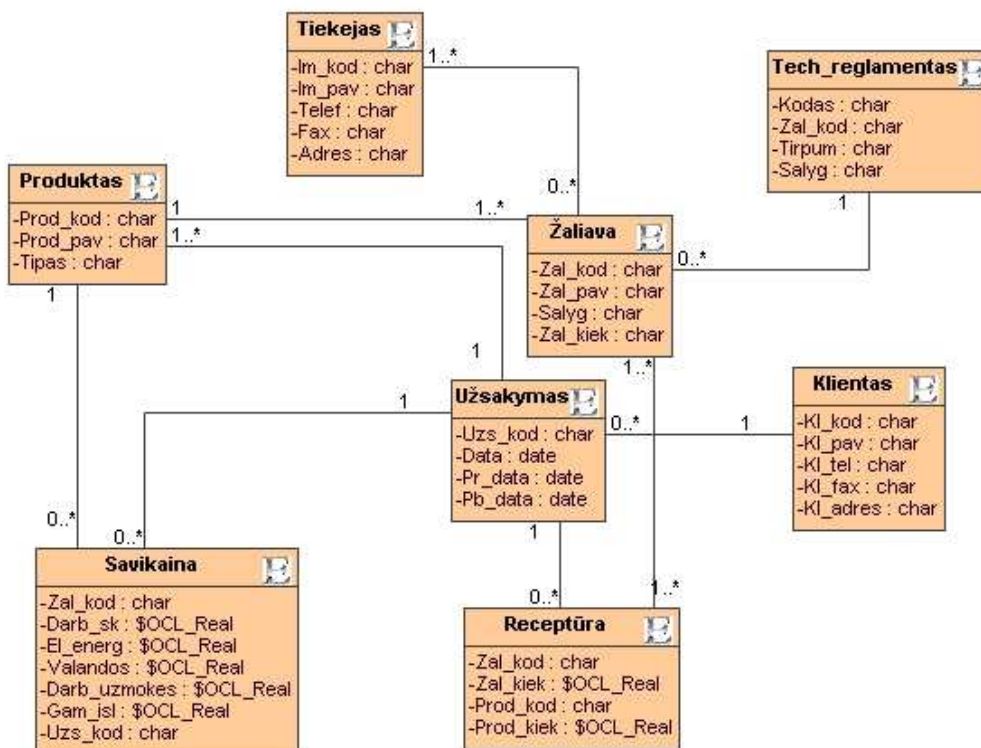




41 pav. Gamybos planavimo sistemos objektų modelis be atributų

Šaltinis: sudaryta autoriaus

Sudarytas objektų modelis be atributų (41 pav.). Jame yra šios esybės: receptūra, klientas, savikaina, žaliava, užsakymas, produktas, tiekėjas, tech\_reglamentas.

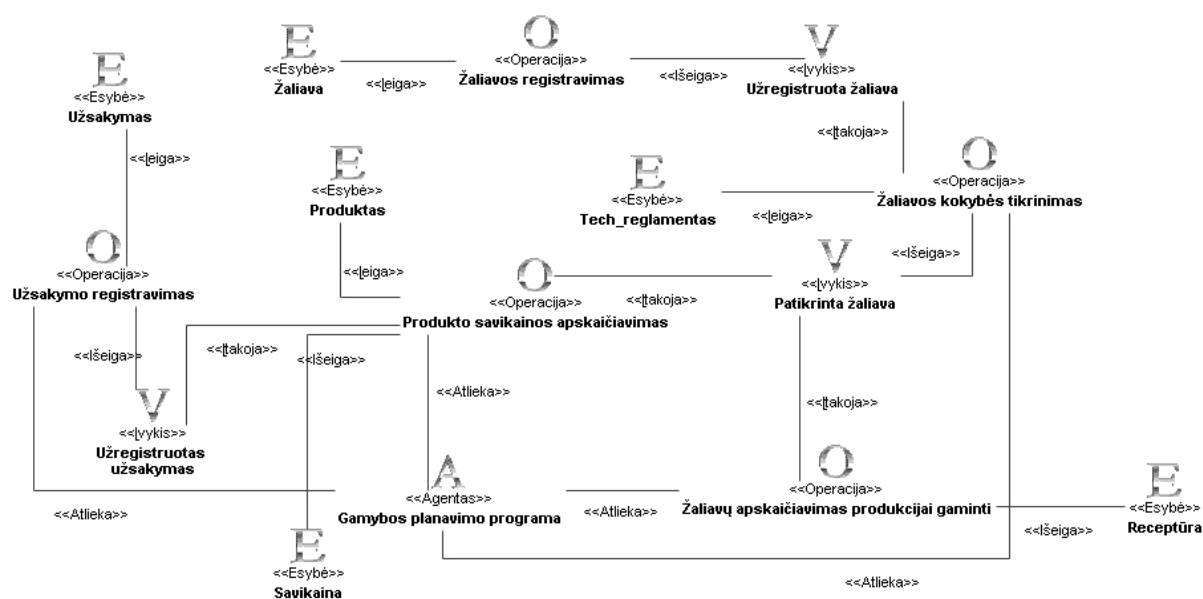


42 pav. Gamybos planavimo sistemos objektų modelis su atributais

Šaltinis: sudaryta autoriaus

Objektų modelyje tiek esybė, tiek agentas sąveikauja su kitomis esybėmis arba agentais tokiais pačiais ryšiais. Šiuo atveju objektais yra esybės, kurios sąveikauja tarpusavyje. Esysbės yra tos pačios kaip ir modelyje be atributų.

### 3.2.4. Gamybos planavimo sistemos operacijų modelis



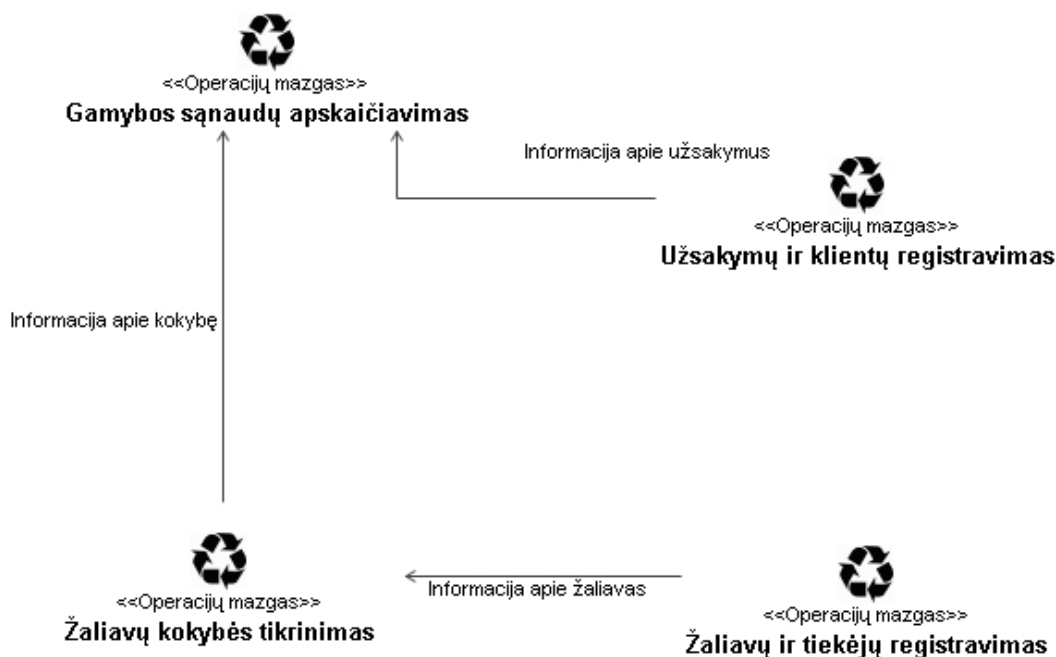
43 pav. Gamybos planavimo sistemos operacijų modelis

Šaltinis: sudaryta autoriaus

Operacijų modelis yra panašus į UseCase modelį, tik Operacijos nustatomos apklausiant dalykinės srities objektus (43 pav.). Įsidėmėtina tai, kad kiekviena operacija turi turėti priešastį, t.y. turi būti atliekama agento arba inicijuojama tam tikro įvykio, kuris gali būti išorinis arba sukurtas kitos operacijos. Mūsų atveju operacijos yra žaliavų registravimas, užsakymo registravimas, žaliavų kokybės tikrinimas, produkto savikainos apskaičiavimas, žaliavų apskaičiavimas produkcijai

gaminti. Esysbės yra tos pačios kaip ir objektų modelyje. Įvykiai yra šie: užregistruota žaliava, patikrinta žaliava, užregistruotas užsakymas.

### 3.2.5. Gamybos planavimo sistemos operacijų mazgų modelis

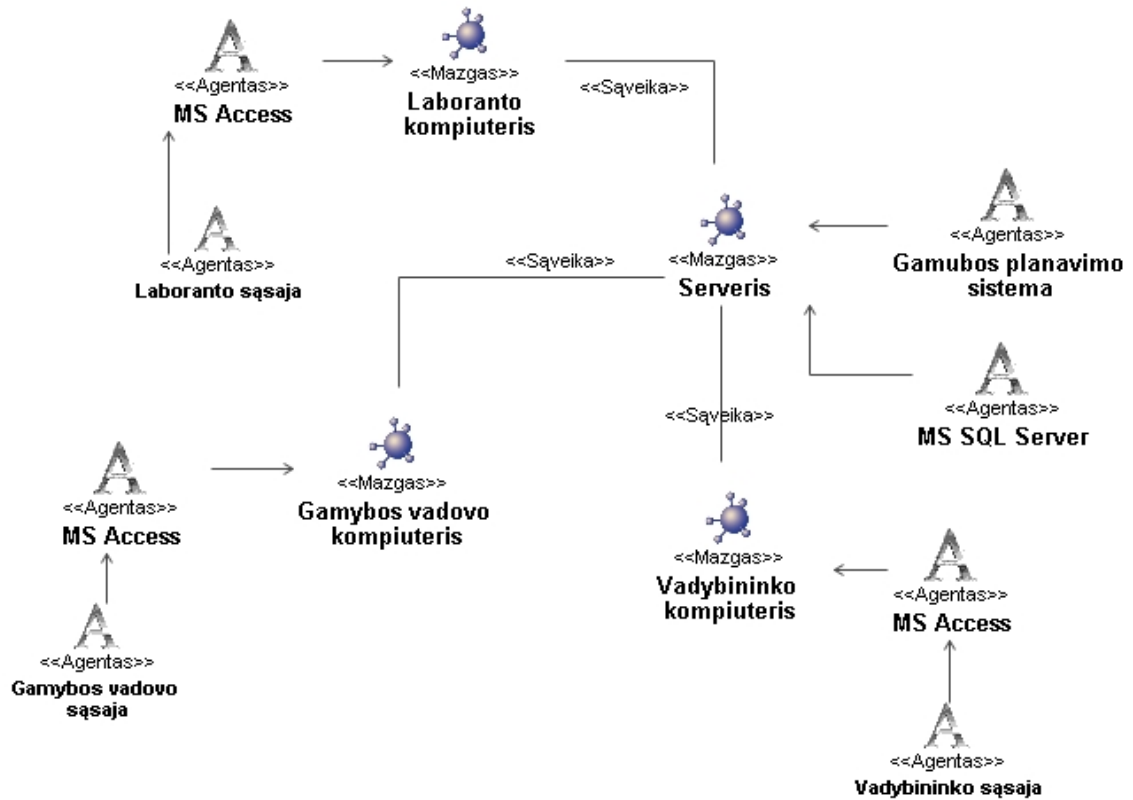


44 pav. Gamybos planavimo sistemos operacijų mazgų modelis

Šaltinis: sudaryta autoriaus

Operacijų mazgų modelis visas operacijas leidžia grupuoti ir pateikti tarpusavio ryšius, t.y. informacijos pasidalijimą identifikuoja. Yra iš viso keturi operacijų mazgai. Mazgas – žaliavų kokybės tikrinimas teikia informaciją apie kokybę kitam mazgui – gamybos sąnaudų apskaičiavimas. Šis savo ruožtu gauna informaciją apie užsakymus iš mazgo – užsakymų ir klientų registravimas. Paskutinis operacijų mazgas – žaliavų ir tiekėjų registravimas teikia informaciją apie žaliavas mazgui – žaliavų kokybės tikrinimas.

### 3.2.6. Gamybos planavimo sistemos sąsajų modelis



45 pav. Gamybos planavimo sistemos sąsajų modelis

Šaltinis: sudaryta autoriaus

Šis modelis parodo sąveikaujančias sistemas (45 pav.). Šis modelis parodo sąveikaujančias sistemas. Sistema veikia per serverį, kuriama saugoma duomenų bazė. Serveris sąveikauja su vartotojais. Gamybos vadovas turi savo sąsają per MS Access, vadybininkas taip pat turi savo vartotojo sąsają per MS Access, laborantas lygiai taip pat. Jų visų kompiuteriai naudojami MS SQL serveriu, kuriame saugoma duomenų bazė.

### 3.3. Tikslais orientuoto ir objektais orientuoto metodų palyginimas

Pirmiausia analizavome veiklą tradiciniu objektais orientuotu metodu. Nustatėme veiklą, kurią analizuosime. Tai UAB „Ventos polimerai“ gamybos palanavimas. Palengvinti gamybos procesui, nuspręsta išanalizuoti šiuos procesus ir pabandyti išskirti reikalavimus kuriamai informacinei sistemai. Tradiciniu metodu buvo sudarytos tokios diagramos: panaudojimo atvejų, veiklos, seku, klasių, išdėstymo diagramos. Tai pagrindiniai UML modeliai, naudojami reikalavimų išsiaiškinimui. Nubraižius tradiciniu metodu modelius, toliau braižėme pagal Magicdraw profilius diagramas. Tokiu būdu patikrinome KAOS papildyto metodo veiksmingumą.

13 lentelė

**Panaudojimo atvejų diagramos ir atsakomybių modelio palyginimas**

	<b>Panaudojimo atvejų diagrama</b>	<b>Atsakomybių modelis</b>
<b>Tikslas</b>		✓
<b>Procesas</b>	✓	
<b>Ryšiai</b>	✓	✓
<b>Savybės</b>	Specifikuoja sistemos funkcijų sąveiką su vartotojais	Specifikuoja sistemos tikslus, už kuriuos atsakingi objektai

Šaltinis: sudaryta autoriaus

Sudarius pirmąją tikslų diagramą, tapo aiškūs informacinės sistemos tikslai, kas už juos atsakingi. Tikslų hierarchija sukonkretina, kas bus siekiama. Tokios informacijos tradicinis metodas iškart neišskiria. Tikslų diagrama pateikia daug informacijos kuriamai informacijos sistemai.

Lyginant panaudojimo atvejų diagramą su atsakomybių diagrama (13 lentelė), atsakomybių modelyje parodoma sistemos tikslų sąveika su objektais, kurie gali būti ir žmogus, ir sistema. O panaudojimo atvejų diagramoje vartotojai yra tik žmonės, kurie valdo atitinkamas sistemos funkcijas. Be to, tam, kad sudarytume sistemos panaudojimo atvejų diagramą, pirmiausia rekomenduojama sudaryti bendrą veiklos panaudojimo atvejų diagramą.

### Sekų diagramos ir operacijų modelio palyginimas

	Sekų diagrama	Operacijų modelis
Ivykis		✓
Operacija		✓
Objektas	✓	✓
Ryšiai	✓	✓
Savybės	specifikuoja dinaminę elgseną tarp objektų chronologine tvarka	Operacijos nustatomos pagal dalykinės srities objektus

Šaltinis: sudaryta autoriaus

Operacijų modelyje žymiai detaliau ir aiškiau aprašomos operacijos, nei sekų diagramoje (14 lentelė). Operacijų diagramoje išskiriamos ir esybės, ir operacijos, ir įvykiai, ir agentai. Kiekviena operacija turi savo priežastį. Sekų diagramoje išskiriami tik objektai.

### Klasių diagramos ir objektų modelio palyginimas

	Klasių diagrama	Objektų modelis
Esybė	✓	✓
Agentas		✓
Atributai	✓	✓
Ryšiai	✓	✓
Savybės	specifikuoja statinę objektų struktūrą, įtraukdama atributus ir operacijas, ryšius	Modelyje gali būti atvaizduojami dviejų tipų objektai: esybės ir agentai

Šaltinis: sudaryta autoriaus

Objektų modelis su klasių diagrama yra panašūs (15 lentelė). Jų tikslumas priklauso nuo to, kaip informacija buvo išanalizuota ankstesnėse diagramose. Objektų modelyje gali būti atvaizduojami dviejų tipų objektai: esybės ir agentai. Klasių modelyje yra tik esybės.

Sistemos sąsajų diagrama atitinka išdėstymo diagramą, todėl jos nubraižytos tokios pat, t.y. pateikia informaciją apie programinės įrangos išsidėstymą.

Operacijų mazgų modelio atitikmens nėra objektiškai orientuotame metode. Šis modelis paimtas iš DoDAF metodoogijos.

### 3.4. Papildyto KAOS metodo palyginimas

KAOS metodas buvo papildytas dviem DoDaf modeliais, kurie papildė KAOS modelius ir suteikė daugiau funkcijų. Sistemos projektavimas buvo atliktas su papildytu metodu. Tokiu būdu daugiau labiau identifikuoti buvo reikalavimai kuriamai sistemai (16 lentelė).

16 lentelė

**KAOS metodo ir papildyto KAOS metodo palyginimas**

	KAOS metodas	Papildytas KAOS metodas DoDaf modeliais
Tikslų atvaizdavimas	✓	✓
Atsakomybių atvaizdavimas	✓	✓
Objektų atvaizdavimas	✓	✓
Operacijų atvaizdavimas	✓	✓
Sistemos sąsajų atvaizdavimas		✓
Operacijų mazgų atvaizdavimas		✓

Šaltinis: sudaryta autoriaus

Projektuojant gamybos planavimo sistemą papildyti modeliai padėjo suprojektuoti sistemą. Sistemos sąsajų identifikavimas modeliavo programą vykdančias komponentes ir programinę įrangą, kuri dirba su šiomis komponentėmis. Operacijų mazgų modelis padėjo struktūrizuoti operacijas. Operacijų modelyje identifikuotų operacijų yra daug, todėl labai patogu ir naudinga sugrupuoti ir pateikti operacijų mazgų tarpusavio ryšius. Tokia informacija padeda įvertinti trūkstamas operacijas ir ar tinkamai išskirtos operacijos.

Tikslais grindžiamas KAOS metodas yra geras būdas vartotojams valdyti reikalavimus, nes tikslai siūlo tinkamą abstrakcijų lygmenį, leidžiantį sprendimo priėmėjams patvirtinti pasirinkimus, padarytus tarp alternatyvų ir siūlant kitas alternatyvas. Šis metodas yra tinkamas naudoti, kuomet kai reikia modeliuoti tikslus, pagrindines funkcijas, atsakomybes už tikslus ir funkcijas bei statinius ir dinامينius objektus.

## IŠVADOS

1. Tikslais orientuota reikalavimų inžinerija apima platesnę sistemos projektavimo perspektyvą lyginant su tradiciniais reikalavimų inžinerijos metodais: srities ypatybės nustatomos iš reikalavimų detalizavimo proceso.
2. Tikslais orientuotas modeliavimas pateikia paprastą mechanizmą, kaip suformuoti sudėtingus reikalavimų dokumentus.
3. Viena iš reikalavimų inžinerijos problemų yra daugialypių požiūrių konfliktų valdymas. Tikslai yra naudojami, kad valdytų konfliktus tarp reikalavimų.
4. Sistemos projektavimui pasirinkta KAOS metodologija, nes atlieka reikalavimų detalizavimą, susiedama tikslus su funkciniais ir nefunkciniais komponentais.
5. KAOS metodologija nėra skirta verslo sistemai modeliuoti, vis tik ji gali būti taikoma kai kuriais biznio sistemos modeliavimo atvejais, tokiais kaip tikslų, pagrindinių funkcijų, atsakomybių už tikslus ir funkcijas bei statinių ir dinaminių objektų modeliavimui.
6. KAOS metodologiją nuspręsta papildyti šiais DoDAF modeliais: sistemos sąsajų modelis, operacijų mazgų modelis. Jais papildytas metodas užtikrina visapusišką reikalavimų inžinerijos realizaciją kuriamai informacinei sistemai.
7. KAOS notacija nėra gerai žinoma. Modeliuojant KAOS UML kalba, nereikia naujos programinės įrangos, modeliuojančios specifinę KAOS notaciją. Todėl KAOS metodo modeliams sukurti profiliai su Magicdraw UML įrankiu.
8. Buvo suprojektuota gamybos planavimo sistema. Naudojantis sukurtais profiliais, modelių sudarymas tapo patogus ir greitas.
9. Suprojektuota gamybos planavimo sistema pateikia visapusišką reikalavimų identifikavimą, todėl papildytas KAOS metodas yra tinkamas naudoti sistemų projektavimui.



## LITERATŪRA

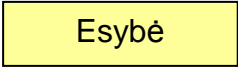


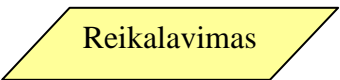
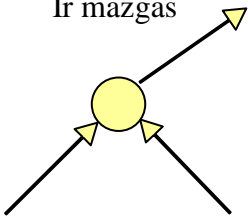
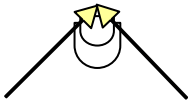
1. ŠARKIŪNAITĖ I., KRIKŠČIŪNIENĖ D., SIMUTIS R (2007). Magistro BD metodiniai nurodymai 2007 [interaktyvus]. [žiūrėta 2007 m. gegužės 21 d.].
2. KAVAKLI, Evangelia. (2003) *Goal Driven Requirements Engineering: Evaluation of Current Methods* [interaktyvus]. [žiūrėta 2008 m. vasario 13 d. ]. Prieiga per internetą: <<http://www.emmsad.org/2003/Final%20Copy/05.pdf>>
3. STIRNA, Janis. (2007) *Ten Years Plus with EKD: Reflections from Using an Enterprise Modeling Method in Practice* [interaktyvus]. [žiūrėta 2007 m. gegužės 21 d.]. Prieiga per internetą: <<http://www.via-nova-architectura.org/files/EMMSAD2007/Stirna.pdf>>
4. Stirna, Janis. (2007) *EKP - Enterprise Knowledge Patterns* [interaktyvus]. [žiūrėta 2008 m. balandžio 16 d.]. Prieiga per internetą: <[http://people.dsv.su.se/~js/ekp/ekd\\_method.html](http://people.dsv.su.se/~js/ekp/ekd_method.html)>
5. Software Measurement Services (2004) *The GQM Method* [interaktyvus]. Software Measurement Services. [žiūrėta 2007 gruodžio 3 d.]. Prieiga per internetą: <<http://www.measuresw.com/services/tools/gqm.html>>
6. SOLINGEN, R. (1999) *GQM method application* [interaktyvus]. [žiūrėta 2008 sausio 15 d.]. Prieiga per internetą: <<http://ivs.cs.uni-magdeburg.de/sw-eng/us/java/GQM/link1.shtml>>
7. CARES, Carlos (2005) *A Comparative Analysis of i\*-Based Agent-Oriented Modeling Languages* [interaktyvus]. [žiūrėta 2007 m. gruodžio 12 d.]. Prieiga per internetą: <<http://www.lsi.upc.edu/~cayala/Papers/2005-AOSDM-camera-ready.pdf>>
8. LIU, Lin. (2002) *From Requirements to Architectural Design –Using Goals and Scenarios* [interaktyvus]. [žiūrėta 2008 m. kovo 1o d.]. Prieiga per internetą: <[http://www.cs.toronto.edu/~liu/publications/From\\_R2A-revised.pdf](http://www.cs.toronto.edu/~liu/publications/From_R2A-revised.pdf)>
9. DALONS, Gautier. (2004) *A template-based analysis of GRL* [interaktyvus]. [žiūrėta 2007 m. gruodžio 10 d.]. Prieiga per internetą: <<http://www.via-nova-architectura.org/files/EMMSAD2005/Dallons.pdf>>
10. University of Toronto (2007) *GRL - Goal-oriented Requirement Language* [interaktyvus]. University of Toronto Canada.[žiūrėta 2007 m. gruodžio 12 d.]. Prieiga per internetą: <<http://www.cs.toronto.edu/km/GRL/>>
11. A KAOS Tutorial, (2003) *Objectiver*, [interaktyvus]. [žiūrėta 2008 m. Sausio 10 d.]. Prieiga per internetą: <[www.objectiver.com/download/documents/KaosTutorial.pdf](http://www.objectiver.com/download/documents/KaosTutorial.pdf)>
12. MYLOPOULOS, John *Goal-Oriented Requirements Engineering: Part II*; 14th IEEE Requirements Engineering Conference, Minneapolis, September 15, 2006 [interaktyvus]. [žiūrėta 2008 m. kovo 24 d.]. Prieiga per internetą: <[http://www.ifi.unizh.ch/req/events/RE06/ConferenceProgram/RE06\\_slides\\_Mylopoulos.pdf](http://www.ifi.unizh.ch/req/events/RE06/ConferenceProgram/RE06_slides_Mylopoulos.pdf)>
13. WENTZ, Andreas. (2004) *KAOS – a goal-driven RE approach*. [interaktyvus]. [žiūrėta 2007 m. lapkričio 23 d.]. Prieiga per internetą: <<http://www.info.ucl.ac.be/~avl/ReqEng.html>>.


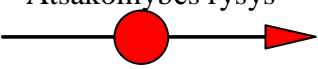
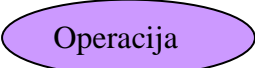
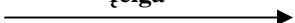
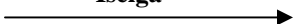
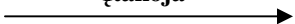

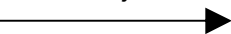

14. LAMS, (2005) *Goal Driven Requirements Engineering Overview*; 2002-2005 Marc-André Lüthi [interaktyvus]. [žiūrėta 2007 m. lapkričio 24 d.]. Prieiga per internetą: <<http://lamswww.epfl.ch/reference/goal/#1.1.%20Some%20History> >
15. DODAF (2006). *DoD Architecture Framework Version 1.5* [interaktyvus]. [žiūrėta 2007 m. lapkričio 24 d.]. Prieiga per internetą: <[http://www.defenselink.mil/cionii/docs/DoDAF\\_Volume\\_II.pdf](http://www.defenselink.mil/cionii/docs/DoDAF_Volume_II.pdf)>
16. Modeling DoDAF (2004) *Modeling DoDAF Compliant Architectures* [interaktyvus]. [žiūrėta 2008 01 15]. Prieiga per internetą: <[http://www.uml-forum.com/docs/papers/White\\_Paper\\_Modeling\\_DoDAF\\_UML2.pdf](http://www.uml-forum.com/docs/papers/White_Paper_Modeling_DoDAF_UML2.pdf) >
17. MATULEVIČIUS R. (2006) *Ontological Analysis of KAOS Using Separation of Reference* [interaktyvus] [žiūrėta 2008 m. sausio 15 d.]. Prieiga per internetą: <<http://www.via-nova-architectura.org/files/EMMSAD2006/Matulevicius2.pdf>>
18. MYLOPOULOS, John (2004) *Conceptual Modeling* [interaktyvus], [žiūrėta 2008 m. balandžio 14 d.]. Prieiga per internetą: <<http://www.cs.toronto.edu/~jm/2507S/Notes04/KAOS.pdf> >
19. BROWN Greg (2006). *Goaloriented Specification of Adaptation Requirements Engineering in Adaptive Systems* [interaktyvus] [žiūrėta 2008 m. sausio 15 d.]. Prieiga per internetą: <<http://www.cse.msu.edu/~zhangji9/publications/Brown06Goal.pdf>>
20. ZHANG Charles, JACOBSEN Arno (2004) *Linking Goals to Aspects* [interaktyvus]. [žiūrėta 2008 m. sausio 15 d.]. Prieiga per internetą: <<http://www.msrg.utoronto.ca/publications/msrg-Jan-05-re.pdf> >
21. MAIBAUM Tom (2005) *Evaluating the Effectiveness of a Goal-Oriented Requirements Engineering Method* [interaktyvus]. [žiūrėta 2008 m. vasario 15 d.]. Prieiga per internetą: <<http://www.di.unipi.it/CERE06/AlSubaie.pdf> >
22. ANDRE Marc (2002) *Goal Driven Requirements Engineering Overview* [interaktyvus]. [žiūrėta 2008 m. vasario 3 d.]. Prieiga per internetą: <<http://lamswww.epfl.ch/reference/goal>>
23. ARENAS Alvaro, AZIZ Bejnamin (2007) *Modelling Security Properties in a Grid-based Operating System with Anti-Goals* [interaktyvus]. [žiūrėta 2008 m. vasario 15 d.]. Prieiga per internetą: <<http://epubs.cclrc.ac.uk/bitstream/1979/aziz-securityGrid.pdf>>
24. ERNST Neil (2002) *Integrating requirements engineering and cognitive work analysis: A case study* [interaktyvus]. [žiūrėta 2008 m. kovo 15 d.]. Prieiga per internetą: <<http://www.neilernst.net/docs/pubs/paper142.pdf>>
25. AIB Issam (2007) *A Business-Driven Approach to Policy Optimization* [interaktyvus]. [žiūrėta 2008 m. kovo 27 d.]. Prieiga per internetą: <<http://www.cs.uwaterloo.ca/~iaib/publications/thesis-issam-aib.pdf>>
26. POPOVA Viara, SHARPANSKYKH Alexei (2006) *Formal Modelling of Goals in Organizations* [interaktyvus]. [žiūrėta 2008 m. kovo 27 d.]. Prieiga per internetą: <[http://dare.uvu.vu.nl/bitstream/1871/11732/1/goals\\_popova\\_sharpanskykh.pdf](http://dare.uvu.vu.nl/bitstream/1871/11732/1/goals_popova_sharpanskykh.pdf)>

27. BOCK Konrad (2000) *Goal-driven Modeling* [interaktyvus]. [žiūrėta 2008 m. kovo 27 d.]. Prieiga per internetą: <<http://www.conradbock.org/goal.html>>
28. JURETA Ivan (2005) *Engineering Requirements for Information Systems using KAOS and Request frameworks Modeling* [interaktyvus]. [žiūrėta 2008 m. kovo 27 d.]. Prieiga per internetą: <<http://www.isys.ucl.ac.be/staff/stephane/GETI2100Slide/KAOS.pdf>>
29. HEAVEN William, FINKELSTEIN Anthony (2006) *A UML profile to support requirements engineering with KAOS* [interaktyvus]. [žiūrėta 2008 m. kovo 27 d.]. Prieiga per internetą: <<http://www.cs.ucl.ac.uk/staff/A.Finkelstein/papers/umlreprofile.pdf> >

## PRIEDAS

## KAOS notacija

Elementas	Aprašymas
	<p>Esybės apibūdina nepriklausomus, pasyvius objektus. Nepriklausomus, nes esybės egzistavimui neturi įtakos kiti objektai. Pasyvus, šiuo atveju apibūdina tai, kad objektas negali atlikti jokios operacijos. Esybe pvz., gali būti durys, lempa ir pan. Esybės gali turėti atributus.</p>
	<p>Agentas – aktyvus, nepriklausomas objektas. Agentas nuo esybės skiriasi tuo, kad gali atlikti įvairias operacijas – tai nėra pasyvus objektas. Agentas gali būti žmogus, mašina, kitas objektas galintis atlikti tam tikrą operaciją. Agentai gali turėti atributus.</p>
<p>____ <u>Asociacija 1..n</u> _____</p>	<p>Asociacija – pasyvus priklausomas objektas, apibūdinantis dviejų esybių sąsają. Asociacijos atributas parodo sąsajos daugybiškumą. Asociacija pvz., gali apibūdinti sąsają tarp būto ir durų, kur daugybiškumas bus durų skaičius bute.</p>
<p>Intereso ryšys</p> <p>Siejasi →</p>	<p>Intereso ryšys suriša reikalavimą (tikslą) su objektu reikalingu šiam reikalavimui įgyvendinti.</p>
	<p>Tikslas arba kitaip vadinamas reikalavimas, tai bene svarbiausias notacijos elementas. Tikslai jungiami į hierarchinius medžius, hierarchiniais ryšiais. Tikslai gali būti įgyvendinti agentų.</p>
	<p>Reikalavimas – tai tas pats tikslas, su šiek tiek kitokia prasme. Reikalavimas apibūdina tikslą, kuris privalo būti įgyvendintas siekiant aukštesnio tikslo įgyvendinimo, tačiau dėl aplinkos sąlygų gali būti neįgyvendintas.</p>
<p>Tobulinimo ryšys</p> <p>→</p>	<p>Tobulinimo ryšys, tai pats paprasčiausias hierarchinis ryšys tarp dviejų tikslų.</p>
<p>Ir mazgas</p> 	<p>Šis mazgas dedamas ant tobulinimo ryšio ir reiškia, jog panaudojus šį mazgą visi žemiau esantys tikslai privalo būti įgyvendinti, norint, kad tėvinis jų atžvilgiu tikslas būtų įgyvendintas.</p>
<p>Arba mazgas</p> 	<p>Arba mazgas dedamas ant tobulinimo ryšio ir reiškia, jog jį panaudojus turi būti įvykdytas bent vienas iš sūninių tikslų norint, kad būtų įvykdytas tėvinis tikslas.</p>

<p>Konfliktas tarp tikslų</p> 	<p>Konfliktas tarp tikslų apibūdina tokią situaciją, kuriai esant tikslai logiškai vienas kitam prieštarauja.</p>
<p>Atsakomybės ryšys</p> 	<p>Atsakomybės ryšys naudojamas parodyti, koks agentas atsakingas už tam tikro tikslo įgyvendinimą. Rodyklė visuomet yra tikslo pusėje.</p>
<p>Operacija</p> 	<p>Operacijas atlieka agentai. Operacija – tai tam tikras veiksmas, kuris sukuria objektą, pakeičia objekto atributo reikšmę, išskviečia įvykį.</p>
<p>Įėjimo ryšys Įeiga</p> 	<p>Įėjimo ryšys suriša agentą su operacija, t.y. parodo, koks agentas atlieka operaciją.</p>
<p>Išėjimo ryšys Išeiga</p> 	<p>Išėjimo ryšys parodo ką įtakoja atlikta operacija. Tai gali būti ryšys su esybe arba įvykiu.</p>
<p>Priežasties ryšys Įtakoja</p> 	<p>Priežasties ryšys artimas įėjimo ryšiui, tačiau esminis skirtumas tarp šių ryšių yra tai, kad priežasties ryšys parodo įvykio sąsają su operacija, o įėjimo ryšys – agento sąsają su operacija.</p>
<p>Įvykis</p> 	<p>Įvykis apibūdina tam tikrą ženklą, kuris gali būti tiek išorinis, tiek vidinis, t.y. sukeliamas tam tikros operacijos. Įvykis gali įvykdyti arba sustabdyti operaciją.</p>
<p>Paveldėjimas</p> 	<p>Paveldėjimas, tai ryšys tarp objektų, parodantis objektų hierarchiškumą.</p>
<p>Atlikimo ryšys Atlieka</p> 	<p>Atlikimo ryšys apibūdina koks agentas atlieka kokią operaciją.</p>

Šaltinis: Sudaryta autoriaus pagal A KAOS Tutorial, Objectiver