



**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA**

Giedrė Drašutavičiūtė

**VEIKLOS TAISYKLIŲ VAIZDAVIMAS DUOMENŲ BAZĖS
STRUKTŪRŲ SCHEMOMIS**

MAGISTRO DARBAS

Vadovas: doc. B. Paradauskas

KAUNAS, 2005

TURINYS

1. Įvadas.....	3
2. Šiuolaikinė veiklos taisyklės samprata	4
2.1 Veiklos taisyklė.....	4
2.2 Reikalavimai veiklos taisyklių formalizavimui.....	4
2.3 Priemonių verslo taisyklių informacinėse sistemose realizuoti analizė	7
2.4 Apibendrinimas.....	8
3. Veiklos taisyklių struktūrizavimo modelių analizė.....	8
3.1 Veiklos taisyklių struktūrizavimo modeliai.....	8
3.1.1 GUIDE veiklos taisyklių projektas.....	9
3.1.2 R.Roso metodas.....	21
3.1.3 Veiklos taisyklių klasifikacija projektuotojo požiūriu.....	24
3.1.4 Veiklos taisyklių struktūrizavimo modelių palyginimas.....	27
4. Veiklos taisyklių modeliavimui naudojamos kalbos, sistemos.....	28
4.1 UML ir OCL.....	28
4.2 Duomenų modelių vaizdavimas duomenų struktūrų schemomis.....	32
4.3 Apibendrinimas.....	36
5. Veiklos taisyklių struktūrizavimo proceso modelis.....	37
6. Eksperimentinės sistemos realizacija.....	41
6.1 Veiklos taisyklių rinkinio sudarymas.....	41
6.2 Projektavimo dalis.....	48
7. Sistemos realizacija.....	58
7.1 Pagrindinis meniu.....	58
7.2 Pasirinkimas Tvardaraštis.....	59
7.3 Pasirinkimas Tvardaraštis info.....	60
7.4 Pasirinkimas Auditorijos (Pradinių domenų įvedimas).....	62
7.5 Pasirinkimas Grupės (Pradinių domenų įvedimas).....	63
7.6 Pasirinkimas Paskaitų laikai (Pradiniai duomenys).....	64
7.7 Pasirinkimas Dėstytojai (Laiko fondai).....	65
7.8 Pasirinkimas Auditorijos (Laiko fondai).....	66
7.9 Pasirinkimas Grupės (Laiko fondai).....	66
8. Išvados.....	67
Literatūros šaltiniai.....	69
Priedas.....	70

1. Įvadas

Pastaruoju metu niekas neabejoja žinių panaudojimo naudingumu IS kūrimo etape. Labai svarbus uždavinys šiuolaikiniame IS kūrimo etape yra tiksliai ir aiškiai nustatyti vartotojo poreikius, surinktas žinias kaupti ir saugoti. Augant vartotojų poreikiams, didėjant informacijos apdorojimo poreikiui dinamiškai kintančioje aplinkoje, tradiciniai IS projektavimo metodai nėra pakankamai efektyvūs. Todėl šiuo metu labai daug dėmesio skiriama verslo modeliavimui įvairiais požiūriais, siekiant sukurti vieningą sistemą. Modeliuojant verslo informacinę sistemą būtina atkreipti dėmesį, kad verslo taisyklės paprastai keičiasi greičiau, negu probleminės srities duomenų struktūros ar objektų elgsena. Dažnai reikia verslo taisyklių rinkinį papildyti, peržiūrėti, keisti. Modeliavimas verslo taisyklių požiūriu yra naujas modeliavimo būdas, paremtas verslo taisyklių išskyrimu. Kuriant IS viena iš problemų yra veiklos taisyklių stuktūrizavimas. Labai svarbu aiškiai suprasti reikalavimų nustatymo etape veiklos procesus, jų keliamus apribojimus. Šio darbo tyrimo objektas – veiklos taisyklės, jų vaizdavimas duomenų bazės struktūrų schemomis.

Darbo tikslas - ištyrus veiklos taisykles, nustatyti ir išskirti tik tas taisykles, kurias po to galime struktūriškai realizuoti duomenų bazės struktūromis.

Šiuo metu daugiausia naudojami du analizės metodai: analizuojama taikant struktūrišką projektavimą arba pateikiami veiklos aprašymai naudojant unifikuotą modeliavimo kalbą UML(Jacobson 2001). Dar yra žinomi ir kiti analizės metodai, tokie kaip Petri tinklai, IDEA, kurie nėra taip plačiai išplitę.

Informacinių sistemų (IS) kūrimo naudojant vien tik grafinius duomenų atvaizdavimo modelius, be aiškios jų realizacijos, turi trūkumų. Duomenų modeliai aprašo tik duomenų struktūras. Mažai dėmesio skiriama jų praktiniam realizavimui, panaudojimui. Todėl įvesdami veiklos taisykles IS projektavimo metodu, bandome išspręsti šiuos trūkumus.

Jau pirmuose straipsniuose pradėtiems publikuoti prieš dešimtmetį, mokslininkai apie veiklos taisykles pradėjo nagrinėti, kaip neišvengiamą projektavimo etapą, be kurio neįmanoma pilnai apibrėžti visos organizacijos veiklos. Šiuo metu skaičius įmonių, naudojančių veiklos taisyklių principą, nėra daug, tačiau jų skaičius sparčiai didėja.

Veiklos taisyklių pranašumai prieš tradicinį požiūrį (Kapočius 2002):

- Egzistuojantys informacinių sistemų aprašymo ir projektavimo modeliai neefektyviai atspindi dalykinėje srityje galiojančias taisykles, jos padrikai paskirstytos po visą sistemą. Veiklos taisyklių nepriklausomumo sluoksnio koncepcijos taikymas išsprendžia šią problemą, kadangi taisykles saugomos vientisoje nepriklausomoje taisyklių saugykloje.

- Tradiciniais metodais sukurtoje IS pakeitimo ar pritaikymo prie pasikeitusios situacijos kaštai yra pernelyg dideli. Veiklos taisyklėmis paremtoje sistemoje, esant poreikiui, nesunkiai jas gali keisti patys verslo atstovai, analitikų pagalba nereikalinga.

Kadangi analizuojamas objektas bus susijęs su informacine sistema, tai naudinga būtų pasirinkti projektavimo metodą, kuris nuo ankstyvosios projektavimo fazės iki galutinės realizacijos nagrinėja veiklos procesus organizacijoje. Projektavimui, pateiktoms diagramoms kurti ir analizuoti panaudotos komunikacinės kilpos, vaizduojamos diagramos, naudojant projektuotojo požiūriu paremtą klasifikavimo būdą, išskiriamos veiklos taisyklės, kurias po to realizuojamos atitinkamai arba programos kode arba duomenų bazės duomenų modelyje.

Analitinėje darbo dalyje pateikiama bendra VT koncepcijos analizė, aptariami ir palyginami svarbiausi žinomi VT struktūrizavimo metodai, pateikiami VT išskyrimo bei analizės principai. Tyrimo dalyje pasiūlomas VT klasifikavimo bei VT kūrimo proceso modelis. Pagaliau eksperimentinėje dalyje aprašomas tvarkaraščio sudarymo uždavinio dalykinė sritis. Darbo pabaigoje pateikiamas išvados. Papildomai autoriaus paskelbtas straipsnis pateikiamas priede.

2. Šiuolaikinė veiklos taisyklių samprata

Šiame skyrelyje bus apžvelgti pastarųjų keleto metų darbai, atlikti veiklos taisyklių tyrinėjimo srityje. Pateiktos analizės lyginimo rezultatai, išvados.

2.1 Veiklos taisyklė

Skirtinguose šaltiniuose veiklos taisyklė yra apibrėžiama panašiai. Ir GUIDE(Kolber 1997) projekto autoriai ir OMG (Omg 2001) grupės veiklos taisyklės (VT) apibrėžia dviem aspektais:

- Veiklos taisyklė – tai verslo požiūriu yra direktyva, skirta įtakoti ar valdyti verslo elgseną, tuo būdu realizuojant verslo politiką.
- IS požiūriu verslo taisyklė yra teiginys, kuris nusako ar apriboja, kuri nors verslo aspektą, deklaruoja verslo struktūrą, kontroliuoja arba kitaip įtakoja verslo procesus.

Darbe (Gottesdiener 1997) taip pat pateikiamas veiklos taisyklės apibrėžimas, jų skaidymo būdas. Kadangi veiklos taisyklė nusakyta veiklos žmogaus, dažniausiai įvardijama dviprasmiškai ir negriežtai, todėl dažnai veiklos taisyklės galima išskaidyti į detalesnius veiklos atvejus.

Išskaidomos veiklos taisyklės iki savo elementariosios formos, jos dalomos tol, kol apsiribojama viena išbaigta mintimi.

2.2 Reikalavimai veikos taisyklių formalizavimui

Norint geriau suvokti veiklos taisyklių esmę, reikia gerai žinoti jiems keliamus reikalavimus. Dažniausiai šie reikalavimai būna kaip rinkinys pateikiamas kartu su veiklos taisyklių klasifikavimo modeliais, tačiau yra išskiriami ir bendri reikalavimai bendri visoms veiklos taisyklėms (Plotkin 1999). Taisyklės turi būti glaustos ir pateiktos tokia forma, kad užtikrinti formalizavimo ir taisyklių saugyklos sudarymo vientisumą.

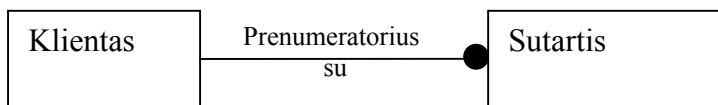
- Deklaratyvios – pažymėtina, kad veiklos taisyklės yra labiau deklaratyvios, nei procedūrinės prigimties.
- Tikslī išraiška – veiklos taisyklę reikia išreikšti tiksliai: grafiškai arba formalia kalba. Struktūrinės taisyklės gali būti išreikštos esybių ryšių (ER) diagrama. Įvykių rezultatai gali būti išreikšti duomenų srautų diagramomis. Tikslī išraiška, naudojant grafinę anotaciją yra gerai parodytas Rosso metode (Ross 1997).
- Atominė – logiškai nedali. VT turi būti nedaloma t.y. jei mes veiklos taisyklę skaidome galime prarasti informaciją.
- Nuosekli – negali būti konfliktuojančių taisyklių.
- Neperteklinė – veiklos taisyklių aibėje negali būti tą pačią informaciją konstatuojančių taisyklių.
- Orientuotos į veiklą – veiklos taisyklė privalo būti formuluojama naudojant terminus, suprantamus konkrečios veiklos srities atstovams.
- Priklausomos veiklai – veiklos taisyklės formuluoti, keisti, paskelbti negaliojančiomis gali tik veiklos atstovai.

Visais atvejais veiklos taisyklės turi būti griežtos, nes jų pagrindu generuojamas kodas (Nemuraitė 1999). Veiklos taisyklės išreikštos deklaratyvia forma Pvz.: „Apmokėti tiekėjo važtaraštį tik tuomet, jei jis buvo patvirtintas“, reikia sugebėti jas išreikšti ir natūralia kalba. Griežtai išreikšta anksčiau minėta taisyklė atrodo taip: Apmokėti *Tiekėjo* Važtaraštį tik tada, jeigu jo statusas yra „*Patvirtintas*“.

Veiklos taisyklių kategorijos

Veiklos taisyklių bendrumas, kiekviena veiklos taisyklė gali būti priskirta vienam iš šių tipų, kaip pateikiama (Butleris 1999) darbe:

Struktūriniai teiginiai – apibrėžta sąvoka arba veiklos teiginys, išreiškiantis tam tikrą objekto struktūros aspektą. Struktūrinis teiginys vartojamas kaip probleminės srities sąvoka arba yra susietas su kitu teiginiu. Ir suprantamas kaip statinis aspektas.



1 pav. Struktūrinio teiginio pavyzdys

Struktūriniai teiginiai yra dviejų rūšių: sąvokos ir ryšių aprašai. Kliento ir sutarties pavyzdyje (1 pav.) ryšio aprašas nusako, kad kiekviena sutartis gali būti sudaryta su klientu ir kiekvienas klientas gali būti prenumeratorius daug sutarčių.

Elgsenos apribojimai – teiginiai apie apribojimus ir sąlygas, apibrėžiantys reikalavimus organizacijos valdymui. Jie nusako dinaminis veiklos aspektus. Specifikuoja rezultatus, kurie yra veiksmų pasekmė. Naudojami žodžiai: „privalo“, „negali“.

Elgsenos apribojimus skirstome į tris pagrindines klases:

Sąlyga – taisyklė, kad jei kažkas atsitiko, tai reiškia pritaikyti kitą veiklos taisyklę. „Ar studentas pateikė studento pažymėjimą?“

Pilnumo sąlyga – taisyklė, kuri privalo būti teisinga. Jei sąlygos apribojimas tik tikrina apribojimą „Ar turi studentas studento pažymėjimą?“, ir tada imtis veiksmų, tai šiuo atveju užduodama griežta sąlyga: „Studentas privalo turėti studento pažymėjimą“ ir uždrausti visus veiksmus, kurie pažeidžia sąlygą.

Teisė – teiginys nusakantis prievolę, remiantis viena ar daugiau sąlygų. Pvz. (Tik) vartotojas gali atlikti veiksmą.

Teisės suteikiamos tipams, kurie gali savarankiškai atlikti veiksmą t.y. žmonėms, skyriams ir t.t. Pvz. „Tik registratorė gali perrašyti prenumeratą kitu adresu“.

Kilmės apibrėžimai – žinomos taisyklės išvestos iš kitų žinių. Jis yra gaunamas iš aplinkos ir saugomos sistemoje. Išvestinis teiginys yra sukuriamas loginio išvedimo ar matematinio skaičiavimų pagalba iš veiklos terminų, struktūrinių teiginių, kitų išvestinių teiginių ar veiklos apribojimų.

2.3 Priemonių verslo taisyklių informacinėse sistemose realizuoti analizė

Verslo taisyklės gali būti realizuotos kaip taikomosiose programose taip ir duomenų bazių valdymo sistemose. Kai turime taisykles kurias dažnai įmonėse vykstantys procesai reikalauja projektuoti, redaguoti, trinti, valdyti, tam kad palengvinti darbą su tokiais verslo taisyklėmis yra naudojamos taip vadinamos verslo taisyklių valdymo sistemos. Tokių sistemų esmė yra atskirti taisykles nuo taikomųjų programų bei duomenų bazių, tai gali būti kaip loginis arba fizinis atskyrimas. Šios taisyklės yra atskiriamos ir saugomos, projektuojamos atskirai taisyklių saugykloje. Tokias taisykles gali keisti ne tik programuotojai bet ir patys verslo subjektai.

Verslo taisyklių valdymo sistemos naudojamos tam kad taisyklių valdymo ir vykdymo procesas būtų greitesnis ir efektyvesnis, Tam pasiekti sistemoje yra naudojami sudėtingi algoritmai, kompiuterių resursų optimizavimui pasiekti vykdant tokias taisykles. Norint verslo taisyklių valdymo sistemą integruoti į verslo sistemą kiekviena taisyklė yra užrašoma kaip nepriklausomas teiginys, tokiu atveju veiklos taisyklė dabar dirbs kaip nepriklausomas autonominis vienetas, kuris bus atskirtas nuo programinės dalies.

Bet kiek būtų privalumų naudoti tokias sistemas, tačiau ne visais atvejais yra tikslinga. Verslo taisyklių valdymo sistemas tikslinga naudoti kai:

- Verslo sistema apima sprendimų priėmimą
- Verslo taisyklės yra sudėtingos
- Verslo taisyklės yra dažnai keičiamos
- Kai taisyklės turi būti naudojamos keliose programose ar organizacijose

Verslo taisyklių valdymo sistemos nerekomenduojamos naudoti, kai:

- Verslo taisyklės yra statinės ir suskaičiuojamos
- Verslo taisyklės yra paprastos, vienodos, kartojamos

Egzistuoja ne mažai įvairių valdymo sistemų, kurios atlieka skirtingas funkcijas, daro verslo sprendimus įvairiausiuose lygmenyse, jų paskirtis yra skirtinga taip pat. Yra pateikiam šių sistemų klasifikacija pagal tai kokias taisykles sistemos gali valdyti:

- **Paprastos verslo taisyklės (Simple Business Rules)** – tai tokios taisyklės kurias galima išreikšti paprastu žodynu ir iškviečiant verslo proceso arba programos kintamumo taškuose. Tai tokios produktai kaip ILOG, BlazeAdvisor, BRBeans, QuikRules ir kt.
- **Dirbtinės intelektualios taisyklės (Artificial Intelligence rules)** – taisyklės kurios įtakoja algoritmus. Tokiai grupei galime priskirti produktus Intelligent Mineramp, DB2Amp ir kt.
- **Įvykių apibendrinimo taisyklės (Event correlation rules)** – taisyklės kurios naudojamos įvykių apibendrinimui. Produktai Tivoliamp, Eventconsole.
- **Taisyklės orientuotos į duomenis (Data Centric rules)** – taisyklės, valdančios duomenų užklausimą ir atnaujinimą. Produktas Versta.

2.4 Apibendrinimas

Tiek verslo taisyklių valdymo sistemose tiek modeliavimo kalbos kaip UML trūksta bendro prieito standarto kuriant verslo sistemas orientuotas į veiklos taisykles. Produktai verslo taisyklėms realizuoti skiriasi kokio lygio taisyklėms skirtos, kokios naudojamos platformos ir verslo taisyklių užrašymui skirtomis kalbomis. Kuriant tokias valdymo sistemas reikia daugiau dėmesio skirti modeliavimo standartizavimui ir taisyklių apskaitimo tarp kitų sistemų standartų.

3. Veiklos taisyklių klasifikavimo metodų analizė

3.1 Veiklos taisyklių struktūrizavimo modeliai

Nuo pat veiklos taisyklių koncepcijos atsiradimo pradžios buvo siūloma įvairūs VT klasifikavimo ir struktūrizavimo modeliai. Klasifikavimo modelių sąrašas yra pateiktas 1 lentelėje (Gottesdiener 1999).

1 lentelė VT klasifikacijos sąrašas

Klasifikavimo metodas	Trumpas taisyklių klasifikacijos aprašymas
Business Rules Group (GUIDE Business Rules Project)	Struktūrinės taisyklės (terminai, faktai), veiksmo taisyklės (integralumo apribojimai, sąlygos, autorizacija), išvestinės taisyklės (skaičiavimai, loginės išvados)

Ronald Ross	Nedalomos(ryšių tipai, subtipai) ir išvestinės
Tom Romeo	Struktūrinės taisyklės (ryšiai, kardinalumas), elgesio taisyklės (prieš sąlygos, po sąlygos, išvestys)
Margaret Thorpe	Apibrėžimai, integralumo taisyklės, bendrieji deklaratyvūs apribojimai, procedūriniai apribojimai,išvestys
Barbara von Halle	Apibrėžimai, faktai, apribojimai, išvestys, loginės išvados
Usoft Corporation	Apribojimai išvestys, elgesio taisyklės, atvaizdavimo taisyklės
Vision Software	Validavimo taisyklės , išvestys, ryšiai, sąlygos veiksmai

Pastebime, jog dauguma tiek ir didelės kompanijos, kuriančios IS kūrimo įrankius (Usoft Corporation, Vision Software) , tiek ir nepriklausomi tyrinėtojai, tokie kaip Barbara von Halle , Margaret Thorpe pasiūlyti metodai pasižymi nepastovumu bei susiaurinta panaudojimo sritimi. Bet nepaisant to, yra sukurti metodai, galintys atitikti keliamus veiklos taisyklių reikalavimus. Tai VT klasifikavimo modeliai tokie, kaip mokslininkų pasiūlytas GUIDE projektas, Microsoft kompanijos bendras darbas su Meta Data Coalition sukurtas modelis, taip pat pažymėtinas yra Ronaldo Rosso metodas.

3.1.1 GUIDE veiklos taisyklių projektas.

Projekto tikslas – pateikti veiklos taisyklių teorinę analizę, kuri galėtų tapti pagrindu, kuriant realias taisyklių koncepcijos pagrindu veikiančias sistemas. GUIDE projekte(Kolber 1997) nagrinėjamos veiklos taisyklių klasifikavimas, jų struktūra bei išskyrimo principai.

Nagrinėjamame GUIDE veiklos taisyklių projekte, atsižvelgiama būtent į informacijos sistemos perspektyvą. Į veiklos taisykles galima žiūrėti iš dviejų perspektyvų. Iš veiklos perspektyvos, jos siejamos su bet kokiais apribojimais, kurie naudojami organizacijos žmonių elgesiui, pavyzdžiui pertraukos pietums apribojimai organizacijos patalpose. Iš informacijos sistemų perspektyvų jos siejamos su faktais, kurie yra užrašomi kaip duomenys ir apribojimai faktų reikšmių pasikeitimams. Kitaip tariant nagrinėjama kokie duomenys ir apribojimai gali būti užrašomi į sistemą o kokie ne. Atitinkamai VT išreiškia ypatingus apribojimus informacinės sistemos

saugomų duomenų keitimui, išsaugojimui ir išmetimui. Pavyzdžiui, įrašas apie pirkinio užsakymą negali būti išsaugotas, jeigu kliento kreditavimo reitingas yra neadekvatus. Tuo tarpu toks apribojimas gali būti glaudžiai susijęs su veiklos taisykle, kuri sako, kad „įmonė negali parduoti klientui nieko su blogu kreditavimo limitu“. Toks apribojimu naudojimo požiūris teikia tris pranašumus projektui: Pirma susikoncentravus į informacijos sistemos perspektyvą daro problemas labiau apdorojamas. Tapo įmanoma suprasti ir modeliuoti veiklos taisykles kaip apribojimus duomenims. Bus daug sudėtingiau padaryti tokiu pat aiškius teiginius apie veiklos taisykles naudojamus veikoje.

Veiklos taisyklės sakinyse pagrindinės kategorijos gali būti:

- Veiklos terminų apibrėžimas (Business Rule)

Vienas iš pagrindinių veiklos taisyklių elementų yra kalba naudojama išreikšti ją. Pats termino apibrėžimas yra veiklos taisyklė, kuri apibūdina kaip žmogus galvoja ir kalba apie daiktus. Tokiu būdu apibrėžiant terminą yra sukuriama veiklos taisyklės kategorija.

- Struktūros teiginiai (Faktai susiejantys terminus vienas su kitais)

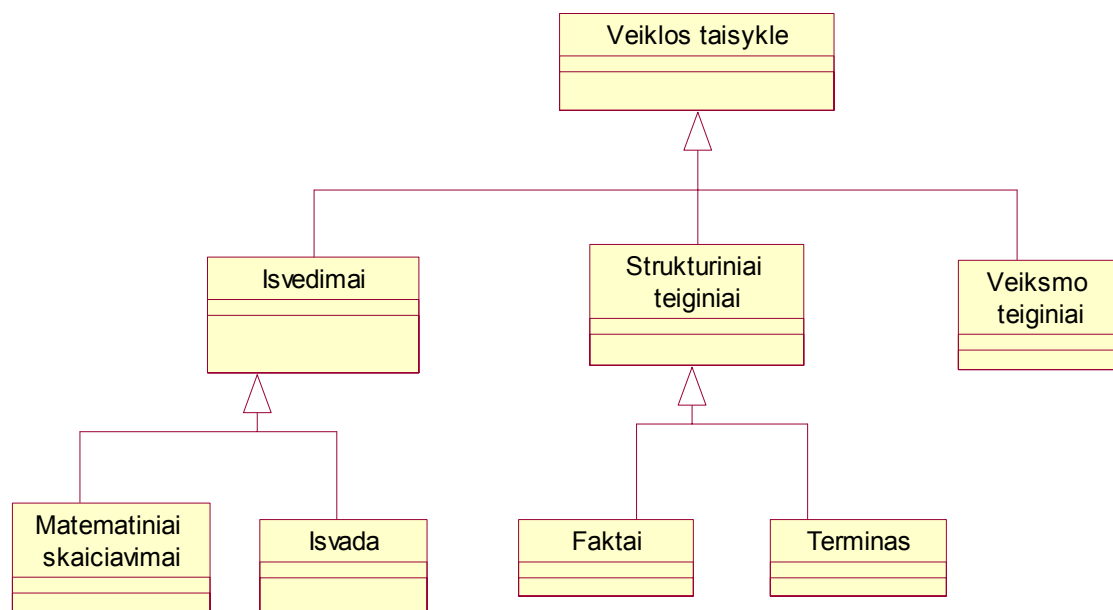
Organizacijos pobūdžio struktūra gali būti aprašoma terminais faktų, kurie sieja terminus su vienas su kitu. Pasakymas kad klientas gali pateikti užsakymą yra veiklos taisyklė. Faktai gali būti dokumentuojami kaip natūralios kalbos sakiniai arba kaip sąryšiai, atributai ir apibendrinimo struktūros grafiniame modelyje.

- Veiksmo teiginiai (arba apribojimai)

Kiekviena organizacija apriboja elgseną tam tikru būdu, ir tai yra susiję su apribojimo duomenimis t.y. apibrėžiama kokie duomenys gali arba negali būti keičiami. Apsaugoti nuo įrašo sukūrimo, daugeliu atveju yra apsaugojama kad veiksmas neįvyktų.

- Išvedimai (derivations)

Veiklos taisyklės apibrėžia kaip žinias iš vienos formos gali būti transformuojamos į kitas žinias, formas.



2 pav. GUIDE projekto veiklos taisyklių klasifikacija

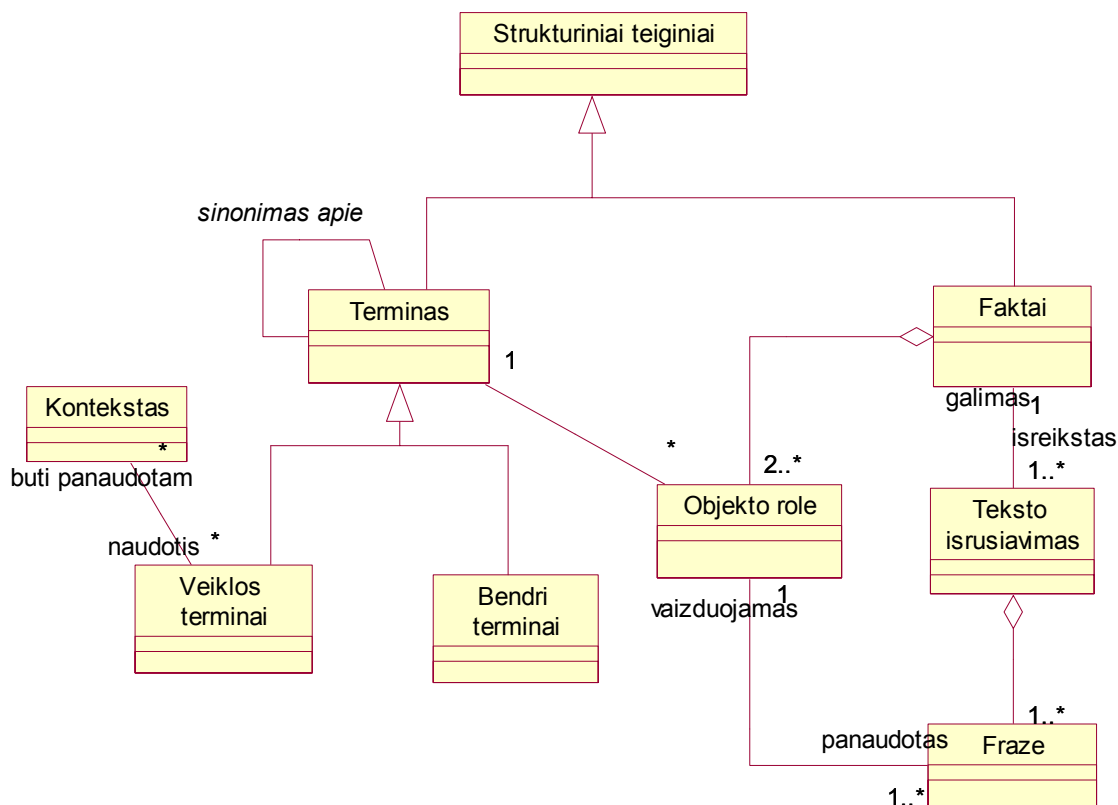
Parodytame paveiksle aukščiau aprašytos veiklos taisyklių klasifikacija. Kiekvieną iš jų panagrinėsime detaliau.

Struktūros teiginiai

Teiginys apie struktūrą yra teiginys, kuris nurodo, kad egzistuoja kažkas svarbaus apie veiklą kaip dominantis konceptas arba egzistuoja kaip sąryšis su kitu dominančiu daiktu(objektu). Detaliau galima apibrėžti taip, kad teiginiai apie struktūrą yra statiniai veiklos aspektai, išreiškiantys daiktus(objektus) arba kaip daiktai siejasi tarpusavyje.

Terminai ir faktai

Teiginiai apie struktūrą yra dviejų rūšių: terminai ir faktai. Terminas yra žodis arba frazė, kuri turi aiškia prasmę veiklai. Terminas gali būti dviejų tipų: veiklos ir bendri terminai. Veiklos terminas yra žodis arba frazė, kuri turi aiškia prasmę veiklai tam tikrame suprojektuotame kontekste. Kiekvienas veiklos terminas privalo būti panaudotas bent viename kontekste, ir kiekvienas kontekstas gali būti panaudotas vieno ar daugiau veiklos termino. Terminai ir faktai parodyti 3 paveiksle.



3 pav. Struktūrinio teiginio klasifikacija

Pavyzdžiui veiklos terminai mašinų nuomos veiklos kontekste gali būti „rezervacija“, „nuoma“, „klientas“ ir kiti. Bendri terminai yra žodžiai naudojami kasdieninėje kalboje. Bendrais terminais gali būti žodžiai iš naudojamo mūsų žodyno, „mašina“, „miestas“, naudojant aiškias visiems suprantamas reikšmes be papildomų pridedamų apibrėžimų.

Tiek veiklos terminai tiek bendri yra naudojami kaip terminai konstruojant sakinius, kurie yra veiklos faktų teiginiai. Skirtumas tarp veiklos ir bendrų terminų yra tai kad veiklos terminas privalo būti aiškiai apibrėžtas vieno ar daugiau faktų terminais. Kiekvienas faktas gali būti panaudotas apibrėžime vieno ar daugiau veiklos terminų.

Faktas nurodo sąryšį tarp dviejų ar daugiau terminų. Faktas gali apimti kelis terminus ir terminas gali būti panaudotas viename ar daugiau faktų. Pavyzdžiui „klientas gali pateikti užklaimą automobilio modeliui nuomos įmonei tam tikrą dieną“ yra faktas, kuris apima keturis bendrus terminus: klientas, mašina, nuomos įmonė, data. Veiklos terminas šiuo atveju „mašinos užklausa nuomai“

Objekto rolės yra reikalingos aprašymui kiekvienai semantinei rolei, kurią terminas atlieka fakte. Pavyzdžiui įmoneje yra faktas, kad gali egzistuoti sąryšis tarp terminų „klientas“ ir „sutartis“ Šiame fakte galimi dvi objekto rolės – viena teigianti, kad terminas „sutartis“ gali būti atlikėjas

semantinės rolės kaip objekto rolės, kuri yra fakto dalis ir kita rolė – teigianti, kad terminas „*klientas*“ gali būti atlikėjas semantinės rolės kaip objekto rolės, kuri yra fakto dalis.



4 pav. Kliento ir sutarties pavyzdys

Kiekvienas faktas turi susidėti iš vienos ar daugiau objektų rolių. Pavyzdyje pateikto kliento/sutarties fragmente, galimi abu atvejai, kad „*kiekviena sutartis gali būti su klientu*“ ir kad „*kiekvienas klientas gali būti nuomotojas daugelyje sutarčių*“

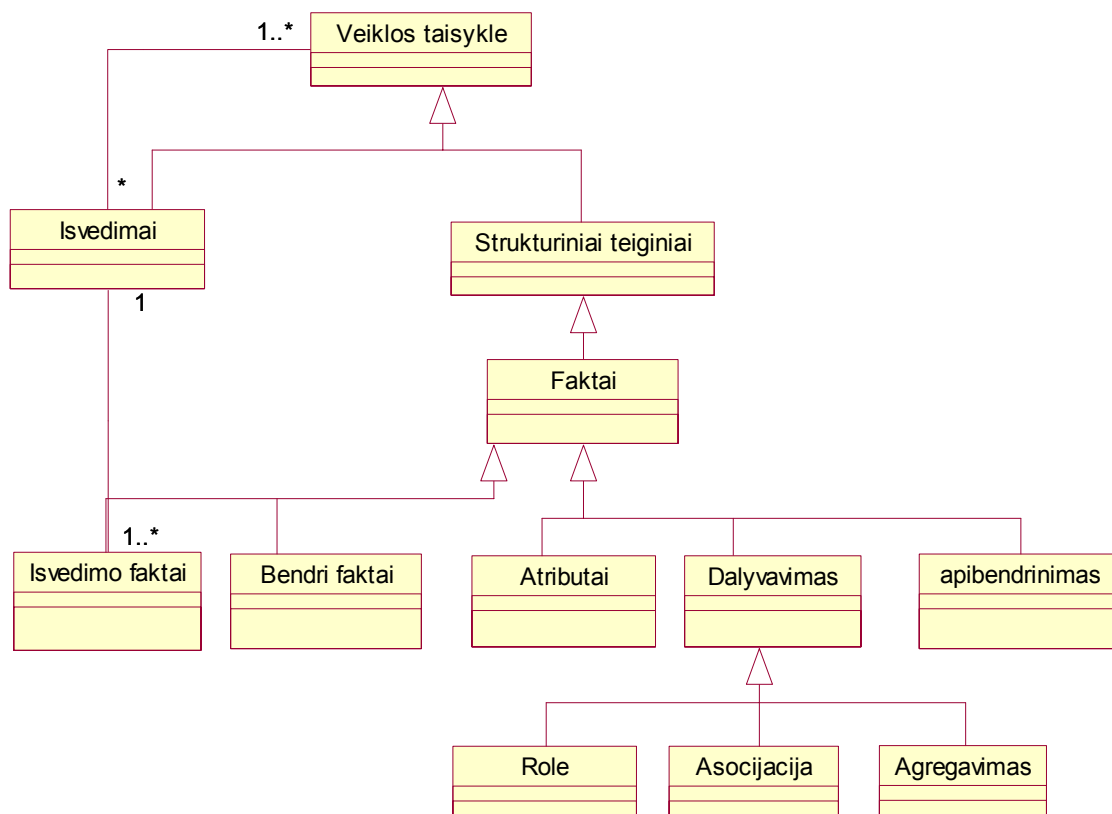
Taip pat struktūrinio teiginio klasifikacijos 3 paveiksle pateikta esybė teksto išrūšiovimas. Tai yra, kad kiekvienas faktas gali būti išreikštas kaip vienas ar daugiau teksto išrūšiovimų, kur teksto išrūšiovimas vaizduoja vieną iš galimų fakto formulavimų. Be to nustatant objekto rolės seką teksto išrūšiovime, frazė taip pat teikia tekstą ir nurodo sintaksinę objekto rolės rolę.

Anksčiau minėtame pavyzdyje „*Kiekvienas klientas gali būti nuomotoju daugelyje sutarčių*“ yra vienas fakto teksto išrūšiovimas. Frazė „*kiekvienas klientas*“ aprašo objekto rolę kaip dalyką šiame fakto teksto išrūšiovime. Taip pat frazė nurodo kad *klientas* pasirodo pirmoje vietoje šiame išrūšiovime ir jis teikia teksto frazė „*kiekvienas*“.

Taigi tos struktūros tarp faktų yra išreiškiamos kaip viena ar daugelis frazių(per teksto išrūšiovimą) ir faktas susideda iš vienos ar daugiau objekto rolių.

Fakto tipai

Faktai taip pat turi savo klasifikaciją, kaip parodyta 5 paveiksle. Pagrindinės kategorijos būtų bendri faktai, kurie yra paprasčiausiai pateikti ir išvesti faktai, kurių reikšmės yra sudaryta iš kitų bendrų faktų. Kita faktų klasifikacija skiria juos į tris tipus: atributus, dalyvavimą, apibendrinimą.



5 pav. Fakto teiginių klasifikacija

Bendri(pagrindiniai) faktai ir išvedimo faktai

Faktas gali būti išvestinis ir pagrindinis. Pagrindinis faktas yra fiksuojamas, kad taip gali būti. Išvestas faktas yra teiginys, kuris yra sudaromas iš kitų teiginių. Tai gali būti skaičiuojama reikšmė arba tai gali būti vaizdas. Kiekvienas išvestinis faktas turi būti išvestas naudojant išvedimą(derivation) – išvadas arba matematinius skaičiavimus. Kiekvienas išvedimas gali būti pagrįstas viena ar daugiau veiklos taisyklių ir veikos taisyklė gali būti naudojama daugelyje išvedimų. Veiklos taisyklės pavyzdys:

„Paskaitos pravedamos laisvoje auditorijai“

Išvestas faktas gali būti:

„Paskaitos laboratoriniams pravesti turi būti kompiuteriu auditorijoje“

„Auditorijos pasirinkimas priklauso nuo studentų skaičiaus ir priemonių paskaitai pravesti kiekio“

Atributai, dalyvavimas ir apibendrinimas

Faktas taip pat turi sekančia klasifikaciją: atributai, kurie pažymi terminą kaip kito termino atributą, apibendrinimas, kuris pažymi viena terminą kaip apibendrinama(super tipas) vieno ar kelių terminų(potipių), dalyvavimas, kuris pažymi terminą kaip sąryšį tarp kitų terminų.

Atributas išreiškia faktą, kuriame terminas apibūdina aspektus kito termino.

Atributo pavyzdžiai studentų tvarkaraščio pavyzdyje:

„Vardas yra dėstytojo atributas“

„StudentoNr yra studento atributas“

Apibendrinimas išreiškia fakto tipą, kuriame terminas apibūdina aspektus kito termino.

Apibendrinimo pavyzdys gali būti:

„Studentas arba dėstytojas yra asmuo.“

„Dėstytojas yra universiteto darbuotojas“

Dalyvavimas išreiškia faktą, kuriame aibė terminų yra sujungti tam tikra prasme ir prasmingi veikloje. Dalyvavimas gali būti vienas iš šių tipų: Agregavimas – yra dalis/susideda iš sąryšis, Rolė – apibūdina kaip terminas dalyvauja kaip aktorius per jo sąveika su jo aplinka, Asociacija – bet kokios kitos rūšies sąryšis.

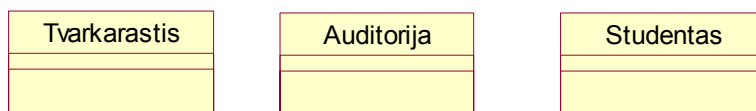
Dalyvavimo(asociacijos) pavyzdžiu gali būti

„Paskaitos gali vesti dėstytojas“

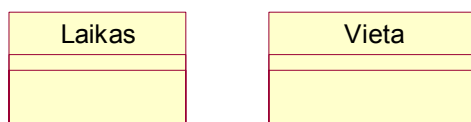
„Tvarkaraštis paskaitų turi būti žinomas studentams“

Terminų ir faktų vaizdavimas UML modelyje

Išanalizavus veiklos taisyklių kategorija struktūriniai teiginiai ir detaliau terminų apibrėžimo ir jų klasifikavimą galime teigti kad tiek bendri tiek veiklos terminai UML modelyje gali būti vaizduojami klasėmis, kurias galime praktiškai realizuoti Oracle SQL sakiniiais.



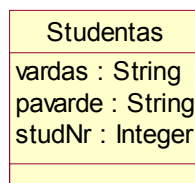
6 pav. Veiklos terminai



7 pav. Bendri terminai

Faktai nurodantys sąryšį tarp dviejų ar daugiau terminų UML vaizduojami ryšiais tarp klasių. Ryšio tipas priklauso nuo to, kuria faktų rūšiai priklauso vaizduojamas faktas.

Jeigu faktas priklauso atributų klasei, tai klasėje atvaizduojamas jam priklausantis atributas:



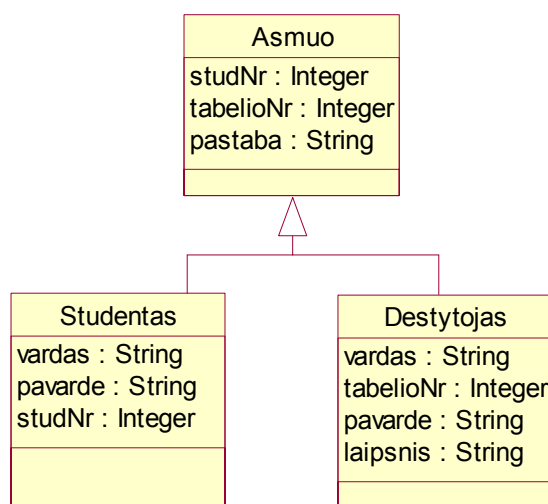
8 pav. Atributo pavyzdys: veiklos taisyklė „*StudentoNr yra studento atributas*“

Oracle SQL sakinyje CREATE TABLE yra išvardijami klasės atributai atskiriami kableliais ir nurodomi jų duomenų tipą.

CREATE TABLE STUDENTAS

```
(student_nr NUMBER (6),
  vardas varchar (20),
  pavarde varchar (30));
```

Apibendrinimo klasei priklausančios veiklos taisyklės vaizduojamas apibendrinimo sąryšiu tarp klasių.



9 pav. Apibendrinimo klasės pavyzdys „*Studentas arba dėstytojas yra asmuo*“

Toki apibendrinimą realizuoti galimi toks variantas: sukurti tris atskiras lenteles ir sukurti sąryšius tarp lentelių *Studentas – Asmuo* ir *Destytojas – Asmuo* ir uždėti apribojimą *Asmuo* lentelei, kad išorinių raktų laukai nebūtų abu užpildyti, naudojant funkcija – CHECK.


```
ALTER TABLE Asmuo
```

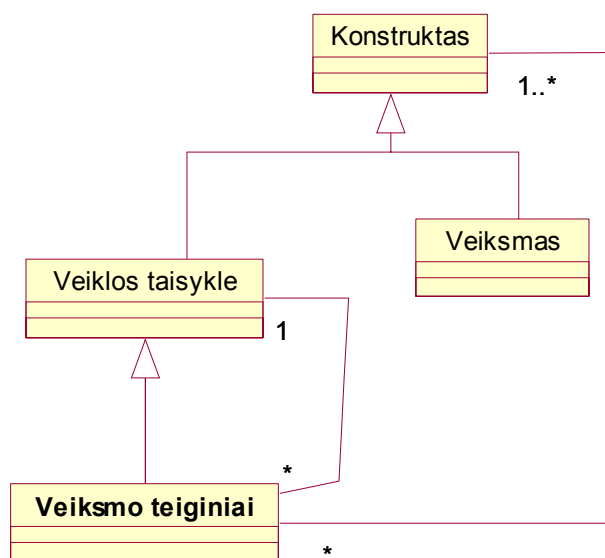
```
Add constrain tikrinti
```

```
CHECK (
```

```
(studNr is null and tabelioNr is not null) or (studNr is not  
null and tabelioNr is null));
```

Teiginiai apie veiksmus

Teiginys apie veiksmą yra teiginys, kuris liečia veiklos dinaminis aspektus. Jie specifikuoja apribojimus rezultatams, kuriuos gali duoti veiksmas. Apribojimai yra aprašomi neprocedūriškai, kitų atominių veiklos taisyklių terminais. Kur teiginiai apie struktūrą aprašo galimybes, tai teiginiai apie veiksmus uždeda apribojimus – privalo arba neprivalo. Teiginiai apie veiksmus parodytas 10 paveiksle.



10 pav. Teiginiai apie veiksmus

Teiginio komponentai apie veiksmus

Teiginys apie veiksmą susideda iš pagrindinio objekto ir vieno arba daugiau atitinkamų objektų. Pagrindiniu objektu gali būti bet kuri veiklos taisyklė. Kiekviena VT turi būti veiklos taisyklės savybė. Bei veiklos taisyklė gali būti pagrindiniu objektu vieno ar kelių teiginių apie veiksmus. Frazėje „Jeigu ...tuomet...“ frazė po „Jeigu“ Pagrindinis objektas yra dažnai teiginys apie struktūrą, bet tai gali būti ir teiginys apie veiklą. Atitinkamas objektas gali būti arba kita veiklos taisyklė arba veiksmas, pateiktas apibendrinimo termino konstruktas, kur jis yra arba kita

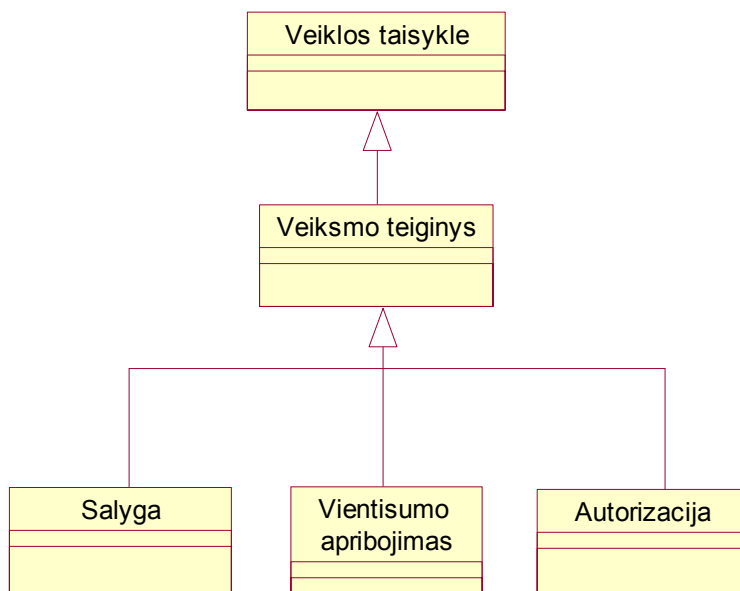
veiklos taisyklė arba veiksmas. Tai konstruktas yra atitinkamas objektas teiginyje apie veiksmą. Veiksmas gali būti: „*Tvarkaraštis pataisytas*“, „*laikas pakeistas*“. Kiekvienas iš šių veiksmų gali būti atitinkamais objektais teiginyje apie veiksmą. Teiginys apie veiksmą yra įvertintas naudojant vieną ar daugiau konstruktų. Jie aprašo apribojančias sąlygas(jos eitų po žodelio „tuomet“)

Teiginys apie veiksmą gali būti pavyzdyje:

„*Studentas privalo turėti studento numerį.*“

Veiklos taisyklė „*studentas*“ (kuris yra terminas) yra pagrindinis objektas teiginyje apie veiksmą. Veiklos taisyklė, kuri nurodo galimybę, kad „*studentas gali turėti studento numerį*“ (faktas) yra atitinkantis objektas teiginyje apie veiksmą

Taip pat teiginiai apie veiksmą turi savo klasifikaciją kaip parodyta 11 paveiksle. Veiksmo teiginys gali būti arba sąlyga arba vientisumo apribojimas arba autorizacija.



11 pav. Veiksmo teiginių klasifikacija

Sąlyga yra teiginys, kuris jeigu kažkas yra teisinga, tuomet kita veiklos taisyklė bus vykdoma. Vientisumo apribojimas yra teiginys, kuris privalo visada būti teisingas. Jis turi iškart uždrausti bet kokią veiksmą, kuris sąlygotų netiesą. Sąlyga gali patikrinti reikšmę ir tuomet nurodyti tam tikrą veiksmą pagal tikrinamą reikšmę. Tačiau vientisumo apribojimas gali teigti, kad „*studentas privalo turėti studento numerį*“ ir uždrausti visus veiksmus, kurie sąlygotų netiesą šiam teiginiui. Toks vientisumo apribojimas uždraustų sukūrimą naujo studento egzemplioriaus , jeigu jis neturi studento numerio ir taip pat užraustų visus veiksmus, kurie galėtų nustatyti studento numeriui tuščią reikšmę.

Autorizacija apibrėžia specifines, išimtines teises bei privilegijas. Tai yra teiginys tokio pavidalo:

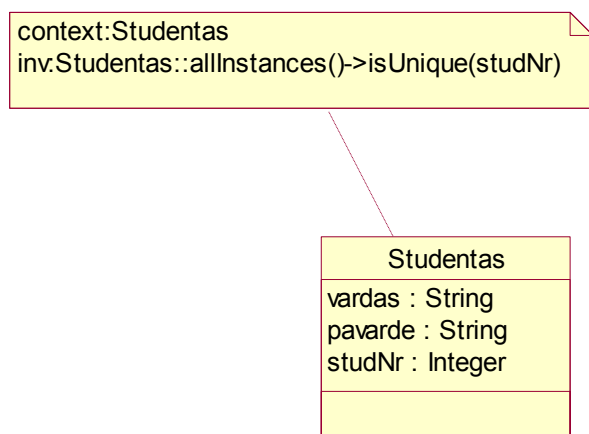
„(Tik) administratorius gali atlikti ištrinimo veiksmą“.

Autorizacija dažniausiai yra suteikiama tik tipams sugebantiems savarankiškai veikti (žmonės, įstaiga, kompiuteris).

Teiginių apie veiksmus vaizdavimas UML

Panagrinėsime veiklos taisykles, kurios priklauso vientisumo apribojimo teiginiams apie veiksmus. Vaizduosime UML modelyje, taip pat pateiksime Oracle SQL realizacija, duotiems apribojimams.

Privalomo atributo apribojimas – „*Studentas privalo turėti unikalų studento numerį*“ pavaizduotas 12 paveiksle.



12 pav. Privalomo atributo apribojimo pavyzdys UML notacija

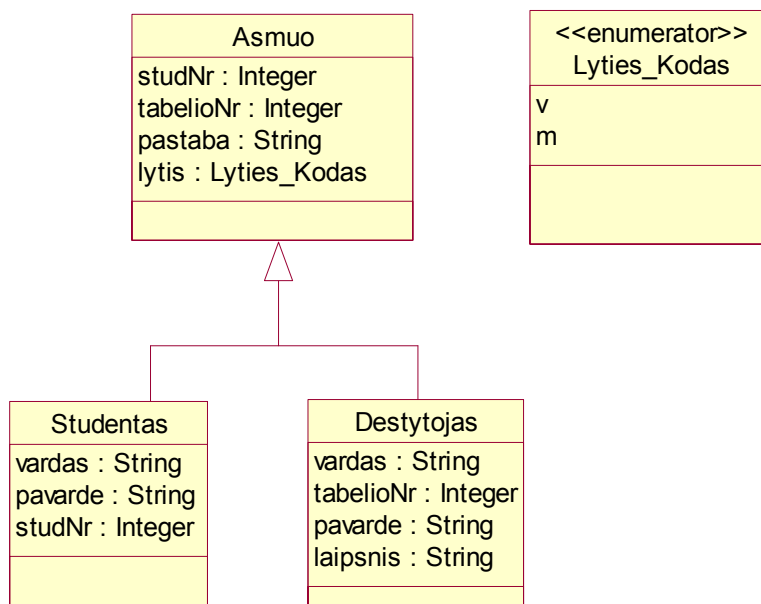
Veiklos taisyklės užrašas Oracle SQL notacija:

```
alter table Studentas add constraint unikalumas Unique(studNr);
```

arba kuriant lentelę *Studentas* nurodant, kad laukas *studento numeris* privalo būti užpildytas ir yra unikalus.

```
CREATE TABLE STUDENTAS (
    vardas      string (20) NULL,
    pavarde    string(30) NULL,
    studNr     number (6) NOT NULL,
    CONSTRAINT unikalumas1 UNIQUE (studNr);
```

Atributo reikšmės apribojimas – neleidžiantis atributui įgyti neleistinių reikšmių pavaizduotas 13 paveiksle.



13 pav. Atributo reikšmės apribojimo pavyzdys UML notacija

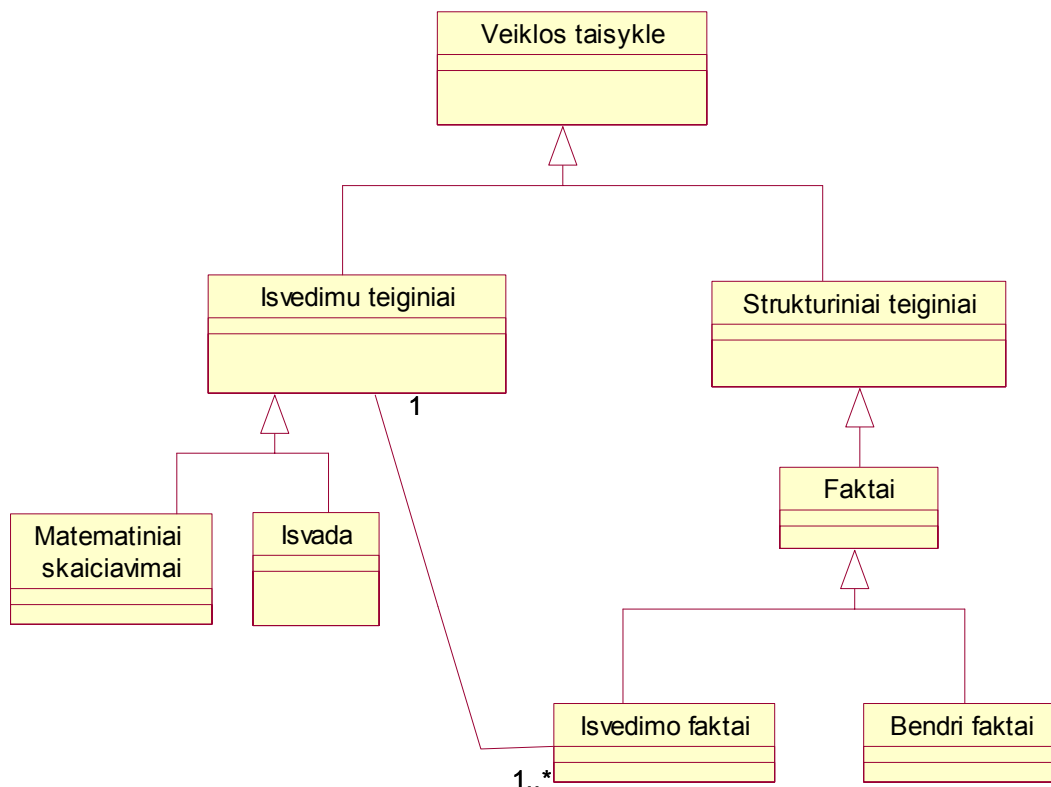
Šiuos apribojimus galima realizuoti CHECK funkcija lentelės atributams.

```

Alter table Asmuo add
Constraint reiksmes_apribojimas
Check (Lytis in ('M', 'V'));
  
```

Išvedimų teiginiai

Bendras faktas yra faktas, kuris egzistuoja pateiktame pasaulyje ir yra išsaugotas sistemoje. Išvestas faktas yra sukurtas išvadų arba matematiniais skaičiavimais iš terminų, faktų arba kitų išvedimų, arba kitų teiginių apie veiksmą. Žemiau esančiame 14 paveiksle pateiktas faktas yra pavaizduotas kaip fakto rūšis. Kiekvienas išvestas faktas privalo būti išvestas naudojant vieną išvedimą. Pats išvedimas gali būti iš vienos ar daugiau veiklos taisyklių



14 pav. Išvedimo teiginiai UML notacija

Išvedimo teiginiai gali būti arba matematiniai skaičiavimai(toks kaip „nuomos kaina yra lygi padauginus pagrindinę kainą iš nuomojamų dienų skaičiaus“) arba išvados (inference).

Matematinis skaičiavimas sukuria išvestą faktą pagal nurodytą algoritmą. Išvada sukuria išvestą faktą naudojant logines indukcijas arba dedukcijas.

3.1.2 R. Rosso metodas.

Labai detalų veiklos taisyklių klasifikavimą pateikia Ronaldas Rossas(Ross 1997). Pagrindinė šio metodo idėja – suskirstyti taisyklės į standartinius, aukšto lygio taisyklių tipus, kurie nurodo, kokį patikrinimą turi atlikti taisyklės.

Taisyklės gali būti nedalomos (atominės) ir išvestinės. Klasifikacijos šeimos smulkiau suskirstytos į poaibius(2 lentelė).

2 lentelė.Rosso šeimų tipai

Atominių taisyklių šeimos	Išvestinių taisyklių šeimos
Egzempliorių patvirtintojai	Egzempliorių testavimo
Tipo patvirtintojai	Pozicijos patikrinimo
Pozicijos patvirtintojai	Atributų modifikavimo

Funkciniai patvirtintojai	Sekos valdymo
Lyginamieji įvertintojai	Sekos specifikuojimo
Matematiniai įvertintojai	Kompozicijos struktūrų testavimo
	Sąlyginio laiko įvertinimo
	Veiklos koordinavimo
	Egzempliorių įgalinimo
	Egzempliorių kopijavimo
	Egzempliorių iškvietimo

Išskiriamos 7 atominių (nedalomų) taisyklių šeimos, kurios savo ruožtu turi 32 išskirtus veiklos taisyklių tipus. Išvestinės taisyklė pagal R. Rosą – tai taisyklė, kuri išreiškiama kitų taisyklių aprašais. Išvestinės taisyklės nėra atominės ir gali būti sudarytos iš keleto atominių taisyklių arba kitų išvestinių taisyklių. Išvestinės taisyklės skirstomos į 12 šeimų, kurios turi dar 58 savo tipus. Labai daug dėmesio R. Rosas skiria taisyklių tipų aprašymui, todėl šis modelis klasifikavimo atžvilgiu yra tikslus.

Veiklos taisyklių modeliavimo metodika pagal R. Rosą

Be detalios taisyklių klasifikavimo sistemos R. Rosas pasiūlė ir unikalią VT modeliavimo metodiką, užrašant jas grafine forma.

Ronaldas Rossas pasiūlė penkis klausimus, į kuriuos reikia atsakyti prieš pradėdant modeliuoti taisykles:

- Ar taisyklė yra apribojimas ar sąlyga, ar taisyklė reikalauja vykdymo, ar yra išreikšta kaip patikrinimas?
- Kas yra taisyklės bazė, kuriam tipui duomenų modelyje priskirti taisyklė?
- Kas yra taisyklės korespondentai t.y. duomenų tipas, toks kaip duomenų objektas, atributo tipas, būtini norint patikrinti taisyklę?
- Koks taisyklės tipas?
- Koks ryšys yra tarp taisyklės bazės ir korespondento, kaip jie susiję tarpusavyje?

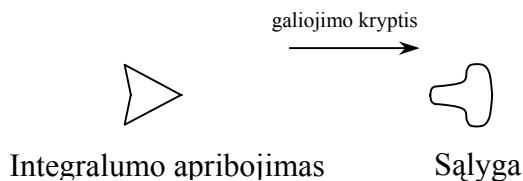
Norint formaliai užrašyti taisykles reikia naudotis dalykinės srities duomenų modeliu. Prieš tai reikėtų apibrėžti, kad duomenų tipai turi būti žinomi: tai esybės, atributai, asociacijos, sąryšiai, potipiai.

Veiklos taisyklės kategorijos. Taisyklės pagal R. Rossą gali būti tik dviejų rūšių – sąlyga arba integralumo apribojimas.

Sąlygos rūšies taisyklės rezultatas gali būti arba loginis nulis, arba loginis vienetas. Rezultatas gali būti ir nežinomas. Jos naudingos, kai reikia nustatyti kokias taisykles vykdyti arba testuoti (t.y. kokias taisykles iškviešti) tam tikrais atvejais.

Integralumo apribojimo rūšies taisyklės rezultatas visada yra loginis vienetas. Kadangi jos negali duoti rezultato loginis nulis, jos privalo būti vykdomos.

Sąlygoms ir integralumo apribojimams Rosas siūlo grafinį žymėjimą parodytame 15 paveiksle.



15 pav. Integralumo ir sąlygos žymėjimas

Simboliai gali būti sukami į bet kokią pusę, tik nereikia pamiršti galiojimo krypties.

Taisyklės bazė. Kiekviena taisyklė apibrėžia kokį nors duomenų modelio egzempliorių. Šis egzempliorius vadinamas *taisyklės baze*. Taisyklės bazė gali būti duomenų objektas, atributas, tačiau gali būti ir kita taisyklė. Iš esmės, taisyklė priklauso bazei, todėl ją galima laikyti bazės savybe. Kiekviena taisyklė privalo turėti bazę.

Korespondentai. Kiekviena taisyklė, be bazės remiasi dar bent vienu duomenų modelio egzemplioriumi. Šie egzemplioriai vadinami taisyklės *korespondentais*. Kiekviena taisyklė privalo turėti bent vieną korespondentą. Korespondentai yra tiesiogiai susiję su tikrinimu, kurį atlieka taisyklė. Korespondentas dažniausiai yra duomenų tipas (duomenų objektas, atributas), tačiau taip pat gali būti kita taisyklė ar veiksmas.

Korespondentai turi šias savybes:

- Atominės taisyklės, priklausančios patvirtinimo tipo šeimai, privalo turėti du arba daugiau korespondentus;
- Atominės taisyklės, priklausančios matematinių įvertintojų šeimai gali turėti daugiau nei vieną korespondentą;
- Kitoms šeimoms priklausančios atominės taisyklės gali turėti *tik vieną* korespondentą.

Taisyklių tipo parinkimas. Taisyklės tipo parinkimas yra svarbiausias žingsnis interpretuojant taisyklę. Nuo taisyklės tipo priklauso, kokį patikrinimą atliks taisyklė. Taisyklių tipai parenkami iš standartinės atominių ir išvestinių taisyklių aibės (2 lentelė). Svarbu pažymėti, kad visi taisyklių tipai gali būti naudojami tiek integralumo apribojimams, tiek sąlygoms.

Taisyklės bazės ir korespondentų susiejimas. Labai svarbu kaip yra susieta bazė su korespondentu. Šis ryšys turi atsispindėti duomenų modelyje, taisyklė naujo ryšio įvesti negali.

Duomenų modelį galima priimti kaip aibe faktų, kurie aprašo veiklos sritį. Faktai gali būti šių tipų:

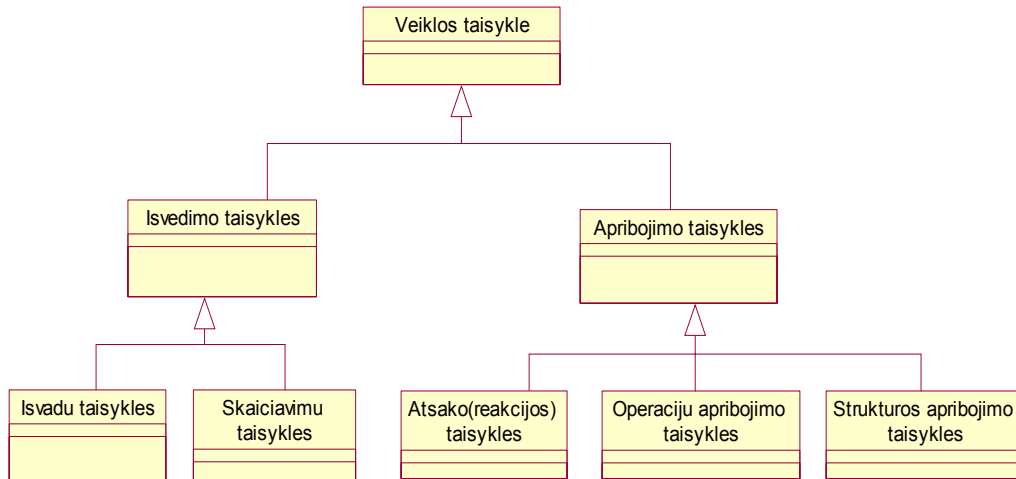
- Sąryšio tipo: ‘Dėstytojas teikia duomenis’;
- Potipių sąryšio: ‘Dėstytojas yra darbuotojas’;
- Atributų sąryšio: ‘Studentas turi vardą’.

Kiekvienas sąryšis tarp bazės ir korespondento yra svarbus taisyklės interpretavimui, todėl taisyklės diagramoje kitų sąryšių vaizduoti nereikėtų.

Kadangi R.Rossas nesiūlo sukurtai veiklos taisyklių klasifikacijai duomenų modelio, kuriame šios taisyklės būtų saugomos, todėl darbe(Kapočius 2001) remiantis Rosso metodu, pasiūlė VT saugyklos architektūrą, išnagrinėjo saugyklos principus(plačiau žr. sk. VT repozitorius). Taip pat VT saugyklos architektūros modeliavimo analizė pateikta (Motiejūnas 2002) darbe.

3.1.3 Veiklos taisyklės klasifikacija projektuotojo požiūriu

Kadangi labai svarbus uždavinys yra veiklos taisyklių klasifikavimas, darbe naudosime veiklos taisyklių klasifikavimo būdą kuris labiau tinka projektuotojui, kai visos veiklos taisyklės suskirstomos į dvi stambias grupes(Nemuraitė 2002).Šis klasifikavimo būdas panašus savo sudėtimi į GUIDE veiklos taisyklių projektą. Kadangi visi klasifikavimo būdai tarpusavyje yra panašūs, pasirinkome tokį klasifikavimo būdą, kuris labiausiai tiktų projektuotojui, kuriant IS. Dažniausiai pasitaiko tokios taisyklių grupės :struktūrinės taisyklės – šios taisyklės apibrėžia duomenų struktūras. Veiksma taisyklės – apibrėžia operacijas ir sekas, kurios turi būti įvykdytos. VT klasifikavimo būdas išdėstytas 16 paveiksle.



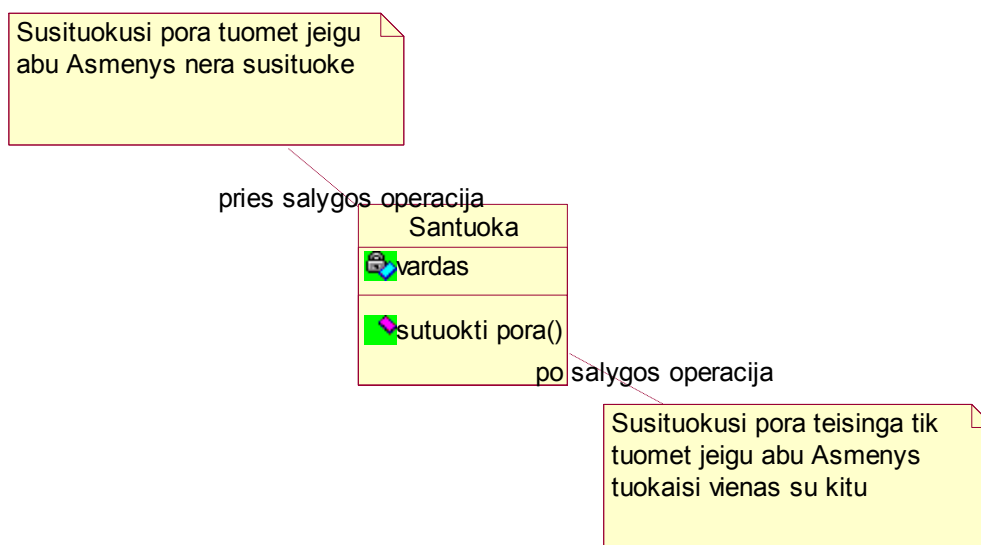
16 pav. Veiklos taisyklių kategorijos projektavimo požiūriu

Veiklos taisyklės čia skaidomos į dvi pagrindines taisyklių kategorijas: taisyklės kurios yra išvedamos iš kitų veiklos taisyklių ar yra priklausomos ir vadinamos jos išvedimo taisyklėmis. Taip pat kita kategorija atribojimo taisyklės - teiginiai apie atribojimus ir sąlygas, apibrėžiantys reikalavimus organizacijos valdymui. Jie nusako dinaminčius veiklos aspektus. Specifikuoja rezultatus, kurie yra veiksmų pasekmė. Atribojimo taisyklės skaidomos į atsako, operacijų ir struktūros atribojimo taisykles. Atribojimo taisyklės gali būti suskirstytos į dvi stambias kategorijas pagal jų kilmę: struktūriniai atribojimai, kuriais mes nusakome kaip objektai siejasi vienas su kitu, tai pastovūs faktai ir elgesio atribojimai, kurie atriboja taisykles vykdymui ar nusako priklausomybę viena nuo kitos.

Atsako (reakcijos) taisyklės atriboja elgesį, nusako sąlygas, prie kurių galima vykdyti operaciją. Pavyzdžiui: *Kai studentas vykdo prisijungimą prie sistemos jeigu jis turi studento numerį tuomet vykdomas prisijungimas prie sistemos.* Reakcijos taisyklės priklausomos nuo konteksto. Taisyklės Jeigu sąlygą tikrinama tik tada Kai įvyksta tam tikro tipo įvykis. Tokios taisyklės sutinkamos pvz. Projektuojant darbų sekas arba būsenų perėjimo diagramose. Tokio tipo taisyklės dar vadinamos ECA (Event-Condition-Action) taisyklėmis.

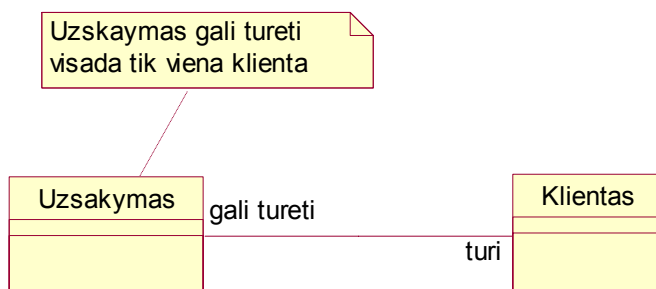
Operacijų atribojimo taisyklės nurodo, kokios sąlygos turi būti tenkinamos prieš operaciją ir po jos, kad būtų užtikrintas vykdymo teisingumas. Tokie atribojimai yra gyvybiškai svarbūs operacijos vykdymui ir nepriklauso nuo operacijos sužadinimo konteksto. Į šias taisykles galime žiūrėti kaip į surišėją, kuris sieja operaciją su jos iškvietėju. Operacijų taisyklės užrašas atrodo taip: Jei iškviesi operaciją, kai įvykdyta prieš sąlyga, gausi galutinę būseną, kurioje bus patenkinta po-sąlyga. Pvz.: *Pasiūlyti tarnautojui vadovo pareigas tik tuomet jeigu tarnautojas nėra vadovas.* Operacijos po-sąlyga garantuoja rezultatus. Kitas pavyzdys: *Susituokusi pora yra*

tik tuomet jeigu abu šie žmonės nėra susituokę anksčiau. Tai būtų prieš sąlyga būtina įvykti santuokai. Ir po sąlyga išpildoma, teisinga bus tada jeigu: Susituokusi pora yra teisinga išpildyta tik tuomet kai abu žmonės yra vedę vienas kitą.



17 pav. Operacijų apdorojimo veiklos taisyklės pavyzdys

Struktūros apribojimo taisyklės aprašo objektų ir jų sąryšių sąlygas, kurių negalima pažeisti. Struktūros apribojimo taisyklės savo ruožtu turi savo tipus.: taip pat gali apriboti atributo reikšmes, esybių egzempliorius, ryšio kardinalumus. Struktūros taisyklės neturi nuorodų į operacijas, jos turi tenkinti bet kuriomis veikimo aplinkybėmis, nesvarbu ar pasikeitė būsenos, ar ne.



18 pav. Struktūros apdorojimo veiklos taisyklės pavyzdys

arba tai gali būti pavaizduota naudojantis ryšio kardinalumu.



19 pav. Struktūros apdorojimo veiklos taisyklės pavyzdys panaudojant kardinalumą

Išvadų taisyklės yra taisyklės kurios konstatuoja kaip esybės(objektai) ir ryšiai yra išvedami, gaunamos iš veiklos taisyklių naudojant išvadas.

Išvados taisyklės gali išreikšti esybes(objektus).Pavyzdžiui: „*Jeigu objektas yra Darbininkas tai jis yra Asmuo*“ „*Asmuo yra Darbininkas tuomet ir tik tuomet asmuo dirba Imoneje*“

Skaičiavimo taisyklės apibrėžia algoritmus, kuriuos vykdant galima gauti norimą rezultatą.

3.1.4 Veiklos taisyklių struktūrizavimo modelių palyginimas

Išanalizavus klasifikavimo modelius, galime išskirti keletą kriterijų(Kapočius 2002), ir palyginti žinomus metodus(5 lentelė). Įvertinimo skalė nuo 0(neatitinka pateiktų kriterijų) iki 5(pilnai išpildo kriterijus).

3 lentelė. Klasifikacijos metodų palyginimas

Kriterijus	GUIDE projektas	Rosso metodas	Projektuotojo pož.
Klasifikacijos užbaigtumas	4	5	4
Modelio apibrėžtumas	3	5	3
Modeliavimo metodika	3	5	2
Saugojimo modelis	3	2	2

Atlikus metodų analizę prieita prie išvados, kad detaliausias VT klasifikavimo metodas modeliavimo aspektu yra R.Rosso pateikta studija. Tačiau projektavimui labiau tinka veiklos taisyklių klasifikacija projektuotojo požiūriu, kuris lenkia R. Rosso metodą. Kaip teorinė studija labai naudingas yra GUIDE grupės sukurtas modelis.

Kadangi labai svarbus uždavinys yra veiklos taisyklių klasifikavimas, darbe naudosime veiklos taisyklių klasifikavimo būdą, kuris labiau tinka projektuotojui, kai visos veiklos taisyklės suskirstomos į dvi stambias grupes. Šis klasifikavimo būdas panašus savo sudėtimi į GUIDE veiklos taisyklių projektą. Kadangi visi klasifikavimo būdai tarpusavyje yra panašūs, pasirinktas toks klasifikavimo būdas, kuris labiausiai tinka projektuotojui., kuriant informacinę sistemą. Realizacijai pasirinktas veiklos taisyklių klasifikavimas GUIEDE projekto pagrindu, kuris atitinka projektuotojo požiūrį į veiklos taisyklių projektavimo galimybių atitikimą.

4. Veiklos taisyklių modeliavimui naudojamos kalbos

4.1 UML ir OCL

Šiuo metu daugelis informacinių sistemų projektavimo įrankių naudoja UML (Unified Modeling Language), kurioje yra specialiai taisyklių aprašymui sukurta kalba – OCL (Object Constraint Language). OCL – yra formali logikos pagrindu paremta kalba, neturinti grafinės notacijos (Rational 2002).

Verslo taisyklės skirtos skirtingais būdais riboti verslą. Iš verslo sistemų nuleidžiant reikalavimus žemyn jie atsiranda informacinėse sistemos ir tokiu pat būdu programų sistemose, realizuotos kaip programų sistemų kodas.

Verslo sistemoje verslo taisyklė susideda iš trijų dalių: inkaro (*angl. anchor*), korespondentų (*angl. correspondent*), bei sąlygos. Inkaras suteikia jungimosi tašką, nuo kurio reikia pradėti interpretuoti taisyklę. Visos taisyklės savo sąlygose kreipiasi į modelio egzempliorius, jas vadina korespondentais (OMG 2003). Pagal (OMG 2003) verslo sistemoje išskiriamos sekančios verslo taisyklių šeimos: egzempliorių tikrinimo (*angl. instance verifiers*), rūšies tikrinimo (*angl. type verifiers*), pozicijos tikrinimo (*angl. position verifiers*), funkcinio tikrinimo (*angl. functional verifiers*), palyginimo (*angl. comparative evaluators*), matematinio įvertinimo (*angl. mathematical evaluators*), projekcijos kontrolės (*angl. projection controllers*). Kiekvienas iš aukščiau paminėtų taisyklių tipų gali būti tiesiogiai atvaizduotas į atitinkanti OCL loginį operatorių, arba jų derinį (OMG 2003). OCL kalba tapo UML specifikacijos dalimi nuo 1.1 versijos. Sakiniai teksto pavidalu gali būti prikabinėti prie UML klasės, arba klasės metodo.

OCL kalbos teiginiai negali egzistuoti be UML modelio, jų teisingumas gali būti tikrinamas tik atsižvelgiant į atitinkamą modelio realizaciją. OCL sakinio struktūra:

Context <Klasifikatorius> inv: <RibojimoPavadinimas>: Sąlyga

Kiekvienas OCL ribojimas prasideda nurodant kontekstą (*angl. Context*) kuriame jis galioja. Kaip kontekstas gali būti naudojamas UML modelio klasifikatorius (pvz. klasės pavadinimas, atributas) arba klasifikatoriaus metodas. Priklausomai nuo ribojimo konteksto gali būti naudojama invariantas, raktinis žodis *inv:*, kai kontekstas klasifikatorius, arba prieš-sąlygos, raktinis žodis *pre:*, ir po-sąlygos, raktinis žodis *post*. Ribojimas visada baigiasi sąlyga. Invarianto sąlygą turi tenkinti kiekvienas modelio egzempliorius. Kai ribojimas uždedamas UML klasės metodui prieš-sąlyga turi būti patenkinta prieš metodo vykdymą o po-sąlyga turi būti teisinga po metodo vykdymo pabaigos.

OCL sakiniai gali būti atvaizduotos aktyvioje duomenų bazėje kelias skirtingais būdais: panaudojant tvirtinimo sakinius, naudojant rodinius. Siekiant transformuoti OCL sakinius į ECA taisyklės, paprastai realizuojamas kaip trigeriai, būtina išspręsti sekančias problemas (Demuth 2001).

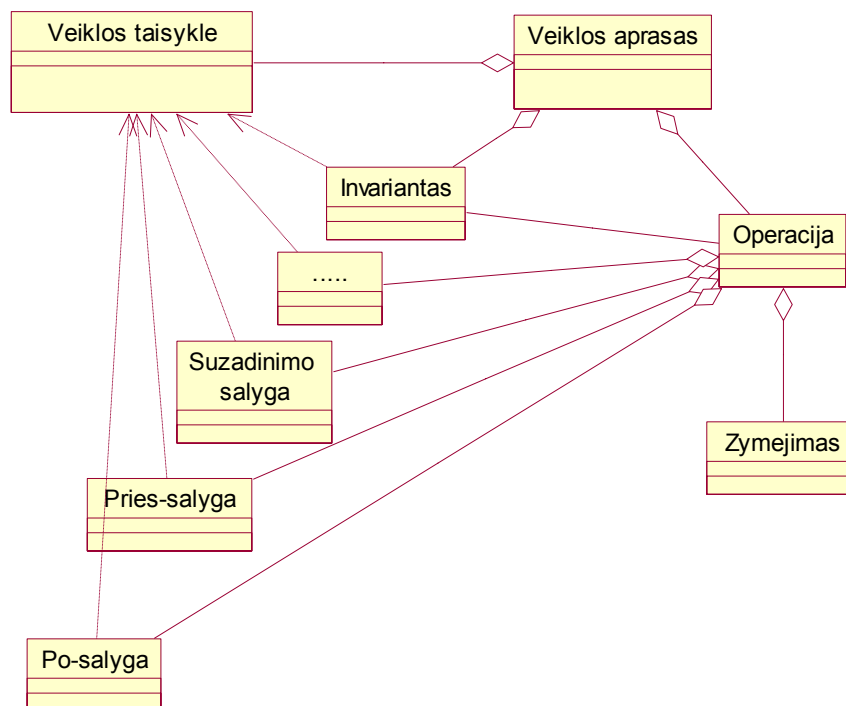
- Įvykis kuris iššaukia trigerį;
- Trigerio inicijavimo dažnis (angl. *granularity*);
- Trigerio sąlyga;
- Trigerio suveikimo laikas;

Iš OCL specifikacijos aišku, kad OCL invarianto sakinis turi būti tenkinamas bet kokio modelio egzemplioriaus. Siekiant patenkinti būtina šitą reikalavimą būtina sekti kiekvieno OCL sąlygoje panaudoto atributo būklę. Tai reiškia, kad invarianto atveju trigeris turi būti iššauktas įterpian/keičiant/trinant kortežus kurių atributai yra panaudoti OCL sąlygoje.

Trigerio suveikimo laikas numato kada bus patikrinta sąlyga, prieš įvykdant trigerio įvykį ar po jo. Pagal (Demuth 2001) jei trigeris suveikia ir sąlyga yra įvertinama prieš iššaukusį įvykį veiksmą sąlyga neturi prieigos prie keičiamų duomenų. Tikrinti sąlyga ir prieš ir po yra ne efektyvu, todėl sąlyga turi būti tikrinama po veiksmo atlikimo.

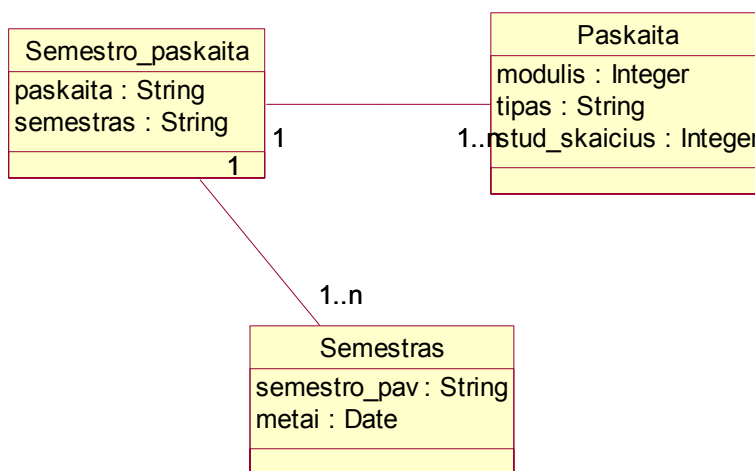
Įterpian n įrašų iš eilės trigeris gali suveikti n arba 1 kartą, priklausomai nuo trigerio inicijavimo dažnio. Trigeriai gali turėti skirtingą inicijavimo dažnį; teiginio lygio, eilutės lygio, įvertinant OCL sąlygą reikalingas eilutės lygio inicijavimo dažnis.

Organizacijoje esančią veiklą galime specifikuoti kaip veiklos taisyklių rinkinį, iš kitos pusės kaip operacijų ir su jomis susijusių invariantų rinkinį. Operacijos susideda iš jos žymėjimo, prieš ir po sąlygų, sužadinimo būsenų ir kitų taisyklių rinkinio. Veiklos taisyklės modelis pateiktas 20 paveiksle.



20 pav. Veiklos taisyklės aprašas OCL kalboje

OCL kalbos pavyzdžiai. Tvarkarasčio sudarymo uždavinio klasių diagramos fragmentas. pateikiamas 21 paveiksle.



21 pav. Dalykinės srities klasių diagrama

OCL Invariantai. OCL kalboje naudojami invariantai. Pateiktame pavyzdyje bus parodyta jų naudojimo ypatumai.

Taisyklė 1 - atributo tipo apribojimas, kuris tikrina, ar darbuotojų kiekis didesnis už 50. Tokiu būdu ši taisyklė reikalauja, kad sąlyga būtų vykdoma.

OCL:

```
context Paskaita inv ArUztenkaStud:
```

```
self.stud_skaicius > 50
```

SQL:

```
create table Paskaita( ... ,  
constraint ArUztenkaStud check (stud_skaicius > 50))
```

OCL invariantų formalus užrašymas

Vardas: OCL INVARIANT

Aprašymas: formos OCL invariantas

```
context <klasės vardas> inv <apribojimo vardas>:  
<OCL išraiška(self)>
```

Galimas kitoks aprašymas:

```
<klasės vardas>.allInstances -> forAll (<OCL išraiška(self)>).
```

Taip pat OCL išraiškas galime aprašyti ir SQL predikatais

Apribojimas aprašytas sekančiai:

```
create assertion <apribojimo vardas>  
check (not exists  
(select * from <context lentelės> SELF  
where not (<OCL išraiška(self)>))
```

Apibendrinant galima pasakyti, kad siūlomi modeliuoti veiklos taisykles UML ir OCL, bei vėliau juos realizuoti DB technologijomis yra vienas iš būdų. Tačiau naudojant šį metodą neišvengiamos problemos:

- Modeliuoti veiklos taisykles UML konstrukcijų neužtenka, todėl naudojama OCL, ji yra formali kalba, neturinti grafinės notacijos, todėl sunku VT modelius suprasti probleminės srities specialistams.
- Siūlomi būdai nenagrinėja taisyklių, kurios specializuoja verslo algoritmus, procedūras ar tvarką, jų automatinė realizacija negalima siūlomą būdu.

4.2 Veiklos taisyklių vaizdavimas duomenų schemomis

Tiesiogine schemų inžinerija suprantamas transformavimo procesas, kurio metu išsamios EER diagramos keičiamos standartizuotu RDB aprašu (Paradauskas). Šiame procese galima patirti semantinius nuostolius, nes ne visos transformacijos išsaugo schemų semantiką. Toliau pateiktos geriausios žinomos transformavimo taisyklės, leidžiančios tiksliausiai atvaizduoti RDB dalykinės srities semantiką.

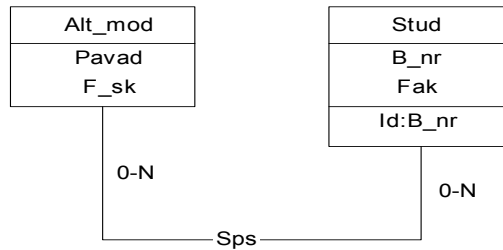
Taisyklės sugrupuotos į tris grupes. Pirmai grupei priskirtos taisyklės, kuriomis EER diagrama pertvarkoma EER modelio ribose. Antrajai grupei priskirtos taisyklės, kuriomis eliminuojami apibendrinimo ryšiai ir EER modelis paverčiamas ER. Trečiajai grupei priskirtos taisyklės, kuriomis ER modelis verčiamas logine RDB schema. Kartais jos atliekamos dviem žingsniais: pirmuoju žingsniu transformuojama konceptualioji schema, antruoju ji paverčiama logine RDB schema. Šios transformacijos vadinamos neutraliosiomis, kadangi jos nepriklauso nuo konkrečios DBVS.

Tokios pat transformacijos vykdomos ir atvaizduojant OO modelį reliacine schema. OO modelių transformacijos į reliacines schemas išanalizuotos (Rumbaugh, 1991), kur nagrinėjami ir daugialypio apibendrinimo atvejai, kai potipis turi keletą supertipų. Daugialypis paveldėjimas nėra realizuotas ir žinomose objektinėse duomenų bazėse, todėl norint realizuoti konceptualiuosius tokio tipo modelius reikia sukurti papildomas konstrukcijas (įvesti tipo ir jo supertipų ryšio esybę).

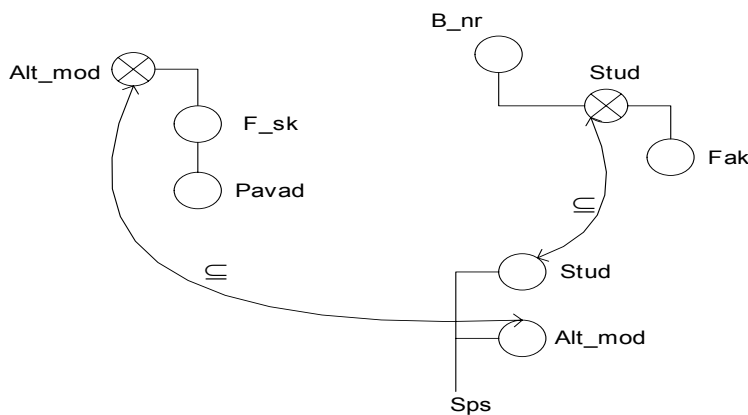
Sakykime, nustatyti tokie dalykinės srities apribojimai:

1. Studentas turi vieną bilietą su unikaliu numeriu (B_NR) ir studijuoja viename universiteto fakultete (FAK).
2. Sudarydamas individualiąją studijų programą (SPS) einamiesiems mokslo metams studentas gali nepasirinkti jokio alternatyvaus modulio (ALT_MOD) arba gali pasirinkti keletą.
3. Kiekvienas alternatyvus modulis turi vieną pavadinimą (PAVAD) su modulio vedimo formų skaičiumi (F_SK).
4. Alternatyvų modulį gali būti nepasirinkęs joks studentas arba gali būti pasirinkę keli studentai.

Šie apribojimai pavaizduoti EER diagrama (22 pav.) ir objektų savybių (OP) modelio konceptualiąja schema (23 pav.).



22 pav.EER diagrama



23 pav. Studijų programos OP diagrama

OP schemeje neparodyti savaime suprantami vidiniai funkciniai tipai - griežtosios ir pilnosios surjekcijos:

$Alt_mod1 : ALT_MOD \rightarrow PAVAD;$

$Alt_mod2 : ALT_MOD \rightarrow F_SK;$

$Stud1 : STUD \rightarrow FAK;$

$Sps1 : SPS \rightarrow STUD;$

$Sps2 : SPS \rightarrow ALT_MOD ;$

vidinės griežtosios bijekcijos:

$stud2: STUD \rightarrow B_NR:$

$stud3 : B_NR \rightarrow STUD$

ir išorinės griežtosios bijekcijos:

$sps-alt_mod1 : SPS.ALT_MOD \rightarrow ALT_MOD;$

$sps-stud1 : SPS.STUD \rightarrow STUD.$

Panagrinėkime vienos konkrečios dalykinės srities būsenos konceptų ALT_MOD, STUD ir SPS

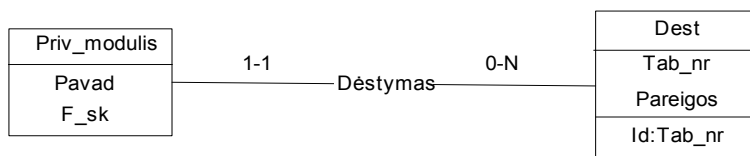
24 paveiksle pateiktas kitas pavyzdys su tokiais apribojimais:

Privalomas modulis PRIV_MOD turi vieną pavadinimą ir tam tikrą dėstyto formų skaičių.

Privalomas modulis turi būti dėstomas (DĖSTYMAS) ir tam moduliui turi būti vienas atsakingas dėstytojas.

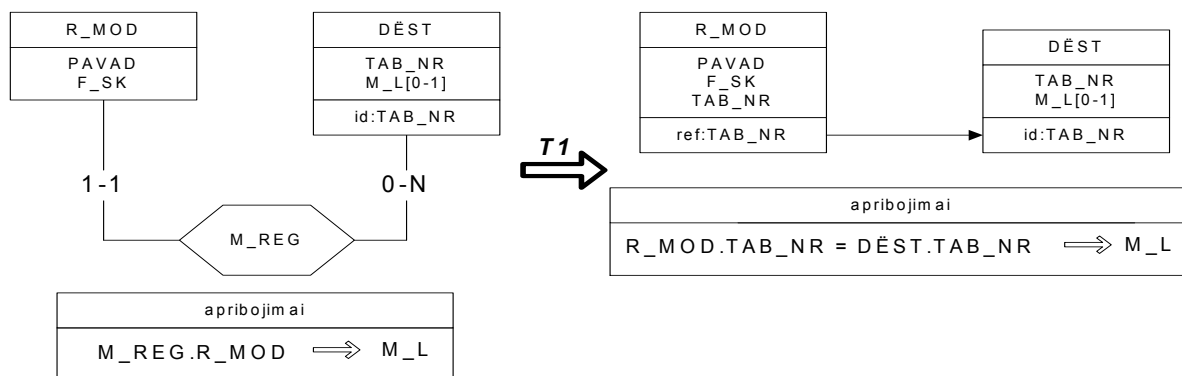
1. Visi dėstytojai (DĖST) turi individualius tabelio numerius (TAB_NR) ir užima tam tikras pareigas.

2. Kai kurie dėstytojai gali nebūti atsakingi už privalomų modulio dėstymą ir tas pats dėstytojas gali būti atsakingas už kelių modulių dėstymą.

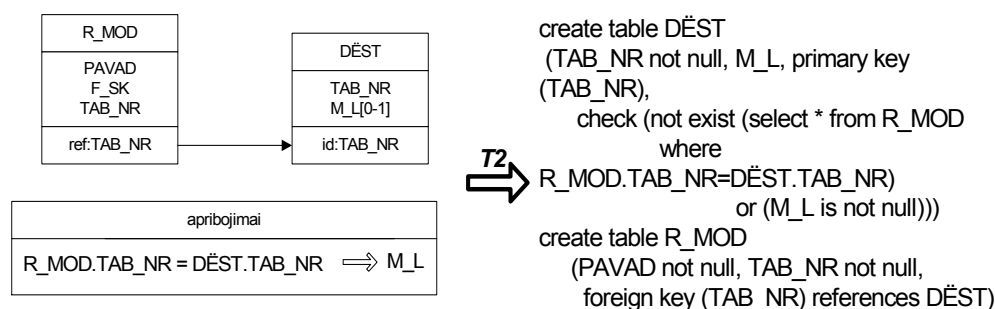


24 pav. pavyzdys

Apribojimai, nustatyti atributų implikacijomis, kontroliuojami įvedant apribojimų tikrinimo funkcijas *check()*. Paimkime kitą pavyzdį apie modulių registraciją (M_REG). Modulus (R_MOD) – tiek privalomus, tiek alternatyvius – turi teisę registruoti mokslo laipsnį (M_L) turintys dėstytojai. Tačiau esybės DĖST egzemplioriais gali būti ir dėstytojai, neturintys mokslo laipsnio. Todėl esybės DĖST atributas yra neprivalomas ir jam nurodomas kardinalumas [0-1] (25 pav.). Apribojimas, kad kiekvienas dėstytojas, registruojantis modulį, turi turėti mokslo laipsnį, priskiriamas ryšiui M_REG ir gali būti pateikiamas implikacijos pavidalu.



25 pav. Ryšio M_REG su nustatytais apribojimais eliminavimas naudojant EER modelio transformaciją T1



26 pav. EER modelio su implikacijomis nustatytais apribojimais transformavimas T2 į RDB loginės schemos aprašą

3 lentelė. Funkcijos *check()* galimos loginės reikšmės

Exist()	not exist()	M_L is not	check()
0	1	1	1
0	1	0	1
1	0	1	1
1	0	0	0

Pateiktoje RDB loginės schemos specifikacijoje (27 pav.) yra du išoriniai apribojimai: išorinio rakto apribojimas ir atributinės implikacijos.

Dėl pirmojo apribojimo *foreign key (TAB_NR) references DĖST* į lentelę R_MOD negalima įterpti įrašo su atributo TAB_NR reikšme, kurios nėra lentelėje DĖST.

Dėl antrojo apribojimo sudaryta loginio tikrinimo funkcija *check()*. Dėstytojui registruojant koki nors modulį, dėl funkcijos *check()* lentelėje DĖST atributui M_L būtina suteikti kokią nors reikšmę (DR, MAG). Funkcija *check()* kontroliuoja implikacijos

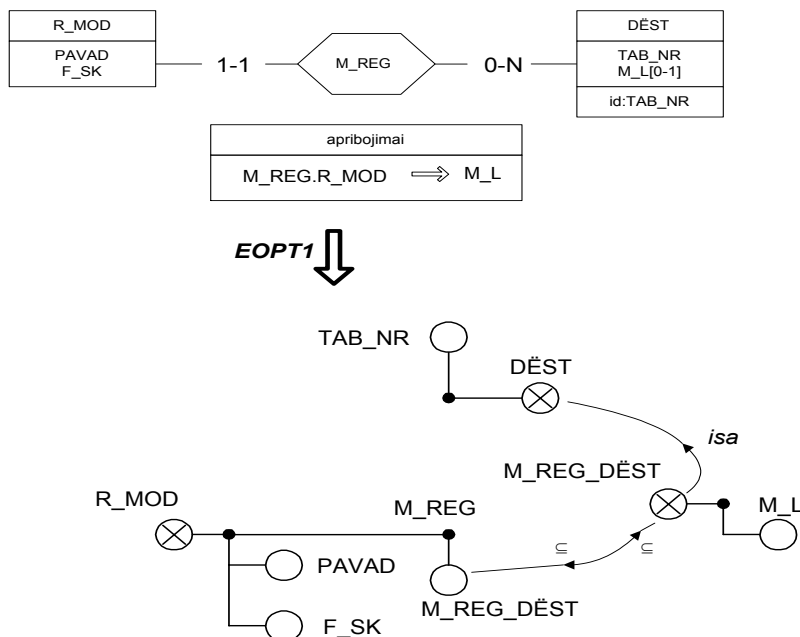
*exist (select * from R_MOD where R_MOD.TAB_NR=DĖST.TAB_NR)*

$\Rightarrow M_L \text{ is not null}$

teisingumą (3 lent.). Jeigu esant lygybei $R_MOD.TAB_NR = DĖST.TAB_NR$ atributas M_L turi neapibrėžtą (neįvestą) reikšmę \perp , ši funkcija įgyja reikšmę 0. Konkrečios RDBVS terpėje sugeneruota procedūra su vidine funkcija *check()* leis vykdyti DB modifikavimo programą, jeigu lentelėje DĖST įvedant kortežą iš karto bus nurodyta atributo M_L reikšmė arba lentelėje

R_MOD įterpiančią kortelę su nauja TAB_NR reikšme kartu bus įvedama lentelės DĖST kortelė (su ta pačia atributo TAB_NR reikšme) atributo M_L reikšmė.

Norint specifikuoti tuos pačius apribojimus, eliminuojant ryšį M_REG, vietoje transformacijos *TI* naudojama transformaciją *EOPI* (28 pav.), kurios metu įvedamas naujas objekto R_MOD atributas – mokslo laipsnį turintis dėstytojas (M_REG_DĖST), kuris traktuojamas kaip savarankiškas objektas su privaloma savybe M_L. Naujas objektas M_REG_DĖST bus sujungtas su objektu DĖST poaibio priklausomybe.



28 pav. Išorinio ryšio M_REG su nustatytais apribojimais eliminavimas naudojant *EOPI* transformaciją

4.3 Apibendrinimas

Pagal pateiktą trumpą verslo taisyklių modeliavimo ir realizacijos technologijų apžvalgą, matyti, kad siūlomi įvairūs būdai panaudoti verslo taisyklių koncepciją (4 lentelė). Pasirinkto VT požiūrio taikyme, modeliuojant IS aktyviai dalyvauja tiek mokslininkai, tiek komercinės kompanijos. Pastarieji siūlo produktus, kuriuos tikslinga naudoti su jų kompanijos siūlomomis paslaugomis.

4 lentelė. Palyginimų lentelė

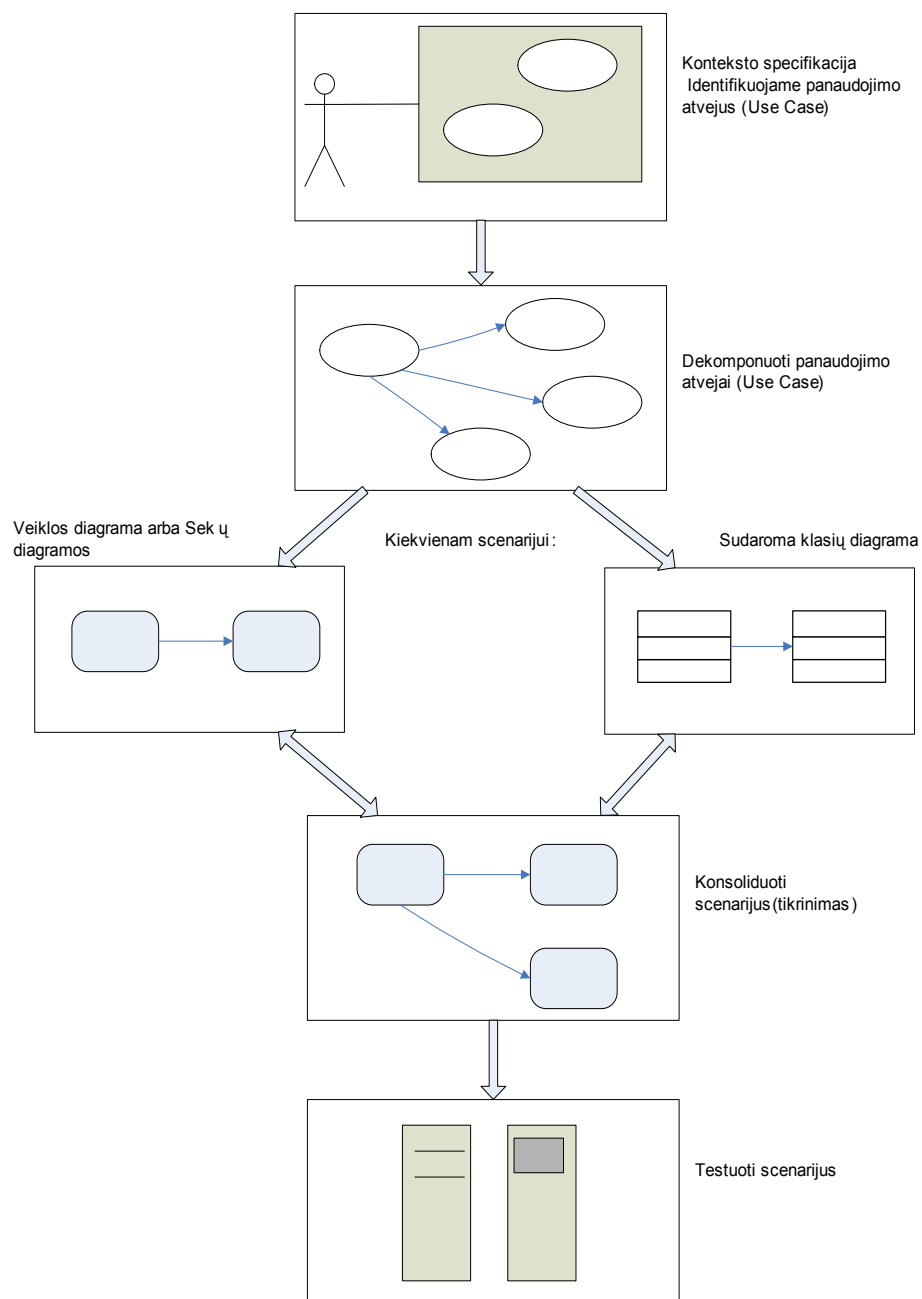
Kriterijus	UML/OCL	Duomenų schemas
Adaptyvumas	+	+
Konfliktų aptikimas	-	+
Deklaratyvi kalba	+	+

Efektyvumas	+	+
VT išreiškiamos aiškiai	+	+
Formalus pagrindas	+	+
VT identifikavimas	+	+
Vientisumas ir nuoseklumas	+	+
Vaizdavimas	+	+
Klasifikavimo palaikymas	+	+
Integracija su egzistuojančiomis sistemomis	+	+

5. Veiklos taisyklių struktūrizavimo proceso modelis

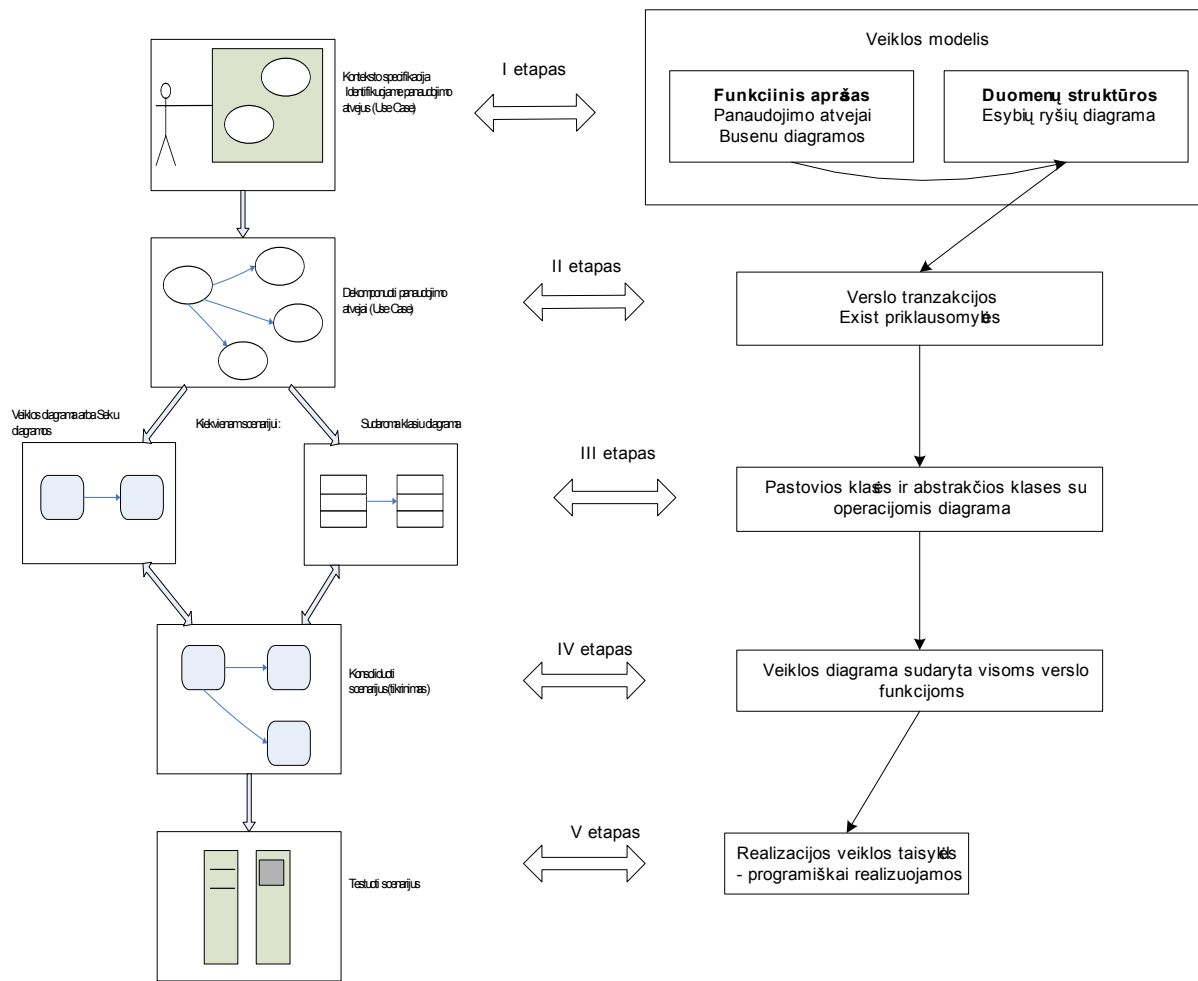
Analizės dalyje išnagrinėti veiklos taisyklių klasifikavimo metodai. Kadangi projektuojame informacinę sistemą, veiklos taisyklės klasifikuojame žiūrint iš projektuotojo pusės, klasifikuojame taisykles naudojantis projektuotojo požiūriu išdėstytu klasifikavimo metodu.

Taip pat projektavimo etape naudojama tipinė projektavimo schema. Ja vadovaujantis yra išskiriami projektavimo etapai veiklos taisyklių požiūriu, kurie atvaizduoti proceso modelyje.



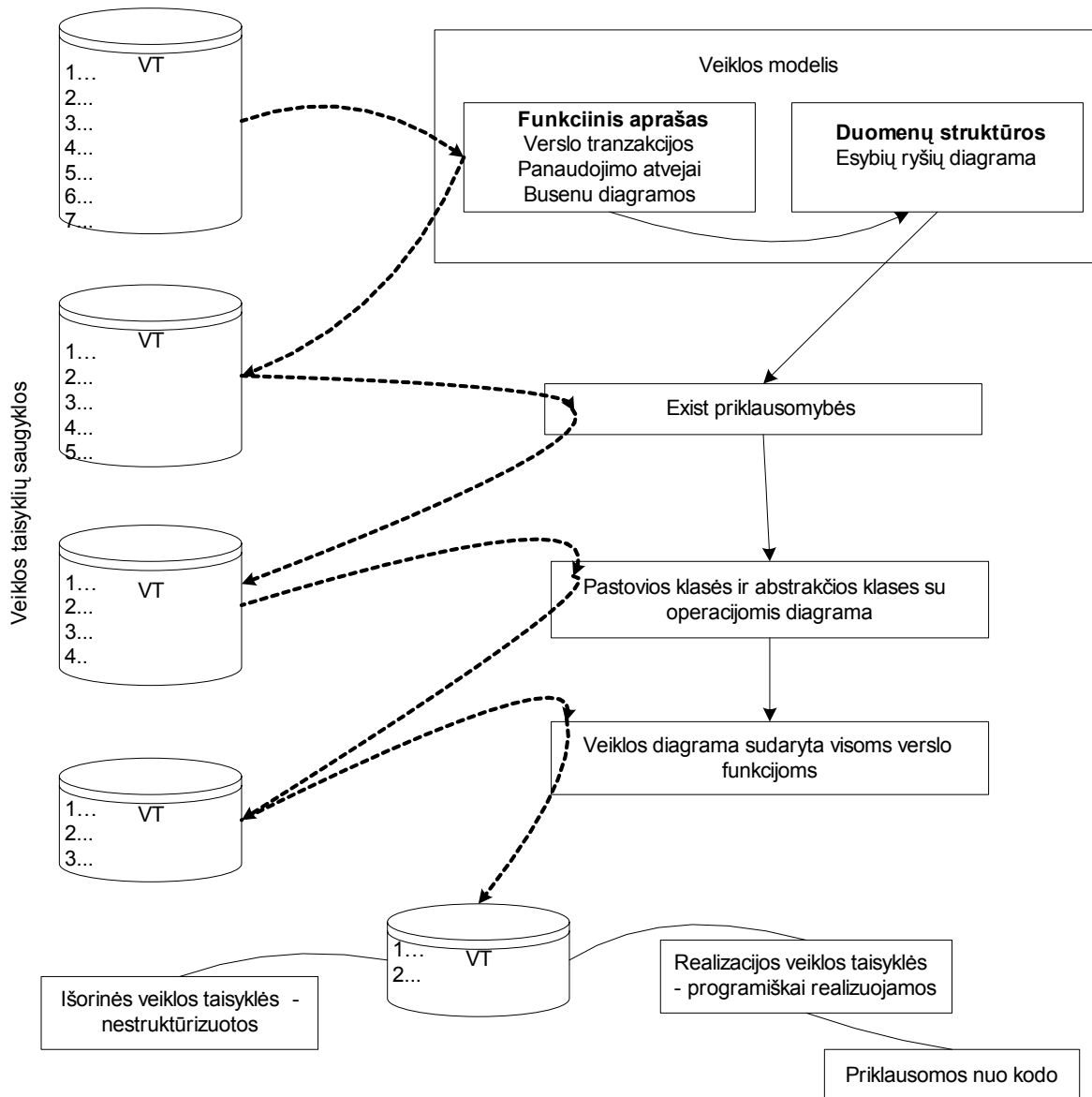
29 pav. Tipinė projektavimo schema

Remiantis tipine projektavimo schema yra siūloma veiklos taisyklių kūrimo technologija, panaudojant duomenų bazės struktūras. Siūlomas VT struktūrizavimo proceso modelis iš esmės pagrįstas veiklos taisyklių struktūrizavimu. Pradedant projektavimą nuo veiklos modelio, pažingsniui einama link išbaigto duomenų modelio.



30 pav. Tipinės projektavimo schemas ir struktūrizavimo proceso projektavimo etapai

Veiklos modelį galima aprašyti naudojant verslo tranzakcijas, esybių ryšių diagramas, kur veiklos taisyklės „įvedamos“, kai sudaromi ryšiai, nurodomi būsimi konceptai, būsenos objektų, jų elgesys. Vykdoma realizacija šiame etape SQL kalba naudojame CHECK funkcija, kuri nurodo apribojimus duomenų struktūroms. Operacijoms, veiklos apribojimams realizuoti naudojama SQL funkcija TRIGGER.



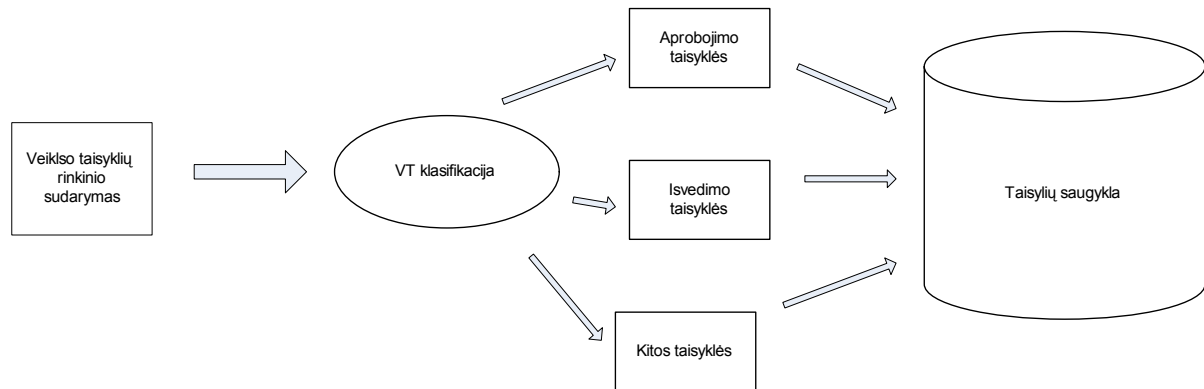
31 pav. Veiklos taisyklių struktūrizavimo proceso modelis

Projektavimo metu veiklos taisyklės įvertinant kiekviename etape, jų lieka realizuoti programiškai vis mažiau, ir lieka tik „grynos“ veiklos taisyklės, kurias galima realizuoti tik programiškai. Tokiu būdu siekiama jau projektavimo metu, struktūrizuoti verslo taisyklės per verslo tranzakcijas, sudėtinius objektų savybių modelius, aiškiai nubrėžiant, kur būtent yra „prikabinamos“ verslo taisyklės.

6. Eksperimentinė sistemos realizacija

6.1 Veiklos taisyklių rinkinio sudarymas

Veiklos taisyklių rinkinio sudarymo proceso apraše, parodomas saugyklos sudarymo principas, vykdant veiklos taisyklių IS projektavimą .



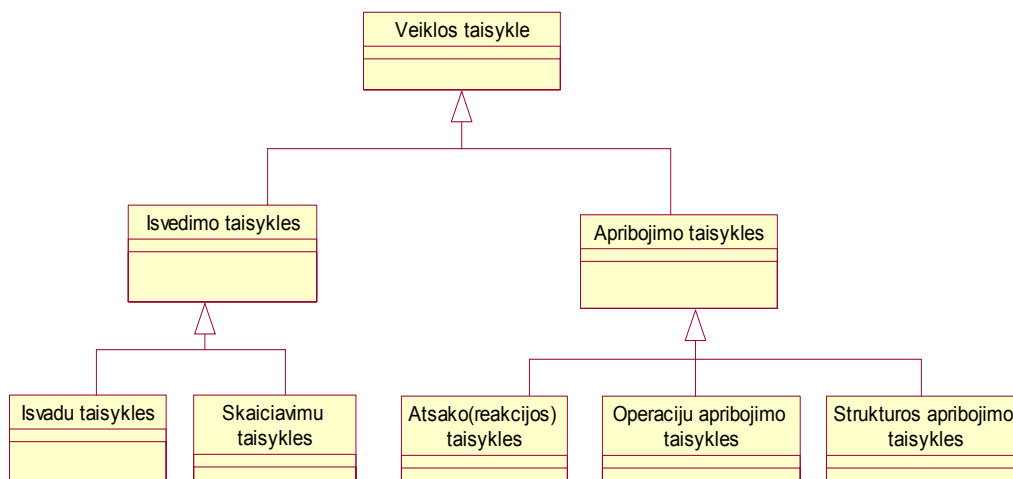
32 pav. Veiklos taisyklių klasifikavimo procesas

Pagal pasirinktą dalykinę veiklos sritį buvo sudarytas veiklos taisyklių rinkinys. Sudarant šį rinkinį nebuvo siekta pilnai charakterizuoti veiklą. Kadangi VT realizuotas dalinai, tai buvo parinktos tokios taisyklės, kurias sistema galėtų išpildyti.

Buvo suformuluoti šie VT teiginiai:

1. Paskaitos vyksta kiekvieną dieną, išskyrus savaitgaliais
2. Tvarkaraštį keisti gali tvarkaraščio sudarinėtojas
3. Studentų grupės skaičius turi būti didesnis už 20.
4. Studentas privalo turėti studento pažymėjimą
5. Paskaitos pravedamos laisvoje auditorijai
6. Negali vykti paskaitos nuo 19.00.
7. Paskaitos laboratoriniams pravesti turi būti kompiuteriu auditorijoje
8. Auditorijos pasirinkimas priklauso nuo studentų skaičiaus ir priemonių paskaitai pravesti kiekio
9. Vardas yra dėstytojo atributas
10. StudentoNr yra studento atributas
11. Studentas arba dėstytojas yra asmuo
12. Dėstytojas priklauso katedrai
13. Paskaitos gali vesti dėstytojas
14. Tvarkaraštis paskaitų turi būti žinomas studentams
15. Studentas prisijungia prie sistemos tik su studento numeriu.
16. Tvarkaraščio sudarinėtojas sudaro tvarkaraštį.

Kadangi visi klasifikavimo būdai tarpusavyje yra panašūs, pasirinkome tokį klasifikavimo būdą, kuris labiausiai tiktų projektuotojui, kuriant IS. Dažniausiai pasitaiko tokios taisyklių grupės: struktūrinės taisyklės – šios taisyklės apibrėžia duomenų struktūras. Veiksmo taisyklės – apibrėžia operacijas ir sekas, kurios turi būti įvykdytos. VT klasifikavimo būdas išdėstytas 33 paveiksle.



33 pav. Veiklos taisyklių klasifikacija

Toliau yra vykdoma veiklos taisyklių klasifikacija pagal pasirinktą klasifikacijos metodą.

Aprobojimo taisyklės

. Aprobojimo taisyklės skaidomos į atsako, operacijų ir struktūros aprobojimo taisykles. Aprobojimo taisyklės gali būti suskirstytos į dvi stambias kategorijas pagal jų kilmę: struktūriniai aprobojimai, kuriais mes nusakome kaip objektai siejasi vienas su kitu, tai pastovūs faktai ir elgesio aprobojimai, kurie aproboja taisykles vykdymui ar nusako priklausomybę viena nuo kitos.

1. Struktūros aprobojimo taisyklės

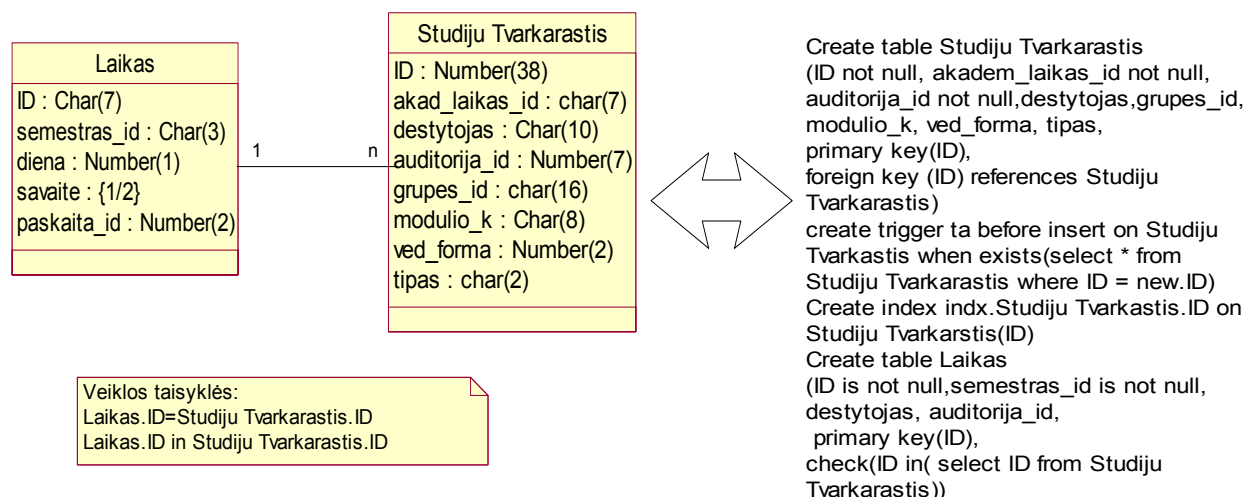
4 lentelė Struktūros aprobojimo taisyklės

Indeksas	Veiklos taisyklė	Projektavimo elementas
VST1.1	„Vardas yra dėstytojo atributas“	Klasių diagrama
VST1.2	„StudentoNr yra studento atributas“	Klasių diagrama
VST1.3	„Dėstytojas yra universiteto darbuotojas“	Klasių diagrama
VST1.4	„Paskaitos gali vesti dėstytojas“	Klasių diagrama
VST1.5	„Tvarkaraštis paskaitų turi būti žinomas studentams“	Klasių diagrama
VST1.6	„Studentas privalo turėti studento pažymėjimą“	Klasių diagrama
VST1.7	„Paskaitos pravedamos laisvoje auditorijai“	Klasių diagrama

VST1.8	„Dėstytojas priklauso katedrai“	Klasių diagrama
VST1.9	„Paskaitos laboratoriniams praveisti turi būti kompiuteriu auditorijoje“	Klasių diagrama

Išanalizavus veiklos taisyklių kategorija struktūriniai teiginiai ir detaliau terminų apibrėžimo ir jų klasifikavimą galime teigti kad tiek bendri tiek veiklos terminai UML modelyje gali būti vaizduojami klasėmis.

Detaliau panagrinėkime veiklos taisyklių vaizdavimą duomenų bazių schemomis ir jų užrašymą RDB schemomis:



34 pav. Duomenų modelio struktūra ir jos užrašas RDB schema

Struktūrinės taisyklės galima užrašyti ir Oracle Sql kalba:

Oracle SQL sakinyje CREATE TABLE yra išvardijami klasės atributai atskiriami kableliais ir nurodomi jų duomenų tipą.

CREATE TABLE STUDENTAS_T

```

(student_nr NUMBER (6),
vardas varchar (20),
pavarde varchar (30));
    
```

```

ALTER TABLE Studentas_T
  Add constrain tikrinti
CHECK (
  (studNr is null and tabelioNr is not null) or (studNr is not
  null and tabelioNr is null));

CREATE TABLE STUDENTAS (
  vardas      string (20) NULL,
  pavarde    string(30) NULL,
  studNr     number (6) NOT NULL,
  CONSTRAINT unikalumas1 UNIQUE (studNr);
  
```

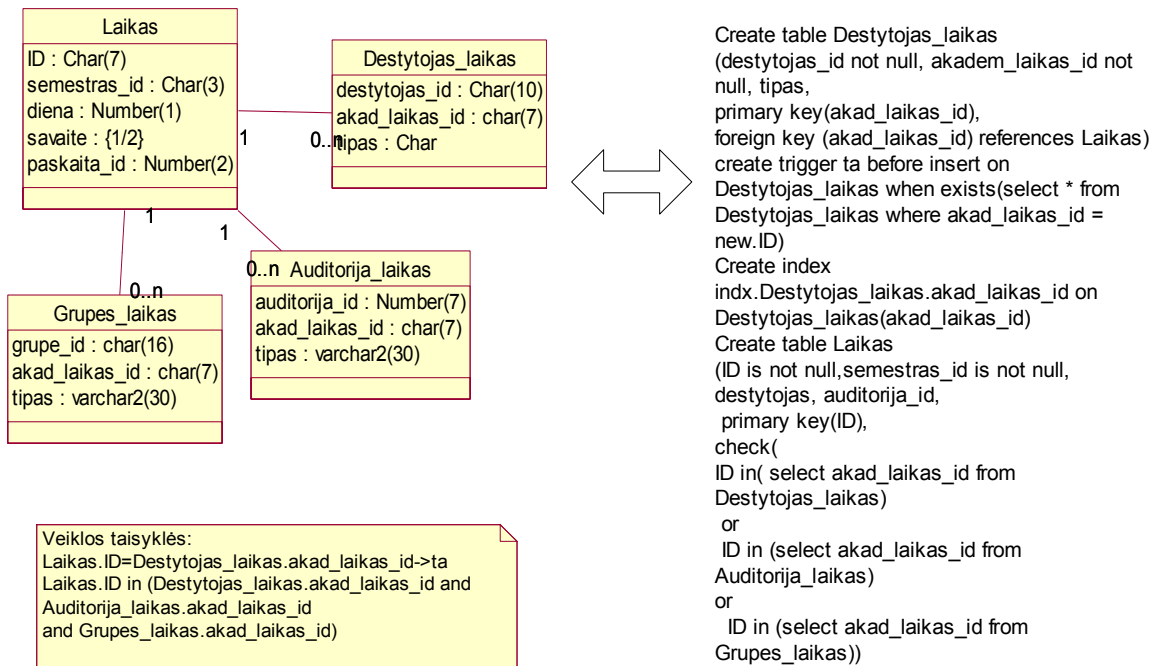
2. Operacijų apribojimo taisyklės

5 lentelė Operacijų apribojimo taisyklės

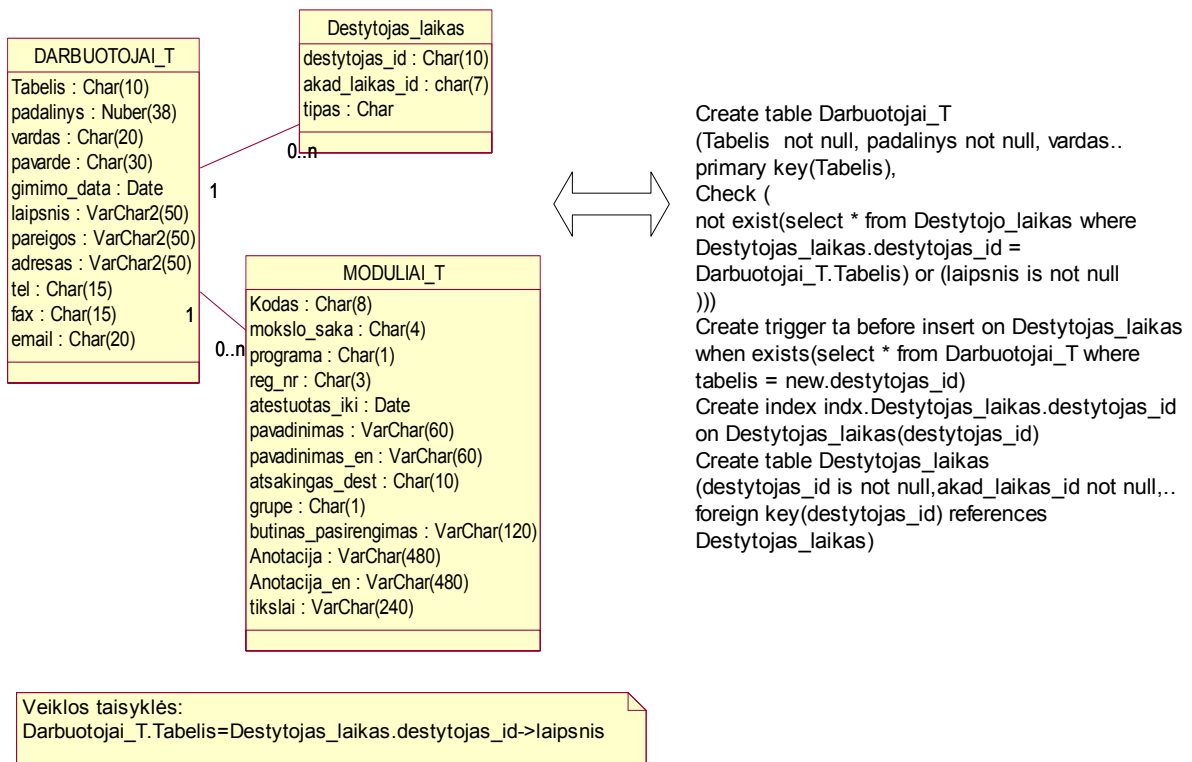
Indeksas	Veiklos taisyklė
VOP1.1	„Tvarkaraštis yra pataisytas“
VOP1.2	„Laikas įvestas privalo“.
VOP1.3	„(Tik) administratorius gali atlikti ištrinimo veiksmą“.
VOP1.4	„Tas pats studentas negali priklausyti dviem skirtingoms to pačio modulio grupėms.“
VOP1.5	„Paskaitos turi tenkinti dėstytojų, grupių, auditorijų laikų fondus.“
VOP1.6	„ Toje pačioje auditorijoje negali vykti skirtingos paskaitos tuo pačiu metu,“
VOP1.7	„Tas pats dėstytojas negali turi skirtingų paskaitų tuo pačiu metu.“
VOP1.8	„Dėstytojas privalo turėti mokslinį laipsnį, kad galėtų registruoti modulį“

Detaliau panagrinėkime veiklos taisyklių vaizdavimą duomenų bazių schemomis ir jų užrašymą RDB schemomis:

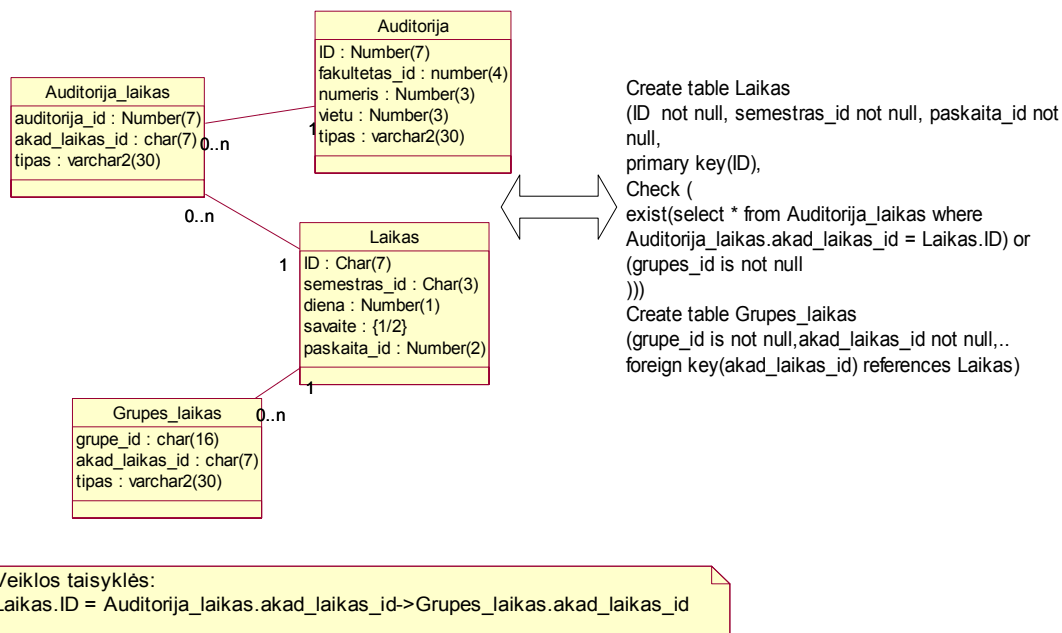
„Paskaitos turi tenkinti dėstytojų, grupių, auditorijų laikų fondus.“



35 pav. Taisyklės „Paskaitos turi tenkinti dėstytojų, grupių, auditorijų laikų fondus“



36 pav. Taisyklės „Dėstytojas privalo turėti mokslinį laipsnį, kad galėtų registruoti modulį“



37 pav. „Toje pačioje auditorijoje negali vykti skirtingos paskaitos tuo pačiu metu,“

3. Atsako(reakcijos) taisyklės

6 lentelė Reakcijos taisyklės

Indeksas	Veiklos taisyklė
VAT1.1	<i>Kai studentas vykdo prisijungimą prie sistemos jeigu jis turi studento numerį tuomet vykdomas prisijungimas prie sistemos</i>
VAT1.2	<i>„(Tik) administratorius gali atlikti ištrinimo veiksmą“.</i>
VAT1.3	<i>„Tas pats studentas negali priklausyti dviem skirtingoms to pačio modulio grupėms.“</i>

Išvedimo taisyklės

1. Išvadų taisyklės

7 lentelė Išvadų taisyklės

Indeksas	Veiklos taisyklė
VIS1.1	<i>„Grupių turimų paskaitų akademinių valandų kiekis turi būti lygus modulio vedimo formoje nurodytam akademinių valandų kiekiui“</i>

Išvedimo teiginiai gali būti arba matematiniai skaičiavimai(toks kaip „nuomos kaina yra lygi padauginus pagrindinę kainą iš nuomojamų dienų skaičiaus“) arba išvados (inference).

Matematinis skaičiavimas sukuria išvestą faktą pagal nurodytą algoritmą. Išvada sukuria išvestą faktą naudojant logines indukcijas arba dedukcijas.

2. Skaičiavimo taisyklės

8 lentelė Skaičiavimo taisyklės

Indeksas	Veiklos taisyklė
VSK1.1	„Grupių turimų paskaitų akademinų valandų kiekis turi būti lygus modulio vedimo formoje nurodytam akademinų valandų kiekiui“

Veiklos taisyklių klasifikavimo klasių realizacijos būdai

9 lentelė Veiklos taisyklių realizavimas iš veiklos modelio į IS lygį naudojant turimas technologijas

Veiklos taisyklių klasifikavimo klasė	Realizacija
Struktūros apribojimo taisyklės	If-then sąlygos programavimo kalbose; DOMAIN, CHECK and CONSTRAINT funkcijos SQL lentelių aprašuose; CREATE ASSERTION sąlygos SQL duomenų bazių schemų apraše
Išvedimo taisyklės	SQL CREATE VIEW funkcijos
Atsako(reakcijos)taisyklės	If-then sąlygos programavimo kalbose; CREATE TRIGGER funkcijos SQL'e;

Veiklos taisyklių „mapinimas“ nėra realizuojamas taip paprastai programavimo kalbomis. SQL turi savyje standartinių funkcijų, kuriomis tokias sudėtingas veiklos taisykles galima realizuoti. Toliau yra pateikiamas SQL trigerio pavyzdys:

```
CREATE trigger Rule1 ON Moduliai
FOR UPDATE
AS
IF UPDATE(status)
BEGIN
if((SELECT statusas FROM INSERTED)='Patvirtintas' and (SELECT id FROM INSERTED))
BEGIN
UPDATE Moduliai
SET Finish_date=getdate()
WHERE id=(SELECT id FROM INSERTED)
```

END

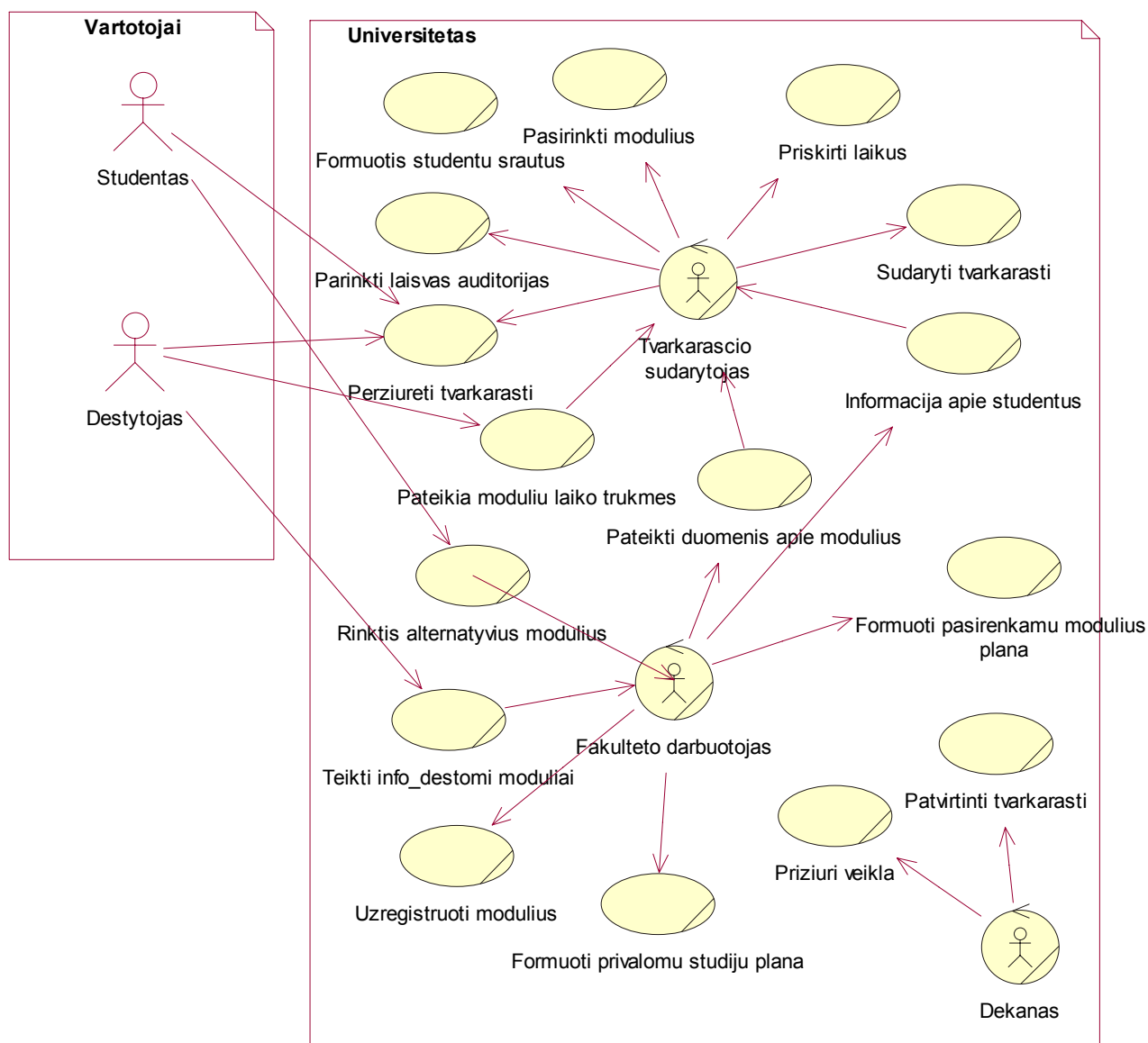
END

6.2 Projektavimo dalis

Vartotojų poreikių analizei ir duomenų modelio projektavimui buvo taikomas objektinio projektavimo metodas Rational Unified ir UML kalba. Projekto sudarymui buvo naudojamas Rational Rose paketas.

I etapas. Sudaromas panaudojimo atvejų modelis

Veiklos panaudojimo atvejų modelis



38 pav. Veiklos panaudojimo atvejų modelis

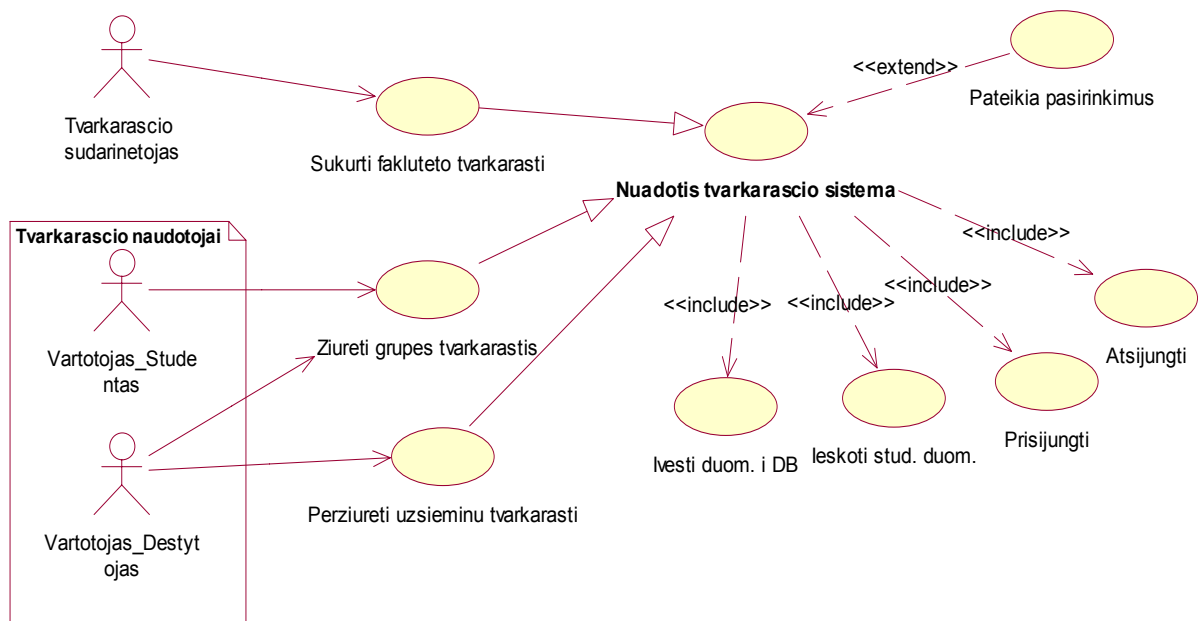
Iš veiklos panaudojimo atvejų diagramos matome, kokius darbo procesus atlieka kiekvienas organizacijos veikėjas

Iš veiklos panaudojimo atvejų diagramos matome, kad fakulteto darbuotojas, atsakingas už tvarkaraščio sudarymą, atlieka tokius organizacijos darbo procesus: (redaguoja) pildo auditorijų laikų fondus, pildo grupių laikų fondus, pildo (redaguoja) dėstytojų laikų fondus (dėstytojų laikų fondus užpildo patys dėstytojai, tačiau raštinės darbuotojas gali atlikti pakeitimus, kurie gali būti būtini korektiškam tvarkaraščio sudarymui), sudaro (redaguoja) studentų grupes (studentų grupes sudarinėja dėstytojai, tačiau raštinės darbuotojas gali atlikti redagavimą), pildo (redaguoja) paskaitas tvarkaraštyje.

Iš veiklos panaudojimo atvejų diagramos, matome, jog dėstytojas atlieka tokius organizacijos darbo procesus: pildo savo laikų fondą, sudaro studentų grupes (sukuria studentų grupes ir priskiria joms studentus) savo dėstomiems moduliams, peržiūri savo paskaitų tvarkaraštį

Iš veiklos panaudojimo atvejų diagramos, matome, jog studentas tik peržiūri savo paskaitų tvarkaraštį.

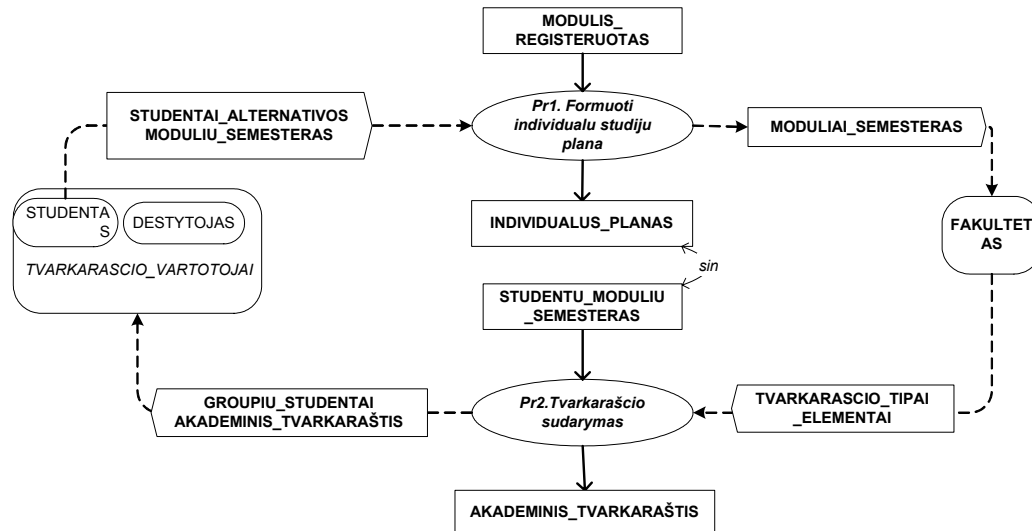
Kompiuterizuojami panaudojimo atvejai:



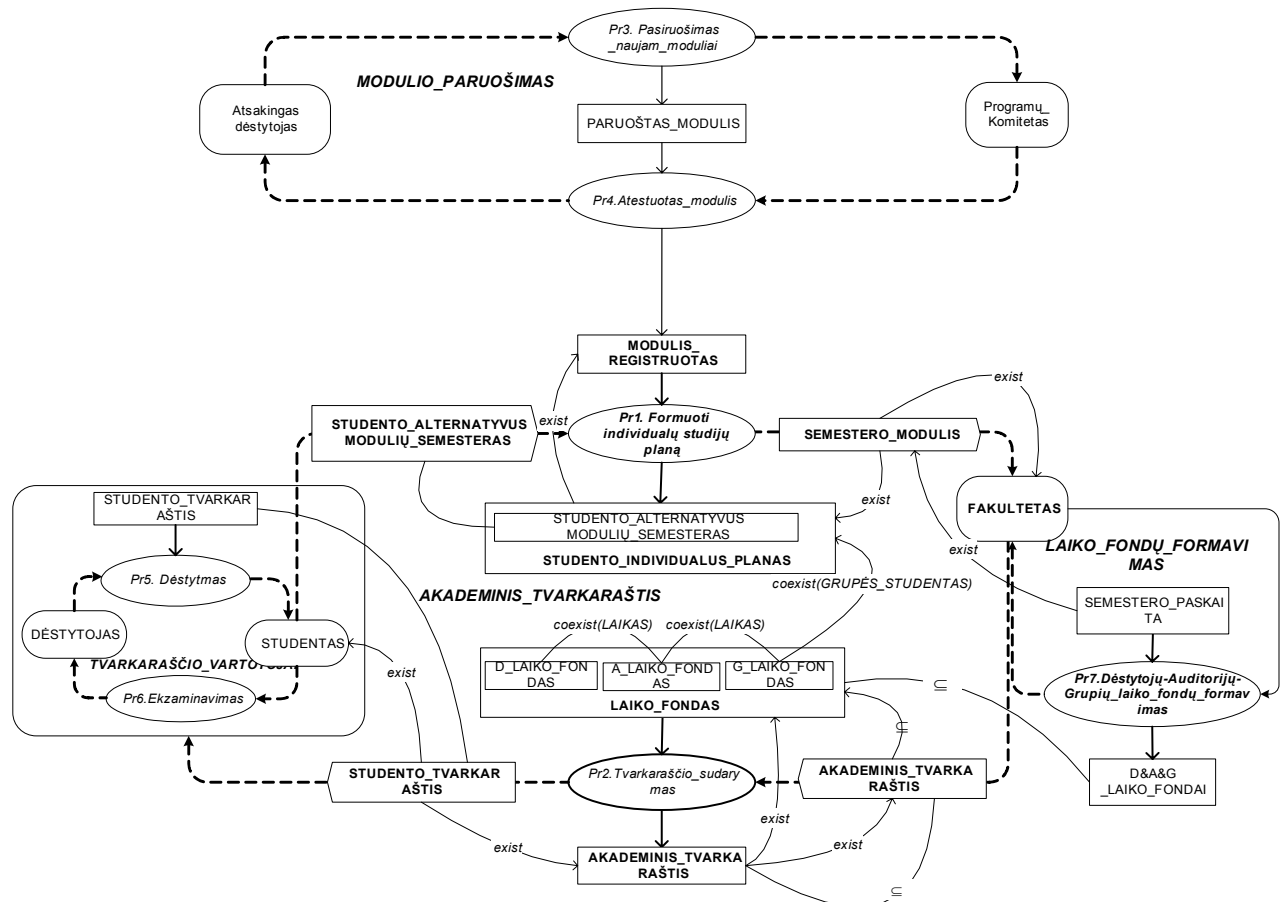
39 pav. Kompiuterizuoti panaudojimo atvejai

2 etapas : Dekomponuojame use casus

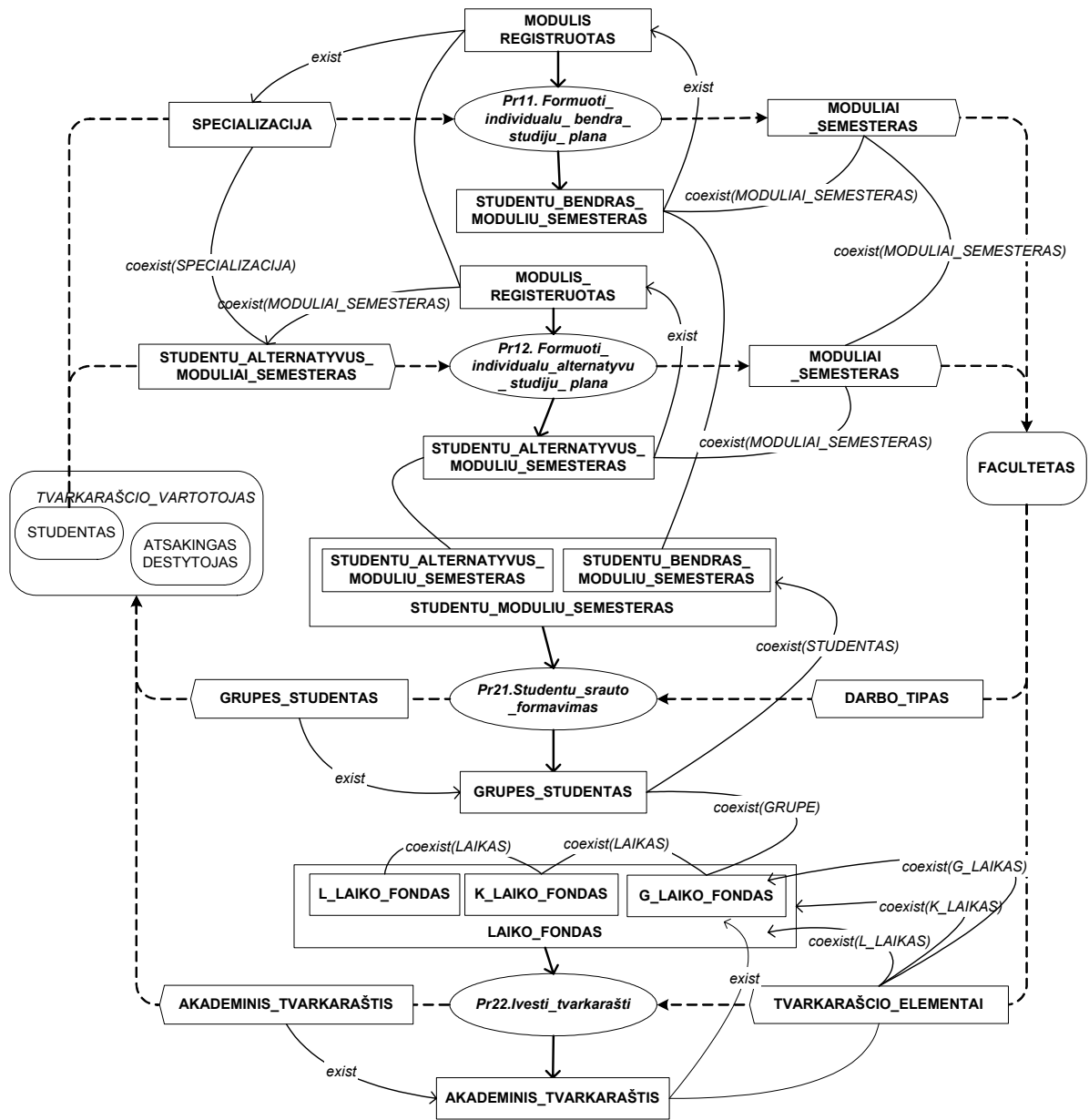
Nagrinėjama veiklos sritis vaizduojama komunikacinėmis kilpomis.



40 pav. Tvarkaraščio kūrimo funkciniai reikalavimai komunikacinėmis kilpomis



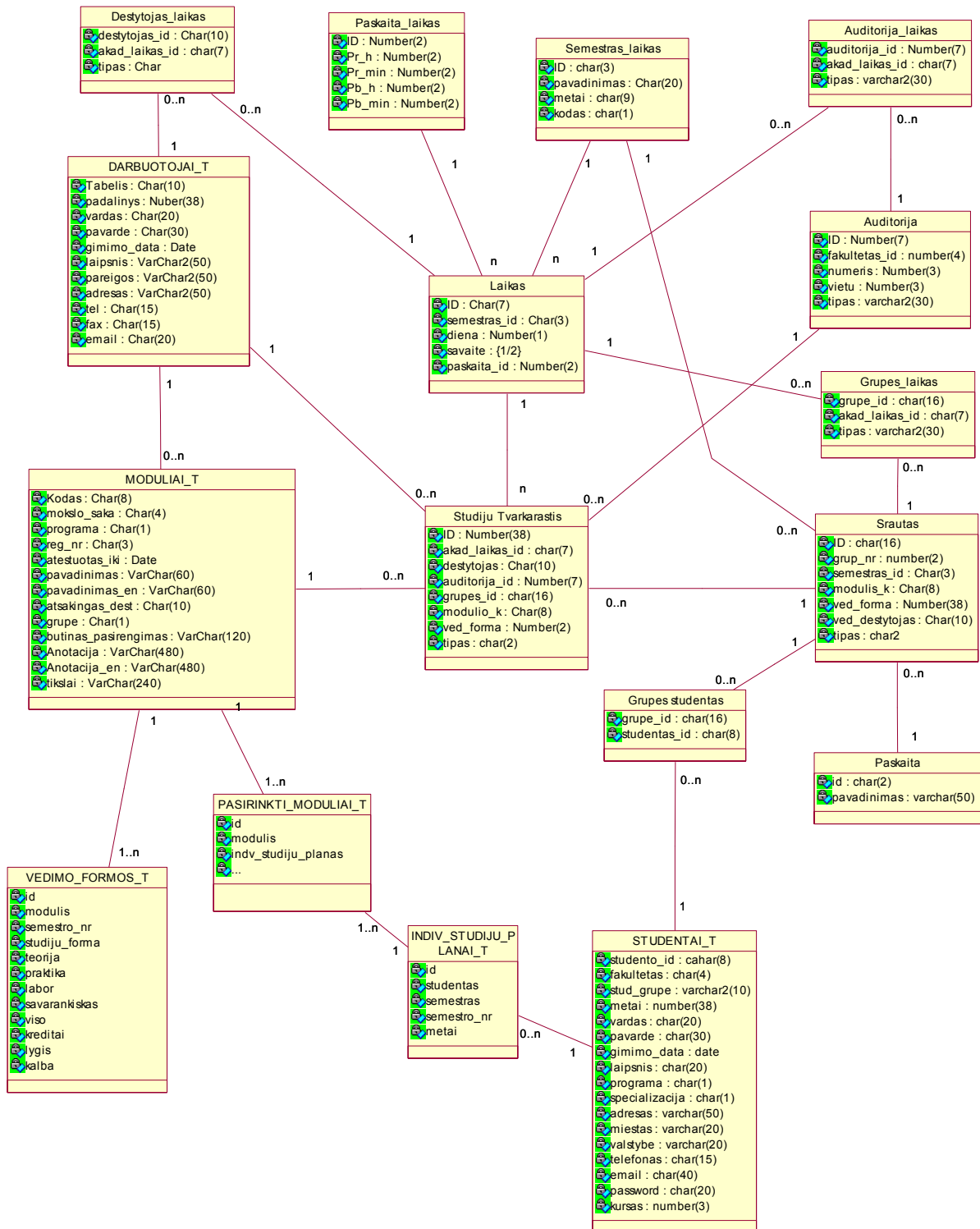
41 pav. Tvarkaraščio kūrimo funkciniai reikalavimai komunikacinėmis kilpomis



42 pav. Tvarkaraščio kūrimo uždavinys

III etapas. Sudaromos klasių diagramos

Dalykinės srities klasių diagrama:



43 pav. Dalykinės srities klasių diagrama

- Egzistuojančios lentelės iš KTU DB:

Klasė **DARBUOTOJAI_T** aprašo KTU darbuotojus (tuos kurie gauna atlyginimą, tokie kaip dėstytojai), projektuojant DB atsižvelgti reikia į tai, kad galima būtų prijungti reikalingus laukus.

Atributai

tabelis - darbuotojo id
padalinys – padalinys, kuriame darbuotojas dirba
vardas – darbuotojo vardas
pavarde – darbuotojo pavardė
gimimo_data – darbuotojo gimimo data
laipsnis - žodžiu parašytas darbuotojo laipsnis
pareigos – darbuotojo pareigos. pvz, vedėjas
adresas – darbuotojo adresas
tel – nurodo darbuotojo telefono numerį
fax – nurodo darbuotojo fakso numerį
email – nurodo darbuotojo elektroninio pašto adresą

Klasė **MODULIAI_T** aprašo dėstomus modulius, pvz.. Informacinės sistemos

Atributai

kodas – nurodomas modulio kodas;
mokslo šaka - nurodoma mokslo šaka
programa – nurodoma modulio programa
reg_nr – nurodomas modulio registracijos numeris
atestuotas_iki – nurodoma data iki kada modulis atestuotas
pavadinimas – nurodo modulio pavadinimą
pavadinimas_en – nurodo modulio pavadinimą anglų kalba
atsakingas_dest - nurodo už modulį atsakingo dėstytojo kodą
grupė – nurodo modulio grupę
butinas_pasirengimas – aprašo moduliui būtina pasirengimą
anotacija – nurodo informaciją apie modulį (anotaciją)
anotacija_en – nurodo informaciją apie modulį anglų kalba
tikslai – nurodo modulio tikslus
atsiskaitymai – nurodo atsiskaitymus už modulį

Klasė **INDV_STUDIJU_PLANAI_T** aprašo studento individualius studijų planus

Atributai

id - nurodomas individualaus studijų plano id
studentas – nurodo studento pažymėjimo numerį
semestras - nurodo semetrą
semestro_nr – nurodo semestro numerį
metai – nurodo metus

Klasė **STUDENTAI_T** aprašo informaciją apie semestrus

Atributai

Studento_id - nurodomas studento pažymėjimo numeris
fakultetas – nurodo fakultetą, kuriame mokosi studentas
stud_grupe - nurodo studento grupę
metai – nurodo studento įstojimo metus
vardas – nurodo studento vardą
pavarde – nurodo studento pavardę

gimimo_data – nurodo studento gimimo datą
laipsnis – nurodo studento laipsnį
programa - nurodomas studento programą
specializacija – nurodo studento specializaciją
adresas - nurodo studento adresą
miestas – nurodo miestą, iš kurio yra studentas
valstybe – nurodo valstybę, kurioje gimė studentas
telefonas – nurodo studento telefono numerį
email – nurodo studento elektroninio pašto adresą

- Naujai sukurtos lentelės:

Klasė **Paskaita_laikas** aprašo paskaitų laikus, pvz.. pirma paskaita 8:00 – 9:45.

Atributai

id – paskaitos laiko identifikatorius;
Pr_h – paskaitos pradžios valanda;
Pr_min – paskaitos pradžios minutės;
Pb_h – paskaitos pabaigos valanda;
Pb_min – paskaitos pabaigos minutės.

Klasė **Semestras_laikas** aprašo akademinis semestrus, pvz.. 2003/2004 – pavasario semestras.

Atributai

id – semestro identifikatorius;
Pavadinimas – semestro pavadinimas;
Kodas – pavadinimo kodas (pirma raidė R, P, V);
Metai – semestro metai.

Klasė **Laikas** aprašo akademinis laikus.

Atributai

id – akademinio laiko identifikatorius;
Semestras_id – semestro identifikatorius;
Savaite – paskaitų savaitė (1,2);
Diena – savaitės diena;
Paskaita_id – paskaitos identifikatorius.

Klasė **Auditorija** aprašo informaciją apie auditorijas ir klases.

Atributai

id – auditorijos identifikatorius;
Fakultetas_id – fakulteto identifikatorius;
Numeris – auditorijos, klasės kabineto numeris;
Vietu – skaičius, kiek auditorijoje, klasėje telpa studentų;
Tipas – auditorijos, klasės tipas.

Klasė **Auditorija_laikas** auditorijos užimtumo laiko fondas.

Atributai

Auditorijos_id – auditorijos identifikatorius;
Akad_laikas_id – akademinio laiko identifikatorius;
Tipas – užimtumo tipas;

Klasė **Paskaita** paskaitų tipai: TH teorija, PR praktika, LA laboratoriniai darbai.

Atributai

id – paskaitos tipo identifikatorius;
Pavadinimas – paskaitos tipo pavadinimas;

Klasė **Srautas** paskaitų grupės, sudaromos paskaitų tvarkaraščiui sudaryti.

Atributai

id – paskaitų grupės identifikatorius;
Grup_nr – grupės numeris;
Semestras_id – semestro identifikatorius;
Modulis_k – grupės modulio kodas;
Ved_forma – grupės modulio vedimo forma.
Ved_destytojas – grupės paskaitų vedantysis dėstytojas.
Tipas – paskaitos tipas.

Klasė **Grupes studentai** paskaitų grupių studentų sąrašas.

Atributai

Grupe_id – paskaitų grupės identifikatorius;
Studentas_id – studento identifikatorius.

Klasė **Grupes laikas** auditorijos užimtumo laiko fondas.

Atributai

Grupe_id – paskaitų grupės identifikatorius;
Akad_laikas_id – akademinio laiko identifikatorius;
Tipas – užimtumo tipas.

Klasė **Destytojas laikas** dėstytojo užimtumo laiko fondas.

Atributai

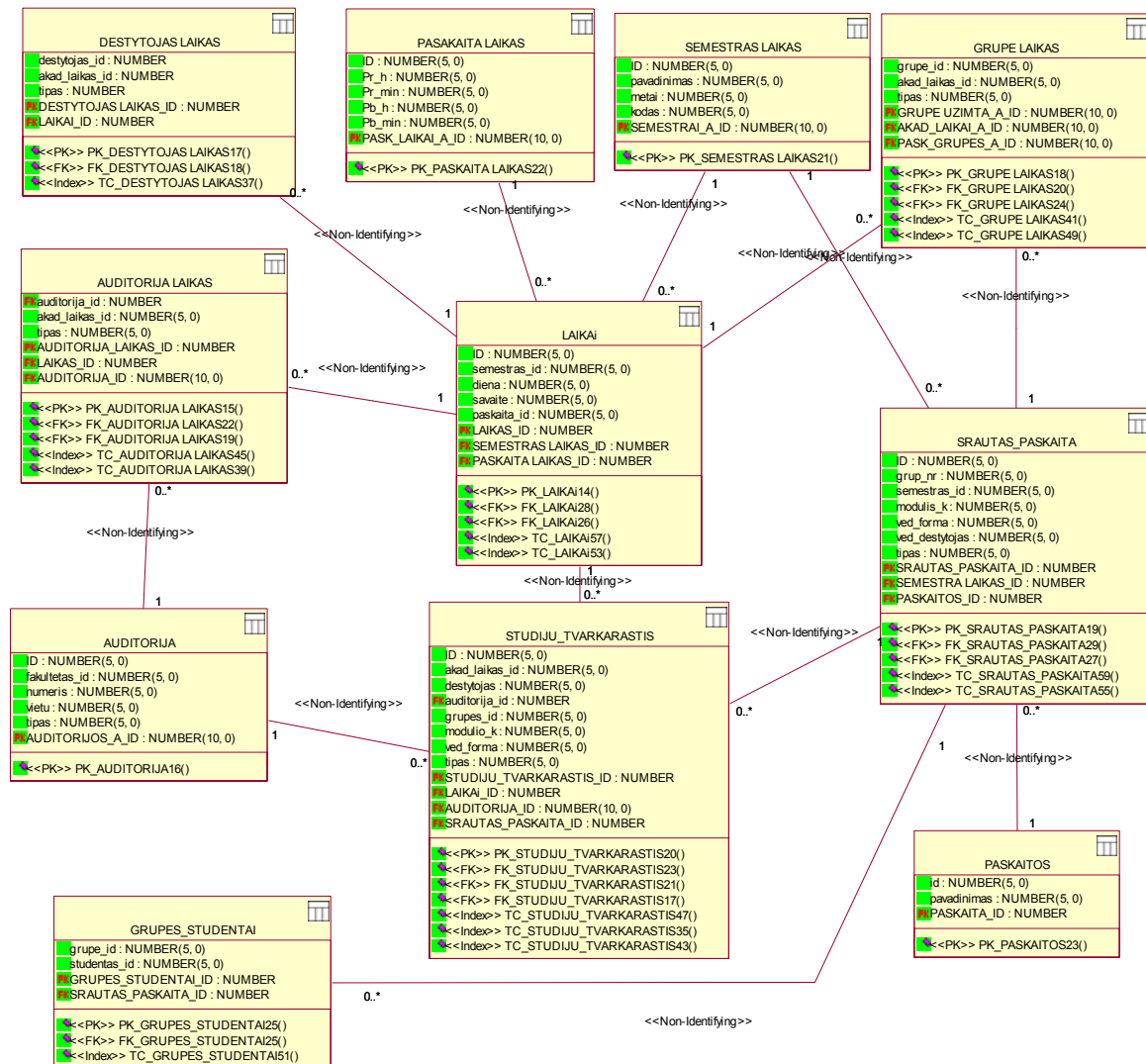
Dėstytojas_id – dėstytojo tabelio numeris;
Akad_laikas_id – akademinio laiko identifikatorius;
Tipas – užimtumo tipas.

Klasė **Studiju Tvarkarastis** sudarinėjamo tvarkaraščio paskaitos.

Atributai

id – paskaitų tvarkaraščio identifikatorius;
Akad_laikas_id – akademinio laiko identifikatorius;
Auditorija_id – auditorijos identifikatorius;
Grupe_id – paskaitų grupės identifikatorius;
Modulis_k – paskaitos modulio kodas;
Forma – paskaitos modulio vedimo forma.
Tipas – paskaitos tipas.

Duomenų bazės diagrama:



44 pav. Duomenų bazės diagrama

7. Sistemos realizacija

Prisijungus prie sistemos vartotojas atsiduria pagrindiniame meniu.



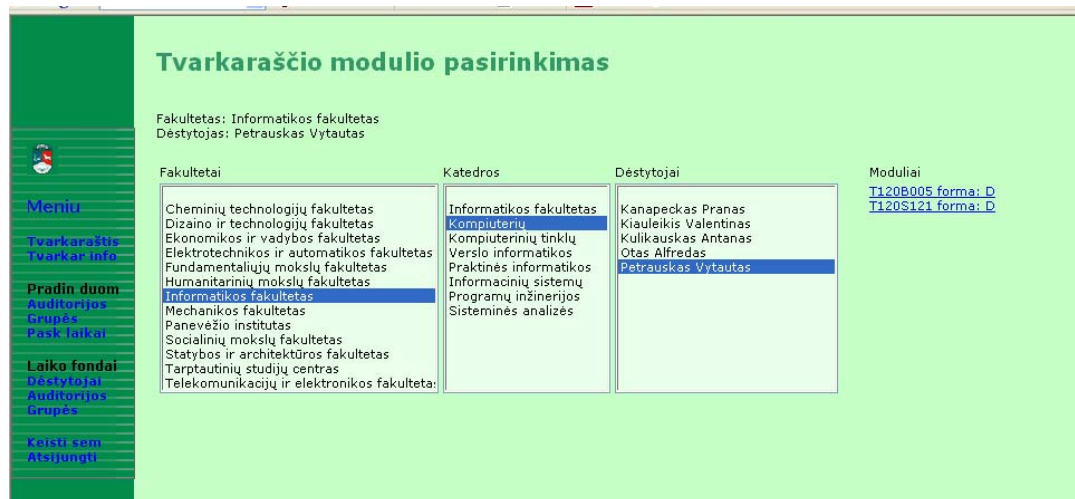
45 pav. Pagrindinio meniu puslapis

Šiame puslapyje galimi tokie pasirinkimai:

1. Tvarkaraštis – paskaitų tvarkaraščio pildymas;
 2. Tvarkaraštis info – informacija apie tvarkaraščio pildymo progresą;
- Pradiniai duomenys:
3. Auditorijos – auditorijų sąrašas, naujų auditorijų įvedimas;
 4. Grupės – paskaitų grupių sudarymas;
 5. Paskaitų laikai – paskaitų laikų redagavimas;
- Laiko fondai:
6. Dėstytojai – dėstytojo laiko fondo pildymas;
 7. Auditorijos – auditorijos laiko fondo pildymas;
 8. Grupės – grupės laiko fondo pildymas;
 9. Pasirinkti sem – semestro keitimas;
 10. Atsijungti – atsijungimas nuo sistemos.

7.1 Pasirinkimas Tvarkaraštis

Iš „Meniu“ puslapio pasirinkus „Tvarkaraštis“ nuorodą užkraunamas „Tvarkaraščio modulio pasirinkimas“ puslapis, kuriame pasirenkame modulį ir vedimo formą, kuriam įvedinėsimė paskaitas.



46 pav. Tvarkaraščio modulio pasirinkimo puslapis

Atsidariusiame puslapyje iš fakultetų sąrašo pasirenkame mus dominantį fakultetą. Išsirinkus fakultetą atsiranda fakultetui priklausančių katedrų ir modulių (ir jų vedimo formų), dėstomų pasirinkto fakulteto dėstytojų sąrašai. Pasirinkus katedrą atsiranda dėstytojų, kurie dirba pasirinktoje katedroje, sąrašas ir atsinaujina modulių sąrašas, dabar matome tik tuos modulius, kuriuos dėsto pasirinktos katedros dėstytojai. Pasirinkus pageidaujamą dėstytoją, dešinėje lango pusėje matome modulius, kuriuos dėsto pasirinktas dėstytojas. Išsirinkus pageidaujamą modulį, kuriam norime sudaryti paskaitų tvarkaraštį, sistema atidaro „Tvarkaraštis pildyti“ puslapį.



47 pav. Tvarkaraščio pildymo puslapis

„Tvarkaraštis pildyti“ - pagrindinis sistemos puslapis, naudojamas tvarkaraščio paskaitoms suvedinėti ir redaguoti. Šiame puslapyje išvedama lentelė, kurios eilutės yra paskaitų dienos ir laikai, o stulpeliai – studentų grupės, pasirinkusios šį modulį. Eilučių ir stulpelių susikirtimuose įvedama paskaitos informacija. Laukų grupė, laikas, vedantysis dėstytojas suvedinėti nereikia, šią informaciją paimama iš lentelės eilučių ir stulpelių, raštinės darbuotojui tereikia įvesti auditorijos identifikatorių, jei identifikatoriaus nežinome, puslapyje yra nuorodos į fakultetų ir auditorijų sąrašus. Suvedus paskaitas spaudžiame mygtuką „Išsaugoti“. Įvestos paskaitos įvedamos į duomenų bazę. Kad vartotojas nesuvestų blogai paskaitų (pvz. tas pats dėstytojas tuo pačiu metu veda dvi skirtingas paskaitas), įvestas ląstelių spalvinimas spalvomis, kas reiškia perspėjimą arba galimą klaidą.

Pilka - Grupės paskaita.

Mėlyna - Klaida, du ar daugiau modulių vyksta toje pačioje auditorijoje tuo pačiu laiku.

Žydra - Klaida, auditorija užimta ir negali joje vykti paskaitos.

Žalia - Perspėjimas, dėstytojas tuo laiku jau turi paskaitą.

Raudona - Klaida, dėstytojas dėl asmeninių priežasčių užimtas ir negali vesti paskaitos.

7.2 Pasirinkimas Tvarkaraštis info

Iš pagrindinio meniu arba „Tvarkaraštis pildyti“ puslapio paspaudus nuorodą „Tvarkaraštis info“ keliaujame į puslapį kuriame galime pažiūrėti informaciją apie dėstytojo modulį ir studentų grupių paskaitas. Parodo kokios grupės kiek, kur ir kada vyksta paskaitos. Labai svarbus puslapis raštinės darbuotojui peržiūrėti tvarkaraščio pilnumą ir vientisumą, ieškant klaidų, žiūrėti kaip paskaitos pasiskirsčiusios tarp grupių.

Planas
Auditorijos

Fakultetas:
Informatikos fė

Katedra:
Kompiuterių

Dėstytojai:
Petrauskas

T120B005 :D
T120S121 :D

Tvarkaraščio informacija

Semestras: 2002/2003 Pavasario semestras
Fakultetas: 7
Katedra: 1401
Dėstytojas: 1493

[Fakultetu sąrašas](#)
[Auditorijų sąrašas](#)
[Atnaujinti](#)

Spalvos:
Raudona - per daug priskirta paskaitų, viršijamas akademinis valandų norma.
Mėlyna - per mažai priskirta paskaitų.

Modulis, atsakingas dėstytojas, vedimo forma	Grupės, vedantis dėstytojas	Paskaitos laikas, auditorija	
T120B005: D Petrauskas Vytautas TH teorija: 64 val. PR praktika: 32 val. LA laboratoriniai d.: 16 val. SA Savarankiškas d.: 32 val.	Grupė: TH1: 96 val. Dėstytojas: Petrauskas Vytautas	Sav:1 diena:1 laikas:08:00-09:45, auditorija: 7103 Sav:1 diena:1 laikas:10:00-11:45, auditorija: 7103 Sav:1 diena:3 laikas:10:00-11:45, auditorija: 7103 Sav:2 diena:1 laikas:08:00-09:45, auditorija: 7103 Sav:2 diena:3 laikas:10:00-11:45, auditorija: 7103 Sav:2 diena:5 laikas:10:00-11:45, auditorija: 7103	
	Grupė: TH2: 64 val. Dėstytojas: Petrauskas Vytautas	Sav:1 diena:1 laikas:08:00-09:45, auditorija: 7103 Sav:1 diena:3 laikas:10:00-11:45, auditorija: 7103 Sav:2 diena:3 laikas:10:00-11:45, auditorija: 7103 Sav:2 diena:5 laikas:10:00-11:45, auditorija: 7103	
	Grupė: PR1: 32 val. Dėstytojas: Petrauskas Vytautas	Sav:1 diena:2 laikas:10:00-11:45, auditorija: 7523 Sav:2 diena:2 laikas:10:00-11:45, auditorija: 7523	
	Grupė: PR2: 32 val. Dėstytojas: Petrauskas Vytautas	Sav:1 diena:2 laikas:12:30-14:00, auditorija: 7523 Sav:2 diena:2 laikas:12:30-14:00, auditorija: 7523	
	Grupė: LA1: 48 val. Dėstytojas: Petrauskas Vytautas	Sav:1 diena:1 laikas:08:00-09:45, auditorija: 7103 Sav:2 diena:1 laikas:08:00-09:45, auditorija: 7103 Sav:2 diena:4 laikas:08:00-09:45, auditorija: 7201	
	Grupė: LA2: 16 val. Dėstytojas: Petrauskas Vytautas	Sav:1 diena:4 laikas:08:00-09:45, auditorija: 7201	
	Grupė: LA3: 16 val. Dėstytojas: Petrauskas Vytautas	Sav:1 diena:4 laikas:08:00-09:45, auditorija: 7201	
	T120S121: D Petrauskas Vytautas TH teorija: 64 val. PR praktika: 32 val. LA laboratoriniai d.: 16 val. SA Savarankiškas d.: 32 val.	Grupė: TH1: 0 val. Dėstytojas: Paradauskas Bronius	Nėra
		Grupė: PR1: 16 val. Dėstytojas: Petrauskas Vytautas	Sav:2 diena:1 laikas:10:00-11:45, auditorija: 7312

48 pav. Tvarkaraščio informacijos puslapis

Atkreipti vartotojo dėmesį apie neteisingus studentų grupių turimus paskaitų valandų kiekius, naudojame spalvinius pažymėjimus:

Raudona - per daug priskirta paskaitų, viršijamas akademinis valandų norma.

Mėlyna - per mažai priskirta paskaitų.

7.3 Pasirinkimas Auditorijos (Pradinių domenu įvedimas)

„Auditorijos“ puslapyje galima peržiūrėti auditorijų sąrašą ir informaciją, tokią kaip kiek auditorijoje telpa studentų, koks auditorijos tipas, kokiam fakultetui priklauso. Auditorijų filtravimas atliekamas pagal fakultetą ir auditorijos tipą. Šiame puslapyje galima atlikti auditorijų įvedimą ir šalinimą. Paspaudus šalia auditorijos ant nuorodos „laiko fondas“ pereinama į puslapį „Auditorija užimta“.

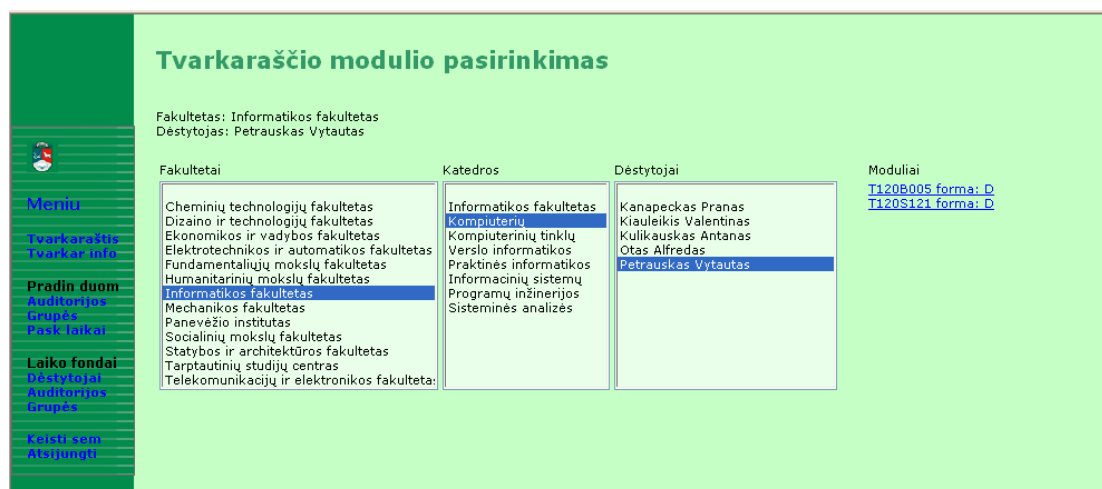


49 pav. Auditorijų puslapis

Norint įvesti naują auditoriją, įvedame laukus esančius puslapio apačioje ir spaudžiame mygtuką „Įvesti“. Norint sužinoti fakulteto identifikatorių spaudžiame ant „Fakultetas_id“ nuorodos.

7.4 Pasirinkimas Grupės (Pradinių domenu įvedimas)

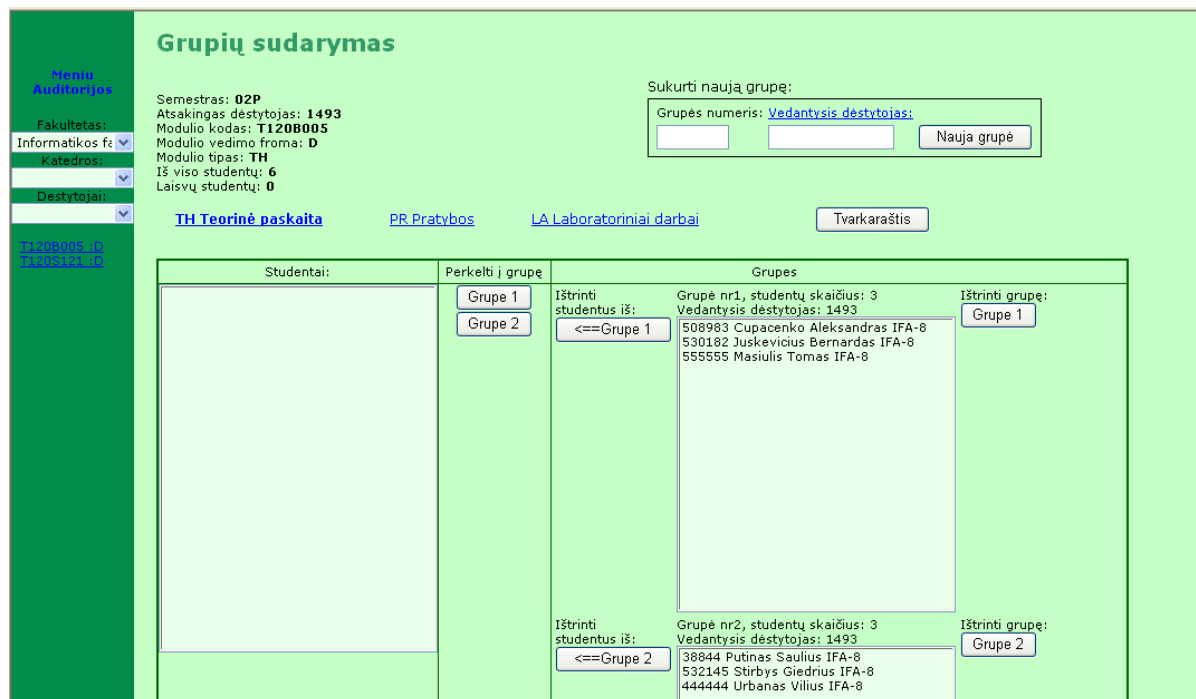
Iš „Meniu“ puslapio pasirinkus „Grupės“ nuorodą užkraunamas „Grupės modulio pasirinkimas“ puslapis, kuriame pasirenkame modulį ir modulio vedimo formą, kuriam kursime paskaitų grupes.



50 pav. Tvarkaraščių modulio pasirinkimo puslapis

Atsidariusiame puslapyje iš fakultetų sąrašo pasirenkame mus dominantį fakultetą. Išsirinkus fakultetą atsiranda fakultetui priklausančių katedrų ir modulių (ir jų vedimo formų),

dėstomų pasirinkto fakulteto dėstytojų sąrašai. Pasirinkus katedrą atsiranda dėstytojų, kurie dirba pasirinktoje katedroje, sąrašas ir atsinaujina modulių sąrašas, dabar matome tik tuos modulius, kuriuos dėsto pasirinktos katedros dėstytojai. Pasirinkus pageidaujamą dėstytoją, dešinėje lango pusėje matome modulius, kuriuos dėsto pasirinktas dėstytojas. Išsirinkus pageidaujamą modulį, kuriam norime sudaryti paskaitų grupes, sistema atidaro „Grupių sudarymas“ puslapį.



51 pav. Grupių sudarymas puslapis

„Grupė sudaryti“ puslapis skirtas studentų grupėms sudaryti ir studentams paskirstyti po grupes. Šiame puslapyje kairėje pusėje išvedamas studentų pasirinkusių modulį sąrašas. Dešinėje pusėje yra grupių studentų sąrašai. Pasirinkus studentus iš kairiojo sąrašo ir paspaudus ant „įvesti į grupę“ mygtuko, studentai iš kairiojo sąrašo perkeliama į dešinę grupės sąrašą. Galimas ir atvirkščias veiksmas. Pažymimi studentai, kuriuos norime pašalinti iš grupės ir spaudžiamas „ištrinti studentus iš Grupe“ mygtukas.

Jei nėra sukurta grupių arba yra, bet per mažai, tai galima padaryti lango viršuje įvedus grupės numerį, vedančiojo dėstytojo tabelio numerį (norint sužinoti dėstytojų tabelio numerius spaudžiame ant nuorodos „Vedantysis dėstytojas“) ir paspaudus mygtuką „Sukurti“.

Norint ištrinti grupę spaudžiame „ištrinti grupę“ mygtuką. Studentai priklausę grupei perkeliama į laisvų studentų sąrašą, esanti kairėje puslapio pusėje.

7.5 Pasirinkimas Paskaitų laikai (Pradiniai duomenys)

Puslapyje “Pask. Laikai” galima peržiūrėti paskaitų laikus, juos redaguoti ir sukurti naujus.

Paskaitų laikai

ID	Pradžia	Pabaiga	Sk
1	08:00	09:45	trinti
2	10:00	11:45	trinti
3	12:30	14:00	trinti
4	14:15	15:45	trinti
5	16:00	17:30	trinti
6	17:45	19:15	trinti

id Paskaitos pradžia Paskaitos pabaiga

id Paskaitos pradžia Paskaitos pabaiga

53 pav. Paskaitų laikų puslapis

Norint redaguoti paskaitos laiką, užpildome laukus esančius po paskaitų laikų sąrašu: nurodome paskaitos laiko identifikatorių, kurį norime pakeisti, įvedame naujus pradžios ir pabaigos laikus ir spaudžiame mygtuką „Keisti“.

Norint įvesti naują paskaitos laiką užpildome laukus esančius puslapio apačioje ir spaudžiame mygtuką „Įvesti“.

Paskaitos pradžios ir pabaigos laikas turi būti skaičiai (valandos nuo 0-23, minutes 0-60).

7.7 Pasirinkimas Dėstytojai (Laiko fondai)

„Meniu“ puslapyje pasirinkus mygtuką „Dėstytojai“, užkraunamas „Dėstytojo pasirinkimas“ puslapis, kuriame pasirenkame dėstytojo tabelį, kuriam norime pildyti užimtumo fondą. Iš fakultetų sąrašo išsirenkame fakultetą, kuriame dėstytojas dirba. Išsirinkus fakultetą atsiranda fakultetui priklausančių katedrų ir dėstytojų, priklausančių pasirinktam fakultetui, sąrašai. Pasirinkus katedrą dėstytojų sąrašė lieka tik dirbantys toje katedroje dėstytojai. Iš dėstytojų sąrašo pasirinkus pageidaujamą dėstytoją sistema atidaro „Dėstytojas užimtas“ puslapį.

Dėstytojo laiko fondas

Meniu
Laiko fondai
 Kadencijos
 Dėstytojas

Dėstytojo tabelis: **1493**
 Dėstytojas: **Petrauskas Vytautas**
 Fakultetas: **Informatikos fakultetas**
 Katedra: **Kompiuterių**
 Laipsnis: **Dėstytojas**
 Pareigos: **Dėstytojas**
 Telefonas:

Fakultetas: Informatikos fakult. [Pilka spalva - Dėstytojas veda paskaitą](#) [Redaguoti užimtumo tipus](#) [Atnaujinti](#)

Pirma savaitė			Antra savaitė		
Diena	Laikas	Užimtumas	Diena	Laikas	Užimtumas
1	08:00 - 09:45	Vyksta paskaita T120B005:D	1	08:00 - 09:45	Vyksta paskaita T120B005:D
	10:00 - 11:45	Vyksta paskaita T120B005:D		10:00 - 11:45	Vyksta paskaita T120S121:D
	12:30 - 14:00			12:30 - 14:00	
	14:15 - 15:45			14:15 - 15:45	
	16:00 - 17:30	Seminaras		16:00 - 17:30	
	17:45 - 19:15	Seminaras		17:45 - 19:15	
2	08:00 - 09:45		2	08:00 - 09:45	
	10:00 - 11:45			10:00 - 11:45	
	12:30 - 14:00			12:30 - 14:00	
	14:15 - 15:45			14:15 - 15:45	
	16:00 - 17:30			16:00 - 17:30	Mokslinė veikla
	17:45 - 19:15			17:45 - 19:15	Mokslinė veikla
3	08:00 - 09:45		3	08:00 - 09:45	
	10:00 - 11:45	Vyksta paskaita T120B005:D		10:00 - 11:45	Vyksta paskaita T120B005:D
	12:30 - 14:00			12:30 - 14:00	
	14:15 - 15:45			14:15 - 15:45	
	16:00 - 17:30			16:00 - 17:30	

54 pav. Dėstytojo laiko fondo puslapis

Šiame puslapyje išvedama lentelė, kurios eilutės yra dienos ir paskaitų laikai, o stulpeliai – paskaitų savaitės. Eilučių ir stulpelių susikirtimuose įvedame kada dėstytojas yra užimtas ir negali vesti paskaitų. Paspaudus mygtuką „Įvesti pakeitimus“ duomenys išsaugomi.

Jei nėra reikiamo užimtumo sąrašė, paspaudus nuorodą „Redaguoti užimtumo tipus“ užkraunamas puslapis, kuriame galima sukurti reikiamą užimtumo tipą. Sukūrus naują užimtumo tipą ir norint jį pamatyti reikia atnaujinti puslapį. Tą galima padaryti paspaudus nuorodą „Atnaujinti“.

Pilka spalva pažymimos dėstytojo paskaitos. Užimtumas, sutampantis su dėstytojo paskaitos laiku, negali būti pasirinktas.

7.8 Pasirinkimas Auditorijos (Laiko fondai)

„Meniu“ puslapyje paspaudus mygtuką „Auditorijos“, užkraunamas „Auditorijos pasirinkimas“ puslapis, kuriame pasirenkame auditorijos identifikatorių, kuriai norime pildyti užimtumo fondą. Auditorijų filtravimas atliekamas pagal du požymius: fakultetą ir auditorijos tipą. Iš auditorijų sąrašo pasirinkus pageidaujamą auditoriją sistema atidaro „Auditorija užimta“ puslapį.

Auditorijos laiko fondas

Auditorijos id **7103**
 Fakultetas **Informatikos fakultetas**
 Numeris **103**
 Vietų skaičius **180**
 Tipas **auditorija**

[Redaguoti užimtumo tipus](#) [Atnaujinti](#)

Pirma savaitė			Antra savaitė		
Diena	Laikas	Užimtumas	Diena	Laikas	Užimtumas
1	08:00 - 09:45	Vyksta paskaita P000B001:D	1	08:00 - 09:45	Vyksta paskaita P000B001:D
	10:00 - 11:45	Vyksta paskaita T120B005:D		10:00 - 11:45	
	12:30 - 14:00			12:30 - 14:00	
	14:15 - 15:45			14:15 - 15:45	
	16:00 - 17:30			16:00 - 17:30	
	17:45 - 19:15			17:45 - 19:15	
2	08:00 - 09:45	Kita priežastis	2	08:00 - 09:45	
	10:00 - 11:45	Kita priežastis		10:00 - 11:45	
	12:30 - 14:00			12:30 - 14:00	
	14:15 - 15:45			14:15 - 15:45	
	16:00 - 17:30			16:00 - 17:30	
	17:45 - 19:15			17:45 - 19:15	
3	08:00 - 09:45		3	08:00 - 09:45	
	10:00 - 11:45	Vyksta paskaita T120B005:D		10:00 - 11:45	Vyksta paskaita T120B005:D
	12:30 - 14:00			12:30 - 14:00	
	14:15 - 15:45			14:15 - 15:45	
	16:00 - 17:30			16:00 - 17:30	
	17:45 - 19:15			17:45 - 19:15	

55 pav. Auditorijų laiko fondo puslapis

Šiame puslapyje išvedama lentelė, kurios eilutės yra dienos ir paskaitų laikai, o stulpeliai – paskaitų savaitės. Eilučių ir stulpelių susikirtimuose įvedame kada auditorija yra užimta ir negali turėti paskaitų. Paspaudus mygtuką „Ivesti pakeitimus“ duomenys išsaugomi.

Jei nėra reikiamo užimtumo sąrašo, paspaudus nuorodą „Redaguoti užimtumo tipus“ užkraunamas puslapis, kuriame galima sukurti reikiamą užimtumo tipą. Sukūrus naują užimtumo tipą ir norint jį pamatyti reikia atnaujinti puslapį. Tą galima padaryti paspaudus nuorodą „Atnaujinti“.

Pilka spalva pažymimos dėstytojo paskaitos. Užimtumas, sutampantis su auditorijos paskaitos laiku, negali būti pasirinktas.

7.8 Pasirinkimas Grupės (Laiko fondai)

Iš „Menu“ puslapio pasirinkus „Grupės“ nuorodą užkraunamas „Grupės pasirinkimas“ puslapis, kuriame pasirenkame modulį ir modulio vedimo formą, kuriam kursime paskaitų grupes.

Žiūrėti 4.12.7 skyrius „Pasirinkimas Grupės“ ir 4.12.7.1 pav. ir po paveikslėliu esantį aprašymą.

Pasirinkus pageidaujimą modulį, kurio grupėms norime pildyti laikų fondus, sistema atidaro „Grupė užimta“ puslapį.

56 pav. Grupių laiko fondo puslapis

Puslapyje pildomi grupių laikų fondai. Šiame puslapyje išvedama lentelė, kurios eilutės yra dienos ir paskaitų laikai, o stulpeliai – paskaitų savaitės. Eilučių ir stulpelių susikirtimuose įvedame kada grupė yra užimta ir negali turėti paskaitų. Paspaudus mygtuką „Įvesti pakeitimus“ duomenys išsaugomi duomenų bazėje.

Jei nėra reikiamo užimtumo sąrašė, paspaudus nuorodą „Redaguoti užimtumo tipus“ užkraunamas puslapis, kuriame galima sukurti reikiamą užimtumo tipą. Sukūrus naują užimtumo tipą ir norint jį pamatyti reikia atnaujinti puslapį. Tą galima padaryti paspaudus nuorodą „Atnaujinti“.

Pilka spalva pažymimos grupių paskaitos. Užimtumas, sutampantis su grupės paskaitos laiku, negali būti pasirinktas.

8. Išvados

1. Išanalizuota veiklos taisyklių koncepcija ir šiuolaikinės jos pritaikymo galimybės. Tinkamas veiklos taisyklių modeliavimas gali patobulinti informacijos sistemų modeliavimo bei projektavimo metodus ir pagerinti realizuotos sistemos funkcionavimą.
2. Taisyklių pagalba galima formaliai aprašyti operacijų semantiką, valdyti verslo tranzakcijas. Šiuolaikinėse sistemose reikalingas taisyklių palaikymo mechanizmas, kuris leistų saugoti taisykles kaip pastovius objektus, jas modifikuoti, atsekti jas ten, kur jos įtrauktos į programų kodus. Veiklos taisyklių pritaikymo sritis yra dar nauja, todėl naujų sprendimų paieška ir esančių tobulinimas yra aktualūs. Veiklos taisyklių principais veikiančios IS ar šiuos principus taikančios IS projektavimo priemonės nėra susilaukusios tokio pripažinimo kaip tradiciniai IS organizavimo sprendimai.
3. Apžvelgti žinomi VT struktūrizavimo modeliai bei smulkiai išanalizuoti ryškiausi sprendimai – GUIDE projektas, Ronaldo Roso metodas ir projektuotojo požiūriu paremtas klasifikavimo būdas.
4. Atlikus lyginamąją šių modelių analizę, nustatyta, kad kiekvienas iš jų gerai apibrėžia tam tikrą taisyklių struktūrizavimo proceso aspektą, nedetalizuoja viso VT modeliavimo proceso.
5. Atsižvelgiant į analizės metu išskirtus žinomų VT struktūrizavimo modelių trūkumus, pasiūlytas konceptualus VT struktūrizavimo proceso modelis. Modelis sudarytas tipinės projektavimo schemos pagrindu. Modelio pagrindu pasirinktas projektavimo požiūriu paremtas VT klasifikavimo metodas, nes jis turi pagrindines veiklos taisyklių klasifikavimo klases, apibrėžtos būtent tos klasės kurios efektyviai naudojamos projektavimo metu, yra perspektyviausias iš analizės metu nagrinėtų struktūrizavimo modelių.
6. Šiame darbe atliktas konkrečių modeliavimo kalbų tyrimas parodė, kad duomenų pateikimui galima naudoti paprastesnes priemones, pavyzdžiui, SQL užklausas ir standartines funkcijas, vaizduoti veiklos taisykles duomenų bazių schemomis.
7. Pasiūlyta modeliavimo metodika naudinga informacinių sistemų projektuotojams ir programuotojams, kurie naudos pateiktą metodiką ir tyrimo išvadas projektuodami kitus uždavinius, kuriantiems įvairias informacines sistemas, taip pat jas modifikuojant, adaptuojant prie pasikeitusių sąlygų.

8. Universiteto tvarkaraščio sudarymo dalykinės srities pagrindu, buvo vykdomas eksperimentas, kurio metu bus išskirti struktūrizuotos veiklos taisyklės, remiantis pasirinkto projektuotojo požiūriu paremtu VT klasifikavimo metodu. Taip pat buvo vaizduojamos duomenų bazių loginės struktūros ir realizuojamos programiškai veiklos taisyklės,. Sukurtoje sistemoje galima peržiūrėti ir įvesti sudėtingų struktūrų duomenis, sudaryti fakulteto tvarkaraščius, registruoti laiko fondus bei peržiūrėti tvarkaraščius.

Literatūros šaltiniai

- [Butleris] Butleris R., Gruzdis G., Veiklos taisyklių specifikuojimas: modeliavimo metodų apžvalga, „Informacinės technologijos 1999“, Kaunas, Technologija, 1999, p. 116-121
- [Demuth] Demuth B, Hussmann, OCL as a specification language for business rules in database, m Gogolla red.:UML 2001, Springer-Verlang, p.104-117
- [Gottesdiener] Gottesdiener E.,Business Rules, Show power promise, 1997
http://www.ebgconsulting.com/powerpromise_article.htm
- [Jacobson] Jacobson I., Booch G., Rumbaugh J. - The Unified Modelling Language Reference Manual, Addison-Wesley, 2001
- [Jeleva] Jeleva L.,CDM RuleFrame Speaking CDM RuleSLang, „ODTUG 2000 konferencijų medžiaga“, 2000, <http://www.odtug.com>
- [Kapočius 2002] Kapočius K.,Butleris R., Veiklos taisyklių struktūrizavimo projektuojant informacijos sistemas modelių analizė, „Informacinės technologijos 2002“, Kaunas, Technologija, 2002, p. 298-304
- [Kapočius 2001] Kapočius K.,Butleris R., Struktūrizuotų veiklos taisyklių saugyklos architektūra, „Informacijos mokslai 2001“, Vilniaus universitetas, 2001, p. 46-56
- [Kolber] Kolber A., Hay D. ir kiti.GUIDE Business Rules Project, Final Report r. 1.2. GUIDE International Corporation, 1997, URL.: <http://www.guide.org/ap/apbrules>.
- [Motiejunas] Motiejunas L., R.Butleris, K.Kapocius, Veiklos taisykliu specifikuojimo ir manipuliavimo modelis, „Informacinės technologijos verslui-2002“, Kaunas, Technologija, 2002, 24-30 psl.
- [Nemuraitė] Nemuraitė L., Ažubalis K., Veiklos taisyklių modeliavimas objektiškai orientuotose metoduose, „Informacinės technologijos 1999“, Kaunas, Technologija, 1999, p. 171-178
- [Plotkin] Plotkin D., Business Rule Everywhere, 1999,
http://www.intelligententerprise.com/db_area/archives/1999/990903/feat2.shtml
- [Paradauskas] Paradauskas B., Nemuraitė L., Duomenų semantiniai modeliai, Kaunas, Technologija, 2002, 260 psl.
- [RationalSoftwareCorp]
RationalSoftwareCorp.,http://www.rational.com/media/uml/resources/media/ad970808_uml11_0cl.pdf, 2002
- [Ross] Ross R.G., The Business Rule Book: Classifying, Defining and Modeling Rules, Business Rule Solution, Second Edition. Houston, 1997

[OMG] Object Management Group. Business Rules in Models.2002, URL.:
<http://cgi.omg.org/cgi-bin/doc>

[OMG 2003] Object Management Group. UML 2.0 OCL Specification.pdf, 2003

Priedas

Drąsutavičiūtė G. Veiklos taisyklių vaizdavimas duomenų bazių struktūrų schemomis. Informacinė visuomenė ir universitetinės studijos '2004, Konferencijos pranešimų medžiaga, Kaunas, 2004, (spaudoje).

VEIKLOS TAISYKLIŲ VAIZDAVIMAS DUOMENŲ BAZIŲ STRUKTŪRŲ SCHEMOMIS

Giedrė Drašutavičiūtė

*Kauno technologijos universitetas
Informacijos sistemų katedra*

Labai svarbu kuriant organizacijos IS aiškiai suprasti reikalavimų nustatymo etape, veiklos procesus, jų keliamus apribojimus. Kuriant IS viena iš problemų yra veiklos taisyklių stuktūrizavimas. Straipsnyje plačiau analizuojama veiklos taisyklių klasifikacija ir vaizdavimo būdai duomenų bazių modeliavimo ir projektavimo metuose, išskiriami jų privalumai ir trūkumai. Remiantis atlikta analize pasiūlytas veiklos taisyklių projektavimo proceso modelis, nurodomos jo vystymo galimybės.

Įvadas

Pastaruoju metu niekas neabejoja žinių panaudojimo naudingumu IS kūrimo etape. Labai svarbus uždavinys šiuolaikiniame IS kūrimo etape yra tiksliai ir aiškiai nustatyti vartotojo poreikius, surinktas žinias kaupti ir saugoti. Augant vartotojų poreikiams, didėjant informacijos apdorojimo poreikiui dinamiškai kintančioje aplinkoje, tradiciniai IS projektavimo metodai nėra pakankamai efektyvūs. Todėl šiuo metu labai daug dėmesio skiriama verslo modeliavimui įvairiais požiūriais, siekiant sukurti vieningą sistemą. Modeliuojant verslo informacinę sistemą būtina atkreipti dėmesį, kad verslo taisyklės paprastai keičiasi greičiau, negu probleminės srities duomenų struktūros ar objektų elgsena. Dažnai reikia verslo taisyklių rinkinį papildyti, peržiūrėti, keisti.

Modeliavimas verslo taisyklių požiūriu yra naujas modeliavimo būdas, paremtas verslo taisyklių išskyrimu. Kuriant IS viena iš problemų yra veiklos taisyklių stuktūrizavimas. Labai svarbu aiškiai suprasti reikalavimų nustatymo etape veiklos procesus, jų keliamus apribojimus, kad toliau galėtume išskirti tik tas veiklos taisykles, kurias reikia programiškai realizuoti.

Veiklos taisyklės

Skirtinguose šaltiniuose veiklos taisyklė yra apibrėžiama panašiai. Ir GUIDE[1] projekto autoriai ir OMG [2] grupės veiklos taisykles (VT) apibrėžia dviem aspektais:

- Veiklos taisyklė – tai verslo požiūriu yra direktyva, skirta įtakoti ar valdyti verslo elgseną, tuo būdu realizuojant verslo politiką.
- IS požiūriu verslo taisyklė yra teiginys, kuris nusako ar apriboja, kurį nors verslo aspektą, deklaruoja verslo struktūrą, kontroliuoja arba kitaip įtakoja verslo procesus.

Darbe [3] taip pat pateikiamas veiklos taisyklės apibrėžimas, jų skaidymo būdas. Kadangi veiklos taisyklė nusakyta veiklos žmogaus, dažniausiai įvardijama dviprasmiškai ir negriežtai, todėl dažnai veiklos taisykles galima išskaidyti į detalesnius veiklos atvejus. Išskaidomos veiklos taisykles iki savo elementariosios formos, jos dalomos tol, kol apsiribojama viena išbaigta mintimi.

Reikalavimai veiklos taisyklėms

Norint geriau suvokti veiklos taisyklių esmę, reikia gerai žinoti jiems keliamus reikalavimus. Dažniausiai šie reikalavimai būna kaip rinkinys pateikiamas kartu su veiklos taisyklių klasifikavimo modeliais, tačiau yra išskiriami ir bendri reikalavimai bendri visoms veiklos taisyklėms[4]. Taisyklės turi būti glaustos ir pateiktos tokia forma, kad užtikrinti formalizavimo ir taisyklių saugyklos sudarymo vientisumą.

- Deklaratyvios – pažymėtina, kad veiklos taisyklės yra labiau deklaratyvios, nei procedūrinės prigimties.
- Tikslī išraiška – veiklos taisyklę reikia išreikšti tiksliai: grafiškai arba formalia kalba. Struktūrinės taisyklės gali būti išreikštos esybių ryšių (ER) diagrama. Įvykių rezultatai gali būti išreikšti duomenų srautu diagramomis. Tikslī išraiška, naudojant grafinę notaciją yra gerai parodytas Rosso metode [5].
- Atominė – logiškai nedali. VT turi būti nedaloma t.y. jei mes veiklos taisyklę skaidome galime prarasti informaciją.
- Nuosekli – negali būti konfliktuojančių taisyklių.

- Neperteklinė – veiklos taisyklių aibėje negali būti tą pačią informaciją konstatuojančių taisyklių.
- Orientuotos į veiklą – veiklos taisyklė privalo būti formuluojama naudojant terminus, suprantamus konkrečios veiklos srities atstovams.
- Priklausomos veiklai – veiklos taisyklės formuluoti, keisti, paskelbti negaliojančiomis gali tik veiklos atstovai.

Visais atvejais veiklos taisyklės turi būti griežtos, nes jų pagrindu generuojamas kodas[6]. Veiklos taisyklės išreikštos deklaratyvia forma pvz.: „Apmokėti tiekėjo važtaraštį tik tuomet, jei jis buvo patvirtintas“, reikia sugebėti jas išreikšti ir natūralia kalba. Griežtai išreikšta anksčiau minėta taisyklė atrodo taip: Apmokėti *Tiekėjo* Važtaraštį tik tada, jeigu jo statusas yra „*Patvirtintas*“

Veiklos taisyklių klasifikavimo metodai

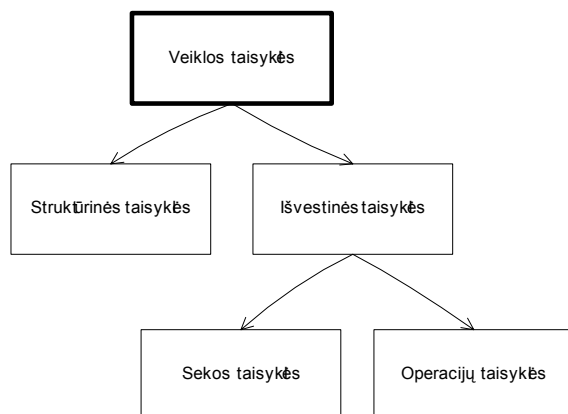
Nuo pat veiklos taisyklių koncepcijos atsiradimo pradžios buvo siūloma įvairūs VT klasifikavimo ir struktūrizavimo modeliai. Klasifikavimo modelių sąrašas yra pateiktas 1 lentelėje[3].

1 lentelė VT klasifikacijos sąrašas

Klasifikavimo metodas	Trumpas taisyklių klasifikacijos aprašymas
Business Rules Group (GUIDE Business Rules Project)	Struktūrinės taisyklės (terminai, faktai), veiksmo taisyklės (integralumo apribojimai, sąlygos, autorizacija), išvestinės taisyklės (skaičiavimai, loginės išvados)
Ronald Ross	Nedalomos (ryšių tipai, subtipai) ir išvestinės
Tom Romeo	Struktūrinės taisyklės (ryšiai, kardinalumas), elgesio taisyklės (prieš sąlygos, po sąlygos, išvestys)
Margaret Thorpe	Apibrėžimai, integralumo taisyklės, bendrieji deklaratyvūs apribojimai, procedūriniai apribojimai, išvestys
Barbara von Halle	Apibrėžimai, faktai, apribojimai, išvestys, loginės išvados
Usoft Corporation	Apribojimai išvestys, elgesio taisyklės, atvaizdavimo taisyklės
Vision Software	Validavimo taisyklės, išvestys, ryšiai, sąlygos veiksmams

Pastebime, jog dauguma tiek ir didelės kompanijos, kuriančios IS kūrimo įrankius (Usoft Corporation, Vision Software), tiek ir nepriklausomi tyrinėtojai, tokie kaip Barbara von Halle, Margaret Thorpe pasiūlyti metodai pasižymi nepastovumu bei susiaurinta panaudojimo sritimi. Bet nepaisant to, yra sukurti metodai, galintys atitikti keliamus veiklos taisyklių reikalavimus. Tai VT klasifikavimo modeliai tokie, kaip mokslininkų pasiūlytas GUIDE projektas, Microsoft kompanijos bendras darbas su Meta Data Coalition sukurtas modelis, taip pat pažymėtinas yra Ronaldo Rosso metodas.

Kadangi labai svarbus uždavinys yra veiklos taisyklių klasifikavimas, siūloma veiklos taisyklių klasifikavimo būdas, kai visos veiklos taisyklės suskirstomos į dvi stambias grupes. Struktūrinės taisyklės – šios taisyklės apibrėžia duomenų struktūras. Veiksmo taisyklės – apibrėžia operacijas ir sekas, kurios turi būti įvykdytos. VT klasifikavimo būdas išdėstytas 1 paveiksle.



1 pav. Siūlomas klasifikavimo būdas

Veiklos taisyklių modeliavimo metodai

UML ir OCL

Šiuo metu daugelis informacinių sistemų projektavimo įrankių naudoja UML (Unified Modeling Language), kurioje yra specialiai taisyklių aprašymui sukurta kalba – OCL (Object Constraint Language). OCL – yra formali logikos pagrindu paremta kalba, neturinti grafinės notacijos[7].

Praktiniai realizacijai naudojamas SQL rodinių bei trigerių šablonų(templates) generavimas iš OCL kalba išreikštų apribojimų. Detaliau yra aprašytas [8].

Pagrindinis pasiūlyto būdo elementas yra SQL rodinių generavimas pagal OCL kalbą parašytą sąlygą. Tikrinant rodinio reikšmes galime realizuoti toliau reikalingus veiksmus. Yra siūlomi trys būdai:

1. Rodinio tikrinimas iš taikomosios programos. Rodinio reikšmių tikrinimas nesiejamas su duomenų bazės vientisumo ir nuoseklumo tikrinimo mechanizmais, tokiais kaip trigeriai. Programos kode turi būti parašyta rodinio tikrinimo sąlyga. Tačiau toks būdas netenkina sąlygos, kad VT rinkinio realizacija turi būti atskirta nuo kitų IS komponentų.

2. Tvirtinimų pakeitimas. Trigeriai, kurie naudojami įvertinant duomenų neprieštarumą bei korektiškumą, stabdo programos vykdymą. Toks trigeris yra aktyvizuojamas po kurio nors duomenų atnaujinimo operacijos. Tačiau tokiu būdu realizuojant, jei rodinyje yra duomenų, turi būti atliekamas veiksmas - klaidos pranešimas. Dėl to tik maža dalis VT gali būti suformuluota ir realizuota tokiu būdu.

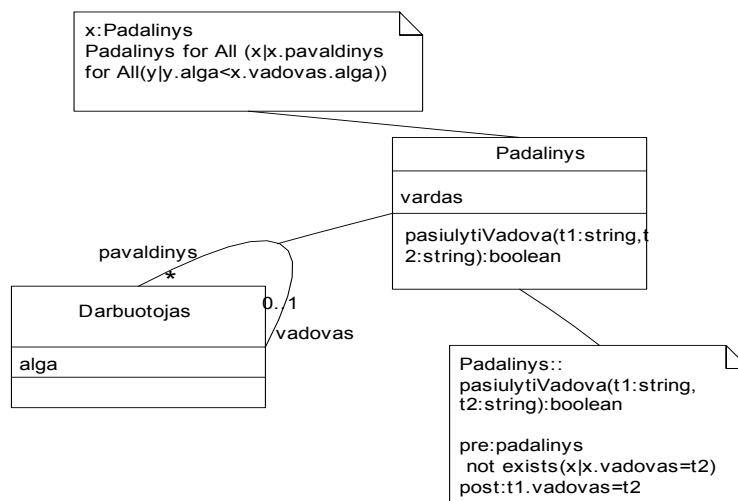
3. ECA trigerio šablonas. Verslo taisyklės savo standartu panašios į ECA taisykles. ECA taisyklės yra tokios formos: įvykis įvykiui, jei tenkinama sąlyga, vykdomas veiksmas. Įvykis nurodo, kada verslo taisyklė turi būti aktyvuojama, sąlyga apibrėžia, kas turi būti patikrinama prieš aktyvuojant taisyklę. ECA struktūra leidžia apibrėžti ir aktyvius ar pasyvius ribojimus, bet ir sudėtingus verslo procesus. Siūlomas būdas leidžia gauti tik trigerio šabloną, t.y. sugeneruojamas tik trigerio įvykis ir sąlygos tikrinimas, o veiksmo realizacija paliekama programuotojui.

Apibendrinant galima pasakyti kad siūlomi modeliuoti veiklos taisykles UML ir OCL, bei vėliau juos realizuoti DB technologijomis yra vienas iš būdų. Tačiau naudojant šį metodą neišvengiamos problemos:

Modeliuoti veiklos taisykles UML konstrukcijų neužtenka, todėl naudojama OCL, ji yra formali kalba, neturinti grafinės notacijos, todėl sunku VT modelius suprasti probleminės srities specialistams.

Siūlomi būdai nenagrinėja taisyklių, kurios specializuoja verslo algoritmus, procedūras ar tvarką, jų automatinė realizacija negalima siūlomu būdu.

Pateikiamas pavyzdys modeliuojamas UML ir naudojantis OCL kalbą aprašyti apribojimus (2 pav.).

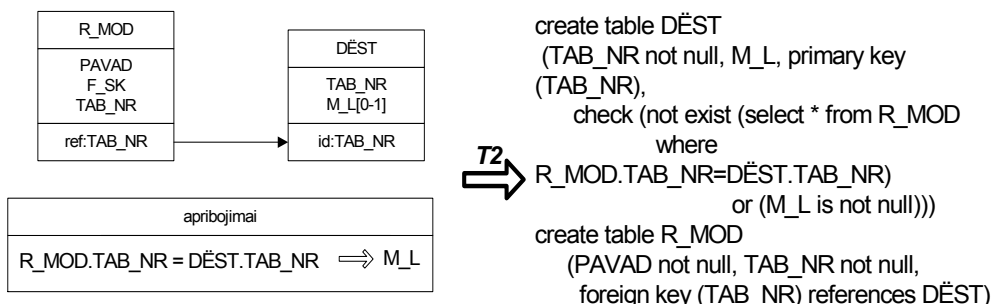


2 pav. Padalinio ir darbuotojo klasės su apribojimais

Duomenų struktūros EER ir OP

Reikalavimų analizės etape išsamiai analizuojama organizacijos veikla, aprašomos vartotojų grupių funkcijos bei apdorojama informacija[9]. Tačiau duomenų struktūros paprastai lieka netiksliai apibrėžtos, neformalizuotos. Konceptualiojo projektavimo etape vartotojams reikalinga informacija aprašoma pasirinktu semantiniu modeliu (objektiniu, ER, EER), kuris loginio projektavimo etape atvaizduojamas logine duomenų bazės schema (objektine ar reliacine), nepriklausoma nuo konkrečios DBVS. Fizinio projektavimo etape loginė schema atvaizduojama pasirinktos DBVS elementais.

Paimkime kitą pavyzdį apie modulių registraciją (M_REG). Modulus (R_MOD) – tiek privalomas, tiek alternatyvius – turi teisę registruoti mokslo laipsnį (M_L) turintys dėstytojai. Tačiau esybės DĖST egzemplioriais gali būti ir dėstytojai, neturintys mokslo laipsnio. Todėl esybės DĖST atributas yra neprivalomas ir jam nurodomas kardinalumas [0-1] (3 pav.). Apribojimas, kad kiekvienas dėstytojas, registruojantis modulį, turi turėti mokslo laipsnį, priskiriamas ryšiui M_REG ir gali būti pateikiamas implikacijos pavidalu. Eliminuojuant ryšį su kardinalumu 1-1 anksčiau nurodytu būdu, apribojimai perkeliama esybių R_MOD ir DĖST atributams. Norint realizuoti specifikuotus apribojimus RDB terpėje, naudojama transformacija T2 (3 pav.). Galimos funkcijos check (3 pav.) reikšmės pateiktos 2 lentelėje.



3 pav. EER modelio su implikacijomis nustatytais apribojimais transformavimas T2 į RDB loginės schemas aprašą

Pateiktoje RDB loginės schemas specifikacijoje (3 pav.) yra du išoriniai apribojimai: išorinio rakto apribojimas ir atributinės implikacijos.

Dėl pirmojo apribojimo *foreign key (TAB_NR) references DĖST* į lentelę R_MOD negalima įterpti įrašo su atributo TAB_NR reikšme, kurios nėra lentelėje DĖST.

Dėl antrojo apribojimo sudaryta loginio tikrinimo funkcija *check()*. Dėstytoju registruojant kokį nors modulį, dėl funkcijos *check()* lentelėje DĖST atributui M_L būtina suteikti kokią nors reikšmę (DR, MAG). Funkcija *check()* kontroliuoja implikacijos

$$\begin{aligned}
 & \text{exist (select * from R_MOD where R_MOD.TAB_NR=DĖST.TAB_NR)} \\
 & \Rightarrow M_L \text{ is not null}
 \end{aligned}$$

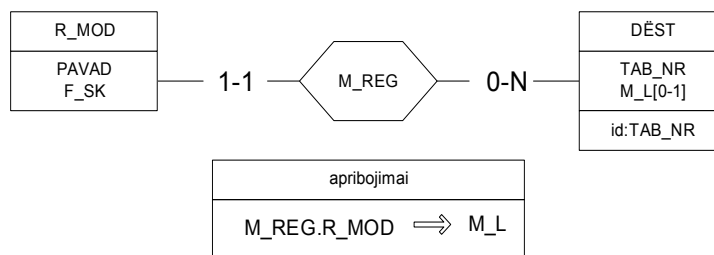
teisingumą (2 lent.).

2 lentelė. Funkcijos *check()* galimos loginės reikšmės

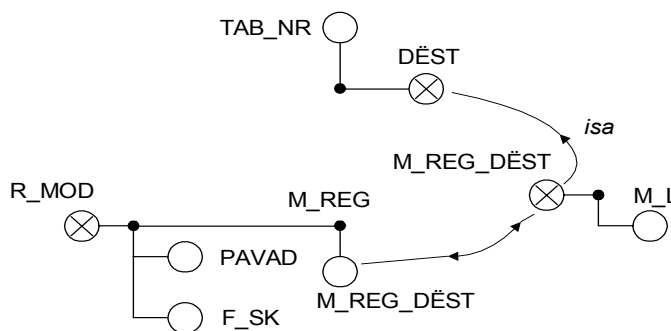
Exist()	not exist()	M_L is not	check()
0	1	1	1
0	1	0	1
1	0	1	1
1	0	0	0

Jeigu esant lygybei $R_MOD.TAB_NR = DĖST.TAB_NR$ atributas M_L turi neapibrėžtą (neįvestą) reikšmę \perp , ši funkcija įgyja reikšmę 0. Konkrečios RDBVS terpėje sugeneruota procedūra su vidine funkcija *check()* leis vykdyti DB modifikavimo programą, jeigu lentelėje DĖST įvedant kortežą iš karto bus nurodyta atributo M_L reikšmė arba lentelėje R_MOD įterpiant kortežą su nauja TAB_NR reikšme kartu bus įvedama lentelės DĖST kortežo (su ta pačia atributo TAB_NR reikšme) atributo M_L reikšmė.

Norint specifikuoti tuos pačius apribojimus, eliminuojant ryšį M_REG, įvedamas naujas objekto R_MOD atributas – mokslo laipsnį turintis dėstytojas (M_REG_DĖST), kuris traktuojamas kaip savarankiškas objektas su privaloma savybe M_L. Naujas objektas M_REG_DĖST bus sujungtas su objektu DĖST poabio priklausomybe



4 pav. Modulio registravimo ER diagrama po pakeitimo



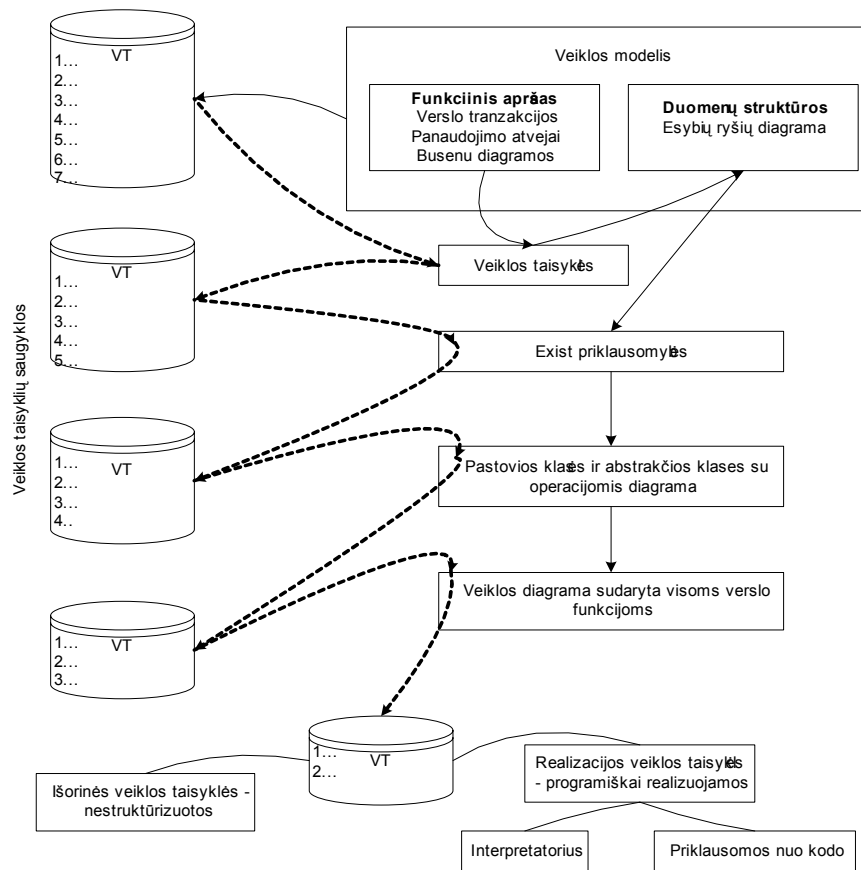
5 pav. Modulio registravimo OP diagrama

Veiklos taisyklių struktūrizavimo proceso modelis

Šio darbo tikslas išskirti tas veiklos taisykles, remiantis klasifikacijos metodu, kurias galime realizuoti struktūriškai duomenų bazės struktūrų schemomis. Kuriant tokią technologiją, būtina atsižvelgti į reikalavimus:

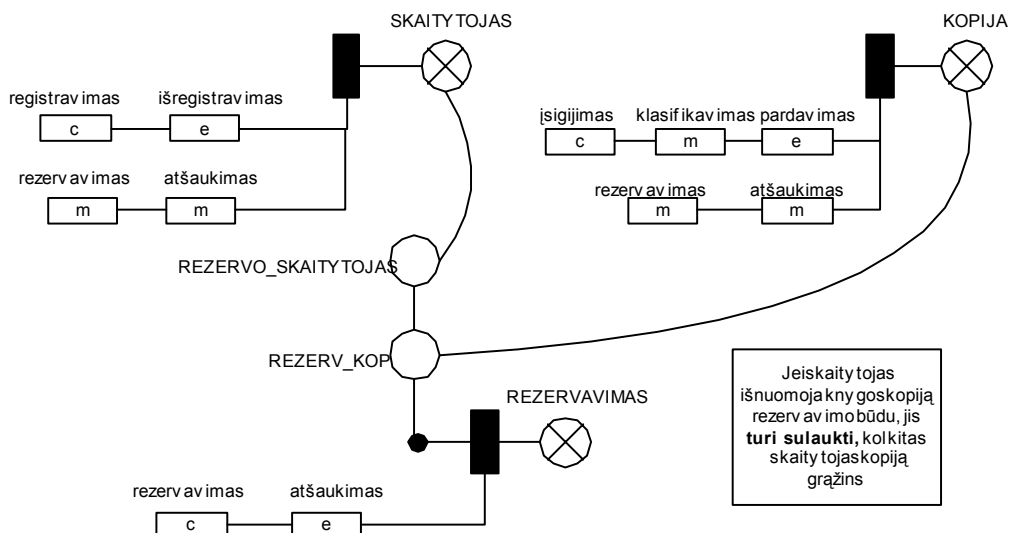
1. Kuriama informacinė sistema turi palaikyti taisyklių versle koncepciją
2. Turi būti kuriamas veiklos taisyklių modelis.
3. Tenkinami žinių vaizdavimo kalbų reikalavimai
4. Taisyklės turi būti struktūrizuotos bei modeliuojamos taikant VT požiūrį.
5. Tenkinami verslo taisyklių sistemų reikalavimai.

Remiantis šiais kriterijais yra siūloma veiklos taisyklių kūrimo technologija, panaudojant duomenų bazės struktūras. Siūlomas VT struktūrizavimo proceso modelis iš esmės pagrįstas veiklos taisyklių struktūrizavimu. Pradedant projektavimą nuo veiklos modelio, pažingsniui einama link išbaigto duomenų modelio. Veiklos modelį galima aprašyti naudojant verslo tranzakcijas, esybių ryšių diagramas, kur veiklos taisyklės „įvedamos“, kai sudaromi ryšiai, nurodomi būsimi konceptai, būsenos objektų, jų elgesys. Projektavimo metu veiklos taisyklės įvertinant kiekviename etape, jų lieka realizuoti programiškai vis mažiau, ir lieka tik grynos veiklos taisyklės. Tokiu būdu siekiama jau projektavimo metu, struktūrizuoti verslo taisykles per verslo tranzakcijas, sudėtinius objektų savybių modelius, aiškiai nubrėžiant, kur būtent yra „prikabinamos“ verslo taisyklės.

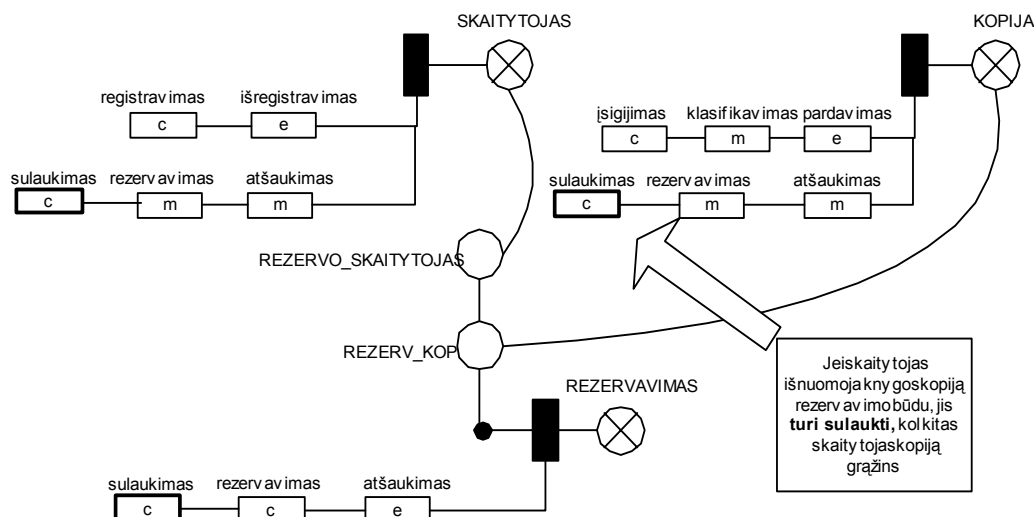


6 pav. Veiklos taisyklių struktūrizavimo proceso modelis

Pateiktas pavyzdys, kuriame nagrinėjamas knygos nuomos dalykinė sritis[9]. Kad skaitytojas galėtų išnuomoti rezervuotą knygą, jis turi jos sulaukti (7,8 pav.).



7 pav. Detalus knygos nuomos verslo taisyklės pavyzdys



8 pav. Detalus knygos nuomos verslo taisyklės pavyzdys

Išvados ir tolimesnio darbo kryptys

Tinkamas veikos taisyklių modeliavimas gali patobulinti informacijos sistemų modeliavimo bei projektavimo metodus ir pagerinti realizuotos sistemos funkcionavimą. Taisyklių pagalba galima formaliai aprašyti operacijų semantiką, valdyti verslo tranzakcijas. Šiuolaikinėse sistemose reikalingas taisyklių palaikymo mechanizmas, kuris leistų saugoti taisykles kaip pastovius objektus, jas modifikuoti, atsekti jas ten, kur jos įtrauktos į programų kodus. Ateityje bus vykdomas eksperimentas, kurio metu bus išskiriamos struktūrizuotos veikos taisyklės, bus vaizduojamos duomenų bazių loginėmis struktūromis ir realizuojamos programiškai, projektavimo etape pritaikant OP modelį, verslo tranzakcijas. Naudojant verslo tranzakcijų principą ir pritaikius interpretatoriaus mechanizmo modelį, kuris galėtų tas taisykles parinkti ir jas tikslingai vykdyti.

Literatūros sąrašas

- [1] A. Kolber, D. Hay ir kiti, GUIDE Business Rules Project, Final Report r. 1.2, GUIDE International Corporation, 1997, URL.: <http://www.guide.org/ap/apbrules>
- [2] Object Management Group, Business Rules in Models, 2002, URL.: <http://cgi.omg.org/cgi-bin/doc>
- [3] E. Gottesdiener, Show power promise, Business Rules, 1997, URL.: http://www.ebgconsulting.com/powerpromise_article.htm
- [4] D. Plotkin, Business Rule Everywhere, 1999, URL.: http://www.intelligententerprise.com/db_area/archives/1999/990903/feat2.shtml
- [5] R. G. Ross, The Business Rule Book: Classifying, Defining and Modeling Rules, Business Rule Solution, Second Edition. Houston, 1997
- [6] L. Nemuraitė, K. Ažubalis, Veiklos taisyklių modeliavimas objektiškai orientuotose metoduose, Informacinės technologijos 1999, Kaunas, Technologija, 1999, p. 171-178
- [7] RationalSoftwareCorp., 2002, URL.: http://www.rational.com/media/uml/resources/media/ad970808_uml11_ocl.pdf
- [8] I. Valatkaitė, O. Vasilecas, Informacinių sistemų modeliavimas dalykinės srities žinių požiūriu, Informacinės technologijos 2003, Kaunas, Technologija, 2003, p. XIV-18 – XIV-29
- [9] B. Paradauskas, L. Nemuraitė, Duomenų bazės ir semantiniai modeliai: monografija, Kaunas, Technologija, 2002, 336 p.

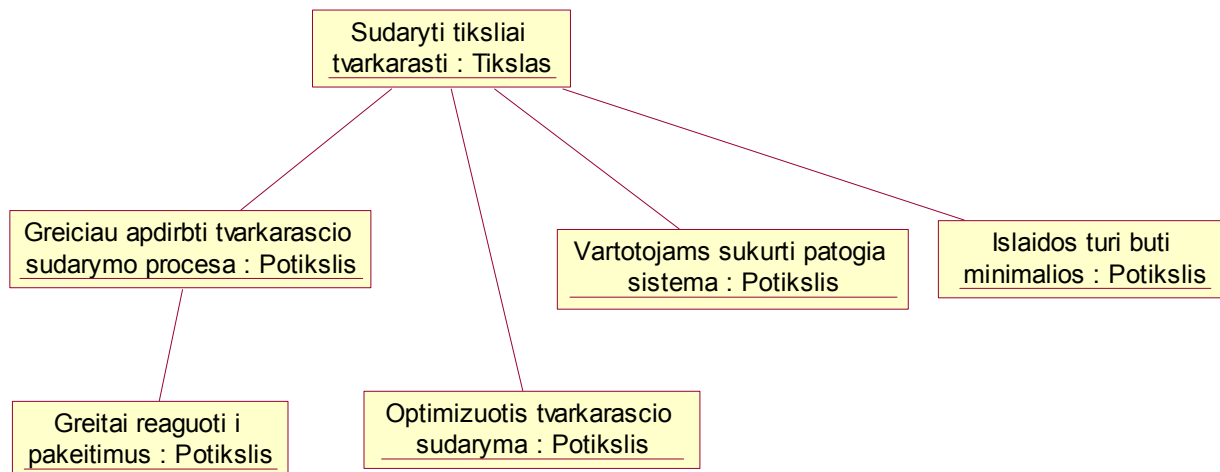
DESCRIPTION OF BUSINESS RULES IN DATABASE STRUCTURING APPROACH

While information system is design it is very important to understand organization processes, their constrains in requirements level. The business rules structuring is one of the problem to design information system. In this article at large is analyzed business rules classification and description ways in database designing and planning methods, including presentation of their advantages and disadvantages. Basing on the analysis work, the model of the

business rules structuring process is proposed and the possibilities for the further development of this models are described.

Priedas 2

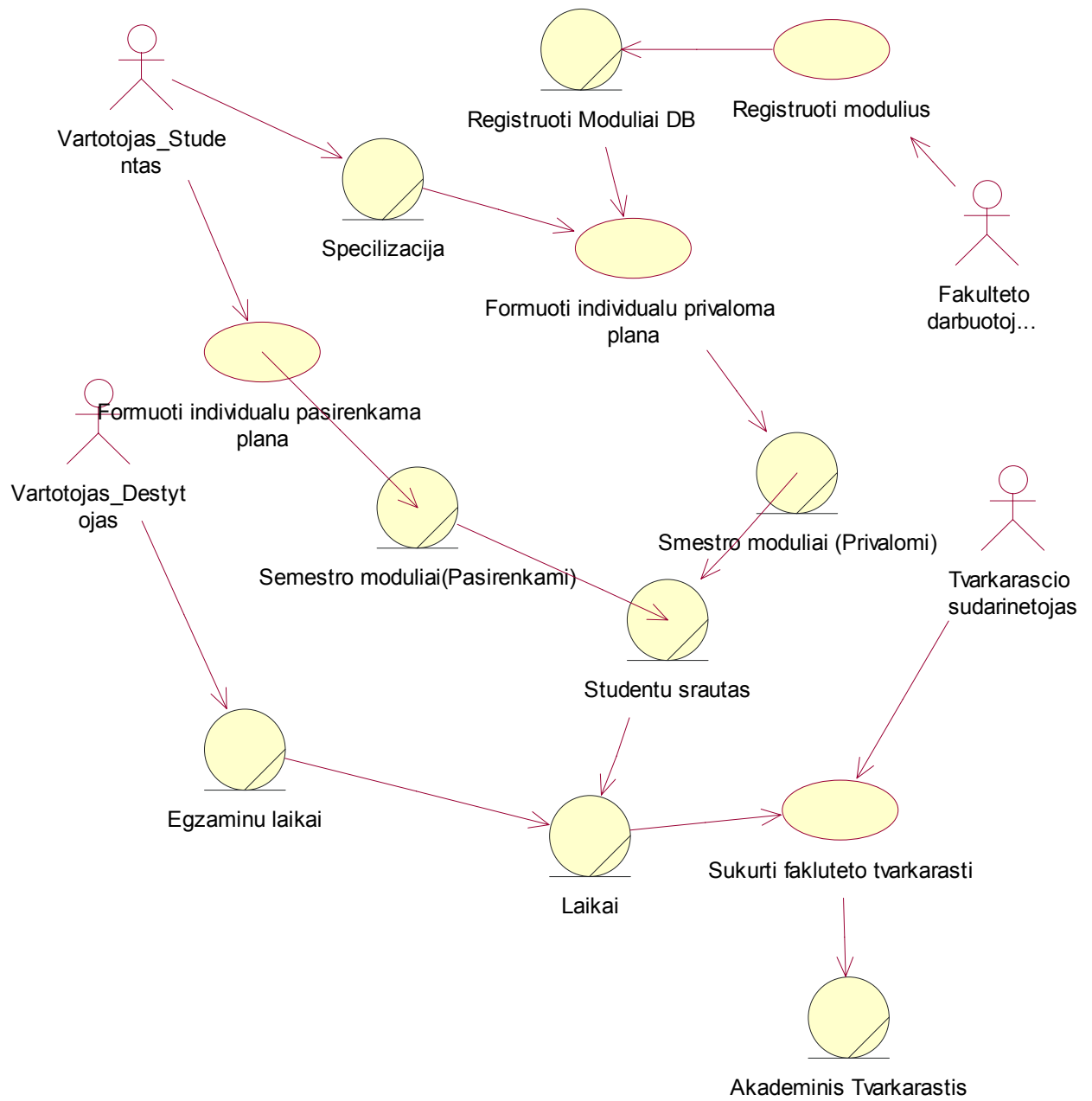
Veiklos tikslų modelis



1 pav. Tikslų modelis

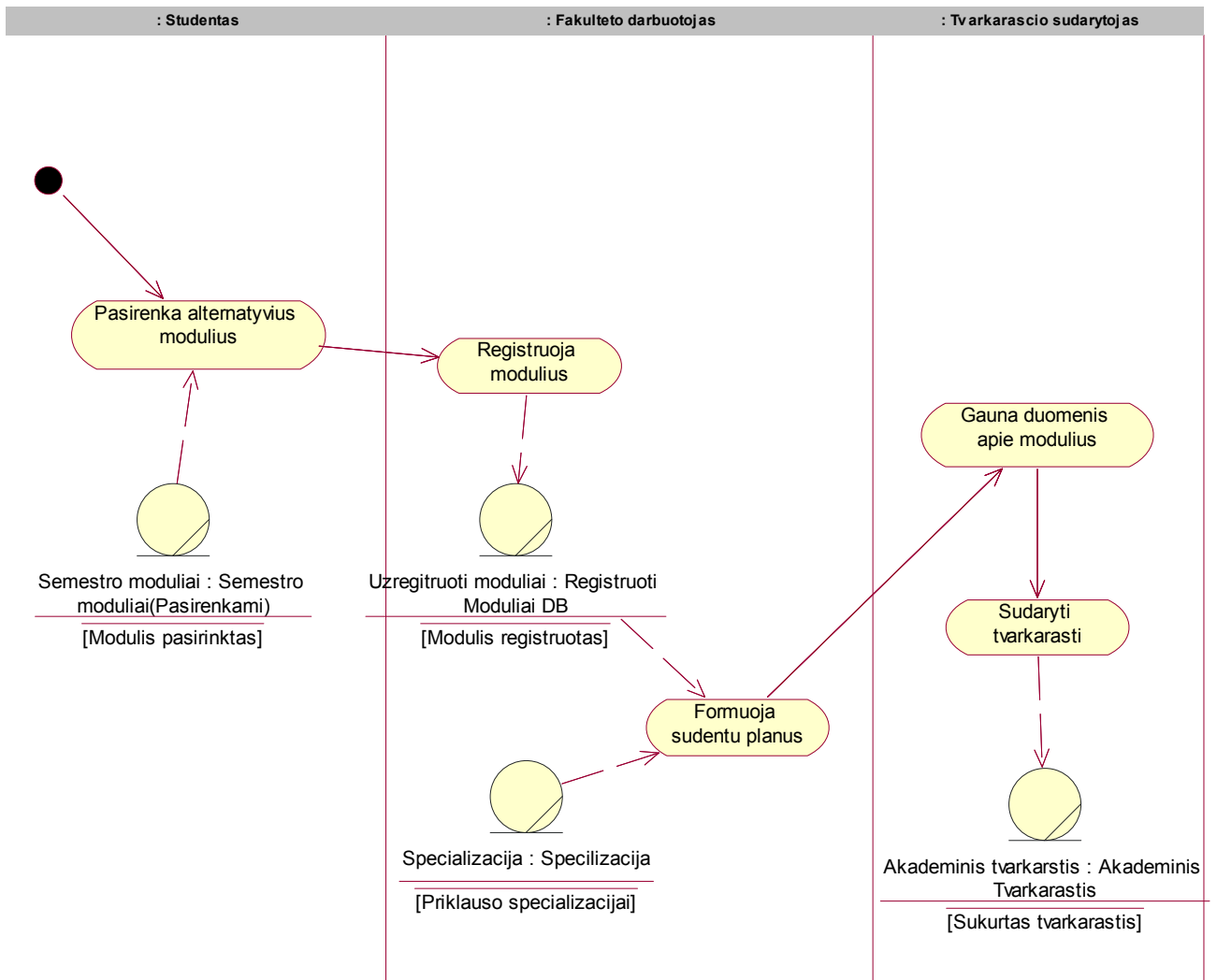
Visi organizacijos tikslai yra atvaizduoti tikslų modelyje. Organizacijos tikslus apima visos veiklos, kuriomis siekiama įgyvendinti organizacijos pagrindinius uždavinius. Pagrindinis tikslas yra organizacijos paskaitų tvarkaraščio sudarymas

Veiklos objektų klasių modelis



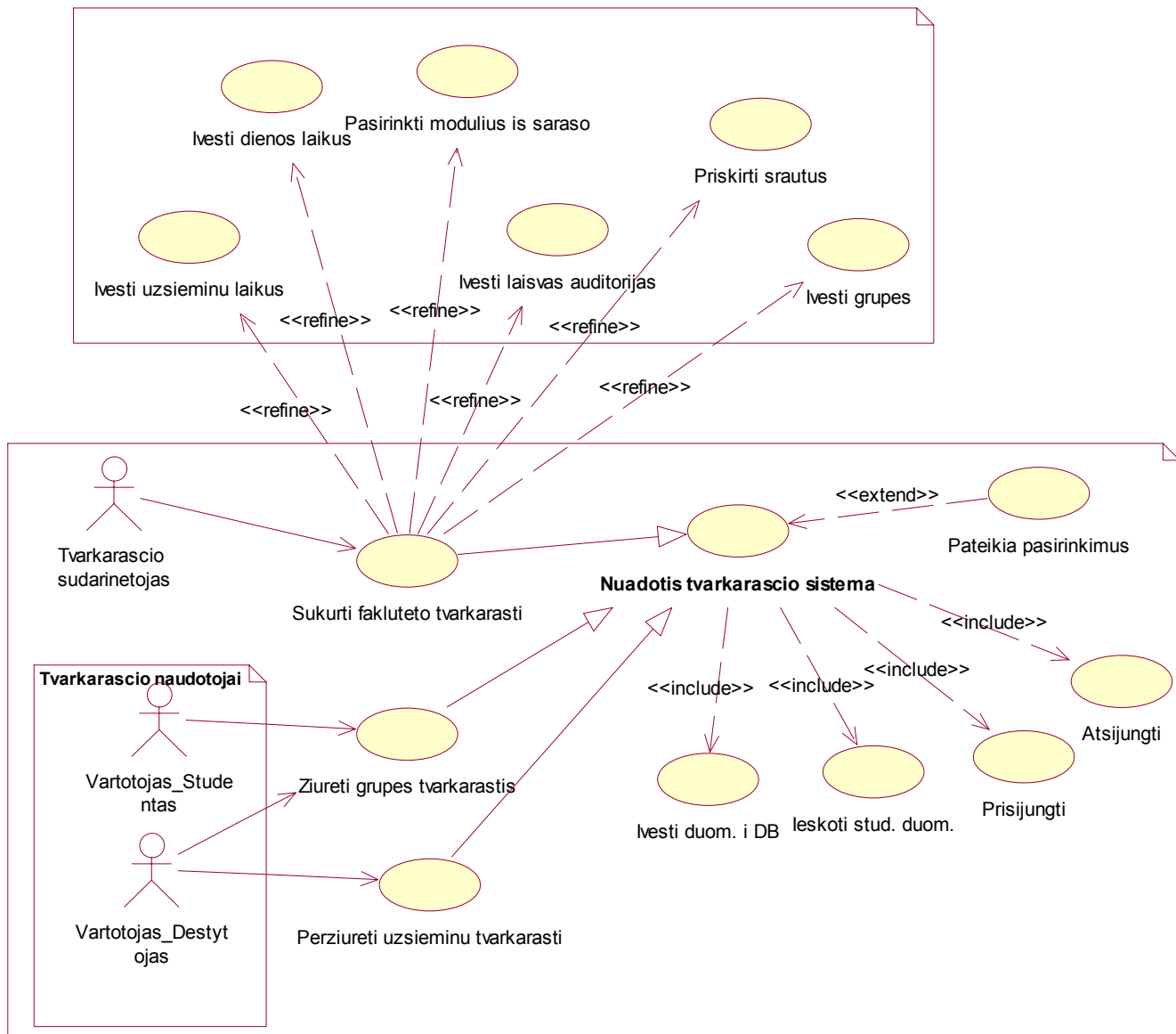
4 pav. Veiklos objektų klasių modelis

Veiklos procesų modelis



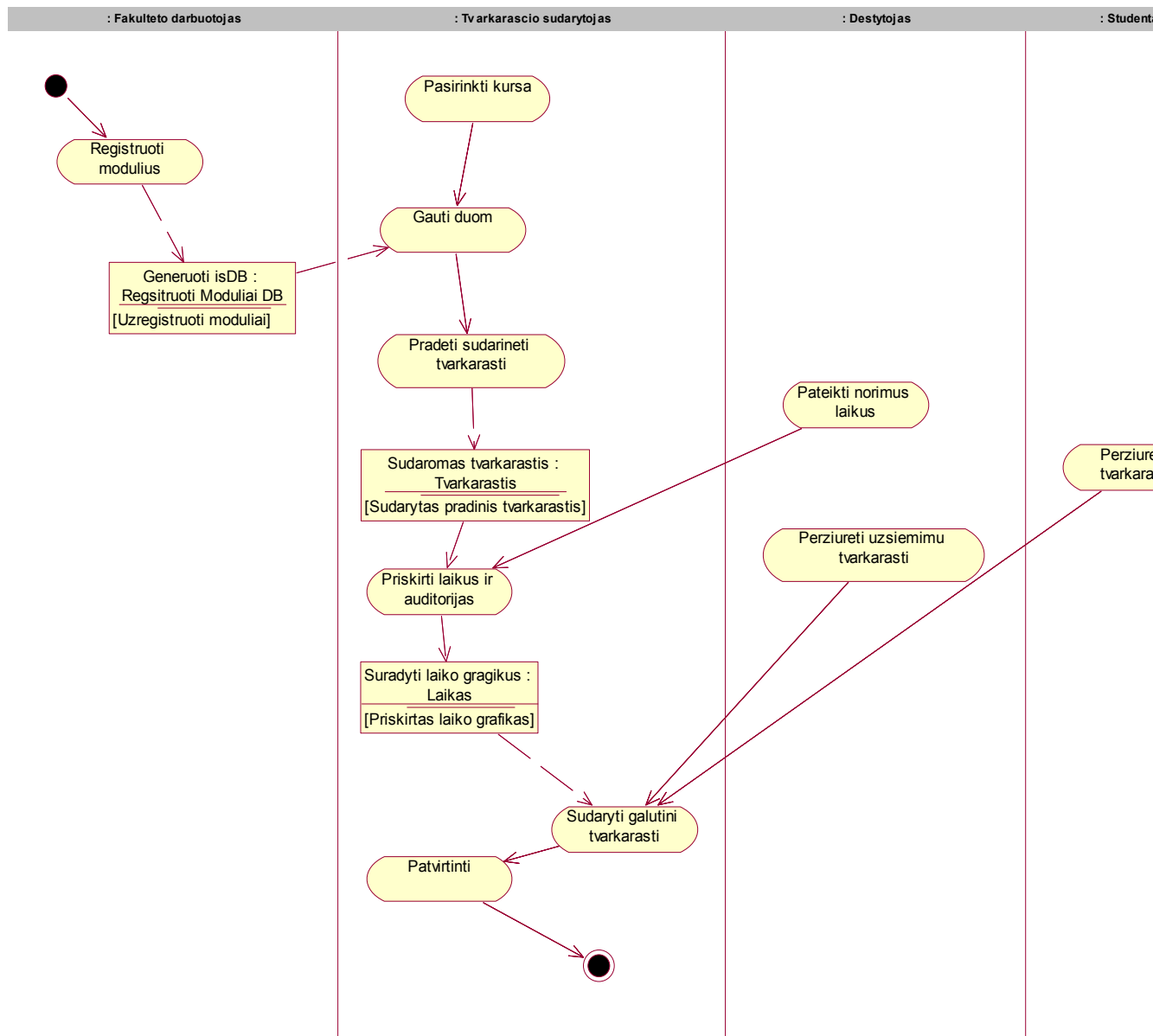
5 pav. Veiklos procesų modelis

Sistemos panaudojimo atvejų modelis



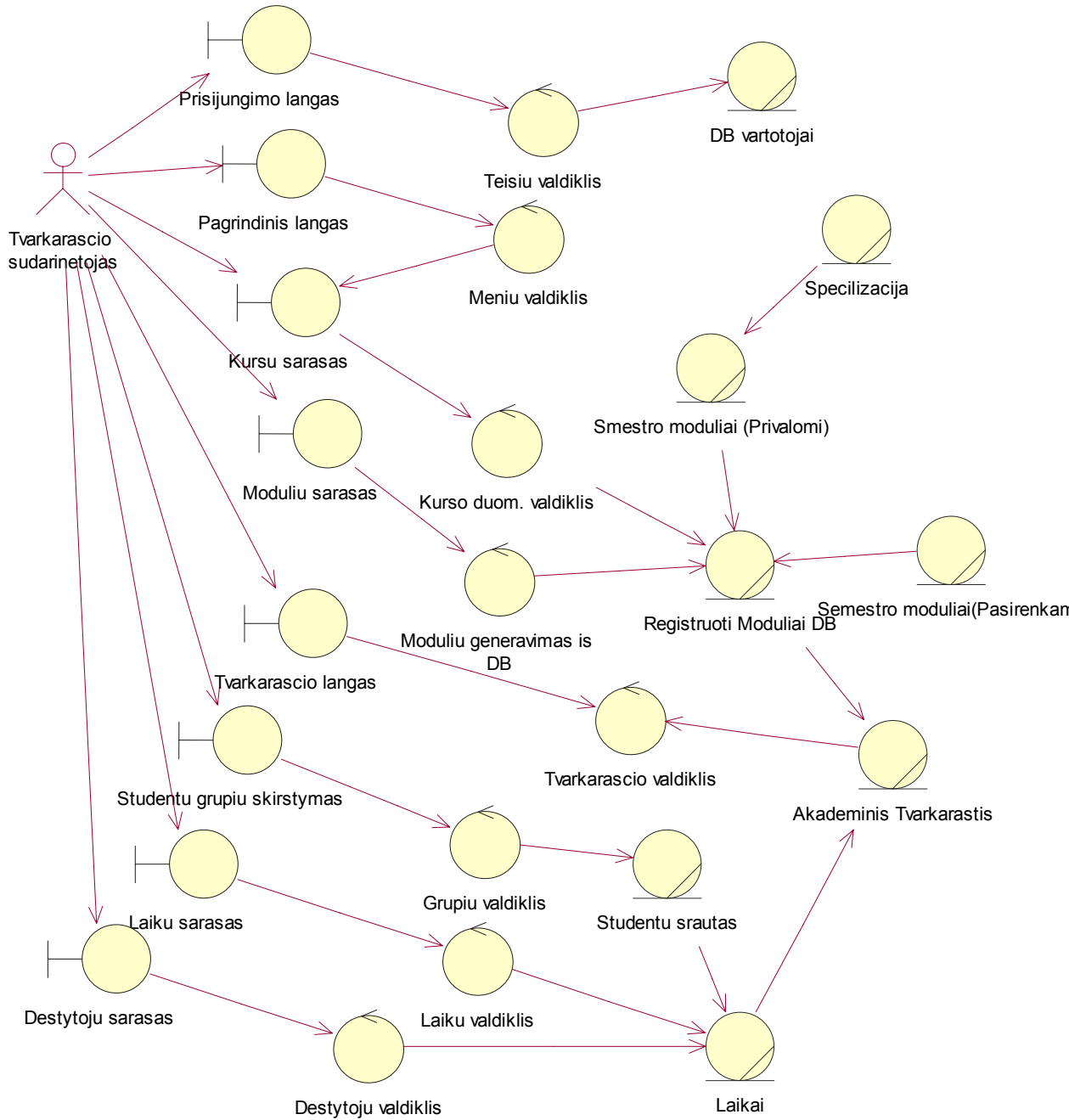
6 pav. Sistemos panaudojimo atvejų modelis

Sistemos veiklos diagrama



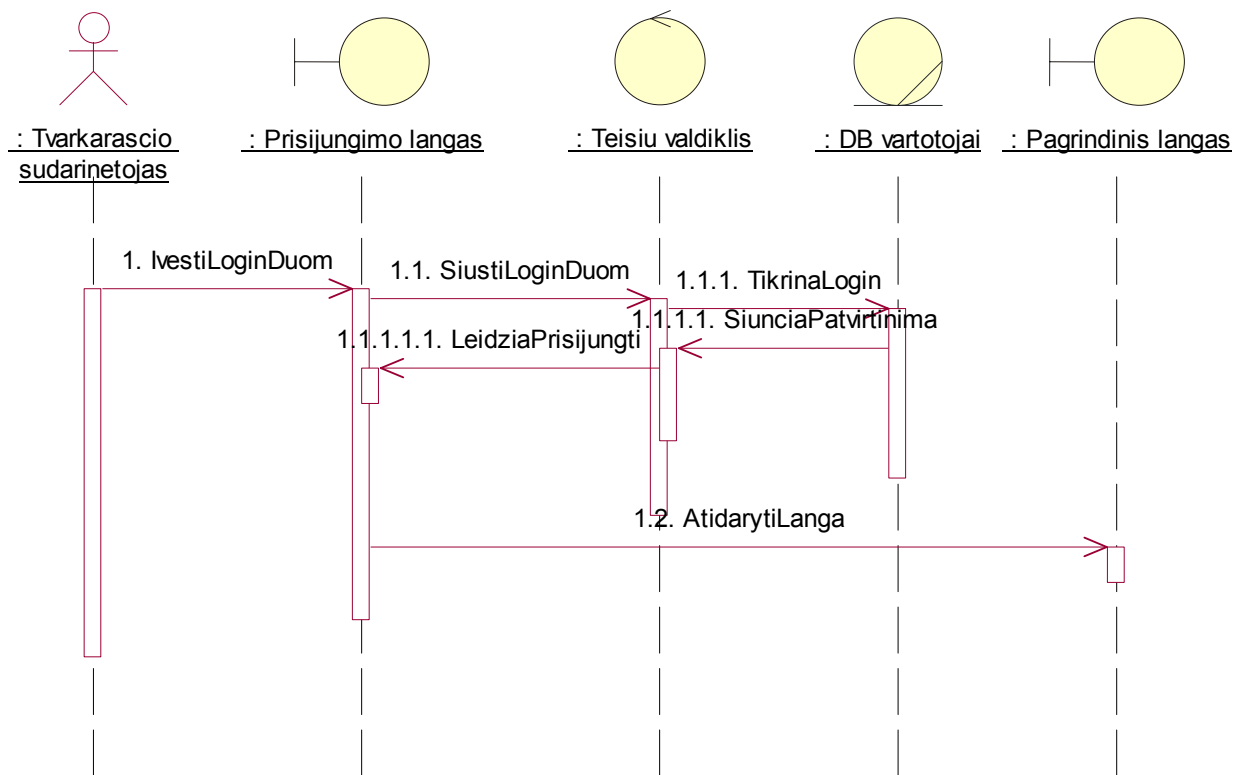
7 pav. Sistemos veiklos diagrama

Sistemos klasių modelis (ribinės klasės, valdikliai, esybės)

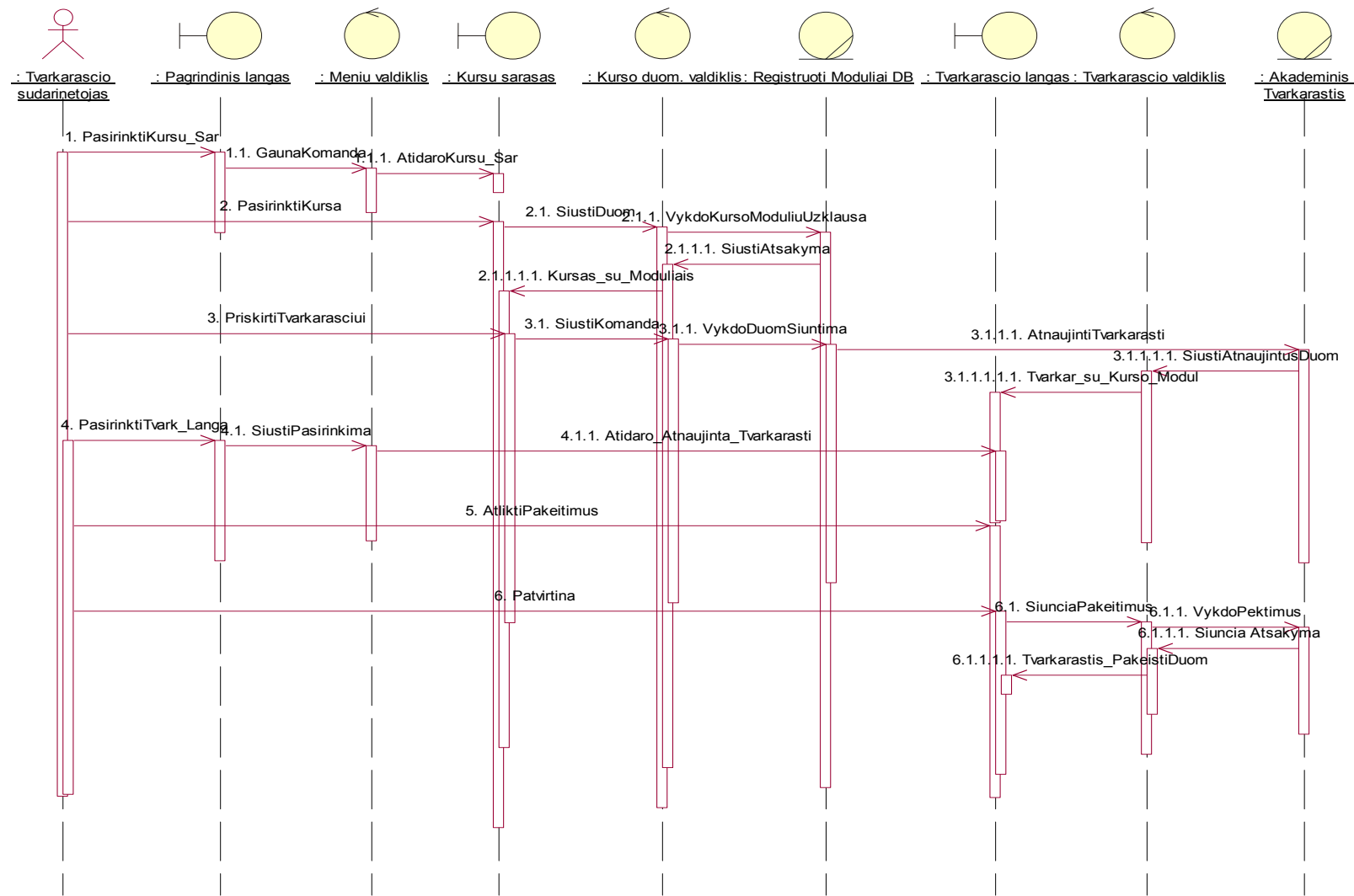


8 pav. Sistemos klasių modelis

Specifikacijos pasirinktam panaudojimo atvejui (Vartotojo ir sistemos sekų diagramos)

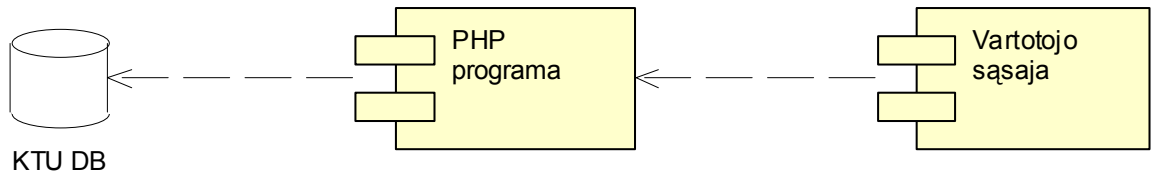


9 pav. Panaudojimo atvejo Prisijungimas prie sistemos sekų diagrama



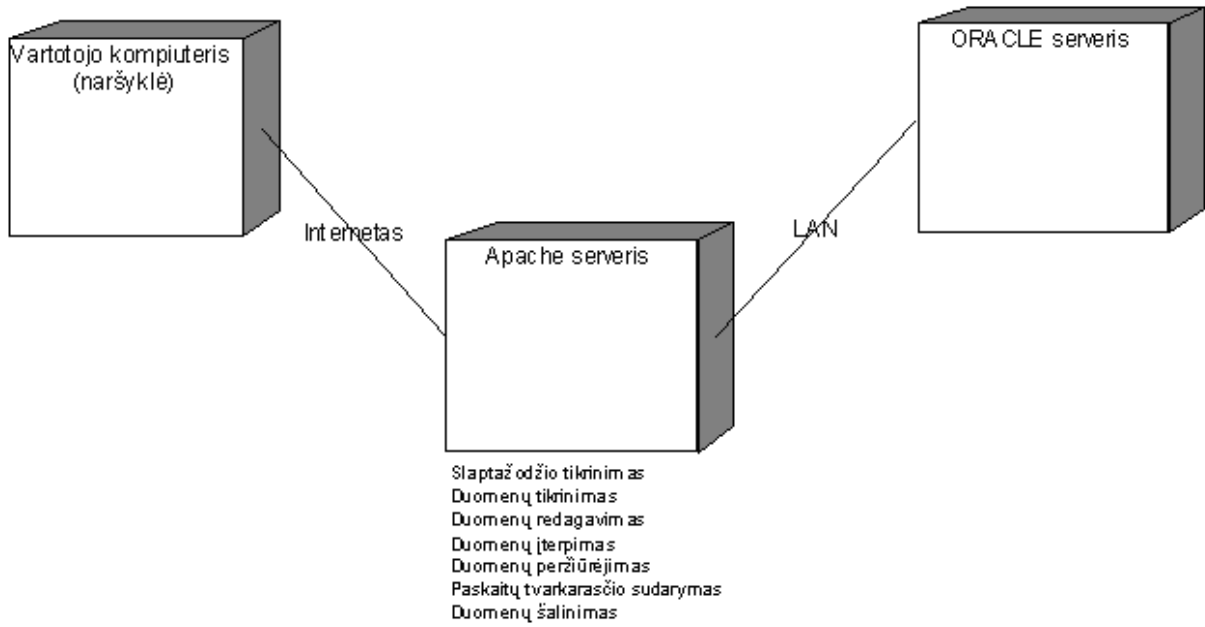
10 pav. Panaudojimo atvejo tvarkarascio sudarymo sekų diagrama

Komponentų diagrama



11 pav. Duomenų bazės diagrama

Paskirstymo diagrama



12 pav. Duomenų bazės diagrama