

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Raminta Žeknytė

**Veiklos taisyklių specifikavimo natūralios
kalbos šablonais ir jų realizavimo
duomenų bazėse metodika**

Magistro darbas

Darbo vadovas

prof. L. Nemuraitė

Kaunas, 2011

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Raminta Žeknytė

**Veiklos taisyklių specifikavimo natūralios
kalbos šablonais ir jų realizavimo
duomenų bazėse metodika**

Magistro darbas

Recenzentas

doc. dr. A. Lenkevičius

2011-05-27

Darbo vadovas

prof. L. Nemuraitė

2011-05-27

Atliko

IFM-9/4 gr. stud.
Raminta Žeknytė

2011-05-25

Kaunas, 2011

SUMMARY

Methodology for specifying business rules, using natural language templates, and implementing them in database

The main goal of this project was to improve the quality of requirement specifications by creating templates for specifying business rules in Lithuanian language and implementing them in database.

Methodology presents structured Lithuanian language templates for specifying *SBVR* business vocabularies and business rules with intention to implement them in a database. For this purpose *SBVR* Structured English templates were supplemented with detail templates for data formats, derivation rules and other elements. Metatables were proposed for keeping names for fact types and roles.

The suitability of the templates were verified by describing and implementing an experimental flight ordering system. Various types of business rules related with ordering flights were implemented by *SQL* triggers and procedures.

Keywords: *SBVR*, fact type, business rule, templates, database, *SQL*, trigger, procedures.

Turinys

1. ĮVADAS	7
2. VEIKLOS TAISYKLIŲ SPECIFIKAVIMO METODŲ IR FORMŲ ANALIZĖ ...	9
2.1. TYRIMO TIKSLAI IR UŽDAVINIAI	9
2.2. VEIKLOS TAISYKLIŲ KLASIFIKACIJŲ ANALIZĖ	10
2.2.1. <i>Veiklos taisyklių samprata</i>	<i>10</i>
2.2.2. <i>Taisyklių klasifikacija pagal GUIDE veiklos taisyklių projektą.....</i>	<i>10</i>
2.2.3. <i>Taisyklių klasifikacija pagal SBVR.....</i>	<i>12</i>
2.2.4. <i>Taisyklių klasifikacija pagal Witt.....</i>	<i>13</i>
2.2.5. <i>Veiklos taisyklių klasifikacijų apibendrinimas.....</i>	<i>14</i>
2.3. VEIKLOS TAISYKLIŲ SPECIFIKAVIMO ŠABLONŲ ANALIZĖ	15
2.3.1. <i>RuleSpeak veiklos taisyklių šablonų analizė.....</i>	<i>15</i>
2.3.2. <i>SBVR veiklos taisyklių šablonų analizė</i>	<i>15</i>
2.3.3. <i>Witt veiklos taisyklių šablonų analizė.....</i>	<i>16</i>
2.3.4. <i>Esamų taisyklių specifikuojamųjų metodų palyginimas.....</i>	<i>21</i>
2.4. SIEKIAMAS SPRENDIMAS	22
2.4.1. <i>Darbo formuluotė.....</i>	<i>22</i>
2.4.2. <i>Taisyklių realizavimo technologijos pasirinkimas ir analizė.....</i>	<i>23</i>
2.5. ANALIZĖS IŠVADOS	25
3. VEIKLOS TAISYKLIŲ SPECIFIKAVIMO METODIKA TAIKANT SBVR NATŪRALIOS KALBOS ŠABLONUS.....	26
3.1. METODIKAI KELIAMSI REIKALAVIMAI.....	26
3.2. FAKTŲ TIPŲ ŠABLONAI	27
3.3. VEIKLOS TAISYKLIŲ ŠABLONAI.....	29
3.4. SBVR KONCEPTŲ TRANSFORMAVIMAS Į DB SCHEMĄ.....	34
3.4.1. <i>Terminų transformavimas.....</i>	<i>34</i>
3.4.2. <i>Faktų tipų ir kai kurių veiklos taisyklių transformavimas</i>	<i>35</i>
3.4.3. <i>Struktūrinių veiklos taisyklių transformavimas į realizavimo DB priemones .</i>	<i>36</i>
3.4.3.1. Taisyklės, reiškančios privalomus atributus	36
3.4.3.2. Taisyklės, reiškančios kardinalumo ribojimus (1:0..*)	36
3.4.3.3. Taisyklės, reiškančios kardinalumo ribojimus (1:0..1)	37
3.4.3.4. Taisyklės, reiškančios kardinalumo ribojimus (1:1..*)	38
3.4.3.5. Taisyklės, reiškančios išskirtinumo ribojimą tarp 2 ar daugiau DB stulpelių.....	39
3.4.3.6. Taisyklės, reiškančios išskirtinumo ribojimą tarp 2 ar daugiau ryšių su kitomis DB lentelėmis	39
3.4.3.7. Taisyklės, reiškančios, kad DB lentelės atributo reikšmė turi būti iš išvardytos atributo reikšmių aibės	40
3.4.3.8. Taisyklės, apimančios 2 ar daugiau objektų tipų vaidmenų, kurie pavaizduojami skirtingų DB lentelių stulpeliais	41
3.4.3.9. Taisyklės, reiškančios, kad DB lentelės atributų reikšmės turi būti iš išvardytos atributo reikšmių aibės	42
3.4.3.10. Taisyklės, apimančios objektų tipus ir individualius konceptus.....	42
3.4.3.11. Išvedimo (skaičiavimo) taisyklės, apimančios dviejų laukų (pvz., datos ir laiko) sujungimą	45
3.4.3.12. Taisyklės, reiškančios atributų reikšmių ribojimus	48
3.4.4. <i>Operacinių veiklos taisyklių transformavimas.....</i>	<i>48</i>
3.4.5. <i>DB schemas semantikos išplėtimas.....</i>	<i>50</i>

4.	EKSPERIMENTINIS METODIKOS TYRIMAS.....	52
4.1.	EKSPERIMENTO APIBRĖŽIMAS	52
4.1.1.	<i>Eksperimento tikslas</i>	52
4.1.2.	<i>Eksperimento planas</i>	52
4.2.	EKSPERIMENTINĖS SKRYDŽIO UŽSAKYO SISTEMOS REIKALAVIMAI	53
4.2.1.	<i>Panaudojimo atvejai jų specifikacijos</i>	53
4.2.2.	<i>Dalykinės srities modelis</i>	61
4.2.3.	<i>Šablonų tinkamumo įvertinimas</i>	63
4.2.3.1.	Dalykinės srities aprašymas natūralia kalba ir veiklos taisyklėmis	63
4.2.3.2.	Faktų tipų sudarymas pagal <i>SBVR</i>	63
4.2.3.3.	Veiklos taisyklių sudarymas iš faktų tipų	64
4.2.3.4.	Lietuviškų šablonų sudarymas	65
4.2.3.5.	DB schema su išsaugojamais vaidmenų pavadinimais	68
4.3.	EKSPERIMENTINĖS SKRYDŽIO UŽSAKYO SISTEMOS PROJEKTAS	69
4.3.1.	<i>Reikalavimų analizė</i>	69
4.3.2.	<i>Loginė visos sistemos architektūra</i>	71
4.3.3.	<i>Vartotojo paslaugos</i>	71
4.3.4.	<i>Sistemos elgsenos modeliai</i>	72
4.3.5.	<i>Duomenų bazės schema</i>	75
4.3.6.	<i>Realizacijos modelis</i>	76
5.	EKSPERIMENTINĖS SKRYDŽIO UŽSAKYO SISTEMOS REALIZACIJA..	77
5.1.	SKRYDŽIO UŽSAKYO SISTEMOS PASKIRTIS, PAGRINDINĖS FUNKCIJOS.....	77
5.2.	REALIZUOTOS EKSPERIMENTINĖS SKRYDŽIO UŽSAKYO SISTEMOS DUOMENŲ BAZĖS SCHEMA IR LENTELIŲ SPECIFIKACIJOS	77
5.3.	SKRYDŽIO UŽSAKYO SISTEMOS VEIKIMO APRAŠYMAS.....	82
5.3.1.	<i>Skrydžio užsakymo sistemos testavimo planas, duomenys ir rezultatai</i>	87
5.4.	METODIKOS ĮVERTINIMAS REMIANTIS EKSPERIMENTINĖS SISTEMOS ANALIZĖ.....	88
5.5.	EKSPERIMENTO IŠVADOS	88
6.	IŠVADOS.....	89
7.	LITERATŪRA.....	91
	PRIEDAI.....	93
	1 priedas. Straipsnis „ <i>SBVR</i> šablonai duomenų bazėse realizuojamoms veiklos taisyklėms specifiukuoti“	93
	2 priedas. Faktų tipai	97
	3 priedas. Veiklos taisyklių sakiniai (RS)	100
	4 priedas. 4 konceptų stiliai.....	104

Terminų ir santrumpų žodynas

DB -	„duomenų bazė” atsirado XX amžiaus 6-ojo dešimtmečio pabaigoje, tačiau ir šiuo metu vis dar įvairiai apibrėžiamas. DB - tai kartu saugomų ir susijusių duomenų, skirtų apdoroti kompiuteriu, visuma.
ECA taisyklės -	įvykis (angl. Event), sąlyga (angl. Condition) ir veiksmas (angl. Action)- tai taisyklės, kurias galima aprašyti naudojant schemą: JEIGU <įvykis> IR <sąlyga> TAI <veiksmas>.
Fakto tipas -	tai ryšys tarp dviejų ar daugiau daiktavardžių sąvokų.
FT -	faktų tipo santrumpa.
KFT -	kitų faktų tipo santrumpa.
RS –	(angl. <i>Rule Sentence</i>) – taisyklės (sakinio) santrumpa.
SBVR -	(angl. <i>Semantics of Business Vocabulary and Business Rules</i>) – yra OMG (angl. <i>Object Management Group</i>) priimtas standartas, kurį sukūrė Business Rule Team konsorciumas. <i>SBVR</i> specifikacija apibrėžia veiklos žodyno, veiklos faktų tipų ir veiklos taisyklių semantiką.
SQL -	(angl. <i>Structured Query Language</i>) – populiariausia iš šiuo metu naudojamų kalbų, skirtų aprašyti duomenis ir manipuliuoti jais reliacinių duomenų bazių valdymo sistemose.
Trigeris -	(angl. <i>trigger</i> – iššaukti, sukelti) – tai speciali procedūra, kuri susiejama su lentele ir iškviečiama įvykus konkrečiam įvykiui.
T-SQL -	(angl. <i>Transact- Structured Query Language</i>) – Transakcija-„struktūrizuota užklausų kalba“).
UML -	(angl. <i>Unified Modeling Language</i>) – vaizdinio modeliavimo kalba, 1997m. OMG (angl. <i>Object Management Group</i>) patvirtinta standartu.
Veiklos taisyklė –	(angl. <i>Business rule</i>) teiginys, kuris apibrėžia arba veikia kai kuriuos veiklos aspektus, t.y. skirtas kontroliuoti verslo struktūrą ar veikti veiklos elgesį.
Veiklos taisyklė –	būdas struktūrizuoti verslo žinias.
Veiklos taisyklė –	taisyklė, kuri priklauso veiklai.

1. Įvadas

Šio tiriamojo darbo [15] tikslas yra pagerinti veiklos taisyklių specifikuojimo procesą, sudarant natūralios kalbos šablonų sudarymo ir jų realizavimo duomenų bazėse metodiką, sukuriant pavyzdinę sistemą. Pagrindinė problema yra ta, kad nėra tinkamos metodikos, kaip išsamiai ir vienareikšmiškai aprašyti veiklos taisykles, apibrėžiant IS reikalavimus ir kaip jas patogiai realizuoti duomenų bazėse. Antra, saugant dalykinės srities informaciją duomenų bazėje, prarandama dalis semantinės informacijos, kuri reikalinga norint vykdyti semantines užklausas ar pateikti duomenų bazių turinį Pasaulio Semantiniame tinkle. Analizuojama, kaip reikėtų formuluoti reikalavimus *SBVR* [4] žodynų ir taisyklių pavidalu, kad aprašytų žinių pakaktų juos vienareikšmiškai realizuoti duomenų bazėje neprarandant semantikos. Todėl siekiant pagerinti veiklos taisyklių specifikuojimo ir realizavimo duomenų bazėse procesą, buvo sukurta metodika ir pavyzdinė sistema, kurioje yra įgyvendintos įvairios veiklos taisyklės.

Tyrimo tikslas – pagerinti reikalavimų specifikuojimo kokybę, sudarant rekomendacijas ir šablonus veiklos taisyklėms specifikuoti lietuvių kalba ir realizuoti *SQL* šablonais.

Parašyta metodika naudinga siekiant tiksliai aprašyti reikalavimus duomenų bazėms kurti bei automatizuoti DB schemų generavimą iš *SBVR* veiklos žodynų.

Šablonų rinkinio tinkamumui ir įgyvendinamumui patikrinti buvo aprašyta eksperimentinė skrydžio užsakymo sistema, kuri - realizuota *Microsoft SQL Server 2008* ir *Microsoft Visual Studio 2008* [13] priemonėmis. Eksperimentinėje skrydžio užsakymo sistemoje buvo įgyvendintos įvairios veiklos taisyklės apie skrydžio užsakymą, kurios - realizuotos *SQL* trigeriais [11], [14], [17] ir procedūromis [12]. Šios sistemos paskirtis – skrydžio bilieto užsakymo sudarymas bei skrydžio registravimo įvykdymas.

Darbe buvo sprendžiami šie **uždaviniai**:

- Išanalizuoti veiklos taisyklių klasifikacijas, egzistuojančius veiklos taisyklių šablonus;
- Išanalizuoti DB technologijas, tinkamas veiklos taisyklėms realizuoti;
- Sudaryti veiklos konceptų ir taisyklių specifikuojimo lietuvių kalba šablonų rinkinį, apimantį konceptus ir taisyklių tipus, reikalingus realizuojant taisykles duomenų bazėje;
- Sudaryti veiklos taisyklių realizavimo duomenų bazėje šablonus;
- Realizuoti ir ištestuoti sudarytais šablonais aprašytų veiklos taisyklių rinkinį;
- Įvertinti savo sukurtą metodiką.

Metodikos vertinimo kriterijai - šablonais užrašomos taisyklių aibės vienareikšmiškumas ir išsamumas, t.y. siekiama aprašyti visas DB kūrimui reikalingas taisykles, kurias galima vienareikšmiškai realizuoti duomenų bazių priemonėmis.

Veiklos taisyklių klasifikacijų ir šablonų analizei atlikti buvo analizuojami *SBVR*, *RuleSpeak* ir autoriaus *Witt* literatūros šaltiniai. Veiklos taisyklėms realizuoti - naudojama techninė literatūra apie procedūras, trigerius bei programines priemones.

Darbo struktūra. Darbo analizės dalyje analizuojami veiklos taisyklėms aprašyti taikomi šablonai ir pagrindžiamas *SBVR* šablonų pasirinkimas. Taip pat atliktas veiklos taisyklių realizavimo technologijos pasirinkimas, analizė ir apibrėžtas siekiamas sprendimas. Šios dalies pabaigoje pateikti analizės rezultatai ir suformuluoti tikslai tolesniam tyrimui.

Pagrindinėje darbo dalyje aprašyti metodikai keliami reikalavimai, faktų tipų ir veiklos taisyklių šablonai bei *SBVR* konceptų transformavimo į DB schemą taisyklės. Šios dalies pabaigoje apibendrinami naujai įvesti šablonai ir pateikiamos metaduomenų lentelės, leidžiančios išsaugoti daugiau semantikos.

Eksperimento dalyje apibrėžtas eksperimento apibrėžimas, užrašyti eksperimentinės skrydžio užsakymo sistemos reikalavimai ir sudarytas jos projektas.

Realizacijos dalyje pateikiamas eksperimentinė skrydžio užsakymo sistema, jos veikimo aprašymas bei įvertinti eksperimento rezultatai.

Pabaigoje pateikiamos atlikto darbo pagrindinės išvados.

Prieduose pateikiamas straipsnis „*SBVR* šablonai duomenų bazėse realizuojamoms veiklos taisyklėms specifikuoti“, faktų tipai ir veiklos taisyklės bei keturi konceptų stiliai.

2. Veiklos taisyklių specifikuojamųjų metodų ir formų analizė

Šiame skyriuje yra apibrėžiama tyrimo sritis, objektas, bei problema. Pagal tyrimo problemą, nustatomi tyrimo tikslai bei uždaviniai. Atliekamas esamųjų taisyklių specifikuojamųjų metodų palyginimas bei veiklos taisyklių realizavimo technologijos pasirinkimas. Įvertinami atliktos analizės tyrimo rezultatai bei nustatomi tolesni darbo tikslai.

2.1. Tyrimo tikslai ir uždaviniai

Tyrimo sritis - veiklos taisyklių specifikuojamasis ir DB technologijos.

Tyrimo objektas - veiklos taisyklių specifikuojamasis ir realizavimo DB procesas.

Tyrimo problemos - nėra tinkamos metodikos, kaip išsamiai ir vienareikšmiškai aprašyti veiklos taisykles, specifikuojant IS reikalavimus taip, kad jas būtų patogu realizuoti DB. Dažnai veiklos taisyklės yra užregistruojamos realizuotojų realizavimo metu, todėl sunku atsekti, kaip programinė realizacija atitinka reikalavimus.

Tyrimo tikslas – pagerinti reikalavimų specifikuojamųjų kokybę, sudarant rekomendacijas ir šablonus veiklos taisyklėms specifikuoti lietuvių kalba ir realizuoti *SQL* šablonais.

Tyrimo uždaviniai:

1. Išanalizuoti veiklos taisyklių klasifikacijas, egzistuojančius veiklos taisyklių šablonus;
2. Išanalizuoti DB technologijas, tinkamas veiklos taisyklėms realizuoti;
3. Sudaryti veiklos konceptų ir taisyklių specifikuojamųjų lietuvių kalba šablonų rinkinį, apimančių konceptus ir taisyklių tipus, reikalingus realizuojant taisykles duomenų bazėje;
4. Sudaryti veiklos taisyklių realizavimo duomenų bazėje šablonus;
5. Realizuoti ir ištestuoti sudarytais šablonais aprašytų veiklos taisyklių rinkinį;
6. Įvertinti savo sukurtą metodiką.

Metodikos vertinimo kriterijai - šablonais užrašomos taisyklių aibės vienareikšmiškumas ir išsamumas, t.y. siekiama aprašyti visas DB kūrimui reikalingas taisykles, kurias galima vienareikšmiškai realizuoti duomenų bazių priemonėmis.

Analizės tikslas – apibrėžti veiklos taisyklių sąvokas, esamas klasifikacijas, išanalizuoti veiklos taisyklių šablonus, taisyklių realizavimo kalbos galimybes, bei patikslinti tyrimo uždavinius.

Analizės metodai:

- Literatūros analizė – analizuojama įvairi literatūra apie veiklos taisyklių klasifikacijas bei šablonus.

- Lyginamoji analizė – panašių metodų ir šablonų palyginimas pagal tam tikrus kriterijus.

2.2. Veiklos taisyklių klasifikacijų analizė

Šiame poskyryje apibrėžiama veiklos taisyklių samprata, apžvelgiamos bei apibendrinamos skirtingų autorių veiklos taisyklių klasifikacijos. Išnagrinėtos šios klasifikacijos:

- Taisyklių klasifikacija pagal *GUIDE* veiklos taisyklių projektą [2];
- Taisyklių klasifikacija pagal *SBVR*;
- Taisyklių klasifikacija pagal Witt “A Practical Method of Developing Natural Language Rule Statements” [1].

2.2.1. Veiklos taisyklių samprata

Veiklos taisyklės yra rinkinys taisyklių, kurios nusako, riboja ar veikia konkrečius veiksmus, juos vykdant konkrečiomis sąlygomis. Veiklos taisyklės yra apibrėžiamos projekto dalykinės srities reikalavimų rinkimo etape, kai yra bandoma surinkti visus veiksmus, kurie turi ar gali veikti konkretų projekte kompiuterizuojamą objektą ar jo operaciją. Idealiu atveju veiklos taisyklių rinkinį turėtų būti galima papildyti bet kuriuo metu, kai to reikalauja verslo pokyčiai.

Visos organizacijos yra valdomos taisyklių. Kiekviena organizacija pasirenka būti taip valdoma dėl įvairių priežasčių: rizikos mažinimas, kainos mažinimas, pajamų apsauga ir t.t. Veiklos taisyklės turi būti įgyvendintos pačioje organizacijoje. Jei taisyklių nesilaikoma, organizacijai kyla finansinių nuostolių pavojus ir, galbūt, net verslo praradimas [1 (1 dalis)].

Veiklos taisyklių (angl. business rules) apibrėžimų yra keletas:

1. Veiklos taisyklė yra teiginys, kuris apibrėžia arba veikia kai kuriuos veiklos aspektus, t.y. skirtas kontroliuoti verslo struktūrą ar veikti veiklos elgesį [2].
2. Veiklos taisyklė – būdas struktūrizuoti verslo žinias [3].
3. Veiklos taisyklė – taisyklė, kuri priklauso veiklai.[4].

2.2.2. Taisyklių klasifikacija pagal *GUIDE* veiklos taisyklių projektą

GUIDE veiklos taisyklių projektas pateikė apibendrintas veiklos taisyklių klasifikavimo bei užrašymo idėjas. *GUIDE* veiklos taisyklių projektas turėjo 4 konkrečius tikslus:

1. Apibrėžti ir apibūdinti veiklos taisykles ir susijusias sąvoka taip nustatant, kas yra ir kas nėra veiklos taisyklė.

2. Apibrėžti veiklos taisyklių abstraktų modelį, kad galėtų išreikšti (informacinių technologijų profesionalams suprantamais terminais), kas yra veiklos taisyklė ir kaip ji siejasi su informacinėmis sistemomis.
3. Nustatyti tikslų pagrindą veiklos taisyklėms išgauti iš esamų sistemų.
4. Nustatyti tikslų pagrindą naujoms sistemoms projektuoti, remiantis formaliais veiklos taisyklių apibrėžimais.

Pagal *GUIDE* veiklos taisyklių projektą, veiklos taisyklės yra keturių kategorijų:

- *Veiklos terminų apibrėžimas* (angl. Definitions of business tems)

Pagrindinis veiklos taisyklių elementas yra kalba, kuria jos yra išreiškiamos. Kiekvienos veiklos taisyklės užrašymas apibūdina, kaip žmogus galvoja ir kalba apie daiktus. Veiklos terminai dažnai vaizduojami esybių/ryšių, *UML* modeliuose.

- *Faktai susieja terminus tarpusavyje* (angl. Facts relating tems to each other)

Šios veiklos taisyklės yra kilusios iš organizacijos ar veiklos prigimties. Dažniausiai užrašomos natūralia kalba ar grafiniame modelyje vaizduojamos ryšiais, atributais.

- *Ribojimai* (angl. Constraints) dar vadinami struktūriniais (angl. Structural Assertions) arba veiksmų ribojimais (angl. Action assertions)

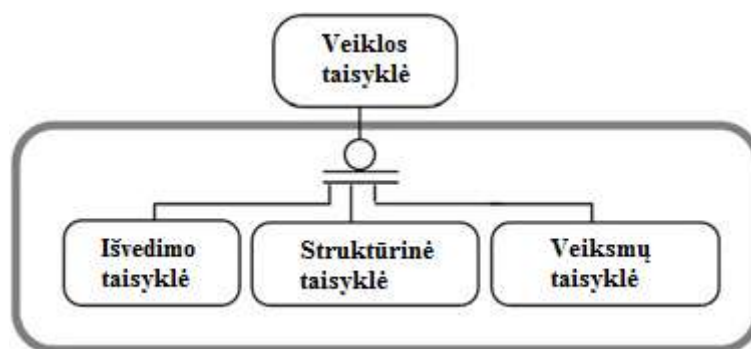
Kiekvienos įmonės struktūros ar veiksmų ribojimas tam tikru būdu, kuris yra susijęs su informacijos priėmimu, atnaujinimu.

- *Išvedimai* (angl. Derivations)

Veiklos taisyklės, kurios apibūdina, kaip vienas žinias galima gauti iš kitų žinių.

GUIDE veiklos taisyklių projektas pateikė veiklos taisyklių modelį, kuriame yra trys pagrindiniai veiklos taisyklių tipai (2.1 pav.):

- *Išvedimo* (angl. Derivation) *taisyklės* – išreiškia žinias, gaunamas iš kitų veiklos žinių.
- *Struktūrinės* (angl. Structural Assertion) *taisyklės* – apibūdina fakto idėją ar formulavimą struktūriškai.
- *Veiksmų* (angl. Action Assertion) *taisyklės* – išreiškia veiksmų ribojimus.



2.1 pav. Veiklos taisyklių tipai pagal *GUIDE* veiklos taisyklių projektą [2]

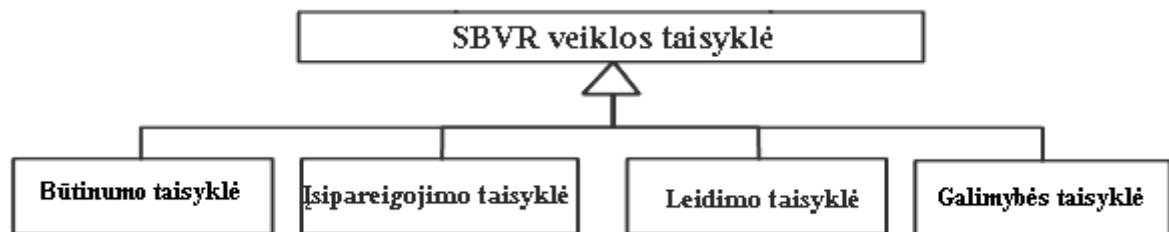
2.2.3. Taisyklių klasifikacija pagal SBVR

SBVR (angl. Semantics of Business Vocabulary and Business Rules) yra *OMG* (angl. Object Management Group) priimtas standartas, kurį sukūrė Business Rule Team konsorciumas. *SBVR* specifikacija apibrėžia veiklos žodyno (angl. business vocabulary), veiklos faktų tipų (angl. business fact types) ir veiklos taisyklių (angl. business rules) semantiką.

SBVR pateikia būdą užfiksuoti specifikacijas natūralia kalba ir išreikšti juos formalia logika taip, kad juos būtų galima apdoroti automatizuotai. *SBVR* numatyta išreikšti veiklos žodyną ir veiklos taisykles, ir specifikuoti veiklos reikalavimus informacinėms sistemoms natūralia kalba.

4 pagrindiniai veiklos taisyklių tipai pagal *SBVR* (2.2 pav.):

- *Būtinumo* (angl. necessity) *taisyklės* – modalinė formuluotė, kuri formuluoja, kad į ją įterpta loginė formuluotė turi būti teisinga visais galimais atvejais.
- *Įsipareigojimų* (angl. obligation) *taisyklės* – modalinė formuluotė, kuri formuluoja, kad į ją įterpta loginė formuluotė turi būti teisinga visais priimtinais atvejais.
- *Leidimo* (angl. permission) *taisyklės* – modalinė formuluotė, kuri formuluoja, kad į ją įterpta loginė formuluotė turi būti teisinga kai kuriais priimtinais atvejais.
- *Galimos* (angl. possibility) *taisyklės* – modalinė formuluotė, kuri formuluoja, kad į ją įterpta loginė formuluotė turi būti teisinga kai kuriais galimais atvejais.



2.2 pav. Veiklos taisyklių tipai pagal *SBVR* [4]

Pagal *SBVR* veiklos taisyklės yra sudaromos tokiomis sakinių pradžios formomis, kurios yra pateiktos 2.1 lentelėje. Šioje lentelėje taip pat matomi anglų ir lietuvių kalba išversti veiklos taisyklių pavyzdžiai.

2.1 lentelė. *SBVR* veiklos taisyklių forma

Sakinių forma	Pavyzdys (anglų kalba)	Pavyzdys (lietuvių kalba)
It is necessary that	It is necessary that amount of the loan is not greater than 10000.	Būtina, kad paskolos dydis neviršytų 10000.
It is obligatory that	It is obligatory that a bank checks validity of a loan.	Įsipareigojama, kad bankas tikrintų paskolos galiojimą.
It is permitted that	It is permitted that a debtor requests a loan.	Leidžiama, kad skolininkas prašytų paskolos.
It is possible that	It is possible that a debtor gets at most 3 loan.	Galima, kad skolininkas gautų daugiausia tris paskolas.

2.2.4. Taisyklių klasifikacija pagal *Witt*

Natūralios kalbos taisyklių sakinių praktinio metodo (A Practical Method of Developing Natural Language Rule Statements) straipsnius rašo *Graham Witt*. Jo straipsniuose aptariamas praktinis taisyklių, susijusių su natūralia kalba, formulavimo metodas (patvirtintas tarpininkų ir naudojamas tiek organizacijos viduje, tiek išorėje). Autorius pritaikė šį metodą didelei Australijos vyriausybės įstaigai, kuri pasirinko Veiklos taisyklių metodą ir *OMG's SBVR*, kaip geriausių taisyklių praktikos pavyzdžius.

Graham Witt sudarė veiklos taisyklių taksonomiją. Jo apžvelgti veiklos taisyklių tipai:

1. *Kardinalumo* (angl. cardinality rules) *taisyklės*: nurodo, kiek tam tikrų duomenų elementų yra reikalinga arba leidžiama tam tikrame kontekste.
 - 1.1. *Privalomų duomenų* (angl. mandatory data rules) *taisyklės*: vienas ar daugiau duomenų elementų yra privalomi tam tikrame kontekste.
 - 1.1.1. *Privalomų duomenų elementų* (angl. mandatory data item rules) *taisyklės*: turi būti tam tikras vienas duomenų elementas.
 - 1.1.2. *Privalomo pasirinkimo* (angl. mandatory option selection rules) *taisyklės*: turi būti nurodytas vienas pasirinkimas iš rinkinio.
 - 1.1.3. *Privalomos grupės* (angl. mandatory group rules) *taisyklės*: turi būti bent vienas duomenų elementas iš grupės.
 - 1.2. *Draudžiamų duomenų* (angl. prohibited data rules) *taisyklės*: nurodo, kad tam tikras duomenų elementas yra neleidžiamas tam tikrame kontekste.
 - 1.3. *Vienintelio duomenų elemento* (angl. singular data rules) *taisyklės*: tik vienas tam tikro duomenų elemento pavyzdys yra leidžiamas tam tikrame kontekste.
 - 1.4. *Priklausomo kardinalumo* (angl. dependent cardinality rules) *taisyklės*: duomenų elemento pavyzdžių skaičius priklauso nuo kokių nors kitų duomenų.

2. *Duomenų turinio* (angl. data content rules) *taisyklės*: duomenys turi būti tam tikrų reikšmių.
 - 2.1. *Reikšmių rinkinio* (angl. value set rules) *taisyklės*: duomenų elementas (arba duomenų elementų kombinacija) turi turėti reikšmę iš nustatyto rinkinio.
 - 2.2. *Atitikimo* (angl. match rules) *taisyklės*: nustato, kad tam tikras duomenų elementas turi būti toks kaip arba kitoks nei kuris nors kitas duomenų elementas.
 - 2.3. *Ribų* (angl. range rules) *taisyklės*: duomenų elementas (arba duomenų elementų kombinacija) turi turėti reikšmę tam tikrose ribose.
 - 2.4. *Unikalumo* (angl. uniqueness constraints) *reikalavimai*: duomenų elementas (arba elementų kombinacija) turi būti kitoks nei kiti tų pačių duomenų elemento pavyzdžiai.
3. *Proceso* (angl. process rules) *taisyklės*: proceso pradžia yra suvaržyta.
 - 3.1. *Procesų intervalų išdėstymo laike* (angl. inter-process interval rules) *taisyklės*: proceso pradžia yra varžoma kai, jis gali prasidėti prieš arba po kito tokio pat proceso prasidėjimo [1].

2.2.5. Veiklos taisyklių klasifikacijų apibendrinimas

Veiklos taisyklių klasifikacijų apibendrinimas pateiktas 2.2 lentelėje. Šioje lentelėje matomi visi nagrinėti šaltiniai, bei tų šaltinių veiklos taisyklių skirstymas į kategorijas.

2.2 lentelė. Veiklos taisyklių klasifikacijų apibendrinimas

Šaltinis	Kategorijos	
Guide Business Rules Project	Struktūrinė taisyklė	
	Veiksmo taisyklė	
	Išvedimo taisyklė	
Semantics of Business Vocabulary and Business Rules (SBVR)	Struktūrinė taisyklė	
	Operacinė taisyklė	
A Practical Method of Developing Natural Language Rule Statements (Witt)	Kardinalumo taisyklė	Privalomų duomenų
		Draudžiamų duomenų
		Vienintelio duomenų elemento
		Priklausomo kardinalumo
	Duomenų turinio taisyklė	Reikšmių rinkinio
		Atitikimo
		Ribų
		Unikalumo
Proceso taisyklė	Procesų intervalų išdėstymo laike	

2.3. Veiklos taisyklių specifikavimo šablonų analizė

Paprasčiausias veiklos taisyklių užrašymo būdas – natūrali kalba. Tačiau šis būdas tinkamas tik pačioje reikalavimų analizės pradžioje, nes yra neformalus ir jo naudojimas yra naudingas toje reikalavimų analizės būsenoje.

Viena didžiausių problemų dabartinėje informacinių sistemų inžinerijoje yra bendravimas tarp sistemos užsakovo ir kūrėjų. Dažniausiai iškyla problemos IS kūrime, kai reikalavimų analizės etape IS projektuotojas ne taip supranta užsakovo reikalavimus, o užsakovas patvirtina projektą, nors ir nesupranta IS projektuotojo sudaryto sistemos modelio technine kalba (pvz. *UML*).

Tokiai problemai spręsti buvo pasiūlytas veiklos taisyklių užrašymas natūralios kalbos šablonais. Nacionaline kalba (t.y. lietuvių) sukurti šablonai leidžia struktūrizuoti taisyklių užrašymą dalykinės srities žinovams suprantamesne kalba.

2.3.1. *RuleSpeak* veiklos taisyklių šablonų analizė

RuleSpeak naudoja anglų kalba suformuotas veiklos taisyklių sakinių formas, kurių struktūra pateikta 2.3 lentelėje. Jie ragina naudoti aiškų subjektą kiekvienos taisyklės sakinio pradžioje. Toks būdas suteikia sakiniui aiškumo ir pastovumo.

RuleSpeak šablonuose raktiniai žodžiai įpinti sakinio viduryje.

2.3 lentelė. *RuleSpeak* taisyklių sakinių formos

<i>RuleSpeak</i> taisyklių sakinių forma (anglų kalba)	<i>RuleSpeak</i> taisyklių sakinių forma (lietuvių kalba)
Each rental always has exactly one requested car group.	Kiekviena nuoma visada turi tik vieną mašinų grupę.
A driver of a rental must be qualified.	Nuomos vairuotojas turi būti kvalifikuotas.
The drop-off branch of a rental need not be the return branch of the rental.	Nuomos išvykimo šaka neturi būti nuomos grįžimo šaka.
The notification date/time of a bad experience that occurs during a rental is sometimes after the actual return date/time of the rental.	Pranešimo apie blogą nutikimą data/laikas kartais būna vėlesnis, negu faktinis nuomos grąžinimo data/laikas.

2.3.2. *SBVR* veiklos taisyklių šablonų analizė

SBVR naudoja anglų kalba suformuotas veiklos taisyklių sakinių formas, kurių struktūra pateikta 2.4 lentelėje. Kiekvienos taisyklės sakinio pradžioje aiškiai išskiriamas subjektas, kuris suteikia sakiniui aiškumo ir pastovumo.

SBVR šablonai prasideda raktiniais žodžiais, kurie leidžia iš karto nustatyti taisyklės tipą.

2.4 lentelė. SBVR taisyklių sakinių formos

SBVR taisyklių sakinių forma (anglų kalba)	SBVR taisyklių sakinių forma (lietuvių kalba)
It is necessary that each rental has exactly one requested car group.	Būtina, kad kiekvienas nuomos mokestis turėtų tik vieną mašinų grupę.
It is obligatory that each driver of a rental is qualified.	Įpareigojama, kad kiekvienas nuomojamas vairuotojas būtų kvalifikuotas.
It is permitted that the drop-off branch of a rental is not the return branch of the rental.	Leidžiama, kad nuomos išvykimo šaka nebūtų nuomos grįžimo šaka.
It is possible that the notification date/time of a bad experience that occurs during a rental is after the actual return date/time of the rental.	Galima, kad pranešimo apie blogą nutikimą data/laikas būtų vėlesnis, negu faktinis nuomos grąžinimo data/laikas.

Daugiau apie SBVR veiklos taisykles aprašyta 2.2.3 skyriuje „Taisyklių klasifikacija pagal SBVR“.

2.3.3. Witt veiklos taisyklių šablonų analizė

Angliški veiklos taisyklių sakinių šablonai pagal Witt „A Practical Method of Developing Natural Language Rule Statements“:

- „Privalomų duomenų elementų“, „Vienintelio duomenų elemento“ taisyklių pavyzdžiai:

Pradinės sakinio formos:

“Skrydžių užsakymo” procese turi būti privaloma išvykimo data.

“Skrydžių užsakymo” procesas turi atmesti bet kokią skrydžio užsakymą, kuriame nenurodyta grįžimo data.

Taisyklių sakiniai (RS):

(Taisyklės sakinyje yra sutrumpinta sakinio forma.)

RS1. Kiekviename skrydžio užsakymo lape turi būti nurodyta tik viena išvykimo data.

RS2. Kiekviename grįžtamojo skrydžio užsakymo prašyme turi būti nurodyta tik viena grįžimo data.

Taisyklių sakiniuose RS1 ir RS2 forma yra panaši, t.y. naudojama ta pati sintaksė, kuria galima lengviau patikrinti, ar nėra dubliavimo.

Taisyklių formuluočių yra paremtos faktų tipais (FT):

Taisyklių sakinyje RS1 paremtas:

FT1. skrydžio užsakymo prašyme yra nurodyta išvykimo data.

Taisyklių sakinyje RS2 paremtas:

FT2. skrydžio užsakymo prašymas yra skirtas kelionei.

FT3. atgalinė kelionė yra kelionės kategorija.

FT4. skrydžio užsakymo prašyme yra nurodyta grįžimo data.

FT6. skrydžio užsakymo prašymas yra skirtas kelionei atgal (išvestas iš FT2 ir FT3).

Taisyklių sakinių RS1 ir RS2 šablonas (RT):

RT1. Each <term 1> {<qualifying clause>|} must <verb phrase 1> <cardinality> <term 2>.

Kur: <term 1> and <term 2> yra įterpiniai tose vietose, kur terminai gali būti pakeisti; <qualifying clause> yra įterpinys vietoje, kurioje sakinio dalis gali būti pakeistas: viena sakinio dalies forma yra "that <verb phrase 2> a <term 3>", bet jei <verb phrase 2> prasideda „is“, „that is“ („yra“, „tai yra“), gali būti praleistas;

simboliai { | } leidžia alternatyvas šablone: šiame šablone gali pasirodyti arba <qualifying clause> (paruošiamasis sakinytis), arba nulis, t.y., <qualifying clause> yra laisvai pasirenkamas;

<verb phrase> (veiksmažodžio frazė) yra įterpinys vietoje, kur veiksmažodžio frazė gali būti pakeista: <term 1> <verb phrase> <term 2> turi būti fakto tipas;

<cardinality> (kardinalumas) yra įterpinys toje vietoje, kur kardinalumas gali būti pakeistas, pvz., tiksliai vienu, bent vienu, daugiausia vienu.

Pagalbinis šablonas (ST) formai <cardinality>:

ST1. <cardinality> ::= {exactly|at {least|most}|at least <positive integer> and at most} <positive integer>

Žinoma, galima naudoti ir kitokias išraiškas, tokias kaip: virš trijų (up to three), vienas arba du (either one or two), nuo vieno iki trijų imtinai (from one to three inclusive).

- „Draudžiamų duomenų“ taisyklės pavyzdys:

Pradinės sakinio forma:

“Skrydžių užsakymo” procesas turi atmesti skrydžio užsakymo prašymą, kuriame nurodyta grįžimo data skrydžiui į vieną pusę.

Taisyklės sakinytis (RS):

RS3. Skrydžio užsakymo prašyme kelionei į vieną pusę neturi būti nurodyta atgalinė data.

Taisyklės formuluoė yra paremta faktų tipais:

FT2. skrydžio užsakymo prašymas yra skirtas kelionei.

FT5. vienkryptė kelionė yra kelionės kategorija.

FT7. skrydžio užsakymo prašymas yra skirtas vienkryptei kelionei (išvestas iš FT2 ir FT5).

Taisyklės sakinio RS3 šablonas (RT):

RT6. {A|An} <term 1> {<qualifying clause>|} must not <verb phrase 1> {a|an} <term 2>.

Šiame šablone dažniau naudojamas nežymimasis artikelis ("A" or "An") nei "Each" (kiekvienas). Taip yra dėl to, kad nepatogu anglų kalba pasakyti "Kiekvienas...neturi..." (angl. "Each ... must not ...").

Pagalbinis šablonas (ST) formai <qualifying clause>:

ST6. <qualifying clause> ::= that <verb phrase> {a|an|each|that|the} <term> {<qualifying clause>|}

- „Aibės apribojimų“ operacinis pavyzdys:

Pradinė sakinio forma:

Neleidžiama apibrėžti skirtingų keleivių išvykstančiams ir sugrįžtančiams skrydžiams.

Taisyklės sakiny (RS):

(Taisyklės sakiny yra sutrumpinta sakinio forma.)

RS28. Keleiviai, rezervavę atgalinį skrydį skrydžių užsakymo patvirtinime turi būti tokie patys, kokie ir rezervavę išvykstančiame skrydyje.

Taisyklės sakinio RS28 šablonas (RT):

RT10. The set of <term 1> <qualifying clause> must be the same as the set of <term 2> <qualifying clause> {{if|unless} <conditional clause>|}.

Kadangi kiekvienas taisyklės tipas gali būti sąlygojamas, tai tada kiekvienam šablonui reikia pagalbinio sakinio. Šablonams pritaikęs pagalbinius sakinius, A Practical Method of Developing Natural Language Rule Statements, autorius *Graham Witt* sudarė šablonus, kurie pateikti 2.5 lentelėje.

2.5 lentelė. Taisyklių tipai, šablonai ir Metataisyklės

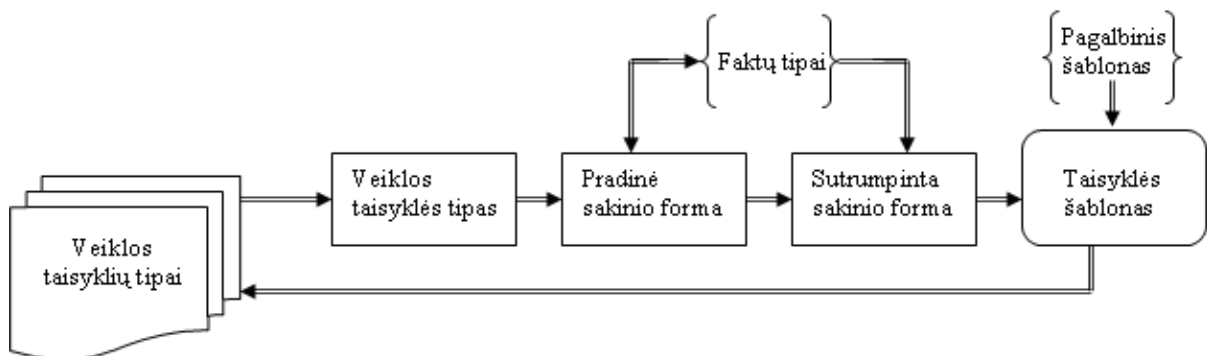
Taisyklės tipas		Šablonas	Šablono konstrukcija	Metataisyklės	
Kardinalumo taisyklės	Privalomų duomenų taisyklės	Privalomų duomenų elementų taisyklė	RT1	Each <term 1> {<qualifying clause>} must <verb phrase 1> <cardinality> <term 2>.	<cardinality> ::= {exactly <positive integer> at least <positive integer> {and at most <positive integer> }
		Privalomo pasirinkimo taisyklė	RT2	Each <term 1> {<qualifying clause>} must {{if unless} <conditional clause>} specify whether {{or not}} {it the <term 2>} <verb phrase 1> {<article> <term 3> } {it the <term 4>} <verb phrase 2> <or-list>.	
	Privalomos grupės taisyklė	RT3	Each <term 1> {<qualifying clause>} must <verb phrase> {a an} <term 2>, {a an} <term 3> or both {{if unless} <conditional clause> }.	if either or both of 2 items in the group (jei grupėje yra 2 elementai)	
		RT4	Each <term 1> {<qualifying clause>} must <verb phrase> {a an} <term 2> or	if only 1 of 2 items in the group (jei grupėje yra tik 1 iš	

			{a an} <term 3> but not both { {if unless} <conditional clause> }.	2 elementų)	
		RT5	Each <term 1> { <qualifying clause> } must { {if unless} <conditional clause> } <verb phrase> <cardinality> of the following: <or-list>.	if more than 2 items in the group (jei grupėje yra daugiau nei 2 elementai)	
	Draudžia- mų duomenų taisyklė	RT6	{A An} <term 1> { <qualifying clause> } must not <verb phrase> {a an} <term 2> { {if unless} <conditional clause> }.		
	Vienintelio duomenų elemento taisyklė	RT14	{ {The Each} <term 1> { (if any) } } that <verb phrase 1> } each <term 2> { <qualifying clause> } must <verb phrase 2> <cardinality> <term 3> { <preposition> { each the } <term 4> } { {if unless} <conditional clause> }.	<cardinality> ::= { exactly at most } one	
	Priklauso mo kardinalu mo taisyklė	RT10	The <set function> of {the } <term 1> <qualifying clause> must be { <non-time range predicate> <equality predicate> } { {if unless} <conditional clause> }.	<set function> ::= number	
Duomenų turinio taisyklės	rinkinio	Reikšmių taisyklė	RT7	The <term 1> { <qualifying clause> } { (if any) } that <verb phrase> each <term 2> { <qualifying clause> } must be <predicate> { {if unless} <conditional clause> }.	<predicate> ::= <value predicate> if only 1 data item (jei tik 1 duomenų elementas)
			RT8	The combination of <simple and-list> { <qualifying clause> } { (if any) } that <verb phrase> each <term 2> { <qualifying clause> } must be <combination predicate> { {if unless} <conditional clause> }.	<combination predicate> ::= <combination value set predicate> if combination of data items (jei duomenų elementų kombinacija)
	Atitikimo taisyklė	Operacinis	RT7	The <term 1> { <qualifying clause> } { (if any) } that <verb phrase> each <term 2> { <qualifying clause> } must be <predicate> { {if unless} <conditional clause> }.	<predicate> ::= <match predicate>
			Struktūrinis	RT9	The <term 1> { <qualifying clause> } { (if any) } that <verb phrase> each <term 2> { <qualifying clause> } is by definition <predicate> { {if unless} <conditional clause> }.
	Ribų taisyklė	RT7		The <term 1> { <qualifying clause> } { (if any) } that <verb phrase> each <term 2> { <qualifying clause> } must be <predicate> { {if unless} <conditional clause> }.	<predicate> ::= <range predicate>
	Unikalumo reikalavimas		RT7	The <term 1> { <qualifying clause> } { (if any) } that <verb phrase> each <term 2> { <qualifying clause> } must be <predicate> { {if unless} <conditional clause> }.	<predicate> ::= <uniqueness predicate> if only 1 data item (jei tik vienas duomenų elementas)
			RT8	The combination of <simple and-list> { <qualifying clause> } { (if any) } that <verb phrase> each <term 2> { <qualifying clause> } must be <combination predicate> { {if unless} <conditional clause> }.	<combination predicate> ::= <combination uniqueness predicate> if combination of data items (jei duomenų elementų kombinacija)

	Aibės apribojimai	Operacinis	RT11	The set of <term 1> <qualifying clause> must be the same as the set of <term 2> <qualifying clause> { {if unless} <conditional clause> }.	
		Struktūrinis	RT12	The set of <term 1> <qualifying clause> is by definition the same as the set of <term 2> <qualifying clause> { {if unless} <conditional clause> }.	
Proceso taisyklės	Procesų intervalų išdėstymo laike taisyklė		RT14	Each <term 1> <qualifying clause> must <verb phrase> {<determiner> <term 2> {<qualifying clause> }} <time range predicate> { {if unless} <conditional clause> }.	

Apibendrinta veiklos taisyklių šablonų sudarymo schema pavaizduota 2.3 paveiksle.

1. Iš visų esamų veiklos taisyklių tipų pasirenkamas vienas veiklos taisyklės tipas;
2. Sudaroma pradinė sakinio forma (ji sudaroma remiantis jau esamais faktų tipais arba sukuriant naujus faktų tipus);
3. Iš pradinės sakinio formos sudaromas taisyklės sakiny, t.y. sutrumpinta sakinio forma (remiantis faktų tipais yra suformuluojama taisyklė);
4. Remiantis veiklos taisyklės sakiniu, sudaromas taisyklės šablonas;
5. Jei reikia, taisyklės šablonui pritaikomas pagalbinis šablonas.
6. Taisyklės šablonas yra pritaikomas vienam veiklos taisyklės tipui ar keliems veiklos taisyklių tipams.



2.3 pav. Veiklos taisyklių šablonų sudarymo schema

Autoriaus *Witt* šablonai yra skirti detalesniems taisyklių tipams, pagal kuriuos galima specifikuoti kardinalumą, reikšmių, loginius ir kt. apribojimus. Tačiau pagal autoriaus rekomendacijas yra gaunama daug perteklinių taisyklių.

2.3.4. Esamų taisyklių specifavimo metodų palyginimas

Teisingas veiklos taisyklių užrašymas yra problema: sunku užfiksuoti veiklos taisykles taip, kad jos būtų vienareikšmiškai suprantamos, teisingos ir lengvai realizuojamos programinėje įrangoje. Yra sukurtų šablonų, kurie leidžia struktūrizuoti veiklos taisyklių užrašymą, tačiau jie skirti anglų kalbai.

R. Ross buvo pirmasis žmogus, pasiūlęs visapusišką taisyklių dokumentavimo metodą, diagraminio vaizdavimo techniką, kuri galbūt buvo per sudėtinga, kad iš tikrųjų būtų palanki verslui. Tuomet jis išrado *RuleSpeak*, kuris buvo pirmosios pastangos link taisyklių natūralia kalba.

Terry Halpin tuo tarpu dirbo su *ORM* (angl. Object Role Modeling models) ir aprašė metodą, išreiškiantį duomenų modelį (ir susijusias taisykles) natūralios kalbos sakiniiais. 2008m sausį *OMG* (angl. Object Management Group) išleido 1.0 versiją *SBVR*. Nors *SBVR* turi, autoriaus manymu, kelis trūkumus, jis reikšmingas, nes jame daug kalbinių logiškų sąvokų, sudarančių verslo taisyklių pagrindą, ir „taisyklių sakiniai“ (*Graham Witt* šį metodą pritaikė didelei Australijos vyriausybės įstaigai), t.y. jų terminija natūralia kalba. Sudėtingumas, būdingas šiai analizei, yra iššūkis specialistams, ypač ne taip gerai išmanantiems kalbos ir logikos teoriją [1].

RuleSpeak ir *SBVR* naudoja anglų kalba suformuotas veiklos taisyklių sakinių formas, kurių struktūra yra šiek tiek skirtinga, tačiau abiem atvejais kiekvienos taisyklės sakinio pradžioje aiškiai išskiriamas subjektas, kuris suteikia sakiniui aiškumo ir pastovumo. *SBVR* šablonai yra patogesni realizuoti kompiuteriu, kadangi jie prasideda raktiniais žodžiais, leidžiančiais iš karto nustatyti taisyklės tipą. *RuleSpeak* šablonuose raktiniai žodžiai įpinti sakinio viduryje. Ir vieni, ir kiti šablonai neturi elementų, kurių reikia specifikuojant detalius reikšmių ribojimus, formatus ir pan. *Witt* rekomendacijos pasižymi didesniu detalumu, tačiau kompiuterizuoti tinkamesnė *SBVR* sakinių forma.

RuleSpeak, *SBVR* ir *Witt* metoduose egzistuojančių šablonų palyginimas pateiktas 2.6 lentelėje.

2.6 lentelė. Metoduose egzistuojančių šablonų palyginimas

Metodai	Šablonai	Struktūrinių taisyklių	Išvedimo taisyklių	Operacinių taisyklių	Procesų	Kardinalumo ir unikalumo	Duomenų turinio	Duomenų formatų	Skaičiavimų	Detalumo laipsnis	Tinkamumas apdoroti kompiuteriu
<i>RuleSpeak</i>		+	+		+					vidutinis	vidutinis
<i>SBVR</i>		+		+						vidutinis	didelis
<i>Witt</i>					+	+	+			didelis	vidutinis
Kuriamas metodas		+	+	+	+	+	+	+	+	didelis	didelis

Atlikus lyginamąją analizę, buvo nutarta formuluoti lietuviškus šablonus remiantis *SBVR* stiliumi ir papildyti juos *Witt* rekomendacijomis, kad šablonai būtų lengvai kompiuterizuojami ir reikiamo detalumo lygio.

2.4. Siekiamas sprendimas

Šiame darbe siekiama sukurti lietuviškus veiklos taisyklių specifikuojimo šablonus ir metodiką, kuria galėtų naudotis veiklos analitikai, rašydami veiklos taisykles, ir kūrėjai, realizuodami veiklos taisykles.

2.4.1. Darbo formuluotė

Darbui reikalingi du vartotojai: analitikas ir realizuotojas. Kiekvienas vartotojas atlieka skirtingas funkcijas. Analitikas specifikuoja veiklos žodyną ir taisykle, apibrėžia jų ypatumus. Realizuotojas, naudodamasis veiklos taisyklėmis, kuria *SQL* trigerius ir procedūras. Abu vartotojai privalo būti savo srities specialistai, t.y. informatikai.

Vartotojų tikslus geriausiai atspindi 2.4 ir 2.5 paveikslėliai, kuriuose pavaizduotos panaudojimo atvejų diagramos.

Analitiko panaudojimo atvejų diagramoje yra matoma tai, kad jis turi specifikuoti veiklos žodyną ir taisykles.



2.4 pav. Analitiko panaudojimo atvejų diagrama

Šiame etape yra žinoma tik tai, kad realizuotojas turi kurti *SQL* trigerius ir procedūras iš veiklos taisyklių.



2.5 pav. Realizuotojo panaudojimo atvejų diagrama

Šiuo metu vartotojams esančios problemos yra pateiktos 2.7 lentelėje.

2.7 lentelė. Vartotojų problemos

Problema	Kaip viskas vyksta dabar
Nėra aiškios metodikos, kaip rašyti veiklos taisykles.	Analitikai siekia užrašyti veiklos taisykles analizės etape, tačiau nėra formalių priemonių, kaip jas užrašyti, dėl to daugelis taisyklių lieka neužfiksuotos arba užrašomos nevienareikšmiškai.
Veiklos taisyklės dažnai yra suformuluojamos tik realizavimo metu.	Daugelį taisyklių užregistruoja tik realizuotojai, kurie realizuoja veiklos taisykles DB priemonėmis, todėl sunku atsekti, kaip programinė realizacija atitinka reikalavimus.

Vartotojams, norėdami išspręsti šias problemas, reikia:

- sudaryti metodiką ir šablonus;
- tiksliai specifikuoti veiklos taisykles surinkimo metu;
- išnagrinėti veiklos taisyklių realizavimą DB sistemose;
- parengti *SQL* šablonus, kurie realizuotų tam tikrus veiklos taisyklių tipus.

2.4.2. Taisyklių realizavimo technologijos pasirinkimas ir analizė

Šiame darbe veiklos taisyklės bus realizuojamos *SQL* trigeriais ir procedūromis.

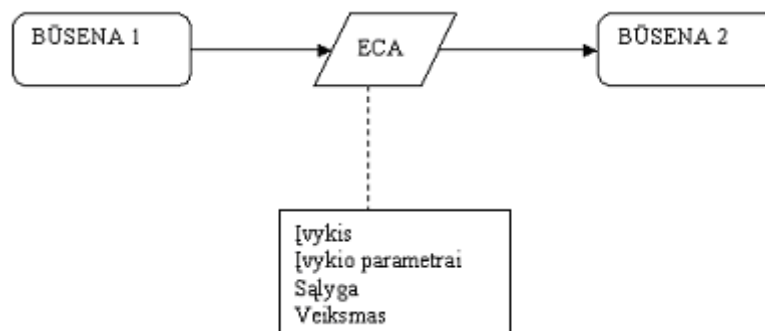
Kalbant programuotojų kalba, trigeris (angl. trigger – iššaukti, sukelti) – tai speciali procedūra, kuri susiejama su lentele ir iškviečiama įvykus konkrečiam įvykiui. Duomenų bazėse tokie įvykiai gali būti trys:

- INSERT – vienos arba kelių eilučių įterpimas į lentelę
- UPDATE – vienos arba kelių eilučių atnaujinimas lentelėje
- DELETE – vienos arba kelių eilučių trynimasis iš lentelės

Kiekvienas trigeris gali reaguoti į vieną iš šių įvykių arba į visus įvykius vienu metu. Be to, prie kiekvienos lentelės galima prikaltinti kelis trigerius, ir jeigu jie reaguoja į tuos pačius įvykius, galima nustatyti jų vykdymo eiliškumą.

Trigerį galima nagrinėti kaip *ECA* taisyklės realizaciją, kurios pagrindinės dalys yra: įvykis (angl. Event), sąlyga (angl. Condition) ir veiksmas (angl. Action). *ECA* – tai taisyklės, kurias galima aprašyti naudojant schemą: JEIGU <įvykis> IR <sąlyga> TAI <veiksmas>. *ECA* taisyklės galima modeliuoti modifikuotomis būsenų perėjimo diagramomis (2.6 pav.)

[6]. Triggeris, kuris reaguoja į kurį nors nustatytą įvykį ir priklausomai nuo tikrinimo sąlygos rezultato atliekamas veiksmas. Galimi trys įvykio variantai: eilutės įterpimas (INSERT), eilutės trynimasis (DELETE) arba eilutės atnaujinimas (UPDATE). Jeigu įvykio nustatytas įvykis, yra tikrinama sąlyga: tai gali būti įrašytų ar ištrintų duomenų patikrinimas, duomenų iš kitų lentelių patikrinimas ir t.t. Priklausomai nuo sąlygos patikrinimo rezultato atliekamas veiksmas: tai gali būti bet kokia *SQL* kalbos komanda: transakcijos atšaukimas, duomenų pakeitimas tiek bazinėje lentelėje, tiek bet kurioje kitoje lentelėje.



2.6 pav. *ECA* taisyklių vaizdavimo būdas būsenų perėjimo diagramomis [6]

MS SQL Server 2008 triggerio (*T-SQL* [5]) sintaksė [9] apibrėžiama taip:

```
CREATE TRIGGER [ schema_name . ]trigger_name
ON table
[ WITH <dml_trigger_option> [ ...,n ] ]
AFTER
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
[ NOT FOR REPLICATION ]
AS { sql_statement [ ...n ] }
```

kur:

- `trigger_name` – nurodo triggerio vardą, kuriuo jis identifikuojamas duomenų bazėje;
- `table` – nurodo lentelėje pavadinimą, kurioje veikia triggeris;
- `<dml_trigger_option>` - leidžia nustatyti `ENCRYPTION` ar/ir `EXECUTE AS` sakinius. `ENCRYPTION` užkoduoja *T-SQL* triggerio apibrėžimą ir padaro jį nematomą sisteminėse lentelėse. `EXECUTE AS` leidžia nustatyti triggerio saugumo kontekstą, kuriame tas triggeris bus vykdomas.
- `[INSERT] [,] [UPDATE] [,] [DELETE]` – įvykiai, kuriems įvykus aktyvuojamas triggeris. Galima nurodyti nuo vieno iki trijų įvykių;
- `[NOT FOR REPLICATION]` – jei prieš tai yra įvykdytas veiksmas lentelėje, tai triggeris neturi būti įvykdytas. Žymi, kad triggerio negalima vykdyti kai lentelei vykdomas replikavimo modifikavimas.
- `sql_statement [...n]` – (*T-SQL* kalbos komanda, -os) veiksmas, kuris vykdomas patikrinus sąlygą.

Veiklos taisyklių sistemai kurti pasirinkta *Microsoft SQL Server 2008* programa. Ši programa pasirinkta dėl to, kad šiuo metu ji yra naujausia *Microsoft SQL Server* programa.

„*Microsoft SQL Server 2008*“ - tai patikima, efektyvi ir intelektualiai duomenų bazių (angl. DataBase) platforma, kuri leis vykdyti daugiausia išteklių reikalaujančias užduotis, sumažinti programų kūrimui ir valdymui skiriamą laiką bei išlaidas ir geriau suvokti visos organizacijos poreikius [7].

„*SQL Server 2008*“ yra svarbi produkto versija, pateikianti daug naujų funkcijų ir svarbių patobulinimų. Šiuo metu tai yra kokybiškiausia ir išsamiausia *SQL* serverio versija.

SQL Server 2008 skiriasi nuo kitų versijų tuo, kad yra:

- a) platforma ypač svarbioms užduotims vykdyti;
- b) dinaminės programų kūrimo galimybės;
- c) daugiau nei tik sąryšiniai duomenys;
- d) gili verslo įžvalga [8].

2.5. Analizės išvados

1. Literatūros ir esamos IS kūrimo praktikos analizė parodė, kad teisingas veiklos taisyklių užrašymas sukelia problemų: sudėtinga užfiksuoti veiklos taisykles taip, kad jos būtų vienareikšmiškai suprantamos, teisingos ir lengvai realizuojamos programinėje įrangoje.
2. Literatūroje yra pateikiamas šios problemos sprendimas - šablonų taikymas, kuris leidžia struktūrizuoti veiklos taisyklių užrašymą.
3. *SBVR*, *RuleSpeak* ir *Witt* šablonų analizė parodė, kad jie visi panašūs, tačiau *SBVR* šablonų forma tinkamiausia apdoroti kompiuteriu. Tačiau *SBVR* šablonams trūksta detalumo, kuriuo pasižymi *Witt* šablonai, dėl to nutarta išplėsti *SBVR* šablonus detaliais *Witt* šablonų elementais.
4. Tolesnio darbo tikslas – pritaikyti išplėstus veiklos taisyklių šablonus lietuvių kalbai ir patikrinti šių šablonų tinkamumą realizuojant tipines veiklos taisykles programine įranga.
5. Veiklos taisyklėms realizuoti pasirinktos *DBVS* priemonės ir *SQL* trigeriai, o eksperimentinei sistemai kurti - *Microsoft SQL Server 2008* ir *Microsoft Visual Studio 2008*.

3. Veiklos taisyklių specifikuojimo metodika taikant *SBVR* natūralios kalbos šablonus

Šiame skyriuje aprašyti faktų tipų šablonai, lietuvių kalbai pritaikyti *SBVR* struktūrinių taisyklių šablonai ir jų transformavimo į DB schemą taisyklės, operacinių taisyklių šablonai ir jų vaizdavimas programiniais komponentais, taip pat pateikiamos metaduomenų lentelės, leidžiančios išsaugoti daugiau semantikos.

3.1. Metodikai keliami reikalavimai

Analitikas specifikuoja veiklos žodyną ir įvairias veiklos taisykles. 3.1 paveiksle matomi esami angliški *SBVR* veiklos taisyklių tipai ir naujai įvesti veiklos taisyklių tipai (duomenų turinio, duomenų formatų ir skaičiavimo taisyklės, kurios pažymėtos tamsesne spalva).

Specifikuojant veiklos žodyną, reikia užrašyti terminus ir faktų tipus. Apibrėžiant taisykles, reikia užrašyti operacines ir struktūrines veiklos taisykles. Struktūrinės taisyklės susideda iš: išvedimo, procesų, skaičiavimo ir ribojimų taisyklių. Detalesnis veiklos taisyklių sudarymas matomas analitiko panaudojimo atvejų diagramoje (3.1 pav.).



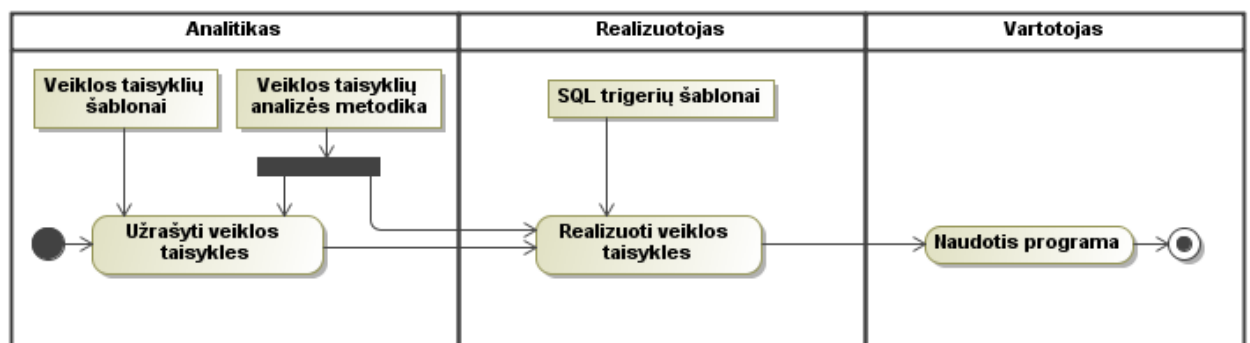
3.1 pav. Detalizuota analitiko panaudojimo atvejų diagrama

Detalizuota realizuotojo panaudojimo atvejų diagrama pavaizduota 3.2 paveiksle. Realizuotojas, iš veiklos faktų tipų ir taisyklių, kuria DB schemą, o iš veiklos taisyklių – *SQL* trigerius ir procedūras.



3.2 pav. Detalizuota realizuotojo panaudojimo atveju diagrama

3.3 paveiksle pavaizduotas procesas, kuris yra susijęs su veiklos taisyklių specifikuojimu ir realizavimo metodika. Šis procesas susideda iš trijų žingsnių: analitikas išanalizuoja ir užrašo veiklos taisykles, realizuotojas jas realizuoja, o vartotojas naudoja sukurta sistema.



3.3 pav. Veiklos taisyklių specifikuojimo ir realizavimo procesų diagrama

Šiame darbe taisyklėms formuluoti ir metodikai patikrinti bus kuriama eksperimentinė skrydžio užsakymo sistema.

3.2. Faktų tipų šablonai

Faktų tipų skirstomi į tokius šablonus:

1. Asociacijų faktų tipai ir jų šablonas:

```
{<terminas-tipas> <veiksmazodis {visi, išskyrus „priskirtas, priklauso“}> <terminas-tipas>}
```

FT30. Keleivis vežasi bagažą.

FT91. Draudimas gautas draudimo agentūroje.

FT93. Mieste veikia oro uostas.

2. Savybių faktų tipai ir jų šablonas:

```
{<terminas-tipas> <veiksmazodis {turi}> <terminas-savybė>}
```

FT13. Keleivis turi keleivio vardą.

FT17. Keleivis turi asmens kodą.

Fakto tipo FT17 sinoniminė forma: Asmens kodas priklauso keleiviui.

FT20. Bilietas turi numerį.

Fakto tipo FT20 sinoniminė forma: Numeris priskirtas bilietui.

3. Unariniai faktų tipai ir jų šablonas:

```
{<terminas-tipas> <veiksmazodis {yra_patikrintas | yra_patvirtintas | yra_nekorektiiskas}>}
```

FT5. Užsakytas skrydis yra patikrintas.

FT6. Užsakytas skrydis yra patvirtintas.

FT7. Užsakytas skrydis yra nekorektiškas.

4. Kategorizavimo faktų tipai, reiškiantys apibendrinimo - specializavimo ryšius. Jų šablonas:

```
{<terminas-tipas> <veiksmazodis {yra}> <terminas-tipas>}
```

FT48. Skrydžio užsakovas yra keleivis.

KFT80. Grįžtamas skrydis yra užsakytas skrydis.

5. Kategorizavimo faktų tipai, reiškiantys duomenų tipų priskyrimą atributams. Jų šablonai:

```
{<terminas-tipas> <veiksmazodis {yra}> <terminas-duomenų_tipas>}
```

KFT1. Išvykimo data yra data.

KFT4. Keleivio vardas yra tekstas.

KFT14. Amžius nuo yra realus skaičius.

Gali būti ir toks šablonas:

```
{<veiksmazodis> <veiksmazodis {yra}> <terminas-duomenų_tipas>}
```

KFT15. Yra patvirtintas yra boolean.

6. Egzempliorių faktų tipai, reiškiantys atributų reikšmių priskyrimą. Jų šablonas:

```
{<individas> <veiksmazodis {yra}> <terminas-savybė>}
```

FT54. Jaunuolis yra keleivio tipo vardas.

FT60. Dvipusis skrydis yra skrydžio tipo vardas.

FT74. Ekonominė klasė yra kelionės klasės tipas.

7. Agregavimo faktų tipai. Jų šablonas:

```
{<terminas-tipas> <veiksmazodis {susideda iš}> <terminas-tipas>}
```

KFT25. Maršrutas susideda iš reisų.

KFT26. Reisas susideda iš skrydžių.

8. Egzempliorių faktų tipai, reiškiantys objektų priklausomybę objektų tipams. Jų šablonas:

```
{<individas> <veiksmazodis {yra}> <terminas-tipas>}
```

KFT22. Vilnius yra miestas.

KFT24. Jonas yra keleivis.

9. Unikumo faktų tipai. Jų šablonas:

```
{<terminas-savybė> <veiksmazodis {yra}> <terminas-duomenų_tipas>}
```

KFT27. Skrydžio tipo vardas yra unikalus.

KFT28. Užsakymo numeris yra unikali seka.

KFT29. Asmens kodas yra unikali seka.

10. Reikšmių ribojimų faktų tipai. Jų šablonas:

```
{<terminas-duomenų_tipas> <veiksmazodis {turi}><individas><terminas-duomenų_tipas>}
```

KFT30. Tekstas turi @ simbolių.

KFT31. Tekstas turi 11 simbolių.

3.3. Veiklos taisyklių šablonai

Veiklos taisyklių šablonus sudaro struktūrinės (išvedimo, proceso, skaičiavimo ir ribojimo taisyklės: duomenų formatų, duomenų turinio, kardinalumo ir unikalumo) ir operacinės taisyklės.

- Struktūrinių veiklos taisyklių šablonai:

1. Išvedimo taisyklės. Jų šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmazodis {yra}> <individas> jei  
<terminas-tipas> <terminas-savybė> <nustatytas veiksmazodis {yra_mažesnis_nei |  
yra_ne_mažesnis_nei}> <individas>}
```

RS100. Privaloma, kad keleivis yra kūdikis, jei keleivio amžius yra mažesnis nei 2 metai.

RS101. Privaloma, kad keleivis yra suaugęs keleivis, jei keleivio amžius yra ne mažesnis nei 18 metų.

Gali būti ir toks šablonas:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmazodis {yra}> <individas> jei  
<terminas-tipas> <terminas-savybė> <nustatytas veiksmazodis {yra_ne_mažesnis_nei}>  
<individas> ir <nustatytas veiksmazodis {yra_mažesnis_nei}> <individas>}
```

RS102. Privaloma, kad keleivis yra jaunuolis, jei keleivio amžius yra ne mažesnis nei 2 metai ir yra mažesnis nei 18 metų.

2. Proceso taisyklės. Jų šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <nustatytas  
veiksmazodis {yra_ne_vėlesnė(-is)_nei | yra_ne_ankstesnė_nei}> <terminas-tipas> <terminas-  
savybė>}
```

RS107. Privaloma, kad skrydžio išvykimo data yra ne vėlesnė nei skrydžio atvykimo data.

RS110. Privaloma, kad reiso atvykimo laikas yra ne ankstesnė nei reiso išvykimo laikas.

Gali būti ir tokie šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-tipas> <terminas-savybė>  
<nustatytas veiksmožodis {yra_ne_ankstesnė(-is)_nei}> <terminas-tipas> <terminas-savybė>}
```

RS111. Privaloma, kad užsakyto skrydžio skrydžio išvykimo data yra ne ankstesnė nei užsakyto skrydžio užsakymo data.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-tipas> <terminas-savybė>  
<nustatytas veiksmožodis {yra_ne_ankstesnė(-is)_nei}> <terminas-tipas> <terminas-tipas>  
<terminas-tipas> <terminas-savybė>}
```

RS112. Privaloma, kad užsakyto skrydžio skrydžio išvykimo data yra ankstesnė nei užsakyto skrydžio grįžtamo skrydžio skrydžio grįžimo data.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-tipas> <terminas-tipas>  
<terminas-savybė> <nustatytas veiksmožodis {yra_ne_ankstesnė(-is)_nei}> <terminas-tipas>  
<terminas-savybė>}
```

RS113. Privaloma, kad užsakyto skrydžio skrydžio reiso faktinis išvykimo laikas yra ne ankstesnis nei faktinis užsakyto skrydžio užsakymo laikas.

3. Skaičiavimo taisyklės. Jų šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <veiksmožodis  
{skaičiuojamas_taip}> : <terminas-tipas> <terminas-tipas> <terminas-savybė> <veiksmožodis  
{minus}> <terminas-tipas> <terminas-savybė>}
```

RS95. Privaloma, kad keleivio amžius skaičiuojamas taip: keleivio užsakyto skrydžio užsakymo data minus keleivio gimimo data.

Gali būti ir tokie šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <veiksmožodis  
{skaičiuojama_taip}> : <terminas-tipas> <terminas-savybė> <veiksmožodis {padaugintas_iš}>  
<terminas-tipas> <terminas-savybė> jei <terminas-tipas> <terminas-savybė> <nustatytas  
veiksmožodis {yra_ne_didesnis_nei}> <individas>}
```

RS96. Privaloma, kad bagažo suminė kaina skaičiuojama taip: bagažo bagažo svoris padaugintas iš bagažo tipo svorio vieneto kainos, jei bagažo maksimalus svoris yra ne didesnis nei 20 kg.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <veiksmožodis  
{skaičiuojama_taip}> : <terminas-tipas> <terminas-savybė> <veiksmožodis {minus}>  
<terminas-tipas> <terminas-savybė> <veiksmožodis {padaugintas_iš}> <terminas-tipas>  
<terminas-savybė> jei <terminas-tipas> <terminas-savybė> <nustatytas veiksmožodis  
{yra_ne_didesnis_už}> <terminas-tipas> <terminas-savybė>}
```

RS97. Privaloma, kad bagazo suminė kaina skaičiuojama taip: bagazo bagazo svoris minus bagazo tipo maksimalus svoris padaugintas iš bagazo tipo papildomos kainos, jei bagazo bagazo svoris yra ne didesnis už bagazo tipo maksimalų svorį.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <veiksmazodis {skaičiuojama_taip}> : <veiksmazodis {suma}>, kuri <veiksmazodis {susideda iš}> <terminas-tipas> <terminas-tipas> <terminas-savybė>}
```

RS98. Privaloma, kad keleivio visų bagažų kaina skaičiuojama taip: suma, kuri susideda iš keleivio bagažo suminių kainų.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <veiksmazodis {skaičiuojama_taip}> : <terminas-tipas> <terminas-savybė> <veiksmazodis {padauginta iš}> <terminas-tipas> <terminas-savybė> <veiksmazodis {plius}> <terminas-tipas> <terminas-savybė>}
```

RS99. Privaloma, kad užsakyto skrydžio suminė kaina skaičiuojama taip: skrydžio kaina padauginta iš kelionės klasės koeficiento plus keleivio visų bagažų kaina.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <veiksmazodis {skaičiuojamas_taip}> : <terminas-tipas> <terminas-savybė> <veiksmazodis {sujungiama su}> <terminas-tipas> <terminas-savybė>}
```

RS105. Privaloma, kad reiso faktinis išvykimo laikas skaičiuojamas taip: skrydžio išvykimo data sujungiama su reiso išvykimo laiku.

4. Ribojimų taisyklės:

4.1. Duomenų formatų taisyklės. Jų šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-savybė> <veiksmazodis {yra}> <terminas-duomenų_tipas>}
```

7. Privaloma, kad telefonas yra tekstas.

16. Privaloma, kad amžius nuo yra realus kaičius.

Gali būti ir tokie šablonai:

```
{<raktinis žodis {privaloma, kad}> <veiksmazodis> <veiksmazodis {yra}> <terminas-duomenų_tipas>}
```

18. Privaloma, kad yra patvirtintas yra boolean.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <terminas-duomenų_tipas> <veiksmazodis {turi}> <individas> <terminas-duomenų_tipas>}
```

1. Privaloma, kad keleivio asmens kodo tekstas turi 11 simbolių.

```
{<raktinis žodis {privaloma, kad}> <terminas-savybė> <terminas-savybė> <veiksmazodis {turi}> <individas> <terminas-duomenų_tipas>}
```

2. Privaloma, kad el pašto tekstas turi @ simbolį.

4.2. Duomenų turinio taisyklės. Jų šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmazodis {turi}> <raktinis žodis {viena_arba_daugiau_iš_aibės}> <terminas-savybė> <terminas-savybė>}
```

RS90. Privaloma, kad užsakytas skrydis turi vieną arba daugiau iš aibės: telefonas, el paštas.

Gali būti ir tokie šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmazodis {turi}> <raktinis žodis {viena_iš_aibės}> <terminas-tipas> <terminas-tipas>}
```

RS91. Privaloma, kad apmokėtas skrydis turi vieną iš aibės: elektroninis pervedimas, kreditinė kortelė.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <veiksmazodis {yra}> <raktinis žodis {vienas_iš_aibės}> <individas> <individas> <individas>}
```

RS94. Privaloma, kad keleivio tipo keleivio tipo vardas yra vienas iš aibės: suaugęs keleivis, jaunuolis, kūdikis.

4.3. Kardinalumo taisyklės. Jų šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmazodis {turi}> <raktinis žodis {tiksliai_vieną}> <terminas-savybė>}
```

RS1. Privaloma, kad reisas turi tiksliai vieną reiso numerį.

RS2. Privaloma, kad reisas turi tiksliai vieną išvykimo laiką.

Gali būti ir tokie šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmazodis {apmokėta (-as) | vykdoma (-as) | apibūdinama (-as) | matuojama (-as) | gauta (-as) | užsakyta (-as) | susijęs su | veikia | išvyksta iš | atvyksta į | vežasi }> <raktinis žodis {tiksliai_viena (-as) | tiksliai_vieno | tiksliai_viename | tiksliai_vienu | tiksliai_vienam | tiksliai_vieną}> <terminas-tipas>}
```

RS56. Privaloma, kad užsakytas skrydis apibūdinamas tiksliai viena kelionės klase.

RS57. Privaloma, kad užsakytas skrydis susijęs su tiksliai vienu skrydžiu.

RS63. Privaloma, kad apmokėtas skrydis apmokėtas tiksliai vieno skrydžio užsakovo.

```
{<raktinis žodis {privaloma, kad}> <individas> <veiksmazodis {turi}> <raktinis žodis {tiksliai_vieną | du}> <terminas-tipas>}
```

RS103. Privaloma, kad vienpusis skrydis turi tiksliai vieną užsakytą skrydį.

RS104. Privaloma, kad dvipusis skrydis turi du užsakytus skrydžius.

{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmožodis {priskirtas}> <raktinis žodis {bent_vienas}> <terminas-tipas> }

RS88. Privaloma, kad reisui priskirtas bent_vienas skrydis.

RS89. Privaloma, kad užsakytam skrydžiui priskirtas bent_vienas keleivis.

{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmožodis {priskirta (-as) | susijęs_su | vežasi}> <raktinis žodis {ne_daugiau_kaip_vienas (-as) | ne_daugiau_kaip_vienu | ne_daugiau_kaip_3}> <terminas-tipas> }

RS82. Privaloma, kad bilietai priskirtas ne_daugiau_kaip_vienas registruotas skrydis.

RS83. Privaloma, kad keleivis vežasi ne_daugiau_kaip_3 bagažus.

RS87. Privaloma, kad apmokėtas skrydis susijęs_su ne_daugiau_kaip_vienu užsakytu skrydžiu.

4.4. Unikumo taisyklės. Jų šablonai:

{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <veiksmožodis {yra}> <terminas-duomenų_tipas> }

19. Privaloma, kad užsakyto skrydžio užsakymo numeris yra unikali seka.

Gali būti ir toks šablonas:

{<raktinis žodis {privaloma, kad}> <terminas-savybė> <veiksmožodis {yra}> <terminas-duomenų_tipas> }

20. Privaloma, kad skrydžio tipo vardas yra unikalus.

- Operacinių veiklos taisyklių šablonai:

1. Leidimo taisyklės. Jų šablonai:

{<raktinis žodis {leidžiama, kad}> <terminas-tipas> <veiksmožodis {prisijungia_prie | sukuria | patvirtina | gauna}> <terminas-tipas>, jei <terminas-tipas> <veiksmožodis {yra | sukuria}> <terminas-tipas> }

RS116. Leidžiama, kad keleivis sukuria registruotą skrydį jei keleivis yra prisijungęs vartotojas.

RS118. Leidžiama, kad skrydžio užsakovas patvirtina užsakytą skrydį, jei skrydžio užsakovas sukuria užsakytą skrydį.

Gali būti ir toks šablonas:

{<raktinis žodis {leidžiama, kad}> <terminas-tipas> <veiksmožodis {prisijungia_prie | sukuria | patvirtina | gauna}> <terminas-tipas> }

RS124. Leidžiama, kad keleivis sukuria registruotą skrydį.

2. Įsipareigojimo taisyklės. Jų šablonai:

{<raktinis žodis {įpareigojama, kad}> <terminas-tipas> <veiksmazodis {įveda | tikrina | nurodo | patvirtina}> <terminas-tipas> , jei <terminas-tipas> <veiksmazodis {patvirtina| yra | sukuria}> <terminas-tipas>}

RS128. Įpareigojama, kad skrydžio užsakymo sistema tikrina registruotą skrydį, jei keleivis patvirtina registruotą skrydį.

RS134. Įpareigojama, kad keleivis nurodo apmokėtą skrydį, jei keleivis sukuria registruotą skrydį.

Gali būti ir toks šablonas:

{<raktinis žodis {įpareigojama, kad}> <terminas-tipas> <veiksmazodis {įterpia | tikrina | nurodo | patvirtina}> <terminas-tipas>}

RS138. Įpareigojama, kad skrydžio užsakymo sistema tikrina registruotą skrydį.

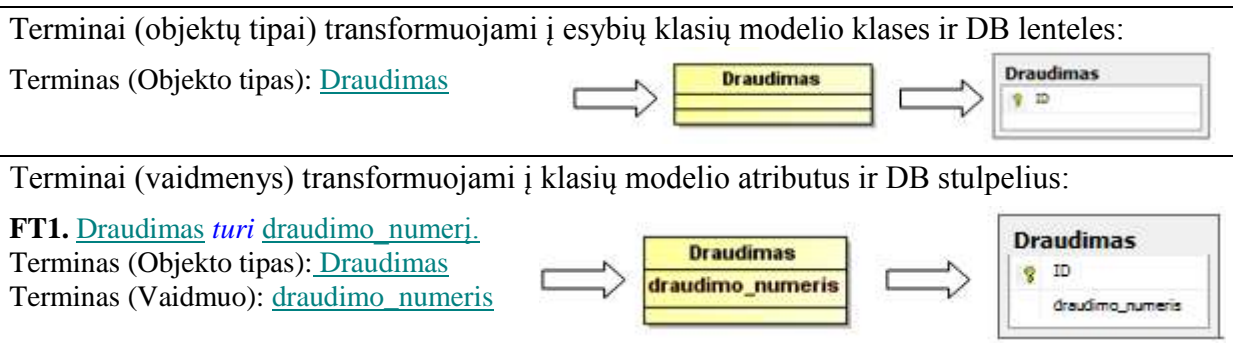
RS139. Įpareigojama, kad keleivis nurodo apmokėtą skrydį.

3.4. SBVR konceptų transformavimas į DB schemą

Šiame poskyryje aprašytas SBVR konceptų transformavimas į DB schemą: veiklos taisyklių ir faktų tipų, kardinalumo ir atributų reikšmių ribojimų transformacija, struktūrinių veiklos taisyklių transformavimas į DB lenteles bei DB schemas semantikos išplėtimas.

3.4.1. Terminų transformavimas

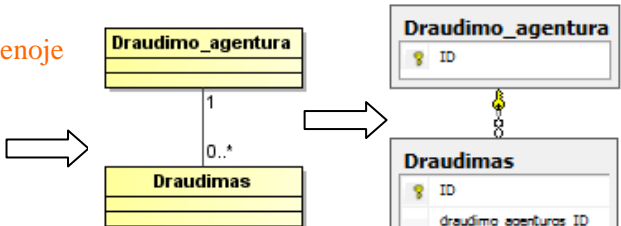
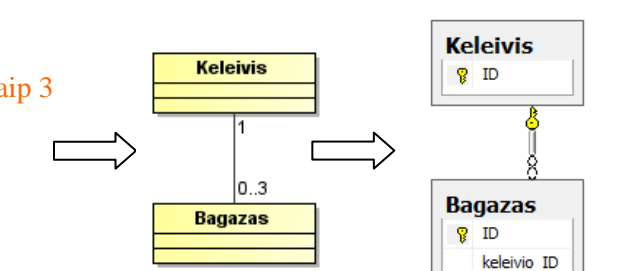
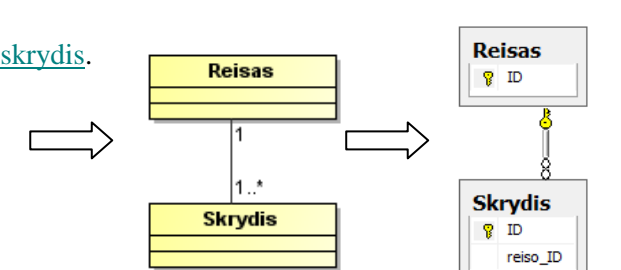
Terminų (objektų tipų) transformavimas į DB lenteles skiriasi nuo transformavimo į esybių klasių modelį tuo, kad lentelėms reikia sukurti pirminius raktus (ID stulpelius), be to, vaidmenims reikia aprašyti terminus, vaizduojančius duomenų tipus (pvz., [tekstas](#), [sveikas skaičius](#) ir t.t.) bei formatus (pvz., [teksto](#), [simbolių skaičių](#)).



3.4.2. Faktų tipų ir kai kurių veiklos taisyklių transformavimas

Faktų tipams reikia sukurti išorinius raktus, todėl kartu reikia nagrinėti ir taisyklės, išreiškiančias kardinalumo ribojimus.

Struktūrinės taisyklės, reiškiančios kardinalumo ribojimus, transformuojamos į klasių modelio kardinalumo ribas:

<p>DB schemoje 1:0..* kardinalumai vaizduojami privalomu išoriniu raktu:</p> <p>RS1. Privaloma, kad <u>draudimas gautas tiksliai vienoje draudimo agentūroje</u>.</p> <p>FT1. <u>Draudimas gautas draudimo agentūroje</u>.</p> <p>Terminas (Objekto tipas): <u>Draudimas</u></p> <p>Terminas (Objekto tipas): <u>Draudimo agentūra</u></p> <p>Asociacijos ryšys</p>	
<p>DB schemoje 1:0..1 kardinalumai vaizduojami trigeriais. Pavyzdys pateiktas DB schemoje 1:0..3 kardinalumu:</p> <p>RS1. Privaloma, kad <u>keleivis vežasi ne daugiau kaip 3 bagažus</u>.</p> <p>FT1. <u>Keleivis vežasi bagažą</u>.</p> <p>Terminas (Objekto tipas): <u>Bagažas</u></p> <p>Terminas (Objekto tipas): <u>Keleivis</u></p> <p>Asociacijos ryšys</p>	
<p>DB schemoje 1:1..* kardinalumai vaizduojami trigeriais ir procedūromis:</p> <p>RS1. Privaloma, kad <u>reisui priskirtas bent vienas skrydis</u>.</p> <p>FT1. <u>Reisui priskirtas skrydis</u>.</p> <p>Terminas (Objekto tipas): <u>Reisas</u></p> <p>Terminas (Objekto tipas): <u>Skrydis</u></p> <p>Asociacijos ryšys</p>	
<p>Unariniai faktų tipai realizuojami stulpeliais, kurių tipas yra <i>boolean</i>. Bendros taisyklės yra taikomos visiems unariniais faktų tipams. Pavyzdžiui:</p> <p>RS1. Privaloma, kad <u>yra patikrintas, yra patvirtintas, yra nekorektiškas – unariniai fakto tipai, kurių tipas yra „boolean“, o pradinė reikšmė yra „false“</u>.</p> <p>Bendra RS1 taisyklė užrašoma remiantis RS2-RS6 taisyklėmis:</p> <p>RS2. Galima, kad <u>objekto tipas turi pradinę reikšmę</u>.</p> <p>FT1. <u>Objekto tipas turi pradinę reikšmę</u>.</p> <p>Terminas (Objekto tipas): <u>Objekto tipas</u></p> <p>Terminas (Vaidmuo): <u>Pradinė reikšmė</u></p> <p>RS3. Privaloma, kad <u>unarinis fakto tipas turi pradinę reikšmę</u>.</p> <p>FT2. <u>Unarinis fakto tipas turi pradinę reikšmę</u>.</p> <p>Terminas (Objekto tipas): <u>Unarinis fakto tipas</u></p> <p>Terminas (Vaidmuo): <u>Pradinė reikšmė</u></p>	

RS4. Privaloma, kad unarinio fakto tipo pradinė reikšmė lygi „false“.

RS5. Privaloma, kad unarinis fakto tipas turi tipą.

FT1. Unarinis fakto tipas turi tipą.

Terminas (Objekto tipas): Unarinis fakto tipas

Terminas (Vaidmuo): Tipas

RS6. Privaloma, kad unarinio fakto tipo tipas yra „boolean“.

3.4.3. Struktūrinių veiklos taisyklių transformavimas į realizavimo DB priemonės

Struktūrinės veiklos taisyklės aprašo išvestinius objektų tipus.

3.4.3.1. Taisyklės, reiškiančios privalomus atributus

Taisyklės, reiškiančios privalomus atributus, transformuojamos į „not null“ ribojimus.

Pavyzdys: Veiklos taisyklės RS1-RS3 yra realizuojamos į DB lentelę „Reisas“:

RS1. Privaloma, kad reisas turi tiksliai vieną reiso numerį.

RS2. Privaloma, kad reisas turi tiksliai vieną išvykimo laiką.

RS3. Privaloma, kad reisas turi tiksliai vieną atvykimo laiką.

Column Name	Data Type	Constraints
ID	int	PK, not null
reiso_numeris	varchar(50)	not null
isvykimo_laikas	time(7)	not null
atvykimo_laikas	time(7)	not null
marsruto_ID	int	FK, null

3.4.3.2. Taisyklės, reiškiančios kardinalumo ribojimus (1:0..*)

Taisyklės, reiškiančios kardinalumo ribojimus (1:0..*), transformuojamos į privalomus išorinius raktus FK (*angl. foreign key*).

Pavyzdys: Veiklos taisyklės RS56-RS60 yra realizuojamos į privalomas išorinius raktus FK:

RS56. Privaloma, kad užsakytas skrydis apibūdinamas tiksliai viena kelionės klase.

RS57. Privaloma, kad užsakytas skrydis susijęs su tiksliai vienu skrydžiu.

RS58. Privaloma, kad užsakytas skrydis apibūdinamas tiksliai vienu skrydžio tipu.

RS59. Privaloma, kad užsakytas skrydis užsakytas tiksliai vieno skrydžio užsakovo.

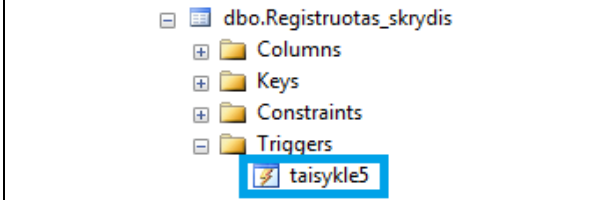
RS60. Privaloma, kad užsakytam skrydžiui priskirta tiksliai viena valiuta.

Key Name	Referenced Table	Referenced Column
PK_UzsakytasSkrydis	UzsakytasSkrydis	PK
FK_Uzsakytas_skrydis_Keliones_klase	Keliones_klase	PK
FK_Uzsakytas_skrydis_Skrydis	Skrydis	PK
FK_Uzsakytas_skrydis_Skrydzio_tipas	Skrydzio_tipas	PK
FK_Uzsakytas_skrydis_Skrydzio_uzsakovas	Skrydzio_uzsakovas	PK
FK_Uzsakytas_skrydis_Valiuta	Valiuta	PK

3.4.3.3. Taisyklės, reiškiančios kardinalumo ribojimus (1:0..1)

Taisyklės, reiškiančios kardinalumo ribojimus (1:0..1), vaizduojamos trigeriais.

1 pavyzdys: Veiklos taisyklė RS82 (kardinalumo 1:0..1 ribojimus) yra realizuojama trigeriai:

RS82. Privaloma, kad <u>bilietui priskirtas ne daugiau kaip vienas registruotas skrydis.</u>	
--	--

Trigerio kodas:

<pre>CREATE TRIGGER [dbo].[taisykle5] ON [dbo].[Registruotas_skrydis] AFTER INSERT AS DECLARE @skaicius1 int; SELECT @skaicius1 = (SELECT COUNT(bilieto_ID) FROM BILIETAS, REGISTRUOTAS_SKRYDIS WHERE BILIETAS.ID = REGISTRUOTAS_SKRYDIS.BILIETO_ID AND BILIETAS.ID = (SELECT bilieto_ID FROM inserted)) IF @skaicius1 > 1 BEGIN RAISERROR (,Veiklos taisyklė: Privaloma, kad bilietui priskirtas ne daugiau kaip vienas registruotas skrydis.', 16, 1) ROLLBACK END</pre>

2 pavyzdys: Veiklos taisyklė RS83 (kardinalumo 1:0..3 ribojimus) yra realizuojama trigeriai:

RS83. Privaloma, kad <u>keleivis vežasi ne daugiau kaip 3 bagažus.</u>	
--	--

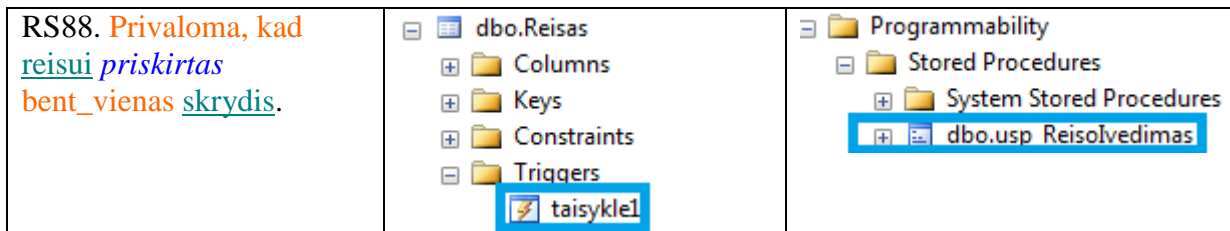
Trigerio kodas:

<pre>CREATE TRIGGER [dbo].[taisykle3] ON [dbo].[Bagazas] after INSERT AS DECLARE @skaicius int; SELECT @skaicius = (SELECT COUNT(keleivio_ID) FROM KELEIVIS, BAGAZAS WHERE KELEIVIS.ID = BAGAZAS.KELEIVIO_ID AND KELEIVIS.ID = (SELECT keleivio_ID FROM inserted)) IF @skaicius > 3 BEGIN RAISERROR (,Veiklos taisyklė: Privaloma, kad keleivis vežasi ne daugiau kaip 3 bagažus.', 16, 1) ROLLBACK END</pre>
--

3.4.3.4. Taisyklės, reiškiančios kardinalumo ribojimus (1:1..*)

Taisyklės, reiškiančios kardinalumo ribojimus (1:1..*), vaizduojamos trigeriais ir procedūromis.

Pavyzdys: Veiklos taisyklė RS88 yra realizuojama trigeriai ir procedūra:



Trigerio kodas:

```
CREATE TRIGGER [dbo].[taisykle1] ON [dbo].[Reisas]
AFTER INSERT
AS
IF NOT EXISTS (SELECT reiso_ID from Skrydis.dbo.Skrydis WHERE reiso_ID =
(SELECT ID FROM inserted) )
BEGIN
RAISERROR (,Veiklos taisyklė: Privaloma, kad reisui priskirtas bent vienas
skrydis.', 16, 1 )
ROLLBACK
END
```

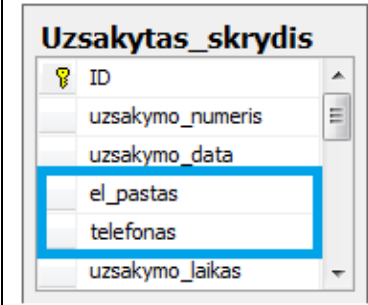
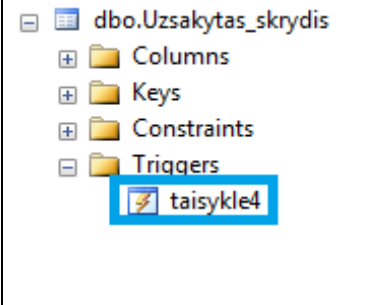
Procedūros kodas:

```
CREATE PROCEDURE [dbo].[usp_ReisoIvedimas] -vartotojo sudarytoms
proceduroms pavadinimas prasideda usp_ (nebutina, bet rekomenduotina)-
--Patikrina parametrus-
@reiso_nr int,
@isvykimo_laikas time(7),
@atvykimo_laikas time(7),
@marsruto_id int,
@isvykimo_data date,
@atvykimo_data date
AS
BEGIN
BEGIN TRANSACTION;
DECLARE @Reiso_ID int;
--Atjungia (disable) Reiso trigeri-
DISABLE TRIGGER [dbo].[taisykle1] ON [dbo].[Reisas]
--Įrašo reiso įrašą-
INSERT INTO [skrydis].[dbo].[Reisas]
([reiso_numeris],[isvykimo_laikas],[atvykimo_laikas],[marsruto_ID])
VALUES
(@reiso_nr,@isvykimo_laikas ,@atvykimo_laikas ,@marsruto_id);
--Iš Reiso paima ID ir įterpia į Skrydžio lentelę -
SELECT @Reiso_ID = (SELECT IDENT_CURRENT(,Reisas'));
INSERT INTO [skrydis].[dbo].[Skrydis]
([isvykimo_data] , [atvykimo_data],[reiso_ID])
VALUES
(@isvykimo_data,@atvykimo_data,@Reiso_ID);
--Įjungia (enable) Reiso trigeri-
ENABLE TRIGGER [dbo].[taisykle1] ON [dbo].[Reisas]
COMMIT; --Patvirtina transakcija (commit)-
END
```

3.4.3.5. Taisyklės, reiškiančios išskirtinumo ribojimą tarp 2 ar daugiau DB stulpelių

Taisyklės, reiškiančios išskirtinumo ribojimus tarp 2 ar daugiau DB stulpelių, vaizduojamos trigeriais.

Pavyzdys: Veiklos taisyklė RS90 yra realizuojama trigeriau:

	RS90. Privaloma, kad <u>užsakytas_skydis</u> turi vieną arba daugiau iš aibės: <u>telefonas</u> , <u>el_paštas</u> .	
---	--	---

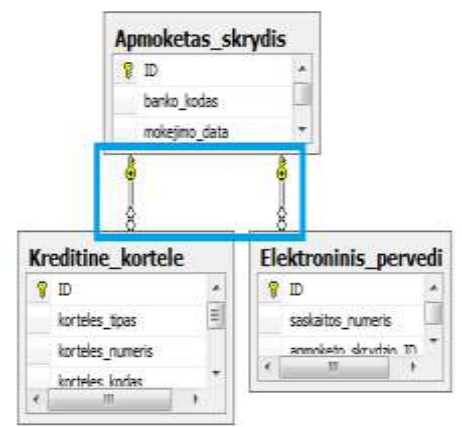
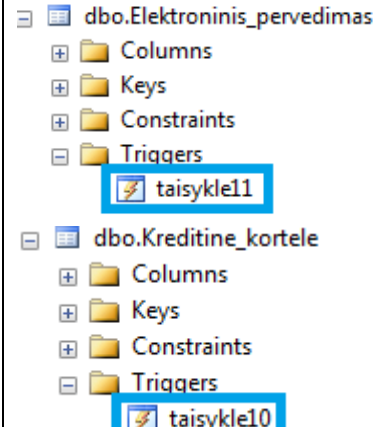
Trigerio kodas:

```
CREATE TRIGGER [dbo].[taisykle4] ON [dbo].[Uzsakytas_skydis]
AFTER INSERT
AS
IF NOT EXISTS (SELECT el_pastas, telefonas FROM inserted WHERE (el_pastas IS NOT NULL AND telefonas IS NOT NULL) OR (el_pastas IS NOT NULL OR telefonas IS NOT NULL))
BEGIN
RAISERROR (,Veiklos taisyklė: Privaloma, kad užsakytas skydis turi vieną arba daugiau iš aibės: telefonas, el_paštas.', 16, 1 )
ROLLBACK
END
```

3.4.3.6. Taisyklės, reiškiančios išskirtinumo ribojimą tarp 2 ar daugiau ryšių su kitomis DB lentelėmis

Taisyklės, reiškiančios išskirtinumo ribojimus tarp 2 ar daugiau ryšių su kitomis DB lentelėmis, vaizduojamos trigeriais.

1 pavyzdys: Veiklos taisyklė RS91 yra realizuojama 2 trigeriais:

	RS91. Privaloma, kad <u>apmokėtas_skydis</u> turi vieną iš aibės: <u>elektroninis_pavedimas</u> , <u>kreditinė_kortelė</u> .	
---	--	---

1 trigerio kodas:

```
CREATE TRIGGER [dbo].[taisykle11] ON [dbo].[Elektroninis_pervedimas]
AFTER INSERT
AS
DECLARE @kreditines_kort_ID int;
SELECT @kreditines_kort_ID = (SELECT COUNT(apmoketo_skrydzio_ID) FROM
KREDITINE_KORTELE WHERE apmoketo_skrydzio_ID = (SELECT apmoketo_skrydzio_ID
FROM inserted) )
IF @kreditines_kort_ID = 1
BEGIN
    RAISERROR (,Veiklos taisyklė: Privaloma, kad apmokėtas skrydis turi
viena iš aibės: elektroninis pervedimas, kreditinė kortelė.', 16, 1 )
    ROLLBACK
END
```

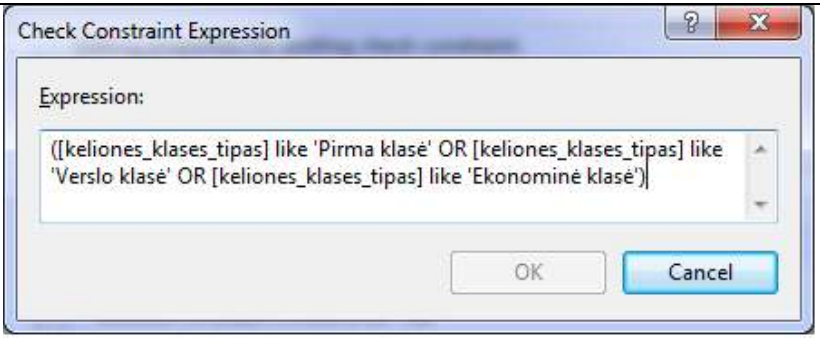
2 trigerio kodas:

```
CREATE TRIGGER [dbo].[taisykle10] ON [dbo].[Kreditine_kortele]
AFTER INSERT
AS
DECLARE @elektroninio_pervedimo_ID int;
SELECT @elektroninio_pervedimo_ID = (SELECT COUNT(apmoketo_skrydzio_ID)
FROM ELEKTRONINIS_PERVEDIMAS WHERE apmoketo_skrydzio_ID = (SELECT
apmoketo_skrydzio_ID FROM inserted) )
IF @elektroninio_pervedimo_ID = 1
BEGIN
    RAISERROR (,Veiklos taisyklė: Privaloma, kad apmokėtas skrydis turi
viena iš aibės: elektroninis pervedimas, kreditinė kortelė.', 16, 1 )
    ROLLBACK
END
```

3.4.3.7. Taisyklės, reiškiančios, kad DB lentelės atributo reikšmė turi būti iš išvardytos atributo reikšmių aibės

Taisyklės, reiškiančios, kad DB lentelės atributo reikšmė turi būti iš išvardintos atributo reikšmių aibės, vaizduojamos „check“ apribojimu.

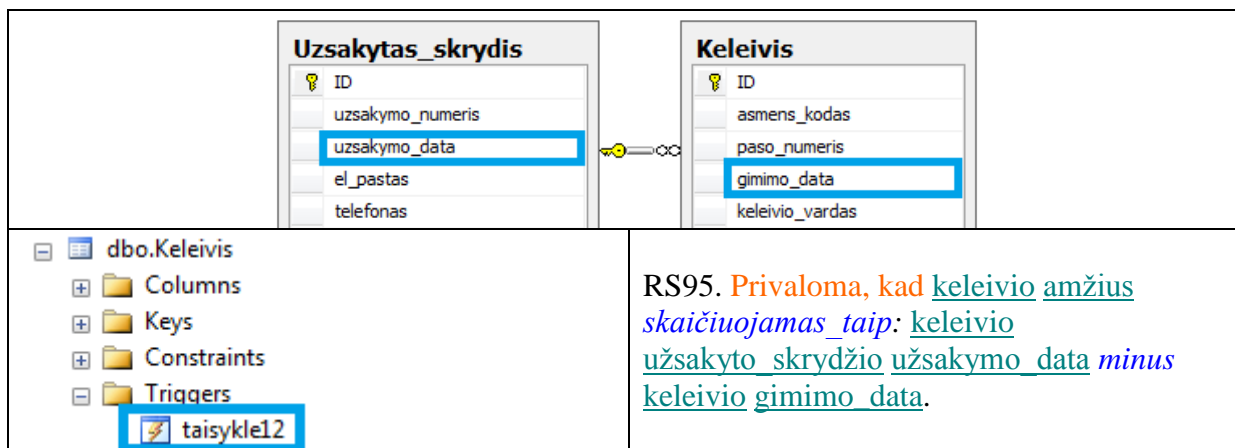
Pavyzdys: Veiklos taisyklė RS92 yra realizuojama „check“ apribojimu:

<table border="1"><thead><tr><th colspan="2">Keliones_klase</th></tr><tr><th>ID</th><th>keliones_klases_tipas</th></tr></thead><tbody><tr><td>1</td><td>Pirma klasė</td></tr><tr><td>2</td><td>Verslo klasė</td></tr><tr><td>3</td><td>Ekonominė klasė</td></tr></tbody></table>	Keliones_klase		ID	keliones_klases_tipas	1	Pirma klasė	2	Verslo klasė	3	Ekonominė klasė	<p>RS92. Privaloma, kad <u>kelionės klasės kelionės klasės tipas yra vienas iš aibės: <u>pirma klasė</u>, <u>verslo klasė</u>, <u>ekonominė klasė</u>.</u></p>
Keliones_klase											
ID	keliones_klases_tipas										
1	Pirma klasė										
2	Verslo klasė										
3	Ekonominė klasė										
<ul style="list-style-type: none">dbo.Keliones_klase<ul style="list-style-type: none">ColumnsKeysConstraints<ul style="list-style-type: none">CK_Keliones_klase											

3.4.3.8. Taisyklės, apimančios 2 ar daugiau objektų tipų vaidmenų, kurie pavaizduojami skirtingų DB lentelių stulpeliais

Taisyklės, apimančios 2 ar daugiau objektų tipų vaidmenų, kurie pavaizduojami skirtingų DB lentelių stulpeliais, vaizduojamos trigeriais.

Pavyzdys: Veiklos taisyklė RS95 yra realizuojama trigeriu:



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'dbo.Keleivis' table is expanded to show the 'Triggers' folder, with 'taisykle12' selected. The main area displays the trigger definition for 'taisykle12' on the 'Keleivis' table. The trigger is linked to the 'uzsakymo_data' column in the 'Uzsakytas_skyrdis' table and the 'gimimo_data' column in the 'Keleivis' table. The trigger code is as follows:

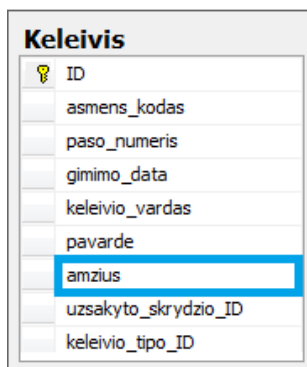
```
CREATE TRIGGER [dbo].[taisykle12] ON [dbo].[Keleivis]
    for INSERT
AS
-- Privaloma, kad keleivio amžius skaičiuojamas taip: keleivio
užsakyto_skyrdžio užsakymo_data minus keleivio gimimo_data.-
DECLARE @uzsakymo_data date
SET @uzsakymo_data = (SELECT uzsakymo_data from Uzsakytas_skyrdis where ID =
(SELECT uzsakymo_skyrdzio_ID from inserted))
DECLARE @gimimo_data date
SET @gimimo_data= (SELECT gimimo_data from inserted)
Update keleivis set amzius = datediff(yy, @gimimo_data, @uzsakymo_data)
```

Trigero kodas:

```
CREATE TRIGGER [dbo].[taisykle12] ON [dbo].[Keleivis]
    for INSERT
AS
-- Privaloma, kad keleivio amžius skaičiuojamas taip: keleivio
užsakyto_skyrdžio užsakymo_data minus keleivio gimimo_data.-
DECLARE @uzsakymo_data date
SET @uzsakymo_data = (SELECT uzsakymo_data from Uzsakytas_skyrdis where ID =
(SELECT uzsakymo_skyrdzio_ID from inserted))
DECLARE @gimimo_data date
SET @gimimo_data= (SELECT gimimo_data from inserted)
Update keleivis set amzius = datediff(yy, @gimimo_data, @uzsakymo_data)
```

Pastaba: po įvykdytos šitos taisyklės, apskaičiuotas keleivio amžius

(@keleivio_amzius) yra įterpiamas į DB lentelės „Keleivis“ stulpelį „amzius“.



Keleivis	
ID	
asmens_kodas	
paso_numeris	
gimimo_data	
keleivio_vardas	
pavarde	
amzius	
uzsakyto_skyrdzio_ID	
keleivio_tipo_ID	

3.4.3.9. Taisyklės, reiškiančios, kad DB lentelės atributų reikšmės turi būti iš išvardytos atributo reikšmių aibės

Taisyklės, reiškiančios, kad DB lentelės atributų reikšmės turi būti iš išvardytos atributo reikšmių aibės, vaizduojamos trigeriais.

Pavyzdys: Veiklos taisyklės RS100-RS102 yra realizuojamos trigeriais:

Keleivio_tipas			
ID	keleivio_tipo_...	amzius_nuo	amzius_iki
1	Suaugęs keleivis	18	999
2	Jaunuolis	2	18
3	Kūdikis	0	2

RS100. Privaloma, kad keleivis yra kūdikis, jei keleivio amžius yra mažesnis nei 2 metai.

RS101. Privaloma, kad keleivis yra suaugęs keleivis, jei keleivio amžius yra ne mažesnis nei 18 metų.

RS102. Privaloma, kad keleivis yra jaunuolis, jei keleivio amžius yra ne mažesnis nei 2 metai ir yra mažesnis nei 18 metų.

Trigerio kodas:

```
CREATE TRIGGER [dbo].[taisykle16] ON [dbo].[Keleivis]
AFTER INSERT
AS
-- Privaloma, kad keleivis yra kūdikis, jei keleivio amžius
yra mažesnis nei 2 metai.-
-- Privaloma, kad keleivis yra suaugęs keleivis, jei keleivio amžius
yra ne mažesnis nei 18 metų.-
-- Privaloma, kad keleivis yra jaunuolis, jei keleivio amžius
yra ne mažesnis nei 2 metai ir yra mažesnis nei 18 metų.-
DECLARE @keleivioTupoID int;
DECLARE @amzius int;
SET @amzius = (SELECT amzius FROM inserted)
SET @keleivioTupoID = (SELECT ID FROM keleivio_tipas WHERE amzius_nuo <=
@amzius AND amzius_iki > @amzius )
Update keleivis set keleivio_tipo_ID = @keleivioTupoID where ID = (SELECT
ID FROM inserted)
```

3.4.3.10. Taisyklės, apimančios objektų tipus ir individualius konceptus

Taisyklės, apimančios objektų tipus ir individualius konceptus, vaizduojamos procedūromis ir trigeriais.

1 pavyzdys: Veiklos taisyklė RS103 yra realizuojama procedūra:

RS103. Privaloma, kad vienpusis skrydis turi tiksliai vieną užsakytą skrydį.

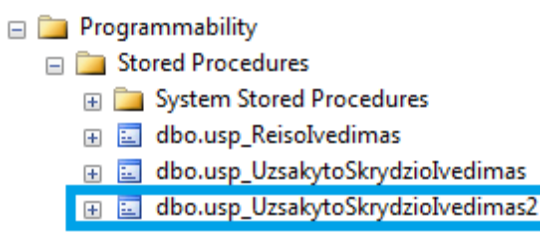
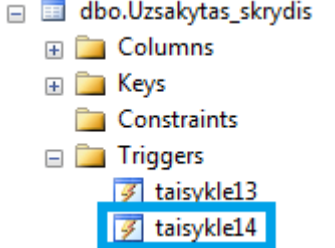
Vienpusio skrydžio procedūros kodas:

```
CREATE PROCEDURE [dbo].[usp_UzsakytoSkrydzioIvedimas]
    @uzsakymo_numeris varchar(50),          @uzsakymo_data date,
    @uzsakymo_laikas time(7),              @yra_patvirtintas bit,
    @yra_nekorektiskas bit,                @yra_patikrintas bit,
    @atsaukimo_data date,                  @el_pastas varchar(50),
    @telefonas varchar(12),                 @keliones_klases_ID int,
    @skrydzio_tipo_ID int,                  @skrydzio_ID int,
    @skrydzio_uzsakovo_ID int,              @valiutos_ID int,
    @asmens_kodas varchar(11),              @paso_numeris varchar(50),
    @gimimo_data date,                      @keleivio_vardas varchar(50),
    @pavarde varchar(50)
AS
BEGIN
    --Privaloma, kad vienpusis_skrydis turi tiksliai_vieną užsakyta_skrydį.-
    DECLARE @skrydzioTipas varchar(50) = ',';
    SET @skrydzioTipas = (SELECT skrydzio_tipo_vardas FROM Skrydzio_tipas
    WHERE ID = @skrydzio_tipo_ID)
    BEGIN TRY
        BEGIN TRANSACTION;
        DECLARE @uzsakyto_skrydzio_ID int;
        DISABLE TRIGGER [dbo].[taisykle2] ON [dbo].[Uzsakytas_skrydis]
            INSERT INTO [skrydis].[dbo].[Uzsakytas_skrydis]
            ([uzsakymo_numeris],[uzsakymo_data],[uzsakymo_laikas],[yra_patvirtintas],[y
            ra_nekorektiskas],[yra_patikrintas],[atsaukimo_data],[el_pastas],[telefonas
            ],[keliones_klases_ID],[skrydzio_tipo_ID],[skrydzio_ID],[skrydzio_uzsakovo_
            ID],[valiutos_ID])
                VALUES
            (@uzsakymo_numeris,@uzsakymo_data,@uzsakymo_laikas,@yra_patvirtintas,@yra_n
            ekorektiskas,@yra_patikrintas,@atsaukimo_data,@el_pastas,@telefonas,@kelion
            es_klases_ID,@skrydzio_tipo_ID,@skrydzio_ID,@skrydzio_uzsakovo_ID,@valiutos
            _ID);
            SELECT @uzsakyto_skrydzio_ID = (SELECT
            IDENT_CURRENT('Uzsakytas_skrydis'));
            INSERT INTO [skrydis].[dbo].[Keleivis]
            ([asmens_kodas],[paso_numeris],[gimimo_data],[keleivio_vardas],[pavarde],
            [uzsakyto_skrydzio_ID])
                VALUES
            (@asmens_kodas,@paso_numeris,@gimimo_data,@keleivio_vardas,@pavarde,
            @uzsakyto_skrydzio_ID);
            ENABLE TRIGGER [dbo].[taisykle2] ON [dbo].[Uzsakytas_skrydis]
        COMMIT;
    END TRY
    BEGIN CATCH
        SELECT ERROR_NUMBER() as error_number,
            ERROR_MESSAGE() as error_message;
        ROLLBACK
    END CATCH
END
```

Vienpusio skrydžio trigerio kodas:

```
CREATE TRIGGER [dbo].[taisykle13] ON [dbo].[Uzsakytas_skrydis]
    for INSERT
AS
DECLARE @skrydis int;
DECLARE @skrydzioTipas varchar(50);
SET @skrydis = (SELECT count(ID) FROM griztamas_Skrydis WHERE
uzsakyto_skrydzio_ID = (SELECT ID FROM inserted) )
SET @skrydzioTipas = (SELECT skrydzio_tipo_vardas FROM Skrydzio_tipas WHERE
ID = (SELECT skrydzio_tipo_ID FROM inserted))
IF (PATINDEX(@skrydzioTipas, 'Vienpusis skrydis') = 1 AND @skrydis <> 0)
BEGIN
RAISERROR (, 'Veiklos taisyklė: Privaloma, kad vienpusis_skrydis turi
tiksliai viena užsakyta skrydį.', 16, 1 )
END
```

2 pavyzdys: RS104 veiklos taisyklei realizuojama analogiška procedūra, tik kurioje „prisideda“ grįžtamo skrydžio parametrai:

<p>RS104. Privaloma, kad <u>dvipusis skrydis turi du užsakytus skrydžius.</u></p>		
---	---	--

Dvipusio skrydžio procedūros kodas:

```
ALTER PROCEDURE [dbo].[usp_UzsakymoSkrydzioIvedimas2] - userio sudarytoms
proceduroms pavadinimas prasideda usp_ (nebutina, bet rekomenduotina)
    @uzsakymo_numeris varchar(50), @uzsakymo_data date,
    @uzsakymo_laikas time(7), @yra_patvirtintas bit,
    @yra_nekorektiskas bit, @yra_patikrintas bit,
    @atsaukimo_data date, @el_pastas varchar(50),
    @telefonas varchar(12), @keliones_klases_ID int,
    @skrydzio_tipo_ID int, @skrydzio_ID int,
    @skrydzio_uzsakovo_ID int, @valiutos_ID int,
    @asmens_kodas varchar(11), @paso_numeris varchar(50),
    @gimimo_data date, @keleivio_vardas varchar(50),
    @pavarde varchar(50), @grizimo_laikas time(7),
    @grizimo_data date
AS
BEGIN
--Privaloma, kad dvipusis_skrydis turi du užsakytus skrydžius.-
    DECLARE @skrydzioTipas varchar(50) = ',';
    SET @skrydzioTipas = (SELECT skrydzio_tipo_vardas FROM Skrydzio_tipas
WHERE ID = @skrydzio_tipo_ID)
    BEGIN TRY
    BEGIN TRANSACTION;
    DECLARE @uzsakyto_skrydzio_ID int;
    DISABLE TRIGGER [dbo].[taisykle2] ON [dbo].[Uzsakytas_skrydis];
    DISABLE TRIGGER [dbo].[taisykle13] ON [dbo].[Uzsakytas_skrydis]; --
grįžtamam skrydžiui
        INSERT INTO [skrydis].[dbo].[Uzsakytas_skrydis]
```

```

([uzsakymo_numeris],[uzsakymo_data],[uzsakymo_laikas],[yra_patvirtintas],[y
ra_nekorektiskas],[yra_patikrintas],[atsaukimo_data],[el_pastas],[telefonas
],[keliones_klases_ID],[skrydzio_tipo_ID],[skrydzio_ID],[skrydzio_uzsakovo_
_ID],[valiutos_ID])
VALUES
(@uzsakymo_numeris,@uzsakymo_data,@uzsakymo_laikas,@yra_patvirtintas,@yra_n
ekorektiskas,@yra_patikrintas,@atsaukimo_data,@el_pastas,@telefonas,@kelion
es_klases_ID,@skrydzio_tipo_ID,@skrydzio_ID,@skrydzio_uzsakovo_ID,@valiutos
_ID);
SELECT @uzsakyto_skrydzio_ID = (SELECT
IDENT_CURRENT('Uzsakytas_skrydis'));
INSERT INTO [skrydis].[dbo].[Griztamas_skrydis] -griztamam
skrydžiui
([grizimo_laikas],[grizimo_data],[uzsakyto_skrydzio_ID],[skrydzio_ID]) -
griztamam skrydžiui
VALUES
(@grizimo_laikas,@grizimo_data,@uzsakyto_skrydzio_ID,@skrydzio_ID); --
griztamam skrydžiui
INSERT INTO [skrydis].[dbo].[Keleivis]
([asmens_kodas],[paso_numeris],[gimimo_data],[keleivio_vardas],[pavarde],
[uzsakyto_skrydzio_ID])
VALUES
(@asmens_kodas,@paso_numeris,@gimimo_data,@keleivio_vardas,@pavarde,
@uzsakyto_skrydzio_ID);
ENABLE TRIGGER [dbo].[taisykle2] ON [dbo].[Uzsakytas_skrydis];
ENABLE TRIGGER [dbo].[taisykle13] ON [dbo].[Uzsakytas_skrydis];
COMMIT;
END TRY
BEGIN CATCH
SELECT ERROR_NUMBER() as error_number,
ERROR_MESSAGE() as error_message;
ROLLBACK
END CATCH
END

```

Dvipusio skrydžio trigerio kodas:

```

CREATE TRIGGER [dbo].[taisykle14] ON [dbo].[Uzsakytas_skrydis]
for INSERT
AS
DECLARE @skrydis int;
DECLARE @skrydzioTipas varchar(50);
SET @skrydis = (SELECT count(ID) FROM griztamas_Skrydis WHERE
uzsakyto_skrydzio_ID = (SELECT ID FROM inserted) )
SET @skrydzioTipas = (SELECT skrydzio_tipo_vardas FROM Skrydzio_tipas
WHERE ID = (SELECT skrydzio_tipo_ID FROM inserted))
IF (PATINDEX(@skrydzioTipas,'Dvipusis skrydis') = 1 AND @skrydis <> 1)
BEGIN
RAISERROR (,'Veiklos taisyklė: Privaloma, kad dvipusis_skrydis turi du
užsakytus_skrydžius.', 16, 1 )
END
END

```

3.4.3.11. Išvedimo (skaičiavimo) taisyklės, apimančios dviejų laukų (pvz., datos ir laiko) sujungimą

Išvedimo (skaičiavimo) taisyklės, apimančios dviejų laukų (pvz., datos ir laiko) sujungimą, vaizduojamos procedūromis ir trigeriais.

1 pavyzdys: Veiklos taisyklės RS105 ir RS106 yra realizuojamos pažymėtu įrašų procedūroje:



Procedūros kodas, kuriame realizuotos RS105 ir RS106 taisyklės (pažymėta dalis):

```
CREATE PROCEDURE [dbo].[usp_ReisoIvedimas]
    @reiso_nr int,
    @atvykimo_laikas time(7),
    @isvykimo_data date,
    @isvykimo_laikas time(7),
    @marsruto_id int,
    @atvykimo_data date
AS
BEGIN
    --Privaloma, kad išvykimo_data ir išvykimo_laikas skaičiuojamas taip:
    --skrydžio išvykimo_data sujungiama su reiso išvykimo_laiku.--
    declare @isvykimas varchar(19);
    set @isvykimas = convert(varchar(20), @isvykimo_data) + ', , ' +
    convert(varchar(8), @isvykimo_laikas);

    --Privaloma, kad atvykimo_data ir atvykimo_laikas skaičiuojamas taip:
    --skrydžio atvykimo_data sujungiama su reiso atvykimo_laiku.--
    declare @atvykimas varchar(19);
    set @atvykimas = convert(varchar(20), @atvykimo_data) + ', , ' +
    convert(varchar(8), @atvykimo_laikas);

    declare @PilnaIsvykimoData datetime;
    declare @PilnaAtvykimoData datetime;
    SET @PilnaIsvykimoData = (SELECT convert(datetime, @isvykimas, 120))
    - 120 yyyy-mm-dd hh:mm:ss(24h)
    SET @PilnaAtvykimoData = (SELECT convert(datetime, @atvykimas, 120))
    --Privaloma, kad skrydžio išvykimo_data yra ne vėlesnė nei skrydžio
    atvykimo_data. -
    --Privaloma, kad reiso išvykimo_laikas yra ne vėlesnis nei reiso
    atvykimo_laikas.-
    --Privaloma, kad skrydžio atvykimo_data yra ne ankstesnė nei skrydžio
    išvykimo_data.-
    --Privaloma, kad reiso atvykimo_laikas yra ne ankstesnis nei reiso
    išvykimo_laikas.-
    IF (@PilnaIsvykimoData >= @PilnaAtvykimoData)
    BEGIN TRANSACTION;

    DECLARE @Reiso_ID int;
    DISABLE TRIGGER [dbo].[taisykle1] ON [dbo].[Reisas]
    INSERT INTO [skrydis].[dbo].[Reisas]
    ([reiso_numeris],[isvykimo_laikas],[atvykimo_laikas],[marsruto_ID])
    VALUES
    (@reiso_nr,@isvykimo_laikas ,@atvykimo_laikas ,@marsruto_id);
    SELECT @Reiso_ID = (SELECT IDENT_CURRENT('Reisas'));
```

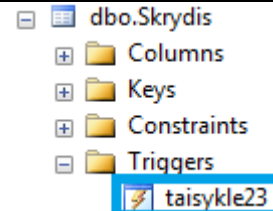
```

INSERT INTO [skrydis].[dbo].[Skrydis]
([isvykimo_data] , [atvykimo_data], [reiso_ID])
VALUES
(@isvykimo_data, @atvykimo_data, @Reiso_ID);
ENABLE TRIGGER [dbo].[taisykle1] ON [dbo].[Reisas]
COMMIT;
END

```

2 pavyzdys: Veiklos taisyklės RS107-RS108 realizuojamos trigeriu:

RS107. Privaloma, kad skrydžio išvykimo data
yra ne vėlesnė nei skrydžio atvykimo data.
RS108. Privaloma, kad skrydžio atvykimo data
yra ne ankstesnė nei skrydžio išvykimo data.



Trigerio kodas:

```

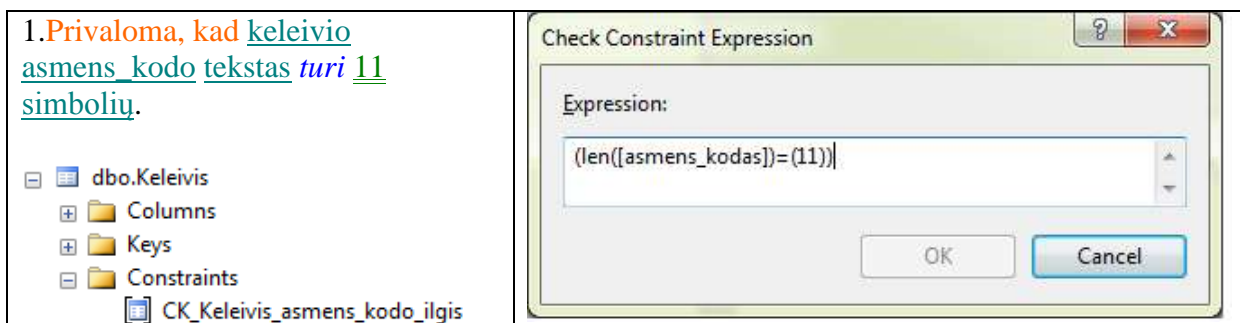
CREATE TRIGGER [dbo].[taisykle23] ON [dbo].[Skrydis]
FOR INSERT
AS
----Privaloma, kad skrydžio išvykimo_data yra ne vėlesnė nei skrydžio
atvykimo_data. -
DECLARE @skydzio_isvykimo_data date
DECLARE @skydzio_atvykimo_data date
SET @skydzio_isvykimo_data = (SELECT isvykimo_data FROM inserted)
SET @skydzio_atvykimo_data = (SELECT atvykimo_data FROM inserted)
IF (@skydzio_atvykimo_data > @skydzio_isvykimo_data )
BEGIN
RAISERROR (,Veiklos taisyklė: Privaloma, kad skrydžio atvykimo_data
yra ne ankstesnė nei skrydžio išvykimo_data.', 16, 1 )
END

```

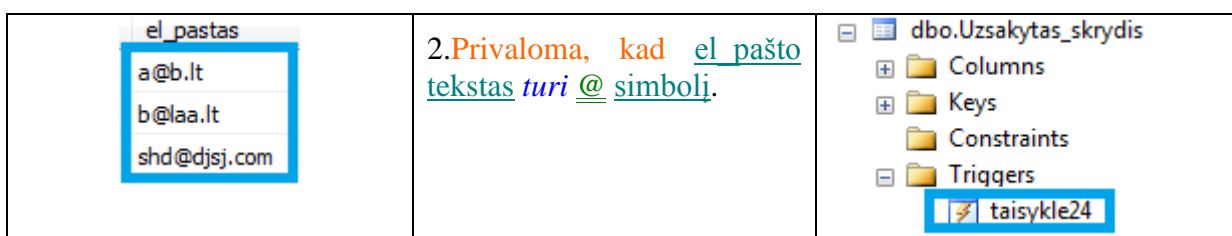
3.4.3.12. Taisyklės, reiškiančios atributų reikšmių ribojimus

Taisyklės, reiškiančios atributų reikšmių ribojimus, realizuojamos „check“ apribojimu arba trigeriu.

1 pavyzdys:



2 pavyzdys:



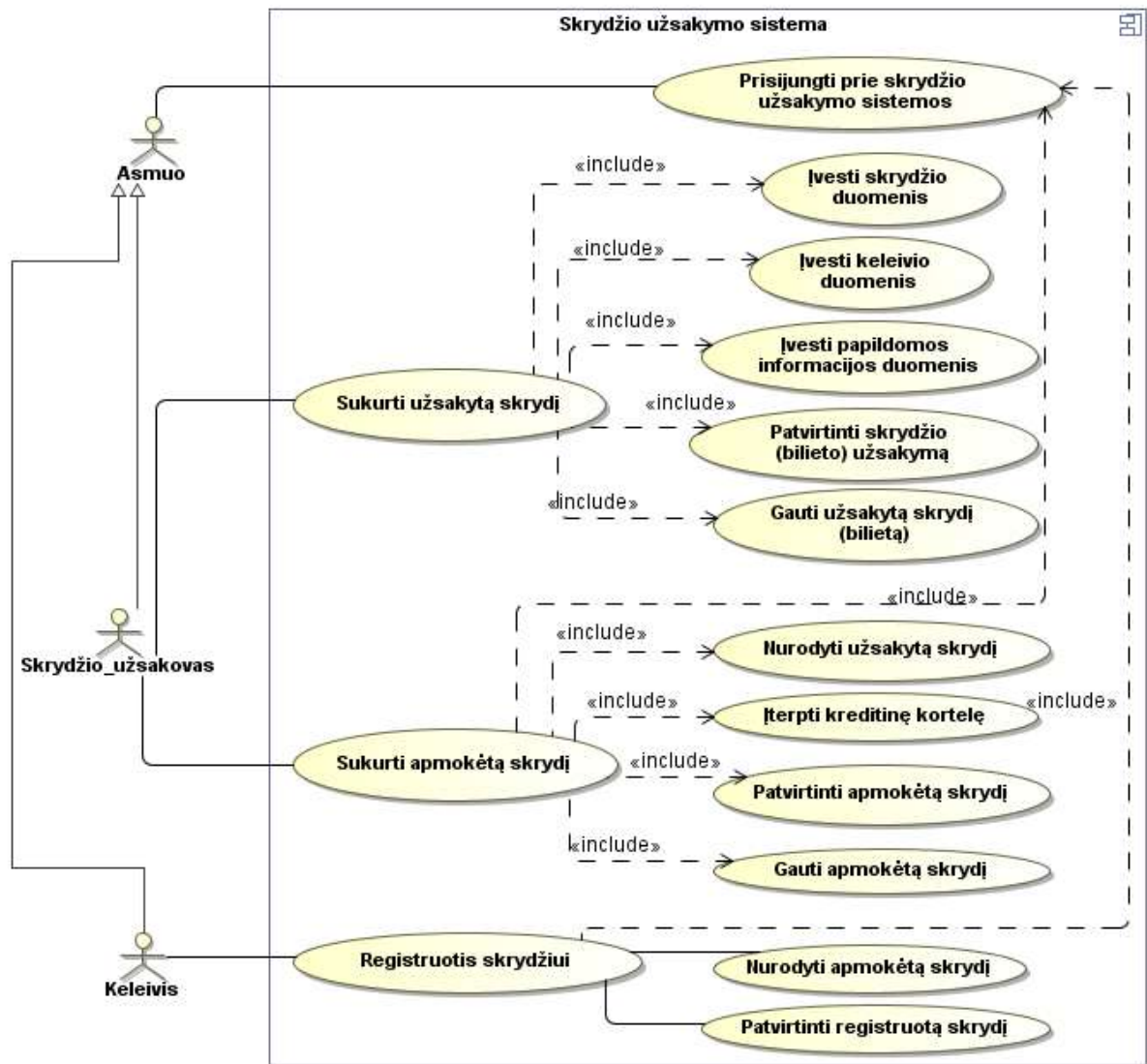
Trigero kodas:

```
CREATE TRIGGER [dbo].[taisykle24] ON [dbo].[Uzsakytas_skyrdis]
AFTER INSERT
AS
IF EXISTS (SELECT el_pastas FROM inserted)
BEGIN
    DECLARE @eta as int = 0;
    SET @eta = (select charindex(,@', (SELECT el_pastas FROM inserted))
    IF (@eta = ,')
    BEGIN
        RAISERROR (,Veiklos taisyklė: Privaloma, kad el.paštas turi „@“
simboli.', 16, 1 )
    END
END
```

3.4.4. Operacinių veiklos taisyklių transformavimas

Operacinės taisyklės vaizduoja operacijas – veiksmus, kuriuos vykdo vartotojas arba sistema. Jos realizuojamos taikomuosiose programose, kurios manipuliuoja DB saugomais duomenimis.

Tiriant veiklos taisyklių specifikuojimo procesą, paaiškėjo, kad norint specifikuoti operacines veiklos taisykles, panaudojimo atvejus reikėtų formuluoti kitaip. Nauja panaudojimo atvejų versija pateikiama 3.4 paveiksle.



3.4 pav. Atnaujinta skrydžio (bilieto) užsakymo sistemos panaudojimo atvejų diagrama

Operacinės taisyklės yra dviejų tipų – leidimo ir įpareigojimo taisyklės. Leidimo taisyklių šablonas leidžia aprašyti funkcijas, kurias leidžiama vykdyti tam tikriems vartotojams.

Leidimo taisyklės:

RS114. **Leidžiama**, kad skrydžio užsakovas *sukuria* užsakytą skrydį jei skrydžio užsakovas yra prisijungęs vartotojas.

RS122. **Leidžiama**, kad keleivis *patvirtina* registruotą skrydį, jei keleivis *sukuria* registruotą skrydį.

RS123. **Leidžiama**, kad skrydžio užsakovas *sukuria* užsakytą skrydį.

Įpareigojimo taisyklių šablonas leidžia aprašyti funkcijas, kurias įsipareigoja vykdyti sistema ar tam tikri vartotojai, esant išpildytoms tam tikroms sąlygoms.

Įsipareigojimo taisyklės:

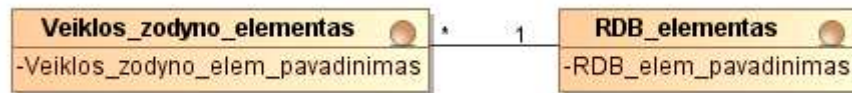
RS126. Įpareigojama, kad skrydžio užsakymo sistema tikrina užsakytą skrydį, jei skrydžio užsakovas patvirtina užsakytą skrydį“.

RS129. Įpareigojama, kad skrydžio užsakovas įterpia skrydį, jei skrydžio užsakovas sukuria užsakytą skrydį.

RS138. Įpareigojama, kad skrydžio užsakymo sistema tikrina registruotą skrydį.

3.4.5. DB schemos semantikos išplėtimas

Norint išsaugoti *SBVR* veiklos žodyno ir taisyklių aprašomą dalykinės srities semantiką, sukurtą DB schemą reikia papildyti. DB lentelių ir stulpelių vardai paprastai nerašomi lietuvių kalba. Nors šios galimybės jau egzistuoja naujose DBVS versijose, lietuviškos raidės gali sukelti problemų apdorojant DB duomenis įvairiomis taikomosiomis programomis. Antra, užrašant faktų tipus ir taisykles naudojamos įvairios to paties žodžio formos, o DB lentelių ir stulpelių vardai kuriami tik iš vienos žodžio formos, kuri apibrėžiama kaip privilegijuotoji (angl. *preferred*). Norint neprarasti šios informacijos, DB schemą reikia papildyti esybėmis „Veiklos_zodyno_elementas“ ir „RDB_elementas“, kurios saugotų žodyno konceptų ir DB elementų vardų atitikmenis (3.5 pav.).



3.5 pav. Esysbės, leidžiančios išsaugoti *SBVR* veiklos žodyno ir RDB elementų vardų atitikmenis

Šių lentelių duomenų pavyzdžiai pateikti 3.6 ir 3.7 pav.

ID	RDB_elem_ID	veiklos_zodyno_elem_pavadinimas
1	1	užsakytas_skrydis
2	2	užsakytam_skrydžiui
3	3	užsakytu_skrydžiu
4	4	keleivis
5	5	keleiviui
6	6	keleivio
7	7	bagażas
8	8	bagażą
9	9	bagażui
10	10	bagażus
11	11	bagażo
12	12	bagażo_tipas
13	13	bagażo_tipu

3.6 pav. Lentelės „Veiklos_zodyno_elementas“ duomenų pavyzdys

ID	RDB_elem_pavadinimas
1	uzsakytas_skrydis
2	uzsakytam_skrydziui
3	uzsakytu_skrydžiu
4	keleivis
5	keleiviui
6	keleivio
7	bagazas
8	bagaza
9	bagazui
10	bagazus
11	bagazo
12	bagazo_tipas
13	bagazo_tipu

3.7 pav. Lentelės „RDB_elementas“ duomenų pavyzdys

Be to, DB schemoje neišsaugomi faktų tipų ir vaidmenų vardai, todėl norint juos atkurti iš duomenų bazės lentelių, reikia sukurti lenteles: „Objekto_tipo_pavadinimas“, „Rysio_tipo_pavadinimas“ ir „Vaidmens_pavadinimas“ (3.8 pav.).



3.8 pav. Veiklos taisyklėms atkurti iš DB reikalingos esybės

Kai kurie terminai gali būti ir vaidmenų, ir objektų tipų vardai, jei fakto tipas naudoja objektų tipų vardus kaip vaidmenų vardus. Šiuo atveju esybių klasių modelyje ryšio galas neturi pavadinimo. Tas pats veiksmažodis gali būti naudojamas įvairiuose faktų tipuose. Kad būtų galima teisingai atstatyti vardus, metaduomenų lentelėse reikia išsaugoti visus jų taikymo variantus.

Objektų lentelė saugo kai kurių objektų tipų (kurie virsta lentelėmis, t.y. esybių) ir atitinkamų lentelių pavadinimus.

Vaidmens lentelė – objektų tipų (kurie virsta stulpeliais, t.y. atributų) ir juos atitinkančių stulpelių pavadinimus.

Pavyzdžiui, turime tris faktų tipus, kur objektų tipų vardai sutampa su vaidmenų vardais ir dviejuose faktų tipuose naudojamas tas pats veiksmažodis: „miestas turi miesto_vardą“, „mieste veikia oro_uostas“, „oro_uostas turi oro_uosto_vardą“.

Šių faktų tipų įrašai metaduomenų lentelėse pateikiami 3.9-3.11 pav.

ID	objekto_pavadinimas
1	miestas
2	oro_uostas
3	marsrutas

3.9 pav. Lentelės „Objekto_tipo_pavadinimas“ duomenų pavyzdys

ID	ryσιο_pavadinimas
1	turi
2	veikia
3	priskirtas
4	pradedamas
5	baigiamas

3.10 pav. Lentelės „Rysio_pavadinimas“ duomenų pavyzdys

ID	objekto_pavad...	ryσιο_pavad_ID	vaidmens_pava...
1	1	1	miesto_vardas
2	1	2	oro_uostas
3	2	1	oro_uosto_vardas

3.11 pav. Lentelės „Vaidmens_pavadinimas“ duomenų pavyzdys

4. Eksperimentinis metodikos tyrimas

Hipotezė – faktų tipus ir veiklos taisykles galima užrašyti šablonais ir juos realizuoti duomenų bazėje.

4.1. Eksperimento apibrėžimas

Pirmas žingsnis, atliekant eksperimentą, yra eksperimento apibrėžimas, kuriame nustatoma eksperimento metu tiriama problema, tikslas ir uždaviniai.

Apibrėžimo šablonas: Išanalizuoti <tyrimo objektą> siekiant <tikslo> <kokybinių kriterijų> atžvilgiu <perspektyvos> požiūriu <konteksto> kontekste [16].

4.1.1. Eksperimento tikslas

Eksperimento tikslas – išanalizuoti eksperimentinę skrydžio užsakymo sistemą, siekiant patikrinti sudarytų taisyklių šablonų tinkamumą, atsižvelgiant į analitiko ir realizuotojo poreikius, kuriant IS, kai veiklos taisyklės realizuojamos DB kontekste.

Tinkamumu šiame darbe suprantamas veiklos taisyklių vaizdavimo šablonais metodikos vienareikšmiškumas, išsamumas ir veiklos konceptų semantikos išsaugojimas. Metodikos vienareikšmiškumas reiškia, kad taikant sukurtus veiklos konceptų, faktų tipų ir taisyklių šablonus, visus juos galima vienareikšmiškai realizuoti duomenų bazėje.

Metodikos išsamumas reiškia, kad kiekvienam konceptui, fakto tipui ar taisyklei galima pritaikyti vieną iš metodikoje aprašytų šablonų.

Veiklos konceptų semantikos išsaugojimas reiškia, kad veiklos konceptus, faktų tipus ir taisykles turi būti galima atstatyti iš duomenų bazės schemas. Esami DB schemų generavimo įrankiai praranda šią semantiką, nes jie sugeneruoja dirbtinius DB stulpelių laukus ir neišsaugo veiklos taisyklių aprašų.

4.1.2. Eksperimento planas

Eksperimento planas:

1. Panaudojant sukurtą lietuviškų šablonų rinkinį, realizuoti ir išanalizuoti eksperimentinę skrydžio užsakymo sistemą.
2. Patikrinti sukurtų šablonų tinkamumą, t.y. vienareikšmiškumą, išsamumą ir veiklos konceptų semantikos išsaugojimą, remiantis skrydžių užsakymų sistemos atveju.

3. Palyginti veiklos konceptų išsaugojimo galimybes, taikant sukurtą metodiką, su esamų DB schemų generavimo įrankių galimybėmis.
4. Išanalizuoti ir įvertinti eksperimento rezultatus.

4.2. Eksperimentinės skrydžio užsakymo sistemos reikalavimai

Šiame poskyryje aprašomi eksperimentinės sistemos panaudojimo atvejai, jų specifikacijos ir dalykinės srities modelis.

4.2.1. Panaudojimo atvejai jų specifikacijos

Kadangi pasirinkta skrydžių sritis, todėl vartotojo panaudojimo atvejai yra susiję su skrydžio bilieto užsakymu. Vartotojas, norėdamas užsakyti skrydžio bilietą (sukurti užsakytą skrydį), turi įvesti pasirinkto skrydžio, savo (kaip keleivio) ir papildomos informacijos duomenis. Užsisakant skrydį, galima apmokėti už skrydį. Vartotojas, gavęs bilietą, registruojasi skrydžiui. Kilus neaiškumams, galima peržiūrėti pagalbą, kurioje vartotojas ras paaiškinimus, kaip jam elgtis neaiškioje situacijoje. Detalizuota vartotojo panaudojimo atvejų diagrama pavaizduota 4.1 paveiksle.



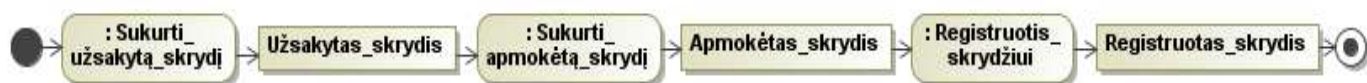
4.1 pav. Detalizuota vartotojo panaudojimo atvejų diagrama

Skrydžio užsakymo ir vykdymo procesų diagrama pavaizduota 4.2 paveiksle.



4.2 pav. Skrydžio užsakymo ir vykdymo procesų diagrama

Skrydžio užsakymo ir vykdymo procesų diagrama su objektų būsenomis pavaizduota 4.3 paveiksle.



4.3 pav. Skrydžio užsakymo ir vykdymo procesų diagrama su objektų būsenomis

Sudėtingesnės vartotojo panaudojimo atvejų specifikacijos pateiktos 4.1-4.5 lentelėse.

Panaudojimo atvejo „Sukurti užsakytą skrydį“ specifikacija pateikta 4.1 lentelėje.

4.1 lentelė. Panaudojimo atvejo „Sukurti užsakytą skrydį“ specifikacija

Panaudojimo atvejis „Sukurti užsakytą skrydį“		
Aktorius	Vartotojas	
Prieš sąlyga	Vartotojas turi būti pasileidęs sistemą	
Sužadinimo sąlyga	Vartotojas nori užsisakyti bilietą skrydžiui	
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	„Įvesti skrydžio duomenis“ <<include>> „Įvesti keleivio duomenis“ <<include>> „Įvesti papildomos informacijos duomenis“ <<include>>
	Specializuoja PA	-
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	
1. Vykdomas panaudojimo atvejis „Įvesti skrydžio duomenis“		
2. Vykdomas panaudojimo atvejis „Įvesti keleivio duomenis“		
3. Vykdomas panaudojimo atvejis „Įvesti papildomos informacijos duomenis“		
Po sąlyga	Užsakytas skrydis sukurtas, t.y. užsakytas bilietas	
Alternatyvūs scenarijai		
-	-	

Skrydžio (skrydžio bilieto) užsakymo procesas pavaizduotas 4.4 paveiksle.



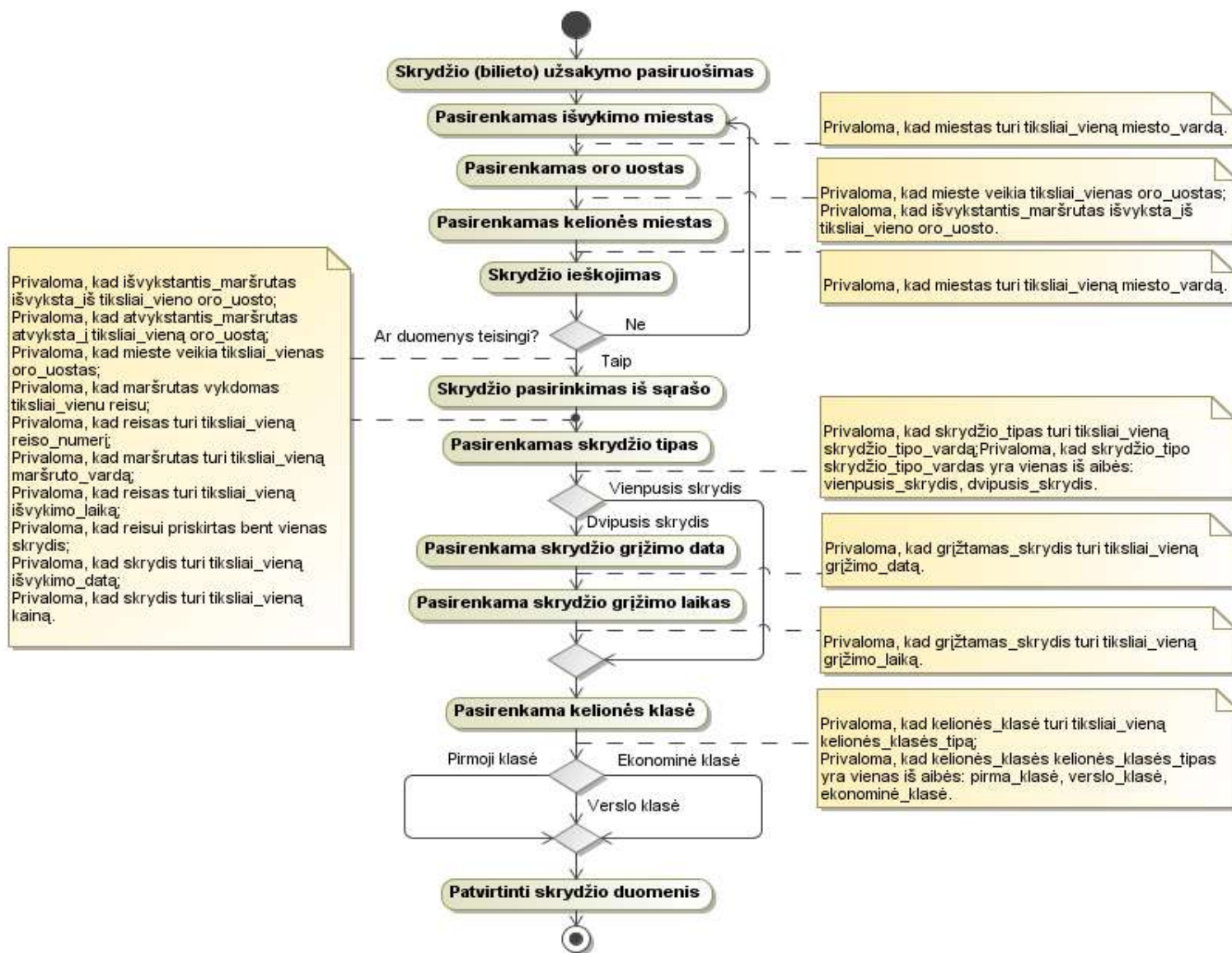
4.4 pav. Skrydžio (skrydžio bilieto) užsakymo procesų diagrama

Kadangi šis procesas yra gan sudėtingas, tad jis – detalizuojamas. Skrydžio bilieto užsakymo procesas skaldomas į tris stambesnius blokus: įvesti skrydžio duomenis (4.5 pav.), įvesti keleivio duomenis (4.6 pav.), bei įvesti papildomos informacijos duomenis (4.7 pav.). Prie kiekvieno skrydžio (skrydžio bilieto) užsakymo proceso bloko (4.5-4.7 pav.) pateiktos veiklos taisyklės, kurios yra reikalingos tam tikrame proceso žingsnyje.

Panaudojimo atvejo „Įvesti skrydžio duomenis“ specifikacija pateikta 4.2 lentelėje.

4.2 lentelė. Panaudojimo atvejo „Įvesti skrydžio duomenis“ specifikacija

Panaudojimo atvejis „Įvesti skrydžio duomenis“		
Aktorius	Vartotojas	
Prieš sąlyga	Vartotojas turi būti pasileidęs sistemą ir meniu juostoje pasirinkęs „Užsakyti bilietą“	
Sužadinimo sąlyga	Vartotojas nori įvesti skrydžio duomenis	
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	
1. Vartotojas pasirenka skrydžio duomenų įvedimą	1.1. Sistema pateikia skrydžio duomenų įvedimo formą	
2. Vartotojas įveda skrydžio duomenis (išvykimo miestą, oro uostą ir atvykimo miestą)	2.1. Sistema pateikia skrydžių sąrašą	
3. Vartotojas pasirenka skrydį ir suveda kitus kelionės duomenis, bei patvirtina.	3.1. Sistema patikrina įvestus duomenis ir juos išsaugo 3.2. Sistema išsaugo užsakymą	
Po sąlyga	Sistemoje išsaugoti skrydžio duomenys	
Alternatyvūs scenarijai		
2a. Jei vartotojas, užpildydamas skrydžio formą, padarė klaidų	2.a1 Sistema suformuoja klaidos pranešimą ir neišsaugo duomenų	
2.b Vartotojas ištaiso klaidas ir paspaudžia dar kartą „Patvirtinti“	2.b1. Atliktas skrydžio duomenų įvedimas	

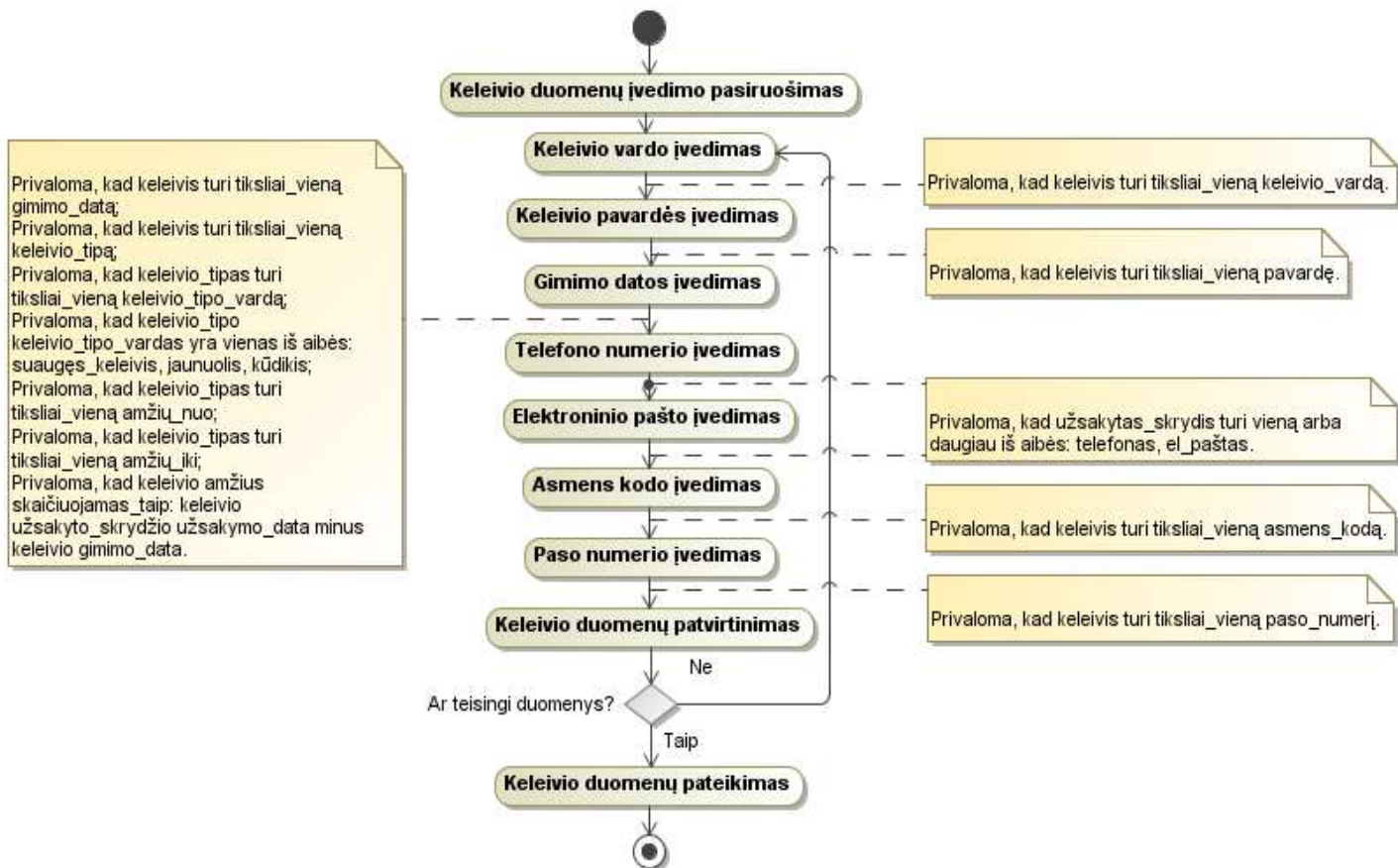


4.5 pav. Skrydžio (skrydžio bilieto) užsakymo procesas. Skrydžio duomenų įvedimo blokas

Panaudojimo atvejo „Įvesti keleivio duomenis“ specifikacija pateikta 4.3 lentelėje.

4.3 lentelė. Panaudojimo atvejo „Įvesti keleivio duomenis“ specifikacija

Panaudojimo atvejis „Įvesti keleivio duomenis“		
Aktorius	Vartotojas	
Prieš sąlyga	Vartotojas turi būti pasileidęs sistemą ir meniu juostoje pasirinkęs „Užsakyti bilietą“	
Sužadinimo sąlyga	Vartotojas nori įvesti keleivio (savo) duomenis	
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas		
1. Vartotojas pasirenka keleivio duomenų įvedimą	1.1. Sistema pateikia keleivio duomenų įvedimo formą	
2. Vartotojas įveda keleivio duomenis (keleivio vardą, pavardę, telefono numerį ir t.t.)	2.1. Sistema patikrina įvestus duomenis ir juos išsaugo	
Po sąlyga	Sistemoje išsaugoti keleivio duomenys	
Alternatyvūs scenarijai		
2.a Jei vartotojas, užpildydamas keleivio formą, padarė klaidų	2.a1 Sistema suformuoja klaidos pranešimą ir neišsaugo duomenų	
2.b Vartotojas ištaiso klaidas ir paspaudžia dar kartą „Patvirtinti“	2.b1. Atliktas keleivio duomenų įvedimas	

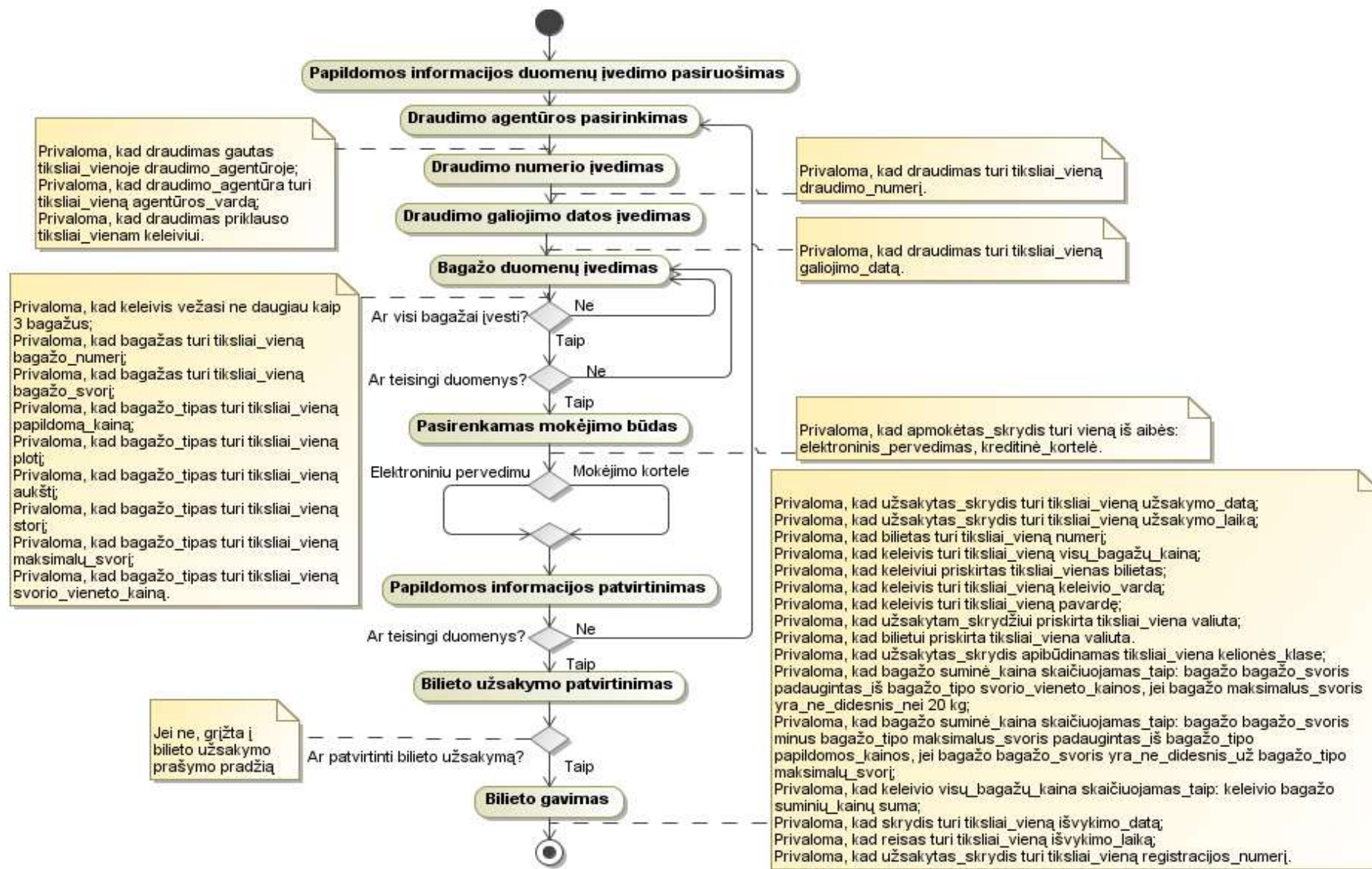


4.6 pav. Skrydžio (skrydžio bilieto) užsakymo procesas. Keleivio duomenų įvedimo blokas

Panaudojimo atvejo „Įvesti papildomos informacijos duomenis“ specifikacija pateikta 4.4 lentelėje.

4.4 lentelė. Panaudojimo atvejo „Įvesti papildomos informacijos duomenis“ specifikacija

Panaudojimo atvejis „Įvesti papildomos informacijos duomenis“		
Aktorius	Vartotojas	
Prieš sąlyga	Vartotojas turi būti pasileidęs sistemą ir meniu juostoje pasirinkęs „Užsakyti bilietą“	
Sužadinimo sąlyga	Vartotojas nori įvesti papildomos informacijos duomenis	
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas		
Sistemos reakcija ir sprendimai		
1. Vartotojas pasirenka papildomos informacijos duomenų įvedimą	1.1. Sistema pateikia papildomos informacijos duomenų įvedimo formą	
2. Vartotojas įveda papildomos informacijos duomenis (draudimą, bagažus)	2.1. Sistema patikrina įvestus duomenis ir juos išsaugo	
3. Vartotojas patvirtina bilieto užsakymą	3.1. Sistema išsaugo bilieto užsakymą	
Po sąlyga	Sistemoje išsaugoti papildomos informacijos duomenys	
Alternatyvūs scenarijai		
2.a Jei vartotojas, užpildydamas papildomos informacijos formą, padarė klaidų	2.a1 Sistema suformuoja klaidos pranešimą ir neišsaugo duomenų	
2.b Vartotojas ištaiso klaidas ir paspaudžia dar kartą „Patvirtinti“	2.b1. Sistema išsaugo įvestus papildomos informacijos duomenis	



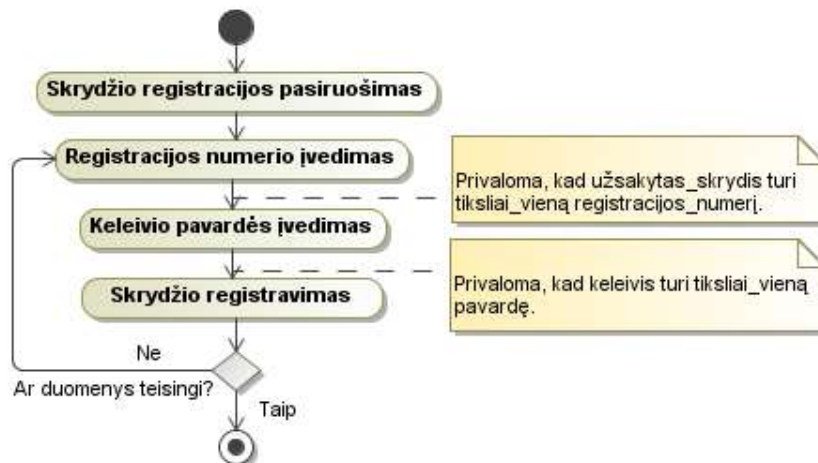
4.7 pav. Skrydžio (skrydžio bilieto) užsakymo procesas. Papildomos informacijos duomenų įvedimo blokas

Panaudojimo atvejo „Registruotis skrydžiui“ specifikacija pateikta 4.5 lentelėje.

4.5 lentelė. Panaudojimo atvejo „Registruotis skrydžiui“ specifikacija

Panaudojimo atvejis „Registruotis skrydžiui“		
Aktorius	Vartotojas	
Prieš sąlyga	Vartotojas turi būti pasileidęs sistemą ir užsisakęs bilietą skrydžiui	
Sužadinimo sąlyga	Vartotojas nori registruotis skrydžiui	
Susiję panaudojimo atvejai	Išplečia PA	-
	Apima PA	-
	Specializuoja PA	-
Pagrindinis įvykių srautas		
Sistemos reakcija ir sprendimai		
1. Vartotojas pasirenka skrydžio registracijos įvedimą	1.1. Sistema pateikia skrydžio registracijos formą	
2. Vartotojas įveda skrydžio registracijos duomenis (registracijos numerį ir pavardę)	2.1. Sistema patikrina įvestus duomenis ir juos išsaugo	
Po sąlyga	Atliktas skrydžio registravimas	
Alternatyvūs scenarijai		
-	-	

4.8 paveiksle pavaizduotas skrydžio registravimo procesas. Šiame procese matomi duomenys, kurie yra reikalingi registruojantis skrydžiui. Prie skrydžio registravimo proceso (4.8 pav.) pateiktos veiklos taisyklės, kurios yra reikalingos tam tikrame proceso žingsnyje.



4.8 pav. Skrydžio registravimo procesas

- Kiti procesai

Skrydžio apmokėjimo procesas buvo nerealizuojamas eksperimentinėje skrydžio užsakymo sistemoje.

Skrydžio apmokėjimas yra galimas dviem būdais: užsisakant skrydį internetu arba oro uoste. Skrydžio apmokėjimo procesas pateiktas 4.9 paveiksle.



4.9pav. Skrydžio apmokėjimo procesas

4.2.2. Dalykinės srities modelis

Išanalizavus veiklos konceptus, buvo gautos tokios esybės: „Miestas“, „Oro_uostas“, „Marsrutas“, „Reisas“, „Skrydis“, „Griztamas_skrydis“, „Registruotas_skrydis“, „Keliones_klase“, „Bilietas“, „Skrydzio_tipas“, „Apmoketas_skrydis“, „Uzsakytas_skrydis“, „Valiuta“, „Bagazo_tipas“, „Kreditine_kortele“, „Skrydzio_uzsakovas“, „Matavimo_vienetas“, „Keleivio_tipas“, „Keleivis“, „Draudimas“, „Draudimo_agentura“, „Bagazas“, „Elektroninis_pervedimas“. Sudaryta dalykinės srities esybių klasių diagrama pateikta 4.10 paveiksle.

Dalykinės srities esybių klasių diagramoje naudojamos semantikos išsaugojimo metalentelės pateiktos 3.5 ir 3.8 paveiksluose.

4.2.3. Šablonų tinkamumo įvertinimas

Siekdami patikrinti šablonų tinkamumą, paimsime dalį veiklos taisyklių ir parodysime jų formavimą 4.2.3.1-4.2.3.4 poskyriuose, o 4.2.3.5 poskyryje pateiksime sugeneruotą DB schemą iš esybių klasių modelio, kuriame galima išsaugoti vaidmenų pavadinimus.

4.2.3.1. Dalykinės srities aprašymas natūralia kalba ir veiklos taisyklėmis

Dalykinė sritis buvo pasirinkta daugumai gerai pažįstama – skrydžio užsakymas (skrydžio bilieto užsakymas). Užsisakant bilietą (internetu), vartotojui reikia įvesti norimo skrydžio duomenis. Kiekvieną įvedimo žingsnį galima užrašyti veiklos taisykle.

Ši dalykinė sritis pasirinkta ir dėl to, kad norint pilnai realizuoti skrydžio užsakymą, reikia užrašyti ne tik jau esamus angliškus *SBVR* veiklos taisyklių tipus, bet ir naujai įvesti tokius veiklos taisyklių tipus, kaip skaičiavimo, duomenų turinio ar formatų.

4.2.3.2. Faktų tipų sudarymas pagal *SBVR*

Pagal *SBVR* aprašymą faktų tipai ir veiklos taisyklės apibrėžiami keturiais konceptų stiliais: [terminas](#), [raktažodis](#), [literalus simbolis/vardas](#) ir [veiksmazodis](#) (4 priedo 1 lentelė).

Atsižvelgus į dalykinę sritį ir *SBVR* konceptų stilius, sudaromi faktų tipai (2 priedas). Fakto tipas – tai ryšys tarp dviejų ar daugiau daiktavardžių sąvokų (vaidmenų ir juos vaidinančių terminų).

Pavyzdys pateiktas 4.6 lentelėje.

4.6 lentelė. Faktų tipų užrašymas natūralia ir struktūrizuota kalba

Faktų tipai natūralia kalba	Faktų tipai struktūrizuota kalba
FT1. Užsakytas skrydis turi užsakymo datą.	FT1. Užsakytas skrydis turi užsakymo datą.
FT2. Užsakytas skrydis turi užsakymo numerį.	FT2. Užsakytas skrydis turi užsakymo numerį.
FT8. Užsakytas skrydis turi el. pašta.	FT8. Užsakytas skrydis turi el pašta.
FT9. Užsakytas skrydis turi telefoną.	FT9. Užsakytas skrydis turi telefoną.
FT12. Užsakytam skrydžiui priskirtas keleivis.	FT12. Užsakytam skrydžiui priskirtas keleivis.
FT13. Keleivis turi keleivio vardą.	FT13. Keleivis turi keleivio vardą.
FT14. Keleivis turi pavardę.	FT14. Keleivis turi pavardę.
FT16. Keleivis turi gimimo datą.	FT16. Keleivis turi gimimo datą.
FT17. Keleivis turi asmens kodą. Sinoniminė forma: Asmens kodas priklauso keleiviui.	FT17. Keleivis turi asmens kodą. Sinoniminė forma: Asmens kodas priklauso keleiviui.
FT18. Keleivis turi amžių.	FT18. Keleivis turi amžių.
FT30. Keleivis vežasi bagažą.	FT30. Keleivis vežasi bagažą.
FT31. Bagažas turi bagažo numerį.	FT31. Bagažas turi bagažo numerį.
FT32. Bagažas turi suminę kainą.	FT32. Bagažas turi suminę kainą.
FT33. Bagažas turi bagažo svorį.	FT33. Bagažas turi bagažo svorį.
FT49. Keleivis apibūdinamas keleivio tipu.	FT49. Keleivis apibūdinamas keleivio tipu.

FT50. Keleivio tipas turi keleivio tipo vardą.	FT50. <u>Keleivio tipas turi keleivio tipo vardą.</u>
FT51. Keleivio tipas turi amžių nuo.	FT51. <u>Keleivio tipas turi amžių nuo.</u>
FT52. Keleivio tipas turi amžių iki.	FT52. <u>Keleivio tipas turi amžių iki.</u>
FT53. Suaugęs keleivis yra keleivio tipo vardas.	FT53. <u>Suaugęs keleivis yra keleivio tipo vardas.</u>
FT54. Jaunuolis yra keleivio tipo vardas.	FT54. <u>Jaunuolis yra keleivio tipo vardas.</u>
FT55. Kūdikis yra keleivio tipo vardas.	FT55. <u>Kūdikis yra keleivio tipo vardas.</u>
FT107. Keleivis turi visų bagažų kainą.	FT107. <u>Keleivis turi visų bagažų kainą.</u>

4.2.3.3. Veiklos taisyklių sudarymas iš faktų tipų

SBVR buvo įvestas Veiklos taisyklių metodo principas.

"Business rules are based on facts, and facts are based on terms".

SBVR faktų tipai ir veiklos taisyklės glaudžiai siejasi su kalba. Natūrali kalba yra paprasčiausias veiklos taisyklių užrašymo būdas, bet neformalus.

Struktūrizuotų veiklos taisyklių (3 priedas) sudarymas iš faktų tipų matomas 4.7 lentelėje.

4.7 lentelė. Veiklos taisyklių sudarymas iš faktų tipų

Veiklos taisyklė (RS)	Fakto tipas (FT)
RS10. Privaloma, kad <u>užsakytas skrydis turi tiksliai vieną užsakymo numerį.</u>	FT2. <u>Užsakytas skrydis turi užsakymo numerį.</u>
RS11. Privaloma, kad <u>užsakytas skrydis turi tiksliai vieną užsakymo datą.</u>	FT1. <u>Užsakytas skrydis turi užsakymo datą.</u>
RS28. Privaloma, kad <u>bagażas turi tiksliai vieną bagažo numerį.</u>	FT31. <u>Bagažas turi bagažo numerį.</u>
RS29. Privaloma, kad <u>bagażas turi tiksliai vieną bagažo svorį.</u>	FT33. <u>Bagažas turi bagažo svorį.</u>
RS36. Privaloma, kad <u>keleivis turi tiksliai vieną visų bagažų kainą.</u>	FT107. <u>Keleivis turi visų bagažų kainą.</u>
RS37. Privaloma, kad <u>keleivis turi tiksliai vieną asmens kodą.</u>	FT17. <u>Keleivis turi asmens kodą.</u>
RS39. Privaloma, kad <u>keleivis turi tiksliai vieną gimimo datą.</u>	FT16. <u>Keleivis turi gimimo datą.</u>
RS40. Privaloma, kad <u>keleivis turi tiksliai vieną keleivio vardą.</u>	FT13. <u>Keleivis turi keleivio vardą.</u>
RS41. Privaloma, kad <u>keleivis turi tiksliai vieną pavardę.</u>	FT14. <u>Keleivis turi pavardę.</u>
RS50. Privaloma, kad <u>keleivio tipas turi tiksliai vieną keleivio tipo vardą.</u>	FT50. <u>Keleivio tipas turi keleivio tipo vardą.</u>
RS51. Privaloma, kad <u>keleivio tipas turi tiksliai vieną amžių nuo.</u>	FT51. <u>Keleivio tipas turi amžių nuo.</u>
RS52. Privaloma, kad <u>keleivio tipas turi tiksliai vieną amžių iki.</u>	FT52. <u>Keleivio tipas turi amžių iki.</u>

RS78. Privaloma, kad <u>keleivis apibūdinamas tiksliai vienu keleivio tipu.</u>	FT49. <u>Keleivis apibūdinamas keleivio tipu.</u>
RS83. Privaloma, kad <u>keleivis vežasi ne daugiau kaip 3 bagažus.</u>	FT30. <u>Keleivis vežasi bagažą.</u>
RS89. Privaloma, kad <u>užsakytam skrydžiui priskirtas bent vienas keleivis.</u>	FT12. <u>Užsakytam skrydžiui priskirtas keleivis.</u>
RS90. Privaloma, kad <u>užsakytas skrydis turi vieną arba daugiau iš aibės: telefonas, el paštas.</u>	FT8. <u>Užsakytas skrydis turi el pašta.</u> FT9. <u>Užsakytas skrydis turi telefoną.</u>
RS94. Privaloma, kad <u>keleivio tipo keleivio tipo vardas yra vienas iš aibės: suaugęs keleivis, jaunuolis, kūdikis.</u>	FT50. <u>Keleivio tipas turi keleivio tipo vardą.</u> FT53. <u>Suaugęs keleivis yra keleivio tipo vardas.</u> FT54. <u>Jaunuolis yra keleivio tipo vardas.</u> FT55. <u>Kūdikis yra keleivio tipo vardas.</u>
RS95. Privaloma, kad <u>keleivio amžius skaičiuojamas taip: keleivio užsakymo skrydžio užsakymo data minus keleivio gimimo data.</u>	FT18. <u>Keleivis turi amžių.</u> FT1. <u>Užsakytas skrydis turi užsakymo datą.</u> FT16. <u>Keleivis turi gimimo datą.</u>
RS98. Privaloma, kad <u>keleivio visų bagažų kaina skaičiuojama taip: suma, kuri susideda iš keleivio bagažo suminių kainų.</u>	FT107. <u>Keleivis turi visų bagažų kainą.</u> FT32. <u>Bagažas turi suminę kainą.</u>
RS100. Privaloma, kad <u>keleivis yra kūdikis, jei keleivio amžius yra mažesnis nei 2 metai.</u>	FT18. <u>Keleivis turi amžių.</u> FT55. <u>Kūdikis yra keleivio tipo vardas.</u>
RS101. Privaloma, kad <u>keleivis yra suaugęs keleivis, jei keleivio amžius yra ne mažesnis nei 18 metų.</u>	FT18. <u>Keleivis turi amžių.</u> FT53. <u>Suaugęs keleivis yra keleivio tipo vardas.</u>
RS102. Privaloma, kad <u>keleivis yra jaunuolis, jei keleivio amžius yra ne mažesnis nei 2 metai ir yra mažesnis nei 18 metų.</u>	FT18. <u>Keleivis turi amžių.</u> FT54. <u>Jaunuolis yra keleivio tipo vardas.</u>

4.2.3.4. Lietuviškų šablonų sudarymas

Atlikus lyginamąją analizę, buvo nutarta formuluoti lietuviškus šablonus remiantis SBVR stiliumi ir papildyti juos Witt rekomendacijomis, kad šablonai būtų lengvai kompiuterizuojami ir reikiamo išsamumo lygio.

Šablonai sudaromi tiek faktų tipams (3.2 poskyris), tiek veiklos taisyklėms (3.3 poskyris). Kadangi, faktų tipų šablonai išsamiai užrašyti 3.2 poskyryje, tad vykdant eksperimentą, tai nebenagrinėjama.

Eksperimento etape užrašytoms veiklos taisyklėms sudaromi tokie šablonai:

- Išvedimo taisyklėms sudaromi du šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmazodis {yra}> <individas> jei
<terminas-tipas> <terminas-savybė> <nustatytas veiksmazodis {yra_mažesnis_nei |
yra_ne_mažesnis_nei}> <individas>}
```

RS100. Privaloma, kad keleivis yra kūdikis, jei keleivio amžius yra mažesnis nei 2 metai.

RS101. Privaloma, kad keleivis yra suaugęs keleivis, jei keleivio amžius yra ne mažesnis nei 18 metų.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmazodis {yra}> <individas> jei  
<terminas-tipas> <terminas-savybė> <nustatytas veiksmazodis {yra_ne_mažesnis_nei}>  
<individas> ir <nustatytas veiksmazodis {yra_mažesnis_nei}> <individas>}
```

RS102. Privaloma, kad keleivis yra jaunuolis, jei keleivio amžius yra ne mažesnis nei 2 metai ir yra mažesnis nei 18 metų.

- Skaičiavimo taisyklėms sudaromi du šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <veiksmazodis  
{skaičiuojamas_taip}> : <terminas-tipas> <terminas-tipas> <terminas-savybė> <veiksmazodis  
{minus}> <terminas-tipas> <terminas-savybė>}
```

RS95. Privaloma, kad keleivio amžius skaičiuojamas taip: keleivio užsakyto skrydžio užsakymo data minus keleivio gimimo data.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <veiksmazodis  
{skaičiuojama_taip}> : <veiksmazodis {suma}>, kuri <veiksmazodis {susideda iš}> <terminas-  
tipas> <terminas-tipas> <terminas-savybė>}
```

RS98. Privaloma, kad keleivio visų bagažų kaina skaičiuojama taip: suma, kuri susideda iš keleivio bagažo suminių kainų.

- Duomenų turinio taisyklėms sudaromi du šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmazodis {turi}> <raktinis žodis  
{vieną_arba_daugiau iš_aibės}> <terminas-savybė> <terminas-savybė>}
```

RS90. Privaloma, kad užsakytas skrydis turi vieną arba daugiau iš aibės: telefonas, el paštas.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <veiksmazodis {yra}>  
<raktinis žodis {vienas iš aibės}> <individas> <individas> <individas>}
```

RS94. Privaloma, kad keleivio tipo keleivio tipo vardas yra vienas iš aibės: suaugęs keleivis, jaunuolis, kūdikis.

- Kardinalumo taisyklėms sudaromi keturi šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmazodis {turi}> <raktinis žodis  
{tiksliai_vieną}> <terminas-savybė>}
```

RS10. Privaloma, kad užsakytas skrydis turi tiksliai vieną užsakymo numerį.

RS11. Privaloma, kad užsakytas skrydis turi tiksliai vieną užsakymo datą.

RS28. Privaloma, kad bagażas turi tiksliai vieną bagažo numerį.

RS29. Privaloma, kad bagażas turi tiksliai vieną bagažo svorį.

RS36. Privaloma, kad keleivis turi tiksliai vieną visų bagažų kainą.

RS37. Privaloma, kad keleivis turi tiksliai vieną asmens kodą.

- RS39. Privaloma, kad keleivis turi tiksliai_vieną gimimo datą.
- RS40. Privaloma, kad keleivis turi tiksliai_vieną keleivio vardą.
- RS41. Privaloma, kad keleivis turi tiksliai_vieną pavardę.
- RS50. Privaloma, kad keleivio tipas turi tiksliai_vieną keleivio tipo vardą.
- RS51. Privaloma, kad keleivio tipas turi tiksliai_vieną amžių nuo.
- RS52. Privaloma, kad keleivio tipas turi tiksliai_vieną amžių iki.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmazodis {apibūdinamas}> <raktinis žodis { tiksliai_vienu}> <terminas-tipas>}
```

- RS78. Privaloma, kad keleivis apibūdinamas tiksliai_vienu keleivio tipu.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmazodis {priskirtas}> <raktinis žodis {bent_vienas}> <terminas-tipas> }
```

- RS89. Privaloma, kad užsakytam skrydžiui priskirtas bent_vienas keleivis.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <veiksmazodis {vežasi}> <raktinis žodis {ne_daugiau_kaip_3}> <terminas-tipas>}
```

- RS83. Privaloma, kad keleivis vežasi ne_daugiau_kaip_3 bagažus.

- Unikalumo taisyklei sudaromas vienas šablonas:

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <veiksmazodis {yra}> <terminas-duomenų tipas>}
```

19. Privaloma, kad užsakyto skrydžio užsakymo numeris yra unikali seka.

- Duomenų formatų taisyklėms sudaromi keturi šablonai:

```
{<raktinis žodis {privaloma, kad}> <terminas-savybė> <veiksmazodis {yra}> <terminas-duomenų tipas>}
```

5. Privaloma, kad suminė kaina yra skaičius.
6. Privaloma, kad keleivio vardas yra tekstas.
7. Privaloma, kad telefonas yra tekstas.
11. Privaloma, kad bagazo svoris yra realus kaičius.
16. Privaloma, kad amžius nuo yra realus kaičius.
21. Privaloma, kad užsakymo data yra data.
22. Privaloma, kad užsakymo numeris yra tekstas.

```
{<raktinis žodis {privaloma, kad}> <terminas-tipas> <terminas-savybė> <terminas-duomenų tipas> <veiksmazodis {turi}> <individas> <terminas-duomenų tipas>}
```

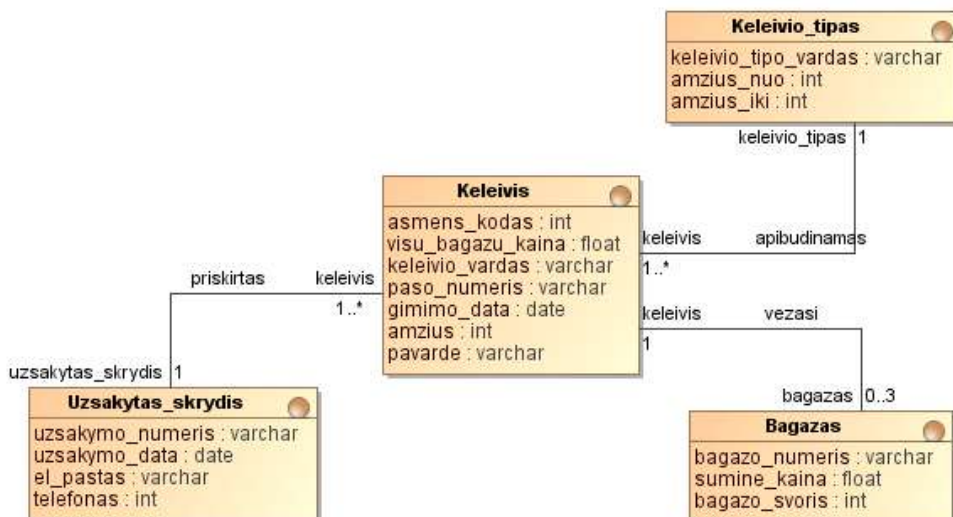
1. Privaloma, kad keleivio asmens kodo tekstas turi 11 simbolių.

```
{<raktinis žodis {privaloma, kad}> <terminas-savybė> <terminas-savybė> <veiksmazodis {turi}> <individas> <terminas-duomenų tipas>}
```

2. Privaloma, kad el pašto tekstas turi @ simbolį.
3. Privaloma, kad telefono tekstas turi 9 simbolius.

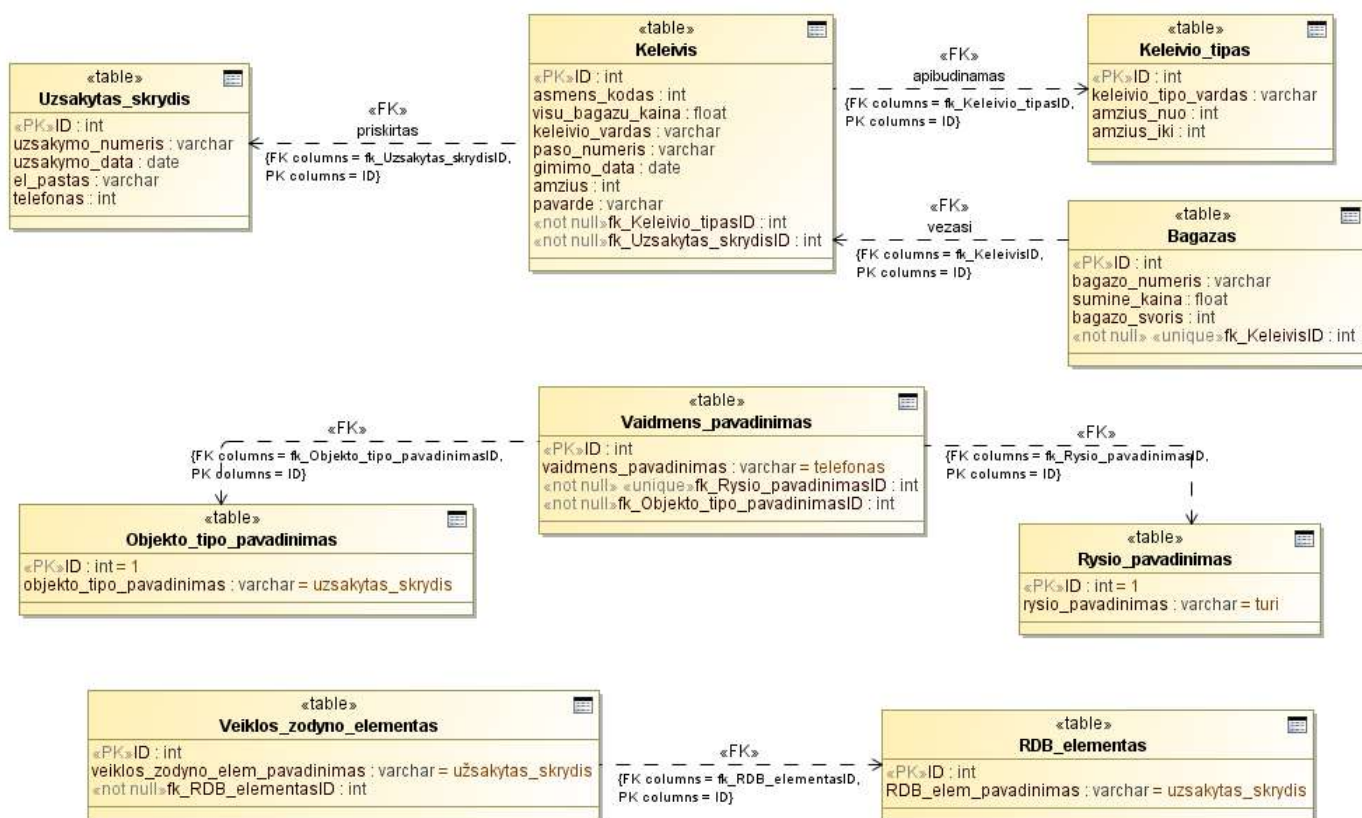
4.2.3.5. DB schema su išsaugojamais vaidmenų pavadinimais

Taikant sukurtą metodiką, iš skrydžio užsakymo sistemos esybių klasių modelio fragmento, pateikiamo 4.11 pav., turėtų būti sukuriama DB schema su papildomomis metalentelėmis. Ji pavaizduota 4.12 paveiksle.

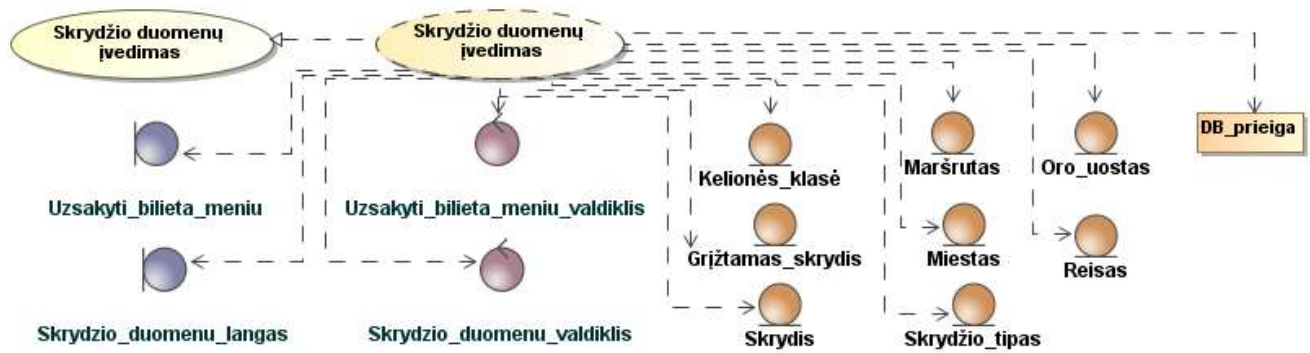


4.11 pav. Faktų tipų ir struktūrinių taisyklių transformavimas į esybių klasių modelį

„Objekto_tipo_pavadinimas“, „Vaidmens_pavadinimas“ ir „Rysio_pavadinimas“ metalentelėse pateiktas fakto tipas „Uzsakytas_skrydis turi telefoną“. Į šias metalenteles rašomi visi faktų tipai.



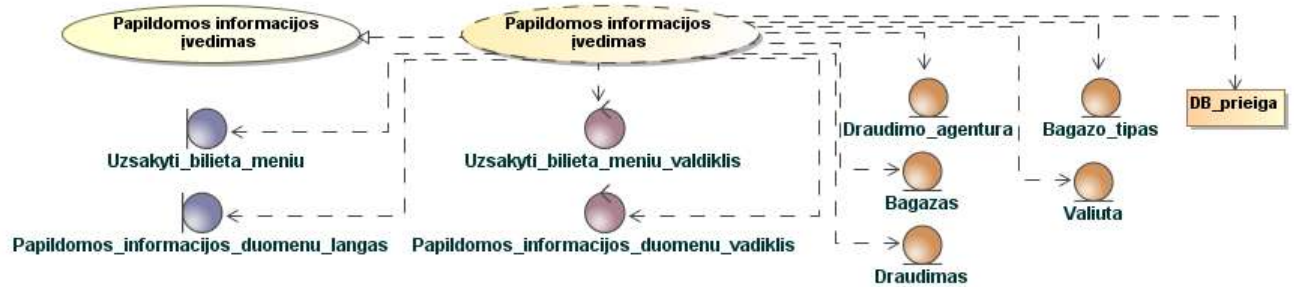
4.12 pav. Sugeneruota DB schema iš esybių klasių modelio



4.14 pav. PA “Skrydžio duomenų įvedimas” realizacijos diagrama



4.15 pav. PA “Keleivio duomenų įvedimas” realizacijos diagrama



4.16 pav. PA “Papildomos informacijos duomenų įvedimas” realizacijos diagrama

Skrydžio registracijos analizės schema pateikta 4.17 paveiksle, o 4.18 pav. Pavaizduota skrydžio registracijos sistemos panaudojimo atvejų realizacijos diagrama.



4.17 pav. Skrydžio registracijos analizės diagrama

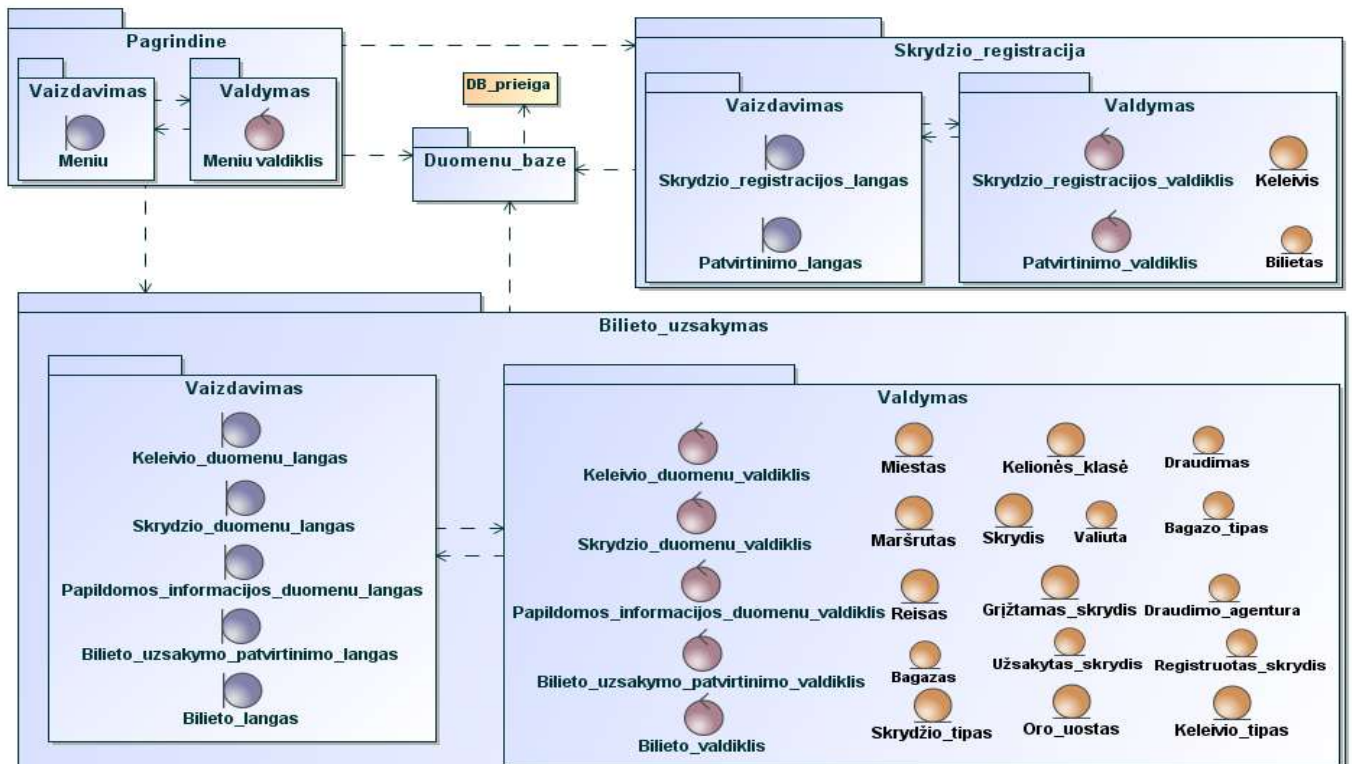


4.18 pav. PA “Registruotis skrydžiui” realizacijos diagrama

4.3.2. Loginė visos sistemos architektūra

Eksperimentinė skrydžio užsakymo sistema yra išskaidoma į atskirus paketus. Kiekvienam paketui yra taikomas trijų lygių architektūros modelis.

Kuriamų paketų priklausomybė ir klasės, kurios bus realizuotos atitinkamuose paketuose, pateiktos 4.19 paveiksle.



4.19 pav. Trijų lygių loginė architektūros schema

4.3.3. Vartotojo paslaugos

Vartotojo navigavimo planą matome 4.20 paveikslėlyje.

Norint dirbti su skrydžio užsakymo sistema, ją reikia pasileisti t.y. atsidaro „Pagrindinis“ langas. „Pagrindiniame lange“ yra tik meniu juosta. Meniu juostos funkcijos: užsakyti bilietą, skrydžio registracija ir pagalba.



4.20 pav. Vartotojo navigavimo planas

Pasirinkus „Bilieto uzsakymo meniu“, atidaromi „Skrydžio duomenų“, „Keleivio duomenų“ ir „Papildomos informacijos duomenų“ langai. „Skrydžio duomenų“ lange galima įvesti skrydžio duomenis. Iš šio lango, taip pat galima iškviesti „Keleivio duomenų“ langą, tik reikia paspausti mygtuką „Uzsakyti“. Šiame lange įvedami keleivio duomenys. Iš „Keleivio duomenų“ lango, taip pat galima iškviesti „Papildomos informacijos duomenų“ langą, tik reikia paspausti mygtuką „Patvirtinti“. Šiame lange įvedami papildomos informacijos duomenys. Teisingai įvedus visus duomenis ir paspaudus mygtuką „Patvirtinti“, atsiranda „Bilieto uzsakymo patvirtinimo“ langas. Lange paspaudus mygtuką „Taip“, gaunamas „Bilietas“.

Pasirinkus „Skrydžio registracijos meniu“, atidaromas „Skrydžio registravimo“ langas, kuriame įvedami skrydžio registracijos duomenys. Jei teisingai užregistruotas skrydis, gaunamas registruoto skrydžio „Patvirtinimo“ langą.

Pasirinkus „Pagalbos meniu“, atidaromas „Pagalbos“ langas, kuriame galima rasti papildomos informacijos, susijusios su bagažu, kelione ir t.t. (pvz., kai vartotojas renkasi bagažą, jis turėtų rasti tokios informacijos: kiek ir kokio bagažo jis gali pasirinkti, kiek jam tai kainuos ir pan.)

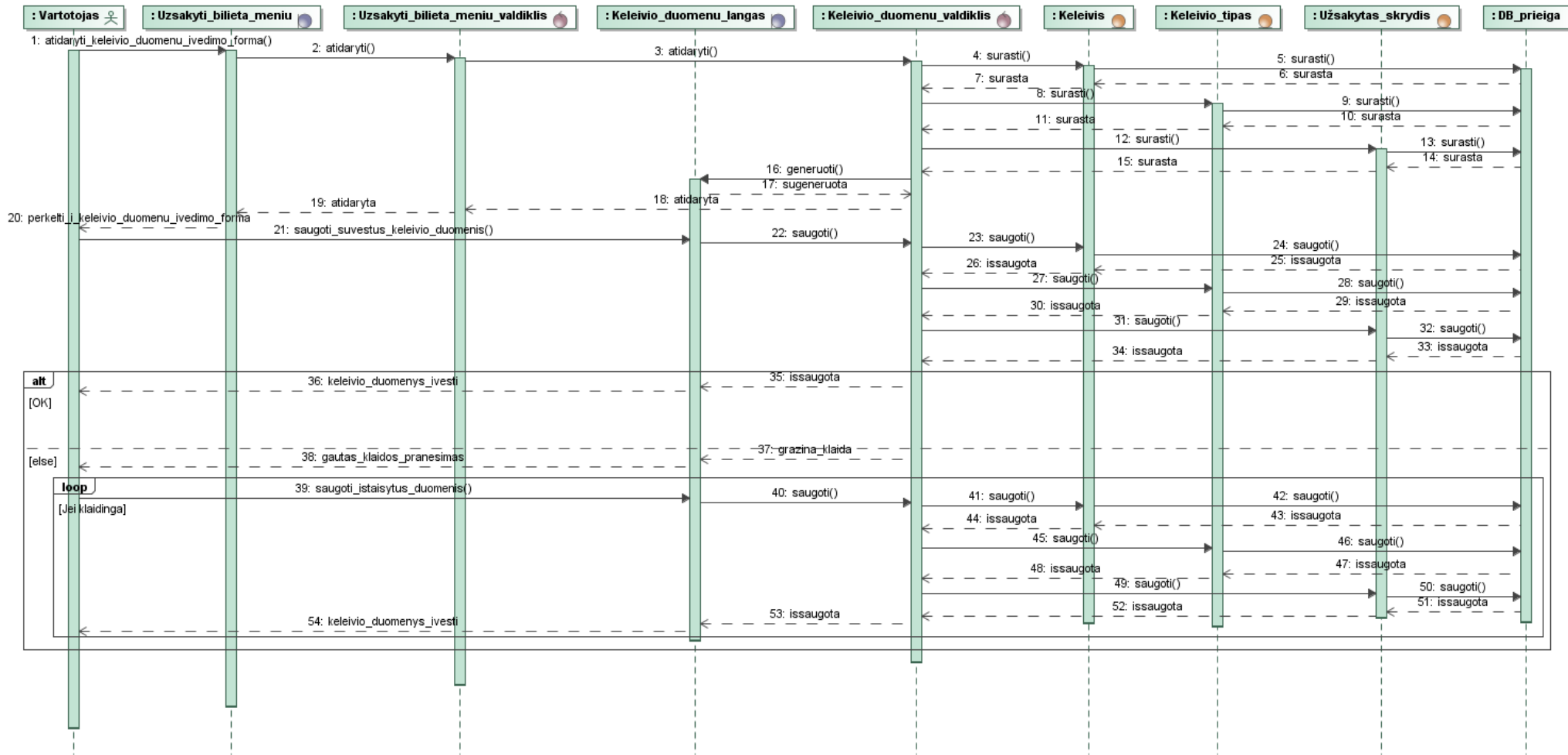
4.3.4. Sistemos elgsenos modeliai

Sistemos elgsenos modelis – tai modelis, kuriame parodomi konkretūs veiksmai atliekami su sistema.

Sekų diagramos vaizduoja bendradarbiavimą tarp ribinių, valdymo ir esybių klasių.

Vartotojo sekų diagramos pateikiamos 4.21 ir 4.22 paveiksluose.

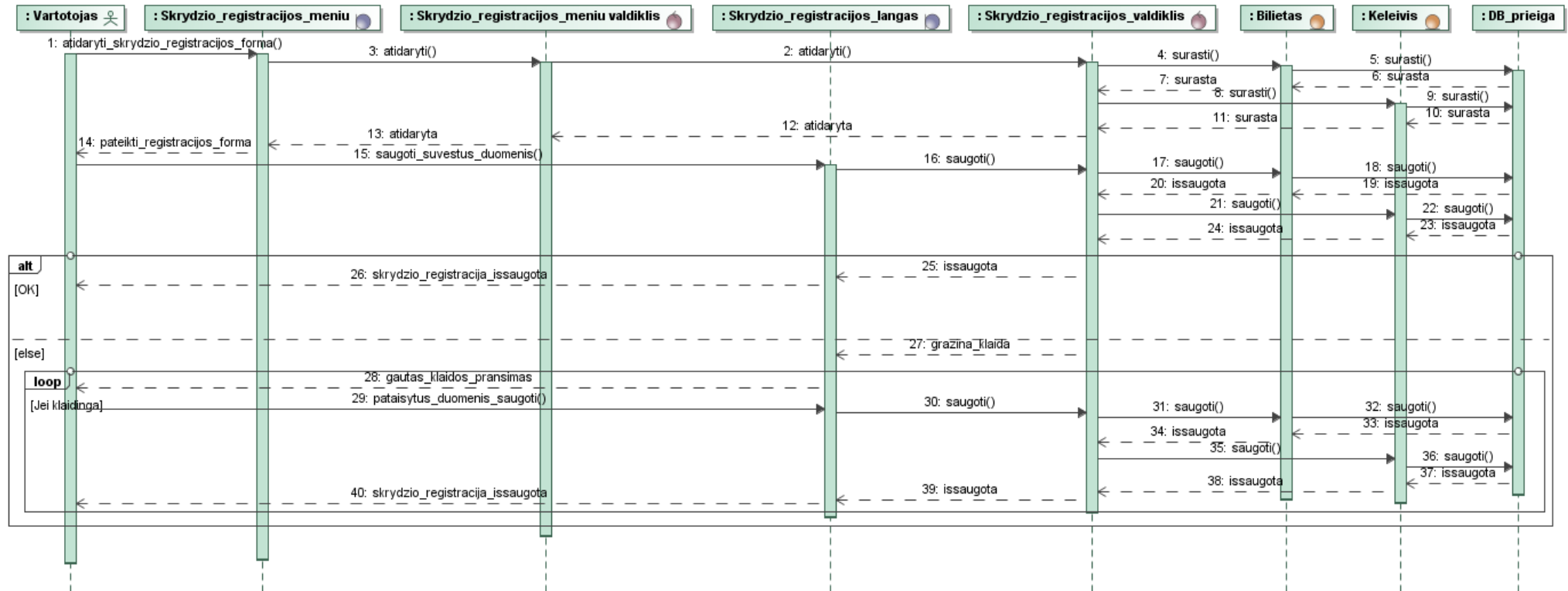
Panaudojimo atvejo „Įvesti keleivio duomenis“ sekos diagrama matoma 4.21 paveiksle.



4.21 pav. Keleivio duomenų įvedimo sekų diagrama

Analogiškos sekų diagramos yra panaudojimo atvejų „Įvesti skrydžio duomenis“ ir „Įvesti papildomos informacijos duomenis“.

Panaudojimo atvejo „Registruotis skrydžiui“ sekos diagrama matoma 4.22 paveiksle.

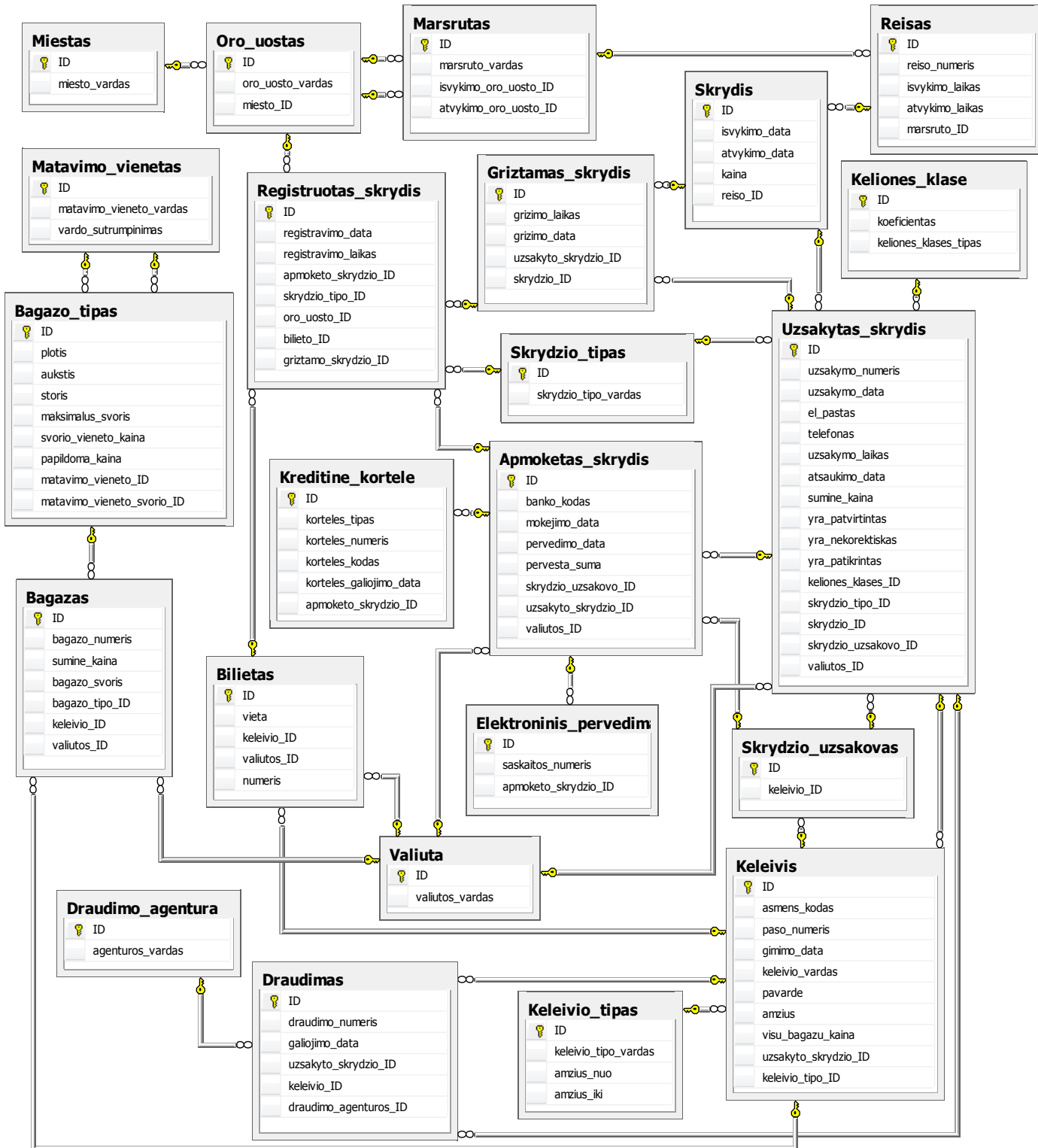


4.22 pav. Skrydžio registracijos sekų diagrama

4.3.5. Duomenų bazės schema

Iš veiklos konceptų sudaryta duomenų bazės schema, kuri matoma 4.23 paveiksle.

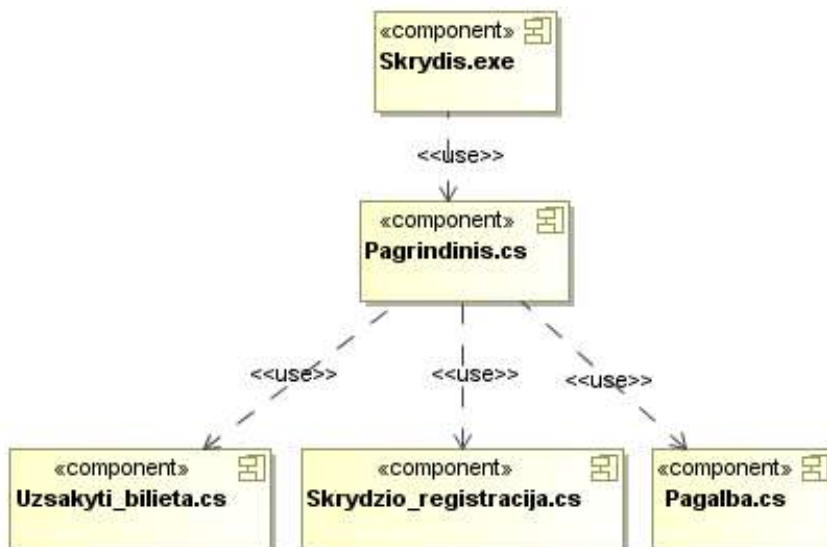
Eksperimentinės sistemos semantikos išsaugojimo metalentelės pateiktos 3.5 ir 3.8 paveiksluose.



4.23 pav. Duomenų bazės schema

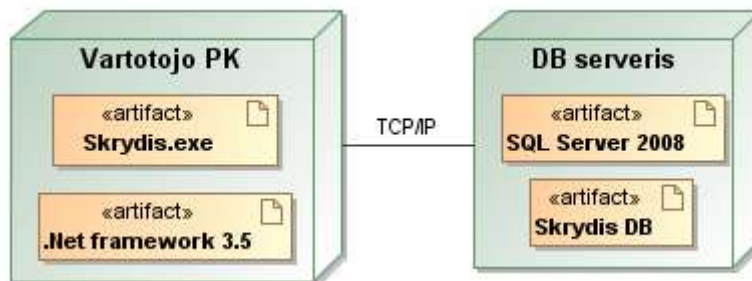
4.3.6. Realizacijos modelis

Į Skrydis.exe komponentą įeina Pagrindinis, kuris susideda iš: Bilieto_uzsakymas, Skrydzio_registracija ir Pagalba . Skrydzio užsakymo sistemos komponentų diagrama pateikiama 4.24 paveiksle.



4.24 pav. Skrydzio užsakymo sistemos komponentų diagrama

Skrydzio užsakymo sistemos diegimo diagrama pavaizduota 4.25 paveiksle.



4.25 pav. Skrydzio užsakymo sistemos diegimo diagrama

Duomenų bazių serveryje turi būti įdiegtas *Microsoft SQL Server 2008* ir Skrydis DB (duomenų bazių failus). Vartotojo ar administratoriaus kompiuteryje būtina įdiegti .Net framework 3.5 versiją ir Skrydis.exe failą.

5. Eksperimentinės skrydžio užsakymo sistemos realizacija

Šiame skyriuje aprašoma eksperimentinė skrydžio užsakymo sistema ir kaip ji veikia.

5.1. Skrydžio užsakymo sistemos paskirtis, pagrindinės funkcijos

Pagrindinė eksperimentinės skrydžio užsakymo sistemos paskirtis – skrydžio bilieto užsakymo sudarymas bei skrydžio registravimo įvykdymas.

Eksperimentinės skrydžio užsakymo sistemos pagrindinės funkcijos: skrydžio paieška pagal įvestus kriterijus, skrydžio/keleivio/papildomos informacijos įvedimas, skrydžio registravimas, bei pagalbos peržiūra.

Eksperimentinė skrydžio užsakymo sistema realizuota su *Microsoft SQL Server 2008* (duomenų bazės kūrimui ir *SQL* šablonų realizavimui) ir *Microsoft Visual Studio 2008* (vartotojo interfeiso kūrimui) programomis.

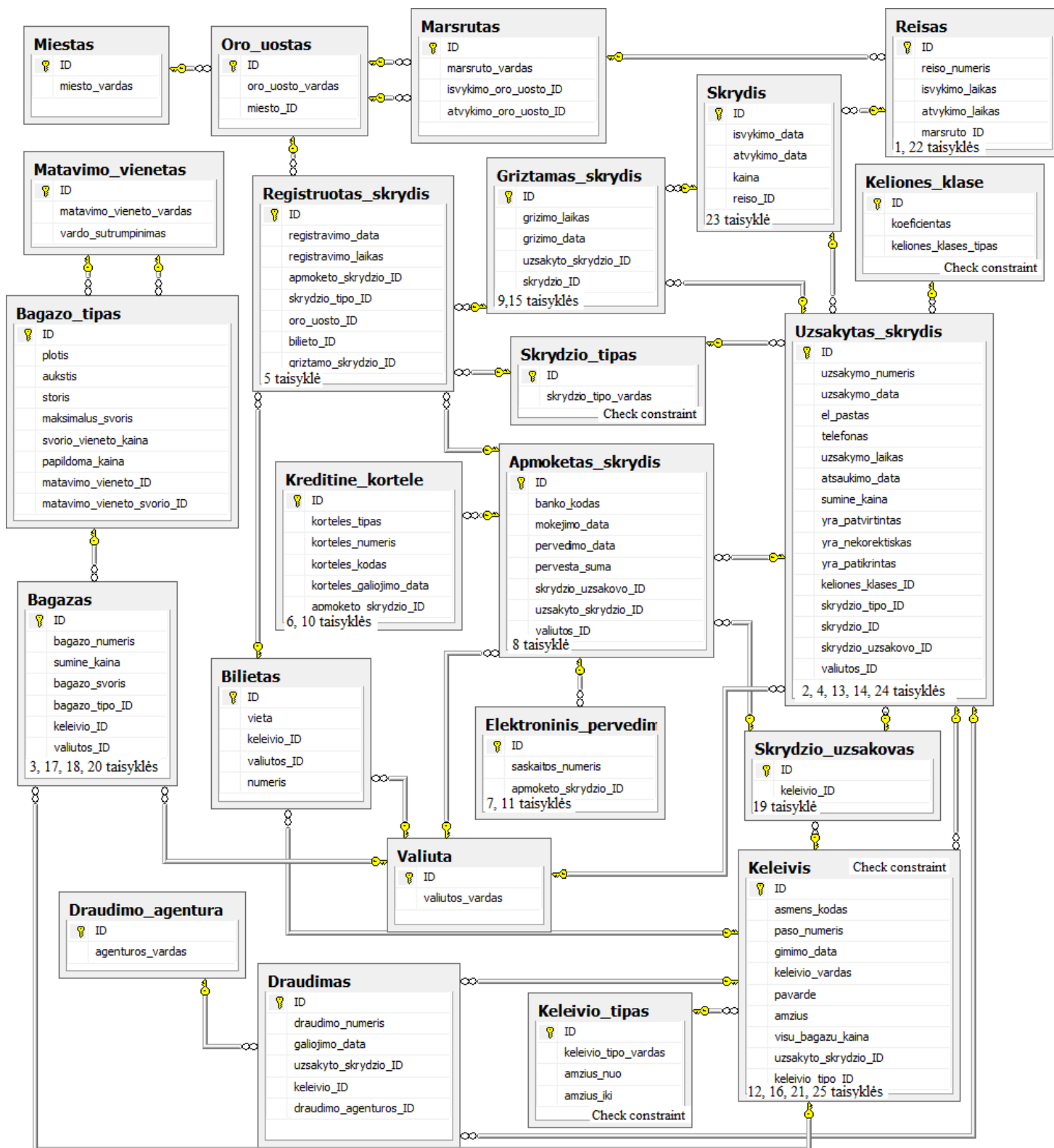
5.2. Realizuotos eksperimentinės skrydžio užsakymo sistemos duomenų bazės schema ir lentelių specifikacijos

Išsamesnė eksperimentinės skrydžio užsakymo sistemos duomenų bazės schema pateikta 5.1 paveiksle. Šitoje schemoje pateikta DB lentelių išsamesnė informacija, t.y. trigeriai (vadinami taisyklėmis) ir apribojimai (*Check constraint*).

Trigeriai realizuoti šitose DB lentelėse: „Registruotas_skrydis“, „Reisas“, „Skrydis“, „Griztamas_skrydis“, „Kreditine_kortele“, „Apmoketas_skrydis“, „Uzsakytas_skrydis“, „Bagazas“, „Elektroninis_pervedimas“, „Skrydzio_uzsakovas“, „Keleivis“.

Apribojimai realizuoti šitose DB lentelėse: „Keliones_klase“, „Skrydzio_tipas“, „Keleivis“, „Keleivio_tipas“.

Semantikos išsaugojimo metalentelės pateiktos 3.5 ir 3.8 paveiksluose.



5.1 pav. Eksperimentinės skrydžio užsakymo sistemos duomenų bazės schema

Duomenų bazės schemos detalūs lentelių aprašymai pateikti 5.1 lentelėje.

5.1 lentelė. Lentelių specifikacijos

ATRIBUTAS	TIPAS	PASKIRTIS
MIESTAS		
ID	int	Pirminis raktas
MIESTO_VARDAS	varchar(50)	Miesto pavadinimas, vardas
ORO_UOSTAS		
ID	int	Pirminis raktas
ORO_UOSTO_VARDAS	varchar(50)	Oro uosto pavadinimas, vardas
MIESTO_ID	int	Išorinis raktas. Miestas
MARSRUTAS		
ID	int	Pirminis raktas
MARSRUTO_VARDAS	varchar(100)	Maršruto pavadinimas, vardas
ISVYKIMO_ORO_UOSTO_ID	int	Išorinis raktas. Oro uostas
ATVYKIMO_ORO_UOSTO_ID	int	Išorinis raktas. Oro uostas
REISAS		
ID	int	Pirminis raktas
REISO_NUMERIS	varchar(50)	Reiso numeris
ISVYKIMO_LAIKAS	time(7)	Reiso išvykimo laikas
ATVYKIMO_LAIKAS	time(7)	Reiso atvykimo laikas
MARSRUTO_ID	int	Išorinis raktas. Maršrutas
GRIZTAMAS_SKRYDIS		
ID	int	Pirminis raktas
UZSAKYTO_SKRYDZIO_ID	int	Išorinis raktas. Užsakytas skrydis
SKRYDZIO_ID	int	Išorinis raktas. Skrydis
GRIZIMO_DATA	date	Grįžtamo skrydžio grįžimo data
GRIZIMO_LAIKAS	time(7)	Grįžtamo skrydžio grįžimo laikas
DRAUDIMAS		
ID	int	Pirminis raktas
DRAUDIMO_NUMERIS	varchar(50)	Draudimo numeris
GALIOJIMO_DATA	date	Draudimo galiojimo data
UZSAKYTO_SKRYDZIO_ID	int	Išorinis raktas. Užsakytas skrydis
KELEIVIO_ID	int	Išorinis raktas. Keleivis
DRAUDIMO_AGENTUROS_ID	int	Išorinis raktas. Draudimo agentūra
SKRYDZIO_UZSAKOVAS		
ID	int	Pirminis raktas
KELEIVIO_ID	int	Išorinis raktas. Keleivis
APMOKETAS_SKRYDIS		
ID	int	Pirminis raktas
BANKO_KODAS	varchar(50)	Banko kodas
MOKEJIMO_DATA	date	Apmokėto skrydžio mokėjimo data
PERVEDIMO_DATA	date	Apmokėto skrydžio pervedimo data
PERVESTA_SUMA	float	Apmokėto skrydžio pervesta suma
SKRYDZIO_UZSAKOVO_ID	int	Išorinis raktas. Skrydžio užsakovas
UZSAKYTO_SKRYDZIO_ID	int	Išorinis raktas. Užsakytas skrydis
VALIUTOS_ID	int	Išorinis raktas. Valiuta

BAGAZAS		
ID	int	Pirminis raktas
BAGAZO_NUMERIS	varchar(50)	Bagažo numeris
SUMINE_KAINA	float	Bagažo suminė kaina
BAGAZO_SVORIS	int	Bagažo svoris
BAGAZO_TIPO_ID	int	Išorinis raktas. Bagažo tipas
KELEIVIO_ID	int	Išorinis raktas. Keleivis
VALIUTOS_ID	int	Išorinis raktas. Valiuta
KELEIVIS		
ID	int	Pirminis raktas
ASMENS_KODAS	int	Keleivio asmens kodas
PASO_NUMERIS	varchar(50)	Keleivio paso numeris
GIMIMO_DATA	date	Keleivio gimimo data
KELEIVIO_VARDAS	varchar(50)	Keleivio vardas
PAVARDE	varchar(50)	Keleivio pavardė
UZSAKYTO_SKRYDZIO_ID	int	Išorinis raktas. Užsakytas skrydis
KELEIVIO_TIPO_ID	int	Išorinis raktas. Keleivio tipas
VISU_BAGAZU_KAINA	float	Keleivio visų bagažų kaina (suma)
AMZIUS	int	Keleivio amžius
BAGAZO_TIPAS		
ID	int	Pirminis raktas
PLOTIS	int	Bagažo plotis
AUKSTIS	int	Bagažo aukštis
STORIS	int	Bagažo storis
SVORIS	int	Bagažo svoris
MAKSIMALUS_SVORIS	int	Bagažo maksimalus svoris
SVORIO_VIENETO_KAINA	float	Vieno bagažo svorio kaina
PAPILDOMA_KAINA	float	Bagažo papildoma kaina
MATAVIMO_VIENETO_ID	int	Išorinis raktas. Matavimo vienetas
MATAVIMO_VIENETO_SVORIO_ID	int	Išorinis raktas. Matavimo vienetas
UZSAKYTAS_SKRYDIS		
ID	int	Pirminis raktas
UZSAKYMO_NUMERIS	varchar(50)	Užsakyto skrydžio užsakymo numeris
UZSAKYMO_DATA	date	Užsakyto skrydžio užsakymo data
SUMINE_KAINA	float	Užsakyto skrydžio suminė kaina
YRA_PATVIRTINTAS	bit	Yra patvirtintas?
YRA_NEKOREKTISKAS	bit	Yra nekorektiškas?
YRA_PATIKRINTAS	bit	Yra patikrintas?
ATSAUKIMO_DATA	date	Užsakyto skrydžio atšaukimo data
EL_PASTAS	varchar(50)	Užsakyto skrydžio elektroninis paštas
TELEFONAS	int	Užsakyto skrydžio telefono numeris
KELIONES_KLASES_ID	int	Išorinis raktas. Kelionės klasė
SKRYDZIO_TIPO_ID	int	Išorinis raktas. Skrydžio tipas
SKRYDZIO_ID	int	Išorinis raktas. Skrydis
SKRYDZIO_UZSAKOVO_ID	int	Išorinis raktas. Skrydžio užsakovas
UZSAKYMO_LAIKAS	time(7)	Užsakyto skrydžio užsakymo laikas
VALIUTOS_ID	int	Išorinis raktas. Valiuta
KELEIVIO_TIPAS		
ID	int	Pirminis raktas
KELEIVIO_TIPO_VARDAS	varchar(50)	Keleivio tipo pavadinimas, vardas
AMZIUS_NUO	int	Keleivio amžius nuo
AMZIUS_IKI	int	Keleivio amžius iki

ELEKTRONINIS_PERVEDIMAS		
ID	int	Pirminis raktas
SASKAITOS_NUMERIS	varchar(50)	Elektroninio pervedimo sąskaitos numeris
APMOKETO_SKRYDZIO_ID	int	Išorinis raktas. Apmokėtas skrydis
VALIUTA		
ID	int	Pirminis raktas
VALIUTOS_VARDAS	varchar(50)	Valiutos pavadinimas, vardas
BILIETAS		
ID	int	Pirminis raktas
VIETA	int	Vieta, lėktuvo sėdynės numeris
KELEIVIO_ID	int	Išorinis raktas. Keleivis
VALIUTOS_ID	int	Išorinis raktas. Valiuta
NUMERIS	int	Bilieto numeris
MATAVIMO_VIENETAS		
ID	int	Pirminis raktas
MATAVIMO_VIENETO_VARDAS	varchar(50)	Matavimo vieneto vardas
VARDO_SUTRUMPINIMAS	varchar(5)	Matavimo vieneto vardo sutrumpinimas
SKRYDIS		
ID	int	Pirminis raktas
ISVYKIMO_DATA	date	Skrydžio išvykimo data
ATVYKIMO_DATA	date	Skrydžio atvykimo data
REISO_ID	int	Išorinis raktas. Reisas
KAINA	float	Skrydžio kaina
KREDITINE_KORTELE		
ID	int	Pirminis raktas
KORTELES_TIPAS	varchar(50)	Kreditinės kortelės tipas
KORTELES_NUMERIS	varchar(50)	Kreditinės kortelės numeris
KORTELES_KODAS	varchar(50)	Kreditinės kortelės kodas
KORTELES_GALIOJIMO_DATA	date	Kreditinės kortelės galiojimo data
APMOKETO_SKRYDZIO_ID	int	Išorinis raktas. Apmokėtas skrydis
DRAUDIMO_AGENTURA		
ID	int	Pirminis raktas
AGENTUROS_VARDAS	varchar(50)	Agentūros vardas, pavadinimas
REGISTRUOTAS_SKRYDIS		
ID	int	Pirminis raktas
REGISTRAVIMO_DATA	date	Registruoto skrydžio registravimo data
REGISTRAVIMO_LAIKAS	time(7)	Registruoto skrydžio registravimo laikas
APMOKETO_SKRYDZIO_ID	int	Išorinis raktas. Apmokėtas skrydis
SKRYDZIO_TIPO_ID	int	Išorinis raktas. Skrydžio tipas
ORO_UOSTO_ID	int	Išorinis raktas. Oro uostas
BILIETO_ID	int	Išorinis raktas. Bilietas
GRIZTAMO_SKRYDZIO_ID	int	Išorinis raktas. Grįžtamas skrydis
KELIONES_KLASE		
ID	int	Pirminis raktas
KELIONES_KLASES_TIPAS	varchar(50)	Kelionės klasės tipo pavadinimas
KOEFICIENTAS	float	Kelionės klasės koeficientas kainai

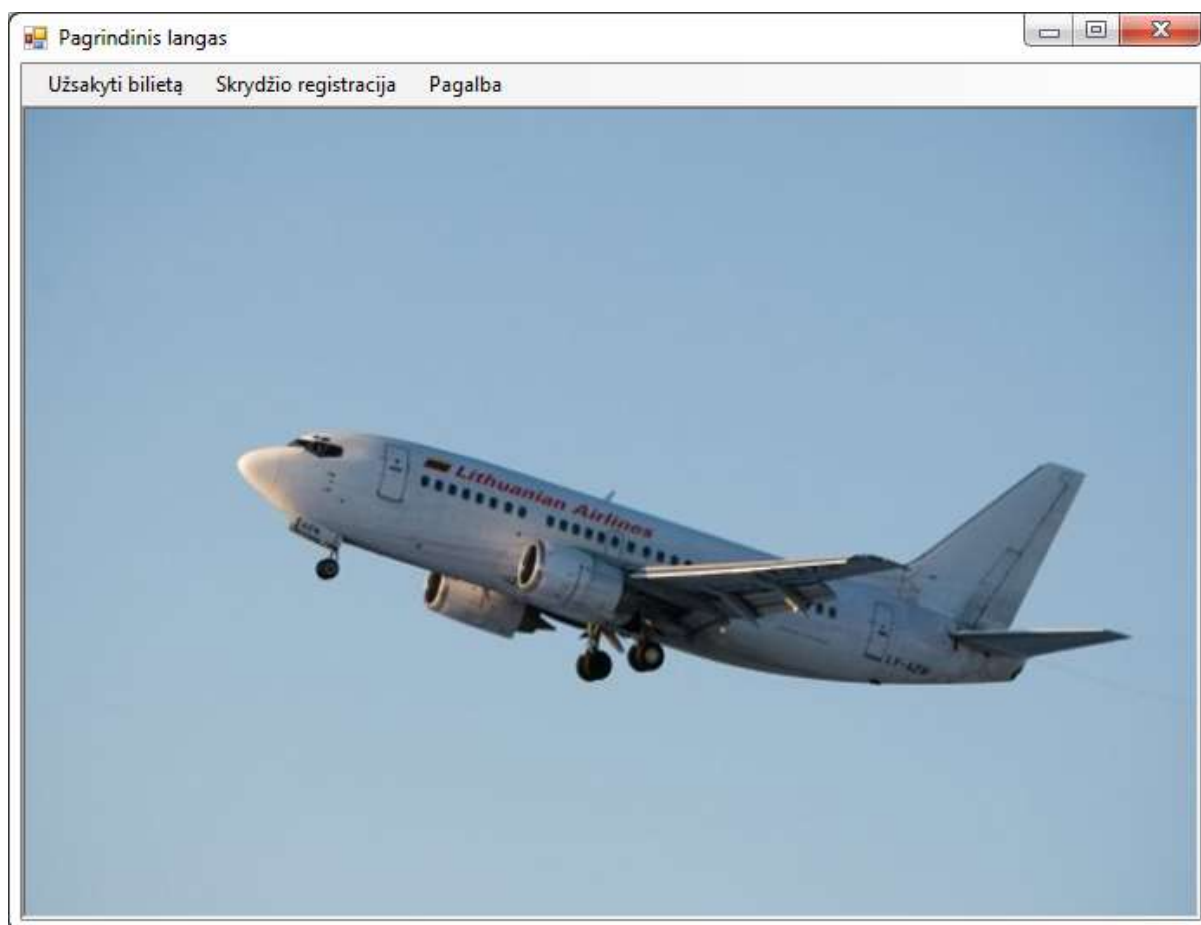
SKRYDZIO_TIPAS		
ID	int	Pirminis raktas
SKRYDZIO_TIPO_VARDAS	varchar(50)	Skrydžio tipo pavadinimas, vardas

5.3. Skrydžio užsakymo sistemos veikimo aprašymas

Kaip viskas vyksta skrydžio srityje, sprendžiama kuriant procesą ir taisykles, o realizacija turi tai atitikti. Eksperimentinė skrydžio užsakymo sistema sukurta bilieto užsakymo ir skrydžio registravimo procesams.

Vartotojui paleidus eksperimentinę programą, atsidaro „Pagrindinis langas“. Šioje formoje vartotojas mato tris galimus veiksmus: užsakyti bilietą (sudaryti skrydžio užsakymo prašymą), registruotis skrydžiui (jei ja turi bilietą) ir peržiūrėti pagalbą (veiklos taisyklės, kurios gali būti reikalingos, iškilus neaiškumams, vartotojui).

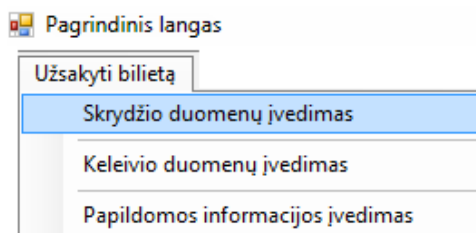
Eksperimentinės skrydžio užsakymo sistemos „Pagrindinis langas“ pateiktas 5.2 paveiksle.



5.2 pav. Eksperimentinės skrydžio užsakymo sistemos „Pagrindinis langas“

- Skrydžio (bilieto) užsakymo procesas:

Vartotojas, norėdamas užsisakyti bilietą, turi suvesti skrydžio, keleivio (savo) ir papildomos informacijos duomenis. Šita eiga pateikta 5.3 paveiksle.



5.3 pav. „Užsakyti bilietą“ eiga

Pasirinkus „Skrydžio duomenų įvedimą“, pateikiama „Skrydžio užsakymo“ forma. Ši forma pateikta 5.4 paveiksle.

Suvedus visus reikalingus duomenis, bus įvykdytas skrydžio duomenų įvedimas.

	Reiso numeris	Maršrutas	Iš oro uosto	Į oro uostą	Išvykimo data	Išvykimo laikas	Ki
	1	Vilnius - Kaunas	Vilnius	Kaunas	2010.10.01	08:00:00	50
	1	Vilnius - Kaunas	Vilnius	Kaunas	2010.10.11	09:10:10	42
	3	Vilnius - Kaunas	Vilnius	Kaunas	2011.01.27	10:10:10	50
	3	Vilnius - Kaunas	Vilnius	Kaunas	2011.01.27	10:10:10	50

Kelionės informacija

Skrydžio tipas: | vieną pusę | abi puses

Grįžimo data: 2011 m. sausis 26 d.

Grįžimo laikas: 12:15:40

Kelionės klasė: Ekonominė klasė

Užsakyti

5.4 pav. Skrydžio duomenų įvedimo forma

Užpildžius skrydžio duomenis, pateikiama „Keleivio informacijos“ forma. Ši forma pateikta 5.5 paveiksle.

Suvedus visus reikalingus duomenis, bus įvykdytas keleivio duomenų įvedimas.

Uždakymo data: 2011-01-26 12:35

Vardas: Raminta

Pavardė: Zeknyte

Gimimo data: 1986 m. sausis 4 d. Amžius: 25 Keleivio tipas: Suaugęs keleivis

Telefono numeris: +37066666666

El. pašto adresas: 1@2.lt

Asmens kodas: 2222222222

Paso numeris: 1111111122

Atšaukti Patvirtinti

5.5 pav. Keleivio duomenų įvedimo forma

Užpildžius keleivio duomenis, pateikiama „Papildomos informacijos“ forma. Ši forma pateikta 5.6 paveiksle.

Suvedus visus reikalingus duomenis, bus įvykdytas papildomos informacijos duomenų įvedimas.

Draudimo informacija

Draudimo agentūra: ePolisas

Draudimo nr.: 4352673

Galioja iki: 2011 m. vasaris 5 d.

Bagažo informacija

Registruoti Šalinti Vieno bagazo tipo maksimalus svoris yra 20 kg

Bagažo numeris	Bagažo tipas	Bagažo svoris
1	10x10x10, Svorio vieneto kaina, Lt.: 0,5	12
2	5x5x5, Svorio vieneto kaina, Lt.: 0,5	34
3	20x20x20, Svorio vieneto kaina, Lt.: 2	36

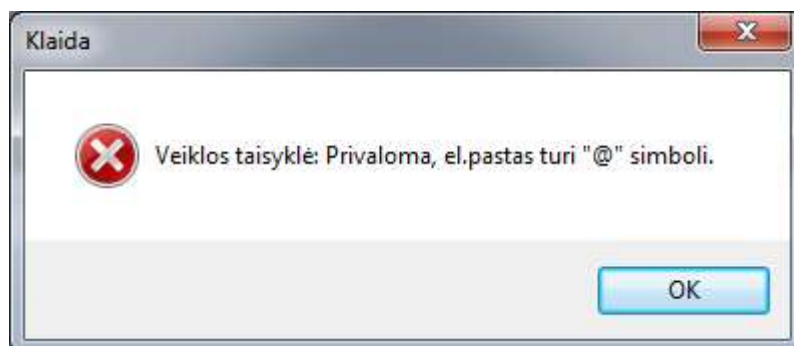
Mokėjimo informacija

Mokėjimo tipas: dropdown

Atšaukti Patvirtinti

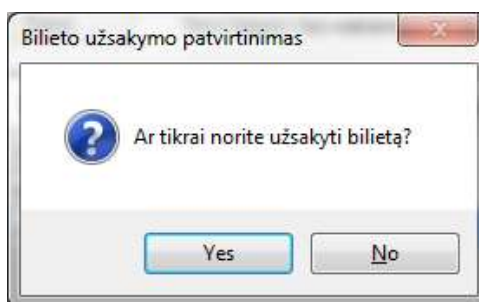
5.6 pav. Papildomos informacijos duomenų įvedimo forma

Suvedus visus reikalingus duomenis ir paspaudus patvirtinimo mygtuką, pateikiama „Klaidos“ formą (5.7 pav.), jei vartotojas įveddamas el. pašta, neįrašė „@“ simbolio.



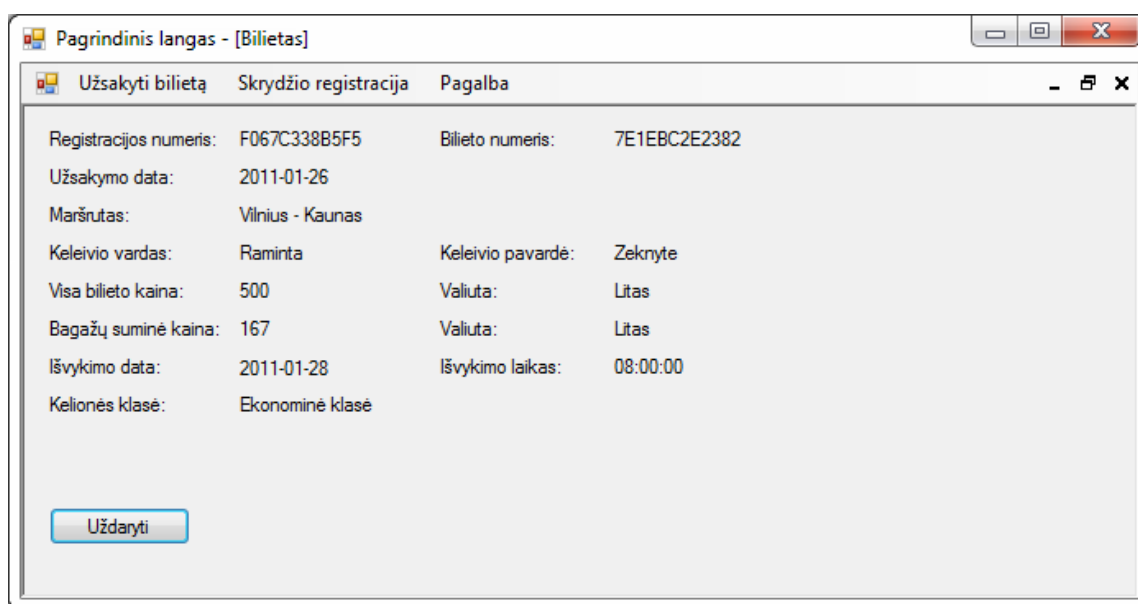
5.7 pav. Klaidos pranešimo forma

Suvedus visus teisingai reikalingus duomenis ir paspaudus patvirtinimo mygtuką, pateikiama „Bilieto užsakymo patvirtinimo“ forma (5.8 pav.).



5.8 pav. Bilieto užsakymo patvirtinimo forma

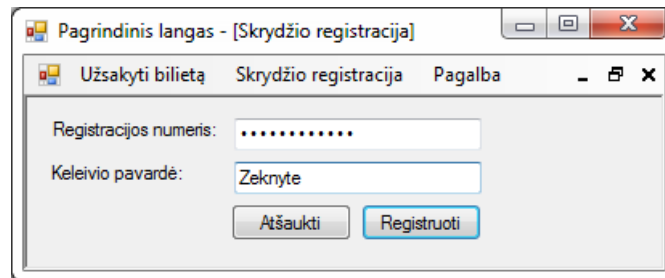
Patvirtinus bilieto užsakymą, pateikiama „Bilieto“ forma (5.9 pav.).



5.9 pav. Bilieto forma

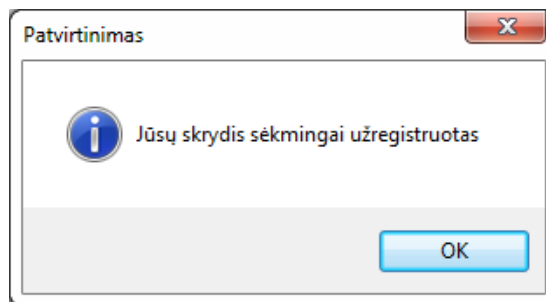
- Registravimo procesas:

Gavus bilietą, vartotojas gali registruotis skrydžiui. „Skrydžio registracijos“ forma pateikta 5.10 paveiksle.



5.10 pav. Skrydžio registracijos forma

Jeigu duomenys suvesti teisingai, tai bus pateikiamas sėkmingos registracijos „Patvirtinimo“ forma. Ši forma pateikta 5.11 paveiksle.

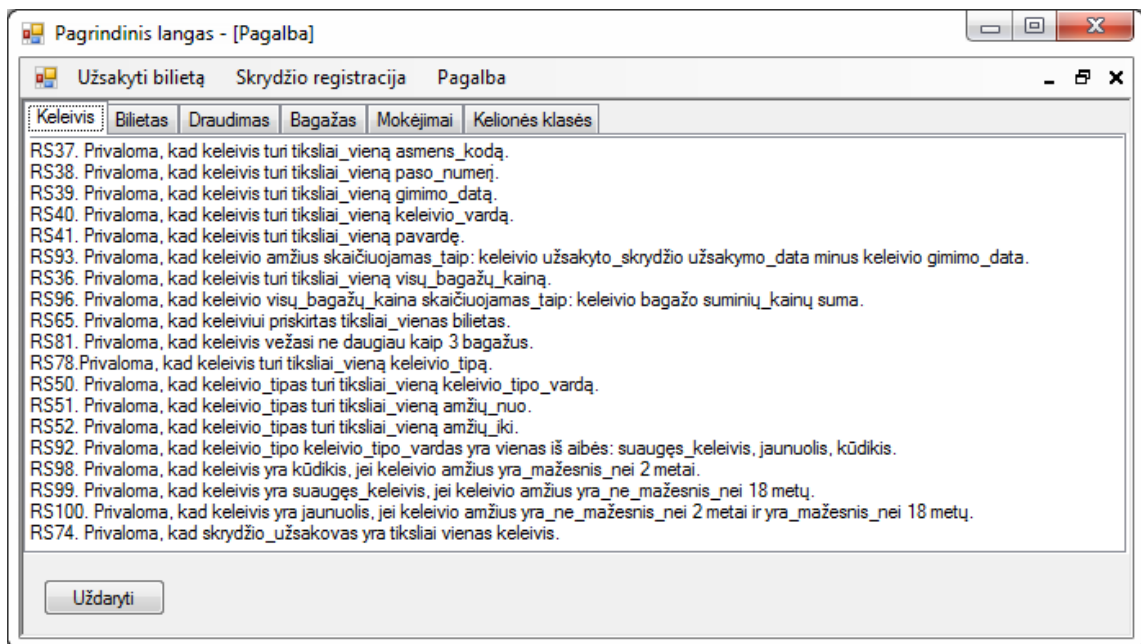


5.11 pav. Patvirtinimo forma

- Pagalbos peržiūra

Iškylus neaiškumams, vartotojas gali peržiūrėti „Pagalbą“. Šitoje formoje vartotojas galės susirasti jam reikiamos informacijos apie: keleivį, bilietą, draudimą, bagažą, mokėjimą, bei kokios kelionės klasės yra siūlomos.

„Pagalbos“ forma pateikta 5.12 paveiksle.



5.12 pav. Pagalbos forma

5.3.1. Skrydžio užsakymo sistemos testavimo planas, duomenys ir rezultatai

Testavimo tikslas yra patikrinti ar realizuota skrydžio užsakymo sistema teisingai vykdo visus panaudojimo atvejus, t.y. ar šios sistemos veikimas atitinka panaudojimų atvejų specifikacijas. Skrydžio užsakymo sistemos testavimui yra sudaromas testavimo planas, kuriame atsispindi, kokius veiksmus reikia atlikti testuojant šią sistemą, taip pat, kokia turi būti šitos sistemos reakcija į konkrečius veiksmus.

Skrydžio užsakymo sistemos testavimo planas yra pateiktas 5.2 lentelėje.

5.2 lentelė. Testavimo planas

NR.	TESTAVIMO VEIKSMŲ SEKA	LAUKIAMA SISTEMOS REAKCIJA
1. Skrydžio (keleivio, papildomos informacijos) duomenų įvedimas		
1.1	Spausti mygtuką „Paieška“ neužpildžius skrydžio paieškos duomenų.	Pranešimas „Nenurodytas išvykimo/atvykimo miestas“.
1.2	Neteisingai įvesti grįžimo datą (ankstesnę, nei skrydžio) ir paspausti mygtuką „Užsakyti“.	Pranešimas: „Nenurodyta sugrįžimo data“.
1.3	Įvesti 18 metų neturinčio keleivio gimimo datą ir paspausti mygtuką „Patvirtinti“.	Pranešimas „Keleivis turi būti ne jaunesnis nei 18 m.“.
1.4	Teisingai įvesti ir pasirinkti privalomi laukai: oro uostas, išvykimo miestas, atvykimo miestas. Paspausti mygtuką „Paieška“.	Skrydžio užsakymo sistema pateikia skrydžių sąrašą.
1.5	Teisingai įvesti skrydžio, keleivio ir papildomos informacijos duomenys. Paspausti mygtuką „Patvirtinti“.	Pranešimas: „Ar tikrai norite užsakyti bilietą?“.
2. Registruotis skrydžiui		
2.1	Spausti mygtuką „Registruotis skrydžiui“ neužpildžius skrydžio registracijos formos.	Skrydžio užsakymo sistema neregistruoja skrydžio.
2.2	Teisingai įvesti registracijos laukai: registracijos numeris ir keleivio pavardė. Paspausti mygtuką „Registruoti“.	Pranešimas: „Jūsų skrydis sėkmingai užregistruotas“.

5.4. Metodikos įvertinimas remiantis eksperimentinės sistemos analize

Šablonų rinkinio tinkamumui ir įgyvendinamumui patikrinti jais buvo aprašyta eksperimentinė skrydžių užsakymų sistema, kuri buvo realizuota *Microsoft SQL Server 2008* ir *Microsoft Visual Studio 2008* priemonėmis.

Eksperimentinėje skrydžio užsakymo sistemoje realizuotos 159 veiklos taisyklės (26 operacinės ir 133 struktūrinės taisyklės). Struktūrinių taisyklių tipai apima išvedimo, procesų, skaičiavimų, kardinalumo ir unikalumo, duomenų turinio ir formatų taisykles.

5.3 lentelėje pateikiami šiame darbe sukurtų šablonų ir pagal juos užrašytų taisyklių skaičiai.

5.3 lentelė. Metoduose egzistuojančių šablonų palyginimas

Metodai	Šablonai	Operacinių taisyklių	Struktūrinių taisyklių	Išvedimo taisyklių	Procesų	Kardinalumo ir unikalumo	Duomenų turinio	Duomenų formatų	Skaičiavimų
<i>RuleSpeak</i>			+	+	+				
<i>SBVR</i>		+	+						
<i>Witt</i>					+	+	+		
Kuriamas metodas		+	+	+	+	+	+	+	+
Sukurtų šablonų skaičius		4	26	2	4	7	3	4	6
Pagal šabloną sukurtų taisyklių skaičius		26	133	3	7	94	5	17	7

5.5. Eksperimento išvados

1. Vykdamas eksperimentą, sukurti lietuviški šablonai buvo patikrinti realizuojant skrydžio užsakymo sistemą.
2. Atlikus eksperimentą, buvo įsitikinta, kad sukurtus struktūrizuotas lietuvių kalbos veiklos konceptų, faktų tipų ir veiklos taisyklių šablonus galima vienareikšmiškai realizuoti duomenų bazėje.
3. Kiekvienam konceptui, fakto tipui ar veiklos taisyklei buvo galima pritaikyti šabloną iš metodikoje aprašytų šablonų aibės.
4. Eksperimentinės skrydžio užsakymo sistemos realizacijoje veiklos taisyklės užrašomos ne tik *SQL* trigeriais ar procedūromis, bet ir pateikiamos pastabų laukeliuose.
5. Veiklos konceptus, faktų tipus ir veiklos taisykles galima atkurti iš duomenų bazės schemos, neprarandant semantikos.

6. Išvados

1. Literatūros analizė parodė, kad teisingas veiklos taisyklių užrašymas yra problema: sunku užfiksuoti veiklos taisykles taip, kad jos būtų vienareikšmiškai suprantamos, teisingos ir lengvai realizuojamos programinėje įrangoje.
2. Literatūroje yra pateikiamas šios problemos sprendimas - šablonų sukūrimas. Sukurti šablonai leidžia struktūrizuoti veiklos taisyklių užrašymą.
3. Lyginamoji analizė parodė, kad formuluoti lietuviškus šablonus reikia remiantis *SBVR* stiliumi ir papildyti juos *Witt* rekomendacijomis, kad šablonai būtų lengvai kompiuterizuojami ir reikiamo detalumo lygio. Buvo pritaikyti esami angliški *SBVR* šablonai ir įvesti nauji skaičiavimų, duomenų formatų, turinio šablonai taisyklėms specifikuoti.
4. Veiklos taisyklėms realizuoti pasirinktos *DBVS* priemonės ir *SQL* trigeriai ir procedūros, o eksperimentinei skrydžio užsakymo sistemai kurti - *Microsoft SQL Server 2008* ir *Microsoft Visual Studio 2008*.
5. *SBVR* metamodelio pagrindu buvo sudarytas struktūrizuotos lietuvių kalbos šablonų rinkinys, leidžiantis vienareikšmiškai aprašyti reikalavimus ir veiklos taisykles, kurias reikia specifikuoti norint sukurti DB schemą ir ja manipuluojančią programinę įrangą.
6. Norint išsaugoti atitikmenis tarp dalykinėje srityje naudojamų lietuviškų vardų bei jų kombinacijų ir DB lentelių bei stulpelių pavadinimų, kurie leidžia išsaugoti tik vieną žodžio formą ir nenaudoja lietuviškų simbolių, buvo sudarytos tam skirtos metalentelės, papildančios DB schemą.
7. Siekiant patikrinti sudarytų taisyklių šablonų tinkamumą IS analitikų ir realizuotojų poreikiams, kai veiklos taisyklės realizuojamos duomenų bazėse, buvo atliktas eksperimentas, kuris parodė, kad sukurta metodika leidžia vienareikšmiškai realizuoti visų nagrinėtų veiklos konceptų ir taisyklių aprašus duomenų bazėse ir išsaugo jų semantiką.
8. Sudarytus šablonus ir metalenteles tikslinga taikyti praktikoje, siekiant formaliai ir įvairiems vartotojams suprantamai aprašyti reikalavimus duomenų bazėms kurti, o ateityje – automatizuoti DB schemų generavimą iš *SBVR* veiklos žodynų.

9. Šio tyrimo pagrindu buvo išspausdintas straipsnis ir pristatytas pranešimas konferencijoje „Informacinės technologijos 2011 Mag&Dok IT2011“.

7. Literatūra

- [1] Graham, W. A Practical Method of Developing Natural Language Rule Statements (Part 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13), *Business Rules Journal*, Vol. 10, No. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 [žiūrėta 2009-10-30]. Prieiga per internetą: <<http://www.brcommunity.com>>
- [2] Defining Business Rules ~What Are They Really? The Business Rules Group, formerly, known as the GUIDE Business Rules Project, Final Report, revision 1.3, July, 2000, 1-77. Prieiga per internetą: <[http://www.docstoc.com/docs/11646109/Defining-Business-Rules-~-What-Are-They-Really-\(3rd-edition\)](http://www.docstoc.com/docs/11646109/Defining-Business-Rules-~-What-Are-They-Really-(3rd-edition))>
- [3] Barzdaitė L., Ribikauskas A. Verslo taisyklių formalaus specifikavimo galimybės. 10-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“, įvykusios Vilniuje 2007 m. balandžio 13 d. medžiaga 2007, 347-352
- [4] Semantics of Business Vocabulary and Business Rules (SBVR), v1.0. Object Management Group. [interaktyvus], [žiūrėta 2009-11-08]. Prieiga per internetą: <<http://www.omg.org/spec/SBVR/1.0/PDF>>
- [5] *Transact-SQL* Reference (Database Engine) [interaktyvus], [žiūrėta 2009-11-19]. Prieiga per internetą: <<http://msdn.microsoft.com/en-us/library/bb510741%28SQL.100%29.aspx>>
- [6] Laučius J., Vasilecas O., Lebedys E. Įvykis – sąlyga – veiksmas taisyklių realizavimas naudojant aktyvių duomenų bazių valdymo sistemų triggerius. ISSN 1392–0561. Informacijos mokslai. 2003 26. Mokslo darbai 26 tomas. Vilnius. 2003, 129-133.
- [7] *Microsoft SQL Server 2008*. [interaktyvus], [žiūrėta 2009-11-28]. Prieiga per internetą: <<http://www.microsoft.com/lietuva/sql/default.aspx> >
- [8] „*Microsoft SQL Server 2008*“ produkto apžvalga. [interaktyvus], [žiūrėta 2009-11-28]. Prieiga per internetą: <http://download.microsoft.com/download/9/e/c/9ecf8364-7357-4cf1-991a-a9dd841166b1/SQL2008_ProductOverview.doc>F
- [9] Sack J. *SQL Server 2008 Transact-SQL Recipes: A Problem – Solution Approach*. Apress, Portland, 2008, 375 p.
- [10] Nemuraitė, L., Skersys, T., Šukys, A., Šinkevičius, E., Ablonskis, L.. VETIS tool for editing and transforming SBVR business vocabularies and business rules into

- UML&OCL models. In A. Targamadze, R. Butleris, R. Butkiene (Eds.) Information Technologies' 2010 : proceedings of the 16th international conference on Information and Software Technologies, IT 2010, Kaunas, Lithuania, April 21-23, 2010, Kaunas University of Technology, Kaunas, Technologija, 2010, 377-384.
- [11] *SQL triggers*. iSeries Information Center, Version 5 Release 3. [interaktyvus], [žiūrėta 2010-10-16]. Prieiga per internetą: <<http://publib.boulder.ibm.com/infocenter/series/v5r3/index.jsp?topic=%2Fsqlp%2Frbafysqltrig.htm>>
- [12] Create procedure (*Transact-SQL*) [interaktyvus], [žiūrėta 2010-10-19]. Prieiga per internetą: <<http://msdn.microsoft.com/en-us/library/ms187926.aspx>>
- [13] Connecting to *Microsoft SQL Server 2008* from *Microsoft Visual Studio 2005* and *2008*. [interaktyvus], [žiūrėta 2010-10-03]. Prieiga per internetą: <<http://msdn.microsoft.com/en-us/library/cc440724%28v=vs.90%29.aspx>>
- [14] Create trigger (*Transact-SQL*) [interaktyvus], [žiūrėta 2010-10-19]. Prieiga per internetą: <<http://msdn.microsoft.com/en-us/library/ms189799.aspx>>
- [15] Butkienė R., Čeponienė L., Nemuraitė L.. Informacinių sistemų inžinerijos magistrų darbų metodiniai nurodymai. Mokomoji knyga. Kaunas, Technologija, 2009.
- [16] Basili, V. R., Selby, R.W., Hutchens, D. H. Experimentation in Software Engineering, *IEEE Trans. Software Eng.*, July 1986, 733-743.
- [17] Moschoyiannis S., Marinos A., Krause P.. Generating SQL Queries from *SBVR* Rules. M. Dean et al. (Eds.): Springer-Verlag Berlin Heidelberg, *RuleML 2010*, LNCS 6403, pp. 128-143, 2010.

Priedai

1priedas. Straipsnis „SBVR šablonai duomenų bazėse realizuojamoms veiklos taisyklėms specifikuoti“

SBVR ŠABLONAI DUOMENŲ BAZĖSE REALIZUOJAMOMS VEIKLOS TAISYKLĖMS SPECIFIKUOTI

Raminta Žeknytė, Lina Nemuraitė

Kauno technologijos universitetas, Informacijos sistemų katedra, Studentų 50-313a. LT-51368
Kaunas, Lietuva, tukas@nku.lt, lina.nemuraitte@ktu.lt

Santrauka. Straipsnyje aprašomi struktūrizuotos lietuvių kalbos šablonai, leidžiantys aprašyti tiriamos srities SBVR veiklos žodyną ir veiklos taisyklės taip, kad juos būtų galima realizuoti duomenų bazėje. Šiam tikslui pritaikyti SBVR struktūrizuotos anglų kalbos šablonai, papildant juos duomenų formatais, išvedimo taisyklėmis ir kitais elementais, kurių reikia norint sukurti DB schemą. Siekiant neprarasti informacijos, DB schemą reikia papildyti metaduomenų lentelėmis, kuriose išsaugomi SBVR faktų tipų ir vaidmenų vardai. Šablonų tinkamumas patikrintas realizuojant eksperimentinę skrydžių užsakymų sistemą.

Raktiniai žodžiai: SBVR, fakto tipas, veiklos taisyklė, duomenų bazė, SQL, trigeris, transformavimas

1. Įvadas

Šio darbo tikslas yra sudaryti SBVR (angl. *Semantics of Business Vocabulary and Business Rules*) veiklos žodyną ir taisyklių šablonus, tinkamus aprašyti IS reikalavimus struktūrizuota lietuvių kalba siekiant juos realizuoti duomenų bazėje (DB). Pagrindinė sprendžiama problema yra ta, kad sunku išsamiai ir vienareikšmiškai aprašyti reikalavimus, ypač jei siekiama automatiškai generuoti programinį kodą ar jo dalį; antra, saugant dalykinės srities informaciją duomenų bazėje, prarandama dalis semantinės informacijos, kuri reikalinga norint vykdyti semantines užklausas ar pateikti duomenų bazių turinį Pasaulio Semantiniame tinkle. SBVR veiklos žodyną ir taisyklių metamodelis leidžia išsamiai aprašyti dalykinės srities semantiką. Todėl šiame darbe analizuojama, kaip reikėtų formuluoti reikalavimus SBVR žodyną ir taisyklių pavidalu, kad aprašytų žinių pakaktų juos vienareikšmiškai realizuoti duomenų bazėje neprarandant semantikos. Šablonų rinkinio tinkamumui ir įgyvendinamumui patikrinti jais buvo aprašyta eksperimentinė skrydžių užsakymų sistema, kuri buvo realizuota *Microsoft SQL Server 2008* ir *Microsoft Visual Studio 2008* priemonėmis.

Antroje straipsnio dalyje analizuojami veiklos taisyklėms aprašyti taikomi šablonai ir pagrindžiamas SBVR šablonų pasirinkimas. Trečioje dalyje aprašyti lietuvių kalbai pritaikyti SBVR struktūrinių taisyklių šablonai ir jų transformavimo į DB schemą taisyklės; ketvirtoje – operacinių taisyklių šablonai ir jų vaizdavimas programiniiais komponentais. Penktoje dalyje apibendrinami naujai įvesti šablonai ir pateikiamos metaduomenų lentelės, leidžiančios išsaugoti daugiau semantikos. Pabaigoje pateikiamos išvados.

2. Veiklos taisyklių specifikuojamų šablonų analizė

Veiklos taisyklės apibrėžiamos dalykinės srities reikalavimų rinkimo etape, kai siekiama nustatyti visus veiksmus, kurie turi ar gali veikti konkretyje projekte kompiuterizuojamą objektą ar jo operaciją. Idealiu atveju veiklos taisyklių rinkinį turėtų būti galima papildyti bet kuriuo metu, kai to reikalauja veiklos pokyčiai [1].

GUIDE klasifikacijoje veiklos taisyklė suprantama kaip teiginys, kuris apibrėžia arba veikia tam tikrus veiklos aspektus, t.y. riboja veiklos struktūrą ar veikia veiklos eigseną [2]. SBVR aprašyme veiklos taisyklė suprantama taisyklė, kuri priklauso veiklai [3]. SBVR veiklos taisyklės remiasi faktų tipais, o faktų tipai – terminais. Fakto tipas – tai ryšys tarp dviejų ar daugiau daiktavardžių sąvokų (vaidmenų ir juos vaidinančių terminų). SBVR faktų tipai ir veiklos taisyklės glaudžiai siejasi su kalba. Natūrali kalba yra paprasčiausias veiklos taisyklių užrašymo būdas, tačiau jis tinkamas tik reikalavimų analizės pradžioje, nes yra neformalus [4].

Teisingas veiklos taisyklių užrašymas yra problema: sunku užfiksuoti veiklos taisyklės taip, kad jos būtų vienareikšmiškai suprantamos, teisingos ir lengvai realizuojamos programinėje įrangoje. Yra sukurtų šablonų, kurie leidžia struktūrizuoti veiklos taisyklių užrašymą, tačiau jie skirti anglų kalbai. *RuleSpeak* [5], *SBVR* [3] ir *Witt* [1] metoduose egzistuojančių šablonų palyginimas pateiktas 1 lentelėje.

1 lentelė. Metoduose egzistuojančių šablonų palyginimas

Metodai	Šablonai	Struktūrinių taisyklių	Išvedimo taisyklių	Operacinių taisyklių	Procesų	Kardinalumo ir ankialumo	Duomenų turinio	Duomenų formatai	Skaitėdavimų	Detalumo laipsnis	Tinkamumas apdoroti kompiuteriu
<i>RuleSpeak</i>		+	+		+					vidutinis	vidutinis
<i>SBVR</i>		+		+						vidutinis	didelis
<i>Witt</i>					+	+	+			didelis	vidutinis
Kuriamas metodas	+	+	+	+	+	+	+	+	+	didelis	didelis

RuleSpeak ir *SBVR* naudoja anglų kalba suformuotas veiklos taisyklių sakinių formas, kurių struktūra yra šiek tiek skirtinga, tačiau abiem atvejais kiekvienos taisyklės sakinio pradžioje aiškiai išskiriamas subjektas, kuris suteikia sakiniui aiškumo ir pastovumo. *SBVR* šablonai yra patogesni realizuoti kompiuteriu, kadangi jie prasideda raktiniais žodžiais, leidžiančiais iš karto nustatyti taisyklės tipą. *RuleSpeak* šablonuose raktiniai žodžiai įpinti sakinio viduryje. Ir vieni, ir kiti šablonai neturi elementų, kurių reikia specifikuojant detalius reikšmių ribojimus, formatus ir pan. *Witt* rekomendacijos pasižymi didesniu detalumu, tačiau kompiuterizuoti tinkamesnė *SBVR* sakinių forma. Atlikus lyginamąją analizę, buvo nutarta formuluoti lietuviškus šablonus remiantis *SBVR* stiliumi ir papildyti juos *Witt* rekomendacijomis, kad šablonai būtų lengvai kompiuterizuojami ir reikiamo detalumo lygio.

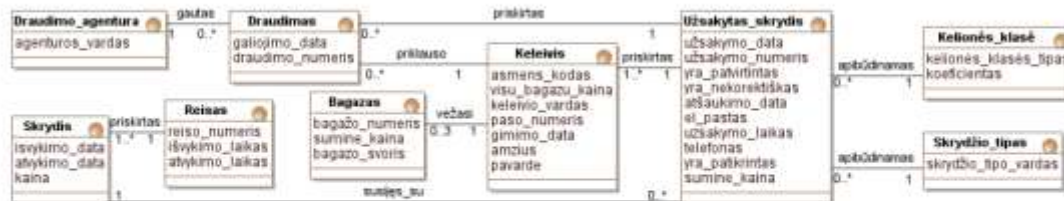
3. SBVR šablonai ir jų transformavimas į DB schema

Pagal *SBVR* aprašymą faktų tipai ir veiklos taisyklės apibrėžiami keturiais teksto stiliais [4]: terminas, raktažodis, literalus simbolis/vardas ir veikmažodis (2 lentelė).

2 lentelė. Veiklos taisyklių sudarymas iš faktų tipų

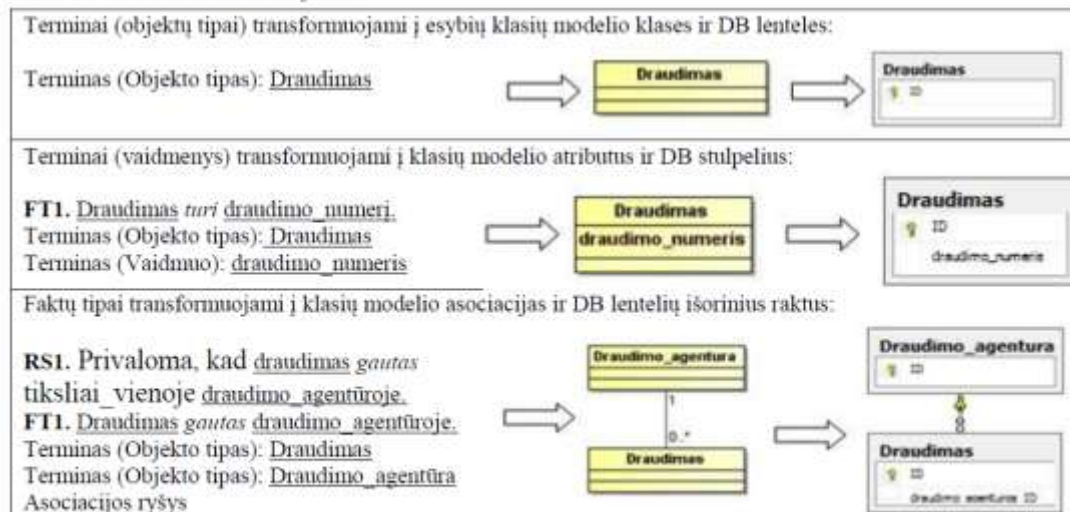
Veiklos taisyklė (RS)	Fakto tipas (FT)
RS1. Privaloma, kad <u>reisas turi bent vieną skrydį</u> .	FT1. <u>Reisas turi skrydį</u> .
RS2. Privaloma, kad <u>užsakytas skrydis turi vieną iš aibės arba daugiau: telefonas, el. paštas</u> .	FT1. <u>Užsakytas skrydis turi telefoną</u> . FT2. <u>Užsakytas skrydis turi el. pašta</u> .
RS3. Privaloma, kad <u>skrydžio tipo vardas yra vienas iš aibės: vienpusis skrydis, dvipusis skrydis</u> .	FT1. <u>Skrydžio tipas turi skrydžio tipo vardą</u> . FT2. <u>Vienpusis skrydis yra skrydžio tipo vardas</u> . FT3. <u>Dvipusis skrydis yra skrydžio tipo vardas</u> .

Norint iš veiklos žodyno ir taisyklių gauti DB schema, galima sugeneruoti esybių klasių modelį [4] ir panaudoti esamus DB schemų generavimo įrankius, arba generuoti DB schema iš karto. Eksperimentinės skrydžių užsakymų sistemos esybių klasių modelio fragmentas pateikiamas 1 paveiksle.



1 pav. Faktų tipų ir struktūrinių taisyklių transformavimas į esybių klasių modelį

Terminų (objektų tipų) transformavimas į DB lenteles skiriasi nuo transformavimo į esybių klasių modelį tuo, kad lentelėms reikia sukurti pirminius raktus (ID stulpelius), be to, vaidmenims reikia aprašyti terminus, vaizduojančius duomenų tipus (pvz., tekstas, sveikas skaičius ir t.t.) bei formatus (pvz., teksto simbolių skaičių). Faktų tipams reikia sukurti išorinius raktus, todėl kartu reikia nagrinėti ir taisykles, išreiškiančias kardinalumo ribojimus.



Kiekvienam veiklos taisyklių tipui buvo sudaryti šablonai, pavyzdžiui, taisyklė „Privaloma, kad reisas turi tiksliai vieną reiso numerį“ užrašoma pagal šablona:

```
{<raktinis zodis {privaloma, kad}> <terminas-tipas> <veiksmažodis {turi}> <raktinis zodis {tiksliai_viena}> <terminas-savybė>}
```

Struktūrinės taisyklės, reiškiančios kardinalumo ribojimus, transformuojamos į esybių klasių modelio savybių (atributų ir ryšių) kardinalumo ribas. DB schemoje šie ribojimai realizuojami įvairiais elementais:

- DB schemoje 1:0..* kardinalumai vaizduojami privalomu išoriniu raktu; 1:0..1 kardinalumai vaizduojami trigeriais; 1:1..* kardinalumai vaizduojami trigeriais ir procedūromis;
- Taisyklės, reiškiančios privalomus atributus, transformuojamos į „not null“ ribojimus;
- Taisyklės, ribojančios atributų reikšmių aibes, realizuojamos „check“ ribojimais arba trigeriais;
- Taisyklės, reiškiančios išskirtinumo, ekvivalentiškumo ir kitus ribojimus dviem ar daugiau stulpelių, priklausančių skirtingoms lentelėms, realizuojamos trigeriais;
- Išvedimo (skaičiavimo) taisyklės, apimančios įvairias duomenų operacijas (aritmetines, operacijas su simbolių eilutėmis, pvz., datos ir laiko sujungimą), realizuojamos procedūromis arba trigeriais.

Pavyzdžiui, skaičiavimo taisyklė „Privaloma, kad užsakymo skrydžio suminė kaina skaičiuojama taip: skrydžio kaina padauginta iš kelionės klasės koeficiento plus keleivio visu bagažu kaina“ realizuojama procedūros kodu:

```
DECLARE @koeficientas float; SET @koeficientas = (select koeficientas from keliones_klase
where ID = @keliones_klases_ID) DECLARE @skrydzioKaina float; SET @skrydzioKaina =
(select kaina from skrydis where ID = @skrydzio_ID) DECLARE @bagoSumineKaina float; SET
@bagoSumineKaina = 0 DECLARE @sumine_kaina float; set @sumine_kaina = @skrydzioKaina *
@koeficientas + @bagoSumineKaina
```

Atributo reikšmių ribojimas „Privaloma, kad keleivio asmens kodo tekstas turi 11 simbolių“ realizuojamas *Check Constraint Expression*: `(len([asmens_kodas])=(11))`, o „Privaloma, kad el pašto tekstas turi @ simbolių“ – trigeriai:

```
CREATE TRIGGER [dbo].[taisykle24] ON [dbo].[Uzsakytas_skyrdis]
ALTER INSERT AS IF EXISTS (SELECT el_pastas FROM inserted) BEGIN DECLARE @eta as int = 0;
SET @eta = (select charindex('@', (SELECT el_pastas FROM inserted))) IF (@eta
= '') BEGIN RAISERROR ('Veiklos taisyklė: Privaloma, kad el.paštas turi "@" simbolių.',
16, 1) END END
```

4. Operacinių veiklos taisyklių šablonai

Operacinės taisyklės vaizduoja operacijas – veiksmus, kuriuos vykdo vartotojas arba sistema. Jos realizuojamos taikomiosiose programose, kurios manipuliuoja DB saugomais duomenimis. Skiriamos dvių tipų – leidimo ir įpareigojimo taisyklės. Leidimo taisyklių šablonas leidžia aprašyti funkcijas, kurias leidžiama vykdyti tam tikriems vartotojams. Pavyzdžiui, „Leidžiama, kad skrydžio užsakovas sukuria užsakyta skrydį jei skrydžio užsakovas yra prisijungęs vartotojas“.

```
{<raktinis zodis {leidžiama, kad}> <terminas-tipas> <veiksmažodis {prisijungia_prie | sukuria | patvirtina}> <terminas-tipas> , jei <terminas-tipas> <veiksmažodis {yra | sukuria}> <terminas-tipas>}
```

Įpareigojimo taisyklių šablonas leidžia aprašyti funkcijas, kurias įsipareigoja vykdyti sistema ar tam tikri vartotojai, esant išpildytoms tam tikroms sąlygoms Pavyzdžiui, „Įpareigojama, kad skrydžių užsakymų sistema tikrina užsakyta skrydį, jei skrydžio užsakovas patvirtina užsakyta skrydį“.

```
{<raktinis zodis {įpareigojama, kad}> <terminas-tipas> <veiksmažodis {įterpia | tikrina | patvirtina}> <terminas-tipas> , jei <terminas-tipas> <veiksmažodis {patvirtina | yra | sukuria}> <terminas-tipas>}
```

5. DB schemos semantikos išplėtimas

Norint išsaugoti SBVR veiklos žodyno ir taisyklių aprašomą dalykinės srities semantiką, sukurtą DB schemą reikia papildyti. DB lentelių ir stulpelių vardai paprastai nėra šios kalbos. Nors šios galimybės jau egzistuoja naujose DBVS versijose, lietuviškos raidės gali sukelti problemų apdorojant DB duomenis įvairiomis taikomiosiomis programomis. Antra, užrašant faktų tipus ir taisykles naudojamos įvairios to paties žodžio formos, o DB lentelių ir stulpelių vardai kuriami tik iš vienos žodžio formos, kuri apibrėžiama kaip privilegijuotoji (angl. *preferred*). Norint neprarasti šios informacijos, DB schemą reikia papildyti lentelėmis „Veiklos žodyno elementas“ ir „RDB elementas“, kurios saugotų žodyno konceptų ir DB elementų vardų atitikmenis. Šiam tikslui skirtas esybių klasių modelis pavaizduotas 2 paveiksle.



2 pav. Esys, leidžiančios išsaugoti žodyno ir RDB elementų vardų atitikmenis bei faktų tipų ir vaidmenų vardus

Be to, DB schemoje neišsaugomi faktų tipų ir vaidmenų vardai, todėl norint juos atkurti iš duomenų bazės lentelių, reikia sukurti lenteles „Objekto tipo pavadinimas“, „Rysio tipo pavadinimas“ ir „Vaidmens pavadinimas“. Kai kurie terminai gali būti ir vaidmenų, ir objektų tipų vardai, jei fakto tipas naudoja objektų tipų vardus kaip vaidmenų vardus. Šiuo atveju esybių klasių modelyje ryšio galas neturi pavadinimo. Tas pats veiksmazodis gali būti naudojamas įvairiuose faktų tipuose. Kad būtų galima teisingai atstaryti vardus, metaduomenų lentelėse reikia išsaugoti visus jų taikymo variantus.

ID	RDB_elem_pavad_ID	veiklos_zodyno_elem_pavad.
1	1	užsakytas_skydis
2	2	užsakytam_skydžiui
3	3	užsakytu_skydžiu
4	4	keleivis
5	5	keleiviniui
6	6	keleivio

3 pav. Lentelių „Veiklos_zodyno_elementas“ (a) ir „RDB_elementas“ (b) duomenų pavyzdžiai

Pavyzdžiui, turime tris faktų tipus, kur objektų tipų vardai sutampa su vaidmenų vardais ir dviejuose faktų tipuose naudojamas tas pats veiksmazodis: „miestas turi miesto vardą“, „mieste veikia oro uostas“, „oro uostas turi oro uosto vardą“.

Šių faktų tipų įrašai metaduomenų lentelėse pateikiami 4 paveiksle.

ID	objekto_tipo_pavad.
1	miestas
2	oro_uostas

ID	rysio_pavad.
1	turi
2	veikia

ID	objekto_tipo_pavad_ID	rysio_pavad_ID	vaidmens_pavad.
1	1	1	miesto vardas
2	1	2	oro uostas
3	2	1	oro uosto vardas

4 pav. Lentelių „Objekto_tipo_pavadinimas“, „Rysio_pavadinimas“ ir „Vaidmens_pavadinimas“ duomenų pavyzdžiai

5. Išvados

Šiame darbe SBVR metamodelio pagrindu buvo sudarytas struktūrizuotos lietuvių kalbos šablonų rinkinys, leidžiantis vienareikšmiškai aprašyti reikalavimus ir veiklos taisykles, kurias reikia specifiukuoti norint sukurti DB schemą ir ja manipuliuojančią programinę įrangą. Šių šablonų taikymas aprašant ir realizuojant eksperimentinę skydžių užsakymų sistemą parodė, kad jie leidžia spręsti problemas, kylančias aprašant DB reikalavimus. Dalis šių problemų susijusi su įvairių formatų, ribojimų, skaičiavimų aprašais, kurie nėra detalizuoti SBVR metamodelyje, todėl jiems buvo sudaryti papildomi šablonai. Antroji problema – išsaugoti atitikmenis tarp dalykinėje srityje naudojamų lietuviškų vardų bei jų kombinacijų ir DB lentelių bei stulpelių pavadinimų, kurie leidžia išsaugoti tik vieną žodžio formą ir nenaudoja lietuviškų simbolių. Sudarytus šablonus ir metalenteles tikslinga taikyti praktikoje siekiant formaliai ir įvairiems vartotojams suprantamai aprašyti reikalavimus duomenų bazėms kurti, o ateityje – automatizuoti DB schemų generavimą iš SBVR veiklos žodynų.

Literatūros sąrašas

- [1] **Graham, W.** A Practical Method of Developing Natural Language Rule Statements (Part 1–13). *Business Rules Journal*, 10(2)–(14), 2010.
- [2] **Defining Business Rules ~What Are They Really?** The Business Rules Group, formerly, known as the GUIDE Business Rules Project. Final Report, revision 1.3, July, 2000, 1–77. Prieiga per internetą: [http://www.docstoc.com/docs/11646109/Defining-Business-Rules---What-Are-They-Really-\(3rd-edition\)](http://www.docstoc.com/docs/11646109/Defining-Business-Rules---What-Are-They-Really-(3rd-edition)).
- [3] **OMG, 2008.** *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0.* OMG Document Number: formal/2008-01-02.
- [4] **Nemuraitė, L., Skersys, T., Šukys, A., Šinkevičius, E., Ablonkskis, L.** VETIS tool for editing and transforming SBVR business vocabularies and business rules into UML&OCL models. *In Information Technologies' 2010: proceedings of the 16th international conference on Information and Software Technologies, IT 2010, Kaunas, Lithuania, April 21-23, 2010, Kaunas University of Technology, Kaunas, Technologija, 2010, 377–384.*
- [5] **Ross, R.G.** *Business Rule Speak Guidelines. Version 2.2.* Business Rule Solutions, LLC, 2001–2009.

SBVR TEMPLATES FOR SPECIFYING BUSINESS RULES IMPLEMENTABLE IN DATABASE

Abstract. The paper presents structured Lithuanian language templates for specifying SBVR business vocabularies and business rules with intention to implement them in a database. For this purpose SBVR Structured English templates were supplemented with detail templates for data formats, derivation rules and other elements. Metatables were proposed for keeping names for fact types and roles. The suitability of the templates were verified by describing and implementing an experimental flight ordering system.

2 priedas. Faktų tipai

Fakto tipas – tai ryšys tarp dviejų ar daugiau daiktavardžių sąvokų.

- FT1. Užsakytas skrydis turi užsakymo datą.
- FT2. Užsakytas skrydis turi užsakymo numerį.
- FT3. Užsakytas skrydis turi suminę kainą.
- FT4. Užsakytas skrydis turi atšaukimo datą.
- FT5. Užsakytas skrydis yra patikrintas.
- FT6. Užsakytas skrydis yra patvirtintas.
- FT7. Užsakytas skrydis yra nekorektiškas.
- FT8. Užsakytas skrydis turi el paštą.
- FT9. Užsakytas skrydis turi telefoną.
- FT10. Užsakytas skrydis turi užsakymo laiką.
- FT11. Užsakytam skrydžiui priskirta valiuta.
- FT12. Užsakytam skrydžiui priskirtas keleivis.
- FT13. Keleivis turi keleivio vardą.
- FT14. Keleivis turi pavardę.
- FT15. Keleivis turi paso numerį.
- FT16. Keleivis turi gimimo datą.
- FT17. Keleivis turi asmens kodą.
- FT18. Keleivis turi amžių.
- FT19. Keleiviui priskirtas bilietas.
- FT20. Bilietas turi numerį.
- FT21. Bilietas turi vietą.
- FT22. Bilietui priskirta valiuta.
- FT23. Bilietui priskirtas registruotas skrydis.
- FT24. Registruotas skrydis vykdomas oro uoste.
- FT25. Registruotas skrydis susijęs su apmokėtu skrydžiu.
- FT26. Registruotas skrydis priskirtas grįžtamam skrydžiui.
- FT27. Registruotas skrydis turi registravimo datą.
- FT28. Registruotas skrydis turi registravimo laiką.
- FT29. Registruotas skrydis apibūdinamas skrydžio tipu.
- FT30. Keleivis vežasi bagažą.
- FT31. Bagažas turi bagažo numerį.
- FT32. Bagažas turi suminę kainą.
- FT33. Bagažas turi bagažo svorį.
- FT34. Bagažas apibūdinamas bagažo tipu.
- FT35. Bagažui priskirta valiuta.
- FT36. Valiuta turi valiutos vardą.
- FT37. Matavimo vienetas turi matavimo vieneto vardą.
- FT38. Matavimo vienetas turi vardo sutrumpinimą.
- FT39. Bagažo tipas matuojamas matavimo vienetu.
- FT40. Bagažo tipas turi matavimo vieneta svorio.
- FT41. Bagažo tipas turi plotį.
- FT42. Bagažo tipas turi aukštį.
- FT43. Bagažo tipas turi storį.
- FT44. Bagažo tipas turi svorio vieneto kainą.
- FT45. Bagažo tipas turi maksimalų svorį.
- FT46. Bagažo tipas turi papildomą kainą.

- FT47. Užsakytas skrydis *užsakytas* skrydžio užsakovo.
- FT48. Skrydžio užsakovas *yra* keleivis.
- FT49. Keleivis *apibūdinamas* keleivio tipu.
- FT50. Keleivio tipas *turi* keleivio tipo vardą.
- FT51. Keleivio tipas *turi* amžių nuo.
- FT52. Keleivio tipas *turi* amžių iki.
- FT53. Suaugęs keleivis *yra* keleivio tipo vardas.
- FT54. Jaunuolis *yra* keleivio tipo vardas.
- FT55. Kūdikis *yra* keleivio tipo vardas.
- FT56. Keleivis *turi* amžių.
- FT57. Užsakytas skrydis *apibūdinamas* skrydžio tipu.
- FT58. Skrydžio tipas *turi* skrydžio tipo vardą.
- FT59. Vienpusis skrydis *yra* skrydžio tipo vardas.
- FT60. Dvipusis skrydis *yra* skrydžio tipo vardas.
- FT61. Užsakytas skrydis *susijęs su* skrydžiu.
- FT62. Skrydis *turi* atvykimo datą.
- FT63. Skrydis *turi* išvykimo datą.
- FT64. Skrydis *turi* kainą.
- FT65. Grįžtamas skrydis *susijęs su* užsakytu skrydžiu.
- FT66. Grįžtamas skrydis *susijęs su* skrydžiu.
- FT67. Grįžtamas skrydis *turi* grįžimo datą.
- FT68. Grįžtamas skrydis *turi* grįžimo laiką.
- FT69. Užsakytas skrydis *apibūdinamas* kelionės klase.
- FT70. Kelionės klasė *turi* kelionės klasės tipą.
- FT71. Kelionės klasė *turi* koeficientą.
- FT72. Pirma klasė *yra* kelionės klasės tipas.
- FT73. Verslo klasė *yra* kelionės klasės tipas.
- FT74. Ekonominė klasė *yra* kelionės klasės tipas.
- FT75. Apmokėtas skrydis *susijęs su* užsakytu skrydžiu.
- FT76. Apmokėtas skrydis *turi* banko kodą.
- FT77. Apmokėtas skrydis *turi* mokėjimo datą.
- FT78. Apmokėtas skrydis *turi* pervedimo datą.
- FT79. Apmokėtas skrydis *turi* pervestą sumą.
- FT80. Grįžtamas skrydis *yra* užsakytas skrydis.
- FT81. Apmokėtam skrydžiui *priskirta* valiuta.
- FT82. Apmokėtam skrydžiui *priskirta* kreditinė kortelė.
- FT83. Apmokėtam skrydžiui *priskirtas* elektroninis pervedimas.
- FT84. Kreditinė kortelė *turi* kortelės tipą.
- FT85. Kreditinė kortelė *turi* kortelės numerį.
- FT86. Kreditinė kortelė *turi* kortelės kodą.
- FT87. Kreditinė kortelė *turi* kortelės galiojimo datą.
- FT88. Draudimas *priklauso* keleiviui.
- FT89. Draudimas *turi* galiojimo datą.
- FT90. Draudimas *turi* draudimo numerį.
- FT91. Draudimas *gautas* draudimo agentūroje.
- FT92. Draudimo agentūrą *turi* agentūros vardą.
- FT93. Mieste *veikia* oro uostas.
- FT94. Miestas *turi* miesto vardą.
- FT95. Oro uostas *turi* oro uosto vardą.
- FT96. Oro uostui *priskirtas* išvykstantis maršrutas.

- FT97. Oro uostui priskirtas atvykstantis maršrutas.
 FT98. Maršrutas turi maršruto vardą.
 FT99. Maršrutas pradedamas išvykimo oro uoste.
 FT100. Maršrutas baigiamas atvykimo oro uoste.
 FT101. Maršrutas vykdomas reisu.
 FT102. Reisui priskirtas skrydis.
 FT103. Reisas turi reiso numerį.
 FT104. Reisas turi išvykimo laiką.
 FT105. Reisas turi atvykimo laiką.
 FT106. Apmokėtas skrydis apmokėtas skrydžio užsakovo.
 FT107. Keleivis turi visų bagažų kainą.
 FT108. Elektroninis pervedimas turi sąskaitos numerį.
 FT109. Užsakytam skrydžiui priskirtas draudimas.
 FT110. Apmokėtas skrydis yra užsakytas skrydis.

Sinoniminės formos

- Fakto tipo FT17 sinoniminė forma: Asmens kodas priklauso keleiviui.
 Fakto tipo FT20 sinoniminė forma: Numeris priskirtas bilietui.
 Fakto tipo FT103 sinoniminė forma: Reiso numeris priskirtas reisui.
 Fakto tipo FT108 sinoniminė forma: Sąskaitos numeris priskirtas elektroniniui pervedimui.

Kiti faktų tipai (KFT)

- KFT1. Išvykimo data yra data.
 KFT2. Išvykimo laikas yra laikas.
 KFT3. Suminė kaina yra skaičius.
 KFT4. Keleivio vardas yra tekstas.
 KFT5. Telefonas yra tekstas.
 KFT6. Paso numeris yra tekstas.
 KFT7. Koeficientas yra skaičius.
 KFT8. Pervesta suma yra skaičius.
 KFT9. Bagažo svoris yra realus skaičius.
 KFT10. Aukštis yra realus skaičius.
 KFT11. Storis yra realus skaičius.
 KFT12. Plotis yra realus skaičius.
 KFT13. Vieta yra realus skaičius.
 KFT14. Amžius nuo yra realus skaičius.
 KFT15. Yra patvitintas yra boolean.
 KFT16. Reisas turi faktinį išvykimo laiką.
 KFT17. Reisas turi faktinį atvykimo laiką.
 KFT18. Užsakytas skrydis turi faktinį užsakymo laiką.
 KFT19. Grijtamas skrydis yra užsakytas skrydis.
 KFT20. Apmokėtas skrydis yra užsakytas skrydis.
 KFT21. Stuka yra lėktuvas.
 KFT22. Vilnius yra miestas.
 KFT23. ePolisas yra draudimo agentūra.
 KFT24. Jonas yra keleivis.
 KFT25. Maršrutas susideda iš reisu.
 KFT26. Reisas susideda iš skrydžių.

- KFT27. Skrydžio tipo vardas yra unikalus.
 KFT28. Užsakymo numeris yra unikali seka.
 KFT29. Asmens kodas yra unikali seka.
 KFT30. Tekstas turi @ simbolį.
 KFT31. Tekstas turi 11 simbolių.

3 priedas. Veiklos taisyklių sakiniai (RS)

Struktūrinės veiklos taisyklės

- RS1. Privaloma, kad reisas turi tiksliai vieną reiso numerį.
 RS2. Privaloma, kad reisas turi tiksliai vieną išvykimo laiką.
 RS3. Privaloma, kad reisas turi tiksliai vieną atvykimo laiką.
 RS4. Privaloma, kad draudimas turi tiksliai vieną draudimo numerį.
 RS5. Privaloma, kad draudimas turi tiksliai vieną galiojimo datą.
 RS6. Privaloma, kad apmokėtas skrydis turi tiksliai vieną banko kodą.
 RS7. Privaloma, kad apmokėtas skrydis turi tiksliai vieną mokėjimo datą.
 RS8. Privaloma, kad apmokėtas skrydis turi tiksliai vieną pervedimo datą.
 RS9. Privaloma, kad apmokėtas skrydis turi tiksliai vieną pervestą sumą.
 RS10. Privaloma, kad užsakytas skrydis turi tiksliai vieną užsakymo numerį.
 RS11. Privaloma, kad užsakytas skrydis turi tiksliai vieną užsakymo datą.
 RS12. Privaloma, kad užsakytas skrydis turi tiksliai vieną užsakymo laiką.
 RS13. Privaloma, kad užsakytas skrydis turi tiksliai vieną atšaukimo datą.
 RS14. Privaloma, kad skrydžio tipas turi tiksliai vieną skrydžio tipo vardą.
 RS15. Privaloma, kad draudimo agentūra turi tiksliai vieną agentūros vardą.
 RS16. Privaloma, kad kelionės klasė turi tiksliai vieną kelionės klasės tipą.
 RS17. Privaloma, kad kelionės klasė turi tiksliai vieną koeficientą.
 RS18. Privaloma, kad grįžtamas skrydis turi tiksliai vieną grįžimo laiką.
 RS19. Privaloma, kad grįžtamas skrydis turi tiksliai vieną grįžimo datą.
 RS20. Privaloma, kad kreditinė kortelė turi tiksliai vieną kortelės tipą.
 RS21. Privaloma, kad kreditinė kortelė turi tiksliai vieną kortelės numerį.
 RS22. Privaloma, kad kreditinė kortelė turi tiksliai vieną kortelės kodą.
 RS23. Privaloma, kad kreditinė kortelė turi tiksliai vieną kortelės galiojimo datą.
 RS24. Privaloma, kad bilietas turi tiksliai vieną vietą.
 RS25. Privaloma, kad bilietas turi tiksliai vieną numerį.
 RS26. Privaloma, kad registruotas skrydis turi tiksliai vieną registravimo datą.
 RS27. Privaloma, kad registruotas skrydis turi tiksliai vieną registravimo laiką.
 RS28. Privaloma, kad bagažas turi tiksliai vieną bagažo numerį.
 RS29. Privaloma, kad bagažas turi tiksliai vieną bagažo svorį.
 RS30. Privaloma, kad bagažo tipas turi tiksliai vieną papildomą kainą.
 RS31. Privaloma, kad bagažo tipas turi tiksliai vieną plotį.
 RS32. Privaloma, kad bagažo tipas turi tiksliai vieną aukštį.
 RS33. Privaloma, kad bagažo tipas turi tiksliai vieną storį.
 RS34. Privaloma, kad bagažo tipas turi tiksliai vieną maksimalų svorį.
 RS35. Privaloma, kad bagažo tipas turi tiksliai vieną svorio vieneto kainą.
 RS36. Privaloma, kad keleivis turi tiksliai vieną visų bagažų kainą.
 RS37. Privaloma, kad keleivis turi tiksliai vieną asmens kodą.
 RS38. Privaloma, kad keleivis turi tiksliai vieną paso numerį.
 RS39. Privaloma, kad keleivis turi tiksliai vieną gimimo datą.
 RS40. Privaloma, kad keleivis turi tiksliai vieną keleivio vardą.

- RS41. Privaloma, kad keleivis turi tiksliai_vieną pavarde.
- RS42. Privaloma, kad matavimo vienetas turi tiksliai_vieną matavimo vieneto varda.
- RS43. Privaloma, kad matavimo vienetas turi tiksliai_vieną vardo sutrumpinima.
- RS44. Privaloma, kad skrydis turi tiksliai_vieną išvykimo data.
- RS45. Privaloma, kad skrydis turi tiksliai_vieną atvykimo data.
- RS46. Privaloma, kad skrydis turi tiksliai_vieną kaina.
- RS47. Privaloma, kad miestas turi tiksliai_vieną miesto varda.
- RS48. Privaloma, kad maršrutas turi tiksliai_vieną maršruto varda.
- RS49. Privaloma, kad valiuta turi tiksliai_vieną valiutos varda.
- RS50. Privaloma, kad keleivio tipas turi tiksliai_vieną keleivio tipo varda.
- RS51. Privaloma, kad keleivio tipas turi tiksliai_vieną amžių nuo.
- RS52. Privaloma, kad keleivio tipas turi tiksliai_vieną amžių iki.
- RS53. Privaloma, kad bagažo tipas turi tiksliai_vieną matavimo vieneta svorio.
- RS54. Privaloma, kad elektroninis pervedimas turi tiksliai_vieną sąskaitos numerį.
- RS55. Privaloma, kad oro uostas turi tiksliai_vieną oro uosto varda.
- RS56. Privaloma, kad užsakytas skrydis apibūdinamas tiksliai_viena kelionės klase.
- RS57. Privaloma, kad užsakytas skrydis susijęs su tiksliai_vienu skrydžiu.
- RS58. Privaloma, kad užsakytas skrydis apibūdinamas tiksliai_vienu skrydžio tipu.
- RS59. Privaloma, kad užsakytas skrydis užsakytas tiksliai_vieno skrydžio užsakovo.
- RS60. Privaloma, kad užsakytam skrydžiui priskirta tiksliai_viena valiuta.
- RS61. Privaloma, kad užsakytam skrydžiui priskirtas tiksliai_vienas draudimas.
- RS62. Privaloma, kad apmokėtam skrydžiui priskirta tiksliai_viena valiuta.
- RS63. Privaloma, kad apmokėtas skrydis apmokėtas tiksliai_vieno skrydžio užsakovo.
- RS64. Privaloma, kad bilietai priskirta tiksliai_viena valiuta.
- RS65. Privaloma, kad keleiviui priskirtas tiksliai_vienas bilietas.
- RS66. Privaloma, kad registruotas skrydis vykdomas tiksliai_viename oro uoste.
- RS67. Privaloma, kad registruotas skrydis susijęs su tiksliai_vienu apmokėtu skrydžiu.
- RS68. Privaloma, kad registruotas skrydis priskirtas tiksliai_vienam grįžtamam skrydžiui.
- RS69. Privaloma, kad registruotas skrydis apibūdinamas tiksliai_vienu skrydžio tipu.
- RS70. Privaloma, kad grįžtamas skrydis susijęs su tiksliai_vienu skrydžiu.
- RS71. Privaloma, kad bagažui priskirta tiksliai_viena valiuta.
- RS72. Privaloma, kad bagažas apibūdinamas tiksliai_vienu bagažo tipu.
- RS73. Privaloma, kad bagažo tipas matuojamas tiksliai_vienu matavimo vienetu.
- RS74. Privaloma, kad skrydžio užsakovas yra tiksliai_vienas keleivis.
- RS75. Privaloma, kad mieste veikia tiksliai_vienas oro uostas.
- RS76. Privaloma, kad išvykstantis maršrutas išvyksta iš tiksliai_vieno oro uosto.
- RS77. Privaloma, kad atvykstantis maršrutas atvyksta į tiksliai_vieną oro uosta.
- RS78. Privaloma, kad keleivis apibūdinamas tiksliai_vienu keleivio tipu.
- RS79. Privaloma, kad maršrutas vykdomas tiksliai_vienu reisu.
- RS80. Privaloma, kad draudimas gautas tiksliai_vienoje draudimo agentūroje.
- RS81. Privaloma, kad draudimas priklauso tiksliai_vienam keleiviui.
- RS82. Privaloma, kad bilietai priskirtas ne daugiau kaip_vienas registruotas skrydis.
- RS83. Privaloma, kad keleivis vežasi ne daugiau kaip_3 bagažus.
- RS84. Privaloma, kad apmokėtam skrydžiui priskirta ne daugiau kaip_viena kreditinė kortelė.
- RS85. Privaloma, kad apmokėtam skrydžiui priskirta ne daugiau kaip_vienas elektroninis pervedimas.
- RS86. Privaloma, kad grįžtamas skrydis susijęs su ne daugiau kaip_vienu užsakytu skrydžiu.

- RS87. Privaloma, kad apmokėtas skrydis susijęs su ne daugiau kaip vienu užsakytu skrydžiu.
- RS88. Privaloma, kad reisui priskirtas bent vienas skrydis.
- RS89. Privaloma, kad užsakytam skrydžiui priskirtas bent vienas keleivis.
- RS90. Privaloma, kad užsakytas skrydis turi vieną arba daugiau iš aibės: telefonas, el paštas.
- RS91. Privaloma, kad apmokėtas skrydis turi vieną iš aibės: elektroninis pervedimas, kreditinė kortelė.
- RS92. Privaloma, kad kelionės klasės kelionės klasės tipas yra vienas iš aibės: pirma klasė, verslo klasė, ekonominė klasė.
- RS93. Privaloma, kad skrydžio tipo skrydžio tipo vardas yra vienas iš aibės: vienpusis skrydis, dvipusis skrydis.
- RS94. Privaloma, kad keleivio tipo keleivio tipo vardas yra vienas iš aibės: suaugęs keleivis, jaunuolis, kūdikis.
- RS95. Privaloma, kad keleivio amžius skaičiuojamas taip: keleivio užsakyto skrydžio užsakymo data minus keleivio gimimo data.
- RS96. Privaloma, kad bagazo suminė kaina skaičiuojama taip: bagazo bagazo svoris padaugintas iš bagazo tipo svorio vieneto kainos, jei bagazo maksimalus svoris yra ne didesnis nei 20 kg.
- RS97. Privaloma, kad bagazo suminė kaina skaičiuojama taip: bagazo bagazo svoris minus bagazo tipo maksimalus svoris padaugintas iš bagazo tipo papildomos kainos, jei bagazo bagazo svoris yra ne didesnis už bagazo tipo maksimalų svorį.
- RS98. Privaloma, kad keleivio visų bagažų kaina skaičiuojama taip: suma, kuri susideda iš keleivio bagažo suminių kainų.
- RS99. Privaloma, kad užsakyto skrydžio suminė kaina skaičiuojama taip: skrydžio kaina padauginta iš kelionės klasės koeficiento plus keleivio visų bagažų kaina.
- RS100. Privaloma, kad keleivis yra kūdikis, jei keleivio amžius yra mažesnis nei 2 metai.
- RS101. Privaloma, kad keleivis yra suaugęs keleivis, jei keleivio amžius yra ne mažesnis nei 18 metų.
- RS102. Privaloma, kad keleivis yra jaunuolis, jei keleivio amžius yra ne mažesnis nei 2 metai ir yra mažesnis nei 18 metų.
- RS103. Privaloma, kad vienpusis skrydis turi tiksliai vieną užsąkytą skrydį.
- RS104. Privaloma, kad dvipusis skrydis turi du užsąkytus skrydžius.
- RS105. Privaloma, kad reiso faktinis išvykimo laikas skaičiuojamas taip: skrydžio išvykimo data sujungiama su reiso išvykimo laiku.
- RS106. Privaloma, kad reiso faktinis atvykimo laikas skaičiuojamas taip: skrydžio atvykimo data sujungiama su reiso atvykimo laiku.
- RS107. Privaloma, kad skrydžio išvykimo data yra ne vėlesnė nei skrydžio atvykimo data.
- RS108. Privaloma, kad skrydžio atvykimo data yra ne ankstesnė nei skrydžio išvykimo data.
- RS109. Privaloma, kad reiso išvykimo laikas yra ne vėlesnis nei reiso atvykimo laikas.
- RS110. Privaloma, kad reiso atvykimo laikas yra ne ankstesnė nei reiso išvykimo laikas.
- RS111. Privaloma, kad užsakyto skrydžio skrydžio išvykimo data yra ne ankstesnė nei užsakyto skrydžio užsakymo data.
- RS112. Privaloma, kad užsakyto skrydžio skrydžio išvykimo data yra ankstesnė nei užsakyto skrydžio grįžtamo skrydžio skrydžio grįžimo data.
- RS113. Privaloma, kad užsakyto skrydžio skrydžio reiso faktinis išvykimo laikas yra ne ankstesnis nei faktinis užsakyto skrydžio užsakymo laikas.

Operacinės veiklos taisyklės

- RS114. Leidžiama, kad skrydžio užsakovas *sukuria* užsakytą skrydį jei skrydžio užsakovas yra prisijungęs vartotojas.
- RS115. Leidžiama, kad skrydžio užsakovas *sukuria* apmokėtą skrydį jei skrydžio užsakovas yra prisijungęs vartotojas.
- RS116. Leidžiama, kad keleivis *sukuria* registruotą skrydį jei keleivis yra prisijungęs vartotojas.
- RS117. Leidžiama, kad asmuo *prisijungia prie* skrydžio užsakymo sistemos.
- RS118. Leidžiama, kad skrydžio užsakovas *patvirtina* užsakytą skrydį, jei skrydžio užsakovas *sukuria* užsakytą skrydį.
- RS119. Leidžiama, kad skrydžio užsakovas *gauna* užsakytą skrydį, jei skrydžio užsakovas *sukuria* užsakytą skrydį.
- RS120. Leidžiama, kad skrydžio užsakovas *patvirtina* apmokėtą skrydį, jei skrydžio užsakovas *sukuria* apmokėtą skrydį.
- RS121. Leidžiama, kad skrydžio užsakovas *gauna* apmokėtą skrydį, jei skrydžio užsakovas *sukuria* apmokėtą skrydį.
- RS122. Leidžiama, kad keleivis *patvirtina* registruotą skrydį, jei keleivis *sukuria* registruotą skrydį.
- RS123. Leidžiama, kad skrydžio užsakovas *sukuria* užsakytą skrydį.
- RS124. Leidžiama, kad keleivis *sukuria* registruotą skrydį.
- RS125. Leidžiama, kad skrydžio užsakovas *sukuria* apmokėtą skrydį.
- RS126. Įpareigojama, kad skrydžio užsakymo sistema *tikrina* užsakytą skrydį, jei skrydžio užsakovas *patvirtina* užsakytą skrydį“.
- RS127. Įpareigojama, kad skrydžio užsakymo sistema *tikrina* apmokėtą skrydį, jei skrydžio užsakovas *patvirtina* apmokėtą skrydį“.
- RS128. Įpareigojama, kad skrydžio užsakymo sistema *tikrina* registruotą skrydį, jei keleivis *patvirtina* registruotą skrydį“.
- RS129. Įpareigojama, kad skrydžio užsakovas *įveda* skrydžio duomenis, jei skrydžio užsakovas *sukuria* užsakytą skrydį.
- RS130. Įpareigojama, kad skrydžio užsakovas *įveda* keleivio duomenis, jei skrydžio užsakovas *sukuria* užsakytą skrydį.
- RS131. Įpareigojama, kad skrydžio užsakovas *įveda* papildomą informaciją, jei skrydžio užsakovas *sukuria* užsakytą skrydį.
- RS132. Įpareigojama, kad skrydžio užsakovas *įveda* kreditinę kortelę, jei skrydžio užsakovas *sukuria* apmokėtą skrydį.
- RS133. Įpareigojama, kad skrydžio užsakovas *nurodo* užsakytą skrydį, jei skrydžio užsakovas *sukuria* apmokėtą skrydį.
- RS134. Įpareigojama, kad keleivis *nurodo* apmokėtą skrydį, jei keleivis *sukuria* registruotą skrydį.
- RS135. Įpareigojama, kad skrydžio užsakymo sistema *patvirtina* užsakytą skrydį jei užsakytas skrydis yra teisingas užsakytas skrydis.
- RS136. Įpareigojama, kad skrydžio užsakymo sistema *patvirtina* apmokėtą skrydį jei apmokėtas skrydis yra teisingas apmokėtas skrydis.
- RS137. Įpareigojama, kad skrydžio užsakymo sistema *patvirtina* registruotą skrydį jei registruotas skrydis yra teisingas registruotas skrydis.
- RS138. Įpareigojama, kad skrydžio užsakymo sistema *tikrina* registruotą skrydį.
- RS139. Įpareigojama, kad keleivis *nurodo* apmokėtą skrydį.

Atributų reikšmių ribojimai

1. Privaloma, kad keleivio asmens kodo tekstas turi 11 simbolių.
2. Privaloma, kad el pašto tekstas turi @ simbolį.
3. Privaloma, kad telefono tekstas turi 9 simbolius.

Duomenų tipų taisyklės

4. Privaloma, kad išvykimo laikas yra laikas.
5. Privaloma, kad suminė kaina yra skaičius.
6. Privaloma, kad keleivio vardas yra tekstas.
7. Privaloma, kad telefonas yra tekstas.
8. Privaloma, kad paso numeris yra tekstas.
9. Privaloma, kad koeficientas yra skaičius.
10. Privaloma, kad pervesta suma yra skaičius.
11. Privaloma, kad bagazo svoris yra realus kaičius.
12. Privaloma, kad aukštis yra realus kaičius.
13. Privaloma, kad storis yra realus kaičius.
14. Privaloma, kad plotis yra realus kaičius.
15. Privaloma, kad vieta yra realus kaičius.
16. Privaloma, kad amžius nuo yra realus kaičius.
17. Privaloma, kad išvykimo data yra data.
18. Privaloma, kad yra patvirtintas yra boolean.
19. Privaloma, kad užsakyto skrydžio užsakymo numeris yra unikali seka.
20. Privaloma, kad skrydžio tipo vardas yra unikalus.
21. Privaloma, kad užsakymo data yra data.
22. Privaloma, kad užsakymo numeris yra tekstas.

4 priedas. 4 konceptų stiliai

Pagal SBVR standartą veiklos taisyklės ir fakto tipai yra apibrėžti keturiais konceptų stiliais [10]. Šie konceptų stiliai pateikti 4 priede 1 lentelėje.

1 lentelė. Keturi SBVR konceptų stiliai

Šriftas (angliškai)	Šriftas (lietuviškai)	Apibūdinimas	Pavyzdys (angliškai)	Pavyzdys (lietuviškai)
<u>term</u>	<u>terminas</u>	bendros sąvokos (daiktavardžiai) ir vaidmenys	<u>e-mail</u> , <u>passenger name</u> , <u>address</u> etc.	<u>el.paštas</u> , <u>keleivio vardas</u> , <u>adresas</u> ir kt.
<u>name</u> , <u>literal</u>	<u>vardas</u> , <u>literalus</u> , <u>simbolis</u>	tikriniai daiktavardžiai irrieveiksmai	<u>Lithuania</u> , <u>Vilnius</u> etc. <u>today</u> , <u>2 years of age</u> etc.	<u>Lietuva</u> , <u>Vilnius</u> ir kt. <u>šiandien</u> , <u>prieš 2 metus</u> ir kt.
<u>keyword</u>	<u>raktažodis</u>	žodžių kombinacija	each, at most one, it is permitted that etc.	Kiekvienas, tik vienas, leidžiama, kad ir kt.
<u>verb</u>	<u>veiksmažodis</u>	veiksmažodžiai	<u>specify</u> , <u>is for</u> , <u>includes</u> etc.	<u>turi</u> , <u>priklauso</u> , <u>apima</u> ir kt.