

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Registų apibendrinto integracijos modelio sukūrimas
Development of typical integration model for registers

Magistro baigiamasis darbas

Atliko: Martynas Lyčius (parašas)

Darbo vadovas: Doc. dr. Albertas Šermokas (parašas)

Darbo recenzentas: Prof. dr. (HP) Saulius Gudas (parašas)

Vilnius – 2018

Santrauka

Šiame darbe nagrinėjama registrų integracinių sąsajų bendrumo problema. Ši problema kyla dėl to, kad registruose sąsajos dažniausiai kuriamos nesinaudojant jokiais taisyklėmis ir kuriant integracines sąsajas kiekvienai naujai sistemai atskirai. Šiai problemai spręsti kuriamas registrų apibendrintas integracijos modelis, kuris aprašytų registrų sąsajų kūrimo taisyklės ir pateiktų būdą patikrinti ar sukurta sąsaja atitinka šias taisykles. Šio darbo tikslas ir yra sukurti ir aprašyti registrų apibendrintą integracijos modelį, kuriuo būtų nusakomos tam tikros taisyklės skirtos integracinių taškų duomenų aibės pasirinkimui registruose, pagal jų duomenų modelį. Modelio taisyklės kuriamos pasinaudojant ontologijų kūrimo taisyklėmis, o validavimui kuriama registrų apibendrinto integracijos modelio XML schema.

Raktiniai žodžiai: Registrų integracinės sąsajos, apibendrintas integracijos modelis, ontologija, XML schema.

Summary

This research is focused on registers integration points unification problem. This problem occurs because of lack of integration point development rules for registers and point to point integration of new systems and registers. Integration model for information systems and registers is being developed during this research. This integration model consists of registers integration point creation rules and means of these rules validation. The purpose of this research is development of such register and information systems integration model. Integration model rules are created using ontology development rules and XML schema is used for validation.

Key words: registers integration points, registers integration model, ontology, XML schema.

TURINYS

Įvadas	6
1. Informacinių sistemų ir registrų integracijų apžvalga ir analizė	8
1.1. Lietuvos registrų ir su jais besiintegruojančių sistemų apžvalga	8
1.1.1. Registrų apžvalga	8
1.1.2. Su registrais besiintegruojančios sistemos	19
1.1.3. Registrų ir su jais besiintegruojančių sistemų apžvalgos apibendrinimas.....	24
1.2. Informacinių sistemų integracijų duomenų modelių kūrimas.....	25
1.2.1. Verslo informacinių sistemų integracija.....	25
1.2.2. Esybių ryšių modelio pritaikymas duomenų integracijose.....	27
1.2.3. Integracijų su registrais ypatybės	29
1.3. Integracinių modelių kūrimo pavyzdžiai ir praktikos	29
1.3.1. Registrų ir valstybinių informacinių sistemų integracinių taškų kūrimo praktikos. Ontologijos.	30
1.3.2. XML duomenų standartai.....	31
1.3.3. Ontologijos kūrimas	34
2. Registrų apibendrinto integracijos modelio sudėtinės dalys ir taisyklės.....	37
2.1. Registrų integracinių sąsajų ontologija	38
2.1.1. Klasių apibrėžimas	38
2.1.2. Taksonomijos sudarymas	44
2.1.3. Ryšių nustatymas.....	46
2.1.4. Atributų aibės formavimas	47
2.1.5. Egzempliorių nustatymas	50
2.1.6. Ontologijos taisyklių nustatymas	51
2.1.7. Apibendrinimas	53
2.2. XML schema sąsajų sudarymui	53
2.3. Procesas modelio taikymui.....	55
3. Registrų apibendrinto integracijos modelio taikymas.....	57
3.1. Adresų registro sąsaja pagal sukurtą modelį	57

3.2. Adresų registro sąsajų palyginimas.....	59
3.3. Gyventojų registro sąsaja pagal sukurtą modelį.....	61
3.4. Gyventojų registro sąsajų palyginimas	62
3.5. Juridinių asmenų registro sąsaja pagal sukurtą modelį	63
3.6. Juridinių asmenų registro sąsajų palyginimas	65
3.7. Įgaliojimų registro sąsaja pagal sukurtą modelį.....	66
3.8. Įgaliojimų registro sąsajų palyginimas.....	67
3.9. Nekilnojamojo turto registro sąsaja pagal modelį.....	69
3.10. Nekilnojamojo turto registro sąsajų palyginimas	70
3.11. Taikymo apibendrinimas	72
Rezultatai ir išvados	74
Literatūros sąrašas.....	76
1 Priedas. Registrų integracinių sąsajų ontologijos taksonomija.....	79
2 Priedas. Registrų integracinių sąsajų ontologijos taksonomija su papildomais sąryšiais 80	
3 Priedas. Registrų integracinių sąsajų modelio XML schema	81
4 Priedas. Adresų registro atributų aibė pagal sukurtą modelį	84
5 Priedas. Gyventojų registro atributų aibė pagal sukurtą modelį	88
6 Priedas. Įgaliojimų registro atributų aibė pagal sukurtą modelį	89
7 Priedas. Nekilnojamojo turto registro atributų aibė pagal sukurtą modelį	91

Ivadas

Beveik kiekvienoje šiuolaikinėje informacinėje sistemoje iškyla integravimosi poreikis. Šis reikalavimas atsiranda dėl didėjančio poreikio gauti informacinėse sistemose saugomus duomenis iš išorės (pvz. interneto). Tai pat reikalingos naujos galimybės jau egzistuojančioms sistemoms dalintis duomenimis ar vykdomais procesais tarpusavyje ir su kitomis naujomis sistemomis [LIN03].

Norint užtikrinti informacinių sistemų integralumą svarbu atsižvelgti į esamas galimybes bei apribojimus. Kuriant integracines sąsajas labai dažnu atveju yra labiau atsižvelgiama į technologijas naudojamas integraciniam taškui sukurti, bet ne į pakankamos duomenų imties bei būtinų duomenų kuriant išorinę sąsają apibrėžimą. Tokiu būtu daugelis šiuolaikinių informacinių sistemų sukuria tam tikras sąsajas, kuriomis perduodamų duomenų aibė yra skirtinga, bei dažnai nepatenkina vėliau su šiomis sistemomis integruojamų kitų sistemų poreikių.

Tradiciškai integracijos yra kuriamos „iš taško į tašką“ (angl. *point-to-point*) metodu, kai integracinės sąsajos yra sukuriamos tik konkrečiam bendravimui tarp dviejų sistemų. [Win07] Kuriant tokiu būdu, kiekvienai informacinių sistemų porai, kurią reikia suintegruoti, beveik garantuotai reikalingas naujas integracinis taškas dėl skirtingų duomenų poreikio. Tai sukelia problemas ateityje, nes dažnu atveju vėliau sukurtos sistemos turi poreikį integruotis su anksčiau sukurtomis. Anksčiau sukurtų sistemų kūrimo projektai, bei projektavimas yra jau užbaigti ir reikalingi nauji kūrimo darbai bei tobulinimas. Tai ne tik stabdo integracijos sukūrimą, bet ir reikalauja papildomų išteklių, kurių būtų galima išvengti.

Šios problemos sprendimas būtų integracinių sąsajų kūrimas ne integracijai su konkrečia sistema, bet pateikiant reikalingą ir pakankamą duomenų aibę, kuria galėtų pasinaudoti išoriniai naudotojai. Duomenų aibės, kuria būtų galima nusakyti integracijai pakankamą duomenų kiekį sudarymas, yra vienas iš pagrindinių darbų, kurį reikia atlikti. Iš to iškyla poreikis sukurti tokį integracinių sąsajų kūrimo modelį, kurį būtų galima panaudoti kuriant informacinių sistemų integracines sąsajas.

Siekiant spręsti anksčiau minėtą integracijų problemą atliekamas mokslinis tiriamasis darbas. Darbo tikslas yra **sukurti ir aprašyti registrų apibendrintą integracijos modelį, kuriuo būtų nusakomos tam tikros taisyklės skirtos integracinių taškų duomenų aibės pasirinkimui registruose, pagal jų duomenų modelį**. Šis modelis spręstų sąsajų bendrumo problemą, kuri kyla, nes registrų esamos sąsajos nepatenkina poreikių ir kiekvienai naujai sistemai kuriama nauja integracinė sąsaja registre. Sukūrus tokį integracijos modelį bus siekiama parodyti jo tinkamumą bei pritaikomumą. Siekiant darbo tikslo įgyvendinami tokie uždaviniai:

1. Išanalizuoti literatūrą susijusią su integracinėmis sąsajomis bei jų aprašymu, panagrinėti esamų registrų ir informacinių sistemų specifikacijas, bei apžvelgti naudojamus registrų integracinių sąsajų kūrimo būdus, standartus ir pavyzdžius,
2. Aprašyti taisyklių rinkinį skirtą integracinės sąsajos registru aprašymui pasinaudojant registre saugomų duomenų modelių,
3. Pasiūlyti apibendrintą modelį registrų integracinėms sąsajoms kurti, į kurį įeitų anksčiau minėtos taisyklės, bei įrankis skirtas validuoti pagal taisykles sukurtą sąsajos aprašą,
4. Pagrįsti modelio tinkamumą, pritaikant jį pasirinktiems valstybės registrams.

Šiame darbe pateikiama keletas pasirinktų, dažnai naudojamų, Lietuvos Respublikos registrų integracinių sąsajų bei su jais besiintegruojančių sistemų apžvalga, nagrinėjami integravimo būdai. Nagrinėjamos kiekviename iš registrų esančios duomenų teikimo sąsajos, bei lyginamos saugomų ir teikiamų duomenų aibės. Taip pat analizuojamas sąsajų universalumas, pagrindžiant tai palyginimu tarp aprašomų besiintegruojančių sistemų duomenų poreikio bei realiai gaunamų duomenų aibės. Be to darbe nagrinėjami XML duomenų standartai, standartinis esybių ryšių modelis, bei kitų šalių praktikos kuriant integracinius taškus registrams. Taip pat aprašomas ontologijų kūrimas, bei pasiūlomas integracijos modelio taisyklių aprašymo būdas pasinaudojant ontologijomis. Analizuojant esamas sąsajas, kitų šalių patirtį, bei įvairius integracinių taškų sudarymo taisyklių kūrimo būdus, siekiama padėti pagrindus apibendrinto integracijos modelio tarp informacinių sistemų ir registrų sukūrimui. Aprašius apibendrinto integracijos modelio kūrimo principus, sudaromas pats modelis, bei aprašomas jo pritaikymas pasirinktiems valstybės registrams.

1. Informacinių sistemų ir registrų integracijų apžvalga ir analizė

Šiame skyriuje atliekama registrų integracinių sąsajų apžvalga, aprašomi su jais besiintegruojančių sistemų pavyzdžiai, nagrinėjama integracinių sąsajų sudarymą aprašanti literatūra. Taip pat aprašomos kitų šalių praktikos sprendžiant minėtą integracinių sąsajų bendrumo problemą. Be to nagrinėjami XML duomenų standartai ir ontologijos, bei pasiūloma kuriamo apibendrinto integracijos modelio struktūra.

1.1. Lietuvos registrų ir su jais besiintegruojančių sistemų apžvalga

Šiame poskyryje pateikiama Lietuvos Respublikos registrų ir pasirinktų su jais besiintegruojančių sistemų apžvalga. Apžvelgiant egzistuojančias integracines sąsajas siekiama identifikuoti pagrindines integracijų problemas, bei išsiaiškinti, ką modelis turėtų spręsti. Taip pat ieškoma tam tikrų bendrumų tarp registrų ir jų duomenų modelių, kuriuos vėliau bus galima panaudoti apibendrintam registrų integracijos modeliui kurti.

1.1.1. Registrų apžvalga

Pagal paskutinius duomenis elektroninėje erdvėje yra 94 Lietuvos Respublikos registrai [Reg18]. Šiame skyrelyje apžvelgiami penki pagrindiniai Lietuvos respublikos registrai [LRS18], gilinamasi į jų specifikacijas, bei juos reglamentuojančius įstatymus. Pagrindinis tyrimų objektas yra registrų integracinės sąsajos, kurios yra numatytos iš anksto. Nagrinėjami duomenų perdavimo būdai, bei lyginama saugomų ir perduodamų duomenų aibės.

Apžvelgiami šie registrai:

- Lietuvos Respublikos adresų registras
- Lietuvos Respublikos gyventojų registras
- Juridinių asmenų registras
- Įgaliojimų registras
- Nekilnojamojo turto registras

1.1.1.1. Lietuvos Respublikos adresų registras

Lietuvos Respublikos adresų registras – tai Lietuvos Respublikos registras, kuriame registruojami Lietuvos Respublikos administracinių vienetų ir jų ribų įstatyme nustatyti, Adresų formavimo taisyklėse, bei adresų registro nuostatuose numatyti Registro objektai [LRV15a]. Kaip vienas iš daugelio veiksnių su registrais (rinkimas, kaupimas, apdorojimas, saugojimas ir t. t.) adresų registro nuostatuose [LRV15a] nurodytas ir duomenų teikimas, kuris aktualus mūsų nagrinėjamoje temoje. Šiame skyrelyje detaliau panagrinėsime Lietuvos Respublikos adresų registrą ir jo specifikaciją ir apžvelgsime adresų registro duomenų aibę, bei jos pasiekiamumą per registre sukurtas duomenų gavimo priemones [Vir08a].

Pirmiausia panagrinėkime adresų registre saugomų duomenų aibę. Nuostatuose [LRV15a] aiškiai apibrėžiami septyni objektai, kurie saugomi adresų registre:

- Apskritis,
- Savivaldybės,
- Seniūnijos,
- Gyvenamosios vietovės,
- Gatvės,
- Adresai,
- Pavadinimai suteikti pastatui, statiniui ar kitam objektui.

Nuostatuose taip pat pateikiami bendrieji duomenys apie kiekvieną registro objektą, bei specializuoti kiekvieną iš jų apibūdinantys duomenys. Taip pat verta paminėti, kad saugomi duomenys yra ne tik tekstiniai bet ir grafiniai t. y. informacija apie objekto vietą ir geografinius parametrus. Žvelgiant iš duomenų mainų pusės ir nuostatuose nurodytos formuluotės – informacija apie visus registro objektus turi būti teikiama. Taigi, teikiamų ir kažkokia forma perduodamų duomenų aibėje turi atsirasti bent dalis duomenų, kurie aprašytų aukščiau minėtus registro objektus.

Toliau panagrinėkime informaciją apie adresų registro duomenų teikimą, kuri pateikta jau minėtuose adresų registro nuostatuose [LRV15a]. Registre teikiamus duomenis galima pasiekti viešai, naudojantis paieška, kuri pateikiama adresų registro tvarkytojo (Šiuo atveju – Valstybės įmonės Registrų centro) svetainėje. Ši sąsaja nors ir prieinama viešai, tačiau duomenys gauti iš jos negali būti panaudojami naudotojo registrų ar informacinių sistemų, todėl reikalingos kitokios priemonės duomenų apsaugai. Nuostatuose aiškiai apibrėžiama, kad duomenys gali būti teikiami kitoms sistemoms automatiškai arba rankiniu būdu įvairiomis elektroninių ryšių priemonėmis. Konkreti perduodamų duomenų aibė nėra nustatyta. Nuostatuose teigiama, kad teikiant duomenis elektroniniu būdu teikiamų duomenų aibė nurodoma duomenų teikimo sutartyse, nustatant duomenų perdavimo sąlygas, tvarką ir formatus. Taigi, įstatymiškai yra apibrėžta tik saugomų duomenų aibė.

Nepaisant to, kad nuostatuose nieko nepasakyta apie konkrečius teikiamus duomenis bei jų perdavimo būdą, specifikuojant adresų registrą yra numatytos standartinės sąsajos. Šios sąsajos skirtos integracijai su informacinėmis sistemomis ar kitais registrais. Joms yra apibrėžtos tam tikros teikiamų duomenų aibės ir perdavimo tvarka. Duomenų teikimui adresų registre sukurta duomenų rinkinių teikimo duomenų perdavimo kanalais posistemė, kuri užtikrina duomenų perdavimą duomenų gavėjams, kurie yra sudarę duomenų gavimo sutartį su registro tvarkytoju [Vir08a]. Per minėtąją sąsają duomenų gavėjas gali gauti pilnus arba dalinius duomenų rinkinių, bei duomenų pasikeitimų protokolus t. y. informaciją apie atributų pasikeitimus nuo paskutinės

duomenų versijos. Visa ši informacija yra pasiekama Registrų centro internetiniame puslapyje tekstinių bylų pavidalu ir duomenų įkėlimas į trečių šalių sistemas vyksta tų sistemų numatytais priemonėmis. Taip yra apibrėžiama vienintelio tipo standartinė sąsaja, kuri sukurta Adresų registre ir yra naudojama išorinių sistemų. Adresų registro specifikacijoje taip pat apibrėžiama šia sąsaja perduodamo duomenų rinkinio struktūra su aprašymais, kurią ir panagrinėsime bei palyginsime su nuostatuose pateikta saugomų duomenų aibe [LRV15a].

Adresų registro specifikacijoje kaip priedas pateikiamas galimas perduoti duomenų rinkinys. Visi šiame rinkinyje aprašyti duomenys gali būti pasiekiami informacijos gavėjui pasirašiusiam sutartį, t. y. yra aprašyta anksčiau minėta sąsaja, per kurią šiuos duomenis galima pasiekti tekstinių bylų pavidalu. Norėdami detaliau panagrinėti sprendimą, palyginkime perduodamų duomenų aibę su nustatyta saugomų duomenų aibe. Perduodamų duomenų aibėje galime matyti, kad perduodami keturių tipų objektai:

- Administraciniai vienetai,
- Gyvenamosios vietovės,
- Gatvės,
- Adresai.

Kaip matome perduodami objektai nesutampa su anksčiau minėtų saugomų objektų aibe. Tačiau panagrinėjus atidžiau galime matyti, kad visi šie objektai lengvai susisieja su saugomų objektų aibe ir apima informaciją apie kiekvieną iš jų (1 lentelė).

1 lentelė. Adresų registro saugomų ir teikiamų objektų aibių susiejimas

Teikiamas duomenų objektas	Saugomas duomenų objektas
Administraciniai vienetai	Apskritis, savivaldybės, seniūnijos
Gyvenamosios vietovės	Gyvenamosios vietovės
Gatvės	Gatvės
Adresai	Adresai, Pavadinimai suteikti pastatui, statiniui ar kitam objektui

Detaliau panagrinėkime ne tik pačius objektus, bet ir šių objektų atributus. Visi bendrieji registre saugomų objektų atributai taip pat aprašomi ir teikiamų duomenų aibėje. Kalbant apie perduodamus administracinius vienetus iš registro specifikacijos ir nuostatų matome, kad saugomų duomenų aibė sutampa su perduodamų, išskyrus tai, kad neperduodamos administracinio vieneto ribos LKS-94 formatu. Tą pačią išvadą galime pasakyti ir apie gyvenamųjų vietovių aprašymą. Gatvės objekto tipui tarp perduodamų atributų taip pat trūksta LKS-94 ašinės linijos koordinatų, bei neperduodamas gatvės tipas. Adresų objektams yra keletas smulkių neatitikimų, atsiradusių dėl po specifikacijos sudarymo sukurtų duomenų tipų. Tai pat prie adresų objektų nėra adreso

vietos taško. Adresų registras taip pat papildomai teikia klasifikatorius, kurie yra reikalingi aprašant registro objektus.

Išnagrinėjus adresų registro saugomų ir teikiamų duomenų aibes galime teigti, kad jos beveik sutampa. Pagrindinis skirtumas yra tas, kad specifikacijoje nėra numatyta kaip perduodami erdviniai duomenys, surišantys adresų registre saugomus objektus su konkrečia geografinė vietoje. Kadangi, šie duomenys taip pat gali būti reikalingi jiems turėtų būti kuriama atskira sąsaja, kuri nenumatyta specifikacijoje.

Apibendrinant, galime teigti, kad Lietuvos Respublikos adresų registras naudoja integracinę sąsają, kuria yra perduodama pilna duomenų atributų aibė kiekvienam registro objektui. Gaunami keturių tipų objektai, kurių pilnas registre esančias aibes galima gauti atskirai. Tai reiškia, kad registro duomenų gavėjas gali gauti tik pilną tam tikros rūšies objektų sąrašą su visais to tipo objektams priklausančiais negrafiniais atributais. Adresų registre tokio tipo sąsaja gali būti numatyta naudojantis logika, kad daugeliui sistemų-gavėjų yra svarbūs visi adresai, o ne kokie nors konkretūs atrenkami tik pagal tam tikrus parametrus. Taip pat, galima pasirinkti pagal objekto tipą priklausomai nuo to kiek detalaus adreso yra norima. Jeigu sistema-gavėjas tenkina aukščiau nurodytas sąlygas, tokio tipo adresų registro integracinė sąsaja jam tiks. Kitu atveju iškyla poreikis kitokiai integracinei sąsajai.

1.1.1.2. Lietuvos Respublikos gyventojų registras

Lietuvos Respublikos gyventojų registras – tai Lietuvos Respublikos registras, kuriame registruojami nuostatuose [LRV17] numatyti Registro objektai:

- Lietuvos Respublikos piliečiai,
- asmenys be pilietybės ir kitų valstybių piliečiai, gaunantys Lietuvos Respublikoje išduodamus asmens dokumentus, deklaruojantys gyvenamąją vietą Lietuvoje ar kurių civilinės būklės aktai yra registruojami Lietuvos Respublikos institucijose.

Registre yra apdorojami, saugomi bei teikiami duomenys apie aukščiau minėtus objektus. Nuostatuose [LRV17] pateikiama visa aibė atributų, kurie aprašo saugomą Registro objektą. Atributų aibė yra bendra abiem numatytiems registro objektams, tiek Lietuvos Respublikos piliečiams, tiek asmenims, kurie neturi pilietybės, bet dėl nuostatuose aprašytų priežasčių, jų duomenys saugomi gyventojų registre. Dėl šios priežasties duomenų aibės požiūriu tikslinga kalbėti tik apie vieną bendrą registre saugomą objektą – asmenį. Be apie asmenį saugomų duomenų taip pat nurodyta, kad naudojama vienuolika klasifikatorių. Taigi, saugomų duomenų aibė yra pakankamai tiksliai apibrėžta.

Panagrinėkime nuostatuose [LRV17] pateiktą informaciją apie gyventojų registro duomenų teikimą. Nuostatuose teigiama, kad duomenys yra vieši ir vadovaujantis įvairiais Lietuvos Respublikos įstatymais yra teikiami pagal registro tvarkytojo (Valstybės įmonė Registrų centras)

ir duomenų gavėjo pasirašytą asmens duomenų teikimo sutartį daugkartinio duomenų gavimo atveju arba duomenų gavėjo prašymą vienkartinio duomenų gavimo atveju.

Nuostatuose aprašyta, kad duomenys teikiami pagal Lietuvos Respublikos gyventojų registro duomenų teikimo taisyklės [LRT15a], kurias panagrinėsime detaliau. Kadangi nagrinėjame integracines sąsajas su sistemomis, aiškinsimės tik duomenų teikimą pagal sutartis, kuris yra skirtas daugkartiniam naudojimui apsikeitimui tarp sistemų. Duomenų teikimo taisyklėse [LRT15a] duomenys gali būti teikiami įvairiais būdais, tarp kurių ir integracines sąsajas tarp sistemų atitinkantys būdai - leidžiamosios kreipties arba paketinis duomenų teikimas internetu arba kitais elektroninių ryšių tinklais. Tikslus duomenų perdavimo būdas numatomas konkrečioje sutartyje. Ten pat nurodoma ir perduodamų duomenų aibė. Taip pat verta pastebėti, kad yra teikiami tik tie duomenys, kurių teikimui yra sukurtos techninės galimybės. Detaliau nei taisyklėse, nei nuostatuose apie konkrečią perduodamų duomenų aibę nėra užsimenama.

Detalesnei gyventojų registro teikiamų duomenų integracinių sąsajų analizei panagrinėkime šio registro specifikaciją [S4i10]. Specifikacijoje numatyti penki duomenų gavimo būdai. Paminėsime tik tuos duomenų perdavimo būdus, kurie yra skirti duomenų teikimui į kitas sistemas:

- Gyventojų registro objekto duomenų perdavimas duomenų perdavimo kanalu pagal duomenų gavėjo pateiktą automatizuotą užklausą. Šiuo būdu perduodant duomenis naudojamos duomenų bazių procedūros (angl. stored procedures). Duomenys teikiami per virtualų privatų tinklą (angl. VPN),
- Gyventojų duomenų bazės išrašo ir gyventojų registro duomenų pasikeitimų perdavimas sutartu periodiškumu. Atliekamas naudojant materializuotas virtualias lenteles (angl. materialized view) ir XML formatu išsaugotus registro objektų sąrašus. Teikiama per virtualų privatų tinklą (angl. VPN) FTP protokolu.

Taigi, kaip matome numatyti keli pakankamai lankstūs duomenų perdavimo būdai.

Taip pat panagrinėkime Lietuvos Respublikos gyventojų registro teikiamų duomenų aibę. Specifikacijoje nenurodyti konkretūs duomenys, kurie teikiami per aukščiau nurodytus duomenų perdavimo kanalus, tačiau dėl kanalų lankstumo galima tikėtis, kad pagal sutartį galima perduoti net ir visą registro duomenų aibę. Ši teiginį patvirtina ir vidinių duomenų srautų schema, kurioje išskiriama atskira duomenų bazė duomenų teikimui ir nurodoma, kad duomenys iš pagrindinės duomenų bazės yra replikuojami į duomenų teikimo duomenų bazę.

Apžvelgus gyventojų registrą reglamentuojančius įstatymus, bei jo specifikaciją matome, kad šiame registre yra numatytos standartinės sąsajos, bei aibės duomenų perdavimui. Projektuojant registrą buvo nuspręsta suteikti galimybę gauti visus saugomus duomenis pagal poreikį, tačiau integruojantis su nauja sistema neišvengiamai reikalingi tam tikri pritaikymai ir duomenų aibės apribojimai priklausomai nuo asmens duomenų teikimo sutarties.

Gyventojų registras pateikia gana lanksčią sąsają, kurią galima pritaikyti besintegruojančioms sistemoms. Nepaisant to, toks lankstumas sąlygoja tai, kad kiekvienai sistemai gali tekti pritaikyti esančias sąsajas rašant naujas duomenų bazių procedūras, kuriant virtualias materializuotas lenteles ar XML struktūras. Iš nuostatų ir techninės specifikacijos galima spręsti, kad Lietuvos Respublikos gyventojų registro duomenų teikimo sąsaja nėra visiškai standartizuota duomenų atributų aibės atžvilgiu ir nėra sukurtos konkrečios sąsajos, kurios būtų universalios jas naudojančioms sistemoms.

1.1.1.3. Juridinių asmenų registras

Juridinių asmenų registras – tai Lietuvos Respublikos registras, kuriame registruojami nuostatuose nurodyti [LRV16a] Registro objektai ir registruojami, įrašomi, renkami, kaupiami, apdorojami, sisteminami, saugomi ir teikiami duomenys, informacija ir dokumentai apie juos. Registro objektai:

- Privatūs juridiniai asmenys,
- Viešieji juridiniai asmenys,
- Juridinių asmenų filialai ir atstovybės.

Juridinių asmenų registro nuostatuose [LRV16a] nurodyta saugomų juridinio asmens objekto duomenų aibė. Apžvelgiant šią aibę galima matyti, kad viešieji ir privatieji juridiniai asmenys atributų aibe beveik nesiskiria todėl tikslinga juos nagrinėti kaip vieną objektą – juridinį asmenį. Filialai ir atstovybės turi atskirus atributų rinkinius. Registre saugomi objektų atributai išskiriami į tokias grupes:

- Bendrieji duomenys,
- Bendroji informacija,
- Specialieji registro duomenys,
- Specialioji registro informacija.

Iš viso yra 24 bendrųjų duomenų atributų tipai, kurie skirti aprašyti juridiniams asmenims. Tuo tarpu filialams bei atstovybėms pateikiami 9 bendrųjų duomenų atributų tipai. Taip pat nurodomi 32 juridinių asmenų bendrosios informacijos atributų tipai, kai filialams ir atstovybėms išskiriami 5 tokių atributų tipai. Taip pat yra numatomi keturi atributai skirti laikinam juridinio asmens įrašymui į registrą. Specialieji registro duomenys bei informacija registre yra grupuojami pagal teisinę formą, taip pat jie yra pateikiami atskirai filialams bei atstovybėms. Be visų šių atributų nurodoma aštuoniolika klasifikatorių, kurie skirti kai kurių atributų reikšmėms nurodyti.

Taip pat verta pastebėti, kad nuostatuose [LRV16a] nurodyta, jog juridinių asmenų registre naudojami duomenys iš kitų mūsų nagrinėjamų registrų – Lietuvos Respublikos gyventojų registro, Nekilnojamojo turto registro, bei Lietuvos Respublikos adresų registro. Detalus duomenų teikimo būdas iš kitų registrų nenurodomas, tačiau reikalaujama, kad duomenys būtų atnaujinti ne

daugiau kaip per 24 valandas nuo jų pasikeitimo automatiniu būdu. Duomenų aibė reikalinga juridinių asmenų registrai aiškiai nurodoma kiekvienam kitam besiintegruojančiam registrai.

Panagrinėjime nuostatuose [LRV16a] nurodytą informaciją apie duomenų teikimą. Duomenų gavimui yra numatyti penki skirtingi būdai, tačiau mes nagrinėsime tik vieną – automatinį, nes visi kiti būdai yra skirti gauti išrašus ar raštiškus duomenis asmenis, bet ne sistemoms. Registro tvarkytojas (Valstybės įmonė Registrų centras) ir duomenų gavėjas turi sudaryti duomenų gavimo sutartį tam, kad gavėjas galėtų gauti jam reikiamą informaciją automatiniu būdu. Sutartyje individualiai nurodoma tokių duomenų gavimo tvarka bei apimtis. Detalesnė informacija apie duomenų teikimą automatiniu būdu nuostatuose nepateikiama.

Detaliau analizuodami juridinių asmenų registro integracines sąsajas panagrinėjime šio registro specifikaciją [Vir10]. Specifikacijoje daug rašoma apie duomenų gavimo būdus iš juridinių asmenų registro svetainės, tačiau konkretūs automatiniai duomenų gavimo būdai nėra aprašomi. Vienas iš duomenų gavimo būdų, kuris greičiausiai gali būti automatinis ir yra aprašomas specifikacijoje, yra duomenų apie juridinius asmenis, bei finansines ataskaitas susijusias su registre saugomais objektais gavimas XML formatu. Šiuo formatu galima gauti tokius duomenis:

- JAR klasifikatorius,
- JA sąrašus,
- JA duomenų išrašus,
- Finansinių ataskaitų klasifikatorius,
- Vieno JA finansinių ataskaitų sąrašą,
- Vienos JA finansinės ataskaitos išrašą.

Labiausiai mus dominantys yra juridinių asmenų duomenų išrašai, kurie yra trijų tipų: trumpasis, pagrindinis ir išplėstinis. Konkrečios teikiamų išrašų duomenų aibės yra pateiktos Juridinių asmenų registro tvarkymo taisyklėse [Vir13]. Trumpajame duomenų išrašė nurodomi 11 atributų juridiniam asmeniui ir 9 filialams bei atstovybėms. Pagrindiniame duomenų išrašė prisideda dar 6 atributai juridiniams asmenims, bei dar 2 filialams ir atstovybėms. Išplėstiniame duomenų išrašė galima gauti visus duomenis saugomus registre, bei istorinius duomenis. Trumpojo bei pagrindinio išrašų duomenų aibių pasirinkimas greičiausiai buvo atliktas pagal dažniausiai naudojamus registro duomenis. Tokiu būdu buvo sukurtos kelios išskirtos sąsajos pagal poreikį neužkertant kelio gauti pilną duomenų aibę, jeigu naujai besiintegruojančiai sistemai to prireiktų. Taigi apibendrinant, galima teigti, kad duomenys iš juridinių asmenų registro gali būti teikiami trijų skirtingo dydžio duomenų aibių pavidalu pagal poreikį.

Juridinių asmenų registras turi standartizuotas duomenų aibes teikiamas per numatytus duomenų perdavimo kanalus. Visos saugomos duomenų aibės suskirstymas į tam tikrus poaibius

leidžia manyti, kad sąsaja buvo konstruota galvojant apie išorinių sistemų poreikį ir numatant tam tikras minimalias ir maksimalias reikalingas ir perduodamas duomenų aibes. Toks suskirstymas leidžia manyti, kad sąsajos yra pakankamai universalios, tačiau verta toliau nagrinėti ar jos yra pakankamos didžiąjai daliai sistemų ir gali būti efektyviai panaudojamos.

1.1.1.4. Įgaliojimų registras

Įgaliojimų registras – tai Lietuvos Respublikos registras, kuriame registruojami šie nuostatuose [LRV16b] nurodyti registro objektai:

- Notarų, konsulinių pareigūnų patvirtinti įgaliojimai,
- Notarų patvirtintiems prilyginami įgaliojimai,
- Informacinių technologijų priemonėmis sudaryti įgaliojimai.

Registre yra renkami, kaupiami, apdorojami, sisteminami, saugomi ir teikiami duomenys, informacija ir dokumentai apie registro objektus, bei atliekami kiti su registro tvarkymu susiję veiksmai [LRV16b].

Kiekvieną iš registro objektų aprašo tam tikra saugomų atributų aibė, kuri nurodyta įgaliojimų registro nuostatuose [LRV16b]. Notarų ir konsulinių pareigūnų patvirtintiems įgaliojimams bei notarų patvirtintiems prilyginamiems įgaliojimams aprašyti numatyti penkiolika atributų. Tuo tarpu informacinių technologijų priemonėmis sudarytiems įgaliojimams numatyta dešimt atributų. Taip pat papildomai saugomi dar keturi atributai aprašantys įgaliojančius asmenis bei įgaliotinius, bei dar keturi atributai, kurie skirti apibūdinti duomenų teikėjams. Kai kurių atributų reikšmės nurodomos kaip klasifikatoriaus reikšmė, todėl registre saugomi ir 9 klasifikatoriai.

Įgaliojimų registras naudojami duomenimis iš kelių mūsų nagrinėjamų registrų – Lietuvos Respublikos gyventojų registro fizinių asmenų duomenims gauti, bei Juridinių asmenų registro juridinių asmenų duomenims gauti. Nuostatuose [LRV16b] nurodyta kokie konkrečiai duomenys gaunami iš minėtų registrų. Detalesnė sąveika aprašyta duomenų teikimo sutartyse, kurios sudarytos su kiekvieno registro valdytoju.

Panagrinėkime nuostatuose [LRV16b] aprašomą registro duomenų teikimą. Pagal nuostatus yra trys formos, kuriomis gali būti pateikti registro objektų duomenys:

- Teikiant registro išrašus apie įgaliojimo įregistravimą, įgaliojimo pakeitimo duomenų įrašymą, įgaliojimo išregistravimą,
- Teikiant registro išrašus,
- Teikiant užsakomuosius registro duomenų rinkinius (suvestines)

Registro duomenys gali būti teikiami raštu, elektroniniu būdu, automatiniu būdu, leidžiamosios krypties (angl. authorized access) būdais ar kitokiomis saugiomis ryšio priemonėmis. Mes

nagrinėsime tik elektroniniu būdu perduodamus duomenis, kai vyksta sąveika tarp sistemų (angl. system to system).

Detalesnė informacija apie Įgaliojimų registro duomenų teikimą pateikiama Įgaliojimų registro objektų registravimo ir duomenų teikimo taisyklėse [Vic16]. Šiose taisyklėse pateikiamos teikiamų duomenų atributų aibės kiekvienai iš aukščiau nurodytų teikimo formų. Taip pat čia nurodyta, kad pvz. išrašai yra teikiami tik pateikus tam tikrus paieškos parametrus. Detaliau panagrinėjus taisyklėse nurodytas duomenų aibes galima pastebėti, kad vienokia ar kitokia forma įmanoma gauti visus registre saugomus duomenų atributus. Nors ir galima gauti visus duomenis, jie yra paskirstyti į keletą skirtingų aukščiau minimų duomenų rinkinių (formų) ir tokiu būdu bandoma atskirti sąsajas pagal tam tikrą paskirtį. Kodėl pasirinkti būtent tokie aibių suskirstymai nėra minima ir niekur neminima, kad šių aibių sudarymui naudotasi kažkokiu bendru modeliu. Taigi, čia matomas dar vienas bandymas pateikti duomenis iš registro, kuris skiriasi nuo anksčiau nagrinėtų tuo, kad pilna duomenų aibė yra išskirstoma per kelis duomenų rinkinius ir visą ją galima gauti tik naudojant kelias sąsajas.

Dabar panagrinėkime Įgaliojimų registro techninėje specifikacijoje [Vic10] numatytus duomenų teikimo taškus. Čia aiškiai pateikiami penki duomenų gavimo iš registro būdai:

- Paieška pagal fizinio asmens kodą ir įgaliojimo notarinio registro numerį,
- Paieška pagal asmens gimimo datą, vardą, pavardę ir įgaliojimo notarinio registro numerį,
- Paieška pagal juridinio asmens kodą ir įgaliojimo notarinio registro numerį,
- Paieška pagal juridinio asmens kodą arba registravimo numerį, užsienio valstybę, kurioje įregistruotas juridinis asmuo, pavadinimą ir įgaliojimo notarinio registro numerį,
- Paieška pagal įgaliojimo identifikavimo kodą registre.

Pastaba. Visose šiose sąsajose įgaliojimo notarinio registro numerį gali pakeisti konsulinio pareigūno ar asmens turinčio teisę atlikti notarinius veiksmus, suteiktas registracijos numeris.

Šiomis sąsajomis gali būti gaunami tokie duomenų poaibiai:

- Išsami informacija
- Dokumentų santrauka
- Archyvo išrašas

Šie duomenų teikimo poaibiai yra jau anksčiau minėtas ir teikimo taisyklėse [Vic16] apibūdintas būdas išskirstyti saugomų duomenų aibę į skirtingus duomenų rinkinius pagal poreikį.

Apžvelgus Įgaliojimų registro specifikaciją [Vic10] matome, kad yra numatytos kelios sąsajos pagal skirtingus parametrus grąžinančios tam tikrus saugomų duomenų aibės poaibius. Pagrindinis akivaizdžiai matomas šių sąsajų trūkumas – bet kokiai informacijai gauti reikalingas

įgaliojimo numeris. Tai reiškia, kad sistemai besiintegruojančiai su Įgaliojimų registru ir norinčiai gauti duomenis apie kažkokio konkretaus asmens visus įgaliojimus standartinėmis sąsajomis to pasiekti nepavyks. Iš to išplaukia, kad šio registro duomenų teikimas nėra visiškai lankstus ir dėl naujų sistemų poreikių gali tekti keisti integracinių taškų skaičių.

Įgaliojimų registro sąsajų konstravimas panašus į Juridinių asmenų registro sąsajų sudarymą. Čia taip pat yra išskiriamos skirtingo dydžio atributų aibės pagal skirtingas užklausas. Pagrindinis skirtumas yra tas, kad Įgaliojimų registras reikalauja konkretaus įgaliojimo numerio, kad būtų galima gauti kokius nors duomenis. Tai reiškia, kad sistema-gavėjas būtinai turi žinoti šį numerį, jei nori naudoti sąsają, kuri skirta visoms sistemoms bendrai. Tai stipriai apriboja galimų gauti duomenų aibę. Be to, iš karto matyti, kad sąsaja tikrai nėra universali ir skirta tik toms sistemoms, kurios žino arba gali gauti iš naudotojo įgaliojimo numerį.

1.1.1.5. Nekilnojamojo turto registras

Nekilnojamojo turto registras – tai Lietuvos Respublikos registras, kuriame registruojami nuostatuose [LRV15b] nurodyti Registro objektai ir tvarkomi jų duomenys. Nuostatuose apibrėžiami tokie registro objektai: Nekilnojamojo turto registro įstatyme nurodyti nekilnojamieji daiktai, daiktinės teisės į juos, šių teisių suvaržymai ir su nekilnojamaisiais daiktais, daiktinių teisių į juos suvaržymais bei daiktinėmis teisėmis susiję juridiniai faktai, taip pat įmonių dovanojimo, pirkimo–pardavimo ir nuomos sutartys [LRV15b].

Registre saugomi kelių tipų duomenys apie registro objektus:

- Bendrieji duomenys,
- Specialieji duomenys,
- Archyviniai Registro objektų duomenys.

Bendrieji duomenys yra bendri visiems registro objektams ir jiems aprašyti yra numatyti 4 atributai, kurie gali būti sudėtiniai t. y. susidėti iš kelių kitų atributų. Specialieji duomenys gali būti specifiniai ir kai kurie skirti tik tam tikro tipo saugomiems objektams apibūdinti ir jiems aprašyti numatyta 14 atributų, tarp kurių taip pat yra sudėtinųjų. Archyviniai duomenys susideda iš tų pačių bendrųjų ir specialiųjų duomenų, tačiau jie skirti apibūdinti istoriniams duomenims apie registro objektus. Kai kurių iš minėtų atributų reikšmės yra imamos iš klasifikatorių, kurių registre iš viso yra saugomi trys.

Informacijos aktualumui ir tikslumui užtikrinti Nekilnojamojo turto registras turi integracinius taškus su trimis iš mūsų nagrinėjamų kitų registrų. Integracinės sąsajos sukurtos su Lietuvos Respublikos gyventojų registru, Juridinių asmenų registru, bei Lietuvos Respublikos adresų registru. Nekilnojamojo turto registro nuostatuose [LRV15b] nurodyta, kad duomenys iš minėtųjų registrų turi būti gaunami automatinio būdu ne vėliau kaip per 24 valandas nuo jų įrašymo

ar pakeitimo registre. Taip pat nuostatuose kiekvienam iš besiintegruojančių registų yra nurodomi konkretūs duomenys, kurie yra gaunami ir naudojami Nekilnojamojo turto registro.

Panagrinėkime nuostatuose [LRV15b] aprašomą registro duomenų teikimą. Nuostatuose nurodyti keli duomenų gavimo būdai:

- Perdavimas automatinio būdu elektroninių ryšių tinklais,
- Teikimas raštu arba elektroninių ryšių priemonėmis,
- Teikimas peržiūrėti leidžiamosios krypties (angl. authorized access) būdu internetu arba kitais elektroninių ryšių tinklais.

Šiame darbe nagrinėsime tik mus dominantį perdavimą automatinio būdu, kai sistema bendrauja su kita sistema.

Taip pat nuostatuose nurodomos tokios duomenų teikimo formos:

- Registro išrašas, apimantis aktualius registro duomenis,
- Registro duomenų bazės išrašas, apimantis dalį arba visus aktualius ir archyvinius Registro duomenis,
- Pažyma, patvirtinanti įregistravimo faktą,
- Registro duomenų peržiūros suvestinė,
- Analitinė Registro informacija teikiama pagal atskiras registro duomenų teikimo sutartis,
- Kadastro žemėlapių ištrauka,
- Registro išrašas, apimantis nekilnojamojo turto sandoriui atlikti reikalingus nekilnojamojo turto kadastro ir registro duomenis,
- Dokumentų, susijusių su Registre įregistruotais objektais, skaitmeninės kopijos,
- Nekilnojamųjų daiktų kadastro bylose esančių dokumentų kopijos.

Labiausiai mus dominančios formos yra Registro duomenų bazės išrašas, bei informacija teikiama pagal atskiras duomenų teikimo sutartis. Iš šių duomenų teikimo formų galima spręsti, kad galima gauti tiek pilną visų Registre saugomų duomenų aibę, tiek šią aibę suskirstytą į poaibius. Taip pat galima spėti, kad pasirašius sutartį su Registro tvarkytoju (Valstybės įmonė Registrų centras) galima gauti tam tikrus, tik duomenų gavėjo sistemai reikalingus duomenų poaibius. Konkretūs teikiamų duomenų poaibiai nuostatuose nėra aprašomi.

Taip pat panagrinėkime kaip registro duomenų teikimas yra aprašomas Nekilnojamojo turto registro specifikacijoje [Vir08b]. Specifikacijoje yra aprašomas panaudojimo atvejis „Pateikti duomenis XML formatu“, kuriame nurodoma kaip registro duomenų teikimo posistemė bendrauja su išorinėmis informacinėmis sistemomis. Čia nurodoma, kad duomenys teikiami pagal specializuotą užklausą, kurios parametrai, bei pagal ją gaunami rezultatai XML formatu yra nustatomi sutartyje su įmone, kuri nori gauti nekilnojamojo turto duomenis. Gavusi nurodytos

formos užklausa duomenų teikimo posistemė suformuoja rezultatą XML formatu ir perduoda jį išorinei sistemai, kuri gautą XML rezultatą apdoroja. Specifikacijoje nėra numatytos jokios specifinės sąsajos skirtos duomenų tarp sistemų apsikeitimui.

Pagal anksčiau nurodytas duomenų teikimo formas galime spręsti, kad duomenų aibė gali būti tokia, kokios reikės sistemai gaunančiai duomenis, tačiau iškyla problema, kad kiekvienai iš naujų sistemų gali tekti kurti naują integracinę sąsają. Gali būti, kad integracinės sąsajos nėra tokios lanksčios ir yra numatytos standartinės duomenų teikimo sutartys, tačiau iš Nekilnojamojo turto registrą apibrėžiančių dokumentų tas nėra aišku ir galime daryti prielaidą, kad universalios duomenų sistemai į sistemą teikimo taško nėra.

Nekilnojamojo turto registras gali perduoti įvairias duomenų aibes pagal įvairaus tipo užklausas, kurių nustatymas paminėtas aukščiau. Išnagrinėjus šio registro nuostatus bei specifikaciją galima daryti išvadą, kad registras nėra orientuotas į universalios duomenų teikimo sąsajos sukūrimą ir duomenų atributų aibės yra kuriamos pagal pareikalavimą. Toks sąsajos konstravimo būdas iššaukia jau anksčiau minėtas dažnų pakeitimų ir pritaikymo prie kitų sistemų problemas, bei nėra labai praktiškas, nes kiekvienai sistemai reikia kurti naujas sąsajas.

1.1.2. Su registrais besiintegruojančios sistemos

Bendresniam vaizdui susidaryti reikia panagrinėti ne tik pačius registrus ir jų siūlomas integracines sąsajas bei teikiamus duomenis, bet ir keletą su registrais besiintegruojančių valstybės informacinių sistemų. Tokiu būdu galėsime pastebėti koks yra duomenų poreikis ir susidaryti tam tikrą vaizdą, kaip reikėtų konstruoti registrų informacijos perdavimo sąsajas. Nagrinėsime valstybės informacines sistemas, kurios integruojasi su anksčiau nagrinėtais registrais. Pasirinktos šios valstybės informacinės sistemos:

- Lietuvos teismų informacinė sistema,
- Valstybinės mokesčių inspekcijos integruota mokesčių informacinė sistema,
- Antstolių informacinė sistema.

1.1.2.1. Lietuvos teismų informacinė sistema

Lietuvos teismų informacinė sistema (LITEKO) – valstybės informacinė sistema, kurios tikslas yra elektroniniu būdu tvarkyti Lietuvos teismuose nagrinėjamų ir išnagrinėtų bylų duomenis, fiksuoti bylų nagrinėjimo eigą ir teikti teisės aktuose numatytas taikinamojo tarpininkavimo ir viešąsias elektronines paslaugas. [Nta16]

Lietuvos teismų informacinės sistemos nuostatuose [Nta16] numatyti šios informacinės sistemos duomenų teikėjai. Čia randame ir anksčiau nagrinėtus registrus t. y. Juridinių asmenų registrą, Lietuvos Respublikos adresų registrą ir Lietuvos Respublikos gyventojų registrą. Taip pat nuostatuose yra numatytas duomenų rinkinys, kuris turi būti gaunamas iš nurodytų registrų. Detaliau panagrinėsime kiekvieną iš nuostatuose nurodytų rinkinių.

Iš Juridinių asmenų registro pagal nuostatus LITEKO turi gauti tokius duomenis:

- Juridinio asmens pavadinimą, kodą ir buveinės adresą,
- Teisinę formą,
- Statusas
- Telefono numeris, elektroninio pašto adresas.

Pažvelgus į Juridinių asmenų registro teikiamus duomenis matome, kad visus juos galime gauti (su nedidele aibe perteklinių duomenų) imdami trumpąjį registro duomenų išrašą per standartinę Juridinių asmenų registro sąsają. Taigi, galime teikti, kad šiuo atveju pasiteisina Juridinių asmenų registre taikomas duomenų aibės skirstymas.

Iš Lietuvos Respublikos adresų registro LITEKO gauna visą arba dalinį Adresų registro duomenų rinkinį. Čia matomas standartinis pavyzdys kaip besiintegruojanti sistema prisitaiko prie registro integracinių sąsajų ir privalo imti tokį duomenų rinkinį koks yra suteikiamas. LITEKO nereikalauja erdvinių adresų duomenų, todėl esanti Adresų registro teikiamų duomenų aibė pilnai tenkina šios sistemos reikalavimus. Akivaizdu, kad kai kurie duomenys Lietuvos teismų informacinei sistemai yra pertekliniai, nes realiai čia naudojami tik adresų duomenys nurodantys vietoves ir panašiai, o visiškai nesvarbūs duomenys, kurie susiję su adresą patvirtinančiais dokumentais ir t. t. Taigi, galime teikti, kad iš Adresų registro gaunami ne tik reikalingi duomenys, bet ir duomenų perteklius.

Iš Lietuvos Respublikos gyventojų registro LITEKO gauna vardą, pavardę, asmens kodą, gimimo datą ir mirties datą, jei ji yra. Tai yra pakankamai maža duomenų aibė ir sprendžiant iš Gyventojų registro apžvalgos ją būtų tokia ir galima gauti. Iš duomenų aibės paprastumo taip pat galima spręsti, kad įmanoma panaudoti ir kitai sistemai skirtą sąsają. Gyventojų registras nespacificuoja konkrečios perduodamos duomenų aibės todėl didelė tikimybė, kad tokių sąsajų tarp registro ir sistemų galima rasti pakankamai daug. Bet kokiu atveju registras turėjo prisitaikyti prie LITEKO sistemos reikalavimų ir greičiausiai sukurta nauja specifinė sąsaja. Taigi, galima matyti standartinę anksčiau minėtą naujos sąsajos kūrimo naujai sistemai pavyzdį.

Panagrinėjus LITEKO ir registrų sąsajas aiškiai galime matyti įvairovę ir bendrumo trūkumą. Pateikti pavyzdžiai rodo duomenų aibės skirstymo, duomenų pertekliaus ir specifinių sąsajų kūrimo atvejus. Tokios sąsajų įvairovės įgyvendinimui reikalingi papildomi kaštai, registrų sąsajų keitimas ir panašiai. Problemos sprendimas – tam tikro registrų sąsajų bendrumo nustatymas ir jo laikymasis.

1.1.2.2. Valstybinės mokesčių inspekcijos integruota mokesčių informacinė sistema

Valstybinė mokesčių inspekcijos integruota mokesčių informacinė sistema (IMIS) – tai valstybės informacinė sistema, kurios paskirtis yra užtikrinti informacines sąsajas su VMI ir kitų

institucijų valdomais valstybės registrais ir valstybės informacinėmis sistemomis bei kitų juridinių asmenų valdomomis informacinėmis sistemomis, siekiant apjungti mokesčių mokėtojų registravimo, deklaracijų apdorojimo, mokesčių apskaitos ir išieškojimo procesų rezultatus, duomenų konfidencialumo, vientisumo ir prieinamumo principais, kaupiant moduluotus duomenis vienoje IS. [VMI09] Iš oficialaus IMIS sistemos aprašo pateikto jos nuostatuose, galima spręsti, kad viena pagrindinių šios sistemos dalių yra integracinės sąsajos. Dėl sistemos veiklos ir tvarkomų informacinių objektų pobūdžio integracijos su tam tikrais registrais yra būtinos teisingam IMIS sistemos darbui.

Nuostatuose [VMI09] nurodyta, kad Valstybinės mokesčių inspekcijos integruota mokesčių informacinė sistema integruojasi su šiais mūsų nagrinėtais valstybės registrais:

- Lietuvos Respublikos gyventojų registru,
- Lietuvos Respublikos adresų registru,
- Nekilnojamojo turto registru,
- Juridinių asmenų registru.

Nuostatuose taip pat pateikiami ir IMIS sistemos posistemiuose tvarkomi duomenys prie kurių nurodomas ir išorinis duomenų gavimo šaltinis (jeigu jis yra). Detalesni iš registrų gaunamų duomenų rinkiniai aprašomi IMIS sistemos specifikacijoje [VMI07]. Toliau ir panagrinėsime kiekvieno iš registrų IMIS sistemai teikiamus duomenis.

Specifikacijoje apibrėžiama, kad iš Lietuvos Respublikos gyventojų registro Integruota mokesčių informacinė sistema gauna vardą, pavardę, asmens kodą ir kitus duomenis numatytus mokesčių įstatymuose ir jų pagrindu priimtuose teisės aktuose. Pirmi trys atributai yra vieni pagrindinių gyventojų registro atributų ir juos gauti turėtų būti labai paprasta. Pagrindinė problema čia iškyla dėl kitų duomenų gavimo. Pagal šių duomenų apibrėžimo formuluotę galima spręsti, kad jų aibė keisis, greičiausiai reikės keisti ir naudojamą gyventojų registro sąsają. Tai reiškia, kad greičiausiai sąsaja bus pritaikyta būtent IMIS sistemai, nes kitu atveju būtų paveikiamos ir kitos su registru besiintegruojančios sistemos. Šią problemą galima būtų spręsti turint tam tikrą universalią sąsają, kurioje būtų perduodami tam tikri numatyti duomenų rinkiniai. Tada keičiantis IMIS sistemai ji galėtų tiesiog panaudoti perteklinius duomenis arba užklausti didesnės duomenų aibės.

Dar vienas registras su kuriuo integruojasi IMIS yra Lietuvos Respublikos adresų registras. Specifikacijoje [VMI07] numatyta, kad gaunamas duomenų apie administracinį vienetą rinkinys, kuriame pateikiami duomenys apie jo identifikavimą, registravimą, bei duomenys, kurie reikalingi mokesčių administravimo funkcijoms vykdyti ir yra numatyti įstatymuose arba pagal juos parengtuose teisės aktuose. Čia problema gali kilti tik jeigu bus reikalingi adresų registre saugomi erdviniai duomenys, nes visi kiti duomenys yra perduodami per standartinę sąsają pilnu duomenų

rinkiniu. Visas duomenų rinkinys šiuo atveju greičiausiai bus perteklinis, nes kalbama tik apie administracinius vienetus, o ne visus registro objektus. Čia gelbsti tai, kad galima gauti pilną duomenų rinkinį skirtą vien tik administraciniams vienetams, taigi pertekliniai duomenys bus teikiami tik šio tipo objektams.

IMIS taip pat integruojasi su Nekilnojamojo turto registru. Specifikacijoje [VMI07] aprašoma iš Nekilnojamojo turto registro gaunamų duomenų aibė: žemės sklypų ir kito nekilnojamojo turto identifikavimo, registravimo, savininko nuosavybės ir kiti duomenys, kurie numatyti įstatymuose ar įstatymų pagrindu priimtuose teisės aktuose. Pagal duomenų rinkinio aprašą galima spręsti, kad čia taip pat reikalinga atskira sąsaja skirta tik IMIS sistemai. Tai neišvengiamai didina registro integracinių taškų skaičių, didina sudėtingumą ir sumažina duomenų teikimo universalumą. To išvengti būtų galima turint universalų sąsajos kūrimo modelį.

Taip pat IMIS integruojasi su Juridinių asmenų registru. Iš šio registro Integruota mokesčių informacinė sistema gauna ūkio subjektų registravimo duomenis. Detaliau duomenų aibė specifikacijoje nėra aprašoma. Iš turimų duomenų galima spręsti, kad gaunamų duomenų aibė sutampa su standartine ūkio subjektų (juridinių asmenų) registravimo duomenų aibe. Juridinių asmenų registras pateikia galimybę gauti duomenis keliais skirtingų dydžių duomenų poaibiais ir dėl to IMIS sistemos kūrėjai galėjo pasirinkti pakankamą jiems duomenų aibę, kuri gali būti ir pilna registre saugomų duomenų aibė. Vienintelė problema čia galėtų būti pertekliniai duomenys.

IMIS sistema integruojasi su keturiais iš mūsų nagrinėtų registru. Beveik visos integracijos reikalauja specifinių duomenų IMIS sistemai ir iš specifikacijos galima spręsti, kad laikui bėgant šios duomenų aibės gali keistis pagal įstatymų ir teisės aktų reikalavimus. Specifinių sąsajų poreikis ir kūrimas specialiai vienai sistemai padidina integracinių sąsajų sudėtingumą ir dėl to iškyla poreikis sukurti universalias sąsajas, kurias būtų galima pritaikyti ne vien tik specifinių duomenų reikalaujančioms sistemoms, bei nereiktų visada teikti pilnos duomenų aibės.

1.1.2.3. Antstolių informacinė sistema

Antstolių informacinė sistema (AIS) – tai valstybės informacinė sistema, kurios tikslas yra sudaryti sąlygas gyventojams ir verslo subjektams nuotoliniu būdu dalyvauti antstolių vykdymo procese, varžytinėse ir aukcionuose sukuriant elektroninių vykdomųjų bylų, elektroninių varžytinių ir elektroninių aukcionų paslaugas [LRT15b]. Ši valstybės informacinė sistema kaip ir daugelis kitų integruojasi su registrais.

Antstolių informacinės sistemos nuostatuose [LRT15b] nurodyta, kad ši sistema integruojasi su visais penkiais mūsų nagrinėtais registrais – Nekilnojamojo turto registru, Juridinių asmenų registru, Lietuvos Respublikos adresų registru, Lietuvos Respublikos gyventojų registru, bei Įgaliojimų registru. Panagrinėsime kiekvieną iš AIS integracinių sąsajų su minėtais registrais detaliau.

AIS integracijos su Lietuvos Respublikos gyventojų registru teikiamų duomenų aibėje nuostatuose [LRT15b] nurodyta 12 atributų. AIS specifikacijoje [LRT15c] nurodyta, kad duomenys perduodami tiesiogiai prisijungiant prie Gyventojų registro duomenų bazės. Sprendžiant iš Gyventojų registro sąsajų analizės, AIS sistamai turėtų būti sukurta atskira materializuota virtuali lentelė (angl. materialized view), kurioje pateikiami AIS reikalingi duomenys. Tai atitinka atskiros sąsajos kūrimo atskiroms sistemoms principą ir iššaukia pakeitimus Gyventojų registre, jei atsiranda kitokie AIS sistemos poreikiai.

Sąsajoje tarp Antstolių informacinės sistemos ir Nekilnojamojo turto registro dalyvauja 11 atributų, tarp kurių yra sudėtinių (susidedančių iš kelių kitų atributų) [LRT15b]. Specifikacijoje [LRT15c] nurodyta, kad duomenys iš Nekilnojamojo turto registro gaunami naudojant internetines tinklines paslaugas (angl. web service) realiu laiku. Kadangi AIS reikalingi ir aktualūs ir archyviniai duomenys naudojamas Nekilnojamojo turto registro duomenų pateikimo forma, kai pateikiamas išrašas apimantis dalinį aktualių ir archyvinių duomenų rinkinį. Išrašas yra pritaikytas AIS sistamai ir kaip ir Gyventojų registro atveju remiasi principu – viena sistema, vienas integracinis taškas.

Nuostatuose [LRT15b] nurodoma, kad AIS iš Juridinių asmenų registro gauna 39 atributus, tarp kurių yra ir sudėtinių atributų. Į šiuos duomenis įeina tiek bendroji, tiek specialioji Juridinių asmenų registro informacija. Specifikacijoje [LRT15c] nurodoma, kad duomenys perduodami per tinklines sąsajas (angl. web service) realiu laiku. Kadangi duomenų aibė yra pakankami didelė ir apima beveik visus registro duomenis, AIS turėtų naudoti sąsają, kuri perduoda išplėstinį Juridinių asmenų registro duomenų rinkinį. Tokiu būdu naudojama universalesnė sąsaja ir būtent šiuo atveju perteklinių duomenų kiekis yra minimalus (1-2 atributai) ir perduodamų duomenų atributų skaičiaus atžvilgiu nesudarantis realios įtakos integracinės sąsajos našumui ar duomenų apdorojimo efektyvumui.

AIS naudoja dvylika Lietuvos Respublikos adresų registro teikiamų atributų [LRT15b]. Tai sudaro tik po kelis atributus iš kiekvieno Adresų registro teikiamų objektų atributų aibės. Pagal AIS specifikaciją [LRT15c] duomenys gaunami tiesiogiai jungiantis prie Registrų centro duomenų bazės ir pasiekiami realiu laiku. Tokia sąsaja Adresų registre nėra numatyta ir yra specifinė, kas reiškia, kad ji yra kurta specialiai AIS sistamai. Tai iškelia jau anksčiau minėtas specialių sąsajų vienai sistamai kūrimo problemas. Adresų registre esančios standartinės sąsajos naudojimas taip pat neišspręstų visų problemų, todėl, kad standartinė sąsaja perduodama duomenų aibė yra žymiai didesnė, nei AIS reikalingų ir atsirastų didelis duomenų perteklius. Žinant, kad Adresų registre saugomas didelis duomenų kiekis (visi Lietuvos Respublikos adresai), kiltų nemenkos našumo ir apdorojimo problemos. Taip pat verta pastebėti, kad nebūtų išpildomas reikalavimas gauti duomenis realiu laiku. Iš to atsiranda bendresnės ir lanksčiau pritaikomos sąsajos poreikis.

Iš Įgaliojimų registro AIS gauna 15 atributų neelektroniniu būdu ir 10 atributų elektroniniu būdu sudarytiems įgaliojimams. Tarp abiejų tipų atributų yra sudėtinių [LRT15b]. Specifikacijoje [LRT15c] nurodyta, kad duomenys gaunami per internetines tinklines sąsajas (angl. web service). Duomenys gaunami pagal naudotojų užklausas apie konkretų registro objektą. Gaunamų duomenų aibė yra pilna Įgaliojimų registre saugomų duomenų aibė. Sąsaja gauti visiems duomenims pagal konkrečią užklausą egzistuoja Įgaliojimų registre kaip standartinė ir nekyla nei duomenų pertekliaus nei sistemos prisitaikymo problemų. Šiuo atveju pilnas duomenų aibės teikimas yra naudingas, tačiau šių duomenų gali reikėti ne visada todėl negalime teigti, kad tokia sąsaja yra universali besiintegruojančioms sistemoms.

Antstolių informacinė sistema integruojasi su visais mūsų nagrinėtais registrais. Didžioji dalis sąsajų yra specifinės AIS sistemai. Iš to išplaukia visos anksčiau minėtos tokių sąsajų pasikeitimų ir pritaikymų problemos. Taigi, ir čia iškyla sąsajų universalumo problema.

1.1.3. Registrų ir su jais besiintegruojančių sistemų apžvalgos apibendrinimas

Išnagrinėjus penkis pagrindinius Lietuvos Respublikos registrus ir tris su jais besiintegruojančias sistemas buvo identifikuoti jų integraciniai taškai, juose kylančios problemos, bei atsirandantys bendrumai. Integracinės sąsajos buvo analizuojamos per integracijai naudojamų technologijų, bei teikiamos atributinių duomenų aibės prizmę. Analizės metu pastebėta, kad skirtingi registrai integracines sąsajas taip pat kuria skirtingai ir dažnu atveju informacinėms sistemoms tenka prisitaikyti prie jų specifikos arba kurti visiškai naujas integracines sąsajas, skirtas konkrečioms sistemoms.

Pastebėtos tokios problemos kylančios dėl to, kad nėra bendro būdo kurti integracines sąsajas:

- Sąsajos nėra konkrečiai apibrėžtos, kuriant registrų nuostatus ir specifikacijas. Tai sukuria prielaidą, kad integracinės sąsajos bus kuriamos kiekvienai naujai sistemai atskirai,
- Kiekvienam registrui sąsajos kuriamos taip kaip nusprendžia registro kūrėjas. Tokiu būdu gaunama didelė sąsajų įvairovė ir apsunkinamas naujų prie registrų sąsajų besijungiančių sistemų prijungimas. T. y. Reikia prisitaikyti prie kiekvieno registro sąsajos atskirai,
- Neegzistuoja vieningas sprendimas kaip kurti tinklines sąsajas registrams perduodamų duomenų atžvilgiu.

Toliau buvo pastebėta, kad registruose yra taikomi keli integracijos būdai duomenų aibės atžvilgiu:

- Perduodama visa atributų aibė,

- Aibė skirstoma į poaibius nuo mažiausio iki didžiausio, kurie yra perduodami pagal besiintegruojančios sistemos poreikį,
- Kiekvienai naujai besiintegruojančiai sistemai kuriamas tam tikras integracinis taškas pagal šios sistemos poreikį.

Taip pat buvo pastebėti tam tikri bendrumai tarp registrų duomenų modelių:

- Registrų integracinėse sąsajose apibrėžiamos duomenų teikimo operacijos,
- Registrų duomenų aibes galima skirstyti pagal bendrus kriterijus (pvz. Juridinių asmenų registro ar Įgaliojimų registro duomenų aibių skaidymas),
- Registrai apibrėžia pagrindinius saugomus objektus, bei pagrindinius objektus apibūdinančius papildomus objektus (Pvz. klasifikatorius).

Tolesnis žingsnis – surastos informacijos apie registrus pritaikymas kuriant apibendrintą integracijos modelį. Šiam tikslui įgyvendinti reikia panagrinėti kaip kuriamos informacinių sistemų integracijos, bei aprašomi duomenų modeliai. Taip pat reikia panagrinėti kitų šalių valstybės informacinių sistemų integracijų kūrimo praktikas ir panagrinėti kokie modeliai yra taikomi ten.

1.2. Informacinių sistemų integracijų duomenų modelių kūrimas

Pirmiausia pradėsime nuo teorinės duomenų modelių kūrimo apžvalgos. Aptarsime verslo sistemų integravimą, bei modelio tokioms integracijoms kūrimą. Taip pat pateiksime esybių ryšių modelį kaip vieną labiausiai taikomų teorinių modelių ryšiams tarp duomenų kurti. Galiausiai pateiksime registrų integracijų išskirtinumus.

1.2.1. Verslo informacinių sistemų integracija

Panagrinėsime verslo informacinių sistemų integracijos (angl. Enterprise application integration) teorines gaires. Verslo sistemų integracija yra plati sritis nusakanti planus, metodus ir įrankius skirtus modernizuoti, konsoliduoti ir koordinuoti informacinių sistemų veikimą versle. Paprastai jau egzistuoja tam tikros atskiros verslo sistemos, tačiau jos būna menkai susietos ir tam yra reikalingos verslo sistemų integracijos praktikos, padedančios sukurti vieningą ir apibendrintą sistemų integracijų sistemą [LSH03]. Šio tipo integracijose dažnai bandoma susieti tos pačios dalykinės srities aplikacijas, tačiau taikomas praktikas galima įgyvendinti ir daug plačiau, pavyzdžiui panaudojant jas mūsų nagrinėjamam registrų integracinių sąsajų, skirtų jais besinaudojančioms sistemoms, kūrimui.

Mūsų nagrinėjamas ir literatūroje aprašomas būdas kurti integracines sąsajas susideda iš trijų žingsnių [Lin03]:

- Duomenų identifikavimas,
- Duomenų susisteminimas,

- Verslo (mūsų atveju dalykinės srities) metaduomenų modelio sudarymas.

Panagrinėsime kiekvieną iš šių žingsnių detaliau ir aprašysime problemas kylančias registru integracijos atveju.

Duomenų identifikavimo fazėje reikia surasti duomenų struktūras, kurios turėtų būti naudojamos integracijos kontekste [Lin03]. Tokių struktūrų, kurios tenkintų besiintegruojančias sistemas, suradimo būdo aprašymas ir yra vienas iš didžiausių integracinio modelio kūrimo uždavinių. Mūsų apžvelgiamose praktikose [Lin03], skirtose bendros dalykinės srities verslo sistemoms, rekomenduojama išsiaiškinti duomenimis besinaudojančių sistemų poreikį, nagrinėjant konkrečias egzistuojančias sistemas. Registru atveju, su jais besiintegruojančios sistemos gali dar nebūti sukurtos. Iš to seka, kad reikalingos tam tikros taisyklės pagal kurias reiktų kurti universalų registro integracinį tašką. Taigi, tam tikros duomenų aibės nustatymo ir dalykinės srities nagrinėjimo būdai gali būti viena iš atliekamo modelio kūrimo užduočių.

Kitas svarbus duomenų identifikavimo žingsnyje pabrėžtas ir būtent registru integracinėms sąsajoms svarbus dalykas yra duomenų gavimo greitis. Išskiriamos trys tokio duomenų poreikio rūšys [Lin03]:

- Realiu laiku gaunami duomenys,
- Beveik realiu laiku gaunami duomenys,
- Vienkartinis duomenų gavimas.

Realiu laiku gaunami duomenys yra patys svarbiausi, todėl jų gavimo užtikrinimui turi būti numatytos sąsajos, kuriomis gaunami duomenys yra su naujausiais pakeitimais arba yra labai mažas pakeitimų pasirodymo uždelsimas [Lin03]. Tokio tipo duomenų gavimas yra reikalingas pavyzdžiui tikrinant įgaliojimus Įgaliojimų registre. Įgaliojimo galiojimas turi būti aktualus tikrinimo momentu, todėl periodiškai gaunami duomenys būtų nekorektiški ir neprasmingi be tikrinimo laiko.

Beveik realiu laiku gaunami duomenys – tai tokie duomenys, kurie yra gaunami iš sistemos – tiekėjo periodiškai. Tokio tipo duomenų poreikis yra gana dažnas registruose, kai duomenų aibės keičiasi retai ir tuo pasinaudojant galima su efektyvinti darbą kviečiant integracinį tašką periodiškai. Tokio tipo sąsajos pavyzdys galėtų būti adresų registro sąsaja. Adresų registro duomenys dažniausiai yra naudojami realiu laiku (pvz. pildant formas su adresais ir pan.), tačiau teikti užklausas kiekvieną kartą kai jis naudojamas yra neefektyvu, nes tai užima daug laiko, o adresų duomenys nesikeičia labai dažnai. Todėl tokio tipo sąsajai užtenka periodinio duomenų gavimo.

Vienkartinio duomenų gavimo atveju sąsajos greitis ir efektyvumas nėra labai svarbus, tačiau mūsų nagrinėjamame kontekste, tai yra viena iš rečiau pasitaikančių duomenų gavimo formų, todėl kuriant modelį ristišis prie šio mažiausiai reikalaujančio teikimo būdo neverta. Tokio

būdo pavyzdys galėtų būti kokių nors mažai besikeičiančių informacinio pobūdžio duomenų vienkartinis suvedimas, pavyzdžiui, valstybių sąrašas, kuris sistemos gyvavimo metu dažniausiai nepasikeičia arba gali pasikeisti labai minimaliai, nesutrikdydamas sistemos darbo.

Duomenų susisteminimo žingsnyje verslo aplikacijose yra surenkami visi galimi aplikacijos duomenys ir duomenų identifikavimo žingsnyje gautos struktūros [Lin03]. Duomenų susisteminimo žingsnis – tai paruošiamasis žingsnis prieš kuriant aplikacijos metaduomenų modelį. Registrų atveju turime tik vieną sistemą ir iš anksto numatytą saugomų duomenų modelį, todėl pats susisteminimas susiveda į pirmajame žingsnyje aprašytus duomenų surinkimo ir struktūrizavimo veiksmus.

Aplikacijos metaduomenų modelio sudarymas – tai integracijoje pateikiamų duomenų atrinkimas iš anksčiau susistemintų duomenų [Lin03]. Susisteminti duomenys šioje vietoje pateikia aibę alternatyvių struktūrų, kurias galima perduoti per integracinį tašką. Būtent aplikacijos metaduomenų modelio sudarymo stadijoje yra parenkamos tam tikros perduodamos duomenų struktūros. Integracinių taškų kūrimo sudėtingoms verslo sistemoms, šiame žingsnyje taip pat atliekamas ir struktūrų supaprastinimas susiejant dekomponuotas jų dalis įvairiais ryšiais ir taip supaprastinant integracinius taškus. Metaduomenų modelio integracijai parinkimas yra vienas iš pagrindinių universalios registro sąsajos kūrimo uždavinių. Sukurtas modelis turėtų pasiūlyti kaip parinkti duomenų aibę, kad būtų patenkintas besiintegruojančių sistemų poreikis.

Apžvelgus integracijų kūrimo verslo sistemose kūrimo gaires, matyti kylantys modelio kūrimo uždaviniai. Universalus modelis turėtų apibrėžti tam tikrą duomenų struktūrų parinkimo, bei besiintegruojančias sistemas tenkinantį jų pateikimo būdą. Nagrinėjant detaliau, apibendrinto integracijos modelio kūrimas turėtų susivesti į duomenų identifikavimo, sisteminimo ir susiejimo taisyklių aprašymą, bei šių taisyklių validavimo būdo pateikimą. Panašūs modelio kūrimo žingsniai yra pritaikomi praktiškai aprašant įvairius duomenų modelius.

1.2.2. Esybių ryšių modelio pritaikymas duomenų integracijose

Vienas iš dažniausiai taikomų praktinių duomenų aibės sudarymo ir ryšių nustatymo būdų yra klasikinis esybių ryšių modelis, kuris yra naudojamas reliacinių duomenų bazių kūrimui [Tha13]. Šis modelis yra naudojamas duomenų esybių ir tarp jų susidarančių ryšių nustatymui. Esybių ryšių modelis sukuria tam tikrą struktūrą pagal kurią galima sukurti duomenų modelį, bei duomenų bazių lenteles ir ryšius tarp jų. Kaip jau išsiaiškinome anksčiau, kuriant duomenų integracinius taškus taip pat iškyla teikiamų duomenų modelio sudarymo problema. Šiam duomenų modeliui kurti taip pat būtų galima pritaikyti esybių ryšių modelio principus. Be to nagrinėsime tik šį modelį, nes jis yra naudojamas Lietuvos Respublikos registrų specifikacijose aprašant saugomų duomenų aibes. Bet kokio kito modelio taikymas integracinėms sąsajoms pagal

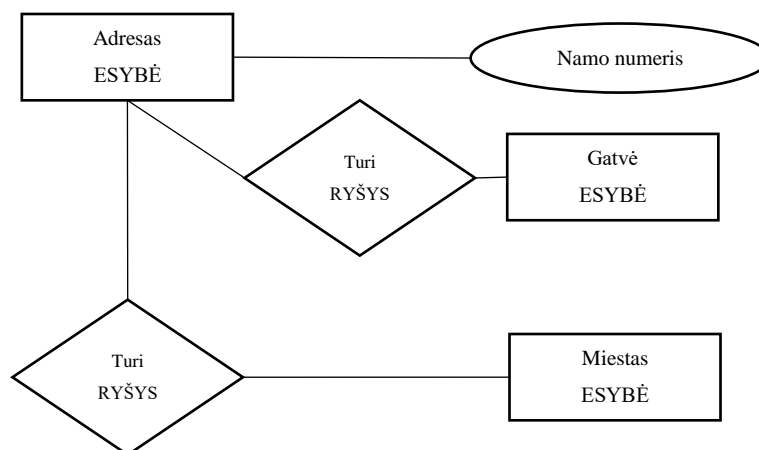
saugomų duomenų modelį kurti apsunkintų darbą, nes būtų reikalinga konvertavimas tarp modelių.

Detaliau panagrinėkime esybių ryšių modelį skirtą reliacinėms duomenų bazėms. Esybių ryšių modelis – tai modelis, kuris teigia, kad modeliuojamus objektus galima aprašyti esybėmis ir ryšiais tarp jų [Che76]. Kuriant struktūrą pirmiausia apibrėžiami baziniai objektai – esybės (angl. entities). Esybėse saugoma pagrindinė informacija, kurią norima saugoti kuriamoje struktūroje. Kiekvienai esybei yra apibrėžiami atributai, kurie ją apibūdina. Taip pat yra nustatomi ryšiai tarp esybių, kurie gali turėti įvairius tipus pvz. vienas su vienu, vienas su daug ir daug su daug. Paprastas modelio pavyzdys pateikiamas 2 paveikslėlyje.

Yra keli būdai kaip tokios esybės gali būti parenkamos ir susiejamos su objektais:

- Esybės atitinka objektus, o ryšiai atitinka kompleksinius objektus turinčius asociacijas. Esybių ir ryšių rinkinius atitinka klasės. Toks būdas yra naudojamas objektiniame programavime,
- Objektai traktuojami kaip duomenų saugyklos, kurios komunikuoja per funkcijas, kurias iššaukia įvykiai. Toks būdas naudojamas funkciniam programavime.

Pirmasis iš šių variantų yra klasikinis ir dažniausiai naudojamas. [Tha13]



1 pav. Esybių ryšių modelio pavyzdys.

Pagrindinis būdas sukurti esybes ir ryšius kuriant modelį yra išsiaiškinti kokių duomenų aplikacijai reikės, išsiaiškinant kliento poreikį ir sistemoje naudojamus duomenis. [Bag11] Tokiu būdu išsiaiškinamos esybės ir sukuriama tam tikri ryšiai tarp jų. Pritaikant šią praktiką kuriant tinklines sąsajas, pirmiausia reikia išsiaiškinti, kokius duomenis norėsime perduoti. Čia iškyla problema – kaip tinkamai parinkti esybes, kad jos būtų tinkamos didžiajai daliai paslaugos

naudotojų, nekeičiant pačios paslaugos. Šią problemą galima spręsti panaudojant esamą sistemos esybių ryšių modelį. Šio modelio transformacijos taisyklės skirtos paversti jį į tinklinėje paslaugoje naudojamą duomenų ryšių modelį ir yra vienas iš apibendrinto integracinio modelio sukūrimo svarbiausių etapų.

Esybių ryšių modelis naudojamas duomenų bazėje yra vienas iš sistemos duomenų struktūros atvaizdavimo būdų. Turėdami tokią duomenų struktūrą galime ją pritaikyti ir duomenų modelio integraciniam taškams sukūrimui. Duomenų bazės esybių ryšių modelis galėtų būti įeities taškas, o išeitis galėtų būti naujas integracinės sąsajos esybių ryšių modelis. Taisyklės kaip iš vieno duomenų modelio gauti kitą ir yra apibendrinto integracijos modelio esminė dalis.

1.2.3. Integracijų su registrais ypatybės

Apžvelgtos verslo sistemų integracijos (angl. Enterprise application integration) ir esybių ryšių modelis yra pakankamai bendrai pritaikomi. Registrai savo ruožtu yra siaura sritis, kurioje egzistuoja tam tikri ypatumai į kuriuos reikia atsižvelgti. Apžvelgsime registrų integracijas verslo sistemų integracijų kūrimo, bei duomenų modelio sudarymo kontekste.

Duomenų identifikavimas, sisteminimas ir struktūrizavimas registrų integracijų atveju yra gana griežtai apibrėžiamas pagal registruose saugomų duomenų aibę. Registruose aprašomi pagrindiniai juose saugomi objektai ir papildomi juos aprašantys duomenys. Tai jau yra tam tikras duomenų identifikavimo elementas. Taip pat registruose nusakoma duomenų hierarchinė struktūra, kurią reiktų naudoti ir kuriant integracijas. Objektų atributai gali būti skaidomi į duomenų rinkinius, kuriuos apibrėžia registrai. Kaip skaidyti į tokius duomenų rinkinius integracijos atveju turėtų nustatyti tam tikros taisyklės.

Registrų integracijos duomenų modelio sudarymo taisyklių esmė – saugomų registro duomenų modelio transformacija į perduodamų duomenų modelį. Duomenų modelis gali būti sudaromas įvairiai, tačiau registrų specifikacijose ir juos apibrėžiančiuose dokumentuose aprašomas naudojant būtent mūsų nagrinėtą esybių ryšių modelį. Iš to išplaukia, kad esybių ryšių modelis gali būti naudojamas ir integracijos duomenų modelio sudarymui, nes tokiu būdu nereikės atlikti transformacijos tarp skirtingų teorinių modelių.

Atsižvelgdami į registrų ypatybes turime panagrinėti praktinį duomenų identifikavimo, sisteminimo ir struktūrizavimo įgyvendinimą, bei integracijos duomenų modelio sudarymą. Šiam tikslui pasiekti reikia apžvelgti kitų šalių praktikas kuriant registrų ir valstybės informacinių sistemų integracijos taškus.

1.3. Integracinių modelių kūrimo pavyzdžiai ir praktikos

Čia pateikiamos kitų šalių integracinių modelių kūrimo praktikos bei taikomi metodai. Taip pat aprašomas šių praktikų kūrimo teorinis pagrindas ir literatūroje pateikiami aprašymai. Remiantis kitų šalių praktikomis ir jau sukurtais tam tikrais integracijos modelių kūrimo principais

nustatomos konkrečios gairės Lietuvos Respublikos registrų apibendrinto integracijos modelio kūrimui.

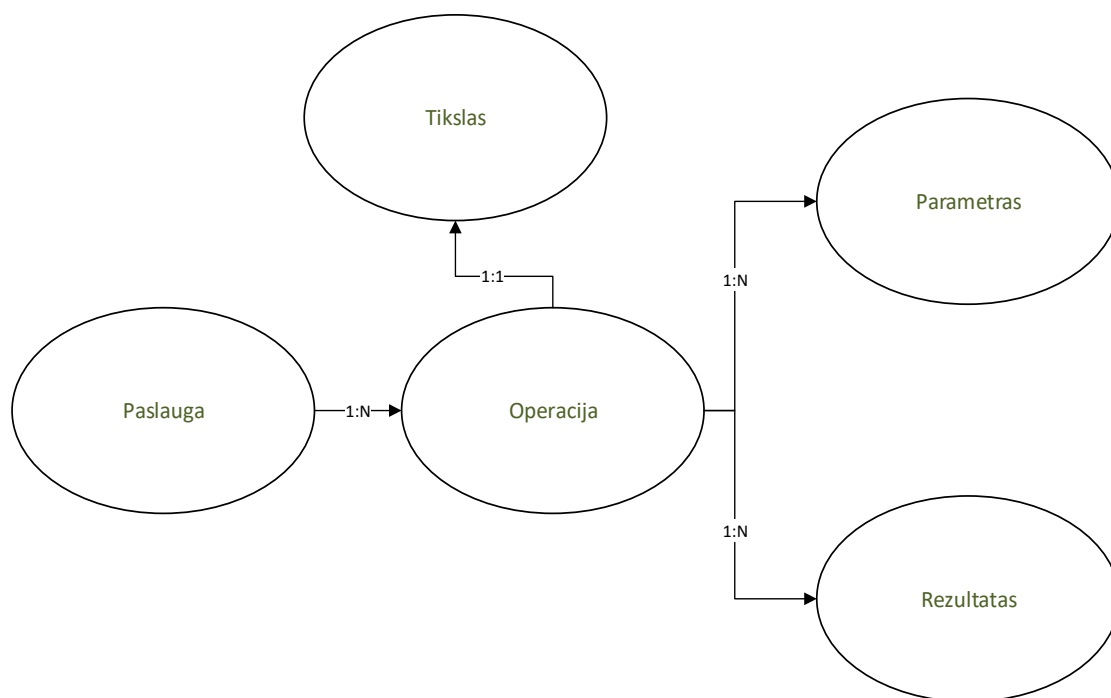
1.3.1. Registrų ir valstybinių informacinių sistemų integracinių taškų kūrimo praktikos. Ontologijos.

Šiame skyriuje apžvelgsime kitų šalių naudojamas viešojo sektoriaus registrų ir repozitorijų integracinių taškų kūrimo praktikas. Šios praktikos remiasi XML standartų sudarymo principu. Remiantis tuo yra sudaroma aibė XML schemų, kurios gali būti iš naujo panaudojamos, kuriant registrų integracinius taškus. Keletas pavyzdžių [Sou08]:

- Jungtinėje Karalystėje yra sukurta XML schemų biblioteka (angl. XML Schema Library), kurioje yra pateikiama apie 78 XML schemas,
- Danijoje egzistuoja InfoStructureBase sistema, kurioje yra tarptautinių standartų repozitorija su verslo procesų aprašymais, duomenų modelių aprašai, sąsajų aprašymai, XML schemas, bei jų fragmentai iš viešų ir privačių organizacijų, bei UDDI (angl. Universal Description, Discovery, and Integration) repozitorija su informacija apie tinklines paslaugas. (UDDI – tai universalus aprašymo, paslaugų paieškos ir integravimo protokolas. Kitaip sakant, tai yra registras, kuriame yra aprašytos tinklinės paslaugos su apibūdinimais ir jis gali būti apdorojamas automatiškai ieškant reikiamų tinklinių paslaugų [Zha14]),
- Vokietijoje yra XML InfoPoint, kuriame saugoma informacija apie atliktus, atliekamus ir planuojamus projektus su XML nuorodomis,
- Italijoje sukurtas Arianna projektas (angl. Arianna project), kuriame yra aprašyta elektroninių viešojo sektoriaus paslaugų ontologija, bei sudaryta paslaugų repozitorija su aprašais.

Kaip matome iš pavyzdžių daugelis jų remiasi XML schemomis ir tinklinių paslaugų ontologijomis. Ontologija – tai bendras žodynas, kuris gali būti panaudojamas modeliuoti tam tikrai sričiai. Jį sudaro aprašomos srities objektų ir sąvokų tipų, savybių, bei ryšių aprašas [AF108]. Ontologijos sukūrimas ir panaudojimas kuriant tinklines paslaugas iš karto nustato tam tikras taisykles kaip tokio tipo paslaugos turėtų būti kuriamos. Tai reiškia, kad galima apibrėžti tokią ontologiją, kuri būtų apibendrinta sričiai ir aiškiai apibrėžtų kaip reiktų sudarinėti tinklines paslaugas. Mūsų atveju apibrėžiama sritis galėtų būti Lietuvos Respublikos registrai ir ontologija aprašytų žodyną tinklinėms paslaugoms skirtoms registrų duomenų perdavimui į išorines sistemas. Tokiu būdu ontologija taptų tarsi taisyklių rinkinys, kurį būtų galima išreikšti per XML schemas. Dar viena ontologijos sukūrimo privalumas yra jos platus pritaikomumas. Ateityje ontologijos apibrėžimas leistų sukurti jau minėtą tinklinių paslaugų repozitorija UDDI ir registrų tinklinių paslaugų klientai (jais besinaudojančios išorinės sistemos) galėtų automatiškai rasti jiems

reikalingą registrą ar registro duomenis ir visais registrais naudotis kaip viena paslauga [Med05]. Bendras žodynas užtikrintų tikslų norimos informacijos gavimą, bei vienareikšmi interpretavimą tiek registro, tiek juo besinaudojančios sistemos pusėje.



2 pav. Ontologijos aprašo pavyzdys.

Panagrinėkime paprastą tinklinių paslaugų ontologijos pavyzdį (2 pav.). Šiame pavyzdyje labai abstrakčiai parodytos pagrindinės tinklinės paslaugos operacijos sąvokos ir objektai: paslauga, operacija, jos parametrai ir rezultatai, bei operacijos tikslas. Kiekvienas objektas yra susietas su kitais ryšiais, kurie yra apibrėžiami jų kardinalumu (1:1 – vienas su vienu, 1:N – vienas su daug, N:N – daug su daug). Išskirtinis objektas šiame paveikslėlyje yra Tikslas. Jis yra skirtas jau anksčiau minėtai automatinei tinklinių paslaugų paieškai. T. y. turėdami tam tikrą tikslų žodyną galėtume lengvai rasti reikalingas operacijas. Tokią ontologiją būtų galima pritaikyti beveik kiekvienai tinklinei paslaugai. Kuriant apibendrintą registrų ir valstybės informacinių sistemų integracinį modelį tikslinga būtų sukurti daug detalesnę ontologiją, kuri galėtų apibrėžti griežtesnes taisykles skirtas būtent registrams. Būtent tokiu būdu atsirastų ir reikalingi ribojimai, aprašyti nagrinėjant egzistuojančius registrus.

1.3.2. XML duomenų standartai

Turint apibrėžtą ontologiją reikalingas tam tikras būdas ją išreikšti. Kaip jau minėta ir aprašyta kitų valstybių praktikose, tą galima padaryti panaudojant XML schemas. Formalesnis tokių XML schemų iš ontologijų sudarymas galėtų būti XML duomenų standarto sukūrimas.

XML duomenų standartai (angl. XML Data Standards) – tai standartai pateikiantys bendras XML schemas, kurias galima taikyti įvairių veiklos sričių duomenims XML formatu vaizduoti [Lam09]. Šios schemas apibrėžia tam tikrą duomenų struktūrų sudarymo modelį. Šis modelis gali

būti pritaikomas verslo sistemoms, kurioms tas XML standartas yra skirtas. Tokiu būdu konkrečioms sistemų rūšims galima sukurti vieningą duomenų struktūrų sudarymo būdą. Taigi, galima sukurti ir tam tikrą universalią duomenų aibę apibrėžiant jos sudarymą XML standartu. Apibrėžiant panašius standartus registrų duomenų teikimo sąsajų kūrimui būtų galima apibrėžti ir apibendrintą informacinių sistemų ir registrų integracinį modelį.

Yra keletas dabar egzistuojančių XML duomenų standartų [Lam09]:

- cXML (commerce eXtensible Markup Language),
- eBIS-XML,
- OAGIS (Open Applications Group Integration Specification),
- UBL (Universal Business Language),
- XBRL (eXtensible Business Reporting Language),
- xCBL (XML Common Business Library).

Jų idėja yra labai panaši, todėl bendram principui suprasti aptarsime tik keletą iš jų.

1.3.2.1.cXML

cXML – tai atviras XML standartas skirtas užtikrinti lengvą apsikeitimą duomenimis apie transakcijas tarp partnerių [Shi00]. Jis skirtas elektroninės prekybos sistemoms. Šiame standarte aprašomi keli komunikavimo būdai – siunčiant užklausą ir gaunant atsakymą į ją ir vienos krypties asinchroninė komunikacija, kai tik siunčiami tam tikri duomenys nesitikint atsakymo. Žinutės susideda iš dviejų standartinių dalių t. y. antraštės (angl. header), kuriame įrašoma autentifikacijos bei kita meta informacija ir kūno (angl. envelope), kurio viduje yra aprašoma visi perduodami duomenys.

Standartas aprašo transakcijas, kurios susideda iš dokumentų [Ari17]. Dokumentai tai yra realių dokumentų naudojamų e-prekybos organizacijose atitikmuo. Standarte išskiriami trys pagrindiniai, labiausiai naudojami [Ari17] dokumentų tipai:

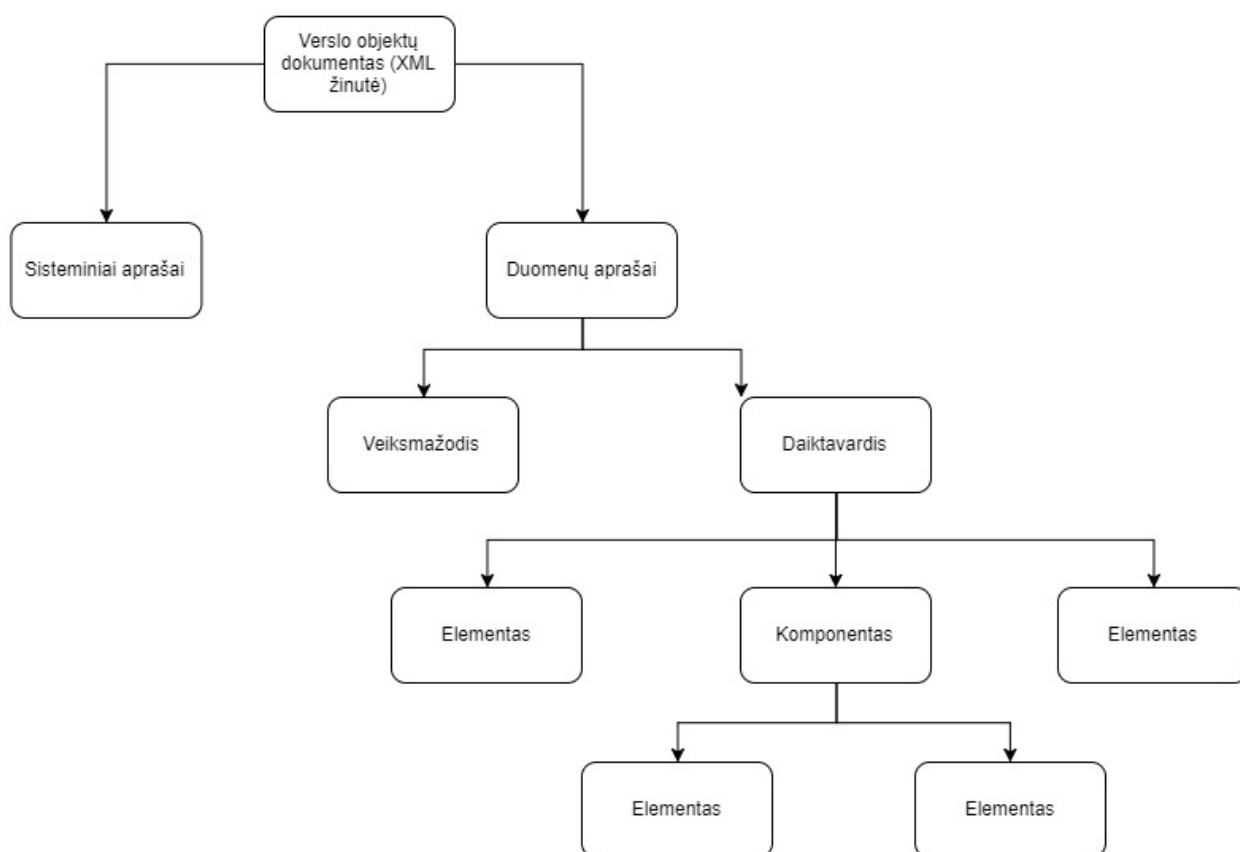
- Katalogai (angl. Catalogs) – bylos turinčios savyje informaciją apie prekes,
- Specialus apsikeitimo protokolas (angl. PunchOut) skirtas bendravimui tarp aplikacijų,
- Užsakymai (angl. Purchase orders) – standartinis prekių užsakymo dokumentas.

Šie tipai atitinka realius, standartą kūrusių prekybos organizacijų sistemose naudojamus, dokumentus [Ari17]. Tai reiškia, kad standartas apibrėžiamas randant esybių panašumus tarp visų vienos veiklos rūšies sistemų. Be to, pagal standarto schemą sukurtas žinutes galima labiau pritaikyti konkrečiai sistemai, modifikuojant schemą. Keičiant schemą laikomasi tų pačių taisyklių, todėl nėra nukrypstama nuo apibendrinto standarto.

Taigi, cXML yra tam tikras apibendrintos sąsajos kūrimo standarto pavyzdys skirtas e-prekybos sistemoms. Tai yra vienas iš būdų aprašyti modelį ir jo modifikacijas. Nepaisant to, naudoti cXML visoms registrų sąsajoms negalima, nes šis standartas apibrėžia pakankamai siaurą bendrą kontekstą sritį – e-prekybos. Daug praktiškesnis variantas būtų aprašyti panašų modelį ir sąsajų sudarymo taisykles skirtas būtent registrų sistemoms. Tokiu atveju svarbu apgalvoti kaip apibendrinti skirtingų veiklos sričių esybes.

1.3.2.2. OAGIS

OAGIS (angl. Open Applications Group Integration Specification) – tai specifikacija aprašanti standartą apsikeitimui duomenimis tarp verslo sistemų. Jis skirtas e-prekybos, gamybos, logistikos, klientų ryšių valdymo (angl. Customer Relationship Management) ir verslo išteklių planavimo (angl. Enterprise Resource Planning) sistemoms. OAGIS standartas pateikia pilną WSDL biblioteką, kurią galima panaudoti kuriant tinklines sąsajas [Chi17].



3 pav. OAGIS žinutės struktūra.

OAGIS specifikuoja 1057 verslo objektų dokumentus (angl. Business Objects Document) ir 64 verslo procesų scenarijus, kuriuose šie dokumentai gali būti panaudojami. Verslo objektų dokumentas – tai bendra XML žinutės struktūra, kurią pateikia OAGIS. Šios struktūros yra užrašomos XSD formatu. Pagrindinė tokių žinučių sudarymo idėja yra matoma pirmame paveikslėlyje. Žinutės remiasi dar kartą panaudojamais komponentais ir susidaro iš daiktavardžio (angl. noun), atitinkančio verslo objektą, ir veiksmažodžio (angl. verb), apibūdinančio koks

veiksmas atliekamas su objektu. Taip pat žinutėje gali būti pateikiami ir tam tikri sisteminiai duomenys, kurie gali būti naudojami autentifikacijai ar panašiai. Žinutės duomenų dalies sudarymo iš daiktavardžio ir veiksmožodžio pavyzdys galėtų būti toks: modelyje aprašytas *adreso* objektas ir veiksmas *gauti*, tada galėtume aprašyti žinutę *gautiAdresą*. Toks žinučių sudarymo būdas yra geras tuo, kad konkretus veiksmas nėra pririšamas prie objekto ir galima tą pačią logiką naudoti daugeliui elementų. Be to papildomam plečiamumui užtikrinti OAGIS turi specialius tipus *UserArea* prie kiekvieno veiksmožodžio, kuriuose leidžiama aprašyta papildomą informaciją ir pridėti papildomus duomenis kiekvienam iš standarte nurodytų objektų.

1.3.3. Ontologijos kūrimas

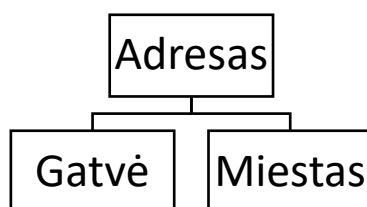
Norint aprašyti apibendrinto registrų ir informacinių sistemų integracijos modelio taisykles galima sukurti ontologiją, kuri tas taisykles ir apibrėžtų. Šiame skyriuje detaliau apžvelgsime iš ko susideda ontologijos ir kaip jos gali būti kuriamos.

Ontologija apibrėžia tam tikros srities žodyną. Aprašant srities ontologiją paprastai reikia apibrėžti šiuos objektus [Ber07]:

- Klases – objektus, kurie skirti atvaizduoti aprašomos srities sąvokoms,
- Ryšius, kurie skirti atvaizduoti sąsajas tarp apibrėžiamų sąvokų,
- Atributus, kurie skirti apibūdinti sąvokų savybes,
- Formalias aksiomas, skirtas apibrėžti visada ontologijoje teisingiems sakiniams,
- Funkcijas – specialių tipų ryšius,
- Egzempliorius (angl. instance), skirtus apibrėžti konkrečius vienetinius objektus.

Ontologijų kūrimui egzistuoja įvairių metodikų, tokių kaip TOVE, Enterprise model approach, METHONTOLOGY, On-To-Knowledge ir pan. [Iqb13] Pagrindiniai visų tokių ir panašių metodologijų žingsniai skirti ontologijų kūrimui yra tokie [Ber07]:

- Apibrėžti klases,
- Sustruktūrizuoti jas į taksonomiją. Taksonomija – tai hierarchinė struktūra sukurta remiantis tos struktūros elementų savybėmis [Van03] 4 paveikslėlyje pateikiamas paprastas taksonomijos pavyzdys.



4 pav. Taksonomijos pavyzdys su adreso, gatvės ir miesto klasėmis.

- Nustatyti ryšius tarp klasių,
- Nustatyti atributus ir jų galimas reikšmes,

- Nustatyti egzempliorius,
- Turint užtekinai laiko ir žinių nustatyti aksiomas ir funkcijas. Verta pastebėti, kad be šios dalies ontologija dažnai yra laikoma „silpna“ arba iš vis tai laikoma tik taksonomija. Nepaisant to, pagal poreikį gali užtekti gerai apibrėžtos taksonomijos. [Ber07].

Taip pat apibrėžiamos kelios ontologijų kūrimo idėjos: [Ber07]

- Sritį apibrėžti sąvokomis yra sunku ir dažnu atveju egzistuoja keletas apibrėžimo variantų, todėl nėra bandoma pagrįsti, kuris variantas geresnis, o bandoma surasti tokį, kuris būtų priimtinas visoms pusėms,
- Ontologijos kūrimas yra iteratyvus procesas ir kūrimo žingsniai yra kartojami iteracijomis,
- Vienas iš pagrindinių būdų pereiti nuo realaus pasaulio iki abstrakcijos, kuriant ontologiją, yra įsigilinti į apibrėžiamos srities daiktavardžius ir veiksmažodžius. Daiktavardžiai atitinka klases, o veiksmažodžiai ryšius. Verta pastebėti, kad panašiu principu remiasi ir mūsų apžvelgtas OAGIS XML duomenų standartas.

Siekiant sukurti ontologiją reikia detaliau panagrinėti kiekvieną jos kūrimo žingsnį. Toliau aprašysime šiuos žingsnius. Verta paminėti, kad skirtingos metodologijos siūlo skirtingą jų atlikimo tvarką, tačiau esmė išlieka panaši.

Pirmiausia apžvelgsime pagrindinių ontologijos objektų t.y. klasių, atributų, egzempliorių ir ryšių apibrėžimą. Vienas iš būdų apibrėžti visus šiuos objektus yra sudaryti apibrėžiamos srities daiktavardžių ir veiksmažodžių sąrašą. Kiekvieną iš sąrašo elementų reikia stengtis apibūdinti kuo plačiau, nes vėliau tai padės nustatyti jo tipą. Sudarius tokią elementų aibę galima juos išskirstyti pagal tipus. Visi veiksmažodžiai yra ryšiai, todėl prie jų verta pasižymėti kokius daiktavardžius jie sujungia. Daiktavardžiai gali būti skirstomi į klases, atributus arba egzempliorius. Klases apibūdinantys daiktavardžiai dažniausiai yra kažkokios abstrakcijos pvz. tai galėtų būti adresas. Atributai apibūdina kitus daiktavardžius pvz. namo numeris yra adreso atributas. Egzemplioriai nurodo kokį nors vienetinį, specifinį objektą, kuris nėra apibendrintas pvz. Seimo pastato adresas. Atrinkti visas sąvokas ir jas sudėlioti teisingai iš karto beveik neįmanoma, todėl procesas atliekamas iteratyviai, vis grįžtant prie šio žingsnio [Ber07].

Turint tam tikrus objektus reikia juos klasifikuoti į taksonomiją. Tai daroma surandant hierarchinius ryšius tarp objektų. Tokių ryšių taksonomijoje gali būti keturių tipų [Cor05]:

- Poklasis (angl. Subclass-Of). Objektas B yra A poklasis tada ir tik tada kai kiekvienas B egzempliorius yra ir A egzempliorius. Tai yra standartinis klasių paveldėjimo sąryšis,

- Suskaidymo dekomponavimas (angl. Disjoint-Decomposition). Tai yra objekto A poklasių aibė, kurie neturi bendrų egzempliorių ir nesudaro visos A poklasių aibės, t. y. gali būti tokių A egzempliorių, kurie nepriklauso nei vienai klasei iš dekomponavimo poklasių aibės. Tokio ryšio pavyzdys galėtų būti administracinių vienetų savivaldybių ir seniūnijų poklasiai. Vienas administracinis vienetas negali būti ir savivaldybė ir seniūnija, bei ne visi administraciniai vienetai yra arba savivaldybės arba seniūnijos,
- Išsamus dekomponavimas (angl. Exhaustive-Decomposition). Tai yra objekto A poklasių poaibis, kuris pilnai padengia A poklasių ir šie poklasiai gali turėti bendrus egzempliorius. Tokio dekomponavimo pavyzdys galėtų būti juridinių asmenų dekomponavimas į viešus ir privačius. Kiekvienas juridinis asmuo yra arba viešas arba privatus, o egzistuoja ir toks variantas, kai valdyti asmenį gali tiek viešasis sektorius, tiek privatus asmuo bendrai,
- Skaidymas (angl. partition). Objekto A skaidymas į poklasių yra tokios poklasių aibės sudarymas, kuri pilnai padengia objekto A poklasių, bei poklasiai negali turėti bendrų egzempliorių. Tokio skaidymo pavyzdys galėtų būti gyventojų skaidymas į pilnamečius ir nepilnamečius. Visi gyventojai priklauso arba vienam arba kitam poaibiui ir nėra tokių gyventojų, kurie nepriklausytų nei vienam iš jų.

Šių paminėtų taksonominių ryšių nustatymas tarp objektų ir yra pagrindinis taksonomijos sudarymo uždavinys.

Ryšiai nustatomi iš jau anksčiau sudarytų veiksmažodžių sąrašo. Kiekvienam ryšiui yra suteikiamas pavadinimas, nustatoma ryšio kryptis, kardinalumas (kiek vienos klasės egzempliorių gali būti susieta su kitos klasės egzemplioriais) ir t. t. Kuriant ryšius atsiranda panašių veiksmažodžių, kuriuos galima apibendrinti ir taip sukurti tam tikrus ryšių tipus. [Ber07]

Atributai nustatomi iš daiktavardžių sąrašo. Pagrindinės gairės nustatant atributą: reikia bandyti nurodyti atributą bendriausiai įmanomai klasei, kuri gali jį turėti, atributas turėtų turėti apibrėžtą tipą, kuris nebūtų klase, rekomenduotina apibrėžti galimas reikšmes, tikslumą ir pan. [Ber07]

Aksiomos ir taisyklės (funkcijos) nustatomos, kai norima pasiekti ontologijos pilnumo. Aksiomos pavyzdys galėtų būti – negali būti dviejų identiškų adresų. Taisyklės (funkcijos) pavyzdys galėtų būti – pirmų gyventojų asmens kodo skaičių gavimas iš lyties, bei gimimo datos.

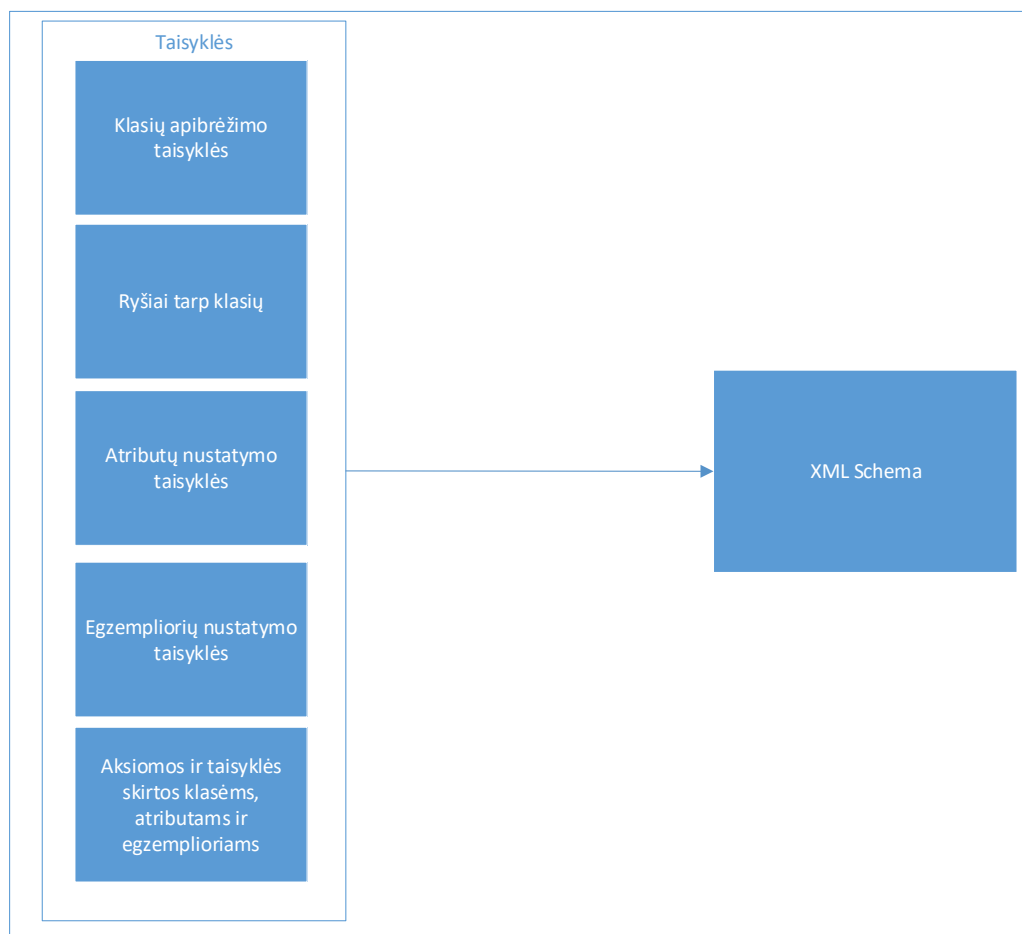
Pasinaudojant šiame skyriuje apibrėžta metodika, bei reikalingais aprašyti objektais galima vienareikšmiškai aprašyti norimą ontologiją. Taigi, turėdami tam tikrą ontologijų kūrimo metodiką ir XML duomenų standartus, techniškai aprašančius ontologijas, galėsime juos išplėtoti, taip sukurdami specializuotą ontologiją registrams ir informacinėms sistemoms.

2. Registrų apibendrinto integracijos modelio sudėtinės dalys ir taisyklės

Šiame skyriuje aprašomas integracijos modelio kūrimas pasinaudojant anksčiau gautais literatūros analizės rezultatais. Išanalizavus esamas registrų integracines sąsajas ir užsienio praktikas, galima teigti, kad vienas pagrindinių būdų kurti valstybinių sistemų ir registrų apibendrintą integracijos modelį yra kurti XML schemą šiems registrams. Tokios XML schemas kūrimas prasideda nuo tam tikro teorinio taisyklių rinkinio (pvz. ontologijos) sukūrimo.

Pasinaudojant visa surinkta informacija apibrėškime, kas sudarys registrų apibendrintą integracijos modelį. Modeliui sudaryti bus kuriama ontologija, skirta Lietuvos Respublikos registrams, kuri bus išreiškiama per XML schemas, panašiai kaip kuriant XML duomenų standartą. Paėmus bet kurio registro esybių ryšių modelį, jam turėtų būti galima pritaikyti pagal ontologijos taisyklės sukurtą XML schemą ir gauti naują integracinį duomenų modelį, o tuo pačiu ir WSDL schemą integraciniam taškui. Taigi modelio sudedamosios dalys – sukurtos registrų integracinių sąsajų kūrimo taisyklės, bei jas atitinkanti XML schema. (5 pav.)

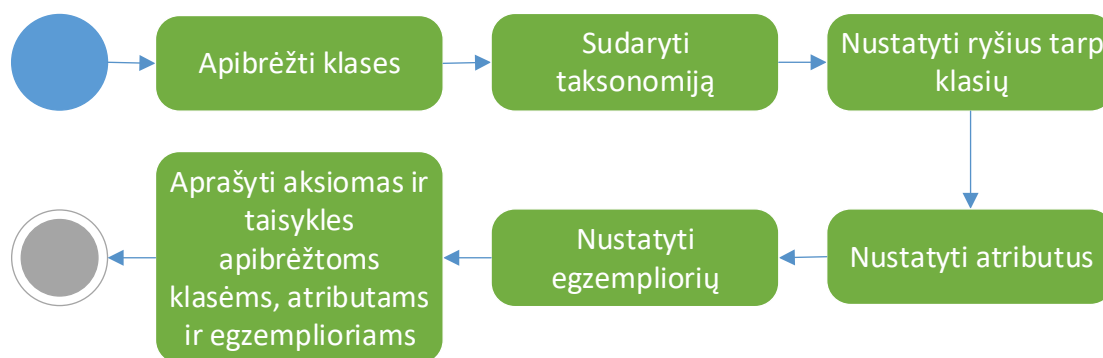
Registrų integracinių sąsajų taisyklės (ontologija) kuriamos pagal išnagrinėtų registrų specifikacijas, bei Lietuvos Respublikos įstatymus reglamentuojančius registrų kūrimą [LRS18]. XML schema sudaroma XML schemų, bei XML duomenų standartų sudarymo principais.



5 pav. Registrų apibendrinto integracijos modelio sudedamosios dalys.

2.1. Registrų integracinių sąsajų ontologija

Šiame skyriuje remiantis ontologijos kūrimo taisyklėmis (1.3.3 punktas) aprašoma registrų integracinių sąsajų ontologija. Ontologijai sukurti apibrėžiamos klasės, sudaroma taksonomija, nustatomi kiti ryšiai. Taip pat nustatoma kaip bus formuojama atributų aibė, aprašomi egzemplioriai. Atlikus šiuos veiksmus aprašomos ontologijos taisyklės ir pateikiamas apibendrinimas. Taisyklės aprašomos atsižvelgiant į registrų specifiką ir jų kūrimo nuostatuose nurodomus ypatumus. Ypatingas dėmesys skiriamas griežtam saugomos objektų ir atributų aibės apibrėžimui, pagal kurį ir yra nustatoma teikiamų duomenų aibė. Veiksmų seka pagal kurią kuriamos ontologijos taisyklės pateikiama (6 pav.)



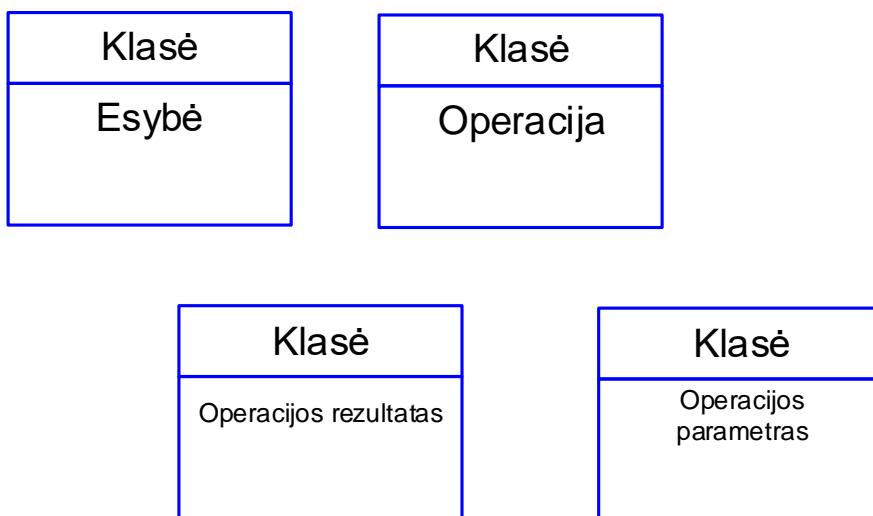
6 pav. Taisyklių rinkinio pagal ontologijos kūrimo taisykles sudarymas.

2.1.1. Klasių apibrėžimas

Pirmiausia apibrėšime pačias stambiausias ir bendriausias klases. Registrų tinklinių sąsajų atveju galime apibrėžti tokias klases (7 pav.):

- Tinklinės sąsajos esybė (toliau Esybė). Esybės klase gali būti apibrėžiamos visos tinklinėje sąsajoje esančios esybės atitinkančios esybes saugomas registruose. Tokios esybės pavyzdys galėtų būti adresas Adresų registre, juridinis asmuo Juridinių asmenų registre ir pan.,
- Tinklinės sąsajos operacija (toliau Operacija). Operacijos klase apibrėžiami visi veiksmai, kuriuos galima atlikti su esybėmis. Kadangi kalbame apie registrus, operacijų sąrašas yra griežtai apibrėžtas (pagal registrus apibrėžiančius įstatymus), o tinklinių sąsajų atveju dažniausiai galimos tik skirtingos duomenų gavimo operacijos. Operacijų pavyzdžiai – Pilnos aibės duomenų gavimas, dalinės aibės duomenų gavimas ir t. t.,
- Tinklinės sąsajos operacijos parametrai (toliau Operacijos parametrai). Šia klase apibrėžiami operacijų parametrai. Pavyzdys galėtų būti parametras asmens kodas siekiant gauti informaciją apie gyventoją iš Gyventojų registro,
- Tinklinės sąsajos operacijos rezultatas (toliau Operacijos rezultatas). Šia klase apibrėžia operacijos su esybėmis rezultata. Rezultatas registrų atvejų paprastai bus

esybių rinkinys, su tam tikrais papildomais metaduomenimis. Pvz. visų Lietuvos gyvenamųjų vietovių sąrašas iš Adresų registro kartu su duomenų atnaujinimo registre, bei duomenų gavimo datomis.



7 pav. Pagrindinės abstrakčios klasės

Apibrėžę keturias abstrakčias pagrindines klases galime detaliau panagrinėti kiekvieną iš jų ir sukurti tam tikras jas atitinkančias klases – poklasius.

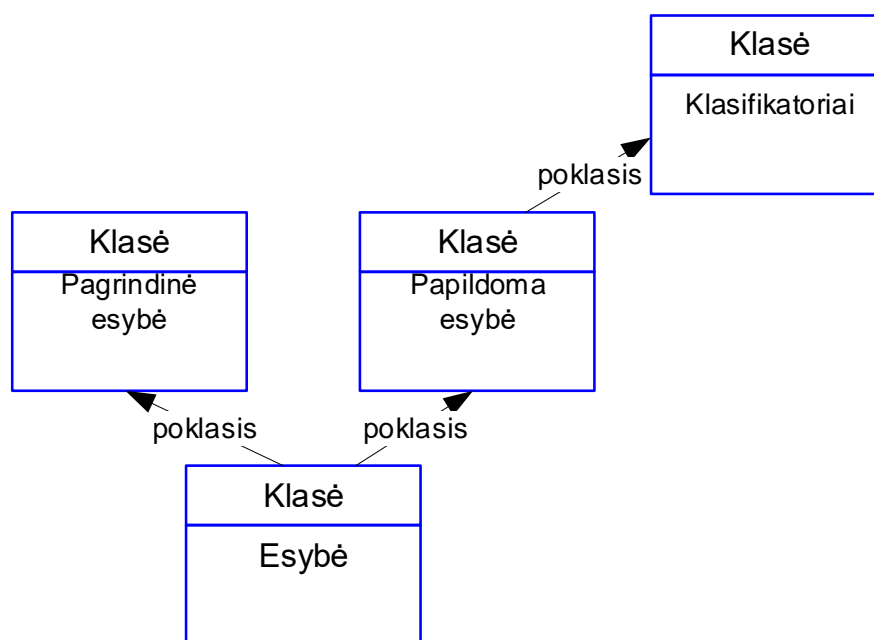
Pirmiausia panagrinėkime esybės klasę. Kiekvienas registras turi tam tikrą savyje saugomų esybių rinkinį. Pagal nagrinėtas registrų specifikacijas galima matyti, kad esybės skiriasi pagal savo pobūdį. Registruose yra išskiriamos pagrindinės esybės, bet egzistuoja ir kitos kurios papildo pagrindines. Taigi, modelio kontekste išskirsime pagrindines ir papildomas esybes. Kaip pavyzdį galime paimti Adresų registrą. Šio registro atveju adresas yra pagrindinė esybė, o administraciniai vienetai, gyvenamosios vietovės ir gatvės papildomos, skirtos apibūdinti adresą. Verta paminėti, kad papildomos esybės nėra atributai, o pilnavertės atskiros esybės registre. Taigi, išskirsime du pagrindinius esybės klasės poklasius:

- Pagrindinė esybė – esybė, kuri vienareikšmiškai apibūdina pagrindinį registre saugojamą objektą ir yra neatskiriama registro dalis. Tokių esybių pavyzdžiai nagrinėtuose registruose:
 - Adresų registras – adresas,
 - Juridinių asmenų registras – juridinis asmuo,
 - Gyventojų registras – fizinis asmuo,
 - Įgaliojimų registras – įgaliojimas,
 - Nekilnojamojo turto registras – nekilnojamasis turtas.

Pastaba. Registre gali būti ir daugiau nei viena pagrindinė esybė. Tai priklauso nuo registre saugojamų objektų.

- Papildoma esybė – esybė, kuri skirta papildyti pagrindinę esybę duomenimis ir yra saugoma kaip atskiras registro objektas. Verta paminėti, kad kai kurie pagrindinių esybių atributai bus papildomos esybės klasės tipo. Papildomų esybių pavyzdžiai nagrinėtuose registruose:
 - Adresų registras – administraciniai vienetai, gyvenamosios vietovės, gatvės, klasifikatoriai, GIS duomenys,
 - Juridinių asmenų registras – juridinio asmens klasifikatoriai, finansinės ataskaitos ir pan.,
 - Gyventojų registras – gyventojų registro klasifikatoriai,
 - Įgaliojimų registras – įgaliotiniai, įgaliojimai, įvairūs klasifikatoriai ir pan.,
 - Nekilnojamojo turto registras – teisių į nekilnojamąjį turtą aprašai, dokumentai, kadastrė saugomų žemėlapių duomenų ištraukos, klasifikatoriai.

Kaip matome visuose registruose kaip papildomos esybės yra įtraukiami klasifikatoriai. Tai rodo, kad klasifikatoriai gali būti kvalifikuojami kaip atskiras papildomų esybių poklasis. Klasifikatoriai turi tam tikrą griežtą struktūrą, kurią galima apibrėžti visiems registrams ir naudoti vieningai, todėl tai yra puikiai apibendrinimui tinkanti esybė. Esybių klasės poklasius galima matyti 8 pav.

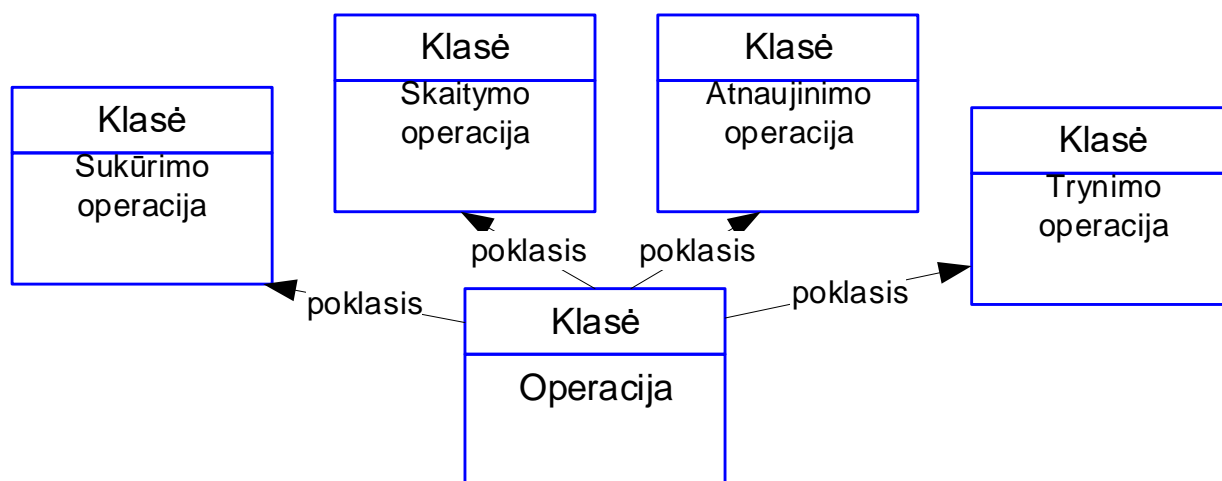


8 pav. Esybių poklasiai.

Toliau nagrinėsime operacijų klasę. Šią klasę tai pat galime padalinti į kelis poklasius priklausomai nuo operacijos tipo. Kadangi kuriame apibendrintą modelį apibrėšime bazinius operacijų tipus pagal bazinės operacijas su duomenimis. Šios bazinės operacijos – tai CRUD (Create, Read, Update, Delete) [Mar83]. Apibrėšime kiekvieną iš operacijų poklasių atskirai:

- Sukūrimo operacijos. Tai tokios operacijos, kurios skirtos sukurti registro esybėms. Pvz. Juridinių asmenų registre galėtų būti operacija skirta juridinių asmenų pridėjimui į registrą. T. y. registravimui.
- Skaitymo operacijos. Tai tokios operacijos, kurios skirtos gauti registro esybes. Pvz. Adresų registre operacija skirta gauti adreso esybei.
- Atnaujinimo operacijos. Tai tokios operacijos, kurios skirtos atnaujinti registro esybėms. Pvz. Gyventojų registre pasikeitė fizinio asmens pavardė ir ją galima pakeisti pasinaudojant duomenų atnaujinimo operacija.
- Trynimo operacijos. Tai tokios operacijos, kurios leidžia ištrinti registro esybę. Šiuo atveju trynimas nebūtinai reiškia visišką duomenų pašalinimą. Dažniausiai bus nustatomas požymis reiškiantis esybės nebegaliojimą. Tokios operacijos pavyzdys – Nekilnojamojo turto paskelbimas nebegaliojančiu (Pvz. nugriautas pastatas) nekilnojamojo turto registre.

Visos šios naujai apibrėžtos klasės sudaro poklasius operacijos klasei. Iš visų minimų poklasių detaliau nagrinėsime tik skaitymo operacijas. Kitas operacijas įtrauksime į modelį ir paliksime abstrakčias. Tai daroma dėl kelių priežasčių – pirmiausia registru esybių kūrimas, atnaujinimas ir trynimas dažniausiai vykdomas iš labai specifinių, registru tvarkyti skirtų sistemų, antra – mus domina registro teikiamų duomenų aibės, o ne jų keitimas. Operacijų klasės poklasiai pateikiami 9 pav.

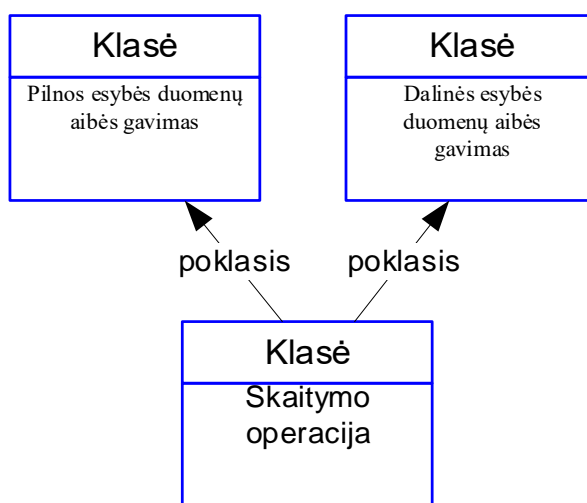


9 pav. Operacijų poklasiai.

Detaliau nagrinėkime skaitymo operacijas. Visų literatūros apžvalgoje nagrinėtų registru sąsajos suteikia galimybę vykdyti skaitymo (duomenų gavimo) operacijas. Iš literatūros analizės rezultatų galime matyti tris duomenų gavimo sąsajų įgyvendinimo būdus – pilnos esybės duomenų aibės gavimas, esybės duomenų aibės skaidymas į poaibius ir šių duomenų teikimas, bei individualių sąsajų kiekvienai naujai sistemai kūrimas. Kaip ir minėta visi šie būdai atskirai turi tam tikrų privalumų ir trūkumų ir reikalingas universalesnis tokių operacijų apibrėžimas. Pagal

iškeltą uždavinį ir problematiką iš karto galime atmesti ir nenaudoti paskutinio duomenų gavimo sąsajos kūrimo būdo – individualių sąsajų įgyvendinimo. Pirmus du gavimo operacijų sudarymo būdus pabandydysime sujungti į vieną taip padengdami dalį reikalavimų kylančių kuriant integracines sąsajas. Taigi, išskirkime tokius skaitymo operacijų poklasisius (10 pav.):

- Pilnos esybės duomenų aibės gavimas – tai skaitymo operacija, kurios rezultatas yra esybė su visais jai priklausančiais atributais,
- Dalinės esybės duomenų aibės gavimas – tai skaitymo operacija, kurios rezultatas yra esybė su daliniu, iš anksto nuspręstu (operacijos parametru ribojamu) atributų kiekiu.



10 pav. Skaitymo operacijos poklasiai.

Operacijas išskyrėme pagal tai kokio dydžio atributų aibę vienai esybei operacija grąžins. Esybių kiekis priklausys nuo užklaustos parametru (ieškoma konkreti esybė ar jų visa aibė ir pan.), bei nuo to ar vienos esybės atributas yra kitos esybės tipo. Pvz. Adreso atveju gaunant konkretų adresą reikia gauti ir administracinį vienetą, ir gatvę, ir gyvenamąją vietovę, nes visos šios papildomos esybės apibūdina adresą ir yra jo dalis.

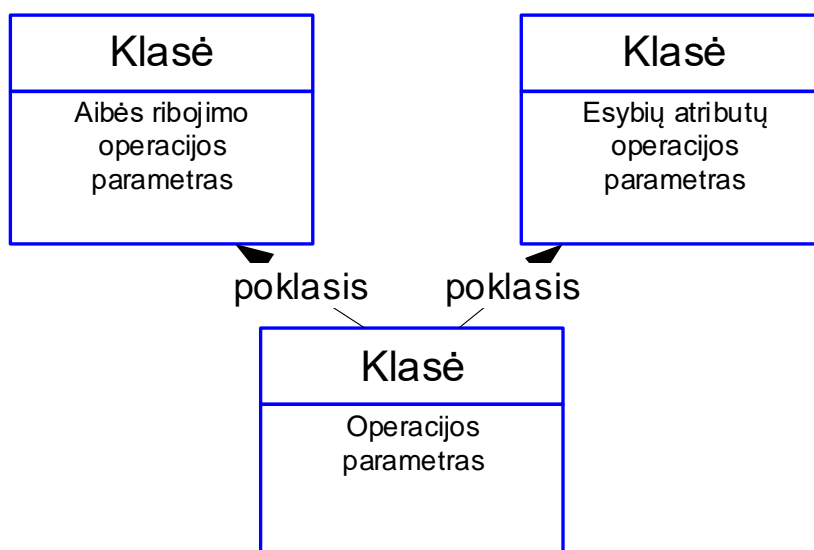
Kita nagrinėjama klasė – operacijos parametrai. Visos aibės duomenų gavimui ši klasė nėra aktuali ir dažniausiai operacijai nereikalinga. Pagrindinė operacijos parametru funkcija yra filtruoti gaunamų duomenų aibę priklausomai nuo poreikio. Taigi, teisingų operacijos parametru apibrėžimas gali būti vienas iš būdų padaryti universalią sąsają, pagal kurią sistema gali gauti ne visus, o tik jai reikalingus duomenis.

Jei detaliau panagrinėsime literatūros apžvalgoje pateikiamas registrų sąsajas pastebėsime, kad vienas iš pagrindinių parametru ten yra registro esybės identifikatorius. To pavyzdys galėtų būti paieška juridinių asmenų registre pagal juridinio asmens kodą, paiešką gyventojų registre pagal asmens kodą ar įgaliojimo paieška pagal įgaliojimo numerį. Taigi, esybės identifikatorius yra vienas iš pačių paprasčiausių ir aiškiausių parametru pagal kurį vienareikšmiškai galima gauti esybę.

Kiti parametrai nagrinėtose sąsajose taip pat susiję su esybių atributais. Pavyzdžiui, įgaliojimo duomenų gavimas, pagal įgaliotojo ir įgaliotinio duomenis. Verta paminėti, kad identifikatorius nors ir atlieka papildomą funkciją, taip pat yra atributas ir čia nebus išskiriamas. Detaliau jis nagrinėjamas prie atributų klasėms apibrėžimo.

Taip pat yra vienas specifinis parametras, kuris nurodomas nagrinėtose sąsajose. Tai gaunamų duomenų detalumą apibūdinantis parametras. Tokio parametro pavyzdys galėtų būti juridinių asmenų išrašo tipo nurodymas. Kaip ir minėta anksčiau gali būti trumpasis, pagrindinis ir išplėstinis išrašas. Nurodžius konkretų tipą prie operacijos parametro gaunamas skirtingas esybės atributų skaičius. Kaip ir esybių atributų operacijos parametrus, taip ir šį parametą išskirsime. Pavadinsime jį aibės ribojimo operacijos parametru.

Taigi, išskirkime du operacijos parametų poklasius – esybių atributų operacijos parametrai ir aibės ribojimo operacijos parametrai. Jų schema pavaizduota 11 pav.

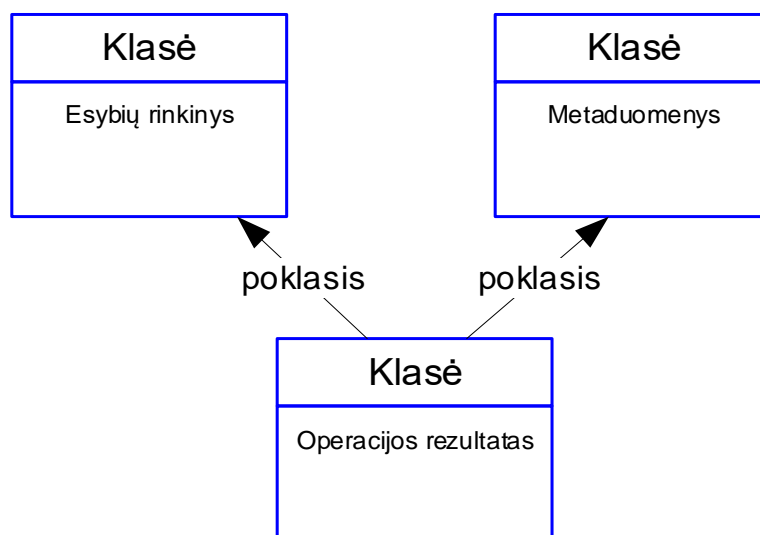


11 pav. Operacijų tipų poklasiai.

Kita nagrinėjama klasė – operacijos rezultatas. Operacijos rezultatas yra glaudžiai susijęs su operacijos tipu. Sukūrimo, atnaujinimo ir trynimo operacijoms rezultatas bus daugiau mažiau vienodas ir greičiausiai tai bus patvirtinimas, kad operacija atlikta sėkmingai arba išvestas klaidos pranešimas. Daug plačiau galima nagrinėti operacijos rezultatą skaitymo (kitaip duomenų gavimo) operacijoms. Duomenų gavimo operacijų rezultatas yra registro esybių ar skirtingos aibės jų atributų rinkinys. Pvz. Adresų registro atveju administracinių vienetų skaitymo operacijos rezultatas yra administracinių vienetų esybės su atitinkamais atributais, kurie parinkti gražinti per sąsają. Taip pat operacijose gali būti teikiami metaduomenys, tokie kaip duomenų gavimo data, paskutinė duomenų atnaujinimo data, informacija apie pavykusią operaciją (kaip jau minėta prie sukūrimo, atnaujinimo ir trynimo operacijų) ir panašiai. Taigi, galime išskirti du operacijos rezultatų poklasius:

- Esybių rinkinys – tai operacijos rezultatas susidedantis iš esybių ir tam tikros jų atributų aibės.
- Metaduomenys – tai operacijos rezultatas skirtas apibūdinti pačios operacijos parametrus. Pvz. Operacijos sėkmės patvirtinimas ar įvykusių klaidų pranešimas.

Apibrėžtus operacijos rezultato poklasius galime matyti 12 pav.



12 pav. Operacijos rezultato poklasiai.

Šiame skyriuje buvo sukurtos pagrindinės tinklinės sąsajos klasės, kurios bus įrašomos į ontologiją. Taip buvo atliktas pirmasis ontologijos kūrimo žingsnis. Norint padaryti sąsają dar labiau pritaikytą ir universalesnę šios apibendrintos klasės vėliau bus detalizuojamos ir pritaikomos konkrečioms registrams.

2.1.2. Taksonomijos sudarymas

Antras žingsnis ontologijų, o tuo pačių ir apibendrinto integracijos modelio kūrimo yra taksonomijos sudarymas. Klasių apibrėžimo skyriuje jau nustatyti klasių poklasiai. Juos nagrinėjant reikia peržvelgti ryšius ir nustatyti jų tipus pagal taksonomijos kūrimo taisykles. Peržvelgsime kiekvieną iš poklasių ryšių ir nustatysime jam tipą. Tada peržiūrėsime bendrą visos aibės vaizdą ir išanalizuosime ar tarp skirtingų pagrindinių klasių ar jų poklasių yra hierarchinių ryšių.

Pirmiausia nagrinėsime esybių klasę ir poklasius. Esybė turi du poklasius – pagrindinės ir papildomos esybės. Šios klasės pilnai padengia visą esybių klasę ir viena esybė negali būti ir pagrindinė ir papildoma. Tai reiškia, kad šie poklasiai taksonomijoje turi skaidymo ryšį su esybės klase. Papildomai esybei yra apibrėžtas vienas poklasis – klasifikatoriai. Šis poklasis tikrai nepadengia visų papildomų esybių (pvz. gatvė yra papildoma esybė adresų registre, tačiau tai nėra klasifikatorius) ir taip pat nebus tokių papildomų esybių, kurios bus ir klasifikatorius ir kitokia papildoma esybė. Tai reiškia, kad papildomos esybės klasė su esybės klase yra susijusi suskaidymo dekomponavimo ryšiu.

Toliau nagrinėkime operacijos klasę. Ši klasė turi sukūrimo, atnaujinimo, skaitymo ir trynimo operacijų poklasius. Nagrinėdami šiuos poklasius teigsime, kad tai visos registrų sąsajose galimos operacijos, nes kuriama ontologija apibūdina kokias operacijas ji gali apibrėžti ir kitokių operacijų naudojantis ja apibrėžti nepavyks. Taip pat, poklasiai galėtų turėti bendrų egzempliorių. Pavyzdžiui, operacija, kuri atnaujina duomenis kurdama naują esybės objektą, o seną ištrindama (nustatydamą tinkamą ištrynimo datą prie esybės). Tokio tipo operacija padengia net tris poklasius – atnaujinimo, sukūrimo ir trynimo. Pagal pastebėtus galimus ryšius ir 1.3.3 skyriuje pateiktą taksonomijos hierarchinių ryšių klasifikaciją galime teigti, kad operacijos poklasiai sudaro išsamaus dekomponavimo ryšį su operacijos klase.

Nagrinėdami operacijos klasę detaliau apibrėžėme ir skaitymo operacijos poklasius. Nutarta, kad bus du skaitymo operacijų poklasiai – pilnos esybės duomenų aibės gavimas, bei dalinės esybės duomenų aibės gavimas. Abu šie poklasiai tikrai nepadengia visų galimų skaitymo operacijų (pvz. egzistuoja skaitymo operacijos, kur perduodama kelių esybių dalinės atributų aibės sujungtos į naują esybę ir pan.). Šie poklasiai negali turėti bendrų egzempliorių, dėl to, kad aiškiai apibrėžiamas perduodamas aibės dydis – vienur jį yra tik pilna, o kitur tik dalinė. Ir pilna, ir dalinė aibė vienu metu būti negali. Taigi, skaitymo operacijos poklasiai su skaitymo operacija sudaro suskaidymo dekomponavimo ryšį.

Kita nagrinėjama hierarchija priklauso operacijos parametrų klasei. Ši klasė suskaidyta į aibės ribojimo operacijų parametrų ir esybių atributų operacijos parametrų poklasius. Abu šie poklasiai yra pakankamai platūs ir registrų integracinių sąsajų kūrimo kontekste tarsime, kad kitokių parametrų būti negali ir jie pilnai apibrėžia visą operacijos parametrų aibę. Taip pat tarsime, kad aibės ribojimo operacijos parametras negali būti esybės atributas – t. y. jis apibūdina tik aibę, o ne pačią esybę. Iš šių prielaidų galime spręsti, kad operacijos parametrų klasė su savo poklasiais susijusi skaidymo ryšiu.

Paskutinė nagrinėjama klasė – operacijos rezultatai. Jie taip pat skaidomi į du poklasius – esybių rinkinį ir metaduomenis. Kadangi metaduomenimis mes laikome visą kitą gražinamą informaciją, kuri nėra esybės atributas, tai šie du poklasiai pilnai padengia visą operacijos rezultatų aibę ir negali turėti bendrų egzempliorių. Tai reiškia, kad operacijos rezultatų poklasiai su operacijos rezultatų klase yra susiję skaidymo ryšiu.

Apibrėžėme visus hierarchinius ryšius tarp skirtingų klasių ir poklasių ir sudarėme kuriamos ontologijos taksonomiją. Bendrą šių ryšių schemą galime matyti 1 Priedase. Schemoje pavaizduotos visos klasės ir poklasiai, bei vizualiai pateikti visi hierarchiniai ryšiai. Šiame išskaidymo lygmenyje daugiau klasių ir poklasių nėra, todėl ir hierarchinių ryšių aprašyti daugiau nepavyks. Tarp susijusių klasių taip pat egzistuoja kiti, ne hierarchiniai ryšiai (pvz. esybės klasės ir operacijos rezultato poklasio esybių rinkinio). Jie bus nustatomi kitame skyriuje.

2.1.3. Ryšių nustatymas

Sudarant taksonomiją buvo sudaryti hierarchiniai ryšiai tarp klasių ir jų poklasių. Šiame žingsnyje bus sudaromi kitokio tipo ryšiai. Pereisime per visas gautas klases ir išsiaiškinsime kokius ryšius jos sudaro vienos su kitomis. Tada gilinsimės į gautus poklasius (smulkesnes klases) ir žiūrėsime ar jie turi kokių nors kitokių ryšių su kitomis klasėmis.

Pirmiausia pradėsime nuo pačios operacijos, kuri vykdoma per tinklinę sąsają. Egzistuoja akivaizdus loginis ryšys tarp operacijos bei jos parametrų ir rezultato. Pagal mūsų kuriamą logiką operacija gali turėti daug įeinančių parametrų ir iš jos išeinančių rezultatų. Taigi, galime sudaryti tokius ryšius:

- Operacijos parametrai *pateikiamas* operacijai – šio ryšio kardinalumas yra daug su vienu t. y. vienai operacija gali būti pateikiama daug parametrų,
- Operacija *grąžina* operacijos rezultatus – šio ryšio kardinalumas yra vienas su daug t. y. viena operacija gali grąžinti daug rezultatų.

Esybės klasė tiesioginio sąryšio su pagrindinėmis operacijos, operacijos parametro, bei operacijos rezultato klasėmis neturi. Taip yra dėl to, kad operacijos parametras ar jos rezultatas gali būti nebūtinai pati esybė, o pvz. metaduomenys. Todėl detaliau esybių nehierarchinius sąryšius nagrinėsime apžvelgdami operacijos parametrų, bei rezultatų poklasius.

Išnaginėjus pagrindines klases, toliau nagrinėsime poklasius. Pirmiausia nagrinėkime operacijos parametro poklasius. Aibės ribojimo operacijos parametrai neturi jokių kitokių nei hierarchinių sąryšių su kitomis klasėmis. Esybių atributų operacijos parametras yra susijęs su esybėmis. Taigi, esybės šiai klasei pateikia atributus ir dėl to galima kurti sąryšį tarp esybių ir esybių atributų operacijos parametro klasių.

Toliau nagrinėkime operacijos rezultato poklasius. Esybių rinkinio klasė turi turėti sąsają su esybės klase. Esybės klasė kaip ir esybių atributų operacijos parametro atveju pateikia atributų rinkinį, bei pačias esybes. Taigi, galime aprašyti dar vieną nehierarchinį ryšį tarp esybės ir esybių rinkinio klasių.

Peržvelgus visas kitas klases, kurios yra pagrindinių klasių poklasiai daugiau aiškių nehierarchinių ryšių nematyti. Taip yra dėl to, kad mūsų kuriamoje ontologijoje sąryšiai yra sudaromi per operaciją ir jo parametrus ir rezultatus. Tiesioginio sąryšio esybei su operacija nenumatyta. Taip pasirinkta dėl to, kad operacija nebūtų pririšta vien tik prie esybių ir galėtų turėti papildomus, su esybėmis nesusijusius, parametrus ir rezultatus. Visi kiti ryšiai tik nurodo hierarchiją ir operacijų arba esybių tipus.

Šiame skyriuje buvo apžvelgti ir sukurti ontologijos nehierarchiniai ryšiai. Šių ryšių diagrama kartu su anksčiau gauta taksonomija pateikiama 2 Priedase. Sukūrus šiuos ryšius galime pereiti prie atributų klasifikavimo ir nustatymo.

2.1.4. Atributų aibės formavimas

Vienas iš ontologijos kūrimo žingsnių yra aprašyti atributus, kurie priklauso ontologijoje aprašytoms klasėms. Pirmiausia aprašysime atributus pagrindinėms klasėms – esybėms, operacijoms, operacijų parametrams ir operacijų rezultatams. Šiuos atributus gali paveldėti ir poklasiai. Taip pat apibrėšime atributus ir jiems.

Viena svarbiausių užduočių kuriant registrų integracinių sąsajų ontologiją yra aprašyti integracinių sąsajų esybių atributų aibės klasifikavimą. Norint sukurti universalų atributų aibės formavimą reikia suskirstyti atributus į tam tikras grupes, kurias būtų galima pritaikyti konkrečioms registrams. Žinant kokias grupes priklauso atributas būtų galima lengvai suformuoti duomenų aibes, kurios yra perduodamos per integracinę sąsają. Tai leistų pasiekti tam tikrą universalumą ir anksčiau aprašytą perduodamų atributų aibės skirstymą į grupes – pilnas, dalinis ir pan. Pirmiausia išskirstykime esybių atributų aibę į dvi dalis:

- Pagrindiniai atributai – tai esybės atributai, kurie apibūdina esybę ir be jų esybės egzempliorius negalėtų egzistuoti. Pavyzdžiui, adresų registre adreso esybės pagrindiniai atributai yra administracinis vienetas, gyvenvietė, gatvė, namo numeris ir pan. Be jų vienareikšmiškai adreso negalima apibrėžti. Egzistuoja variantų kada vienas ar keli pagrindiniai atributai gali būti nebūtini (adresas neturi gatvės ir namo numerio, o tik gyvenvietę), tačiau jie vis tiek priskiriami pagrindiniams, o jų pateikimas per sąsają priklauso nuo atskiro registro taisyklių,
- Papildomi atributai – tai esybės atributai, kurie apibūdina esybę, tačiau esybės egzempliorius gali egzistuoti ir be jų. Pvz. juridinių asmenų registre gali būti nurodytas juridinio asmens kontaktinis telefonas, tačiau tai nėra privalomas duomuo ir juridinis asmuo gali būti pilnai apibūdinamas ir be to.

Atskirti, kuris atributas yra pagrindinis, o kuris papildomas leidžia registro specifiką ir registre saugomi objektai nurodomi registrų nuostatuose. Taip pat nagrinėjant registrus apibrėžiančius nuostatus ir įstatymus (1 skyrius) buvo pastebėta, kad registre saugomų objektų atributai jau yra skirstomi į pagrindinius ir papildomus (Pastaba. Tai gali būti apibrėžiama kitomis sąvokomis, kurios reiškia tą patį).

Norėdami labiau detalizuoti registrų integraciniams sąsajoms svarbius esybių atributus, panagrinėkime kiekvieną iš atributų grupių detaliau. Pirmiausia nagrinėkime pagrindinius atributus. Čia iš karto reiktų išskirti vieną labai svarbų ir kiekvienai esybei (pagal esybių ryšių modelį aprašomą 1.2.2 punkte) privalomą pagrindinį atributą – esybės identifikatorių. Esybės identifikatorius – tai unikalus esybės atributas, kuris vienareikšmiškai apibrėžia esybę. Tokio identifikatoriaus pavyzdys galėtų būti asmens kodas gyventojų registre. Dažnai siekiant išvengti pasikartojimų ir užtikrinti visišką unikalumą, registras suteikia savą esybės identifikatorių, kuris

neturi jokių reikšminių atitikmenų realiame pasaulyje. Kiti pagrindiniai atributai kaip jau minėta priklauso nuo registro specifikos ir daugiau bendrumų rasti yra sudėtinga.

Toliau panagrinėkime papildomus atributus. Kadangi jie kaip ir daugelis pagrindinių priklauso nuo konkretaus registro, čia reikia nagrinėti jų išskyrimą į kelias grupes. Pavyzdžiui, jau nagrinėtame juridinių asmenų registre perduodami išrašai yra išskiriami į tris grupes (1.1.1.3 punkte). Tai reiškia, kad tarp jų egzistuoja ir trys atributų grupės. Viena iš jų yra pagrindinė, o kitos perduodamos kaip papildomos. T. y. trumpajame išrašė perduodami tik pagrindiniai atributai, pagrindiniame – pagrindiniai atributai ir papildomi atributai, o išplėstiniame – pagrindiniai atributai ir dvi papildomų atributų grupės. Norėdami pritaikyti panašią logiką apibrėšime papildomų atributų grupes:

- Pirminiai papildomi atributai – papildomi atributai, kurie apibūdina papildomus esybės duomenis, kurie yra skirti tiesiogiai apibūdinti esybę ir dažniausiai turi realius atitikmenis realiame pasaulyje. Pvz. juridinio asmens kontaktinis telefonas, priešingai, nei juridinio asmens pavadinimas yra realus ne tik su registru susijusiems duomenims tvarkyti reikalingas atributas,
- Išplėstiniai papildomi atributai – papildomi atributai, kurie apibūdina papildomus esybės duomenis, tokius kaip registravimo duomenys, pagrindiniams atributams nepriklausančios papildomos nuorodos į esybes, GIS duomenys, kur GIS esybė nėra pagrindinė (pvz. adresų registre) ir pan.

Šių dviejų minėtų grupių atskyrimas taip pat yra uždavinys, kuris yra specifinis kiekvienam registru. Remiantis nagrinėtomis registrų specifikacijomis ir nuostatais galime matyti, kad panašus išskyrimas jau egzistuoja registruose, todėl šį uždavinį išspręsti yra lengviau. Turint aukščiau pateiktą šių atributų apibūdinimą šį uždavinį spręsti yra lengviau, o išskirsčius į tokias grupes jau gaunama ir bendra struktūra, kuri bus bendra visiems šių ontologijų, bei sukurta modelį naudojantiems registrams.

Nustatant esybės klasės atributus buvo išskirtos trys pagrindinės jų grupės – pagrindiniai, pirminiai papildomi ir išplėstiniai atributai. Visos šios grupės buvo apibūdintos ir vėliau bus panaudotos analizuojant ir sudarinėjant atributų aibes kitoms klasėms.

Taip pat nustatysime atributus specifiniam papildomų esybių poklasiui – klasifikatoriams. Paprastuoju atveju klasifikatoriaus objektas susidės iš kodo ir reikšmės atributų. Specialus atvejis galėtų būti, kai klasifikatoriai būtų hierarchiniai ir reiktų nustatyti sąryšius. Dėl šios priežasties prie kodo ir reikšmės atributų klasifikatoriams pridėsime nuorodą į tėvinį klasifikatorių (tėvinio klasifikatoriaus objekto kodą). Jei šis atributas neužpildytas reiškia klasifikatoriaus objektas neturi hierarchijos arba yra šakninis.

Toliau nagrinėsime operacijos klasę ir jos atributus. Išskirkime pagrindinius atributus, kurie operacijos klasės objektuose būtinai turi būti nurodomi:

- Pavadinimas – tai yra reikšminis operacijos identifikatorius, kuris nurodo operacijos tikslą ir dažniausiai susideda iš veiksmažodžio, kuris apibūdina operacijos tipą (pagal poklasių žr. 2.1.1 punktą), bei daiktavardžio, kuris apibūdins gaunamas esybes.
- Tikslas – pilnas aprašymas ką daro operacija. Metaduomuo skirtas pilnai apibūdinti operaciją.
- Operacijos parametrai – nuorodos į operacijos parametrų klases,
- Operacijos rezultatai – nuorodos į operacijos rezultatų klases.

Laikysime, kad išskirti atributai yra pakankami apibrėžti operaciją registru integracinių sąsajų ontologijoje.

Nagrinėkime operacijos parametrus ir operacijos rezultatus. Šios abi klasės apibūdina įeities ir išeities duomenis operacijoms. Pats duomuo yra vienas iš atributų. Jam apibūdinti reikalingas pavadinimas. Dėl šios priežasties operacijos parametro ir operacijos rezultato klasėms suteiksime po du atributus – pavadinimas ir reikšmė. Reikšmė šiuo atveju gali būti įvairi. Parametro atveju tai arba aibę ribojanti reikšmė, arba esybės atributas, o rezultato atveju arba esybių rinkinio reikšmė arba Metaduomuo. Verta pastebėti, kad šis suskirstymas taip pat nurodo, kad tiek operacijos parametro, tiek operacijos rezultato klasės yra susiję ir būtų galima sukurti jų tėvinę klasę – pavadinimo-reikšmės pora.

Toliau detalčiau panagrinėkime esybės atributų grupių sąsajas su operacijos parametrais ir rezultatais. Viena iš aibės ribojimo operacijos parametro reikšmių galėtų būti aibės dydį nusakantis klasifikatorius. Aibės dydį galima nusakyti pagal esybės atributus – pagrindinis, pirminis ir išplėstinis. Nusakius šį aibės dydį gaunamas atitinkamas operacijos rezultatas, todėl galime sakyti kad esybių rinkinio operacijos rezultato reikšmė gali būti arba tik esybė su pagrindiniais atributais, esybės su pagrindiniais ir pirminiais papildomais atributais arba esybė su pagrindiniais, pirminiais papildomais ir išplėstiniais atributais (atitinka pilną esybės atributų aibę). Taip pat atskiri esybės atributai, nepriklausomai nuo jų grupės, galėtų būti aibės ribojimo operacijos parametro reikšmės. Taigi, taip nustatome sąsają tarp esybių atributų ir operacijos parametrų bei rezultatų ir realiai sudarome pagrindinę skirtingų atributų aibių gavimo logiką.

Apibrėžiant atributus klasėms buvo aprašyti konkretūs klasėms privalomi atributai ir atributų grupės į kuriuos bus skirstomi specifiniai registruose egzistuojantys esybių atributai. Atributų grupių sudarymas leido sudaryti taisykles pagal kurias konstruojamos operacijos ir kuriamos apibendrintos registru sąsajos skirtos informacinėms sistemoms. Aprašius atributus galime prieiti prie ontologijos egzempliorių išskyrimo bei aksiomų ir taisyklių nustatymo.

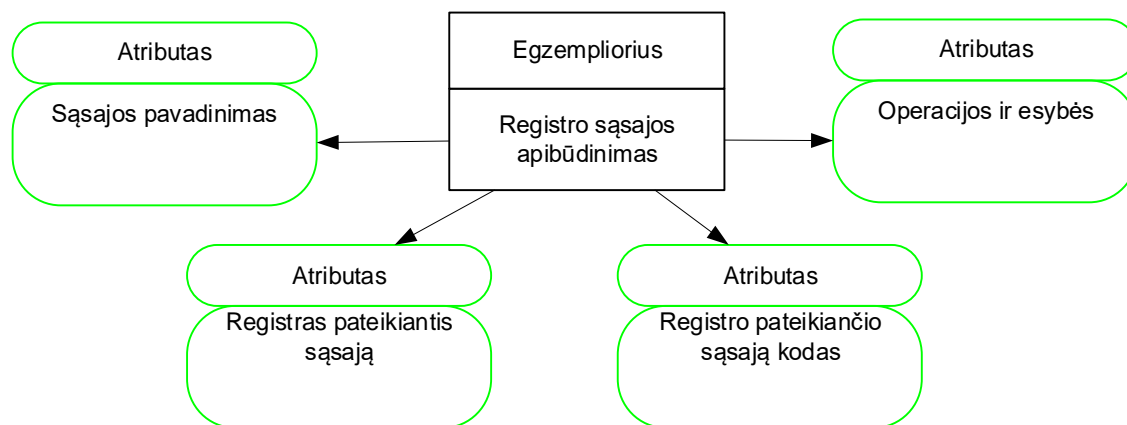
2.1.5. Egzempliorių nustatymas

Šiame žingsnyje bus nustatyti egzemplioriai, kurie gali būti nebūtinai kažkurios iš klasių tipo, tačiau yra bendri visoms pagal integracinį modelį kuriamoms sąsajoms. Darome prielaidą, kad ontologijoje egzistuoja vienas egzempliorius, kuris nepriklauso nuo kitų klasių ir bus skirtas pačios tinklinės sąsajos apibrėžimui ir aprašymui. Taip pasirinkta dėl to, kad vienetinių bendrų objektų, kurie turėtų tą pačią reikšmę visų registrų logikoje beveik nėra, dėl skirtingos registrų specifikos. Pavyzdžiui, jau anksčiau minėtas seimo pastato adresas adresų registre gali būti aiškus ir registre saugomus objektus atitinkantis egzempliorius, tačiau juridinių asmenų registre jis arba netektų prasmės arba ją keistų iš esmės (Tai galėtų būti tik atributas nurodyti, kažkurio iš juridinių asmenų adresui).

Taigi, išskirsime egzempliorių, kuris apibūdins pačią registro sąsają. Toks egzempliorius reikalingas, kad būtų galima aiškiai identifikuoti skirtingų registrų sąsajas. Šį egzempliorių vadinsime registro sąsajos apibūdinimu ir jam suteiksime tokius atributus:

- Sąsajos pavadinimas – laisva forma pasirenkamas sąsajos pavadinimas, skirtas sąsajos funkciją suprasti žmogui,
- Registras pateikiantis sąsają – pilnas registro pateikiančio sąsają pavadinimas,
- Registro pateikiančio sąsają kodas – registro teikiančio sąsają kodas. Šį atributą laikysime neprivalomu, tačiau norint ateityje sukurti anksčiau minėtą automatinę (žr. 1.2.2 skyrius ir [Med05]) tinklinių sąsajų paiešką būtina sudaryti tokių kodų duomenų bazę ir kiekvieną naują registrą registruoti joje.
- Teikiamos operacijos ir joje dalyvaujančios esybės – tai yra metaduomenys nusakantys sąryšį tarp teikiamų operacijų ir esybių, kurios dalyvauja tose operacijose. Vėlgi, norint sukurti automatinę tinklinių sąsajų operacijų paiešką reikalingas tiek operacijų tipų, tiek esybių kodavimas ir šių kodų duomenų bazių sudarymas.

Registro sąsajos apibūdinimui apibrėžime pagrindinę minimalią atributų aibę (13 pav.). Priklausomai nuo registro poreikio ši aibė gali skirtis, tačiau papildomi atributai dažniausiai bus susiję su vidinėmis sąsajos savybėmis arba mūsų nenagrinėjamomis temomis, pavyzdžiui, sąsajos saugumo užtikrinimu.



13 pav. Registro sąsajos apibūdinimo egzempliorius.

Apibrėžus registro sąsajos apibūdinimo egzempliorių jau turime visus reikalingus bazinius ontologijos elementus. Norėdami sustiprinti ontologiją turėsime nustatyti aksiomas ir tam tikras ontologijos taisykles (funkcijas). Tokiu būdu turėsime griežtesnę ontologijos struktūrą ir taip pat dar bendresnį registrų integracinių sąsajų kūrimo modelį.

2.1.6. Ontologijos taisyklių nustatymas

Šiame skyriuje apibrėšime registrų integracinių sąsajų ontologijos taisykles, kurios turės galioti visoms pagal ontologiją kuriamoms sąsajoms. Registrų integracinių sąsajų ontologijos taisykles – tai aksiomų ir funkcijų rinkinys nusakantis kaip turi būti aprašoma sąsaja ir kaip turi būti panaudojamos apibrėžtos klasės.

Pirmiausia apibrėšime paprasčiausias aksiomas, kurios nusako klasių privalomumus ir ribojimus:

- Turi egzistuoti bent viena esybė,
- Turi egzistuoti bent viena operacija,
- Kiekvienai operacijai turi egzistuoti rezultatas,
- Visi privalomi atributai turi būti užpildyti,
- Turi egzistuoti vienas ir tik vienas registro sąsajos apibūdinimo egzempliorius,
- Operacijos gali turėti esybių atributų operacijos parametrus,
- Negali būti dviejų tokių pačių esybių,
- Negali būti dviejų tokių pačių operacijų,
- Negali būti dviejų tokių pačių parametrų vienai operacijai (Pvz. aibės ribojimo parametras negali būti nurodytas du kartus tame pačiame operacijos kvietime),
- Negali būti dviejų tokių pačių rezultatų vienai operacijai (Pvz. tie patys du esybių rinkiniai negali būti gražinami kaip vienos operacijos rezultatas),
- Kiekvienai klasei turi egzistuoti bent vienas atributas.

Toliau, aprašysime taisyklę skirtą išskirti pagrindinį registru integraciniame modelyje naudojamą duomenų gavimo operacijų tipą. Ankstesniuose skyriuose apibrėžėme vieno tipo aibės ribojimo operacijų parametro reikšmes ir jas susiejome su esybės atributų grupėmis. Norint, kad šis susiejimas būtų panaudojamas visose pagal modelį kuriamose sąsajose, reikia formaliai apibrėžti tokios operacijos būtinumą. Apibrėžkime taisyklę - Turi egzistuoti tokia operacija, kuri:

- turi privalomą aibės ribojimo parametą galintį turėti tokias reikšmes: pagrindinis, pirminis ir išplėstinis,
- turi nebūtiną registro esybės identifikatoriaus parametą,
- rezultate pateikia esybę arba esybes, priklausomai nuo to ar buvo nurodytas konkrečios esybės identifikatorius,
- rezultate pateikia esybės ar esybių atributų aibę priklausomai nuo aibės ribojimo parametro pagal tokią logiką:
 - jei aibės ribojimo parametro reikšmė yra pagrindinis, tai pateikiami tik pagrindiniai atributai,
 - jei aibės ribojimo parametro reikšmė yra pirminis, tai pateikiami pagrindiniai ir pirminiai papildomi atributai,
 - jei aibės ribojimo parametro reikšmė yra išplėstinis, tai pateikiami pagrindiniai, pirminiai papildomi ir išplėstiniai atributai,
- rezultate pateikia tik vieną esybės tipą (Pvz. adresų registro atveju konkretų adresą), o norint gauti kito tipo esybę turi būti kviečiama kita operacija skirta tai esybei.

Sprendžiant iš paskutinio punkto apibrėžiančio operaciją turime sugriežtinti taisyklę ir teigti, kad kiekvienai registro esybei (išskyrus klasifikatoriaus tipo) turėtų egzistuoti tokia operacija. Taip pat čia reikia apibrėžti, kad pilnos esybės duomenų aibės gavimo operacijų aibės ribojimo parametro reikšmė yra „išplėstinis“, o dalinės esybės duomenų aibės gavimo operacijų – „pagrindinis“ ir „pirminis“.

Klasifikatoriaus tipo esybių gavimo operacija apibrėžiama atskirai. Klasifikatoriaus gavimo operacija:

- turi neprivalomą klasifikatoriaus objekto kodo (esybės identifikatorius) operacijos parametą,
- turi neprivalomą tėvinio klasifikatoriaus objekto kodo parametą, jei klasifikatorius hierarchinis,
- rezultate pateikia klasifikatoriaus esybę arba esybes priklausomai nuo pateiktų parametų,
- rezultate pateikta klasifikatoriaus esybė ar esybės turi pilną savo atributų aibę – kodą, reikšmę, tėvinio klasifikatoriaus objekto kodą.

Apibrėžėme pagrindines taisykles, kurios skirtos ontologijai ir sąsajos pagal ją aprašymui. Taisyklių rinkinys gali būti plečiamas bei tobulinamas, tačiau čia buvo aprašytos pagrindinės taisyklės, kurios susijusios su mūsų sprendžiamu atributų aibės, bei sąsajų operacijų kiekio nustatymu. Iš taisyklių galime matyti, kad visoms esybėms bus sukurta po vieną operaciją, o atributai bus pateikiami priklausomai nuo aibės ribojimo parametro. Įgyvendinus tokius ribojimus ir taisykles registrams turėsime apibrėžtas gražinamų duomenų aibes ir galėsime apibrėžti sąsajas.

2.1.7. Apibendrinimas

Aprašius klases, taksonomiją, ryšius, atributus, egzempliorius, bei ontologijos taisykles remiantis 1.3.3 punkte apibrėžtais ontologijos sudarymo žingsniais buvo gauta pilna ontologija. Apibrėžta registrų integracinių sąsajų ontologija leidžia sukurti XML schemą, pagal kurią būtų galima sudarinėti konkrečių registrų sąsajų XML schemas. Aprašytos ontologijos taisyklės ir sukurta XML schema sudarys modelį, pagal kurį turėtų būti sudaromos integracinės sąsajos registras. Tolesniuose skyriuose ši ontologija bus pritaikoma jau anksčiau minėtos XML schemas sudarymui, bei modelio pritaikymui konkrečiam pavyzdžiui.

2.2. XML schema sąsajų sudarymui

Remiantis 1.3.2 skyriuje apibrėžta XML duomenų standartų, bei sukurta registrų integracinių sąsajų ontologija apibrėšime XML schemą, kuri atspindės ontologijos praktinį įgyvendinimą. Pasinaudojant sudaryta schema bus galima sukurti kitas konkrečių registrų sąsajų schemas, pagal kurias bus galima sukurti sąsajų WSDL arba kokius nors kitus aprašus, priklausomai nuo naudojamos technologijos. Jeigu kalbėsime tik apie tinklines sąsajas, sukūrus konkrečių registrų sąsajų schemų duomenų bazę, galėsime atlikti automatinę šių sąsajų paiešką. Šios duomenų bazės, kaip ir anksčiau minėtų sąsajų, bei operacijų kodų duomenų bazių sudarymas nėra šio darbo uždavinys, tačiau sukurta XML schema suteikia galimybę šią užduotį atlikti naudojantis tam tikromis aiškiais taisyklėmis.

Kurdami schemą visus atributus dėsime į šakninį elementą, kurį pavadinsime *RegistroSasaja*. Pirmiausia pradėsime nuo esybių aprašymo. Kadangi esybės yra skaidomos į pagrindines ir papildomas, schemoje sukursime du elementus – *PagrindineEsybe* ir *PapildomaEsybe*, kurie bus viduje tėvinio elemento *Esybe*. Taip pat sukursime dar vieną elementą *Klasifikatorius* tame pačiame lygmenyje, nes jis yra papildomos esybės poklasis suskaidymo dekomponavimo būdu ir jo negalime laikyti vidine papildomos esybės dalimi (nėra padengiama visa Papildomos esybės klasė). Visų esybių tipų (išskyrus klasifikatorius) atributus suskirstysime į *PagrindinisAtributas*, *PagrindinisPapildomasAtributas* ir *IsplestinisPapildomasAtributas*. Visų šių tipų atributai bus skirti tik išskyrimui, o viduje bus laikoma ta pati struktūra: *Pavadinimas* ir *Reiksme*. Klasifikatorių atveju sukurta tokia vidinė struktūra: *Kodas*, *Reikšmė* ir *TevinioObjektoKodas*. Visoms esybės sukurtas atributas *Pavadinimas*, kuris nurodo kaip pavadinta esybė.

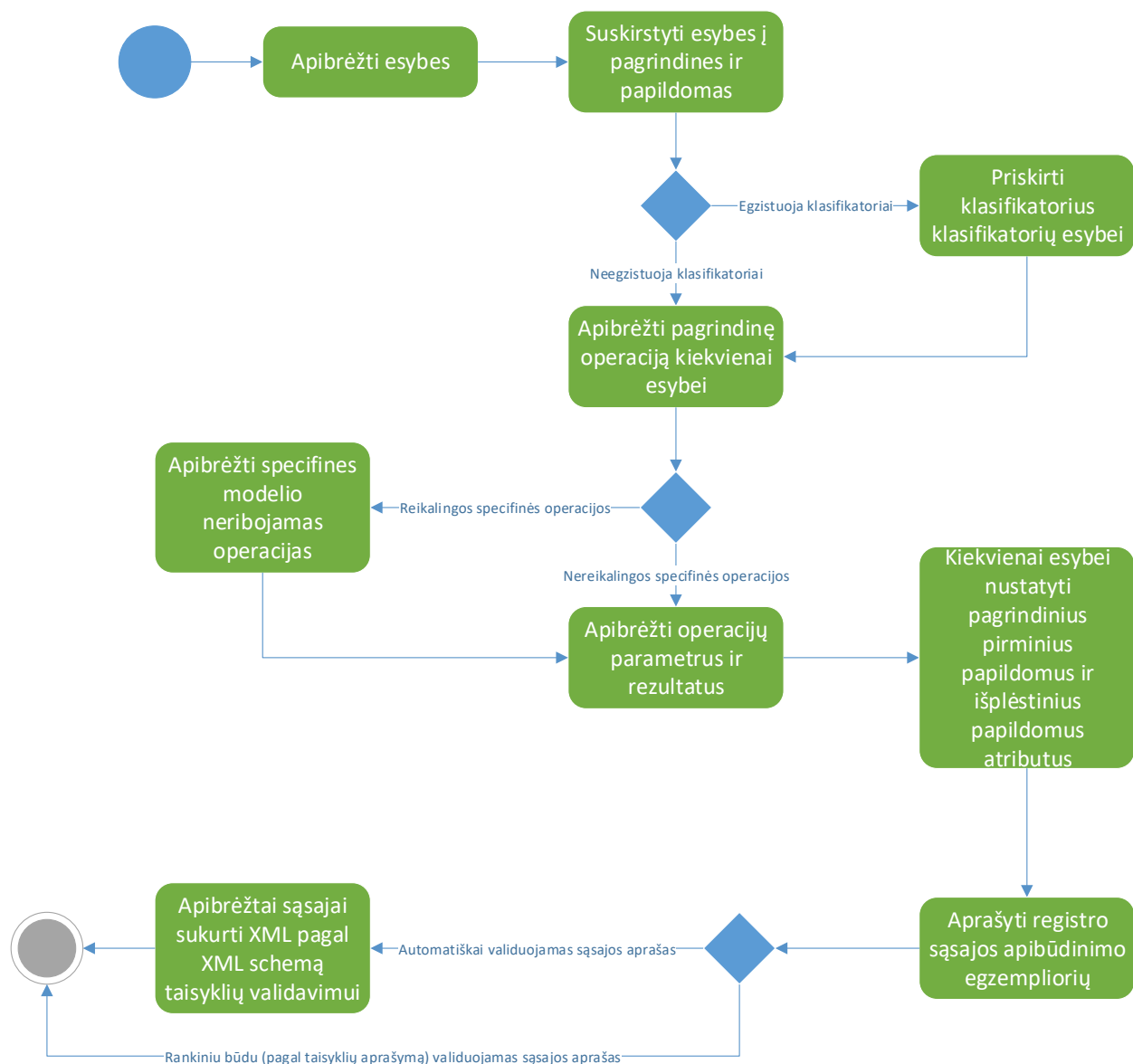
Toliau aprašysime operacijos klasę. Jos šakninį elementą pavadinsime *Operacija* ir pridėsime jam du atributus *Pavadinimas* ir *Tikslas*. Taip pat į operacijos vidų sudėsime elementus *OperacijosParametras* ir *OperacijosRezultatas*, kadangi šios dvi klasės yra neatsiejama operacijos dalis. Operacijos parametro elemento viduje dėl skaidymo ryšio įterpsime tokias klases: *EsybiuAtributuOperacijosParametras* ir *AibesRibojimoOperacijosParametras*. Priklausomai nuo jų funkcijos ir atributų aprašytos ontologijoje sukursime vidinę jų struktūrą. Operacijos rezultato elemento viduje taip pat dėl skaidymo sąryšio sukursime du elementus: *EsybiuRinkinys* ir *Metaduomuo*. Kaip ir operacijos parametro atveju šiems elementams suteiksime atitinkamus atributus ir ribojimus. Verta pastebėti, kad operacijų rezultatų elemento *EsybiuRinkinys* tipas parinktas toks pats kaip esybės. Taip padaryta dėl to, kad esybės ir esybių rinkinio struktūra sutaptų. Ontologijos taisyklėse aprašytų operacijų ribojimas turėtų būti sukurtas kiekvienoje registro sąsajoje, ją programuojant. Ta pati struktūra padės išvengti papildomo darbo, kurį reiktų atlikti vykdant susiejimus.

Taip pat į schemą įtrauksime ir mūsų sukurtą registro sąsajos apibūdinimo egzempliorių (*RegistroSasajosApibudinimas*). Jį apibrėšime kaip atskirą žymę ir suteiksime jam tokias vidines žymes pagal atributus: sąsajos pavadinimas – *Pavadinimas*, registras pateikiantis sąsają – *Registras* ir registro pateikiančio sąsają kodas – *RegistroKodas*. Operacijų ir esybių XML schemas viduje atskirai neiškirsime tam, kad nesukeltume dviprasmiškumo. Tiek esybės tiek operacijos, tiek esybės XML schemeje jau yra nurodytos, o jų susiejimą su egzemplioriumi rodys tai, kad jos yra vienoje schemeje.

Sudarytos registrų integracinių modelių XML schemas abstraktų variantą galima rasti 3 Priedase. Čia pateikiama schema gali būti pritaikoma konkrečių registrų schemų sudarymui ir aprašymui. Kuriant kiekvieną iš naujų schemų, ją galima papildyti įvairiais specifiniais duomenimis, tačiau reikia atsižvelgti į ontologijos ir schemas reikalavimus. Tokiu būdu bus išlaikoma bendra struktūra ir sąsajos universalumas.

2.3. Procesas modelio taikymui

Aprašytos registrų apibendrinto integracijos modelio taisyklės mums leidžia bendrai aprašyti registrų integracines sąsajas. Šių taisyklių taikymui yra reikalingas tam tikras procesas, kurį būtų galima pritaikyti (14 pav.). Proceso įėjtis yra registre saugomų duomenų aibė, o išeitis yra aprašyta registro integracinė sąsaja pagal registrų apibendrintą integracijos modelį.



14 pav. Procesas modelio pritaikymui integracinės sąsajos kūrimui.

Modelis pritaikomas pasinaudojant veiksmų seka pateikta 14 paveikslėlyje. Pirmiausia pagal registro duomenų modelį apibrėžiamos esybės. Tada jos yra suskirstomos į pagrindines ir papildomas, bei klasifikatoriaus esybes, jei tokios esybės egzistuoja. Tada kiekvienai esybei yra apibrėžiama pagrindinė operacija apibrėžta 2.1.6. punkte. Tada, jei reikia galima apibrėžti ir specifines operacijas. Jos yra ribojamos tik aksiomų apibrėžtų 2.1.6. punkte. Apibrėžus operacijas aprašomi operacijų parametrai bei rezultatai. Toliau iš registre saugomų objektų atributų kiekvienai esybei nustatomi atributai ir priskiriami pagrindiniams, pirminiems papildomiems ir

išplėstiniams papildomiems atributams. Paskutinis darbas pritaikant taisykles yra registro sąsajos apibūdinimo egzemplioriaus apibrėžimas. Toliau priklausomai nuo validavimo būdo (rankinis pagal aprašą arba automatinis pagal schema) kuriamas arba nekuriamas sąsajos XML. Procesas gali būti iteratyvus, laikant, kad aukščiau apibrėžtas procesas yra viena iteracija.

3. Registrų apibendrinto integracijos modelio taikymas

Šiame skyriuje sukurtą modelį praktiškai pritaikysime nagrinėtiems Lietuvos Respublikos registrams. Kiekvienam registrui modelio taikymas susidės iš kelių žingsnių – pirmiausia aprašysime naują sąsają pagal modelį, paskui palyginsime esamos sąsajos galimybes su naujai siūlomos ir pateiksime išvadas.

Ne visi nagrinėti registrai turi aiškiai išreikštas dabar egzistuojančias sąsajas. Kaip ir minėta anksčiau (1.1 poskyryje) kai kurios iš jų yra apibrėžiamos kiekvienai iš naujų sistemų atskirai. Dėl šios priežasties ne visiems registrams pavyks tiksliai apibūdinti esamos ir naujos sąsajų galimybes. Nepaisant to, bus parodyta, kad bendrą sąsają sukurti įmanoma ir įvertinti jos pranašumus prieš sąsajas, kurios kuriamos kiekvienai sistemai atskirai.

3.1. Adresų registro sąsaja pagal sukurtą modelį

Pirmiausia pradėsime nuo Adresų registro. Pritaikysime Registre saugomus objektus sukurtam modeliui. Adresų registre yra saugomi septyni objektai (1.1.1.1 papunktis) :

- Apskritys,
- Savivaldybės,
- Seniūnijos,
- Gyvenamosios vietovės,
- Gatvės,
- Adresai,
- Pavadinimai suteikti pastatui, statiniui ar kitam objektui.

Vienas iš būdų supaprastinti sąsają būtų apjungti kai kuriuos objektus, kurių atributai yra tokie pat tiesiog skiriasi tipas. Tai nėra būtinas veiksmas ir analogiška sąsaja galima ir be to, tačiau mes atliksime šiuos pakeitimus vien dėl to, kad jie atlikti ir kuriant esamą sąsają. Vienodi objektai leis paprasčiau palyginti sąsajas. Taigi, sutraukime apskritis, savivaldybes ir seniūnijas į administracinių vienetų esybę, o adresus ir pavadinimas suteiktus pastatui, statinius ar kitam objektui į bendrą adreso esybę. Gausime tokias esybes:

- Administraciniai vienetai,
- Gyvenamosios vietovės,
- Gatvės,
- Adresai.

Modelyje išskiriamos pagrindinės ir papildomos esybės, taigi tą patį reikia padaryti ir kuriant sąsają. Šiame registre labai akivaizdžiai matyti (vien iš pavadinimo), kad pagrindinė esybė bus Adresas. Administracinius vienetus, gyvenamąsias vietas ir gatves priskirsime papildomoms esybėms. Taip pat yra dar keturi adresų registro klasifikatoriai [LRV15]:

- administracinių vienetų tipų klasifikatorius,
- gyvenamųjų vietovių tipų klasifikatorius,
- gatvių tipų klasifikatorius,
- institucijų, kurių nėra Juridinių asmenų registre, klasifikatorius.

Juos priskirsime papildomų esybių poklasiui – klasifikatoriams.

Toliau nagrinėsime operacijas. Adresų registrui kursime tik pagrindines operacijas, kurios yra aprašytos ontologijos taisyklėse t. y. operacijas skirtas visų esybių gavimui su aibės ribojimo parametru bei papildomu neprivalomu identifikatoriaus parametru. Kiekvienai esybei sukursime po operaciją ir jas pavadinsime – *gautiAdresus*, *gautiAdministraciniusVienetus*, *gautiGyvenamąsiasVietoves*, *gautiGatves*, *gautiAdmVienetuTipus*, *gautiGyvenamujuVietoviuTipus*, *gautiGatviuTipus*, *gautiPapildomasInstitucijas*. Iš anksčiau minėtų operacijų apibrėžimo išplaukia ir kitų dviejų klasių – operacijos parametrų ir operacijos rezultatų klasės objektai. Kiekvienai operacijai pagal sąryšius apibrėžkime parametrus ir rezultatus:

- *gautiAdresus*
 - Parametrai – *aibesRibojimas*, *adresoId*
 - Rezultatai – adresų objektų rinkinys
- *gautiAdministraciniusVienetus*
 - Parametrai – *aibesRibojimas*, *administracinioVienetoId*
 - Rezultatai – administracinių vienetų objektų rinkinys
- *gautiGyvenamąsiasVietoves*
 - Parametrai – *aibesRibojimas*, *gyvenamosiosVietovesId*
 - Rezultatai – gyvenamųjų vietovių objektų rinkinys
- *gautiGatves*
 - Parametrai – *aibesRibojimas*, *gatvesId*
 - Rezultatai – gatvių objektų rinkinys
- Klasifikatorių operacijos
 - Parametrai – *kodas*
 - Rezultatai – klasifikatoriaus kodų ir reikšmių rinkinys

Apibrėžę pagrindines operacijas, pereikime prie atributų aibių nustatymo. Kiekvienai iš esybių reikia nustatyti pagrindinius, pirminius papildomus ir išplėstinius papildomus atributus. Šiuo atveju tai padarysime naudodamiesi registro nuostatuose [LRV15a] pateiktais saugomais atributais ir logika apibūdinta atributų formavimo taisyklėse (2.1.4 punktą). Kai kuriuos iš atributų priskirsime vieniems ar kitiems savo nuožiūra, tačiau esminio skirtumo tai nesudarys, nes

visuose registruose atributų tipai turės būti skirstomi naudojantis ne tik taisyklėmis, bet ir specifine registro logika.

Klasifikatorių atributai yra specifiniai ir jau aprašyti (kodas, reikšmė), todėl jų dar kartą atskirai neaprašysime. Kiekvienos iš likusių esybių atributus aprašome atskirai ir detalizuojame skirstydami į atitinkamas atributų grupes. Atributų aibės suskirstymą galime pamatyti 4 Priedase.

Taip pat aprašysime registro sąsajos apibūdinimo egzempliorių:

- Registro sąsajos pavadinimas – Adresų registro sąsaja,
- Registro pateikiančio sąsają pavadinimas – Lietuvos Respublikos Adresų registras,
- Registro pateikiančio sąsają kodas – ADRESU_REGISTRAS.

Apibrėžę atributų esybės, operacijas, operacijų parametrus, operacijų rezultatus ir suformavę atributų aibes gauname adresų registro sąsajos aprašymą. Norint kurti anksčiau minėtą registru duomenų bazę reikia sukurti adresų registro sąsajos XML schemą, kurią būtų galima saugoti toje duomenų bazėje. Sukūrus tokią schemą daug paprasčiau sukurti sąsają paremtą XML duomenimis, pavyzdžiui, SOAP tinklinę sąsają (angl. SOAP web-service). Bet kokių atveju verta pabrėžti, kad sukurta ontologija ir registro sąsajos aprašas neverčia naudoti jokios technologijos ir ją galima laisvai pasirinkti.

3.2. Adresų registro sąsajų palyginimas

Šiame skyriuje palyginsime adresų registro esamą sąsają (toliau – esama Adresų registro sąsaja) ir pagal modelį sukurtą naująją sąsają (toliau – adresų registro sąsaja pagal modelį). Lyginimą atliksime palygindami nagrinėtas sąsajos charakteristikas: naudojamas technologijas, esybės, atributų aibę ir operacijas (2 Lentelė).

2 Lentelė. Adresų registro sąsajų palyginimas.

Charakteristika \ Sąsaja	Esama Adresų registro sąsaja	Adresų registro sąsaja pagal modelį
Naudojamos technologijos	Pateikiami adresų registro failai	Technologija nėra ribojama. Rekomenduojama sudaryti XML schemą.
Esybės	Pateikiamos keturios registro objektų esybės – administraciniai vienetai, gyvenamosios vietovės, gatvės adresai. Taip pat pateikiami keturi klasifikatoriai.	Pateikiamos keturios registro objektų esybės – administraciniai vienetai, gyvenamosios vietovės, gatvės adresai. Taip pat pateikiami keturi klasifikatoriai.

Atributų aibė	Per sąsają gaunama tik pilnos esybės su atributų aibėmis. Gaunama atributų aibė yra pilna, išskyrus tokius atributus kaip grafiniai duomenys, kai kurių objektų tipai (pvz. gatvių) ir pan. Detaliau aprašyta 1.1.1.1 punkte nagrinėjant sąsają.	Per sąsają galima gauti trijų rūšių duomenų rinkinius – pagrindinį, pirminį ir išplėstinį. Juose gaunami atributai priklauso nuo to, kuriai grupei jie priskirti. Išplėstinis duomenų rinkinys leidžia gražinti visą duomenų aibę.
Operacijos	Pateikiami failai visoms keturioms teikiamoms registro objektų esybėms, taip pat ir klasifikatorių esybėms. Kadangi pateikiami failai, operacijos parametrų neturi. Rezultatas – pilnas faile pateikiamų esybių sąrašas.	Pateikiamos operacijos visoms apibrėžtomis registro objektų ir klasifikatorių esybėms. Operacijos gali priimti du parametrus – aibės ribojimo ir registro objekto identifikatoriaus. Rezultatas priklauso nuo parametrų – gali būti gaunama viena arba visos esybės su skirtingomis atributų aibėmis priklausomai nuo aibės ribojimo parametro. Aibės ribojimo parametrui su reikšmė išplėstinis gaunama pilna atributų aibė.

Palyginus sąsajas galime matyti, kad adresų registro sąsaja pagal modelį yra daug platesnė ir leidžia lanksčiau gauti Adresų registre saugomus duomenis. Taip pat peržvelgus kiekvieną charakteristiką detaliau galime matyti, kad esama sąsaja yra siauresnis registro sąsajos pagal modelį variantas. Tai leidžia daryti išvadą, kad paėmus dabartinę sąsają ir ją pritaikius sukurtoms taisyklėms (Pridėjus aibės ribojimo parametrus ir pritaikius bendrumą užtikrinančias taisykles) galėsime sudaryti sąsają pagal modelį, jai kurti XML schemą ir tokiu būdu gauti apibendrintą sąsają.

3.3. Gyventojų registro sąsaja pagal sukurtą modelį

Aprašysime Gyventojų registro sąsają pagal registrų apibendrintą integracijos modelį. Pirmiausia pritaikysime Gyventojų registro objektus. Pagal literatūros apžvalgą (1.1.1.2. papunktis) registre saugomi dviejų tipų objektai:

- Lietuvos Respublikos piliečiai,
- asmenys be pilietybės ir kitų valstybių piliečiai, gaunantys Lietuvos Respublikoje išduodamus asmens dokumentus, deklaruojantys gyvenamąją vietą Lietuvoje ar kurių civilinės būklės aktai yra registruojami Lietuvos Respublikos institucijose.

Kadangi šie du objektų tipai turi tą pačią atributų aibę (1.1.1.2. papunktis), tai galime supaprastinti esybų aibę ir apibrėžti vieną bendrą esybę – **asmenį**.

Kaip matome šiuo atveju sąsajoje egzistuos tik viena esybė, kuri yra pagrindinė. Be šios esybės registre saugoma ir 11 klasifikatorių (1.1.1.2. papunktis), kuriuos priskirsime papildomoms esybėms ir jų poklasiui – klasifikatoriams. Taigi, gavome sąsajos esybes.

Toliau aprašysime operacijas. Kursime tik pagrindines operacijas, kurio privalomos pagal registrų apibendrinto integracijos modelio taisykles (2.1.6. punktas). Kiekvienai esybei reikia sukurti po operaciją. Šiuo atveju bus viena operacija pagrindinei esybei – *gautiAsmenis* ir operacijos klasifikatorių reikšmėms gauti. Aprašydami šias operacijas taip pat gausime ir operacijos parametro ir rezultato klases:

- *gautiAsmenis*
 - Parametrai – *aibesRibojimas, asmensId*
 - Rezultatai – asmenų objektų rinkinys
- Klasifikatorių operacijos
 - Parametrai – *kodas*
 - Rezultatai – klasifikatoriaus kodų ir reikšmių rinkinys

Toliau nustatysime atributų aibes. Reikia nustatyti pagrindinius, pirminius papildomus ir išplėstinius papildomus atributus (2.1.4 punktas). Šių atributų aibių nustatymui pasinaudosime Gyventojų registro nuostatuose [LRV17] pateikta registre saugomų atributų aibe. Šiuo atveju aibes reikės nustatinėti vienai – asmens esybei. Kaip ir Adresų registru, atskirai neparašinėsime klasifikatorių atributų, nes jie yra specifiniai ir apibrėžti bendrai visoms sąsajoms kuriamoms pagal registrų apibendrintą integracijos modelį. Pilną atributų aibės aprašymą galima rasti 5 priede.

Taip pat aprašysime Gyventojų registro sąsajos apibūdinimo egzempliorių:

- Registro sąsajos pavadinimas – Gyventojų registro sąsaja,
- Registro pateikiančio sąsają pavadinimas – Lietuvos Respublikos Gyventojų registras,
- Registro pateikiančio sąsają kodas – GYVENTOJU_REGISTRAS.

Apibrėžę visas reikalingas sąsajos pagal registru apibendrintą integracijos modelį dalis Gyventojų registrui gauname naują integracinę sąsają. Turėdami tokį sąsajos aprašą kaip ir Adresų registro sąsajos atveju galime sukurti XML schema, bei kurti realią SOAP ar kitomis technologijomis pagrįstą integracinį tašką.

3.4. Gyventojų registro sąsajų palyginimas

Palyginsime pagal modelį sukurtą (toliau – Gyventojų registro sąsaja pagal modelį) ir jau esamas Gyventojų registro sąsajas (toliau – esamos Gyventojų registro sąsajos). Palyginimas bus vykdomas pagal tas pačias charakteristikas kaip ir Adresų registro atveju – naudojamas technologijas, esybes, atributų aibę ir operacijas (3 lentelė).

3 Lentelė. Gyventojų registro sąsajų palyginimas.

Charakteristika \ Sąsaja	Esamos Gyventojų registro sąsajos	Gyventojų registro sąsaja pagal modelį
Naudojamos technologijos	Duomenų bazių procedūros (angl. stored procedures), materializuotos virtualios lentelės (angl. materialized views), registro objektų XML sąrašai	Technologija nėra ribojama. Rekomenduojama sudaryti XML schema.
Esybės	Perduodama asmens esybė, bei yra galimybė gauti klasifikatorius	Pateikiama viena registro esybė - asmuo. Taip pat pateikiama 11 klasifikatorių.
Atributų aibė	Atributų aibė stipriai priklauso nuo sistemos ir nėra aiškiai apibrėžta. Tai sąlygoja, kad gali tekti pritaikyti vis naują sąsają naujai sistemai.	Per sąsają galima gauti trijų rūšių duomenų rinkinius – pagrindinį, pirminį ir išplėstinį. Juose gaunami atributai priklauso nuo to, kuriai grupei jie priskirti. Išplėstinis duomenų rinkinys leidžia grąžinti visą duomenų aibę.
Operacijos	Pateikiamos operacijos duomenų gavimui. Aibė ir operacijos parametrai nėra	Pateikiamos operacijos visoms apibrėžtoms registro objektų ir klasifikatorių esybėms. Operacijos gali

	žinomi ir pritaikomi kiekvienai sistemai atskirai.	priimti du parametrus – aibės ribojimo ir asmens objekto identifikatoriaus. Rezultatas priklauso nuo parametrų – gali būti gaunama viena arba visos esybės su skirtingomis atributų aibėmis priklausomai nuo aibės ribojimo parametro. Aibės ribojimo parametrai su reikšmė išplėstinis gaunama pilna atributų aibė.
--	--	--

Palyginus sąsajas galime matyti, kad esamos Gyventojų registro sąsajos yra labiau pritaikomos kiekvienai sistemai atskirai ir dėl to nebelieka bendrumo, kuris yra sąsajoje sukurtoje pagal modelį. Sąsaja pagal modelį suteikia galimybę gauti tą pačią atributų aibę kaip ir esamos sąsajos, tačiau šią aibę suskirsto dalimis ir leidžia daugeliui sistemų naudotis ta pačia sąsaja. Taip pat sąsaja pagal modelį leidžia pasirinkti daugiau technologinių sprendimų. Iš palyginimo galime matyti, kad iš esmės operacijų ir atributų atžvilgiu sąsaja pagal modelį padengia ir esamas sąsajas. Tam tikrais atvejais esamos sąsajos gali būti efektyvesnės, jeigu perduodama atributų aibė yra ypač specifinė gaunančiai sistemai. Nepaisant to tokia sąsaja greičiausiai ir bus tinkama tik tai vienai sistemai. Taigi, kaip ir Adresų registro atveju, galime daryti išvadą, kad sąsaja pagal modelį teikia panašias galimybes kaip ir esančios sąsajos, tačiau suteikia privalumų užtikrinančių sąsajų bendrumą.

3.5. Juridinių asmenų registro sąsaja pagal sukurtą modelį

Toliau aprašysime Juridinių asmenų registro sąsają pagal registru apibendrintą integracijos modelį. Registre yra saugomi tokie objektai (1.1.1.3. papunktis):

- Privatūs juridiniai asmenys,
- Viešieji juridiniai asmenys,
- Juridinių asmenų filialai ir atstovybės.

Nustatysime esybes pagal saugomus objektus. Privatūs ir viešieji juridiniai asmenys turi beveik vienodas duomenų aibes (1.1.1.3. papunktis) todėl juos išskirsime kaip vieną esybę - juridinis asmuo. Filialus ir atstovybes išskirsime kaip dar vieną esybę – filialas/atstovybė.

Akivaizdu, kad juridiniai asmenys turi būti išskirti kaip pagrindinė esybė Juridinių asmenų registre. Filialus ir atstovybes skirsime prie papildomų esybių, nes savarankiškai, be „tėvinio“ juridinio asmens ši esybė egzistuoti negali (negali būti juridinio asmens filialo be juridinio

asmens). Taip pat registras turi 18 klasifikatorių, kuriuos visus priskirsime papildomų esybių klasės, klasifikatorių poklasiui.

Apibrėšime registro sąsajos operacijas. Kiekvienai esybei bus sukurta registrų apibendrinto integracijos modelio taisyklėse (2.1.6. punktas) nustatyta pagrindinė operacija. Juridinių asmenų esybėms gauti bus sukurta operacija *gautiJuridiniusAsmenis*, o filialams ir atstovybėms gauti – *gautiFilialusAtstovybes*. Taip pat aprašomos standartinės operacijos, skirtos klasifikatoriams gauti. Apibrėžiant šias operacijas apibrėžiami ir operacijos parametrų ir operacijos rezultatų klasės objektai. Kiekvienai operacijai šie objektai bus tokie:

- *gautiJuridiniusAsmenis*
 - Parametrai – *aibesRibojimas, juridinioAsmensId*
 - Rezultatai – juridinių asmenų objektų rinkinys
- *gautiFilialusIrAtstovybes*
 - Parametrai – *aibesRibojimas, filialoAtstovybesId*
 - Rezultatai – filialų ir atstovybių objektų rinkinys
- Klasifikatorių operacijos
 - Parametrai – *kodas*
 - Rezultatai – klasifikatoriaus kodų ir reikšmių rinkinys

Nustatome reikalingas atributų aibes. Reikia apibrėžti pagrindinius, pirminius papildomus ir išplėstinius papildomus atributus (2.1.4 punktas). Šį kartą aibę skirstysime šiek tiek kitaip nei Adresų ar Gyventojų registro atvejais. Juridinių asmenų registro egzistuojančioje integracinėje sąsajoje jau aprašomi trys galimi gauti duomenų rinkiniai (1.1.1.3. punktas). Jie beveik atitinka mūsų skirstymą, tik nėra atskirai išskirtos aibės Juridinių asmenų ir filialų/atstovybių objektams. Siekdami supaprastinti aibės sudarymą ir remdamiesi tuo, kad juridinių asmenų registro projektuotojai geriau supranta registro subtilybes tiesiog pritaikysime jų pasiūlytą atributų aibes Juridinių asmenų ir filialų/atstovybių esybėms. Taikoma tokiu principu:

- Pagrindiniai atributai – trumpajame duomenų rinkinyje esantys atributai,
- Pirminiai papildomi atributai – pagrindiniame rinkinyje esantys atributai, kurie nėra trumpajame rinkinyje,
- Išplėstiniai papildomi atributai – išplėstiniame rinkinyje esantys atributai, kurie nėra nei trumpajame, nei pagrindiniame rinkinyje.

Pastaba. Juridinių asmenų ir filialų bei atstovybių esybėms šie atributai yra išskirstomi pagal jų priklausymą vienam ar kitam saugomam objektui registre.

Taip pat aprašysime Juridinių asmenų registro sąsajos apibūdinimo egzempliorių:

- Registro sąsajos pavadinimas – Juridinių asmenų registro sąsaja,
- Registro pateikiančio sąsają pavadinimas – Juridinių asmenų registras,

- Registro pateikiančio sąsają kodas – JURIDINIU_ASMENU_REGISTRAS.

Apibrėžėme Juridinių asmenų registro sąsają pagal registru apibendrintą integracijos modelį. Kaip ir anksčiau aptartiems Adresų bei Gyventojų registrams Juridinių asmenų registrui galima sukurti XML schemą, kuri būtų naudojama registru sąsajų bibliotekai arba sąsajos WSDL aprašyti.

3.6. Juridinių asmenų registro sąsajų palyginimas

Palyginsime pagal modelį sukurtą Juridinių asmenų registro sąsają ir esamą Juridinių asmenų registro sąsają. Sąsajų palyginimui naudosime tas pačias charakteristikas – naudojamas technologijas, esybes, atributų aibę ir operacijas (4 lentelė).

4 Lentelė. Juridinių asmenų registro sąsajų palyginimas.

Charakteristika \ Sąsaja	Esama Juridinių asmenų registro sąsaja	Juridinių asmenų registro sąsaja pagal modelį
Naudojamos technologijos	Duomenų gavimas XML formatu (nedetalizuojama kaip)	Technologija nėra ribojama. Rekomenduojama sudaryti XML schemą.
Esybės	Pateikiama juridinio asmens esybė, kuria padengiami visi trys registre saugomi objektai	Pateikiama viena pagrindinė registro esybė – juridinis asmuo ir viena papildoma – filialas/atstovybė. Taip pat pateikiama 18 klasifikatorių.
Atributų aibė	Atributų aibė skirstoma taip, kad būtų galima pateikti tris duomenų rinkinius – trumpąjį, pagrindinį ir išplėstinį.	Per sąsają galima gauti trijų rūšių duomenų rinkinius – pagrindinį, pirminį ir išplėstinį. Juose gaunami atributai priklauso nuo to, kuriai grupei jie priskirti. Išplėstinis duomenų rinkinys leidžia gražinti visą duomenų aibę.
Operacijos	Pateikiamos operacijos juridinių asmenų duomenų rinkiniams bei klasifikatoriams gauti. Ar yra pateikiamos trys operacijos, ar	Pateikiamos operacijos visoms apibrėžtomis registro objektų ir klasifikatorių esybėms. Operacijos gali priimti du parametrus – aibės ribojimo ir asmens objekto

	viena operacija su parametru neapibrėžiama.	identifikatoriaus. Rezultatas priklauso nuo parametru – gali būti gaunama viena arba visos esybės su skirtingomis atributų aibėmis priklausomai nuo aibės ribojimo parametro. Aibės ribojimo parametru su reikšmė išplėstinis gaunama pilna atributų aibė.
--	---	--

Iš palyginimo galime matyti, kad esama sąsaja yra panaši į naujai apibrėžtą. Esamoje sąsajoje jau egzistuoja registrų apibendrinto integracijos modelio principas skirstyti duomenų aibę į tris duomenų rinkinius nuo mažiausio iki didžiausio. Sąsaja pagal yra lankstesnė technologijų požiūriu ir padengia ir esamos sąsajos XML duomenų rinkinių perdavimą. Taip pat sąsaja pagal modelį pateikia pagrindines ir papildomas esybes (Juridinius asmenis ir filialus/atstovybes), kas leidžia padaryti sąsają dar lankstesnę ir reikalui esant perdavinėti tik pagrindines esybes. Taip pat naujoji sąsaja yra griežčiau apibrėžta. Taigi, Juridinių asmenų registro sąsaja pagal modelį padengia ir ankstesnę sąsają, bei turi papildomų privalumų.

3.7. Įgaliojimų registro sąsaja pagal sukurtą modelį

Aprašysime Įgaliojimų registro sąsają pagal registrų apibendrintą integracijos modelį. Pradėsime nuo registre saugomų objektų priskyrimo prie sąsajos esybių. Iš Įgaliojimų registro apžvalgos (1.1.1.4 papunktis) galime matyti, kad registre saugomi šie objektai:

- Notarų, konsulinių pareigūnų patvirtinti įgaliojimai,
- Notarų patvirtintiems prilyginami įgaliojimai,
- Informacinių technologijų priemonėmis sudaryti įgaliojimai.

Registro apžvalgoje (1.1.1.4) teigiama, kad notarų ir konsulinių pareigūnų patvirtinti įgaliojimai, bei notarų patvirtintiems prilyginami įgaliojimai turi tą patį duomenų rinkinį todėl šiuos įgaliojimus sutrauksime į vieną esybę – įgaliojimas. Informacinėmis priemonėmis sukurtus įgaliojimus apibūdinsime esybe – IT įgaliojimas. Tiek įgaliojimai, tiek IT įgaliojimai šio registro atveju bus pagrindinės esybės, nes tai yra esminiai registro objektai (apie tai galima spręsti vien iš registro pavadinimo). Taip pat registre yra 9 klasifikatoriai, kuriuos priskirsime papildomų esybių klasės klasifikatorių poklasiui.

Toliau apibrėšime sąsajos operacijas. Kiekvienai esybei apibrėžiamos pagrindinės operacijos reikalingos pagal registrų apibendrinto integracijos modelio taisykles (2.1.6 punktą). Taigi, sąsaja turės dvi operacijas pagrindinėms esybėms ir operacijas gauti klasifikatorių reikšmėms. Įgaliojimams gauti kuriama operacija – *gautiĮgaliojimus*, o IT įgaliojimams –

gautiITĮgaliojimus. Apibrėždami operacijas apibrėžiame ir operacijų paramterų, bei operacijų rezultatų klasių objektus:

- *gautiĮgaliojimus*
 - Parametrai – *aibesRibojimas, įgaliojimoId*
 - Rezultatai – įgaliojimų objektų rinkinys
- *gautiITĮgaliojimus*
 - Parametrai – *aibesRibojimas, itĮgaliojimoId*
 - Rezultatai – IT įgaliojimų objektų rinkinys
- Klasifikatorių operacijos
 - Parametrai – *kodas*
 - Rezultatai – klasifikatoriaus kodų ir reikšmių rinkinys

Toliau nustatysime atributų aibes. Atributų aibės nustatomos remiantis jų formavimo taisyklėmis apibrėžtomis 2.1.4 punkte. Įgaliojimų registro esamoje sąsajoje, kaip ir Juridinių asmenų registre, yra taikomas aibių skirstymas. Pagrindinis skirtumas yra tas, kad aibės yra skirstomos visiškai atskirai, o ne atributų kiekio didėjimo tvarka. Dėl šios priežasties tokio pat principo kaip Juridinių asmenų registre skirstant atributų aibę pritaikyti nepavyks. Atributų aibė skirstoma pasinaudojant ta pačia logika kaip Adresų ir Gyventojų registro atveju. Ši aibė pateikiama 6 priede.

Taip pat aprašysime Įgaliojimų registro sąsajos apibūdinimo egzempliorių:

- Registro sąsajos pavadinimas – Įgaliojimų registro sąsaja,
- Registro pateikiančio sąsają pavadinimas – Įgaliojimų registras,
- Registro pateikiančio sąsają kodas – IGALIOJIMU_REGISTRAS.

Apibrėžėme Įgaliojimų registro sąsają pagal registrų apibendrintą integracijos modelį. Kaip ir ankstesnių registrų atveju toliau galima aprašyti XML schemą arba kurti sąsają pasinaudojant aprašytomis klasėmis ir taisyklėmis.

3.8. Įgaliojimų registro sąsajų palyginimas

Įgaliojimų registro sąsają pagal modelį palyginsime su esama sąsaja. Lyginimas bus vykdomas pagal anksčiau naudotus kriterijus – naudojamas technologijas, esybes, atributų aibę ir operacijas (5 Lentelė2 Lentelė).

5 Lentelė. Įgaliojimų registro sąsajų palyginimas.

Charakteristika \ Sąsaja	Įgaliojimų registro esama sąsaja	Įgaliojimų registro sąsaja pagal modelį
Naudojamos technologijos	Tiksli naudojama technologija nepateikiama.	Technologija nėra ribojama. Rekomenduojama sudaryti XML schemą.
Esybės	Pateikiama viena įgaliojimo esybė.	Pateikiamos dvi registro objektų esybės – įgaliojimai ir IT įgaliojimai. Taip pat pateikiami devyni klasifikatoriai.
Atributų aibė	Atributų aibė skirstoma į keletą poaibių. Šie poaibiai nebūtinai persidengia vienas su kitu ir norint gauti pilną duomenų rinkinį reikia kelių operacijų, skirtų gauti skirtingiems atributų poaibiams	Per sąsają galima gauti trijų rūšių duomenų rinkinius – pagrindinį, pirminį ir išplėstinį. Juose gaunami atributai priklauso nuo to, kuriai grupei jie priskirti. Išplėstinis duomenų rinkinys leidžia gražinti visą duomenų aibę.
Operacijos	Numatytos penkios operacijos, skirtos konkrečiau įgaliojimo paieškai. Operacijos turi parametrus, kurie vienareikšmiškai nusako vieną įgaliojimą. Rezultatas – įgaliojimas pagal pradinius parametrus. Visos aibės gauti nepavyks.	Pateikiamos operacijos visoms apibrėžtomis registro objektų ir klasifikatorių esybėms. Operacijos gali priimti du parametrus – aibės ribojimo ir registro objekto identifikatoriaus. Rezultatas priklauso nuo parametru – gali būti gaunama viena arba visos esybės su skirtingomis atributų aibėmis priklausomai nuo aibės ribojimo parametro. Aibės ribojimo parametru su reikšmė išplėstinis gaunama pilna atributų aibė.

Palyginus Įgaliojimų registro esamą ir sąsają pagal modelį galime matyti, kad sąsaja pagal modelį yra šiek tiek kitoks esamos sąsajos variantas. Sąsaja pagal modelis leidžia gauti duomenis dviejų esybių pavidalu, kai tuo tarpu esama sąsaja informacinėmis technologijomis ir fiziškai sukurtų įgaliojimų neskiria ir turi vieną įgaliojimo esybę. Taip pat esama sąsaja skirsto duomenų aibę į poaibius taip, kad gaunamų duomenų aibė augtų didėjimo tvarka priklausomai nuo aibės ribojimo parametro. Esama sąsaja turi daugiau operacijų, nei sąsaja pagal modelį, todėl leidžia gauti vieną įgaliojimą pagal šiek tiek daugiau kriterijų, tačiau neleidžia gauti pilnos duomenų aibės. Lyginant šias sąsajas galime matyti, kad kai kuriose vietose (pvz. operacijų atžvilgiu) esama sąsaja gali būti pranašesnė. Nepaisant to, esant poreikiui modelis nedraudžia sukurti ir daugiau operacijų, kurios būtų panašios į esamos sąsajos operacijas. Tokiu būdu nebūtų panaikinimas sąsajos pagal modelį bendrumas, nes ji vis tiek atitiktų modelio XML schemą sąsajų sudarymui (2.2), bei registų apibendrinto integracijos modelio taisyklės.

3.9. Nekilnojamojo turto registro sąsaja pagal modelį

Aprašysime Nekilnojamojo turto registro sąsają pagal registų apibendrintą integracijos modelį. Pradėsime nuo registre saugomų objektų priskyrimo sąsajos esybėms. Pagal 1.1.1.5 papunktį registre saugomi šie objektai: Nekilnojamojo turto registro įstatyme nurodyti nekilnojamieji daiktai, daiktinės teisės į juos, šių teisių suvaržymai ir su nekilnojamaisiais daiktais, daiktinių teisių į juos suvaržymais bei daiktinėmis teisėmis susiję juridiniai faktai, taip pat įmonių dovanojimo, pirkimo–pardavimo ir nuomos sutartys. Pagal registro objektus ir registro nuostatuose [LRV15b] pateikiamą saugomų atributų aibę objektus galime suskirstyti į tokias esybes:

- Nekilnojamasis turtas,
- Daiktinės teisės (Pastaba. Čia įeina tiek teisių suvaržymai, bei juridiniai faktai),
- Sutartys.

Nekilnojamojo turto esybę priskirsime pagrindinėms esybėms, nes tai yra esminis Nekilnojamojo turto registro objektas. Daiktinės teisės, bei sutartis priskirsime papildomoms esybėms, nes šios esybės yra iš esmės yra skirtos apibūdinti nekilnojamojo turto priklausomybę, bei veiksmus, atliekamus su juo. Taip pat tris registre saugomus klasifikatorius priskirsime papildomų esybių klasifikatorių poklasiui.

Tolesniame žingsnyje apibrėšime operacijas sąsajai pagal registų apibendrintą integracijos modelį. Kiekvienai esybei apibrėžiamos pagrindinės operacijos numatytos modelio taisyklėse (2.1.6. punktas). Nekilnojamojo turto esybei kuriama operacija – *gautiNekilnojamajiTurta*. Daiktinių teisių ir sutarčių esybėms atitinkamai aprašomos operacijos *gautiDaiktinesTeises* ir *gautiSutartis*. Aprašant šias operacijas aprašomi ir operacijų parametrų bei operacijų rezultatų klasių egzemplioriai:

- *gautiNekilnojamajiTurta*
 - Parametrai – *aibesRibojimas, nekilnojamojoTurtoId*
 - Rezultatai – nekilnojamojo turto objektų rinkinys
- *gautiDaiktinesTeises*
 - Parametrai – *aibesRibojimas, daiktiniuTeisiuId*
 - Rezultatai – daiktinių teisių objektų rinkinys
- *gautiSutartis*
 - Parametrai – *aibesRibojimas, sutartiesId*
 - Rezultatai – sutarties objektų rinkinys
- Klasifikatorių operacijos
 - Parametrai – *kodas*
 - Rezultatai – klasifikatoriaus kodų ir reikšmių rinkinys

Toliau nustatysime atributų aibę (2.1.4). Atributų aibė nustatoma tuo pačiu principu kaip ir anksčiau aprašytose registrų sąsajose pagal modelį – atributai kiekvienai esybei yra skirstomi į pagrindinius, pirminius papildomus ir išplėstinius papildomus. Atributai priskiriami kiekvienai grupei remiantis registro nuostatais [LRV15b] ir logika aprašyta atributų tipų aprašymuose (2.1.4). Išskirstytą atributų aibę galima rasti 7 priede.

Aprašysime Nekilnojamojo turto registro sąsajos apibūdinimo egzempliorių:

- Registro sąsajos pavadinimas – Nekilnojamojo turto registro sąsaja,
- Registro pateikiančio sąsają pavadinimas – Nekilnojamojo turto registras,
- Registro pateikiančio sąsają kodas – NT_REGISTRAS.

Apibrėžėme Nekilnojamojo turto registro sąsają pagal registrų apibendrintą integracijos modelį. Gautą apibrėžimą kaip ir ankstesnių registrų atveju galime taikyti toliau aprašant XML schemą registrui, bei kuriant integracinę sąsają.

3.10. Nekilnojamojo turto registro sąsajų palyginimas

Nekilnojamojo turto registro sąsają pagal registrų apibendrintą integracijos modelį palyginsime su esamomis registro sąsajomis. Lyginimą atliksime pagal anksčiau naudotas charakteristikas: naudojamas technologijas, esybes, atributų aibę ir operacijas (6 Lentelė).

6 Lentelė. Nekilnojamojo turto registro sąsajų palyginimas.

Charakteristika \ Sąsaja	Nekilnojamojo turto registro esamos sąsajos	Nekilnojamojo turto registro sąsaja pagal modelį
Naudojamos technologijos	Teikiami duomenys XML formatu	Technologija nėra ribojama. Rekomenduojama sudaryti XML schemą.

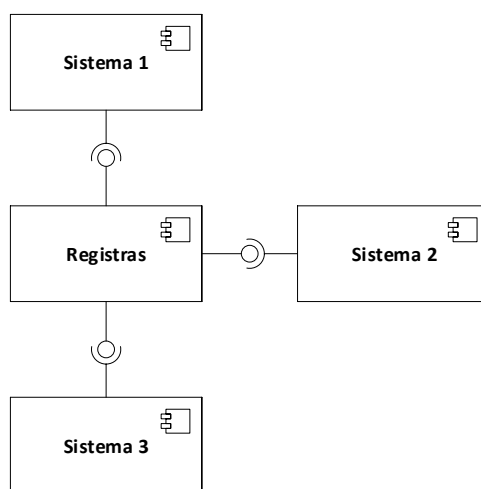
Esybės	Esybių aibė yra apibrėžiama atskirai kiekvienai naujai sąsajai, tačiau neabejotinai tarp perduodamų esybių yra nekilnojamasis turtas	Pateikiamos trys registro objektų esybės – nekilnojamasis turtas, daiktinės teisės ir sutartys. Taip pat pateikiami trys klasifikatoriai.
Atributų aibė	Atributų aibė skirstoma į priklausomai nuo sutarties pasirašytos su duomenis gaunančios sistemos valdytoju. Galima gauti pilną atributų aibę arba jos poaibius.	Per sąsają galima gauti trijų rūšių duomenų rinkinius – pagrindinį, pirminį ir išplėstinį. Juose gaunami atributai priklauso nuo to, kuriai grupei jie priskirti. Išplėstinis duomenų rinkinys leidžia gražinti visą duomenų aibę.
Operacijos	Duomenų gavimo operacijos nustatomos priklausomai nuo sutarties su duomenis gaunančios sistemos valdytoju.	Pateikiamos operacijos visoms apibrėžtomis registro objektų ir klasifikatorių esybėms. Operacijos gali priimti du parametrus – aibės ribojimo ir registro objekto identifikatoriaus. Rezultatas priklauso nuo parametrų – gali būti gaunama viena arba visos esybės su skirtingomis atributų aibėmis priklausomai nuo aibės ribojimo parametro. Aibės ribojimo parametrai su reikšmė išplėstinis gaunama pilna atributų aibė.

Galime matyti, kad esamos sąsajos yra daug mažiau apibrėžtos negu sąsaja pagal modelį. Esamos sąsajos naudoja XML technologiją, kai tuo tarpu sąsajos aprašas pagal modelį nereikalauja jokios konkrečios technologijos. Sąsaja pagal modelį turi apibrėžtą esybių bei atributų rinkinį sąsajai, o esamos sąsajos gali būti labai skirtingos ir greičiausiai kuriamos kiekvienai sistemai atskirai pagal poreikį. Toks pats principas galioja ir operacijoms. Didesnis sąsajos pagal modelį

apibrėžtumas leidžia turėti apibendrintą sąsają prie kurios gali prisitaikyti kitos sistemos. Tokiu atveju kiekvienai sistemai nereiktų kurti naujos sąsajos, kaip esamų sąsajų atveju. Taigi, Nekilnojamojo turto registro sąsaja pagal modelį įneša bendrumo ir yra geriau apibrėžta nei esamos sąsajos.

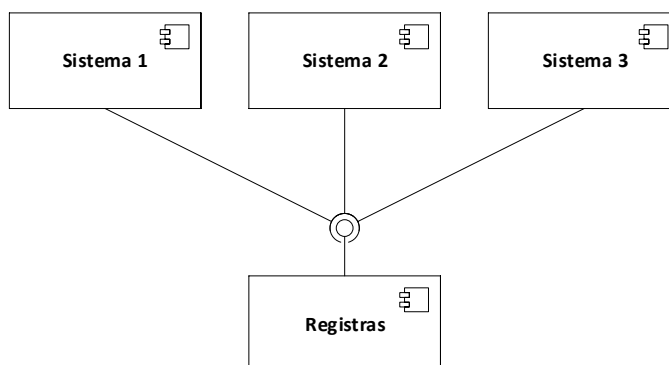
3.11. Taikymo apibendrinimas

Pritaikius registrų apibendrintą integracijos modelį pagrindiniams Lietuvos Respublikos registrams galime pademonstruoti kokie pokyčiai įvyksta sąsajose. Registruose, kalbant apie dabartines sąsajas, egzistuoja panaši situacija. Diagramoje (15 pav.) apibrėžtu atveju registro sąsajos kuriamos kiekvienai naujai sistemai atskirai. Sąsajų kiekis yra augantis priklausomai nuo naujai atsirandančių sistemų kiekio.



15 pav. Esama registrų sąsajų situacija.

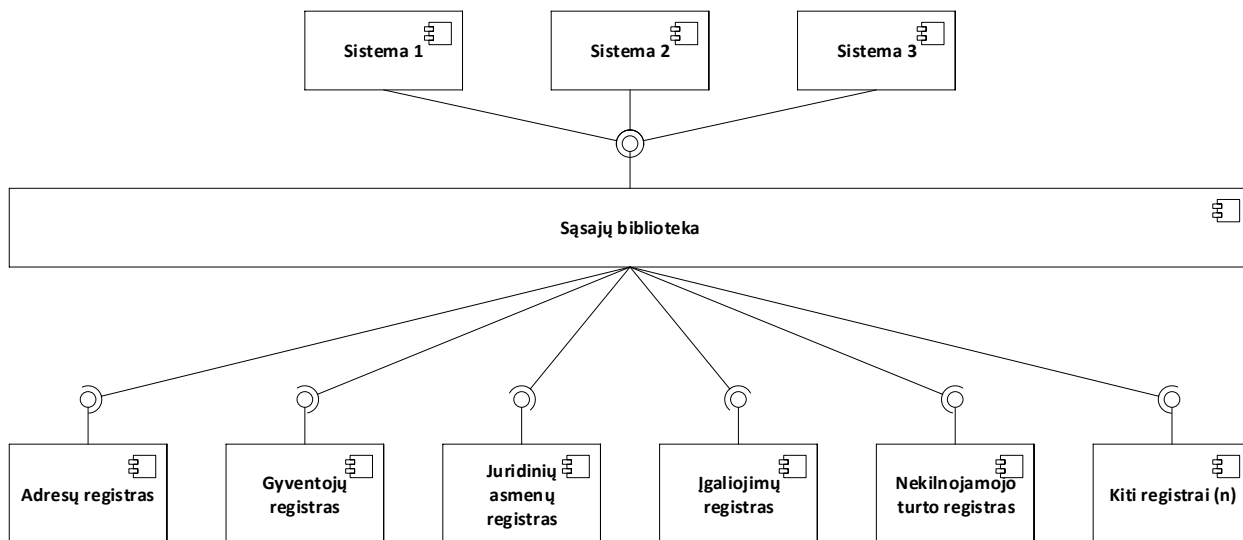
Registrų sąsaja po modelio pritaikymo sukuria kitokią situaciją (16 pav.). Pritaikius registrų apibendrintą integracijos modelį sistemoms yra pateikiama viena sąsaja ir sąsajų skaičius gali augti tik labai specifiniais atvejais. Tokiu būdu sprendžiama bendrumo problema ir registrų nebereikia papildyti naujomis sąsajomis kas kartą atsiradus naujai sistemai.



16 pav. Sąsaja po registrų apibendrinto integracijos modelio pritaikymo.

Registrų apibendrinto integracijos modelio pritaikymas leidžia pasiekti ir dar daugiau – automatizuoti duomenų visuose registruose paiešką. Sukūrus biblioteką su registrų sąsajų XML schemomis, esybėmis ir jų atributais galime sukurti ir vieningą automatizuotą sąsają per kurią

sistemos galėtų gauti duomenis iš visų registrų aibės (17 pav). Tokia sąsaja sukurtų apibendrintą visų registrų sistemą ir leistų sistemoms nebesirūpinti, kuriame registre yra saugomi joms reikalingi duomenys.



17 pav. Registrų sąsajų bibliotekos pritaikymas.

Rezultatai ir išvados

Darbe sukurtas registrų apibendrintas integracijos modelis susidedantis iš dviejų dalių – registrų integracinių sąsajų taisyklių rinkinio sudaryto pagal ontologijų kūrimo taisykles ir XML schemos skirtos sąsajų validavimui. Modelis nustato tam tikras visiems registrams bendras taisykles, kuriomis remiantis galima sukurti registro integracinę sąsają.

Darbe pasiekta tokių rezultatų:

1. Išnagrinėti pasirinkti Lietuvos registrai ir jų sudėtyje esamos sukurtos integracinės sąsajos, bei išskirtos apibendrintos registrų ypatybės, įtakojančios integracinių sąsajų kūrimą,
2. Aprašytas taisyklių rinkinys pagal ontologijos sudarymo principus skirtas registrų integracinių sąsajų sukūrimui,
3. Sukurtas įrankis (XML schema), kuris leidžia validuoti sąsajas sukurtas pagal aprašytą taisyklių rinkinį. Kartu su pačiu taisyklių rinkiniu, XML schema sudaro registrų apibendrintą integracijų modelis, kurio sukūrimas ir buvo darbo tikslas.
4. Sukurtas registrų apibendrintas integracijos modelis pritaikytas pagrindiniams Lietuvos Respublikos registrams. Atliktas esamų registrų sąsajų, bei sąsajų pagal sukurtą modelį palyginimas.

Sukūrus modelį, pritaikius jį pagrindiniams Lietuvos Respublikos registrams, bei atlikus palyginimą prieita tokių išvadų:

- Sąsajos kurtos pagal registrų apibendrintą integracijos modelį pritaikytais atvejais padengia ir esamų sąsajų funkcionalumą,
- Sąsajos pagal registrų apibendrintą integracijos modelį yra griežčiau struktūrizuotos ir apibrėžtos negu esamos sąsajos,
- Esamas sąsajas galima patobulinti joms pritaikant registrų apibendrintą integracijos modelį,
- Registrų integracijos modelis pateikia bendrą struktūrą visoms pagal jį sukurtoms sąsajoms.
- Bendrą pagal registrų integracijos modelį kurtų sąsajų struktūrą galima panaudoti kuriant sąsajų XML schemų, objektų ir operacijų bendrą biblioteką. Šią biblioteką vėliau galima būtų panaudoti reikalingų integracinių sąsajų paieškos automatizavimui.

Remiantis palyginimo rezultatais, bei registrų kūrimo taisyklėmis ir nuostatais (pagal kuriuos ir kurtas modelis), prieita išvados, kad registrų apibendrintas integracijos modelis sprendžia registrų integracinių sąsajų bendrumo problemą iškeltą darbo pradžioje.

Formuojant sąsajas pagal registrų apibendrintą integracijos modelį pastebėta ir tobulintinų vietų modelyje. Modelis niekaip neapibrėžia technologijų, todėl efektyvumas efektyvumo problemos nėra sprendžiamos. Taip pat modelio pasiūlytos operacijos gali netikti specifinėms sistemoms. Modelis nedraudžia kurti ir kitokių operacijų, tačiau tokių specializuotų operacijų kūrimo taisyklių sudarymas galėtų būti tolimesnio tyrimo uždavinys.

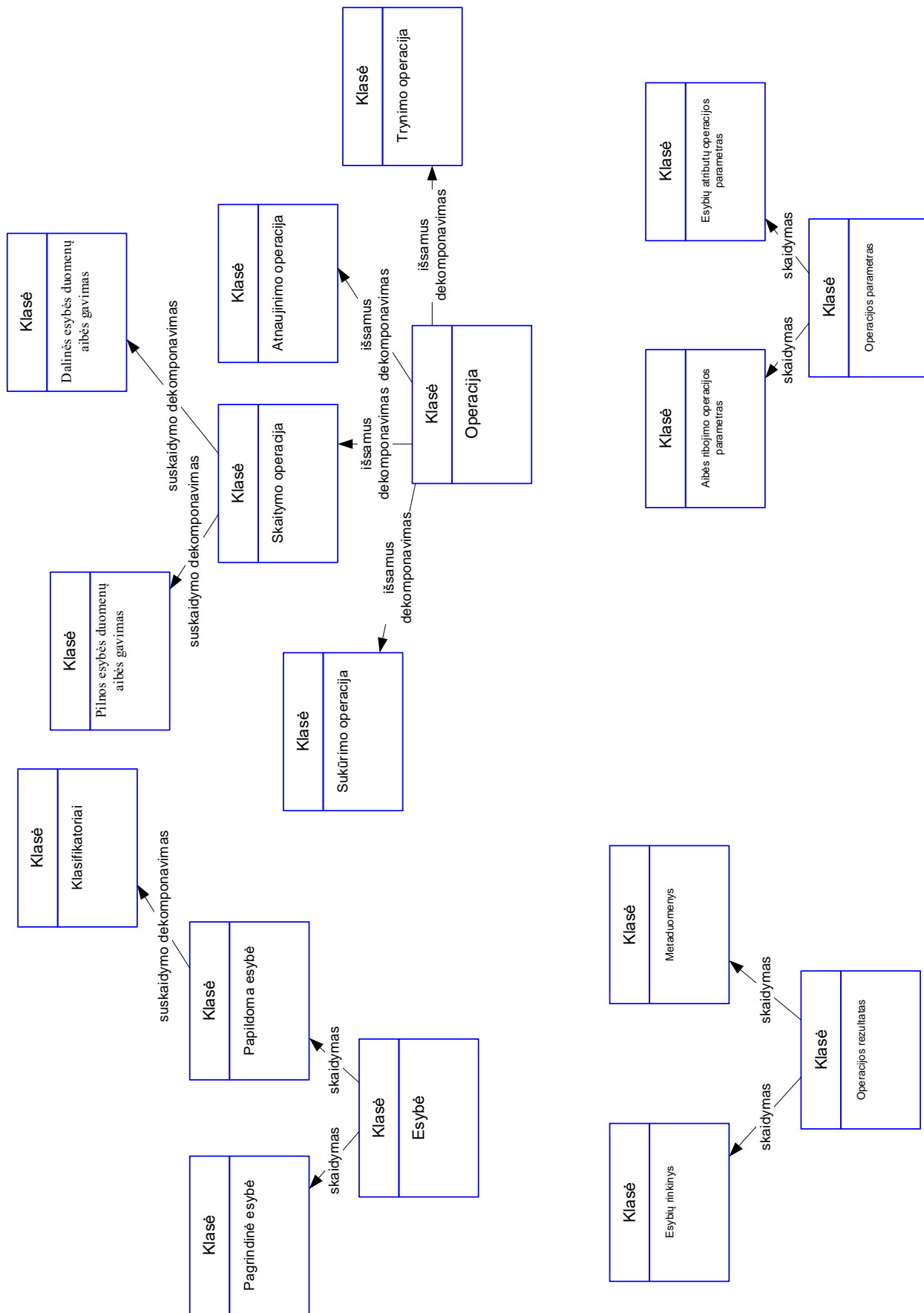
Literatūros sąrašas

- [Ari17] Ariba Inc. cXML version 1.2.034 User's Guide. 2017.
- [AFI08] ARVIDSSON, Fredrik; FLYCHT-ERIKSSON, Annika. Ontologies i. PDF). <http://www.ida.liu.se/janma/SemWeb/Slides/ontologies1.pdf>. Retrieved, 2008, 26.
- [Bag11] BAGUI, Sikha; EARP, Richard. Database design using entity-relationship diagrams. CRC Press, 2011.
- [Ber07] BERMEJO, Julita. A simplified guide to create an ontology. Madrid University, 2007.
- [Chi17] CHITUC, Claudia-Melania. XML interoperability standards for seamless communication: An analysis of industry-neutral and domain-specific initiatives. *Computers in Industry*, 2017, 92: 118-136.
- [Che76] CHEN, PETER PIN-SHAN. The Entity-Relationship Model-Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1976, 1.1: 9-36.
- [Cor05] CORCHO, Oscar, et al. Building legal ontologies with METHONTOLOGY and WebODE. In: *Law and the semantic web*. Springer Berlin Heidelberg, 2005. p. 142-157.
- [Iqb13] IQBAL, Rizwan, et al. An analysis of ontology engineering methodologies: A literature review. *Research journal of applied sciences, engineering and technology*, 2013, 6.16: 2993-3000.
- [Lam09] LAMPATHAKI, Fenareti, et al. Business to business interoperability: A current review of XML data integration standards. *Computer Standards & Interfaces*, 2009, 31.6: 1045-1055.
- [Lin03] LINTHICUM, David S. Next generation application integration: from simple information to Web services. Addison-Wesley Longman Publishing Co., Inc., 2003.
- [LSH03] LEE, Jinyoul; SIAU, Keng; HONG, Soongoo. Enterprise Integration with ERP and EAI. *Communications of the ACM*, 2003, 46.2: 54-60.
- [LRS18] Lietuvos Respublikos Seimas. Lietuvos Respublikos valstybės informacinių išteklių valdymo įstatymas. 2018.
- [LRT15a] Lietuvos Respublikos teisingumo ministerija. Lietuvos Respublikos gyventojų registro duomenų teikimo taisyklės. 2015.
- [LRT15b] Lietuvos Respublikos teisingumo ministerija. Antstolių informacinės sistemos nuostatai. 2015.

- [LRT15c] Lietuvos Respublikos teisingumo ministerija. Antstolių informacinė sistema. AIS. Techninis aprašymas (Specifikacija). 2015.
- [LRV15a] Lietuvos Respublikos Vyriausybė. Lietuvos Respublikos Adresų registro nuostatai. 2015.
- [LRV15b] Lietuvos Respublikos Vyriausybė. Nekilnojamojo turto registro nuostatai. 2015.
- [LRV16a] Lietuvos Respublikos Vyriausybė. Juridinių asmenų registro nuostatai. 2016.
- [LRV16b] Lietuvos Respublikos Vyriausybė. Įgaliojimų registro nuostatai. 2016.
- [LRV17] Lietuvos Respublikos Vyriausybė. Lietuvos Respublikos gyventojų registro nuostatai 2017.
- [Mar83] MARTIN, James. Managing the Data Base Environment. 1983.
- [Med05] MEDJAHED, Brahim; BOUGUETTAYA, Athman. Customized delivery of e-government web services. IEEE Intelligent Systems, 2005, 20.6: 77-84.
- [Nta12] Nacionalinė teismų administracija. Lietuvos teismų informacinės sistemos specifikacija. 2012.
- [Nta16] Nacionalinė teismų administracija. Lietuvos teismų informacinės sistemos nuostatai. 2016.
- [Reg18] Registrų ir valstybės informacinių sistemų registras. (2018-03-08)
- [S4i10] UAB „S4ID“. Lietuvos respublikos gyventojų registras. Specifikacija. 2012.
- [Sou08] SOUROUNI, Aikaterini-Maria, et al. Paving the way to eGovernment transformation: Interoperability registry infrastructure development. Lecture Notes in Computer Science, 2008, 5184: 340-351.
- [Shi00] SHIM, Simon SY, et al. Business-to-business e-commerce frameworks. Computer, 2000, 33.10: 40-47.
- [Tha13] THALHEIM, Bernhard. Entity-relationship modeling: foundations of database technology. Springer Science & Business Media, 2013.
- [Van03] VAN REES, Reinout. Clarity in the usage of the terms ontology, taxonomy and classification. CIB REPORT, 2003, 284.432: 1-8.
- [Vic10] Valstybės įmonė Centrinės hipotekos įstaiga. Notarine tvarka patvirtintų įgaliojimų registro specifikacija. 2010.
- [Vic16] Valstybės įmonė Centrinės hipotekos įstaiga. Įgaliojimų registro objektų registravimo ir duomenų teikimo taisyklės. 2016.
- [Vir08a] Valstybės įmonė Registrų centras. Adresų registro specifikacija. 2008.
- [Vir08b] Valstybės įmonė Registrų centras. Nekilnojamojo turto registro specifikacija. 2008.
- [Vir10] Valstybės įmonė Registrų centras. Juridinių asmenų registro specifikacija. 2010.

- [Vir13] Valstybės įmonė Registrų centras. Juridinių asmenų registro tvarkymo taisyklės. 2013.
- [VMI07] Valstybinė mokesčių inspekcija prie Lietuvos Respublikos finansų ministerijos. Integruota mokesčių informacinė sistema. IMIS. Specifikacija. 2007.
- [VMI09] Valstybinė mokesčių inspekcija prie Lietuvos Respublikos finansų ministerijos. Valstybinės mokesčių inspekcijos integruotos mokesčių informacinės sistemos nuostatai. 2009.
- [Win07] LAM, Wing (ed.). Enterprise Architecture and Integration: Methods, Implementation and Technologies: Methods, Implementation and Technologies. IGI Global, 2007.
- [Zha14] ZHANG, Chijun, et al. Distributed Service Discovery Algorithm Based on Ant Colony Algorithm. JSW, 2014, 9.1: 70-75.

1 Priedas. Registrų integracinių sąsajų ontologijos taksonomija



3 Priedas. Registrų integracinių sąsajų modelio XML schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="RegistroSasaja">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="RegistroSasajosApibudinimas" minOccurs="1" maxOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Pavadinimas" type="xs:string"/>
              <xs:element name="Registras" type="xs:string"/>
              <xs:element name="RegistroKodas" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Esybe" type="EsybesKlasesTipas" minOccurs="1"
maxOccurs="unbounded">
          <xs:unique name="UnikalusAtributoPavadinimas">
            <xs:selector xpath="//Pavadinimas"/>
            <xs:field xpath="."/>
          </xs:unique>
          <xs:unique name="UnikalusKlasifikatoriausKodas">
            <xs:selector xpath="//Kodas"/>
            <xs:field xpath="."/>
          </xs:unique>
        </xs:element>
        <xs:element name="Operacija" minOccurs="1" maxOccurs="unbounded"
type="OperacijosKlasesTipas">
          <xs:unique name="UnikalusOperacijosParametroPavadinimas">
            <xs:selector xpath="//OperacijosParametroTipas"/>
            <xs:field xpath="@Pavadinimas"/>
          </xs:unique>
          <xs:unique name="UnikalusOperacijosRezultatoPavadinimas">
            <xs:selector xpath="//OperacijosRezultatoTipas"/>
            <xs:field xpath="@Pavadinimas"/>
          </xs:unique>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
    <xs:unique name="UnikalusEsybesPavadinimas">
      <xs:selector xpath="Esybe"/>
      <xs:field xpath="@Pavadinimas"/>
    </xs:unique>
    <xs:unique name="UnikalusOperacijosPavadinimas">
      <xs:selector xpath="Operacija"/>
      <xs:field xpath="@Pavadinimas"/>
    </xs:unique>
  </xs:element>
  <xs:complexType name="EsybesKlasesTipas">
    <xs:choice>
      <xs:element name="PagrindineEsybe" type="EsybesAtributas"/>
      <xs:element name="PapildomaEsybe" type="EsybesAtributas"/>
      <xs:element name="Klasifikatorius" type="KlasifikatoriausTipas"/>
    </xs:choice>
    <xs:attribute name="Pavadinimas" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="EsybesAtributas">
    <xs:sequence>
      <xs:element name="PagrindinisAtributas" type="AtributoTipas" minOccurs="1"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

        <xs:element name="PirminisPapildomasAtributas" type="AtributoTipas"
minOccurs="1" maxOccurs="unbounded"/>
        <xs:element name="IsplestinisPapildomasAtributas" type="AtributoTipas"
minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="AtributoTipas">
    <xs:sequence>
        <xs:element name="Pavadinimas" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="Reiksme" type="xs:anyType" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="KlasifikatoriausTipas">
    <xs:sequence>
        <xs:element name="Kodas" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="Reiksme" type="xs:anyType" minOccurs="1" maxOccurs="1"/>
        <xs:element name="TevinioObjektoKodas" type="xs:anyType" minOccurs="0"
maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="OperacijosKlasesTipas">
    <xs:sequence>
        <xs:element name="OperacijosParametras" type="OperacijosParametroTipas"
minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="OperacijosRezultatas" type="OperacijosRezultatoTipas"
minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="Pavadinimas" type="xs:string" use="required"/>
    <xs:attribute name="Tikslas" type="xs:string" use="required"/>
    <xs:attribute name="Tipas" type="OperacijosTipas" use="required"/>
</xs:complexType>
<xs:simpleType name="OperacijosTipas" final="restriction">
    <xs:restriction base="xs:string">
        <xs:enumeration value="SukurimoOperacija"/>
        <xs:enumeration value="SkaitymoOperacija"/>
        <xs:enumeration value="AtnaujinimoOperacija"/>
        <xs:enumeration value="TrynimoOperacija"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="OperacijosParametroTipas">
    <xs:choice>
        <xs:element name="EsybiuAtributuOperacijosParametras" type="AtributoTipas"
minOccurs="1" maxOccurs="1"/>
        <xs:element name="AibesRibojimoOperacijosParametras"
type="AibesRibojimoOperacijosParametroTipas" minOccurs="1" maxOccurs="1"/>
    </xs:choice>
    <xs:attribute name="Pavadinimas" type="xs:string" use="required"/>
</xs:complexType>
<xs:simpleType name="AibesRibojimoOperacijosParametroTipas" final="restriction">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Pagrindinis"/>
        <xs:enumeration value="Pirminis"/>
        <xs:enumeration value="Isplestinis"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="OperacijosRezultatoTipas">
    <xs:choice>
        <xs:element name="EsybiuRinkynys" type="EsybesKlasesTipas" minOccurs="1"
maxOccurs="1"/>
        <xs:element name="Metadomuo" type="AtributoTipas" minOccurs="1"
maxOccurs="1"/>
    </xs:choice>

```

```
    <xs:attribute name="Pavadinimas" type="xs:string" use="required"/>
  </xs:complexType>
</xs:schema>
```

4 Priedas. Adresų registro atributų aibė pagal sukurtą modelį

- Adresas
 - Pagrindiniai atributai
 - Identifikatorius
 - Savivaldybės kodas
 - Savivaldybės pavadinimas
 - Seniūnijos kodas
 - Seniūnijos pavadinimas
 - Gyvenamosios vietovės kodas
 - Gyvenamosios vietovės pavadinimas
 - Gatvės kodas
 - Gatvės pavadinimas
 - Žemės sklypo ar pastato numeris
 - Korpuso numeris
 - Patalpos numeris pastate ar korpuse
 - Pavadinimas
 - Pirminiai papildomi atributai
 - Pašto kodas
 - Virtualusis adresas
 - Unikalus numeris (jei registruota Nekilnojamojo turto registre)
 - Priklausomybė kitam registro objektui
 - Išplėstiniai papildomi atributai
 - Dokumento, kuriuo remiantis įregistruojamas, išregistruojamas ar keičiami registro objekto duomenys rūšis, numeris, data ir priėmusios institucijos pavadinimas (Gali būti skirstoma ir į keletą atributų, tačiau nuostatuose tai nepabrėžiama)
 - Įregistravimo data
 - Išregistravimo data
 - Duomenų įrašymo data
 - Keitimo data
 - Vietos taškas LKS-94
 - Universaliųjų pašto paslaugų teikimo vietos pavadinimas
 - Civilinės saugos ir gelbėjimo duomenys
 - Socialiniai, kultūriniai, ekonominiai duomenys
- Administracinis vienetas

- Pagrindiniai atributai
 - Identifikatorius
 - Vardas (apskritis, savivaldybės ar seniūnijos, priklausomai nuo tipo) kilmininko linksniu
 - Administracinio vieneto tipas
- Pirminiai papildomi atributai
 - Centro kodas (jei ne seniūnija)
 - Centro pavadinimas (jei ne seniūnija)
 - Apskritis, kurios teritorijoje yra savivaldybė kodas (jei savivaldybė)
 - Apskritis, kurios teritorijoje yra savivaldybė pavadinimas (jei savivaldybė)
 - Gyvenamosios vietovės, kurioje yra seniūnijos buveinė kodas (jei seniūnija)
 - Gyvenamosios vietovės, kurioje yra seniūnijos buveinė pavadinimas (jei seniūnija)
 - Savivaldybės, kurios teritorijoje yra seniūnija kodas (jei seniūnija),
 - Savivaldybės, kurios teritorijoje yra seniūnija pavadinimas (jei seniūnija)
- Išplėstiniai papildomi atributai
 - Dokumento, kuriuo remiantis įregistruojamas, išregistruojamas ar keičiami registro objekto duomenys rūšis, numeris, data ir priėmusios institucijos pavadinimas (Gali būti skirstoma ir į keletą atributų, tačiau nuostatuose tai nepabrėžiama)
 - Įregistravimo data
 - Išregistravimo data
 - Duomenų įrašymo data
 - Keitimo data
 - Ribos LKS-94 formatu
 - Plotas
 - Civilinės saugos ir gelbėjimo duomenys
 - Socialiniai, kultūriniai, ekonominiai duomenys
- Gyvenamoji vietovė
 - Pagrindiniai atributai
 - Identifikatorius
 - Vardas vardininko forma

- Vardas kilmininko forma
 - Gyvenamosios vietos tipas
- Pirminiai papildomi atributai
 - Savivaldybės, kurios teritorijoje yra gyvenamoji vietovė kodas
 - Savivaldybės, kurios teritorijoje yra gyvenamoji vietovė pavadinimas
 - Seniūnijos, kurios teritorijoje yra gyvenamoji vietovė kodas
 - Seniūnijos, kurios teritorijoje yra gyvenamoji vietovė pavadinimas
- Išplėstiniai papildomi atributai
 - Dokumento, kuriuo remiantis įregistruojamas, išregistruojamas ar keičiami registro objekto duomenys rūšis, numeris, data ir priėmusios institucijos pavadinimas (Gali būti skirstoma ir į keletą atributų, tačiau nuostatuose tai nepabrėžiama)
 - Įregistravimo data
 - Išregistravimo data
 - Duomenų įrašymo data
 - Keitimo data
 - Gyvenamosios vietovės vardo kirčiuotė
 - Žyma apie gyvenamajai vietai suteiktą kurorto statusą
 - Ribos LKS-94 formatu
 - Civilinės saugos ir gelbėjimo duomenys
 - Socialiniai, kultūriniai, ekonominiai duomenys
- Gatvė
 - Pagrindiniai atributai
 - Identifikatorius
 - Vardas
 - Tipas
 - Pirminiai papildomi atributai
 - Gyvenamosios vietovės, kurioje yra gatvė kodas
 - Gyvenamosios vietovės, kurioje yra gatvė pavadinimas
 - Išplėstiniai papildomi atributai
 - Dokumento, kuriuo remiantis įregistruojamas, išregistruojamas ar keičiami registro objekto duomenys rūšis, numeris, data ir priėmusios institucijos pavadinimas (Gali būti skirstoma ir į keletą atributų, tačiau nuostatuose tai nepabrėžiama)
 - Įregistravimo data

- Išregistravimo data
- Duomenų įrašymo data
- Keitimo data
- Gatvės ašinė linija LKS-94
- Civilinės saugos ir gelbėjimo duomenys
- Socialiniai, kultūriniai, ekonominiai duomenys

5 Priedas. Gyventojų registro atributų aibė pagal sukurtą modelį

- Asmuo
 - Pagrindiniai atributai
 - Asmens kodas (identifikatorius)
 - Vardas (vardai)
 - Pavardė (pavardės)
 - Lytis
 - Gimimo data
 - Pilietybė (pilietybės)
 - Gimimo vietos šalis
 - Gyvenamoji vietovė
 - Mirties data
 - Tautybė
 - Pirminiai papildomi atributai
 - Pilietybės įgijimo data
 - Pilietybės netekimo data
 - Atvykimo į gyvenamąją vietą data
 - Išvykimo vieta (jei išvykęs)
 - Išvykimo data (jei išvykęs)
 - Savivaldybė, kurioje gyvena (jei neturi gyvenamosios vietos)
 - Tėvų, vaikų ir sutuoktinių asmens kodai
 - Asmens kontaktiniai duomenys
 - Išplėstiniai papildomi atributai
 - Veido atvaizdas
 - Pirštų antspaudai
 - Parašas
 - Asmens dokumentų duomenys
 - Prašymo išduoti asmens dokumentą duomenys
 - Duomenys apie asmens neveiknumą tam tikroje srityje ar veiksnio tam tikroje srityje apribojimą
 - Civilinės būklės aktų įrašų duomenys
 - Registro objekto įregistravimo data;
 - Registro objekto duomenų keitimo data.

6 Priedas. Įgaliojimų registro atributų aibė pagal sukurtą modelį

- Įgaliojimas
 - Pagrindiniai atributai
 - Identifikavimo kodas (Identifikatorius)
 - Įgaliojimo sudarymo data
 - Įgaliojimo sudarymo vieta
 - Įgaliotojo duomenys (Pilna aibė pateikiama [LRV16b])
 - Įgaliotinio duomenys (Pilna aibė pateikiama [LRV16b])
 - Įgaliojimo terminas
 - Įgaliojimo pasibaigimo pagrindas
 - Įgaliojimo pasibaigimo data
 - Pirminiai papildomi atributai
 - Įgaliojimą patvirtinusio asmens duomenys (vardas, pavardė arba pavadinimas)
 - Pastabos apie įgaliotinio teisę perįgaluoti kitą asmenį atlikti veiksmus, kitos pastabos, tikslinančios kitus šiame punkte nurodytus duomenis (jeigu jos pateiktos)
 - Išplėstiniai papildomi atributai
 - Įgaliojimo įregistravimo registre data ir laikas
 - Įgaliojimo duomenų keitimo pagrindas
 - Įgaliojimo duomenų keitimo data
 - Įgaliojimo duomenų keitimo registre data ir laikas
 - Įgaliojimo išregistravimo iš registro data ir laikas
 - Įgaliojimo skaitmeninė kopija
- IT įgaliojimas
 - Pagrindiniai atributai
 - Identifikavimo kodas (Identifikatorius)
 - Įgaliojimo sudarymo data ir laikas
 - Įgaliojimo sudarymo vieta
 - Įgaliotojo duomenys (Pilna aibė pateikiama [LRV16b])
 - Įgaliotinio duomenys (Pilna aibė pateikiama [LRV16b])
 - Įgaliojimo terminas
 - Įgaliojimo pasibaigimo pagrindas
 - Įgaliojimo pasibaigimo data
 - Pirminiai papildomi atributai

- Įgaliøjime nurodyto pavedimo turinys.
- Išplėstiniai papildomi atributai
 - Įgaliøjimo įregistravimo registre data ir laikas
 - Įgaliøjimo išregistravimo iš registro data ir laikas

7 Priedas. Nekilnojamojo turto registro atributų aibė pagal sukurtą modelį

- Nekilnojamasis turtas
 - Pagrindiniai atributai
 - Registro numeris (Identifikatorius)
 - Nekilnojamojo daikto adresas
 - Pavadinimas
 - Kadastro vietovė
 - Kadastro blokas
 - Žemės sklypo kadastro numeris
 - Unikalus sklypo numeris
 - Unikalus statinio numeris
 - Unikalus buto arba patalpos numeris
 - Nekilnojamąjį daiktą apibūdinantys kadastro duomenys: pagrindinė tikslinė naudojimo paskirtis, plotas, tūris, kiti geometriniai parametrai, vertės, duomenų nustatymo data ir kiti duomenys, nustatyti Lietuvos Respublikos nekilnojamojo turto kadastro įstatymo 6 straipsnyje
 - Pirminiai papildomi atributai
 - Nekilnojamojo turto kadastro žemėlapių duomenys ir kitų nekilnojamųjų daiktų planai
 - Energinio naudingumo duomenys
 - Investicinio fondo dalyvių nuosavybės teisės duomenys
 - Išplėstiniai papildomi atributai
 - Dokumento, kuriuo remiantis objektas įregistruojamas, duomenys
 - Dokumento, kuriuo remiantis objektas išregistruojamas, duomenys
 - Įregistravimo data
 - Išregistravimo data
 - Nekilnojamųjų daiktų kadastro duomenų archyviniai duomenys
- Daiktinės teisės
 - Pagrindiniai atributai
 - Registro numeris (Identifikatorius)
 - Pavadinimas
 - Duomenys, apibūdinantys daiktines teises, šių teisių suvaržymus ir su nekilnojamaisiais daiktais, daiktinių teisių į juos suvaržymais bei daiktinėmis teisėmis susijusius juridinius faktus

- Daiktinių teisių turėtojų duomenys
 - Pirminiai papildomi atributai
 - Kitų su daiktinėmis teisėmis susijusių asmenų duomenys
 - Daiktinių teisių, tų teisių suvaržymų ir juridinių faktų duomenys (sandorio suma, specialiosios sandorio ar administraciniu aktu nustatytos sąlygos, sandorio, administracinio akto identifikavimo kodas, data, terminai ir panašiai)
 - Išplėstiniai papildomi atributai
 - Dokumento, kuriuo remiantis objektas įregistruojamas, duomenys
 - Dokumento, kuriuo remiantis objektas išregistruojamas, duomenys
 - Įregistravimo data
 - Išregistravimo data
 - Nuosavybės ir kitų daiktinių teisių, tų teisių suvaržymų ir juridinių faktų archyviniai duomenys
- Sutartys
 - Pagrindiniai atributai
 - Registro numeris (Identifikatorius)
 - Sutarties rūšis
 - Sudarymo data
 - Kaina
 - Sutarties šalys
 - Perleidžiamo turto sudėtis
 - Pirminiai papildomi atributai
 - Sutarties priedų duomenys
 - Išplėstiniai papildomi atributai
 - Dokumento, kuriuo remiantis objektas įregistruojamas, duomenys
 - Dokumento, kuriuo remiantis objektas išregistruojamas, duomenys
 - Įregistravimo data
 - Išregistravimo data