



Kauno technologijos universitetas

Informatikos fakultetas

Šablonais grindžiamas koncepcinio duomenų modelio kūrimo metodas

Baigiamasis magistro studijų projektas

Vytautas Volungevičius

Projekto autorius

Doc. Dr. Rita Butkinė

Vadovė

Kaunas, 2020



Kauno technologijos universitetas

Informatikos fakultetas

Šablonais grindžiamas koncepcinio duomenų modelio kūrimo metodas

Baigiamasis magistro studijų projektas
Informacinių sistemų inžinerija (6211BX009)

Vytautas Volungevičius

Projekto autorius

doc. dr. Rita Butkienė

Vadovė

doc. dr. Vytautas Rudžionis

Recenzentas

KAUNAS, 2020



Kauno technologijos universitetas

Informatikos fakultetas

Vytautas Volungevičius

Šablonais grindžiamas koncepcinio duomenų modelio kūrimo metodas

Akademinio sąžiningumo deklaracija

Patvirtinu, kad mano, Vytautas Volungevičiaus, baigiamasis projektas tema „Šablonais grindžiamas koncepcinio duomenų modelio kūrimo metodas“ yra parašytas visiškai savarankiškai ir visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

Vytautas Volungevičius

(vardą ir pavardę įrašyti ranka)

(parašas)

Vytautas Volungevičius. Šablonais grindžiamas koncepcinio duomenų modelio kūrimo metodas. Magistro baigiamasis projektas vadovė doc. dr. Rita Butkienė; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informacijos sistemos, Informatikos mokslai.

Reikšminiai žodžiai: duomenų modelių šablonai, CASE įrankiai, duomenų bazės kūrimas, koncepcinio duomenų modelio kūrimo metodai, dalinai automatizuotas duomenų modelio kūrimas.

Kaunas, 2020. 129 p.

Santrauka

Per pastaruosius kelis dešimtmečius vis daugiau ir daugiau organizacijų savo verslo procesus perkelia į informacinę sistemą. Informacinėse sistemose ne tik vykdoma didelė dalis verslo procesų tačiau taip pat saugoma ir labai didelė dalis organizacijos duomenų. Duomenys informacinėse sistemose yra saugojami duomenų bazėse, kurių projektavimas yra sudėtingas ir lėtas procesas.

Nors per daugelį metų jau yra sukaupta nemažai patirties įvairių dalykinių sričių koncepcinių modelių kūrimo, tačiau ši patirtis ir žinios nėra suskaitmenintos, nėra galimybės jomis pasinaudoti kuriant koncepcinį duomenų modelį.

Šio darbo tikslas – sudaryti sąlygas, kurios leistų pagreitinti koncepcinio duomenų modelio kūrimo procesą. Šiam tikslui pasiekti buvo pasirinkta panaudoti šablonai grindžiamą koncepcinio modelio kūrimo metodą. Analizės etape buvo nustatytas esamo tipinio koncepcinio modelio kūrimo metodo trūkumas, ir projektavimo etape šis metodas buvo praplėstas panaudojant duomenų modelių šablonų biblioteką bei automatizuotą įrankį leidžiantį šablonus įterpti į kuriamą modelį.

Eksperimento metu buvo palygintas laikas reikalingas sukurti duomenų modelį naudojant tipinį koncepcinio modelio kūrimo procesą, bei praplėsta duomenų modelių šablonų biblioteka. Remiantis eksperimento rezultatais pavyko nustatyti, kad naudojantis įrankiu bei šablonų biblioteka galima pagreitinti duomenų modelio kūrimo procesą nuo 37,5% – 86,84%. Sprendimo efektyvumas priklauso nuo to, ar projektuotojas yra susipažinęs su duomenų modelių šablonais. Remiantis apklausos rezultatais šis naujas modelio kūrimo būdas sulaukė teigiamų rezultatų. Modeliai gaunami naudojant šablonus yra lankstesni, kas užtikrina lengvesnę sistemos plėtimą ateityje.

Vytautas Volungevičius. Pattern-Based Method of Conceptual Data Model Development. Master's Final Degree Project supervisor doc. dr. Rita Butkienė; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Information Systems, Computing.

Keywords: Data model patterns, CASE tool. Database development, Conceptual data model development methods, Semi-automated conceptual data model development.

Kaunas, 2020. 129 pages.

Summary

Over the past few decades, more and more organizations shifting their business processes to the information systems. Not only a large portion of the business process runs in an information system, but a considerable part of an organization's data is stored there also. Data in organization information systems is stored in databases, the design of which is a slow and complicated process.

Although much experience has been gained over the years in the development of conceptual models in various subject areas, this experience and knowledge are not digitized, there is no possibility to use it in the development of a conceptual data model.

This work aims to create conditions that would speed up the process of conceptual data model development. To achieve this objective, it was chosen to use a pattern-based conceptual data model development method. In the analysis phase, a shortcoming of the existing typical conceptual model development method was identified. In the design phase, this method was extended to use a data model pattern library and an automated tool that allows patterns to be inserted into the model that is being developed.

During the experiment, the times required to create a data model using a typical conceptual model development process and an extended, using data model patterns library, conceptual model development process were compared. Based on the results of the experiment, it was found that using the tool and pattern library can speed up the data model development process from 37.5% to 86.84%. The effectiveness of the solution depends on whether the designer is familiar with data model patterns or not. According to the result of the survey, this new way of model development has yielded positive results. The models developed using patterns are flexible, which ensures easier expansion of the system in the future.

Turinys

Lentelių sąrašas	9
Paveikslų sąrašas	10
Santrumpų ir terminų sąrašas	12
Įvadas.....	13
1. Probleminės srities analizė.....	15
1.1. Analizės tikslas	15
1.2. Tyrimo objektas, sritis ir problema	15
1.3. Tyrimo objekto analizė.....	15
1.3.1. Trys duomenų bazės lygiai.....	17
1.3.2. Konceptinis duomenų modelis.....	17
1.3.3. Tipinis konceptinio duomenų modelio kūrimo procesas MagicDraw įrankyje.....	18
1.3.4. Duomenų modelių šablonų įvairovė.....	20
1.3.5. Duomenų modelių abstrakcijos lygiai	26
1.4. Tyrimo objekto naudotojų analizė	28
1.5. Esamų problemos sprendimo metodų analizė (Lietuvos ir tarptautiniu mastu)	29
1.6. Darbo tikslas, uždaviniai ir siekiami privalumai	32
1.7. Siekiamo sprendimo apibrėžimas	32
1.8. Analizės išvados	32
2. Duomenų modelių šablonų sprendimo reikalavimų specifikacija ir projektas, formalus aprašas	34
2.1. Reikalavimų specifikacija.....	34
2.1.1. Sistemos aprašas	34
2.2. Sistemos reikalavimai.....	34
2.2.1. Nefunkciniai reikalavimai	34
2.2.2. Panaudojimo atvejai	34
2.3. Dalykinės srities modelis.....	46
2.4. Formalus sprendimo aprašas	49
2.4.1. Sprendimo procesai	49
2.4.2. Pasiūlyti procesai.....	49
2.5. Reikalavimų apibendrinimas	53
3. Duomenų modelių šablonų sprendimo arba eksperimentinės realizacijos projektas	55
3.1. Realizacijos technologijos	55
3.2. Šablonų bibliotekos sudarymas	55
3.3. UML metamodelio praplėtimas.....	56
3.4. Sistemos loginė architektūra.....	56
3.5. Sistemos klasių diagrama	57
3.6. Sistemos komponentų diagrama.....	58
3.7. Sistemos diegimo diagrama.....	59
4. Sprendimo realizacija ir testavimas.....	60
4.1. Sprendimo realizacijos ir veikimo aprašas	60
4.2. Testavimo modelis, duomenys, rezultatai	63
4.2.1. Testavimo duomenys.....	63
4.2.2. Testuojamo modelio kūrimo duomenys	64

4.2.3. Testavimo rezultatai	69
5. Šablonais grindžiamas koncepcinio duomenų modelio kūrimo metodo eksperimentinis tyrimas	71
5.1.Eksperimento planas	71
5.2.Eksperimento rezultatai	72
5.3.Sprendimo veikimo ir savybių analizė, kokybės kriterijų įvertinimas	77
5.4.Sprendimo taikymo rekomendacijos	79
5.5.Ateities perspektyvos.....	80
Išvados	81
Literatūros sąrašas	82
Priedai.....	84
1priedas. Eksperimento vykdymo aprašas.....	84
2priedas. Apklaustos anketa	92
3priedas. Straipsnis.....	93
4priedas. Šablonai.....	93
5.1.Activities and procedures	93
5.2.Actual asset usage.....	94
5.3.Addresses.....	95
5.4.Asset classes and attributes.....	96
5.5.Authorship	97
5.6.Catalogue item types	98
5.7.Complete Assets Model.....	99
5.8.Contract role entity	100
5.9.Contracts.....	101
5.10. Deliveries.....	102
5.11. Divided Activities.....	104
5.12. Document type structure.....	104
5.13. Documenting products.....	105
5.14. Employees	106
5.15. Employment	107
5.16. Employment(Revised).....	108
5.17. Equipment and materials	109
5.18. Geographic location.....	109
5.19. Inventory.....	110
5.20. Kinds of contracts.....	111
5.21. Land parcel	111
5.22. Maintance Work Orders	112
5.23. Organization structure	113
5.24. Organizations.....	113
5.25. Parties	114
5.26. Parts Inventory.....	115
5.27. Position title.....	116
5.28. Positions	117
5.29. Product Categories.....	118
5.30. Production orders.....	119

5.31.	Products and Product types.....	120
5.32.	Products and services	121
5.33.	Purchase orders.....	122
5.34.	Reporting relationship	123
5.35.	Roles and Assignments.....	124
5.36.	Sales orders.....	125
5.37.	Site.....	126
5.38.	Time sheets	127
5.39.	Work orders	128
5.40.	Party contact mechanism.....	129

Lentelių sąrašas

Lentelė 1 Problemos su kuriomis susiduria naudotojai, bei kaip viskas vyksta dabar.....	28
Lentelė 2 Konceptinio modelio kūrimo automatizavimo metodų palyginimas.....	31
Lentelė 3 Panaudojimo atvejo „Ieškoti pagal raktinį žodį“	35
Lentelė 4 Panaudojimo atvejo „Peržiūrėti šabloną“	36
Lentelė 5 Panaudojimo atvejo „Ieškoti šablonų pagal esybę“	37
Lentelė 6 Panaudojimo atvejo „Pasirinkti šablonus“	38
Lentelė 7 Panaudojimo atvejo „Įterpti pasirinktus“	39
Lentelė 8 Panaudojimo atvejo „Pasirinkti esybes“	41
Lentelė 9 Panaudojimo atvejo „Įterpti pasirinktas esybes“	41
Lentelė 10 Panaudojimo atvejo „Įterpti šabloną“	43
Lentelė 11 Panaudojimo atvejo „Redaguoti šabloną“	44
Lentelė 12 Klasių diagramos metamodelio elementų aprašymas.....	48
Lentelė 13 Šablonų bibliotekos elementų aprašymas.....	48
Lentelė 14 Naudotojo sukurtų elementų aprašymas.....	48
Lentelė 15 UML metamodelio praplėstų elementų aprašymas	49
Lentelė 16 Klasių aprašas	58
Lentelė 17 Detalizuoti bendri eksperimento modelio kūrimo efektyvumo rodikliai dalyvių grupėse	73
Lentelė 18 Bendras detalizuotas visų eksperimento dalyvių rezultatas	74
Lentelė 19 Pirmosios eksperimento dalies detalizuotas rezultatas dalyvių grupėse	75
Lentelė 20 Antrosios eksperimento dalies detalizuotas rezultatas dalyvių grupėse	76

Paveikslų sąrašas

1 pav.	Duomenų bazės kūrimo procesas pagal Graeme Simsion.....	16
2 pav.	Tipinis duomenų modelio kūrimo procesas	19
3 pav.	Subjektų šablonas pagal David C. Hay	20
4 pav.	Subjektų šablonas pagal Len Silverston.....	21
5 pav.	Pozicijos šablonas pagal David C. Hay.....	22
6 pav.	Pozicijos šablonas pagal Len Silverston	23
7 pav.	Pozicijos šablonas pagal Michael Blaha	23
8 pav.	Draudimo sutarčių modelis pagal Graeme Simsion.....	25
9 pav.	Nuomos sutarčių modelis, pagrįstas draudimo sutarčių modeliu pagal Graeme Simsion	26
10 pav.	Universalus duomenų modelio šablonas pagal David C. Hay	27
11 pav.	Šablonų abstrakcijos lygiai pagal David C. Hay	27
12 pav.	EIPW įrankio architektūra.....	31
13 pav.	Panaudojimo atvejų diagrama	35
14 pav.	Ieškoti pagal raktinį žodį	36
15 pav.	Peržiūrėti šabloną	37
16 pav.	Ieškoti šablonų pagal esybę.....	38
17 pav.	Pasirinkti šablonus.....	39
18 pav.	Įterpti pasirinktus.....	40
19 pav.	Pasirinkti esybes	41
20 pav.	Įterpti pasirinktas esybes	43
21 pav.	Įterpti šabloną	44
22 pav.	Redaguoti šabloną	46
23 pav.	Dalykinės srities modelis	47
24 pav.	Duomenų modelio kūrimo procesas naudojantis šablonų biblioteka.....	50
25 pav.	Duomenų modelio papildymo procesas naudojantis šablonų biblioteka	51
26 pav.	Tipinio duomenų modelio kūrimo proceso praplėtimas	53
27 pav.	Sukurti stereotipai	56
28 pav.	Sistemos loginė architektūra	57
29 pav.	Sistemos klasių diagrama	57
30 pav.	Sistemos komponentų diagrama.....	58
31 pav.	Sistemos diegimo diagrama	59
32 pav.	Duomenų modelių šablono bibliotekos atidarymas	60
33 pav.	Šablonų pasirinkimas bibliotekoje	61
34 pav.	Rezultatas gautas įvykdžius šablono įterpimą.....	61
35 pav.	„Related library elements“ funkcija	62
36 pav.	Susijusių bibliotekos elementų pasirinkimas	63
37 pav.	Testavimui skirtas modelis.....	64
38 pav.	„Parties“ šablonas.....	65
39 pav.	„Organization structure“ šablonas.....	65
40 pav.	„Employees“ šablonas.....	66
41 pav.	„Employment“ šablonas	66
42 pav.	„Sites“ šablonas	67
43 pav.	„Products and services“ šablonas	68

44 pav. „Positions“ šablonas	69
45 pav. Testavimo metu gautas modelis	70
46 pav. Bendri eksperimentinio modelio kūrimo efektyvumo rodikliai dalyvių grupėse.....	73
47 pav. Bendri visų eksperimento dalyvių rezultatai	74
48 pav. Pirmosios eksperimento dalies efektyvumo rodikliai dalyvių grupėse	75
49 pav. Antrosios eksperimento dalies efektyvumo rodikliai dalyvių grupėse	76
50 pav. Metodo ir įrankio apklausos įvertinimai	78
51 pav. Apsimokymo laiko ir rekomendacijų apklausos įvertinimai.....	79

Santrumpų ir terminų sąrašas

Santrumpos:

ER – esybių ryšiai (angl. *Entity relationship*)

DDL – duomenų aprašymo kalba (angl. *Data definition language*)

EIR – esybių egzempliorių saugykla (angl. *Entity Instance Repository*)

RIR – ryšių egzempliorių saugykla (angl. *Relationship Instance Repository*)

EIP – esybių egzempliorių šablonai (angl. *Entity Instance Repository*)

NL – natūrali kalba (angl. *Natural language*)

DBMS – duomenų bazių valdymo sistema (angl. *Database management system*)

UoD – diskurso visata (angl. *Universe of Discourse*)

Ivadas

Studijų programa: Informacinių sistemų inžinerija

Siekiant paspartinti darbą, turėti vieną vietą, kurioje būtų galima talpinti visus organizacijos duomenis, vis daugiau įmonių savo verslo procesus perkelia į informacines sistemas. Organizacijoms plečiantis ar keičiant savo verslo procesus reikia keisti arba plėsti informacines sistemas.

Darbo problematika ir aktualumas

Poreikis kurti naujas ar tobulinti senas sistemos naudojančias duomenų bazes išlieka. Duomenų bazės projektas yra labai svarbus abejais atžvilgiais, kadangi nuo šio projekto priklauso ar sistema bus lengvai palaikoma, paruošta pokyčiams. Tam, kad duomenų modelis būtų lengvai suprantamas dažnai yra kuriamos lentelės su aiškiais specifiniais, dalykinės srities objektų pavadinimais. Duomenų bazės projekto kūrimo procesas yra ilgas ir sudėtingas. Per pastaruosius kelis dešimtmečius yra sukaupta nemažai informacijos, parengta duomenų modelių šablonų apie įvairių dalykinių sričių kompiuterizavimą, tačiau šiuo metu nėra galimybės jomis pasinaudoti kuriant konkrečias ar plečiant esamas sistemas. Šios žinios nėra susistemintos, nėra parengto šablonų rinkinio bei įrankio, kuriuo pasinaudojant būtų galima dalinai automatizuoti koncepcinio duomenų modelio kūrimo procesą.

Darbo tikslas ir uždaviniai

Pagrindinis šio tyrimo tikslas yra sukurti sąlygas, kurios leistų palengvinti bei pagreitinti duomenų bazės kūrimo procesą, kuriant naujas sistemas ar tobulinant senas. Šiam tikslui pasiekti buvo suformuoti tokie uždaviniai:

1. Apibendrinti įvairių autorių patirtį;
2. Surinkti ir suskaitmeninti įvairių dalykinių sričių tipinius duomenų modelius;
3. Parengti automatizuotą metodiką, leidžiančią šiuos modelius pritaikyti konkrečiam atvejui;
4. Sukurti duomenų modelių šablonų biblioteką ir plėtinį kuris ją panaudotų;
5. Realizuoti "Duomenų modelių šablonų" plėtinį;
6. Eksperimentiškai ištirti sukurtą sprendimą.

Darbo rezultatai ir jų svarba

Šio darbo rezultatas yra paspartintas duomenų bazės kūrimo procesas, įnešti panašumai į įvairių dalykinių sričių, ar specifinių dalykinių sričių duomenų bazių projektus. Panaudojant šablonus duomenų bazės pagrindinė struktūra išlieka panaši skirtingose dalykinės srities sistemose, todėl tai palengvina darbą pasikeitus architektui, kadangi nereikia detaliai analizuoti kiekvienos lentelės ir ryšio. Tokių duomenų bazių struktūra, kurios suskurtos šablonų pagrindu, išlieka panaši ir yra žinoma daugeliui specialistų, kurie yra susipažinę su duomenų modelių šablonais. Sukurtos sąlygos leidžiančios palengvinti analitikų darbą atliekant naujos jiems nežinomos dalykinės srities reikalavimų analizę.

Darbo struktūra

Darbe yra apibrėžiamas tyrimo objektas, tyrimo sritis bei tyrimo problema. Pirmajame skyriuje yra analizuojamas duomenų bazės kūrimo procesas. Pirmojo skyriaus poskyriuose pateikiama tipinio duomenų bazės kūrimo proceso diagrama. Trumpai aprašomas koncepcinis duomenų modelis, 1.3.2 skyrelyje įvardinami sunkumai su kuriais susiduriama kuriant koncepcinį modelį. Duomenų modelių šablonų įvairovė skyrelyje kalbama apie įvairius duomenų modelių šablonus. Poskyryje 1.7 pateikiami keli sprendimo būdai ir 1.8 poskyryje pateikiamos analizės išvados. Antrajame skyriuje pateikiama sprendimo reikalavimų specifikacija. Duomenų modelių šablonų sprendimo eksperimentinės realizacijos projekto skyriuje pateikiamas įskiepio realizacijos projektas. Ketvirtajame skyriuje pateikiami sukurto sprendimo testavimo duomenys bei rezultatai. Penktajame skyriuje pateikiamas eksperimento tyrimo planas, bei gauti eksperimentinio tyrimo rezultatai. Šiame skyriuje taip pat pateikiamos sprendimo taikymo rekomendacijos bei ateities perspektyvos. Darbo pabaigoje pateikiamos išvados.

1. Probleminės srities analizė

1.1. Analizės tikslas

Pagal pasirinktą temą „Šablonais grindžiamas koncepcinio duomenų modelio kūrimas“ yra siekiama išanalizuoti įvairių dalykinių sričių modelius. Kadangi jau yra sukaupta nemažai informacijos apie įvairias dalykines sritis, bus pasinaudojama įvairių autorių sukauptomis žiniomis, siekiama jas susisteminti. Analizuojant literatūrą tai pat bus atsižvelgiama į įvairius artefaktus minimus joje. Taip siekiama geriau pažinti tyrimo sritį, išvelgti problemas bei pasiūlyti galimus sprendimus šioms problemoms.

1.2. Tyrimo objektas, sritis ir problema

Tyrimo problema: Poreikis kurti naujas duomenų bazes naudojančias sistemas ar tobulinti esančias išlieka ir šiomis dienomis, net tuomet kai jų yra sukurta tikrai labai daug. Įprastai sistemos turi kažkokių bendrų bruožų pvz.: reikalinga darbuotojų valdymo dalis, finansų valdymo dalis, ar pačios organizacijos struktūra. Šias bendras dalis projektuotojams tenka kurti vis iš naujo nors dažnai galima panaudoti vieną bendrą šabloną. Patyrę analitikai vykdant reikalavimų specifikacijos surinkimo procesą iš karto pastebi, kokio modelio jiems reikės, ir dalis jų tokius modelius, kuriuos gali perpanaudoti, jau turi. Tačiau nepatyrę asmenys susiduria su įvairiais iššūkiais. Problema, kad nėra surinkto tokių duomenų modelių šablonų rinkinio, kuriuo galėtų naudotis įvairaus lygio analitikai. Taip pat nėra sukurto automatizuoto įrankio, kuriuo naudojantis būtų galima taikyti šablonus iš surinkto jų rinkinio, kas galėtų paspartinti sistemos kūrimą. Kuriant naujas sistemas taip pat dažnai kuriama ir nauja duomenų bazė nuo pagrindų, tačiau tai nėra tikslinga, kadangi jau yra atlikta daug tyrimų, surinkta daug medžiagos apie įvairias dalykines sritis ir duomenų bazės „branduolį“ galima sudaryti pasinaudojant šablonais.

Tyrimo objektas: Koncepcinio duomenų modelio kūrimo procesas įvairioms dalykinėms sritims.

Tyrimo sritis: Koncepcinio duomenų modelio kūrimo metodai, įrankiai, technologijos, dalykinės srities objektų modelių notacijos.

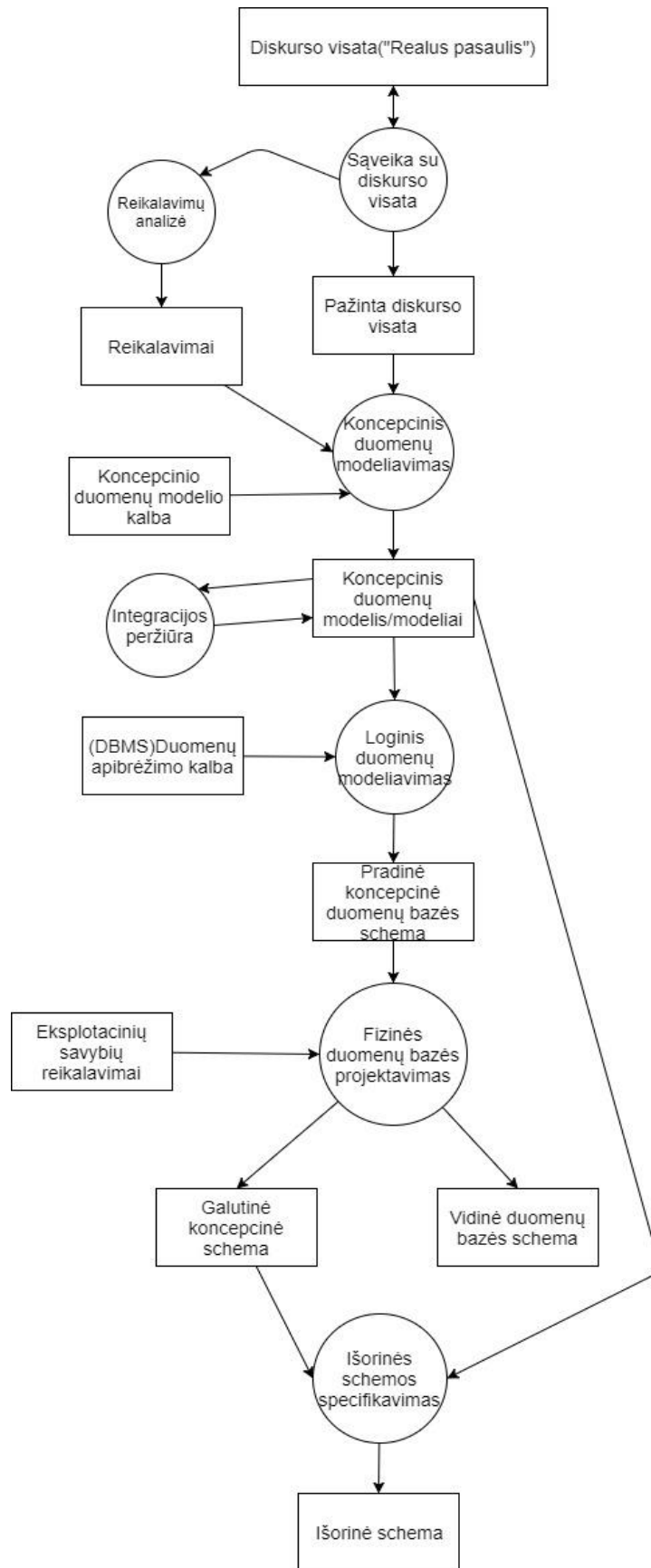
Numatomas problemos sprendimas: Apibrėžtą problemą spręsti yra pasirinkta tokiais būdais:

- sudaryti įvairių dalykinių sričių šablonų biblioteką;
- Sukurti automatizuotą įrankį, leidžiantį šablonus įterpti į UML klasių diagrama atvaizduotą kuriamą duomenų modelį.

1.3. Tyrimo objekto analizė

Šiame poskyryje pateikiama duomenų bazės kūrimo proceso pagrindinių etapų analizė. Žemiau pateikiama diagrama, kuri buvo sudaryta pagal Graeme Simson diagrama [1], kurioje pateikiama seka, pagal kurią dažniausia yra sudaroma duomenų bazės schema.

Duomenų bazės kūrimo procesas prasideda nuo sąveikos su diskurso visata (UoD) (angl. *Universe of Discourse*).



1 pav. Duomenų bazės kūrimo procesas pagal Graeme Simsion

1. Sąveika su diskurso visata. Analitikas surenka informaciją, kuri yra susijusi su tam tikros srities realaus pasaulio objektais, veiksmis.
2. Reikalavimų analizė. Šis etapas ne visuomet būna įtraukiamas į duomenų bazės kūrimo procesą. Tuomet metu kai jis įtraukiamas jis papildo analitiko suvokimą apie modeliuojamą realaus pasaulio objektą verslo taisyklėmis.
3. Konceptinis duomenų modeliavimas. Šiame žingsnyje sukuriama konceptiniai duomenų modeliai, kurie nėra priklausomi nuo jokio DBMS. Jie yra atvaizduojami pasinaudojant kuria nors konceptinio modeliavimo kalba pvz. Esysbių ryšių modeliu [2]. Kai kuriais atvejais šiame etape yra sukuriama keletas atskirų modelių, skirtų atvaizduoti skirtingoms perspektyvoms.
4. Integracijos peržiūra. Ji reikalinga tuomet, kai buvo sukurti keli skirtingi konceptiniai modeliai. Šio etapo rezultatas yra vienas bendras konceptinis duomenų modelis.
5. Loginis duomenų modeliavimas. Šiame etape konceptinis duomenų modelis yra paverčiamas į konceptinę schemą (dažnai vadiname loginiu duomenų modeliu arba loginę duomenų bazės schema) naudojant tam tikra duomenų apibrėžimo kalbą (angl. Data definition language) pagal pasirinktą DBMS.
6. Fizinės duomenų bazės projektavimas. Šiame etape specifikuojama vidinė duomenų bazės schema, kuri gali pakeisti pradinę konceptinę schemą norint pagerinti duomenų bazės efektyvumą.
7. Išorinės schemas specifikacija. Paskutinis etapas, kurio rezultatas yra išorinė schema.

1.3.1. Trys duomenų bazės lygiai

Iš aukščiau pateiktos informacijos, duomenų bazės kūrimo proceso galime išskirti 3 duomenų bazės lygius. Šiame darbe bus gilinamasi į pagrindinį duomenų bazės lygmenį: konceptinį duomenų modelį.

1.3.2. Konceptinis duomenų modelis

Konceptinis duomenų modelis yra pagrindas analizės ir dizaino metodologijų, skirtų informacinių sistemų kūrimui [3]. Per daugelį metų tyrėjai pasiūlė įvairių konceptinio duomenų modelio formalizmų, tokių kaip ER modeliai, įvairios šių modelių variacijos tokios kaip IDEF1X, Oracle CASE notacija, ORM, UML ir kiti [3]. Thalheim [4] nustatė, kad bendras skaičius siūlomų ER modelio variantų yra virš 80.

Plačiausiai praktikoje yra naudojami ER bei UML modeliai [5]. ER modelį pasiūlė Chen [6], šis modelis buvo plačiai naudojamas sistemų analizei ir conceptualiam modeliavimui. ER modelis yra lengvai suprantamas, jame yra atvaizduojamos realaus pasaulio koncepcijos, šis modelis tai pat lengvai paverčiamas duomenų bazės schema [7].

Natūrali kalba (angl. *Natural Language (NL)*) yra plačiausiai paplitęs būdas apibrėžti dalykus ir bendrauti. Tyrimai parodo [5] [8], kad beveik 90% visų industrinių reikalavimų yra aprašomi *NL*. Tačiau *NL* reikalavimus paversti konceptiniu duomenų modeliu yra sudėtinga [9].

1. *NL* reikalavimai gali būti dviprasmiški, todėl reikalinga tiksli analizė, metodai ir taisyklės modeliavimui;
2. Ta pati semantika gali būti pateikiama skirtingai, todėl reikalinga numatyti būdus, kaip susitvarkyti su šiais nuo konteksto priklausomais semantiniiais pokyčiais;
3. Ne visos sąvokos būna aiškiai išreikštos reikalavimų specifikacijos, tačiau yra svarbios sričiai, būna sunku jas atrasti. Tam, kad būtų galima įžvelgti paslėptas esybes, kurios nebuvo aiškiai įvardintos yra reikalingos ekspertinės dalykinės srities žinios.

1.3.2.1. Problemos su kuriomis susiduriama kuriant koncepcinius duomenų modelius

Problemos su kuriomis susiduria daugelis pradedančiųjų projektuotojų jau buvo nagrinėjamos anksčiau [10]. Nors jos ir buvo identifikuotos, tačiau tolimesni tyrimai vis dar rodo, kad jos nebuvo pašalintos [1]:

- **Kombinatorinis sudėtingumas galimuose ryšiuose tarp esybių.**

Didėjant esybių kiekiui, esybių ryšių sudėtingumas didėja kombinatoriniu tempu. Pradedantieji projektuotojai patiria sunkumus modeliuojant netgi paprastus binarinius ryšius [10]. Norinti prasmingai identifikuoti ryšius reikėtų, kad būtų laikomasi bent vieno iš šių kriterijų:

1. Turėtų būti modeliuojama visa semantika taikomoje programoje;
2. Neturėtų būti jokių nereikalingų ryšių konstrukčių;
3. Ryšių turėtų būti minimaliai, tai yra tik tiek kiek reikia, kad atitiktų reikalavimus.

- **Semantiniai nesutapimai**

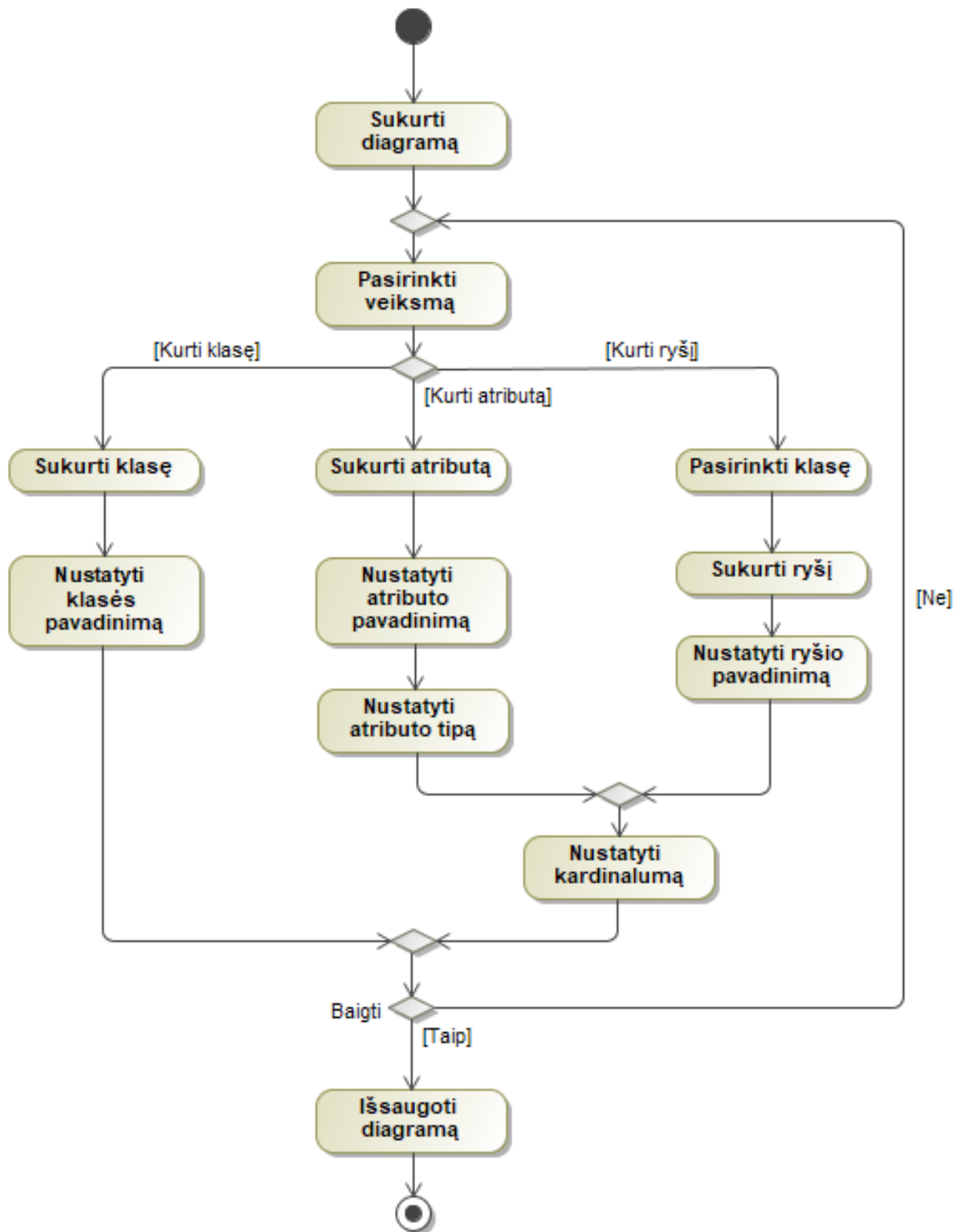
Ne visada galima tiesiogiai versti reikalavimus į duomenų bazės struktūrą [10]. Ne visi prasmingi realaus pasaulio ryšiai gali būti tinkamai išversti į duomenų bazės ryšius duomenų modelyje, kai kurie ryšiai turi būti išvestiniai duomenų bazės lygyje.

- **Patirties/dalykinės srities žinių trūkumas**

Pradedantieji dizaineriai turi ribotą žinių bei įgūdžių kiekį, kai tuo metu ekspertai gali pasinaudoti savo patirtimi iš ankstesnių darbų. Tačiau netgi ekspertai gali suklysti kurdami nežinomos dalykinės srities koncepcinį modelį [11]. Tik dalykinės srities ekspertai gali išvelgti paslėptas esybes, kurios yra reikalingos modelyje. Išskyla svarbi problema, kaip šios žinios gali būti efektyviai perduotos pradedantiesiems projektuotojams.

1.3.3. Tipinis koncepcinio duomenų modelio kūrimo procesas MagicDraw įrankyje

Vienas iš plačiau naudojamų modeliavimo įrankių yra MagicDraw. Šis įrankis taip pat naudojamas kurti duomenų modelius. Vartotojui atsidarius įrankį, sukūrus naują projektą bei naują klasių diagramą kuriant duomenų modelį, kuris bus verčiamas į duomenų bazės modelį vykdomas šis procesas pateiktas 2 pav.



2 pav. Tipinis duomenų modelio kūrimo procesas

Naudotojas kurdamas duomenų modelį *MagicDraw* įrankyje privalo pridėti kiekvieną naują elementą pats. Paprastai atliekama veiksmų seka:

1 žingsnis - pridėti klasės elementą, bei nurodyti jo pavadinimą.

2 žingsnis - pridėti atributus kuriuos turi sukurtą klasė.

3 žingsnis - nurodyti kiekvieno kuriamo naujo atributo tipą bei jo kardinalumą.

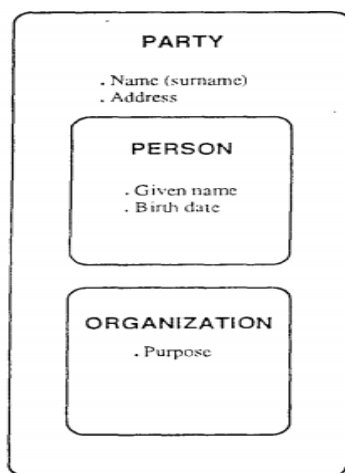
4 žingsnis - jei egzistuoja klasių tarp kurių turi būti sukurtas ryšis pridėti šį ryšį bei nurodyti klasių sąsajos kardinalumus.

Šie proceso žingsniai yra atliekami tol kol yra baigiamas kurti duomenų modelis. Kaskart kuriant naują modelį ir naudojant šį procesą paaiškėja jo trūkumai. Turint daug skirtingų modelių atsiranda galimybė tam tikras modelių dalis perpanaudoti, ko šis procesas nesiūlo. Dažnai tenka kurti pasikartojančius elementus (klases, atributus, ryšius). Pavyzdžiui, kiekvienai sistemai, kuriai reikalinga personalo valdymo dalis reikės sukurti mažiausiai 3 klases kurios yra privalomos (Asmuo, Organizacija, Darbo sutartis). Kuriant daug skirtingų modelių, kuriuose reikės šių elementų, juos kaskart tenka modeliuoti iš naujo.

Per daugelį metų tam tikros duomenų modelių dalys, kurios buvo plačiai naudojamos daugelyje modelių buvo įvardintos kaip šablonai.

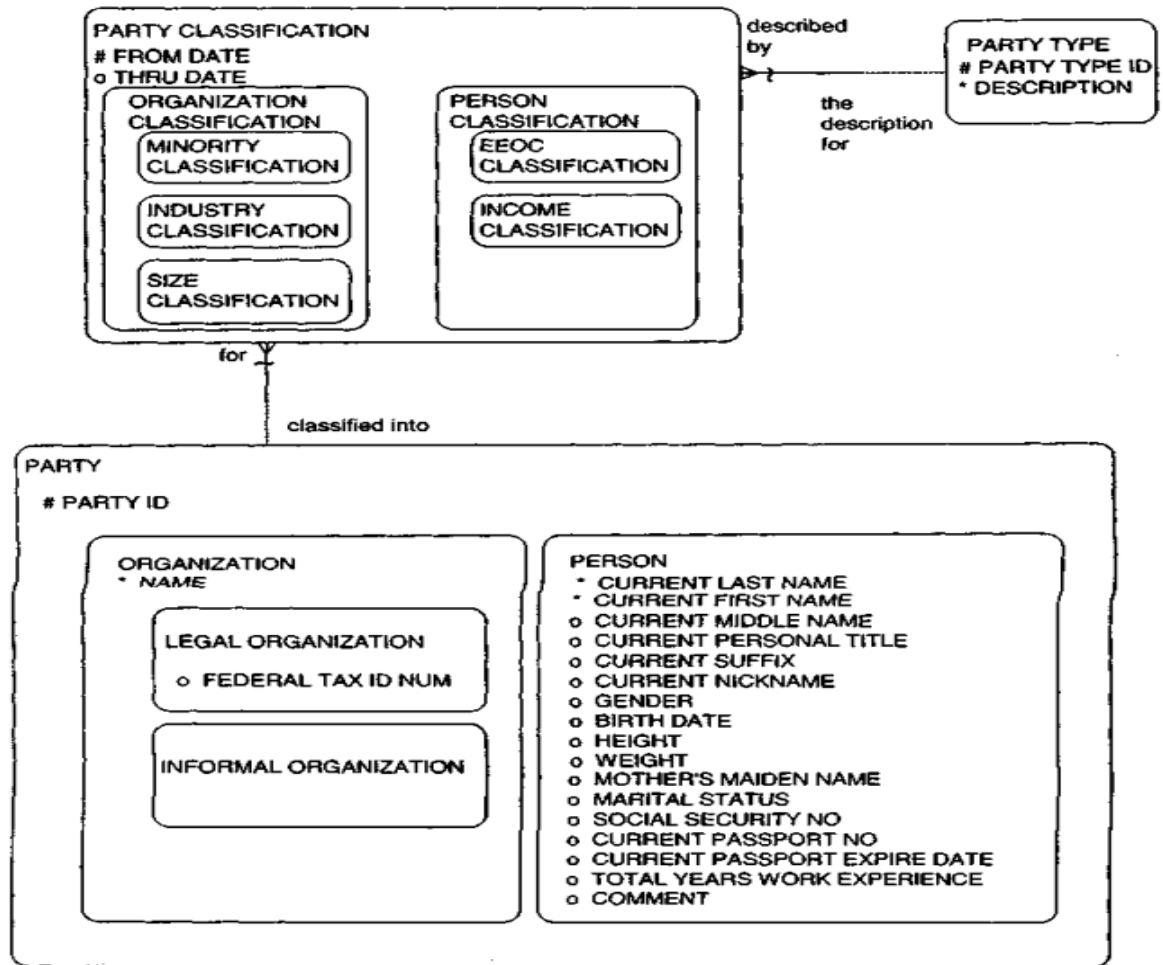
1.3.4. Duomenų modelių šablonų įvairovė

Duomenų modeliai egzistuoja ilgą laiką tarpą, daugelis autorių išvelgė panašumus tam tikrose verslo srityse, kur yra naudojami panašūs elementai ir tai įvardino kaip duomenų modelių šablonus. Tačiau net tie patys šablonai, kuriuos pateikia skirtingi autoriai yra šiek tiek skirtingi. Dauguma jų skiriasi abstrakcijos lygiu, kuris yra pritaikytas šablonui. Norinti, kad šablonas būtų tinkamas naudojimui ir priimtinas įvairioms sritims reikia rasti tinkamą jo abstrakcijos lygį [12].



3 pav. Subjektų šablonas pagal David C. Hay

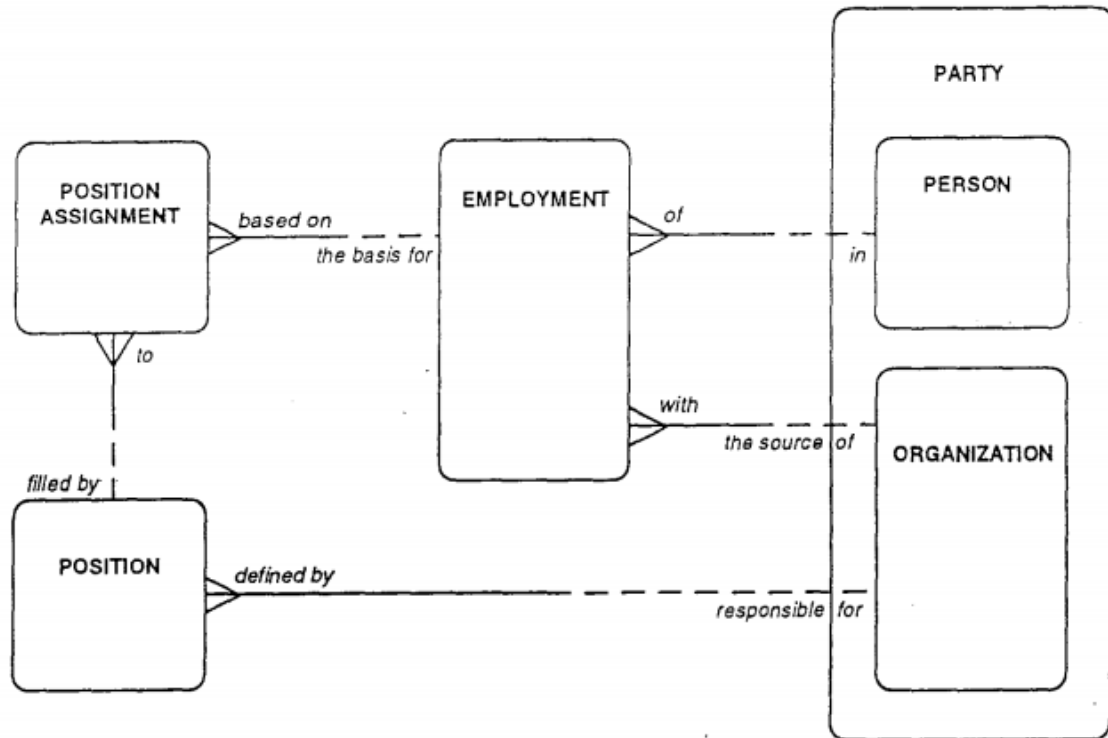
Aukščiau pateiktame 3 pav. yra pateikta „Parties“ šablonas kurį pristatė David C. Hay. Pristatytas šablonas yra labai abstraktus, pateikiamos tik kelios esybės. Viena super tipo (angl. *SuperType*) „Party“, kuri apima tiek „Person“ tiek „Organization“ esybės. Daugelis autorių duomenų modeliuose nusprendė šias dvi esybes, kurios yra svarbios apibrėžti bendru pavadinimu „Party“ kadangi korporacija yra „legalus asmuo“ [13] [14]. Abu autoriai išvelgė, kad ir organizacijos ir asmenys turi panašių charakteristikų kurias jie ir „iškėlė“ į Subjektas (angl. *Party*) esybę.



4 pav. Subjektų šablonas pagal Len Silverston

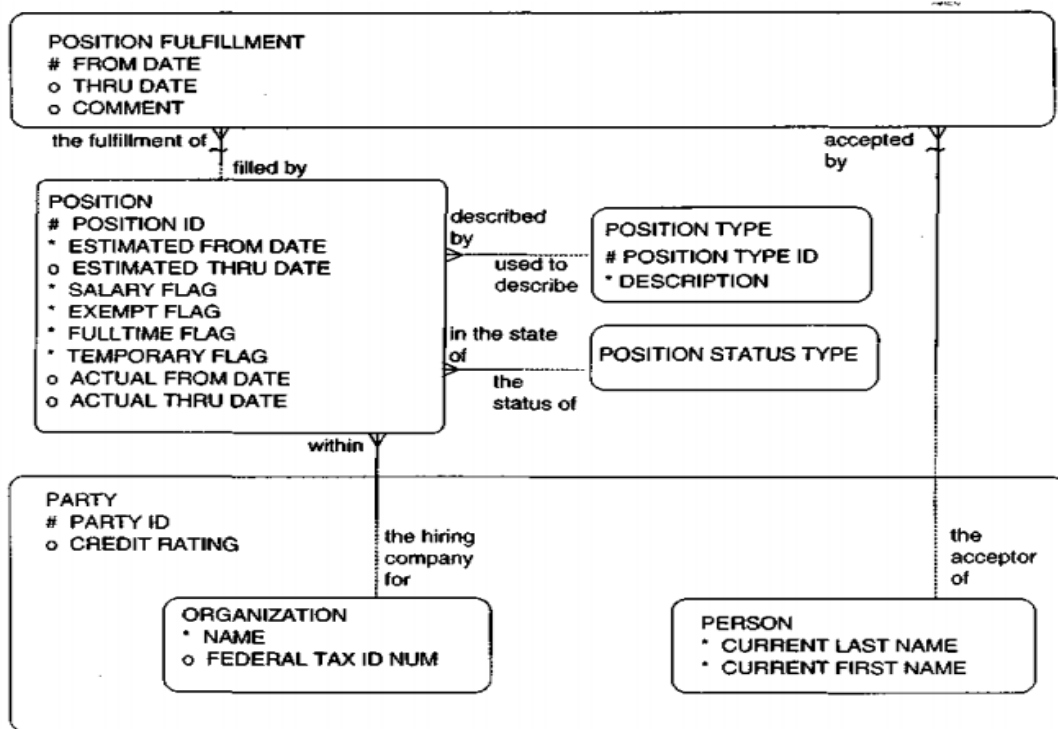
Aukščiau 4 pav. pavaizduotas „Parties“ modelis, kurį pateikė Len Silverston [14]. Autorius modelyje pateikia daugiau elementų nei David C. Hay, tačiau pats bendrinis modelis „Party“ vis tiek išlieka apimantis organizacijas ir asmenis. Šis modelis yra papildytas atributais bei papildytas nauja esybe „Party classification“. „Party classification“ taip pat yra super tipas, kurio viduje klasifikacija yra išskaidyta į organizacijos klasifikacija bei asmens klasifikacija. Len Silverston pateiktas modelis yra šiek tiek detalesnis ir mažiau abstraktus. Tačiau ne visada yra tikslinga naudoti šį modelį, kadangi modelyje pateikiami atributai gali būti nereikalingi, tam tikrame kontekste. Pavyzdžiui informacija apie asmens lytį, ūgį, svorį ir pan. gali būti nereikalinga. Reikia nepamiršti, kad šie šablonai negali pilnai realizuoti viso modelio. Įprastai bus naudojama tik keletas šablonų visame duomenų modelyje, tačiau jie dažniausiai apima esmines esybes [15].

Pozicijos/Įdarbinimo duomenų modelio šablonas



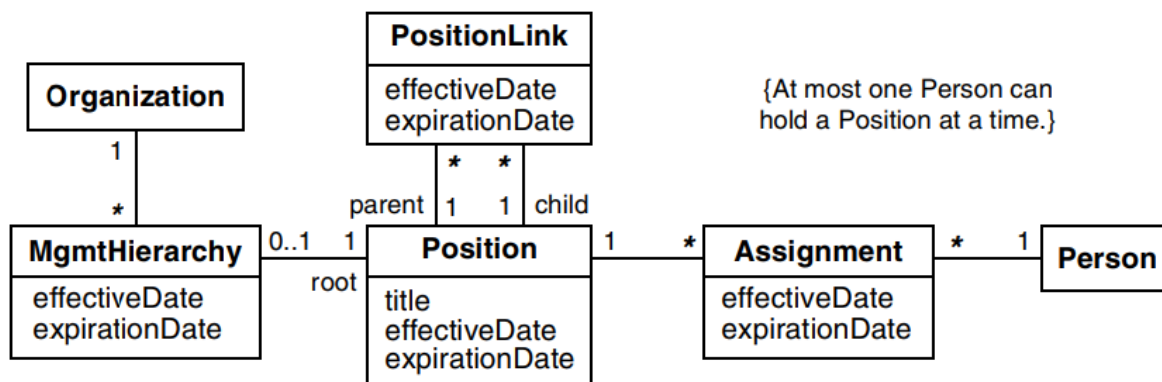
5 pav. Pozicijos šablonas pagal David C. Hay

Dar vienas modelis kurį tai pat savo knygoje „Data Model Patterns: Convention of Thoughts“ pateikė David C. Hay buvo įdarbinimo modelis [13]. Kadangi kiekvienas darbuotojas yra asmuo ir dirba tam tikroje organizacijoje modelyje išlieka tos pačios „Person“ ir „Organization“ esybės. Šį modelį papildžius „Employment“, „Position Assignment“, „Position“ esybėmis gaunamas įdarbinimo modelio šablonas. David C. Hay savo diagramose pateikia tik pagrindines esybes, bei ryšius tarp jų, nepateikdamas atributų, kurie reikalingi šiai diagramai. Pagrindiniai duomenų modeliai, kurie pateikiami David C. Hay yra konceptualus lygmenis.



6 pav. Pozicijos šablonas pagal Len Silverston

Len Silverston tai pat pateikė savo įdarbinimo modelį [14]. Pateiktas modelis yra detalesnis, modelyje yra išskirtos papildomos esybės pozicijos tipas bei pozicijos statuso tipas. Tokiu atveju suteikiama galimybė sekti pozicijos būsenas tokias kaip pvz. Aktyvi, Atidaryta, Nebeaktyvi, Uždaryta. Toks statusas kaip užimta nebūtų aktualus, kadangi šią informaciją galima gauti iš „Position fulfillment“ esybės [14]. Taip pat kas aktualu daugeliui detalesnių modelių, kuriuos pateikia Len Silverston yra tai, kad jis pateikia ir atributus savo modeliuose. Juose taip pat matyti išoriniai bei pirminiai raktai.



7 pav. Pozicijos šablonas pagal Michael Blaha

Dar vienas pozicijos modelis, kurį pateikė Michael Blaha [15]. Pateiktas modelis yra labai paprastas ir kaip pats autorius įvardina nėra pririštas prie jokios konkrečios srities. Michael Blaha savo knygoje „Patterns of Data Modeling“ modelius pateikė naudodamas *UML* ir *IDEF1X* notacija. Modelyje nurodytos pagrindinės esybės, kurios reikalingos sudaryti šabloną. Pagrindinės esybės išlieka visuose trijuose skirtinguose modeliuose, kuriuos pateikė skirtingi autoriai bei laikotarpis tarp šių modelių taip pat labai skyrėsi. Michael Blaha knyga išleista 2010 metais tai 9 metais vėliau nei Len Silverston ir 14 metų vėliau nei duomenų modeliavimo pradininko David C. Hay pasiūlytas modelis. Michael Blaha įvardino, kad jo pateikiami modelių variantai yra tikri šablonai, o šių senesnių autorių modeliai yra tiesiog sėkliniai modeliai (angl. *Seed models*), kurie yra labiau priristi prie tam tikrų taikymo sričių [15].

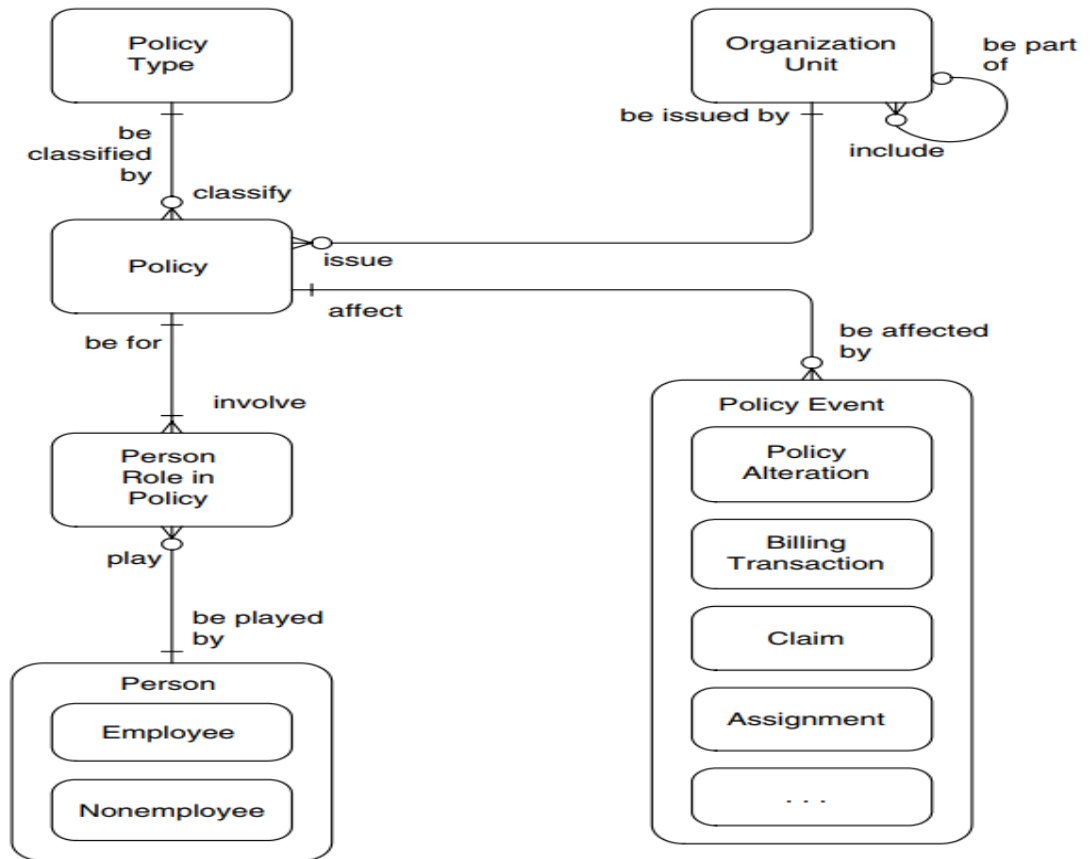
Palyginus keletą skirtingų autorių modelių galima pastebėti, kad jie yra šiek tiek skirtingi. Nors esmė yra ta pati tačiau Len Silverston pateikiami modeliai yra detalesni. Tai yra gerai norint greitai sukurti duomenų bazę, kadangi tai sumažina poreikį atrinkti tam tikrus atributus, kurie yra reikalingi tam tikrai esybei bei pašalina tikimybę, kad analitikas neįvertins ar užsakovas neįvardins tam tikro atributo, kuris gali būti reikalingas tam tikrai esybei apibrėžti. Kartais būna taip, kad užsakovas neįvardina ir pačios esybės [13].

1.3.4.1. Sėkliniai modeliai ir duomenų modelių šablonai

Nors šie du skirtingi modeliai dažnai buvo tapatinami įvairių autorių, tačiau Michael Blaha išskiria juos kaip skirtingus dalykus. Sėkliniai modeliai yra modeliai pritaikyti tam tikrai specifiniam srities problemai išspręsti. Taip šie modeliai yra labai naudingi ir stipriai pagreitina darbą tuo metu kai yra dirba toje pačioje srityje, tačiau su problema galima susidurti tuomet kai sritis pasikeičia. Pasikeitus sričiai vėl reikia rasti naujai sričiai tinkamą modelį arba koreguoti seną modelį tam, kad jis būtų pritaikytas naujai sričiai [15].

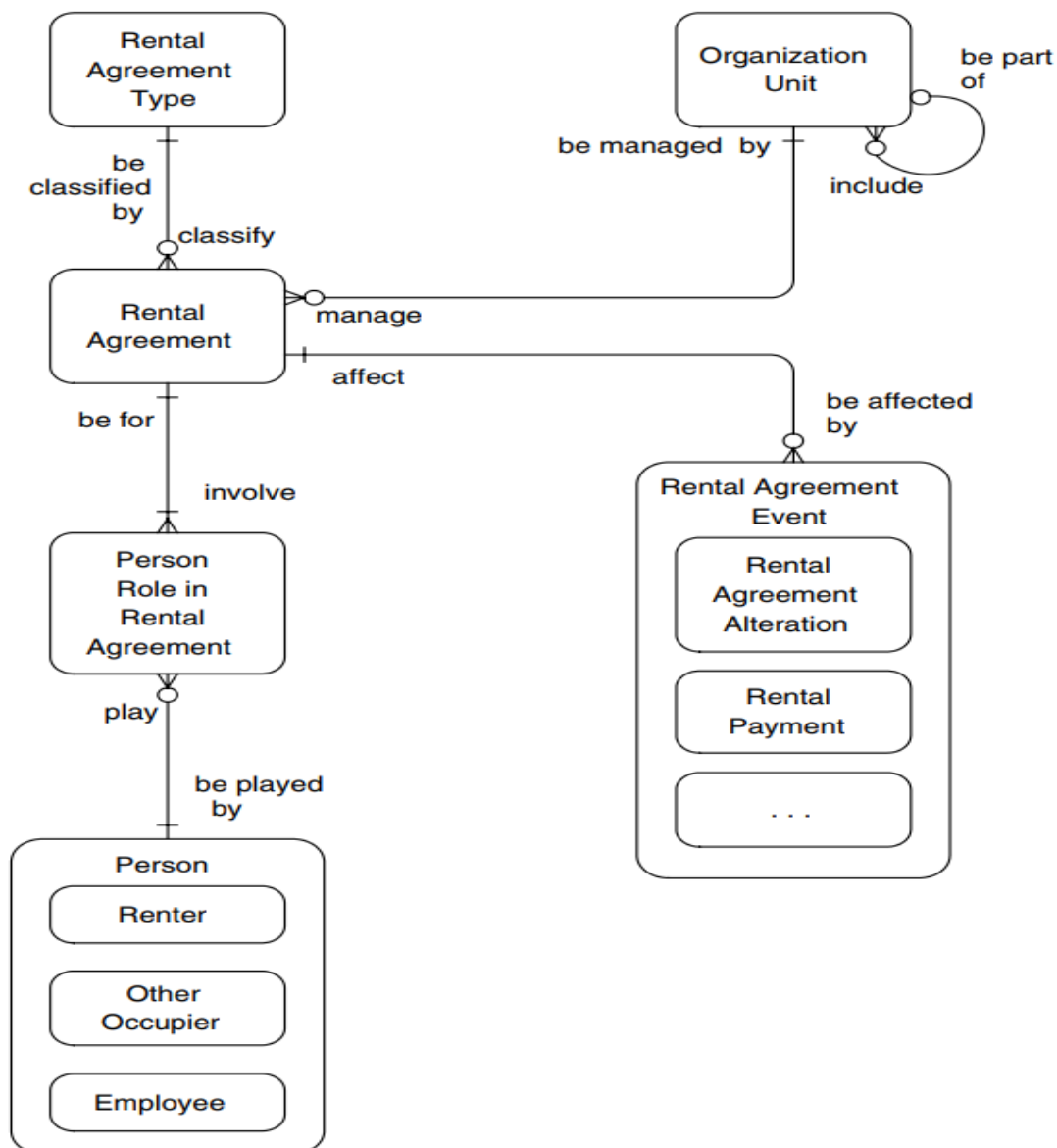
1.3.4.2. Bendriniai modeliai

Bendriniai modeliai (angl. *Generic models*) dar vienas modelių tipas, kurie taip pat gali būti naudojami kaip šablonai. Yra bendriniai modeliai, kurie plačiai atitinka naudotojų reikalavimus ir gali būti tiesiog pritaikomi tam tikrai sričiai. Bendrinio modelio pavyzdį, būdą jų pritaikymui bei variantą kaip vieną bendrinį modelį paversti kitų pateikė Graeme Simsion savo knygoje „Data Modelling Essentials“ [16]. Šioje knygoje autorius pasinaudodamas draudimo modeliu jį pritaikė nuomos sistemai. Draudimo modelio pavyzdys pateikimas žemiau.



8 pav. Draudimo sutarčių modelis pagal Graeme Simsion

Ne visuomet galime rasti bendrinį modelį reikiamai sričiai. Dažniausiai egzistuoja kažkoks bendrinis modelis, kuriuo pasinaudojant galima sukurti naują bendrinį modelį. Visų pirma reikia parinkti tinkamą bendrinį modelį, kuris atitiktų naują reikalingą modelį. Kaip šiuo atveju buvo pasirinktas draudimo sutarčių modelis. Kadangi nuomos paslaugų teikimo klientas įvardino reikalavimus, analitikas išvelgė, kad bus reikalingi asmenys organizacijos bei sutartys tarp jų. Žinant, kad praeityje buvo sukurtas draudimo modelis, kuris taip pat turėjo šias esybes buvo galima juo pasinaudoti ir taip gauti bendrinį nuomos paslaugų modelį.



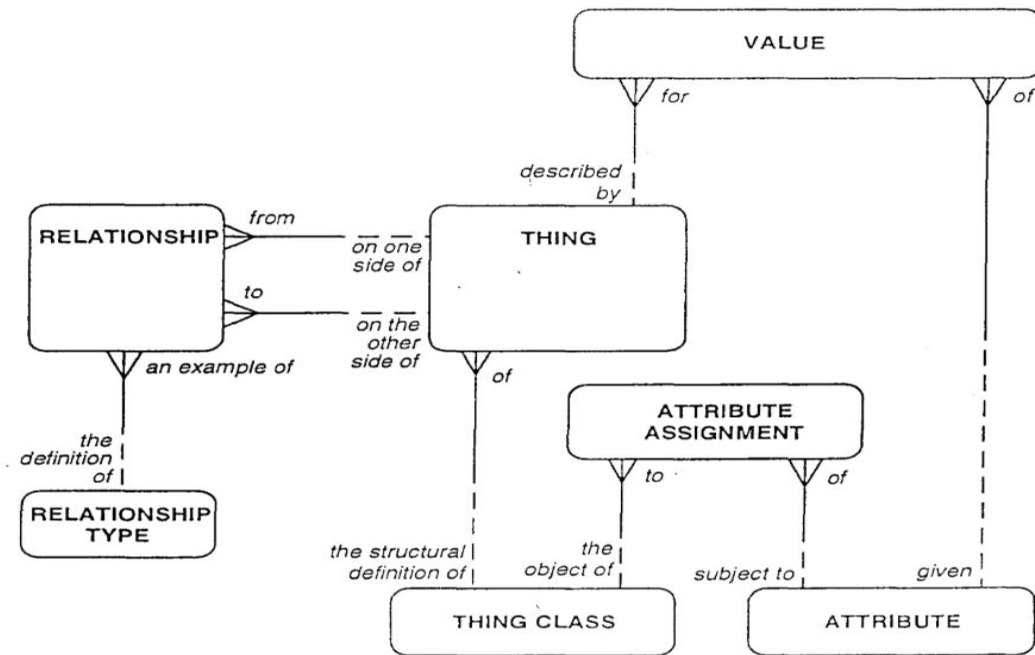
9 pav. Nuomos sutarčių modelis, pagrįstas draudimo sutarčių modeliu pagal Graeme Simson

Kaip matoma pateiktas modelis beveik nepasikeitė. Norint iš jo gauti bendrinį modelį reikėtų tiesiog apibendrinti „Rental Agreement“ į „Agreement“. Tokiu atveju būtų gaunamas bendrinis sutarčių modelis [16].

1.3.5. Duomenų modelių abstrakcijos lygiai

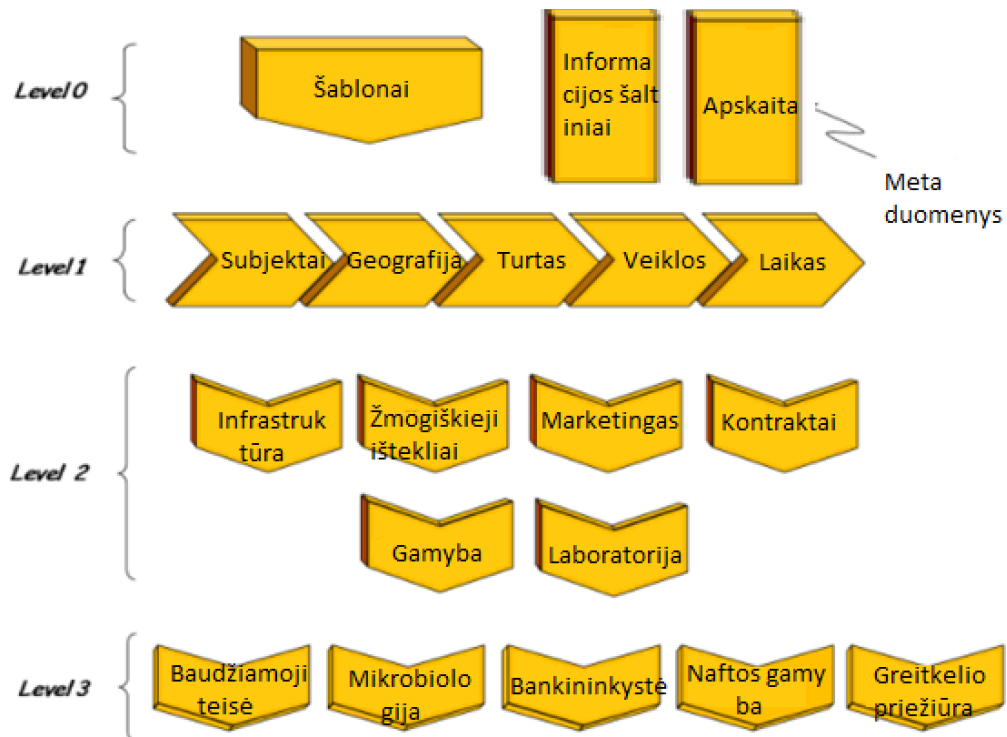
Įmanoma sumodeliuoti bet ką naudojant skirtingų abstrakcijos lygių variacijas. Kiekvieną kartą kuriant naują duomenų modelį iškylą klausimas kokį abstrakcijos lygį reikia parinkti atitinkamam modeliui [12]. Modeliui yra svarbus abstrakcijos lygis, kadangi nuo jo priklauso ar jį bus galima pritaikyti kitoms sritims ar modelis bus lankstus bei galės prisitaikyti prie pokyčių [12].

David C. Hay pateikia universalų duomenų modelį [13], kuriuo naudojantis galima apibrėžti visus dalykus egzistuojančius visatoje.



10 pav. Universalus duomenų modelio šablonas pagal David C. Hay

Modelis yra universalus, tačiau šis universalumas ne visuomet yra geras dalykas. Modelis yra labai abstraktus, todėl negali būti panaudojamas bet kurioje konkrečioje srityje. Bandant panaudoti šį modelį tam tikrame kontekste nekeičiant jo esybių pavadinimų jis nebūtų aiškiai suprantamas.



11 pav. Šablonų abstrakcijos lygiai pagal David C. Hay

Aukščiau pateiktame paveikslėlyje pavaizduoti abstrakcijos lygiai, kuriuos išskyrė David C. Hay [17]. Autorius išskyrė 4 skirtingus abstrakcijos lygius:

- **Lygis 0** visų pirma yra labai žemio lygio bendrinis šablonas. Tai abstrakčiausias modelis, kuris turi kokią nors prasmę. Šie modeliai yra pagrindai reikalingi įmonės modeliams, kurie yra 1 lygyje.
- **Lygis 1** šis lygis yra mažiau abstraktus nei 0 lygis. Modeliai, kurie yra pateikiami šiame lygyje gali būti naudojami kaip pradiniai taškai kuriant bet kokios įmonės duomenų modelius.
- **Lygis 2** šiame lygyje modeliai atvaizduoja funkcines sritis organizacijoje, tačiau jie vis tiek išlieka šablonais. Tokiu atveju šie šablonai gali būti pritaikomi daugeliui organizacijų, kurių veiklos sritys yra panašios.
- **Lygis 3** Šiame abstrakcijos lygyje pateikti modeliai specifinėms pramonės šakoms. Tai nereiškia, kad šiame lygyje bus pateikia modeliai pilnai atspindinys visą pramonės šakos modelį. Tačiau kiekvienai pramonės šakai reikia tam tikro specializuoto modelio, kuris spręstų problemas, kurios yra toje pramonės šakoje.

1.3.5.1. Duomenų modelių šablonų panaudojimas

Pagrindinis duomenų modelio šablonų panaudojimo būdas yra skirtas sudaryti duomenų modeliui, taip palengvinant projektuotojų darbą. Daugelis autorių pripažįsta, kad koncepcinis duomenų modelis yra pagrindinė duomenų bazės proceso dalis [16]. Tolimesni etapai vykdomi praplečiant šią dalį. Daugelis pateiktų šablonų yra koncepcinio modelio pavyzdžiu, šie šablonai yra pripažinti duomenų modeliavimo srityje. Pagal John Giles [18], galima padengti bet kurios įmonės duomenų modelį apie 50% panaudojant neutralius dalykinei sričiai nepriklausomus šablonus, o pasinaudojus tam tikros pramonės šakos šablonais galima padengti apie 75% duomenų modelio. Taigi vienas iš pagrindinių šablonų panaudojimo būdų yra jų naudojimas kuriant koncepcinius bei loginius duomenų bazės modelius.

Dar vienas būdas kaip gali būti panaudojami duomenų modelių šablonai yra galimybė juos panaudoti reikalavimų surinkimo etape kaip klausimų šaltinį pvz.: rengiant struktūrinį interviu. Tokiu atveju, kai analitikas nėra dalykinės srities ekspertas, pasinaudojant duomenų modelio šablonu galima atrasti nemažai klausimų, kurie gali būti aktualūs modeliuojamai sričiai.

1.4. Tyrimo objekto naudotojų analizė

Pagrindiniai tyrimo objekto naudotojai yra:

- Analitikai – tai asmenys, kurie atlieka kuriamos sistemos analizę yra atsakingi už reikalavimų surinkimą. Duomenų modelio esybių bei jų atributų identifikavimą ir teisingą užfiksavimą reikalavimų dokumente. Reikalavimų dokumentas tai reikalavimų analizės gautas produktas. Jis privalo būti patvirtintas visų susijusių šalių, tai yra tiek užsakovo tiek analitiko.
- Projektuotojai – tai asmenys, kurie yra atsakingi už sistemos projektavimą. Kadangi darbo tikslas yra duomenų bazės kūrimas, šiuo atveju bus atsižvelgiama tik į duomenų modelio kūrimo procesą.

Lentelė 1 Problemos su kuriomis susiduria naudotojai, bei kaip viskas vyksta dabar.

Problema	Kaip viskas vyksta dabar
Ne visi reikalavimai būna paminėti reikalavimų surinkimo etapo metu, dalis jų paaiškėja tik	Analitikui susitikus su užsakovu yra sprendžiamos tik esamos problemos, užsakovai

modeliuojant duomenų modelį [13].	nepamini svarbių faktų, kurie jiems atrodo savaime suprantami.
Dalykinės srities žinių trūkumas, nepilni duomenų modeliai.	Projektuotojai sukuria modelius, kurie nepilnai atspindi visą reikalingą įmonės architektūrą.
Duomenų modelio kūrimas yra ilgas ir sudėtingas procesas.	Projektuotojai yra priversti braižyti tuos pačius elementus kuriant naujos sistemos duomenų bazės modelį, kuriame yra tie patys elementai tokie kaip pvz. Organizacija, Asmuo, Pareigos, Produktas...

1.5. Esamų problemos sprendimo metodų analizė (Lietuvos ir tarptautiniu mastu)

Jau ilgą laiko tarpą yra aktuali problema, kaip būtų galima palengvinti duomenų bazės kūrimo procesą. Per šį laiko tarpą buvo pasiūlyta keletas sprendimo variantų. Kadangi viena iš svarbiausių dalių duomenų bazės kūrimo procese yra teisingo duomenų modelio sudarymas, todėl ir nagrinėsiu sprendimus, kurie buvo pasiūlyti, kaip būtų galima automatizuoti, ar pusiau automatizuoti koncepcinio duomenų modelio sudarymą.

Šiame skyriuje paminėsiu keletą pasiūlytų būdų, kuriais pasinaudojus būtų galima pusiau automatizuoti koncepcinio duomenų modelio kūrimo procesą. Tai pat plačiau apžvelgiau vieną iš paminėtų būdų, tą patį kuris yra nagrinėjamas šiame darbe, šablonais paremtas duomenų modelio sudarymas.

Būdai kuriais būtų galima pasinaudoti yra:

- **Lingvistiniu požiūriu pagrįstas būdas.** Šio būdo pagrindas yra taisyklės, kaip anglų kalbos sakiniai galėtų būti išversti į ER modelius, kuriuos pasiūlė Chen [2]. Toliau šios taisyklės buvo praplėstos, pritaikomi įvairūs apribojimai, tačiau šiam būdai taip ir nepavyko pašalinti NL kalbai būdingų limitacijų.
- **Pagal atvejį pagrįstas metodas.** Pagrindinė šio būdo mintis yra sudaryti saugyklą kurioje būtų saugomi koncepciniai modeliai tam tikroms problemoms spręsti. Naudojantis šia saugykla būtų galima rasti jau esamą sprendimą, žinoma, jei jis yra patalpintas į saugyklą, iškilusiai problemai. Šis būdas pasinaudoja galimybe perpanaudoti modelius, tačiau turi ir savų apribojimų. Jei reikalingi kažkokie pokyčiai reikia grįžti prie pradinio, įprasto modeliavimo metodo. Tai pat pokyčiai visada atsiranda keičiantis reikalavimų specifikacijai. Pagrindinis trūkumas yra tai, kad koncepcinių modelių bibliotekų vystymas, indeksavimo mechanizmas yra labai brangus.
- **Ontologija pagrįstas metodas.** Embley [7] teigia, kad pasinaudojus ontologija, galima išspręsti informacinių sistemų semantines problemas. Koncepcinio modeliavimo srityje, dauguma tyrėjų pasinaudoja ontologija, formuojant koncepcinio modeliavimo formalizmus. Ontologija gali būti naudojama kaip tam tikro srities žinių šaltinis ir dizaineriai gali ja pasinaudoti tam, kad įgyti tam tikrų dalykinės srities žinių. Aukštesnio lygio ontologijos pateikia bendras koncepcijas, kurios yra vienodos visose srityse ir susideda iš esybių hierarchijos ir taisyklių, toms esybėms, kurios nepriklauso jokiai konkrečiai probleminei sričiai. Pagrindinė problema su aukštesnio lygio ontologijomis yra tai, kad nėra jokios geros vartotojo sąsajos ar API dirbti su jomis.

- **Šablonais pagrįstas metodas.** Šablonų svarbumas buvo pripažintas Alexander knygoje [19] apie architektūrą ir urbanistiką. Šis autorius pasiūlė, kad projektuotojai geriau ruošų ir naudotų šablonus, negu spręstų problemą iš pirmojo principo. Dabar šablonai pripažintas būdas kaip spręsti pasikartojančias problemas programinės įrangos kūrimo procesuose. Šablonų perpanaudojimas pasiūlo daugelį privalumų, tokių kaip didesnis produktyvumas, geresnė kokybė ir sumažintos laiko bei kainos sąnaudos programinės įrangos kūrimo procese. Tačiau integruoti šablonus į koncepcinių modelių dizainą visada buvo sudėtinga. Šablonai koncepcinio modelio kontekste buvo pripažinti dėl tokių autorių darbų kaip Coad [20], Hay [13] ir Fowler [21]. Šie autoriai sukūrė duomenų modelių biblioteką ir pateikė pavyzdžius kaip bendrus modelius pritaikyti tam tikriems specifiniams reikalavimams.

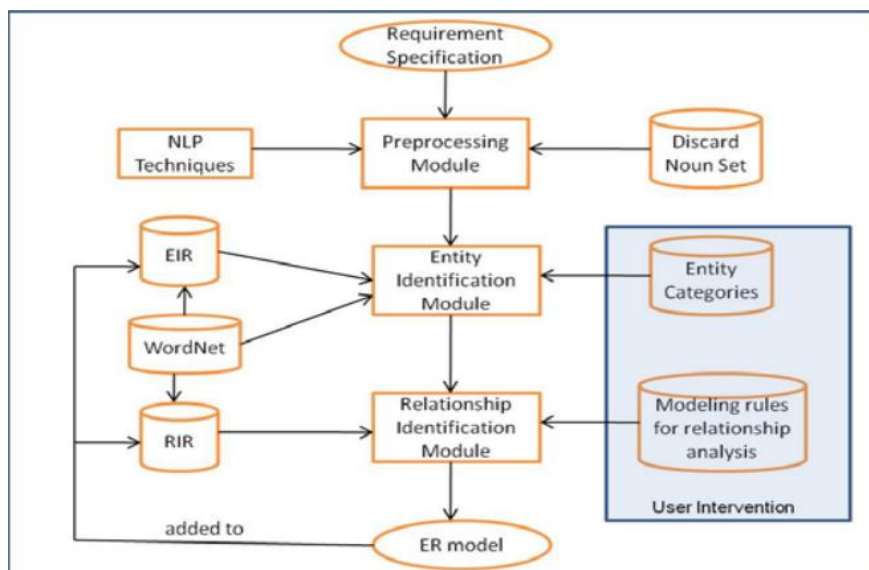
Empiriniai tyrimai parodo, kad ekspertai perpanaudoja šablonus, o pradedantieji dizaineriai ne [22]. Norint išvelgti šablonų privalumus reikia pasinaudoti ne tik komponentais esamais schemeje tačiau ir ryšiais tarp jų. Patį šablonų naudojimo procesą galima suskaidyti į 3 dalis [23]:

- Pasirinkti šablona, kuris tinkamas problemai spręsti;
- Pritaikyti šį šablonai problemai išspręsti;
- Integruoti šį šablona į visą modelį.

Šis sprendimas labai palengvina duomenų modelio kūrimo procesą, tačiau sudaryti šablonus yra sudėtinga, šablonai priklausomi nuo eksperto žinių, jis privalo nuspręsti kokiam abstrakcijos lygyje turi būti pateikiamas šablonas. Šablonų sudarymas užima labai daug laiko, šiuo metu dauguma saugyklų siūlo šablonus tik koncepciniams duomenų modeliams.

Vienas iš sprendimų kaip būtų galima sumažinti laiką ir pastangas reikalingas išgauti šablonams iš tam tikrų schemų yra galimybė sudaryti šablonų artefaktus įvairiose programų srityse. Kas praktiškai padėtų kurti daugkartinio panaudojimo modelius. Metodas išgauti šablonams iš egzistuojančių schemų (DFC) [24] buvo pasiūlytas tam, kad sutaupyti laiko ir išgauti tikrus šablonus.

- **Esybių pavyzdžių saugykla (angl. *Entity Instance Repository-EIR*) ir ryšių pavyzdžių saugykla.** Dar vienas būdas bei sukurtas įrankis yra esybių pavyzdžių saugykla ir ryšių pavyzdžių saugykla bei sukurtas įrankis esybių pavyzdžių šablonų žodžių tinklas (angl. *Entity Instance Patterns WordNet - EIPW*) [25]. EIPW [26] įrankis analizuojant reikalavimus parenka esybes bei jų ryšius taip sudarydamas modelius. Pats metodas naudojami DDL tam kad užpildytų EIR. Pradiniame etape ji buvo užpildyta pasiremiant Silverston parengta biblioteka į šį įrankį buvo įkelta 464 ryšiai ir 1859 atributai. Vėliau EIR ir RIR elementų sąrašas buvo praplėstas remiantis atvejų analize.



12 pav. EIPW įrankio architektūra

Aukščiau pateikiama EIPW įrankio architektūra. Įrankis nagrinėdamas natūralia kalba pateiktus reikalavimus parenka esybės iš surinktos bibliotekos EIR pagal rastus daiktavardžius. Po to pasinaudojant RIR biblioteka yra nustatomi galimi ryšiai tarp atitinkamų esybių ir taip sukuriamas ER modelis.

Aukščiau pateiktus koncepcinio duomenų modelio kūrimo metodus galima palyginti pagal šiuos 4 pasirinktu palyginimo kriterijus:

1. Kaina – kiekvienam modelio kūrimo metodui yra svarbi kaina, kadangi kainos kaštai gali būti didesni naudojant tam tikrą metodą.
2. Pritaikomumas – dar vienas svarbus kriterijus yra pritaikomumas, ne visi šie metodai gali būti lengvai pritaikomi praktikoje.
3. Lankstumas – kiekvienas modelis ir metodas turi būti lankstus, atsiradus sistemos reikalavimų pokyčiams pasirinktas metodus turi leisti šiuos pokyčius įgyvendinti ir modelyje.
4. Sudėtingumas – jei pasirinktas metodas bus per daug sudėtingas įgyvendinti tai gali įtakoti jo suteikiamus privalumus

Lentelė 2 Koncepcinio modelio kūrimo automatizavimo metodų palyginimas

Metodas	Kaina	Pritaikomumas	Lankstumas	Sudėtingumas
Lingvistiniu požiūriu pagrįstas būdas	Aukšta	Žemas	Aukštas	Aukštas
Pagal atvejį pagrįstas būdas	Aukšta	Vidutinis	Žemas	Aukštas
Ontologijomis pagrįstas būdas	Aukšta	Žemas	Aukštas	Aukštas
Šablonais pagrįstas būdas	Žemas	Aukštas	Aukštas	Vidutinis

Pagal pateiktus rezultatus 2 lentelėje matoma, kad pranašiausias metodas yra šablonais pagrįstas kūrimo būdas. Kadangi dauguma šablonų yra pateikti literatūrinuose šaltiniuose jų kainą yra žemiausia. Šablonai priklausomai nuo jų abstrakcijos lygio yra lengvai pritaikomi praktikoje, modeliai gauti panaudojant šablonus yra lankstūs. Šablonų sudėtingumas tiesiogiai priklauso nuo jų abstrakcijos lygio. Tam tikrai atvejais gali būti sudėtinga parinkti tinkamą šabloną, nežinant tolimesnių sistemos plėtimo galimybių ir ateities poreikių.

1.6. Darbo tikslas, uždaviniai ir siekiami privalumai

Pagrindinis šio tyrimo tikslas yra sudaryti sąlygas lengviau kurti duomenų bazę, kuriant naujas sistemas ar tobulinant senas. Šis tikslas bus pasiektas, jei atliktus eksperimentą panašių ar tos pačios sistemos duomenų bazės kūrimo laikotarpis sutrumpės, arba duomenų bazė bus lankstesnė. Šiam tikslui pasiekti buvo suformuoti tokie uždaviniai:

1. Apibendrinti įvairių autorių patirtį;
2. Surinkti ir suskaitmeninti įvairių dalykinių sričių tipinius duomenų modelius;
3. Parengti automatizuotą metodiką, leidžiančia šiuos modelius pritaikyti konkrečiam atvejui;
4. Sukurti duomenų modelių šablonų biblioteką, ir plėtinį kuris ją panaudotų;
5. Realizuoti "Duomenų modelių šablonų" plėtinį;
6. Eksperimentiškai iširti sukurtą sprendimą.

1.7. Siekiamo sprendimo apibrėžimas

Suformuotai problemai yra siūlomas sprendimo būdas:

- Sudaryti įvairių šablonų rinkinių biblioteką;
- Sukurti automatinis įrankį leidžiantį šabloną įterpti į UML diagramą.

Šablonai yra pateikti įvairiose šaltiniuose šie šablonai turi būti suskaitmeninti parengiant UML klasių diagramas. Šablonai turi būti sukelti į vieną biblioteką ir biblioteka patalpinta į „*MagicDraw*“ įrankį. Tuomet suprogramuotas įrankis, kuris leistų pasirinkti atitinkamą šabloną ir jį įkelti į naudotojo kuriamą koncepcinį duomenų modelį.

1.8. Analizės išvados

Atlikus analizę buvo priimtos tokios išvados:

- 1) Išanalizavus duomenų bazės kūrimo procesą nustatyta, kad ilgiausiai užtrunka sukurti koncepcinį duomenų bazės modelį.
- 2) Atlikus literatūros šaltinių analizę nustatyta, kad per pastaruosius 30 metų yra sukurta daugiau nei 50 įvairių dalykinių sričių duomenų modelių šablonų.
- 3) Duomenų bazės kūrimo proceso analizės metu nustatyta, kad vienas iš būdų kaip pagreitinti šį procesą yra panaudoti duomenų modelių šablonus.
- 4) Duomenų modelių šablonų analizės metu nustatyta, kad pasinaudojant bendriniais nuo dalykinės srities nepriklausomais šablonais galima padengti apie 50% bet kurios įmonės duomenų modelio esybių, o pasinaudojus specifinės pramonės srities šablonais galima padengti iki 75% duomenų modelio.
- 5) Duomenų modelių šablonų analizė parodė, kad šablonai gali būti naudojami kaip pagalbiniis įrankis nustatant reikalavimus planuojamai kurti sistemai.

- 6) Siekiamo sprendimo analogų analizės metu nustatyta, kad šiuo metu nėra realizuoto įrankio, kuris leistų iš duomenų modelių šablonų bibliotekos pasirinkti dalykinei sričiai tinkamą modelį ir jį įterpti į kuriamos sistemos duomenų modelį.

Apibendrinant analizės išvadas galima spręsti, kad norint pagreitinti duomenų bazės kūrimo procesą reikia pagreitinti šio proceso koncepcinio modelio kūrimo etapą. Koncepcinio modelio kūrimas gali būti pagreitintas panaudojant šablonus. Siūlomas sprendimas analogų neturi, ir dabartinės technologijos leidžia jį įgyvendinti.

2. Duomenų modelių šablonų sprendimo reikalavimų specifikacija ir projektas, formalus aprašas

2.1. Reikalavimų specifikacija

2.1.1. Sistemos aprašas

Kuriamas įskiepis, skirtas *MagicDraw* modeliavimo įrankiui. Pagrindinė „Plugin“ dalis yra duomenų modelių šablonų biblioteka kurioje patalpinti įvairių dalykinių sričių, bei universalūs, nepriklausantys jokiai dalykiniai sričiai duomenų modeliai. Duomenų modelių šablonų autoriai yra David C. Hay bei Len Silverston. Pagrindinis įrankio tikslas praplėsti *MagicDraw* galimybes kuriant duomenų modelius. Naudojantis parengta duomenų modelių šablonų biblioteka bei parengtu įrankiu vartotojai galės:

- Peržiūrėti esančius šablonus bibliotekoje.
- Atlikti paiešką, pagal raktinius žodžius.
- Pasirinkti ir įterpti šabloną ar šablono esybes į savo modelį.

Įrankyje bus naudojamosi *MagicDraw* egzistuojančiomis funkcijomis jas šiek tiek praplečiant, ar pridėdant naujas reikalingas pasiekti norimą sprendimą. Naudojantis šiuo įrankiu, įvairaus lygio, tiek specialistai tiek pradedantieji projektuotojai galės pagreitinti duomenų modelio kūrimą, kas galiausiai pagreitins ir pačios duomenų bazės kūrimo procesą.

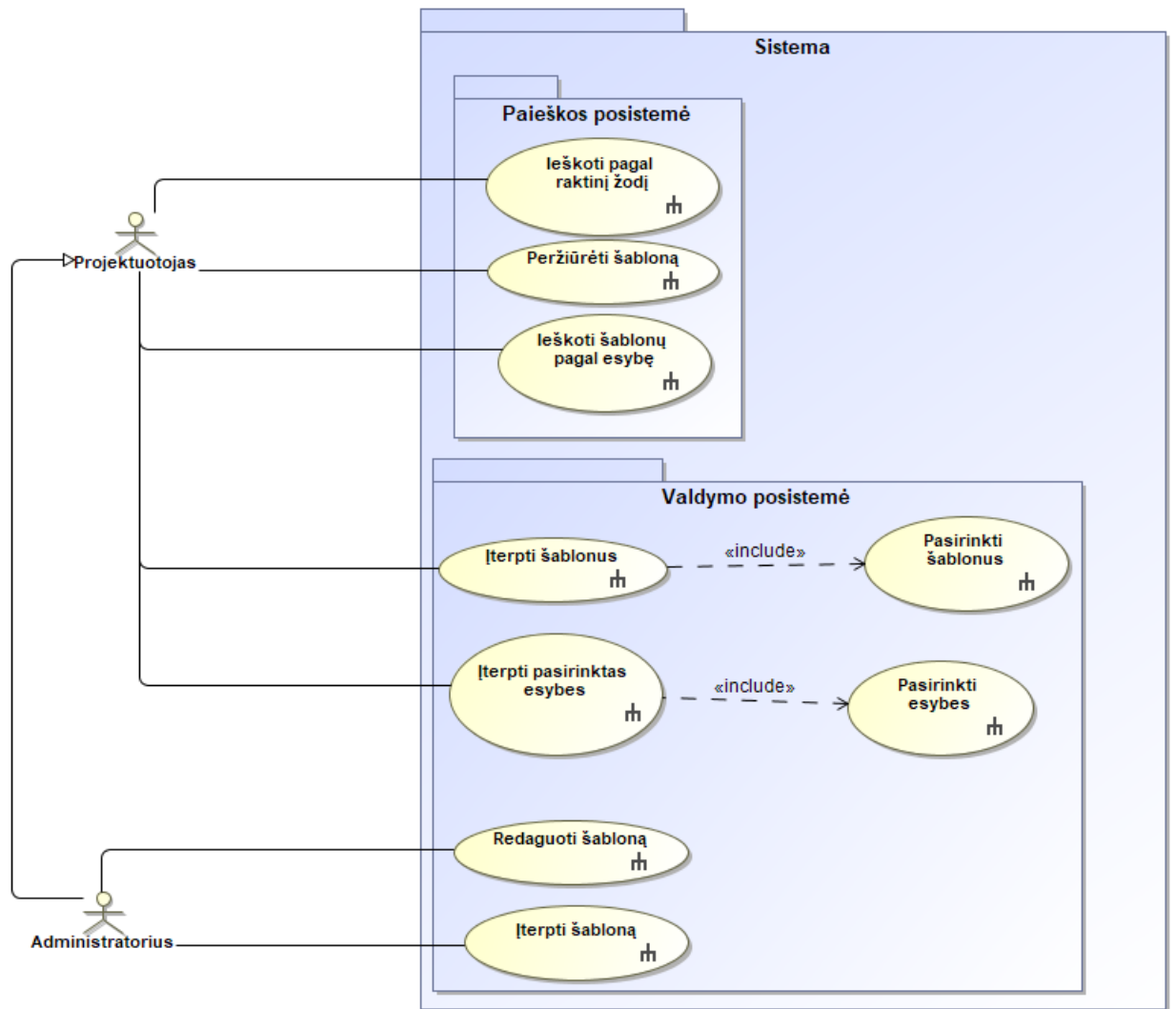
2.2. Sistemos reikalavimai

2.2.1. Nefunkciniai reikalavimai

Sistemai yra keliami nefunkciniai reikalavimai šablonų esybių atvaizdavimui vartotojo modelyje. Kadangi kuriant duomenų modelį naudotojas gali naudoti ne tik šablonų esybes bet ir kurti naujas savo esybes, koncepciniame modelyje šablonų esybės turi būti pažymimos naudojant „Copied class“ stereotipą.

2.2.2. Panaudojimo atvejai

Numatytų kuriamo įskiepio panaudojimo atvejų diagrama pateikiama 13 pav.



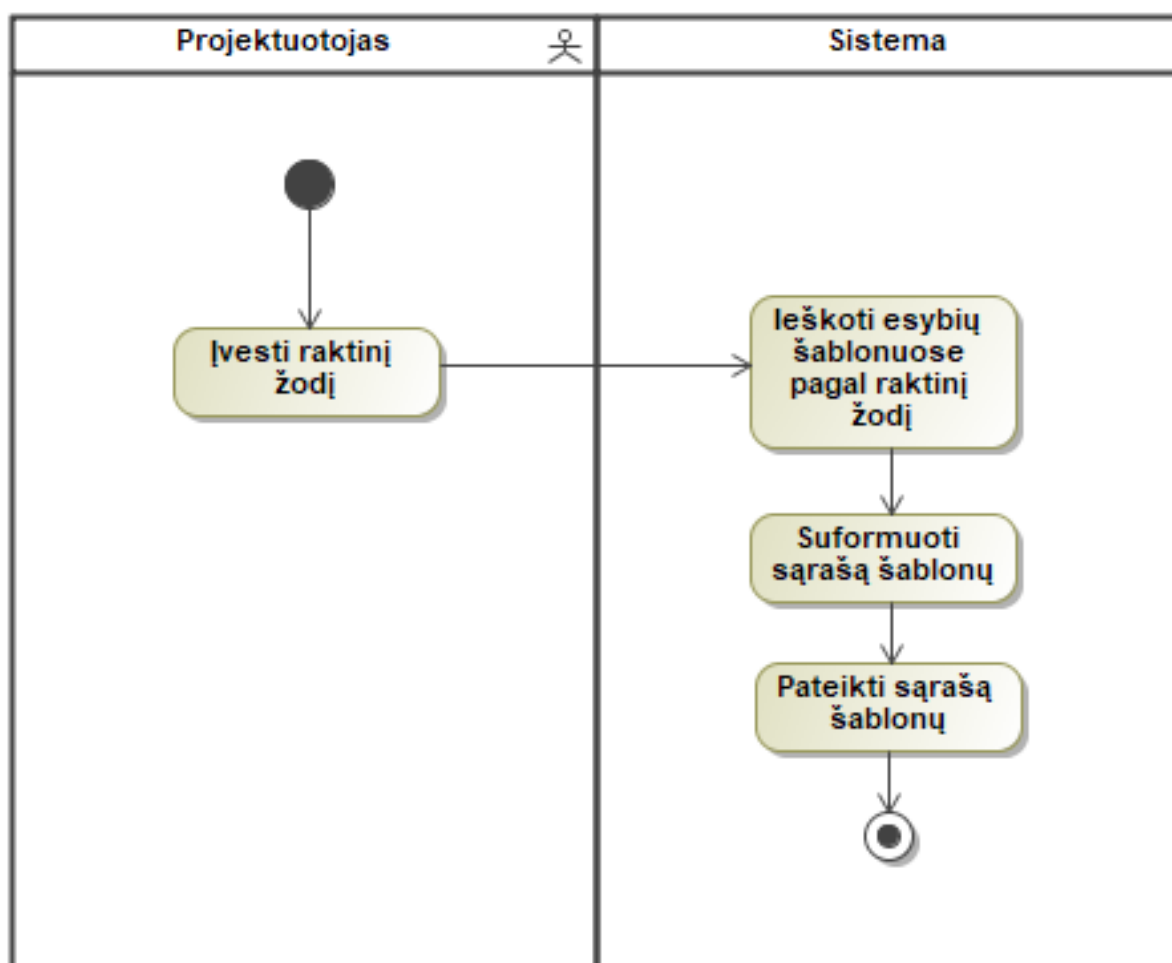
13 pav. Panaudojimo atvejų diagrama

Lentelė 3 Panaudojimo atvejo „Ieškoti pagal raktinį žodį“

PA1 „Ieškoti pagal raktinį žodį“	
Tikslas/uždavinys. Surasti šablonus pagal raktinį žodį	
Aprašymas. Vykdamas šį PA sistemos naudotojas gali atlikti paiešką šablonų bibliotekoje pagal raktinį žodį.	
Prieš-sąlyga	Atsidaręs paiešką
Aktorius	Projektuotojas, Administratorius
Sužadinimo sąlyga	Naudotojas nori surasti šabloną pagal raktinį žodį
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuoti PA
Pagrindinis scenarijus	
Pagrindinis įvykių srautas	Sistemos reakcija

Po-sąlyga	Naudotojui pateikiamas šablonų sąrašas pagal raktinį žodį
------------------	---

14 pav. pateikiama šio panaudojimo atvejo veiklos diagrama.



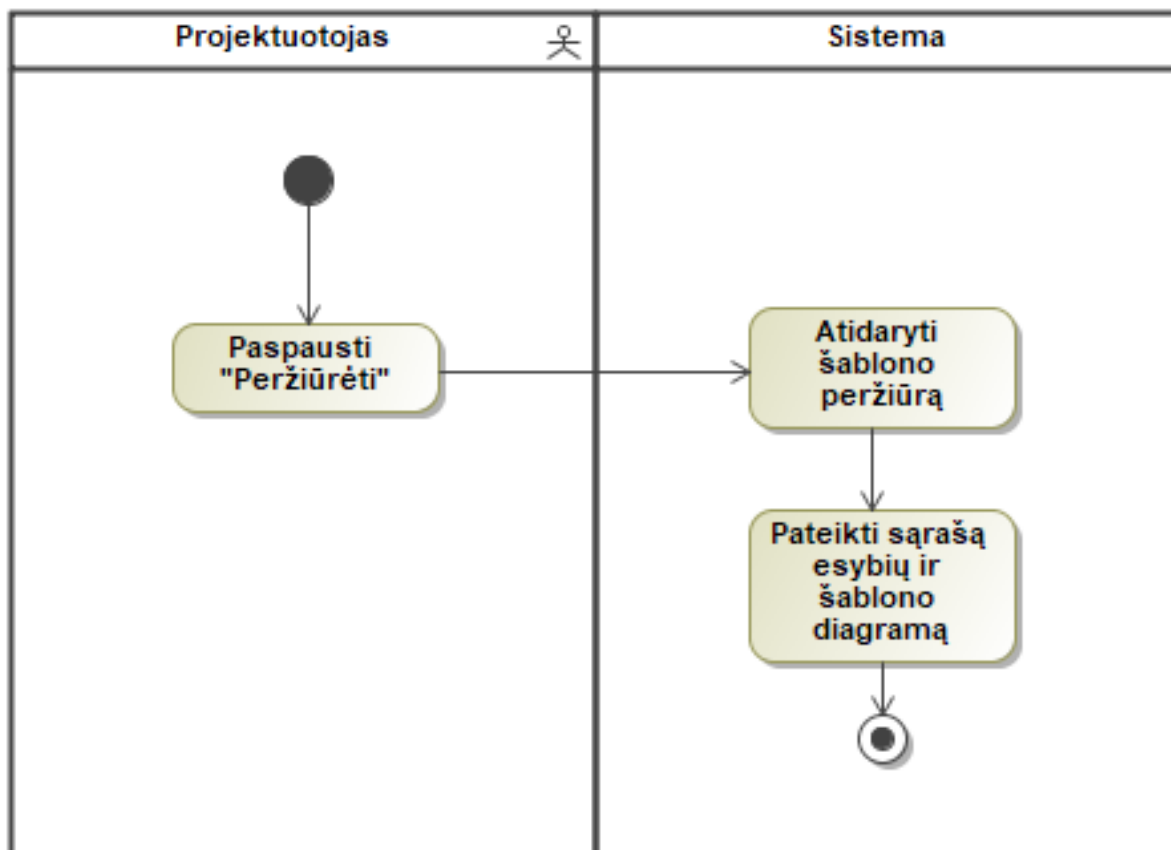
14 pav. Ieškoti pagal raktinį žodį

Lentelė 4 Panaudojimo atvejo „Peržiūrėti šabloną“

PA2 „Peržiūrėti šabloną“	
Tikslas/uždavinys. Peržiūrėti šabloną	
Aprašymas. Vykdamas šį PA sistemos naudotojas gali peržiūrėti šabloną	
Prieš-sąlyga	Naudotojas yra šablonų sąrašė
Aktorius	Projektuotojas, Administratorius
Sužadinimo sąlyga	Naudotojas nori peržiūrėti šabloną
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuoti PA
Pagrindinis scenarijus	
Pagrindinis įvykių srautas	Sistemos reakcija

Po-sąlyga	Naudotojas mato šablono turinį

15 pav. Pateikiama panaudojimo atvejo veiklos diagrama.



15 pav. Peržiūrėti šablona

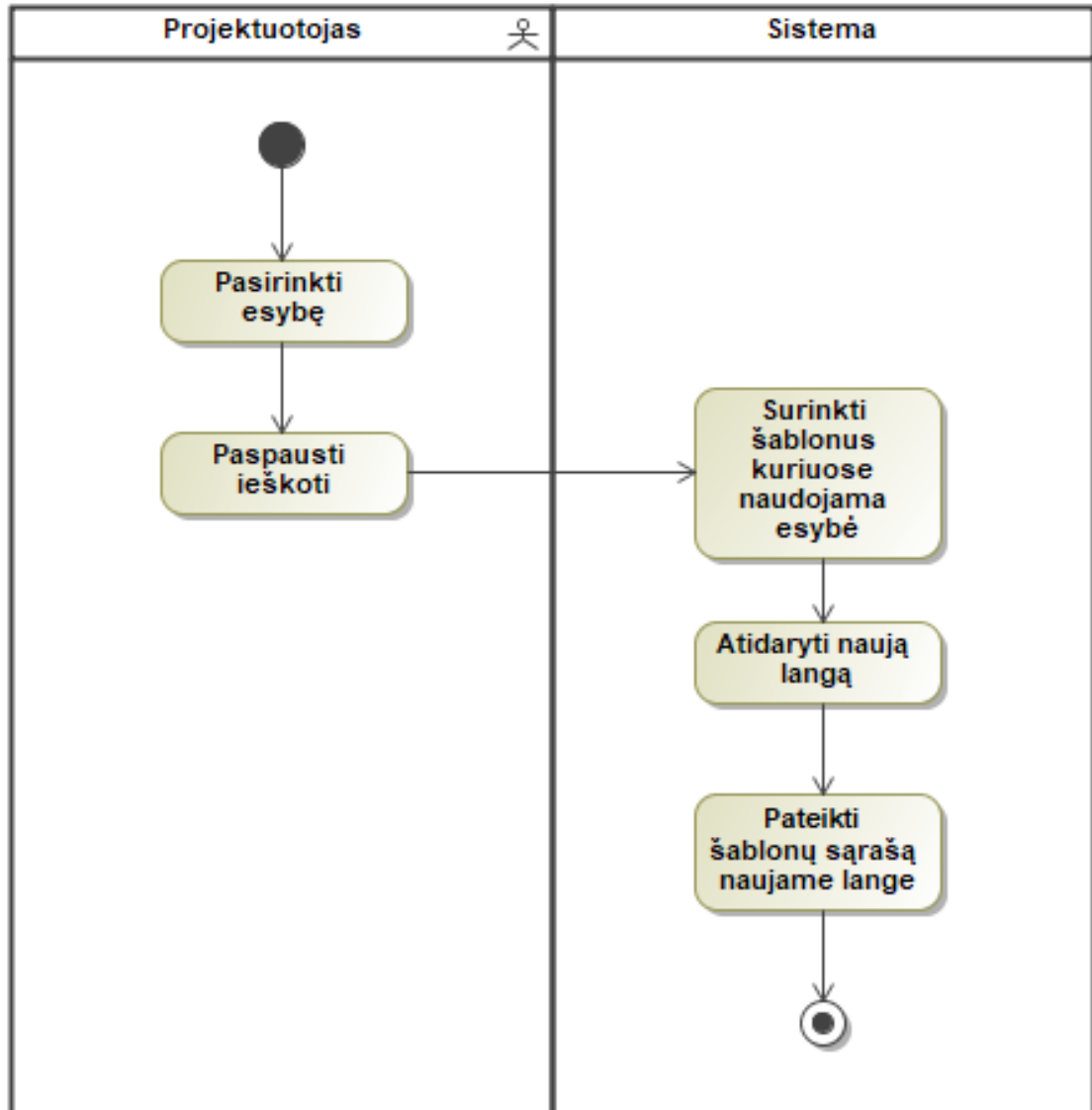
Lentelė 5 Panaudojimo atvejo „Ieškoti šablonų pagal esybę“

PA3 „Ieškoti šablonų pagal esybę“	
Tikslas/uždavinys. Surasti šablonus kuriuose naudojama esybė	
Aprašymas. Vykdamas šį PA sistemos naudotojas gali surasti šablonus kuriuose naudojama pasirinkta esybė	
Prieš-sąlyga	Naudotojas yra paieškos lange
Aktorius	Projektuotojas, Administratorius
Sužadinimo sąlyga	Naudotojas nori surasti šablonus
Susiję panaudojimo atvejai	Išplečiantys PA
	Apimami PA
	Specializuoti PA
Pagrindinis scenarijus	
Pagrindinis įvykių srautas	Sistemos reakcija

Po-sąlyga

Naudotojas gauna šablonų sąrašą

16 pav. Pateikiama panaudojimo atvejo veiklos diagrama.



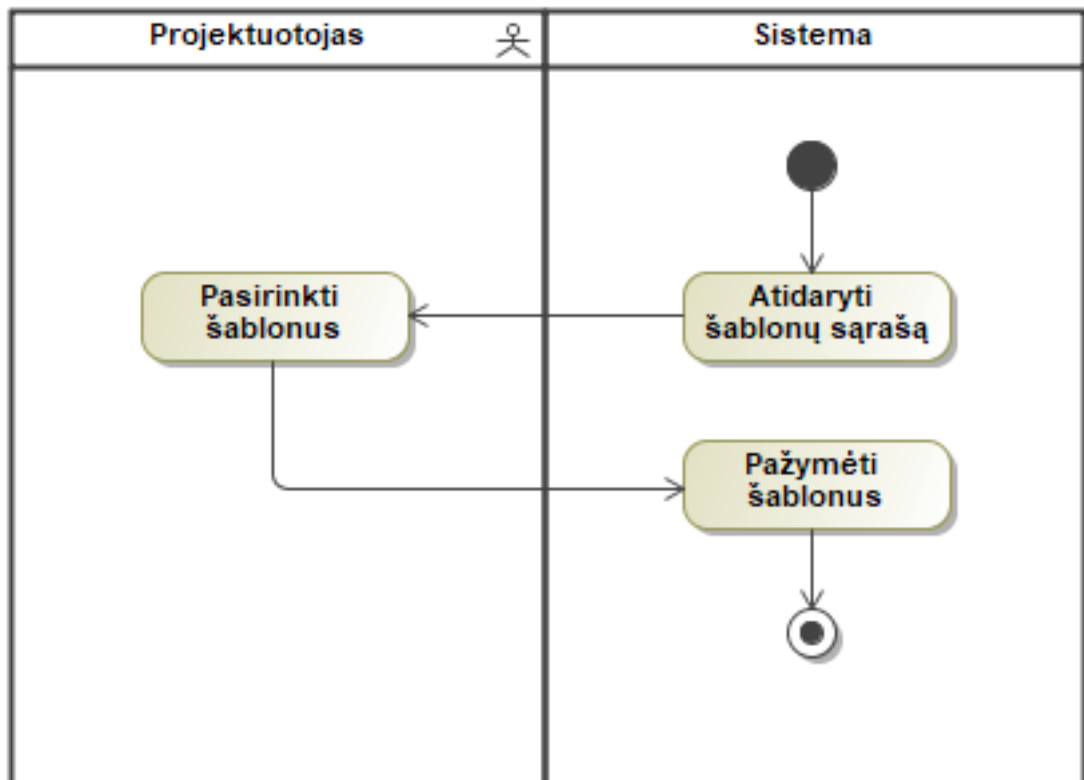
16 pav. Ieškoti šablonų pagal esybę

Lentelė 6 Panaudojimo atvejo „Pasirinkti šablonus“

PA4 „Pasirinkti šablonus“		
Tikslas/uždavinys. Pasirinkti šablonus		
Aprašymas. Vykdam šį PA sistemos naudotojas gali pasirinkti šablonus		
Prieš-sąlyga	Naudotojas yra šablonų sąrašė	
Aktorius	Projektuotojas, Administratorius	
Sužadinimo sąlyga	Naudotojas nori pasirinkti šablonus	
Susiję panaudojimo atvejai	Išplečiantys PA	PA7 „Įterpti pasirinktus“
	Apimami PA	

	Specializuoti PA	
Pagrindinis scenarijus		
Pagrindinis įvykių srautas		Sistemos reakcija
Po-sąlyga		Pažymėti pasirinkti šablonai

17 pav. Pateikiama panaudojimo atvejo veiklos diagrama.



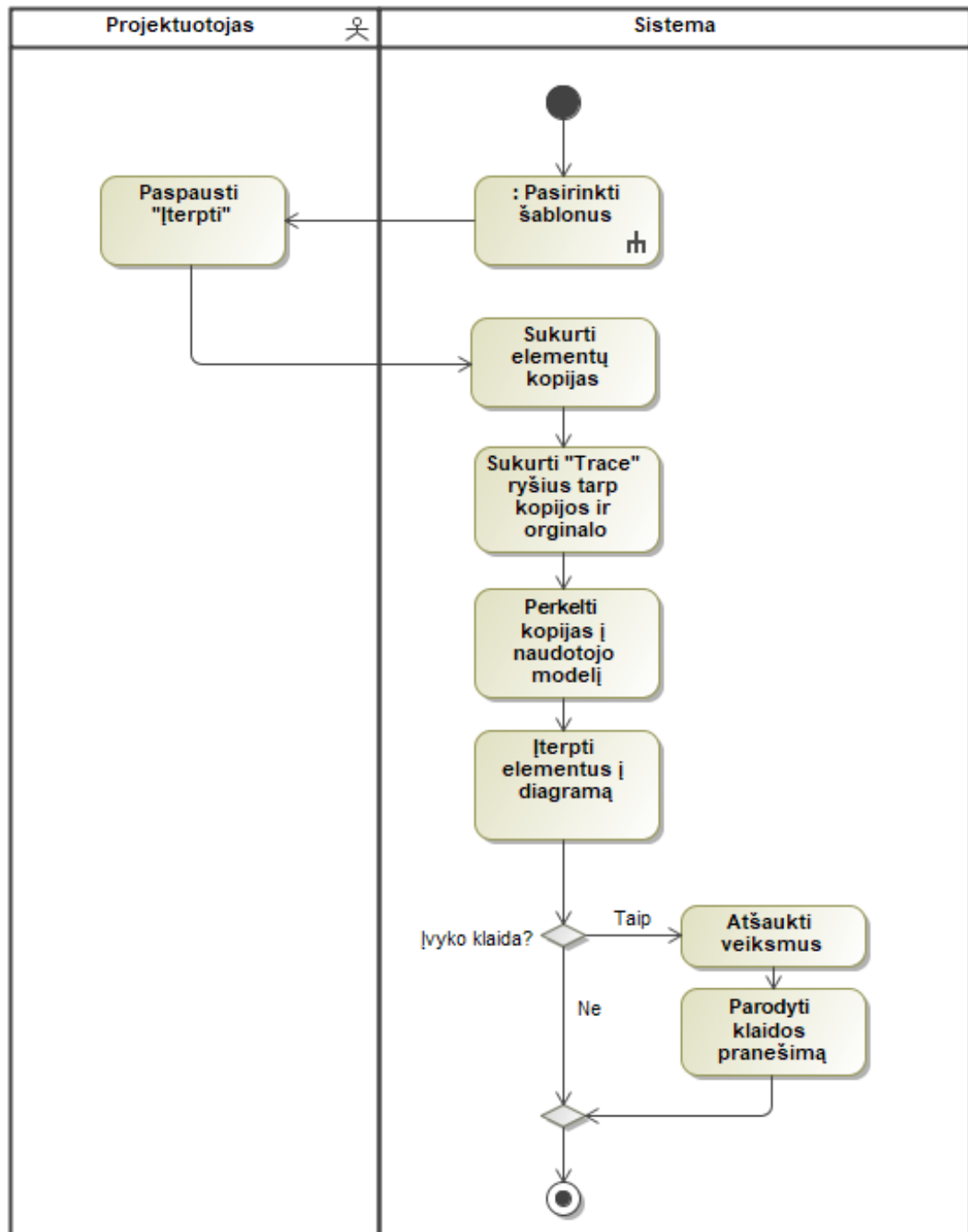
17 pav. Pasirinkti šablonus

Lentelė 7 Panaudojimo atvejo „Įterpti pasirinktus“

PA5 „Įterpti pasirinktus“		
Tikslas/uždavinys. Įterpti pasirinktus		
Aprašymas. Vykdam šį PA sistemos naudotojas gali įterpti pasirinktus šablonus į modelį		
Prieš-sąlyga	Naudotojas yra šablonų sąrašė ir pasirinko bent vieną šabloną	
Aktorius	Projektuotojas, Administratorius	
Sužadavimo sąlyga	Naudotojas nori įterpti šablonus	
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	PA4 „Pasirinkti šablonus“
	Specializuoti PA	
Pagrindinis scenarijus		
Pagrindinis įvykių srautas		Sistemos reakcija

Po-sąlyga	Šablonas įterptas į modelį

18 pav. Pateikiama panaudojimo atvejo veiklos diagrama.

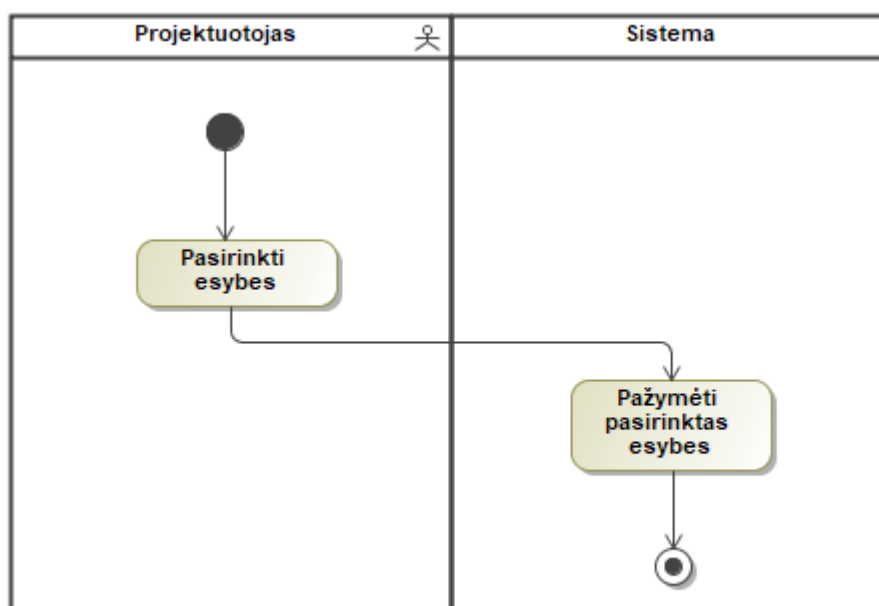


18 pav. Įterpti pasirinktus

Lentelė 8 Panaudojimo atvejo „Pasirinkti esybes“

PA6 „Pasirinkti esybes“		
Tikslas/uždavinys. Pasirinkti esybes		
Aprašymas. Vykdam šį PA sistemos naudotojas gali pasirinkti esybes		
Prieš-sąlyga	Naudotojas yra esybių sąrašė	
Aktorius	Projektuotojas, Administratorius	
Sužadavimo sąlyga	Naudotojas nori pasirinkti esybes	
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuoti PA	
Pagrindinis scenarijus		
Pagrindinis įvykių srautas		Sistemos reakcija
Po-sąlyga	Pažymėtos pasirinktos esybės	

19 pav. Pateikta panaudojimo atvejo veiklos diagrama.



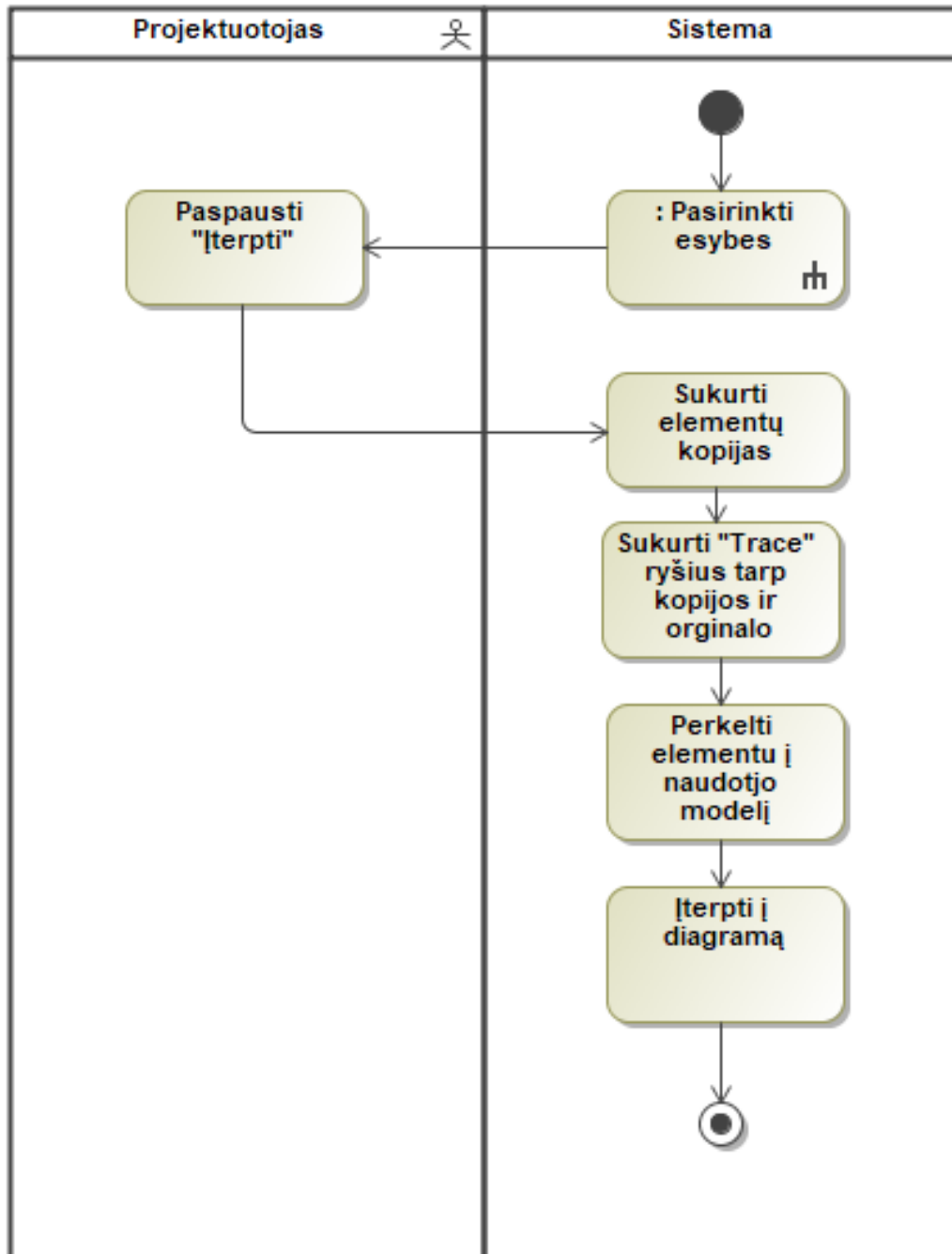
19 pav. Pasirinkti esybes

Lentelė 9 Panaudojimo atvejo „Įterpti pasirinktas esybes“

PA7 „Įterpti pasirinktas esybes“	
Tikslas/uždavinys. Įterpti pasirinktas esybes	
Aprašymas. Vykdam šį PA sistemos naudotojas gali įterpti pasirinktas esybes	
Prieš-sąlyga	Naudotojas yra esybių sąrašė ir pažymėjo bent vieną esybę

Aktorius	Projektuotojas, Administratorius	
Sužadavimo sąlyga	Naudotojas nori įterpti pasirinktas esybes	
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	PA6 „Pasirinkti esybes“
	Specializuoti PA	
Pagrindinis scenarijus		
Pagrindinis įvykių srautas		Sistemos reakcija
Po-sąlyga	Įterptos esybės į modelį	

20 pav. Pateikiama panaudojimo atvejo veiklos diagrama.



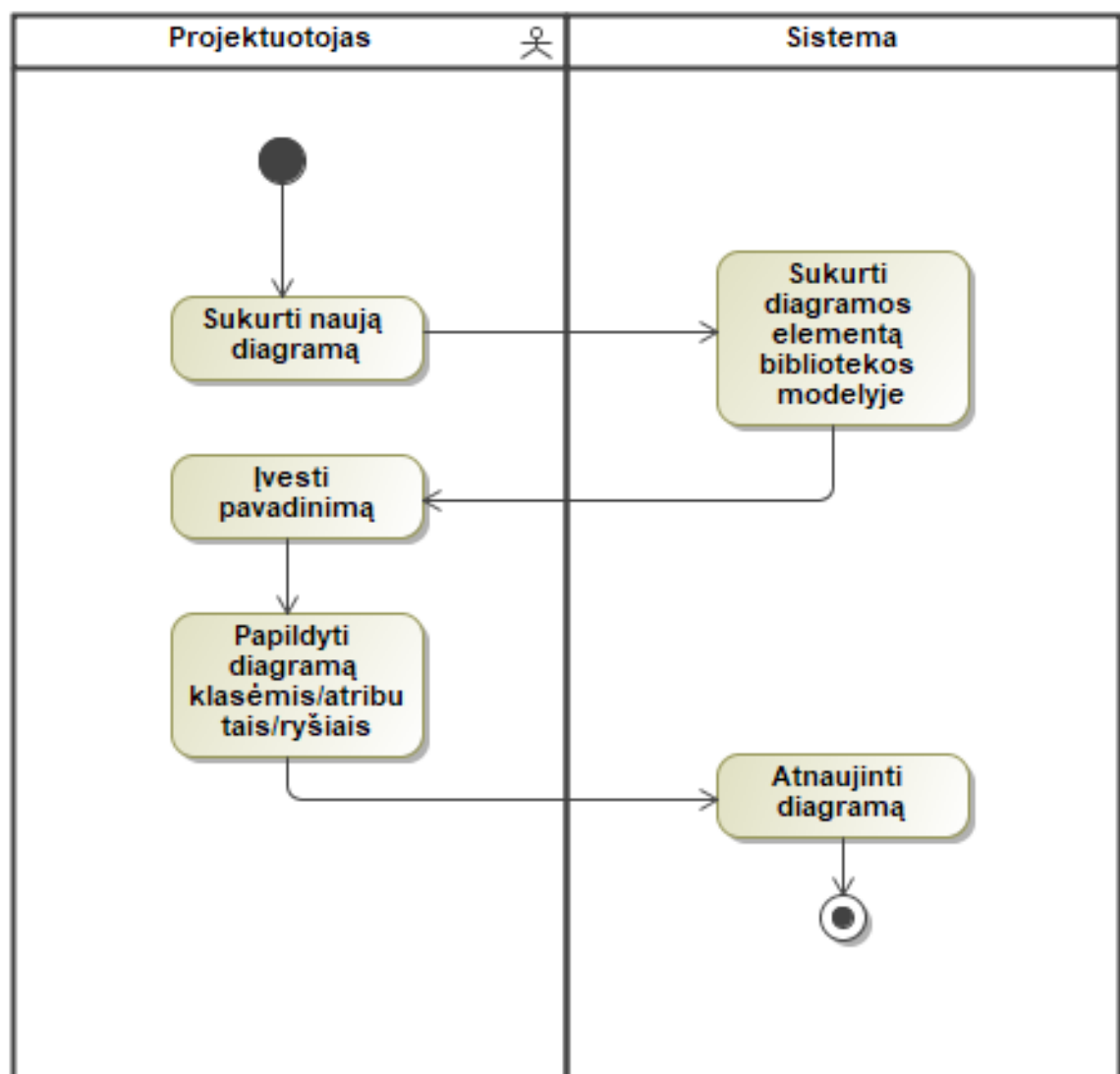
20 pav. Įterpti pasirinktas esybes

Lentelė 10 Panaudojimo atvejo „Įterpti šablona“

PA8 „Įterpti šablona“	
Tikslas/uždavinys. Pridėti naują šablona į biblioteka	
Aprašymas. Vykdanč šį PA sistemos administratorius gali pridėti naują šablona į biblioteka	
Prieš-sąlyga	Naudotojas yra šablono pridėjimo lange
Aktorius	Administratorius

Sužadavimo sąlyga	Naudotojas nori pridėti šabloną	
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuoti PA	
Pagrindinis scenarijus		
Pagrindinis įvykių srautas		Sistemos reakcija
Po-sąlyga	Pridėtas naujas šablonas į biblioteką	

21 pav. Pateikiama panaudojimo atvejo diagrama.

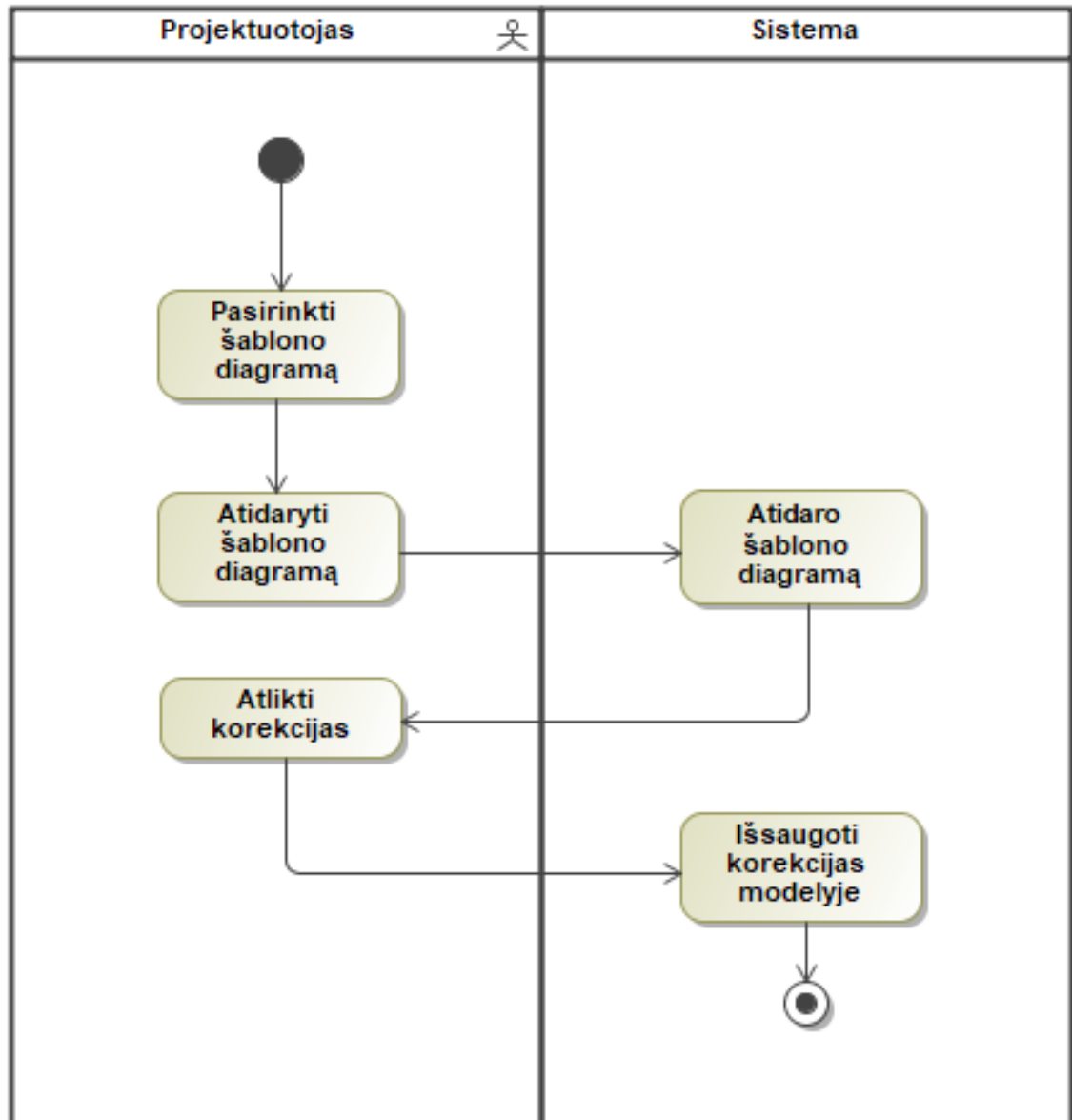


21 pav. Įterpti šabloną

Lentelė 11 Panaudojimo atvejo „Redaguoti šabloną“

PA9 „Redaguoti šabloną”		
Tikslas/uždavinys. Redaguoti šabloną		
Aprašymas. Vykdam šį PA sistemos naudotojas gali pakeisti šabloną		
Prieš-sąlyga	Naudotojas yra šablono redagavimo lange	
Aktorius	Projektuotojas, Administratorius	
Sužadavimo sąlyga	Naudotojas nori pakeisti šabloną	
Susiję panaudojimo atvejai	Išplečiantys PA	
	Apimami PA	
	Specializuoti PA	
Pagrindinis scenarijus		
Pagrindinis įvykių srautas		Sistemos reakcija
Po-sąlyga	Pakeistas šablonas	

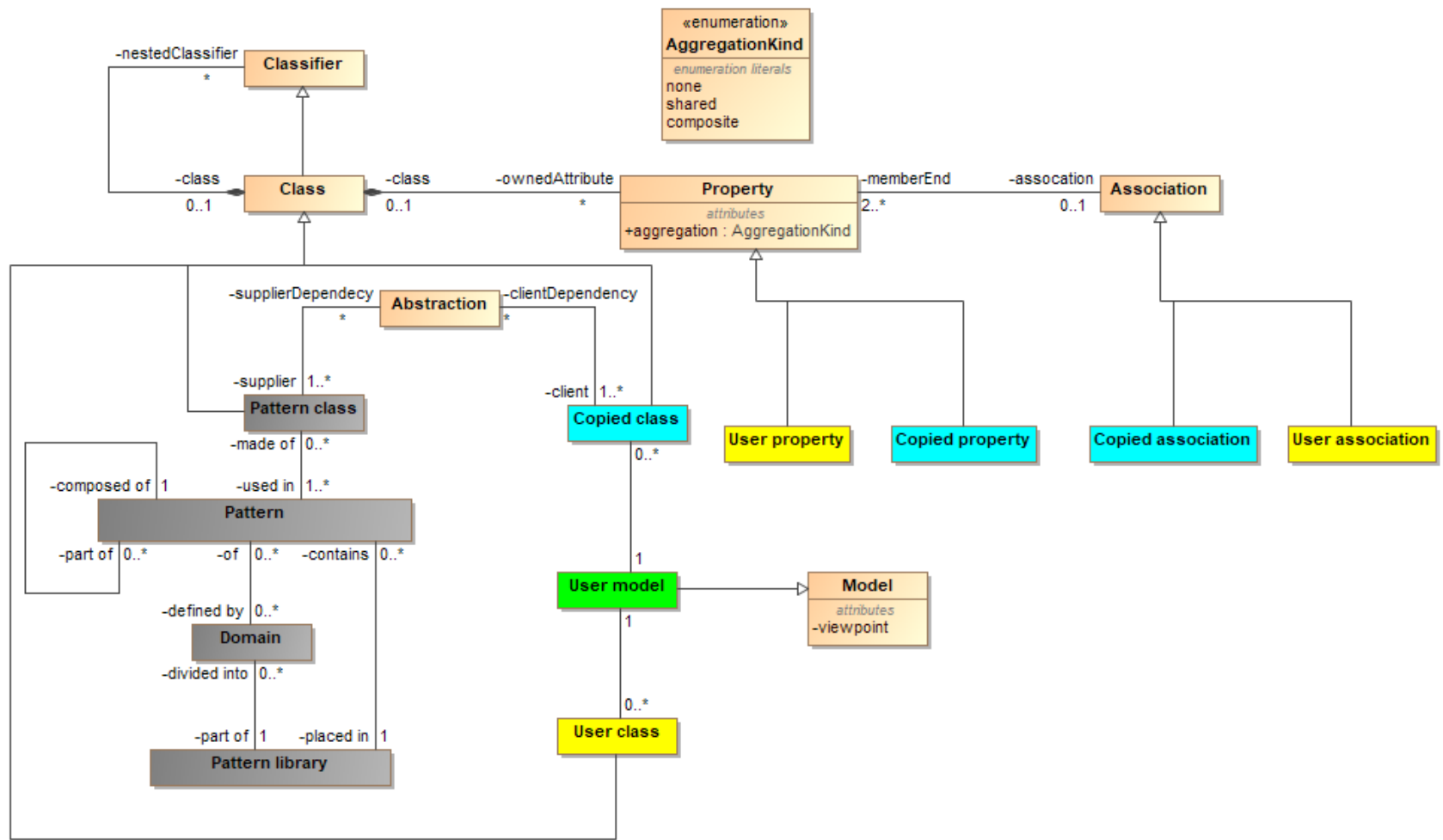
22 pav. Pateikiama panaudojimo atvejo veiklos diagrama.



22 pav. Redaguoti šablona

2.3. Dalykinės srities modelis

Šiame skyriuje pateikiamas nagrinėjamos temos bei kuriamo įrankio dalykinės srities modelis.



23 pav. Dalykinės srities modelis

Dalykinės srities modelis, kuris pateiktas 23 pav. yra suskaidytas ir pažymėtas atitinkamomis spalvomis. Dalis modelio atitinka klasių diagramas metamodelio dalį. Jos aprašytos 12 lentelėje.

Lentelė 12 Klasių diagramos metamodelio elementų aprašymas

Pavadinimams	Aprašymas
Class	Tai elementas, kuris aprašo aibes objektų turinčių tuos pačius atributus, operacijas ryšius ir semantiką.
Property	Tai savybė, kuri aprašo klasę, atributas gali priklausyti tik vienai klasei.
Association	Apibrėžia semantinį ryšį, kuris gali atsirasti tarp egzempliorių. Ryšys nurodomas tarp dviejų egzempliorių, vienas iš jų yra „šaltinis“ (<i>angl. source</i>) kitas „taikinytis“ (<i>angl. target</i>). Asociacija gali būti skirtingų tipų, paprasta asociacija, kompozicija, agregavimas, apibendrinimas. Vienu metu gali būti TIK vieno tipo.
Classifier	Tai elementas, kuris apibūdina atributo tipą. Šis elementas taip pat gali būti bet kokio primityvaus duomenų tipo (Integer, Boolean, Char...)
Abstraction	Tai elementas, kuris skirtas nurodyti ryšį tarp elementų.
Model	Tai elementas, kuris atitinka paketą, tai aukščiausio lygio paketas, kuris yra sukuriamas sukuriant projekto failą.

Sekanti dalis, elementai, kurie diagramoje pavaizduoti pilka spalva yra duomenų modelių šablonų bibliotekos dalis. Šie elementai aprašyti 13 lentelėje.

Lentelė 13 Šablonų bibliotekos elementų aprašymas

Pavadinimas	Aprašymas
Pattern	Klasė atspindinti šabloną bibliotekoje, joje nurodomas šablono pavadinimas bei jo abstrakcijos lygis. Kiekvienas šablonas gali priklausyti daugeliui sričių arba nepriklausyti nei vienai. Kiekvieną šabloną sudaro šablono klasės „Pattern class“ bei šablonas gali būti kito šablono dalis.
Pattern class	Elementas atspindi šablone esančias klases. Negali egzistuoti nepriklausydamas nei vienam šablonui, tačiau vienu metu gali priklausyti daugeliui šablonų.
Domain	Elementas atspindis šablonų bibliotekos sritis. Bibliotekoje šablonai gali būti skaidomi į tam tikras dalykines sritis. Elementas privalo priklausyti bibliotekai.
Pattern library	Tai pagrindinis elementas „Šablonų biblioteka“.

Bibliotekos elementai yra pasiekiami naudotojui tačiau šių elementų modifikacija negalima.

Kita duomenų modelio dalis yra vartotojo nauji elementai, kuriuos vartotojas gali kurti plėsdamas kuriamą duomenų modelį. Šie elementai dalykinės srities diagramoje pavaizduoti geltona spalva. Šie elementai aprašyti 14 lentelėje.

Lentelė 14 Naudotojo sukurtų elementų aprašymas

Pavadinimas	Aprašymas
-------------	-----------

User class	Tai elementas, kuris yra sukuriamas naudotojo, ši elementą sukuria naudotojas ir jis neturi jokio ryšio su šablonų biblioteka.
User property	Tai elementas, kuri sukuria naudotojas, šis elementas naudojamas apibrėžiant „User class“. Taip pat jis gali būti naudojamas norint papildyti klasės, kurios buvo nukopijuotos ir perkeltos į naudotojo modelį iš šablonų bibliotekos.
User association	Tai asociacijos ryšys sukurtas naudotojo, naudojamas parodyti semantiniams ryšiams tarp naudotojo sukurtų klasių.

Mėlyna spalva dalykinės srities diagramoje pavaizduoti elementai, kurie yra perkeliama ir naudojami iš šablonų bibliotekos, bei kuriuos galima modifikuoti naudotojo modelyje. Įprastai tai yra šablonų bibliotekoje esančių šablonų klasių bei jų atributų elementų kopijos. Šie elementai aprašyti 15 lentelėje.

Lentelė 15 UML metamodelio praplėstų elementų aprašymas

Pavadinimas	Aprašymas
Copied class	Tai šablonų bibliotekoje esančių klasių kopijos, kurios yra prieinamos naudotojui, šiuos elementus, esant poreikiu, naudotojas gali modifikuoti. Gali pakeisti pavadinimą, pridėti atributų tiek iš šablonų bibliotekoje esančiu šiai klasei tiek naujų.
Copied property	Tai šablonų bibliotekoje esančių šabloninių klasių atributų kopijos. Šiuos elementus savo modelyje gali modifikuoti naudotojas. Pakeisti pavadinimas, tipą...
Copied association	Tai šablonų bibliotekoje esančių klasių ryšių kopijos, kurios yra perkeltos į naudotojo modelį, šias kopijas vartotojas gali modifikuoti.

Žalia spalva pavaizduotas elementas „User model“ tai duomenų modelis kurį kuria naudotojas naudojantis MagicDraw įrankiu, bei pateikta duomenų modelių šablonų biblioteka ir parengtu įrankiu.

2.4. Formalus sprendimo aprašas

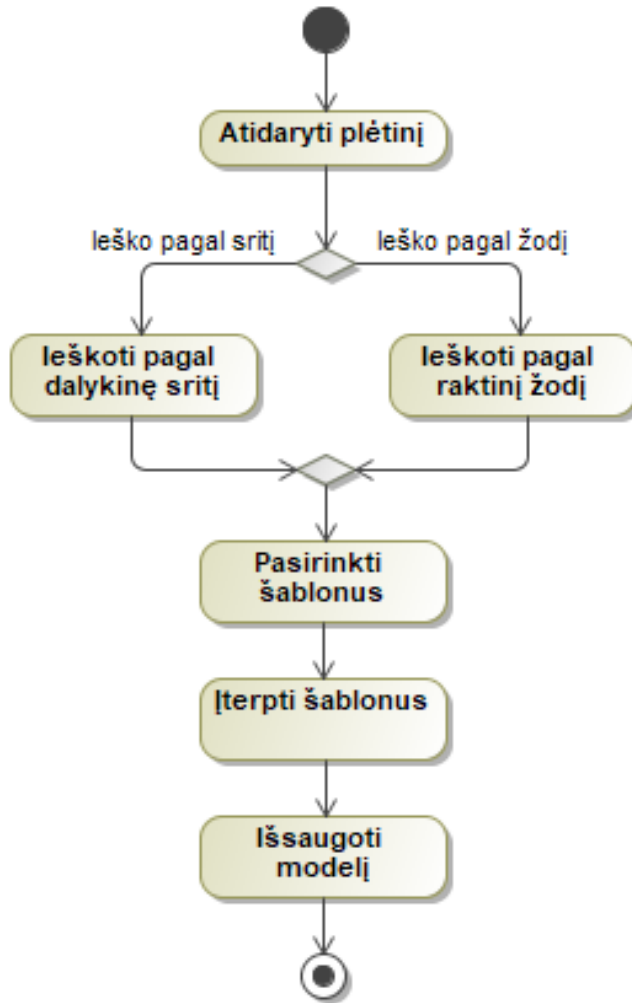
2.4.1. Sprendimo procesai

Šiame skyriuje pateiki tipiniai veiklos procesai, pagal kuriuos kuriamas duomenų modelis, bei procesai, kuriais bus papildytas duomenų modelio kūrimo procesas realizavimus sprendimą.

2.4.2. Pasiūlyti procesai

Analizes metu buvo įvardinta problema dėl pasikartojančių žingsnių tipiniame koncepcinio modelio kūrimo procese. Siekiant sumažinti šių besikartojančių žingsnių skaičių kuriant naujus duomenų modelius yra pateikiami 2 duomenų modelio kūrimo proceso papildymai naudojantis duomenų modelių šablonų biblioteka.

Pirmajame variante pateiktame 24 pav. yra pateikiamas procesas kai vartotojas gali įterpti atitinkamas modelio dalis iš pateiktos šablonų bibliotekos. Šio proceso eiga:



24 pav. Duomenų modelio kūrimo procesas naudojantis šablonų biblioteka

Proceso vykdymas priklauso nuo to kaip naudotojas nori ieškoti šablonų pagal dalykinę sritį, kuriai kuriamas duomenų modelis ar pagal raktinį žodį pvz. Organization, Person...

1 žingsnis – atidaryti plėtinio langą su šablonų sąrašu.

2 žingsnis – pasirinkti dalykinę sritį arba įvesti raktinį žodį.

3 žingsnis – pasirinkti norimus šablonus.

4 žingsnis – įterpti šablonus, juose esančių klasių atributus, ryšius bei išsaugoti modelį

PSEUDO CODE

```

BEGIN
Atidaryti pasirinkimo langą
Surinkti šablonus ir jų esybes iš bibliotekos
Surinkti šablonų esybes iš vartotojo modelio
If paspausta "OK" pasirinkimo lange:
    Sukurti pasirinktų elementų kopijas
    For copy in copies:
  
```

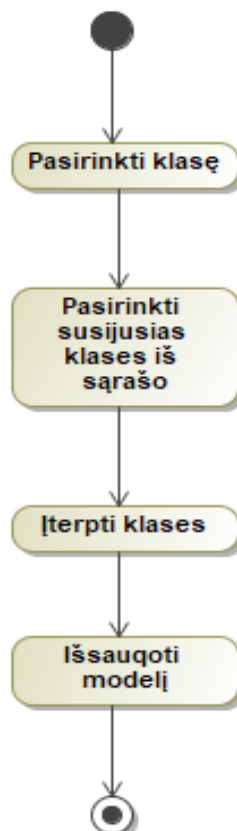
```

Sukurti atgalinį "Trace" ryšį.
Pašalinti papildomus "Property" elementus
Sukurti prezentacijos elementus
For šablono_element in pasirinkti_šablono_elementai:
    For esamas_elementas in esami_elementai:
        Surinkti ryšius
For ryšys in ryšiai:
    Pakeisti ryšio galo elementą iš šablono elemento į jo kopijos
    elementą vartotojo modelyje
Įkelti prezentacijos elementus į diagram
Įvykdyti "Display all paths"
Uždaryti kopijavimo sesija
End if
END

```

Atitinkamai pagal pasirinktus šablonus (kiek klasių buvo šablone) sumažėja tipinio proceso pasikartojančių žingsnių skaičius, taip pat sumažėja ir trukmė, kurios reikia sukurti naują duomenų modelį.

Antras papildymas pateiktas 25 pav. yra tuomet kai yra pasinaudota duomenų modelių šablonų biblioteka ir pastebėtas esybių trūkumas. Tokiu atveju gali būti vykdomas šis procesas:



25 pav. Duomenų modelio papildymo procesas naudojantis šablonų biblioteka

Proceso vykdymas priklauso nuo poreikio, įterpti klases ar įterpti atributus. Tipinis proceso vykdymo scenarijus:

1 žingsnis – pasirinkti klasę iš kuriamo modelio, kuri buvo įterpta iš šablonų bibliotekos.

2 žingsnis – pasirinkti susijusias klases.

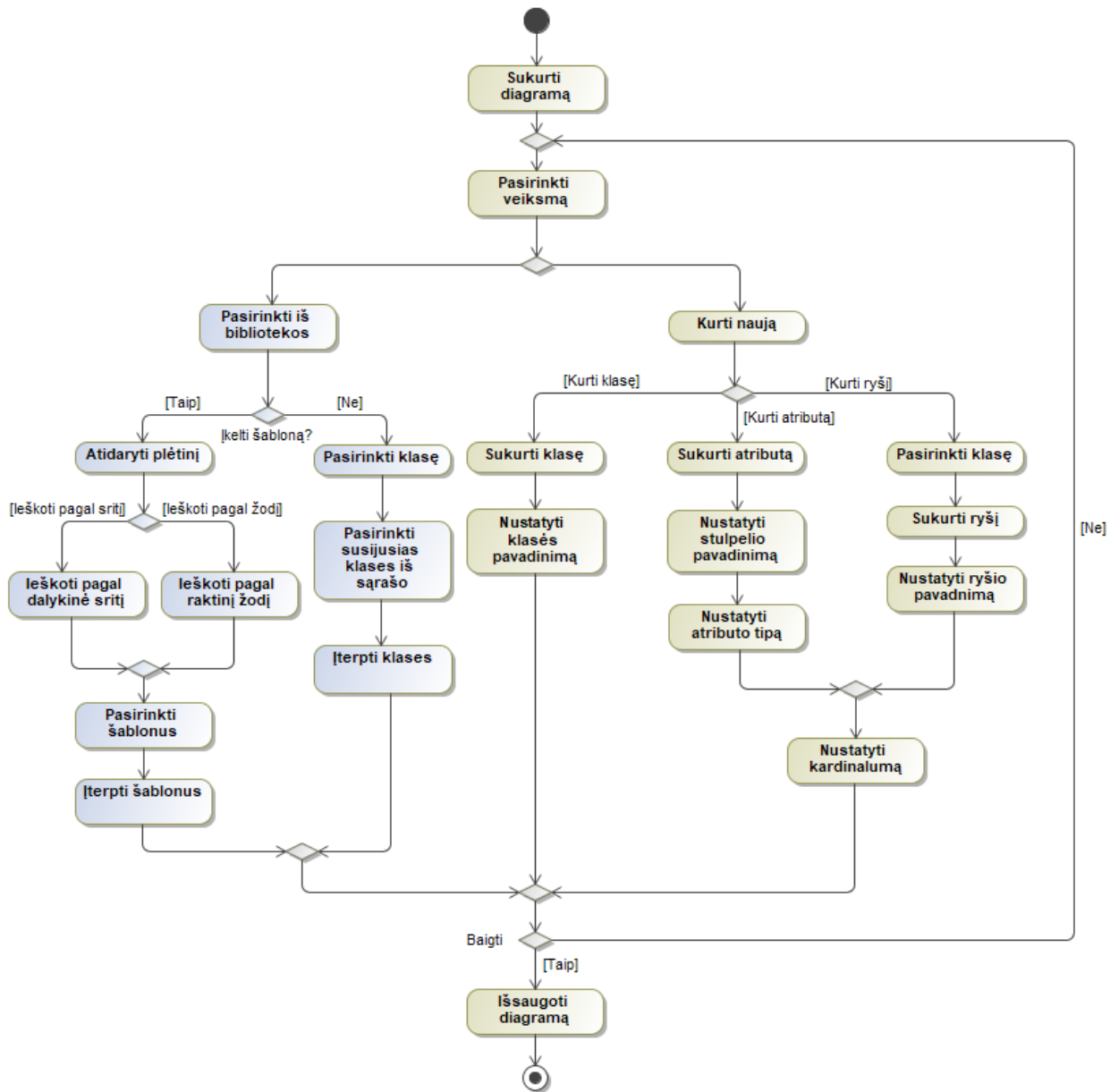
3 žingsnis – įterpti klases, klasių atributus bei visus galimus ryšius tarp pasirinktų ir egzistuojančių klasių.

PSEUDO CODE

```
BEGIN
Atidaryti pasirinkimo langą
Surasti esybės originalą šablonų bibliotekoje
Surinkti susijusius šablonus ir jų esybes iš bibliotekos
Surinkti šablonų esybes iš vartotojo modelio
If paspausta "OK" pasirinkimo lange:
    Sukurti kopijavimo sesija
    Sukurti pasirinktų elementų kopijas
    For copy in copies:
        Sukurti atgalinį "Trace" ryšį.
        Pašalinti papildomus "Property" elementus
        Sukurti prezentacijos elementus
    For šablono_element in pasirinkti_šablono_elementai:
        For esamas_elementas in esami_elementai:
            Surinkti ryšius
    For ryšys in ryšiai:
        Pakeisti ryšio galo elementą iš šablono elemento į jo kopijos
        elementą vartotojo modelyje
    Įkelti prezentacijos elementus į diagram
    Įvykdyti "Display all paths"
    Uždaryti kopijavimo sesija
End if
END
```

Vykdamas šį veiklos procesą suteikia galimybę modifikuoti pateiktus šablonus, taip dar kartą sumažinamas poreikis kurti esybes bei atributus ir paspartinamas duomenų modelio kūrimo procesas.

Šiais dviem pasiūlytais procesais gali būti praplėstas tipinis duomenų modelio kūrimo metodas. Pralėsto duomenų modelio kūrimo diagrama pateikiama 26 pav.



26 pav. Tipinio duomenų modelio kūrimo proceso praplėtimas

Panaudojant praplėsta metodą galima sumažinti perteklinių veiksmų kiekį, kuris buvo reikalingas naudojant įprastą tipinį duomenų modelio kūrimo metodo.

2.5. Reikalavimų apibendrinimas

Šiame skyriuje buvo pateikti reikalavimai keliami duomenų modelių šablonų bibliotekai bei įrankiui. Apibendrinant šiuos reikalavimus galimas išskirti šiuos pagrindinius keliamus reikalavimus:

1. Sistema privalo leisti naudotojui ieškoti esybių ir šablonų
2. Sistema privalo leisti naudotojui įsikelti esybes ir šablonus į savo modelį
3. Sistema privalo kitaip atvaizduoti šablonines esybes naudotojo modelyje.
4. Sprendimas privalo praplėsti esamą tipinį duomenų modelio kūrimo procesą.

Šie pagrindiniai reikalavimai yra keliami kuriamam sprendimui.

3. Duomenų modelių šablonų sprendimo eksperimentinės realizacijos projektas

Pagal nagrinėjama problemą jos sprendimui buvo pasiūlyta parengti duomenų modelių šablonų biblioteką UML formatu pasinaudojant MagicDraw įrankiu bei parengti MagicDraw plėtinį tam, kad naudotojas galėtų patogiai naudotis parengta biblioteka. Duomenų bibliotekai parengti buvo pasinaudota įvairių autorių patirtimi, ši patirtis buvo užfiksuota knygose bei kituose leidiniuose. Naudojantis šiais leidiniais siekiama parengti duomenų modelių biblioteką elektroniniame formate, naudojant UML notaciją. Analizės dalyje nustatyta, kad UML yra plačiausiai naudojama modeliavimo kalba. Biblioteka yra MagicDraw projektas kuriame struktūriškai, pagal autorius, abstrakcijos lygį, dalykinę sritį yra sudėlioti duomenų modelių šablonai. Bibliotekoje yra naudojami klasės (angl. *Class*) tipo elementai bei ryšiai tarp jų. Kadangi dauguma skirtingų autorių pateikė panašius šablonus skirtingai. Buvo naudojamas atsekamumo (angl. *Trace*) ryšys tam, kad būtų galima atsekti panašumus tarp skirtingų autorių šablonų bei juose naudojamų elementų.

Kita šio sprendimo dalis yra MagicDraw plėtinys. Plėtinio tikslas yra sukurti galimybę pasinaudoti parengtu bibliotekos projektu. Taip siekiama pagreitinti bei palengvinti duomenų bazės projektavimą. Plėtinyje galima išskirti 3 pagrindines funkcines grupes:

- Paieška;
- Įterpimas;
- Atvaizdavimas.

Tolimesniuose poskyriuose pateikiama detalesnė informaciją apie plėtinio realizacijos projektą.

3.1. Realizacijos technologijos

MagicDraw įrankis yra sukurtas JAVA programavimo kalbos pagrindu, todėl plėtinys taip pat bus kuriamas naudojant JAVA programavimo kalba. Plėtiniui kurti bus panaudotas Intellij IDEA įrankis bei MagicDraw 18.5 OpenAPI dokumentacija.

Duomenų šablonų bibliotekai parengti bus pasinaudojama pačiu įrankiu kuriam kuriamas plėtinys. MagicDraw patogus įrankis kurti UML modelius.

3.2. Šablonų bibliotekos sudarymas

Šablonų bibliotekai sudaryti buvo naudojamas MagicDraw modeliavimo įrankis. Buvo pasirinkti dviejų autorių [13] [14] duomenų modelių šablonai, kuriuos jie pateikė savo knygose. Daugumos modelių, kuriuos pateikė David C. Hay alternatyvas taip pat pateikė ir Len Silverston. Len Silverston pateikti duomenų modeliai buvo detalesni, tačiau sunkiau perprantami, kadangi bibliotekoje pateikti modeliai bus naudojami ir pradedančių projektuotų buvo pasirinkti palikti tik David C. Hay duomenų modelių šablonų versijas, kai kurias esybes papildant atributais, kurie buvo pateikti Len Silverston šablonuose. Sudarant biblioteką buvo stengiamasi įvertinti atributų poreikį kuriant modelius. Kadangi atributai dažniausiai priklauso nuo konkrečios kuriamos sistemos, šablonų esybėse buvo nuspręsta palikti tik pagrindinius atributus, taip sumažinant galima poreikį vėliau perkėlus esybę į konkretų modelį naikinti perteklinius atributus. Kuriant įrankį buvo numatyta, kad naudotojai galėtų apjungti įvairius šablonus per pasikartojančias esybes juose. Šiam funkcionalumui pasiekti kuriant biblioteką buvo pasirinkta naudoti tik vieną konkrečios esybės elementą modelyje. Bibliotekos struktūra buvo sudaryta išskiriant šablonus pagal autorius. Tuomet

kiekvieno autoriaus šablonai buvo kuriant atskiruose paketuose, sukuriant konkrečiam šablonui unikalias esybes ir diagramą, o likusias reikalingas esybes naudojant iš ankščiau sukurtų šablonų.

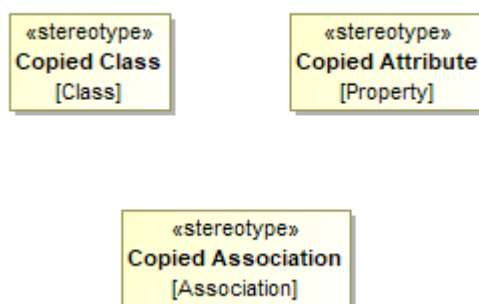
3.3. UML metamodelio praplėtimas.

Pagal nustatytus reikalavimus nuspręsta šablono nukopijuotas klases pažymėti kitaip vartotojo modelyje. Šiame tikslui pasiekti buvo nuspręsta sukurti 3 papildomus stereotipus, kurie pateikti 33 pav.

Copied Class – sukurtas stereotipas, kuris bus pritaikomas Class elementui. Stereotipas nustatomas kopijuojant klases iš duomenų modelių šablonų ir pritaikomas kiekvienam Copied Class elementui.

Copied Attribute – sukurtas stereotipas bus pritaikomas Attribute elementui. Stereotipas pritaikomas kopijuojant atributus iš šablonų bibliotekos ir pritaikomas kiekvienam Copied attribute elementui.

Copied Association – stereotipas bus pritaikomas Association elementui. Stereotipas pritaikomas kopijuojant elementus iš duomenų modelių šablonų bibliotekos.

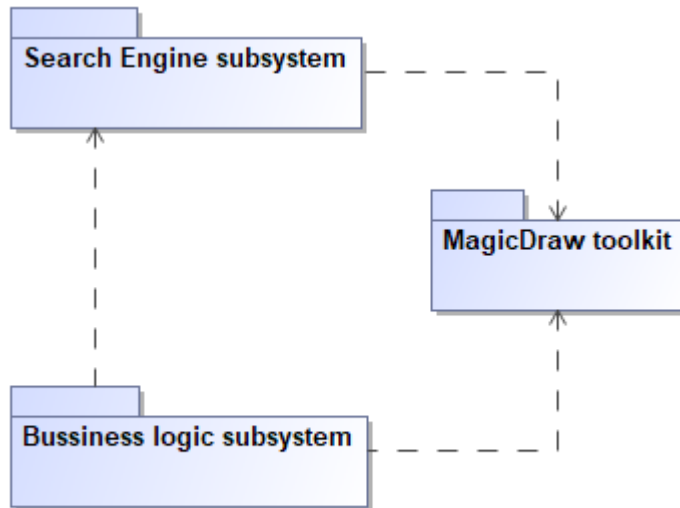


27 pav. Sukurti stereotipai

Šie stereotipai pateikti 27 pav. pritaikomi kopijuojant elementus iš šablonų bibliotekos

3.4. Sistemos loginė architektūra

Skyriuje pateikia kuriamo plėtinio loginės architektūros diagrama bei trumpai aprašoma jos elementų paskirtis.



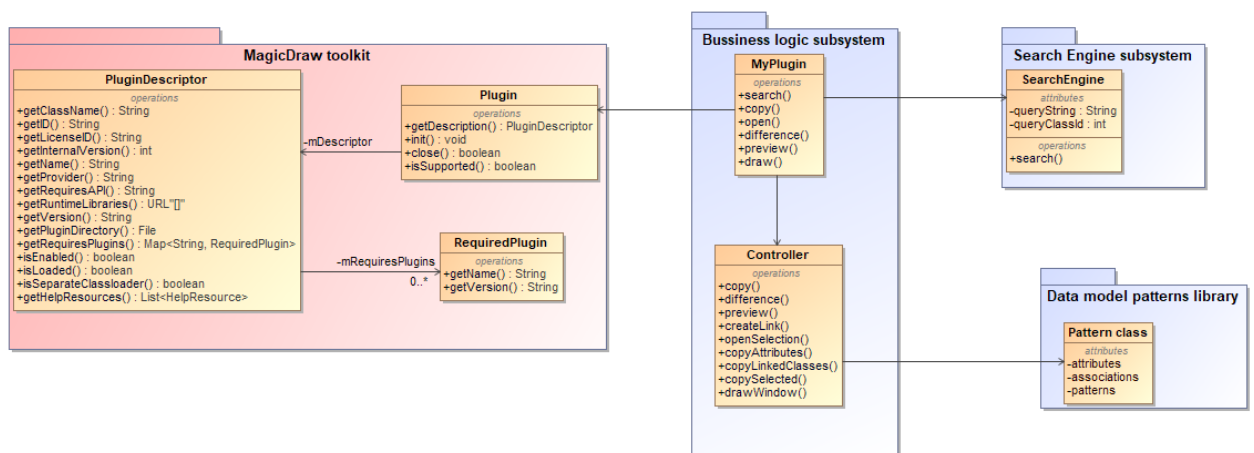
28 pav. Sistemos loginė architektūra

Loginės architektūros elementai pateikti 28 pav.:

- Search engine subsystem – elementas skirtas atvaizduoti sistemos dalį atsakingą už paieškos funkcionalumą.
- MagicDraw toolkit – elementas skirtas atvaizduoti pasirinkto įrankio MagicDraw pateikiamą programinį kodą reikalingą realizuoti plėtinį MagicDraw aplinkoje.
- Bussiness logic subsystem – elementas kuriame bus klasės atsakingos už plėtinio veikimo logiką.

3.5. Sistemos klasių diagrama

Skyriuje pateikiama sistemos klasių diagrama, sistemoje pagrindinė klasė yra *Plugin*, kuri naudojasi pagalbinių klasių funkcijomis.



29 pav. Sistemos klasių diagrama

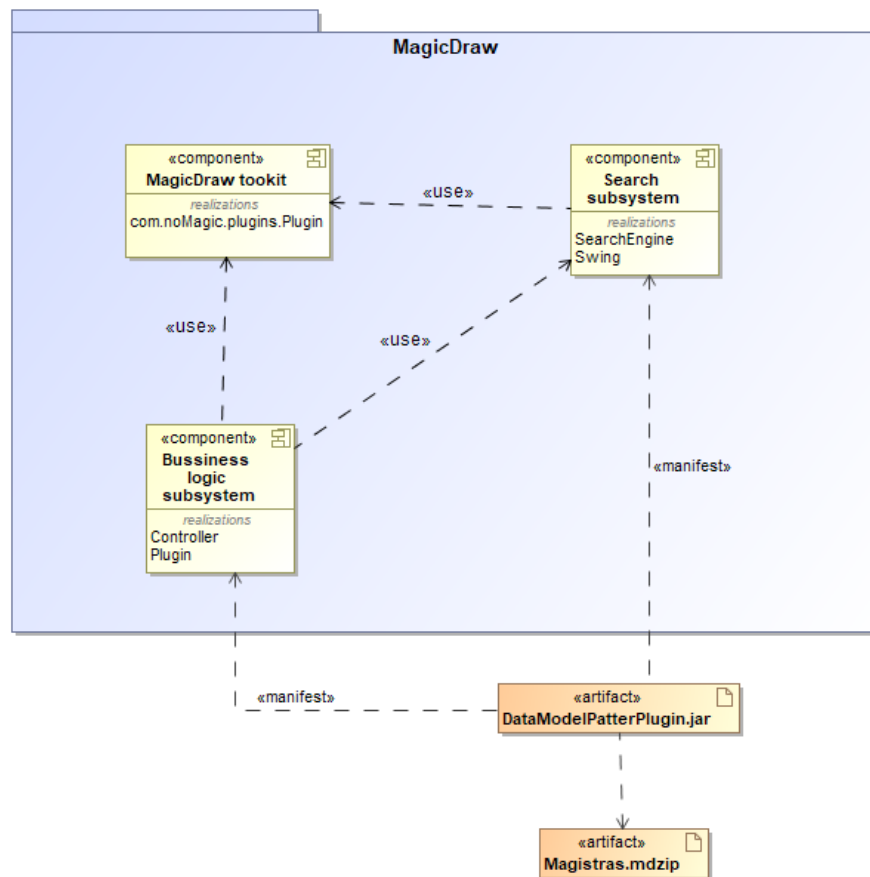
Paveikslėlyje 29 pav. pateikta klasių diagrama. Diagramoje yra nurodytos naudojamos klasės:

Lentelė 16 Klasių aprašas

Klasė	Aprašymas
Com.noMagic.plugins.Plugin	MagicDraw įrankio parengta bendrinė plėtinių klasė kurią paveldint yra kuriame visi kiti plėtiniai galimi MagicDraw įrankyje
Plugin	Klasė kuriami ir realizuos visą siekiamą sprendimą. Klasėje nurodyti metodai skirti užtikrinti tinkamą siekiamo sprendimo veikimą
Controller	Klasė yra pagalbiniė Plugin klasei joje aprašoma pagrindinė veikimo logika
SearchEngine	Klasė yra pagalbiniė Plugin klasei joje išskirta logika skirta paieškai sukurtame bibliotekos projekte

3.6. Sistemos komponentų diagrama

Skyriuje pateikiama sistemos komponentų diagrama.

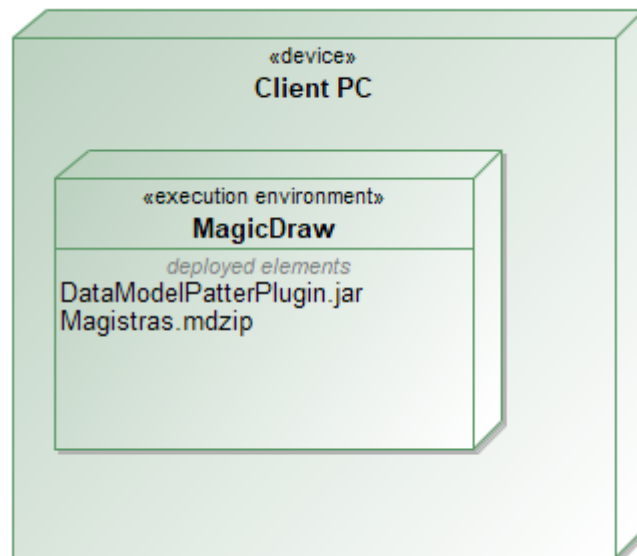


30 pav. Sistemos komponentų diagrama

30 pav. pateiktame paveikslėlyje yra sistemos komponentų diagrama. Diagramoje pateikiamos klasės, kurios realizuoja komponentus bei gaunamas artefaktas visų komponentų. Plėtinys realizuotas Java programavimo kalba todėl gaunamas artefaktas yra Java programa. Kiekvienas komponentas bus MagicDraw įrankyje.

3.7. Sistemos diegimo diagrama

Skyriuje pateikiama sistemos diegimo diagrama. Kadangi realizacijos objektas yra plėtinys jis bus patalpintas MagicDraw įrankyje 31 pav.



31 pav. Sistemos diegimo diagrama

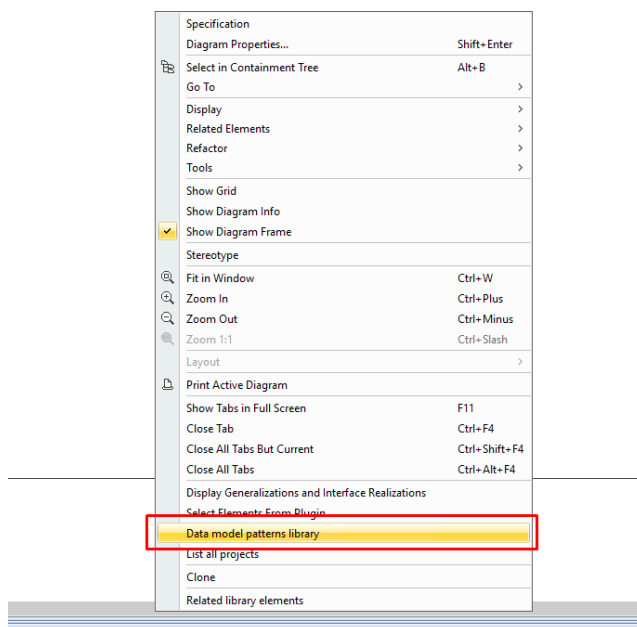
Kiekvienas naudotojas norintis pasinaudoti plėtiniumi privalės jį įsirašyti savo naudojamame kompiuteryje kuriame yra MagicDraw įrankis.

4. Sprendimo realizacija ir testavimas

Šis skyrius skaidomas į du poskyrius, sprendimo realizacijos ir veikimo aprašą bei testavimo modelį, duomenis ir rezultatus.

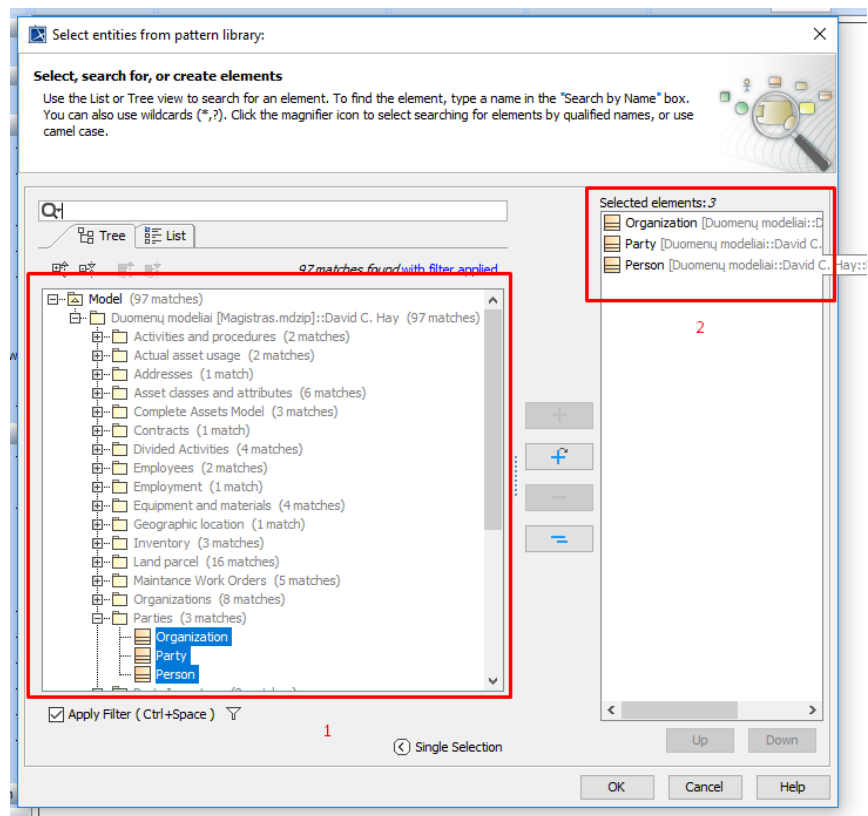
4.1. Sprendimo realizacijos ir veikimo aprašas

Pagal užduotį buvo realizuotas ir sukurtas pirminis siekiamo sprendimo, įskiepio prototipas, kuris jau leidžia naudotojui kurti duomenų modelį pasinaudojant bibliotekoje esančiais šablonais. Naudotojas įsirašęs sukurtą įskiepį ir į naują modelį importavęs šablonų bibliotekos projektą gali pradėti kurti duomenų modelį. Turimoje „Class diagram“ naudotojui paspaudus dešinę pelės mygtuką ir pasirinkus „Data model patterns library“ kaip parodyta 32 pav. yra atidaromas pasirinkimo langas kuriame naudotojas gali rinktis elementus(klases) iš šablono bibliotekos, kurias nori įtraukti į savo kuriamą duomenų modelį.



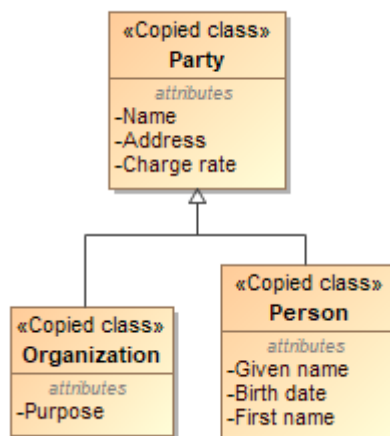
32 pav. Duomenų modelių šablono bibliotekos atidarymas

Naudotojas gali matyti šablonų biblioteką ir joje pateiktus šablonus, pasirinkti elementus, kurie yra bibliotekoje ir įterpti į savo kuriamą modelį kaip pavaizduota 33 pav.



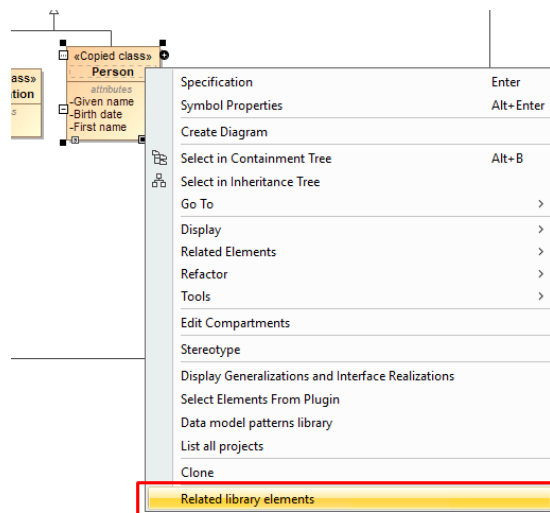
33 pav. Šablonų pasirinkimas bibliotekoje

Paspaudus „Ok“ yra sukuriamos pasirinktų elementų kopijos. Taip pat tam, kad užtikrinti atgalinį ryšį tarp kopijos elemento ir pirminio elemento bibliotekoje yra sukuriamas „Trace“ ryšys. O tam, kad naudotojo modelyje būtų galima atskirti elementus, kurie buvo įkopijuoti iš bibliotekos jiems yra pritaikomas atitinkamas stereotipas. Žemiau pateiktame paveikslėlyje 34 pav. matyti kas gaunama paspaudus „Ok“ ir nustatčius automatinį elementų lygiavimą diagramoje.



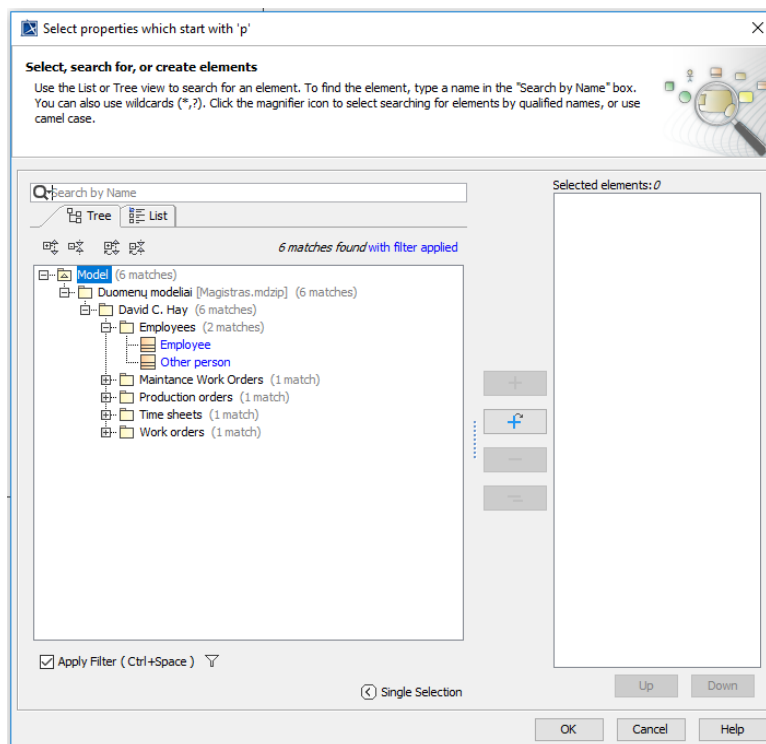
34 pav. Rezultatas gautas įvykdžius šablono įterpimą

Norint toliau plėsti duomenų modelį galima kurti savo klases naudojantis MagicDraw įrankio pateiktomis funkcijomis, arba vėl atsidaryti šablonų biblioteką ir pasirinkti kitus elementus, kurių gali reikėti ir įterpti juos. Jei tarp esamų ir įterpiamų elementų egzistuoja ryšiai bibliotekoje šie ryšiai yra nukopijuojami ir sukuriama naudotojo kuriame modelyje. Dar vienas galimas scenarijus yra pasinaudoti funkcija „Related library elements“, kuri parodyta 35 pav. taip galima pažiūrėti su kuriais kitais bibliotekos elementais yra susijęs pasirinktas elementas ir esant poreikiui įterpti juos į kuriamą modelį.



35 pav. „Related library elements“ funkcija

Įvykdžius funkciją atidaromas langas, kuris atrodo taip pat kaip ir pradinis bibliotekos langas pateikiamas 36 pav. tik elementai yra filtruojami pagal pasirinktą elementą ir parodomi tik tie, kurie turi ryšį su jo pirminiu elementu, kuris yra surandamas pagal ankščiau sukurtą „Trace“ ryšį.



36 pav. Susijusių bibliotekos elementų pasirinkimas

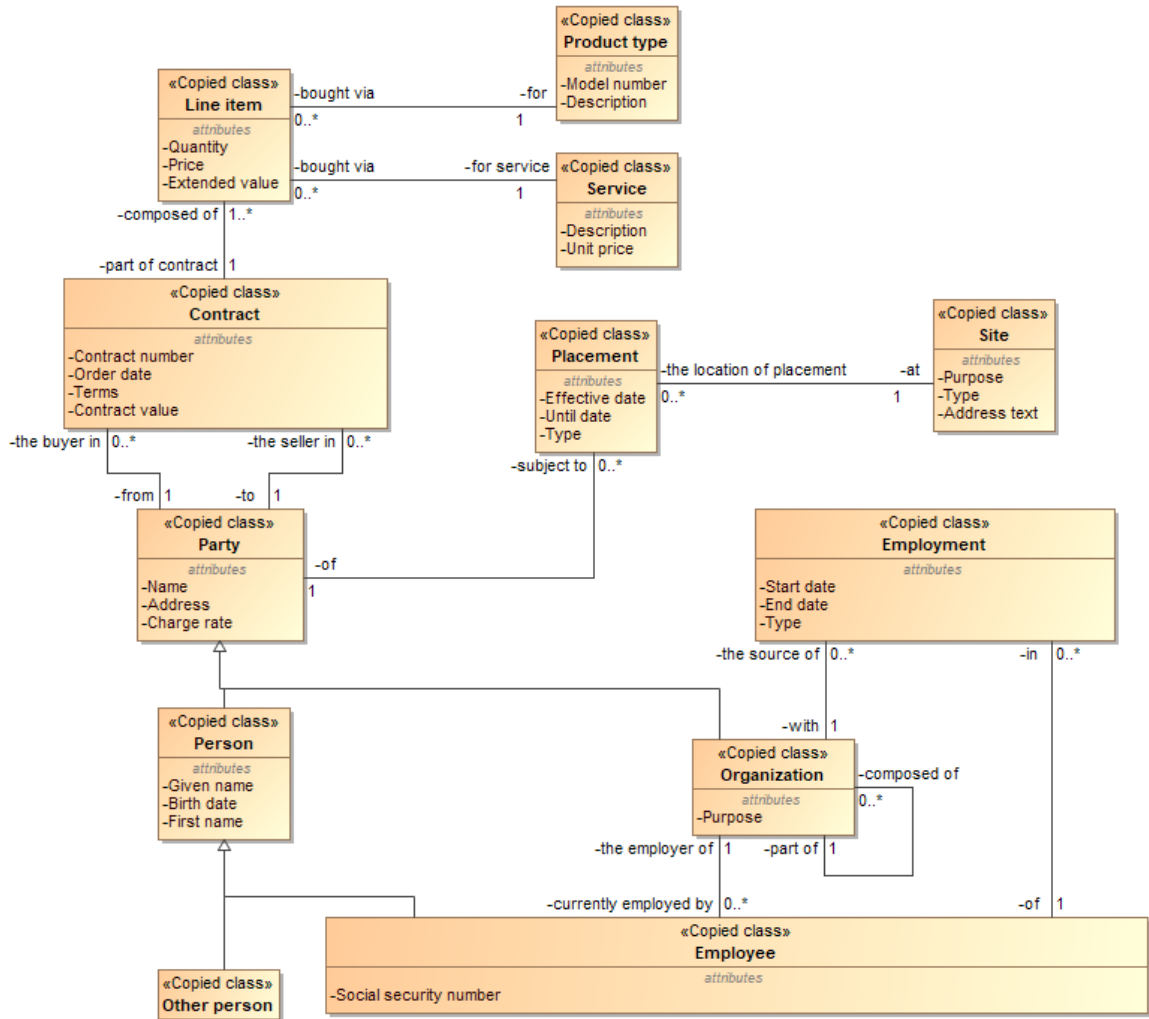
Pasirinkus ir įterpus elementus į naudotojo modelį sukuriama ryšiai ne tik tarp pasirinktos esybės ir pasirinktų naujų elementų iš bibliotekos, bet ir tarp jau esamų elementų naudotojo modelyje ir naujai įkeliamų elementų iš bibliotekos.

4.2. Testavimo modelis, duomenys, rezultatai

Šiame skyriuje pateikimas testavimo modelis ir duomenys, kurie buvo panaudoti bei gauti rezultatai.

4.2.1. Testavimo duomenys

Testavimo duomenų modelis buvo sukurtas taip, kad jį pilnai būtų galima padengti šiuo metu esamais šablonais bibliotekoje.

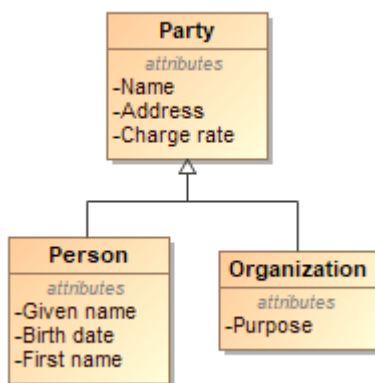


37 pav. Testavimui skirtas modelis

37 pav. Pateiktas testavimo modelis, kuris yra sudarytas iš 6 šablonų, kurie yra bibliotekoje. Toliau bus pateikiamas kiekvienas šablonas, kuris buvo panaudotas sudarant šį testavimo modelį.

4.2.2. Testuojamo modelio kūrimo duomenys

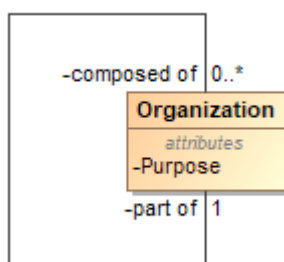
Pirmas iš jų yra „Parties“ šablonas, kuris yra pateikiamas 38 pav.



38 pav. „Parties“ šablonas

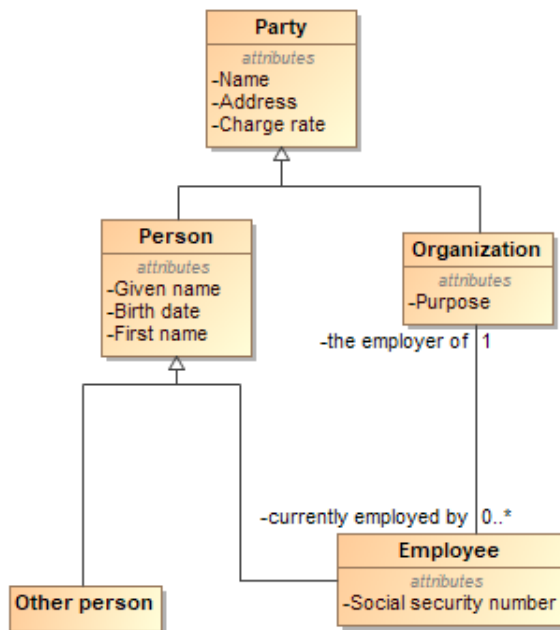
Šis šablonas sudarytas iš 3 esybių. Jame matyti subjektas, kuris gali būti tiek organizacija tiek individualus asmuo. Šablonas yra abstraktus todėl ir atributų skaičius yra labai mažas.

Kitas šablonas, kuris buvo panaudotas kuriant testavimo modelį yra „Organization structure“. Jis pateikiamas 39 pav. Tai šablonas skirtas atvaizduoti organizacijos struktūrai.



39 pav. „Organization structure“ šablonas

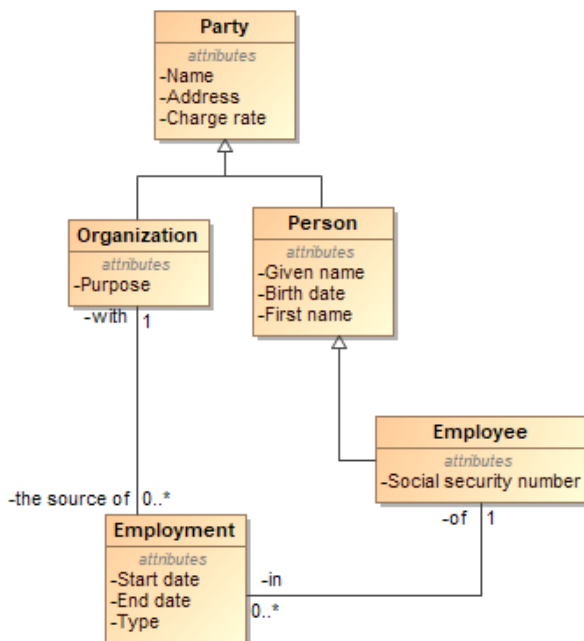
Dar vienas panaudotas šablonas pateikimas 40 pav.



40 pav. „Employees“ šablonas

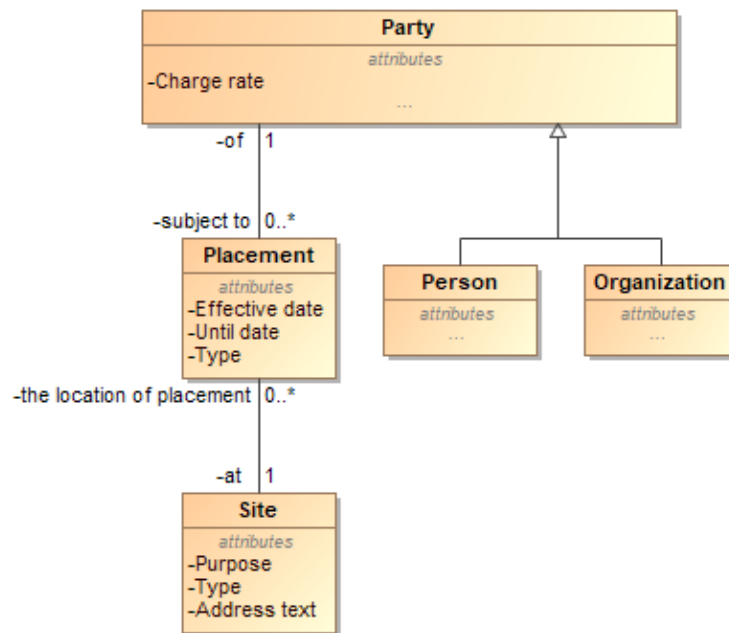
Kaip matoma šis šablonas yra praplėstas „Parties“ šablonas, jis yra papildytas „Employee“ ir „Other person“ esybėmis bei semantiniu ryšių egzistuojančiu tarp „Organization“ ir „Employee“.

Jei yra aktualu turėti informaciją ne tik semantinio ryšio tačiau ir laikotarpio bei istorijos tarp organizacijos ir darbuotojo tam yra panaudojama „Employment“ esybė. Tam yra atskiras šablonas bibliotekoje „Employment“, kuris pateiktas 41 pav.



41 pav. „Employment“ šablonas

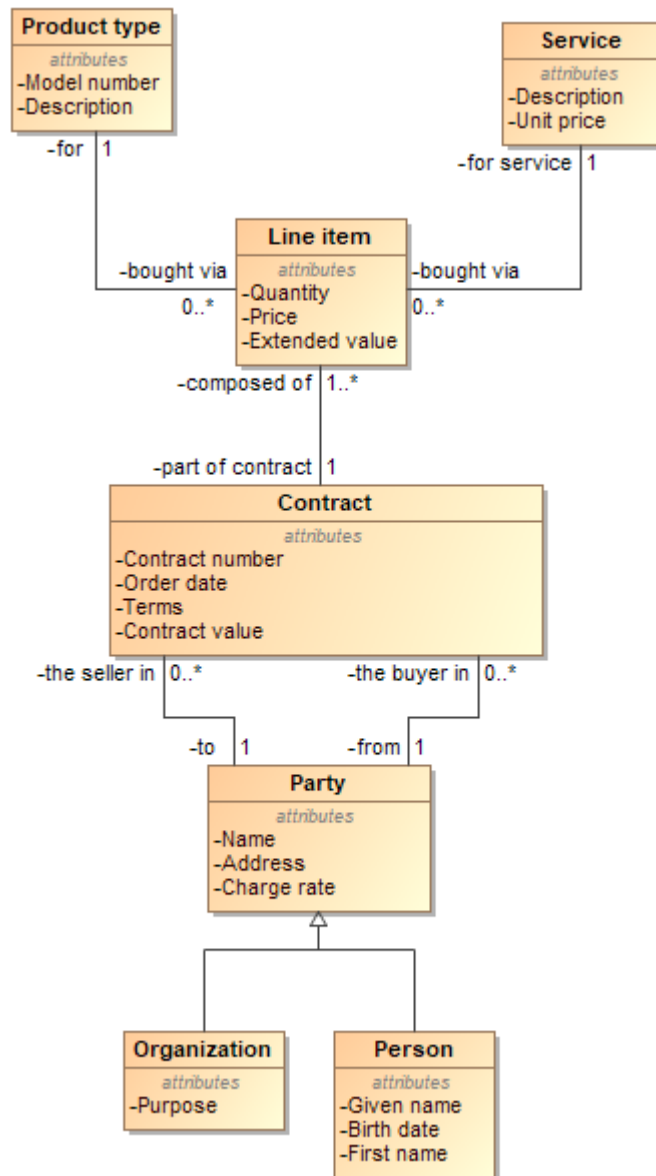
Šablonas suteikia daugiau informacijos valdant darbuotojo ir darbdavio santykius. Dar vienas šablonas panaudotas sukurtame testavimo modelyje yra „Sites“, kuris pateikiamas 42 pav.



42 pav. „Sites“ šablonas

Šis šablonas taip pat yra sudarytas panaudojant „Parties“ šabloną ir suteikia informaciją apie subjekto vietovę.

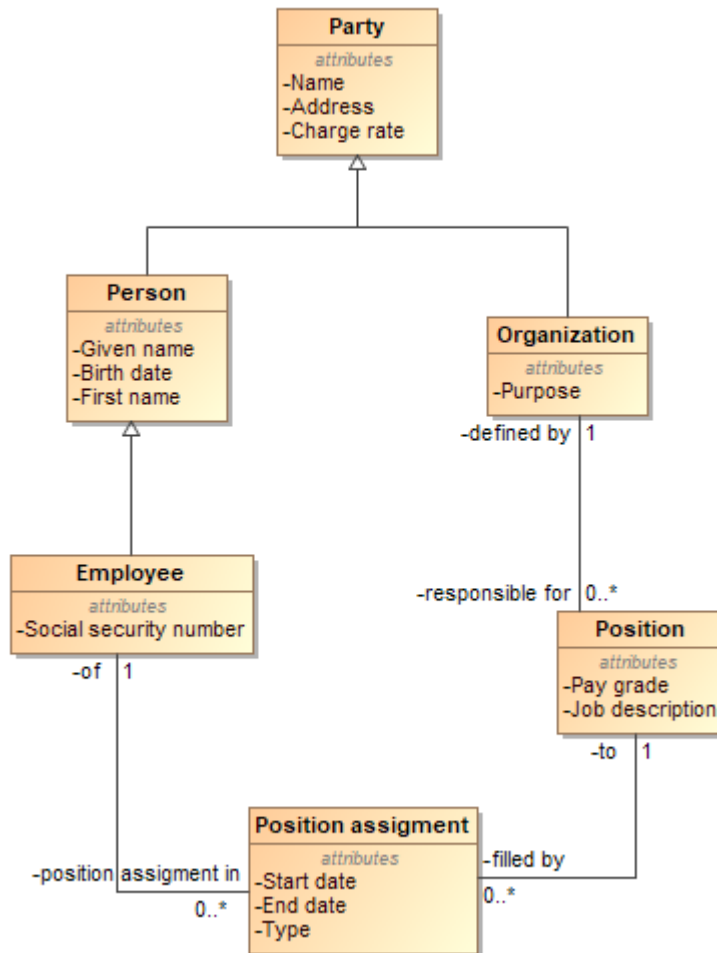
Paskutinis šablonas, kuris yra panaudotas sudarant testavimo modelį yra „Products and services“ jis pateikiamas 43 pav.



43 pav. „Products and services“ šablonas

Šis šablonas taip pat yra sudarytas panaudojant „Parties“ šabloną bei „Contracts“ šabloną. Šablonas pateikia modelį pagal kurį galime saugoti informaciją apie sudarytus kontraktus (angl. *Contract*) tarp subjektų. Bei kontrakte saugomą informaciją apie produktas arba paslaugas, kurios buvo suteiktos subjektui.

Sujungus visus šiuos šablonus ir buvo gautas testavimo modelis. Šis testavimo modelis gali būti naudojamas kaip pagrindas kuriant duomenų modelį įmonės sistemai kurioje reikalinga informacija apie darbuotojus, ir sudaromus kontraktus. Šį modelį praplėtus „Positions“ šablonu, kuris pateikiamas 44 pav. galima praplėsti duomenų bazę ir turėti bazinį personalo valdymo sistemos modelį.

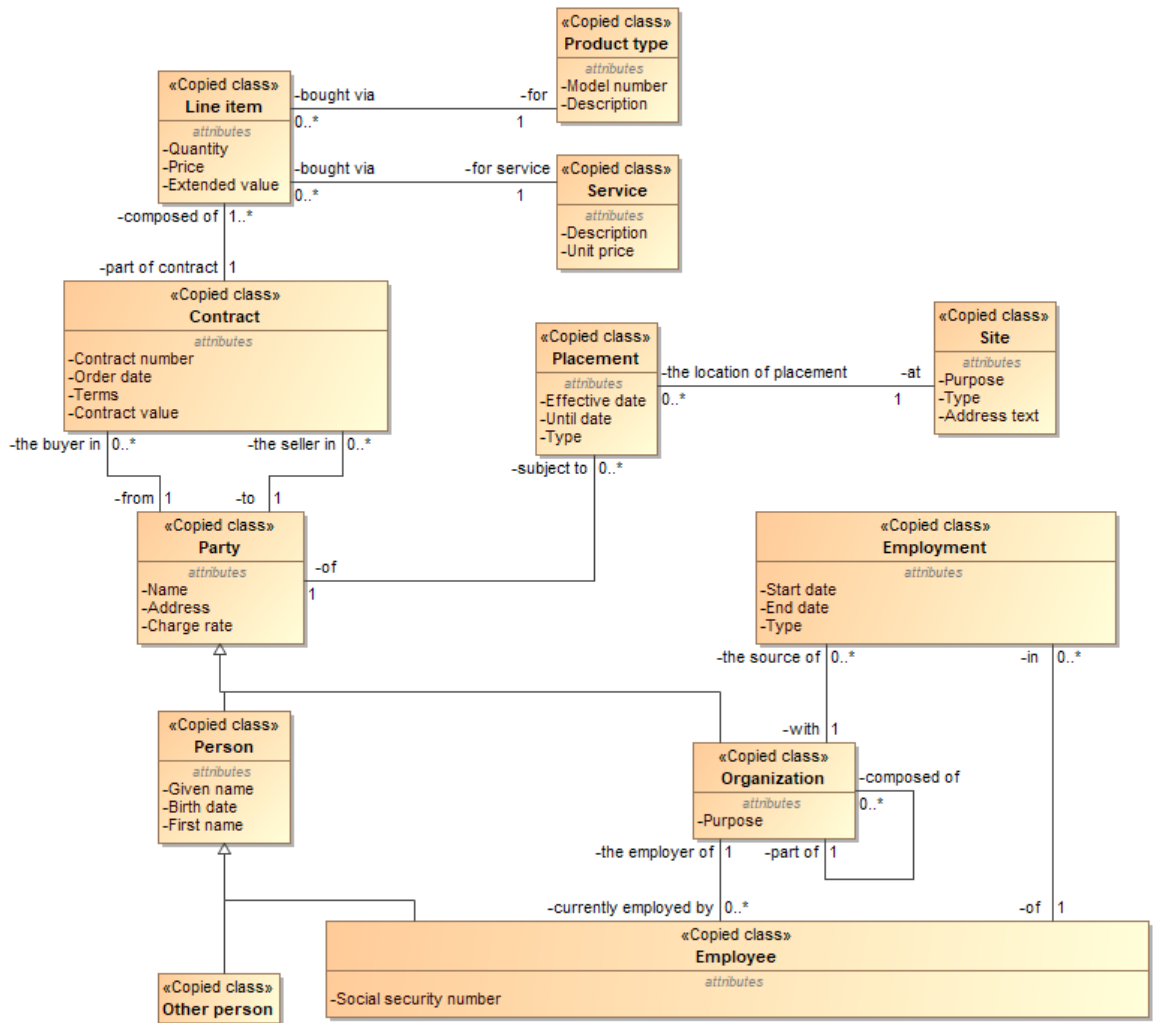


44 pav. „Positions“ šablonas

Visuose paminėtuose šablonuose buvo „Party“ esybė, ši esybė naudojama daugelyje šablonų. Kadangi šablonai naudotas tas pačias esybes per jas galima sujungti keletą šablonų į vieną bendrą modelį.

4.2.3. Testavimo rezultatai

Pasinaudojant sukurtu prototipu ir esančia duomenų modelių šablonų biblioteka, buvo tikimasi gauti tokį pat modelį, kuris buvo pateiktas testavimo dalyje. Testavimo metu buvo gautas modelis pateiktas 45 pav.



45 pav. Testavimo metu gautas modelis

Panaudojant įrankį pavyko gauti tokį pat modelį, kurio buvo tikėtasi panaudojant šablonus. Įprastai sudarant modelį pagal šablonus, kurie jau buvo susisteminti buvo užtrukta 20 minučių. Pasinaudojant sukurtu įrankiu šį modelį galima gauti per 30-60 sekundžių.

5. Šablonais grindžiamas koncepcinio duomenų modelio kūrimo metodo eksperimentinis tyrimas

Šiame skyriuje pateikiamas šablonais grindžiamas koncepcinio duomenų modelio kūrimo metodo eksperimentinio tyrimo planas jo rezultatai bei taikymo rekomendacijos.

5.1. Eksperimento planas

Pagrindinis eksperimento tikslas, nustatyti ar panaudojant įrankį pavyko pasiekti tyrimo tikslą, pagreitinti koncepcinio duomenų modelio kūrimo procesą.

Eksperimento dalyviai buvo parenkami įvairaus lygio projektuotojai. Kadangi eksperimento dalyviai buvo įvairaus lygio projektuotai, todėl jų projektavimo ir numatyto įrankio „MagicDraw“ įgūdžiai labai skyrėsi eksperimente buvo nuspręsta vertinti tik absoliutų procentinį laiko skirtumo duomenų modelio rengimo procese.

Eksperimento vykdymas sudarė 3 dalis:

1. Kiekvienam eksperimento dalyviui bus pristatytas parengtas įrankis ir šablonų biblioteka. Eksperimento dalyviai išbandys įrankio galimybes parengiant paprastą duomenų modelį.
2. Eksperimento dalyviams bus pateikti 2 dalykinės srities aprašai. Pirmajame dalykinės srities apraše bus pristatytas naujas modelis kurį reikėjo sukurti nuo pradžių. Antrajame parengtas modelis bus praplečiamas.
3. Kiekvienas eksperimento dalyvis bus instruktuos kiekvieno dalykinės srities modelio kūrimo metu naudoti laikmatį ir užfiksuoti kiek laiko užtruko suprojektuoti duomenų modelį.
4. Vykdamas eksperimentą dalyvis privalo laikytis tokio plano ir sąlygų:
 - Pagal pirmąjį dalykinės srities aprašą eksperimento dalyvis, pirma rengia modelį įprastiniu būdu.
 - Pabaigus duomenų modelio kūrimą pagal pirmąjį dalykinės srities aprašą eksperimento dalyvis užfiksavo laiką reikalingą sukurti duomenų modelį.
 - Eksperimento dalyvis pagal tą patį pirmąjį dalykinės srities aprašą turi sukurti modelį pasinaudojant šablonų biblioteka bei parengtu įrankiu.
 - Pabaigus kurti modelį naudojantis šablonų biblioteka eksperimento dalyvis užfiksuoja laiką.
 - Pagal antrąjį dalykinės srities aprašą eksperimento dalyvis pasinaudodamas šablonų biblioteka vėl papildoma parengtą modelį reikiamomis dalykinės srities esybėmis bei užfiksuoja laiką reikalingą atlikti korekcijas.
 - Pabaigus korekcijas pasinaudojant šablonų biblioteka eksperimento dalyvis tas pačias korekcijas turi atlikti ir pirminiame modelyje kuriame nebuvo naudojama šablonų biblioteka.

- Atlikus korekcijas eksperimento dalyvis vėl užfiksuoja laiką.

Apibendrinant eksperimento planas yra sudarytas taip, kad dalyviai turėtų galimybę pamatyti šablonų privalumai kai modelis yra žinomas ir nebereikia į jį gilintis, bei modelio kūrimą naudojantis tik šablonais, įvertinant trukmę kurios reikia surasti tinkamą šabloną.

5.2. Eksperimento rezultatai

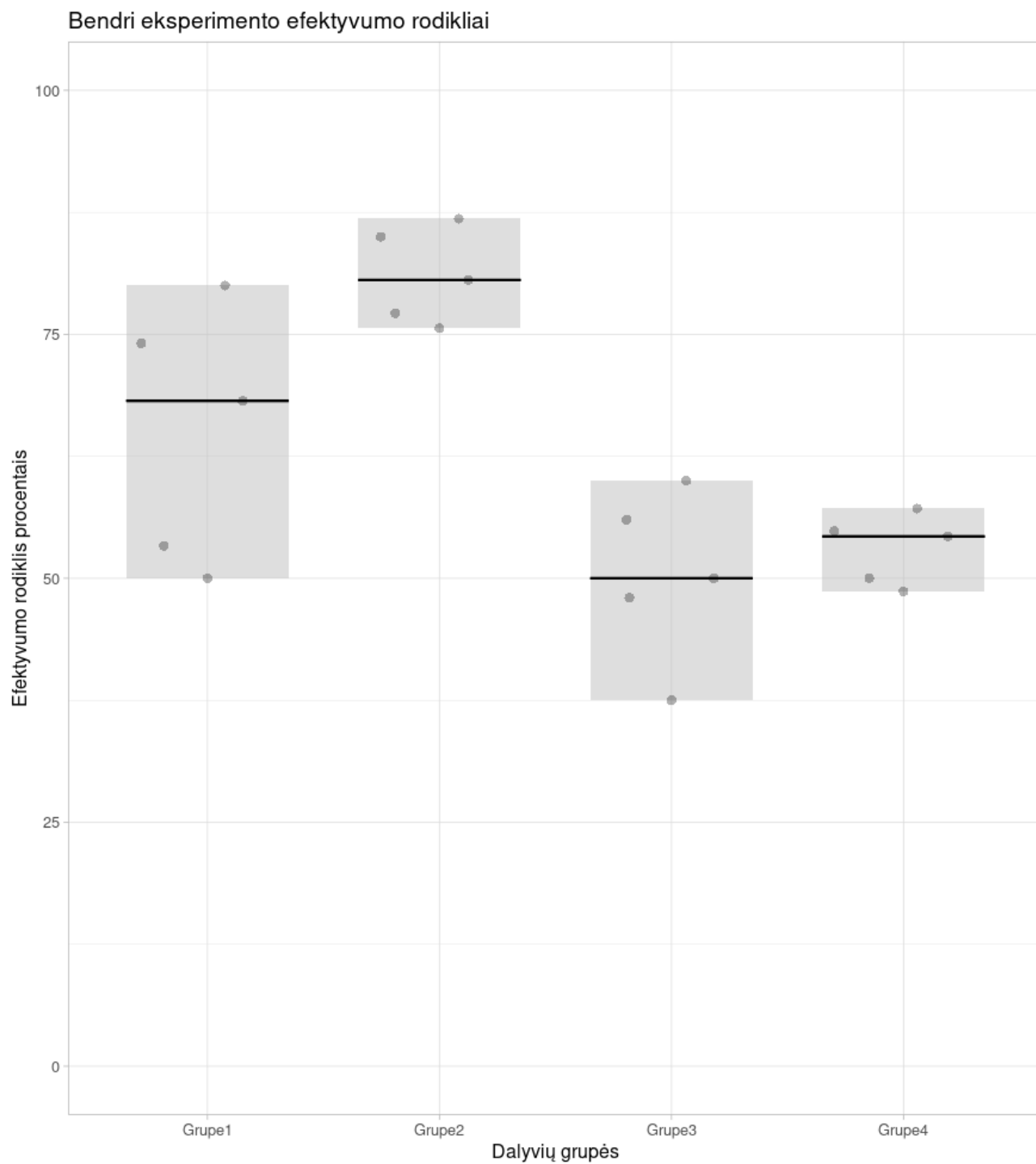
Eksperimento dalyviai buvo suskaidyti į 4 grupes.

1. Pirma grupė – asmenys dirbantys su duomenų bazėmis ir susipažinę su MagicDraw įrankiu;
2. Antra grupė – asmenys dirbantys su duomenų bazėmis nesusipažinę su MagicDraw įrankiu;
3. Trečia grupė – asmenys dirbantys su MagicDraw įrankiu, tačiau turintys mažai patirties duomenų modelių kūrime;
4. Ketvirta grupė – asmenys turinys mažai patirties su MagicDraw įrankiu ir turintys mažai patirties su duomenų modelių kūrimu.

Eksperimento eksperimente dalyvavo dalyviai iš šių organizacijų:

1. Kauno technologijos universiteto;
2. Kauno grūdai, AB;
3. Registrų centras, VĮ;
4. Plum Tree systems;
5. Švyturys – Utenos alus, UAB;

Eksperimento dalyviai pagal dalykinės srities aprašus kūrė 2 duomenų modelius. Eksperimento rezultatai pateikti žemiau.

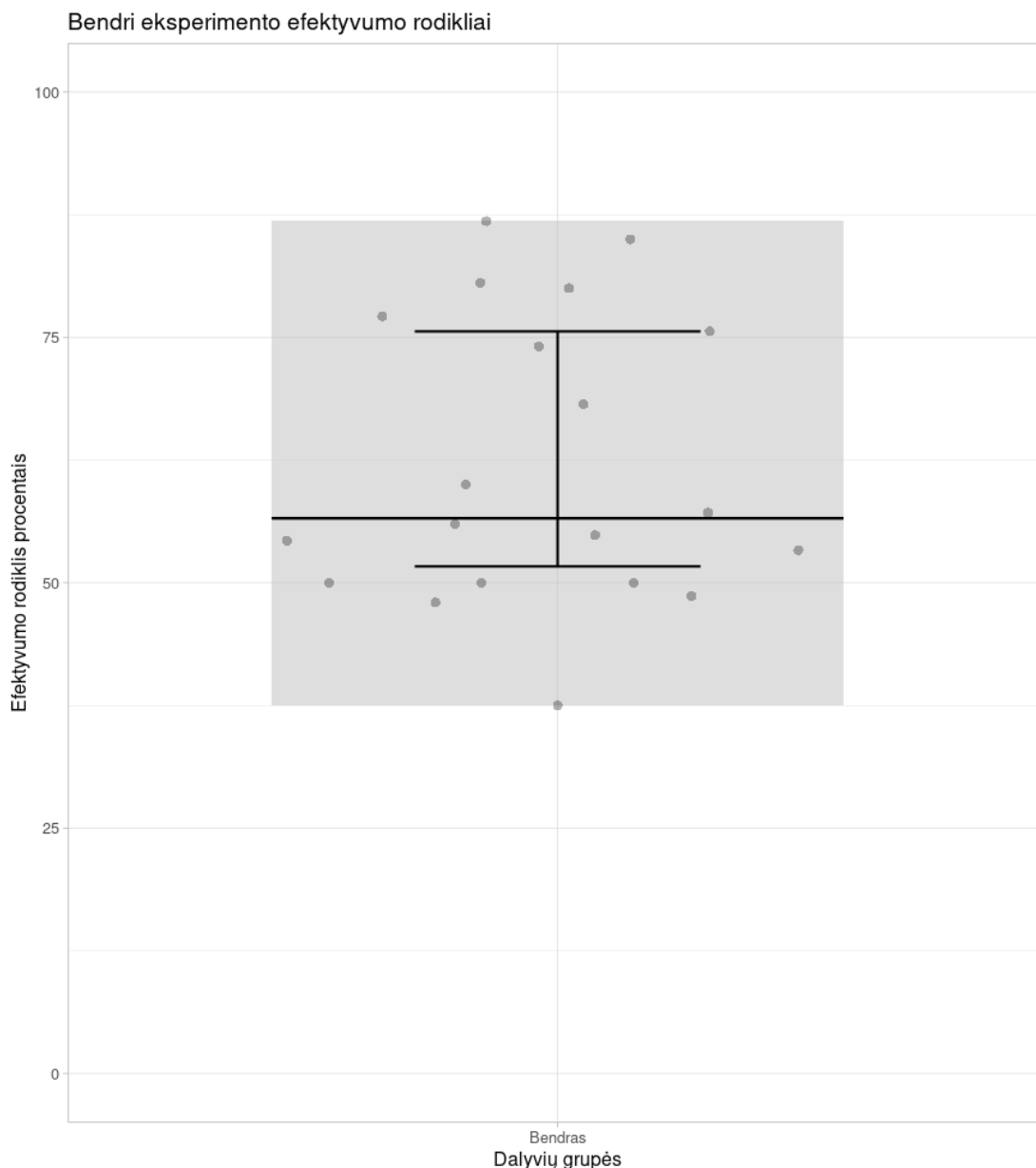


46 pav. Bendri eksperimentinio modelio kūrimo efektyvumo rodikliai dalyvių grupėse

Lentelė 17 Detalizuoti bendri eksperimento modelio kūrimo efektyvumo rodikliai dalyvių grupėse

Grupė	Mediana	95CI mediana
Grupe1	68,18%	50% – 80%
Grupe2	80,56%	75,61% – 86,84%
Grupe3	50%	37,5% – 60%
Grupe4	54,29%	48,65% – 57,14%

Pagal bendrus eksperimento rezultatus pateiktus 46 pav. matyti, kad šis metodas ir įrankis labiausiai paveikė 2 grupės dalyvius. Šios grupės dalyviams esybės esančios šablonuose jau buvo žinomos, todėl modeliuoti naudojant įrankį buvo lengviau nei įprastiniu būdu. Mažiausiai įrankis paveikė 3 grupės dalyvius. Kadangi grupėje dalyviai buvo įgudę MagicDraw įrankio naudotojai laiko skirtumai tarp įprastinio modeliavimo metodo ir įrankio nebuvo labai dideli. Tam tikrai atvejais šabloninių esybių paieška net pailgino laiką reikalingą sukurti modelį nei naudojantis įprastiniu būdu. Tačiau geriau susipažinus su šablonų biblioteka šie laikai pasikeistų.

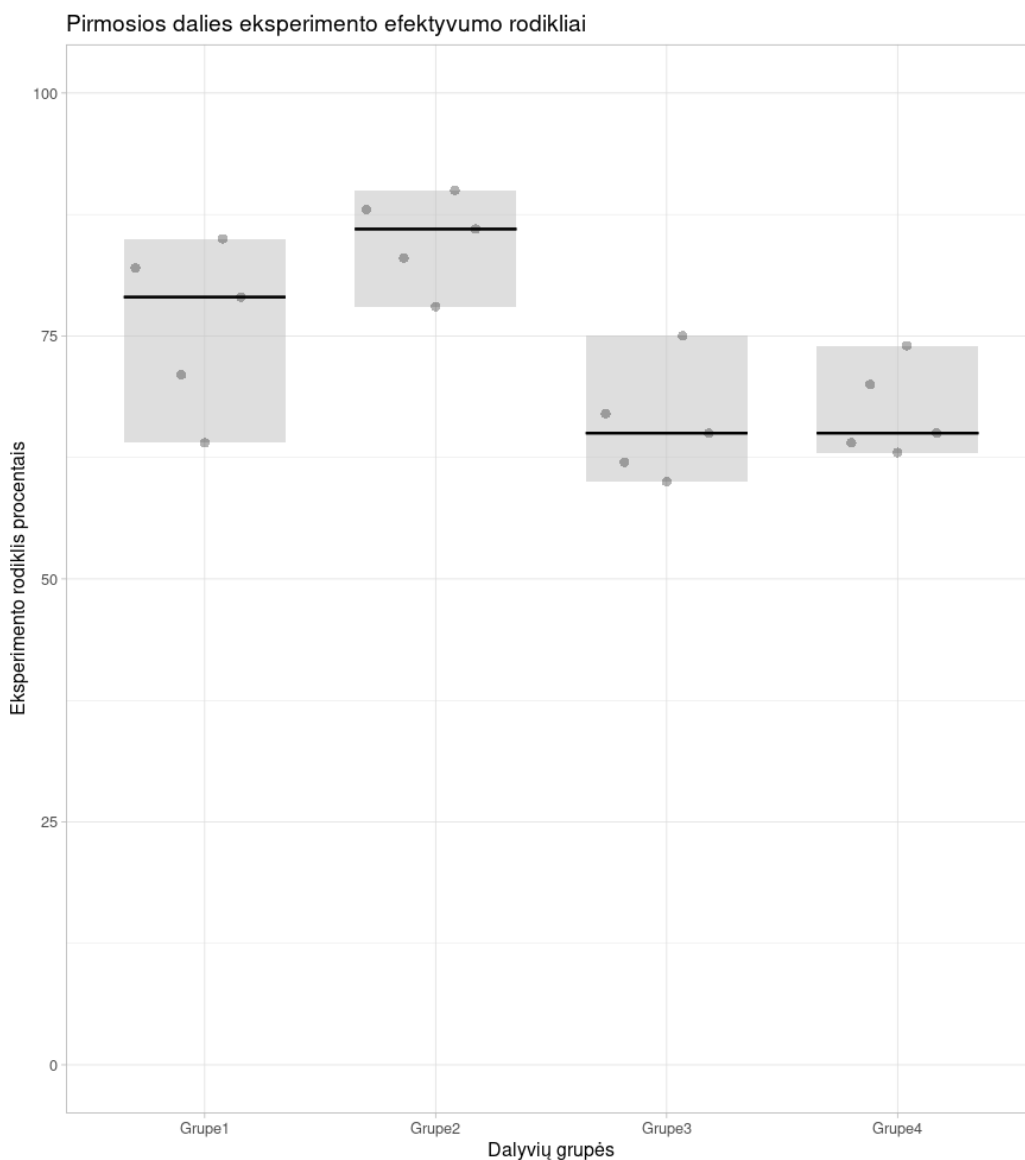


47 pav. Bendri visų eksperimento dalyvių rezultatai

Lentelė 18 Bendras detalizuotas visų eksperimento dalyvių rezultatas

Grupė	Mediana	95CI mediana
Bendras	56,57%	51,67% – 75,61%

Pagal bendrus pateiktus rezultatus 47 pav. matoma, kad įrankis pagreitina modelio kūrimą. Naudojantis 0.95 statistinio patikimumo rodikliu medianai galima matyti, kad įrankis mažiausias pastebimas pagreitėjimas siekia 51,67%, o didžiausias 75,61%. Didėjant dalyvių skaičius apatinė ir viršutinė riba gali kisti. Tačiau iš bendrų rezultatų kuriuos pavyko gauti dalyviams pirmą kartą naudojantis įrankiu ir biblioteka galima teigti, kad naujas duomenų modelio kūrimo metodas ir įrankis pasiekė tikslą ir pagreitino duomenų modelio kūrimą.

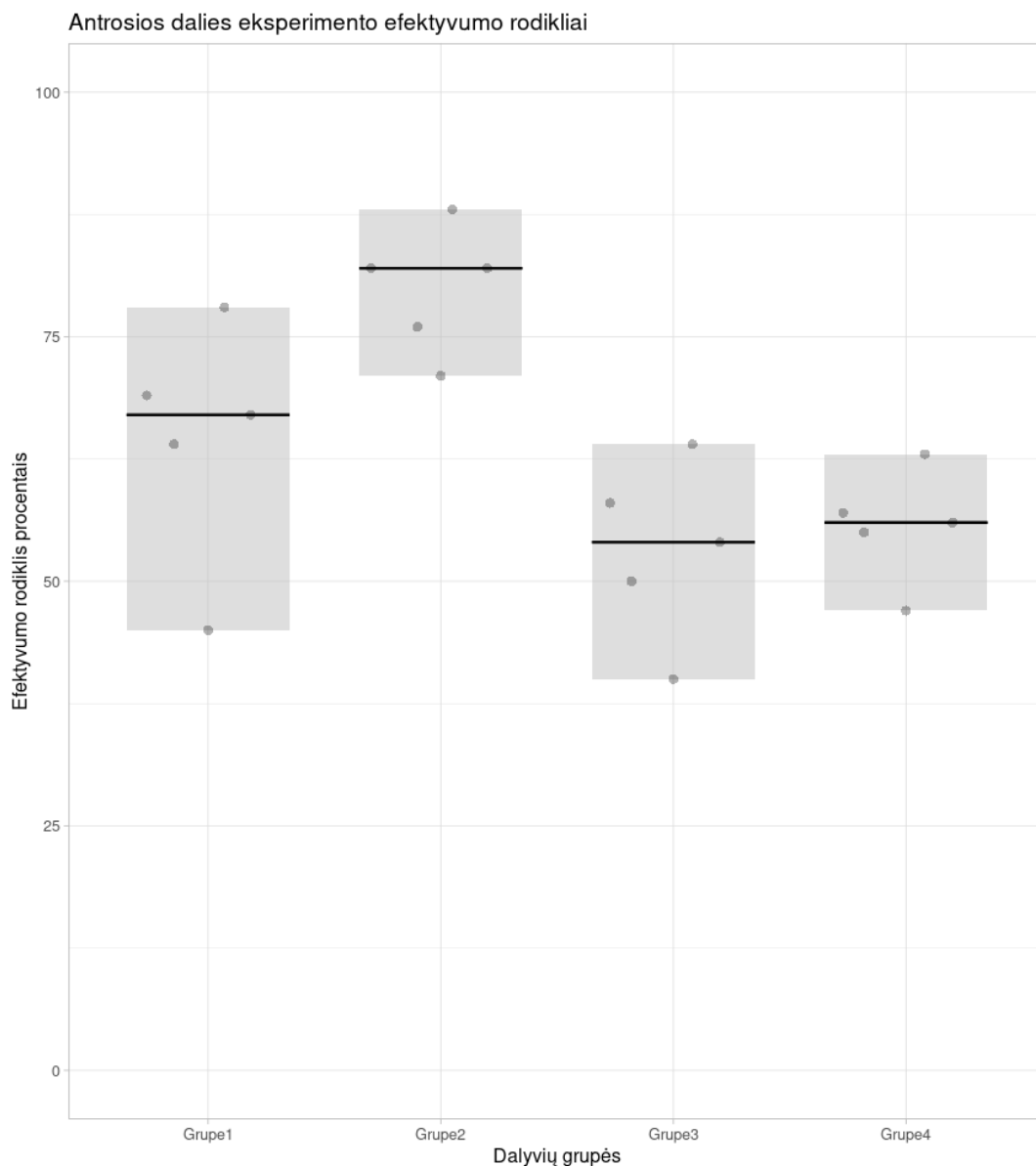


48 pav. Pirmosios eksperimento dalies efektyvumo rodikliai dalyvių grupėse

Lentelė 19 Pirmosios eksperimento dalies detalizuotas rezultatas dalyvių grupėse

Grupė	Mediana	95CI mediana
Grupe1	79%	64% – 85%
Grupe2	86%	78% – 90%
Grupe3	65%	60% – 75%
Grupe4	65%	63% – 74%

Aukščiau pateiktame 48 pav. pateikti pirmosios dalies rezultatai. Šioje dalyje matyti, kad dalyvių laikai pagreitėjo 65% ir daugiau procentais. Šioje dalyje duomenų modelį dalyviai pirmiau kūrė naudodami įprastinį modeliavimo procesą, o po to tą patį modelį kūrė naudojanti šablonų biblioteka. Kadangi modelis buvo nesudėtingas todėl nebuvo sunku atrasti atitikmenis šablonų bibliotekoje.



49 pav. Antrosios eksperimento dalies efektyvumo rodikliai dalyvių grupėse

Lentelė 20 Antrosios eksperimento dalies detalizuotas rezultatas dalyvių grupėse

Grupė	Mediana	95CI mediana
Grupe1	67%	45% – 78%
Grupe2	82%	71% – 88%
Grupe3	54%	40% – 64%
Grupe4	56%	47% – 63%

Aukščiau pateiktame 49 pav. pateikti antrosios eksperimento modelio dalies rezultatai. Antrasis modelis buvo kuriamas pirma naudojantis šablonų biblioteka, šis modelis buvo sudėtingesnis nei pirmasis, ne visos esybės nurodytos dalykinės srities apraše turėjo tiesioginius atitikmenis šablonų bibliotekoje. Todėl sudaryti šį modelį naudojantis šablonais dalyviams, kurie nebuvo susipažinę su šablonų biblioteka buvo sudėtinga. Pagal pateiktus rezultatus matyti mažesnis pagreitėjimas nei sudarant pirmąjį modelį. Šis pagreitėjimas beveik nepasikeitė 2 grupėje, kadangi šios grupės dalyviai turėjo daugiau patirties ir buvo susidūrę su šablonais jiems pavyko greičiau surasti atitikmenis šablonų bibliotekoje ir sudaryti modelį. Prasčiausi rezultatai buvo 3 grupėje, šios grupės dalyviams buvo sudėtinga atrasti šablonus, todėl kuriant modelį įprastiniu būdu jį sudaryti pavyko greičiau. Tam įtakos turėjo ir tai kad jie jau buvo gavę šį modelį su šablonų biblioteka todėl jį reikėjo tiesiog perbraižyti iš jau turimo modelio ir šiek tiek pakeisti esybių pavadinimus.

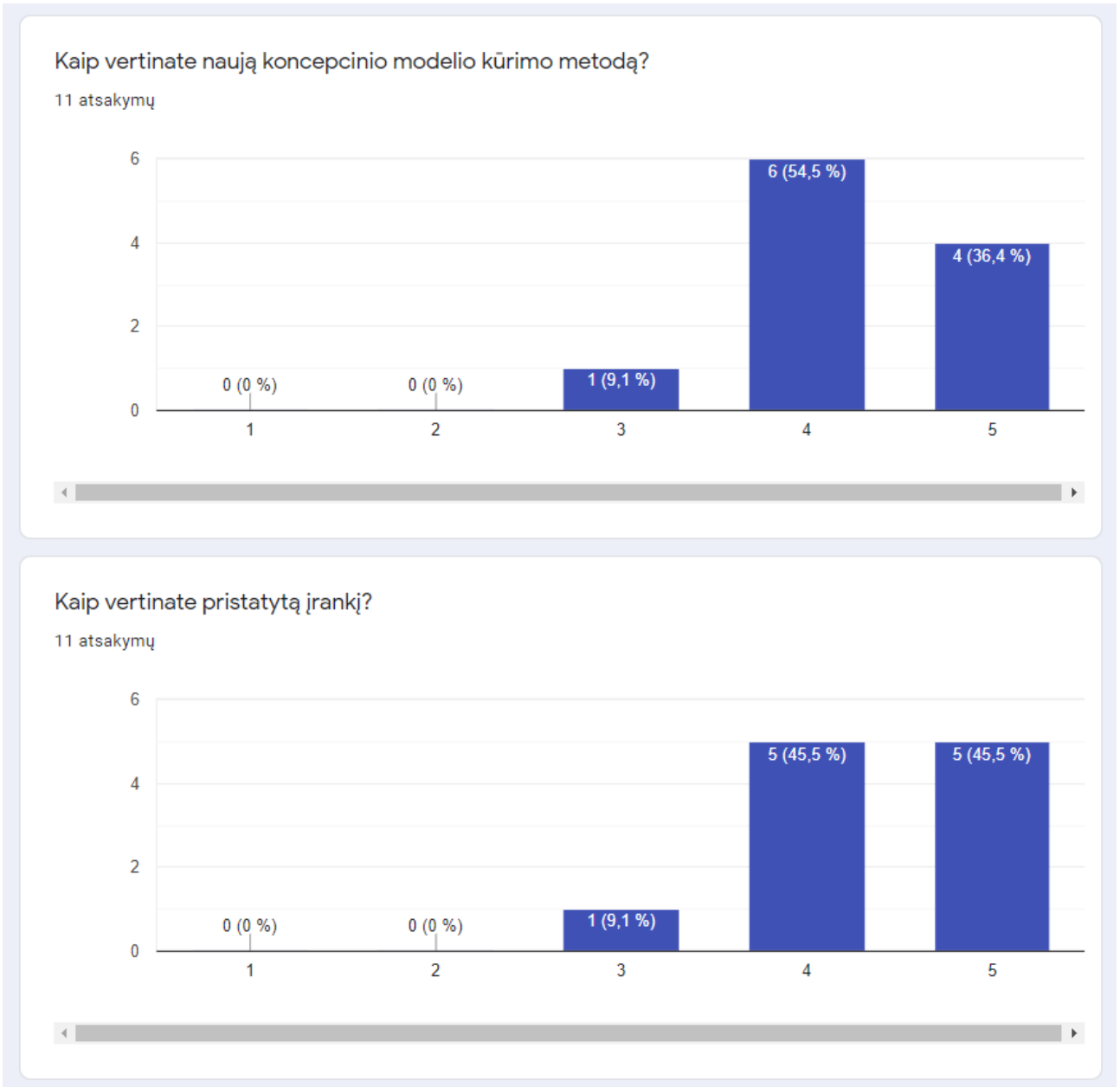
Apibendrinant eksperimento rezultatus matomas duomenų modelio kūrimo pagreitėjimas. Eksperimento dalyvių skaičius nebuvo didelis todėl didėjant dalyvių skaičiui gauti rezultatai gali skirtis, tačiau pagreitėjimas turi išlikti. Eksperimento metu pastebėta, kad norint pilnai išnaudoti parengtą įrankį ir biblioteką duomenų modelio kūrime, reikia susipažinti su šablonais esančiais bibliotekoje. Žinant šablonus projektuotojai jau reikalavimų surinkimo gali žinoti šablonus kurių jiems prireiks kuriant dalykinės srities modelį. Tai pasiteisino ir kuriant pirmosios dalies modelį, kadangi pirmojoje dalyje visos kuriamos esybės buvo šablonų bibliotekoje su atitinkamais angliškais pavadinimais, eksperimento dalyviams lengviau pavyko surasti šablonus ir esybes reikalingas sukurti dalykinės srities modelį.

5.3. Sprendimo veikimo ir savybių analizė, kokybės kriterijų įvertinimas

Analizės dalyje pasirinkti duomenų modelio kūrimo metodai buvo lyginami pagal 4 kriterijus: kaina, pritaikomumas, lankstumas ir sudėtingumas.

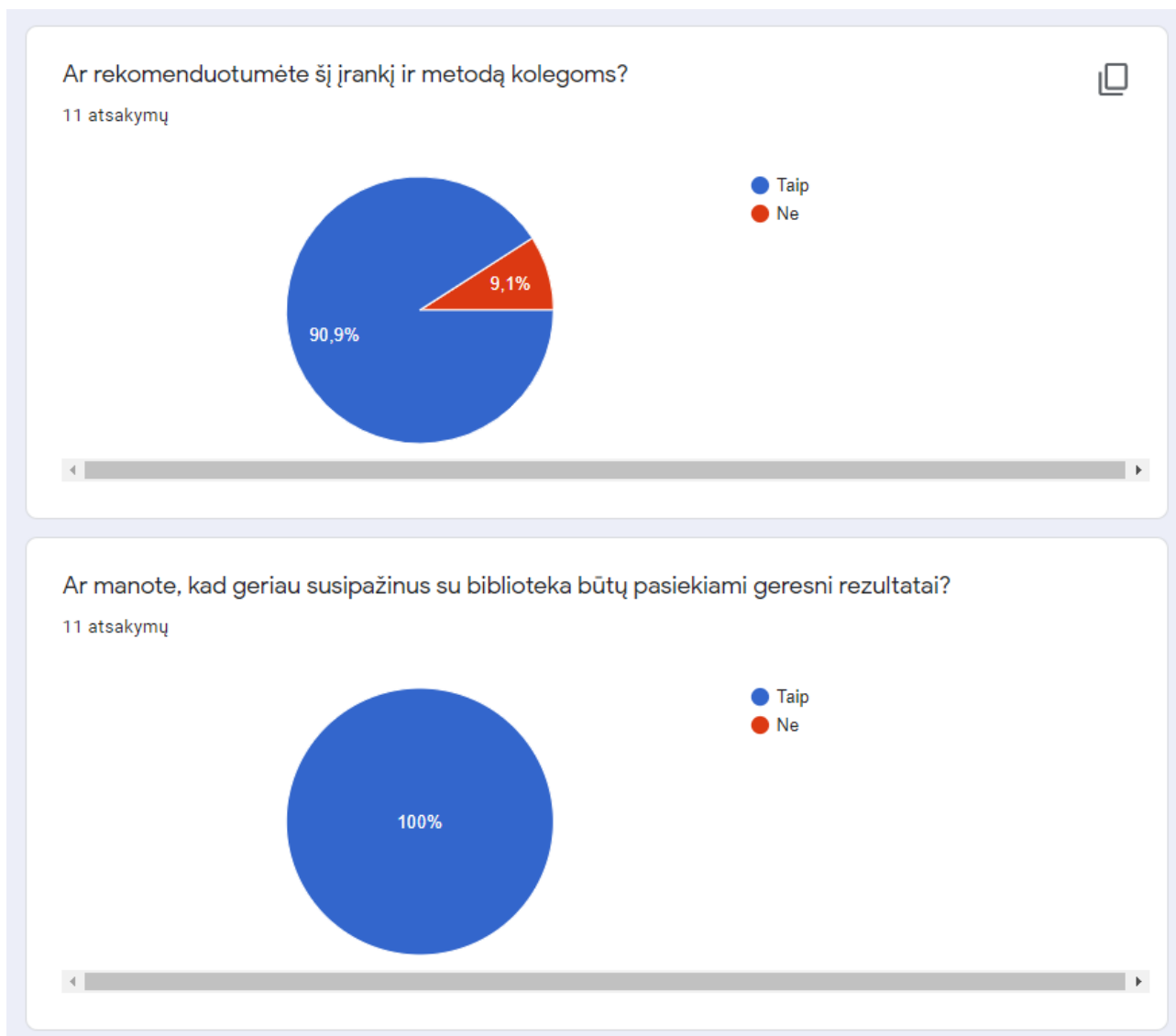
Realizuotas sprendimas atitinka analizėje nurodytus šablonais pagrįsto duomenų modelio kūrimo kriterijus. Realizacijos kaina buvo žema, realizuotas sprendimas nesunkiai įdiegiamas ir pritaikomas duomenų modelio kūrimo procese. Duomenų modeliai gauti panaudojant šablonus yra lankstūs ir lengvai plečiami. Sprendimo sudėtingumas tiesiogiai susijęs su naudotojo turimomis žiniomis apie atitinkamų dalykinių sričių šablonus ir parengtą duomenų modelių šablonų biblioteką. Paiešką bibliotekoje yra sudėtingesnė asmenims, kurie turi mažiau patirties.

Kokybiniam įvertinimui buvo parengta apklausa, kurioje 91% naują koncepcinio modelio kūrimo procesą bei įrankį įvertino **Gerai** ir **Labai gerai** likusioji 9% dalis metodą bei įrankį įvertino **Vidutiniškai**. Detalesni rezultatai pateikti 50 pav.



50 pav. Metodo ir įrankio apklausos įvertinimai

100% apklausos dalyvių sutinka, kad geriau susipažinus su duomenų modelių šablonų biblioteka būtų pasiekiami geresni rezultatai, bendruose atsiliepimuose komentarai dėl poreikio gerai žinoti duomenų modelių šablonų biblioteką taip pat buvo paminėti. Rezultatai pateikti 51 pav.



51 pav. Apsimokymo laiko ir rekomendacijų apklausos įvertinimai

Tik 9,1% iš apklausos dalyvių nerekomendotų šio įrankio kolegoms.

Pagal bendrus apklausos rezultatus galima spręsti, kad naujas modelio kūrimo metodas bei įrankis yra vertinami teigiamai. Dabartinis įvardintas trūkumas – dalyviai nėra susipažinę su duomenų modelių šablonų biblioteka, todėl tinkamo šablono paieška užtrunka ilgiau.

5.4. Sprendimo taikymo rekomendacijos

Pagal gautus rezultatus pavyko nustatyti, kad sprendimas pagreitina duomenų modelio kūrimą. Tačiau šablonai turi tik pačius svarbiausius elementus. Norint taikyti šį sprendimą produkcinės duomenų bazės kūrimo procese reikėtų atsižvelgti į:

1. Sistemos paskirties ir organizacijos veiklos plėtimo galimybes ir pagal tai nuspręsti ar likti prie bendrinių šabloninių esybių pavadinimų, ar juos konkretizuoti;
2. Kuriant naują koncepcinį modelį pasinaudoti įrankiu greitai sudaryti koncepcinio modelio branduolį, ir taip greitai pradėti modelio kūrimą;

3. Plečiant esamą modelį patikrinti šablonų ir esamo modelio esybes dėl galimų sinoniminių ar bendriniais pavadinimais pavadintų esybių. Tam, kad modelyje nebūtų įtrauktų semantinė prasme identiškų esybių;
4. Pasirenkant šabloną įvertinti ar tikrai reikia visų jo esybių, kadangi šablonai dažniausiai siūlo lankstų sprendimą tinkantį daugeliui atvejų paprastoje sistemoje jis gali būti nereikalingas, todėl pagal poreikį įsikelti tik būtinas esybes esamiems ir artimos ateities poreikiams. Pasikeitus poreikiams pasinaudojant įrankiu bus lengva papildyti struktūrą.

Pateiktame sprendime pridėta duomenų modelių šablonų biblioteka kurią kiekvienas projektuotojas gali prisitaikyti pagal save, tačiau pakeitimus bibliotekoje rekomenduojama atlikti ekspertams. Naujus šablonus sukurti yra sudėtinga, kadangi reikia atrasti tinkamą esybių ir šablonų abstrakcijos lygį. Pradedantiesiems projektuotojams siūloma naudotis esamais šablonais bibliotekoje.

5.5. Ateities perspektyvos

Sukūrus sprendimo prototipą ir atlikus eksperimentą buvo gauti teigiami rezultatai, todėl ateityje numatoma šį sprendimą tobulinti:

1. Šablonų bibliotekoje pakeisti rolių vardus, nustatyti skaitymo kryptį.
2. Atlikti šablonų esybių atributų analizę ir pagal gautus rezultatus praplėsti esybių atributų skaičių.
3. Sukurti detalesnę ekspertams skirtą šablonų biblioteką.
4. Įrankyje realizuoti bibliotekos pasirinkimo funkciją.

Šiuo metu numatomi aukščiau išvardinti 4 darbai, tolimesnės plėtimo perspektyvos gali būti nustatomos iš naudotojų atsiliepimų ir funkcijų poreikių nustatymo.

Išvados

1. Išanalizavus duomenų bazės kūrimo procesą nustatyta, kad ilgiausiai užtrunka sukurti koncepcinį duomenų bazės modelį. Vienas iš būdų kaip pagreitinti koncepcinio duomenų modelio kūrimą yra panaudoti duomenų modelių šablonus. Pasinaudojant bendriniais nuo dalykinės srities nepriklausomais šablonais galima padengti apie 50% bet kurios įmonės duomenų modelio esybių, o pasinaudojus specifinės pramonės srities šablonais galima padengti iki 75% duomenų modelio.
2. Siekiamo sprendimo analogų analizės metu nustatyta, kad šiuo metu nėra realizuoto įrankio, kuris leistų iš duomenų modelių šablonų bibliotekos pasirinkti dalykinei sričiai tinkamą modelį ir jį įterpti į kuriamos sistemos duomenų modelį.
3. Tipinio koncepcinio modelio kūrimo metodo analizės metu buvo įvardinta pasikartojančių elementų (klasių, atributų, ryšių) kūrimo problema, kurią pavyko išspręsti praplečiant koncepcinio modelio procesą panaudojant duomenų modelių šablonų biblioteką bei sukurtą MagicDraw plėtinį leidžiantį elementus įkelti į kuriamą modelį. Šablonuose esantys elementai (klasės, atributai, ryšiai) pašalina didelę dalį pasikartojančių elementų, kuriuos įprastai reikia kurti kiekvieną kartą projektuojant naują duomenų bazę.
4. Palyginus eksperimento metu gautas modelio kūrimo trukmes modelį kuriant įprastiniu modeliavimo metodu ir metodu, praplėstu šablonų taikymu, nustatyta, kad šablonų taikymas leidžia sumažinti dažnai skirtinguose modeliuose pasikartojančių elementų (klasių, atributų, ryšių) kūrimą ir dėl to duomenų modelio kūrimo laikas sutrumpėja iki 86%.
5. Remiantis apklausos ir eksperimento rezultatais galima teigti, kad kuriant duomenų modelius šablonų pagrindu gaunami lankstesni modeliai. Atsiradus naujiems reikalavimams tokie modeliai gali būti plečiami įtraukiant kitus susijusius reikalingus šablonus ir esybes.
6. Pagal apklausos rezultatus nustatyta, kad pirmą kartą naudojantis įrankiu nežinant šablonų jų paieška užtrunka ilgiau. Geriau susipažinus su šablonų biblioteka, sutrumpėja šablonų ir esybių paieška bibliotekoje, kas leis greičiau įtraukti visas reikiamas esybes ir šablonus į kuriamą modelį taip išvengiant pasikartojančių elementų kūrimo įprastu būdu.
7. Atlikus eksperimentą ir apibendrinus eksperimentinių duomenų modelių kūrimo laikų rezultatus galima teigti, kad darbo tikslas, sudaryti sąlygas leidžiančias pagreitinti koncepcinio modelio kūrimą buvo pasiektas. Modelio kūrimo laikas skirtingų kompetencijų dalyvių grupėse, kurie pirmą kartą naudojami įrankiu ir šablonų biblioteka sumažėjo nuo 37,5% iki 86,84%. Šablonus efektyviausiai sekėsi taikyti dalyviams, kurie buvo susipažinę su duomenų modelių šablonais. Šioje eksperimento dalyvių imtyje modelio kūrimo laiką vidutiniškai pavyko sumažinti 56,57%.

Literatūros sąrašas

1. G. Simsion, *Data Modeling Theory and Practice*, New York: Technics Publications, LLC, 2007.
2. P. P.-S. Chen, „English sentence structure and entity-relationship diagram,“ *Information Sciences*, t. 29, nr. 2-3, pp. 127-149, 1983.
3. H. Taedong, P. Sandeep ir S. C. Veda, „Generating large-scale repositories of reusable artifacts for conceptual design of information systems,“ *Decision Support Systems*, t. 45, nr. 4, pp. 665-680, 2008.
4. O. Thonggoom, I.-Y. Song ir Y. An, „Semi-automatic Conceptual Data Modelling Using Entity and Relationship Instance Repositories,“ įtraukta *Conceptual Modelling - ER 2011*, Berlin, 2011.
5. O. Thonggoom, I.-Y. Song ir Y. An, „EIPW: A Knowledge-Based Database Modeling Tool,“ įtraukta *Advanced Information Systems Engineering Workshops. CAiSE 2011.*, Berlin.
6. B. Thalheim, *Entity-Relationship Modeling: Foundations of Database Technology*, Springer, 2000.
7. I.-Y. Song, Z. Yongjun, C. Hyithaek ir O. Thonggoom, „Methodologies for Semi-automated Conceptual Data Modelling from Requirements,“ įtraukta *International Conference on Conceptual Modeling*, 2015.
8. I.-Y. Song, J. Trujillo, K. Yano ir S. Lujan-Mora, „A taxonomic class modeling methodology for Object-Oriented analysis,“ įtraukta *Information Modelling methods and Methodologies. Advanced Topics in Databases Series.*, Hershey, Idea Group Publishing, 2004, pp. 216-240.
9. A. Solomon ir M. Vidya, „Data Modeling Patterns: A Method and Evaluation,“ įtraukta *Proceedings of the Fifteenth Americas Conference on Information Systems*, San Francisco, California, 2009.
10. G. Simsion ir G. Witt, *Data Modelling Essentials*, San Francisco: Morgan Kaufmann, 2004.
11. L. Silverston, *The Data Model Resource Book, Vol. 1: A library of Universal Data Models for All Enterprises*, Wiley, 2001.
12. G. Shanks, „Conceptual data modeling process: a study of novice and expert,“ *Proceedings of the 1st International Conference on Object-Role Modeling*, 1994.
13. K. Namgyu, L. Sangwon ir M. Songchun, „Formalized Entity Extraction Methodology for Changeable Business Requirements,“ *Journal of information science and engineering*, t. 24, pp. 649-671, 2008.
14. L. Mich, M. Franch ir P. Inverardi, „Market research for requirements analysis using linguistic tools,“ *Requirements Engineering*, t. 9, nr. 1, pp. 40-56, 2004.
15. D. C. Hay, *Data Model Patterns: Conventions of Thought*, New York: Dorset House Publishing, 1996.
16. D. C. Hay, *Enterprise Model Patterns: Describing the World (UML Version)*, Technics Publications, 2011.

17. D. C. Hay, *Data Model Patterns: A Metadata Map* (The Morgan Kaufmann Series in Data Management Systems), San Francisco: Morgan Kaufmann, 2006.
18. J. Giles, *The Nimble Elephant: Agile Delivery of Data Models using a Pattern-based Approach*, Technics Publications, 2012.
19. M. Fowler, *Analysis Patterns: Reusable Object Models*, Menlo Park: Addison Wesley, 1997.
20. D. W. Embley, „Toward semantic understanding: an approach based on information extraction,“ *Proceedings of the 15th Australasian database conference*, t. 27, pp. 3-12, 2004.
21. P. Coad, D. North ir M. Mayfield, *Object models: strategies, patterns, applications*, New Jersey, USA: Yourdan Press, 1995.
22. P. P.-S. Chen, „The entity-relationship model toward a unified view of data,“ *ACM Transactions on Database Systems (TODS)*, t. 1, nr. 1, pp. 9-36, 1976.
23. M. Blaha, *Patterns of Data Modeling*, 2010.
24. D. Batra, „Cognitive complexity in data modeling: causes and recommendations,“ *Requirements Engineering*, t. 12, nr. 4, pp. 231-244, 2007.
25. C. Alexander, *The Timeless Way of Building*, New York: Oxford University Press, 1979.
26. „Requirement engineering: the state of the practice,“ *IEEE Software*, t. 20, nr. 6, pp. 40-45, 2003.

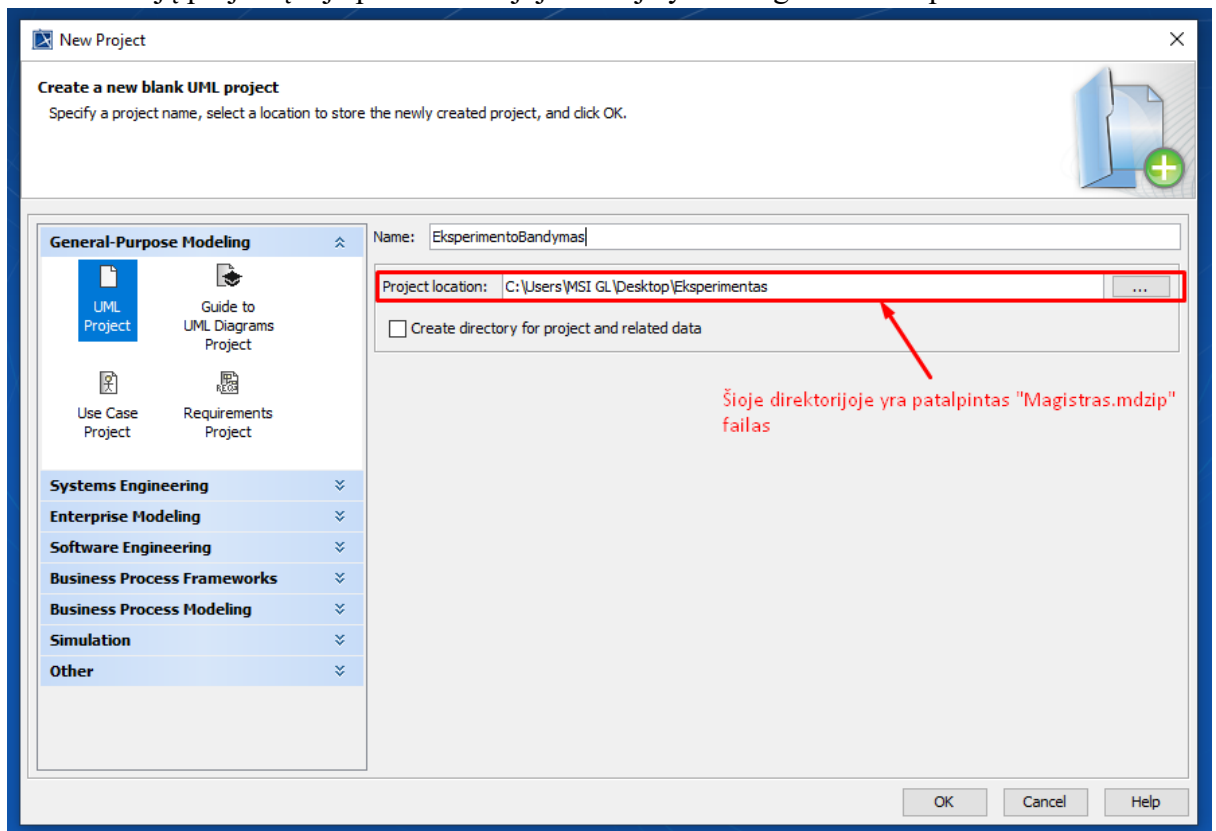
Priedai

1 priedas. Eksperimento vykdymo aprašas Plugin įrašymas.

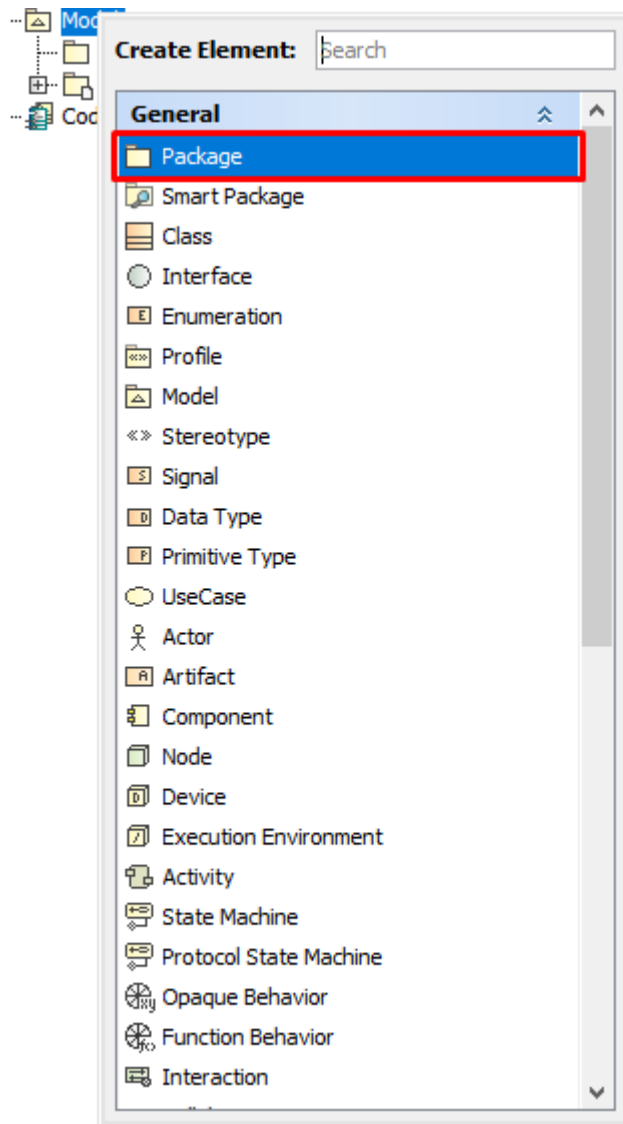
1. Gautą Eksperimentas.7z failą išskleidžiate parsisiuntimo kataloge.
2. Atsidarius MagicDraw instaliacijos katalogą, jame surasti “plugins” direktoriją ir į ją įkelti “data.model.patterns” direktoriją.
3. Iš “Eksperimentas” katalogo nukopijuoti “Magistras.mdzip” failą ir jį įkelti į savo “darbinę” direktoriją. T.y. direktoriją kurioje sukursite naują MagicDraw projektą.

Projekto sukūrimas

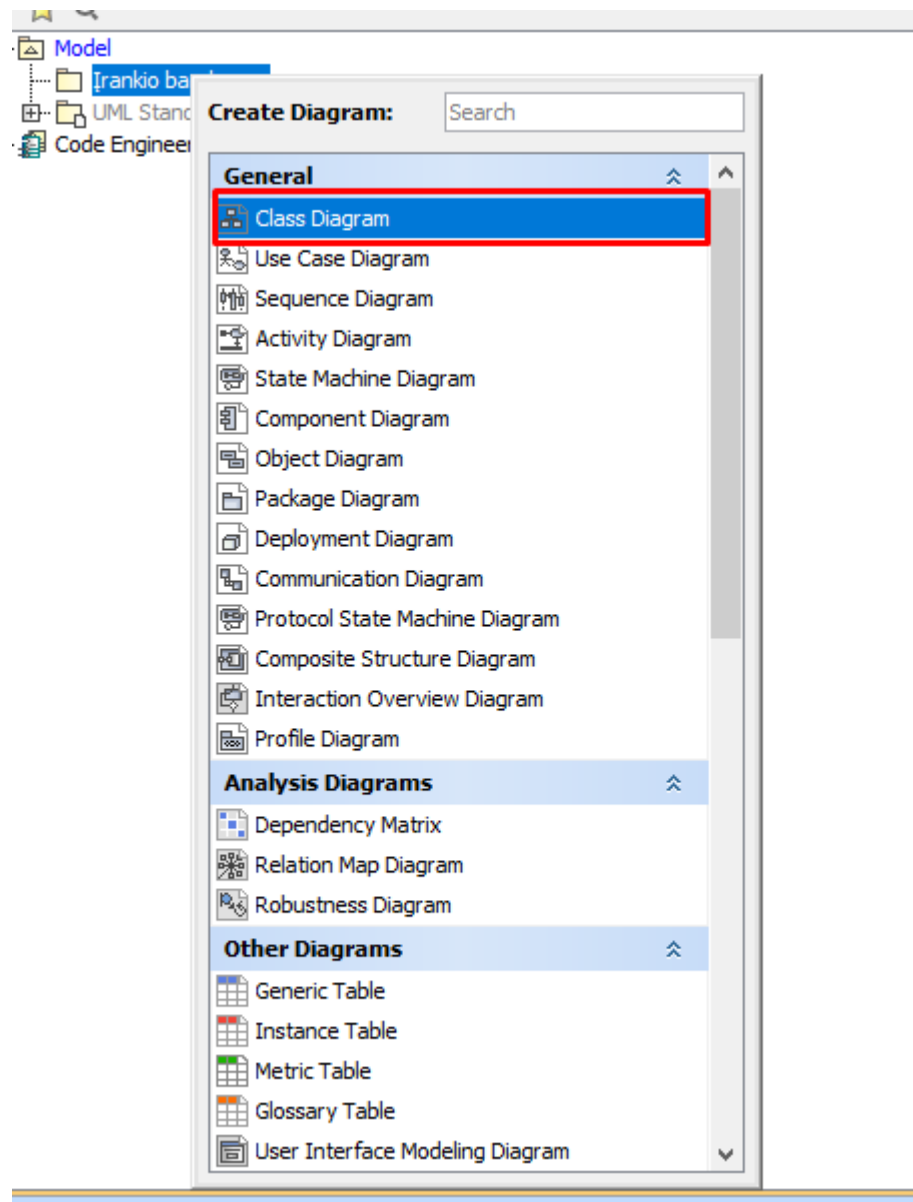
1. Sukurti naują projektą toje pat direktorijoje kurioje yra “Magistras.mdzip”



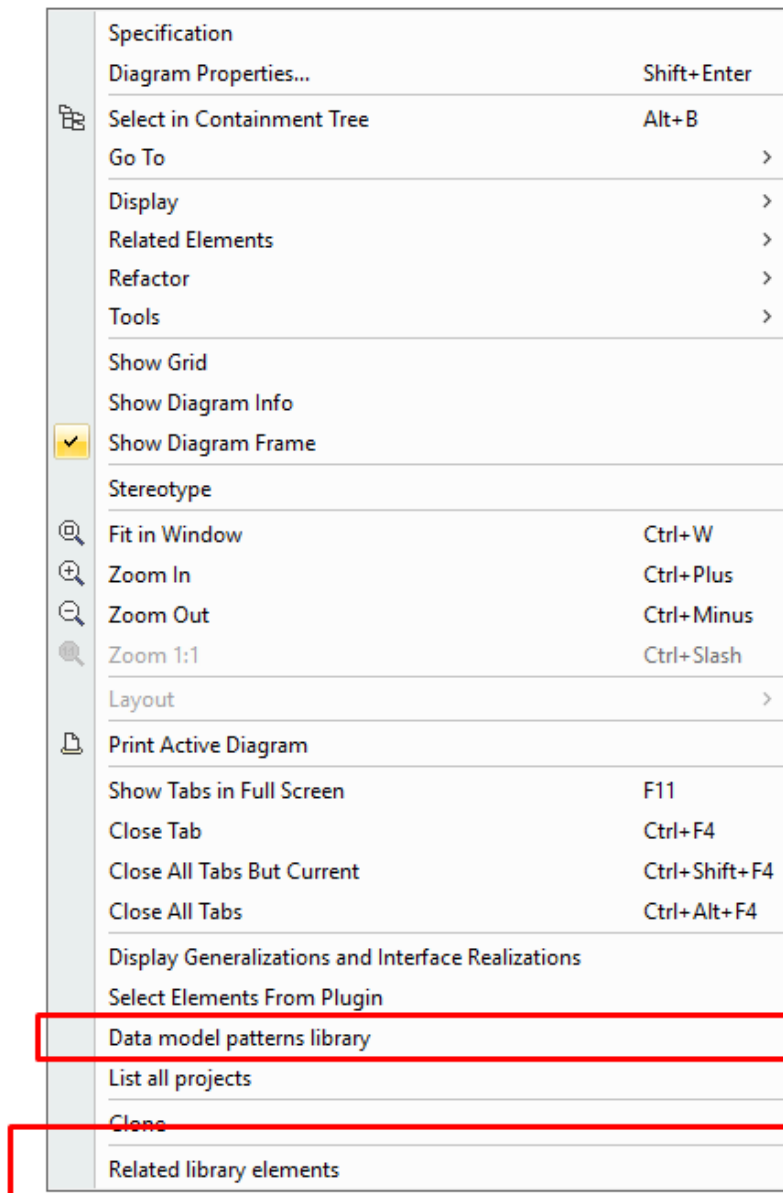
2. Sukurtame projekte sukurti naują “Package” tipo elementą.



3. Sukurtame “Package” elemente sukurti “Class Diagram”.

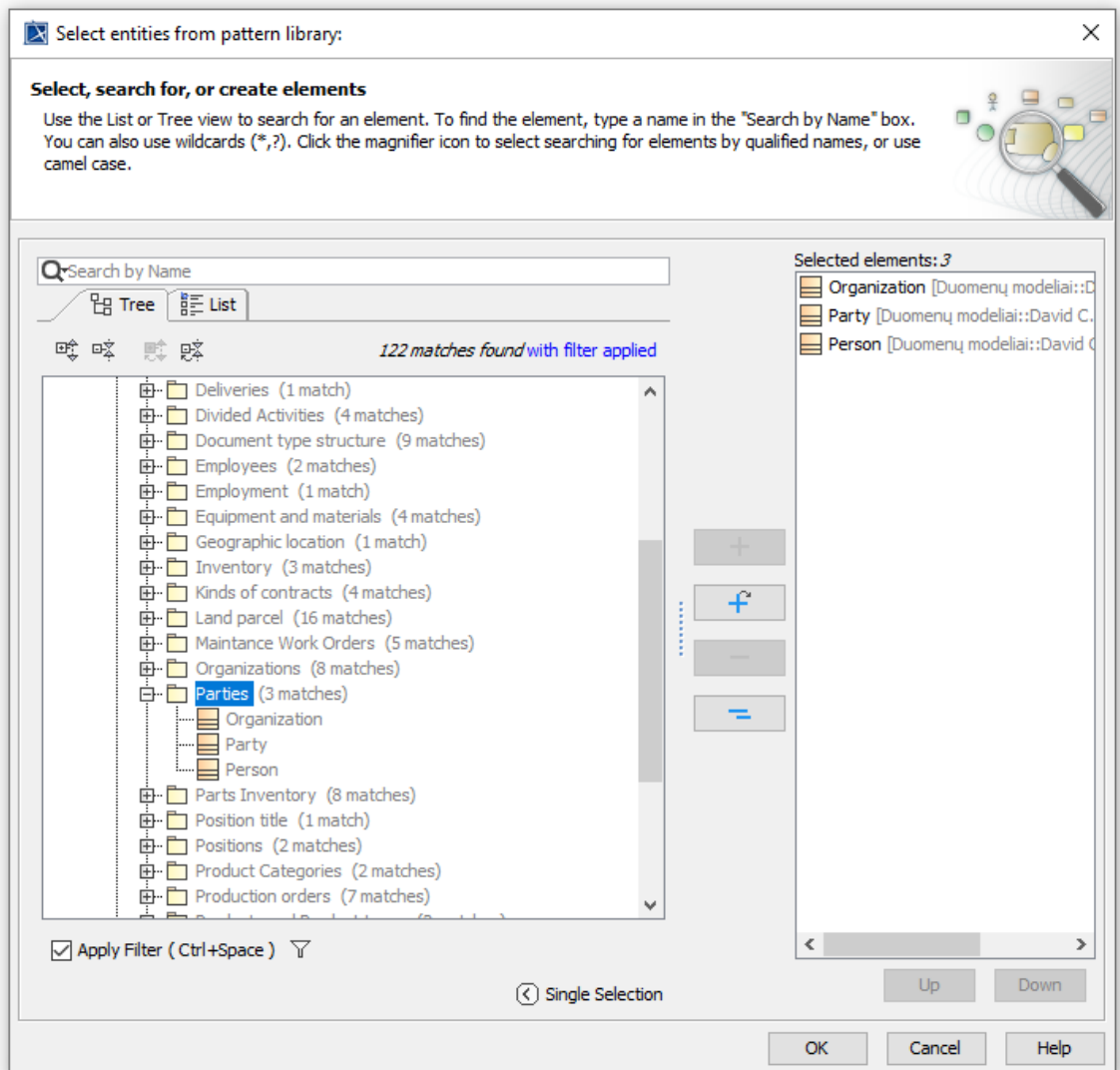


4. Sukurtoje klasių diagramoje paspaudus dešinę pelės mygtuką turėtų būti matomos 2 naujos funkcijos.

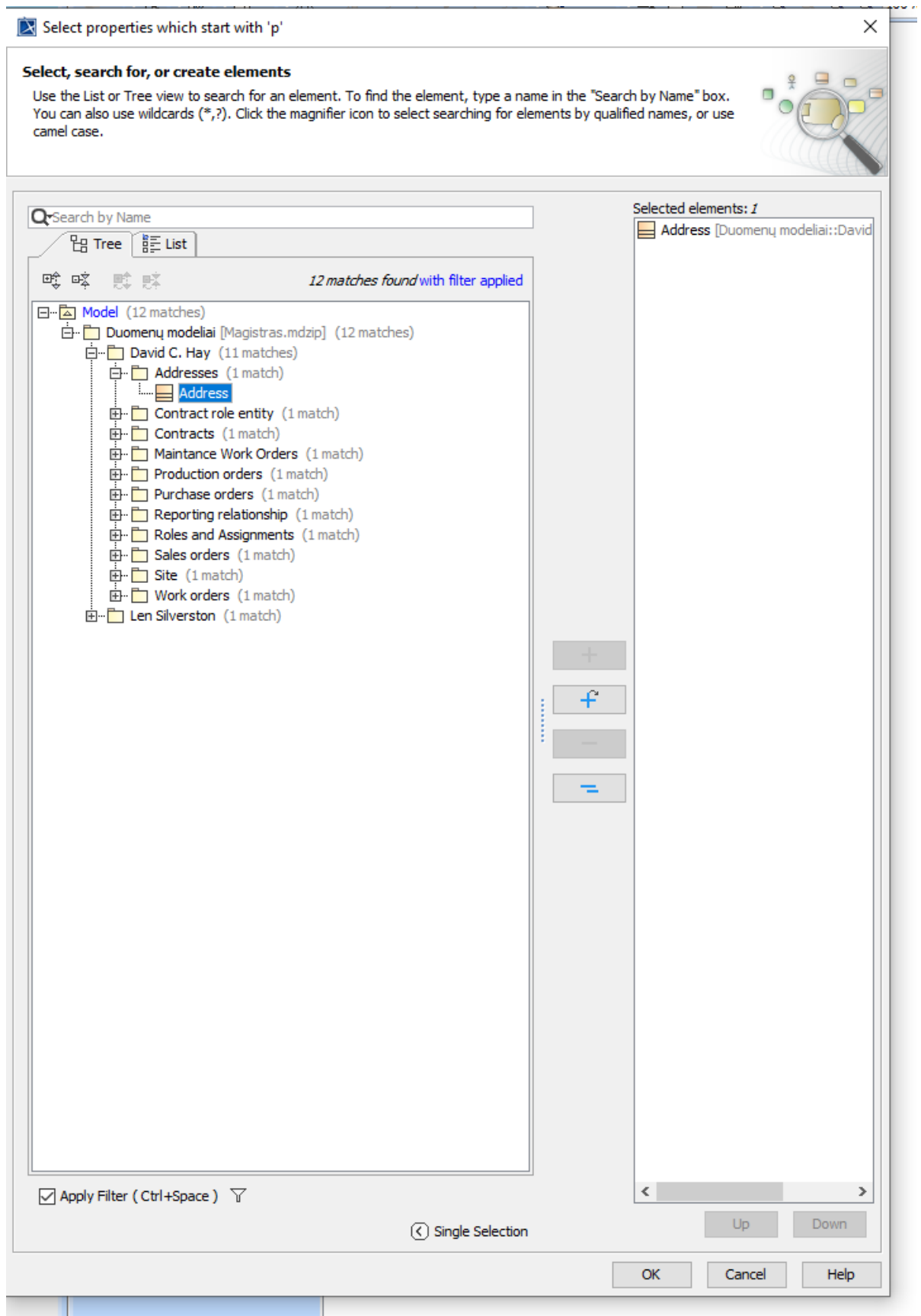


Bandomojo eksperimento modelio sudarymas

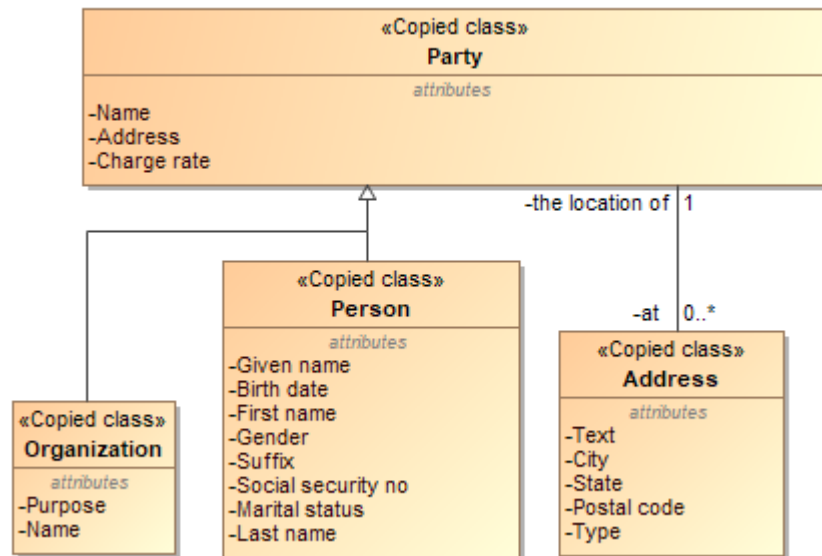
1. Pasirinkus pirmąją funkcija “Data model patterns library” surasti “Parties” šabloną, pasirinkti visus jo elementus ir paspausti OK.



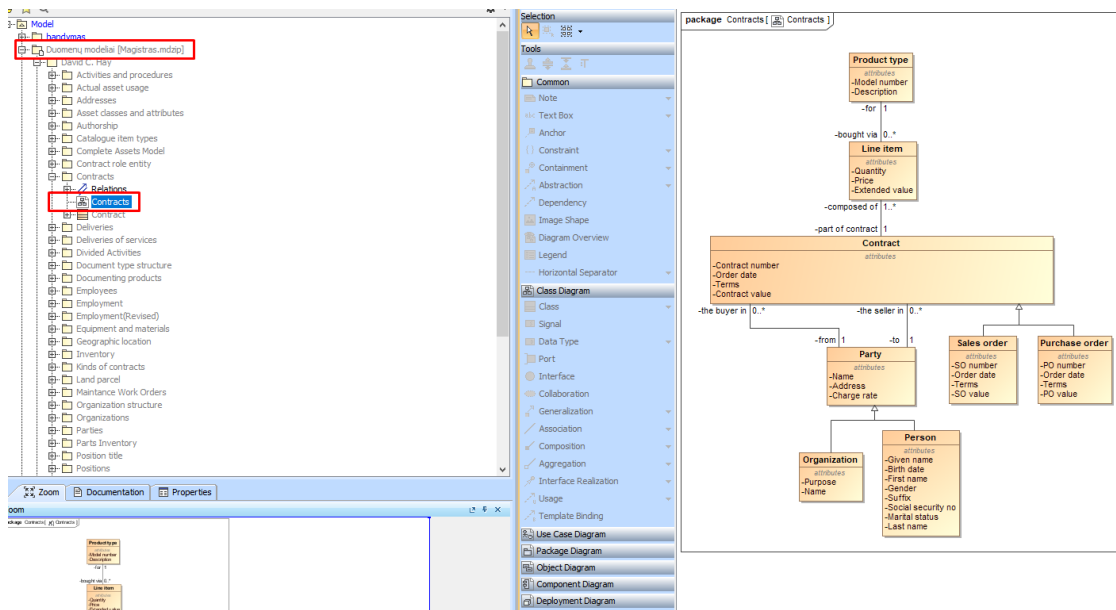
2. Pasirinkti "Party" klasę, ir ant jos paspausti dešinę pelės mygtuką. Tuomet pasirinkti antrąją plėtinio funkciją "Related library elements" atsidariusiame langele iš susijusių esybių pasirinkti "Address" ir paspausti "OK";



Įvykdžius šias 2 paprastas funkcijas gali gauti adresų duomenų modelį.



Prieš pradėdant eksperimentą siūloma susipažinti su duomenų modelių šablono esančiais bibliotekoje. Tai padaryti galite išskleidę bibliotekos katalogą ir atsidarant konkretaus šablono diagramą.



Eksperimento duomenų modeliai skirti patikrinti laiko trukmę reikalinga juos suprojektuoti.

1 dalis (Atliekant šią dalį pirma reikėtų suprojektuoti modelį nenaudojant šablonų bibliotekos ir parengto įrankio, po to pasinaudojant šablonų biblioteka bei įrankiu).

Organizacija projektuoja personalo valdymo posistemę ir tam ruošia duomenų bazės modelį. Personalo valdymo posistemėje bus saugoma informacija apie pačią organizaciją (Pavadinimas, Tikslas, Aprašymas). Organizacijoje dirbančius asmenis (Vardas, Pavardė, Lytis, Asmens kodas, Gimimo data). Kiekvienam organizacijos darbuotojui sukuriama vienas ar daugiau įdarbinimo įrašų (Pradžia, Pabaiga, Tipas). Kiekvienas įdarbinimo įrašas gali turėti daug paskyrimų. Paskyrimuose nurodoma informacija (Pradžia, Pabaiga, Tipas). Kiekviename paskyrimuose yra nurodoma konkreti pozicija, kuriai formuojamas paskyrimas. Pozicijas (Pareigos) apibrėžia organizacija, kiekviena pozicija turi nurodytą atlyginimo dydį, bei darbo aprašymą. Kiekviena pozicija gali turėti kelis pavadinimus, pozicijos pavadinimas taip pat yra įtraukiamas į paskyrimo įrašą. Kiekvienas darbuotojas bei pati organizacija gali turėti kelis adresus. Kiekviename adreso įrašuose saugoma informacija (Miestas, Pašto kodas, Tipas, Valstija, Adreso eilutės tekstas).

2 dalis (Atliekant šią dalį pirma reikėtų projektuoti modelį pasinaudojant biblioteka bei įrankiu, po to projektuojant įprastiniu būdu).

Organizacija taip pat teikia paslaugas bei parduoda prekes. Tiek paslaugų tiek prekių pardavimo metu sudaroma dvišalė sutartis tarp subjektų. Sutartyje nurodoma (Sutarties numeris, Užsakymo data, Sąlygos, Užsakymo vertė). Kiekvieną sutartį sudaro bent viena užsakymo eilutė. Užsakymo eilutėje nurodomas (Kiekis, Kaina, Bendra kaina). Konkretaus daikto pardavimo metu jis taip pat nurodomas užsakymo eilutėje. Paslaugos teikimo atveju, užsakymo eilutėje nurodoma veikla kurią reikia atlikti suteikiant paslaugą. Apie kiekvieną veiklą yra žinoma: Suplanuota pradžios data, Suplanuota pabaigos data, Darbo kaina. Paslaugą gali sudaryti daug veiklų. Paslauga turi savo aprašymą ir bendrą kainą.

2 priedas. Apklausos anketa

Šablonais grindžiamo koncepcinio duomenų modelio kūrimo metodo eksperimento apklausa

Jūsų pateikti asmeniniai duomenys bus anonimizuoti ir naudojami tik skaidyme į naudotojų grupes

Vardas *

Trumpo atsakymo tekstas

Pavardė *

Trumpo atsakymo tekstas

⋮
Kaip vertinate naują koncepcinio modelio kūrimo metodą? *

Blogai 1 2 3 4 5 Labai gerai

Kaip vertinate pristatytą įrankį? *

Blogai 1 2 3 4 5 Labai gerai

Ar rekomenduotumėte šį įrankį ir metodą kolegoms? *

Taip
 Ne

Ar manote, kad geriau susipažinus su biblioteka būtų pasiekiami geresni rezultatai? *

Taip
 Ne

Bendri eksperimento atsiliepimai

Ilgo atsakymo tekstas

3 priedas. Straipsnis

Straipsnis yra pateiktas ICIST 2020. Straipsnio tema: „Research of semi-automated database development using Data Model Patterns“

4 priedas. Šablonai

5.1. Activities and procedures

Šablonas sukurtas pagal David C. Hay [13]

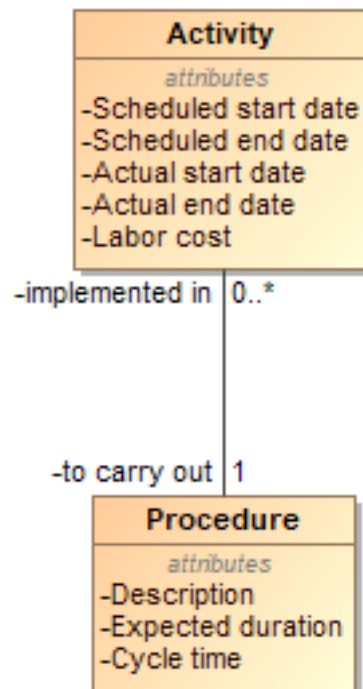


Figure 1. Activities and procedures

5.2. Actual asset usage

Šablonas sukurtas pagal David C. Hay [13]

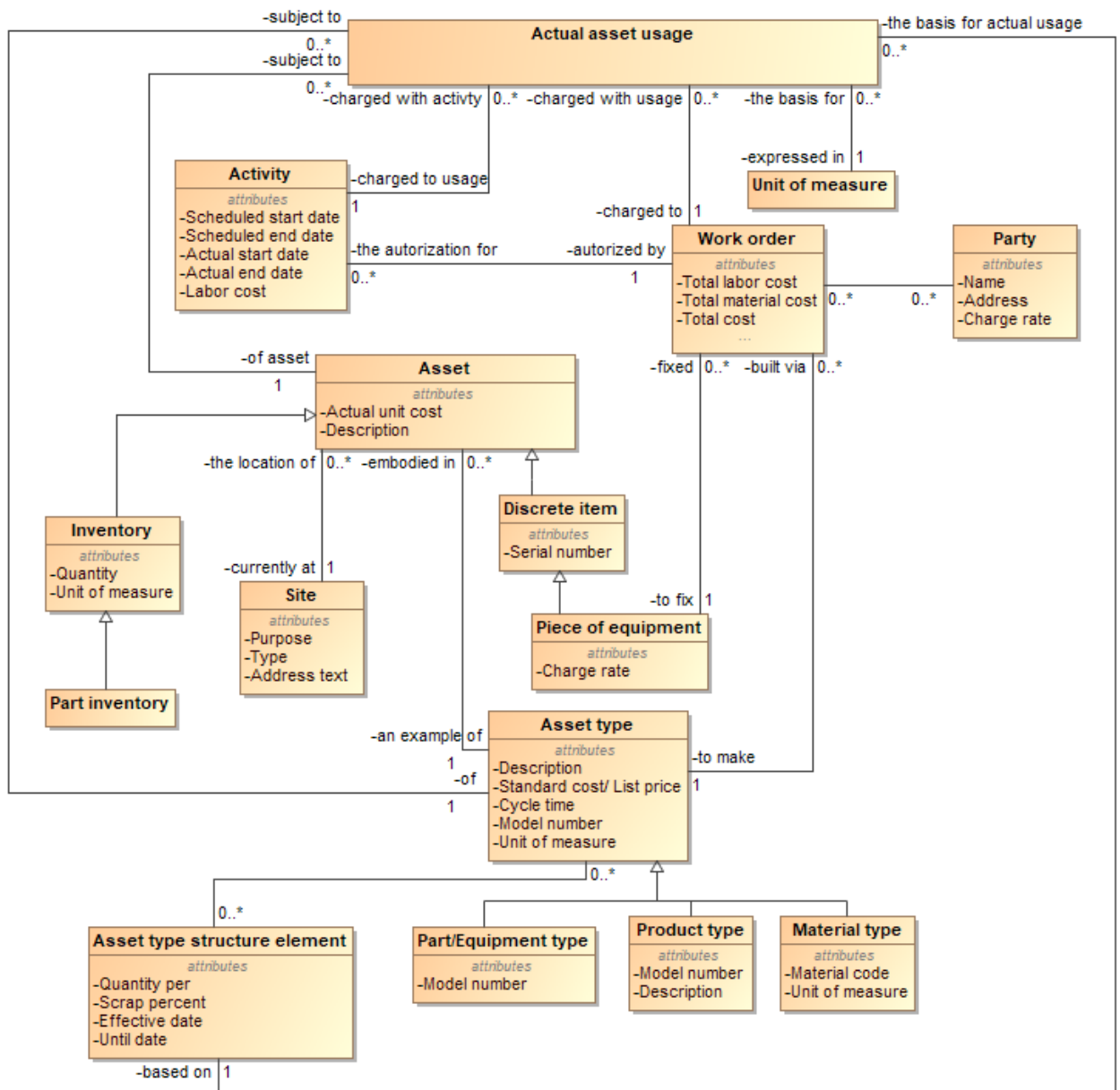


Figure 2. Actual asset usage

5.3. Addresses

Šablonas sukurtas pagal David C. Hay [13]

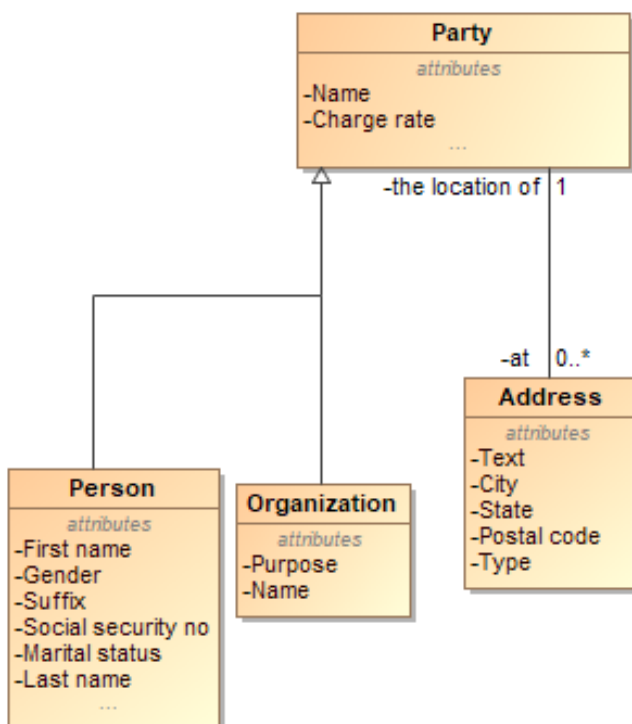


Figure 3. Addresses

5.4. Asset classes and attributes

Šablonas sukurtas pagal David C. Hay [13]

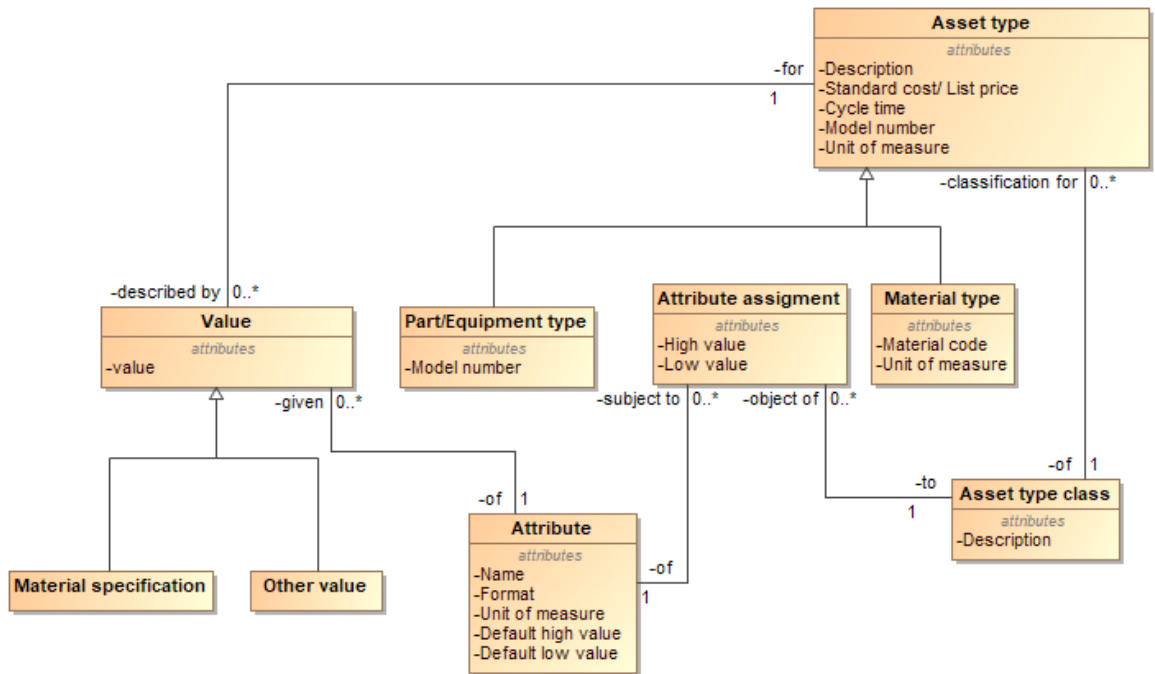


Figure 4. Asset classes and attributes

5.5. Authorship

Šablonas sukurtas pagal David C. Hay [13]

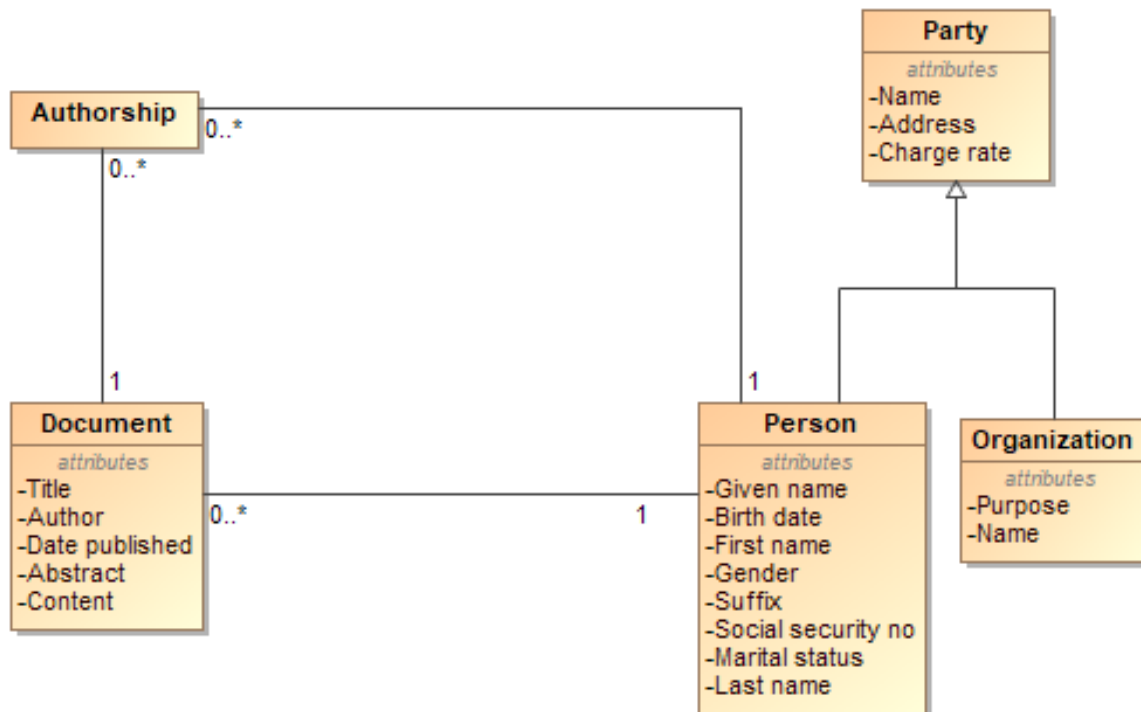


Figure 5. Authorship

5.6. Catalogue item types

Šablonas sukurtas pagal David C. Hay [13]

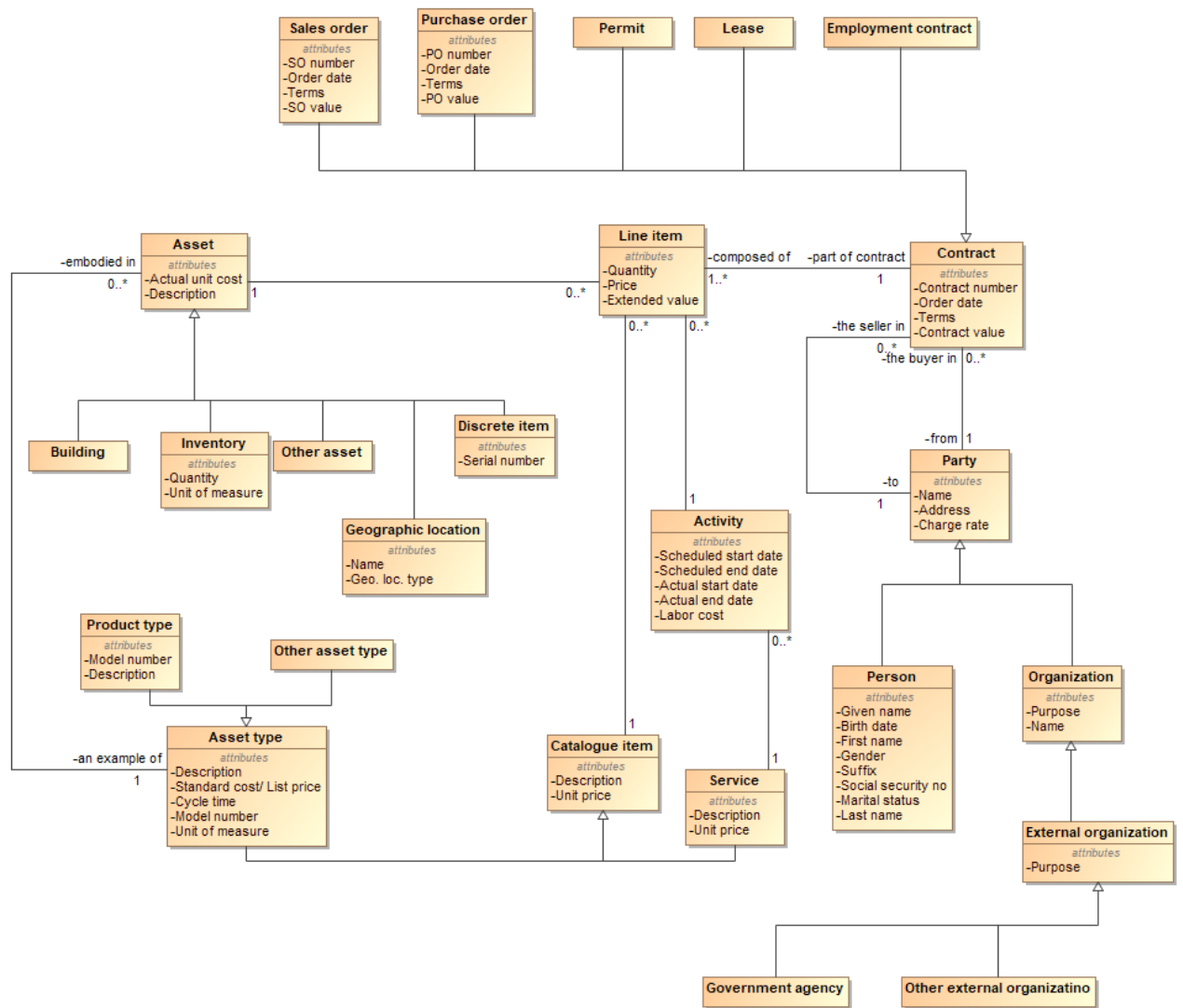


Figure 6. Catalogue item types

5.8. Contract role entity

Šablonas sukurtas pagal David C. Hay [13]

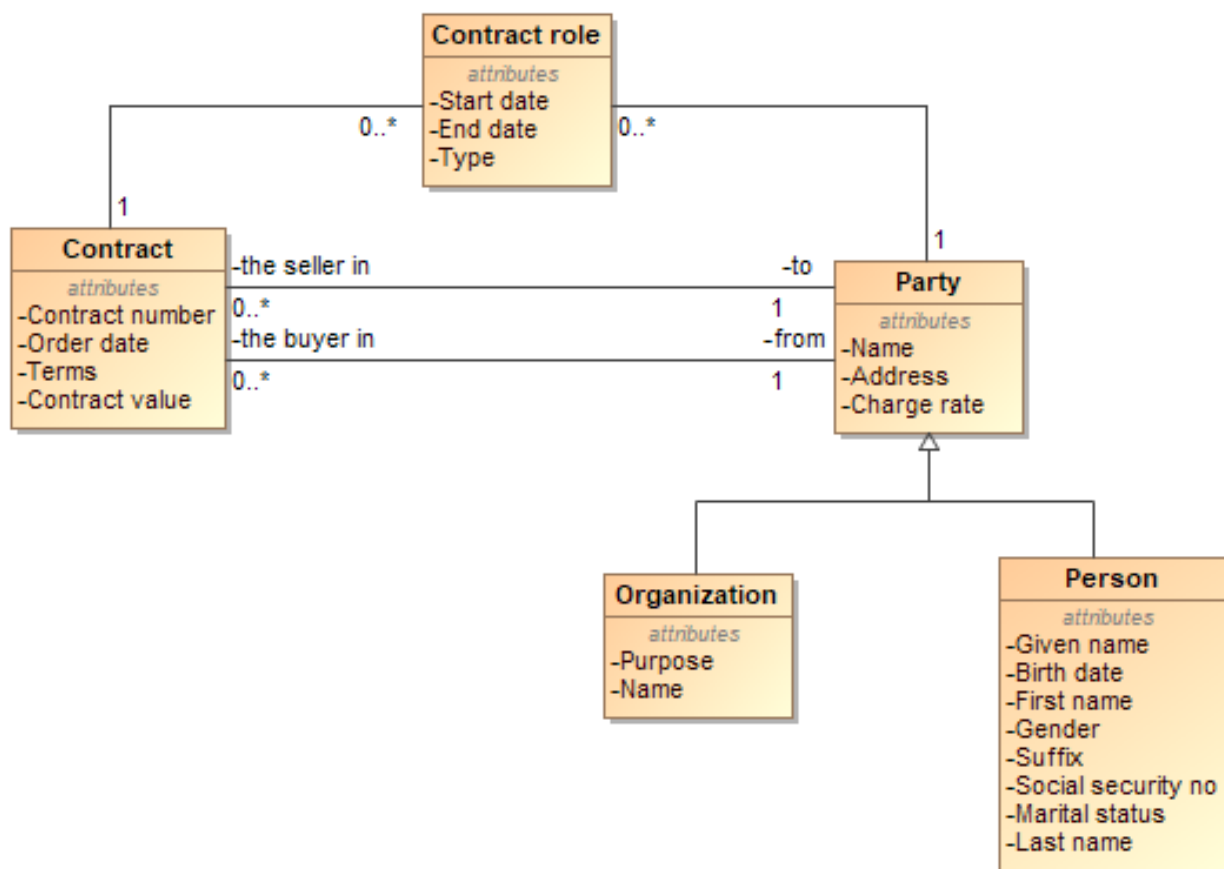


Figure 8. Contract role entity

5.9. Contracts

Šablonas sukurtas pagal David C. Hay [13]

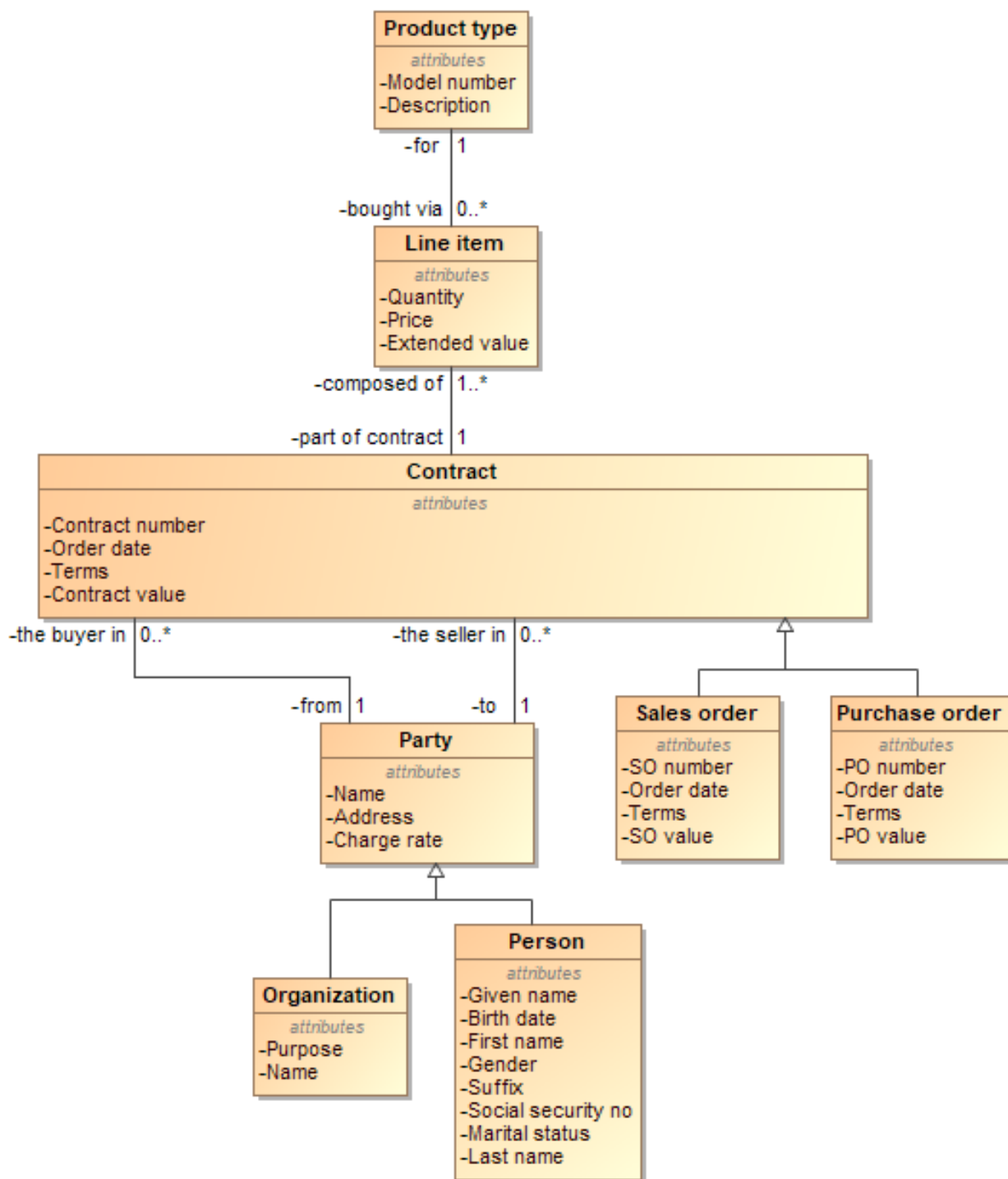


Figure 9. Contracts

5.10. Deliveries

Šablonas sukurtas pagal David C. Hay [13]

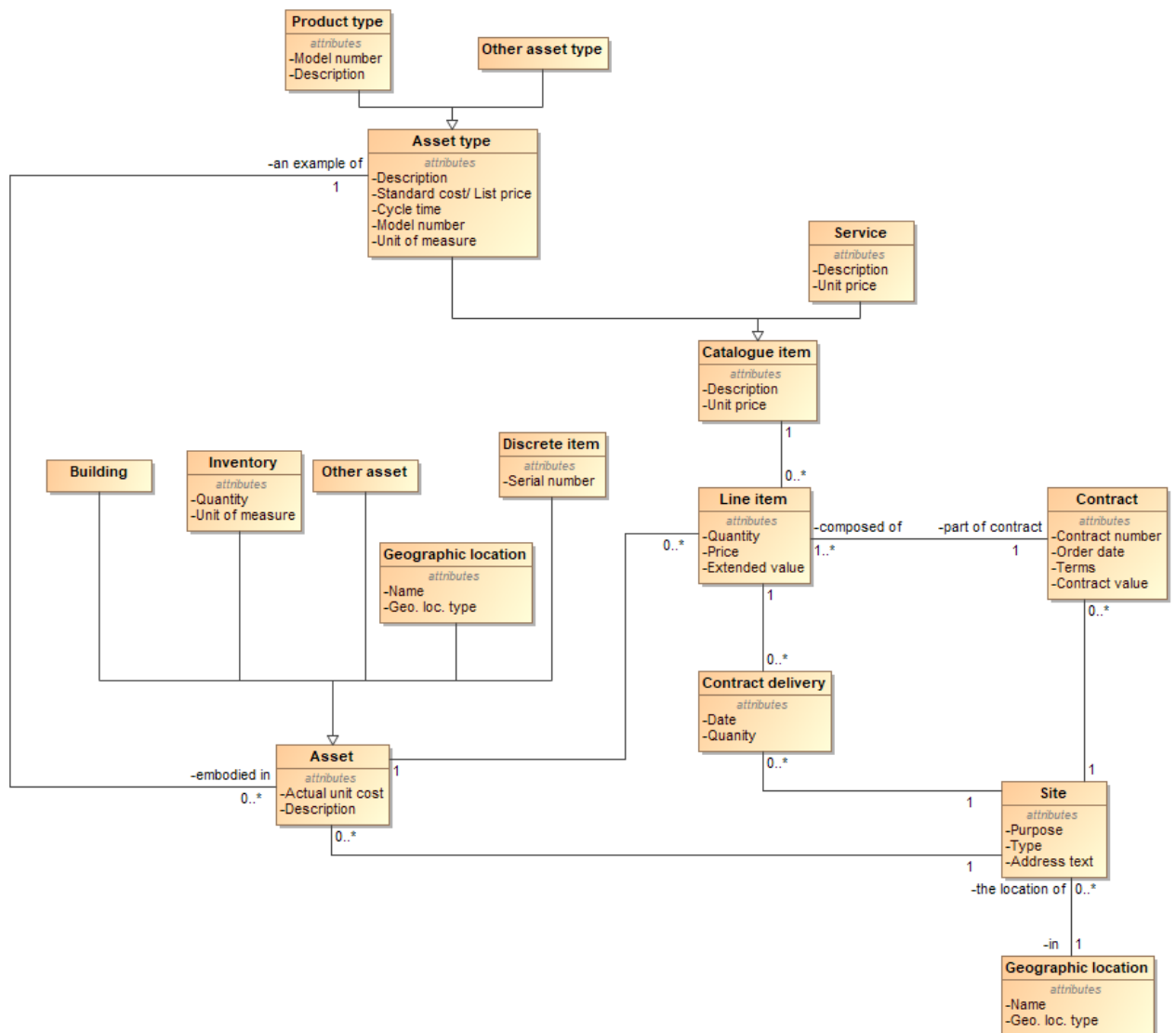


Figure 10. Deliveries

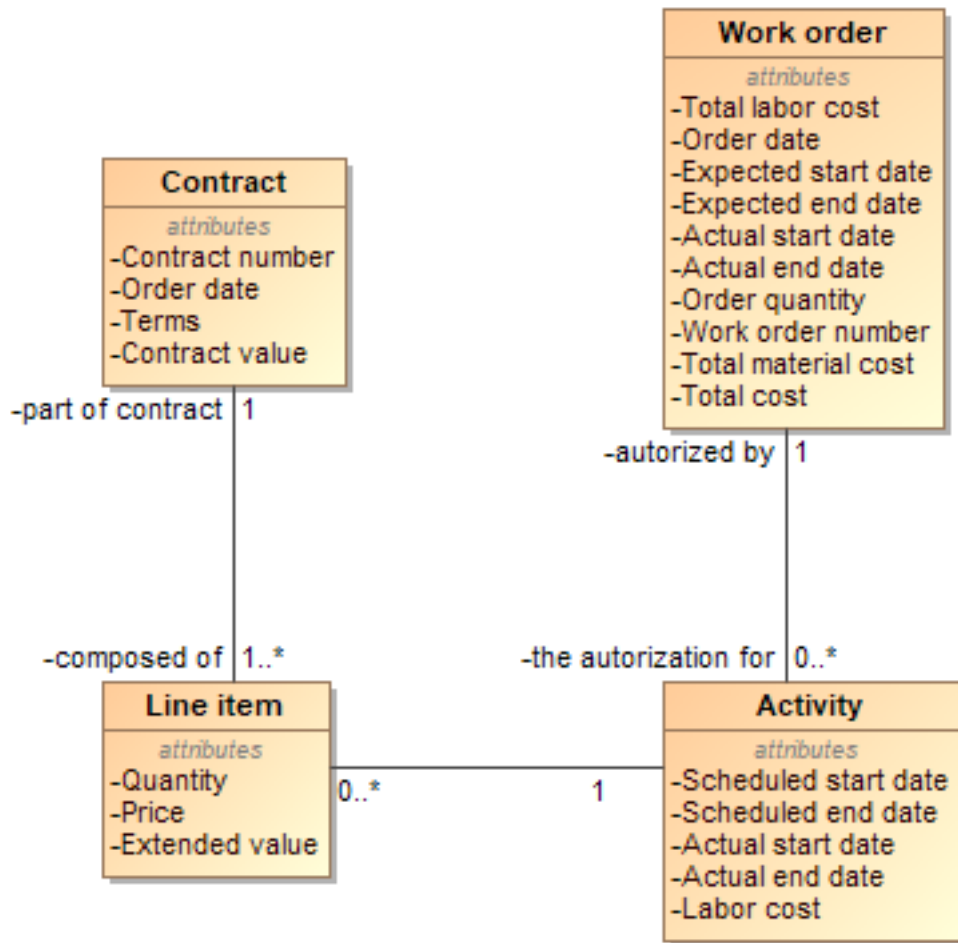


Figure 11. Deliveries of services

5.11. Divided Activities

Šablonas sukurtas pagal David C. Hay [13]

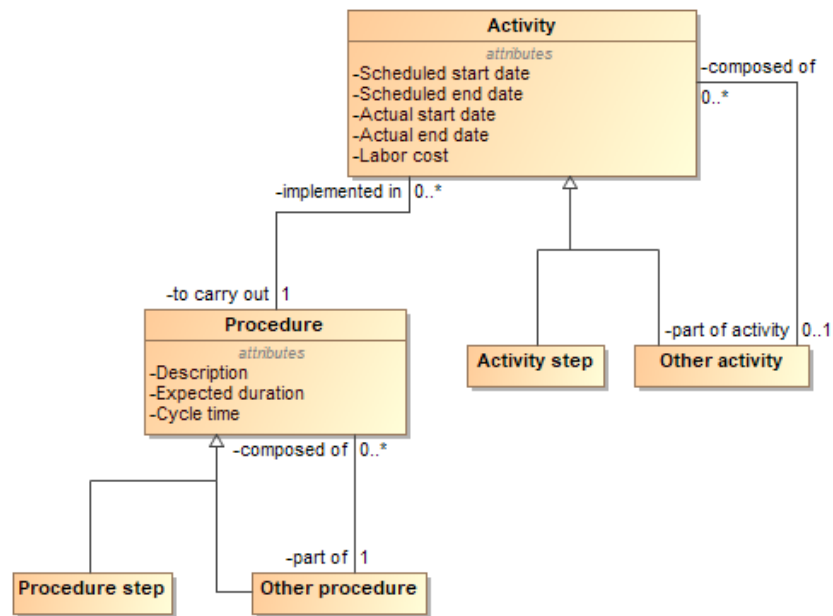


Figure 12. Divided Activities

5.12. Document type structure

Šablonas sukurtas pagal David C. Hay [13]

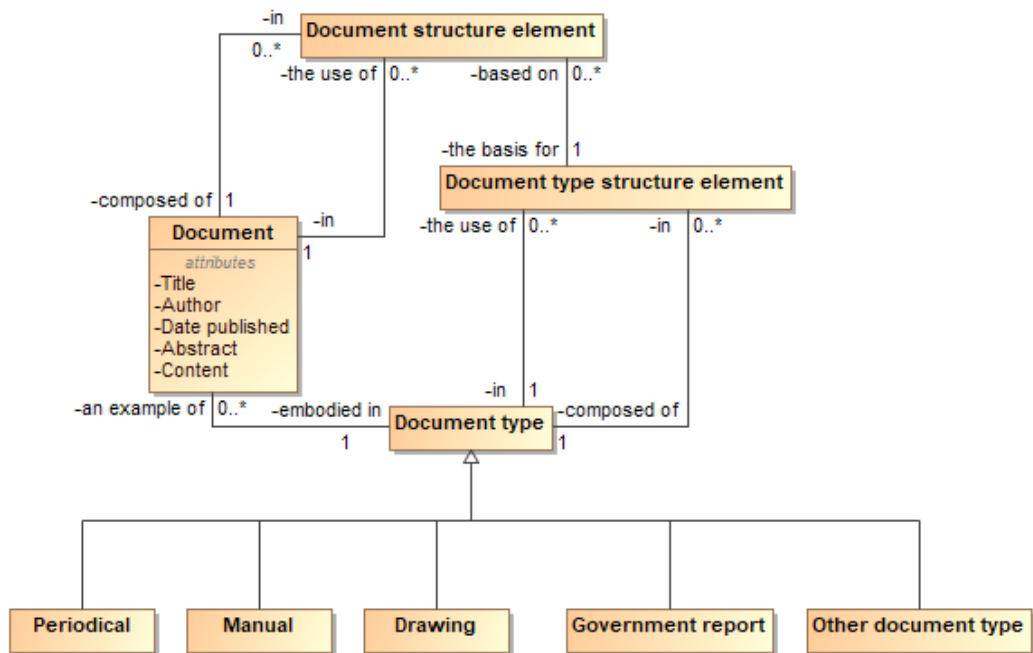


Figure 13. Document type structure

5.13. Documenting products

Šablonas sukurtas pagal David C. Hay [13]

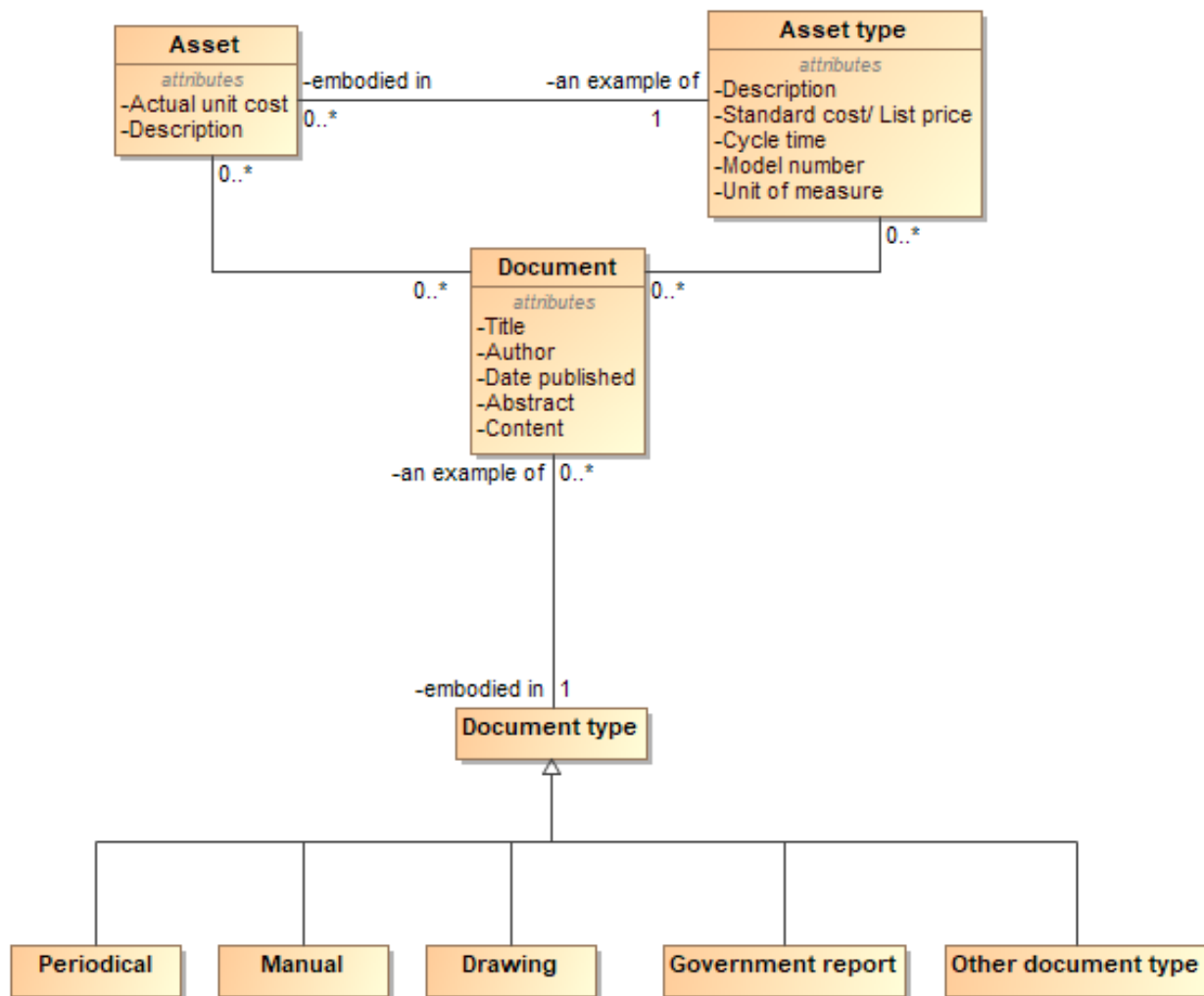


Figure 14. Documenting products

5.14. Employees

Šablonas sukurtas pagal David C. Hay [13]

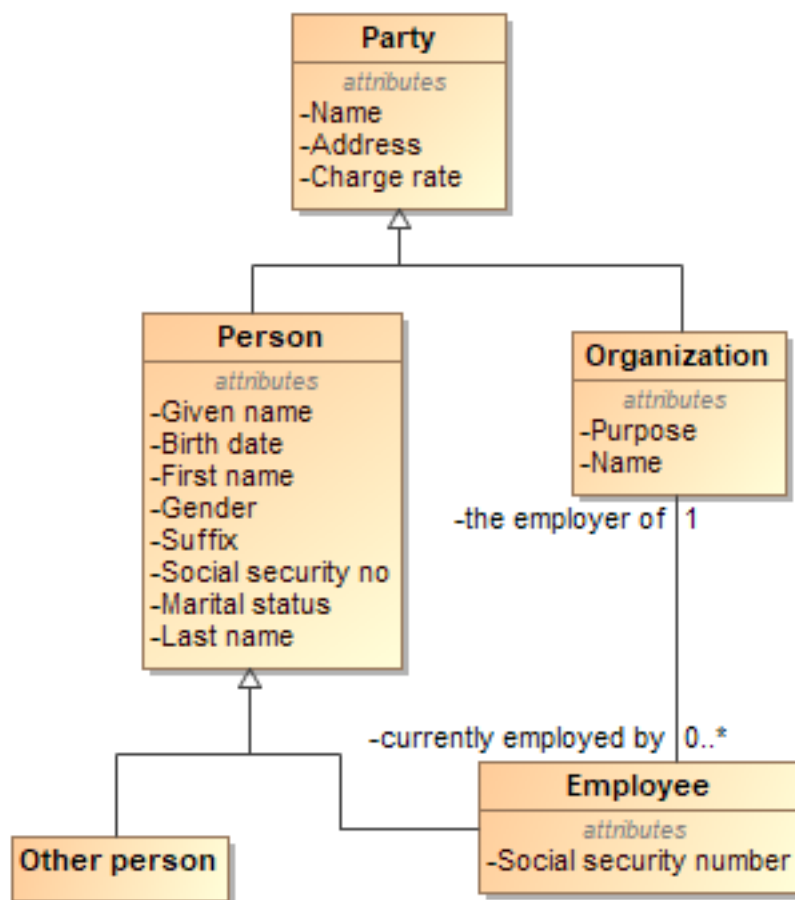


Figure 15. Employees

5.15. Employment

Šablonas sukurtas pagal David C. Hay [13]

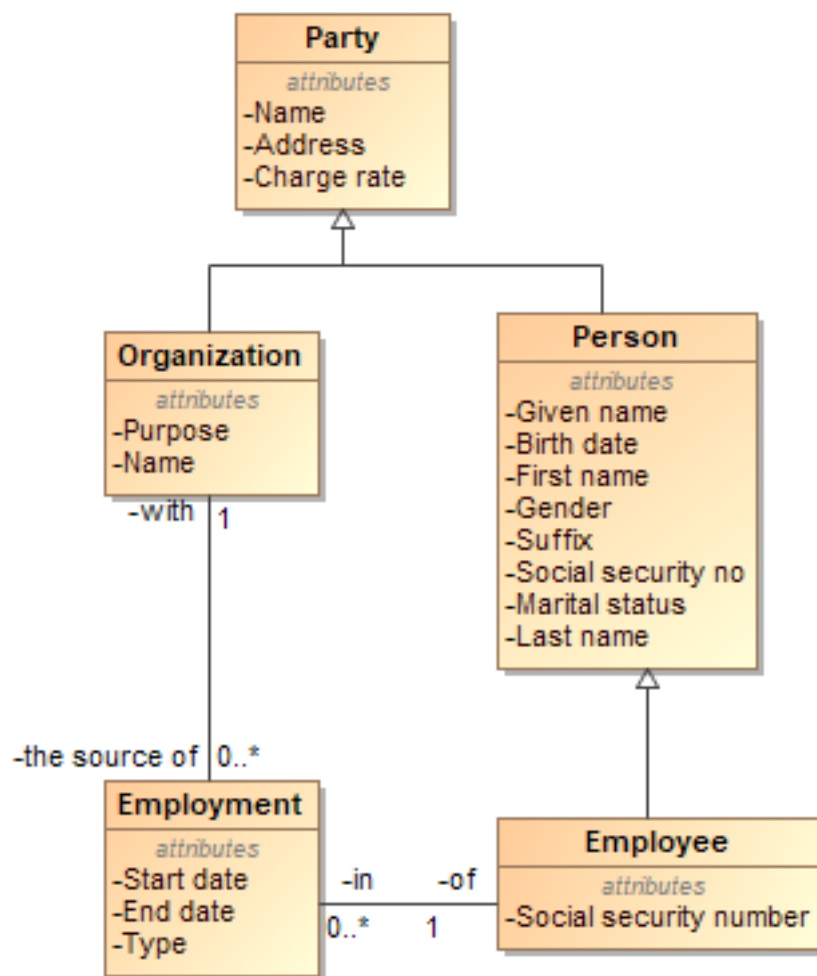


Figure 16. Employment

5.16. Employment(Revised)

Šablonas sukurtas pagal David C. Hay [13]

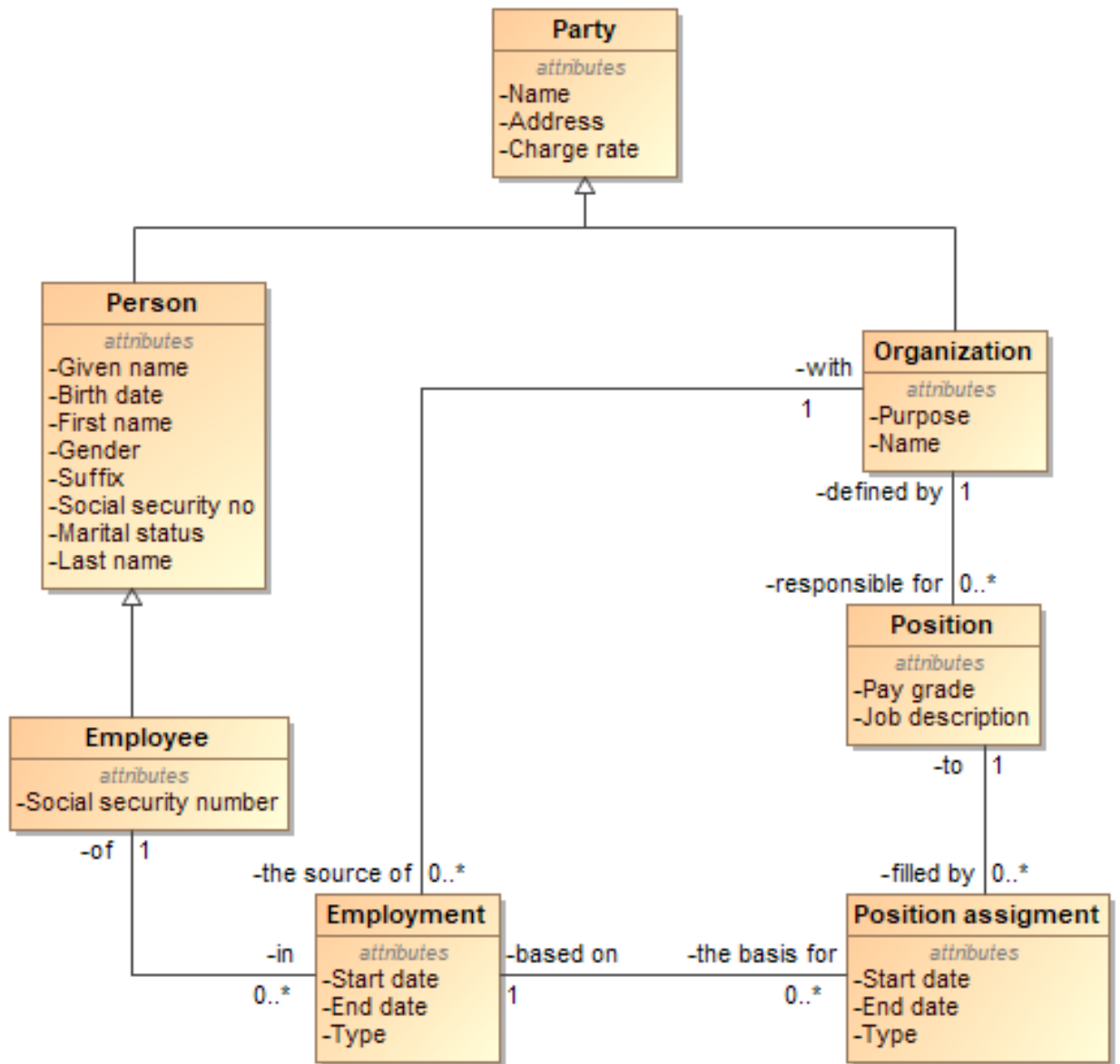


Figure 17. Employment(Revised)

5.17. Equipment and materials

Šablonas sukurtas pagal David C. Hay [13]

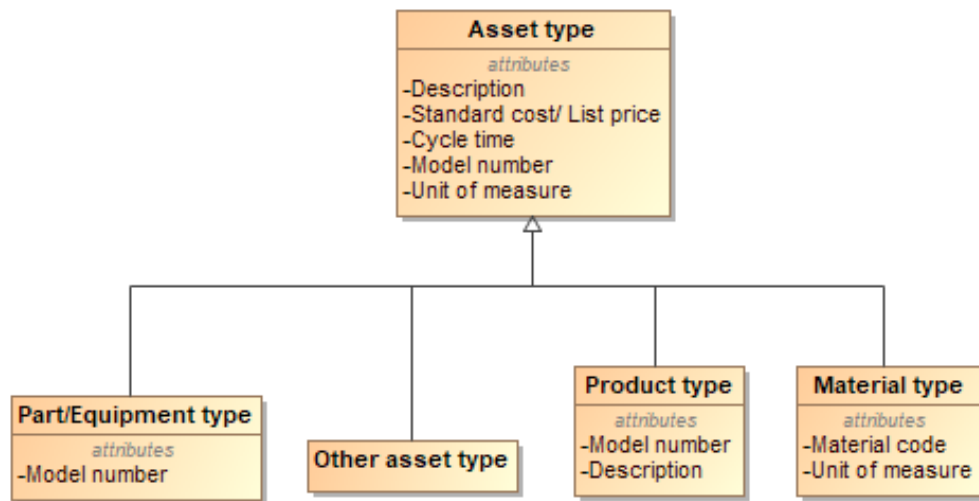


Figure 18. Equipment and materials

5.18. Geographic location

Šablonas sukurtas pagal David C. Hay [13]

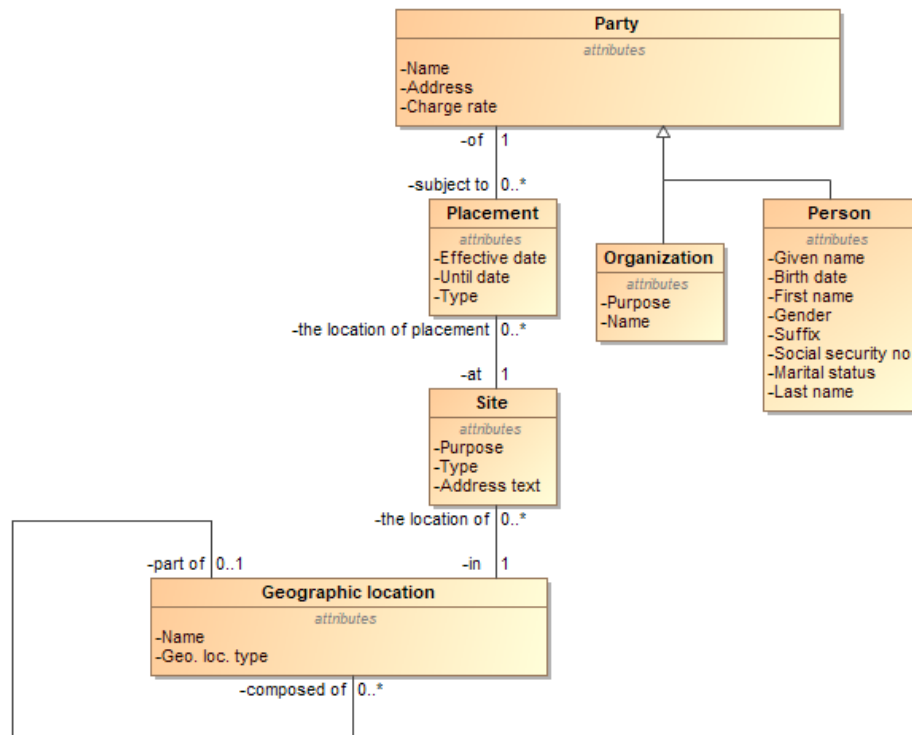


Figure 19. Geographic location

5.19. Inventory

Šablonas sukurtas pagal David C. Hay [13]

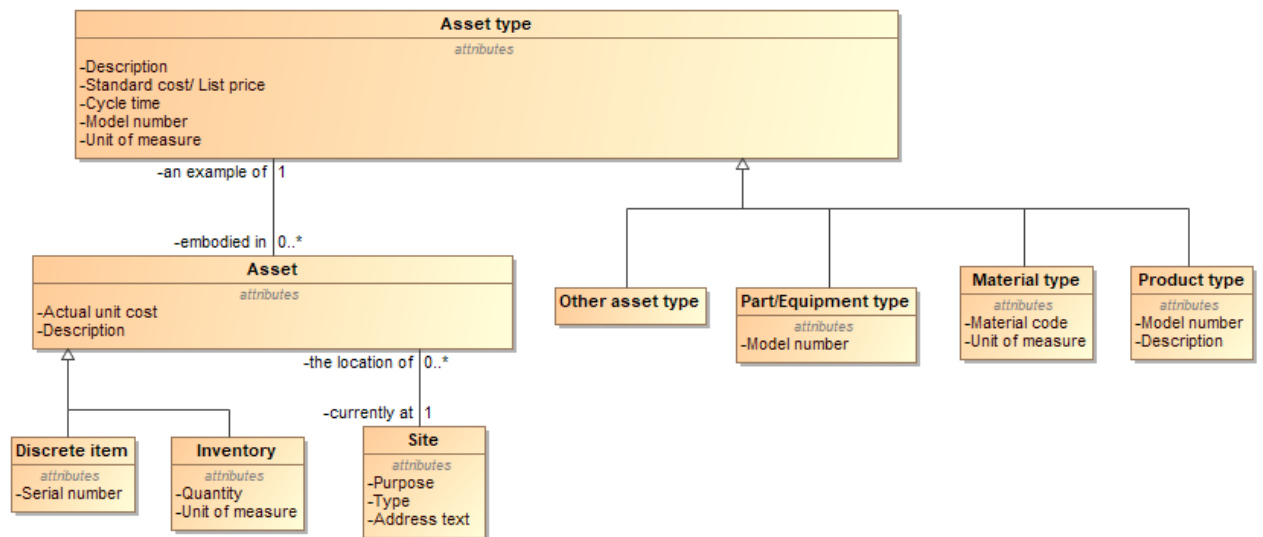


Figure 20. Inventory

5.20. Kinds of contracts

Šablonas sukurtas pagal David C. Hay [13]

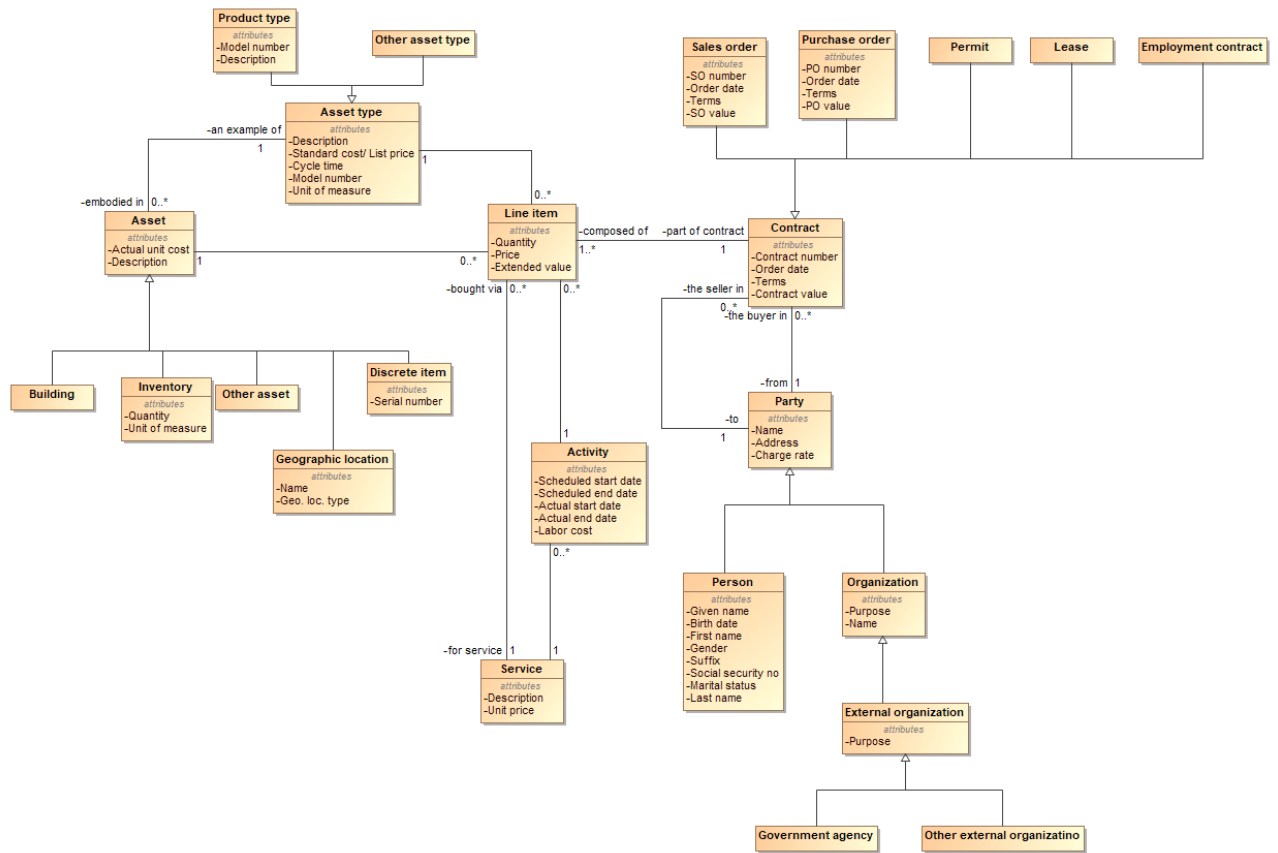


Figure 21. Kinds of contracts

5.21. Land parcel

Šablonas sukurtas pagal David C. Hay [13]

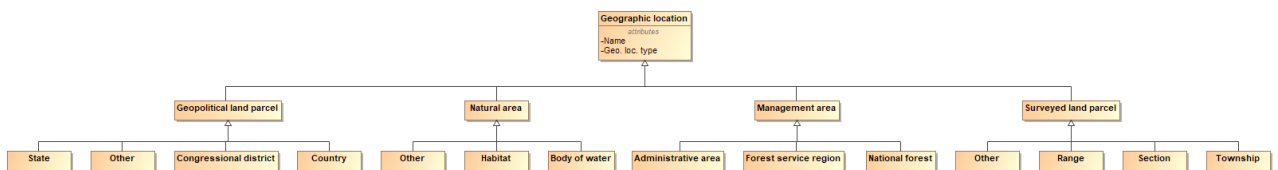


Figure 22. Land parcel

5.22. Maintenance Work Orders

Šablonas sukurtas pagal David C. Hay [13]

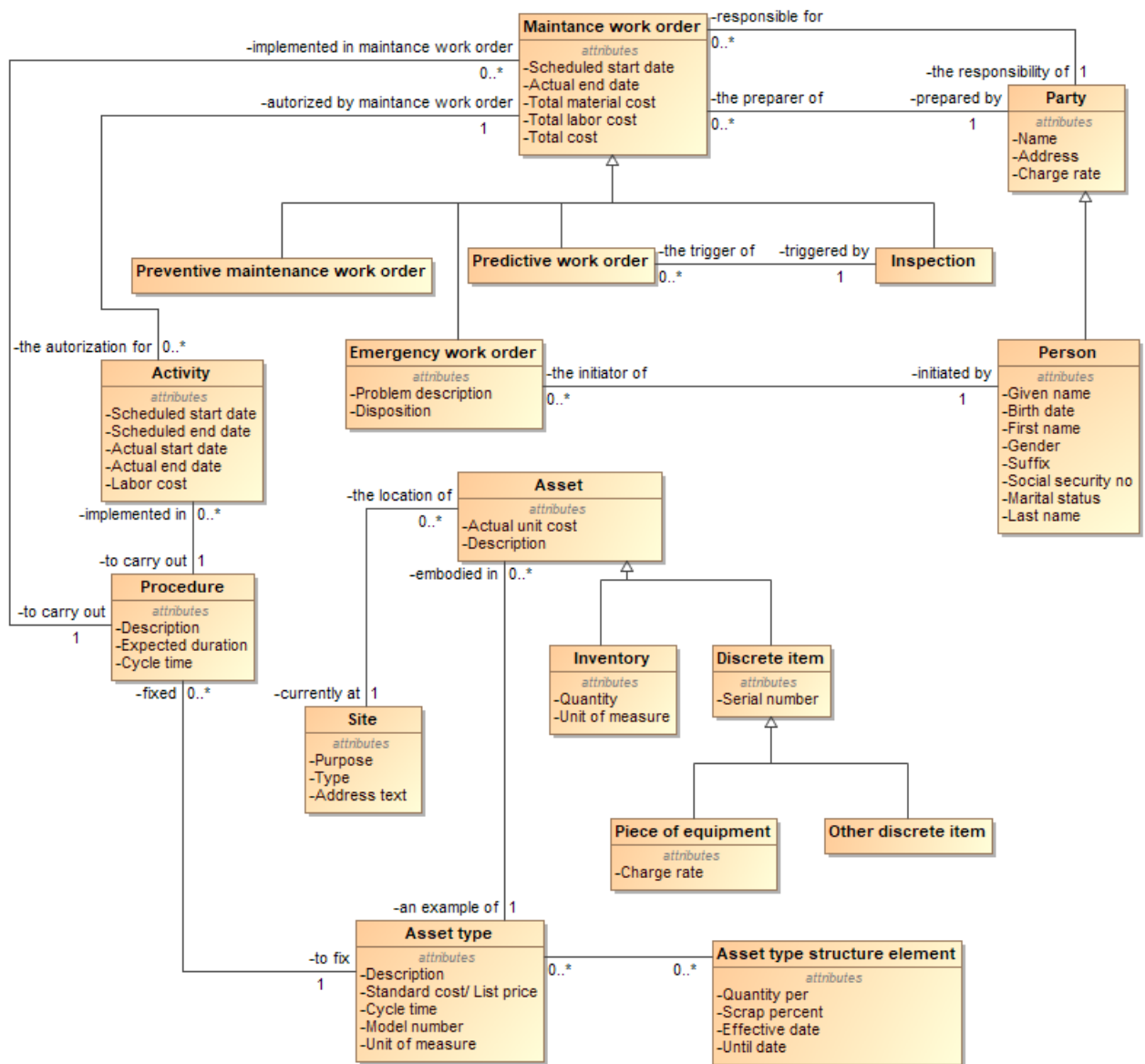
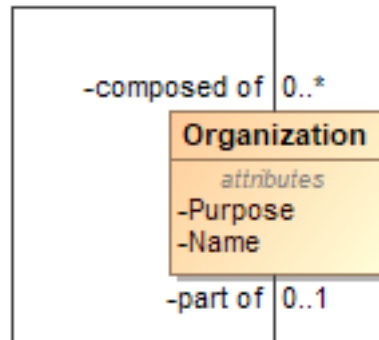


Figure 23. Maintenance Work Orders

5.23. Organization structure

Šablonas sukurtas pagal David C. Hay [13]



5.24. Organizations

Šablonas sukurtas pagal David C. Hay [13]

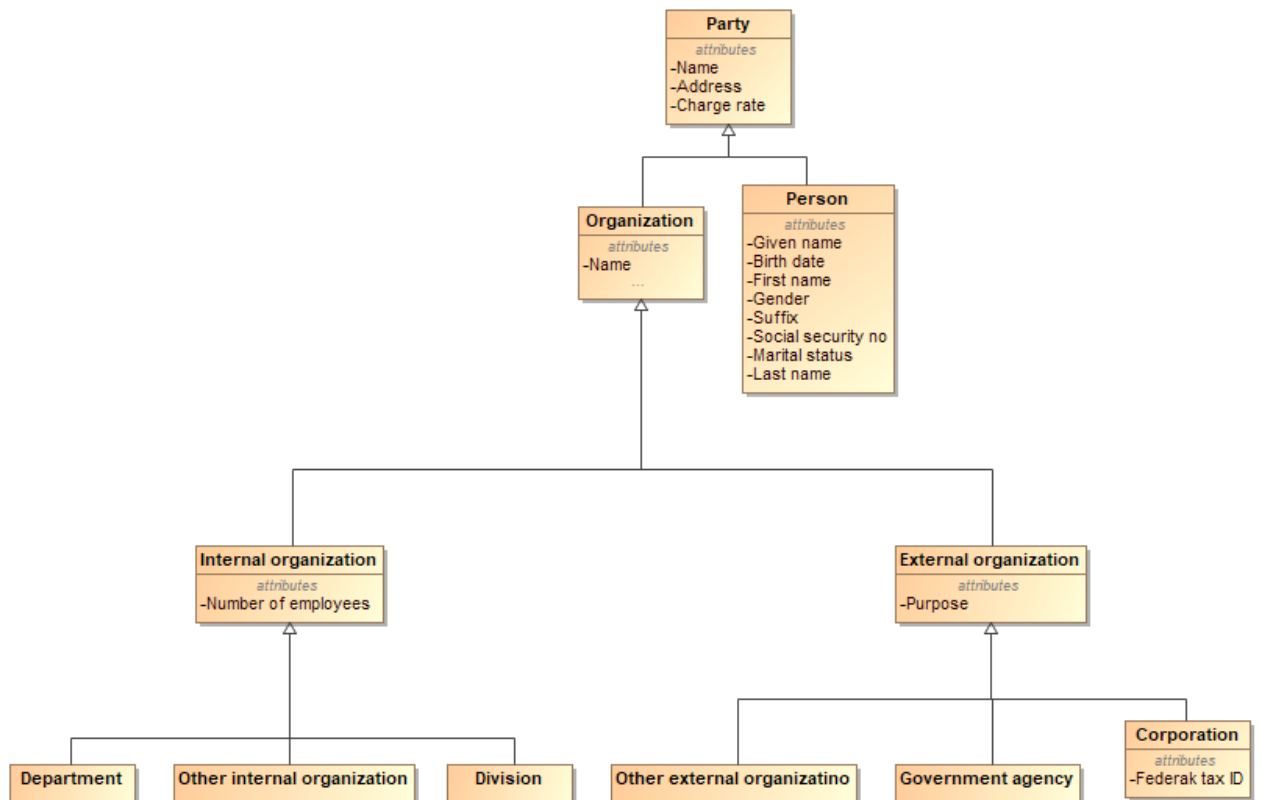


Figure 24. Organizations

5.25. Parties

Šablonas sukurtas pagal David C. Hay [13]

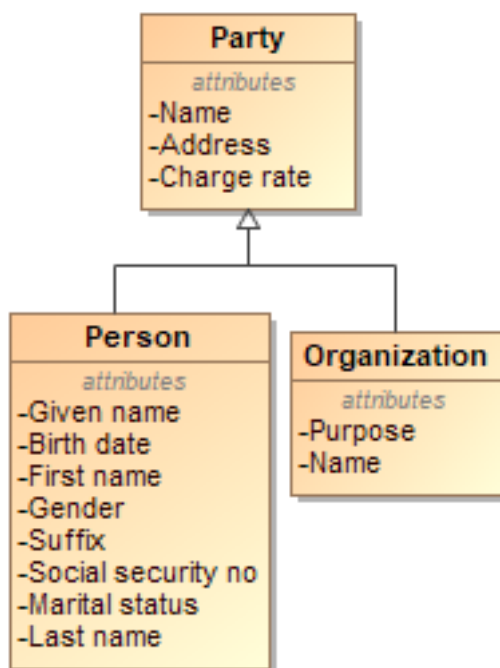


Figure 25. Parties

5.26. Parts Inventory

Šablonas sukurtas pagal David C. Hay [13]

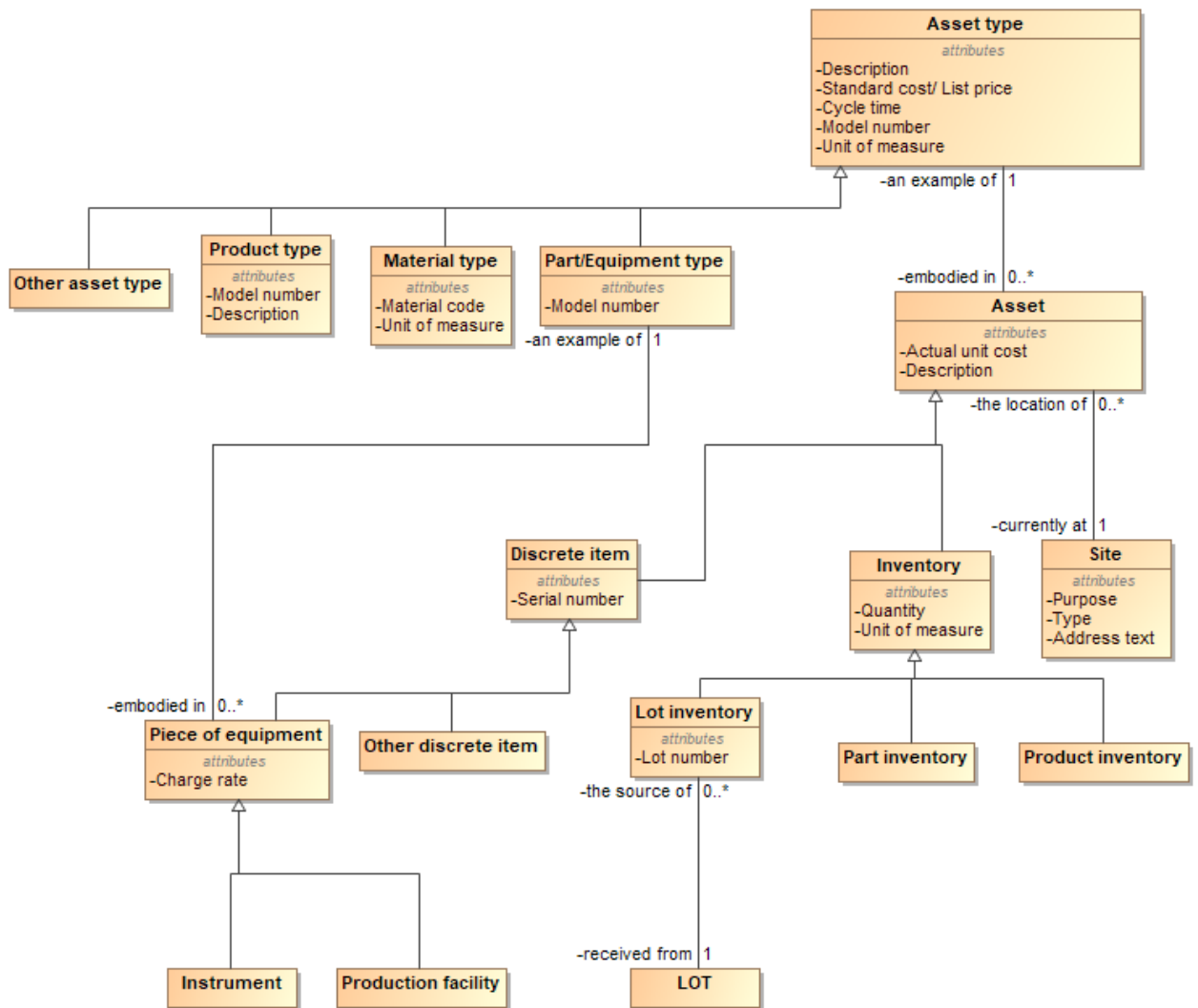


Figure 26. Parts Inventory

5.27. Position title

Šablonas sukurtas pagal David C. Hay [13]

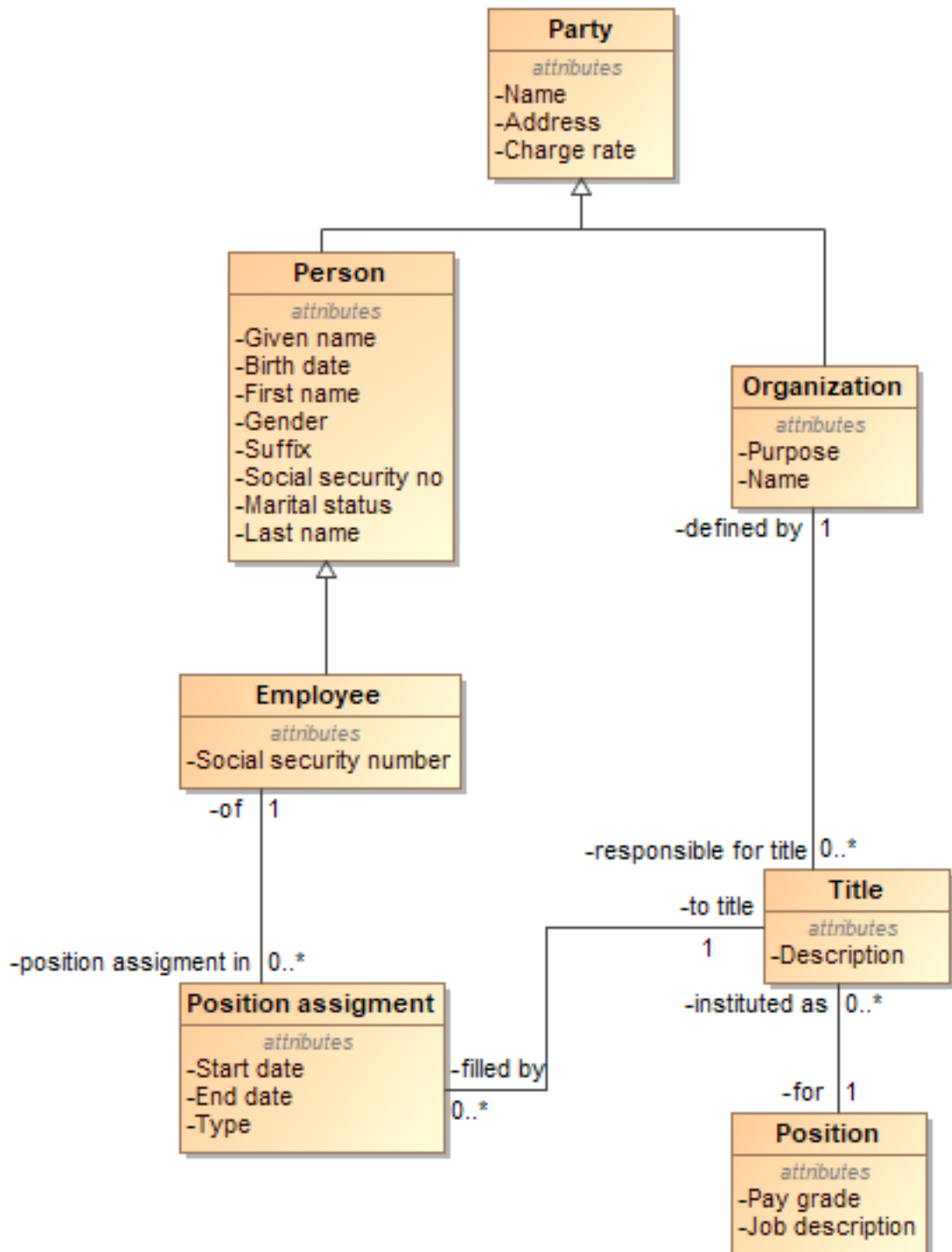


Figure 27. Position title

5.28. Positions

Šablonas sukurtas pagal David C. Hay [13]

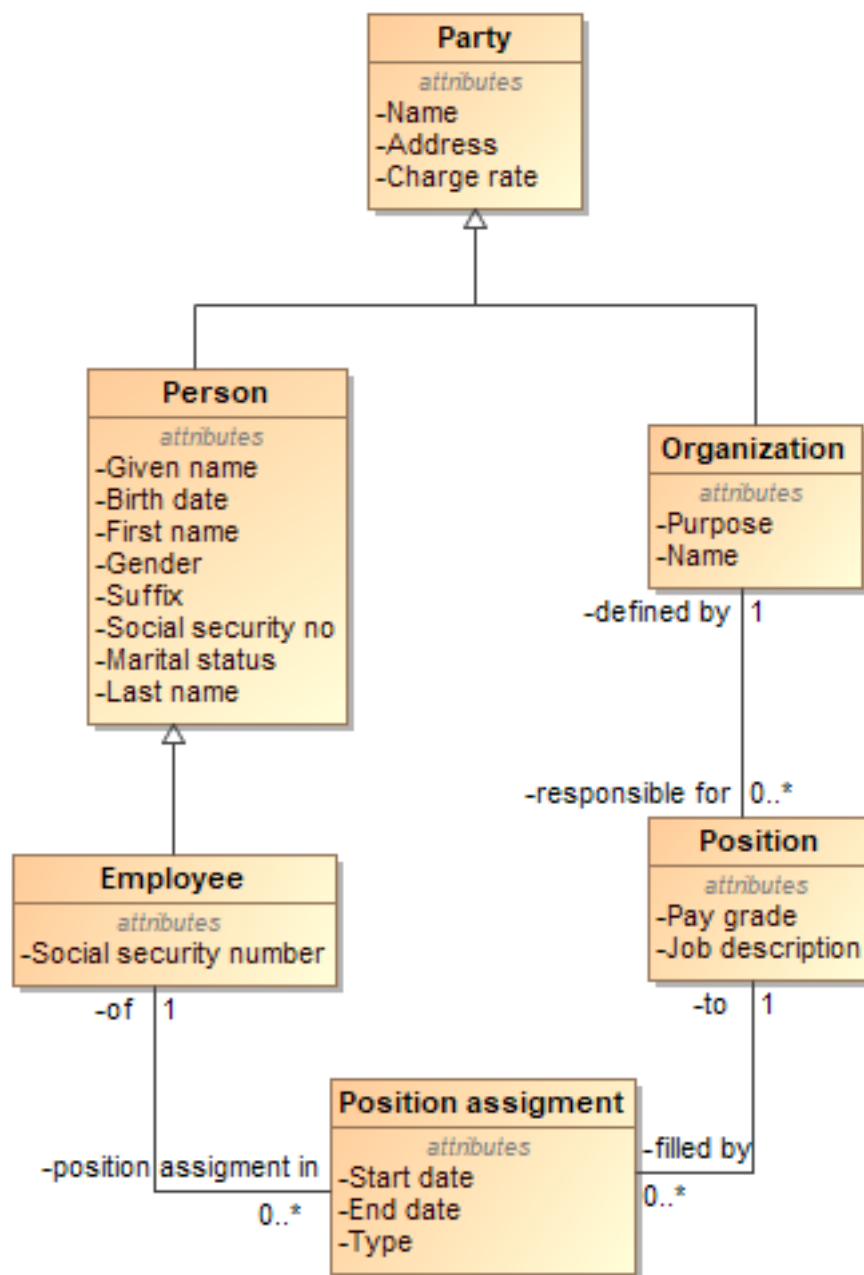


Figure 28. Positions

5.29. Product Categories

Šablonas sukurtas pagal David C. Hay [13]

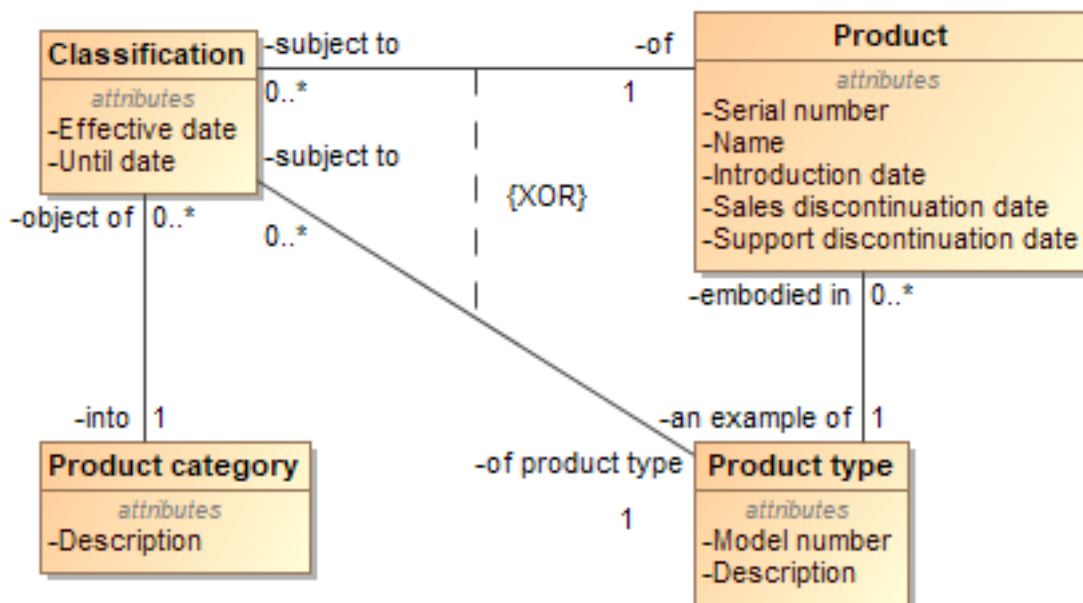


Figure 29. Product Categories

5.30. Production orders

Šablonas sukurtas pagal David C. Hay [13]

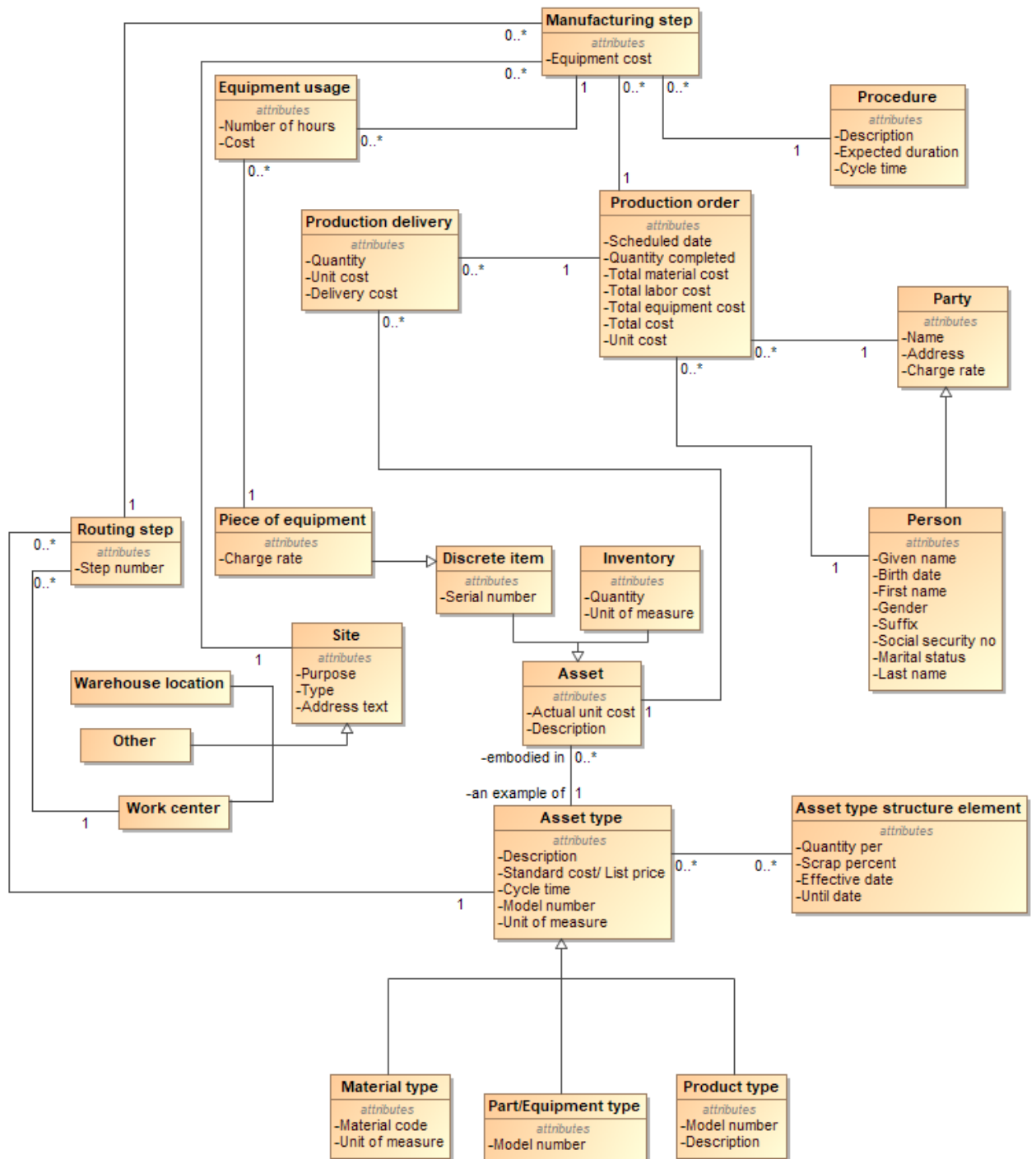


Figure 30. Production orders

5.31. Products and Product types

Šablonas sukurtas pagal David C. Hay [13]

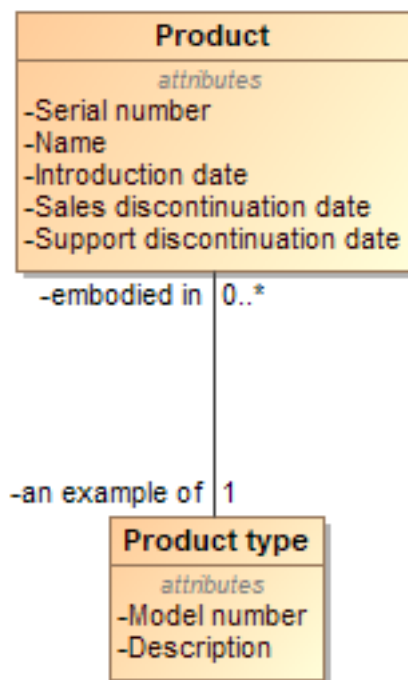


Figure 31. Products and Product types

5.32. Products and services

Šablonas sukurtas pagal David C. Hay [13]

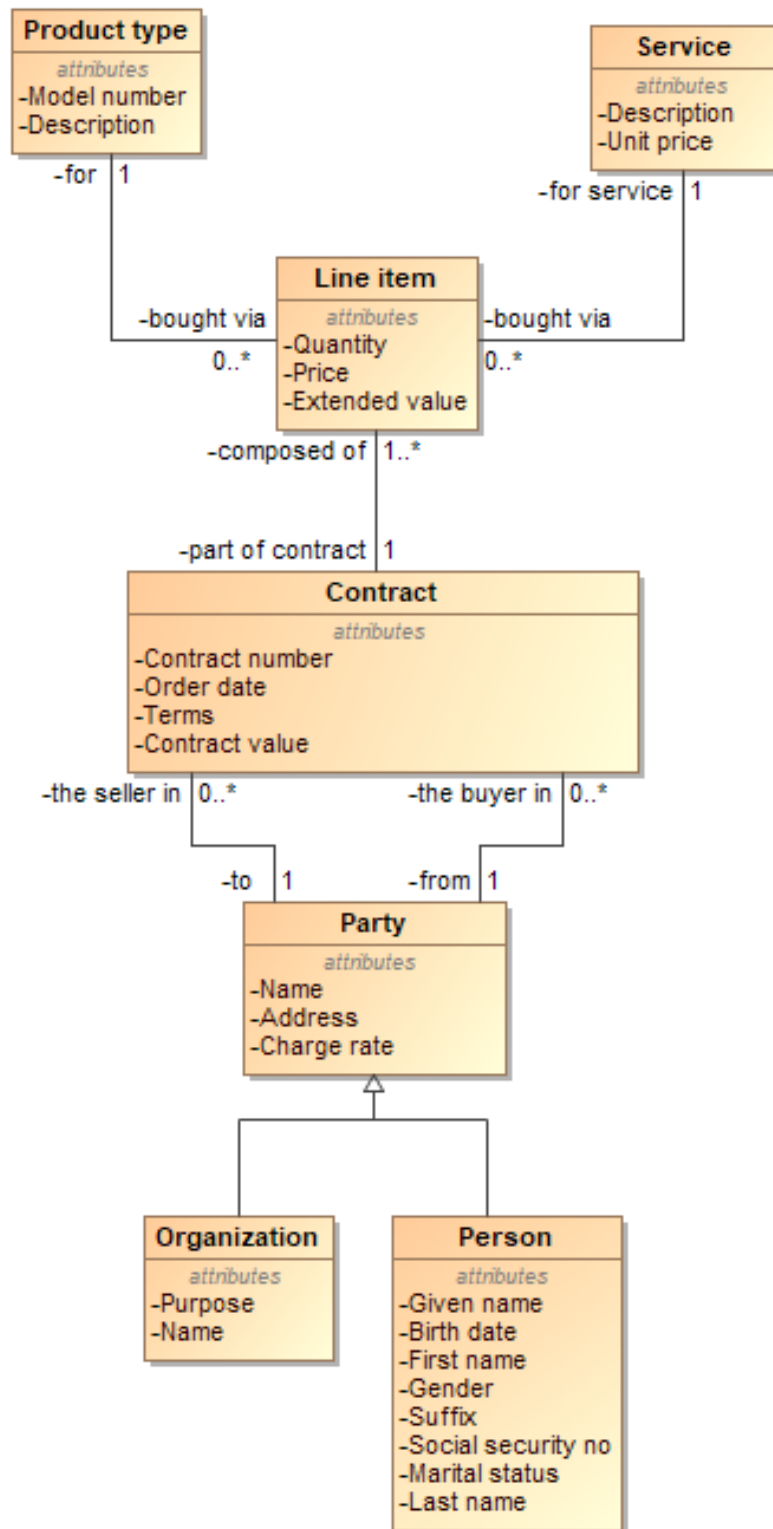


Figure 32. Products and services

5.33. Purchase orders

Šablonas sukurtas pagal David C. Hay [13]

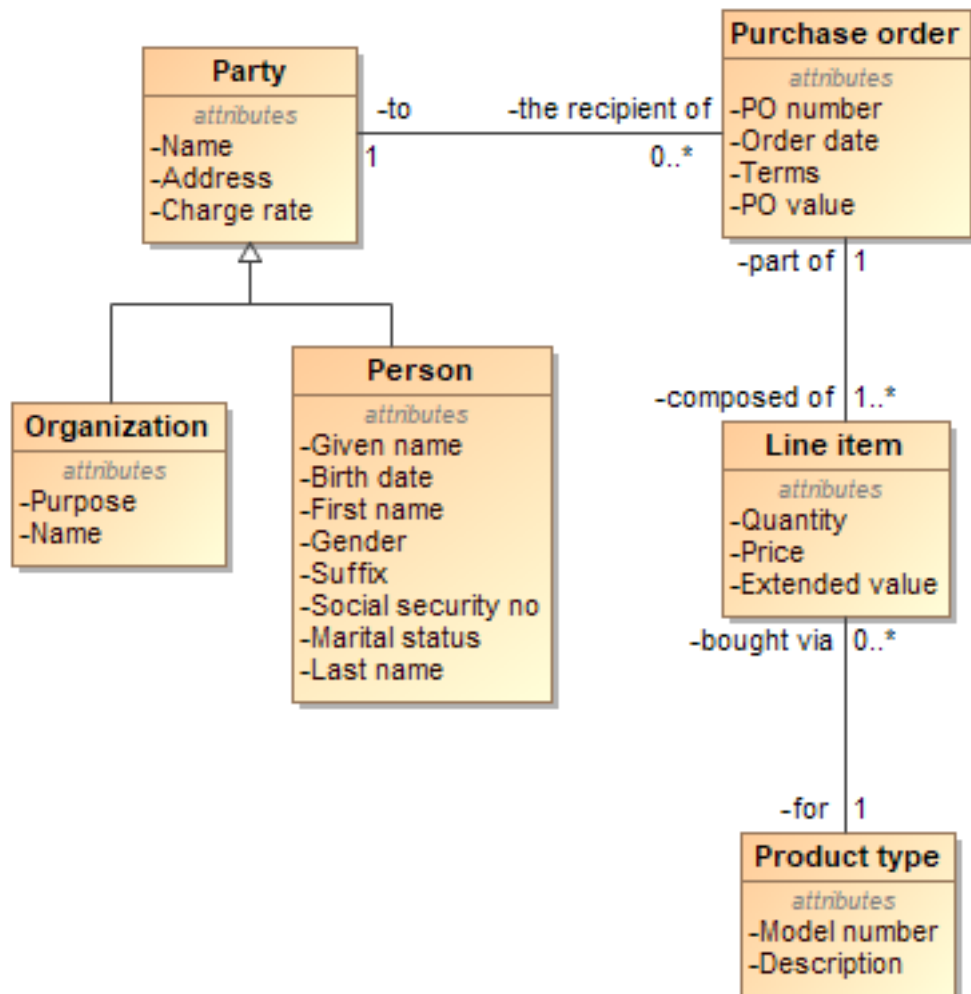


Figure 33. Purchase orders

5.34. Reporting relationship

Šablonas sukurtas pagal David C. Hay [13]

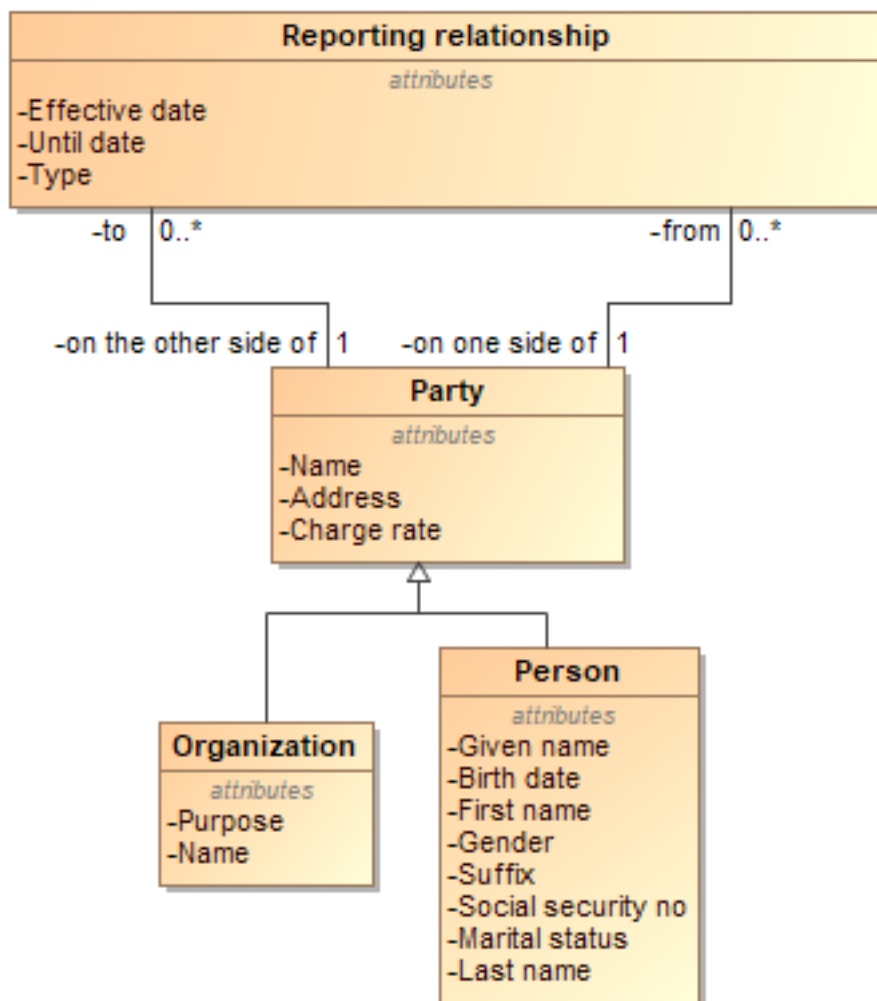


Figure 34. Reporting relationship

5.35. Roles and Assignments

Šablonas sukurtas pagal David C. Hay [13]

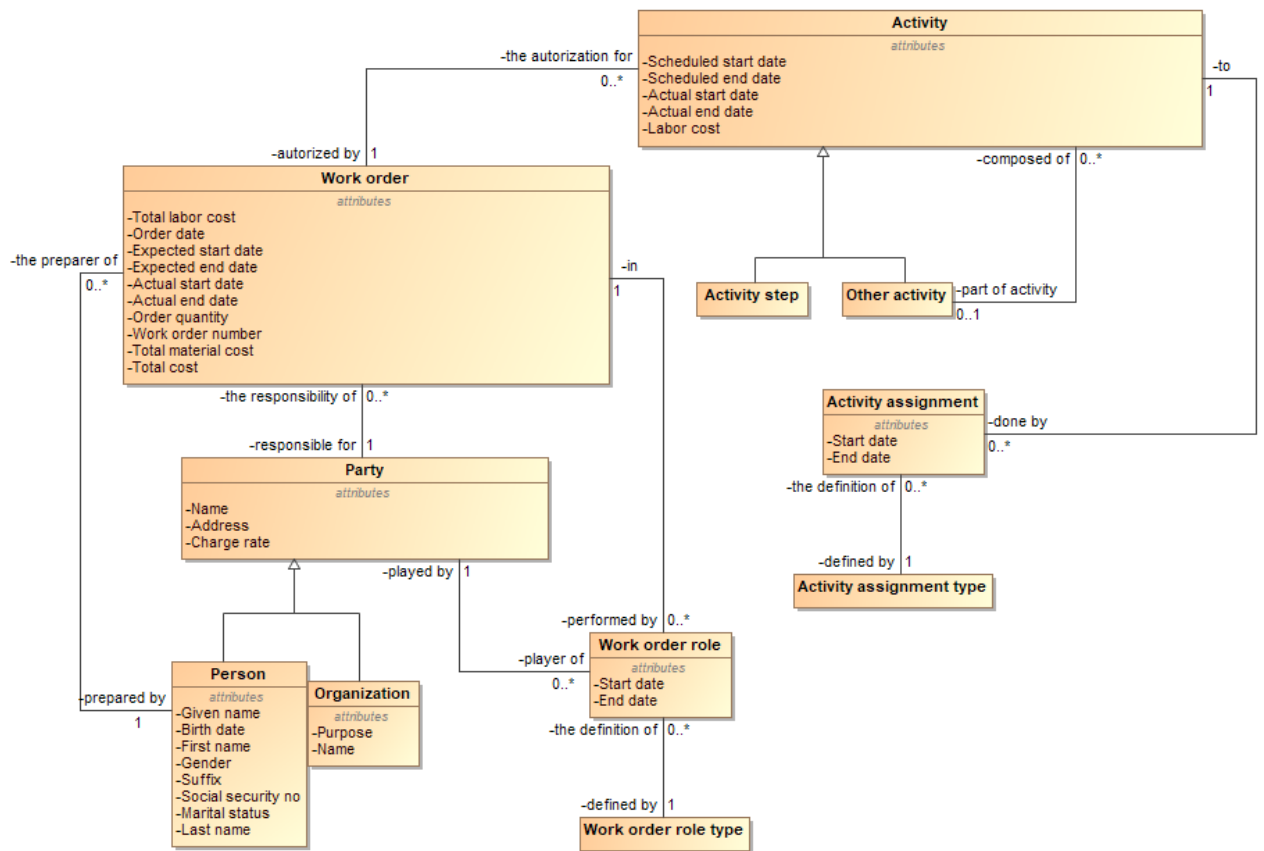


Figure 35. Roles and Assignments

5.36. Sales orders

Šablonas sukurtas pagal David C. Hay [13]

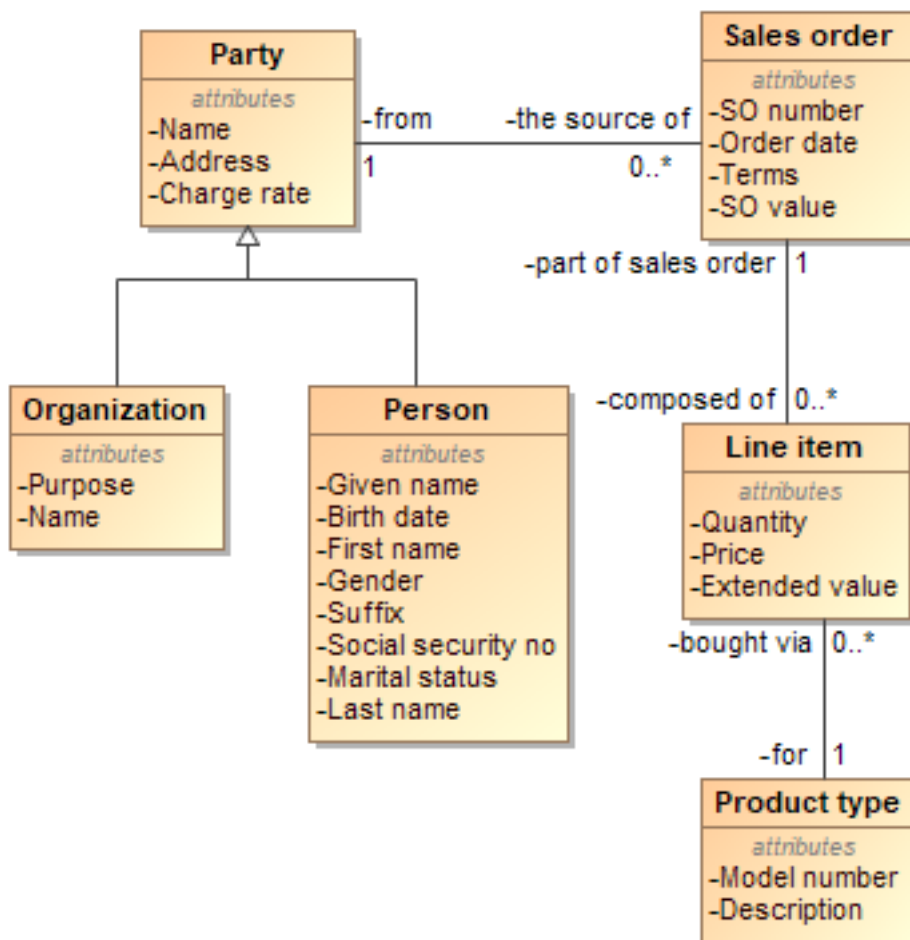


Figure 36. Sales orders

5.37. Site

Šablonas sukurtas pagal David C. Hay [13]

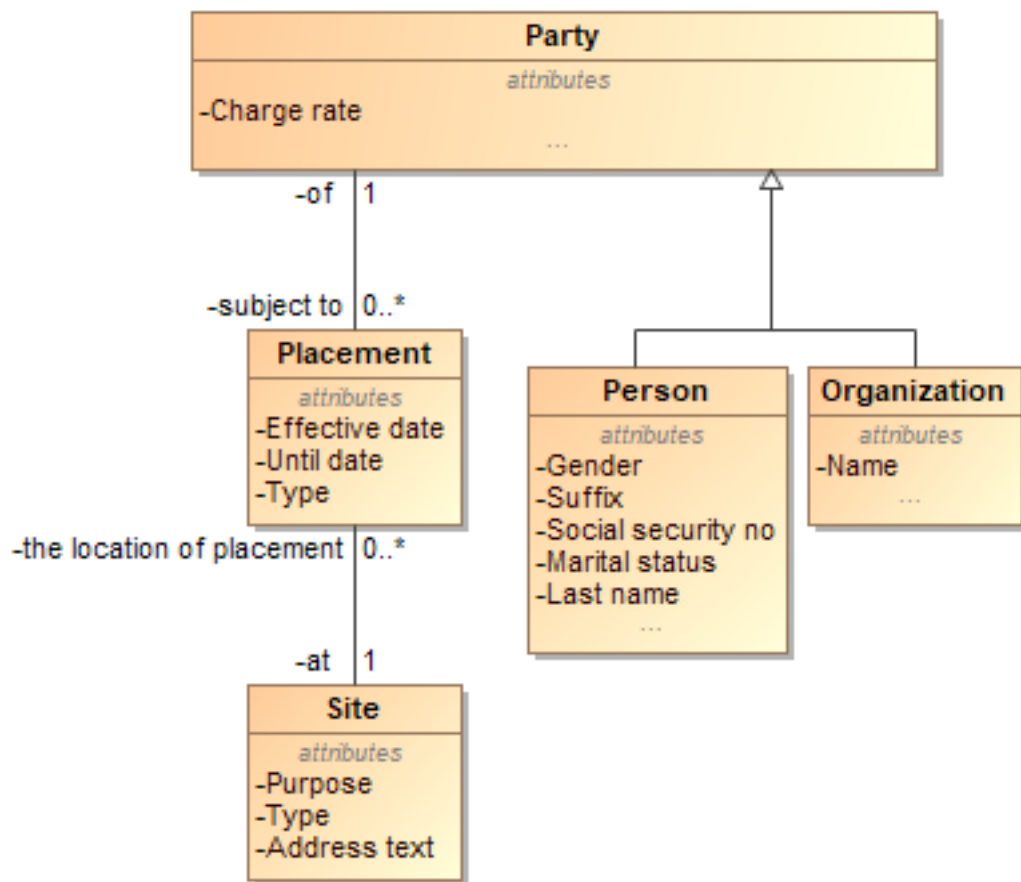


Figure 37. Site

5.38. Time sheets

Šablonas sukurtas pagal David C. Hay [13]

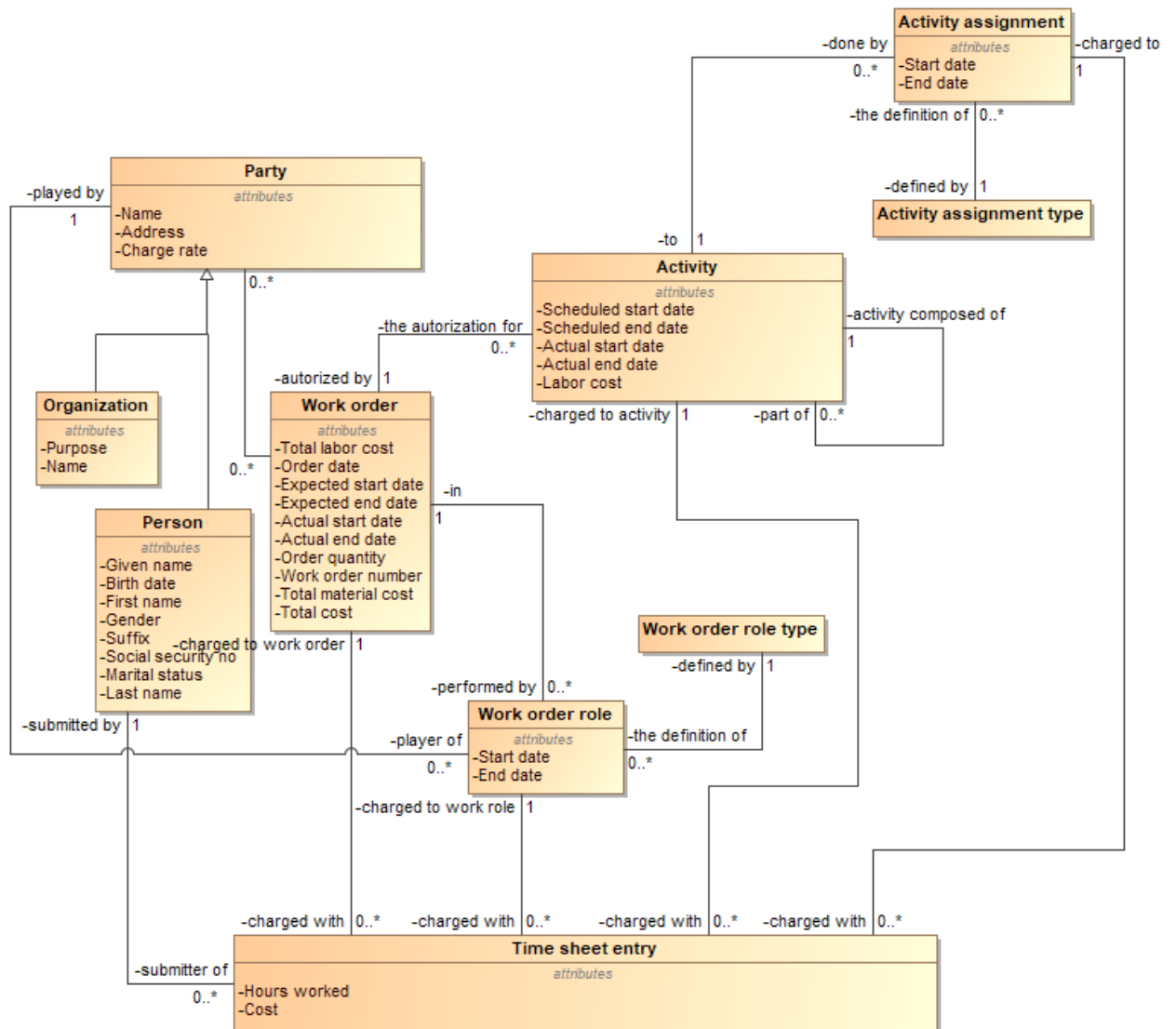


Figure 38. Time sheets

5.39. Work orders

Šablonas sukurtas pagal David C. Hay [13]

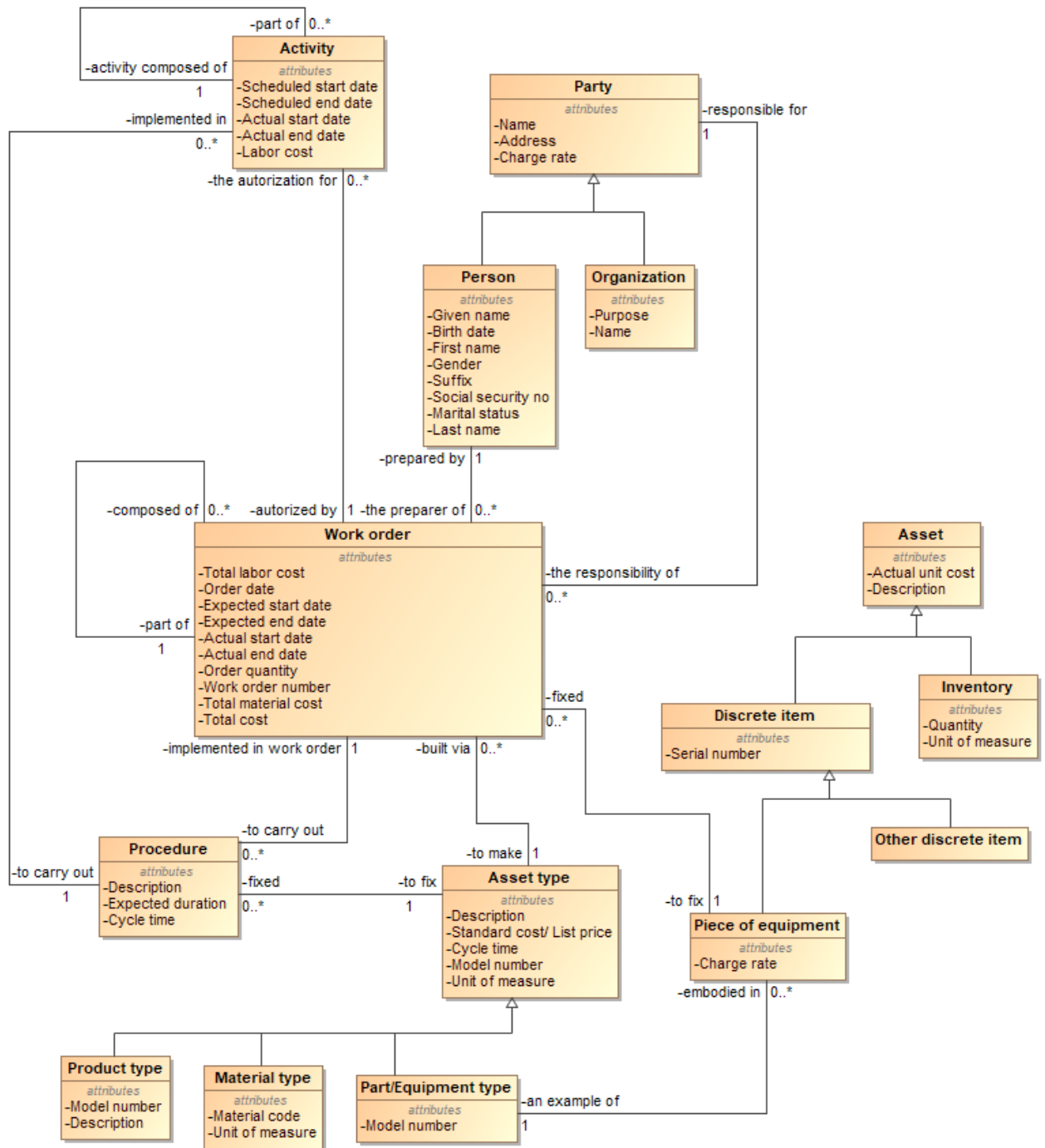


Figure 39. Work orders

5.40. Party contact mechanism

Party contact mechanism pagal Len Silverston [14]

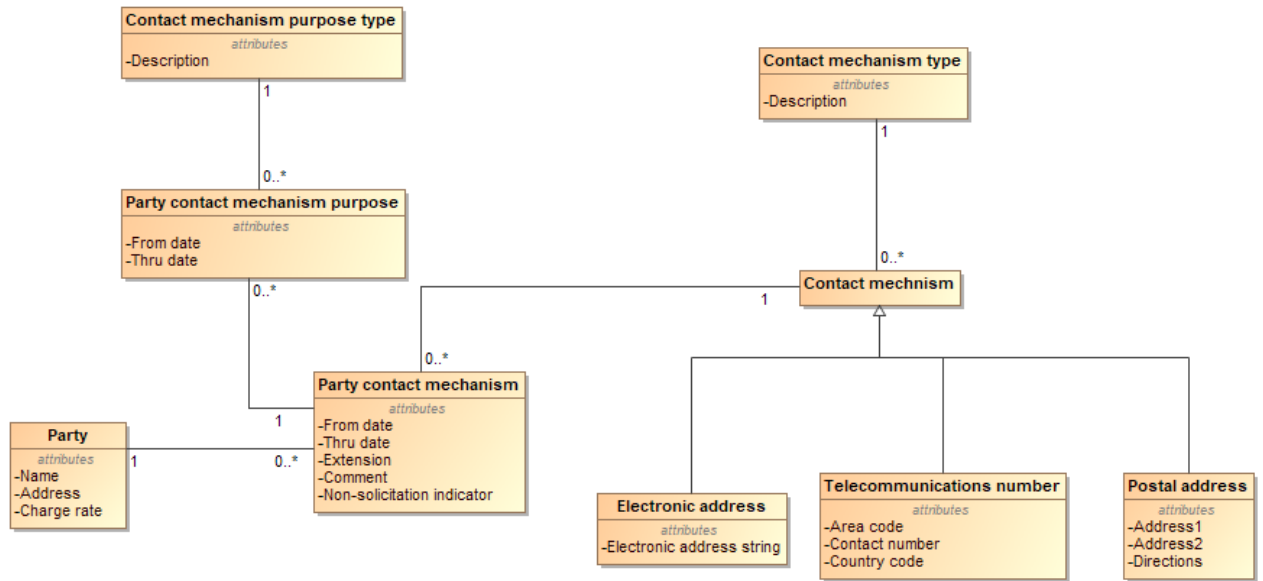


Figure 40. Party contact mechanism