

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Koncepcinio duomenų modeliavimo metodų tyrimas
Conceptual data modeling methods research

Bakalauro darbas

Atliko:	Aurelijus Statkus	(parašas)
Darbo vadovas:	dr. Elita Pakalnickienė	(parašas)
Darbo recenzentas:	dr. Audronė Lupeikienė	(parašas)

Santrauka

Šiame bakalauro darbe yra atliekamas koncepcinio duomenų modeliavimo metodų ORM 2, UML ir E-R tyrimas, nagrinėjami pagrindiniai konceptai, grafiškai atvaizduojami vientisumo reikalavimai. Atliekamas praktinis koncepcinio duomenų modeliavimo metodų išbandymas, po kurio suformuojamos praktinės rekomendacijos koncepcinio duomenų modeliavimo metodo pasirinkimui. Koncepcinių duomenų modelių kokybės vertinimui darbe naudojami dažniausiai literatūroje sutinkami vertinimo kriterijai: aiškumas, išraiškingumas, skirtingų grafinių konceptų kiekis, vientisumo reikalavimų kiekis. Šių kriterijų pagalba siekiama išsiaiškinti kiekvieno iš pasirinktų koncepcinio duomenų modeliavimo metodų privalumus ir trūkumus.

Raktiniai žodžiai: koncepcinis duomenų modeliavimas, ORM 2, UML, E-R, vientisumo reikalavimai, grafiniai konceptai.

Abstract

This Bachelor's thesis presents the analysis of the research on the conceptual data modeling techniques ORM 2, UML and E-R. The work presents the analysis of each technique, discusses the main concepts, displays the integrity constraints in a graphic form. It also provides the results of practical testing of the conceptual data modeling techniques, which lead to certain practical recommendations. The work focuses on the most widely discussed conceptual data models criteria for quality, such as clarity, expressibility, the number of different graphic concepts, the number of integrity constraints. These criteria are used to examine the advantages and disadvantages of each of the selected conceptual data modeling techniques.

Key words: conceptual data modeling, ORM 2, UML, E-R, integrity constraints, graphic concepts.

Turinys

Įvadas	6
1. Modeliavimo metodų analizė	8
1.1. Tyrimo kriterijai.....	8
1.2. Modeliavimo metodai	9
1.2.1. ORM 2 modeliavimo metodas	9
1.2.2. UML modeliavimo metodas.....	10
1.2.3. Esybių-ryšių modeliavimo metodas	11
1.3. Pagrindiniai konceptai	11
1.3.1. Esybės.....	11
1.3.2. Atributai.....	12
1.3.3. Ryšiai.....	13
1.4. Vientisumo reikalavimai	15
1.4.1. Pirminis raktas	15
1.4.2. Išorinis raktas	15
1.4.3. Privalomi ir neprivalomi atributai	16
1.4.4. Alternatyvus privalomumas	17
1.4.5. Vidinis unikalumas.....	19
1.4.6. Išorinis unikalumas.....	19
1.4.7. Vidinis dažnumas	20
1.4.8. Išorinis dažnumas	20
1.4.9. Išskyrimas.....	21
1.4.10. Išskyrimas – arba.....	22
1.4.11. Žiediniai apribojimai	22
1.4.12. Poaibis	25
1.4.13. Lygybė.....	26
1.4.14. Potipių apribojimai apibendrinimo ryšiui.....	26
1.4.15. Agregacija ir kompozicija	28
1.5. Modeliavimo metodų apibendrinimas	29

2.	Praktinis modeliavimo metodų išbandymas.....	32
3.	Praktinės rekomendacijos.....	42
	Išvados	43
	Rezultatai.....	43
	Šaltiniai	44
	Santrumpos.....	46
	Priedai	47
1	priedas. Kompaktinis diskas.....	47

Ivadas

Temos aktualumas

Šiuo metu informacinių sistemų kūrimas ir naudojimas sparčiai auga. Įmonės siekdamos padidinti savo darbų efektyvumą bando pasitelkti vis naujesnes kompiuterines technologijas bei kelia joms vis aukštesnius reikalavimus. Taigi informacinių technologijų kūrimas tampa vis sudėtingesnis. Norint supaprastinti bei pagreitinti informacinių technologijų kūrimo eigą būtina pasitelkti tinkamą duomenų modeliavimo metodą. Duomenų modelis turi didelę įtaką projektavimui, kūrimui taikomosios programos, nepriklausomai ar tai būtų duomenų saugykla, ERP, SOA, OLTP ar, bet kokia kita taikomoji programa. Tai turi ryšį su projekto įvykdymu laiku, neviršijant biudžeto, nes duomenų modeliavimas gali padėti identifikuoti klaidas, kai jos dar gali būti lengvai ištaisytos. Paprastai duomenų modeliavimui sunaudojama mažiau nei 10% projekto biudžeto, bet jis gali sumažinti 70% biudžeto, skirto programavimo darbams.

Duomenų modeliavimas prasideda nuo koncepcinio modelio sudarymo, kurio pagrindinis tikslas yra tiksliai ir išsamiai atvaizduoti duomenis vartotojams bei programuotojams lengvai suprantama schema. Koncepcinio modelio aiškumas rodo modeliavimo kalbos galimybę natūraliu būtu atvaizduoti realų pasaulį, jo objektus. Žmogus suvokia ir modeliuoja pasaulį jam suprantamais konceptais, kurių kiekvienas turi aiškų apibrėžimą, galimus egzempliorius bei grafinius arba tekstinius simbolius. Tačiau koncepcinio duomenų modeliavimo metu naudingesni tampa grafiniai simboliai, kadangi jais perteikiamas vaizdas yra lengviau suvokiamas vartotojų [PN08].

Pagrindinės problemos su kurioms susiduriama koncepcinio duomenų modeliavimo metu yra kalbų apribojimai [PN08] dėl kurių jos negali išpildyti visų dalykinės srities reikalavimų, bei vartotojo sugebėjimas pasinaudoti sukurtu modeliu. Taigi, kyla klausimas, kurį duomenų modeliavimo metodą pasirinkti, kuris padėtų surinkti reikalavimus ir užtikrinti saugomų duomenų kokybę bei būtų lengvai suprantamas?

Tyrimo sritis – koncepcinio modeliavimo metodai, juose naudojami vientisumo reikalavimai.

Darbo tikslas – atlikti koncepcinio duomenų modeliavimo metodų analizę, siekiant palyginti juose naudojamus modelio elementus, jų grafines galimybes bei juose nagrinėjamų vientisumo reikalavimų aibes ir pateikti rekomendacijas koncepcinio duomenų modeliavimo metodo pasirinkimui.

Darbo uždaviniai:

1. Išanalizuoti ORM 2, UML ir E-R koncepcinio duomenų modeliavimo metodus pagal apsibrėžtus kriterijus;

2. Apžvelgti kiekvieno koncepcinio duomenų modeliavimo metodo naudojamus elementus, jų grafinio vaizdavimo galimybes;
3. Palyginti pasirinktus koncepcinio modeliavimo metodus juos praktiškai išbandant pasirinktai dalykinei sričiai;
4. Pateikti rekomendacijas siekiant pasirinkti tinkamiausią koncepcinio duomenų modeliavimo metodą.

Darbo atlikimo eiga:

1. Surinkta ir išnagrinėta su darbo tema susijusi literatūra;
2. Pasirinkti tyrimo kriterijus, kurie leistų įvertinti nagrinėjamus koncepcinio duomenų modeliavimo metodus;
3. Pasinaudojant literatūra išnagrinėti bei tarpusavyje palyginti ORM 2, UML ir E-R modeliavimo metodai, juose naudojami pagrindiniai konceptai bei jiems taikomi vientisumo reikalavimai;
4. Praktiškai pritaikyti darbe nagrinėti modeliavimo metodai konkrečiai dalykinei sričiai;
5. Remiantis modeliavimo metodų tyrimo ir jų praktinio išbandymo rezultatais, pateiktos rekomendacijos koncepcinio duomenų modeliavimo metodo pasirinkimui;
6. Aptariamos darbo išvados ir rezultatai;

Siekiami rezultatai

Išanalizavus literatūrą, pateikti susistemintą UML, E-R bei ORM 2 koncepcinio duomenų modeliavimo metodų palyginamo medžiagą. Pasinaudojant surinkta medžiaga praktiškai išbandyti kiekvieną modeliavimo metodą. Taip pat pateikti rekomendacijas siekiant pasirinkti vieną iš modeliavimo metodų.

1. Modeliavimo metodų analizė

1.1. Tyrimo kriterijai

Koncepcinis duomenų modelis – dalykinės srities konceptų ir jų tarpusavio sąryšių rinkinys, skirtas duomenų atvaizdavimui. Koncepciniame duomenų modeliavime kuriama duomenų schema atvaizduojama paprastam vartotojui suprantamomis sąvokomis bei terminais, taip paslepiant loginį bei fizinį duomenų modelius skirtus atvaizduoti programines ir duomenų bazės struktūras. Šiame darbe taip pat labai svarbi semantinė koncepcinių modelių kokybė, kurios svarbiausiu faktoriumi laikomas koncepcinio modelio atitikimas (tinkamumas) dalykinės srities duomenų objektams. Koncepcinis modelis yra tinkamas, jei jis tenkina dalykinės srities duomenų rinkinius ir leidžia atvaizduoti visas leistinas dalykinės srities būsenas bei neleidžia atvaizduoti neleistinių.

Visuotinai pripažįstamo koncepcinių duomenų modelių kokybės kriterijų rinkinio nėra, tai vis dar yra tyrimų objektas [PN08]. Šiame darbe pasirinkti tyrimo kriterijai, tai dažniausiai literatūroje sutinkami koncepcinių modelių kokybės vertinimo kriterijai. Taigi, didžiausias dėmesys yra skiriamas šiems kriterijams: aiškumas, išraiškingumas, skirtingų grafinių konceptų bei vientisumo reikalavimų kiekis. Tai T. Halpin, T. Morgan, B. Paradausko, L. Nemuraitės, M. Blaha, D. Hay bei kitų autorių literatūroje dažniausiai sutinkami kokybės vertinimo kriterijai. Būtent šių kriterijų pagalba bus siekiama išsiaiškinti kiekvieno iš pasirinktų duomenų modeliavimo metodų taikymo privalumus ir trūkumus.

Detalus kriterijų aprašymas:

- Aiškumas – kriterijus, kurio pagalba įvertinamas modeliavimo kalbos suvokimas, vartotojo gebėjimas pasinaudoti sukurtu modeliu. Vertinamas duomenų modelis turėtų būti nedviprasmiškas, lengvai skaitomas bei perprantamas, intuityvus.
- Išraiškingumas – šio kriterijaus pagalba bus siekiama išsiaiškinti metodo ekspresyvumą bei kaip tiksliai juo yra perteikiama modeliuojamos srities mintis. Pasirinktas modeliavimo metodas turėtų atvaizduoti duomenis kiek galima išsamiau ir tiksliau.
- Skirtingų grafinių konceptų kiekis – skirtingų grafinių elementų kiekis naudojamas kiekviename iš pasirinktų modeliavimo metodų diagramose.
- Vientisumo reikalavimų kiekis – šio kriterijaus pagalba įvertinsime, kuris modeliavimo metodas pasižymi plačiausia vientisumo reikalavimų aibe, kas užtikrina duomenų tikslumą bei nuoseklumą visą jų gyvavimo ciklą, tai vienas svarbiausių aspektų projektavimo, įgyvendinimo bei sistemos naudojimo metu.

1.2. Modeliavimo metodai

1.2.1. ORM 2 modeliavimo metodas

Objekto vaidmenų modeliavimas (trumpiau ORM) – tai semantinis duomenų modeliavimo būdas, modeliuojamą sritį išreiškiantis per objektus ir jų atliekamus vaidmenis.

Vienas pagrindinių ORM plėtotojų, tai daktaras Terry Halpin, kuris šį modeliavimo metodą pradėjo plėtoti dar septintame dešimtmetyje. ORM pirmtakas, tai natūralios kalbos informacijos analizės metodas – NIAM.

Objekto vaidmenų modeliavimo būdas pasižymi :

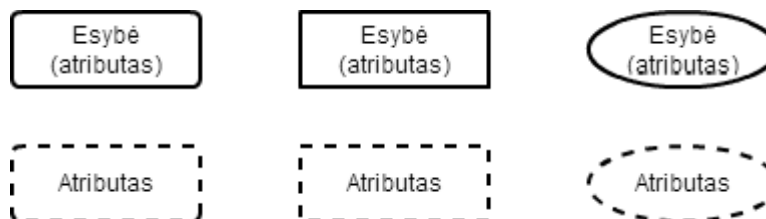
- Natūralios kalbos elementais.
- Lengvai suprantamomis diagramomis.
- Galimybe transformuoti šiuo metodu sukurtus duomenų modelius į kitos kalbos duomenų modelius.
- Išsamia grafine notacija.
- Plačiu vientisumo reikalavimų pasirinkimu.

Visa tai supaprastina ORM duomenų modeliavimo procesą, bei suteikia galimybę aiškiai atvaizduoti verslo procesus.

Darbe nagrinėjama šiuo metu naujausia ORM 2 grafinė notacija, kuri pasižymi [MTH05]:

- Kompaktiškesniu ORM 2 modelių vaizdavimu;
- Pagerinta internacionalizacija – pašalinti tik anglų kalbai būdingi simboliai;
- Pagerintas tekstinių anotacijų palaikymas;
- Supaprastintomis modeliavimo taisyklėmis – dėl ko paprastėja grafiniai redaktoriai;
- Praplėstu funkcionalumu;

ORM 2 modeliavimo metodas be įprastinės notacijos taip pat siūlo 2 alternatyvas (1 pav.). T. Halpin atlikto tyrimo, kurio tikslas buvo išsiaiškinti priimtinausią notaciją, metu buvo apklausta 18 duomenų modeliavimo ekspertų: „Iš 18 apklaustų ekspertų, 12 teikė pirmenybę stačiakampiui apvalintais kampais, 5 teikė pirmenybę elipsės formai ir 1 teikė pirmenybę paprastam stačiakampiui“ [Hal05]. Darbe naudojamas įrankis Visual Studio 2013 su NORMA įskiepiu palaiko stačiakampio apvalintais kampais notaciją.



1 pav. Alternatyvūs esybų ir atributų žymėjimas ORM 2 metodu

1.2.2. UML modeliavimo metodas

UML arba kitaip vieninga modeliavimo kalba (angl. Unified Modeling Language) – tai modeliavimo ir specifikacijų kūrimo kalba, skirta specifiškai, atvaizduoti ir konstruoti objektiškai orientuotų programų dokumentus [PN08].

Ši modeliavimo kalba buvo sukurta 1997 metais remiantis 1991 metais James Rumbaugh sukurta objektinio modeliavimo technika (angl. Object-modeling technique), tais pačiais metais Grady Booch publikuotu „Booch“ metodu ir Ivar Jacobson 1992 metais sukurta objektiškai orientuotos programinės įrangos modeliavimo kalba ir metodologija (angl. Object-oriented software engineering).

Šiuo metu UML yra labiausiai paplitęs programinės įrangos specifikavimo standartas išpopuliarėjęs dar 1997 metais pasirodžius 1.1 UML versijai. 2005 metais tarptautinė standartizacijos organizacija (trumpiau - ISO) patvirtino UML 1.4.2. kaip vieną iš ISO standartų, kurios tikslas yra užtikrinti programinės įrangos architektams, inžinieriams ir kūrėjams įrankius skirtus programinės įrangos analizei, dizainui ir implementacijai, taip pat verslo ir panašių procesų modeliavimui.

Šiuo metu UML turi daug ir įvairių tipų diagramų, kurios yra suskirstytos į 2 pagrindines kategorijas: struktūrinės (angl. Structure diagram) bei elgesnė aprašančios (angl. Behavior diagram). Struktūrinėmis diagramomis pasitelkiant objektus, atributus, operacijas ir ryšius atvaizduojama statinė sistemos struktūra, hierarchija. Kai elgesnė aprašančiomis diagramomis atvaizduojamas modeliuojamos sistemos elgesys, pagrindinis programinės įrangos funkcionalumas.

Kalbant apie koncepcinį duomenų modeliavimą pasitelkiant UML modeliavimo metodą, pagrindinė analizuojamos struktūrinės klasių bei objektų diagramos [Hal98a]:

- Klasių diagrama (angl. Class diagram) atvaizduoja sistemos struktūrą, jos klases ir jų atributus taip pat ryšius tarp klasių ir operacijas.
- Objektų diagrama (angl. Object diagram) atvaizduoja pilną arba dalinį modeliuojamos sistemos vaizdą konkrečiu momentu. Šio tipo diagramos yra konkretesnės nei klasių diagramos.

Kadangi UML modeliavimo metodas nėra pilnai pritaikytas duomenų modeliavimui ir standartinis Magic Draw 18.3 įrankis neužtikrina galimybės kurti koncepcinių duomenų modelių darbe naudojamas Cameo Data Modeler 18.3 įskiepis. Jis papildo Magic Draw įrankį ir sudaro galimybę pažymėti pagrindinius duomenų bazės palaikomus vientisumo reikalavimus: pirminį bei išorinį raktus, koncepcinio duomenų modeliavimo metu.

1.2.3. Esybių-ryšių modeliavimo metodas

Esybių – ryšių modelis (trumpiau E-R) – tai duomenų modeliavimo būdas skirtas verslo procesams bei reikalavimams apibūdinti.

E-R modeliavimą išvystė Peter Chen dar 1976 metais. Šis modeliavimo metodas yra daugelio kitų sistemų analizės ir dizaino metodologijų tokių kaip UML pamatas. Tai vienas patogiausių bei paprasčiausių būdų apibūdinti bei aprašyti verslo procesus. Pagrindiniai šio modeliavimo metodo komponentai, tai esybės bei ryšiai tarp jų. Ryšiai yra naudojami priklausomybių tarp esybių apibūdinimui. Esybės (subjektai) gali savyje turėti įvairių savybių (atributų), kuriais jos yra apibūdinamos. Taigi grafiškai apibūdintos esybės, jų savybės bei ryšiai kartu sudaro esybių – ryšių diagramas.

Koncepcinio duomenų modeliavimo atveju E-R diagramomis yra bendrai nusakoma modeliuojama dalykinė sritis, apibrėžiamos pagrindinės duomenų esybės, aprašoma duomenų struktūra.

Šiame darbe bus nagrinėjama „Crow’s foot“ E-R modelių žymėjimo sistema. Tai vienas iš informatyviausių bei plačiausiai naudojamų E-R modeliavimo metodo žymėjimo būdų. Be to tai taip yra vienas lengviausių būdų modeliuoti kompleksinius duomenis taip, kad jie būtų suprantami tiek verslo atstovams, tiek sistemos projektuotojams. Taigi E-R diagramos yra gera komunikacijos priemonė tarp sistemos projektuotojų bei jos vartotojų siekiant išaiškinti jų reikalavimus.

1.3. Pagrindiniai konceptai

1.3.1. Esybės

Esybė (angl. Entity) - tai lengvai atpažįstamas ir suvokiamas tikro pasaulio objektas (sąvoka, konceptas), atvaizduotas modeliuojamame kontekste [Bar05] (2 pav.).



2 pav. Esybių vaizdavimas

ORM 2 – esybė žymima stačiakampiu apvalintais kampais. Šiame modeliavimo metode esybės viduje žymimas tik vienas – pirminio rakto atributas, likę atributai žymimi atskirai.

UML – esybė atitinka UML klasių diagramos, klasės elementą ir žymima stačiakampiu viršuje nurodant esybės pavadinimą, o žemiau išvardijami visi jai priklausantys atributai.

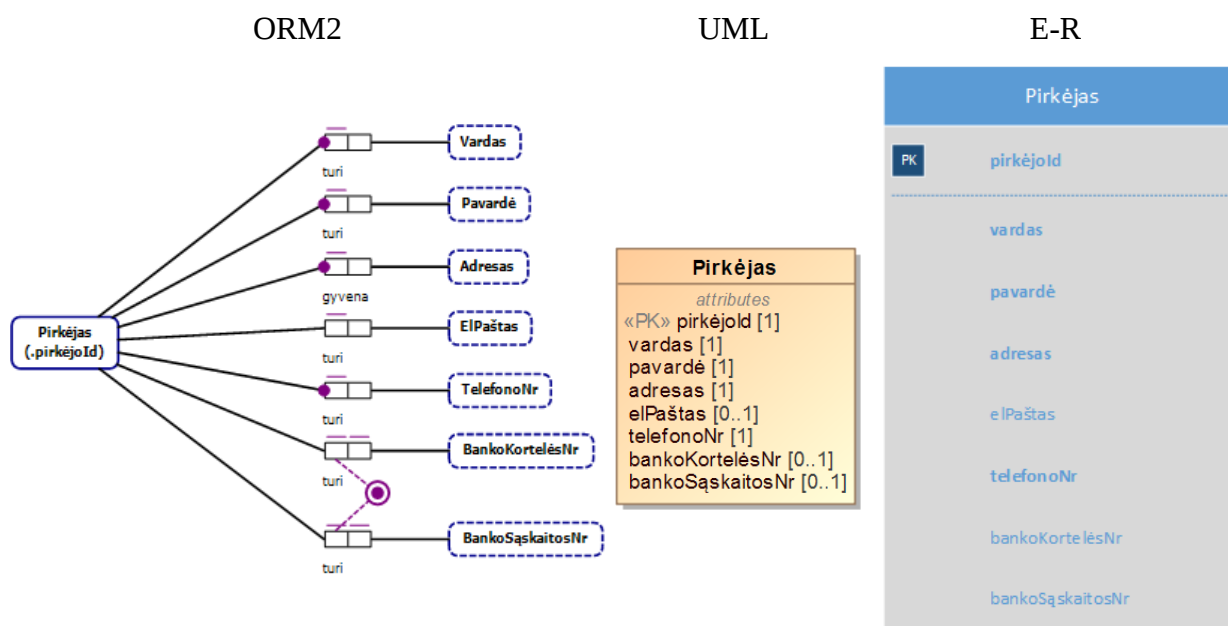
E-R – esybė žymima stačiakampiu viršuje nurodant esybės pavadinimą, o po juo išvardijami visi jai priklausantys atributai.

Kiekvienai esybei yra suteikiamas unikalus pavadinimas. Pavyzdžiui modeliuojama elektroninės parduotuvės informacinė sistema gali apimti tokias esybes: „Prekė“, „Katalogas“, „Pirkėjas“. Visos šios esybės taip pat turi atributus, kurie joms suteikia identitetą.

Kadangi ORM 2 ir E-R modeliai yra specialiai pritaikyti duomenų modeliavimui, o UML ne, anot ORM plėtotėjo Terry Halpin [Hal98a] bei Michael Blaha [Bla13] šiam tikslui UML modeliuose geriausia yra pasirinkti klasių diagramas ir esybes žymėti klasėmis. Tačiau reikia pabrėžti, kad UML modeliavimo kalba yra labiau orientuota į objektiškai orientuotų programų modeliavimą ir UML klasės standartiškai savyje be atributų dar turi ir operacijas, ko nagrinėjamu atveju neprireiks - klasių operacijos bus ignoruojamos. Taigi esybė apibrėžia grupę objektų pasižyminčių panašiomis savybėmis (atributais) bei ryšiais su kitų esybių objektais [Bla13].

1.3.2. Atributai

Atributas (angl. Attribute) – tai esybę apibrėžianti savybė, charakteristika. Būtina pabrėžti, kad atributai yra įvardintos savybės, kuriomis yra apibūdinami visi tam tikros esybės objektai. Taigi trumpiau - atributas yra „vieta“, kurioje saugomos esybės objekto reikšmės [Bla13].



3 pav. Atributų vaizdavimas

ORM 2 – atributas žymimas stačiakampiu apvalintais kampais, apibrėžtu punktyrine linija. Kiekvienas atributas su esybe siejamas sąryšiu, dėl ko turime galimybę nurodyti daugiau įvairių atributo apribojimų lyginant su kitais modeliavimo metodais. Dėl šios priežasties šis modeliavimo metodas yra daug išraiškingesnis lyginant su kitais. Tačiau toks žymėjimas taip pat reikalauja gana daug vietos, dėl ko ORM 2 modeliavimo metodas yra tampa ganėtinai painus.

UML – atributai žymimi sąrašu esybės viduje. Toks žymėjimas yra daug kompaktiškesnis lyginant su ORM 2 atributais, nors ir ne toks ekspresyvus.

E-R – atributai kaip ir UML modelyje žymimi sąrašų esybės viduje, dėl ko taip pat prarandame dalį apribojimų, tačiau kuriamas modelis yra kompaktiškesnis.

Atributai visais atvejais egzistuoja visoms apibrėžtomis esybėms. Taip pat visi vienos esybės objektai turi tuos pačius atributus. Taigi kiekviena esybė turi ją apibrėžiančių atributų aibę, pavyzdžiui esybės PIRKĖJAS atributai gali būti šie: identifikacijos numeris (ID), vardas, pavardė, šalis, miestas, adresas, telefono numeris, el. pašto adresas, banko kortelės numeris, banko sąskaitos numeris ir kita pirkėjo informacija.

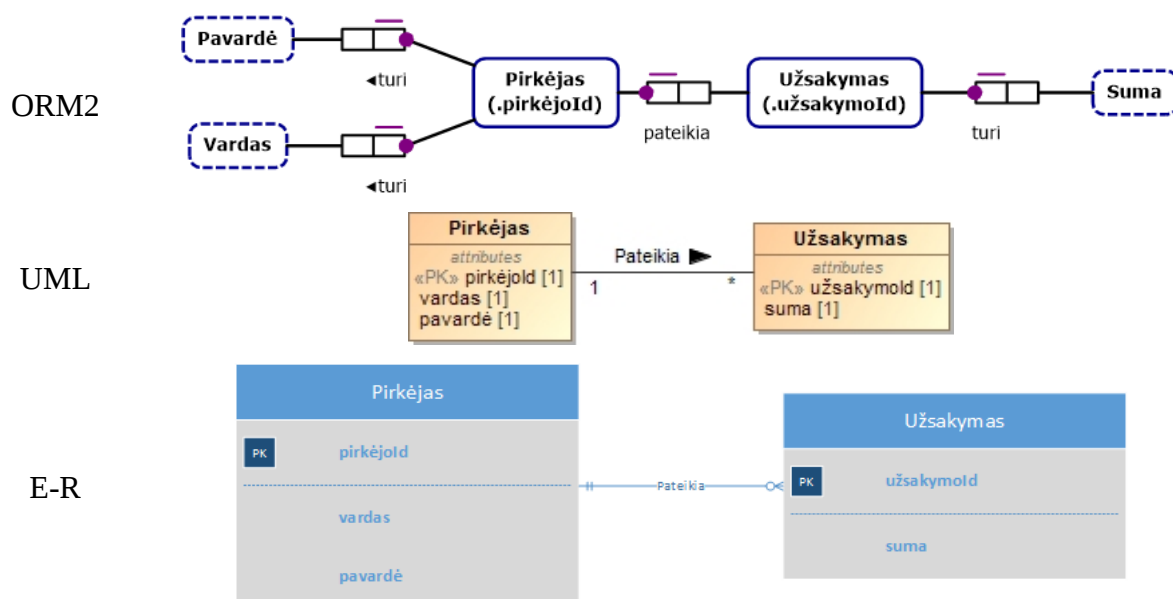
Pateiktame atributų vaizdavimo pavyzdyje (3 pav.) matome, kad visais trimis atvejais esybė PIRKĖJAS turi šiuos atributus: pirkėjo identifikacijos numerį (pirkėjoId), vardą, pavardę, šalį, miestą bei adresą. Iš pateiktų duomenų modelių galime teigti, jog atributai yra:

- Privalomi: vardas, pavardė, adresas, telefono nr. – atvaizduojami visuose 3 modeliuose;
- Neprivalomi: el. paštas, banko kortelės nr., banko sąskaitos nr. – atvaizduojami visuose 3 modeliuose;
- Alternatyvos: banko kortelės nr. arba banko sąskaitos nr. – turi specifinį žymėjimą tik ORM2 modelyje;

1.3.3. Ryšiai

Ryšys (angl. Relationship) – tai santykis tarp dviejų esybių [PN08] (arba esybės ir atributo ORM 2 modelyje).

Žemiau pateiktame pavyzdyje (4 pav.) matome tipinį ryšių tarp esybių (ORM 2 modelyje ir atributų) atvaizdavimą. Pateiktame pavyzdyje matome 2 esybes: „Pirkėjas“, „Užsakymas“. Kiekvienas sąryšis turi pavadinimą, kuris privalo atspindėti ryšio esmę. Pavyzdyje matomas ryšys gali būti skaitomas: „Pirkėjas pateikia užsakymą“.



4 pav. Sąryšio atvaizdavimas

ORM 2 – modeliuose ryšiai atvaizduojami pasinaudojant:

- Rolėmis – vaizduojamos stačiakampio skiltimis sujungtomis su jų esybėmis;
- Predikatais – tai tariniai arba sakiniai apibūdinantys esybės arba atributo atliekamą rolę.

Toks mišrus grafinis bei tekstinis ryšio vaizdavimas leidžia lengviau apibūdinti modelį paprastomis lengvai suvokiamomis sąvokomis, sakiniais [Hal98b]. Taip pat ORM 2 modelis išsiskiria tuo, kad ryšiai taip pat įgalinti tarp esybės ir atributo, kas leidžia nusakyti daug tikslesnę atributo būseną esybėje.

UML – modeliuose ryšiai atvaizduojami pasinaudojant:

- Ryšio pavadinimą – kiekvienas ryšys gali turėti (tačiau neprivalo) daugiausia 1 pavadinimą, kuris žymimas ant, arba virš ryšio linijos;
- Esybių atliekamos rolės – žymimos ryšio galuose ir nurodo esybės atliekamą rolę sąryšyje. Tačiau rolių žymėjimas nėra privalomas;

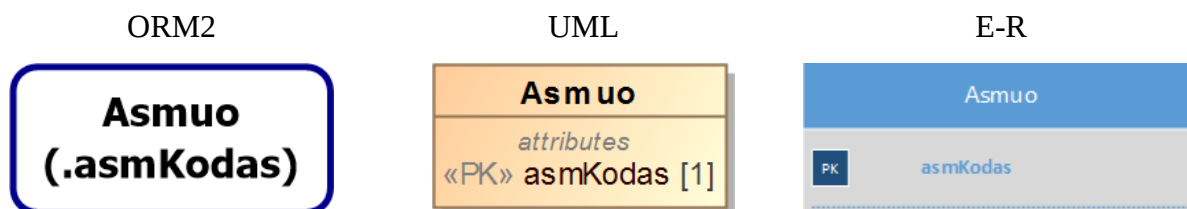
Tokio žymėjimo pagalba, kai nurodomas ryšio pavadinimas, bei esybių atliekamos rolės, supaprastina verbalizaciją UML diagramose;

E-R – modeliuose ryšiai atvaizduojami pasinaudojant ryšio pavadinimu – pavadinimas žymimas ant ryšio linijos. Taigi E-R modelyje ryšiai tarp esybių žymimi paprasčiausiai – linijomis, kurios jungia konkrečias esybes, turinčiomis pavadinimus, kuriais nusako ryšio esmę.

1.4. Vientisumo reikalavimai

1.4.1. Pirminis raktas

Pirminis raktas (angl. Primary key) – unikalus identifikatorius, kurio pagalba vienareikšmiškai nusakomi visi esybės objektai. Tai gali būti paprastas atributas, kuris garantuotai bus unikalus ir kurį turės visi esybės objektai pavyzdžiui: asmens kodas, kuris yra unikalus ir kurį turi kiekvienas žmogus (5 pav.) [Bar05].



5 pav. Pirminio rakto vaizdavimas

ORM 2 – modelyje, pirminis raktas nurodomas esybės viduje po jos pavadinimu, taip jį aiškiai išskiriant iš kitų esybės atributų;

UML – modelyje, pirminis raktas nurodomas prieš atributo pavadinimą, pažymint sutartiniu žymėjimu (stereotipu) <<PK>>. Standartiškai pirminio rakto atributas yra pirmasis esybės atributų sąraše, tačiau tai nėra būtina sąlyga;

E-R – modelyje, pirminis raktas taip pat žymimas raidėmis PK prieš atributo pavadinimą, tačiau kitaip nei UML modelyje turi grafinį skirtuką taškuotą liniją, kuri jį atskiria nuo kitų atributų;

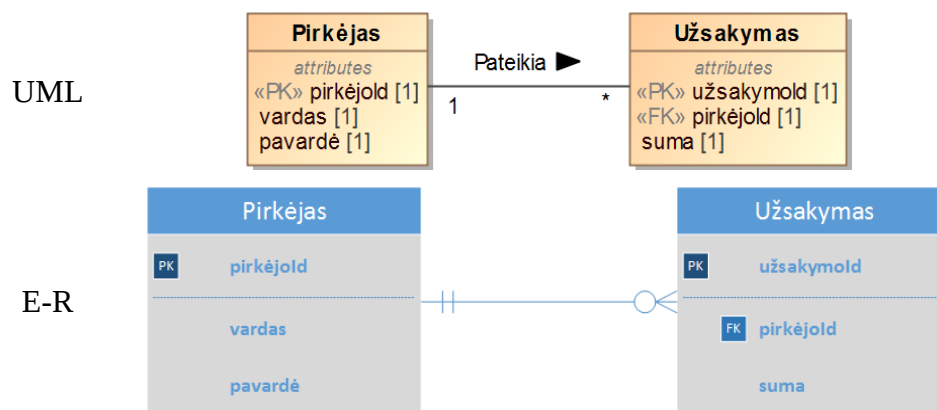
1.4.2. Išorinis raktas

Išorinis raktas (angl. Foreign key) – kitaip vadinamas migruojantis atributas. „Tai esybės atributų rinkinys, kuris kitoje esybėje yra pirminis raktas“ [Bar05]. Išorinis raktas yra naudojamas loginiams ryšiams tarp esybės objektų apibrėžti. Pavyzdžiui turime 2 esybes „Pirkėjas“, kurios pirminis raktas yra pirkėjoID, ir „Užsakymas“. Norėdami apjungti konkretų užsakymą su konkrečiu pirkėju, „Užsakymo“ esybėje privalome turėti išorinį raktą pirkėjoID, kuris nurodys konkretų „Pirkėjas“ esybės objektą. Taip galėsime susieti užsakymą su jo pirkėju.

ORM 2 – neturi grafinio išorinio rakto vaizdavimo galimybes.

UML – modelyje, išorinis raktas nurodomas prieš atributo pavadinimą, pažymint raktiniu simboliu (stereotipu) <<FK>>. Standartiškai išorinio rakto atributas yra sekantis po pirminio rakto esybės atributų sąraše, tačiau tai nėra būtina sąlyga.

E-R – modelyje, išorinis raktas taip pat vaizduojamas pažymėjimu FK prieš atributo pavadinimą ir pagal nutylėjimą žymimas po pirminio rakto atributu, tačiau tai nėra būtina sąlyga.



6 pav. Išorinio rakto vaizdavimas

1.4.3. Privalomi ir neprivalomi atributai

Privalomas atributas (angl. Mandatory attribute) – tai esybės savybė, kuri turi būti nurodyta visiems šios esybės objektams.

Neprivalomas arba pasirenkamas atributas (angl. Optional attribute) – tai esybės savybė, kuri neprivalo būti nurodyta visiems šios esybės objektams.

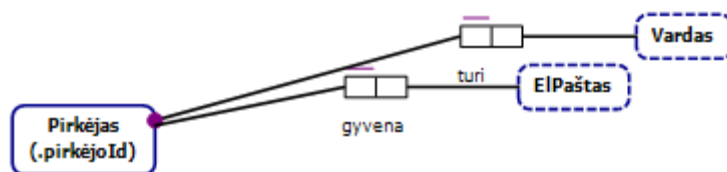
Pasinaudojant 3 pav. pateiktu pavyzdžiu galima teigti, kad kiekvienas el. parduotuvės pirkėjas (esybė - PIRKĖJAS) turi sekančias privalomus ir neprivalomus atributus:

Lentelė 1. Privalomi ir neprivalomi atributai iš 3 paveikslo

Privalo turėti	Neprivalo turėti
Vardas	El. Pašto
Pavardė	BankoKortelėsNr (UML, E-R modeliuose)
TelefonoNr	BankoSąskaitosNr (UML, E-R modeliuose)
Adresas	

ORM 2 – duomenų modelis išsiskiria savo esybių ir atributų žymėjimu, kadangi tarp esybės ir atributo yra nurodomas ryšys – tai esminis skirtumas lyginant šį duomenų modeliavimo metodą su UML ar E-R metodais. ORM 2 duomenų modelyje privalomi atributai žymimi taškais toje vietoje, kurioje atributas jungiasi su esybė arba toje vietoje, kur esybė jungiasi su atliekama role. Kitu atveju atributas – neprivalomas.

Taip pat galimos dviprasmybės, kai esybė jungiasi su daugeliu rolių, bei tampa neįmanoma nustatyti, kuri rolė yra privaloma. Kaip matome 7 paveiksle negalime nustatyti kuris atributas: vardas ar elPaštas yra privalomas. Tokiu atveju atributo privalomumas žymimas šalia jo atliekamos rolės.



7 pav. Dviprasmiškas privalomo atributo žymėjimas ORM 2 modelyje

Taigi kaip matome 3 paveiksle esybė PIRKĖJAS jungiasi su visais privalomais atributais ir šalia esybės atliekamų rolių yra nurodomi tamsūs taškai, kurie nurodo, kad atributai: vardas, pavardė, telefono nr. ir adresas yra privalomi.

UML – modelyje anot Terry Halpin visi atributai pagal nutylėjimą yra privalomi [Hal98a] tačiau David Hay savo publikuotame straipsnyje „UML as a Data Modeling Notation“ [Hay08] bei Micheal Blaha knygoje „UML database modeling“ siūloma atvaizduoti atributo privalomumo sąlygą nurodant jos kardinalumą (angl. Multiplicity). Darbe naudojamas MagicDraw įrankis įgalina šią funkciją ir 3 paveiksle pateiktame UML modelyje, šalia atributų laužtiniuose skliaustuose yra nurodoma, kokį vaidmenį esybėje atributas privalo atlikti. Dažniausiai pasitaikančios UML atributų sąlygos remiantis Micheal Blaha yra privalomi bei neprivalomi atributai, kurie žymimi kardinalumais [Bla13]:

- [1] – atributas yra privalomas ir gali turėti tik 1 nurodytą reikšmę;
- [0..1] – atributas gali, tačiau neprivalo, būti nurodytas ir gali turėti tik 1 reikšmę;
- [0..*] – atributas gali, tačiau neprivalo, turėti nurodytų reikšmių;
- [1..*] – atributas yra privalomas ir gali turėti daugiau nei 1 nurodytą reikšmę (reliacinės duomenų bazės neleidžia atributui turėti daugiau nei 1 reikšmę);

Taigi norėdami pažymėti atributo privalomumo sąlygą UML modelyje pasinaudojame atributų kardinalumo apribojimu (angl. Attribute multiplicity).

E-R – modelyje atributai žymimi paprasčiausiai. Privalomi atributai pažymimi juos tiesiog paryškinant. MS Visio įrankyje privalomiems atributams reikia nustatyti savybę „Set required“, o likę nepažymėti atributai yra neprivalomi.

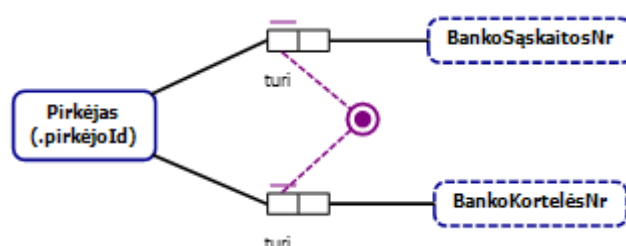
1.4.4. Alternatyvus privalomumas

Alternatyvus privalomumas (angl. Disjunctive mandatory) – tai sąlygą tarp dviejų ar daugiau atributų arba esybių kuomet vienas arba daugiau atributų arba esybių privalo egzistuoti, kad sąlyga būtų tenkinama [HC06]. Tai vadinama logine sudėtimi arba disjunkcija – teiginys „*x arba y*“ yra teisingas, kai teisingas bent vienas iš teiginio elementų *x* arba *y* [Kry04]. Pavyzdžiui PIRKĖJAS norėdamas naudotis elektronine parduotuve ir atsiskaityti už pirkinius privalo nurodyti banko sąskaitos numerį arba banko kortelės numerį. Tik tenkindamas vieną arba abi iš pateiktų sąlygų PIRKĖJAS galės atsiskaityti už pirkinius (2 lentelė).

2 lentelė. Disjunkcijos apibrėžimo aiškinamoji lentelė

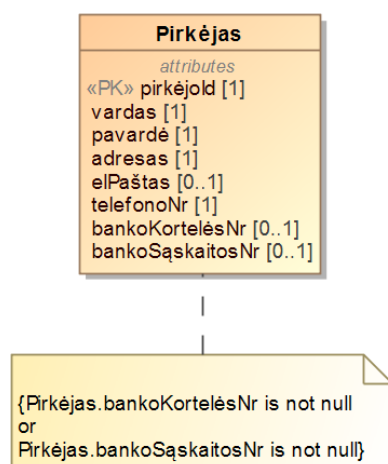
Banko sąskaitos numeris	Banko kortelės numeris	Gali atsiskaityti
Ne	Ne	Ne
Taip	Ne	Taip
Ne	Taip	Taip
Taip	Taip	Taip

ORM 2 – modelyje tokia sąlyga žymima alternatyvaus privalomo vaidmens (angl. Disjunctive mandatory role) apribojimu. Alternatyvus privalomas vaidmuo arba dar kitaip vadinamas „Imtinai-arba“ apribojimu (angl. Inclusive-Or) ORM 2 modelyje žymimas apvestu tašku kuris jungiamas prie apribojamų esybės rolių (8 pav.).



8 pav. Alternatyvaus privalomo atributo pasirinkimo pavyzdys – ORM 2 modelyje

UML ir E-R modeliai grafinio atitikmens ORM 2 modelyje atvaizduojamam alternatyvaus privalomo vaidmens apribojimui neturi. Tačiau Terry Halpin siūlo UML modelyje šalia atributų atvaizduoti kardinalumo [0..1] sąlygą, bei alternatyvios privalomos rolės apribojimą apibrėžti tekstiniu apribojimu prisegtoje žymoje (9 pav.) [Hal98a].



9 pav. Alternatyvaus privalomo atributo žymėjimas teksiniu apribojimu - UML

Autorius taip pat vėliau paprieštarauja sau ir pabrėžia, jog toks apribojimų žymėjimas grafinio atitikmens neturintiems apribojimams UML modeliuose nėra geriausia išeitis ir daugelyje

modelių tokių apribojimų yra tiesiog atsisakoma. Taigi galima teigti, kad ORM 2 modeliai žymint apribojimus atributams yra daug išraiškingesnis lyginant su UML bei E-R modeliavimo metodais.

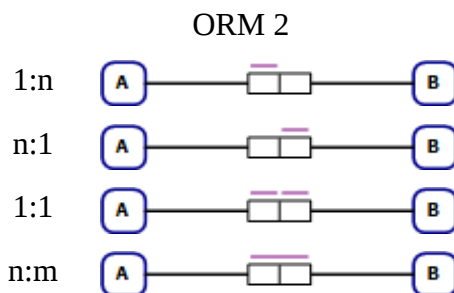
1.4.5. Vidinis unikalumas

Vidinis unikalumas (angl. Internal uniqueness) – nurodo atributų viename sąryšyje unikalumą [HM10]. Pasinaudojant T. Halpin ir T. Morgan knygoje „Information modeling and relational databases“ esančiu pavyzdžiu turime: 2 atributus „Žmogus“ ir „Šalis“. Formuojame sąlygą, kad „Žmogus“ gali būti gimęs daugiausia vienoje šalyje. Arba formuojant atvirkštinį variantą: „Yra neįmanoma, kad tas pats „Žmogus“ būtų gimęs daugiau nei vienoje šalyje“. 3 lentelėje matome, kad žmonės A, B ir C yra gimę tik vienoje šalyje: LT arba LV ir neturime tokio įrašo, kad žmogus A būtų gimęs antrą kartą LV šalyje. Taigi atributas „Žmogus“ yra unikalus.

Lentelė 3. Žmogaus gimusio konkrečioje šalyje pavyzdys

Žmogus	Šalis
A	LT
B	LT
C	LV

Šis apribojimas įgyvendinamas tik ORM 2 modeliavimo metodu. UML ir E-R modeliavimo metodai neturi galimybės atvaizduoti atributų unikalumo koncepciniame modelyje. ORM 2 diagramose unikalumas žymimas virš atributo atliekamos rolės brūkšneliu, priklausomai nuo sąryšio laipsnio (10 pav.).

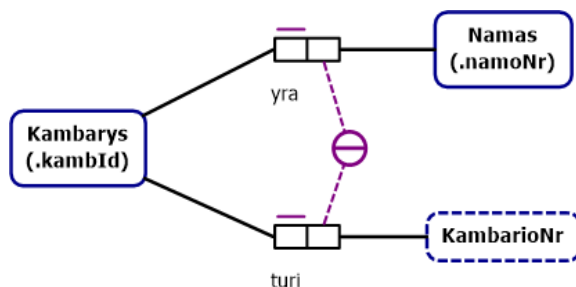


10 pav. ORM 2 vidinio unikalumo žymėjimas

1.4.6. Išorinis unikalumas

Išorinis unikalumas (angl. External uniqueness) - šiuo apribojimu nurodoma, jog dviejų ar daugiau atributų arba esybių objektų junginys gali priklausyti vienam kitos esybės objektui [HM10]. Pavyzdžiui turime tokią situaciją: “Kambarys turi kambario numerį X” ir “Kambarys yra name, kurio numeris Y”. Jei abiem kambario rolėms nurodysime išorinio unikalumo apribojimą, tuomet galime suformuoti tokį sąryšį: “Kiekviena Namo numerio ir Kambario numerio kombinacija gali turėti/priklausyti daugiausia vienam Kambariui”.

ORM2 – modeliuose šis apribojimas žymimas apskritimu su viena horizontalia linija centre, kuris yra sujungiamas su reikiomomis 2 arba daugiau rolių, kaip pavaizduota 11 paveiksle.

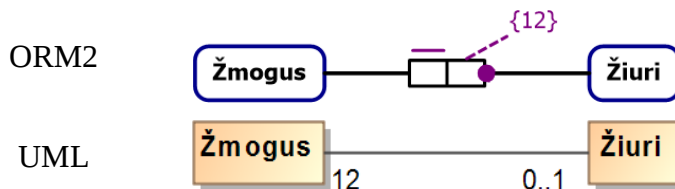


11 pav. Išorinio unikalumo vaizdavimas ORM2 modelyje

UML ir ER modeliai šio apribojimo atitikmens neturi.

1.4.7. Vidinis dažnumas

Vidinis dažnumas (angl. Internal frequency) – nurodo kiek konkrečios esybės objektų gali dalyvauti sąryšyje. Pasinaudosime ORM 2 grafinės notacijos aprašytos Terry Halpin pavyzdžiu [Hal11]. Turime 2 esybes „Žmogus“ ir „Žiuri“, kurioms apibrėžiamas reikalavimas, kad „Žiuri“ privalo sudaryti lygiai 12 „Žmonių“. 12 paveiksle pateiktas pavyzdys vidinio dažnumo reikalavimui pažymėti ORM 2 ir UML diagramose. Pasirinkta E-R notacija negali tiksliai apibrėžti šio apribojimo, kadangi galimi tik „0“, „1“ arba „daug“ kardinalumo žymėjimai.



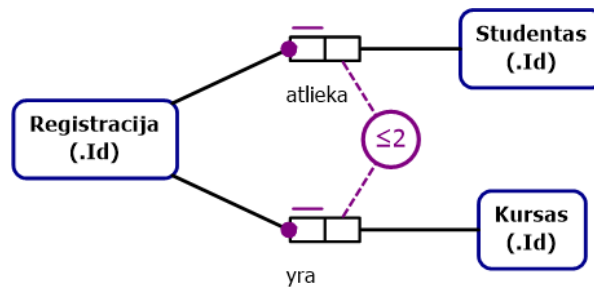
12 pav. Vidinio dažnumo pavyzdys

ORM 2 – modelyje vidinis dažnumas žymimas sąryšio dalyvių skaičių sujungiant su reikiama role.

UML – modelyje vidinis unikalumas žymimas patikslinant sąryšio įvairovės sąlygą.

1.4.8. Išorinis dažnumas

Išorinis dažnumas (angl. External frequency) – nurodo kiek maksimaliai gali būti sudaryta skirtingų atributų kombinacijų. Tikslesniam apibūdinimui pasinaudosiu Terry Halpin ir Tony Morgan knygoje Information Modeling and Relational Databases pateiktu pavyzdžiu [HM10]. Turime 3 esybes „Studentas“, „Dalykas“ ir „Registracija“. Formuojama sąlyga, kad studentas konkrečiam kursui gali užsiregistruoti ne daugiau kaip 2 kartus. Taigi maksimaliai galime sudaryti dvi „Studento“ ir „Kurso“ kombinacijas (13 pav.)



13 pav. Išorinio dažnumo pavyzdys ORM 2 diagramoje

ORM 2 – išorinis dažnumas žymimas apjungiant reikiamas roles apskritimu (žr. 15 paveikslą), kurio viduje nurodoma būtinumo sąlyga - tai gali būti konkretus skaičius arba nelygybė.

UML ir E-R modeliai neturi galimybės grafiškai atvaizduoti šio apribojimo.

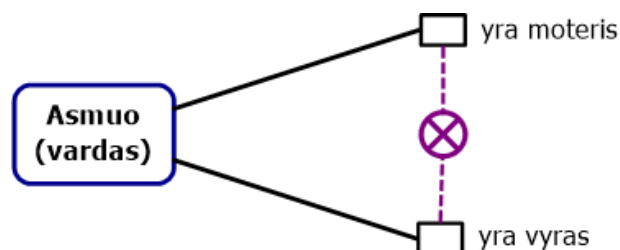
1.4.9. Išskyrimas

Išskyrimas (angl. Exclusion) – apribojami skirtingi atributai taip, kad reikšmę įgytų daugiausia vienas arba nei vienas iš apribotų atributų [HM10]. Pavyzdžiui turime 3 atributus, „Asmuo“, „yraVyras“, „yraMoteris“. Atributams „yraVyras“ ir „yraMoteris“ nurodytas išskyrimo apribojimas. Atributams galimos įgyti reikšmės atvaizduojamos 4 lentelėje:

Lentelė 4. Išskyrimo apribojimo aiškinamoji lentelė

Asmuo	yraVyras	yraMoteris
Jonas	Taip	-
Dalia	-	Taip
Petras	-	-
Tomas	Taip	Taip

Iš pateikto pavyzdžio matyti, kad pirmi 3 įrašai vardais: Jonas, Dalia ir Petras nurodyti gerai. Tai reiškia nurodyta tik viena reikšmė atributams yraVyras ir yraMoteris arba lytis nenurodyta visai. Tačiau įrašo eilutė vardu Tomas nėra galima kadangi reikšmės nurodytos abiem atributams. Šis įrašas duomenų įvedimo metu atmetamas.



14 pav. Išskyrimo atvaizdavimo pavyzdys ORM2 modelyje

ORM 2 – išskyrimas žymimas apjungiant reikiamas roles apskritimu, kurio viduje žymimas X simbolis (14 pav.).

UML ir E-R - modeliai neturi galimybės grafiškai atvaizduoti šio apribojimo.

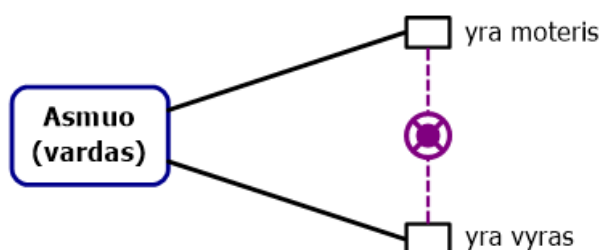
1.4.10. Išskyrimas – arba

Išskyrimas – arba (angl. Exclusive - or) – apribojami skirtingi atributai, taip kad reikšmės įgytų tik vienas iš apribotų atributų, tačiau ne abu [HM10]. Pasinaudojant išskyrimo apribojimo pavyzdžiu (žr. lentelė 4) turime tuos pačius 3 atributus: „Asmuo“, „yraVyras“, „yraMoteris“. Atributams „yraVyras“ ir „yraMoteris“ nurodytas išskyrimo apribojimas. Atributų galimos įgyti reikšmės pateikiamos 5 lentelėje:

Lentelė 5. Išskyrimo – arba apribojimo aiškinamoji lentelė

Asmuo	yraVyras	yraMoteris
Jonas	Taip	-
Dalia	-	Taip
Petras	-	-
Tomas	Taip	Taip

Iš pateikto pavyzdžio matyti, kad pirmi 2 įrašai vardais: Jonas, Dalia nurodyti gerai. Tai - nurodyta tik viena reikšmė atributams yraVyras ir yraMoteris. Tačiau įrašo eilutės vardu Petras ir Tomas nėra galimos, kadangi reikšmės nurodytos abiem arba nei vienam atributui. Šie įrašas duomenų įvedimo metu atmetami.



15 pav. Išskyrimo – arba apribojimo pavyzdys ORM2 modelyje

ORM2 – išskyrimo – arba apribojimas žymimas apskritimu kurio centre žymimas X simboliu perdengtu tašku (15 pav.).

UML ir E-R modeliai neturi galimybės grafiškai atvaizduoti šio apribojimo.

1.4.11. Žiediniai apribojimai

Žiedinis apribojimas (angl. Ring constraint) – ribojimas taikomas tos pačios esybės skirtingiems objektams susietiems binariniu ryšiu. Tokio tipo ryšius, kai esybė rodo pati į save, įgalina visi nagrinėjami modeliavimo metodai, tačiau apribojimų tokio tipo ryšiams nepalaiko nei UML, nei E-R modeliai. ORM 2 modeliai sudaro sąlygas atvaizduoti net keletą skirtingo tipo žiedinių apribojimų [HM10], [Hal98c]:

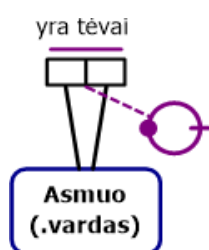
- Nerefleksyvus (angl. Irreflexive);
- Asimetris (angl. Asymmetric);

- Netranzytyvus (angl. Intransitive);
- Nesimetrinis (angl. Antisymmetric);
- Neciklinis (angl. Acyclic);

1.4.11.1. Nerefleksyvus

Nerefleksyvus (angl. Irreflexive) ryšio, kuris sieja tą pačią esybę su pačia savimi, apribojimas apibrėžiamas:

R yra nerefleksyvus tada ir tik tada kai $\forall x, \neg xRx$ [HM10];



Ana	Ana
Ana	Bilas
Bilas	Ana
Bilas	Kolinas
Ana	Kolinas

16 pav. Nerefleksyvaus ryšio pavyzdys

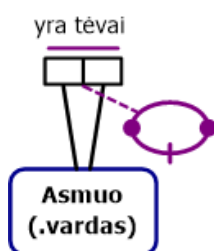
Iš pavyzdžio pateikto 16 paveiksle, nerefleksyvaus ryšio apribojimą galime įvardinti kaip: „Nė vienas žmogus yra tėvas pats sau“. Taigi esybės „Asmuo“ objektas Ana negali būti susietas pats su savimi pasinaudojant žiediniu ryšiu.

1.4.11.2. Asimetrinis

Asimetrinis (angl. Asymmetric) ryšio, kuris sieja tą pačią esybę su pačia savimi, apribojimas apibrėžiamas:

R yra asimetrinis tada ir tik tada kai $\forall x, y: xRy \rightarrow \neg yRx$ [HM10];

Šis apribojimas nusako jog, jeigu galima duomenų kombinacija (A, B) tuomet kombinacija (B, A) nėra galima. Taigi, jeigu A esybės objektas turi ryšį su B esybės objektu, tai atvirkščias ryšys negalimas.



Ana	Bilas
Bilas	Ana
Bilas	Kolinas
Ana	Kolinas

17 pav. Asimetrinio ryšio pavyzdys

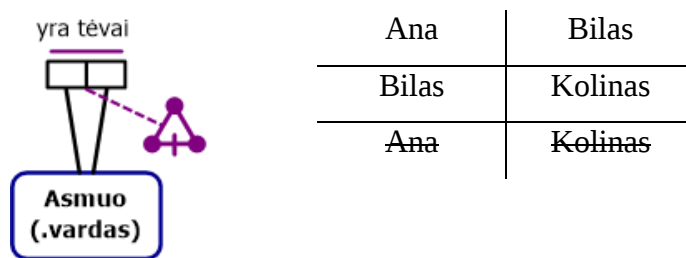
Iš pateikto pavyzdžio 17 paveiksle matome, jog jeigu Ana yra Bilo mama, tai Bilas negali būti Anos tėvu. Beje asimetrinis ryšys privalo būti ir nerefleksyvus [HM10].

1.4.11.3. Netranzytyvus

Netranzytyvus (angl. Intransitive) ryšio, kuris sieja tą pačią esybę su pačia savimi, apribojimas apibrėžiamas:

R yra netranzytyvus tada ir tik tada kai $\forall x, y, z: xRy \wedge yRz \rightarrow \neg xRz$ [HM10];

Apribojimu nurodoma, kad turimoms duomenų kombinacijoms (A, B) ir (B, C), kombinacija (A, C) yra negalima. Taigi jei esybės objektas A turi ryšį su tos pačios esybės B objektu, o B turi ryšį su to pačios esybės C objektu, tuomet ryšys tarp A ir C objektų nėra galimas.



18 pav. Netranzytyvaus ryšio pavyzdys

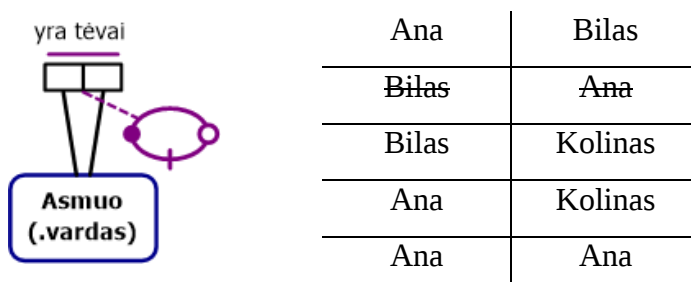
Iš pateikto pavyzdžio 18 paveiksle matyti, kad Ana yra Bilo mama, o Bilas yra Kolino tėvas, tačiau Ana negali būti Kolino mama. Beje netranzytyvus ryšys privalo būti ir nerefleksyvus.

1.4.11.4. Nesimetrinis

Nesimetrinis (angl. Antisymmetric) ryšio, kuris sieja tą pačią esybę su pačia savimi, apribojimas apibrėžiamas:

R yra nesimetrinis tada ir tik tada kai $\forall x, y: x \neq y \wedge xRy \rightarrow \neg yRx$ [HM10].

Nesimetrinis ryšio apribojimas panašiai kaip ir asimetrinio ryšio ribojimas, esant ryšiui tarp skirtingų esybės objektų neleidžia egzistuoti priešingam ryšiui tarp tų pačių esybės objektų, tačiau šis apribojimas nėra nerefleksyvus. Taigi yra leidžiamas ryšys tarp tos pačios esybės vieno objektų.



19 pav. Nesimetrinio ryšio pavyzdys

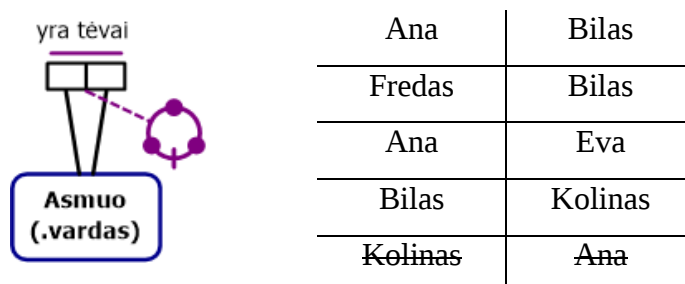
Taigi iš 19 paveiksle matomo pavyzdžio galima teigti, jog jeigu Ana yra Bilo mama, tai Bilas negali būti Anos tėvu. Dėl to ryšys tarp objektų Bilas ir Ana negalimas. Tačiau gaunamas ryšys, jog Ana gali būti pati savo mama, kadangi šis ryšys nėra nerefleksyvus.

1.4.11.5. Neciklinis

Neciklinis (angl. Acyclic) ryšio, kuris sieja tą pačią esybę su pačia savimi, apribojimas apibrėžiamas:

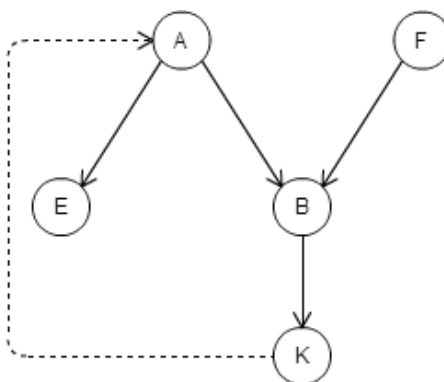
R yra neciklinis tada ir tik tada kai $\forall x: \neg(x \text{ turi kelią į } x)$ [Kee07]

Šio tipo žiedinis apribojimas nurodo, kad esybės objektas A turi ryšį su tos pačios esybės objektu B, o B turi ryšį su tos pačios esybės objektu C, tačiau objektas C negali turėti ryšio su objektu A, kadangi tokiu atveju gauname ciklą sąryšyje.



20 pav. Neciklinio ryšio pavyzdys

20 pav. pateiktame pavyzdyje matome, kad Ana yra Bilo mama, Bilas yra Kolino tėvas, tačiau Kolinas negali būti Anos tėvu, kadangi tokiu atveju gautume ciklą sąryšyje. Šį ciklą atvaizduojamas 21 paveiksle.

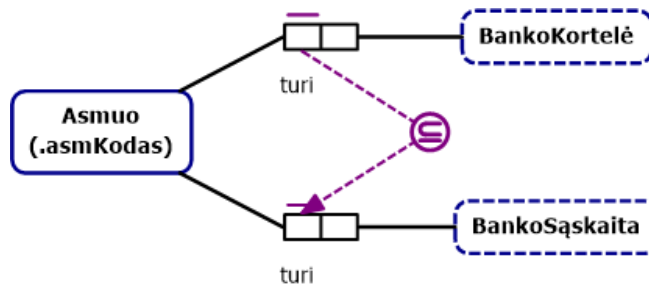


21 pav. Ciklo sąryšyje pavyzdys

Negalimas ryšys tarp Kolino ir Anos žymimas punktyrine linija. Jei šis ryšys būtų leidžiamas gautume ciklą tarp 22 paveiksle pateikto pavyzdžio objektų: Anos, Bilo ir Kolino.

1.4.12. Poaibis

Poaibis (angl. Subset) – apibrėžia, kad vienas ryšys yra kito poaibis, taigi jei egzistuoja poaibio ryšys, privalo egzistuoti ir pagrindinės aibės ryšys [WK03]. Detalesniam apibrėžimui atvaizduojama sąlyga, kad žmogus, kuris turi banko kortelę taip pat turi ir banko sąskaitą. Taigi turime esybę „Žmogus“, kurios atributams „BankoKortelė“ ir „BankoSąskaita“ yra nurodomas poaibio apribojimas. Dėl to gauname sąlygą, kad turint banko kortelę privalu turėti ir banko sąskaitą. Šis pavyzdys atvaizduojamas 22 paveiksle.



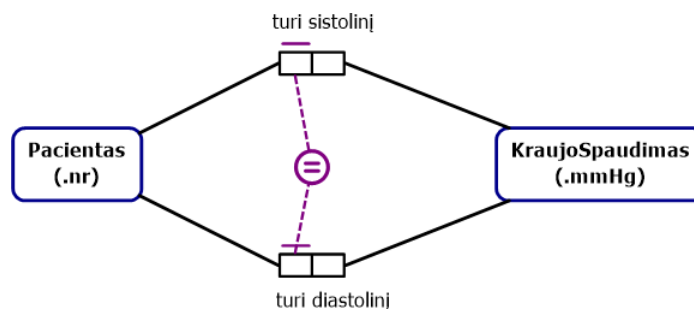
22 pav. Poaibio vaizdavimas ORM 2 modelyje

ORM 2 – poaibio apribojimas žymimas specialiu simboliu „⊆“ apskritime (žr. 24 paveikslą), sujungtu punktyrine linija su rodykle, kuri rodo į super-tipo (angl. Superset) rolę [HM10].

UML ir E-R modeliai neturi galimybės grafiškai atvaizduoti šio apribojimo.

1.4.13. Lygybė

Lygybė (angl. Equality) – nurodoma, kad objektų atributų arba sąryšyje dalyvaujančių objektų aibės privalo būti lygios [HM10]. Pasinaudojant Terry Halpin pateiktu pavyzdžiu turime 2 esybes „Pacientas“ ir „KraujoSpaudimas“, bei roles „turi sistolinį“ ir „turi diastolinį“ tarp jų. Apribojimas nurodo, kad paciento sistolinis kraujo spaudimas gali būti nurodytas tik tada jei nurodytas jo diastolinis kraujo spaudimas (23 pav.).



23 pav. Lygybės vaizdavimas ORM 2 modelyje

ORM 2 – modelyje lygybės apribojimas vaizduojamas simboliu „=“ apskritime, kuris sujungiamas punktyrine linija su ribojamomis rolėmis kaip pavaizduota 25 paveiksle.

UML ir E-R modeliai neturi galimybės grafiškai atvaizduoti šio apribojimo.

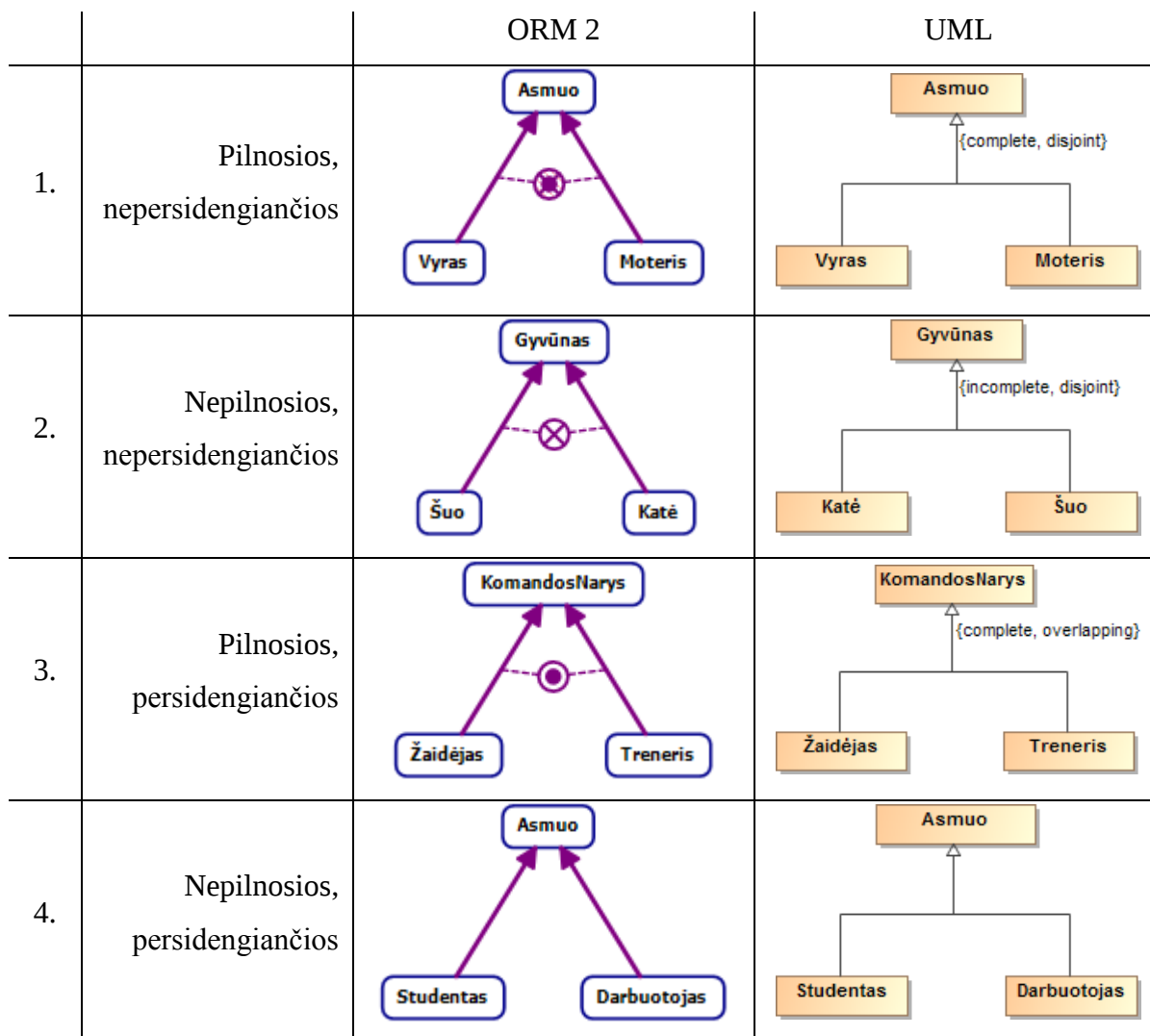
1.4.14. Potipių apribojimai apibendrinimo ryšiui

Potipis (angl. Subtype) – esybės egzempliorių klasifikavimas į konkretesnius jų tipus [HM10]. Pavyzdžiui tėvinę esybę „Žmogus“ galime išskaidyti į 2 vaikinės esybės, kurioms galėsime nusakyti skirtingas savybes, tai „Vyras“ ir „Moteris“. Egzistuoja 4 pagrindiniai potipių tipai [HM10]:

1. Pilnosios (angl. Complete) – visi bendro tipo esybės objektai privalo turėti konkretaus tipo esybės objektus;

2. Nepilnosios (angl. Incomplete) – ne visi bendro tipo esybės objektai privalo turėti konkretaus tipo esybės objektus;
3. Nepersidengiančios (angl. Disjoint) – vienam bendro tipo esybės objektui gali būti nurodytas tik vienas konkretaus tipo esybės objektas;
4. Persidengiančios (angl. Overlapping) – vienam bendro tipo esybės objektui gali būti nurodytas 1 arba daugiau skirtingų konkretaus tipo esybių objektų;

Be pagrindinių poaibių tipų galimos ir jų kombinacijos (24 pav.), būtent jomis ir išreiškiami potipių apribojimai [OMG15]:



24 pav. Potipių apribojimai ORM 2 ir UML modeliuose

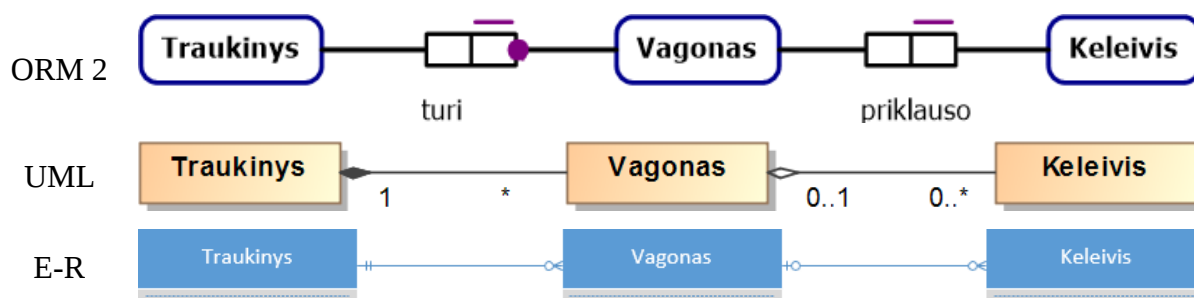
1. Pilnosios, nepersidengiančios (angl. Complete, disjoint) – kiekvienas bendro tipo esybės objektas turi turėti konkretaus tipo esybės objektą ir šie negali persidengti. Pavyzdžiui: bendro tipo esybė „Asmuo“, kurios konkretaus tipo esybės „Vyras“ ir „Moteris“. Taigi apibūdinant asmens lytį yra pateikiami visi galimi subtipo variantai;
2. Nepilnosios, nepersidengiančios (angl. Incomplete, disjoint) – ne visi bendro tipo esybės objektai privalo turėti konkretaus tipo esybių objektus, tačiau šie

- nepersidengia. Pavyzdžiui: bendro tipo esybės „Gyvūnas“, konkretaus tipo esybės „Šuo“ ir „Katė“. Tačiau galimos ir kitos gyvūnų rūšys, kurios neturi konkretaus tipo esybių: „Paukštis“, „Žuvis“, taip pat tas pats „Gyvūnas“ negali būti 2 skirtingų rūšių;
3. Pilnosios, persidengiančios (angl. Complete, overlapping) – kiekvienas bendro tipo esybės objektas turi turėti bendro tipo esybės objektą, tačiau bendro tipo esybės objektas gali turėti 1 arba daugiau konkrečių tipų esybių objektų. Pavyzdžiui: bendro tipo esybė „KomandosNarys“, konkretaus tipo esybės „Žaidėjas“ ir „Treneris“. Visa galima komandos sudėtis nurodyta konkretaus tipo esybėmis, tačiau žaidėjas taip pat gali būti ir treneriu, dėl ko gauname persidengiančias konkretaus tipo esybes;
 4. Nepilnosios, persidengiančios (angl. Incomplete, overlapping) – ne visi bendro tipo esybių objektai privalo turėti konkretaus tipo esybių objektus, tačiau bendro tipo esybės objektas gali turėti 1 arba daugiau konkretaus tipo esybių objektus; Pavyzdžiui: bendro tipo esybė „Asmuo“, konkretaus tipo esybės „Darbuotojas“ ir „Studentas“. Gaunama sąlyga, kad studentas taip pat gali būti ir darbuotoju, tačiau taip pat tai nenurodo pilnos asmens klasifikacijos, asmenys gali būti skirstomi ne tik į darbuotojo ar studento konkrečius tipus;

1.4.15. Agregacija ir kompozicija

Agregacija (angl. Aggregation) apibrėžiama, kaip asociacija (sąryšis) apibūdinanti „visuma - dalis“ sąryšį arba paprasčiau, tai sąryšis, kai viena esybė yra kitos esybės dalis. Agregacija yra pabrėžiama, kad vienos esybės dalis yra nepriklausoma nuo kitos, taigi pašalinus vieną iš esybių likusi gali egzistuoti toliau [HM10].

Kompozicija (angl. Composition) – ryšys, kuriuo yra nurodoma stipri gyvavimo ciklo priklausomybė tarp dviejų esybių arba kitaip – tėvinio ryšio priklausomybė. Taigi pašalinus tėvinę esybę, jos vaikinės esybės taip pat pašalinamos. Tai priešingas sąryšis agregacijai [HM10].



25 pav. Agregacijos ir kompozicijos atvaizdavimas.

Pateiktame pavyzdyje (25 pav.) UML diagramoje tarp esybių „Keleivis“ ir „Vagonas“ vaizduojama agregacija, o tarp „Traukinys“ ir „Vagonas“ kompozicija. Kaip pastebime ORM ir E-R diagramos neturi specifinio agregacijos bei kompozicijos ryšių vaizdavimo, taigi šie elementai yra išreiškiami paprastų ryšių pagalba. Ryšys, kuris UML modelyje atitinka agregacija ORM 2 ir

E-R modeliuose neturi privalomo ryšio apribojimo, 25 paveiksle matome, kad tiek „Keleivis“, tiek „Vagonas“ esybės gali egzistuoti nepriklausomai viena nuo kitos. Kompozicijos – vienai iš pasirinktų esybių nurodomas privalomo ryšio apribojimas, kuris 25 paveiksle nurodytas „Vagono“ esybei, kadangi „Vagono“ esybė negali egzistuoti be „Traukinio“ esybės. Taigi pašalinus Traukinio esybę, pašalinama ir jos vaikinė esybė „Vagonas“.

UML – modeliuose agregacija yra žymima binarine asociacija su tuščiaiduriu rombu šalia esybės, kuri atitinka „visumos“ dalį, panašiai atvaizduojama ir kompozicija tik rombas yra užpildytas, o ne tuščiaiduris.

1.5. Modeliavimo metodų apibendrinimas

Taigi apžvelgus mokslines publikacijas, pateiktus konkrečių grafinių konceptų palyginimo pavyzdžius bei išanalizavus kiekvieno modeliavimo metodo privalumus bei trūkumus galime apibūdinti modeliavimo metodus pagal darbo pradžioje aprašytus kriterijus:

Aiškumas – juo vertinamas modeliavimo kalbos suvokimas, vartotojo gebėjimas pasinaudoti sukurtu modeliu. Kaip minėta anksčiau vertinamas duomenų modelis turėtų būti nedviprasmiškas, lengvai skaitomas bei perprantamas, intuityvus. Paprasta modelių verbalizacija sakiniiais, platesnis sąryšių panaudojimas ORM 2 modeliuose supaprastina jo skaitomumą, padeda išvengti dviprasmybių, taip pat leidžia vartotojui lengviau pasinaudoti sukurtu modeliu, tačiau gana netradicinis sąryšio elementų vaizdavimo būdas reikalauja didesnės patirties iš vartotojo, kuris naudosis sukurtais ORM 2 modeliais. Palyginti E-R „Crow’s Foot“ notacijoje sąryšio kardinalumo vaizdavimas yra daug intuityvesnis ir lengviau įsisavinamas paprastų vartotojų. Apibendrinant UML – tai gana techniškas bei specifikuotas modeliavimo metodas, kuris reikalauja nemažo žinių kiekio iš vartotojo norint perprasti modelyje pateikiamą dalykinę sritį, taip pat jis nepasižymi lengva verbalizacija. Taigi modeliuojant paprastesnę dalykinės srities modelį siūloma rinktis E-R arba UML modeliavimo metodus priklausomai nuo to, koks detalus privalės būti kuriamas duomenų modelis.

Išraiškingumas – tai metodo ekspresyvumo bei modeliuojamos srities pateikimo tikslumo vertinimas. Modeliavimo metodas turėtų atvaizduoti duomenis kiek galima išsamiau ir tiksliau. Taigi išanalizavus mokslinėje literatūroje pateiktus metodų pavyzdinius modelius bei sumodeliuotų diagramų pavyzdžius galima teigti, jog ekspresyviausias metodas iš visų trijų yra ORM 2. Šio metodo ekspresyvumas yra proporcingas jame perteikiamų vientisumo reikalavimų kiekiui, jį ORM 2 modeliavimo metodas turi daugiausia, kas lemia, jog metode duomenys yra atvaizduojami daug išsamiau bei tiksliau. Kiti aspektai dėl ko šį modeliavimo metodą galima vadinti ekspresyviausiu yra sąryšio pateikimas tarp atributo ir esybės, aiškus rolės būtinumo nurodymas. Tačiau ekspresyvumas ORM 2 modeliuose turi ir neigiamą pusę, dėl gana didelio grafinių konceptų kiekio atvaizduojamo ORM 2 diagramose, jos labai greitai plečiasi ir tampa

labai didelės palyginti su likusių modeliavimo metodų diagramomis. UML ir E-R diagramose esybių ypatybės formuojamos atributais atsisakant sąryšių, dėl šios priežasties abu modeliavimo metodai praranda dalį ORM 2 diagramose įgyvendinamų apribojimų, taigi pastarasis metodas yra išraiškingesnis.

Skirtingų grafinių konceptų kiekis – plačiausia grafinių konceptų aibe pasižymi ORM 2 modeliai, po to seka UML ir E-R. E-R modeliuos atvaizduojamų grafinių konceptų kiekis. Visi nagrinėti modeliavimo metodai turi vienodus pagrindinius konceptus, bet žymiai skiriasi gebėjimu pažymėti vientisumo reikalavimus jiems. Taigi ORM 2 modeliai gali daug detaliau perteikti modeliuojamą dalykinę sritį, tačiau grafiniai elementai ORM 2 yra gana neintuityvus, pavyzdžiui atributų vaizdavimas UML ir E-R modeliuose yra gana paprastas, jie tiesiog atvaizduojami reikiamos esybės viduje, na, o ORM 2 modeliuose atributai yra atskiriami nuo esybių sąryšiais, be to esybių ir atributų grafinis atvaizdavimas labai panašus, galintis suklaidinti nepatyrusį vartotoją. Nors ORM modeliavimo metodas turi platesnę grafinių elementų aibę, tačiau E-R ir UML modeliuose jie yra atvaizduojami intuityviau, bei užima mažiau vietos.

Duomenų vientisumas – tai duomenų tikslumo bei nuoseklumo užtikrinimas visą jų gyvavimo ciklą, tai vienas svarbiausių aspektų projektavimo, įgyvendinimo bei sistemos naudojimo metu. Nors duomenų vientisumo supratimas gali būti skirtingas, tačiau iškelti tikslai jiems pasiekti yra vienodi tam pasitelkiami duomenų vientisumo reikalavimai. Duomenų vientisumo reikalavimai – tai duomenų modelio kūrimo technikos skirtos užtikrinti būsimų duomenų tikslumą bei nuoseklumą. Išanalizavus visus tris ORM 2, UML ir E-R duomenų modeliavimo metodus galima teigti, jog didžiausia vientisumo reikalavimų aibę turi ORM 2 modeliai. Pirminio identifikatoriaus, būtinumo/nebūtinumo, ryšių reikalavimai atributams yra svarbiausi reikalavimai, naudojami visuose nagrinėtuose modeliavimo metoduose. Vaizdinio vientisumo reikalavimų modeliavimo požiūriu ORM 2 modelis yra gana išraiškingas, bet tuo pačiu sudėtingas ir tinkamas tik analizės etapui. Nepaisant to, kad yra metodų ir galimybių transformuoti ORM 2 modelius į analizės ir projektavimo etapams skirtą CASE įrankių aplinkas (pavyzdžiui, Microsoft Visio-Based Database Modelling Visual Studio .NET Enterprise Architect), sunku įsivaizduoti, kad ORM 2 būtų galima praktiškai naudoti informacinių sistemų programinės įrangos kūrimo projektuose. Žemiau pateiktoje lentelėje (6 lentelė) apibendrinami vientisumo reikalavimų tipai, grafiškai vaizduojami analizuotuose koncepciniuose modeliuose.

Lentelė 6. Modeliuose palaikomi vientisumo reikalavimai

	ORM 2	UML	E-R
Pirminis raktas	+	+	+
Išorinis raktas	-	+	+
Privalomas ir neprivalomas atributai	+	+	+
Alternatyvus privalomas atributas	+	-	-
Vidinis unikalumas	+	-	-
Išorinis unikalumas	+	-	-
Vidinis dažnumas	+	+	-
Išorinis dažnumas	+	-	-
Išskyrimas	+	-	-
Išskyrimo - arba	+	-	-
Žiediniai apribojimai:			
• Nerefleksyvus	+	-	-
• Asimetris	+	-	-
• Netranzytius	+	-	-
• Nesimetris	+	-	-
• Neciklinis	+	-	-
Poaibis	+	-	-
Lygybė	+	-	-
Potipio:			
• Pilnosios, nepersidengiančios	+	+	-
• Nepilnosios, nepersidengiančios	+	+	-
• Pilnosios, persidengiančios	+	+	-
• Nepilnosios, persidengiančios	+	+	-
Agregacija, kompozicija	+	+	+
Iš viso:	21	9	4

2. Praktinis modeliavimo metodų išbandymas

Dalykinė sritis – modeliuojamas elektroninės komercijos įrankis skirtas prekių paieškai, apskaitai bei užsakymui internetu;

Problema – įmonė užsiimanti prekyba internetu bei turi neribotą prekių ir prekių kategorijų skaičių. Reikia pateikti duomenų modelį, kurio pagalba būtų atvaizduojamas prekių katalogas bei užtikrintas prekių pardavimo procesas;

Dalykinės srities aprašymas:

Elektroninė parduotuvė turi neriboto dydžio prekių katalogą, kuris skirstomas į kategorijas.

Kategorijų tarpusavio ryšiai nusakomi medžio struktūra, kur tėvinė kategorija gali turėti neribotą vaikinių kategorijų skaičių. Kategorija saugo šią informaciją: pavadinimas, aprašymas (neprivalomas). Kiekviena kategorija gali turėti daugiausia 3 ją atvaizduojančius paveikslus. Kiekvienai kategorijai gali būti priskirtas neribotas prekių skaičius. Kiekviena prekė privalo priklausyti tik 1 kategorijai. Prekė saugo šią informaciją: kodas (unikalus), pavadinimas, aprašymas (neprivalomas), vieneto kaina, brūkšninis kodas (neprivalomas, unikalus), kiekis. Prekės kaip kategorijos gali turėti daugiausia 3 ją atvaizduojančius paveikslus.

Sistema gali naudotis tik registruoti vartotojai (turintys prisijungimo slaptažodį), kuriems nurodyta kontaktinė informacija: vardas, pavardė, el. paštas arba telefono numeris (bent 1 privalomas), šalis, miestas, adresas, pašto kodas, fizinio arba juridinio asmens ypatybė, jei asmuo juridinis jam papildomai privalome nurodyti įmonės pavadinimą ir kodą, pvm kodą (neprivalomas).

Prisijungęs vartotojas sudaro savo prekių krepšelį, kuris saugo neribotą pasirinktų prekių skaičių ir jų kiekius. Iš prekių krepšelio vartotojas formuoja užsakymą, kurį privalo minimaliai sudaryti bent 1 prekė. Formuojant užsakymą būtina nurodyti pirkėjo bei pristatymo informaciją, kuri gali sutapti su vartotojo kontaktiniais duomenimis arba gali būti nurodyta individualiai. Taip pat pirkėjo informacijos įvedimo metu vartotojas pasirenką vieną iš atsiskaitymo tipų: grynaisiais arba pavedimu.

Suformuoti užsakymai išsaugomi nurodant: užsakymo numerį (unikalus), datą. Užsakymą gali sudaryti neribotas prekių skaičius (tačiau ne mažiau nei 1 prekė), kurių rinkiniams yra išsaugoma: prekės vieneto kaina, užsakytas kiekis.

Taip pat kiekvieno objekto kūrimo metu privalome išsaugoti jo sukūrimo datą.

Esybės ir jų atributai:

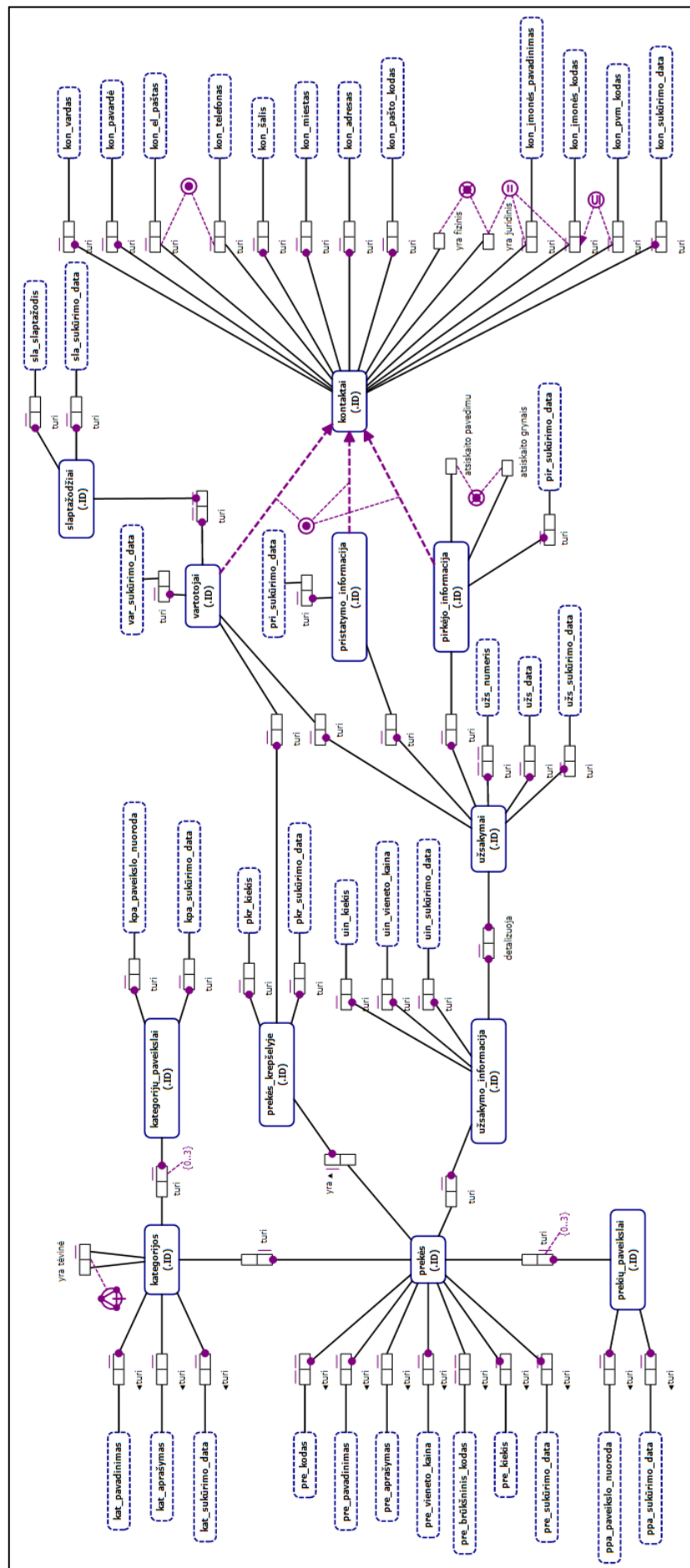
Pagal pateiktą dalykinės srities aprašymą suformuota esybių bei jų atributų lentelė (7 ir 8 lentelės).

7 lentelė. Dalykinės srities esybės bei atributai

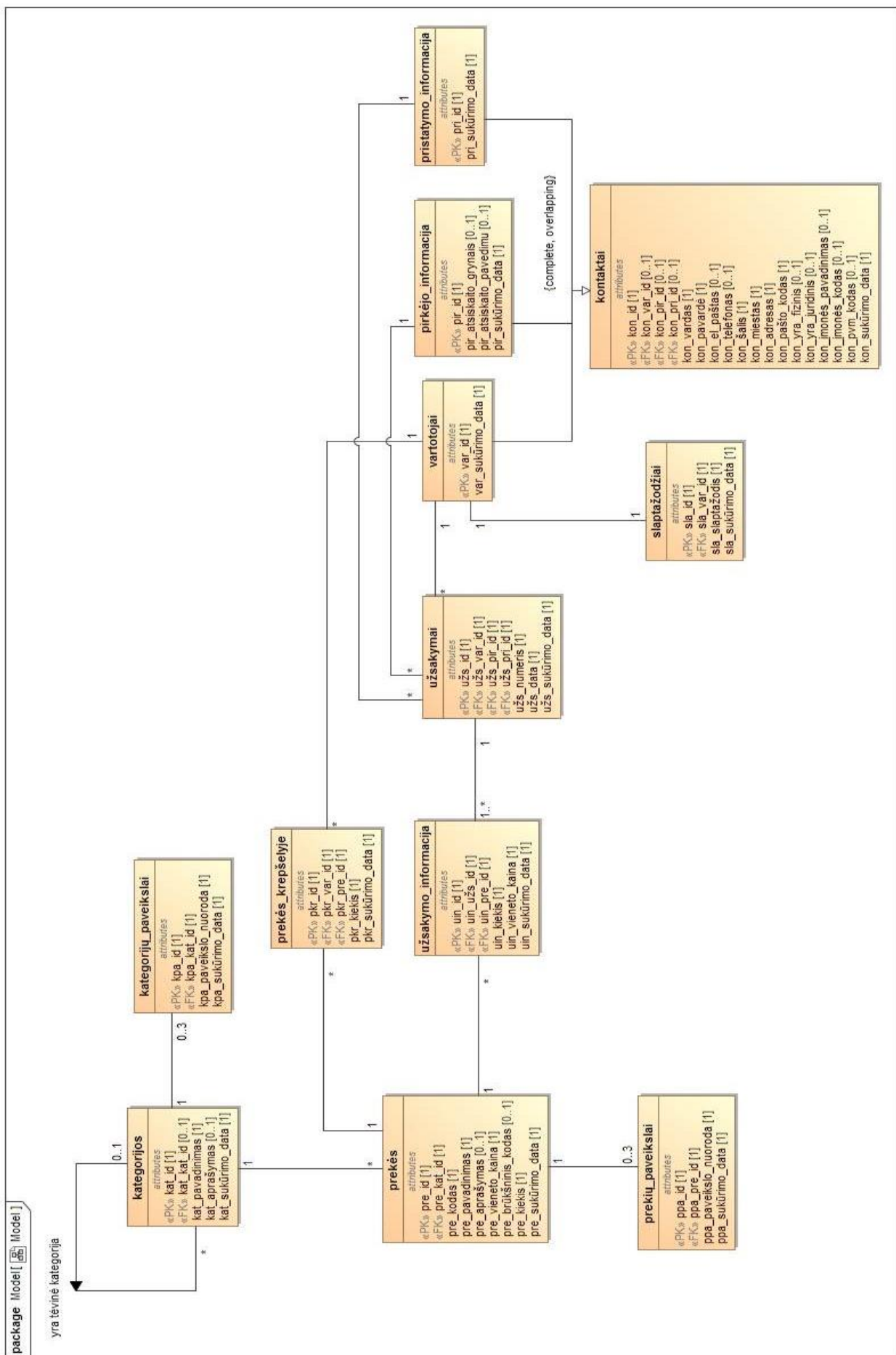
Esybė	Esybės aprašymas ir atributai
Kategorijos	• <u>ID</u> • Pavadinimas • Aprašymas (neprivalomas) • Sukūrimo data ○ Kat_id – nukreipia į Kategorijos – nurodo tėvinės – vaikinės kategorijų ryšį;
Kategorijų paveikslai	• <u>ID</u> • Paveikslo nuoroda • Sukūrimo data • Išoriniai raktai: ○ Kat_id – nukreipia į Kategorijos;
Prekės	• <u>ID</u> • Kodas (unikalus) • Pavadinimas • Aprašymas (neprivalomas) • Vieneto kaina • Brūkšninis kodas (neprivalomas, unikalus) • Kiekis • Sukūrimo data • Išoriniai raktai: ○ Kat_id – nukreipia į Kategorijos;
Prekių paveikslai	• <u>ID</u> • Paveikslo nuoroda • Sukūrimo data • Išoriniai raktai: ○ Pre_id – nukreipia į Prekės
Prekės krepšelyje	• <u>ID</u> • Kiekis • Sukūrimo data • Išoriniai raktai: ○ Var_id – nukreipia į Vartotojai ○ Pre_id – nukreipia į Prekės
Užsakymai	• <u>ID</u> • Numeris (unikalus) • Data • Sukūrimo data • Išoriniai raktai: ○ Var_id – nukreipia į Vartotojai ○ Pir_id – nukreipia į Pirkėjo informacija ○ Pri_id – nukreipia į Pristatymo informacija

8 lentelė. Dalykinės srities esybės bei atributai (7 lentelės tęsinys)

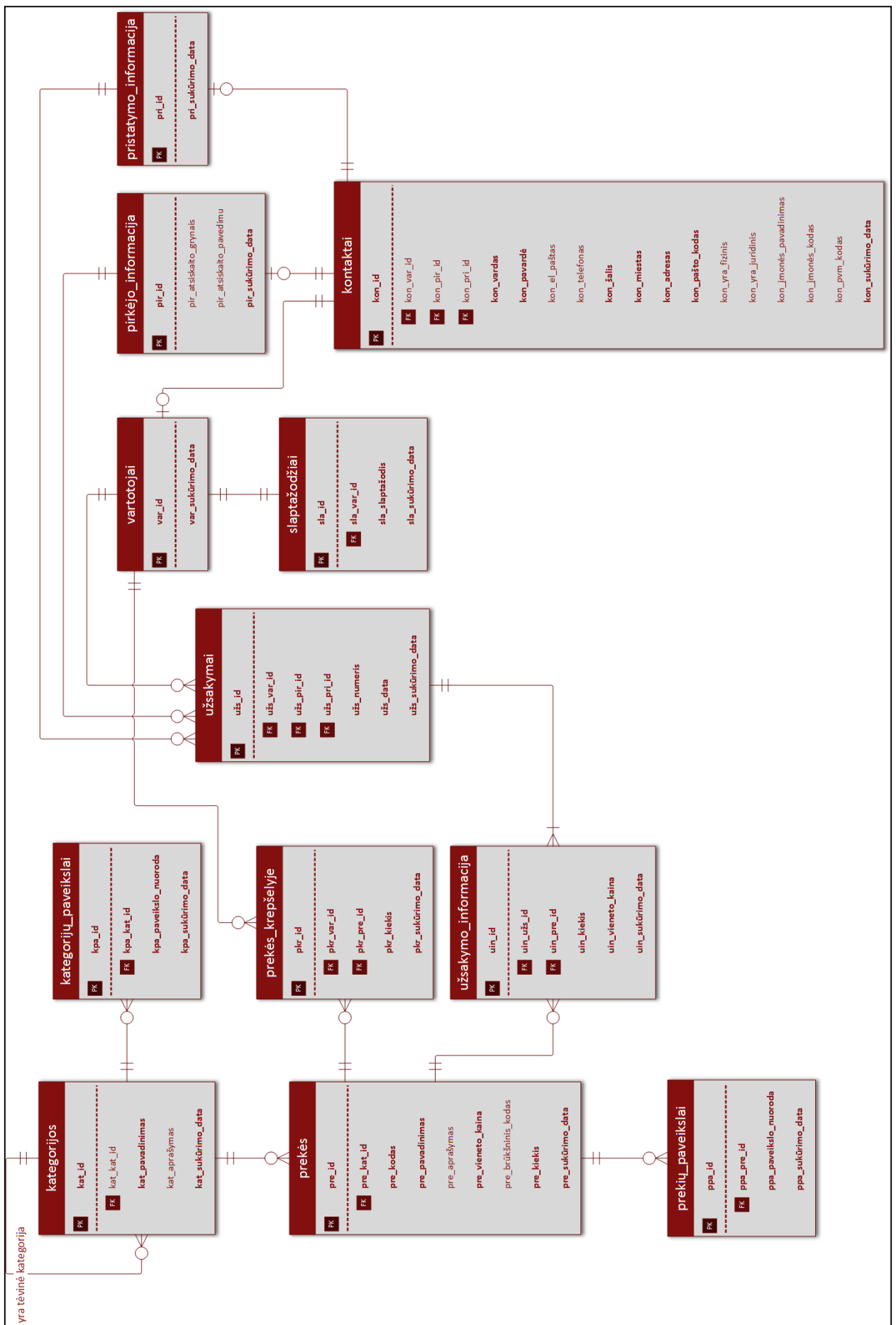
Esybė	Esybės aprašymas ir atributai
Užsakymo informacija	• <u>ID</u> • Kiekis • Vieneto kaina • Sukūrimo data • Išoriniai raktai: ○ Užs_id – nukreipia į Užsakymai ○ Pre_id – nukreipia į Prekės
Vartotojai	• <u>ID</u> • Sukūrimo data
Slaptažodžiai	• <u>ID</u> • Slaptažodis • Sukūrimo data • Išoriniai raktai: ○ Var_id – nukreipia į vartotojai;
Pirkėjo informacija	• <u>ID</u> • Atsiskaito grynais • Atsiskaito pavedimu • Sukūrimo data Privalomas pasirinkti 1 atsiskaitymo tipą
Pristatymo informacija	• <u>ID</u> • Sukūrimo data
Kontaktai	• <u>ID</u> • Vardas • Pavardė • El. paštas (neprivalomas) • Telefonas (neprivalomas) • Šalis • Miestas • Adresas • Pašto kodas • Yra fizinis (neprivalomas) • Yra juridinis (neprivalomas) • Įmonės pavadinimas (neprivalomas) • Įmonės kodas (neprivalomas) • PVM kodas (neprivalomas) • Sukūrimo data • Išoriniai raktai: ○ Var_id – nukreipia į Vartotojai ○ Pir_id – nukreipia į Pirkėjo informacija ○ Pri_id – nukreipia į Pristatymo informacija Privaloma nurodyti bent 1: telefono numeris, el. pašto adresą; Privaloma nurodyti bent 1: yra fizinis, yra juridinis; Pasirinkus juridinio asmens tipą kartu privaloma nurodyti: įmonės pavadinimas, įmonės kodas PVM kodas – gali būti nurodytas tik tuo atveju jei asmens tipas – juridinis ir nurodyti visi privalomi juridinio asmens duomenys



26 pav. ORM 2 koncepcinio duomenų modelio pavyzdys



27 pav. UML koncepcinio duomenų modelio pavyzdys



28 pav. E-R koncepcinio duomenų modelio pavyzdys

Lentelė 9. Praktinio išbandymo metu sukurtuose koncepciniuose modeliuose atvaizduojami vientisumo reikalavimai

Vientisumo reikalavimai	ORM 2	UML	E-R
Pirminis raktas	+	+	+
Išorinis raktas	-	+	+
Privalomas ir neprivalomas atributai	+	+	+
Alternatyvus privalomas atributas	+	-	-
Vidinis unikalumas	+	-	-
Išorinis unikalumas	-	-	-
Vidinis dažnumas	+	+	-
Išorinis dažnumas	-	-	-
Išskyrimas	-	-	-
Išskyrimo - arba	+	-	-
Žiediniai apribojimai:			
• Nerefleksyvus	-	-	-
• Asimetris	-	-	-
• Netranzytyvus	+	-	-
• Nesimetris	-	-	-
• Neciklinis	+	-	-
Poaibis	+	-	-
Lygybė	+	-	-
Potipio:			
• Pilnosios, nepersidengiančios	-	-	-
• Nepilnosios, nepersidengiančios	-	-	-
• Pilnosios, persidengiančios	+	+	-
• Nepilnosios, persidengiančios	-	-	-
Agregacija, kompozicija	-	-	-
Atvaizduota vientisumo reikalavimų:	11	5	3

Tiksliausiai dalykinę sritį perteikė ORM 2 modelis (26 pav.), kuris atvaizdavo visus dalykinės srities reikalavimus (9 lentelė).

UML modelyje (27 pav.) be pagrindinių vidinio, išorinio rakto bei privalomų ir neprivalomų atributų reikalavimų pažymėti ir vidinio dažnumo bei pilnojo persidengiančiojo potipio apribojimai (9 lentelė). Tačiau esybės „Pirkėjo informacija“ atributai tarpusavyje yra visiškai

neribojami. Galimas klaidingų duomenų pateikimas: pavyzdžiui tas pats pirkėjas už savo užsakymą gali atsiskaityti tiek grynaisiais, tiek bankiniu pavedimu, arba išvis nenurodyti atsiskaitymo tipo. Panaši situacija gaunama ir „Kontaktai“ esybėje. Nors pagal dalykinės srities reikalavimus yra privaloma pasirinktinai nurodyti (alternatyvaus privalomumo apribojimas) kontaktinį telefono numerį arba el. pašto adresą, remiantis sukurtu modeliu galime nenurodyti visiškai jokios kontaktinės informacijos, kadangi šie 2 atributai UML modelyje yra neprivalomi. Taip pat panaši situacija gaunama ir su asmens tipo pasirinkimu, kuomet reikia nurodyti tik vieną iš 2 galimų asmens tipų (išskyrimo – arba apribojimas). UML modelyje asmens tipo galime išvis nenurodyti arba pasirinkti abu, kadangi abu atributai taip pat žymimi kaip neprivalomi. Žinoma, tokios problemos galima išvengti modelyje pateikus tik vieną neprivalomą atributą „yra_juridinis“, kuris atskirtų fizinius ir juridinius asmenis, tačiau toks sprendimas nėra tinkamas visais atvejais. Dėl neatvaizduojamo lygybės apribojimo „Kontaktai“ esybės atributams, objektai, kuriems nurodytas juridinio asmens tipas gali egzistuoti be nurodytos juridinio asmens informacijos (įmonės_pavadinimas, įmonės_kodas), kadangi atributai tarpusavyje nesusiję, taigi nurodžius vieną neprivalomą nurodyti likusių. Dėl nenurodyto poaibio apribojimo tarp esybės „Kontaktai“ atributų (pvm_kodas, įmonės_kodas), nėra tikrinama ar pvm kodas konkrečiu atveju gali būti nurodytas, kadangi pagal dalykinę sritį norint pateikti pvm kodą kontaktų esybės objektui privalo būti nurodytos savybės – juridinis asmuo, įmonės pavadinimas ir kodas. Taip pat UML modelyje nepažymėta prekės kodo unikalumo sąlyga, dėl ko negalėsime identifikuoti konkrečių prekių pagal jų kodus, kadangi jie gali dubliuotis. Verta paminėti ir prekių katalogo formavimą – esybę „Kategorijos“. Šiai esybei ORM 2 modelyje nurodytas žiedinis netransityvus, neciklinis apribojimas, kurio UML modelis atvaizduoti nepajėgia. Dėl apribojimo nebūvimo prekių katalogo vaikinėmis kategorijomis galės tapti jų tėvinės kategorijos, taip sudarant nepageidaujamą ciklą, taip pat tėvinės kategorijos galės turėti nuorodas į žemesnio lygio nei jų vaikinės kategorijos objektus, kas iš dalies griauia dalykinėje srityje aprašytą medžio struktūrą. Toks koncepcinis duomenų modelis, nors ir neužtikrina visų dalykinei sričiai reikalingų vientisumo reikalavimų, tačiau jo struktūra yra kompaktiškesnė bei paprasčiau perteikia būsimą duomenų struktūrą. Sukurtas modelis nėra perkrautas įvairiais grafiniais konceptais, dėl ko nėra toks painus, kaip ORM 2.

E-R koncepcinis duomenų modelis (28 pav.) perteikia tik pačius svarbiausius pirminio bei išorinio rakto, ir privalomo bei neprivalomo atributų vientisumo reikalavimus. Sukurtas koncepcinis modelis kaip ir UML neatvaizduoja tų pačių, bei keleto papildomų vientisumo reikalavimų. Esiybėms „Prekių paveikslai“ ir „Kategorijų paveikslai“ nenurodytas vidinio dažnumo apribojimas, dėl šios priežasties neribojamas konkretios prekės arba kategorijos turimų paveikslų kiekis. Taip pat neatvaizduojamas ir potipio apribojimas apibendrinimo ryšiui. Dėl šios

priežasties esybės „Kontaktai“ objektai gali egzistuoti nepriklausomai nuo konkretaus tipo esybių: „Vartotojai“, „Pirkėjo informacija“ ir „Pristatymo informacija“. Taigi E-R modelis atvaizduoja tik duomenų modelio karkasą: esybes, atributus bei ryšius tarp esybių. Taip sukuriant labiau eskizinį dalykinės srities modelį.

Praktinio koncepcinio duomenų modeliavimo metodų išbandymo metu pastebėta:

- ORM 2 modeliavimo metodas (26 pav.):
 - Teigiami aspektai:
 - Modelis išraiškingiausias ir pilnai perteikia modeliuojamą dalykinę sritį;
 - Perteikia daugiausia duomenų vientisumo reikalavimų;
 - Lengva verbalizacija;
 - Laisvųjų atributų (angl. Attribute-free) struktūra leidžia tiksliau apibrėžti ryšius tarp skirtingų atributų ar esybės;
 - Neigiami aspektai:
 - Modelis gana didelis dėl laisvųjų atributų struktūros. Net nedideliai dalykinei sričiai sudarytas modelis sunkiai telpa A4 formato lape, dėl ko susiduriama su nepatogumais dokumentuojant bei pernešant tokio tipo modelius
 - Reikalauja gana daug dalykinės srities patirties siekiant perprasti sukurtą modelį, dėl ko paprastam, nepatyrusiam vartotojui toks modelis tampa nepriimtiniu, sunkiai skaitomu;
 - Dideli modeliavimo kaštai - net ir paprastam modeliui reikalingas patyręs analitikas, bei reikia daugiau laiko nei kitais modeliavimo metodams;
- UML modeliavimo metodas (27 pav.):
 - Teigiami aspektai:
 - Perteikiama aiškesnė duomenų struktūra – esybės viduje vaizduojami atributai padeda lengviau suprasti duomenų bazės lentelių struktūrą;
 - Komapktiškas modelis;
 - Norint pasinaudoti koncepciniu modeliu nereikalaujama didelė patirtis. Paprasti vartotojai gali lengviau perprasti sukurtą modelį;
 - Modeliavimas reikalauja mažesnių kaštų (laiko ir kainos). Modelis kuriamas daug greičiau nei ORM 2;
 - Neigiami aspektai:
 - Neužtikrinta verbalizacija;

- Dėl per pus mažesnio atvaizduojamų vientisumo reikalavimų kiekio lyginant su ORM 2, negali perteikti visų dalykinės srities reikalavimų;
 - Neužtikrina duomenų vientisumo, gautas duomenų modelis neleidžia patikrinti ar jis tik teisingus dalykinei sričiai duomenų rinkinius;
- E-R modeliavimo metodas (28 pav.):
 - Teigiami aspektai:
 - Kompaktiškas modelis;
 - Nereikalauja daug patirties nei iš programuotojo ar analitiko nei iš vartotojo;
 - Modelis kuriamas greitai ir patogiai. Leidžia greit atvaizduoti būsimos duomenų bazės struktūrą;
 - Minimalūs modeliavimo kaštai;
 - Neigiami aspektai:
 - Perteikia tik būtiniausius vientisumo reikalavimus;
 - Neužtikrina verbalizacijos mechanizmo;
 - Negali detaliai perteikti modeliuojamos dalykinės srities;
 - Neužtikrina duomenų vientisumo;

3. Praktinės rekomendacijos

Remiantis koncepcinio duomenų modeliavimo metodų analizės apibendrinimu (1.5. skyrius) bei atlikus praktinį modeliavimo metodų išbandymą (2. skyrius) suformuluotos rekomendacijos koncepcinio duomenų modeliavimo metodo pasirinkimui:

- ORM 2 modeliavimo metodas – tinka didelės apimties ir didelio duomenų tikslumo koncepciniame lygyje reikalaujantiems projektams. Mažesnės apimties projektams toks modeliavimo metodas yra tiesiog per brangus;
- UML modeliavimo metodas – tinka tiek didelės, tiek mažos apimties projektams. Nereikalauja didelių išlaidų, tačiau negali užtikrinti išraiškingo duomenų modelio sudarymo apimančio visų reikalingų vientisumo reikalavimų;
- E-R modeliavimo metodas – tinka mažesnės apimties projektams arba verslo logikai apibūdinti. Dėl paprastos struktūros yra lengvai perprantamas ir nereikalauja didelių išlaidų;

Išvados

1. Darbe nagrinėjamų koncepcinio modeliavimo metodų analizė pagal pasirinktus kriterijus parodė, kad:
 - ORM 2 modeliai yra tiksliai ir paprastai verbalizuojami (pateikiami vientisais teiginiais), kuo nepasižymi likę modeliavimo metodai.
 - ORM 2 modeliai yra ekspresyviausi, nes turi daugiausiai grafiškai žymimų elementų bei pateikia ryšį tarp atributo ir esybės, nurodant atliekamas roles.
 - Vaizdinio vientisumo reikalavimų žymėjimo požiūriu išraiškingiausias yra ORM 2 metodas. ORM 2 apima išsamiausią vientisumo reikalavimų aibę, kurios poaibius nagrinėja kiti metodai;
 - Skirtingi metodai naudoja skirtingų tipų vientisumo reikalavimus ir nei vienas metodas neapima visų vientisumo reikalavimų tipų;
 - Dėl atributų vaizdavimo kaip esybės ypatybių E-R ir UML modeliuose, lyginant su ORM 2, yra prarandama dalis vientisumo reikalavimų;
2. Eksperimentinis metodų išbandymas iliustracinei dalykinei sričiai patvirtino, kad:
 - ORM 2 duomenų modeliai yra gana dideli, painūs, turintys daug linijų. Taip pat reikalauja daug dalykinės srities patirties dėl ko paprastam vartotojui toks modelis nepriimtinas;
 - UML duomenų modeliai, nors ir negali perteikti tiek dalykinės srities reikalavimų kaip ORM 2 metodas, tačiau sukurti modeliai aiškiau perteikia duomenų struktūras, yra kompaktiškesni bei reikalauja mažesnių modeliavimo kaštų;
 - E-R duomenų modeliai yra intuityviausi bei kompaktiškiausi palyginus UML ir ORM 2, tačiau šiuo metodu yra modeliuojami labiau eskiziniai dalykinės srities modeliai, skirti išsiaiškinti, o ne atvaizduoti visus dalykinės srities reikalavimus;
3. Kadangi ORM 2 yra sudėtingas, o E-R perteikia tik pagrindinius modeliuojamus konceptus, jų savybės ir tinka labiau eskiziniam dalykinės srities modeliui, koncepcinių duomenų modelių atvaizdavimui tikslingiausia pasirinkti UML.

Rezultatai

1. Pateiktas koncepcinio modeliavimo metodų: ORM 2, UML ir E-R palyginimas pagal pasirinktus kriterijus;
2. Atliktas eksperimentinis koncepcinio modeliavimo metodų išbandymas konkrečiai dalykinei sričiai;
3. Pateiktos rekomendacijos tinkamiausio koncepcinio modeliavimo metodo pasirinkimui kuriamai informacinei sistemai.

Šaltiniai

- [Bar05] Romas Baronas, Duomenų bazių valdymo sistemos, TEV, 2005, p. 41-42, 71-78.
- [Bla13] Michael Blaha, UML Database Modeling Workbook, Technics Publications, 2013, p. 11-12.
- [Hay08] David Hay, UML as a Data Modeling Notation, 2008.
[žiūrėta 2016-03-25]. Prieiga per internetą:
<<http://tdan.com/uml-as-a-data-modeling-notation-part-1/8457>>
- [Hal05] Terry Halpi, ORM 2 Graphical Notation, 2005.
[žiūrėta 2016-04-06]. Prieiga per internetą:
<http://www.orm.net/pdf/orm2_techreport1.pdf>
- [Hal11] Terry Halpin, ORM 2 Graphical Notation, 2011
[žiūrėta 2016-04-02]. Prieiga per internetą:
<<http://www.orm.net/pdf/ORM2GraphicalNotation.pdf>>
- [Hal98a] Terry Halpin, UML data models from an ORM perspective: part 1, 1998.
[žiūrėta 2016-03-15]. Prieiga per internetą:
<<http://www.orm.net/pdf/icmarticle1.pdf>>
- [Hal98b] Terry Halpin, UML data models from an ORM perspective: part 3, 1998.
[žiūrėta 2016-03-15]. Prieiga per internetą:
<<http://www.orm.net/pdf/ICMArticle3.pdf>>
- [Hal98c] Terry Halpin, UML data models from an ORM perspective: part 7, 1998.
[žiūrėta 2016-03-15]. Prieiga per internetą:
<<http://www.orm.net/pdf/orm-jcm7.pdf>>
- [HC06] Terry Halpin, Matt Curland, ORM 2 Constraint Verbalization Part 1, 2006.
[žiūrėta 2016-03-29]. Prieiga per internetą:
<http://www.orm.net/pdf/orm2_techreport2.pdf>
- [HM10] Terry Halpin, Tony Morgan, Information Modeling and Relational Databases, Morgan Kaufmann, 2010, p. 111-121, 225-229, 238-253, 271-288, 372-375, 379-382.
- [Kee07] Maria Keet, Mapping the Object-Role Modeling language ORM2 into Description Logic language, 2007.
[žiūrėta 2016-04-15]. Prieiga per internetą:
<<http://www.inf.unibz.it/krdp/pub/TR/KRDB07-2.pdf>>
- [Kry04] Aleksandras Krylovas, Diskrečioji matematika, Technika, 2004, p. 8.
- [MTH05] Robert Meersman, Zahir Tari, Pilar Herrero, On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops, Springer, 2005, p. 677.
- [OMG15] Object Management Group, OMG Unified Modeling Language TM (OMG UML), Version 2.5, 2015, p. 117 – 124.
[žiūrėta 2016-05-18]. Prieiga per internetą:
<<http://www.omg.org/spec/UML/2.5/PDF>>

- [PN08] Bronius Paradauskas, Lina Nemuraitė, Duomenų bazių semantiniai modeliai, Mokslo aidai, 2008, p. 9-13, 50-53.
- [WK03] Jos Warmer, Anneke Kleppe, The Object Constraint Language – Getting Your Models Ready for MDA, Addison-Wesley Professional, 2003, p. 62-63.

Santrumpos

- UML – Vieninga modeliavimo kalba (angl. Unified Modeling Language).
- E-R – esybių-ryšių duomenų modelis (angl. Entity–relationship model);
- ORM – Objektų vaidmenų modeliavimas (angl. Object Role Modeling);
- ISO - Tarptautinė standartizacijos organizacija (angl. International Organization for Standardization);

Priedai

1 priedas. Kompaktinis diskas

Kompaktiniame diske yra pateikiami praktinio išbandymo metu paruošti originalaus dydžio koncepciniai duomenų modeliai:

- ORM_2_modelis.png – ORM 2 modeliavimo metodu paruoštas modelis;
- UML_modelis.png – UML modeliavimo metodu paruoštas modelis;
- E-R_modelis.png – E-R modeliavimo metodu paruoštas modelis;

Taip pat skirtinguose kataloguose pateikiami programiniai šių modelių failai:

- ORM_2_modelis – ORM 2 modeliavimo metodu paruošto modelio programiniai failai;
Naudotas įrankis – Visual Studio 2013 su Norma įskiepiu;
- UML_modelis – UML modeliavimo metodu paruošto modelio programiniai failai;
Naudotas įrankis – MagicDraw 18.3 su Cameo Data Modeler 18.3 įskiepiu;
- E-R_modelis – E-R modeliavimo metodu paruošto modelio programiniai failai;
Naudotas įrankis – MS Visio 2016.