

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
Informacijos sistemų katedra

Magistro darbas

UAB „GNT Lietuva“ duomenų integravimo posistemio reinžinerija

Magistrantas: I.Kungytė

Vadovas: Prof. L.Nemuraitė

KAUNAS, 2010



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
Informacijos sistemų katedra

Indrė Kungytė

UAB „GNT Lietuva“ duomenų integravimo posistemio reinžinerija

Magistro darbas

Recenzentas

Doc. Antanas Lenkevičius

2010-05-25

Vadovas

prof. Lina Nemuraitė

2010-05-27

Atliko

IFM-4/4 gr. mag.

Indrė Kungytė

2010-05-27

KAUNAS, 2010

Turinys

REENGINEERING OF DATA INTEGRATION SUBSYSTEM IN JSC "GNT LIETUVA"	5
1. ĮVADAS	6
2. DUOMENŲ INTEGRAVIMO PROCESŲ REINŽINERIJOS GALIMYBIŲ ANALIZĖ	9
2.1 REINŽINERIJOS SAŲOKOS	9
2.2 DUOMENŲ INTEGRAVIMO PRINCIPAI	10
2.3 ESAMŲ ĮMONĖS DUOMENŲ INTEGRAVIMO PROCESŲ ANALIZĖ	14
2.4 PROJEKTAVIMO ŠABLONŲ ANALIZĖ LITERATŪROS ŠALTINIUOSE	14
2.5 DUOMENŲ INTEGRAVIMO TECHNOLOGIJŲ ANALIZĖ	15
2.5.1 Duomenų integravimas SAP Business Objects aplinkoje	15
2.5.2 Duomenų integravimas Oracle aplinkoje	17
2.5.3 Duomenų integravimas MS SQL 2005 Server aplinkoje	18
2.6 SIEKIAMAS DUOMENŲ INTEGRAVIMO PROCESŲ REINŽINERIJOS SPRENDIMAS	19
2.7 ANALIZĖS IŠVADOS	21
3. DUOMENŲ INTEGRAVIMO PROCESŲ METODIKA	23
3.1 DUOMENŲ INTEGRAVIMO ŠABLONO KRITERIJAI	23
3.2 DUOMENŲ INTEGRAVIMO ŠABLONAS	23
4. PATOBULINTO DUOMENŲ INTEGRAVIMO POSISTEMIO REIKALAVIMŲ SPECIFIKACIJA	26
4.1 FUNKCINIAI DUOMENŲ INTEGRAVIMO POSISTEMIO REIKALAVIMAI	26
4.2 NEFUNKCINIAI DUOMENŲ INTEGRAVIMO PROCESO REIKALAVIMAI	37
4.3 DUOMENŲ INTEGRAVIMO PROCESO DALYKINĖS SRITIES MODELIS	38
4.4 DUOMENŲ INTEGRAVIMO PROCESO ARCHITEKTŪRA	41
4.4.1 Duomenų integravimo loginė architektūra	41
4.4.2 Vartotojo paslaugos	41
4.4.3 Veiklos paslaugos	42
4.5 PAGRINDINĖ DUOMENŲ INTEGRAVIMO PROCESO ANALIZĖS DIAGRAMA	43
4.6 PANAUDOJIMO ATVEJŲ REALIZACIJOS	45
4.7 DUOMENŲ INTEGRAVIMO ELGSENOS MODELIS	51
4.8 DUOMENŲ BAZĖS SCHEMA	59
5. DUOMENŲ INTEGRAVIMO POSISTEMIO REALIZACIJA IR TESTAVIMAS	64
5.1 DUOMENŲ INTEGRAVIMO POSISTEMIO REALIZACIJOS MODELIS	64
5.2 DUOMENŲ INTEGRAVIMO PROJEKTŲ APLINKOS	65
5.2.1 MS SQL Server 2000 aplinka	65
5.2.2 MS SQL Server 2005 aplinka	66
5.3 DUOMENŲ INTEGRAVIMO PROCESO VEIKIMO APRAŠAS	68
5.4 REALIZUOTI DUOMENŲ INTEGRAVIMO MODELIAI	68
5.4.1 Plokščias modelis MS SQL Server 2000 aplinkoje	68
5.4.2 Struktūrizuotas modelis MS SQL Server 2000 aplinkoje	73
5.4.3 Struktūrizuotas modelis MS SQL Server 2005 aplinkoje	76
5.4.4 Duomenų integravimo modelių apibendrinimas	81
5.5 DUOMENŲ INTEGRAVIMO PROJEKTŲ TESTAVIMAS	81
5.5.1 Testavimo modelis	82
5.5.2 Integravimo proceso duomenys, DI proceso pavyzdys	82
5.5.3 SSIS ir DTS procesų diegimo planas	83
6. DUOMENŲ INTEGRAVIMO MODELIŲ EKSPERIMENTINIS TYRIMAS IR ĮVERTINIMAS	93

7. IŠVADOS	98
8. LITERATŪRA.....	99
9. TERMINŲ IR SANTRAUKŲ ŽODYNAS	101
10. PRIEDAI	102
10.1 PRIEDAS 1. MS SQL SERVER 2000 APLINKA	102
10.2 2 PRIEDAS. MS SQL SERVER 2005 APLINKA	106
10.3 3 PRIEDAS. INSERT_MASTER_DBM MS SQL UŽKLAUSA.....	118
10.1 3 PRIEDAS. STRAIPSNIS	121

Reengineering of Data Integration Subsystem in JSC "GNT Lietuva"

SUMMARY

The need to share the ever-increasing amounts of data calls for a reliable data integration scheme. Volume of data used in everyday practice grows together with the development of any company, and eventually a requirement to have all valuable data in one secure centralized database emerges. Rapid, efficient and accurate data transfer from remote subsidiaries to a central database where further analysis can be performed is the issue that this research aims to solve.

The research area of thesis covers various data extraction, transfer and integration methods and technologies; the main object of the research is the process of transferring data from remote subsidiaries and integrating it into one central database currently active in JSC "GNT Lietuva".

The goal of this research is to move data integration (DI) processes into a new technological environment and upgrade them without interrupting active daily DI process, ultimately creating a flexible data integration model (pattern), which could be reused in the future. Following tasks were carried out in order to achieve this goal: analysis of reengineering and data integration principles as well as new integration technologies; investigating their adaptability to the current DI processes and their improvement; implementing integration solutions and experimentation to verify the efficiency of new DI processes, and, finally, construction of a flexible integration solution.

The final generated solution was formalized as a data integration pattern. Conclusions drawn from the experiment accomplished in JSC "GNT Lietuva" indicate that practical application of the pattern reduced overall duration of the DI process by 45,4%, whilst the additional application of the SSIS technology resulted in duration decrease of 81,99%. The data integration process became more flexible and new data sources can be easily incorporated from now on.

Keywords - ETL, data integration, reengineering, patterns design.

1. Įvadas

Greitai augančiame ir besikeičiančiame pasaulyje atsiranda poreikis iš esamų duomenų greitai gauti rezultatus, be to, pasitikėjimas rezultatais yra svarbus kaip niekada anksčiau. Efektyvus duomenų integravimas tapo efektyvaus sprendimų priėmimo pagrindu.

Duomenų integravimas pasireiškia vis didėjančiu poreikiu dalintis augančiais duomenų kiekiais. Šiame darbe tiriama įmonės „GNT Lietuva“ padaliniai įsikūrę skirtingose Europos šalyse. Įmonė klesti, auga duomenų kiekiai, atsiranda poreikis turėti duomenis vienoje duomenų bazėje. Todėl pagrindinė šiame tyrime sprendžiama **problema** – greitas, efektyvus, tikslus duomenų perkėlimas iš nutolusių padalinių į centrinę duomenų bazę, kurioje atliekama jų analizė. Tai siekiama padaryti taikant naujas duomenų integravimo technologijas. Duomenims perkelti ir integruoti taikomi ETL (angl. *Extract, Transform, Load*) įrankiai, kurie leidžia išgauti, transformuoti ir įkelti duomenis. Įmonės taiko ETL įrankius perkelti duomenis iš įvairių šaltinių, pakeisti formatą ir įkelti juos į kitas duomenų bazines, duomenų centrus, analizės saugyklas ar į kitas operacines sistemas verslo procesams vykdyti. Šio darbo **tyrimo sritis** – duomenų išgavimo, perkėlimo ir integravimo metodai ir technologijos, **tyrimo objektas** - UAB „GNT Lietuva“ duomenų perkėlimo iš nutolusių padalinių ir integravimo į centrinę duomenų bazę procesas.

UAB „GNT Lietuva“ prekiauja didmenine IT ir buitine technika. Įmonė sudaryta iš penkių padalinių, esančių skirtingose valstybėse. Kiekvieną dieną duomenys iš jų yra keliami į vieną duomenų bazę, kurioje jie sujungiami ir suvienodinami – atliekamas duomenų jungimas. Tyrimo pradžioje tai buvo įgyvendinta *MS SQL Server 2000* duomenų transformavimo DTS (angl. *Data Transform Services*) funkcijomis. Sukurta transakcijų sistema, kuri perkeldavo ir transformuodavo duomenis iš įmonės padaliniuose esančių vidinių duomenų bazių serverių ir vykdė duomenų replikavimą.

Šiuo metu įmonė atnaujina savo procesus ir pereina prie naujos technologijos, o tam visus duomenų integravimo procesus reikia perkelti į naują aplinką. Duomenų integravimo sistemą sudaro apie 100 duomenų integravimo procesų. Kiekvienas šios sistemos procesas turi griežtai nustatytą vykdymo laiką ir dažnumą. Daugelis - iš jų pasižymi sudėtinga duomenų integravimo logika, todėl procesų perkėlimas tampa sudėtingu uždaviniu.

Taigi, šio **darbo tikslas** yra perkelti duomenų integravimo procesus į naują technologinę aplinką ir juos patobulinti, nenutraukiant kasdieninio duomenų integravimo proceso ir sukuriant lankstų duomenų integravimo modelį (šabloną), kurį būtų galima taikyti įmonei plečiantis ateityje. Šiam tikslui pasiekti reikia įvykdyti šiuos **uždavinius**:

- išanalizuoti reinžinerijos ir duomenų integravimo principus bei naujų integravimo technologijų galimybes;
- pritaikyti naujas technologijas įmonės duomenų integravimo procesams kartu juos patobulinant ir padarant efektyvesniais;
- atlikti eksperimentą, kuris leistų patikrinti naujų integravimo procesų efektyvumą;
- sudaryti lankstų integravimo sprendimą, kurį įmonė galėtų taikyti naujiems duomenų šaltiniams įjungti ateityje.

Šiame darbe išanalizuoti duomenų integravimo metodai, reinžinerijos principai bei naujų technologijų galimybės buvo pritaikyti praktiškai, atliekant įmonės duomenų integravimo posistemio reinžineriją. Reinžinerija – egzistuojančios sistemos analizės ir modifikavimo procesas, atliekamas siekiant pertvarkyti sistemą. Reinžinerijos metu neužtenka sukurti analogiškus procesus naujai pasirinkta technologija. Didžiausias reinžinerijos efektas pasiekiamas tada, kai veiklos procesams ne tik pritaikomos naujos technologijos, bet jie kartu patobulinami ar optimizuojami.

Analizuojant įmonės duomenų integravimo procesus pastebėta, kad šie procesai vyksta ilgai, naudoja daug atminties ir serverio resursų, todėl buvo ieškoma būdų jiems pagreitinti. Senieji duomenų integravimo procesai buvo vykdomi dviejose skirtingose aplinkose: *MS SQL Server 2000* ir *MS SQL Server 2005* terpėse. Norint įvertinti, kurioje platformoje duomenų integravimo procesai yra greitesni, stabilesni, sunaudoja mažiau resursų, buvo sukurti trys projektai. Pirmajame projekte informacija buvo jungiama iš lokalių duomenų bazės lentelių ir iš jų sugeneruojamas *XML* dokumentas. Antrajame projekte buvo patobulinta DB lentelių struktūra ir *XML* dokumento generavimas. Trečiajame projekte analogiškas procesas buvo realizuotas *Microsoft SQL Server 2005* integravimo paslaugomis *Integration Services (SSIS)*.

Kol kas įmonėje veikia ir senoji, paremta DTS, integravimo sistema, kadangi ji leidžia patikrinti duomenų integravimo procesų teisingumą. DTS procesas senojoje sistemoje yra ištestuotas, todėl juo užkrauti duomenys turi visiškai sutapti su duomenimis, perkeltais naujuoju *SSIS* procesu. Ateičiai, kai senasis procesas neveiks, sukurtas testavimo modelis, kadangi integruojami duomenys bus testuojami ir veikimo metu. Testavimo modelis apima logines proceso taisykles ir skirtumų radimą tarp dviejose kompanijose esančių produktų informacijos.

Atlikta senojo DTS proceso analizė parodė, kad perkeliamos duomenų struktūros turi pasikartojančių duomenų, kuriuos galima atskirti ir patobulinti perkėlimo procesą. Ši analizė taip pat atskleidė aktualias problemas: tikslinga sukurti tokius perkėlimo procesus ir duomenų struktūras, kad būtų lengva jas papildyti įtraukiant naujus duomenų šaltinius. Pritaikius naujas duomenų struktūras, proceso vykdymo laikas sutrumpėjo, procesas tapo lankstesnis įvairiems pasikeitimams.

Sukurtas **sprendimas** buvo formalizuotas **duomenų integravimo šablono** pavidalu. Šablonai (angl. *patterns*) yra būdas aprašyti programinės įrangos problemų sprendimus [16]. Jų tikslas yra formalizuoti sprendimus, kuriuos galima taikyti projektuojant panašius procesus, juos vystant. Šablono aprašas turi būti išsamus, aiškus, motyvuotas. Šablonai gali turėti savitą struktūrą, tačiau jų pagrindiniai etapai yra vienodi.

Eksperimentas parodė, kad šablono taikymas UAB „GNT Lietuva“ įmonėje sutrumpino proceso trukmę 45,4%, o kartu pritaikytos SSIS technologijos dėka gautas 81,99% sutrumpėjimas. Duomenų integravimo procesas tapo lankstesnis, į jį bus galima paprasčiau įjungti naujus duomenų šaltinius. Eksperimentas buvo atliekamas realioje įmonės aplinkoje įvairiais laiko momentais, esant skirtingiems serverio apkrautumams. Visi procesai buvo bandomi su tais pačiais duomenimis, siekiant išvengti galimos duomenų rinkinių įtakos procesų vykdymo laikui.

Darbo struktūra. Darbas sudarytas iš įvado, kuriame atskleista pagrindinė tyrimo idėja, santrumpų žodyno, kuriame pateikti santrumpų išsamūs aprašymai ir pagrindinės tyrimo dalies bei išvadų ir literatūros sąrašo. Pagrindinę dalį sudaro duomenų integravimo reinžinerijos galimybių analizė, duomenų integravimo procesų metodika, patobulinto duomenų integravimo posistemio reikalavimų specifikacija, duomenų integravimo posistemio realizacija ir testavimas, DI modelių eksperimentinis tyrimas ir įvertinimas.

Analizės skyriuje išanalizuotos reinžinerijos sąvokos, ETL ir duomenų integravimo principai, šablonų kūrimo ir sudarymo kriterijai. Peržiūrėta ir įvertinta įmonėje egzistuojanti duomenų integravimo sistema. Pateikta duomenų integravimo technologijų: *SAP Business objects*, *Oracle*, *MS SQL Server 2005* analizė. Technologijų privalumai ir trūkumai tiriamos įmonės atžvilgiu. Apibendrintas duomenų integravimo procesų reinžinerijos sprendimas.

Reikalavimų specifikacijos etape surinkti visi funkciniai ir nefunkciniai duomenų integravimo proceso reikalavimai. Sudarytas dalykinės srities modelis, pateikta proceso loginė architektūra. Suprojektuota duomenų bazė.

Realizacijos skyriuje sukurti ir ištestuoti trys duomenų integravimo modeliai. Sukurtas testavimo modelis, kuris bus naudojamas ateityje, kaip testavimo logikos pagrindas. Projektai sukurti realioje įmonės aplinkoje. Jų grąžinamas rezultatas vienodas, tačiau vidinė architektūra, technologija skirtinga. Pateiktas SSIS ir DTS paketų įdiegimo į duomenų integravimo sistemą planas. Duomenų integravimo modelių eksperimentiniame tyrimo ir įvertinime atliktas eksperimentas ir pritaikytas empiriniu būdu gautas bendrinių duomenų atskyrimo šablonas. Pagrįstas patobulinto ir perkelta į naują technologiją proceso pagreitėjimas.

Darbas buvo pristatytas kasmetinėje "Informacinė visuomenė ir universitetinės studijos (IVUS)" konferencijoje ir išspausdintas IVUS leidinyje.

2. Duomenų integravimo procesų reinžinerijos galimybių analizė

Greitai augančiame ir besikeičiančiame pasaulyje atsiranda poreikis iš esamų duomenų greitai gauti rezultatus, be to, pasitikėjimas rezultatais yra svarbus kaip niekada anksčiau. Efektyvus duomenų integravimas tapo efektyvaus sprendimų priėmimo pagrindu [2] [3].

Duomenų perkėlimas iš šaltinių, UAB „GNT Lietuva“ įmonėje, vyksta *MS SQL Server 2000* aplinkoje. Procesams įvykdyti naudojamos *MS SQL 2000 Server* funkcijos – *Data Transformations Services (DTS)*.

Nagrinėjama įmonė ketina atsinaujinti ir nori pereiti prie naujos technologijos, perkeliant visą duomenų integravimo procesų sistemą į naują aplinką. Duomenų integravimo sistemą sudaro apie 100 duomenų integravimo procesų. Daugelis iš jų pasižymi sudėtinga duomenų integravimo logika, todėl procesų perkėlimas tampa sudėtingu uždaviniu.

Įmonės darbuotojai naudoja ataskaitas, kurių duomenys yra imami iš bendros duomenų bazės užkrautos DTS procesais. Kadangi tiesioginis DTS vartotojas yra IT skyriaus darbuotojas, jis privalo surasti geriausią alternatyvą DTS ir siekti, kad sistemos vartotojai nepajustų sutrikimų keičiant esamą *MS SQL Server 2000* nauja technologija.

2.1 Reinžinerijos sąvokos

Reinžinerija - egzistuojančios sistemos analizės ir modifikavimo procesas, atliekamas kai norima pertvarkyti sistemą. Veiklos procesą sudaro rinkinys logiškai susietų užduočių, kurios atliekamos norint gauti apibrėžtą rezultatą. Tai apima viešųjų pirkimų, užsakymų įvykdymą, produktų vystymą, klientų aptarnavimą ir pardavimų veiklas.

Reinžinerijos sąvoka sudaryta iš 4 pagrindinių aspektų:

Novatoriškas mąstymas: procesas, kuris visiškai priklauso nuo kūrėjo kūrybiškumo ir įkvėpimo lygio.

Proceso funkcija: rinkinys veikų, kurios grąžina reikiamą rezultatą.

Radikalūs pakeitimai: radikalus organizacijos elemento pakeitimas, transformacija, kuri svarbi organizacijai palaikyti. Pokyčius įtakoja naujos idėjos, technologijos. Organizacija turi suprasti pokyčių svarbą ir būtinumą, mokytis efektyviai valdyti procesus.

Organizacijos vystymas ir vykdymas: Vienas iš būdų spręsti apie organizacijos vystymąsi yra jos palyginimas su kitomis kompanijomis. Tai parodo organizacijos lygį.

Organizacijoje atliekama sistemos ar proceso reinžinerija suskirstyta į etapus:

- verslo teorijos pergalvojimas.

- nebegaliojančių taisyklių panaikinimas ir sunkiai realizuojamų prielaidų atmetimas.
- tradicinių metodų atsisakymas ir išėjimas iš apibrėžtų organizacijos ribų.
- senų procesų perkūrimas informacinių technologijų pagalba, jų automatizavimas.
- didesnis dėmesys išoriniams klientams.

Naujų darbo aplinkų kūrimas. Davenport ir Short išskiria ir apibūdina 5 verslo proceso reinžinerijos žingsnius:

1 – verslo idėjos ir procesų plėtojimas. Verslo proceso reinžinerija yra verslo idėjos valdymas, kuri reiškia konkrečius verslo tikslus tokius kaip kaštų mažinimas, laiko trumpinimas, rezultato kokybės gerinimas.

2 – identifikuoti procesai turi būti pertvarkyti. Dauguma įmonių identifikuoja visus procesus ir tada juos surūšiuoja pagal poreikį, aktualumą.

3 – suprasti ir įvertinti egzistuojantį procesą: etapas reikalingas tam, kad išvengtų pasikartojančių senų klaidų ir objektyviai pateikti tolesnius patobulinimus.

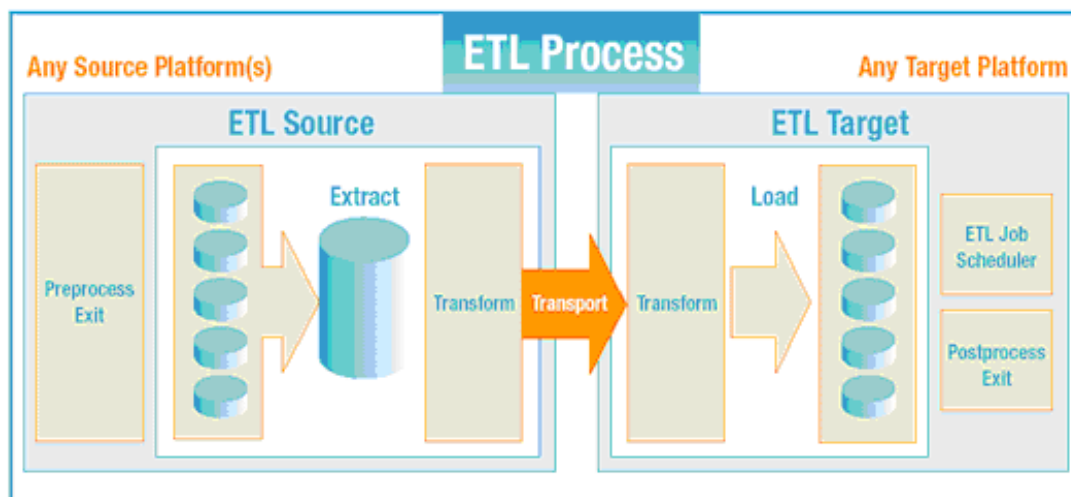
4 – nustatyti informacinių technologijų (IT) lygį.

5 – suprojektuoti ir kurti naujo proceso prototipą [18].

2.2 Duomenų integravimo principai

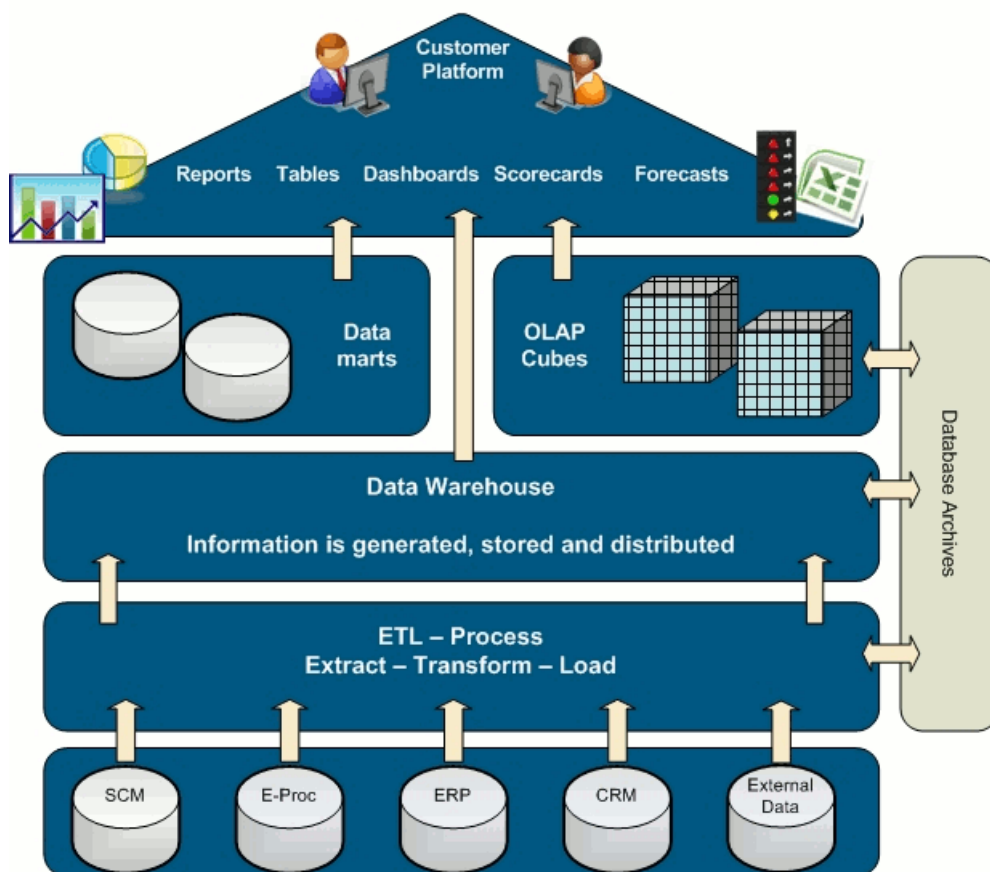
Pagrindinis duomenų integravimo principas – išgauti, transformuoti, įkelti duomenis. Tai ETL platformos pagrindas.

ETL (angl. *Extract, Transform, Load*) yra duomenų konsolidavimo įrankiai, leidžiantys išgauti, transformuoti ir užkrauti duomenis arba duomenų kopijas iš daugialypių platformų ir duomenų šaltinių į vieną saugyklą. 1 paveiksle pavaizduoti ETL žingsniai: duomenų išgavimas (*Extract*), duomenų transformavimas (*Transform*), duomenų užkrovimas (*Load*) [1].



1 pav. ETL procesų eiga

ETL procesai yra sistemos viduje, todėl vartotojams jiems nežinomi. Vartotojas naudojami ETL procesų gautais rezultatais – konsoliduotais duomenimis. 2 paveiksle pateiktas ETL procesų lygmuo įmonės infrastruktūroje.



2 pav. ETL procesai įmonių infrastruktūrose

Kiekvienas ETL etapas turi daug galimybių kaip apdoroti duomenis. Jų panaudojimas priklauso nuo proceso logikos ir sudėtingumo.

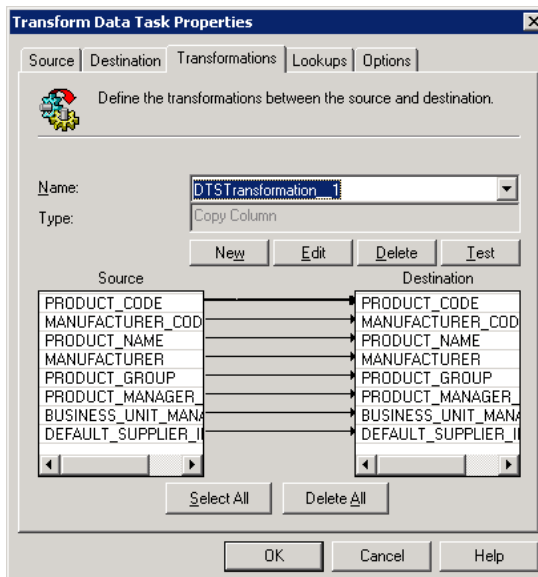
- *Duomenų išgavimas*

Duomenys išgaunami iš skirtingų vidinių ir išorinių šaltinių. Užklausos siunčiamos į šaltinio sistemą, suaktyvinant jungtį (angl. *Connections*). Paimti duomenys išsaugomi ta pačia struktūra kokia buvo šaltinyje.

Duomenų ištraukimo būdas priklauso nuo esamo šaltinio. Jungties (angl. *Connections*) tipas taip pat gali būti parinktas atitinkamai pagal turimą procesą. Naudingiausia ištraukti tik tuos duomenis, kurie tiesiogiai reikalingi sistemos funkcionalumui palaikyti. Papildomi, nereikalingi, duomenys užima vietą diske ir atmintį.

- *Duomenų transformavimas*

Duomenų transformavimo etape duomenys yra išgauti iš duomenų šaltinių ir patalpinti vienoje vietoje. Duomenų transformavimo pagrindinės funkcijos yra duomenų atitikčių nustatymas (*Mapping*), sujungimas, filtravimas, rūšiavimas, duomenų kokybės tikrinimas ir t.t. Duomenų atitikčių nustatymas yra duomenų struktūrų palyginimas tarp pirminio duomenų šaltinio struktūros ir galutinės. 3 paveiksle pateiktas *DTS Mapping* procesas *MS SQL 2000 Server* aplinkoje [10][11].



3 pav. Atitikmens suradimo procesas *MS SQL 2000 Server* aplinkoje

- *Duomenų įkėlimas*

Duomenų įkėlimui, į galutinę duomenų saugyklą, yra svarbus duomenų apimtys dydis ir duomenų įkėlimo greitis. Šiems kriterijams optimizuoti yra daug būdų:

- ✓ Veiksmų fiksavimo (angl. *logging*) įjungimas – kuomet fiksavimas įjungtas, sistema sukuria transakcijų įrašų dokumentą, kuriame įrašo kiekvieną įvykusį veiksmą. Patogu stebėti ir analizuoti kiekvieno veiksmo vykdymo trukmę.
- ✓ Indeksų perkūrimas (angl. *recreate indexes*) - indeksavimas yra susijęs su duomenų įrašymu į duomenų bazę. Galima panaikinti indeksą ir sukurti naują tokį pat kaip buvo, kuomet duomenų įkėlimas yra baigtas. Tai padarys indeksavimą paketiniu apdorojimo procesu (angl. *batch process*) vietoj atliktos veiklos su kiekvienu naujai pridėdamu įrašu į sistemą. Šis lentų perindeksavimas pagreitina duomenų įkėlimą, tačiau ne visais atvejais. Jei duomenų bazės lentelė yra maža ir papildoma tik apie 2% lentos dydžio, tai perindeksavimas gali užtrukti ilgiau, nei pats duomenų įkėlimas.
- ✓ Failo surūšiavimas pagal indeksuotą pirminį raktą – tai pagreitina indeksavimo procesą. Tai viena iš labiausiai rekomenduojamų metodikų. Jeigu reikia sukurti indeksą kitam

laukui (kuris nėra pirminis raktas), esamas indeksas ištrinamas ir sukuriamas naujas vėliau.

- ✓ Subalansuotas indeksų skaičius – per didelis indeksų skaičius padidina duomenų įkėlimo trukmę, sumažina greitį, prailgina vartotojo jungimosi į duomenų bazę laiką.
- ✓ Duomenų bazės lentelės papildymas vietoj pilno jos atnaujinimo – naujų įrašų pridėjimas užima mažiau laiko lyginant su duomenų bazės lentelės perkūrimu nuo nulio. Tačiau, jeigu yra pridedama per daug naujų įrašų ir jų apimtis didelė, tai greičiau yra ištrinti ir perkurti.
- ✓ Limitas atnaujinimo dažnumui – atnaujinimo procesas užima labai daug laiko, todėl reikia pasirinkti maksimalų atnaujinimo periodą.
- ✓ Paralelinės lentelės užkrovimas ir indeksų palaikymas – indeksų išmetimo ir didelių duomenų užkrovimo procesų sugretinimas (kai abu procesai veikia lygiagrečiai, tuo pačiu metu) žymiai pagreitina duomenų įkėlimą. Dauguma duomenų saugyklų platformų teikia šia galimybę [11].

- *Duomenų profiliavimas ir kokybės kontrolė*

Duomenų profiliavimas suteikia tiesioginį įžvalgumą duomenų kokybei. Jis gali parodyti, kiek duomenų eilučių turi dingusias arba neatitinkančias formato reikšmes. Atliekant duomenų profiliavimą prieš ETL proceso projektavimą, reikia įsitikinti, kad duomenų bazės struktūra, į kurią bus keliami duomenys, yra tinkama.

- *Meta-duomenų valdymas*

Informacija apie duomenis, kurie yra apdoroti, dažnai dedama į meta - duomenų saugyklą (duomenų bazė, kuri talpina visus metaduomenis). Visas ETL procesas gali būti kontroliuojamas metaduomenų valdymu. Tačiau šiuo metu minimaliai teikiamos duomenų integravimo ir perkėlimo taisyklės [12].

ETL platformos pagrindu yra sukurta daug įrankių, kurie diegiami įvairiose duomenų bazių sistemose. DI įrankis yra brangus. Jo kainos pagrindą sudaro įrankio licencija ir įrankio palaikymo aparatūros kaina. Neatsižvelgiama į duomenų kiekį įmonėje [12]. 1 lentelėje pateikti ETL įrankiai ir jų užsakovai.

1 lentelė. ETL įrankių sąrašas

Eil. Nr	ETL įrankio pavadinimas	ETL įrankio užsakovas
1.	Oracle Warehouse Builder (OWB)	Oracle
2.	Data Integrator & Services (BODI)	Business Objects, SAP
3.	IBM Information Server (Ascential)	IBM

4.	SAS Data Integration Studio	SAS Institute
5.	PowerCenter	Informatica
6.	Elixir Repertoire	Elixir
7.	Integration Services (SSIS)	Microsoft
8.	Data Migrator	Information Builders
9.	Talend Open Studio	Talend
10.	DataFlow Manager	Group 1 Software (Sagent)
11.	Data Integrator	Pervasive
12.	Transformation Server	IBM DataMirror
13.	Transformation Manager	ETL Solutions Ltd.
14.	Data Manager/Decision Stream	IBM Cognos
15.	DT/Studio	Embarcadero Technologies
16.	ETL4ALL	IKAN
17.	DB2 Warehouse Edition	IBM
18.	Pentaho Data Integration	Pentaho

2.3 Esamų įmonės duomenų integravimo procesų analizė

UAB „GNT Lietuva“ prekiauja didmenine IT ir buitine technika. Įmonė sudaryta iš 5 padalinių, esančių skirtingose valstybėse. Kiekvieną dieną duomenys iš jų yra keliami į vieną duomenų bazę, kuri sujungia ir suvienodina juos – atliekamas duomenų jungimas. Šiuo metu tai yra įgyvendinama *MS SQL Server 2000* DTS funkcijos pagalba.

Naudojant duomenų transformavimo paslaugas (DTS) sukurta transakcijų sistema, į kurią įkeliami ir transformuojami duomenys iš įmonės padaliniuose esančių vidinių DB serverių. Įmonėje naudojamas DTS leidžia atlikti duomenų replikavimą. Be to, ši funkcija suteikia galimybę perkelti indeksus ir procedūras į SQL serverius, kurti paketus, kuriuose grupuojamos perkėlimo operacijos ir objektai. Operacijos paleidžiamos sinchroniškai (viena operacija laukia, kol kita pasibaigs) arba asinchroniškai, naudojant visą SQL serverio apkrovą.

ETL įrankis yra reikalingas įmonei, nes šie įrankiai veikia išgavimo – transformavimo – užkrovimo principu. Nors DTS nėra ETL įrankis, tačiau jis veikia tokiu pat principu. ETL įgalina perkelti duomenis iš vienos duomenų bazės į kitą, kai jų struktūra visiškai skirtinga, todėl ši savybė svarbi įmonėje esantiems duomenų integravimo procesams.

2.4 Projektavimo šablonų analizė literatūros šaltiniuose

Šablonai (angl. *patterns*) yra naujausias būdas sprendžiant programinės įrangos problemas [16]. Ši idėja išsiplėtota daugelyje sričių, įskaitant programavimo raštingumą. Šablono tikslas yra

formalizuotas dokumentas, kuriuo naudojantis galima projektuoti, vystyti procesus, objektus, sistemas ir kt. Šablono aprašas turi būti išsamus, aiškus, motyvuotas. Kiekvienas šablono dokumentas turi savitą struktūrą, tačiau jo pagrindiniai etapai yra vienodi.

Austrų architektas Christopher Alexander yra vienas iš formalizuoto šablono aprašo pradininkas. Jis teigia, kad šablonai yra praktiniai vadovai sprendžiant problemas, kurios nuolat kartojasi. Nors jo aprašyti šablonai yra skirti architektūros sričiai, projektuoti įvairioms teritorijoms ir objektams, tačiau pagrindiniais aspektais susidomėjo ir IT sritis dėl šių priežasčių:

- Nuolatiniam susidūrimams su tomis pačiomis ir panašiomis problemomis reikia sudaryti šablonus, kurie jas išsprendžia. Tuomet pasikartojus problemai nereikia galvoti ir programuoti iš naujo. Problema sprendžiama pasinaudojus sukurtu šablonu.
- Daugkartinis sukurtų modelių panaudojimas.
- Išsamus problemos ir jos sprendimo dokumentavimas [15].

Nepriklausomos ekspertės Linda Rising iš jungtinių Amerikos valstijų teigimu, šablonas yra tam tikra dokumento forma, kurio esmė sukauptų žinių ir patirties išsaugojimas.

Šablonas sudarytas iš 4 pagrindinių elementų:

1. Šablono pavadinimas – trumpas, aiškus, sudarytas iš vieno – dviejų žodžių, kurie apibūdina problemą ir jos sprendimą.
2. Problema – apibūdina, kada reikia taikyti šabloną ar metodą. Paaiškina problemą ir situaciją. Pateikiamas sąlygų sąrašas, kurios turi būti tenkinamos prieš pradėdant naudoti šabloną.
3. Sprendimas – apibūdina elementus, kurie dalyvauja problemos sprendimo metu, ryšius, priklausomybes, elementų tarpusavio bendravimą. Sprendimas neturi būti skirtas vienos problemos sprendimui. Šablonas – tai modelis, kuris turi būti taikomas daugelyje panašaus tipo situacijų.
4. Rezultatas – pateikiamas situacijos aprašymas po šablono taikymo. Apibūdinamas sistemos lankstumo, perkėlimo poveikis, šablono nauda, tęstinumas [16][17].

2.5 Duomenų integravimo technologijų analizė

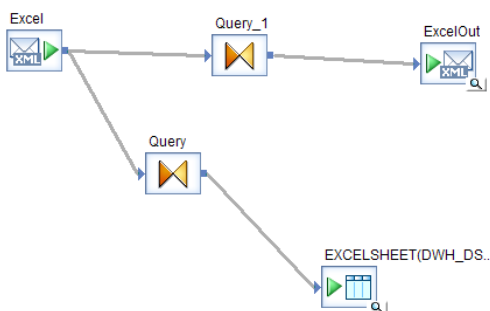
2.5.1 Duomenų integravimas SAP Business Objects aplinkoje

SAP BusinessObjects Data Integrator - ETL įrankis, leidžiantis lengvai tyrinėti, išgauti, transformuoti ir pristatyti duomenis į pasirinktą vietą, pasirinktu dažnumu. Patogi vartotojo sąsaja padidina programuotojo darbo našumą. Duomenų integravimas palaiko įvairias duomenų aplinkas.

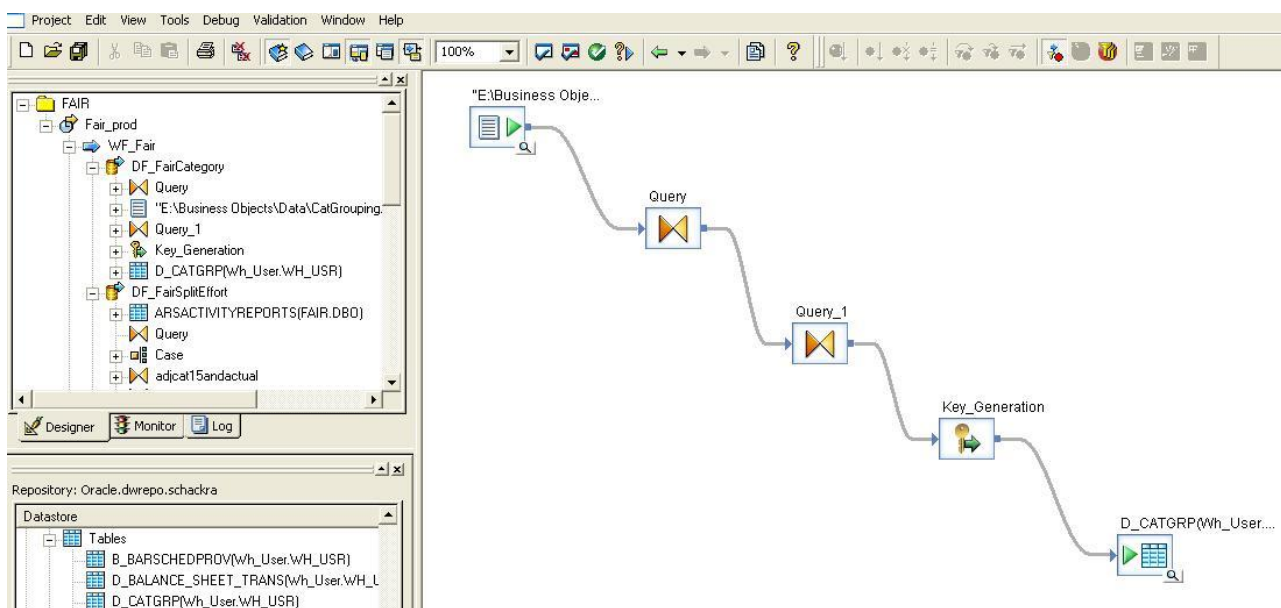
SAP BusinessObjects Data Integrator užtikrina įmonėje esančių duomenų kokybę su patobulintu duomenų profiliavimu, kuris leidžia suprasti turinį, struktūrą, ryšius tarp lentelių ir duomenų šaltinių. Duomenų patikros metu sukuriama filtras tarp duomenų šaltinio ir galutinės sistemos, kuris pašalina duomenis neatitinkančius verslo taisyklių. Įrankis suteikia galimybę stebėti, kaip pasikeitimai duomenų šaltinyje įtakoja verslo procesus, sistemą. Duomenų integratorius yra unikalus savo sugebėjimu pateikti lengvą integraciją su išoriniais duomenimis iš *SAP Business Objects* programinės įrangos verslo sumanumo (angl. *Business Intelligence*) sluoksnio.

Vartotojas turi galimybę matyti kada duomenys atnaujinami, kaip suskaičiuoti, iš kokio duomenų šaltinio užkrauti. Tai padeda įgyti jų pasitikėjimą, kad duomenys tvarkomi teisingai ir atitinka jų sudarytus reikalavimus.

Įrankis padeda pagreitinti ir supaprastinti programavimo pastangas kuriant ir valdant duomenų integravimo procesus vienoje grafinėje aplinkoje. „Paimk ir įdėk“ principas leidžia greitai pasirinkti reikiamus duomenų integravimui elementus, sujungti juos tarpusavyje. 4 - 5 paveiksluose pateikta „*Business Objects Data Service design*“ vartotojo sąsaja ir elementai.



4 pav. *SAP Business Objects Data Integrator* įrankio grafiniai elementai



5 pav. *SAP Business Objects Data Integrator* įrankio vartotojo sąsaja

2.5.2 Duomenų integravimas Oracle aplinkoje

Kompanijos privalo turėti galimybę integruoti didelius duomenų kiekius iš skirtingų sistemų. Mažinti duomenų integravimo kaštus ir pagerinti jo našumą.

Oracle Data Integrator Enterprise Edition užtikrina aukštą procesų vykdymo kokybę, duomenų transformavimą tarp platformų, realiu laiku, sinchroninių arba asinchroniniu režimu. *Oracle Data Integrator Enterprise Edition* yra visapusiška duomenų integravimo platforma, kuri apima visus duomenis tenkinančius aukšto lygio reikalavimus.

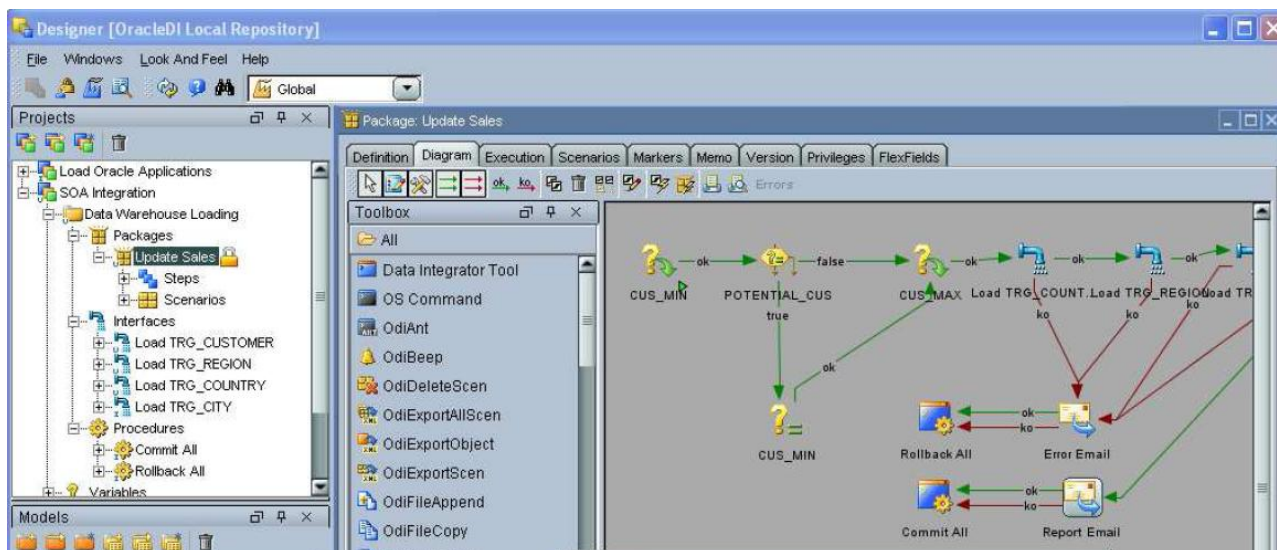
Oracle Data Integrator Enterprise Edition duomenų gavybos, perkėlimo, transformavimo architektūra sugretina skirtingus RDBMS (angl. *relational database management systems*) variklius siekiant vykdyti ir transformuoti duomenis. Tai optimizuoja vykdymą ir mažina sprendimo kainą. Įrankis generuoja paprastą kodą, iš esmės, nesuderinamiems RDBMS varikliams.

Šiuo metu *Oracle* palaiko du duomenų integravimo įrankius: "*Oracle Data Integrator*", "*Oracle Warehouse Builder Enterprise*". Ateityje planuojama abu įrankius sujungti. Ši strategija, *Oracle* įrankių klientams, padės greitai ir saugiai naudoti įrankį.

„*Oracle Warehouse Builder*“ – Šis įrankis sukurtas ETL platformos pagrindu. Visiškai integruotas reliacinis ir erdvinis modeliavimas. Jis palaiko duomenų integravimo ir duomenų valdymo veiklas:

- ✓ ETL duomenų saugyklai;
- ✓ Duomenų sujungimą iš įvairių duomenų šaltinių;
- ✓ Duomenų perkėlimą iš senų sistemų į naujas;
- ✓ Reliacinį ir erdvinį modeliavimą;
- ✓ Įmonės duomenų projektavimą ir valdymą;
- ✓ duomenų gryninimą ir tikrinimą;
- ✓ duomenų priežiūrą ir taisymą.

6 paveiksle grafiškai pavaizduota, duomenų integravimui skirtoje *Oracle* aplinkoje, vykstantis duomenų integravimo procesas.



6 pav. Oracle Data Integrator įrankio vartotojo sąsaja

2.5.3 Duomenų integravimas MS SQL 2005 Server aplinkoje

MS SQL Server 2005 yra išsami įmonėms skirta platforma, kurioje integruoti verslo tyrimų įrankiai, padedantys valdyti duomenis. *MS SQL Server 2005* modulis teikia saugią ir patikimą santykinį ir struktūrinių duomenų saugyklą, todėl galima lengvai kurti ir valdyti veikiančias verslo duomenų programas.

MS SQL Server 2005 suteikia galimybę perkelti duomenis iš vienos duomenų bazės į kitą SSIS (angl. *SQL Server Integration Services*) pagalba. Tai ETL įrankis. Projektavimo aplinka „*Business Intelligence Development Studio (IDS)*“ leidžia kurti SSIS paketus. IDS ir SSIS įrankiai visiškai suderinti, todėl jų tarpusavio konfliktų tikimybė maža.

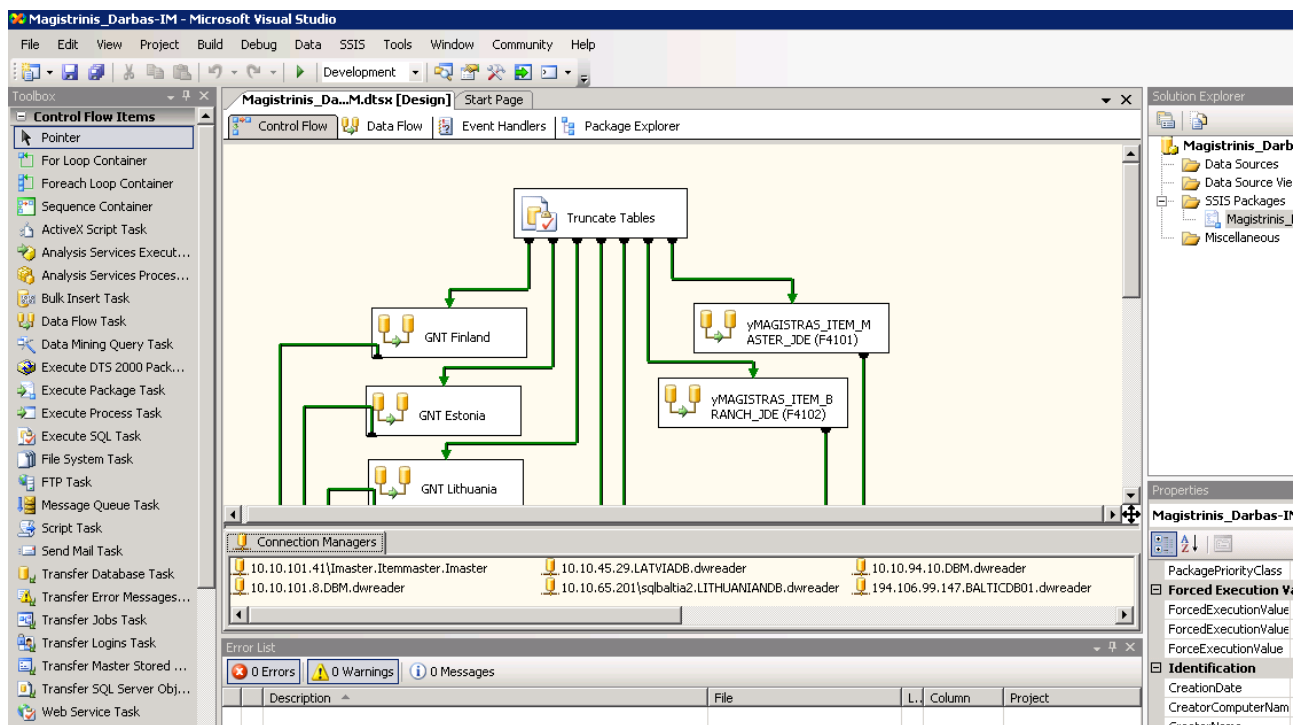
SSIS yra DTS analogas, tačiau tai ne funkcija kaip DTS *MS SQL Server 2000* versijoje, o paslauga, atskiras modulis, kuris įdiegiamas į SQL Server papildomai. Duomenų integravimas *MS SQL Server 2005* aplinkoje vyksta keliais etapais.

1 etapas - Pirmiausia sukuriama SSIS paketas, kuriame yra vykdomos visos operacijos: atitinkamų DB lentelių trynimasis, atnaujinimas, naikinimas, naujų kūrimas, duomenų perkėlimas, kuris gali būti apribotas įvairiomis sąlygomis.

2 etapas – proceso sukūrimas (angl. *Job*). Tai procesas, kurio paleidimas iškviečia SSIS paketo vykdymą. Jis yra būtinas, kai norime, kad procesas vyktų pastoviu režimu.

3 etapas - sukurtas paleidimo procesas (angl. *Job*) gali būti paleidžiamas nustatytu režimu. Paleidimo procesui nustatomas automatinis vykdymosi laikas, pasirenkant valandą ir dažnumą pagal serverio apkrautumą arba informacijos gavimo svarbumą.

7 paveiksle pateiktas grafinės duomenų integravimo aplinkos „*Business Intelligence Development Studio*“ vartotojo sąsaja.



7 pav. MS SQL Server 2005 SSIS įrankio vartotojo sąsaja

2.6 Siekiamas duomenų integravimo procesų reinžinerijos sprendimas

Siekiamas sprendimas yra išlaikyti esamą duomenų struktūrą ir pastoviai vykdyti automatinį duomenų užkrovimą iš skirtingų duomenų šaltinių, esančių įmonės padaliniuose. Duomenų perkėlimo į centrinę duomenų bazę metu turi būti išvengta nesklaidumų jungiantis per pasirinktą naują DI įrankį.

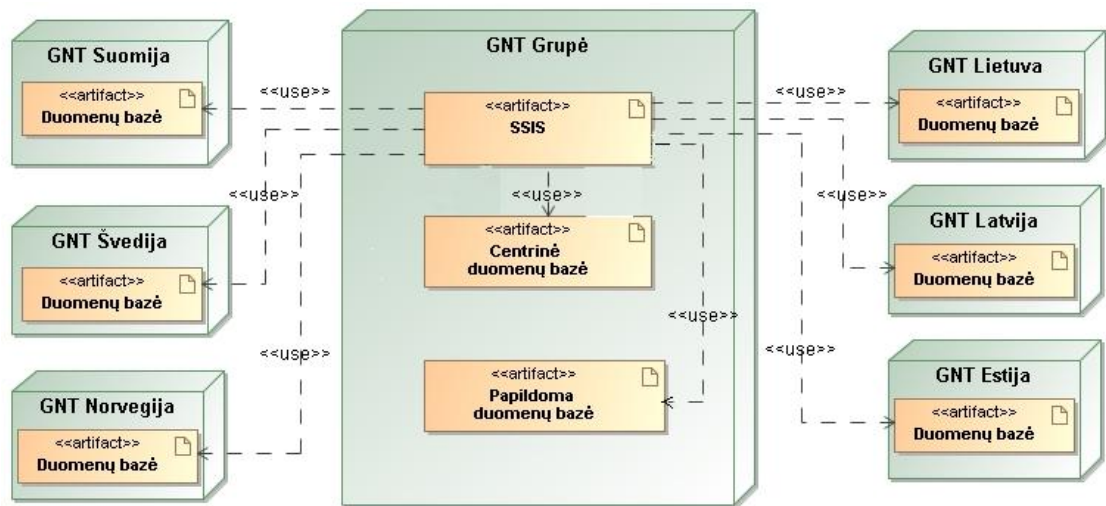
DTS nebuvimas naujoje *MS SQL Server 2005* versijoje privertė ieškoti geriausio sprendimo, kuo pakeisti DTS. Lengviausias ir greičiausias būdas juos pakeisti, yra pereinant iš *MS SQL Server 2000* versijos į *MS SQL Server 2005*. Čia yra galimybė perkelti sukurtus DTS procesus ir juos koreguoti, tačiau nėra funkcijos kurti naujiems DTS. Sprendimas yra tinkamas įmonėms, kurios naudoja nesudėtingus DTS ir nėra numatyta, kad juos reikės kurti ateityje [8] [7].

Kitas būdas pakeisti DTS funkciją yra naujos DBVS pasirinkimas. Įmonės, kurios naudojosi *MS SQL Server 2000*, netekusios galimybės lengvai integruoti duomenis pasirinko naują DBVS, nes jos nepajėgios toliau bendrauti su Microsoft (dėl biudžeto) arba duomenų srautai nėra dideli, kad būtų reikalinga turėti didelę duomenų bazę.

2 lentelė. Esamų sprendimų apibendrinimas

Integravimo Įrankis	MS SQL 2005 Server <i>SSIS</i>	MS SQL 2005 Server <i>DTS</i>	SAP Business Objects <i>DI</i>	Oracle <i>DI</i>	Oracle <i>OWB</i>
Kriterijai					
Windows operacinė sistema	+	+	+	+	+
Dideli duomenų srautai	+	+	+		+
Ateities perspektyvos	+		+	+	+
Nemokama paslauga		+			
Nereikalingi papildomi apmokymai	+	+			
Greitas naujos paslaugos realizavimas	+	+			
Pasirinktas variantas	+				

8 paveiksle pateikta siekiamo sprendimo verslo proceso abstrakti architektūra.



8 pav. Siekiamo sprendimo verslo proceso abstrakti architektūra

9 paveiksle pavaizduotas tiriamos veiklos procesas. Siekiama pakeitus DTS funkciją nauja paslauga, išlaikyti tokią pačia veiklos proceso struktūrą.



9 pav. Tiriamos veiklos procesas

2.7 Analizės išvados

1. Sprendžiant duomenų integravimo procesų patobulinimo ir perkėlimo į naują technologinę aplinką problemą, atlikta reinžinerijos principų analizė parodė, kad tikslinga ne tik perkelti esamus procesus į naują aplinką, bet tuo pačiu ir juos patobulinti, pagerinti jų veikimą.
2. Atlikta ETL įrankių *MS SQL Server 2005 SSIS*, *SAP Business Objects* ir *Oracle* analizė parodė, kad SSIS duomenų integravimo paslaugos yra perspektyviausia technologija

sprendžiant tiriamos įmonės duomenų integravimo problemą, kadangi dėl jos panašumo su ankstesne *MS SQL Server 2000* DTS technologija IT skyriaus darbuotojams nesunku persiorientuoti į šį produktą, galimas greitas SSIS realizavimas.

3. Duomenų integravimo procesų metodika

3.1 Duomenų integravimo šablono kriterijai

Reikalinga esamus duomenų integravimo procesus optimizuoti. Atliktos duomenų integravimo procesų įmonėje analizės metu pastebėta, kad DI procesai ilgai skaičiuojasi, naudoja daug atminties, serverio resursų. Siekiama sudaryti šabloną paremtą duomenų integravimo logika, kuris padėtų greitai ir optimaliai vykdyti DI procesą [18].

- Šablonas turi būti aiškus, tikslus, išsamus, suprantamas programuotojui, kurio aprašyme vartojamos sąvokos ir išsireiškimai yra iš atitinkamos srities, nedviprasmiški.
- Šablonas privalo būti lankstus, gebėti lengvai įtraukti, į jau veikiančią DI procesą, naują duomenų šaltinį. Tai reikalinga dėl tiriamos įmonės plėtimosi į kitas šalis. Naujas duomenų šaltinio įtraukimas turi būti paprastas, vykti be trukdžių, nereikalauti papildomos bendradarbiaujančių objektų korekcijos [18].
- Turi būti galimybė lengvai papildyti šabloną naujais elementais, kurie padėtų spręsti panašaus tipo problemas.

3.2 Duomenų integravimo šablonas

Šablonas sudarytas remiantis skirtingo detalumo duomenų atskyrimo straipsnyje aprašytu šablonu [18].

Modelio pavadinimas

Bendrinių duomenų atskyrimo šablonas

Tikslas

Duomenų integravimo procesą padaryti lankstų, naujo duomenų šaltinio įtraukimo ir pašalinimo atžvilgiu. Sumažinti DI procesų vėlinimo laiką.

Problema

Komplikuotas naujo duomenų šaltinio įtraukimas į duomenų integravimo procesą. Naujo šaltinio atsiradimas reikalauja daugelio susietų objektų modifikavimo. Be to, ilga DI procesų vykdymo trukmė užvėlina duomenų perkėlimą į galutinę saugyklos vietą. Tai sutrikdo procesų sinchroninį duomenų užkrovimą, dėl kurio duomenys yra nekorektiški.

Sprendimas

Duomenų bazėje esančios galutinės lentelės struktūra yra statinė. Tai yra, kiekviename įrašė egzistuoja tiek vienodų laukų, kiek yra duomenų šaltinių. Atsiradus naujam duomenų šaltiniui, reikia papildyti lentelę naujais laukais, sukurti duomenų perkėlimą. Sudėtinga ir atvirkštinė situacija, kai norima pašalinti iš faktinės lentelės atitinkamus laukus, naikinant egzistuojantį šaltinį.

Problema išsprendžiama perorganizavus galutinės lentelės struktūrą. Informaciją, gautą iš duomenų šaltinių, reikia suskirstyti į dvi dalis: pagrindinę ir detalią. Sugrupuota informacija saugoma skirtingose lentelėse.

Taikymas

Šablonas taikomas kuomet galima išskirti bendrinę ir detalią informaciją.

Šablono elementai

Duomenų integravimo įrankis, duomenų bazės lentelės.

Rezultatas

Informacijos suskirstymas leidžia įtraukti ir pašalinti neribotą skaičių duomenų šaltinių vienu metu. Dėl atskyrimo sumažėja duomenų gavybos laiko trukmė.

Pavyzdys

Duomenų integravimo procesu užkraunama informacija apie produktus, esančius skirtinguose trejose įmonės padaliniuose: Lietuvoje, Latvijoje, Estijoje. Produkto informaciją sudaro: produkto kodas, produkto pavadinimas, produkto vadybininkas (10 pav.).

1 duomenų šaltinis	Produkto_kodas	Produkto_pavadinimas	Produkto_vadybininkas
	XXX	Logitech pelė	Indrė Kungytė
2 duomenų šaltinis	Produkto_kodas	Produkto_pavadinimas	Produkto_vadybininkas
	XXX	Logitech pelė	Jonas Jonaitis
3 duomenų šaltinis	Produkto_kodas	Produkto_pavadinimas	Produkto_vadybininkas
	XXX	Logitech pelė	Petras Petraitis

10 pav. Produkto informacija skirtinguose duomenų šaltiniuose

Išgauti duomenys sujungiami ir patalpinami į vieną lentelę (11 pav.).

Produkto_kodas	Produkto_pavadinimas	LT_Produkto_vadybininkas	LV_Produkto_vadybininkas	EE_Produkto_vadybininkas
XXX	Logitech pelė	Indrė Kungytė	Jonas Jonaitis	Petras Petraitis

11 pav. Produkto informacija įrašyta į galutinę DB lentelę

Atlikus duomenų analizę galime išskirti detalią ir pagrindinę informaciją. Kadangi produkto pavadinimas ir kodas visuose duomenų šaltiniuose yra vienodas, tai šiuos duomenis laikome pagrindiniais, o produkto vadybininkai - detali informacija, nes skirtinguose padaliniuose už tą patį produktą yra atsakingi skirtingi asmenys. Ši informacija yra unikali padalinio lygyje. Įvedame papildomą padalinio požymį ir išsaugome duomenis naujose lentelėse (12 pav.).

Pagrindinė produkto informacija

	Produkto_kodas	Produkto_pavadinimas
1	XXX	Logitech pelė

Detali produkto informacija

	Produkto_kodas	Padalinys	Produkto_vadybininkas
1	XXX	LT	Indrė Kungytė
2	XXX	LV	Jonas Jonaitis
3	XXX	EE	Petras Petraitis

12 pav. Produkto duomenys atskirti ir įrašyti į galutinės DB lenteles

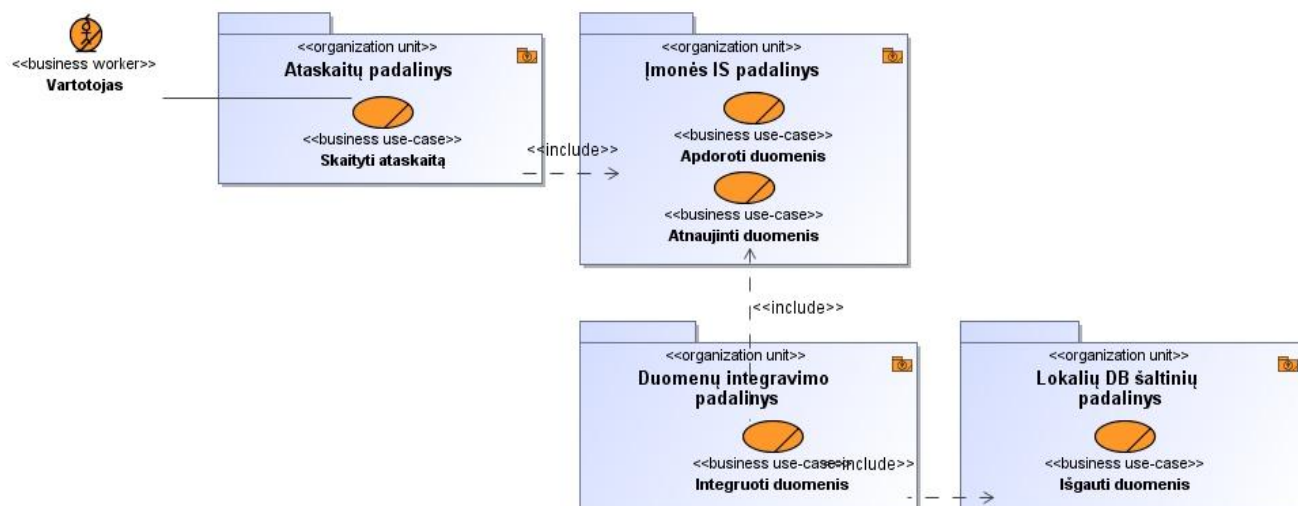
Atsiradus naujam duomenų šaltiniui, šio pavyzdžio atveju, įsirašytų papildoma eilutė detalios produkto informacijos DB lentelėje. Nereikalinga DB lentelių struktūros korekcija.

4. Patobulinto duomenų integravimo posistemo reikalavimų specifikacija

Šiame ataskaitos skyriuje išnagrinėjami visi funkciniai ir nefunkciniai reikalavimai realizuojamam projektui, taip pat pateikiami visi panaudojimų atvejai su detalesnėmis lentelėmis.

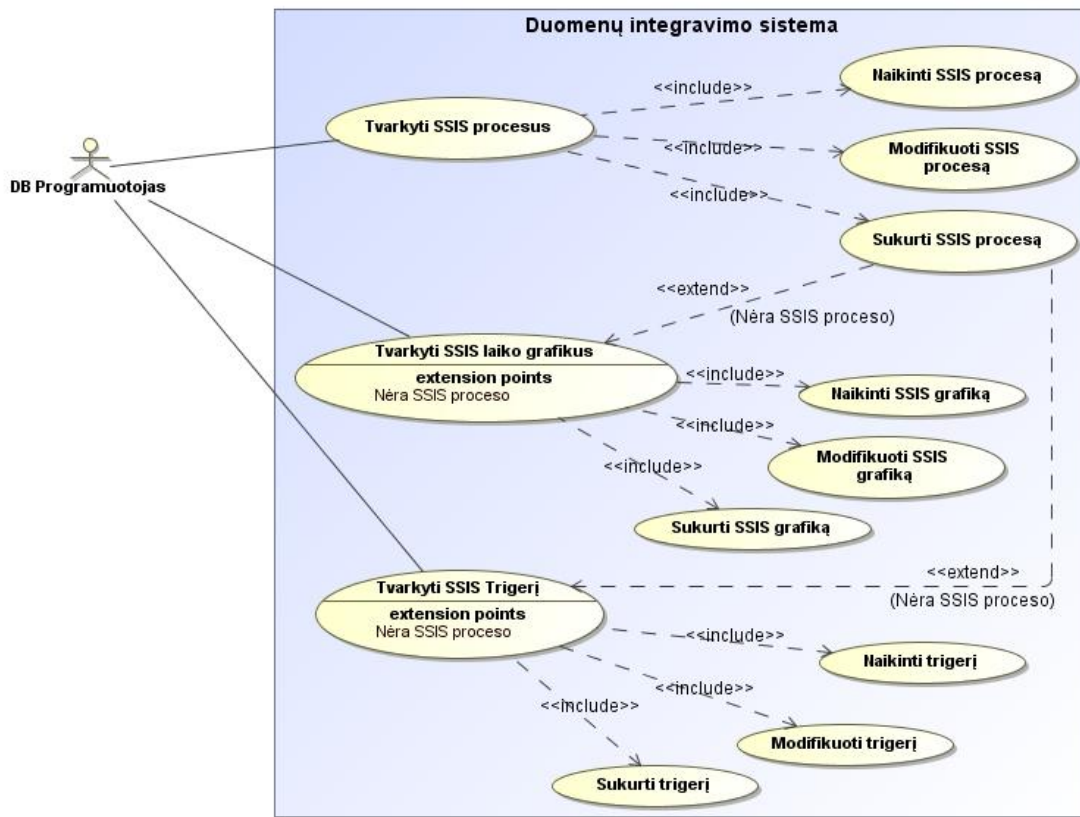
4.1 Funkciniai duomenų integravimo posistemo reikalavimai

Panaudojimo atvejų diagramoje (13 pav.) pavaizduota duomenų integravimo proceso vieta tiriamosios įmonės infrastruktūroje. Duomenų integravimo procesas tiesiogiai neįtakojamas galutinio sistemos vartotojo – tai vidinis procesas.



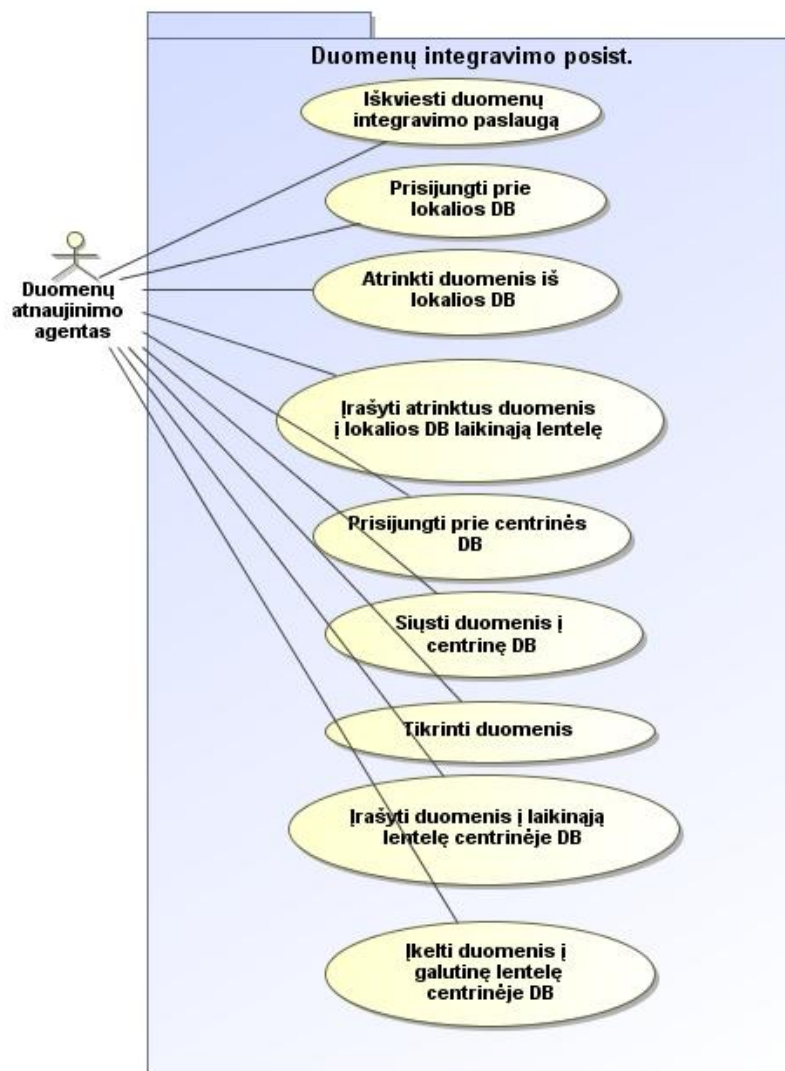
13 pav. Duomenų integravimo procesas įmonės infrastruktūroje

14 paveiksle pateikti panaudojimo atvejai, kuriuos gali atlikti DB programuotojas, kai administruoja duomenų integravimo procesus, sudaro procesams grafikus, trigerius.



14 pav. Duomenų integravimo posistemio panaudojimo atvejų diagrama

15 paveiksle pateikiama principinė kompiuterizuojamų panaudojimo atvejų diagrama. Čia pavaizduoti pagrindiniai procesai reikalingi duomenų integravimui vykdyti.



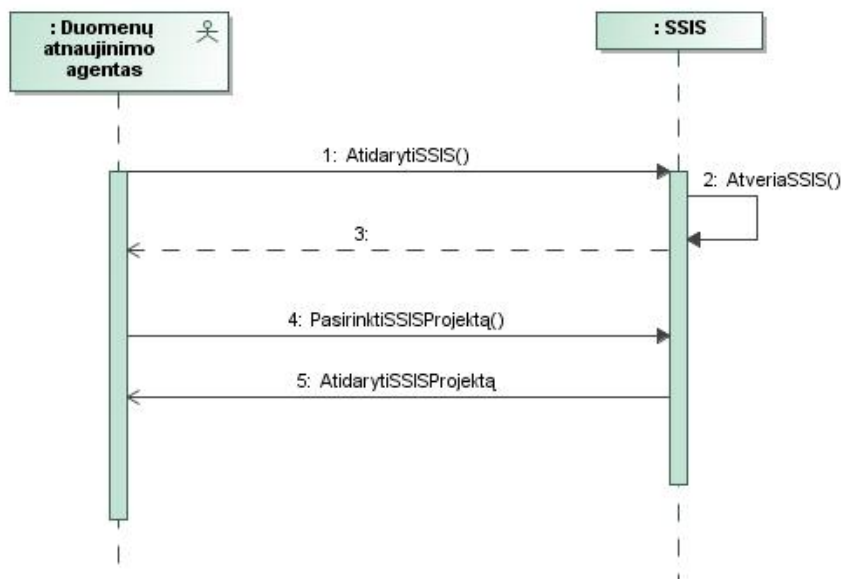
15 pav. Duomenų integravimo proceso panaudojimo atvejų diagrama

Panaudojimo atvejų specifikacij

- ✓ Iškviesti duomenų integravimo paslaugą;
- ✓ Prisijungti prie lokalsios DB;
- ✓ Atrinkti duomenis iš lokalsios DB;
- ✓ Įrašyti atrinktus duomenis į lokalsios DB laikinąją lentelę;
- ✓ Prisijungti prie centrinės DB;
- ✓ Siųsti duomenis į centrinę DB;
- ✓ Tikrinti duomenis;
- ✓ Įrašyti duomenis į laikinąją lentelę centrinėje DB;
- ✓ Įkelti duomenis į galutinę lentelę centrinėje DB.

3 lentelė. Panaudojimo atvejo „Iškviešti duomenų integravimo paslaugą“ specifikacija

PA „Iškviešti duomenų integravimo paslaugą“	
Tikslas. Iškviešti duomenų integravimui skirtą <i>SQL Server Integration Services</i> paslaugą.	
Aprašymas. DB programuotojas atidaro duomenų integravimo procesui reikalingą administruoti programą, kurioje galima kurti naujus integravimo procesus, juos redaguoti, administruoti.	
Prieš sąlyga	DB programuotojas įsijungia SSIS paslaugą
Aktorius	DB Programuotojas
Sužadinimo sąlyga	DB programuotojas nori sukurti \ administruoti\ redaguoti duomenų integravimo procesą.
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	
1.DB programuotojas atidaro SSIS paslaugą	Windows atveria langą.
2.DB programuotojas pasirenka SSIS projektą	SSIS atveria pasirinktą projektą
Po sąlyga:	Atidarytas pasirinktas SSIS projektas
Alternatyvūs scenarijai	

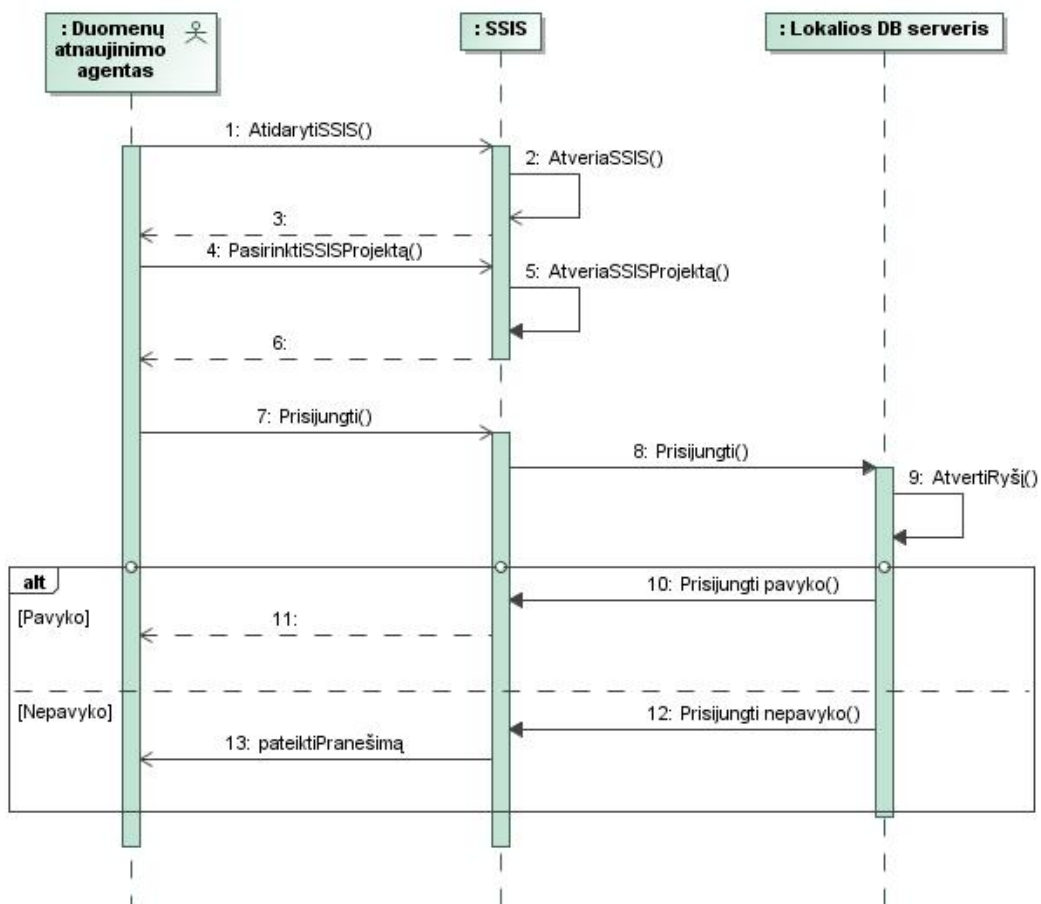


16 pav. Panaudojimo atvejo „Iškviešti duomenų integravimo paslaugą“ sekų diagrama

4 lentelė. Panaudojimo atvejo „Prisijungti prie lokalsios DB“ specifikacija

PA „Prisijungti prie lokalsios DB“	
Tikslas. Duomenims integruoti reikia prisijungti prie lokalsios, kad būtų galima atrinkti atitinkamus duomenis.	
Aprašymas. SSIS paslaugos agentas sužadina procesą, kuris vykdo prisijungimą prie serverio.	
Prieš sąlyga	SSIS integravimo agentas sužadina prisijungimo prie lokalaus serverio procesą.
Aktorius	SSIS integravimo agentas, DB programuotojas
Sužadinimo sąlyga	SSIS procesui reikia prisijungti prie duomenų bazės,

		esančios lokaliame serveryje.
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1.DB programuotojas atsidaro SSIS paslaugą		Windows atveria langą.
2.DB programuotojas pasirenka SSIS projektą		SSIS atveria pasirinktą projektą
3.DB programuotojas paleidžia vykdyti duomenų integravimą		Sistema prisijungia prie lokalaus serverio
Po sąlyga:		Prieinami duomenys lokalaus serverio duomenų bazėje.
Alternatyvūs scenarijai		
1.DB programuotojas paleidžia vykdyti duomenų integravimą		Sistema neprisijungia prie lokalaus serverio, grąžinamas atitinkamas pranešimas.
		SSIS nutraukia duomenų integravimo procesą.

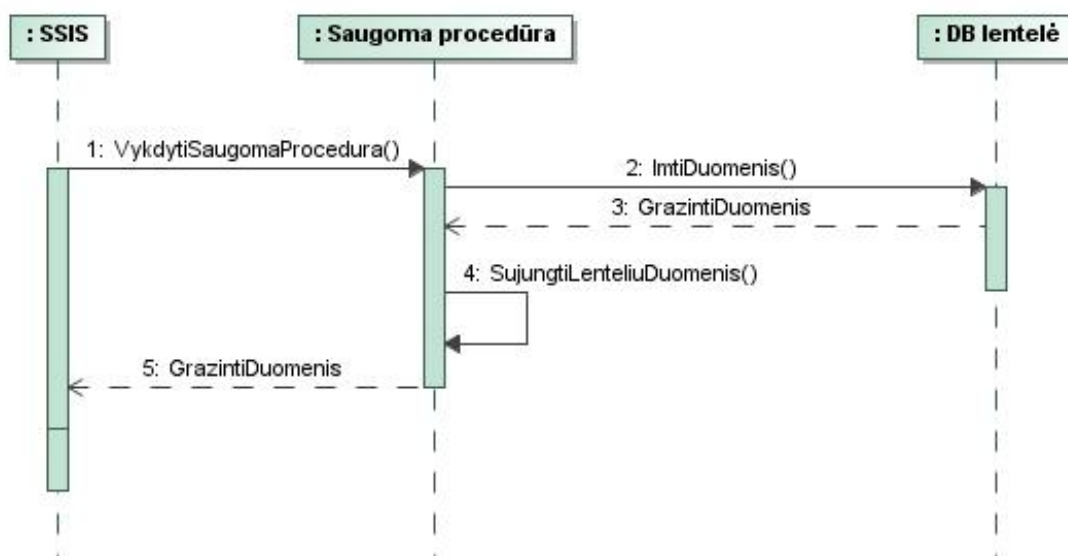


17 pav. Panaudojimo atvejo „Prisijungti prie lokalis DB“ sekų diagrama

5 lentelė. Panaudojimo atvejo „Atrinkti duomenis iš lokalis DB“ specifikacija

PA „Atrinkti duomenis iš lokalis DB“
Tikslas. Reikia atrinkti reikalingus duomenis duomenų integravimui.
Aprašymas. SSIS paslaugos agentas sužadina procesą, kuris vykdo duomenų atrinkimą iš lokalaus serverio DB.

Prieš sąlyga	SSIS integravimo agentas sužadina duomenų atrinkimo procesą.
Aktorius	SSIS integravimo agentas
Sužadinimo sąlyga	SSIS procesui reikia atrinkti duomenis iš duomenų bazės, esančios lokaliame serveryje.
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. SSIS agentas paleidžia vykdyti duomenų atrinkimo procesą	SSIS išskviečia saugomą procedūrą.
	Saugoma procedūra kreipiasi į lokalaus serverio DB lenteles
	Saugoma procedūra sujungia gautus duomenis
	Saugoma procedūra grąžina atrinktus duomenis
Po sąlyga:	Prieinami duomenys lokalaus serverio duomenų bazėje
Alternatyvūs scenarijai	

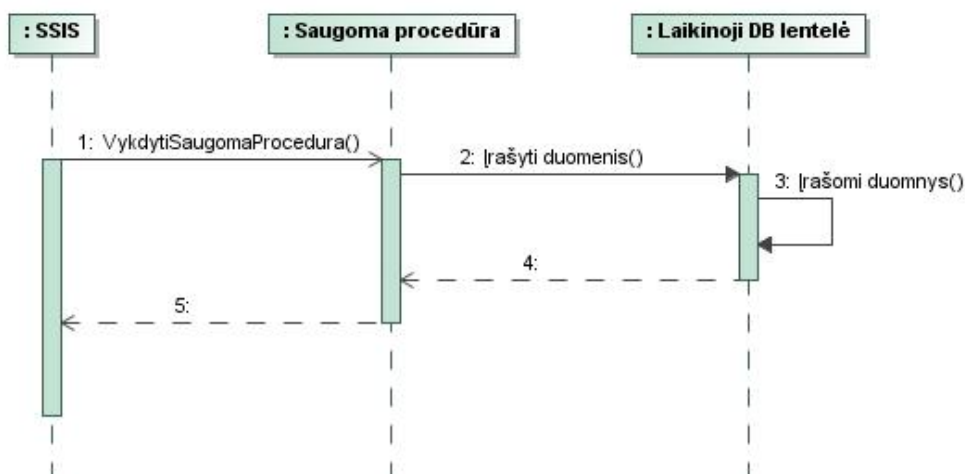


18 pav. Panaudojimo atvejo „Atrinkti duomenis iš lokalsios DB“ sekų diagrama

6 lentelė. PA „Įrašyti atrinktus duomenis į lokalsios DB laikinąją lentelę“ specifikacija

PA „Įrašyti atrinktus duomenis į lokalsios DB laikinąją lentelę“	
Tikslas. Reikia gautą duomenų sąrašą įrašyti į lentelę, kuri vėliau ištrinama.	
Aprašymas. SSIS paslaugos agentas sužadina procesą, kuris vykdo duomenų įrašymą į laikiną DB lentelę lokaliame serveryje.	
Prieš sąlyga	SSIS integravimo agentas sužadina duomenų įrašymą.
Aktorius	SSIS integravimo agentas
Sužadinimo sąlyga	SSIS procesui reikia įrašyti duomenis į laikiną duomenų bazės lentelę, esančią lokaliame serveryje.
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai

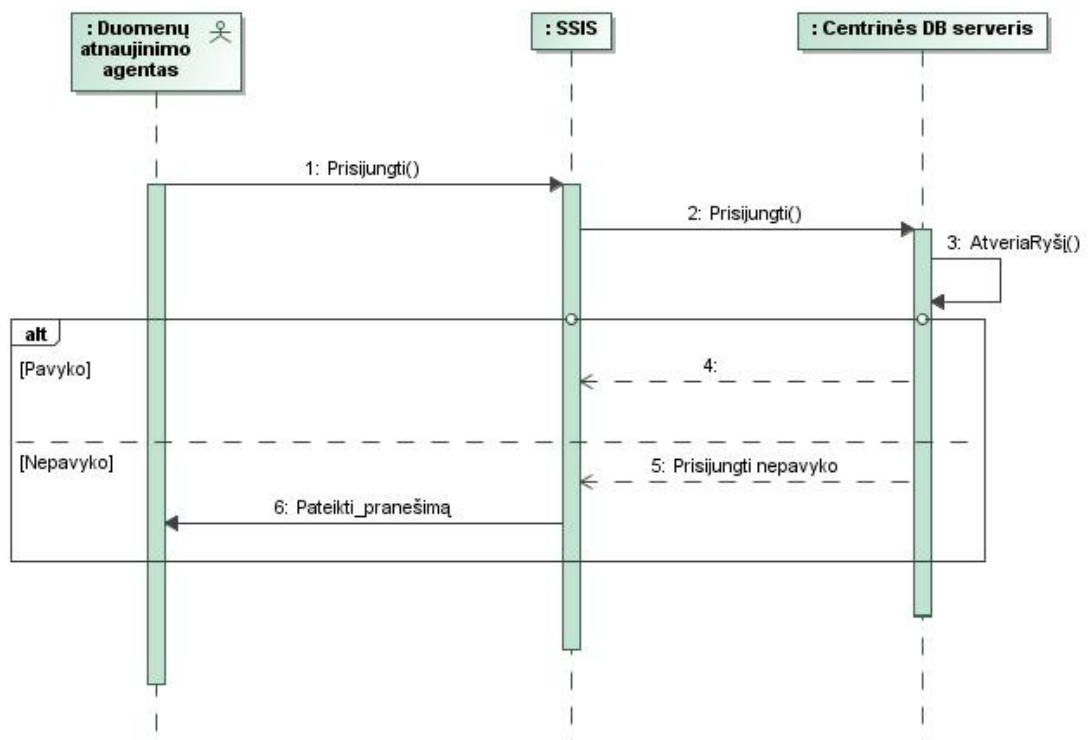
1. SSIS agentas paleidžia vykdyti duomenų įrašymo procesą	SSIS išskviečia saugomą procedūrą.
	Saugoma procedūra kreipiasi į lokalaus serverio DB laikiną lentelę
	Saugoma procedūra įrašo duomenis
Po sąlyga:	įrašyti duomenys laikinoje DB lentelėje
Alternatyvūs scenarijai	



19 pav. PA „Įrašyti atrinktus duomenis į lokalsios DB laikinąją lentelę“ sekų diagrama

7 lentelė. PA „Prisijungti prie centrinės DB“ specifikacija

PA „Prisijungti prie centrinės DB“	
Tikslas. Duomenims integruoti reikia prisijungti prie centrinės DB, kad būtų prieinama vieta į kurią reikia perkelti duomenis.	
Aprašymas. SSIS paslaugos agentas sužadina procesą, kuris vykdo prisijungimą prie serverio.	
Prieš sąlyga	SSIS integravimo agentas sužadina prisijungimą prie centrinės DB procesą.
Aktorius	SSIS integravimo agentas
Sužadinimo sąlyga	SSIS procesui reikia prisijungti prie duomenų bazės, esančios centriniame serveryje.
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. SSIS integravimo agentas paleidžia prisijungimo prie centrinės DB procesą	Sistema prisijungia prie centrinio serverio
Po sąlyga:	Atidarytas ryšys į centrinę DB
Alternatyvūs scenarijai	
1. SSIS integravimo agentas paleidžia prisijungimo prie centrinės DB procesą	Sistema neprisijungia prie centrinio serverio, grąžinamas atitinkamas pranešimas.
	SSIS nutraukia duomenų integravimo procesą.

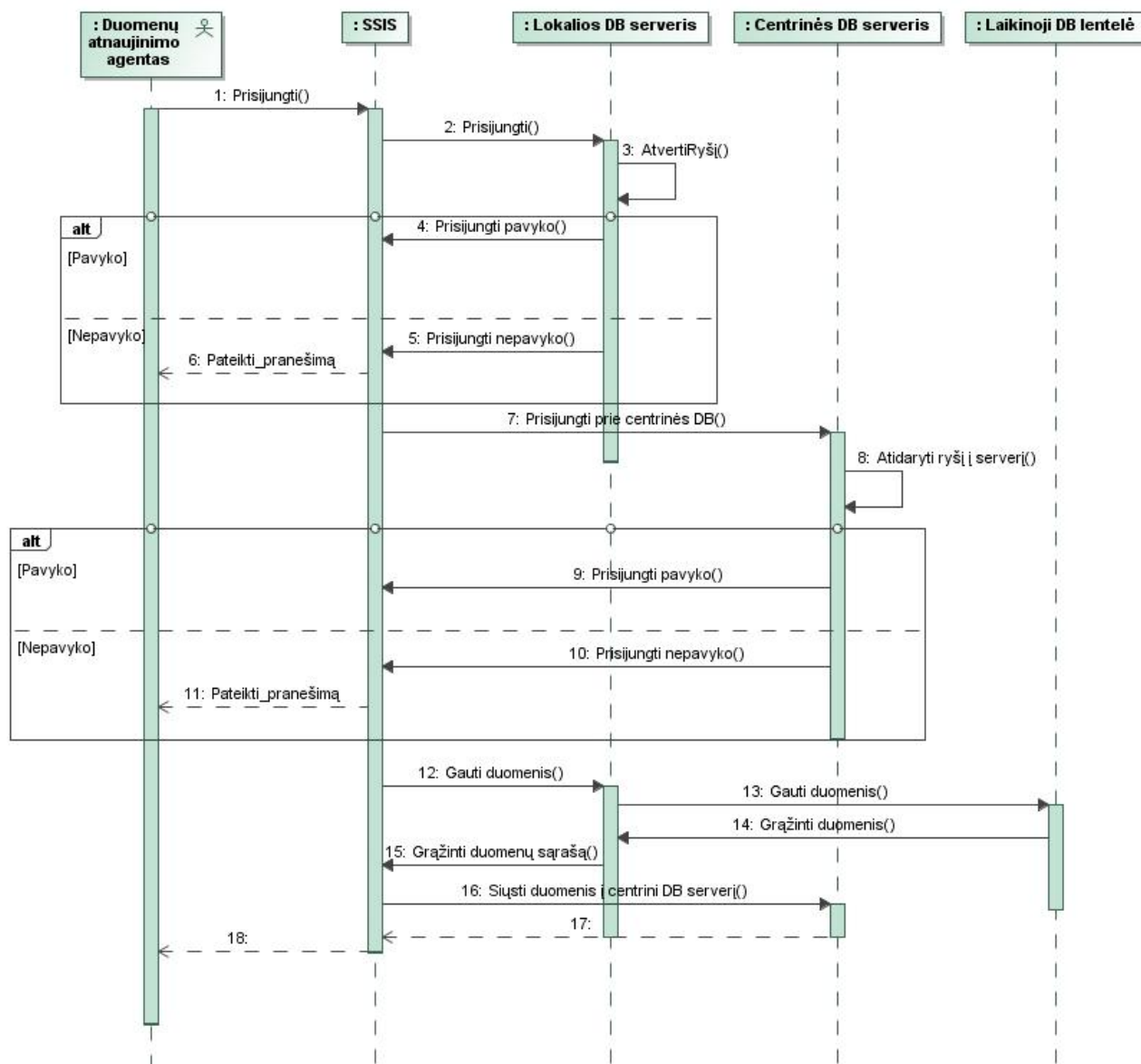


20 pav. PA „Prisijungti prie centrinės DB“ sekų diagrama

8 lentelė. PA „Siųsti duomenis į centrinę DB“ specifikacija

PA „Siųsti duomenis į centrinę DB“		
Tikslas. Duomenims siųsti į centrinę DB reikia duomenų iš laikinos lentelės ir prieinamos centrinės DB .		
Aprašymas. SSIS paslaugos agentas sužadina procesą, kuris vykdo duomenų perkėlimą iš vienos DB į kitą.		
Prieš sąlyga	SSIS integravimo agentas sužadina duomenų perkėlimo procesą.	
Aktorius	SSIS integravimo agentas	
Sužadinimo sąlyga	SSIS procesui reikia prisijungti prie duomenų bazių, esančių centriname ir lokaliame serveriuose.	
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių srautas		
1. SSIS integravimo agentas paleidžia prisijungimo prie lokalsios DB procesą	Sistemos reakcija ir sprendimai Sistema prisijungia prie lokalaus serverio	
2. SSIS integravimo agentas paleidžia prisijungimo prie centrinės DB procesą	Sistema prisijungia prie centrinio serverio	
3. SSIS integravimo agentas vykdo duomenų išgavimo procesą	Sistema gauna duomenis iš laikinos DB lentelės	
4. SSIS integravimo agentas vykdo duomenų perkėlimo procesą	Sistema patalpina duomenis Centrinėje DB	
Po sąlyga:	Perkelti duomenys	
Alternatyvūs scenarijai		

2. SSIS integravimo agentas paleidžia prisijungimo prie lokalsios DB procesą	Sistema neprisijungia prie lokalaus serverio, grąžinamas atitinkamas pranešimas.
	SSIS nutraukia duomenų integravimo procesą.
2. SSIS integravimo agentas paleidžia prisijungimo prie centrinės DB procesą	Sistema neprisijungia prie centrinio serverio, grąžinamas atitinkamas pranešimas.
	SSIS nutraukia duomenų integravimo procesą.

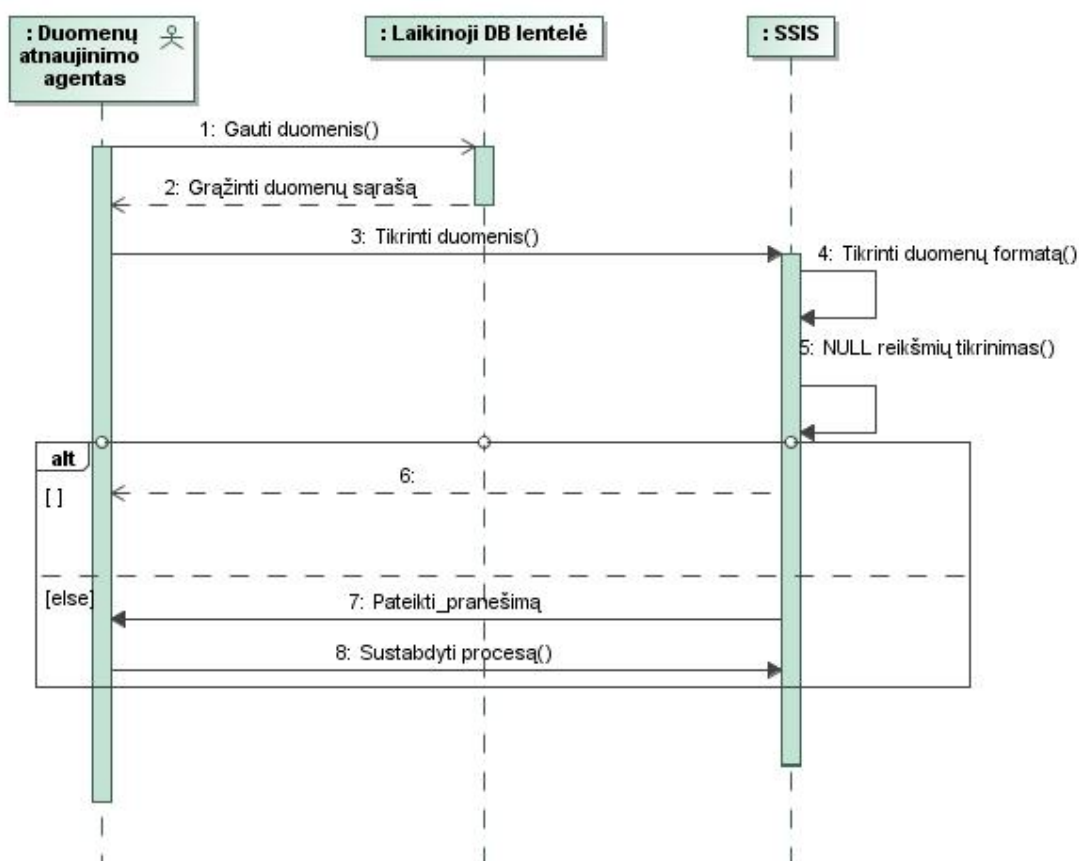


21 pav. PA „Siųsti duomenis į centrinę DB“ sekų diagrama

9 lentelė. PA „Tikrinti duomenis“ specifikacija

PA „Tikrinti duomenis“	
Tikslas. Reikia patikrinti duomenų teisingumą.	
Aprašymas. SSIS paslaugos agentas sužadina procesą, kuris vykdo duomenų tikrinimą.	
Prieš sąlyga	SSIS integravimo agentas sužadina duomenų tikrinimo

		procesą.
Aktorius		SSIS integravimo agentas
Sužadavimo sąlyga		SSIS procesui reikia patikrinti gautų duomenų sąrašą.
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. SSIS integravimo agentas vykdo duomenų paėmimo procesą		Sistema gauna duomenis iš DB laikinos lentelės
2. SSIS integravimo agentas vykdo duomenų formato patikrinimą		Sistema pateikia duomenų sąrašą
3. SSIS integravimo agentas vykdo neužpildytų duomenų reikšmių patikrinimą		Sistema pateikia duomenų sąrašą
Po sąlyga:		
Alternatyvūs scenarijai		
1. SSIS integravimo agentas paleidžia duomenų tikrinimo procesą		SSIS nutraukia duomenų integravimo procesą.

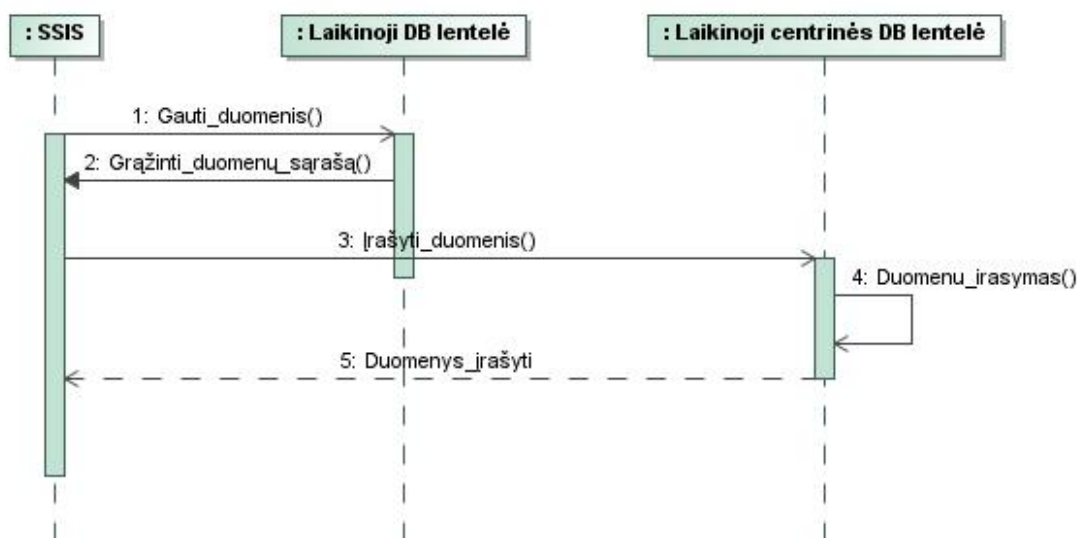


22 pav. PA „Tikrinti duomenis“ sekų diagrama

10 lentelė. PA „Įrašyti duomenis į laikiną lentelę centrinėje DB“ specifikacija

PA „Įrašyti duomenis į laikiną lentelę centrinėje DB“	
Tikslas. Reikia duomenis įrašyti į laikiną lentelę centrinėje DB.	
Aprašymas. SSIS paslaugos agentas sužadina procesą, kuris įrašo duomenis į laikiną centrinės DB lentelę.	
Prieš sąlyga	SSIS integravimo agentas sužadina duomenų įrašymo

		procesą.
Aktorius		SSIS integravimo agentas
Sužadinimo sąlyga		SSIS procesui reikia įrašyti duomenis į lentelę.
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. SSIS integravimo agentas vykdo duomenų paėmimo procesą		Sistema gauna duomenis
2. SSIS integravimo agentas vykdo duomenų įrašymą į lentelę procesą		Sistema įrašo duomenis
Po sąlyga:		
Alternatyvūs scenarijai		

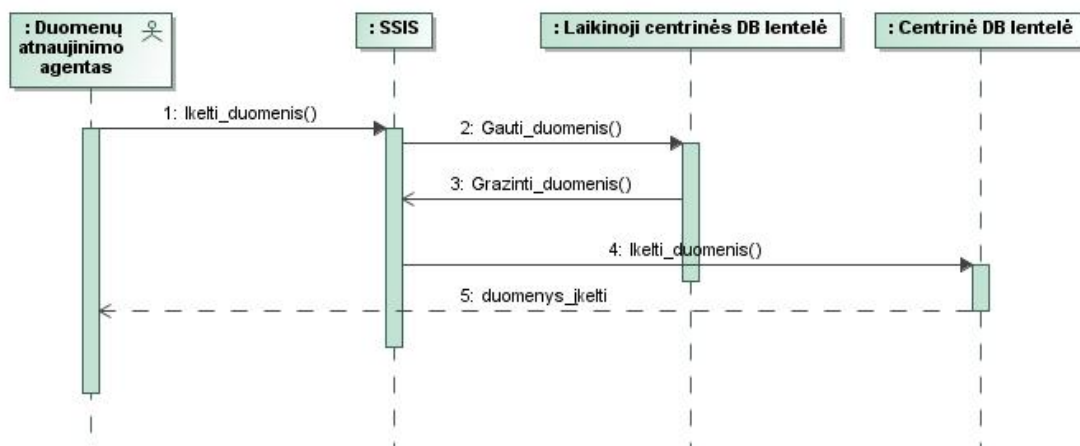


23 pav. PA „Įrašyti duomenis į laikiną lentelę centrinėje DB“ sekų diagrama

24 pav.

11 lentelė. PA „Įkelti duomenis į galutinę lentelę centrinėje DB“ specifikacija

PA „Įkelti duomenis į galutinę lentelę centrinėje DB“		
Tikslas. Reikia duomenis patalpinti galutinėje lentelėje centrinėje DB.		
Aprašymas. SSIS paslaugos agentas sužadina procesą, kuris perkelia duomenis į galutinę centrinės DB lentelę.		
Prieš sąlyga		SSIS integravimo agentas sužadina duomenų įrašymo procesą.
Aktorius		SSIS integravimo agentas
Sužadinimo sąlyga		SSIS procesui reikia įrašyti duomenis į galutinę lentelę.
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	
	Specializuoja PA	
Pagrindinis įvykių srautas		Sistemos reakcija ir sprendimai
1. SSIS integravimo agentas vykdo duomenų paėmimo procesą		Sistema gauna duomenis
2. SSIS integravimo agentas vykdo duomenų įrašymą į lentelę procesą		Sistema įrašo duomenis
Po sąlyga:		
Alternatyvūs scenarijai		



25 pav. PA „Įkelti duomenis į galutinę lentelę centrinėje DB“ sekų diagrama

4.2 Nefunkciniai duomenų integravimo proceso reikalavimai

12 lentelė. Nefunkcinių reikalavimų sąrašas

Nr.	Sritis	Aprašymas
1.	Nefunkcinis reikalavimas sistemai	Pagrindimas: Duomenų integravimo sistema neturi nustoti veikti dėl interneto ryšio.
		Tinkamumas: Turi būti užtikrintas nenutrūkstamas internetinis ryšys, kad duomenų integravimas vyktų teisingai.
2.	Nefunkcinis reikalavimas duomenų integralumui	Pagrindimas: Duomenų integravimo proceso metu neturi įvykti klaida dėl duomenų struktūros nesutapimų.
		Tinkamumas: Duomenys paruošti duomenų integravimui turi sutapti su centrinės duomenų bazės lentelių struktūra.
3.	Nefunkcinis reikalavimas duomenų integralumui	Pagrindimas: Duomenų integravimo proceso metu neturi įvykti klaida dėl duomenų nekorektiškumo.
		Tinkamumas: Duomenų paruoštų duomenų integravimui formatas ir tipas turi sutapti su centrinės duomenų bazės lentelių formatais, tipais.
4.	Nefunkcinis reikalavimas sistemai	Pagrindimas: Duomenų integravimo proceso vykdymosi trukmė neturi būti ilga.
		Tinkamumas: Duomenų integravimo procesai turi būti greiti ir efektyvūs, kad neužvėlintų kitų procesų. Tam reikalinga juos optimizuoti.
5.	Nefunkcinis reikalavimas sistemai	Pagrindimas: DB programuotojas pageidauja gauti informaciją apie duomenų integravimo procesuose įvykusias klaidas. Tam reikalinga į pagrindinius integravimo etapus įdėti informacinius pranešimus, kuriuos gauna DB programuotojas.
		Tinkamumas: Pranešimai turi informuoti apie įvykusių klaidų pirminę priežastį ir vietą duomenų integravimo sistemoje.

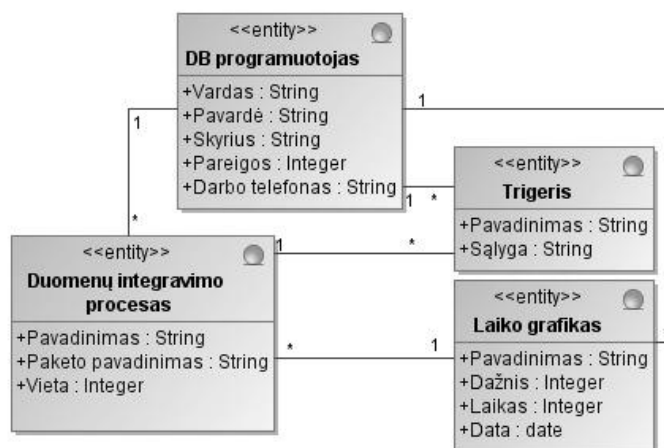
6.	Nefunkcinis reikalavimas sistemai	Pagrindimas: DB programuotojas nori stebėti duomenų integravimo sistemos atliktus veiksmus, tam reikalinga fiksuoti juos dokumente - sukurti log failą.
		Tinkamumas: Fiksuotų sistemos veiksmų sąrašas turi padėti rasti pažeidžiamas ir duomenų integravimui pavojingas sistemos vietas.
7.	Nefunkcinis reikalavimas sistemai	Pagrindimas: Duomenų atstatomumas.
		Tinkamumas: Duomenų integravimo procesų kopijos turi užtikrinti visišką integruojamų duomenų atstatomumą (angl. <i>Recoverability</i>).
8.	Nefunkcinis reikalavimas sistemai	Pagrindimas: Apsauga nuo neteisėtos prieigos.
		Tinkamumas: Turi užtikrinti duomenų integravimo sistemos apsaugą nuo nesankcionuotų veiksmų.

4.3 Duomenų integravimo proceso dalykinės srities modelis

Paveiksluose pavaizduoti įmonės duomenų integravimo procesų metaduomenų modeliai.

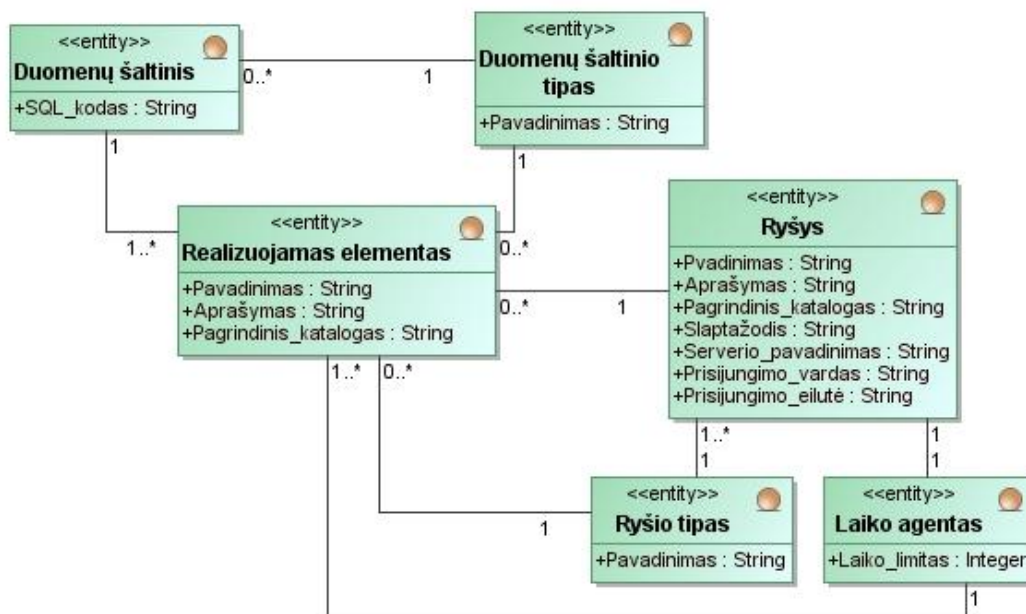
25 paveiksle pateiktas duomenų integravimo sistemos metaduomenų modelis, kurį sudaro duomenų integravimo procesai, DB programuotojo nustatyti laiko grafikai, kuriuos sužadina trigeriai. Trigerius sukuria DB programuotojas SSIS įrankio pagalba.

Visi duomenų integravimo (DI) procesai yra specifikuoti į tam tikrą duomenų perkėlimą ir apdorojimą, pavyzdžiui, klientų duomenų integravimas, darbuotojų duomenų integravimas, pardavimai, pirkimai, grąžinimai ir t.t. Kiekvienas DI procesas turi griežtą laiko grafiką ir trigerį. Laiko ir įvykio agentų pagalba sudaromos DI procesų sekos, kurios užtikrina nuoseklų, taisyklingą duomenų užkrovimą į centrinę duomenų bazę. DI procesų visuma sudaro įmonės duomenų integravimo sistemą.



26 pav. Duomenų integravimo sistemos procesų metaduomenų modelis

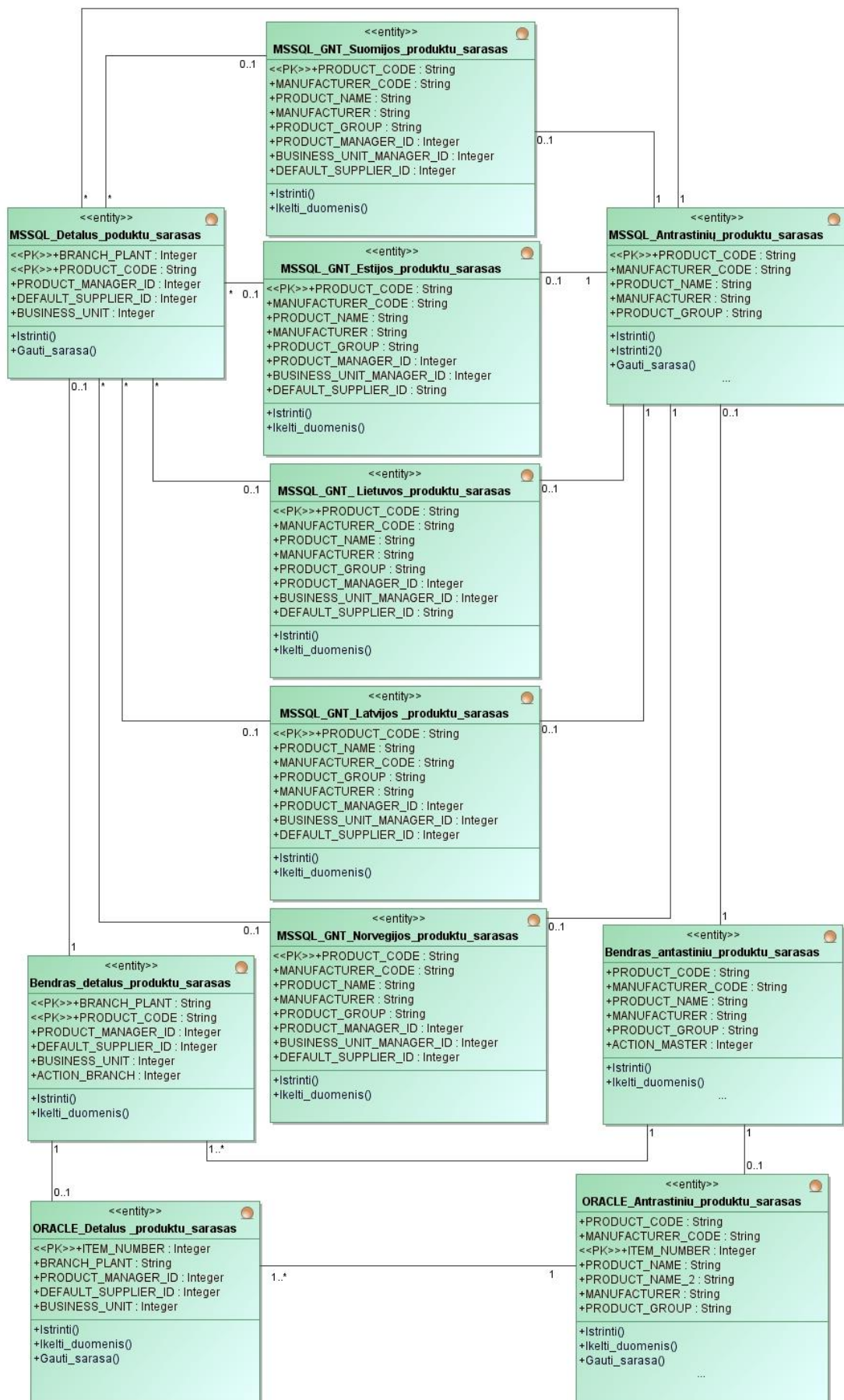
26 paveiksle pateiktas abstraktus duomenų integravimo proceso metaduomenų modelis, kurį sudaro „Realizuojamas elementas“, „Duomenų šaltinis“, „Duomenų šalinio tipas“, „Ryšys“ (angl. *Connection*), „Ryšio tipas“ (angl. *Connection Type*), „Laiko agentas“ (angl. *TimeOut*).



27 pav. Duomenų integravimo proceso metaduomenų modelis

27 paveiksle pateiktas nenormalizuotas duomenų integravimo proceso esybių klasių modelis. Juo siekiama užtikrinti nuolatinį duomenų integravimą iš *Oracle* ir *MS SQL* duomenų bazių į centrinę DB. Vienos iteracijos metu atrenkama informacija apie produktus iš *Oracle* ir DBM duomenų bazių. Šie duomenys integruojami į centrinėje duomenų bazėje esančias lenteles. Kadangi DBM sudaro 5 duomenų bazės, esančios skirtinguose serveriuose, dėl to reikalingas šių duomenų apdorojimas ir sujungimas į vieną duomenų sąrašą. Lokalios duomenų bazės lentelės yra momentinės – duomenys nuolat kinta, dėl to reikalinga užfiksuoti duomenis pasirinktu laiku. DB lentelėse: *MSSQL_GNT_suomijos_produkto_sarasas*, *MSSQL_GNT_estijos_produkto_sarasas*, *MSSQL_GNT_latvijos_produkto_sarasas*, *MSSQL_GNT_lietuvos_produkto_sarasas*, *MSSQL_GNT_norvegijos_produkto_sarasas*, saugomi įrašai forma patogiai dirbti duomenų saugykloje. Suintegruoti duomenys tarpusavyje lyginami ir iš susidariusių skirtumų generuojamas XML dokumentas, kuris atnaujina lentelės duomenis *Oracle* duomenų bazėje (vyksta atvirkštinis duomenų integravimas į *Oracle* DB). Procesas vyksta periodiškai – vieną kartą per dieną.

Proceso tikslas – *Oracle* duomenų bazėje turėti naujausią informaciją apie įmonės padalinių sandėliuose esančius produktus.



28 pav. Dalykinės srities realaus projekto esybių modelis

4.4 Duomenų integravimo proceso architektūra

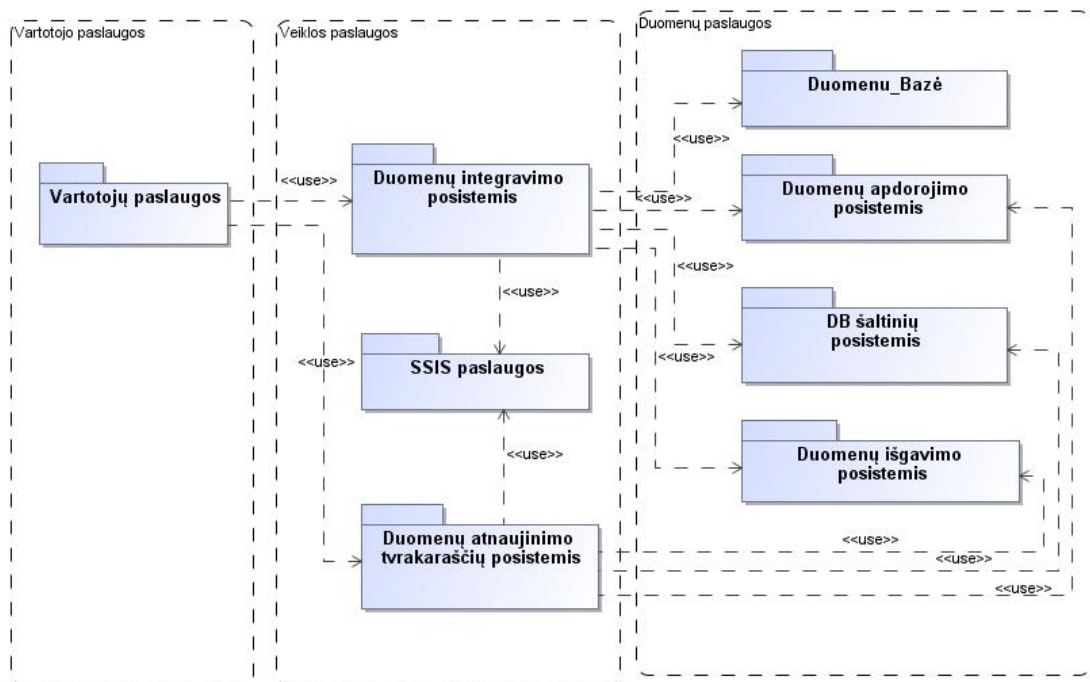
4.4.1 Duomenų integravimo loginė architektūra

Loginė architektūra parodo kokios pagrindinės dalys sudaro kuriamą sistemą. 28 paveiksle pavaizduoti pagrindiniai duomenų integravimui reikalingi loginiai architektūros lygiai.



29 pav. Trijų lygių loginės architektūros principinė schema

SSIS paslaugų lygis sudarytas iš duomenų integravimo ir duomenų atnaujinimo tvarkaraščių posistemių. Architektūros lygis veikia ETL principu. Duomenys gaunami iš duomenų šaltinių, iš DB šaltinių posistemio, kuris yra duomenų paslaugų lygyje. 29 paveiksle pateikta detali duomenų integravimo loginė architektūra.

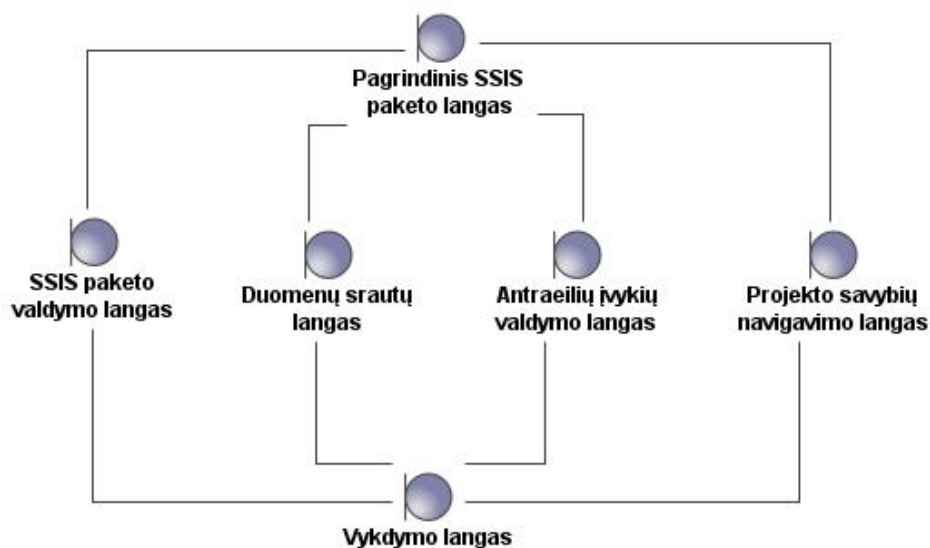


30 pav. Detalus duomenų integravimo loginės architektūros modelis

4.4.2 Vartotojo paslaugos

Duomenų integravimo procesai kuriami ir valdomi per *Business Intelligence Development Studio* aplinką. Kadangi aplinkoje yra paprasta naviguoti ir valdyti procesus, todėl speciali vartotojo sąsaja nekuriama.

30 paveiksle pateiktas navigavimo planas *Business Intelligence Development Studio* aplinkoje, kuriame yra pavaizduoti pagrindiniai langai.



31 pav. Navigavimo planas

SSIS paketo valdymo langas – aplinka, kurioje sudedami visi elementai reikalingi duomenų integravimo logikai išpildyti.

Duomenų srautų langas – lange vykdomas duomenų perkėlimas. Pasirenkami duomenų šaltiniai, nurodomos galutinės duomenų saugyklos.

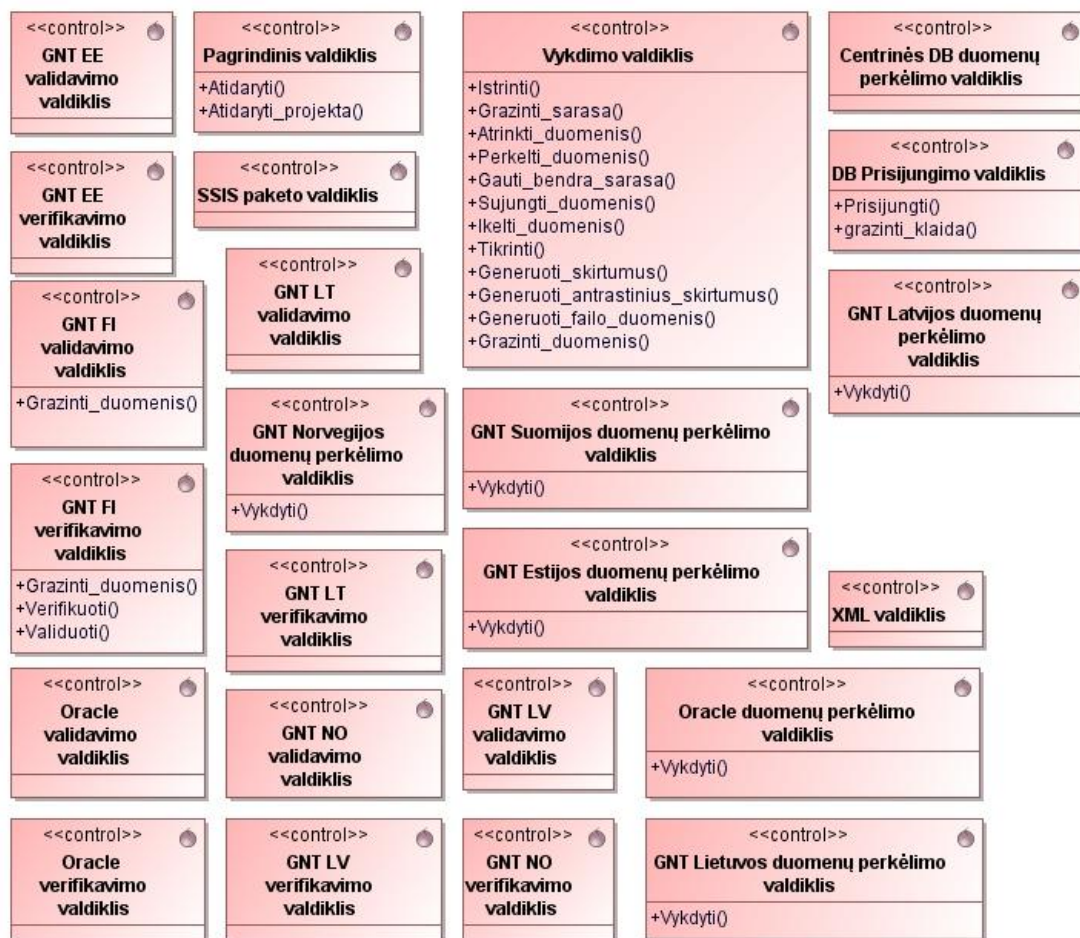
Antraeilių įvykių valdymo langas – įvykus elementui iš SSIS paketo valdymo lango, galima vykdyti žemesnio lygio elementus.

Projekto savybių navigavimo langas – pateikiamas projekte naudojamų elementų, jų savybių, kintamųjų, duomenų bazių jungčių sąrašas.

Vykdymo langas – projekto vykdymo langas ir jo savybės.

4.4.3 Veiklos paslaugos

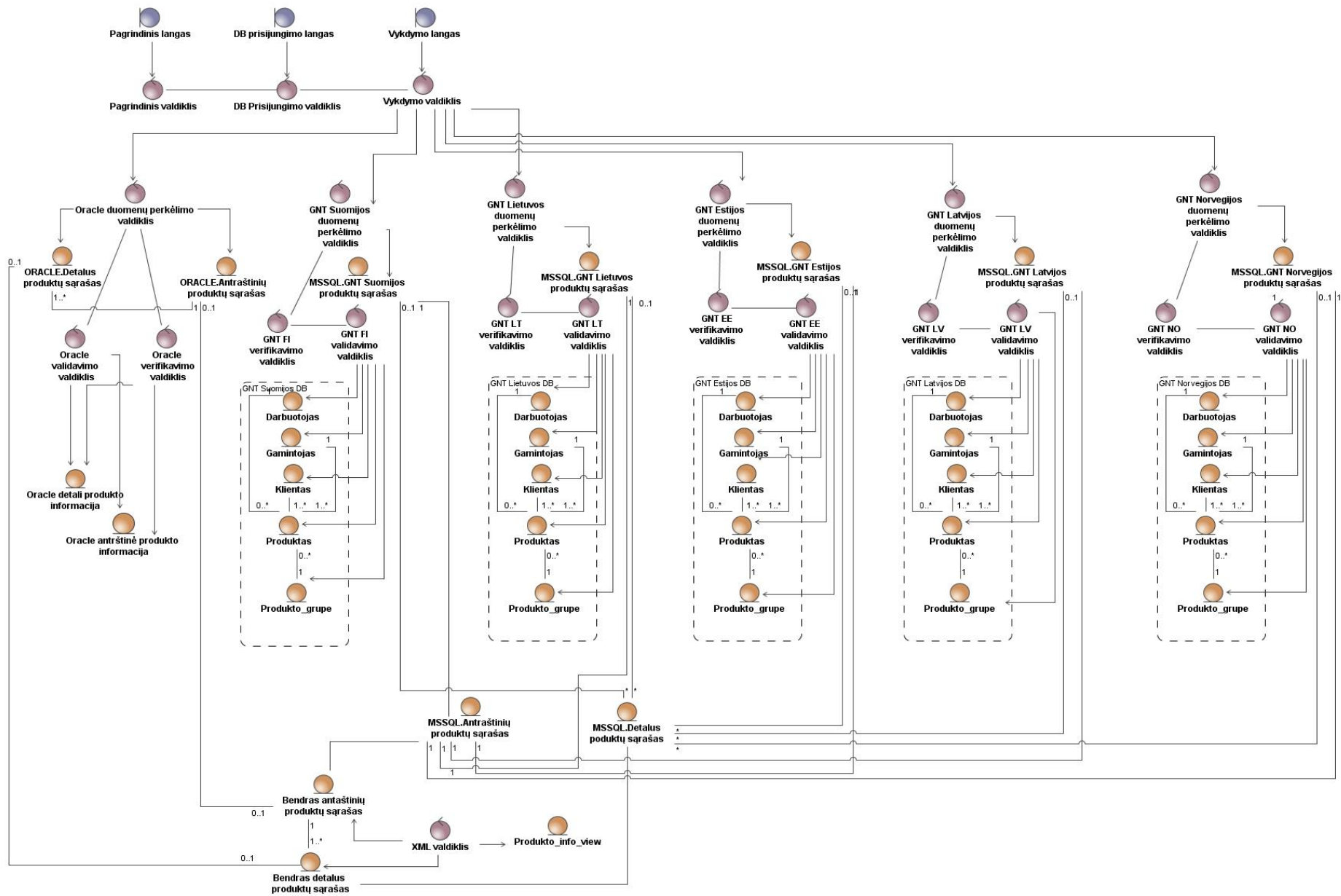
Veiklos paslaugų klasės realizuoja veiklos logiką ir yra tarpinė dalis tarp vartotojo sąsajos ir duomenų paslaugų. Valdymo klasių modelis pateiktas 31 paveiksle. Valdymo klasės yra nepriklausomos viena nuo kitos ir vadinamos valdikliais.



32 pav. Valdymo klasių modelis

4.5 Pagrindinė duomenų integravimo proceso analizės diagrama

Duomenų integravimo proceso analizės diagrama parodo valdiklius ir esybes bendradarbiaujančias tarpusavyje. DI procese duomenų gavyba vyksta iš 5 šaltinių. Duomenys patikrinami ir užkraunami į lenteles. 32 paveiksle pavaizduota duomenų integravimo proceso analizės diagrama.



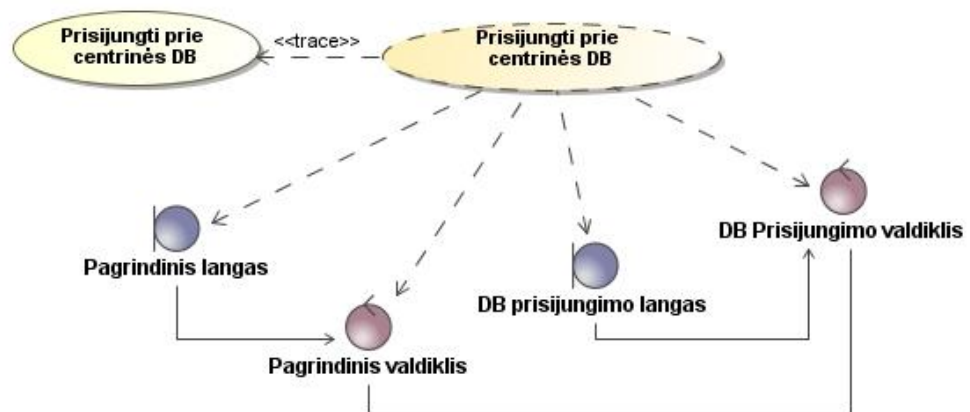
33 pav. Duomenų integravimo proceso analizės diagrama

4.6 Panaudojimo atvejų realizacijos

Panaudojimo atvejų realizacijos diagramos pateiktos kiekvienam panaudojimo atvejui (33 – 43 pav.). PA analizės diagramomis pavaizduoti sudėtingesni panaudojimo atvejai.

- ✓ PA - „Prisijungti prie centrinės DB“;

Panaudojimo atvejai: „Prisijungti prie Centrinės DB“, „Prisijungti prie Oracle DB“, „Prisijungti prie lokalsios DB“ naudojami tais pačiais valdikliais ir langais, jų navigavimas sistemoje vienodas, todėl vaizduojama tik viena PA realizacijos diagrama.

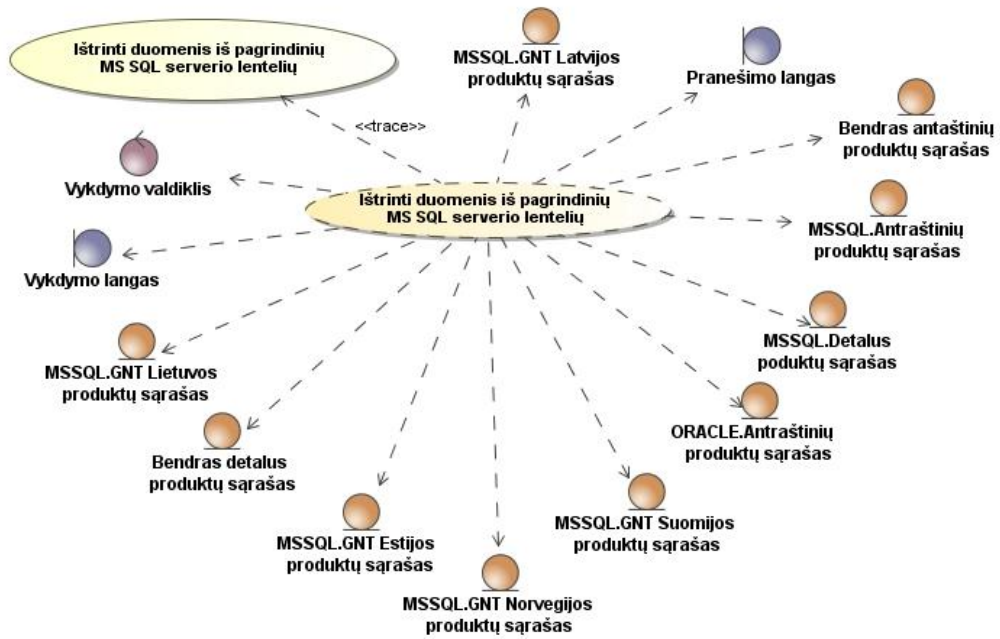


34 pav. Panaudojimo atvejo „Prisijungti prie centrinės DB“ realizacijos diagrama

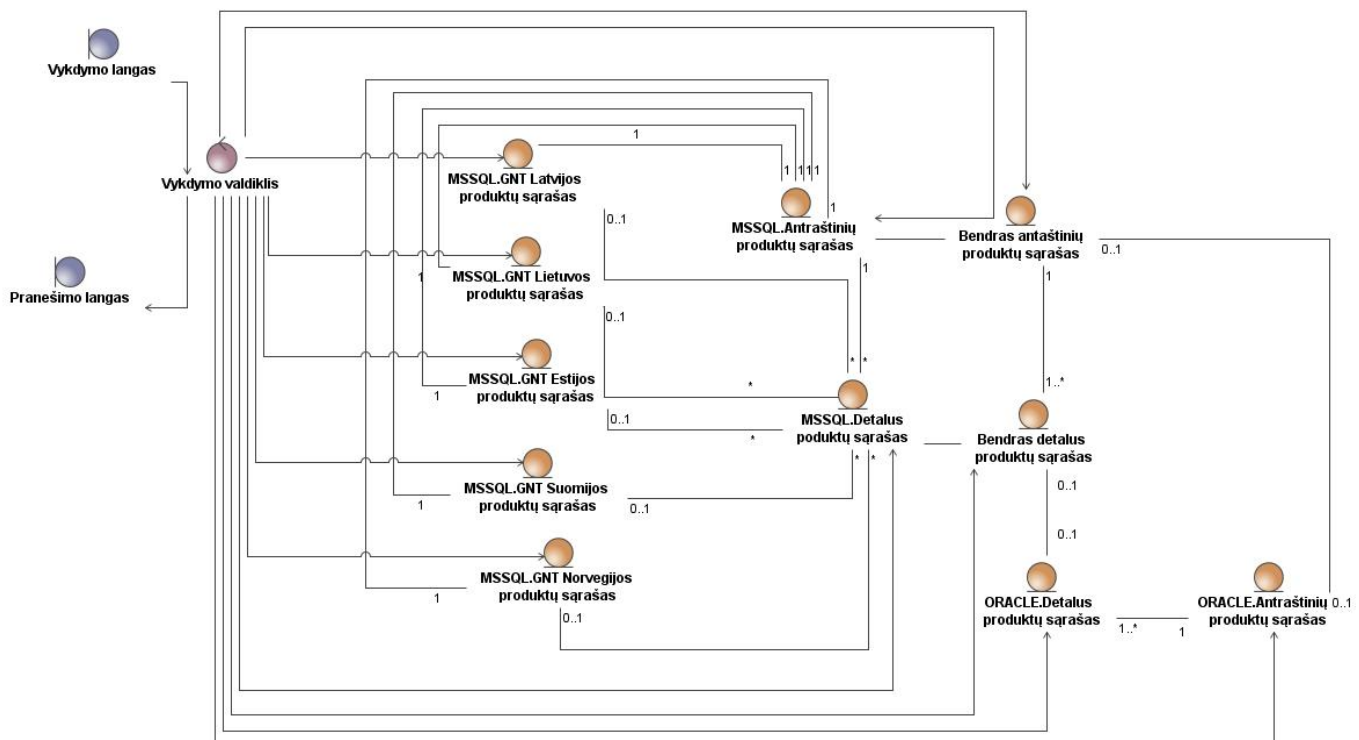
- ✓ PA – „Ištrinti duomenis iš pagrindinių MS SQL serverio lentelių“;

Kadangi šiam panaudojimo atvejui realizuoti naudojama daug esybių, todėl jų tarpusavio ryšiai pateikti analizės diagramoje (34 pav.). Šio panaudojimo atvejo tikslas – išvalyti duomenų bazės atitinkamas lenteles naujų duomenų įkėlimui. Duomenys įkeliami kiekvieną kartą vykdant duomenų integravimo procesą.

Duomenų pašalinimas vyksta buferinėse DB lentelėse, kuriose užkraunami pirminiai įrašai iš duomenų šaltinių, ir lentelėse, kuriose saugomi sujungti ir apdoroti duomenys.

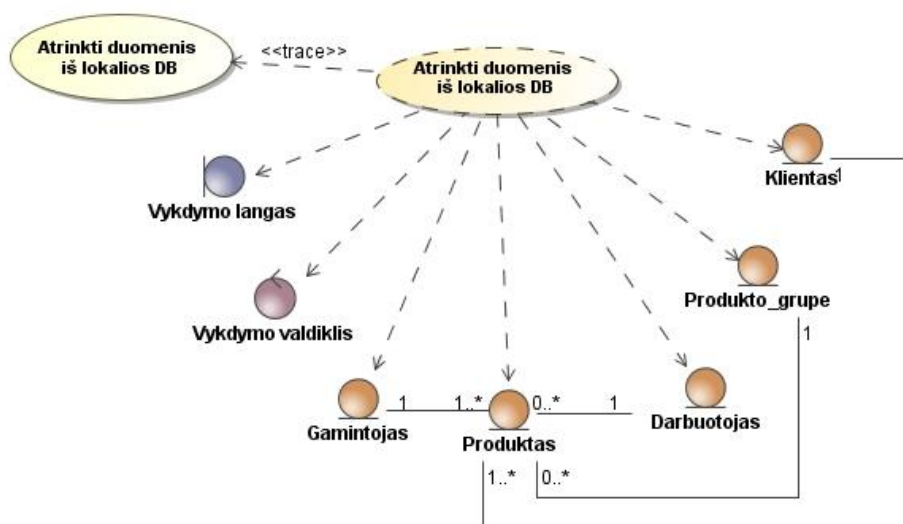


35 pav. Panaudojimo atvejo „Ištrinti duomenis iš pagrindinių MS SQL serverio lentelių“ realizacijos diagrama



36 pav. Panaudojimo atvejo „Ištrinti duomenis iš pagrindinių MS SQL serverio lentelių“ analizės diagrama

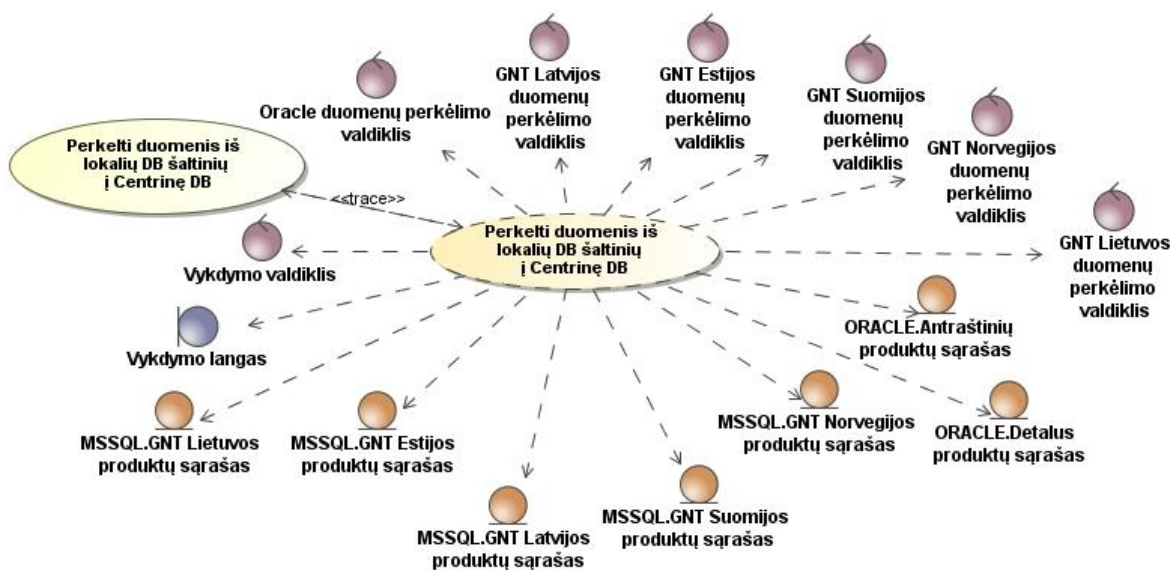
- ✓ PA – „Atrinkti duomenis iš lokaliai DB (36 pav.)“;



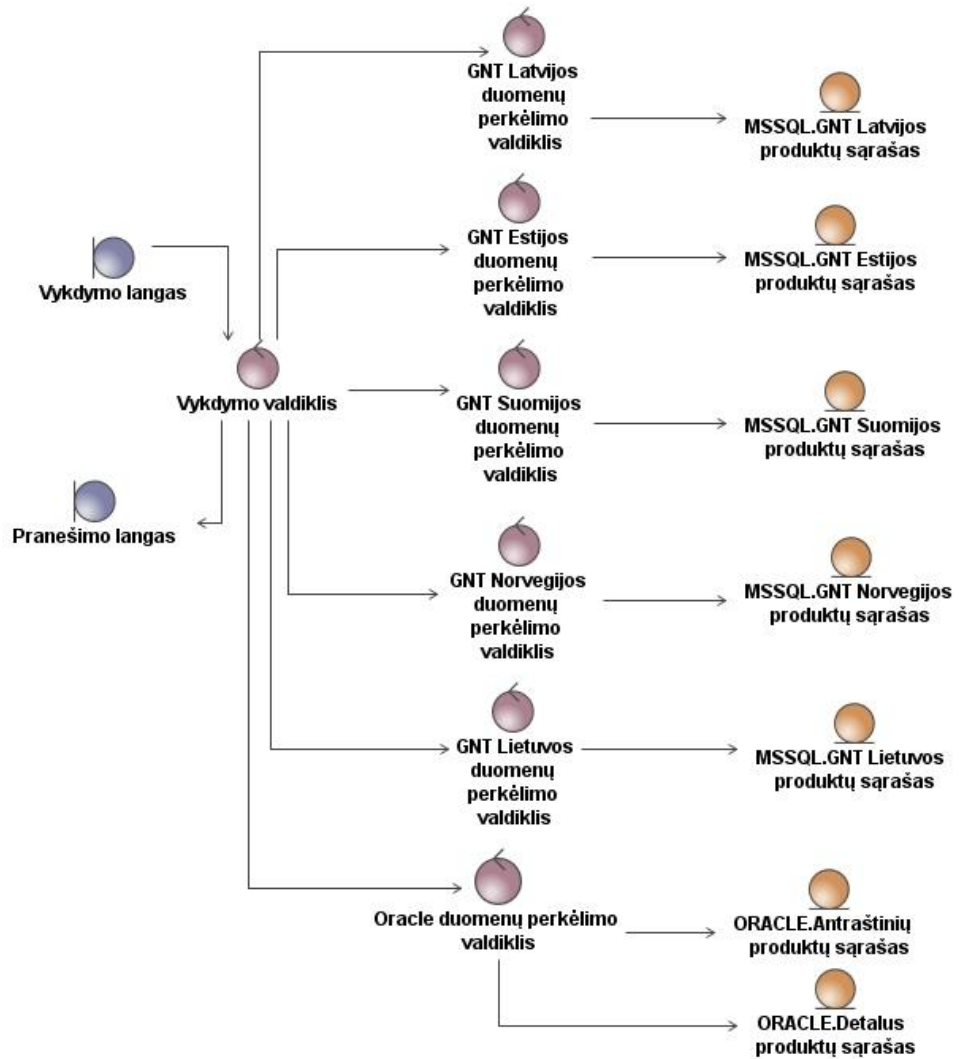
37 pav. Panaudojimo atvejo „Atrinkti duomenis iš lokalsios DB“ realizacijos diagrama

✓ PA – „Perkelti duomenis iš lokalių DB ir Oracle šaltinių į Centrinę DB“ (37 pav.);

PA detalizuotas PA analizės diagrama, kad tiksliau parodyti ryšius tarp objektų (37 paveikslas). Tai procesas, kuris sudarytas iš 6 žemesnio lygio procesų, kurie yra lygiagretūs ir vyksta vienu metu.

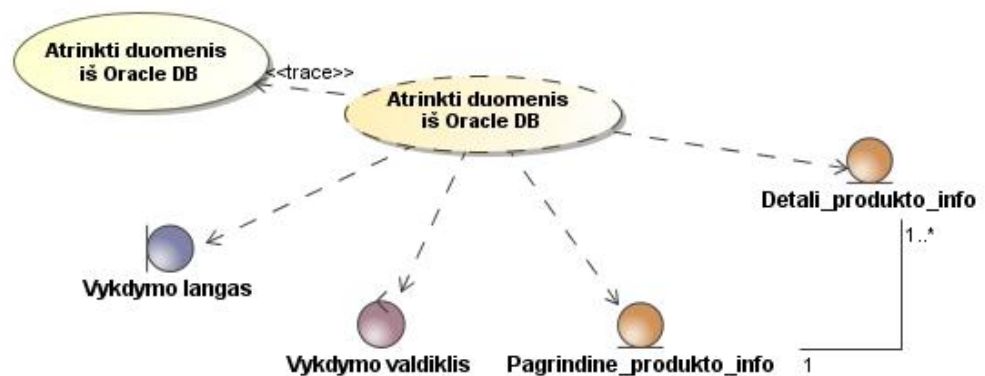


38 pav. Panaudojimo atvejo „Perkelti duomenis iš lokalių ir Oracle DB šaltinių į Centrinę DB“ realizacijos diagrama



39 pav. Panaudojimo atvejo „Perkelti duomenis iš lokalių DB ir Oracle šaltinių į Centrinę DB“ analizės diagrama

✓ PA – „Atrinkti duomenis iš Oracle DB „ (39 pav.);

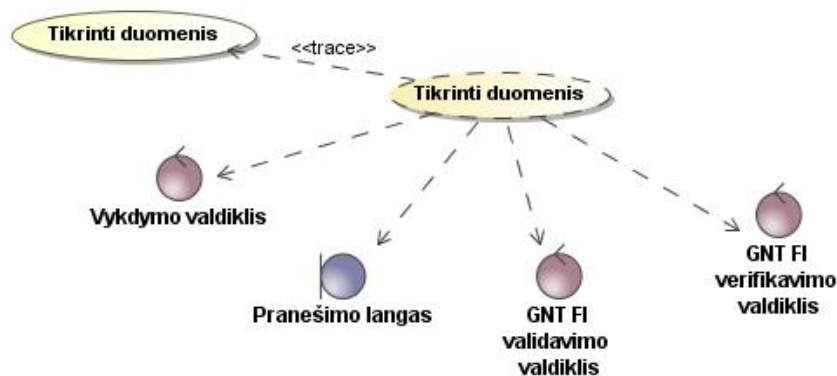


40 pav. Panaudojimo atvejo „Atrinkti duomenis iš Oracle DB“ realizacijos diagrama

✓ PA – „Tikrinti duomenis„ (40 pav.);

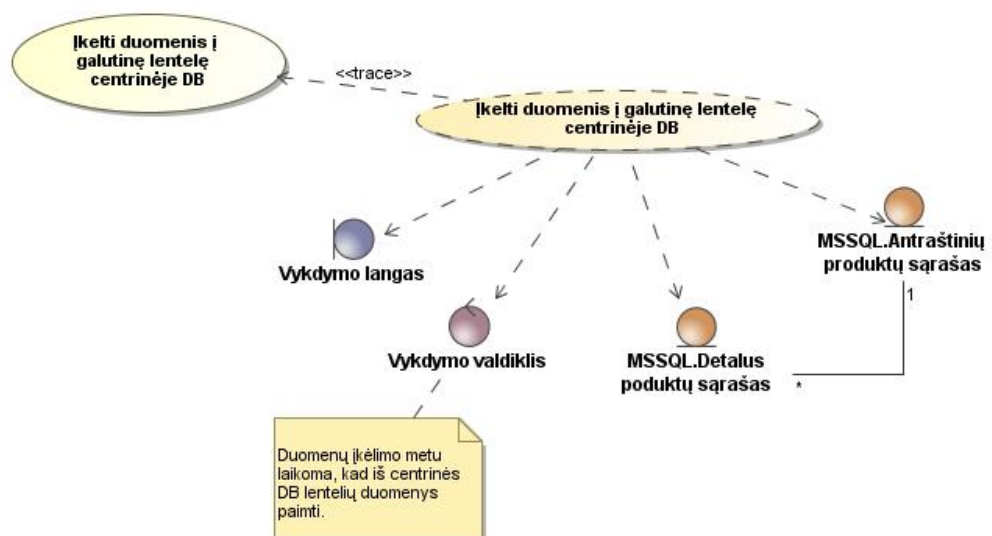
Šis panaudojimo atvejis sudarytas iš 6 žemesnio lygio procesų kaip ir PA „Perkelti duomenis iš lokalių DB ir Oracle šaltinių į Centrinę DB“, kurie vyksta lygiagrečiai, vienu metu. Žemesnio lygio procesai yra šie: „Tikrinti duomenis GNT Suomijos DB“, „Tikrinti duomenis GNT Norvegijos DB“, „Tikrinti duomenis GNT Estijos DB“, „Tikrinti duomenis GNT Latvijos DB“, „Tikrinti duomenis GNT Lietuvos DB“, „Tikrinti duomenis Oracle DB“. Kadangi jų vykdymo planas yra vienodas, dėl to pateikiama vieno žemesnio lygio proceso realizacijos diagrama.

Įkeliamų duomenų tikrinimas yra vidinis procesas. Jo vykdymas nėra tiesiogiai kontroliuojamas sistemos vartotojo. Tikrinimas yra neatsiejamas nuo duomenų integravimo proceso. Todėl kai vykdomas duomenų integravimas, automatiškai vykdomas ir duomenų tikrinimas – validavimas, verifikavimas.



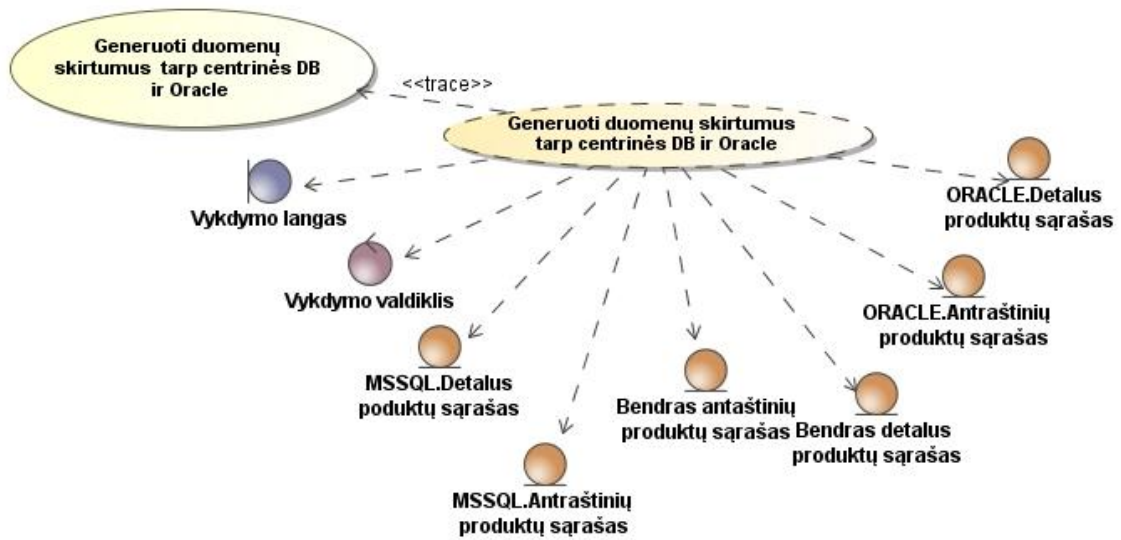
41 pav. Panaudojimo atvejo proceso „Tikrinti duomenis GNT Suomijos DB“ realizacijos diagrama

✓ PA – „Įkelti duomenis į galutinę lentelę centrinėje DB„ (41 pav.);



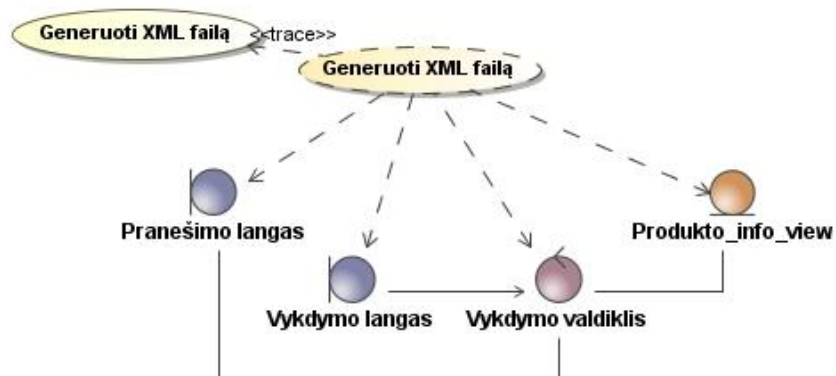
42 pav. Panaudojimo atvejo „Įkelti duomenis į galutinę lentelę centrinėje DB“ realizacijos diagrama

✓ PA – „Generuoti duomenų skirtumus tarp centrinės DB ir Oracle,, (42 pav.);



43 pav. Panaudojimo atvejo „Įkelti duomenis į galutinę lentelę centrinėje DB“ realizacijos diagrama

✓ PA – „Generuoti XML failą,, (43 pav.);

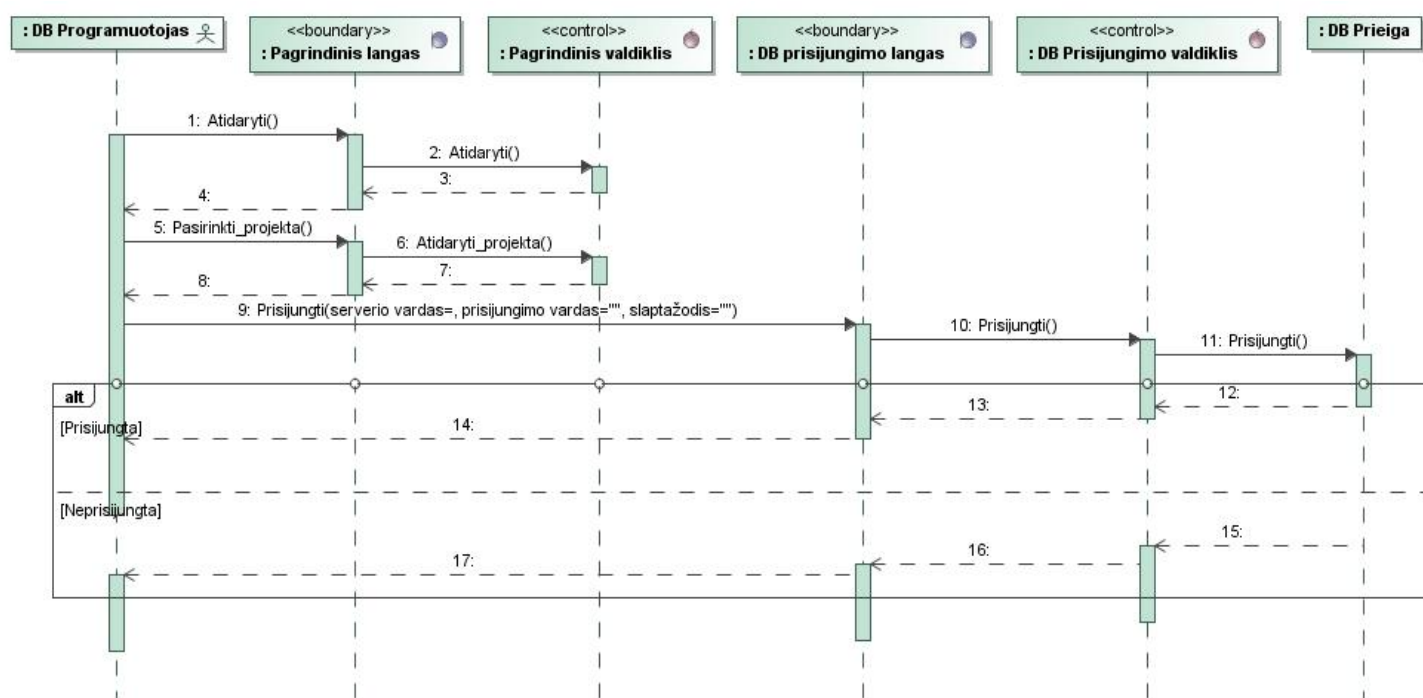


44 pav. Panaudojimo atvejo „Generuoti XML failą“ realizacijos diagrama

4.7 Duomenų integravimo elgsenos modelis

Sistemos elgsenos modelis - tai modelis, kuriame pateikiami veiksmai atliekami su sistema. Teisingas veiksmų atlikimas grąžina korektiškus duomenis. Pateiktos diagramos detalai atvaizduoja sistemos pagrindinių panaudojimų atvejų elgseną.

Panaudojimo atvejo „Prisijungti prie centrinės DB“ elgsenos diagramoje (44 pav.) pateikiamas prisijungimas prie centrinės DB per projekto aplinkos langą. Panaudojimo atvejams: „Prisijungti prie lokalsios DB“, „Prisijungti prie Oracle DB“ prisijungimas vyksta vienodai, todėl pateikiama viena prisijungimo prie DB elgsenos diagrama.



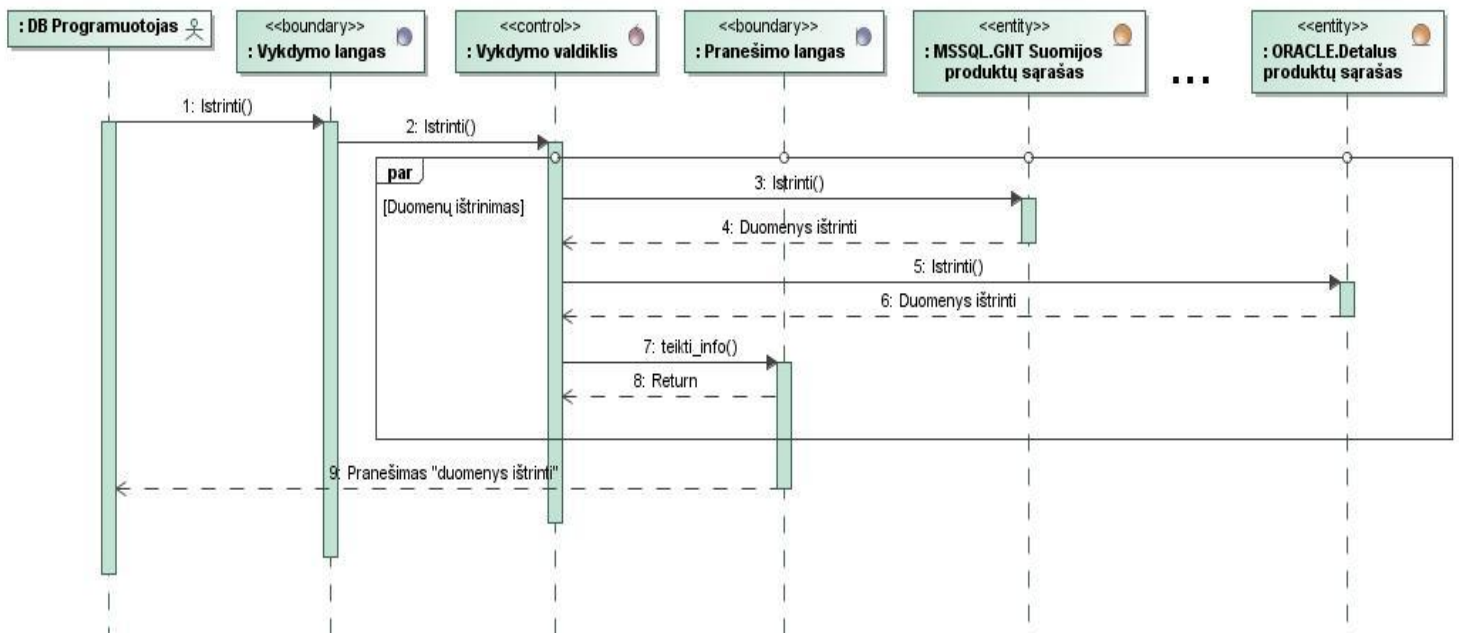
45 pav. Panaudojimo atvejo „Prisijungti prie centrinės DB“ elgsenos diagrama

Panaudojimo atvejo „Ištrinti duomenis iš pagrindinių MS SQL Server lentelių“ elgsenos diagrama (45 pav.) atvaizduoja lygiagretų duomenų pašalinimą iš atitinkamų centrinės DB lentelių. Lygiagretumas elgsenos diagramoje pažymėtas „parallel“ funkcija. Visi valdikliai, vartotojo sąsajos langai ir esybės pateikti 13 lentelėje.

13 lentelė. PA „Ištrinti duomenis iš pagrindinių MS SQL serverio lentelių“ objektai.

Tipas	Schemas elementas
Langai	Vykdyto langas
Langai	Pranešimo langas
Valdikliai	Vykdyto valdiklis
Klasė	MSSQL.GNT Suomijos produktų sąrašas

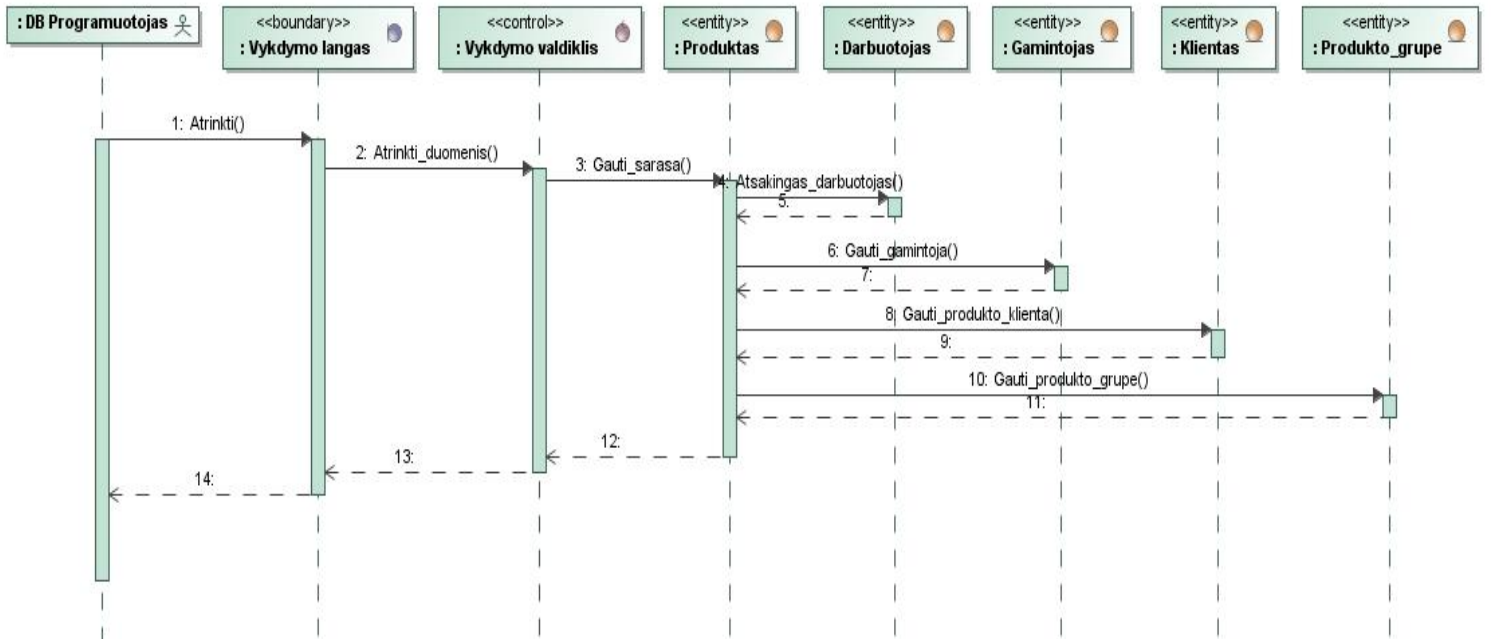
Klasė	MSSQL.GNT Estijos produktų sąrašas
Klasė	MSSQL.GNT Lietuvos produktų sąrašas
Klasė	MSSQL.GNT Latvijos produktų sąrašas
Klasė	MSSQL.GNT Norvegijos produktų sąrašas
Klasė	MSSQL.Antraštinių produktų sąrašas
Klasė	MSSQL.Detalus produktų sąrašas
Klasė	Bendras antraštinių produktų sąrašas
Klasė	Bendras detalus produktų sąrašas
Klasė	ORACLE.Antraštinių produktų sąrašas
Klasė	ORACLE.Detalus produktų sąrašas



46 pav. Panaudojimo atvejo „Ištrinti duomenis iš pagrindinių MS SQL serverio lentelių“ elgsenos diagrama

46 paveiksle pateikta panaudojimo atvejo - „Atrinkti duomenis iš lokaliai DB“ elgsenos diagrama, kuri vaizduoja duomenų atrinkimą lokaliai duomenų bazėje. Projektą sudaro 6 skirtingos DB, kuriose vykdomas toks pats duomenų atrinkimas. Laikome, kad prisijungimas į DB yra įvykęs.

Valdiklyje „vykdymo valdiklis“ yra išsaugota procedūra, kurioje vyksta duomenų sujungimas į vieną sąrašą.

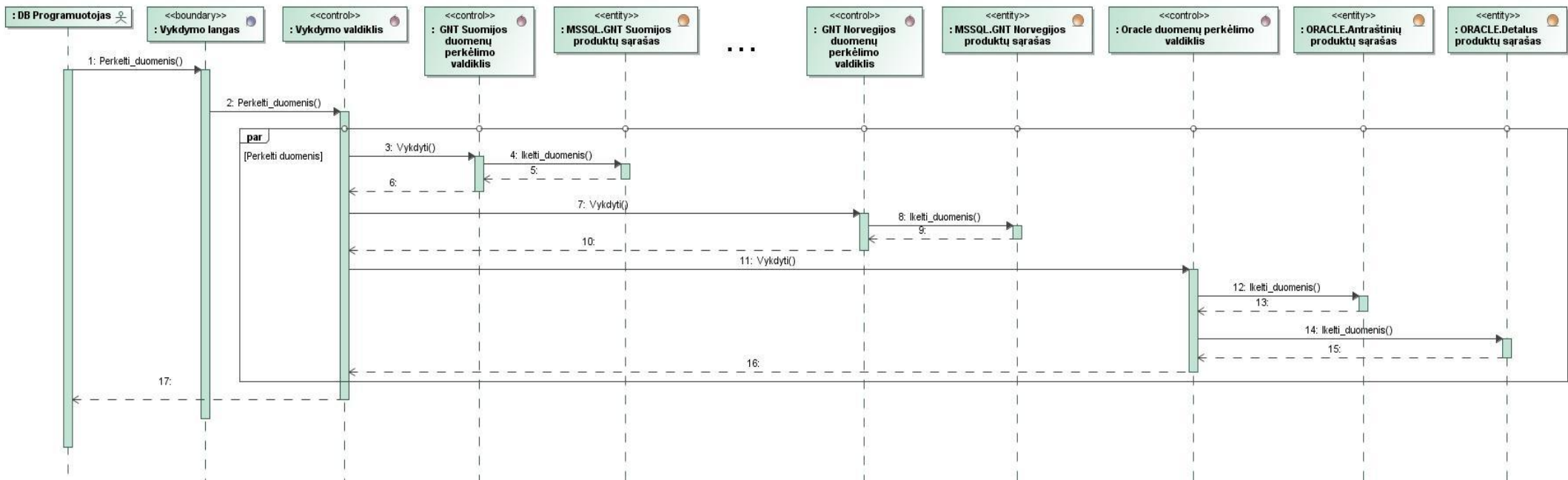


47 pav. Panaudojimo atvejo „Atrinkti duomenis iš lokalsios DB“ elgsenos diagrama

47 paveiksle pateikta panaudojimo atvejo „Perkelti duomenis iš lokalių DB šaltinių į Centrinę DB“ elgsenos diagrama. Kadangi vienoje schemoje visas esybes ir valdiklius sunku pavaizduoti, dėl to diagramoje esantis daugtaškis reiškia kitų šalių objektus. Visi objektai pateikti 14 lentelėje.

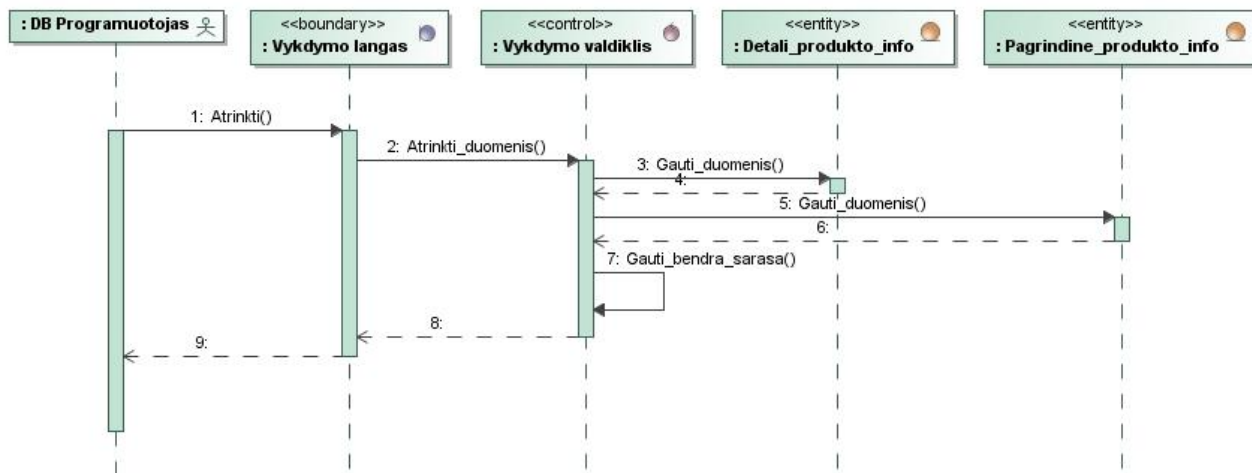
14 lentelė. PA „Perkelti duomenis iš lokalių DB šaltinių į Centrinę DB“ objektai

Tipas	Schemas elementas
Langai	Vykdyto langas
Valdikliai	Vykdyto valdiklis
Klasė	MSSQL.GNT Suomijos produktų sąrašas
Klasė	MSSQL.GNT Estijos produktų sąrašas
Klasė	MSSQL.GNT Lietuvos produktų sąrašas
Klasė	MSSQL.GNT Latvijos produktų sąrašas
Klasė	MSSQL.GNT Norvegijos produktų sąrašas
Klasė	ORACLE.Antraštinių produktų sąrašas
Klasė	ORACLE.Detalus produktų sąrašas
Valdikliai	GNT Suomijos duomenų perkėlimo valdiklis
Valdikliai	GNT Estijos duomenų perkėlimo valdiklis
Valdikliai	GNT Latvijos duomenų perkėlimo valdiklis
Valdikliai	GNT Lietuvos duomenų perkėlimo valdiklis
Valdikliai	GNT Norvegijos duomenų perkėlimo valdiklis
Valdikliai	Oracle duomenų perkėlimo valdiklis



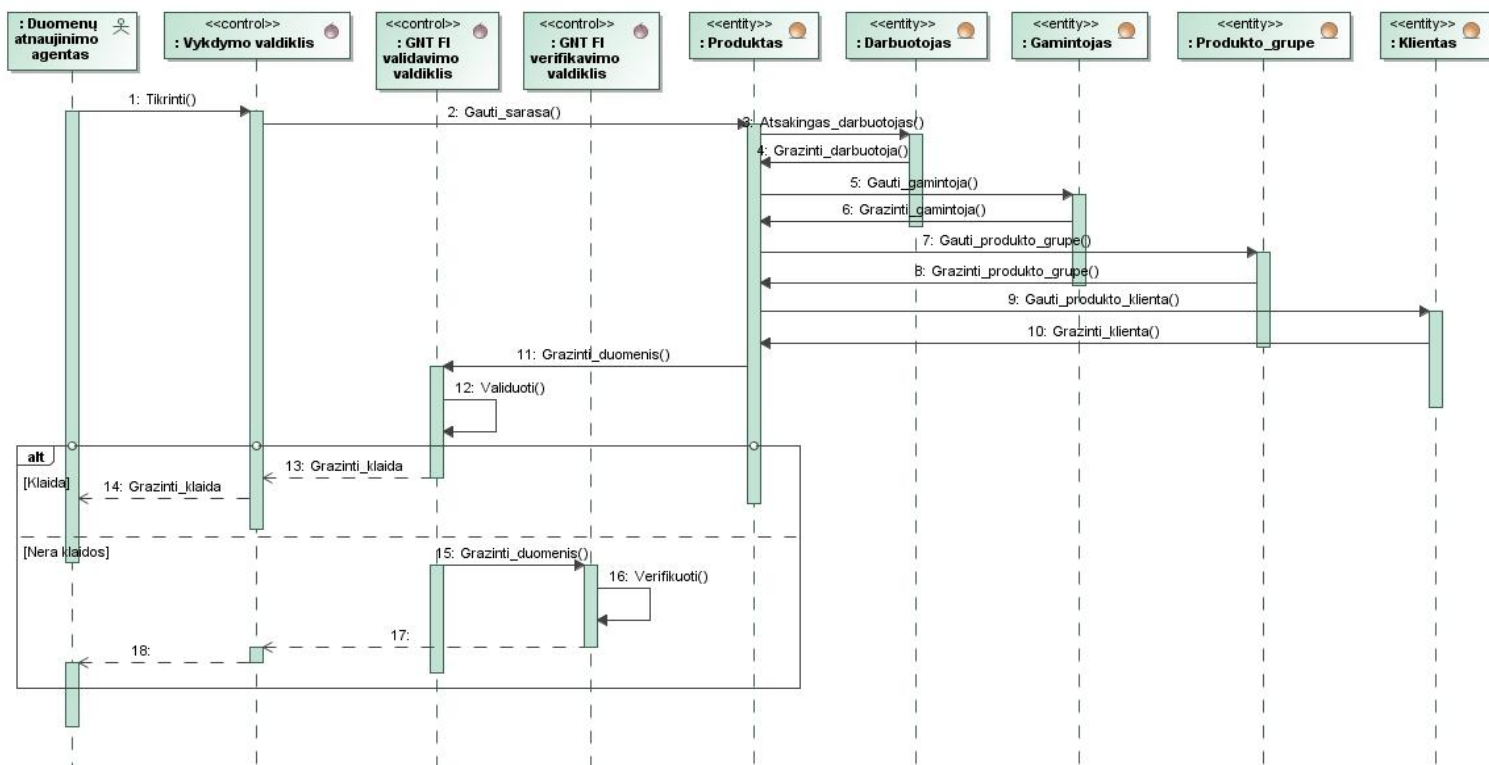
48 pav. Panaudojimo atvejo „Perkelti duomenis iš lokalių DB šaltinių į Centrinę DB“ sekų diagrama

48 paveiksle pateiktas duomenų atrinkimas iš *Oracle* duomenų bazės. *Oracle DB* lentelėse yra atrinkta informacija, todėl duomenų atrinkimas nevykdomas, vyksta duomenų perkėlimas į centrinės duomenų bazės lentelę.



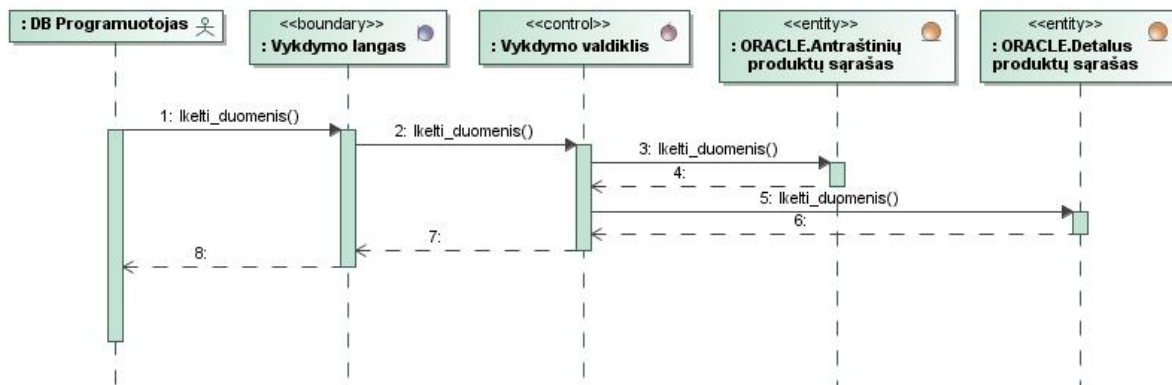
49 pav. Panaudojimo atvejo „Atrinkti duomenis iš Oracle DB“ elgsenos diagrama

49 paveiksle pateiktas „GNT Suomija“ duomenų bazės perkeliamų įrašų duomenų tikrinimas. Duomenų tikrinimo procesas yra analogiškas visuose lokaliuose duomenų integravimo procesuose, todėl pateikiama viena iš jų elgsenos diagrama.



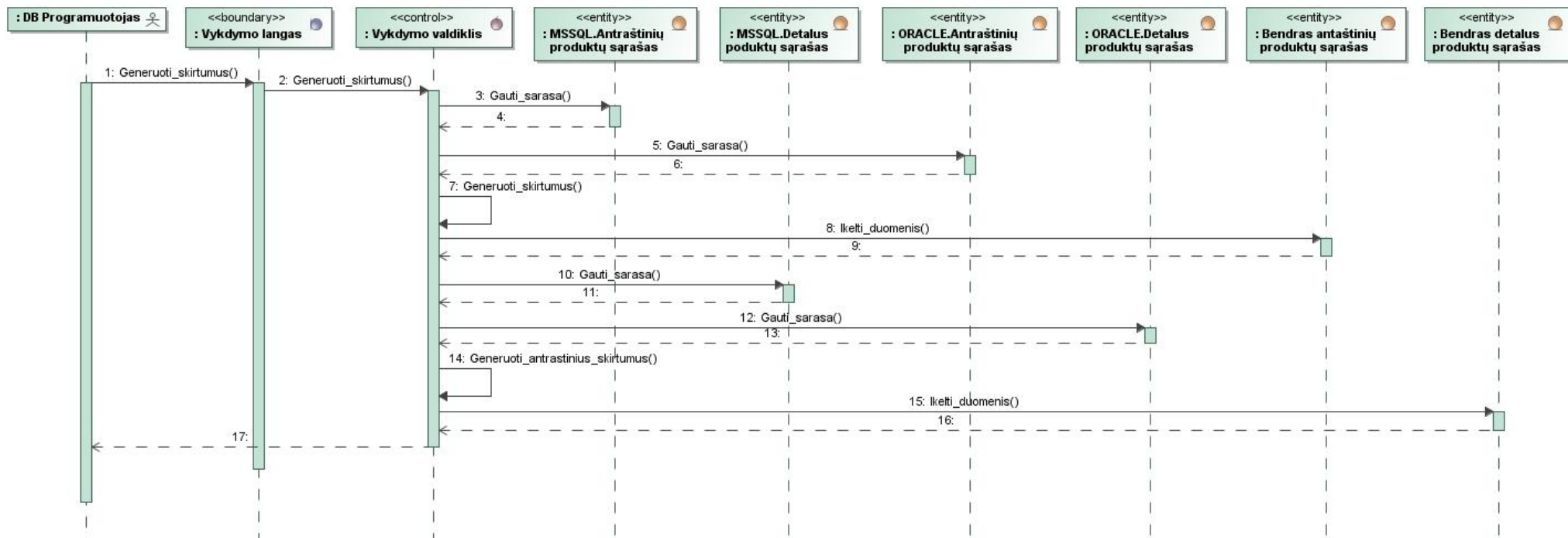
50 pav. Panaudojimo atvejo „Tikrinti duomenis [GNT FI]“ elgsenos diagrama

50 paveiksle pateikta duomenų įrašymo į galutinę DB lentelę elgsenos diagrama.



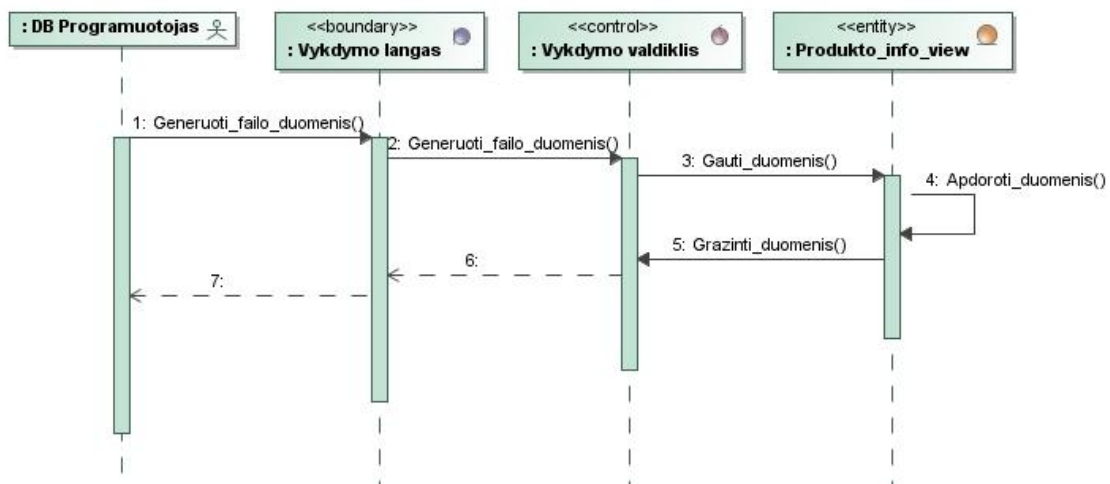
51 pav. Panaudojimo atvejo „Įkelti duomenis į galutinę lentelę centrinėje DB“ sekų diagrama

51 paveiksle pavaizduotas skirtumų generavimo procesas. Elgsenos diagramoje nuosekliai pateikti sistemos žingsniai generuojant bendrą skirtumų sąrašą naudojantis jau suintegruotais duomenimis iš lokalių duomenų bazių. Proceso tikslas – sugeneruoti sąrašą skirtumų, kurie vienoje DB yra, o kitose – ne, ir pateikti jį XML failo sudarymui.



52 pav. Panaudojimo atvejo „Generuoti duomenų skirtumus tarp centrinės DB ir Oracle“ elgsenos diagrama

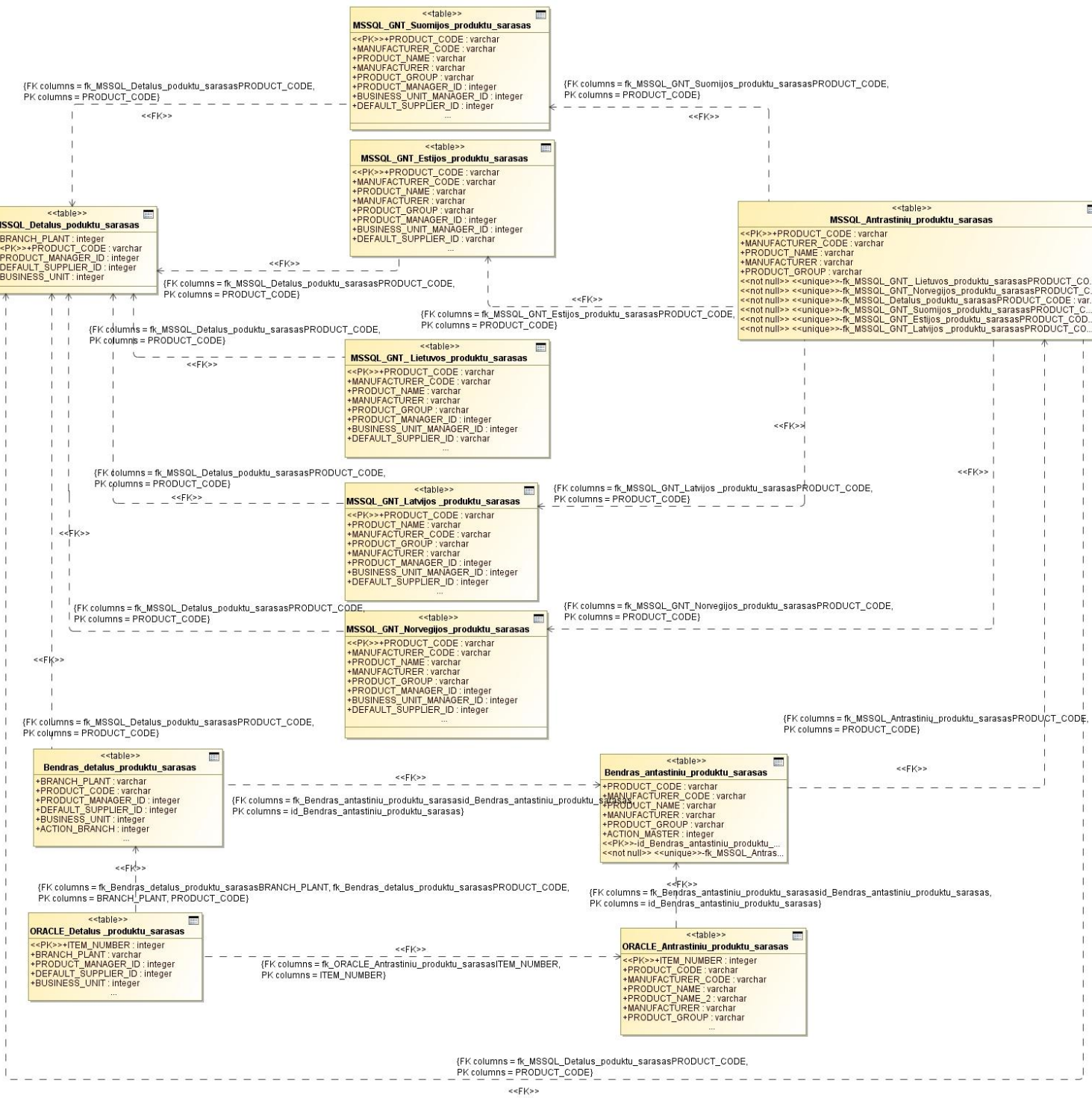
52 paveiksle pavaizduotas XML failo generavimas.



53 pav. Panaudojimo atvejo „Generuoti XML failą“ elgsenos diagrama

4.8 Duomenų bazės schema

Duomenų bazės schema pavaizduota 53 paveiksle. Pateiktas DB lentelių detalus aprašas.



54 pav. Detalaus projekto duomenų bazės schema

15 lentelė. Duomenų bazės lentelės „MSSQL_GNT_Suomijos_produkto_sarasas“ specifikacija

Lauko pavadinimas	Tipas	Aprašymas
PRODUCT_CODE	Varchar(20)	Unikalus produkto kodas (SKU)
MANUFACTURER_CODE	Varchar(20)	Unikalus gamintojo suteiktas kodas
PRODUCT_NAME	Varchar(60)	Produkto pavadinimas
MANUFACTURER	Char(3)	Gamintojo sutrumpintas pavadinimas
PRODUCT_GROUP	Char(3)	Produkto grupės sutrumpintas pavadinimas
PRODUCT_MANAGER_ID	Int	Produkto vadybininko ID
BUSINESS_UNIT_MANAGER_ID	Int	
DEFAULT_SUPPLIER_ID	Int	Tiekėjo ID

16 lentelė. Duomenų bazės lentelės „MSSQL_GNT_Estijos_produkto_sarasas“ specifikacija

Lauko pavadinimas	Tipas	Aprašymas
PRODUCT_CODE	Varchar(20)	Unikalus produkto kodas (SKU)
MANUFACTURER_CODE	Varchar(20)	Unikalus gamintojo suteiktas kodas
PRODUCT_NAME	Varchar(60)	Produkto pavadinimas
MANUFACTURER	Char(3)	Gamintojo sutrumpintas pavadinimas
PRODUCT_GROUP	Char(3)	Produkto grupės sutrumpintas pavadinimas
PRODUCT_MANAGER_ID	Int	Produkto vadybininko ID
BUSINESS_UNIT_MANAGER_ID	Int	
DEFAULT_SUPPLIER_ID	Int	Tiekėjo ID

17 lentelė. Duomenų bazės lentelės „MSSQL_GNT_Latvijos_produkto_sarasas“ specifikacija

Lauko pavadinimas	Tipas	Aprašymas
PRODUCT_CODE	Varchar(20)	Unikalus produkto kodas (SKU)
MANUFACTURER_CODE	Varchar(20)	Unikalus gamintojo suteiktas kodas
PRODUCT_NAME	Varchar(60)	Produkto pavadinimas
MANUFACTURER	Char(3)	Gamintojo sutrumpintas pavadinimas
PRODUCT_GROUP	Char(3)	Produkto grupės sutrumpintas pavadinimas
PRODUCT_MANAGER_ID	Int	Produkto vadybininko ID
BUSINESS_UNIT_MANAGER_ID	Int	
DEFAULT_SUPPLIER_ID	Int	Tiekėjo ID

18 lentelė. Duomenų bazės lentelės „MSSQL_GNT_Lietuvos_produkto_sarasas“ specifikacija

Lauko pavadinimas	Tipas	Aprašymas
PRODUCT_CODE	Varchar(20)	Unikalus produkto kodas (SKU)
MANUFACTURER_CODE	Varchar(20)	Unikalus gamintojo suteiktas kodas
PRODUCT_NAME	Varchar(60)	Produkto pavadinimas
MANUFACTURER	Char(3)	Gamintojo sutrumpintas pavadinimas
PRODUCT_GROUP	Char(3)	Produkto grupės sutrumpintas pavadinimas
PRODUCT_MANAGER_ID	Int	Produkto vadybininko ID
BUSINESS_UNIT_MANAGER_ID	Int	
DEFAULT_SUPPLIER_ID	Int	Tiekėjo ID

19 lentelė. Duomenų bazės lentelės „MSSQL_GNT_Norvegijos_produkto_sarasas“ specifikacija

Lauko pavadinimas	Tipas	Aprašymas
PRODUCT_CODE	Varchar(20)	Unikalus produkto kodas (SKU)
MANUFACTURER_CODE	Varchar(20)	Unikalus gamintojo suteiktas kodas
PRODUCT_NAME	Varchar(60)	Produkto pavadinimas
MANUFACTURER	Char(3)	Gamintojo sutrumpintas pavadinimas
PRODUCT_GROUP	Char(3)	Produkto grupės sutrumpintas pavadinimas
PRODUCT_MANAGER_ID	Int	Produkto vadybininko ID
BUSINESS_UNIT_MANAGER_ID	Int	
DEFAULT_SUPPLIER_ID	Int	Tiekėjo ID

20 lentelė. Duomenų bazės lentelės „MSSQL_Antrastiniu_produkto_sarasas“ specifikacija

Lauko pavadinimas	Tipas	Aprašymas
PRODUCT_CODE	Varchar(20)	Unikalus produkto kodas (SKU)
MANUFACTURER_CODE	Varchar(20)	Unikalus gamintojo suteiktas kodas
PRODUCT_NAME	Varchar(60)	Produkto pavadinimas
MANUFACTURER	Char(3)	Gamintojo sutrumpintas pavadinimas
PRODUCT_GROUP	Char(3)	Produkto grupės sutrumpintas pavadinimas

21 lentelė. Duomenų bazės lentelės „MSSQL_detalus_produkto_sarasas“ specifikacija

Lauko pavadinimas	Tipas	Aprašymas
BRANCH_PLANT	SmallInt	Šalies padalinio kodas (vidinis)
PRODUCT_CODE	Varchar(20)	Unikalus produkto kodas (SKU)
PRODUCT_MANAGER_ID	Varchar(12)	Produkto vadybininko ID
DEFAULT_SUPPLIER_ID	Int	Tiekėjo ID
BUSINESS_UNIT	Char(3)	

22 lentelė. Duomenų bazės lentelės „ORACLE_detalus_produkto_sarasas“ specifikacija

Lauko pavadinimas	Tipas	Aprašymas
ITEM_NUMBER	Int	Unikalus lentelės generuojamas kodas (ID)
BRANCH_PLANT	Nvarchar(12)	Šalies padalinio kodas (vidinis)
PRODUCT_MANAGER_ID	Nvarchar(20)	Produkto vadybininko ID
DEFAULT_SUPPLIER_ID	Int	Tiekėjo ID
BUSINESS_UNIT	Nchar(3)	

23 lentelė. Duomenų bazės lentelės „ORACLE_Antrastiniu_produkto_sarasas“ specifikacija

Lauko pavadinamas	Tipas	Aprašymas
ITEM_NUMBER	Int	Unikalus lentelės generuojamas kodas (ID)
PRODUCT_CODE	Nvarchar(20)	Unikalus produkto kodas (SKU)
MANUFACTURER_CODE	Nvarchar(20)	Unikalus gamintojo suteiktas kodas
PRODUCT_NAME	Nvarchar(30)	Produkto pavadinimas (pirmi 30 simbolių)
PRODUCT_NAME_2	Nvarchar(30)	Produkto pavadinimas (po 30-tojo simbolio dar 30 simbolių)
MANUFACTURER	Nchar(3)	Gamintojo sutrumpintas pavadinimas
PRODUCT_GROUP	Nchar(3)	Produkto grupės sutrumpintas pavadinimas

24 lentelė. Duomenų bazės lentelės „Bendras_detalus_produkto_sarasas“ specifikacija

Lauko pavadinamas	Tipas	Aprašymas
BRANCH_PLANT	SmallInt	Šalies padalinio kodas (vidinis)
PRODUCT_CODE	Varchar(20)	Unikalus produkto kodas (SKU)
PRODUCT_MANAGER_ID	Varchar(12)	Produkto vadybininko ID
DEFAULT_SUPPLIER_ID	Int	Tiekėjo ID
BUSINESS_UNIT	char(3)	
ACTION_BRANCH	Int	Produkto tipas

25 lentelė. Duomenų bazės lentelės „Bendras_Antrastiniu_produkto_sarasas“ specifikacija

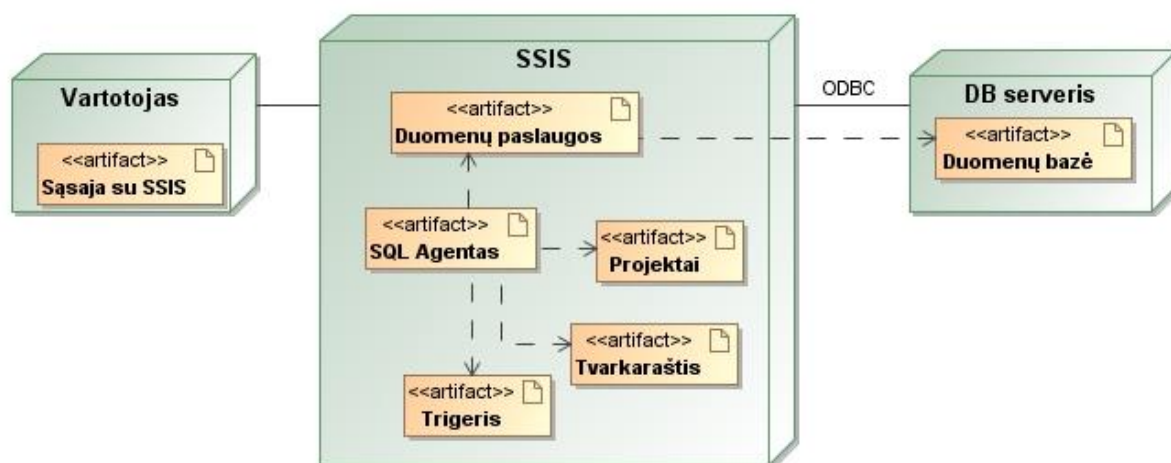
Lauko pavadinamas	Tipas	Aprašymas
PRODUCT_CODE	Varchar(20)	Unikalus produkto kodas (SKU)
MANUFACTURER_CODE	Varchar(20)	Unikalus gamintojo suteiktas kodas
PRODUCT_NAME	Varchar(60)	Produkto pavadinimas
MANUFACTURER	Char(3)	Gamintojo sutrumpintas pavadinimas
PRODUCT_GROUP	Char(3)	Produkto grupės sutrumpintas pavadinimas
ACTION_MASTER	Int	Antraštinio produkto tipas

5. Duomenų integravimo posistemio realizacija ir testavimas

Duomenų integravimo posistemio proceso esmė – perkelti ir sujungti informaciją apie produktus, esančius skirtingose šalyse, į centrinę duomenų bazę tam, kad būtų galima atlikti verslui aktualią analizę. Nustatyti produktų informacijos skirtumus tarp *Oracle* ir Centrinės DB tam, kad būtų galima užtikrinti informacijos korektiškumą. Integravimo procesu gautus duomenis pateikti vartotojams ataskaitų pavidale.

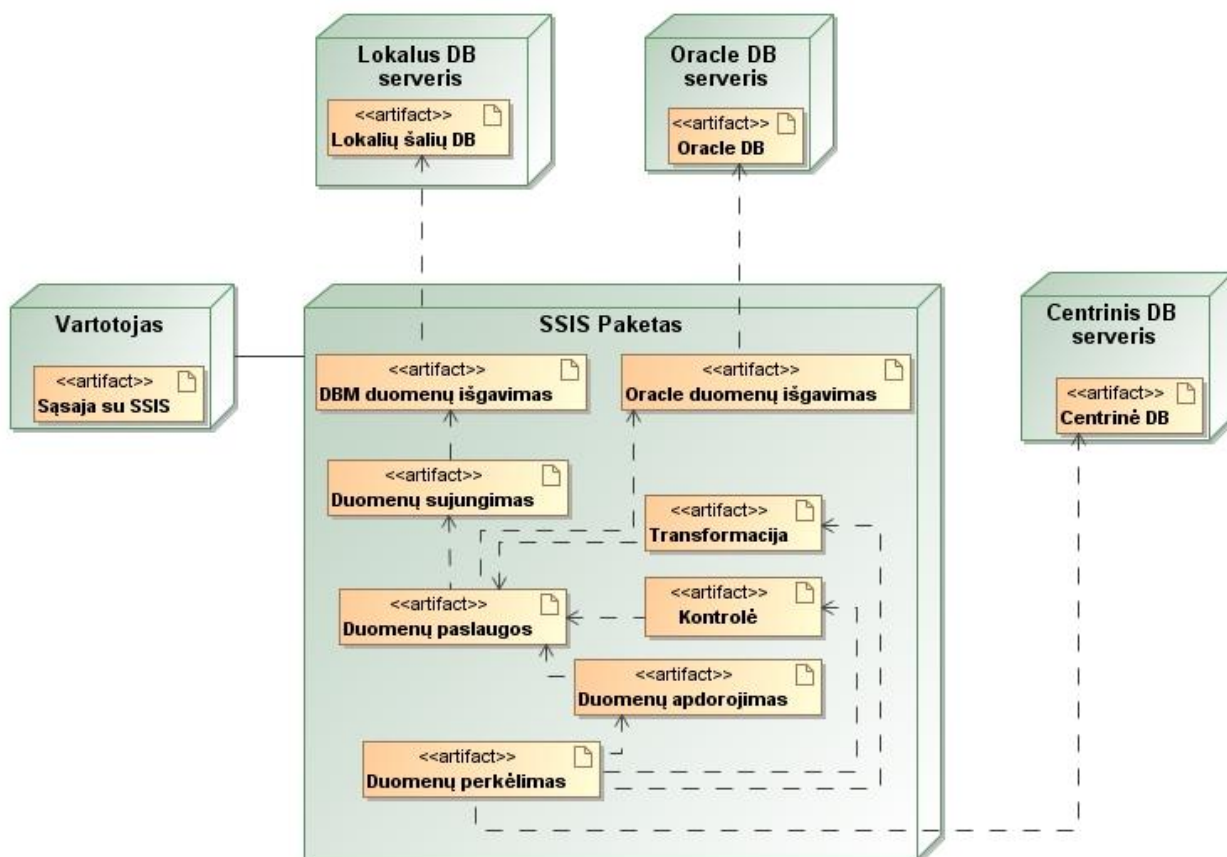
5.1 Duomenų integravimo posistemio realizacijos modelis

Duomenų integravimo procesui realizuoti naudojama MS SQL programavimo kalba ir *SQL Server Integration Services* įrankis. Duomenų bazė realizuojama naudojant *MS SQL Server 2000*, *MS SQL Server 2005* ir *Oracle* serverius. 54 paveiksle pateiktas DI proceso diegimo modelis, kuris apima duomenų integravimo procesų valdymą. Kadangi sistemą sudaro daugybę procesų, būtina kontroliuoti jų vykdymą. Visiems procesams priskiriami atitinkami prioritetai, kad jie vyktų nuosekliai, vienas paskui kitą.



55 pav. Duomenų integravimo proceso diegimo modelis

Duomenų integravimo posistemio komponentų diegimo diagrama pateikta 55 paveiksle. Vartotojas SSIS pagalba iškviečia reikalingas funkcijas. Duomenų išgavimo metu duomenys traukiami iš *Oracle* ir iš lokalaus DB šaltinių, duomenų perkėlimo metu krepiamasi į DB, esančią centriniame DB serveryje.



56 pav. Duomenų integravimo posistemio diegimo modelis

5.2 Duomenų integravimo projektų aplinkos

Duomenų integravimo projektai kuriami ir vykdomi dvejose skirtingose aplinkose: *MS SQL Server 2000* ir *MS SQL Server 2005* terpėse.

Projektų tikslas - įvertinti, kurioje platformoje DI procesas yra greitesnis, stabilesnis, sunaudoja mažiau resursų.

5.2.1 MS SQL Server 2000 aplinka

Enterprise Manager yra pagrindinė *MS SQL Server 2000* įdiegtų įrankių administravimo aplinka. Ji tiekia grafinį vaizdą visų SQL Server įrankių, esančių tinkle. *Enterprise Manager* aplinkoje galima vykdyti aukšto lygio administravimo funkcijas, kurios įtakoja vieną ar kelis serverius, nustatyti bendrus užduočių tvarkaraščius, kurti ir modifikuoti atskirų duomenų bazių struktūrą.

Data Transformation Services paslauga teikia lankstų būdą duomenis integruoti, eksportuoti. DTS turi daug elementų, kurie leidžia išgauti, transformuoti ir sujungti duomenis iš skirtingų šaltinių į vieną ar kelias paskirties (destination) vietas. Įrankiais galima sukurti specializuotą duomenų

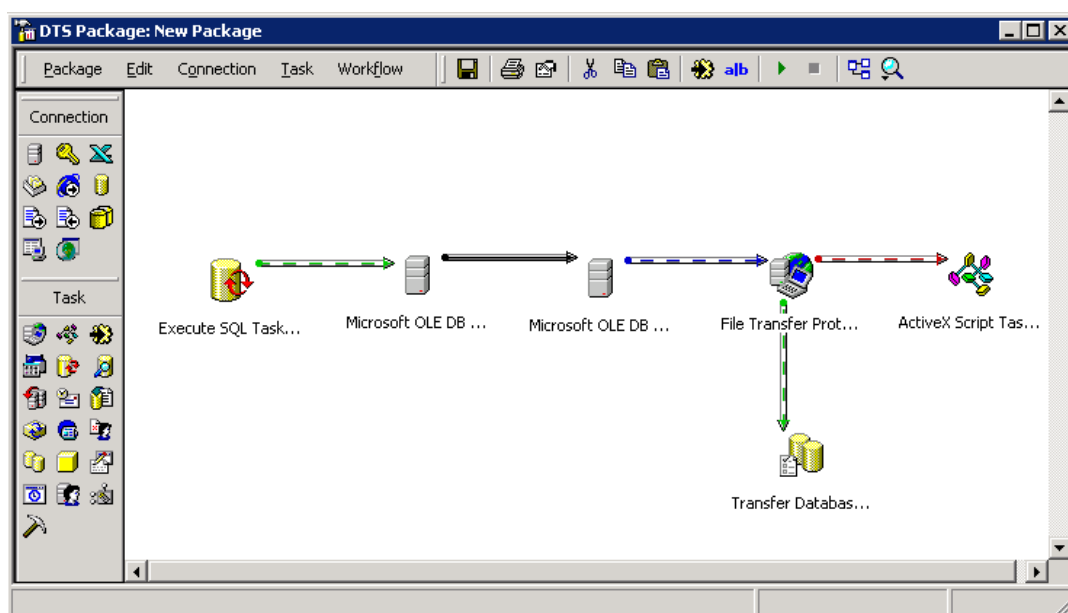
integravimo sprendimą, kuris reikalingas organizacijai. Kiekvienas DI sprendimas gali turėti nustatytą tvarkaraštį, kurio pagalba jis vykdomas automatiškai. DI paketas gali būti išsaugotas *Microsoft Visual Basic* bylomis.

DTS projektavimo aplinka grafiškai realizuoja DTS objektinį modelį, leidžiantį grafiškai sukurti DTS paketą. Jis gali būti paprastas arba sudarytas iš daugiau nei vieno žingsnio. Procesas gali turėti sudėtingą įvykių seką sudarytą iš sudėtinių žingsnių, naudojant sąlygas arba keletą skirtingų duomenų šaltinių.

DI proceso projektavimo aplinka sudaryta iš darbo ploto, skirto paketo turinio kūrimui, įrankių panelės, kurioje yra visi elementai, ir meniu, kuriame yra įvykių sekos tipai ir bendrosios paketo valdymo komandos.

Detalesnė informacija apie *MS SQL Server 2000 DTS Designer* įrankį pateikta 1 priede, kuriame yra elementų aprašymai, DTS paketo sukūrimas ir navigacija.

56 paveiksle pateiktas DTS dizainerio grafinė vartotojo sąsaja.



57 pav. DTS Designer vartotojo grafinė sąsaja

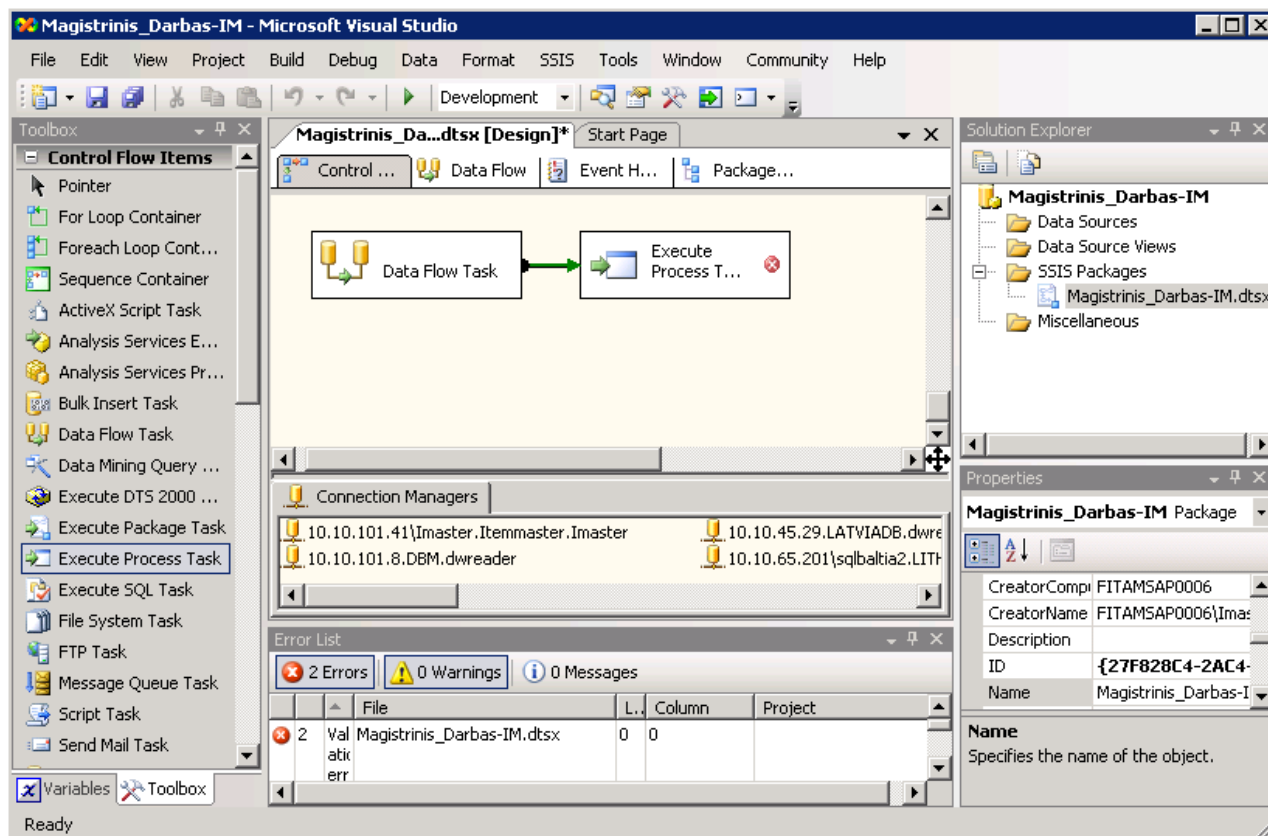
5.2.2 MS SQL Server 2005 aplinka

MS SQL Server 2005 aplinkoje sukurtas atskiras modulis, kuris naudojamas ETL procesams kurti. Naujas ETL procesas kuriamas naudojant *Business Intelligence Development Studio* įrankį, kuris palaiko SSIS paslaugą.

MS SQL Server 2005 versija yra patogi su ja dirbančiam programuotojui, IS analitikui. Įvairūs sistemos ir vartotojo sukurti kintamieji, kurie veikia viename arba keliuose projektuose, konfigūraciniai

failai, greitas kodo keitimas bei daugelis kitų patobulinimų naujoje versijoje, padeda vartotojui labiau susikoncentruoti į užduoties esmę [9].

57 paveiksle pateikta SSIS projektavimo aplinkos vartotojo sąsaja.

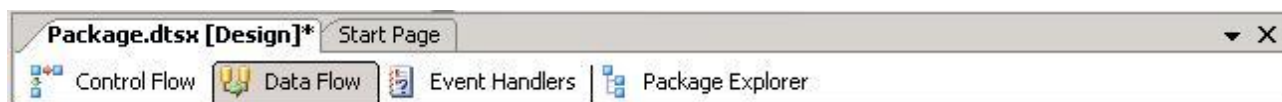


58 pav. SSIS Designer vartotojo grafinė sąsaja

Pagrindinė projekto dalis padalinta į keturių sekcijų valdymą (58 pav.):

- *Control Flow* - valdymo skiltyje formuojamas procesas. Sudedami visi elementai reikalingi projekto logikai išpildyti.
- *Data Flow* – skiltis skirta duomenų perkėlimo veiksams atlikti.
- *Event Handlers* - elemento antriniam veiksams formuoti.
- *Package Explorer* – skiltis yra medžio struktūros. Tai turinys visų panaudotų elementų, parametrų, nustatymų.

58 paveiksle pateiktas projekto valdymo skiltys.



59 pav. Projekto valdymo sekcijos

Detali informacija apie *MS SQL Server 2005 SSIS* projektavimo aplinką pateikta 2 priede. Priede yra visų elementų aprašymai, SSIS paketo kūrimo etapai, navigavimas projekte [14].

5.3 Duomenų integravimo proceso veikimo aprašas

Duomenų integravimo procesas sujungia duomenis iš 5 šaltinių. Šaltiniai yra *MS SQL Server 2000* ir *MS SQL Server 2005*.

Duomenų gavyba vyksta lokaliuose duomenų bazėse, kuriose DB lentelių, naudojamų DI procese, struktūra yra vienoda. Duomenų sąrašas generuojamas užklausos pagalba, kurią sudaro 5 lentelės: „Produktas“, „Gamintojas“, „Produkto grupė“, „Darbuotojas“, „Klientas“. Paimtiems įrašams iš lokalių DB yra ieškomi atitikmenys buferinėje DB lentelėje. Duomenų perkėlimo elementu, duomenys įkeliami į buferinę lentelę. Kiekvieno įrašymo metu buferinė lentelė yra tuščia.

Antrame proceso etape vyksta duomenų sujungimas į vieną sąrašą. Iškviečiama užklausa, kuri paruošia duomenis jų įkėlimui į galutinę paskirties vietą. Duomenų paskirties vieta, prieš įrašant duomenis, yra tuščia.

Trečias etapas – XML dokumento generavimas. Duomenys esantys galutinėje paskirties vietoje įtraukiami į XML dokumentą.

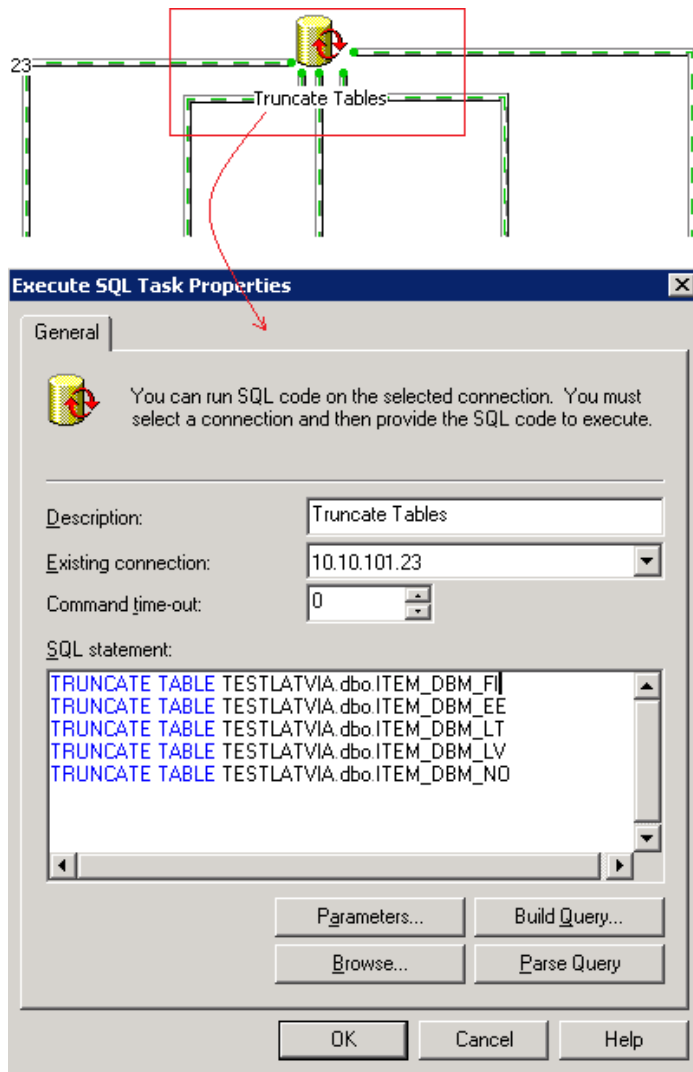
5.4 Realizuoti duomenų integravimo modeliai

5.4.1 Plokščias modelis MS SQL Server 2000 aplinkoje

MS SQL Server 2000 aplinkoje sukurtas DTS procesas, kurio pagrindinis tikslas sujungti ir palyginti duomenis gautus iš visų GNT įmonės padalinių. Duomenys saugomi centrinėje duomenų bazėje iš kurios kuriamos ataskaitos.

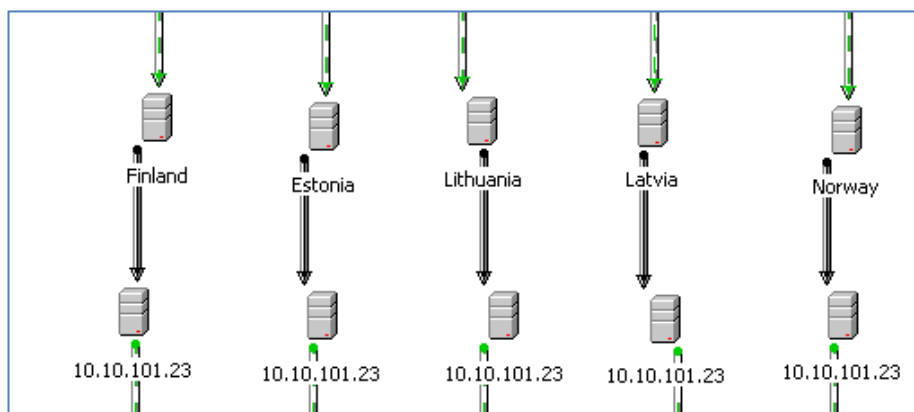
Kiekvienas GNT padalinys yra atskira įmonė turinti savo infrastruktūrą, duomenų bazę, sandėlius, todėl padaliniuose informacija kaupiama skirtingai. Norint palyginti kiekviename padalinyje esančių aktyvių ir neaktyvių produktų skaičių, asortimentą, būtina turėti bendrą ataskaitą, kuri rodytų esamą situaciją GNT šalyse.

Procesas vyksta kiekvieną dieną. Užkraunami duomenys į duomenų bazės lenteles iš lokalių duomenų bazių. Pirmiausia, vyksta DB lentelių, į kurias kraunami duomenys, ištrynimasis. 59 paveiksle pateikiamas DB lentelių išvalymo elementas ir jo kodas.



60 pav. DB lentelių trynimo elementas ir jo kodas

Kai DB lentelės išvalytos, tuomet vyksta duomenų išgavimo – transformavimo - užkrovimo procesai. DTS pakete yra 5 ETL elementai, po vieną kiekvienam GNT padaliniui. 60 paveiksle pateiktas grafinis šių elementų vaizdas DTS pakete.



61 pav. Grafinis duomenų perkėlimo vaizdas

Kadangi visuose padaliniuose duomenų išgavimo logika vienoda, todėl čia pateikiama tik viena užklausa iš penkių.

Užklausa sudaryta iš 5 lokalsios duomenų bazės lentelių. Faktinė lentelė laikoma „T_NIMIKE“ lentelė (produkto lentelė), kitos lentelės jungiamos su LEFT jungimu, kaip papildomos. Traukiami tik aktyvūs ir baigiantys galioti produktai (TN.AKTIIVI IN (1,2)), replikuojami į visas šalis (TN.STATUS=1). Papildomos sąlygos (TN.T_NIMIKE IS NOT NULL AND TN.O_KOODI IS NOT NULL) užtikrina, kad nebūti paimti neteisingi duomenys, kurie atsirado dėl darbuotojų klaidų. Į duomenų sąrašą neįeina duomenų eilutės, kurios neturi produkto kodo arba yra ne GNT prekės (tai prekės, kurios tik sandėliuojamos GNT).

Įvedamas grupavimas pagal unikalų produkto požymį – produkto kodą. Iš vienos šalies turi būti gauti visi unikalūs produktai. Idealiu atveju, produkto kodas yra unikalus, tačiau dėl darbuotojų klaidų atsiranda dubliuotų.

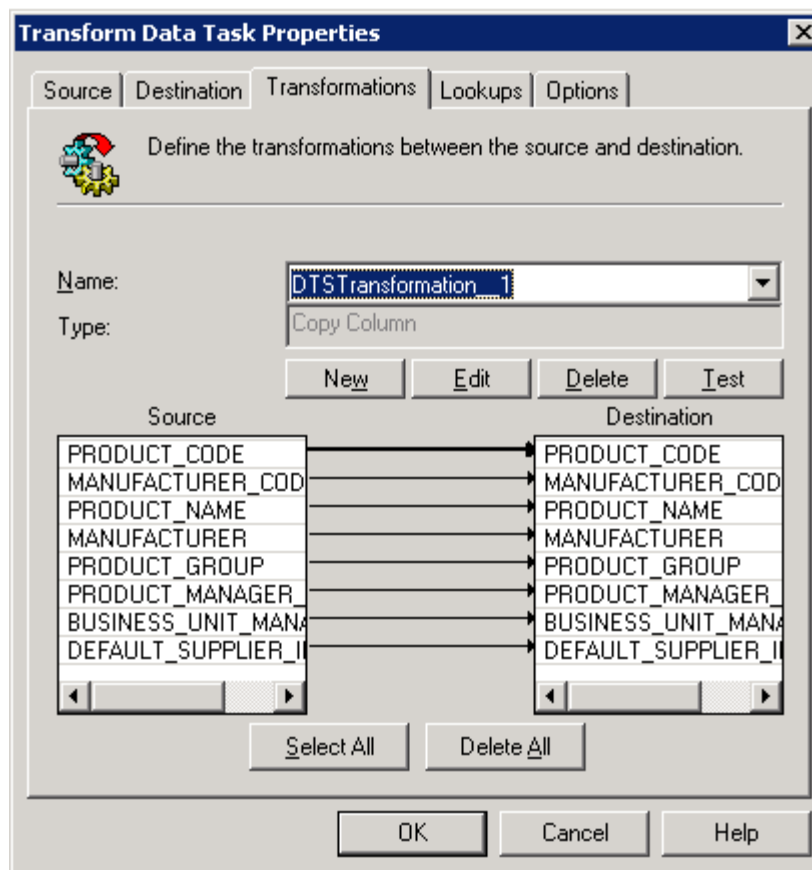
Produkto grupės lentelė (T_RYHMA) turi įrašus skirtingomis kalbomis – vietine, anglų bei standartine. Šiame procese traukiamos tik tos produkto grupės, kurių kalba standartinė arba anglų (KIELI_ID IN (0,2)).

Kadangi reikalingas produkto vadybininko ID reikia prijungti darbuotojų lentelę (KAYTTAJA). Šios lentelės jungimas prie produkto DB lentelės vyksta ne per raktinius laukus, o per vardą ir pavardę. Darbuotojų lentelėje vardas ir pavardė yra skirtinguose laukeliuose, o produkto lentelėje vadybininko vardas ir pavardė yra viename lauke, todėl reikia sujungti vardą ir pavardę, kad gauti PM ID (KA.ETUNIMI + '' +KA.SUKUNIMI).

```
SELECT
    TN.T_KOODI AS PRODUCT_CODE,
    MAX(CASE WHEN ISNULL(TN.TYPE,0)=2 THEN TN.T_KOODI ELSE TN.O_KOODI END) AS
MANUFACTURER_CODE,
    MAX(TN.O_NIMIKE) AS PRODUCT_NAME,
    MAX(VA.LYHENNE) AS MANUFACTURER,
    MAX(TR.LYHENNE) AS PRODUCT_GROUP,
    MAX(KA.ID) AS PRODUCT_MANAGER_ID,
    MAX(ISNULL(KA.CHIEF_ID,0)) AS BUSINESS_UNIT_MANAGER_ID,
    MAX(ISNULL(TM.IP_SUPPLIER_ID,0)) AS DEFAULT_SUPPLIER_ID
FROM DBM.MUSTA_NAAMIO.T_NIMIKE TN(NOLOCK)
LEFT JOIN DBM.MUSTA_NAAMIO.VALM VA (NOLOCK) ON VA.ID=TN.VALM_ID
LEFT JOIN DBM.MUSTA_NAAMIO.T_RYHMA TR (NOLOCK) ON TR.ID=TN.T_RYH_ID AND KIELI_ID IN (0,2)
LEFT JOIN DBM.MUSTA_NAAMIO.KAYTTAJA KA (NOLOCK) ON KA.ETUNIMI + '' +KA.SUKUNIMI=TN.T_PAALL
LEFT JOIN DBM.MUSTA_NAAMIO.TOIM TM (NOLOCK) ON TM.ID=TN.VENDOR_ID
WHERE TN.AKTIIVI IN (1,2) AND TN.STATUS=1 AND TN.T_NIMIKE IS NOT NULL AND TN.O_KOODI IS NOT NULL AND
TN.T_KOODI IS NOT NULL AND TN.OMISTAJA_ID IS NULL
GROUP BY TN.T_KOODI
```

62 pav. Duomenų išgavimo užklausa

61 paveiksle pateikta užklausa gražina duomenis, kurie perkeliami į centrinę duomenų bazės lentelę - TESTLATVIA.dbo.ITEM_DBM_FI (atitinkamai pagal šalį). MS SQL Server 2000 aplinkoje transformacija nėra sudėtinga. Tai atitikčių radimas galutinėje paskirties vietoje (62 pav.).



63 pav. Duomenų sujungimo (Mapping) langas

3 Priede pateikta užklausa, kuri sujungia duomenis, įkeltus iš visų padalinių. Pirmiausia, išvaloma DB lentelė ITEM_MASTER_DBM, po to vykdoma INSERT užklausa. Duomenų sujungimui naudojama MS SQL komanda UNION ALL.

Galutinis rezultatas – suformuojama kiekvienam produktui eilutė, duomenų bazės lentelėje, kurioje yra visų GNT šalių informacija. Tai patogu, kai reikalinga greitai sužinoti esamą situaciją apie produktą visuose padaliniuose. Tačiau, naujo GNT padalinio atsiradimas sukeltų daug nepatogumų, nes galutinė lentelė nėra dinaminė, todėl ją reikėtų plėsti.

Sujungti ir apdoroti duomenys naudojami XML dokumento generavimui. Jis reikalingas asmenims neturintiems tiesioginio priėjimo prie GNT ataskaitų. 63 paveiksle pateiktas sugeneruoto XML

dokumento fragmentas. XML turi bendrąją informaciją, kurioje yra sugeneruoto dokumento laikas, data, versija serverio pavadinimas. Kita dalis yra skirta produktų informacijai. Produkto informacija suskirstyta į „Master“ ir „Branch“ dalis.

„Master“ dalis yra produkto antraštė, kurioje saugoma bendrinė informacija apie produktą: pavadinimas, kodas, produkto grupė, gamintojas. „Branch“ dalyje laikoma detali informacija. Vienas produktas turi tiek „Branch“ lygio dalių kiek jų rasta padaliniuose. Šioje dalyje generuojama informacija yra: šalies kodas (FI,NO,LT ...), Produkto vadybininko ID atitinkamoje šalyje, pardavimo vadybininko ID taip pat atitinkamoje šalyje.

```

<?xml version="1.0" ?>
- <Document>
- <DocumentInfo>
  <Definition>DBM to E1 Item Master</Definition>
  <Date>19.1.2010</Date>
  <Time>6:51:54 PM</Time>
  <SourceSystem>DBM</SourceSystem>
  <SourceSystemEnvironment>Development</SourceSystemEnvironment>
  <TargetSystem>E1</TargetSystem>
  <SourceSystemUser>dwmaster</SourceSystemUser>
  <SourceSystemMachineKey>DW_SQL</SourceSystemMachineKey>
  <Version>1.0</Version>
</DocumentInfo>
- <Item>
+ <ItemMaster>
  <ItemBranchRecords />
+ <ItemBranch>
</Item>
- <Item>
- <ItemMaster>
  <Action>Add</Action>
  <Identifier2ndItem>#300-1441</Identifier2ndItem>
  <DescriptionLine1>PWR SPLY,A145,AC-48VDC CONV</DescriptionLine1>
  <DescriptionLine2 />
  <SalesReportingCode2>SPA</SalesReportingCode2>
  <SalesReportingCode4>YYY</SalesReportingCode4>
</ItemMaster>
  <ItemBranchRecords />
  <ItemBranch>
    <Action>Add</Action>
    <Branchplant>FI</Branchplant>
    <SalesReportingCode1>3688</SalesReportingCode1>
    <SalesReportingCode5 />
    <AddressNumberPlanner>8558</AddressNumberPlanner>
  </ItemBranch>
</Item>
+ <Item>
+ <Item>
+ <Item>
</Document>

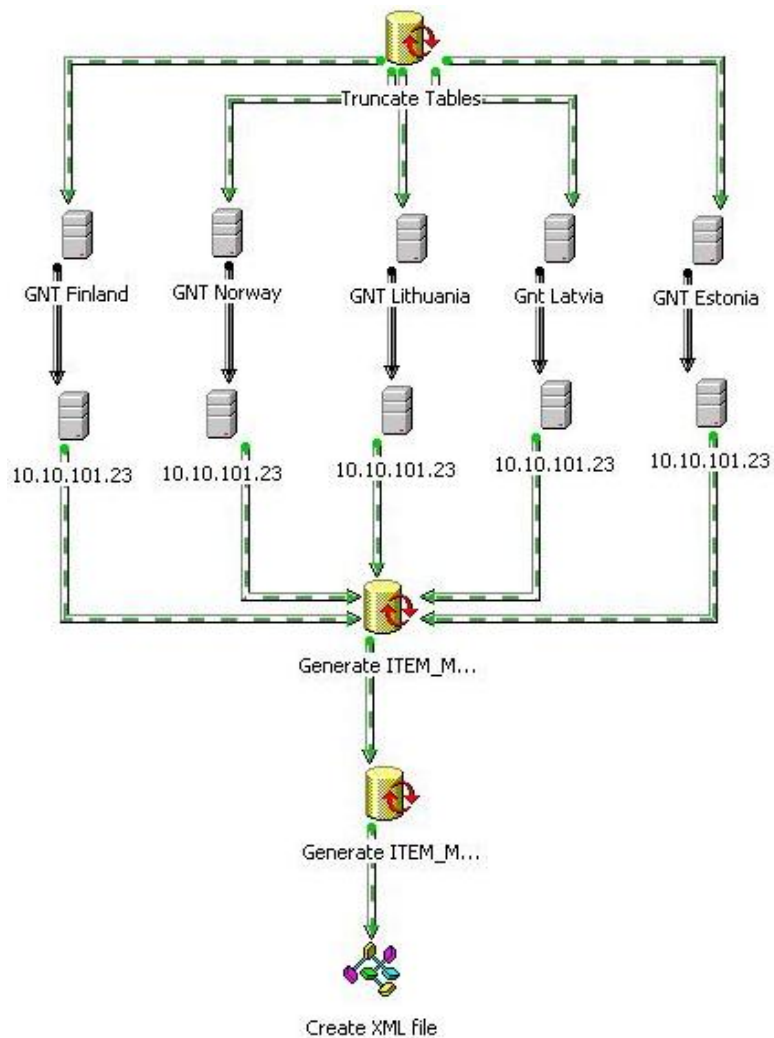
```


DTS proceso vykdymo trukmė yra ilga, dėl to siekiama jį padaryti greitesnį. Komplikuotas naujo padalinio integravimas į procesą reikalauja struktūros patobulino, kuris leistų lanksčiai pašalinti arba pridėti duomenis iš naujo duomenų šaltinio.

5.4.2 Struktūrizuotas modelis MS SQL Server 2000 aplinkoje

DTS proceso, kurio aprašymas pateiktas „Plokščias modelis MS SQL Server 2000 aplinkoje“ skyriuje, vykdymo trukmė ilga. Jo struktūra paprasta, tačiau nelanksti naujo padalinio atsiradimo ir kitų numatomų padaryti pakeitimų atžvilgiu. Atliktos DI proceso analizės metu, pastebėta, kad ilgiausiai trunkantis proceso etapas yra duomenų sujungimas ir įrašymas į lentelę.

64 paveiksle matomas DI procesas *MS SQL Server 2000* aplinkoje. Panaudoti trijų tipų elementai: užklauskos rašymo, transformacijos, *ActiveX* elementai.



65 pav. Detalus projektas *MS SQL Server 2000 DTS Design* aplinkoje

Duomenų gavyba išlieka tokia pati, kaip pradiniame DI procese. Galutinės paskirties vietos restruktūrizavimas pradedamas informacijos suskirstymu. Duomenys padalinami į dvi dalis: pagrindinę ir detalią.

Duomenų sujungimo ir įrašymo metu laikomasi taisyklės, pirmiausia, perkelti duomenis, esančius Suomijos DB lentelėse. GNT Suomijos padalinys yra pagrindinis, nes visi produktai užsakomi per šią šalį. Padalinys tiesiogiai bendrauja su gamintojais. 65 paveiksle pateiktas MS SQL kodo fragmentas, kuriame matyti kad naudojama funkcija „ISNULL“, kuri tuo atveju kai nėra produkto informacijos Suomijos DB, paima duomenis iš kitos šalies DB.

```
UPPER(XX.PRODUCT_CODE) AS PRODUCT_CODE,
UPPER(MAX(isnull(MANUFACTURER_CODE_FI,MANUFACTURER_CODE))) AS MANUFACTURER_CODE,
UPPER(ISNULL(MAX(ISNULL(PRODUCT_NAME_FI,PRODUCT_NAME)),)) AS PRODUCT_NAME,
UPPER(ISNULL(MAX(isnull(MANUFACTURER_FI,MANUFACTURER)),)) AS MANUFACTURER,
UPPER(ISNULL(MAX(isnull(PRODUCT_GROUP_FI,PRODUCT_GROUP)),)) AS PRODUCT_GROUP
```

66 pav. Duomenų iš GNT Suomijos pirmumo taisyklės MS SQL kodo fragmentas.

Antroje lentelėje talpinama detali informacija apie produktą, kuri unikali kiekvienoje šalyje. 66 - 67 paveiksluose pateikta grafinė vieno produkto interpretacija duomenų bazėje plokščioje struktūrizuotoje lentelėje.

PRODUCT_CODE	MANUFACTURER_CODE_FI	MANUFACTURER_CODE_EE	MANUFACTURER_CODE_LT	MANUFACTURER_CODE_LV	MANUFACTURER_CODE_NO	PRODUCT_NAME
1 66A-00137	66A-00137	66A-00137	66A-00137	66A-00137	66A-00137	CCF EXTRNL CONN L/SA OLV I

MANUFACTURER	PRODUCT_GROUP	PRODUCT_MANAGER_ID_FI	PRODUCT_MANAGER_ID_EE	PRODUCT_MANAGER_ID_LT	PRODUCT_MANAGER_ID_LV	PRODUCT_MANAGER_ID_NO
1 MST	OL	8873	104	282	244	9358

BUSINESS_UNIT_MANAGER_ID_FI	BUSINESS_UNIT_MANAGER_ID_EE	BUSINESS_UNIT_MANAGER_ID_LT	BUSINESS_UNIT_MANAGER_ID_LV	BUSINESS_UNIT_MANAGER_ID_NO
1 5689	15	5	205	9348

DEFAULT_SUPPLIER_ID_FI	DEFAULT_SUPPLIER_ID_EE	DEFAULT_SUPPLIER_ID_LT	DEFAULT_SUPPLIER_ID_LV	DEFAULT_SUPPLIER_ID_NO	ACTIVITY_FI	ACTIVITY_EE
1 5121	0	0	0	0	1	1

67 pav. Produkto informacija pateikta „plokščioje“ DB lentelėje

67 paveiksle pateikta produkto informacija struktūrizuotoje DB lentelėje. Pirmoji eilutė yra pagrindinė produkto informacija: Produkto kodas, gamintojo kodas, produkto pavadinimas, produkto grupė, gamintojo pavadinimo trumpinys. Kitoje lentelėje - detali produkto informacija, kurioje saugomas šalies kodas, atsakingo už produktą vadybininko ID ir t.t.

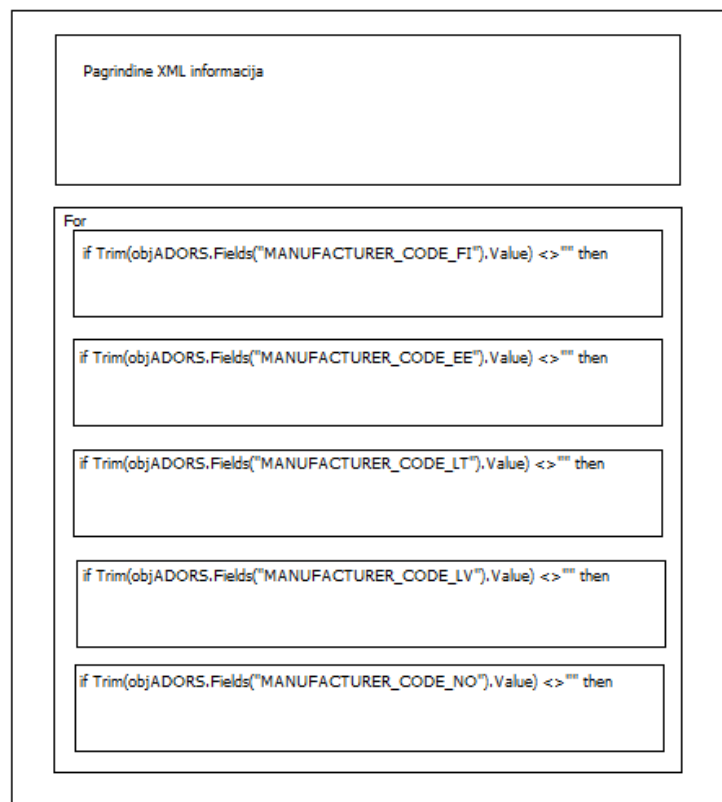
	PRODUCT_CODE	MANUFACTURER_CODE	PRODUCT_NAME	MANUFACTURER	PRODUCT_GROUP	ACTION_MASTER
1	66A-00137	66A-00137	CCF EXTRNL CONN L/SA OLV D 3YRACQY1	MST	OL	1

	BRANCH_PLANT	PRODUCT_CODE	PRODUCT_MANAGER_ID	DEFAULT_SUPPLIER_ID	BUSINESS_UNIT	ACTION_BRANCH	PROFIT_CENTER
1	200	66A-00137	200-9358	0		1	
2	400	66A-00137	400-244	0		1	
3	500	66A-00137	500-282	0		1	
4	100	66A-00137	100-8873	0		1	
5	300	66A-00137	300-104	0		1	

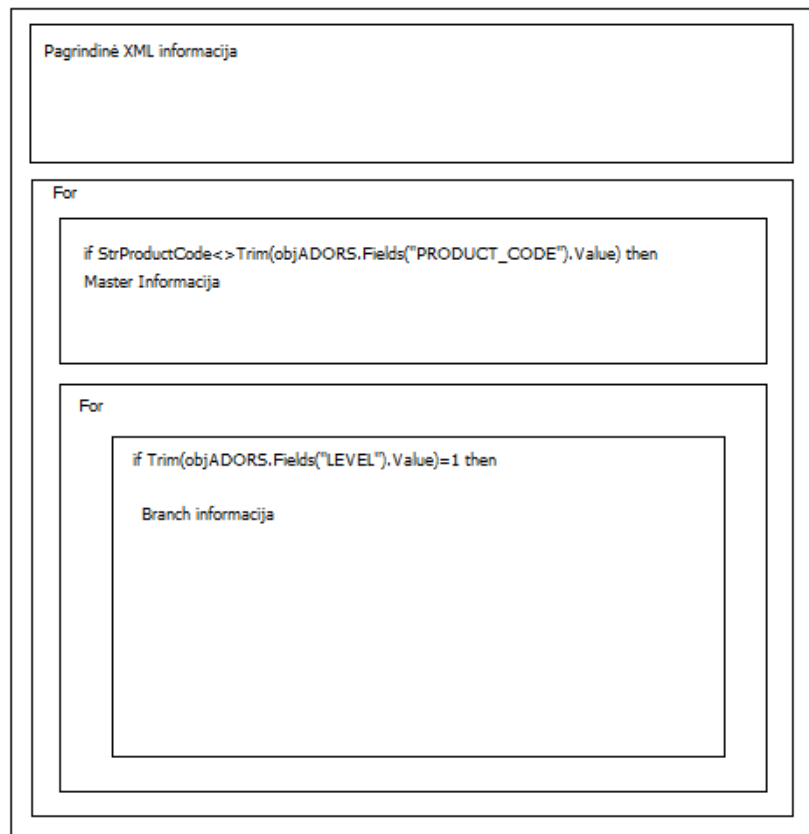
68 pav. Produkto informacija pateikta „struktūrizuotoje“ DB lentelėse

Dėl struktūrizuotų lentelių duomenys saugomi skirtingose vietose, tačiau XML dokumentui patogiu paduoti vieną duomenų sąrašą. Todėl sukuriama poschemė (angl. *View*).

Procese patobulintas XML dokumento generavimas. Pradiniame DI procese pagrindinė informacija apie produktą buvo kartojama pagal duomenų šaltinių skaičių. Patobulintame XML generavime, suprogramuotas ciklas, kuris ima pagrindinę produkto informaciją, vykdo ciklą tol, kol randa duomenų bazėje eilučių su atitinkamu produkto kodu. 68 – 69 paveiksluose pateiktas pradinis ir patobulintas XML generavimo metodas.



69 pav. XML generavimo metodas pradinio projekto variante



70 pav. XML generavimo metodas patobulinto projekto variante

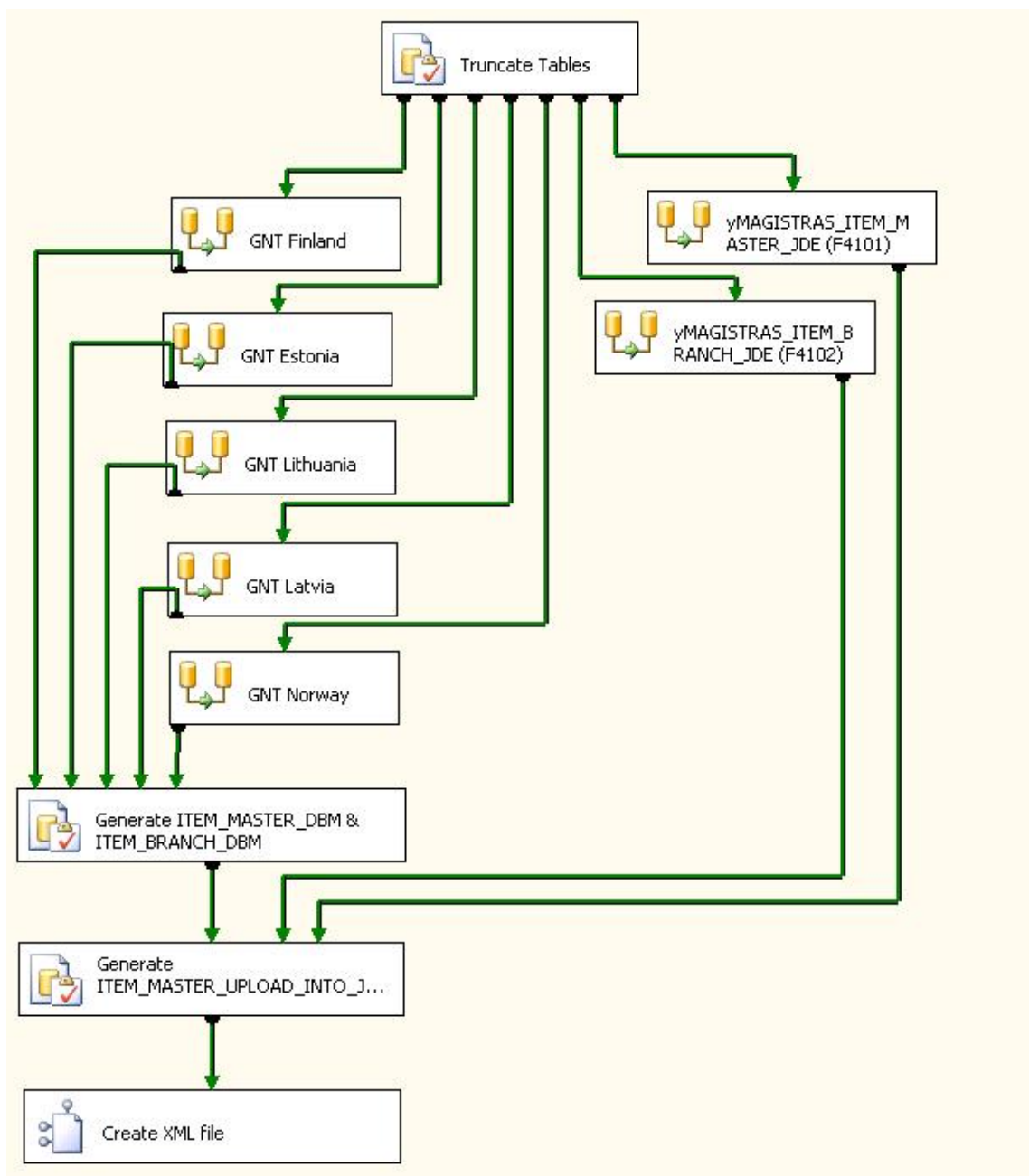
Struktūrizuotame DTS procese yra išpildytas naujo padalinio integravimas, lankstus seno duomenų šaltinio pašalinimas.

5.4.3 Struktūrizuotas modelis MS SQL Server 2005 aplinkoje

Analizės metu nustatyta, kad lengviausias, greičiausias ir efektyviausias perėjimas iš vienos procesų vykdymo terpės į kitą, yra procesų perkėlimas į *MS SQL Server 2005*. Todėl sukurtas ir patobulintas duomenų perkėlimo procesas kuriamas naujoje terpėje. Siekiama išlaikyti tą pačią duomenų struktūrą ir gaunamą proceso rezultatą.

Konsoliduotam produktų informacijos procesui yra įvedama papildoma sąlyga – duomenų palyginimas. Palyginimas vyksta tarp centrinėje DB esančių produktų informacijos ir naujo padalinio, kuris ateityje turi atlikti centrinės DB vaidmenį. Naujo padalinio aplinka yra *Oracle*, turinti savą įmonės infrastruktūrą, todėl tiesiogiai duomenų įkelti į *Oracle* DB lenteles negalima. Priimtas sprendimas XML dokumento pagalba perkelti reikiamą informaciją. XML pasirinktas, nes duomenis galima transformuoti į įvairius formatus.

70 paveiksle pateiktas produktų informacijos sujungimo ir palyginimo procesas *MS SQL Server 2005* aplinkoje.



71 pav. Produktų informacijos sujungimo ir palyginimo procesas *MS SQL Server 2005* aplinkoje

Duomenų perkėlimas ir sujungimas vyksta analogiškai kaip ir patobulintame procese *MS SQL Server 2000* aplinkoje, todėl šios proceso dalys skyrelyje nebeaprašomos.

Duomenų palyginimas reikalauja sukurti naujus žemesnio lygio procesus pakete, kurie trauktų duomenis iš *Oracle* duomenų bazės. Duomenų paėmimui naudojami globalūs kintamieji. Juos patogiau naudoti, kai duomenų traukimo užklausa keičiasi. Pavyzdžiui, duomenų bazės serveryje yra keletas

duomenų bazių, viduje jos talpina identišką DB lenteles, poschemes, saugomas procedūras, skiriasi tik savininko vardas (angl. *owner*). Kuomet pasikeičia savininko vardas užtenka pakeisti kintamojo reikšmę ir užklausa traukia duomenis iš kitos DB. 71 paveiksle pateikti visi globalūs kintamieji, kurie panaudoti SSIS pakete.

Name	Scope	Data Type	Value
DBVardas	Magistrinis_Darbas-IM	String	CRPDTA
GavejuSarasas	Magistrinis_Darbas-IM	String	indre.kungyte@gmail.com
ORACLE_que...	Magistrinis_Darbas-IM	String	SELECT IMITM as ITEM_NUMBER,IMLITM as P...
ORACLE_que...	Magistrinis_Darbas-IM	String	SELECT T1.IBITM as ITEM_NUMBER, T1.IBMCU...
SQL1_templatas	Magistrinis_Darbas-IM	String	SELECT IMITM as ITEM_NUMBER,IMLITM as P...
SQL2_templatas	Magistrinis_Darbas-IM	String	SELECT T1.IBITM as ITEM_NUMBER, T1.IBMCU...

72 pav. Globalūs kintamieji SSIS pakete

Globalių kintamųjų reikšmes galima išsaugoti XML dokumente, kuriame lengva administruoti SSIS paketą, neturint visos informacijos apie jį. 72 paveiksle pateiktas konfigūracinio failo turinys.

```

- <Configuration ConfiguredType="Property" Path="\Package.Connections[ORACLE].Properties[UserName]" ValueType="String">
  <ConfiguredValue>DEV_GARCIAAN</ConfiguredValue>
</Configuration>
- <Configuration ConfiguredType="Property" Path="\Package.Variables[User::_InputDatabase].Properties[Value]" ValueType="String">
  <ConfiguredValue>TESTDTA</ConfiguredValue>
</Configuration>
- <Configuration ConfiguredType="Property" Path="\Package.Variables[User::SendEmailTo].Properties[Value]" ValueType="String">
  <ConfiguredValue>indre.kungyte@gnt.lt</ConfiguredValue>
</Configuration>
</DTSConfiguration>

```

73 pav. Konfigūracinio dokumento turinys

Konfigūracinis failas išsaugomas .dtsConfig formatu.



74 pav. Globalių kintamųjų dokumentas išsaugotas XML struktūra

Vykdomos užklauskos rezultatas – produktų skirtumai. Žemiau pateikta skirtumų radimo užklauskos analizė. Užklauską suskirstyta į 5 dalis.

1-oji užklauskos dalis: Duomenys, esantys centrinėje DB jungiami „LEFT“ jungimu su lentele, kurioje yra duomenys gauti iš *Oracle DB*. Tos eilutės, kurios nesusijungia su „yMAGISTRAS_ITEM_MASTER_DBM“ lentele yra įdedamos į

„yMAGISTRAS_ITEM_MASTER_UPLOAD_INTO_JDE“ lentelę. Visos eilutės įgyja požymį „add“, kuris reiškia, jog šie produktai yra naujai sukurti ir dar neįkelti į *Oracle DB* lentelę. 74 paveiksle pateiktas skirtumo generavimo užklauso dalis, kuri grąžina naujų produktų pagrindinę informaciją.

```
INSERT INTO ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_MASTER_UPLOAD_INTO_JDE
SELECT      IMD.PRODUCT_CODE AS PRODUCT_CODE,
            IMD.MANUFACTURER_CODE AS MANUFACTURER_CODE,
            IMD.PRODUCT_NAME AS PRODUCT_NAME,
            IMD.MANUFACTURER AS MANUFACTURER,
            IMD.PRODUCT_GROUP AS PRODUCT_GROUP,
            1 AS ACTION_MASTER

FROM ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_MASTER_DBM IMD (NOLOCK)
LEFT JOIN ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_MASTER_JDE IMJ (NOLOCK) ON
IMD.PRODUCT_CODE=IMJ.PRODUCT_CODE
WHERE IMJ.PRODUCT_CODE IS NULL
```

75 pav. Skirtumo generavimo užklauso dalis, kuri grąžina naujų produktų pagrindinę informaciją

2-oji užklauso dalis: Duomenys, esantys centrinėje DB jungiami „INNER“ jungimu su lentele, kurioje yra duomenys gauti iš *Oracle DB*. Susijungusių eilučių per produkto kodą informacija yra lyginama tarpusavyje. Jei laukeliuose esančios reikšmės nesutampa, tuomet ši produkto informacija yra įdedama į „yMAGISTRAS_ITEM_MASTER_UPLOAD_INTO_JDE“ lentelę su požymiu „update“. 75 paveiksle pateiktas skirtumo generavimo užklauso dalis, kuri grąžina atnaujinamų produktų pagrindinę informaciją.

```
INSERT INTO ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_MASTER_UPLOAD_INTO_JDE
SELECT      IMD.PRODUCT_CODE AS PRODUCT_CODE,
            IMD.MANUFACTURER_CODE AS MANUFACTURER_CODE,
            IMD.PRODUCT_NAME AS PRODUCT_NAME,
            IMD.MANUFACTURER AS MANUFACTURER,
            IMD.PRODUCT_GROUP AS PRODUCT_GROUP,
            2 AS ACTION_MASTER

FROM ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_MASTER_DBM IMD (NOLOCK)
INNER JOIN ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_MASTER_JDE IMJ (NOLOCK) ON
IMD.PRODUCT_CODE=IMJ.PRODUCT_CODE
WHERE IMD.MANUFACTURER_CODE<>IMJ.MANUFACTURER_CODE
      OR IMD.PRODUCT_NAME<>(IMJ.PRODUCT_NAME+IMJ.PRODUCT_NAME_2)
      OR IMD.MANUFACTURER<>IMJ.MANUFACTURER
      OR IMD.PRODUCT_GROUP<>IMJ.PRODUCT_GROUP
```

76 pav. Skirtumo generavimo užklauso dalis, kuri grąžina atnaujinamų produktų pagrindinę informaciją

3-oji užklauso dalis: Gaunama detali produkto informacija, kurią reikia perkelti į *Oracle DB*. Visuose įmonės padaliniuose yra panašus produktų asortimentas, tačiau dėl rinkos, produkto paklauso, ne visuose padaliniuose produktu prekiaujama. Pavyzdžiui, produktu „AAA“ prekiaujama Suomijoje ir Latvijoje. Proceso vykdymo metu šis produktas patenka į *Oracle DB*, tačiau su požymiu, kad jis yra tik

Suomijoje ir Latvijoje. Kuomet pradedama prekiauti Lietuvoje, duomenų bazėje atsiranda įrašai su šio produkto informacija.

Ši užklausos dalis suranda pagrindinę produkto informaciją, kuri jau yra įkelta į *Oracle DB*, ir ją papildo atitinkamo padalinio detalio produkto informacija (76 pav.).

```
INSERT INTO ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_BRANCH_UPLOAD_INTO_JDE
SELECT
        IBD.BRANCH_PLANT AS BRANCH_PLANT,
        IBD.PRODUCT_CODE AS PRODUCT_CODE,
        IBD.PRODUCT_MANAGER_ID AS PRODUCT_MANAGER_ID,
        IBD.DEFAULT_SUPPLIER_ID AS DEFAULT_SUPPLIER_ID,
        IBD.BUSINESS_UNIT AS BUSINESS_UNIT,
        1 AS ACTION_BRANCH,
        IBD.PROFIT_CENTER AS PROFIT_CENTER

FROM ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_BRANCH_DBM IBD (NOLOCK)
LEFT JOIN ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_MASTER_JDE IMJ (NOLOCK) ON
IBD.PRODUCT_CODE=IMJ.PRODUCT_CODE
LEFT JOIN ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_BRANCH_JDE IBJ (NOLOCK) ON
IMJ.ITEM_NUMBER=IBJ.ITEM_NUMBER
AND CAST(IBD.BRANCH_PLANT AS VARCHAR (12))=LTRIM(IBJ.BRANCH_PLANT)
WHERE IBJ.BRANCH_PLANT IS NULL
```

77 pav. Skirtumo generavimo užklausos dalis, kuri grąžina esamų produktų naują produkto informaciją

4-oji užklausos dalis: Ši užklausos dalis gauna visą detalią informaciją apie produktą ir ją palygina tarpusavyje. Jei yra randami pasikeitimai atitinkamuose stulpeliuose, įrašas įtraukiamas į „yMAGISTRAS_ITEM_BRANCH_UPLOAD_INTO_JDE“ lentelę (77 pav).

```
INSERT INTO ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_BRANCH_UPLOAD_INTO_JDE
SELECT
        IBD.BRANCH_PLANT AS BRANCH_PLANT,
        IBD.PRODUCT_CODE AS PRODUCT_CODE,
        IBD.PRODUCT_MANAGER_ID AS PRODUCT_MANAGER_ID,
        IBD.DEFAULT_SUPPLIER_ID AS DEFAULT_SUPPLIER_ID,
        IBD.BUSINESS_UNIT AS BUSINESS_UNIT,
        2 AS ACTION_BRANCH,
        IBD.PROFIT_CENTER AS PROFIT_CENTER

FROM ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_BRANCH_DBM IBD (NOLOCK)
INNER JOIN ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_MASTER_JDE IMJ (NOLOCK) ON
IBD.PRODUCT_CODE=IMJ.PRODUCT_CODE
INNER JOIN ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_BRANCH_JDE IBJ (NOLOCK) ON
IMJ.ITEM_NUMBER=IBJ.ITEM_NUMBER
AND CAST(IBD.BRANCH_PLANT AS VARCHAR (12))=LTRIM(IBJ.BRANCH_PLANT)

WHERE
        IBD.PRODUCT_MANAGER_ID <>LTRIM(IBJ.PRODUCT_MANAGER_ID)
        OR IBD.DEFAULT_SUPPLIER_ID <>IBJ.DEFAULT_SUPPLIER_ID
        OR IBD.BUSINESS_UNIT<>IBJ.BUSINESS_UNIT
        OR IBD.PROFIT_CENTER<>LTRIM(IBJ.PROFIT_CENTER)
```

78 pav. Skirtumo generavimo užklausos dalis, kuri grąžina atnaujinamų produktų detalią informaciją

5-oji užklauso dalis: Užklauso dalis vykdoma, kai visos užklauso dalys yra atliktos. Ji skirta patikrinti duomenis, kad visa produktų detali informacija turėtų pagrindinę (78 pav.).

```
INSERT INTO ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_MASTER_UPLOAD_INTO_JDE
SELECT      IMD.PRODUCT_CODE AS PRODUCT_CODE,
            IMD.MANUFACTURER_CODE AS MANUFACTURER_CODE,
            IMD.PRODUCT_NAME AS PRODUCT_NAME,
            IMD.MANUFACTURER AS MANUFACTURER,
            IMD.PRODUCT_GROUP AS PRODUCT_GROUP,
            0 AS ACTION_MASTER

FROM ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_MASTER_DBM IMD (NOLOCK)
WHERE IMD.PRODUCT_CODE IN
      (SELECT DISTINCT PRODUCT_CODE FROM
       ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_BRANCH_UPLOAD_INTO_
       JDE (NOLOCK) )
AND    IMD.PRODUCT_CODE NOT IN
      (SELECT DISTINCT PRODUCT_CODE FROM
       ITEMMASTER.IMASTER.yMAGISTRAS_ITEM_MASTER_UPLOAD_INTO_
       JDE (NOLOCK) )
```

79 pav. Tikrinama, ar visa detali produktų informacija turi pagrindinę informaciją

5.4.4 Duomenų integravimo modelių apibendrinimas

Sistemos realizacijos etape sukurti 3 projektai. Pradinio proceso pagrindinė esmė - gauti sujungtą informaciją iš lokalių duomenų bazės lentelių ir iš jų sugeneruoti XML dokumentą. Proceso vykdymo trukmė buvo ilga, todėl nuspręsta procesą patobulinti, padaryti efektyvesnį, našesnį.

Sukurtas antrasis projektas turi patobulintą DB lentelių struktūrą ir XML dokumento generavimą. Proceso vykdymo laikas sutrumpėjo, tapo lankstesnis įvairiems pasikeitimams.

Kadangi pagrindinė idėja perkelti visus DTS procesus į naują terpę, dėl senosios terpės panaikinimo, buvo reikalinga sukurti analogišką procesą SSIS pagalba. Trečiame projekte įvedama papildoma sąlyga – duomenų traukimas iš *Oracle* duomenų bazės ir trūkstamos informacijos įkėlimas į ją. XML dokumentas generuoja produktų skirtumų informaciją.

5.5 Duomenų integravimo projektų testavimas

Testavimas yra viena iš pagrindinių duomenų įkėlimo proceso dalių. Jis vykdomas nuolat, dėl didelio duomenų srauto ir naujų duomenų atsiradimo lokaliuose duomenų bazėse. Neteisingų duomenų priežastis gali būti vartotojas, kuris neteisingai įveda informaciją į atitinkamas sistemos formas, nestabilus interneto ryšys, dėl kurio duomenys yra neteisingai perkeltami į DB.

5.5.1 Testavimo modelis

Kol veikia senoji DTS procesų sistema lengva patikrinti duomenų teisingumą. Kadangi DTS procesas senojoje sistemoje yra ištestuotas, todėl įkelti duomenys turi visiškai sutapti su duomenimis, perkeltais naujuoju SSIS procesu. Pirminis patikrinimas – bendras perkeltų eilučių skaičius. Jei jis nesutampa tai požymis, kad SSIS procese yra klaida.

Testavimo modelis reikalingas nuolatiniam duomenų testavimui. Jis taps svarbus kuomet senoji sistema bus panaikinta ir visiškai pereita prie *MS SQL 2005 Server 2005*.

Duomenų integravimo procese išskirti trys pagrindiniai etapai:

1. Negali būti kuriama produkto antraštė, jei nė vienoje lokatioje DB detalios informacijos apie produktą nėra.

Loginės sąlygos netenkinimo atveju, SQL užklausa grąžina duomenų sąrašą.

```
SELECT product_code
FROM Itemmaster.Imaster.v_yMAGISTRAS_ITEM_MASTER
  where ACTION_MASTER=1
group by product_code
having sum(ACTION_BRANCH) <>0
```

80 pav. Produkto antraštės sukūrimo testavimas, kai produkto detalios informacijos nėra

2. Negali būti kuriama produkto antraštė, jei bent vienoje šalyje detali produkto informacija buvo atnaujinta.

Loginės sąlygos netenkinimo atveju, SQL užklausa grąžina duomenų sąrašą.

```
SELECT product_code
FROM Itemmaster.Imaster.v_yMAGISTRAS_ITEM_MASTER
  where ACTION_MASTER=1
group by product_code
having sum(ACTION_BRANCH) >0
```

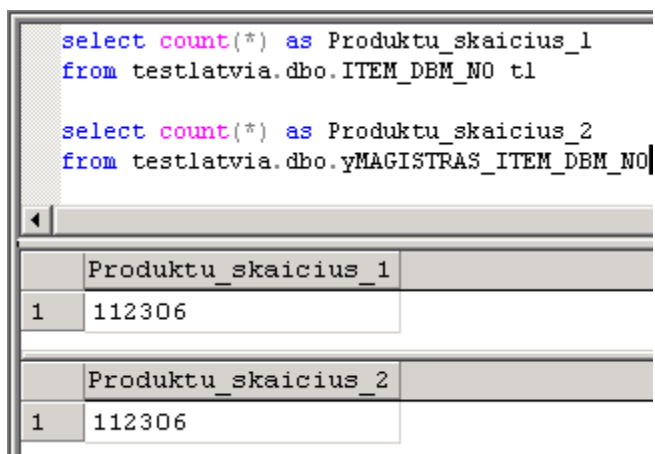
81 pav. Produkto antraštės sukūrimo testavimas, kai produkto detali informacija atnaujinama bent vienoje šalyje

5.5.2 Integravimo proceso duomenys, DI proceso pavyzdys

Produktų informacijos integravimo procesą sudaro 5 GNT šalyse esantys duomenys, todėl duomenų srautas yra labai didelis. Kontroliniame pavyzdyje išnagrinėtas procesas pagal vieną šalį. Pasirinkta šalis yra Norvegija.

Bendras perkeltas produktų skaičius turi sutapti visuose projektuose (3 projektai). Kadangi pirmasis DTS procesas yra ištestuotas ir veikia teisingai, būtina juo remtis. Jei SSIS paketu perkėlus duomenis iš Norvegijos gaunamas skaičius mažesnis arba didesnis, vadinasi, klaida yra užklausoje. 81 –

82 paveiksluose pateiktas bendras eilučių skaičius, gautas perkėlus duomenis iš Norvegijos, SSIS ir DTS paketuose.



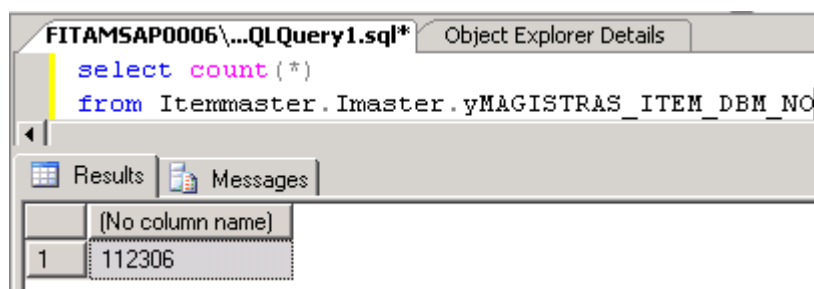
```
select count(*) as Produktu_skaicius_1
from testlatvia.dbo.ITEM_DBM_NO t1

select count(*) as Produktu_skaicius_2
from testlatvia.dbo.yMAGISTRAS_ITEM_DBM_NO
```

Produktu_skaicius_1	
1	112306

Produktu_skaicius_2	
1	112306

82 pav. Bendras produktų skaičius užkrautas pradiniu ir patobulintu DTS procesais



```
select count (*)
from Itemmaster.Imaster.yMAGISTRAS_ITEM_DBM_NO
```

(No column name)
1

83 pav. Bendras produktų skaičius užkrautas SSIS procesu

Tikrinimas remiantis senąja procesu sistema yra baigtas. Toliau būtina patikrinti teisingą skirtumų radimą. Yra dvi pagrindinės loginės taisyklės, kurios aprašytos testavimo modelyje. Surandamos neegzistuojančios duomenų kombinacijos pasinaudojant testavimui sukurtomis užklausomis.

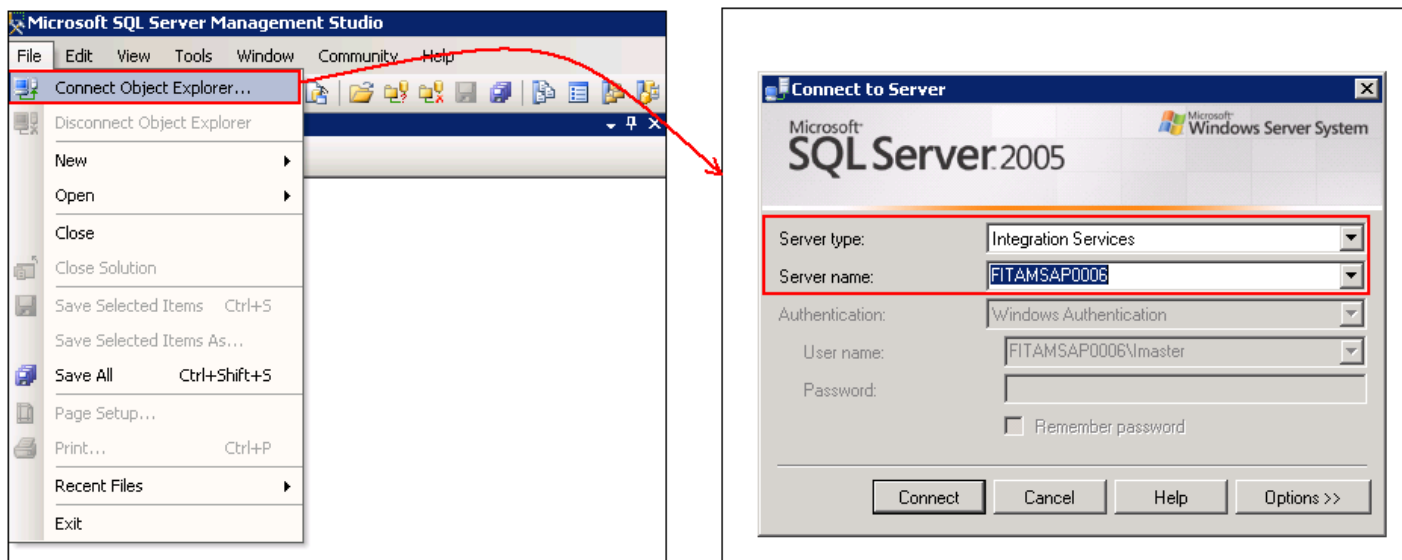
5.5.3 SSIS ir DTS procesų diegimo planas

Naujas DTS ir SSIS projektas turi būti įdiegtas į bendrą procesų grandinę. Įdiegti ir teisingai funkcionuojantys paketai turi nustatyta laiką, kuris kruopščiai parenkamas, įvertinus duomenų svarbą, pirmumo teisę vykdytis procesų grandinėje.

SSIS diegimo planas

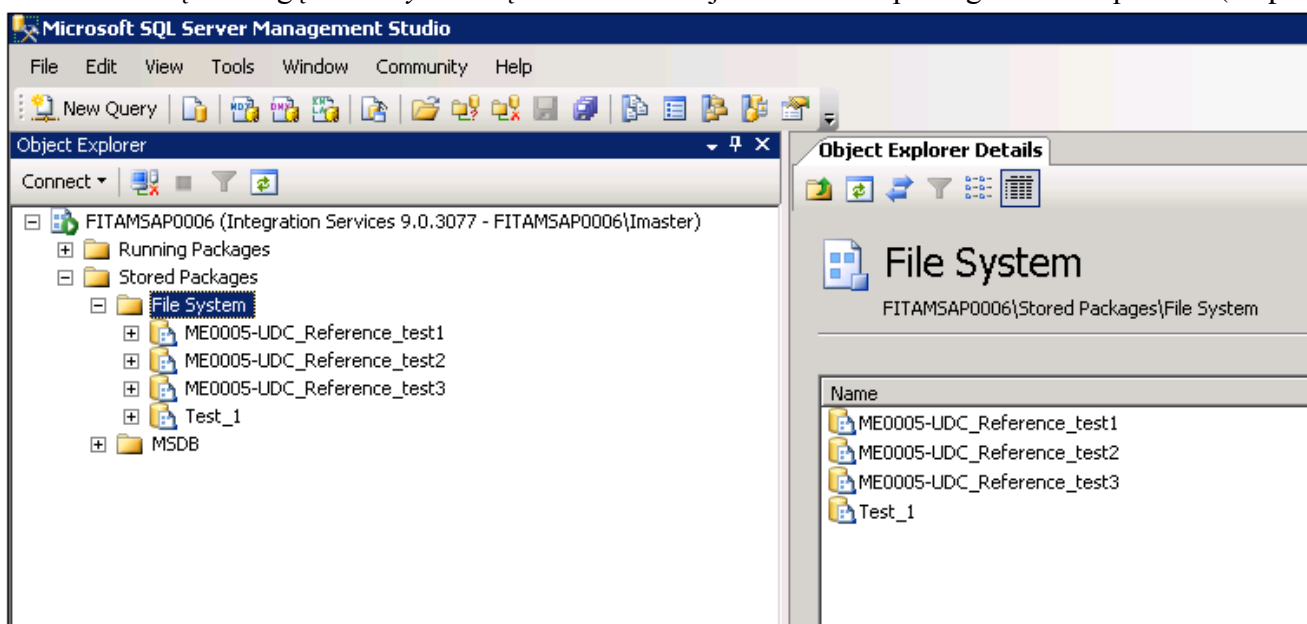
SSIS paketo diegimas susideda iš dviejų dalių: SSIS paketo importavimo ir naujo darbo sukūrimo.

1. Paketo importavimas vyksta *Microsoft SQL Server Managament Studio* aplinkoje, pasirinkus *Integration services* jungtį (83 pav.).



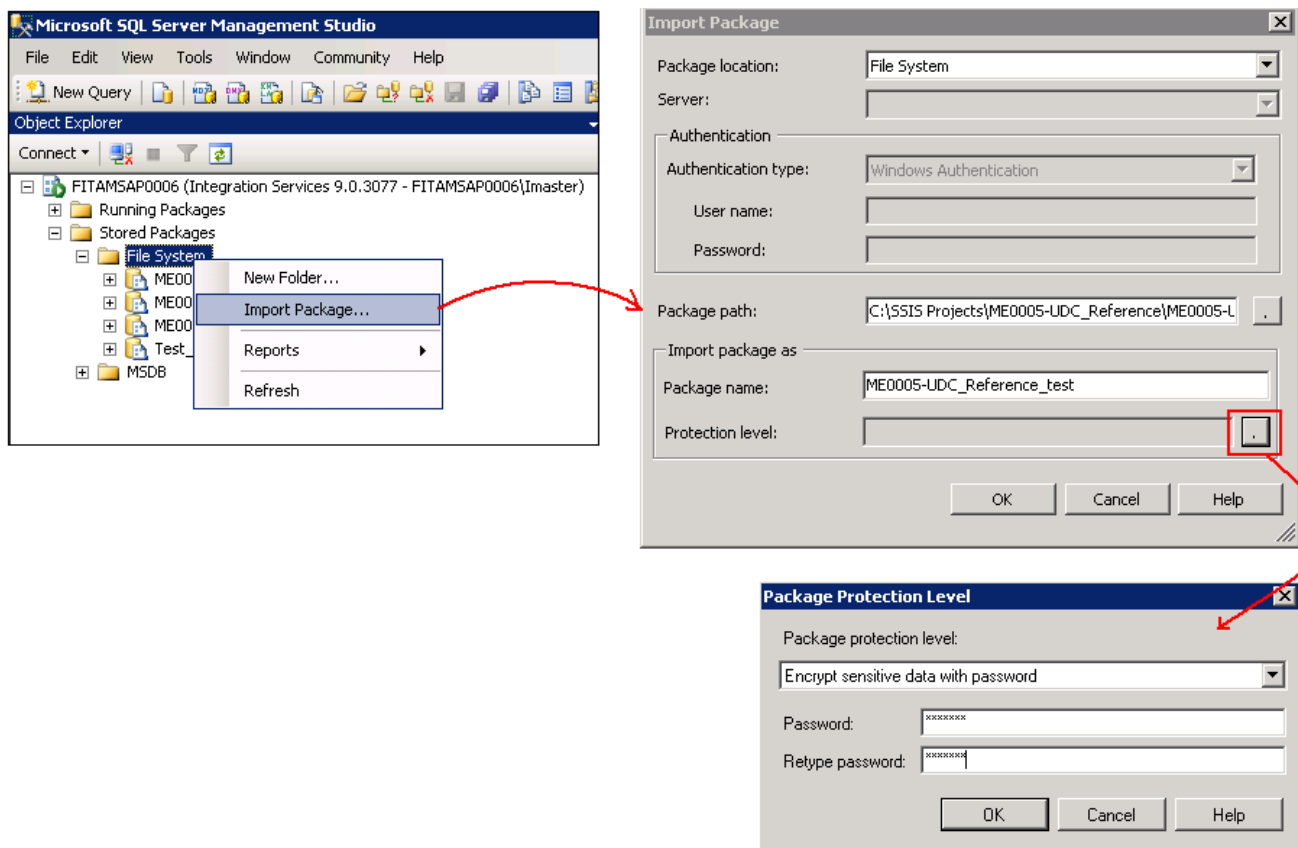
84 pav. "Integration services" jungtis

2. Į katalogą "File System" įtraukiamas naujai sukurtas ir pasaugotas SSIS paketas (84 pav.).



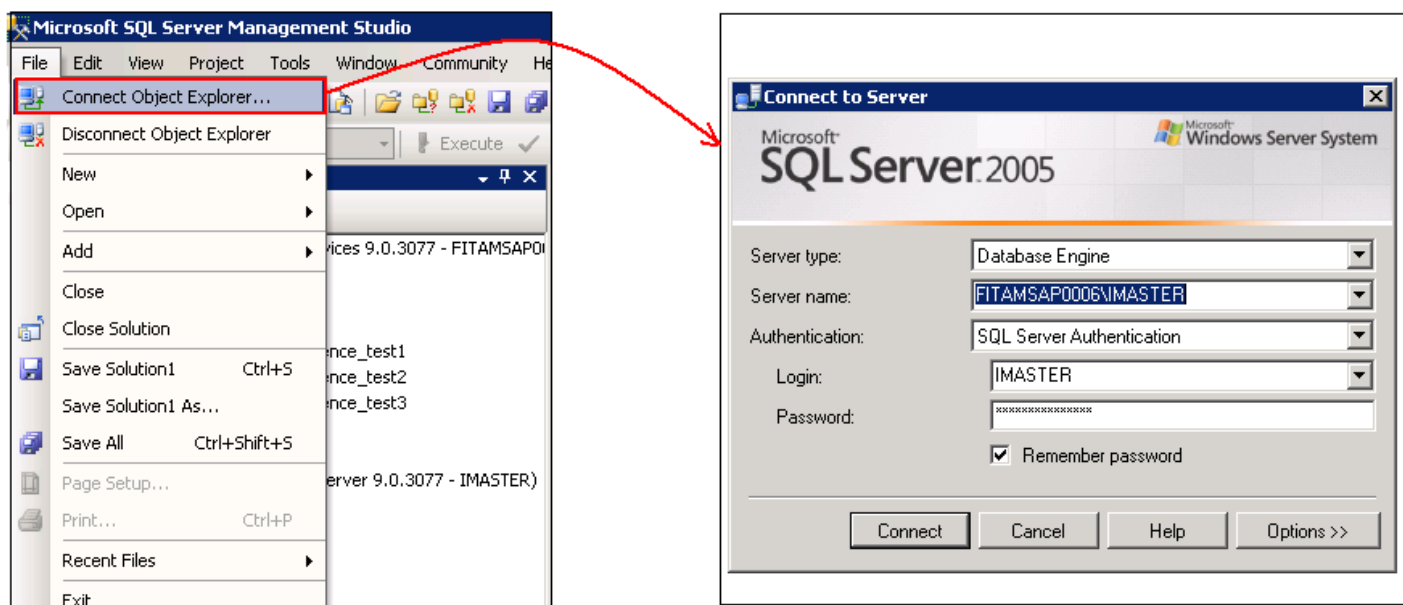
85 pav. "File System" katalogas

3. Importuojamas SSIS paketas. Pasirenkamas atitinkamas projektas, suteikiamas pavadinimas. Būtina nustatyti *Protection Level* metodą – tai apsauga, kuri garantuoja proceso saugumą. Žemiau pateiktame paveiksle yra pasirinktas *Encrypt Sensitive data with password* apsaugos būdas, kuris reikalauja įvesti paketui skirtą prisijungimo vardą ir slaptažodį. ("File System" -> Import Package).



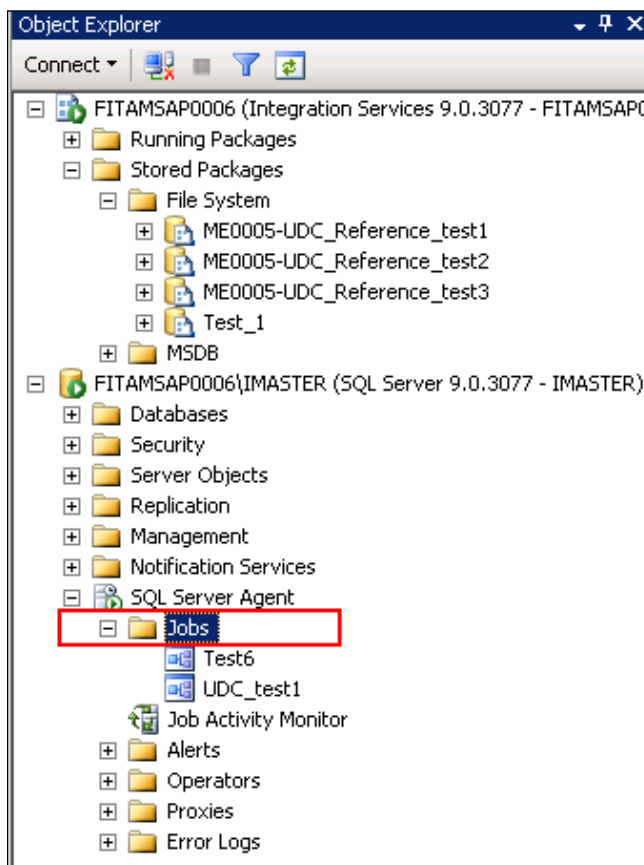
86 pav. SSIS paketo įdiegimo nustatymai

4. Sukuriamas darbas naujai importuotam paketui. Reikalinga prisijungti *DataBase Engine* režimu toje pačioje *Microsoft SQL Server Management Studio* aplinkoje.



87 pav. "DataBase Engine" jungtis

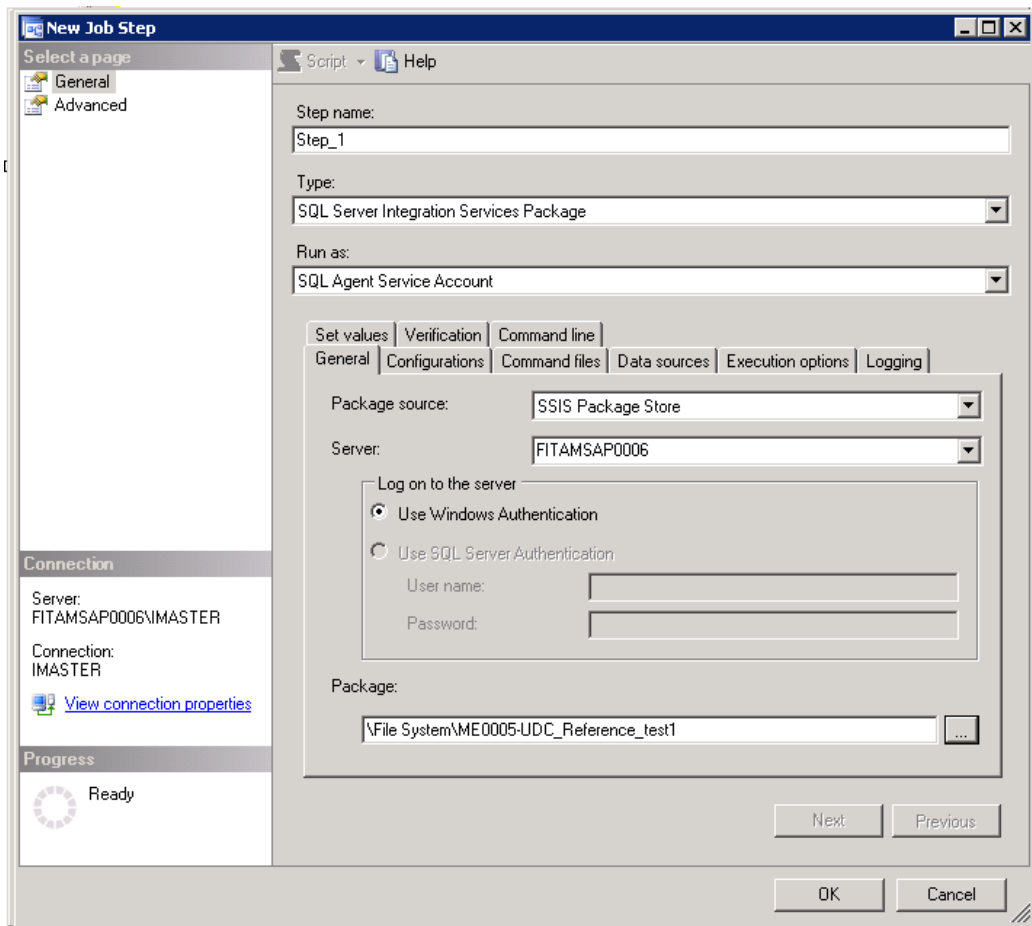
5. Sukuriamas darbas (angl. *jobs*) *SQL Server Agent* kataloge (*SQL Server Agent* -> *Jobs*->*New job*). Šiame kataloge saugomi visi sukurti darbai. Vienas SSIS paketas gali turėti daug skirtingų darbų (87 pav.).



88 pav. Naujo "Jobs" išsaugojimo vieta

6. Darbo parametrų nustatymas. Pagrindiniai nustatymai:

- Type – SQL Server Integration Services Package,
- Run as – SQL Agent Service Account,
- Package source – SSIS Package Store,
- Server – pasirenkamas serveris kuriame vykdomas SSIS paketas,
- Package – pasirenkamas importuotas paketas.



89 pav. Darbo nustatymai

7. Kuriamas automatinis darbo pasileidimas. Būtina suderinti laiką ir darbo pasileidimo dažnumą. Laikas gali įtakoti kitų duomenų teisingumą, korektiškumą. Kadangi duomenų kiekis yra didelis, dažnumas turi būti pasirinktas, kuo didesniais periodais (“Schedules” sector->New). 89 paveiksle pateikti laiko ir periodo nustatymai.

New Job Schedule

Name:

Schedule type: Enabled

One-time occurrence

Date: Time:

Frequency

Occurs:

Recurs every: week(s) on

Monday Wednesday Friday Saturday
 Tuesday Thursday Sunday

Daily frequency

Occurs once at:

Occurs every: Starting at:
Ending at:

Duration

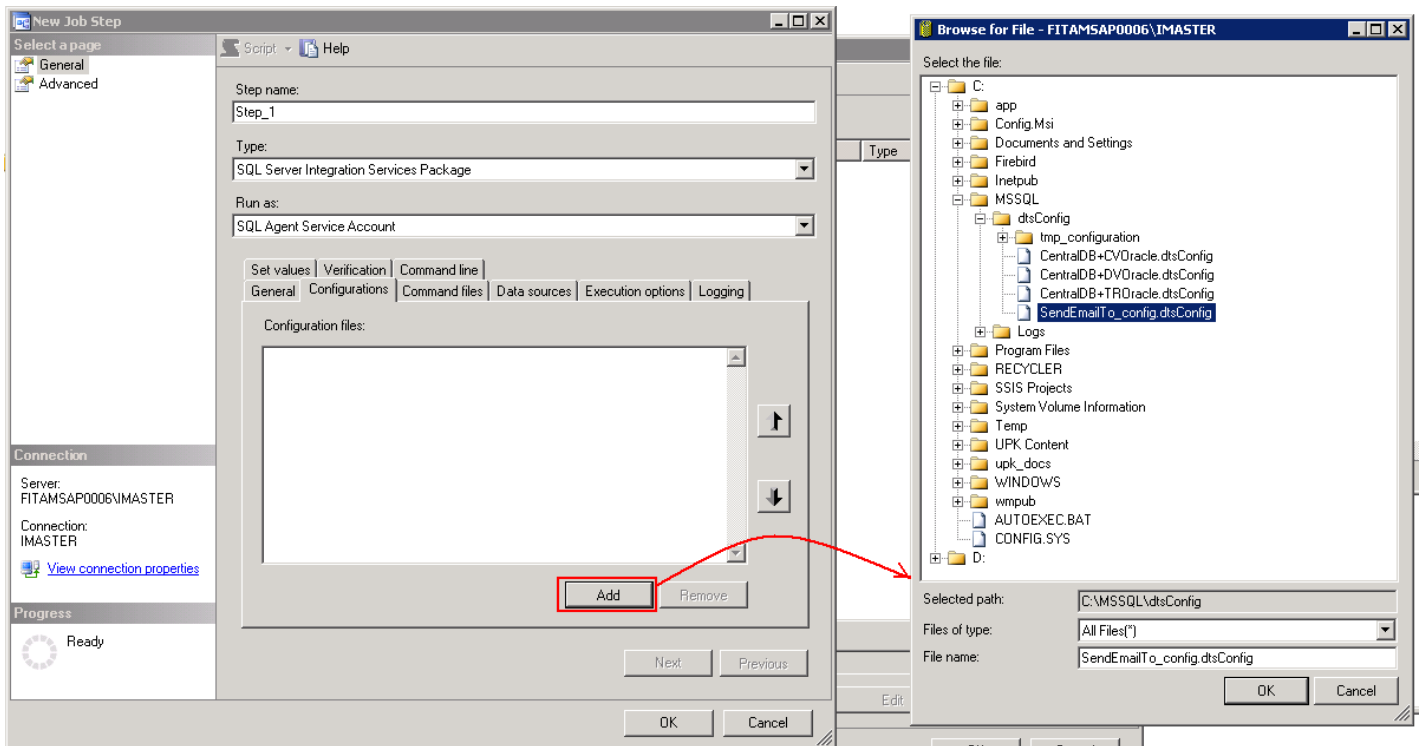
Start date: End date:
 No end date

Summary

Description:

90 pav. Laiko ir darbo periodiškumo nustatymai

Jeigu SSIS paketas naudoja konfigūracinį failą, yra galimybė darbo sukūrimo metu, jį įterpti (90 pav.).

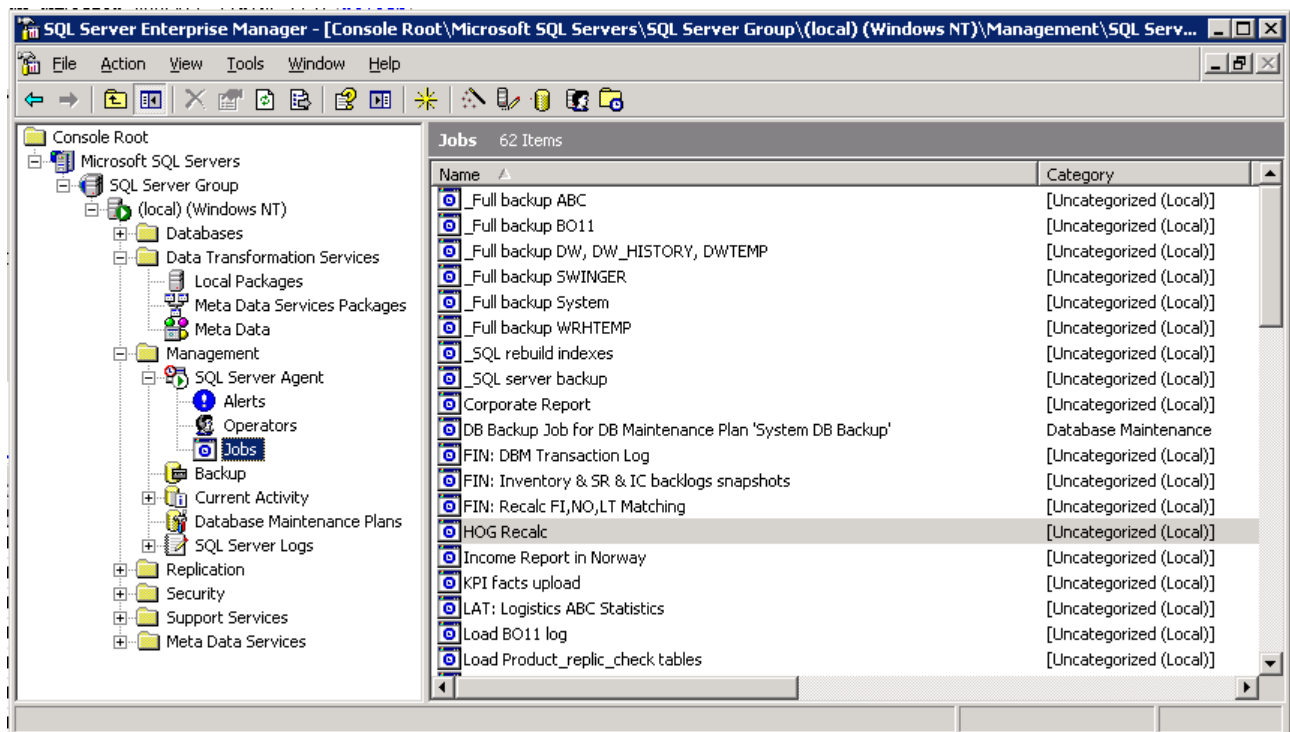


91 pav. SSIS pakete naudojamo konfigūracinio failo įtraukimas

DTS diegimo planas

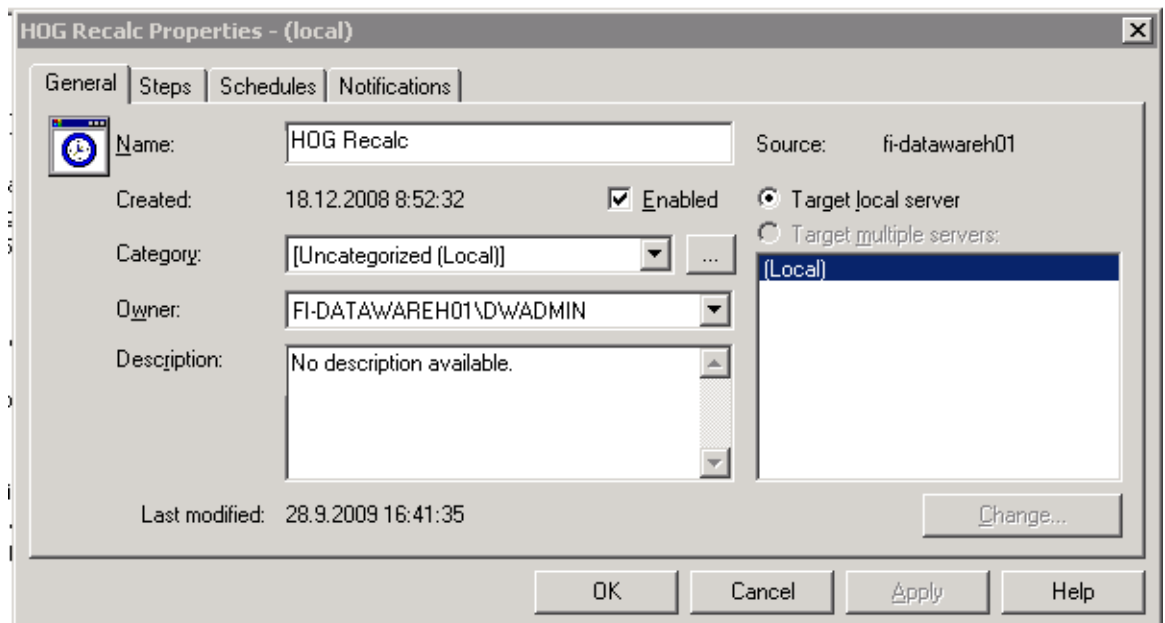
Kiekvienas DTS paketas gali turėti automatinį tvarkaraštį. Visi „automatiniai pasileidimai“ yra saugomi Management->SQL Server Agent - >Jobs skiltyje. Sukurtam „automatiniam darbui“ turi būti nustatytas periodas, laikas, užduotis.

Jobs elementas sudarytas iš keturių dalių: *General, Steps, Schedules, Notifications*.



92 pav. DTS paketų „Jobs“ hierarchija

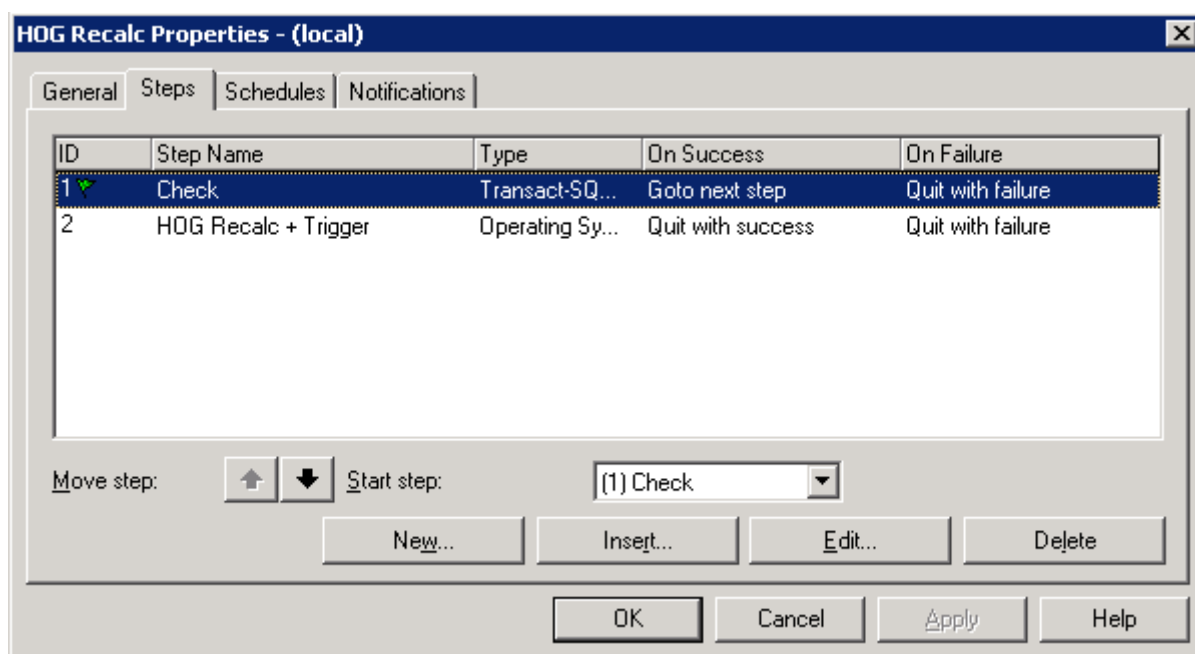
Kuriamo darbo *General* skiltyje įvedama pagrindinė informacija – Darbo pavadinimas, autorius, aprašymas. Sistema automatiškai uždeda varnelę prie *Enabled*, kuri reiškia, kad tik nustačius kitoje skiltyje laiką, darbas pasileis vykdytis (92 pav.).



93 pav. *General* darbo skiltis

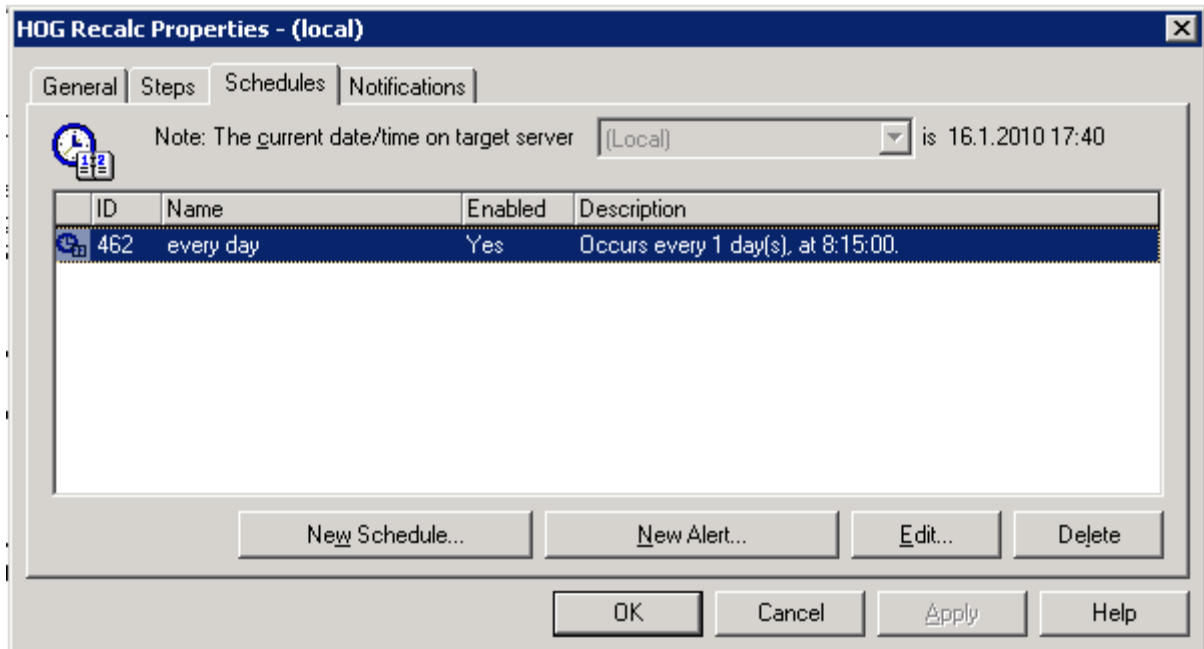
Darbo skiltyje *Steps* sudedami visi žingsniai, kuriuos turi paleisti naujasis darbas. Dažniausiai būna daugiau nei viena užduotis, kurios sudaro veiksmų grandinę, todėl būtina nustatyti, užduočių baigčių scenarijus (93 pav.).

Pavyzdžiui, naujai sukurtas darbas vykdomas tam tikrais periodais, paleidžia 3 užduotis. Standartiškai, jei visos užduotys sėkmingai įvykdomos tuomet darbas yra baigtas. Tačiau kyla problemų, kai viena iš užduočių nepavyksta, tuomet visas darbas sustoja. Kartais tai yra gerai, tačiau jei vykdomų uždavinių logika nesusijusi, tuomet nepaisant nepavykusios užduoties, sekanti turi vykdytis. Tai galima padaryti teisingai nustačius užduoties „*On Success*“, „*On Failure*“ laukus.



94 pav. *Steps* darbo skiltis

Automatinio pasileidimo laiką ir periodiškumą reikia nustatyti logiškai, nes atsitiktinai parinktas laikas gali sugadinti duomenų teisingumą (94 pav.). Pavyzdžiui, Jei pirma bus į duomenų bazę sukelti praeitos dienos pardavimai, o tik vėliau visi aktyvūs, egzistuojantys produktai, tai pardavimai duomenų bazėje bus neteisingi, nes tuomet kai buvo keliami duomenys dar nebuvo užkrauti nauji produktai.



95 pav. Schedules darbo skiltis

6. Duomenų integravimo modelių eksperimentinis tyrimas ir įvertinimas

Eksperimento tikslas yra išanalizuoti patobulintus duomenų integravimo procesus ir įsitikinti jų pagreitėjimu, įvertinant duomenų perdavimo laiką prieš ir po šablono pritaikymo.

Eksperimento planas. Eksperimentas vykdomas realioje įmonės aplinkoje, jame dalyvauja duomenų integravimo sistema ir programuotojas. Analizuojami 3 nepriklausomi kintamieji – projektas 1 (senas DI procesas), projektas 2 (DI procesas, pritaikius šabloną, naujas I), ir projektas 3, kai šablonas buvo pritaikytas naujoje technologinėje *SSIS* aplinkoje (naujas II). Priklausomas kintamasis – duomenų perdavimo laikas, apimantis duomenų įkėlimą, sujungimą ir persikirstymą pagal duomenų detalumą.

Tikrinamos dvi nulinės ir dvi alternatyvios hipotezės „projektas 1 – projektas 2“ ir „projektas 1 – projektas 3“ procesų atžvilgiu.

Nulinės hipotezės H_0 : bendrinių ir detalių duomenų atskyrimo šablono pritaikymas nesumažino duomenų perdavimo laiko, t. y. $\mu_{\text{projektas1}} = \mu_{\text{projektas2}}$, $\mu_{\text{projektas1}} = \mu_{\text{projektas3}}$.

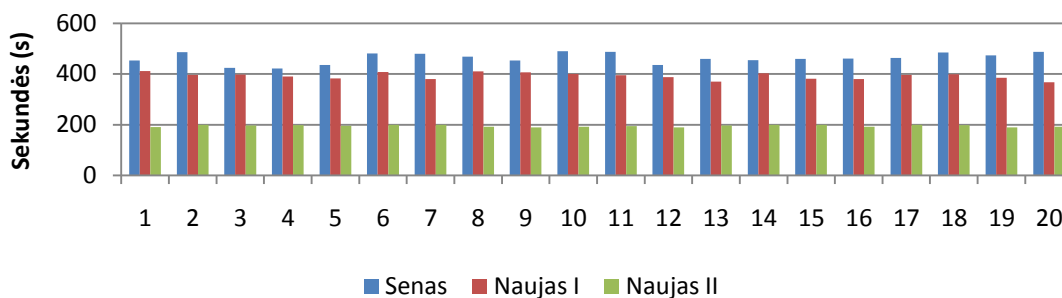
Alternatyvios hipotezės H_a : skirtingo detalumo duomenų integravimo šablono pritaikymas sumažino duomenų perdavimo laiką: $\mu_{\text{projektas1}} > \mu_{\text{projektas2}}$, $\mu_{\text{projektas1}} > \mu_{\text{projektas3}}$.

Eksperimento vykdymas. Atlikta 20 bandymų su kiekvienu projektu. Bandymų rezultatai sekundėmis pateikti 95 paveiksle

26 lentelė. Bandymų rezultatai.

Bandymo Nr.	1 projektas	2 projektas	3 projektas
1	7 min. 33 sek.	6 min. 52 sek.	3 min. 11 sek.
2	8 min. 06 sek.	6 min. 36 sek.	3 min. 19 sek.
3	7 min. 04 sek.	6 min. 38 sek.	3 min. 17 sek.
4	7 min. 01 sek.	6 min. 30 sek.	3 min. 18 sek.
5	7 min. 15 sek.	6 min. 22 sek.	3 min. 16 sek.
6	8 min. 01 sek.	6 min. 48 sek.	3 min. 20 sek.
7	7 min. 59 sek.	6 min. 20 sek.	3 min. 19 sek.
8	7 min. 48 sek.	6 min. 50 sek.	3 min. 12 sek.
9	7 min. 33 sek.	6 min. 46 sek.	3 min. 10 sek.
10	8 min. 10 sek.	6 min. 40 sek.	3 min. 12 sek.
11	8 min. 07 sek.	6 min. 35 sek.	3 min. 15 sek.
12	7 min. 15 sek.	6 min. 27 sek.	3 min. 10 sek.
13	7 min. 40 sek.	6 min. 10 sek.	3 min. 17 sek.
14	7 min. 34 sek.	6 min. 43 sek.	3 min. 20 sek.
15	7 min. 39 sek.	6 min. 21 sek.	3 min. 20 sek.
16	7 min. 41 sek.	6 min. 20 sek.	3 min. 12 sek.

17	7 min. 43 sek.	6 min. 36 sek.	3 min. 19 sek.
18	8 min. 05 sek.	6 min. 39 sek.	3 min. 18 sek.
19	7 min. 53 sek.	6 min. 25 sek.	3 min. 09 sek.
20	8 min. 07 sek.	6 min. 07 sek.	3 min. 12 sek.



96 pav. Eksperimento rezultatai

Analizė ir interpretavimas. Duomenų perdavimo laiko vidurkio, dispersijos, nuokrypio ir pasiklivimo intervalų įvertinimai \bar{x} , s^2 , s ir d apskaičiuoti pagal formules:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \text{ - aritmetinis vidurkis,}$$

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2, \quad s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \text{ - standartinis nuokrypis,}$$

$$\bar{x} - t_{(\alpha/2, n-1)} \frac{s}{\sqrt{n}} \leq x \leq \bar{x} + t_{(\alpha/2, n-1)} \frac{s}{\sqrt{n}} \text{ - pasiklivimo intervalas.}$$

Apskaičiuojame kiekvieno projekto aritmetinį bandymų vidurkį. Projektas 1 duomenų integravimo proceso aritmetinis bandymų vidurkis:

$$\begin{aligned} \bar{x}_1 &= \frac{453 + 486 + 424 + 421 + 435 + 481 + 479 + 468 + 453 + 490 + 487 + 435 + 460 + 454 + 459 + 461 + 463 + 485 + 473 + 487}{20} = \\ &= \frac{9272}{20} = 464 \text{ sek.} = 7 \text{ min } 44 \text{ sek.} \end{aligned}$$

Projektas 2 duomenų integravimo proceso aritmetinis atliktų bandymų trukmės vidurkis:

$$\begin{aligned} \bar{x}_2 &= \frac{412 + 396 + 398 + 390 + 382 + 408 + 380 + 410 + 406 + 400 + 395 + 387 + 370 + 403 + 381 + 380 + 396 + 399 + 385 + 367}{20} = \\ &= \frac{7845}{20} = 392 \text{ sek.} = 6 \text{ min } 32 \text{ sek.} \end{aligned}$$

Projektas 3 duomenų integravimo proceso, esančio naujoje aplinkoje, aritmetinis atliktų bandymų trukmės vidurkis :

$$\bar{x}_3 = \frac{191+199+197+198+196+200+199+192+190+192+195+190+197+200+200+192+199+198+189+192}{20} =$$

$$= \frac{3906}{20} = 195 \text{ sek.} = 3 \text{ min } 15 \text{ sek.}$$

Apskaičiuojame kiekvieno projekto bandymų variacija ir standartinis nuokrypis, kuris rodo rezultatų pasiskirstymo skaidą:

Projektas 1 variacija ir standartinis nuokrypis:

$$s^2 = \frac{1}{19} ((453-464)^2 + (486-464)^2 + (424-464)^2 + (421-464)^2 + (435-464)^2 + (481-464)^2 + (479-464)^2 + (468-464)^2 +$$

$$(453-464)^2 + (490-464)^2 + (487-464)^2 + (435-464)^2 + (460-464)^2 + (454-464)^2 + (459-464)^2 + (461-464)^2 + (463-464)^2 + (485-464)^2 +$$

$$(473-464)^2 + (487-464)^2) = \frac{1}{19} (121+484+1600+1849+849+289+225+16+121+676+529+961+16+100+25+9+1+441+81+529) = 469,737$$

$$s = 21,7$$

Projektas 2 projekto variacija ir standartinis nuokrypis:

$$s^2 = 140,1; \quad s = 11,8$$

Projektas 3 projekto naujoje aplinkoje, variacija ir standartinis nuokrypis:

$$s^2 = 15,2; \quad s = 3,9$$

Pasiklovimo intervalas nurodo sritį iš kurios didžiausia tikimybe, bus gauta rezultato reikšmė.

Žemiau pateikta kiekvieno projekto pasiklovimo intervalai.

Seno projekto pasiklovimo intervalas su pasiklovimo lygmeniu $\alpha = 0.05$:

$$d_{x1} = t_{(\alpha/2, n-1)} \frac{s}{\sqrt{n}} = t_{(0,05/2, 20-1)} \frac{21,7}{\sqrt{20}} = 2,093 \frac{21,7}{\sqrt{20}} = 10,16$$

$$464 - 10,16 \leq x \leq 464 + 10,16$$

$$454,16 \leq x \leq 474,16$$

Patobulinto projekto pasiklovimo intervalas su pasiklovimo lygmeniu $\alpha = 0.05$:

$$d_{x2} = t_{(\alpha/2, n-1)} \frac{s}{\sqrt{n}} = t_{(0,05/2, 20-1)} \frac{11,8}{\sqrt{20}} = 2,093 \frac{11,8}{\sqrt{20}} = 5,53$$

$$392 - 5,53 \leq x \leq 392 + 5,53$$

$$386,47 \leq x \leq 397,53$$

Patobulinto projekto naujoje aplinkoje, pasiklovimo intervalas su pasiklovimo lygmeniu $\alpha = 0.05$:

$$d_x = t_{(\alpha/2, n-1)} \frac{s}{\sqrt{n}} = t_{(0,05/2, 20-1)} \frac{3,9}{\sqrt{20}} = 2,093 \frac{3,9}{\sqrt{20}} = 1,83$$

$$195 - 1,83 \leq x \leq 195 + 1,83$$

$$193,17 \leq x \leq 196,83$$

čia pasiklovimo lygmuo $\alpha = 0.05$, bandymų skaičius $n = 20$, *Student* kvantilis $t_{(\alpha/2, n-1)} = t_{(0,025,19)} = 2,093$.

Įvertinimų rezultatai pateikiami 1 lentelėje.

27 lentelė. Eksperimente apskaičiuoti įvertinimai

Įvertinimas	Projektas 1 (senas)	Projektas 2 (naujas I)	Projektas 3 (naujas II)
\bar{x}	7 min. 44 sek.	6 min. 32 sek.	3 min. 15 sek.
s^2	469,6 sek.	140,1 sek.	11,8 sek.
s	21,7 sek.	11,8 sek.	3,9 sek.
d	10,16 sek.	5,53 sek.	1,83 sek.

Eksperimento rezultatai rodo, kad įvertintas perkėlimo laiko vidurkis projekto 2 ir projekto 3 atvejais yra daug mažesnis, atitinkamai mažesni ir dispersijos, standartinio nuokrypio ir pasiklovimo intervalo įvertinimai, vadinasi, įvertinimo rezultatai yra geresni ir stabilesni.

Nulinę hipotezę H_0 galima atmesti, jei $t_0 > t_{\alpha/2, n+m-2}$, čia

$$t_0 = \frac{\bar{x} - \bar{y}}{s_p \sqrt{\frac{1}{n} + \frac{1}{m}}},$$

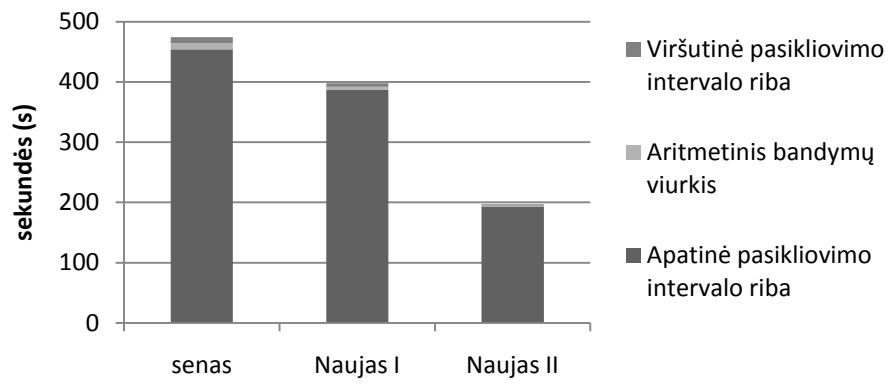
$$s_p = \sqrt{\frac{(n-1)s_x^2 + (m-1)s_y^2}{n+m-2}},$$

s_x^2 ir s_y^2 yra atskirų bandymų variacijos.

Pirmą nulinę hipotezę galima atmesti, nes $t_0 = 12,7$, $t_{0,025;38} = 2,024$ ir $t_0 > t_{0,025;38}$; antrą nulinę hipotezę tuo labiau galima atmesti, kadangi $t_0 = 68,0$; $t_{0,025;38} = 2,024$ ir $t_0 > t_{0,025;38}$.

Eksperimentas parodė, kad šablono taikymas GNT įmonėje sutrumpino proceso trukmę 45,4%, o kartu pritaikytos SSIS technologijos dėka gautas 81,99% sutrumpėjimas. Duomenų integravimo procesas tapo lankstesnis, į jį bus galima paprasčiau įjungti naujus duomenų šaltinius. Eksperimentas buvo atliekamas realioje įmonės aplinkoje įvairiais laiko momentais, esant skirtingiems serverio apkrautumams. Visi procesai buvo bandomi su tais pačiais duomenimis, siekiant išvengti galimos duomenų rinkinių įtakos procesų vykdymo laikui.

Seno ir naujo procesų vidurkiai ir jų pasiklovimo intervalai pavaizduoti 96 paveiksle.



97 pav. Eksperimente tirtų procesų duomenų perdavimo laikų vidurkiai ir jų pasiklovimo intervalų ribos

7. Išvados

1. Atlikta literatūros šaltinių ir įrankių analizė parodė, kad SSIS technologija yra perspektyviausia tiriamos įmonės duomenims integruoti, kadangi dėl jo panašumo į anksčiau taikytas DTS funkcijas IT skyriaus darbuotojams bus nesunku persiorientuoti į šį produktą ir bus galima greičiau pereiti prie SSIS. .
2. ETL procesų realizavimo metu atlikta analizė parodė, kad perkeliamos duomenų struktūros turi pasikartojančių duomenų, kuriuos galima atskirti ir patobulinti perkėlimo procesą. Ši analizė taip pat atskleidė aktualias problemas: tikslinga sukurti tokius perkėlimo procesus ir duomenų struktūras, kad būtų lengva jas papildyti įtraukiant naujus duomenų šaltinius.
3. Realizuojant integravimo procesą, buvo patobulinta duomenų struktūra, dėl to procesas tapo lankstesnis galimiems pasikeitimams ir buvo išspręstas naujų padalinių įjungimo į procesą uždavinys.
4. Įmonės praktinės patirties analizė parodė, kad testuoti perkeliamus duomenis reikia ir veikimo metu, todėl buvo sukurtas testavimo modelis, paremtas loginėmis proceso taisyklėmis ir skirtumų tarp dviejuose serveriuose saugomos produktų informacijos radimu. Jis bus pritaikomas ir ateityje, kai senasis procesas neveiks
5. Nustatyta galimybė atskirti bendrinius ir detalius integruojamų šaltinių duomenis buvo aprašyta kaip bendrinių ir detalių duomenų atskyrimo šablonas, kurį galima taikyti panašioms duomenų integravimo sprendimams.
6. Eksperimentas parodė, kad šablono taikymas GNT įmonėje sutrumpino proceso trukmę 45,4%, o kartu pritaikytos SSIS technologijos dėka gautas 81,99% sutrumpėjimas. Duomenų integravimo procesas tapo lankstesnis, į jį galima paprasčiau įjungti naujus duomenų šaltinius. Eksperimentas buvo atliekamas realioje įmonės aplinkoje įvairiais laiko momentais, esant skirtingiems serverio apkrautumams. Visi procesai buvo bandomi su tais pačiais duomenimis, siekiant išvengti galimos duomenų rinkinių įtakos procesų vykdymo laikui.

8. Literatūra

- [1] Knyga: Wayne Eckerson, Colin White. „Evaluating ETL and Data Integration platforms“,2003 by 101communications LLC, United States. [žiūrėta 2008-10-14]. Prieiga internete: http://mimage.hummingbird.com/alt_content/binary /pdf/collateral/mc/etlreport.pdf
- [2] Srinath Shankar, Ameet Kini, David j DeWitt, Jeffery Naughton (Viskonsino universitetas, kompiuterių mokslo skyrius, Madisonas). „Integrating databases and workflow systems“,2005, rugsėjis, Nr.3 žurnalas: „SIGMOD Record“. [žiūrėta 2008-10-13]. Prieiga internete: <http://www.sigmod.org/record/issues/0509/p5-special-sw-section-1.pdf>
- [3] A.D. Jhingran, N. Mattos, M.Pirahesh. „Information Integration, a Research agenda“, žurnalas: „IBM System“, 2002, spalio, Nr. 4. [žiūrėta 2008-10-13]. Prieiga internete: <http://researchweb.watson.ibm.com/journal/sj/414/jhingran.pdf>
- [4] J. D. D. Daniel, K. N. Goh, S. M. Yusop. „Data Transformation Services (DTS): Creating Data Mart by Consolidating Multi-Source Enterprise Operational Data“, „PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY“ 2007, Spalis. [žiūrėta 2008-10-13]. Prieiga internete: <http://www.waset.org/pwaset/v25/v25-89.pdf>
- [5] Knyga: Louis Davidson. „Professional SQL Server 2000 DataBase Design“, 2001, Wrox Press. United States. [žiūrėta 2008-10-13]. Prieiga internete: http://mulmedp.tu-sofia.bg/ebooks/Wrox-Professional_SQL_Server_2000_Database_Design.pdf
- [6] Kamal Hathi. „An Introduction to SQL Server 2005 Integration Services“. Publikuota: Gegužės 1, 2005. [žiūrėta 2008-10-10]. Prieiga internete: <http://www.microsoft.com/technet/prodtechnol/sql/2005/intro2is.msp>
- [7] decipherinfosys. „DTS vs SSIS: A basic overview“. Publikuota: Kovo 11, 2008.[žiūrėta 2008-10-11]. Prieiga internete: <http://decipherinfosys.wordpress.com/2008/03/11/dts-vs-ssis-a-basic-overview/>
- [8] Allan Mitchell. „DTS to SSIS Migration: Upgrading DTS packages to SQL Server Integration Services“. Publikuota: Spalio 18, 2005.[žiūrėta 2008-10-12] Prieiga internete: <http://www.simple-talk.com/sql/sql-server-2005/dts-to-ssis-migration/>
- [9] Microsoft. „Kas nauja SQL Server 2005“. [žiūrėta 2008-10-12 .] Prieiga internete: <http://www.microsoft.com/lietuva/sql/prodinfo/whats-new-in-sqlserver2005.msp>
- [11] MS SQL 2005 Server, *SQL Server Integration Services* pagalba (*help*)

- [12] Dr David Waddington, „Ten Tips for Selecting a Data Integration Tool“, publikuota: liepa, 2008, [žiūrėta: 2009-05-24], prieiga internete: <http://www.siamultiserve.com/>
- [13] Microsoft TechNet, prieiga internete: [http://technet.microsoft.com/en-us/library/ms137701\(SQL.90\).aspx](http://technet.microsoft.com/en-us/library/ms137701(SQL.90).aspx); [žiūrėta:2010-01-20].
- [14] Brian Mains „Using SQL Server Integration Services (SSIS) Control Flow in SQL Server 2005“, publikuota: 2007 liepa [žiūrėta: 2010-01-21], prieiga internete: <http://dotnetslackers.com/articles/sql/UsingSQLServerIntegrationServicesSSIS.aspx>
- [15] James O. Coplien Naperville, Illinois „Software patterns – Design Pattern Definition“, [žiūrėta: 2010-02-28], prieiga internete: <http://www.hillside.net/component/content/article/50-patterns/222-design-pattern-definition>
- [16] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, S. Angel, A Pattern Language, „Christopher Alexander’s Pattern Language“, publikuota: 1999 liepa, „IEEE TRANSACTIONS ON PROFESSIONAL COMMUNICATION“ leidimas: 42, nr. 2, [žiūrėta:2010-03-05].
- [17] Gamma Helm Johnson Vlissides, knyga: „Design Patterns Elements of Reusable Object Oriented Software“, [žiūrėta 2010-03-11].
- [18] Adeyemi Sidikat, Aremu Mukaila Ayanda, „Impact Assessment of Business Process Reengineering on Organisational Performance“, European Journal of Social Sciences – leidimas 7, Nr .1 (2008).
- [19] Indrė Kungytė, Kauno technologijos universitetas, straipsnis: „Bendrinių duomenų atskyrimo šablonas ir jo taikymas UAB „GNT Lietuva“ ETL procesuose“, publikuota: 2010 gegužė, „Informacinė visuomenė ir universitetinės studijos (IVUS)“ konferencijoje.

9. Terminų ir santraukų žodynas

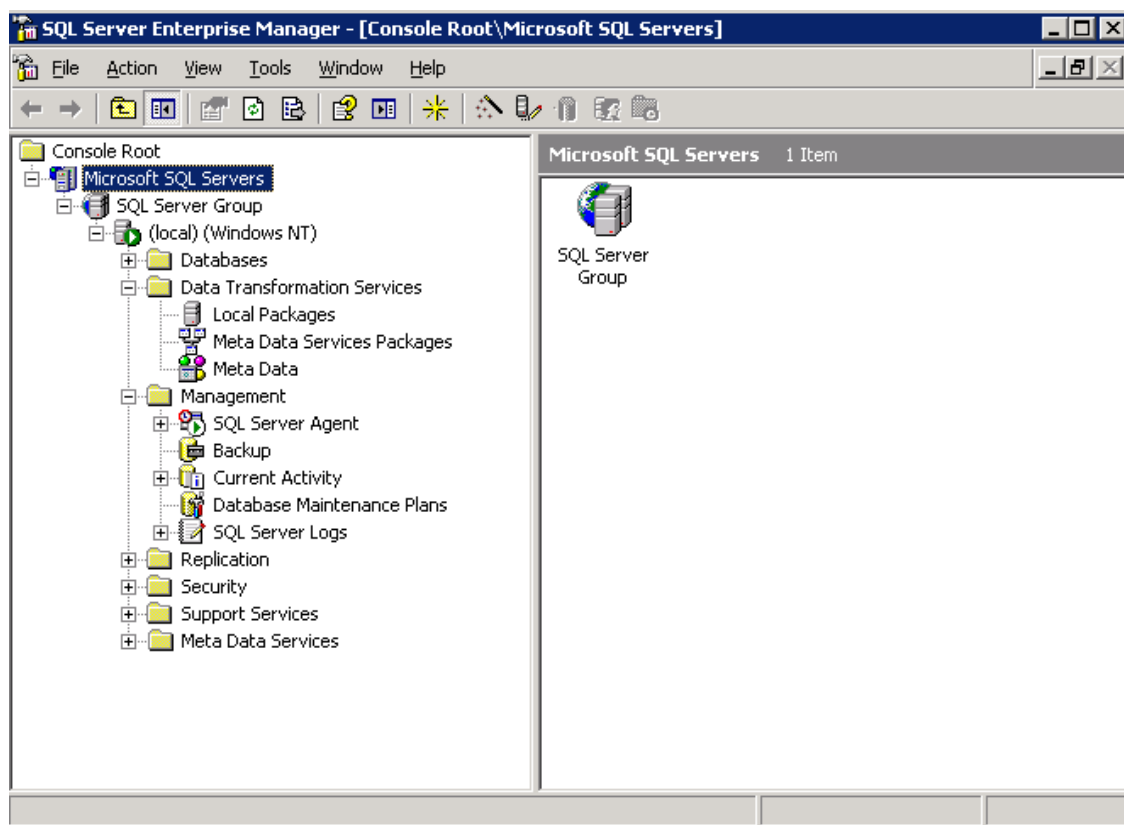
1. DB (angl. Database) – duomenų bazė.
2. SSIS (angl. *SQL Server Integration Services*) – duomenų integravimo paslauga MS SQL 2005 Server aplinkoje.
3. DTS (*Data Transformation Services*) - duomenų integravimo įrankis MS SQL 2000 Server aplinkoje.
4. DBM – vidinė GNT ERP sistema, kuri apima visų skyrių darbą šalies padalinyje. Sistema saugo visas transakcijas, kurios įvyko darbo metu.
5. IDS (angl. *Business Intelligence Development Studio*) - įrankis skirtas kurti ir redaguoti SSIS paketus.
6. ERP (angl. *Enterprise resource planning*) - nauja įmonės resursų planavimo strategija, plačiai plintanti daugelyje Europos valstybių. Tai strategija, kurios pagrindą sudaro visų kompanijos departamentų ir veiklos grupių integravimas į vieną kompiuterinę sistemą, kuri aptarnauja visus departamentus.
7. XML (angl. *Extensible Markup Language*) - W3C rekomenduojama bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba.
8. MAPI (angl. *Messaging Application Program Interface*) – MS Windows programa, kuris siunčia elektroninius laiškus per Windows taikomąsias programas, prikabina dokumentus, pranešimus.
9. FTP (angl. *File Transfer Protocol*) – standartas failų perdavimui. FTP protokolą realizuojanti programinė įranga paprastai vadinama FTP klientais.
10. SMTP (angl. *Simple Mail Transfer*) —standartas elektroninio pašto laiškų perdavimui internete. Naudojamas elektroniniams laiškam pristatyti į gavėjo elektroninę pašto dėžutę.

10. Priedai

- 10.1 Priedas 1. MS SQL Server 2000 aplinka
- 10.2 Priedas 2. MS SQL Server 2005 aplinka
- 10.3 Priedas 3. Insert_Master_DBM MS SQL užklausa

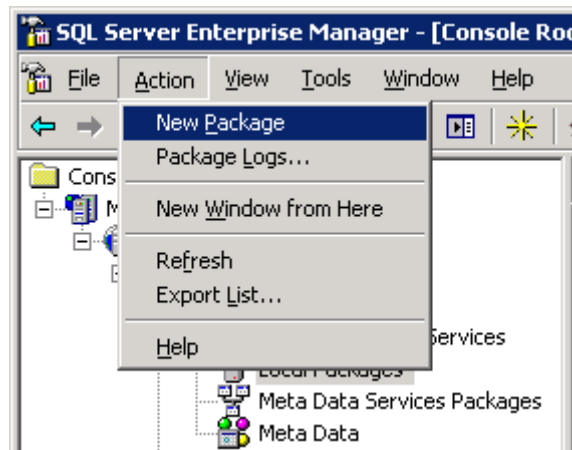
10.1 Priedas 1. MS SQL Server 2000 aplinka

MS SQL Server 2000 Enterprise Manager paslaugos pagalba, kuriamas naujas ETL procesas, skirtas, duomenims perkelti iš vienos vietos į kitą, yra galimybė nustatyti automatinį DTS vykdymo tvarkaraštį [4]. 97 paveiksle pavaizduotas pagrindinis *Enterprise Manager* įrankio navigacinis langas.



98 pav. „MS SQL Server Enterprise Manager“ langas

Naujas DTS procesas *MS SQL Server 2000* aplinkoje pradedamas kurti pasirinkus *Actions* skiltyje *New Package* (98 pav.).

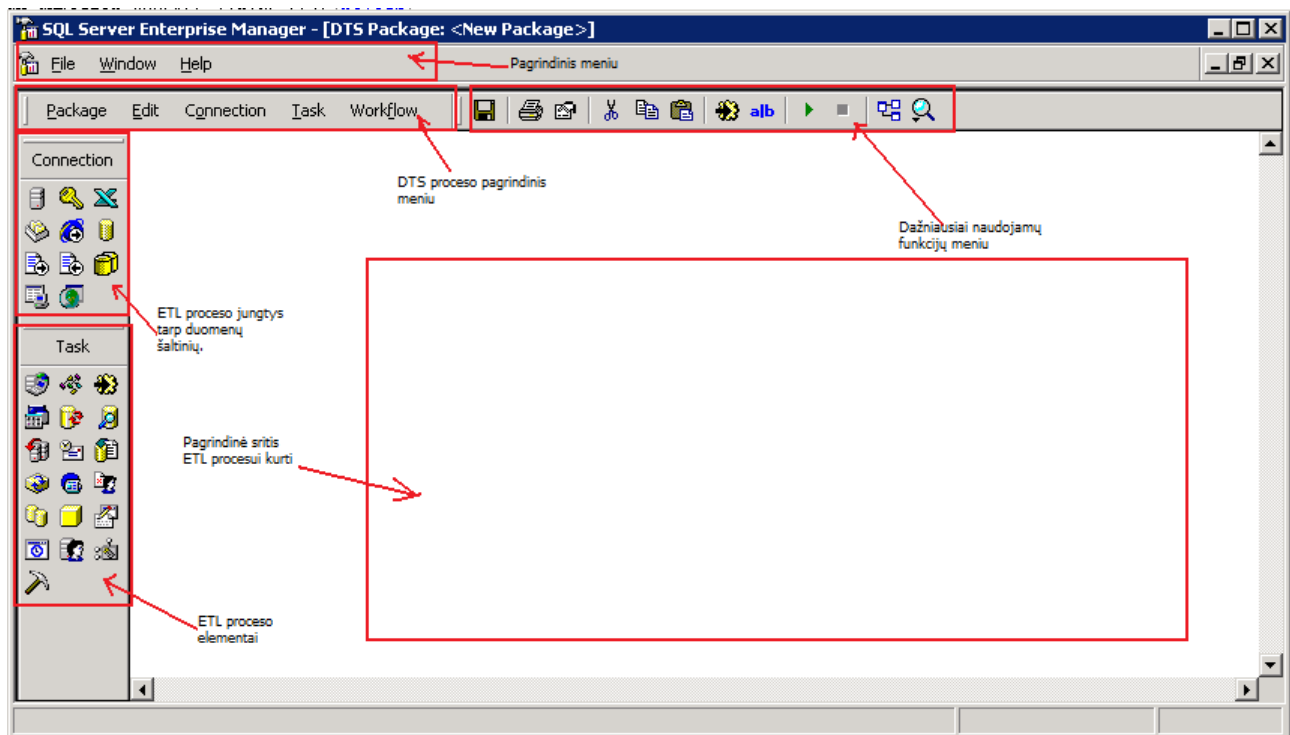


99 pav. Naujo DTS kūrimas

Kuomet sukuriamas DTS paketas, sistema atveria naują aplinką, skirtą ETL procesui kurti. DTS aplinka suskirstyta į 6 sritis:

1. Pagrindinis meniu – šioje juostoje patalpintos standartinės funkcijos, kurios yra ir kitose sistemose, elementuose ir funkcijose. Standartinėmis funkcijomis laikoma *File*, *Windows*, *Help*. Jose yra bendri DTS'o nustatymai, navigacija.
2. DTS meniu – šioje skiltyje galima išsaugoti DTS pakeitimus, pasirinkti ETL proceso kūrimo elementus, ryšio tipą tarp elementų, jungtis tarp šaltinių.
3. Funkcijų meniu – juostoje pateikiami dažniausiai naudojami elementai, kurie yra standartiniai t.y. klijavimas, kopijavimas, kirpimas, spausdinimas ir t.t.
4. Užduočių elementų meniu – Tai DTS meniu dalis, kurioje grafiškai pateikti visi galimi elementai-užduotys. Sritis yra nebūtina, nes visus elementus galima rasti DTS meniu srityje, tačiau atskirta vartotojų patogumui.
5. Jungčių meniu – Tai DTS meniu dalis, kurioje grafiškai pateikiamos visos galimos jungtys tarp duomenų šaltinių.
6. ETL proceso sritis – Šioje srityje formuojamas duomenų perkėlimo ir transformavimo procesas. Jame sudedami visi elementai, jungtys, duomenų šaltiniai.

97 paveiksle patektos visos sritys, matomos vartotojui kuriant DTS paketą arba atsidarius seną - redagavimui [5].



100 pav.

Naujo DTS kūrimas

DTS procese galima naudoti šiuos užduočių elementus :

- Active Script Task – elementas teikia galimybę naudoti programuotojo parašytą "ActiveX script" kodą , kurį galima atnaujinti, tobulinti.
- Transform Data Task – vykdomas duomenų kopijavimas ir transformavimas OLE DB duomenų šaltinyje. Duomenų transformavimo elementas yra pagrindinis duomenų išgavimo variklis DTS'e.
- Execute Process Task – elementas skirtas programų ir paketų iškvietimui ir jų vykdymui. Tačiau elementas gali vykdyti tik standartines programas, tokias kaip MS Excel, MS Word.
- Execute SQL Task – elementas vykdo SQL užklausas paketo vykdymo metu. Yra galimybė aprašyti daug globalių kintamųjų, išsaugoti globaliuose kintamuosiuose gautus įrašus, sunaikinti DB lentelę, sukurti naują objektą prieš įrašant į jį duomenis, leisti išsaugomas procedūras. Viename elemente gali būti saugoma daugiau nei viena užklausa.
- Data Driven Query Task - elementas leidžia lanksčiai vykdyti duomenų operacijas paremtas Transact-SQL, įtraukiant saugomas procedūras ir *INSERT*, *UPDATE*, *DELETE* tipo užklausas. Kiekvienai duomenų sąrašo eilutei, elementas parenka ir vykdo vieną iš keleto MS SQL užklausų.

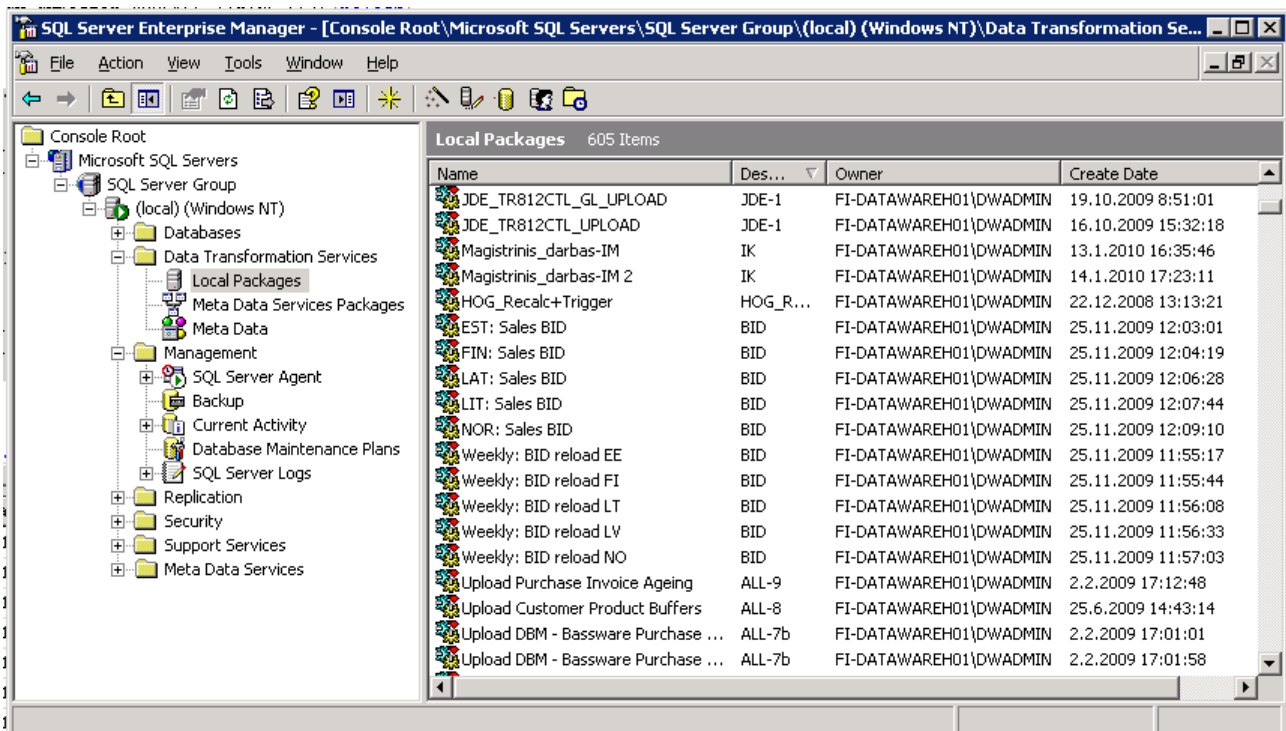
- Copy SQL Server Object Task – elementas leidžia kopijuoti MS SQL Server objektus iš vieno SQL Server paketo į kitą. Galimybė perkelti objektus - duomenis, lenteles. Papildomų objektų pagalba įtraukiami indeksai ir apribojimai.
- Send Mail Task – elementas leidžia išsiųsti elektroninį laišką iš DTS paketo. Kad naudotis šiuo elementu turi būti įdiegta MAPI.
- Bulk Insert Task – elementas leidžia nukopijuoti didelį kiekį duomenų į MS SQL Server lentelę. Užtikrina duomenų judėjimo spartą. Duomenų transformacija nevykdoma, kol į duomenų paskyros vietą nėra perkelti visi nukopijuoti įrašai.
- Execute Package Task – elementas padidina įmonės DTS pajėgumą, leisdamas vykdyti kitus DTS paketus darbų sekoje.
- Message Queue Task – elementas leidžia naudoti pranešimų užklausas. Siųsti ir gauti pranešimus tarp DTS paketų. Šie pranešimai gali būti pateikti įvairia forma: tekstu, dokumentais, globaliais kintamaisiais ir jų reikšmėmis, kurie siunčiami, kai paskirties vieta yra neprieinama, užimta.
- Transform Error Message Task – leidžia atlikti veiksmą, kuomet prieš tai esančio elemento baigtis "False".
- Transfer Databases Task – leidžia kopijuoti serverio informaciją esančią duomenų bazėse. Šios užduotys apima situacijas, kurių negali išspręsti "*Copy SQL Server Objects task*" elementas.
- Analysis services processing Task – elementas vykdo analizės paslaugų objektus: kubus, dimensijas ir gavybos modelius. Keli objektai gali būti tvarkomi tuo pačiu metu.
- Dynamic Properties Task – traukia reikšmes iš šaltinių, esančių DTS paketo išorėje. Paketo vykdymosi metu priskiriamos reikšmės pasirinktam paketui. Išoriniai duomenys prisikirti elementui yra neatpažįstami, kol paketas nepasileidžia vykdytis.
- Transfer Job Task – leidžia kopijuoti "darbus" (angl. *jobs*).
- Transfer Login Task – leidžia kopijuoti prisijungimus.

Šiame skyrelyje nepateikiamas elementų aprašymas, nes elemento pavadinimas parodo iš kokio šaltinio duomenys gali būti paimti ir į kokio tipo paskyros vietą - padedami.

- Microsoft OLE DB Provider,
- Microsoft Access,
- Microsoft Excel 97-2000,
- dBase 5,

- HTML file (source) ,
- Paradox 5.X ,
- Text File (source),
- Text file (destination) ,
- Microsoft ODBC Driver for Oracle ,
- Microsoft Data Link,
- Other Connections .

Sukurti DTS paketai pasiekiami pasirinkus „Local Packages“ sritį (100 pav.).

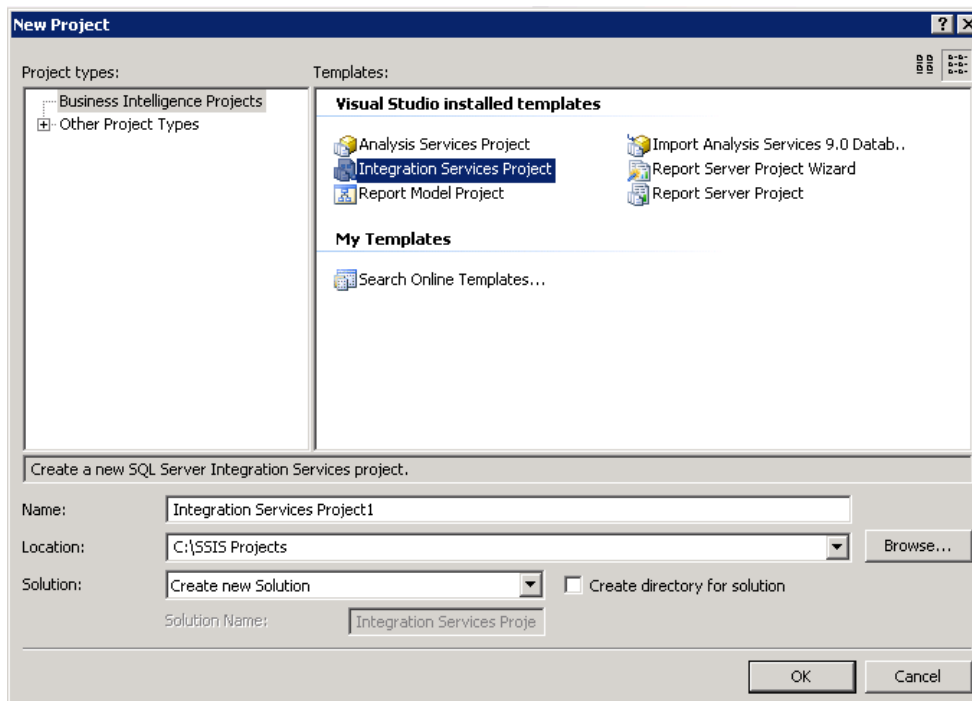


101 pav. DTS paketų hierarchija

10.2 2 Priedas. MS SQL Server 2005 aplinka

MS SQL 2005 serverio aplinkoje yra atskiras modulis, kuris naudojamas ETL procesams kurti. Naujas ETL procesas kuriamas naudojant *Business Intelligence Development Studio* įrankį, kuriame yra SSIS paslauga.

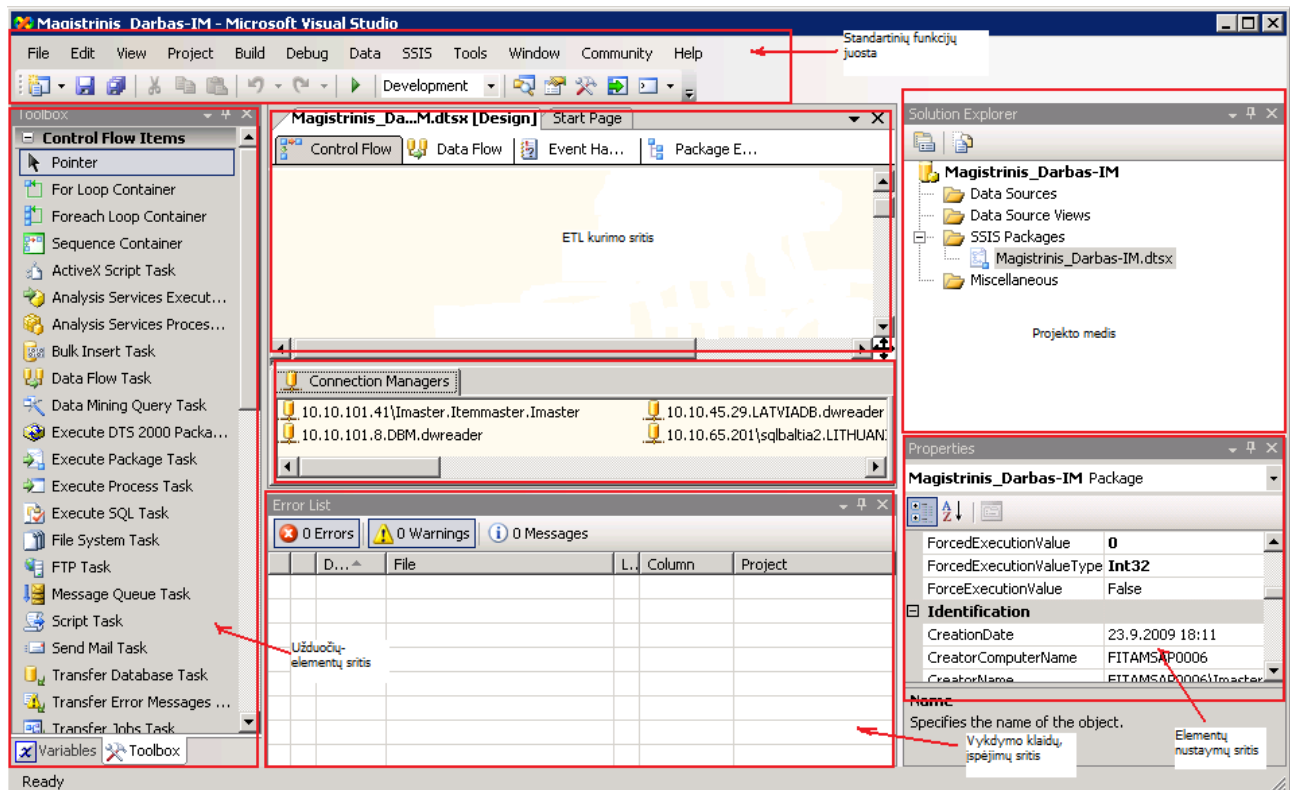
SSIS projektas kuriamas pasirinkus File->New->Project. Tuomet programa atveria naujo projekto langą, kuriame reikia pasirinkti *Integration Services Project* dokumento tipą. Naujo SSIS kūrimo etapas pavaizduotas 101 paveiksle.



102 pav. SSIS paketo sukūrimas

100 paveiksle pateiktas ETL proceso kūrimo aplinka. Langas suskirstytas į 7 pagrindines sritis:

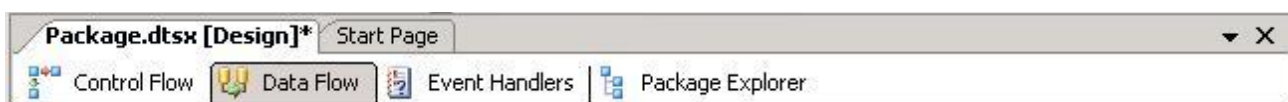
1. Standartinių funkcijų juostos – šioje juostoje yra standartinės funkcijos ir SSIS projektams skirtos paslaugos: *SSIS*, *Build*, *Debug* ir t.t.
2. Užduočių elementų sritis – elementų sąrašas, kurie gali būti panaudoti formuojant ETL procesą. Elementai suskirstyti į grupes pagal atliekamas funkcijas.
3. ETL kūrimo sritis – Šioje srityje formuojamas duomenų perkėlimo ir transformavimo procesas. Jame sudedami visi elementai, jungtys, duomenų šaltiniai.
4. Prisijungimų (angl. *Connecions*) sritis – duomenų šaltinių jungčių sąrašas, kurios naudojamos projekte.
5. Vykdyto klaidų, įspėjimų sritis – klaidų, įspėjimų sąrašas, kuris atsiranda vykdant elementą arba visą projektą.
6. Projekto medis.
7. Elemento savybių sritis – šioje srityje nustatomi elemento parametrai, suteikiamos pradinės elemento reikšmės.



103 pav. SSIS paketo pagrindinis navigacijos langas

Pagrindinė projekto dalis padalinta į keturių sekcijų valdymą (101 pav.):

- Control Flow - valdymo skiltyje formuojamas procesas. Sudedami visi elementai reikalingi projekto logikai išpildyti.
- Data Flow – skiltis skirta duomenų perkėlimo veiksmams atlikti.
- Event Handlers - elemento antriniam veiksmams formuoti.
- Package Explorer – turinys visų panaudotų elementų, parametrų, nustatymų.



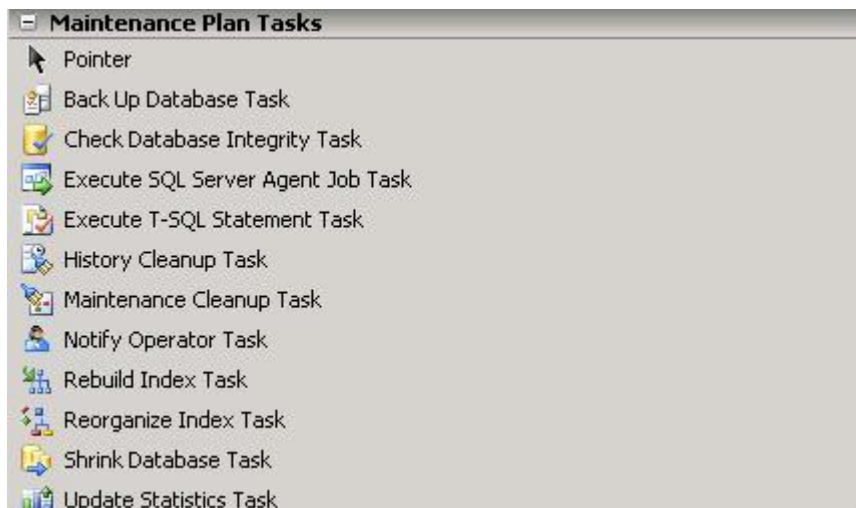
104 pav. Projekto valdymo sekcijos

Control Flow valdymo skydelis

Elementai, kurie gali būti panaudoti *Control Flow* skiltyje yra dviejų tipų: *Maintenance Plan tasks*, *Control Flow Items*. *MS SQL Server 2000* DTS kūrimo aplinkoje šie elementai taip yra, tačiau jų pasirinkimas labai mažas.

MS SQL Server 2005 įgyvendina priežiūros planus SSIS projektuose suteikiant lankstumo individualių užduočių kūrimui. Planai peržiūrimi vykdant Transact-SQL komandas, vietoj ankščiau naudotų neregistruotų saugomų procedūrų *MS SQL Server 2000* aplinkoje.

Kitas būdas, tai proceso vykdymo istorijos peržiūrėjimas, kuris trikdžių priežiūros schemas daro patogesnes. *Maintenance Plan tasks* elementai pateikti 103 paveiksle.



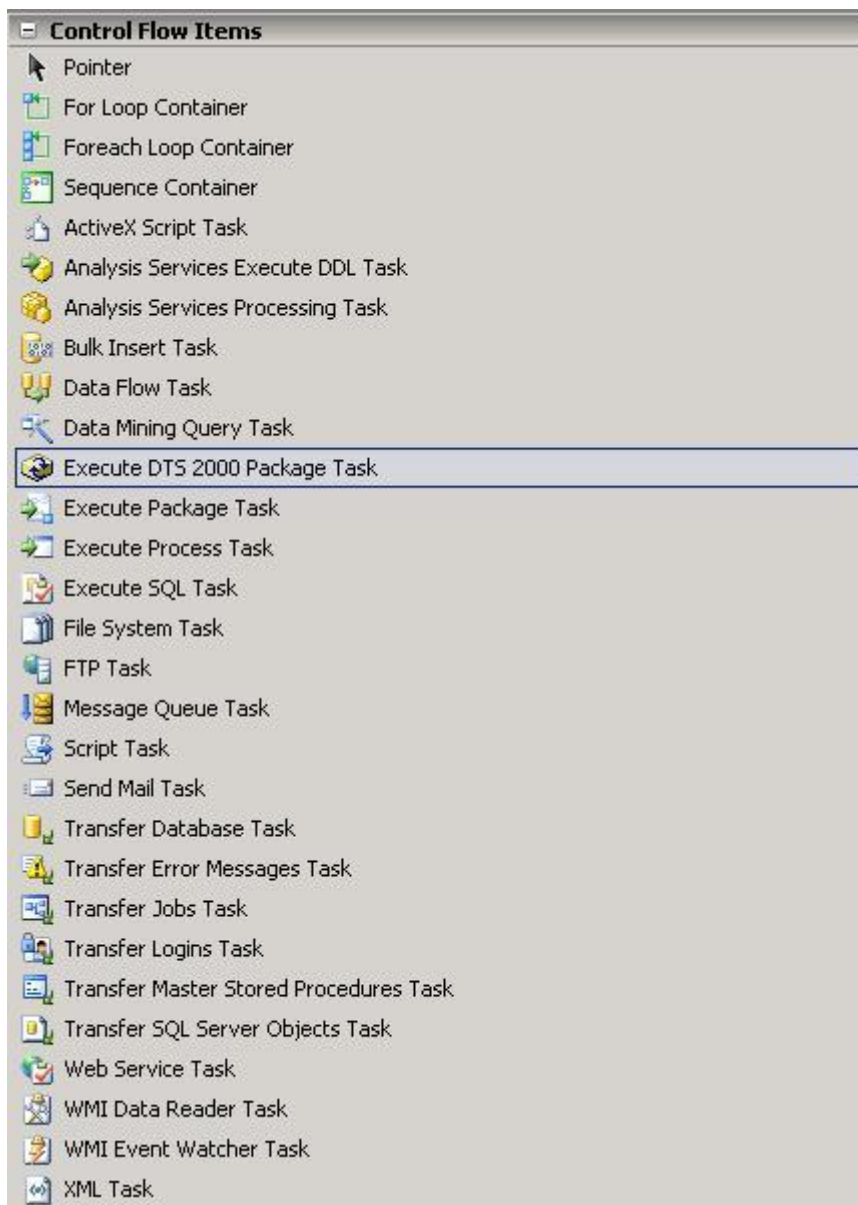
103 pav. „Maintenance Plan tasks“ elementai

Kita elementų grupė - „*Control Flow Items*“. Elementai reikalingi užduoties rezultatui pasiekti, jie pateikti 104 paveiksle.

Dažniausiai naudojamų valdymo elementų projektuose aprašymai:

- For Loop Container – ciklas vyksta tol, kol sąlygos reikšmė lygi *false*. Visi veiksmai sudėti į ciklą taip pat kartojasi tol, kol išeinama iš jo.
- For Each Loop Container – šis elementas leidžia vykdyti ciklą su įvairiais rinkiniais, įtraukiant dokumentus, XML, SSIS kintamuosius.
- Sequence Container - organizuoja užduočių rinkinį konteineryje. Užduotys vykdomos nuosekliai.
- Execute SQL Task – elementas vykdo SQL užklausas, saugomas procedūras ir SQL dokumentus.
- Execute Package Task - elementas gali vykdyti kitą projektą (paketą).
- Bulk Insert Task – elementas greitai įdeda tekstinį dokumentą į duomenų bazės lentelę, tačiau nėra galimybės kontroliuoti duomenų gavybą.
- File System Task – elementas turi galimybę vykdyti įvairias sistemos operacijas.
- FTP Task – elementas leidžia siųsti ir gauti dokumentus per FTP. Jungties (connection) valdymas reikalauja nustatyti FTP serverį.
- Execute Package Task – elementas vykdo vaikinį paketą leidžiamo projekto kontekste.
- Script Task – elementas skirtas pakeisti *ActiveX task* elementą, naudojantį VB.NET scenarijų.
- Send Mail Task – elementas siunčia elektroninį laišką per SMTP.

- XML Task – elementas susijęs su operacijomis, kurias galima atlikti su XML dokumentais.

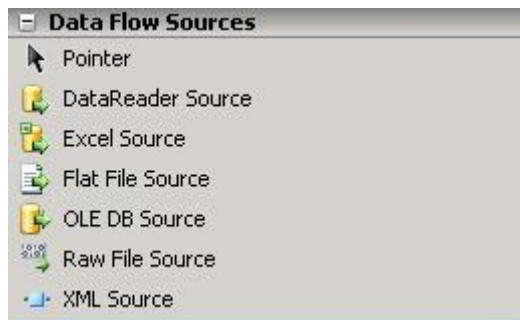


106 pav. Control Flow Items elementai

Data Flow valdymo skydelis

Atliekamas duomenų perkėlimas iš duomenų šaltinio į galutinę duomenų paskirties vietą. Elementai suskirstyti į tris grupes: duomenų išgavimo (105 pav.), duomenų transformavimo (106 pav.), duomenų patalpavimo galutinėje saugykloje (107 pav.).

Duomenys gali būti paimti iš dokumento esančio Excel formate, elementariame tekstiniame, XML faile ar kitame duomenų serveryje. Taip pat yra galimybė duomenis turėti SSIS paketų sistemoje, lokaliai ir pasinaudojus elementu *Raw File Source*, perkelti duomenis į kitur esančią duomenų saugyklą.

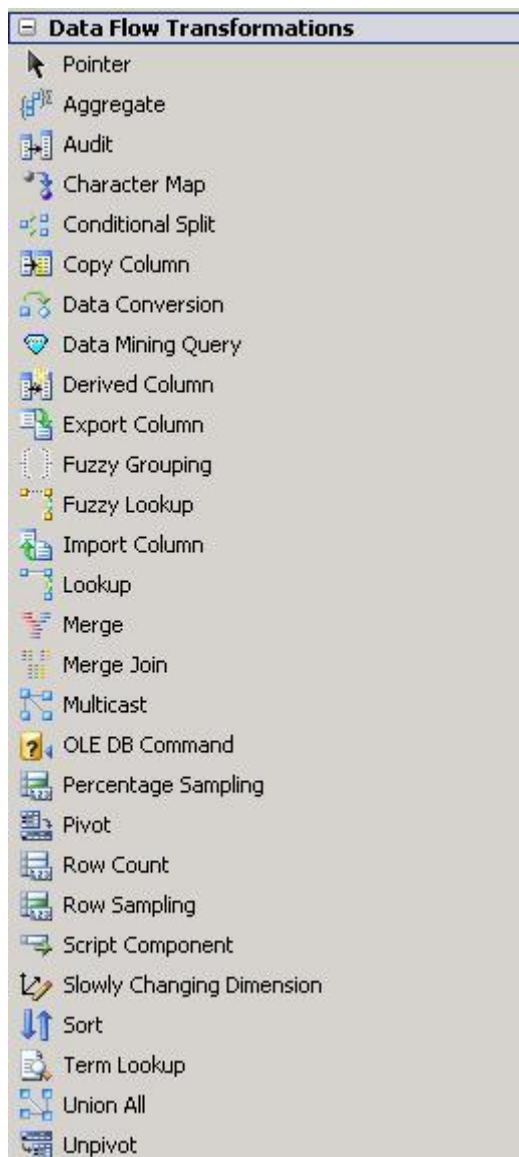


107 pav. Duomenų išgavimo elementai

Duomenų transformavimui ir konvertavimui yra sukurta daug elementų SSIS paketo aplinkoje lyginat su DTS paketu. Žemiau pateiktas duomenų transformavimo elementų sąrašas ir jų aprašymai.

- Aggregate – elementas teikia vidurkio skaičiavimo, sumavimo, minimumo/maksimumo išrinkimo, grupavimo funkcijas.
- Audit – elementas leidžia į duomenų srautą įtraukti duomenis iš aplinkos, kurioje vykdomas paketas. Pavyzdžiui, SSIS paketo, kompiuterio ar operatoriaus vardas, gali būti įkeltas į srautą. *MS SQL Server 2005* SSIS įtraukia sistemos kintamuosius, kuriuose ši sistema egzistuoja.
- Character Map – elementas teikia tik funkcijas susijusias su simboliais, tokias kaip mažųjų raidžių pavertimas į didžiąsias. Elementas vykdomas jei duomenų stulpelis yra „string“ tipo.
- Conditional Split – elementas gali maršrutizuoti duomenų eilutes skirtingiems išėjimams priklausomai nuo duomenų turinio. Tai atitinka CASE sąlygą programavime.
- Copy Column - Elementas sukuria naujus stulpelius, kurie skirti nukopijuotiems stulpeliams iš įeinančio duomenų srauto. Nukopijuotam stulpeliui galima pritaikyti įvairias funkcijas, tokias kaip agregavimo, konvertavimo. Ši transformacija turi vieną įėjimą ir vieną išėjimą.
- Data Conversation - Elementas konvertuoja duomenis iš įeinančio stulpelio į kitą duomenų tipą ir tada kopijuoja rezultata į naują išėjimo stulpelį. Pavyzdžiui, gauti duomenys iš įvairių duomenų šaltinių gali neatitikti duomenų tipų galutinėje duomenų saugykloje.
- Data Mining Query - Elementas vykdo prognozės užklausas. Ši transformacija turi užklauskos kūrimo mechanizmą, kuris leidžia kurti duomenų valdymo plėtinių užklausas. Vienas elementas gali paleisti daug prognozinių užklauskų, jei modeliai sukurti ant tos pačios duomenų valdymo struktūros.
- Derived Column - transformavimo grupės elementas skirtas kombinuotai reikšmei gauti. Reikšmė paimta iš duomenų srauto atitinkamo stulpelio gali įgyti kitą, kombinuotą reikšmę sudarytą iš sistemos kintamųjų, operacijų, funkcijų. Rezultatai gali būti patalpinti automatiškai sukurtame naujame stulpelyje arba egzistuojančios reikšmės stulpelyje.

- Export Column - Elementas skaito duomenis iš duomenų srauto ir surašo juos į failą. Pavyzdžiui, jei duomenų srautas talpina informaciją apie produktus, kurių kiekvienas turi aprašymą ir paveiksluką, tai pasinaudojus šiuo elementu paveikslus galima išsaugoti failuose.
- Fuzzy Grouping - Elementas vykdo duomenų tvarkymo užduotis identifikudamas duomenų eilutes, kurios gali būti sudubliuotos. Transformavimo grupės elementas reikalauja, kad būtų prisijungta prie *MS SQL Server 2005* DB ir sukurtos laikinos lentelės.
- Fuzzy LookUp - Elementas vykdo duomenų tvarkymo užduotis tokias kaip, duomenų standartizavimas, duomenų teisingumas, dingusių reikšmių pateikimas. Šis elementas skiriasi nuo "LookUp" elemento tuo, kad naudoja neapibrėžtą duomenų atitikimą.
- Import Column - Elementas skaito duomenis iš failų ir deda juos į stulpelius duomenų sraute. Naudojant šią transformaciją paketas gali pridėti tekstą ir atskiruose failuose saugojamus paveikslus į duomenų srautą.
- LookUp - Elementas lygina duomenis tarp šaltinio ir nurodyto duomenų sąrašo. Duomenų sąrašas gali būti egzistuojanti lentelė ar poschemė. Elementas naudoja OLE DB jungtį į duomenų bazę, kur talpinami duomenys.
- Merge - Elementas jungia du parūšiuotus duomenų sąrašus į vieną.
- Merge Join - Elementas jungia du surūšiuotus duomenų sąrašus *FULL*, *LEFT* arba *INNER* jungimu ir grąžina atfiltruotus duomenis.



108 pav. Duomenų transformavimo elementai

Galutinė duomenų saugykla gali būti paprastas tekstinis failas, *MS Excel* dokumentas, duomenų bazė. Visi elementai, kurie gali būti panaudoti saugoti duomenims pateikti 107 paveiksle.



109 pav. Duomenų paskirties vietos tipai

Event Handlers valdymo skydelis

Kiekvienas vykdomas elementas grąžina atitinkamą baigties įvykį. Šie įvykių valdymo elementai yra kuriami siekiant didesnio proceso valdymo. Atitinkamas baigties įvykis gali turėti atitinkamą savo veiksmų seką, kuri vykdoma, kai elementas baigia savo pagrindinę darbų seką. *Event Handlers* gali vykdyti tokias užduotis:

- Išvalyti laikinas duomenų saugyklas, kai paketas ar elementas baigia vykdytis.
- Atkurti sistemos informaciją prisijungus prie šaltinio prieš SSIS paketo vykdymą.
- Atnaujinti lentelėje esančius duomenis kuomet palyginimas su nuorodų lentele nepavyksta.
- Siųsti pranešimą kai įvyksta klaida, gaunamas įspėjimas ar užduotis nepavyksta.

Jei įvykis neturi *Event Handler*, tuomet pasibaigus vienai užduočiai - pereinama prie kitos.

Event Handler sukūrimas yra panašus į elemento kūrimą. Jis turi užduotis ir konteinerius, kurie sudedami į seką valdymo sraute.

Event Handler visos elemento baigtys pateiktos 28 lentelėje

28 lentelė. *Event Handler* elementų sąrašas

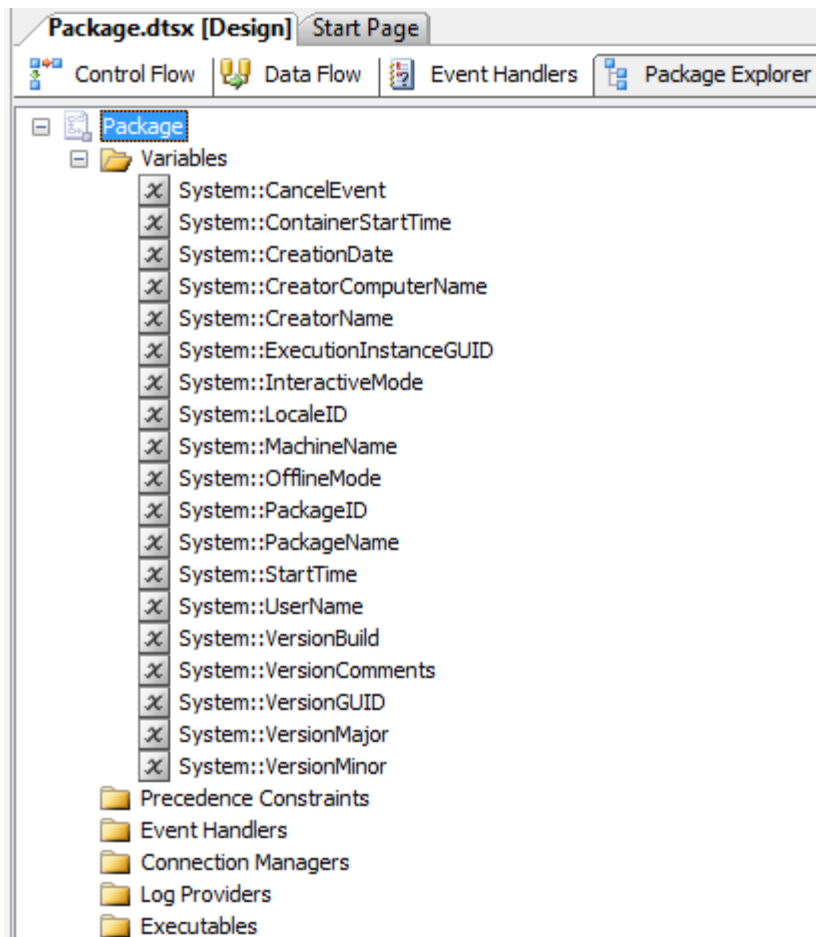
Įvykio baigtis	Aprašymas
OnError	Įvykis yra suaktyvinamas, kai elemento vykdomosi metu įvyksta klaida.
OnExecStatusChanged	Įvykis yra suaktyvinamas, kai pasibaigęs vykdytis elementas pakeičia būsenos statusą.
OnInformation	Įvykis yra suaktyvinamas, kai užduotis vykdoma, validuojama. Visa informacija gauta tuo metu įrašoma ir pateikiama vartotojui. Ataskaitoje nėra informacijos apie įspėjimus, klaidas.

OnPostExecute	Įvykis suaktyvinamas, kai užduotis įvykdyta.
OnPostValidate	Įvykis suaktyvinamas, kai užduoties patikra įvykdyta.
OnPreExecute	Įvykis suaktyvinamas prieš užduoties vykdymąsi.
OnPreValidate	Įvykis suaktyvinamas prieš užduoties validacijos vykdymąsi.
OnProgress	Įvykis suaktyvinamas, kai užduoties vykdymosi metu įvyksta žymus progresas.
OnQueryCancel	Įvykis suaktyvinamas, kai vykdoma užduotis, jis nurodo kada užduotis turi sustoti vykdytis.
OnTaskFailed	Įvykis suaktyvinamas, kai vykdoma užduotis nepavyksta.
OnVariableValueChanged	Įvykis suaktyvinamas, kai užduotyje esančio kintamojo reikšmė pasikeičia. Šis įvykis nevykdomas, jeigu nustatymuose nustatyta kintamajam „False“.
OnWarning	Įvykis suaktyvinamas, kai užduoties vykdyme atsiranda įspėjimų.

Package Explorer valdymo skydelis

Šioje paketo valdymo dalyje yra visi objektai, kurie buvo panaudoti kuriant procesą. Tai medžio tipo hierarchija, kurioje yra kintamieji – sisteminiai, vartotojo sukurti, duomenų bazių panaudotos jungtys, visi proceso elementai, vykdymosi laikas.

Proceso priežiūra gali būti atliekama tik per šią skiltį. 108 paveiksle pateiktas *Package Explorer* hierarchinis medis.



110 pav. *Package Explorer* navigacijos medis

SSIS kintamieji leidžia sukurti paketą, kuris gali būti konfigūruojamas jo vykdymo metu. Išraiškos (angl. *expressions*) – tai formulės, įvairūs pažymėjimai. Galima sukurti adresatų sąrašą laiškams siųsti. Sukuriame kintamąjį (109 pav.), surašome reikalingų asmenų elektroninius paštus, priskiriame išraiškai ir įdedame į konfigūracinį failą. Pasikeitus adresatų sąrašui, pataisomas išraiškos (110 pav.) tekstas konfigūraciniame faile.

Kintamieji (angl. *variables*) yra dviejų tipų: sukurti vartotojo ir sisteminiai. Sisteminio kintamojo savybių vartotojui keisti negalima. 109 paveiksle pavaizduotas vartotojo sukurtas kintamasis. Pateiktos jo savybės (angl. *properties*) [6].

Name	Scope	Data Type	Value
SendTo	tmp	String	indre
SendEmailTo	tmp	String	indre.kun...
EmailSendTo	tmp	String	indre.kun...

Properties

SendEmailTo Variable

Misc

CreationName

Description

EvaluateAsExpression **False**

Expression

ID {858EEAD8-634A-494D-AF56-E81

Name **SendEmailTo**

Namespace **User**

QualifiedName User::SendEmailTo

RaiseChangedEvent **False**

ReadOnly **False**

Scope tmp

SystemVariable False

Value

Value **indre.kungyte@gnt.lt**

ValueType **String**

111 pav. Vartotojo sukurtas kintamasis ir jo parametrai

Send Mail Task Editor

Configure the properties for the e-mail message sent by the SSIS package.

General

Mail

Expressions

Misc

Expressions

MessageSource "Package: " + @[System::PackageName]...

Subject "SSIS error on " + @[System::CreatorComputa...

ToLine @[User::SendEmailTo]

MessageSource

OK Cancel Help

Expression Builder

Specify the expression for the property: MessageSource.

Variables

- System::ContainerStartTime
- System::CreationName
- System::LocaleID
- System::TaskID
- System::TaskName
- System::TaskTransactionOption
- System::Cancel
- System::ErrorCode
- System::ErrorDescription
- System::EventHandlerStartTime
- System::Propagate
- System::SourceDescription
- System::SourceID

Mathematical Functions

String Functions

Date/Time Functions

NULL Functions

Type Casts

Operators

Expression:

"Package: " + @[System::PackageName] + "\n" + "Task: " + @[System::SourceName] + "\n" + "Error: " + @[System::ErrorDescription] + "\n" + "Creation Name: " + @[System::CreationName] + "\n" + "Source Description: " + @[System::SourceDescription] + "\n" + "Creator name: " + @[System::CreatorName] + "\n" + "Machine name: " + @[System::MachineName] + "\n"

Evaluated value:

Evaluate Expression OK Cancel

112 pav. Išraiškos kūrimo langai

10.3 3 Priedas. Insert_Master_DBM MS SQL užklausa

Insert_Master_DBM MS SQL užklausa

```
TRUNCATE TABLE TESTLATVIA.DBO.ITEM_MASTER_DBM
INSERT INTO [TESTLATVIA].[dbo].[ITEM_MASTER_DBM]

SELECT XX.PRODUCT_CODE AS PRODUCT_CODE,
MAX(MANUFACTURER_CODE_FI) AS MANUFACTURER_CODE_FI,
MAX(MANUFACTURER_CODE_EE) AS MANUFACTURER_CODE_EE,
MAX(MANUFACTURER_CODE_LT) AS MANUFACTURER_CODE_LT,
MAX(MANUFACTURER_CODE_LV) AS MANUFACTURER_CODE_LV,
MAX(MANUFACTURER_CODE_NO) AS MANUFACTURER_CODE_NO,

MAX(PRODUCT_NAME) AS PRODUCT_NAME,
max(MANUFACTURER) AS MANUFACTURER,
MAX(PRODUCT_GROUP) AS PRODUCT_GROUP,

MAX(PRODUCT_MANAGER_ID_FI) AS PRODUCT_MANAGER_ID_FI,
MAX(PRODUCT_MANAGER_ID_EE) AS PRODUCT_MANAGER_ID_EE,
MAX(PRODUCT_MANAGER_ID_LT) AS PRODUCT_MANAGER_ID_LT,
MAX(PRODUCT_MANAGER_ID_LV) AS PRODUCT_MANAGER_ID_LV,
MAX(PRODUCT_MANAGER_ID_NO) AS PRODUCT_MANAGER_ID_NO,

MAX(BUSINESS_UNIT_MANAGER_ID_FI) AS BUSINESS_UNIT_MANAGER_ID_FI,
MAX(BUSINESS_UNIT_MANAGER_ID_EE) AS BUSINESS_UNIT_MANAGER_ID_EE,
MAX(BUSINESS_UNIT_MANAGER_ID_LT) AS BUSINESS_UNIT_MANAGER_ID_LT,
MAX(BUSINESS_UNIT_MANAGER_ID_LV) AS BUSINESS_UNIT_MANAGER_ID_LV,
MAX(BUSINESS_UNIT_MANAGER_ID_NO) AS BUSINESS_UNIT_MANAGER_ID_NO,

MAX(DEFAULT_SUPPLIER_ID_FI) AS DEFAULT_SUPPLIER_ID_FI,
MAX(DEFAULT_SUPPLIER_ID_EE) AS DEFAULT_SUPPLIER_ID_EE,
MAX(DEFAULT_SUPPLIER_ID_LT) AS DEFAULT_SUPPLIER_ID_LT,
MAX(DEFAULT_SUPPLIER_ID_LV) AS DEFAULT_SUPPLIER_ID_LV,
MAX(DEFAULT_SUPPLIER_ID_NO) AS DEFAULT_SUPPLIER_ID_NO,

MAX(ACTIVITY_FI) AS ACTIVITY_FI,MAX(ACTIVITY_EE) AS ACTIVITY_EE,
MAX(ACTIVITY_LT) AS ACTIVITY_LT,MAX(ACTIVITY_LV) AS ACTIVITY_LV,
MAX(ACTIVITY_NO) AS ACTIVITY_NO

FROM (SELECT IDFI.PRODUCT_CODE AS PRODUCT_CODE,
IDFI.MANUFACTURER_CODE AS MANUFACTURER_CODE_FI,
" AS MANUFACTURER_CODE_EE," AS MANUFACTURER_CODE_LT,
" AS MANUFACTURER_CODE_LV," AS MANUFACTURER_CODE_NO,
ISNULL(IDFI.PRODUCT_NAME,") AS PRODUCT_NAME,
ISNULL(IDFI.MANUFACTURER,") AS MANUFACTURER,
ISNULL(IDFI.PRODUCT_GROUP,") AS PRODUCT_GROUP,
ISNULL(IDFI.PRODUCT_MANAGER_ID,0) AS PRODUCT_MANAGER_ID_FI,
0 AS PRODUCT_MANAGER_ID_EE,0 AS PRODUCT_MANAGER_ID_LT,
0 AS PRODUCT_MANAGER_ID_LV,0 AS PRODUCT_MANAGER_ID_NO,

ISNULL(IDFI.BUSINESS_UNIT_MANAGER_ID,0) AS BUSINESS_UNIT_MANAGER_ID_FI,
0 AS BUSINESS_UNIT_MANAGER_ID_EE,0 AS BUSINESS_UNIT_MANAGER_ID_LT,
0 AS BUSINESS_UNIT_MANAGER_ID_LV,0 AS BUSINESS_UNIT_MANAGER_ID_NO,

ISNULL(IDFI.DEFAULT_SUPPLIER_ID,0) AS DEFAULT_SUPPLIER_ID_FI,
0 AS DEFAULT_SUPPLIER_ID_EE,0 AS DEFAULT_SUPPLIER_ID_LT,
0 AS DEFAULT_SUPPLIER_ID_LV,0 AS DEFAULT_SUPPLIER_ID_NO,

CASE WHEN IDFI.MANUFACTURER_CODE IS NOT NULL THEN 1 ELSE 0 END AS ACTIVITY_FI,
0 AS ACTIVITY_EE,0 AS ACTIVITY_LT,0 AS ACTIVITY_LV,0 AS ACTIVITY_NO
```

FROM TESTLATVIA.dbo.ITEM_DBM_FI IDFI (NOLOCK)

UNION ALL

```
select IDEE.PRODUCT_CODE AS PRODUCT_CODE,
" AS MANUFACTURER_CODE_FI,
IDEE.MANUFACTURER_CODE AS MANUFACTURER_CODE_EE,
" AS MANUFACTURER_CODE_LT," AS MANUFACTURER_CODE_LV,
" AS MANUFACTURER_CODE_NO,
ISNULL(IDEE.PRODUCT_NAME," AS PRODUCT_NAME,
ISNULL(IDEE.MANUFACTURER," AS MANUFACTURER,
ISNULL(IDEE.PRODUCT_GROUP," AS PRODUCT_GROUP,
0 AS PRODUCT_MANAGER_ID_FI,
ISNULL(IDEE.PRODUCT_MANAGER_ID,0) AS PRODUCT_MANAGER_ID_EE,
0 AS PRODUCT_MANAGER_ID_LT,0 AS PRODUCT_MANAGER_ID_LV,
0 AS PRODUCT_MANAGER_ID_NO,

0 AS BUSINESS_UNIT_MANAGER_ID_FI,
ISNULL(IDEE.BUSINESS_UNIT_MANAGER_ID,0) AS BUSINESS_UNIT_MANAGER_ID_EE,
0 AS BUSINESS_UNIT_MANAGER_ID_LT,0 AS BUSINESS_UNIT_MANAGER_ID_LV,
0 AS BUSINESS_UNIT_MANAGER_ID_NO,

0 AS DEFAULT_SUPPLIER_ID_FI,
ISNULL(IDEE.DEFAULT_SUPPLIER_ID,0) AS DEFAULT_SUPPLIER_ID_EE,
0 AS DEFAULT_SUPPLIER_ID_LT,0 AS DEFAULT_SUPPLIER_ID_LV,
0 AS DEFAULT_SUPPLIER_ID_NO,

0 AS ACTIVITY_FI,
CASE WHEN IDEE.MANUFACTURER_CODE IS NOT NULL THEN 1 ELSE 0 END AS ACTIVITY_EE,
0 AS ACTIVITY_LT,0 AS ACTIVITY_LV,0 AS ACTIVITY_NO
```

FROM TESTLATVIA.dbo.ITEM_DBM_EE IDEE (NOLOCK)

UNION ALL

```
select IDLT.PRODUCT_CODE COLLATE Finnish_Swedish_CI_AS AS PRODUCT_CODE,
" AS MANUFACTURER_CODE_FI," AS MANUFACTURER_CODE_EE,
IDLT.MANUFACTURER_CODE AS MANUFACTURER_CODE_LT,
" AS MANUFACTURER_CODE_LV," AS MANUFACTURER_CODE_NO,
ISNULL(IDLT.PRODUCT_NAME COLLATE Finnish_Swedish_CI_AS," AS PRODUCT_NAME,
ISNULL(IDLT.MANUFACTURER COLLATE Finnish_Swedish_CI_AS," AS MANUFACTURER,
SNULL(IDLT.PRODUCT_GROUP COLLATE Finnish_Swedish_CI_AS," AS PRODUCT_GROUP,
0 AS PRODUCT_MANAGER_ID_FI,0 AS PRODUCT_MANAGER_ID_EE,
ISNULL(IDLT.PRODUCT_MANAGER_ID,0) AS PRODUCT_MANAGER_ID_LT,
0 AS PRODUCT_MANAGER_ID_LV,0 AS PRODUCT_MANAGER_ID_NO,

0 AS BUSINESS_UNIT_MANAGER_ID_FI,0 AS BUSINESS_UNIT_MANAGER_ID_EE,
ISNULL(IDLT.BUSINESS_UNIT_MANAGER_ID,0) AS BUSINESS_UNIT_MANAGER_ID_LT,
0 AS BUSINESS_UNIT_MANAGER_ID_LV,0 AS BUSINESS_UNIT_MANAGER_ID_NO,

0 AS DEFAULT_SUPPLIER_ID_FI,0 AS DEFAULT_SUPPLIER_ID_EE,
ISNULL(IDLT.DEFAULT_SUPPLIER_ID,0) AS DEFAULT_SUPPLIER_ID_LT,
0 AS DEFAULT_SUPPLIER_ID_LV,0 AS DEFAULT_SUPPLIER_ID_NO,

0 AS ACTIVITY_FI,0 AS ACTIVITY_EE,
CASE WHEN IDLT.MANUFACTURER_CODE IS NOT NULL THEN 1 ELSE 0 END AS ACTIVITY_LT,
0 AS ACTIVITY_LV,0 AS ACTIVITY_NO
```

FROM TESTLATVIA.dbo.ITEM_DBM_LT IDLT (NOLOCK)

UNION ALL

```
select IDLV.PRODUCT_CODE COLLATE Finnish_Swedish_CI_AS AS PRODUCT_CODE,
" AS MANUFACTURER_CODE_FI," AS MANUFACTURER_CODE_EE,
" AS MANUFACTURER_CODE_LT,
IDLV.MANUFACTURER_CODE AS MANUFACTURER_CODE_LV,
" AS MANUFACTURER_CODE_NO,
ISNULL(IDLV.PRODUCT_NAME COLLATE Finnish_Swedish_CI_AS," AS PRODUCT_NAME,
ISNULL(IDLV.MANUFACTURER COLLATE Finnish_Swedish_CI_AS," AS MANUFACTURER,
ISNULL(IDLV.PRODUCT_GROUP COLLATE Finnish_Swedish_CI_AS," AS PRODUCT_GROUP,
```

```

0 AS PRODUCT_MANAGER_ID_FI,0 AS PRODUCT_MANAGER_ID_EE,
0 AS PRODUCT_MANAGER_ID_LT,
ISNULL(IDLV.PRODUCT_MANAGER_ID,0)AS PRODUCT_MANAGER_ID_LV,
0 AS PRODUCT_MANAGER_ID_NO,

0 AS BUSINESS_UNIT_MANAGER_ID_FI,0 AS BUSINESS_UNIT_MANAGER_ID_EE,
0 AS BUSINESS_UNIT_MANAGER_ID_LT,
ISNULL(IDLV.BUSINESS_UNIT_MANAGER_ID,0) AS BUSINESS_UNIT_MANAGER_ID_LV,
0 AS BUSINESS_UNIT_MANAGER_ID_NO,

0 AS DEFAULT_SUPPLIER_ID_FI,0 AS DEFAULT_SUPPLIER_ID_EE,
0 AS DEFAULT_SUPPLIER_ID_LT,
ISNULL(IDLV.DEFAULT_SUPPLIER_ID,0) AS DEFAULT_SUPPLIER_ID_LV,
0 AS DEFAULT_SUPPLIER_ID_NO,

0 AS ACTIVITY_FI,0 AS ACTIVITY_EE,0 AS ACTIVITY_LT,
CASE WHEN IDLV.MANUFACTURER_CODE IS NOT NULL THEN 1 ELSE 0 END AS ACTIVITY_LV,
0 AS ACTIVITY_NO

```

```
FROM TESTLATVIA.dbo.ITEM_DBM_LV IDLV (NOLOCK)
```

```
UNION ALL
```

```

select IDNO.PRODUCT_CODE COLLATE Finnish_Swedish_CI_AS AS PRODUCT_CODE,
" AS MANUFACTURER_CODE_FI," AS MANUFACTURER_CODE_EE,
" AS MANUFACTURER_CODE_LT," AS MANUFACTURER_CODE_LV,
IDNO.MANUFACTURER_CODE AS MANUFACTURER_CODE_NO,
ISNULL(IDNO.PRODUCT_NAME COLLATE Finnish_Swedish_CI_AS,") AS PRODUCT_NAME,
ISNULL(IDNO.MANUFACTURER COLLATE Finnish_Swedish_CI_AS,") AS MANUFACTURER,
ISNULL(IDNO.PRODUCT_GROUP COLLATE Finnish_Swedish_CI_AS,") AS PRODUCT_GROUP,
0 AS PRODUCT_MANAGER_ID_FI,0 AS PRODUCT_MANAGER_ID_EE,
0 AS PRODUCT_MANAGER_ID_LT,0 AS PRODUCT_MANAGER_ID_LV,
ISNULL(IDNO.PRODUCT_MANAGER_ID,0) AS PRODUCT_MANAGER_ID_NO,

```

```

0 AS BUSINESS_UNIT_MANAGER_ID_FI,0 AS BUSINESS_UNIT_MANAGER_ID_EE,
0 AS BUSINESS_UNIT_MANAGER_ID_LT,0 AS BUSINESS_UNIT_MANAGER_ID_LV,
ISNULL(IDNO.BUSINESS_UNIT_MANAGER_ID,0) AS BUSINESS_UNIT_MANAGER_ID_NO,

```

```

0 AS DEFAULT_SUPPLIER_ID_FI,0 AS DEFAULT_SUPPLIER_ID_EE,
0 AS DEFAULT_SUPPLIER_ID_LT,0 AS DEFAULT_SUPPLIER_ID_LV,
ISNULL(IDNO.DEFAULT_SUPPLIER_ID,0) AS DEFAULT_SUPPLIER_ID_NO,

```

```

0 AS ACTIVITY_FI,0 AS ACTIVITY_EE,0 AS ACTIVITY_LT,0 AS ACTIVITY_LV,
CASE WHEN IDNO.MANUFACTURER_CODE IS NOT NULL THEN 1 ELSE 0 END AS ACTIVITY_NO

```

```
FROM TESTLATVIA.dbo.ITEM_DBM_NO IDNO (NOLOCK)
```

```
) xx GROUP BY PRODUCT_CODE
```


10.1 3 Priedas. Straipsnis

Bendrinių duomenų atskyrimo šablonas ir jo taikymas UAB „GNT Lietuva“ ETL procesuose

Indrė Kungytė

Informacijos sistemų katedra

Kauno technologijos universitetas

indrose@gmail.com

vadovė Lina Nemuraitė

Informacijos sistemų katedra

Kauno technologijos universitetas

lina.nemuraite@ktu.lt

Santrauka—Straipsnyje analizuojami duomenų išgavimo iš paskirstytų duomenų bazių ir perkėlimo į centrinę saugyklą procesai ir perkeliamų duomenų struktūros, siekiant pagerinti šių procesų veikimo charakteristikas. Procesų patobulinimas vykdomas įmonės perėjimo prie naujų integravimo technologijų metu, taip gaunant didesnę reinžinerijos efektą. Pateikiamas empiriniu būdu gautas bendrinių duomenų atskyrimo šablonas, kurį galima taikyti panašiuose duomenų integravimo sprendimuose.

Raktiniai žodžiai - *ETL, duomenų integravimas, projektavimo šablonai.*

Įvadas

Greitai augančiame ir besikeičiančiame pasaulyje atsiranda poreikis iš esamų duomenų greitai gauti rezultatus, be to, pasitikėjimas rezultatais yra svarbus kaip niekada anksčiau. Efektyvus duomenų integravimas tapo efektyvaus sprendimų priėmimo pagrindu.

Duomenų integravimas pasireiškia vis didėjančiu poreikiu dalintis augančiais duomenų kiekiais. Plečiantis GNT įmonei, jos padaliniai įsikuria skirtingose Europos šalyse. Augant duomenų kiekiams, atsiranda poreikis turėti juos vienoje duomenų bazėje tam, kad būtų galima atlikti analizę ir panaudoti jos rezultatus verslo sprendimams priimti. Šio darbo tikslas – greitas, efektyvus, tikslus duomenų perkėlimas, kurį siekiama atlikti taikant naujas duomenų integravimo technologijas. Šiam tikslui buvo analizuojami duomenų integravimo metodai, reinžinerijos principai bei naujų technologijų pritaikymo galimybės, atliekant įmonės duomenų integravimo posistemio reinžineriją.

Duomenų perkėlimas iš operacinių duomenų bazių į analizei skirtas saugyklas vykdomas ETL (angl. Extract, Transform, Load) procesais, kurie leidžia paimti duomenis iš įvairių šaltinių, pakeisti jų formatus, pašalinti neteisingus ar tuščius duomenis ir atlikti kitas operacijas. Šie procesai jungiami į grandines, kurių vykdymo laikas yra ilgas, todėl paprastai jie vykdomi naktį, kad iki darbo pradžios būtų galima perskaičiuoti visas vartotojams reikalingas ataskaitas.

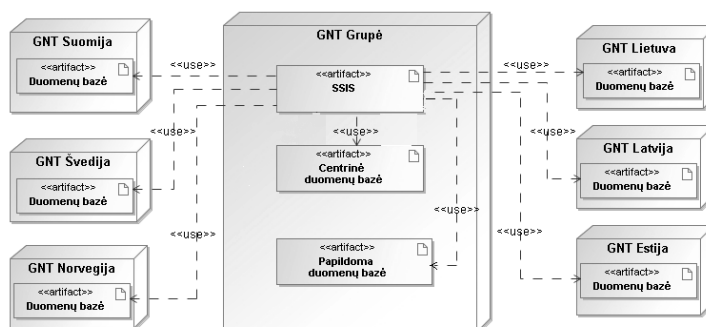
Šiame darbe buvo sprendžiami uždaviniai, kaip perkelti ETL procesus į naujai pasirinktą technologinę platformą, nenutraukiant duomenų integravimo proceso. Pereinant prie naujų technologijų, pasinaudota reinžinerijos idėja, kuri teigia, kad perėjimo prie naujos technologijos metu tikslinga pakeisti ir optimizuoti procesus, jei tai yra reikalinga. Reinžinerijos metu buvo pastebėta, kad perkeliama duomenys yra skirtingo detalumo ir juos galima atskirti. Eksperimentai, kurių metu buvo stebimos ETL procesų charakteristikos, patvirtino, kad po atskyrimo duomenys perkeliama daug greičiau. Ši galimybė pagerinti ETL procesus buvo apibendrinta ir aprašyta kaip bendrinių duomenų atskyrimo šablonas.

Duomenų integravimo posistemis

Duomenų integravimo posistemio proceso esmė – perkelti ir sujungti informaciją apie produktus, esančius skirtingose šalyse, į centrinę duomenų bazę tam, kad būtų galima atlikti verslui aktualią analizę. Siekiant užtikrinti duomenų korektiškumą, perkėlimo metu identifikuojami produktų informacijos

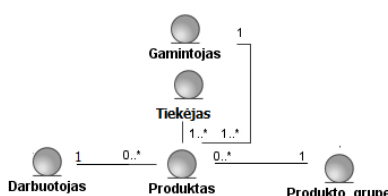
skirtumai tarp tarpinės (ORACLE) ir Centrinės DB. Integravimo procesu gauti duomenys pateikiami vartotojams ataskaitų pavidalu.

Duomenų integravimo (DI) procesas sujungia duomenis iš penkių šaltinių, kurie realizuoti MS SQL Server 2000 ir MS SQL Server 2005 duomenų bazių valdymo sistemose (DBVS). Galimų integravimo technologijų analizė parodė, kad GNT perspektyviausia pasirinkti MS SQL Server integravimo paslaugas (angl. SQL Server Integration Services) SSIS. 113 paveiksle pateikta UAB „GNT“ duomenų integravimo posistemio architektūra.



113 pav. GNT duomenų integravimo sistemos architektūra

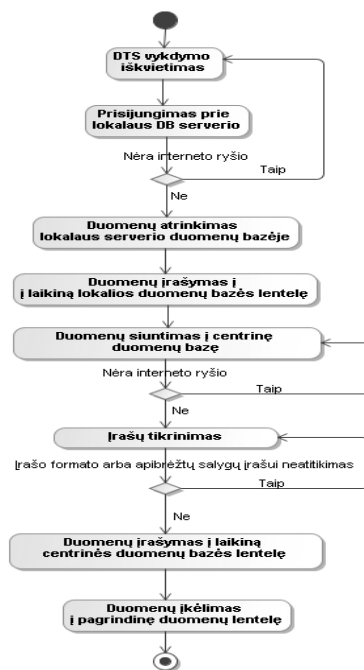
Duomenys registruojami lokaliuose duomenų bazėse, kurių lentelių struktūra yra vienoda. Duomenų sąrašas generuojamas užklausa, kuri formuojama iš penkių lentelių: „Produktas“, „Gamintojas“, „Produkto grupė“, „Darbuotojas“, „Tiekėjas“ (114 pav.).



114 pav. Perkeliamų duomenų esybių klasių modelis

Duomenų perkėlimo (etl) procesai

ETL yra platforma, leidžianti išgauti, transformuoti ir perkelti duomenis arba jų kopijas iš daugialypių platformų ir duomenų šaltinių į vieną ar daugiau saugyklų. Naudingiausia perkelti tik tuos duomenis, kurie reikalingi sistemos funkcionalumui. Papildomi, nereikalingi, duomenys užima vietą diske ir atmintį [4–6]. 115 paveiksle pavaizduoti duomenų integravimo proceso *ETL* elementai.



115 pav. Tipinis duomenų perkėlimo procesas

Skirtingo detalumo duomenų integravimo šablonas

Įmonėje atliktos duomenų integravimo procesų analizės metu pastebėta, kad kiekviena perkeliama lentelė turi bendrinę ir detalią informaciją. Pagrindinė informacija yra unikali viename padalinyje, bet atsikartoja kituose padaliniuose. Detali informacija yra unikali tiek vieno įmonės padalinio, tiek visos įmonės atžvilgiu.

Šių empirinių pastebėjimų rezultate buvo sudarytas skirtingo detalumo duomenų atskyrimo šablonas. Remiantis šablonų sudarymo principais, šablonas turi būti aiškus, tikslus, išsamus, suprantamas programuotojui, naudoti atitinkamos srities sąvokas ir būti nedviprasmiškas [1–3]. Tiriamos įmonės atveju šablonas privalo būti lankstus ir gebėti lengvai įtraukti į jau veikiančią duomenų perkėlimo procesą naują duomenų šaltinį. Tai tikslinga dėl tiriamos įmonės plėtimosi į kitas šalis. Naujo duomenų šaltinio įtraukimas turi būti paprastas, vykti be trikdžių, nereikalauti bendradarbiaujančių objektų koregavimo, taip pat turi būti galimybė lengvai papildyti perkėlimo procesą naujais etapais, kurie padėtų spręsti kitokio tipo problemas.

Bendrinių ir detalių duomenų atskyrimo šablonas

Šiame skyriuje pateikiamas sukurto šablono aprašymas, sudarytas pagal literatūroje aprašytus šablonų struktūros principus [2].

Tikslas. Duomenų integravimo procesą padaryti lankstų naujo duomenų šaltinio įtraukimo ir pašalinimo atžvilgiu, sumažinti integravimo procesų vykdymo laiką.

Problema. Komplikuotas naujo duomenų šaltinio įtraukimas į duomenų integravimo procesą. Naujo šaltinio atsiradimas reikalauja daugelio susietų objektų modifikavimo. Ilga perkėlimo vykdymo

trukmė vėlina duomenų perkėlimą į galutinę saugyklos vietą. Tai sutrikdo sinchroninį duomenų užkrovimą, dėl kurio gaunami nekorektiški duomenys.

Problema atsiranda dėl netobulos perkeliamų duomenų lentelės struktūros. Kiekvienas įrašas turi tiek skirtingų laukų, kiek yra duomenų šaltinių. Atsiradus naujam duomenų šaltiniui, lentelę reikia papildyti naujais laukais ir sukurti naują perkėlimo procesą. Sudėtinga ir atvirkštinė situacija, kai reikia naikinti egzistuojantį šaltinį ir pašalinti iš faktinės lentelės atitinkamus laukus.

Sprendimas. Problema išsprendžiama perorganizavus galutinės lentelės struktūrą. Informaciją, gautą iš duomenų šaltinių, reikia suskirstyti į dvi dalis: pagrindinę ir detalią. Sugrupuota informacija saugoma skirtingose lentelėse.

Taikymas. Modelis taikomas duomenų perkėlimo procesuose, kai galima išskirti bendrinę ir detalią informaciją.

Rezultatas. Informacijos suskirstymas leidžia įtraukti ar pašalinti reikiamą duomenų šaltinių skaičių. Dėl atskyrimo sutrumpėja duomenų perkėlimo laikas.

Pavyzdys. Duomenų integravimo procesu perkeliama informacija apie produktus, esančius trijuose skirtinguose šaltiniuose. Produkto informaciją sudaro produkto kodas, produkto pavadinimas, produkto vadybininkas (116 pav.).

1 duomenų šaltinis	Produkto_kodas XXX	Produkto_pavadinimas Logitech pelė	Produkto_vadybininkas Indrė Kungytė
2 duomenų šaltinis	Produkto_kodas XXX	Produkto_pavadinimas Logitech pelė	Produkto_vadybininkas Jonas Jonaitis
3 duomenų šaltinis	Produkto_kodas XXX	Produkto_pavadinimas Logitech pelė	Produkto_vadybininkas Petras Petraitis

116 pav. Produkto informacija skirtinguose duomenų šaltiniuose

Išgauti duomenys sujungiami į vieną lentelę (117 pav.).

Produkto_kodas	Produkto_pavadinimas	LT_Produkto_vadybininkas	LV_Produkto_vadybininkas	EE_Produkto_vadybininkas
XXX	Logitech pelė	Indrė Kungytė	Jonas Jonaitis	Petras Petraitis

117 pav. Produkto informacija, įrašyta į galutinę DB lentelę

Kadangi produkto pavadinimas ir kodas visuose duomenų šaltiniuose yra vienodas, tai šiuos duomenis laikome pagrindiniais. Produkto vadybininkai yra detali informacija, nes skirtinguose padaliniuose už tą patį produktą yra atsakingi skirtingi asmenys. Ši informacija yra unikali padalinio lygyje. Įvedame papildomą padalinio požymį ir išsaugome duomenis naujose lentelėse (118 pav.).

Pagrindinė produkto informacija

	Produkto_kodas	Produkto_pavadinimas
1	XXX	Logitech pelė

Detali produkto informacija

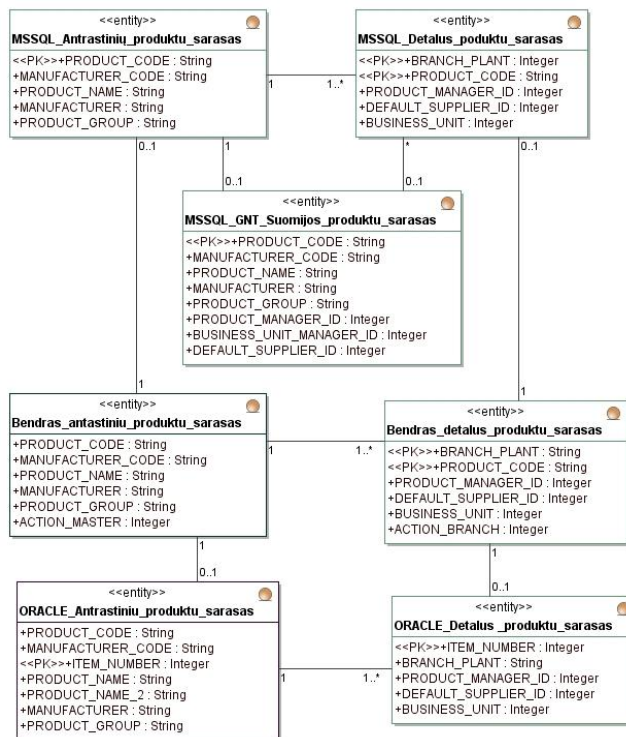
	Produkto_kodas	Padalinys	Produkto_vadybininkas
1	XXX	LT	Indre Kungyte
2	XXX	LV	Jonas Jonaitis
3	XXX	EE	Petras Petraitis

118 pav. Produkto duomenys atskirti ir įrašyti į galutines DB lenteles

Atsiradus naujam duomenų šaltiniui, į detalios produkto informacijos lentelę įsirašytų papildoma eilutė, nereikėtų DB lentelių struktūros koregavimo.

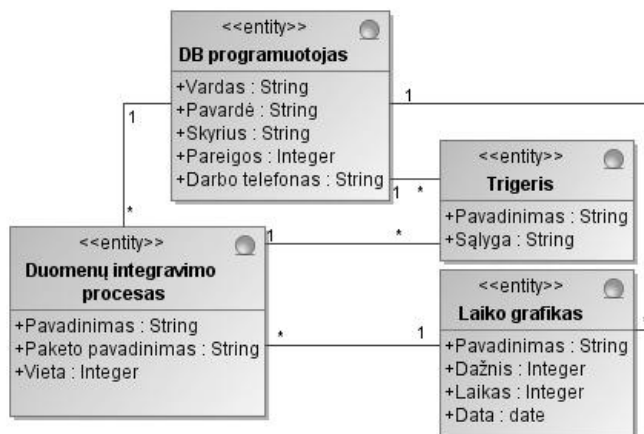
Šablono taikymas UAB „GNT Lietuva“ įmonės ETL procesuose

119 paveiksle pateikta GNT esybių klasių modelio dalis, kurioje pavaizduota bendrinė ir detali vieno iš penkių integruojamų šaltinių (Suomijos) informacija.



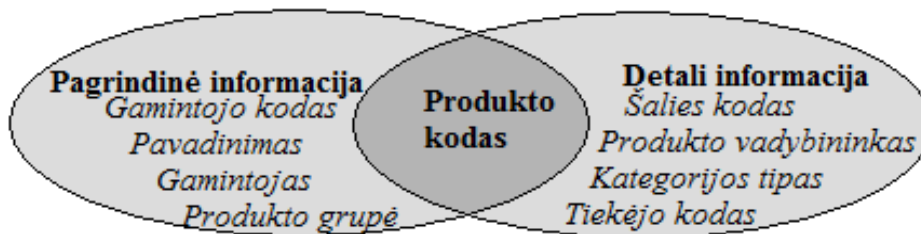
119 pav. Atskirtos bendrinės ir detalios produkto duomenų struktūros

120 paveiksle pateiktas duomenų integravimo (DI) posistemio metaduomenų modelis, kurį sudaro duomenų integravimo procesai ir jiems nustatyti laiko grafikai, kuriuos sužadina trigeriai, sukurti integravimo paslaugų SSIS įrankiu. Visi DI procesai atlieka tam tikrą duomenų perkėlimą ir apdorojimą [5]. Kiekvienas DI procesas turi griežtą laiko grafiką ir trigerį. Laiko ir įvykių agentų pagalba sudaromos DI procesų sekos, kurios užtikrina nuoseklų, taisyklingą duomenų užkrovimą į centrinę duomenų bazę. DI procesų visuma sudaro įmonės duomenų integravimo sistemą.



120 pav. Duomenų integravimo sistemos procesų metaduomenų modelis

Skirtingo detalumo duomenų integravimo šablonas taikomas duomenų perkėlimo į buferines DB lenteles etape. Išgauti duomenys suskirstomi į dvi dalis: bendrinę ir detalią. Bendrinę produktų informaciją sudaro šie laukai: produkto kodas, gamintojo kodas, produkto pavadinimas, gamintojas, produkto grupė. Ši informacija yra vienoda visuose įmonės padaliniuose. Detali informacija sudaryta iš šalies kodo, produkto kodo, produkto vadybininko kodo, tiekėjo kodo, kategorijos tipo. Šie duomenys yra skirtingi įmonės padaliniuose, nes kiekvienas padalinys turi savo darbuotojus, tiekėjus ir produktų kategorijas, į kurias patenka produktai pagal pardavimų skaičiaus ir kainos santykį. Atskirta informacija saugoma skirtingose DB lentelėse, kurios jungiamos per produkto kodą. 121 paveiksle pateikta atskirtos informacijos schema.



121 pav. tskirta bendrinės ir detalios informacijos struktūra

Bendrinės ir detalios informacijos atskyrimo šablono taikymo eksperimentas

Eksperimento tikslas yra išanalizuoti patobulintus duomenų integravimo procesus ir įsitikinti jų pagreitėjimu, įvertinant duomenų perdavimo laiką prieš ir po šablono pritaikymo.

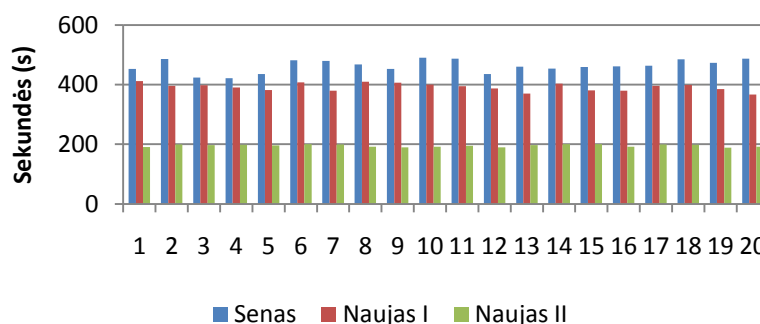
Eksperimento planas. Eksperimentas vykdomas realioje įmonės aplinkoje, jame dalyvauja duomenų integravimo sistema ir programuotojas. Analizuojami 3 nepriklausomi kintamieji – projektas 1 (senas DI procesas), projektas 2 (DI procesas, pritaikius šabloną, naujas I), ir projektas 3, kai šablonas buvo pritaikytas naujoje technologinėje SSIS aplinkoje (naujas II). Priklausomas kintamasis – duomenų perdavimo laikas, apimantis duomenų įkėlimą, sujungimą ir perskirstymą pagal duomenų detalumą.

Tikrinamos dvi nulinės ir dvi alternatyvios hipotezės „projektas 1 – projektas 2“ ir „projektas 1 – projektas 3“ procesų atžvilgiu.

Nulinės hipotezės H_0 : bendrinių ir detalių duomenų atskyrimo šablono pritaikymas nesumažino duomenų perdavimo laiko, t. y. $\mu_{\text{projektas1}} = \mu_{\text{projektas2}}$, $\mu_{\text{projektas1}} = \mu_{\text{projektas3}}$.

Alternatyvios hipotezės H_a : skirtingo detalumo duomenų integravimo šablono pritaikymas sumažino duomenų perdavimo laiką: $\mu_{\text{projektas1}} > \mu_{\text{projektas2}}$, $\mu_{\text{projektas1}} > \mu_{\text{projektas3}}$.

Eksperimento vykdymas. Atlikta 20 bandymu su kiekvienu projektu. Siekiant išvengti serverio apkrautumo ir skirtingų duomenų rinkinių įtakos rezultatams, bandymai buvo atliekami vykdant projektus tuo pačiu metu su tais pačiais duomenimis. Bandymų rezultatai sekundėmis pateikti 122 paveiksle.



122 pav. Eksperimento rezultatai

Analizė ir interpretavimas. Duomenų perdavimo laiko vidurkio, dispersijos, nuokrypio ir pasiklivimo intervalų įvertinimai \bar{x} , s^2 , s ir d apskaičiuoti pagal formules:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2, \quad s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2},$$

čia pasiklivimo lygmuo $\alpha = 0.05$, bandymų skaičius $n = 20$, *Student* kvantilis $t_{(\alpha/2, n-1)} = t_{(0,025, 19)} = 2.093$. Įvertinimų rezultatai pateikiami 1 lentelėje.

TABLE I. EKSPERIMENTE APSKAIČIUOTI ĮVERTINIMAI

vertinimas	Projektas 1 (senas)	Pr ojektas 2 (naujas I)	P rojektas 3 (naujas II)
\bar{X}	464	392	195
s^2	469,6	140,1	11,8
s	21,7	11,8	3,9
d	10,16	5,53	1,83

Eksperimento rezultatai rodo, kad įvertintas perkėlimo laiko vidurkis projekto 2 ir projekto 3 atvejais yra daug mažesnis, atitinkamai mažesni ir dispersijos, standartinio nuokrypio ir pasiklivimo intervalo įvertinimai, vadinasi, įvertinimo rezultatai yra geresni ir stabilesni.

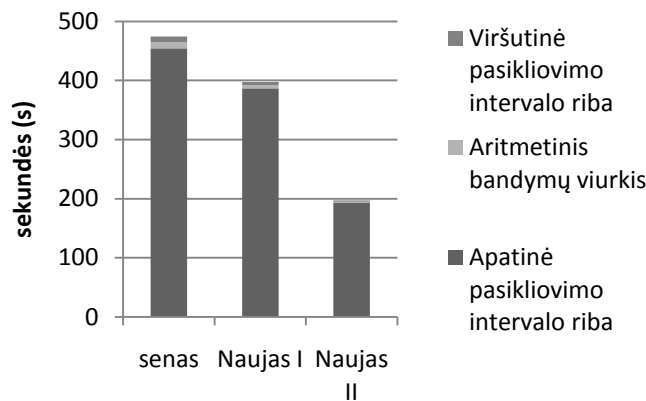
Nulinę hipotezę H_0 galima atmesti, jei $t_0 > t_{\alpha/2, n+m-2}$, čia

$$t_0 = \frac{\bar{x} - \bar{y}}{S_p \sqrt{\frac{1}{n} + \frac{1}{m}}}, S_p = \sqrt{\frac{(n-1)S_x^2 + (m-1)S_y^2}{n+m-2}}$$

S_x^2 ir S_y^2 yra atskirų bandymų variacijos.

Pirmą nulinę hipotezę galima atmesti, nes $t_0=12,7$, $t_{0,025;38}=2,024$ ir $t_0 < t_{0,025;38}$; antrą nulinę hipotezę tuo labiau galima atmesti, kadangi $t_0=68,0$; $t_{0,025;38}=2,024$ ir $t_0 < t_{0,025;38}$.

Remiantis eksperimentu galima teigti, kad bendrinių ir detalių duomenų atskyrimas pagreitino duomenų integravimo procesą 45,4%, o kartu pritaikytos *SSIS* technologijos dėka gautas 81,99% pagreitėjimas. Seno ir naujo procesų vidurkiai ir jų pasiklivimo intervalai pavaizduoti 123 paveiksle.



123 pav. Eksperimente tirtų procesų duomenų perdavimo laikų vidurkiai ir jų pasiklivimo intervalų ribos

Išvados

Atlikta GNT *ETL* procesų analizė parodė, kad perkeliamos duomenų struktūros turi pasikartojančių duomenų, kuriuos galima atskirti ir patobulinti perkėlimo procesą. Ši analizė taip pat atskleidė aktualias problemas: tikslinga sukurti tokius perkėlimo procesus ir duomenų struktūras, kad būtų lengva jas papildyti įtraukiant naujus duomenų šaltinius.

Nustatyta galimybė atskirti bendrinius ir detalius integruojamų šaltinių duomenis buvo aprašyta kaip bendrinių ir detalių duomenų atskyrimo šablonas, kurį galima taikyti panašioms duomenų integravimo sprendimams.

Atliktas eksperimentas parodė, kad šablono taikymas GNT įmonėje sutrumpino proceso trukmę 45,4%, o kartu pritaikytos *SSIS* technologijos dėka gautas 81,99% sutrumpėjimas. Duomenų integravimo procesas tapo lankstesnis, į jį bus galima paprasčiau įjungti naujus duomenų šaltinius.

Literatūra

- [1] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, S. Angel, A Pattern Language, „Christopher Alexander’s Pattern Language“, 1999, „IEEE Transactions on Professional Communication“, Vol. 42, Nr. 2, 117–122.
- [2] W. Eckerson, C. White. „Evaluating ETL and Data Integration platforms“, 101communications LLC, United States, 2003. Prieiga internete: http://mimage.hummingbird.com/alt_content/binary/pdf/collateral/mc/etlreport.pdf.
- [3] K. Hathi. „An Introduction to SQL Server 2005 Integration Services“. 2005. Prieiga internete: <http://www.microsoft.com/technet/prodtechnol/sql/2005/intro2is.mspx>
- [4] A. D. Jhingran, N. Mattos, M. Pirahesh. „Information Integration, a Research agenda“, IBM System, 2002, Nr. 4. Prieiga internete: <http://researchweb.watson.ibm.com/journal/sj/414/jhingran.pdf>
- [5] B. Mains. „Using SQL Server Integration Services (SSIS) Control Flow in SQL Server 2005“, 2007, prieiga internete: <http://dotnetslackers.com/articles/sql/UsingSQLServerIntegrationServicesSSIS.aspx>.