

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA**

Ričardas Kanaitis

**IS XML SRAUTŲ FORMAVIMAS PANAUDOJANT
KOMUNIKACINES KILPAS**

Magistro darbas

**Vadovas
doc. dr. B. Paradauskas**

KAUNAS, 2005

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA**

**TVIRTINU
Katedros vedėjas
doc. R. Butleris
2005-05-**

**IS XML SRAUTŲ FORMAVIMAS PANAUDOJANT
KOMUNIKACINES KILPAS**

Informatikos mokslo magistro baigiamasis darbas

**Kalbos konsultantė
Lietuvių kalbos katedros lektorė
dr. J. Mikelionienė
2005-05-**

**Recenzentas
2005-05-**

**Vadovas
doc. dr. B. Paradauskas
2005-05-**

**Atliko
IFM 9/1 gr. stud.
R. Kanaitis
2005-05-**

KAUNAS, 2005

SUMMARY

INFORMATION SYSTEM XML STREAM FORMATTING USING COMMUNICATION LOOPS

In changing world of computerized information systems modeling of business transactions plays an important role. Business transaction can be defined as extended communicative action loop. Using such approach, business processes and new functional requirements may be represented as choreographed sets of extended communicative action loops with established pragmatic and semantic dependencies. In general case, a computerized information system may be of hybrid nature, that is, it may consist of different versions and types of databases. XML streams' formatting in hybrid information system is discussed in this paper.

TURNYS

IŽANGA	6
1. VERSLO PROCESŲ MODELIAVIMO METODŲ APŽVALGA.....	8
1.1. Verslo procesų modeliavimo ypatumai	8
1.2. Tinklo paslaugų koncepcija kuriant programinę įrangą.....	10
1.3. EbXML veiklos procesų modeliavimo metodologija	11
1.4. Procesų modelių korektiškumas bei vientisumas	15
1.5. Analizės išvados	16
2. VERSLO TRANSAKCIJŲ MODELIAVIMAS KOMUNIKACINĖMIS KILPOMIS	17
2.1. Verslo transakcijos informacijos sistemose	17
2.2. Hibridinės duomenų bazės	19
2.3. Pragmatinės priklausomybės komunikacinių veiksmų kilpose.....	21
2.4. Išplėstųjų komunikacinių kilpų schemų sudarymas.....	24
3. XML ĮRANKIŲ APŽVALGA.....	27
3.1. XML srautų panaudojimo aktualumas	27
3.2. XML technologijos duomenų bazių valdymo sistemose	28
3.2.1. Oracle duomenų bazė ir XML panaudojimo galimybės	28
3.2.2. MS SQL Server duomenų bazė ir XML panaudojimo galimybės	29
3.2.3. Oracle ir MS SQL Server teikiamų XML savybių palyginimas	29
3.2.4. Replikavimo priemonė MS SQL serveryje	32
4. TRANSAKCIJŲ SERVERIAI	33
4.1. Transakcijų serverio apibrėžimas.....	33
4.2. Transakcijų serverių struktūra.....	35
5. KOMUNIKACINIŲ KILPŲ METODO TAIKYMAS.....	37
5.1. Verslo transakcijų realizavimas duomenų bazių transakcijomis.....	37
5.2. Dinaminės priklausomybės ir jų panaudojimas DB transakcijoms sudaryti.....	39
5.3. Akademinio posistemio funkcinių reikalavimų modeliavimas.....	40
5.3.1. Veiklos sąveikų modelis.....	40
5.3.2. Veiklos panaudojimo atvejų modelis	41
5.3.3. Procesų veiklos diagrama.....	44
5.4. Dalykinė sritis aprašyta komunikacinėmis kilpomis.....	45
IŠVADOS	48

	5
LITERATŪRA	51
1 PRIEDAS. Straipsnis “Naujų funkcinių reikalavimų įvedimas informacijos sistemose”	52

IŽANGA

Šiuo metu daugelis organizacijų naudoja elektroninio verslo sistemas („verslas verslui“, „verslas klientui“). Sparčiai kintant rinkos sąlygoms, kinta ir tokių sistemų reikalavimai. Atsiranda poreikis kurti naujas sistemas arba patobulinti esamas. Modeliavimas yra būtinas kuriant dideles, sudėtingas ar su kitomis sistemomis sąveikaujančias informacines sistemas. Modeliuojami ne tik vartotojo poreikiai, dalykinė sritis, programinė įranga, bet ir organizacijos veikla - pastaraisiais metais ši modeliavimo fazė tapo daugelio projektavimo procesų dalimi. Nors egzistuoja iširtos, stabilios veiklos procesų modeliavimo metodologijos (RUP, Provision, Popkin, Demo, Action Workflow ir kitos), tačiau informacinių technologijų plėtra nuolat suteikia galimybių tobulinti verslo procesus ir savo ruožtu kelia didesnius reikalavimus tų procesų modeliavimui. Egzistuojančių veiklos modeliavimo metodų galimybės tapo nepakankamos modeliuoti elektroninio verslo procesus, apimančius ne vieną, bet kelias bendradarbiaujančias organizacijas.

Pirmame darbo skyriuje išnagrinėta labiausiai išvystyta elektroninio verslo modeliavimo ebXML metodologija. Ji apima į kūrimą bei procesų valdymą orientuotą modeliavimą, remiasi ekonomine verslo verte ir aprašo verslo procesus moduliniu principu. Nustatyta, kad ebXML procesų modeliavimo metodologija nėra išbaigta. Naudojant ebXML procesas konstruojamas iš standartinių atominių elementų - transakcijų, bendradarbiavimo protokolų ir verslo dokumentų. Tai supaprastina proceso apibrėžimą, bet giluminės semantinės šių elementų priklausomybės lieka paslėptos.

Įvertinus šiuos ebXML metodologijos trūkumus verslo transakcijų modeliavimui siūloma naudoti komunikacinių kilpų metodą. Šis metodas detaliam išnagrinėtas antrame skyriuje. Pagrindinis komunikacinių kilpų metodo privalumas yra tas, kad šiuo metodu galima užtikrinti semantinio integralumo kriterijus. Be to naudojant pragmatiškai motyvuotas komunikacines kilpas organizacijos veiklos procesus galima aprašyti savarankiškais struktūriniais vienetais.

Realizuojant komunikacinių kilpų informacijos srautus būtina pasirinkti priemones, kurios užtikrintų duomenų apsikeitimą tarp skirtingų duomenų bazių valdymo sistemų. Tai būtina todėl, kad organizacijos naudoja skirtingas duomenų bazes. Norint integruoti skirtingas DB, visų pirma reikia naudoti bendrą, integruojamų duomenų bazių struktūras apjungiantį standartą. Šis standartas turi būti universalus, turėtų būti pritaikomas skirtingoms duomenų bazėms neatsižvelgiant į jų struktūros specifiką. Vienas populiariausių tokių standartų yra XML. Trečiame skyriuje apžvelgiamos šiuolaikinių duomenų bazių valdymo sistemų priemonės skirtos XML srautams formuoti.

Pažymėtina, kad realizuojant sudėtingas „verslas verslui“ ar „verslas klientui“ tipo sistemas duomenų bazių valdymo sistemų teikiamų galimybių dažnai nepakanka. Transakcinių veiksmų vykdymui už duomenų bazės ribų naudojami transakcijų serveriai. Šio tipo sistemos apžvelgtos ketvirtame skyriuje.

Penktame darbo skyriuje aprašytas komunikacinių kilpų metodo taikymas. Nustatyta, kad persiunčiant tarp aktorių XML srautus, visos verslo procesų operacijos turi būti protokoluojamos DB žurnaluose. Tokiu būdu verslo transakcija būtų susiejama su duomenų bazių transakcijomis. Duomenų bazėje turi būti išsaugojami kilpą identifikuojantys parametrai. Tai leistų nesėkmės atveju atstatyti verslo transakciją į pradinę būseną. Taip pat pateikiami formalūs exist- ir coexist- priklausomybių apibrėžimai. Šių priklausomybių panaudojimas leidžia detalizuoti sudėtingas verslo transakcijas ir pateikti jas paprastų binarinių kilpų rinkiniais.

Išnagrinėtam ir siūlomam naudoti komunikacinių kilpų metodui iliustruoti sudarytos universiteto akademinio posistemio naujų funkcinių reikalavimų specifikacijos dviem metodais – UML kalba ir komunikacinėmis kilpomis.

Darbo pabaigoje pateikiamos šio darbo išvados, taip pat nurodoma naudota literatūra, bei priedas – magistrantų ir doktorantų konferencijoje pristatytas straipsnis „Naujų funkcinių reikalavimų įvedimas informacijos sistemose“.

1. VERSLO PROCESŲ MODELIAVIMO METODŲ APŽVALGA

Kuriant elektroninio verslo sistemas, procesų modeliavimas yra svarbiausia veiklos dalis. Elektroninio verslo procesų modeliavimo metodų reikalavimai skiriasi nuo procesų modeliavimo, skirto informacijos sistemoms kurti ar darbų sekų valdymui automatizuoti, reikalavimų. Elektroninio verslo procesų, apimančių kelias bendradarbiaujančias organizacijas, modelius tikslinga konstruoti modulinio principu. Tam egzistuoja reikiama informacinių technologijų parama, tačiau modeliavimo metodologijos nėra iki galo subrendusios.

Modeliavimas yra būtinas kuriant dideles, sudėtingas ar su kitomis sistemomis sąveikaujančias informacines sistemas, nors daugelis programuotojų siekia minimizuoti modeliavimo pastangas. Modeliuojami ne tik vartotojo poreikiai, dalykinė sritis, programinė įranga, bet ir organizacijos veikla - pastaraisiais metais ši modeliavimo fazė tapo daugelio projektavimo procesų dalimi. Nors egzistuoja iširtos, stabilios veiklos procesų modeliavimo metodologijos (RUP, Provision, Popkin, Demo, Action Workflow ir kitos), tačiau informacinių technologijų plėtra nuolat suteikia galimybių tobulinti verslo procesus ir savo ruožtu kelia didesnius reikalavimus tų procesų modeliavimui. Egzistuojančių veiklos modeliavimo metodų galimybės tapo nepakankamos modeliuoti elektroninio verslo procesus, apimančius ne vieną, bet kelias bendradarbiaujančias organizacijas. Kitas postūmis tobulinti veiklos procesų modeliavimo metodus - programinės įrangos paslaugų koncepcijos, XML technologijų bei tinklo paslaugų architektūros (.NET) atsiradimas. .NET architektūra grindžiama bendradarbiavimu, kurio metu jos komponentai sąveikauja tarpusavyje ir su išorinėmis sistemomis. Dėl XML, .NET ir kitų su jomis susijusių informacinių technologijų sudėtingumo reikia keisti ir pačius modeliavimo metodus.

1.1. Verslo procesų modeliavimo ypatumai

Galima išskirti du veiklos procesų modeliavimo metodų tipus:

- veiklos procesų modeliavimas, siekiant suprasti ir paruošti geresnį informacinės sistemos projektą;
- veiklos procesų modeliavimas, siekiant automatizuoti veiklos proceso valdymą. Pastaruoju atveju galima išskirti darbų sekų procesų modeliavimą, labiau orientuotą į įmonės vidinių procesų valdymą, ir išorinių, kelias organizacijas apimančių procesų modeliavimą.

Projektuojant organizacijos informacinę sistemą, veiklos procesų modelis padeda susidaryti bendrą pradinį sistemos elgsenos vaizdą, identifikuoti vartotojų poreikius (panaudojimo atvejus) bei veiklos objektus (esybių ir procesų objektus), suformuluoti sistemos projektavimo užduotis, testuoti sistemą. Dauguma šiuo metu egzistuojančių veiklos procesų modeliavimo metodologijų siekia sudaryti ištinį veiklos proceso aprašą, apimantį visus proceso žingsnius nuo pradinio sužadinančio įvykio iki proceso

rezultato gavimo. Tai būdinga Provision, Popkin metodikai, UML pagrįstiems metodams ir daugeliui kitų. Į projektavimą orientuoti modeliai aprašo proceso vykdymo eigą, bet nesiekia pavaizduoti visų galimų scenarijų aibės. Procesų etapų aprašai naudojami informacinių sistemų projektavimui bei reinžinerijai, kuriant interneto informacines sistemas, kompiuterizuotas vartotojų darbo vietas, diegiant daugiavartotojiškus integruotus taikomųjų programų paketus. Galutinis projekto rezultatas - veikiantys programiniai komponentai, kuriuos vartotojai susieja į procesą vykdymo metu.

Į procesų automatizavimą orientuotais modeliavimo metodais siekiama sudaryti išsamų proceso aprašą - jo apibrėžimą. Tuo pasižymi darbų sekų valdymui skirti veiklos procesų modeliavimo metodai Action Workflow, DEMO ir kiti. Pagal Workflow Management Coalition (WfMC) darbų sekų standartus proceso apibrėžimu, arba schema, suprantamas kompiuterinis formalizuotas jo aprašas, kurį sudaro veiksmų seka, skirta pasiekti bendrą tikslą. Ši seka gali būti gana sudėtinga - ji apima lygiagrečius, nuoseklius veiksmų rinkinius, ciklus bei rekursyvų proceso žingsnių aprašymą - veiksmas gali būti ne tik atominis, jis gali reikšti kito proceso iškvietimą. Proceso apibrėžimas apima procesą sudarančius veiksmus, jų dalyvius, perėjimus nuo vieno veiksmo prie kito, su proceso vykdymu susijusius duomenis bei iškviečiamų taikomųjų programų aprašus. Sudarytas WfMC proceso apibrėžimas transformuojamas į XPDL (angl. *XML Process Definition Language*). WfMC standartus atitinkanti darbų sekų valdymo programinė įranga gali valdyti veiklos procesus pagal XPDL aprašus. Proceso automatizavimo projekto rezultatas - veikianti programa, kuri koordinuoja proceso vykdymą perduodama valdymą iš vieno dalyvio kitam arba prireikus atlieka žingsnius automatiškai, iškviisdama reikiamas taikomąsias programas. Automatiškai iškviečiamos ar vartotojų naudojamos programos, kuriomis jie vykdo proceso žingsnius, turi būti sukurtos iš anksto arba tam sudaromi projektai. Procesų schemas saugomos saugykloje, kuri leidžia daug kartų panaudoti procesų apibrėžimus.

Proceso schema turi apibrėžti visus galimus jo vykdymo scenarijus. WfMC proceso apibrėžimo standarte daug dėmesio skiriama veiksmų jungimo, sinchronizavimo galimybėms. Nors konstrukcijų pasirinkimas didelis, jomis išreikšti proceso semantiką praktiškai ne visuomet lengva. WfMC apsiriboja proceso ir veiksmo sąvokomis, procesų dekompozicija pasiekama pasitelkus rekursiją - kai kurie veiksmi gali tapti procesais. Proceso išskaidymas į verslo transakcijas paliekamas konkrečių modeliavimo metodų kūrėjų kompetencijai, todėl nėra vienodo verslo transakcijos supratimo - vieni metodai visą procesą traktuoja kaip vieną transakciją, kiti naudoja mažesnės apimties transakcijas.

Griežta vykdymo tvarka būdingesnė vidiniams organizacijų procesams, tuo tarpu tiksliai apibrėžti bendradarbiaujančių organizacijų elektroninio verslo procesų scenarijus yra sunkiau: atskiri procesų elementai (transakcijos) gali kartotis neapibrėžtą skaičių kartų, jų vykdymo tvarka ne visada svarbi. Atsiranda poreikis formuoti procesą iš dalių. Bendras proceso apibrėžimas turi apimti skirtingų

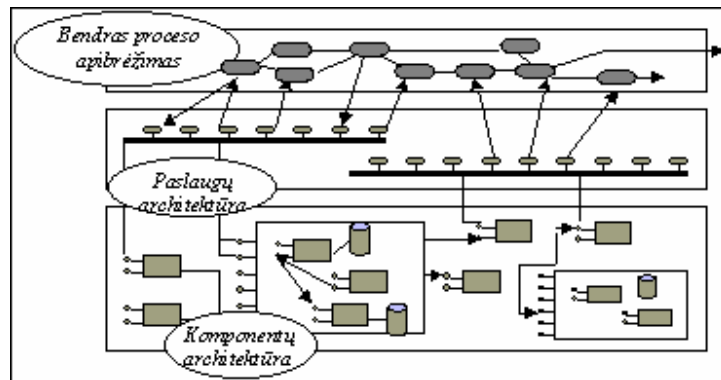
sąveikaujančių organizacijų verslo procesus. Esminė šio apibrėžimo savybė ta, kad jis negali būti nė vieno veiklos dalyvių nuosavybė, bet turi remtis partnerių susitarimais - kontraktais. Kitas svarbus reikalavimas, kurį turi tenkinti elektroninio verslo modeliavimo metodologija, - ji turi remtis ekonominėmis sąvokomis, nes elektroninis verslas visų pirma turi būti gerai suformuluotas verslas. Elektroninio verslo modeliavimo metodologijoje turi būti analizuojami ekonominiai įvykiai, kurių metu veiklos dalyviai keičiasi ekonomine verte. Be to, bendram procesų apibrėžimui keliami dar mažiausiai trys principiniai reikalavimai:

- Turi būti bendras veiklos procesų metamodelis ir su juo susijusi schemas apibrėžimo kalba;
- Turi būti mechanizmas, padedantis kurti sudėtingus procesų bei su jais susijusių dokumentų aprašus iš bazinių daugkartinio naudojimo elementų. Šie baziniai elementai turi turėti modifikavimo bei pritaikymo vertikalių veiklos sričių poreikiams galimybes;
- Turi būti mechanizmas, kuris leistų organizacijoms skelbti apie savo galimybes atlikti tam tikrus vaidmenis bendruose veiklos procesuose, teikiant tinklo paslaugas (angl. *web services*), kad potencialūs partneriai galėtų rasti vienas kitą ir kartu vykdyti veiklos procesus.

Šiuolaikinė elektroninio verslo modeliavimo metodologija turi apimti vientisą programinės įrangos ir veiklos valdymo modeliavimo procesą, kuriame įvertinami ekonominiai tikslai, standartiniu būdu konstruojami schemas elementai ir jų valdymo scenarijai, schema susiejama su informacinių technologijų infrastruktūra - tinklo paslaugų sąveikomis. WfMC schemoms keliamas abstraktus reikalavimas nurodyti išskviečiamas taikomas programas ir duomenis, perduodamus šioms programoms išskvietimo metu. Konkrečiu taikymo atveju darbų sekų apibrėžimas gali būti naudojamas su bet kokia prieinama programų integravimo infrastruktūra ir integruotomis ar neintegruotomis vartotojų valdomomis programomis.

1.2. Tinklo paslaugų koncepcija kuriant programinę įrangą

Tinklo paslaugomis suprantamos taikomosios programos, pateikiamos internete paslaugų pavidalu. Paslaugos yra nuosekli komponentinio kūrimo tąša. Tinklo paslaugomis gali naudotis nutolę vartotojai – pavieniai asmenys ar verslo įmonės; gali vykti ir tiesioginė sąveika tarp paslaugų, kurios atstovauja organizacijoms elektroninio verslo procesuose. Bendro proceso apibrėžimo koncepcija, taikant tinklo paslaugas, parodyta 1.2.1 pav.



1.2.1 pav. Bendras proceso apibrėžimas ir tinklo paslaugų architektūra

Proceso apibrėžimas nurodo verslo įvykių sekas ir paslaugų reikalavimus. Tai pats nepastoviausias architektūros lygis, kuris keičiasi priklausomai nuo verslo dalyvių bendradarbiavimo poreikių. Paslaugų architektūra apibrėžia verslo paslaugas ir tai, kaip jas palaiko komponentų architektūra. Ji sprendžia du pagrindinius uždavinius: atvaizduoja verslo poreikius į juos realizuojančius komponentus, o paslaugas - į atitinkamas semantines sritis. Daugelio komponentų sąsajos tiesiogiai pateikiamos kaip verslo paslaugos. Tais atvejais, kai komponentų ir paslaugų atitiktis nėra paprasta (pavyzdžiui, paslauga apima keleto komponentų sąsajas), reikia sukurti fasado komponentus, kurie integruoja atitinkamas sąsajas ir pateikia juos verslo paslaugų pavidalu. Atvaizdavimas į semantines sritis reikalingas dėl to, kad, įsijungus į pasaulinį tinklą, verslui nepakaks savo apibrėžtos duomenų semantikos, reikės derintis prie rinkos, pramonės ar partnerių standartų. Komponentų architektūra teikia reikiamas funkcijas. Daugelyje organizacijų šią architektūrą sudarys nauji komponentai, taikomųjų programų paketai (angl. *packaged applications*) bei palikuoninės programos su interfeisiniu apvalkalu (angl. *wrapped legacy*).

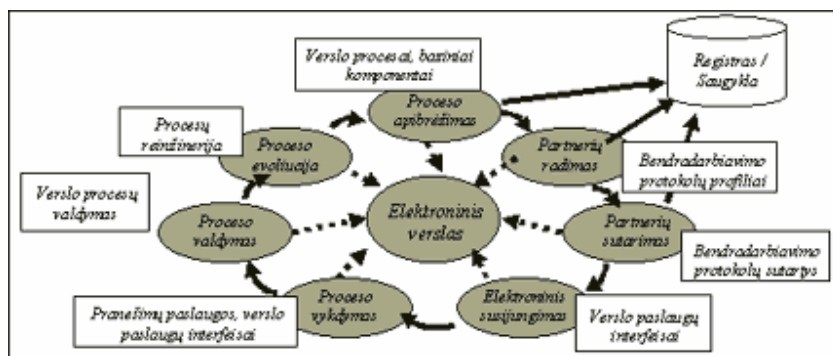
Šiuo metu egzistuoja keletas stambių organizacinių struktūrų, kurios sprendžia elektroninio verslo procesų integravimo ir tinklo paslaugų teikimo problemas, pavyzdžiui, WWW Consortium (W3C), ebXML, RosettaNet, BizTalk, UDDI. Iš jų išsiskiria ebXML, kuriuo siekiama įvertinti ankstesnę pramonės patirtį, įgytą taikant elektroninio keitimosi duomenimis standartą EDI, bei stengiamasi patenkinti visus tris anksčiau minėtus reikalavimus.

1.3. EbXML veiklos procesų modeliavimo metodologija

EbXML yra specifikacijų aibė, kuri leidžia sukurti modulinį elektroninio verslo modelį [11], [12]. EbXML atsirado bendra Jungtinių Tautų centro UN/CEFACT ir Struktūruotos informacijos standartų tobulinimo organizacijos OASIS iniciatyva ir skirta globaliam naudojimui. EbXML specifikacijos apima bendrą veiklos procesų metamodelio standartą - specifikavimo schemą, kurioje procesų sąveikos

suvienijamos dokumentų šrtais; bazinius XML komponentus, kurie leidžia daug kartų naudoti dokumentų ir veiklos procesų schemas; saugyklą, kurioje galima registruoti ir rasti verslo partnerius, tinklo paslaugas bei XML schemas; registrą, per kurį vyksta bendravimas su saugykla; pranešimų paslaugas, kurios leidžia bendradarbiaujančioms organizacijoms keistis pranešimais.

EbXML vizija - sudaryti galimybes sukurti globalią elektroninę rinką, kurioje bet kokio dydžio įmonės, esančios bet kurioje pasaulio vietoje, gali susitikti ir kartu vykdyti verslą pasitelkusios XML pranešimus. B2B bendradarbiavimo procesas, taikant ebXML specifikacijas, pateikiamas 3 paveiksle. EbXML veiklos procesų apibrėžimas iš pradžių sudaromas UML kalba, vėliau jis transformuojamas į XML schemą. EbXML procesų modeliavimo metodiką galima taikyti tiek bendriems tarporganizaciniams, tiek vidiniams įmonės procesams modeliuoti. Proceso specifikacija transformuojama į ebXML reikalavimus atitinkančią programinę įrangą, pavyzdžiui, tinklo paslaugų sąsajų konfigūraciją. Proceso vykdymo aplinka kartu yra programų integravimo įrangą, kadangi sujungiant skirtingus proceso elementus tenka dinamiškai integruoti skirtingų organizacijų įvairių tipų programas. EbXML schema konstruojama modulinio principu, kuris užtikrina reikiamą vykdymo laisvę.

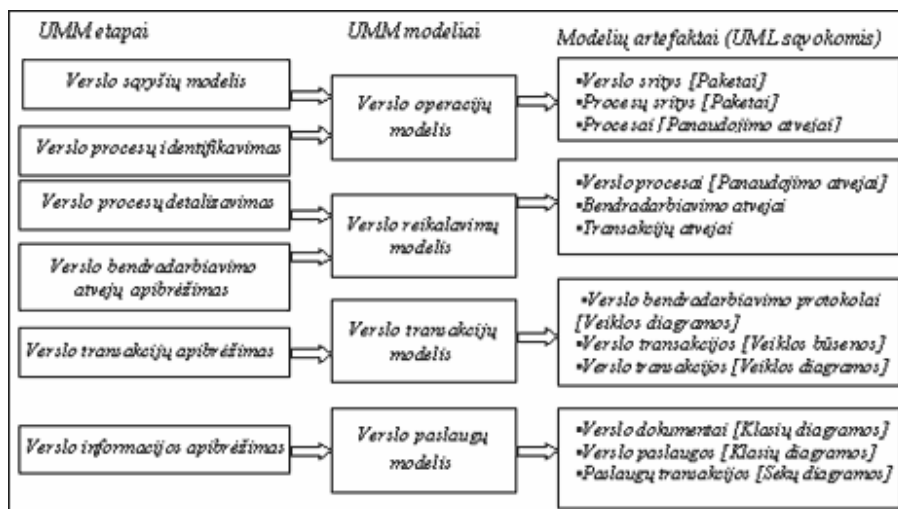


1.3.1 pav. Verslo bendradarbiavimo procesas ir atitinkamos ebXML specifikacijos

EbXML procesų schema remiasi UMM (angl. *UN/CEFACT Modelling Methodology*) metodologija, kuri buvo sukurta pritaikant RUP (angl. *Rational Unified Process*) metodologiją elektroninio verslo modeliavimo tikslams. Tam buvo įvesti stereotipai bei modeliavimo procedūra, kuri leidžia gauti visą veiklos procesų ir informacijos apibrėžimą, nepriklausomą nuo realizavimo technologijos.

Nuo paprasto tinklo paslaugų naudojimo ebXML schema skiriasi tuo, kad joje vartotojų ir paslaugų sąveikų scenarijai modeliuojami transakcijomis, susietomis su veiklos procesais ir atitinkamais verslo dokumentais. Paslaugų naudojimo sąlygos, kurios turi būti tenkinamos komunikavimo metu, derinamos tarp verslo partnerių ir registruojamos bendradarbiavimo protokolų profiliuose bei sutartyse. Šie elementai užtikrina reikiamą paslaugų sąveiką elektroninio verslo realizavimo aplinkose.

EbXML naudoja UMM poaibį, kurio etapai parodyti 1.3.2 pav.

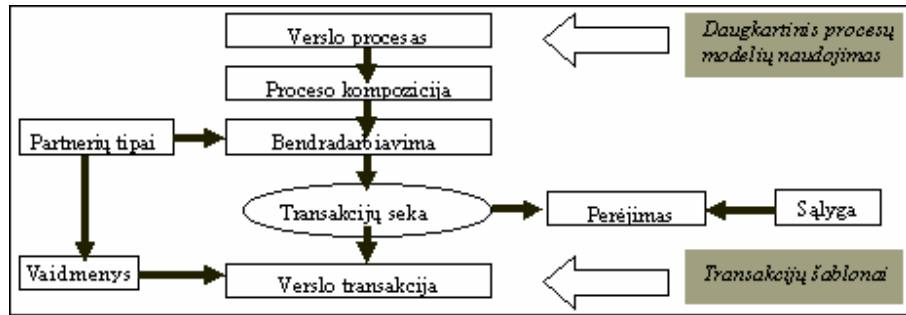


1.3.2 pav. Veiklos procesų modeliavimo etapai ir modeliai, naudojami ebXML

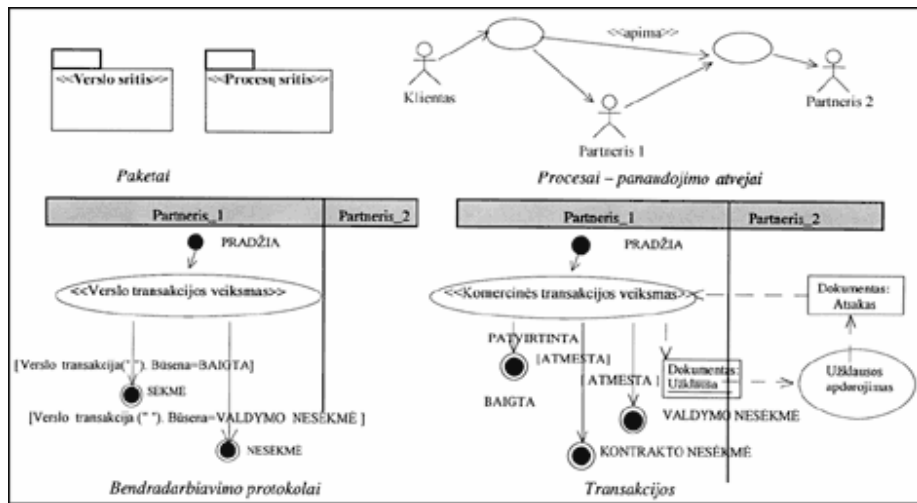
Verslo sąryšių modelis nusako pagrindinius konceptus ir jų tarpusavio ryšius, pateikia bendrą verslo proceso vaizdą. Verslo proceso identifikavimo metu apibrėžiamas aukšto lygio panaudojimo atvejų rinkinys. Verslo procesų detalizavimo etape apibrėžiami aktorių vaidmenys bei verslo procesų pradinės ir galinės būsenos. Verslo operacijų modelis nusako nagrinėjamas verslo sritis ir procesų sritis, kurios vaizduojamos verslo modelio paketų struktūra. Verslo bendradarbiavimo atvejų apibrėžimo metu identifikuojami ekonominiai įvykiai, kuriais suprantamas ekonominių išteklių valdymo perdavimas iš vieno dalyvio kitam. Bendradarbiavimo protokolai apibrėžia bendradarbiaujančius dalyvius ir jų perduodamus informacinius pranešimus. Kiekvieną verslo bendradarbiavimo atvejį sudaro suderinta verslo transakcijų seka (angl. *choreography*). EbXML galimi du bendradarbiavimo tipai: binarinis ir daugelio partnerių (angl. *multiparty*) bendradarbiavimas, kuris išreiškiamas binarinių bendradarbiavimo atvejų aibe. Binarinis bendradarbiavimas apima verslo veiksmų aibę. Verslo veiksmas gali būti verslo transakcija arba kitas binarinis verslo bendradarbiavimo atvejis - tokiu būdu galima rekursyviai apibrėžti bendradarbiavimo atvejus. Verslo veiksmas apibrėžiamas transakcija arba bendradarbiavimo protokolu. Verslo transakcija yra atominis (daugiau neskaidomas) darbo vienetas, kuriame dalyvauja du partneriai ir kuris visuomet baigiasi sėkminga arba nesėkminga būsena. Bendradarbiavimo protokolas naudojamas tada, kai nesėkmės atveju nebūtina atkurti ankstesnę verslo būseną - pavyzdžiui, galimas laipsniškas užsakymo vykdymas dalimis. Verslo transakcijų sekas apibrėžia galimi perėjimai iš vienos veiksmo būsenos į kitą. Verslo transakcijų apibrėžimo etapas labiau techniškai orientuotas negu ankstesni etapai. Jame apibrėžiami realūs veiksmai ir organizacijų vaidmenys, kurie pradeda ir baigia transakcijas. Verslo informacijos apibrėžimo etape aprašomi pranešimai ir verslo paslaugos (klasių diagramomis) bei paslaugų transakcijos (sekų diagramomis). Visi projekto rezultatų elementai (procesai, bendradarbiavimo atvejai,

transakcijos, perėjimai ir kiti), be 1.3.2 paveiksle parodytų UML artefaktų, aprašomi specifikacijomis, kurios vaidina pagrindinį vaidmenį generuojant XML schemą.

Aprašydami transakcijas ir bendradarbiavimo atvejus, veiklos dalyviai turi susitarti dėl savo atliekamų vaidmenų. Verslo procesus galima konstruoti tiek iš viršaus žemyn, tiek iš apačios į viršų, panaudojant saugykloje užregistruotus transakcijų šablonus ar ištisus procesus. Labai supaprastinta veiklos procesų modelio schema pateikiama 1.3.3 pav., pagrindinės proceso apibrėžimo UML konstrukcijos – 1.3.4 paveiksle.



1.3.3 pav. Verslo proceso modelio sudarymo principai



1.3.4 pav. EbXML proceso schemas UML konstrukcijos

EbXML specifikacijas galima realizuoti bet kuria ebXML reikalavimus atitinkančia programine įranga (tinklo paslaugomis). Tai gali būti Microsoft .Net, Oracle 9/Dynamic Services ir kitos technologijos, kurios naudoja XML ir paslaugas. Paslaugos komponuojamos iš egzistuojančių duomenų bazių, interneto ir kitų taikomųjų programų bei procedūrų, o paskui naudojamos portaluose, elektroninio verslo procesuose.

EbXML schemas apibrėžimas įdomus kaip modeliavimo technika, kurią galima taikyti ne tik išoriniams, bet ir vidiniams įmonės procesams. Remiantis EbXML specifikacijomis, buvo sumodeliuoti interneto teikimo paslaugų verslo procesai. Pažymėtina, kad ebXML specifikacijos dar tobulinamos, todėl modeliavimo metodologijoje yra neapibrėžtumų. Neaprašyta UMM ir ebXML metamodelių atitiktis. Dažnai UMM modeliavimo galima atsisakyti ir iš karto konstruoti ebXML schemą UML arba XML kalba. Pastaruoju atveju didėja tikimybė pažeisti procesų modelio korektiškumą bei vientisumą.

1.4. Procesų modelių korektiškumas bei vientisumas

Komponuojant elektroninio verslo proceso apibrėžimą iš bendradarbiavimo protokolų ir transakcijų, iškyla proceso modelio korektiškumo ir vientisumo klausimai, kuriuos ebXML kontekste galima suformuluoti taip:

- Ar ekonomiškai pagrįstas verslo modelis?
- Ar teisingai specifikuota proceso eiga?
- Ar perduodami verslo dokumentų elementai parodo verslo būsenų pasikeitimus?

Formalūs procesų specifikacijų korektiškumo tikrinimo, procesų ir informacinių esybių integralumo klausimai būdingi teoriniams procesų modeliavimo darbams. Tačiau formalūs metodai paprastai būna nesusieti su praktiniais modeliavimo metodais bei realizavimo technologijomis.

Toliau bus nagrinėjamas Enterprise Modeling (EM) metodo semantinio vientisumo kriterijus. Nesunku įžiūrėti sąryšį tarp ebXML ir EM procesų modeliavimo metodų. Literatūroje [5] pateiktas EM metodas, adaptuotas procesų modeliavimui UML kalba (toliau jį vadinsime EMUML, kadangi EMUML yra skirtumų ir išplėtimų, palyginti su originaliu EM). EMUML metodu procesai konstruojami iš transakcijų, kurios detalizuojamos veiksmų kilpomis, susietomis ištisiniu proceso aprašu. Procesai siejami tarpusavyje per esybių objektų būsenas. EbXML metodo EMUML transakcija, kuri siejama su dalyvių tikslais, atitiktų bendradarbiavimo atvejis, kuris ebXML siejamas su ekonominiu įvykiu ir pasikeitimu ekonominiais ištekliais; EMUML paprastą veiksmų kilpą atitiktų ebXML transakcija, EMUML komunikacinį veiksmą - ebXML verslo veiksmas. Išplėsta veiksmų kilpa (EMUML) atitiktų kelių partnerių bendradarbiavimą (ebXML). EM metodas pagal sumanymą skirtas informacinės sistemos kūrimui, ir pagrindinis jo privalumas - pragmatinio ir semantinio vientisumo kriterijų tenkinimas:

- susietumas (angl. *cohesion*) - ryšiai tarp informacinės sistemos būsenų pasikeitimų ir komunikacinių srautų vidinės struktūros, kurie pasiekiami būsenų pasikeitimus atvaizduojant į esybių atributus;
- pilnumas (angl. *completeness*) – nepilnumą rodo probleminės (nesėkmingos) būsenos, kurios gali duoti pradžią naujų galimybių ieškojimui;

- suderinamumas (angl. *coherence*) - galinės būsenos turi būti suderinamos su aktorių tikslais;
- gyvybingumo (angl. *viability*) kriterijų užtikrina tipinė veiksmų kilpa, kurioje tarp būsenų galioja pragmatinės priklausomybės. Žodžiais jas galima išreikšti taip: jei vykdytojo galimybė neigiamai veikia kliento problemą, tuomet vykdytojo galimybė teigiamai veikia kliento tikslą ir kliento tikslas neigiamai veikia vykdytojo problemą (vykdytojas padeda klientui pasiekti tikslą, klientas sumažina arba panaikina vykdytojo problemą).

Komunikacinių veiksmų kilpą galima išplėsti įtraukiant naujus aktorius, kurių problemas neigiamai veikia procese dalyvaujančių aktorių tikslai. Sistemos gyvybingumas suprantamas kaip galimybė reaguoti į nepageidaujamas situacijas ir užtikrinti veiksmus, kurie perkeltų sistemą į pageidaujamą būseną arba sudarytų galimybę į ją pereiti. Nėra apibrėžto sistemos gyvybingumo mato, tačiau jį būtų galima įvertinti alternatyvių scenarijų skaičiumi arba "gerų" scenarijų santykiu su visais scenarijais, kurie sukelia skirtingas galines būsenas.

Naudojant EMUML metodą, vientisumo kriterijai tenkinami susiejant veiklos, būsenų ir klasių diagramas. Tokią procedūrą tikslinga įtraukti į UMM ir ebXML procesų komponavimo metodiką. Susietumo kriterijus užtikrintų tinkamą partnerių perduodamų verslo dokumentų struktūrą; suderinamumo kriterijus - gerai suformuluotą ekonominį verslo pagrindimą; išsamumo ir gyvybingumo kriterijai padėtų atskleisti tinklo paslaugų rinką, stimulą plėtoti tinklo paslaugas.

Procesų eigos korektiškumą UMM metodologijoje užtikrina visą procesą apimanti UML veiklos diagrama, EMUML - esybių objektų būsenomis susietos veiklos diagramos. Pažymėtina, kad UMM veiklos diagrama yra daug paprastesnė, palyginti su veiklos diagrama WfMC arba EMUML atveju. Jos nepakanka pragmatinio ir semantinio vientisumo kriterijams tenkinti, tačiau pakanka korektiškai proceso eigai užtikrinti. Jei EbXML atsisako ištisinės UMM veiklos diagramos, jame turi būti korektiškumo tikrinimo procedūra, kuri tikrintų schemas elementų specifikacijas UML kalba aprašytoje schemoje arba XML specifikacijas, jei schema iš karto sudaroma XML kalba. Tokiai procedūrai sudaryti verslo vykdymo taisyklės reikėtų aprašyti aukšto lygio kalba, panašia į verslo modeliavimo kalbą BML arba taisyklingų išraiškų kalbą.

1.5. Analizės išvados

Šiuolaikines verslo procesų modeliavimo metodų tendencijas geriausiai atskleidžia ebXML veiklos procesų modeliavimo metodologija, kuri apima į kūrimą bei procesų valdymą orientuotą modeliavimą, remiasi ekonomine verslo verte ir aprašo verslo procesus modulinio principu. Kaip modeliavimo rezultatas gaunama vykdoma proceso schema, susieta su tinklo paslaugų architektūra, kuri remiasi XML technologijų infrastruktūra. Apibrėžti su modeliavimo metodologija susiję verslo pranešimų,

bendradarbiavimo protokolų, bazinių komponentų, procesų saugyklos ir registro standartai. Įvertinant į paslaugas orientuotos programinės įrangos kūrimo tendencijas, galima prognozuoti, kad ateityje vyraus į EbXML panašūs veiklos procesų modeliavimo metodai.

Tačiau ebXML procesų modeliavimo metodologija nėra iki galo baigta. Įdomu tai, kad procesas konstruojamas iš standartinių atominių elementų - transakcijų, bendradarbiavimo protokolų ir verslo dokumentų. Tai supaprastina proceso apibrėžimą, bet giluminės semantinės šių elementų priklausomybės lieka paslėptos. Sutrumpintas veiklos proceso apibrėžimo metodas užtikrina sintaksinį schemas korektiškumą, bet neužtikrina pragmatinio ir semantinio vientisumo bei veiksmų eigos korektiškumo. Tobulinant šią metodologiją, reikia aiškiai susieti ebXML schemas sudarymą su išplėsta UMM metodologija ir sudaryti tikrinimo procedūras greitam kūrimui.

2. VERSLO TRANSAKCIJŲ MODELIAVIMAS KOMUNIKACINĖMIS KILPOMIS

2.1. Verslo transakcijos informacijos sistemose

Norint garantuoti, kad verslo informacijos sistemose, ypač susijusiose su organizacijų darbų seku valdymu bei „verslas klientui“ (angl. *Business-To-Customer*) ar „verslas verslui“ (angl. *Business-To-Business*) procesais, suderintai ir patikimai funkcionuotų taikomosios programos, reikia palaikyti verslo transakcijas. Kadangi vartotojų programos ir duomenys tokiose sistemose tendencingai atskiriamos, tai transakcijų mechanizmų svarba tokiomis sąlygomis labai padidėja. Paskirstytiesiems verslo procesams būdingos ilgos trukmės (palyginti su duomenų bazių transakcijų trukmėmis), jie realizuojami įvykdant daugelį DB transakcijų ir kaupiant duomenų srautus paskirstytosiose DB. Verslo procesus galima apibrėžti verslo transakcijų aibėmis.

Elektroninio verslo procesas susideda iš veiksmų, kuriems reikalinga prieiga prie daugialypių duomenų bazių, ir jis turi būti siejamas su transakcijų semantika, nes transakcijų būdingosios vadinamosios ACID (nedalumas, vientisumas, izoliavimas, išliekamumas; angl. *Atomicity, Consistency, Isolation, Durability*), savybės gali būti taikomos tiek verslo procesams, tiek šių procesų dalims. Pažymėtina, kad netikslinga visą verslo procesą traktuoti kaip ACID transakciją, nes tai pareikalautų daug ilgam laikui išaldytų išteklių ir brangaus nepriklausomų paskirstytųjų duomenų bazių koordinavimo. Kartais būsenų atkūrimo (angl. *rollback*) mechanizmai valdant transakcijas gali būti nepageidaujami arba tiesiog neįvykdomi, kadangi verslo procesai paprastai turi išorinius efektus. Todėl įvairiuose šaltiniuose pasirodė įvairūs išplėstiniai transakcijų modeliai. Pavyzdžiui, Saga modelio transakcija sudaryta iš elementariųjų transakcijų grandinės, o kiekviena elementarioji grandinės transakcija dar turi savo

kompensacinę transakciją. Jeigu Saga transakcijoje elementarioji transakcija neįvykdoma teisingai, tai atkuriamas ankstesnė būseną ir valdymas perduodamas atitinkamai kompensacinei transakcijai. Veiklos transakcijų modelyje ATM (angl. *Activity Transaction Model*) ilgos trukmės transakcijos sudėtiniai elementai gali būti jungiami tiek į grandines (angl. *chained*), tiek į rinkinius (angl. *nested*). Rinkinio transakcijos vykdomos tuo pačiu metu, joms taikoma hierarchinė klaidų ir išimčių kontrolė. Čia skiriami uždarujų ir atvirųjų sudėtinių transakcijų modeliai. Uždarujų sudėtinių transakcijų rezultatai išorėje fiksuojami tik po to, kai ji užbaigiama. Tokiu būdu užtikrinami abu - ir nedalumo, ir izoliavimo - principai. Atvirųjų transakcijų atžvilgiu izoliavimo principas pažeidžiamas, nes vienu metu vykdoma ir sudėtinė, ir ją sudaranti elementarioji transakcija, o pastarosios rezultatai perduodami išorėje dar nesibaigus tėvinei transakcijai. Klaidos atveju tėvinis narys sprendžia, ar kartoti klaidingą dalinę transakciją, ar vykdyti nenumatytus uždavinius, ar perkelti klaidą į hierarchijos viršų. Tokiu būdu ATM modelio transakcijų kompensacijos ir nenumatyti uždaviniai asocijuojami su skirtingais hierarchijos lygiais. Vėlesniuose siūlymuose klaidas numatoma perduoti iš vienos kontrolės hierarchijos į kitą. Šaltinyje [7] pateikiami įvairūs ATM modeliai ir transakcijų semantikos formalizmai. Pažymėtina, kad išplėstiniai transakcijų modeliai pernelyg riboja paslaugų uždavinius, išskirstytus laiko ir erdvės atžvilgiais.

Norint apibrėžti išorinių organizacijų verslo procesų transakcijų protokolą BTP (angl. *Business Transaction Protocol*), Standartų ir pramonės konsorciumas OASIS įsteigė Verslo transakcijų techninį komitetą. Šaltinyje [6] verslo transakcija apibrėžiama kaip nuosekli dviejų ar daugiau partnerių verslo ryšių būsenų kaita. BTP naudoja dviejų fazių bendradarbiavimo rezultatų skaičiavimo protokolą ir numato nedalomų darbų vienetų sujungimą į darnią verslo transakciją (derinį). Čia į sekų formavimo uždavinį leidžiama įsikišti iš išorės, patvirtinant arba atšaukiant nedalumus darbų vienetus. BTP užtikrina didelį verslo transakcijos dalyvių veiklos padarinių vertinimo ir patvirtinimo bei atšaukimo įrankių lankstumą. Partneriai gali naudoti kompensacines operacijas, norėdami sugrįžti atgal arba judėti pirmyn ir tokiu būdu koordinuoti skaidomas verslo transakcijas. BTP transakcijų modelyje darbo vienetai gali turėti ACID savybių, bet visuminė verslo transakcija gali būti grąžinama arba jau įvykdytos jos sudėtinės dalys sunaikinamos. Kitaip sakant, transakcijų izoliavimo konceptas nėra griežtas. OASIS BTP palaiko paslaugų vartotojų transakcijų koordinavimą tinklo paslaugų ir verslo bendradarbiavimo aplinkose. Paslaugų vartotojais gali būti tiek pavienės, tiek daugialypės autonomiškos organizacijos.

Pažymėtinos dvi elektroninio verslo modelių sudarymo kalbos – ebXML ir XLANG. ebXML metodologija verslo procesą apibrėžia kaip verslo bendradarbiavimą, tai yra, dviejų ar daugiau partnerių veiklą, kuria norima gauti tam tikrą rezultatą. Daugelio partnerių bendradarbiavimą galima išskaidyti į kelias binariąsias formas. ebXML verslo bendradarbiavimas aprašo sudėtinių transakcijų modelį, kai klaidų atvejais skaidžių transakcijų nebūtina sunaikinti. Binarusis bendradarbiavimas, kuris neskaidomas į

smulkesnius veiklos vienetus, reiškia verslo transakciją, kuri generuoja suskaičiuojamą teisingą arba klaidingą būseną. Iš transakcijų sekos (choreografijos, angl. *choreography*). suformuojama verslo proceso specifikacija (schema). XLANG yra vietinės reikšmės kalba, Microsoft BizTalk Server naudojama verslo procesams formaliai specifikuoti. Šioje kalboje naudojama ilgos trukmės kompensuojamosios transakcijos samprata, tačiau kompensacijos taikomos tik klaidoms apdoroti.

Verslo transakcijų modelių lyginamoji analizė [1], [4] rodo, kad verslo transakcija įvairiuose elektroninio verslo modeliavimo metoduose suprantama skirtingai ir neturi tikslaus apibrėžimo. Kai kuriuose protokoluose ar kalbose (pavyzdžiui, BTP arba XLANG) visai neapibrėžiama, kokios veiksmų sekos gali būti pripažįstamos transakcijomis. ebXML bendradarbiavimas apibrėžiamas ekonominių įvykių pagrindu, t. y. transakcijos reiškia ekonominius įvykius, kurių metu vienas partneris perduoda kitam ekonominę vertę ar ekonominių išteklių kontrolę. Dažniausiai verslo transakcija apibrėžiama kaip universaliosios modeliavimo kalbos UML panaudojimo atvejis: veiksmų seka, kuri duoda tam tikrą reikšmingą rezultatą šią seką inicijavusiam aktoriui. Toks traktavimas sukelia problemų dėl panaudojimo atvejų detalizavimo laipsnio neapibrėžtumo. Darbe [1] verslo transakcijas siūloma apibrėžti komunikacinių veiksmų kilpomis, kurios sudaro organizacijos modeliavimo EM metodo [2] pagrindą.

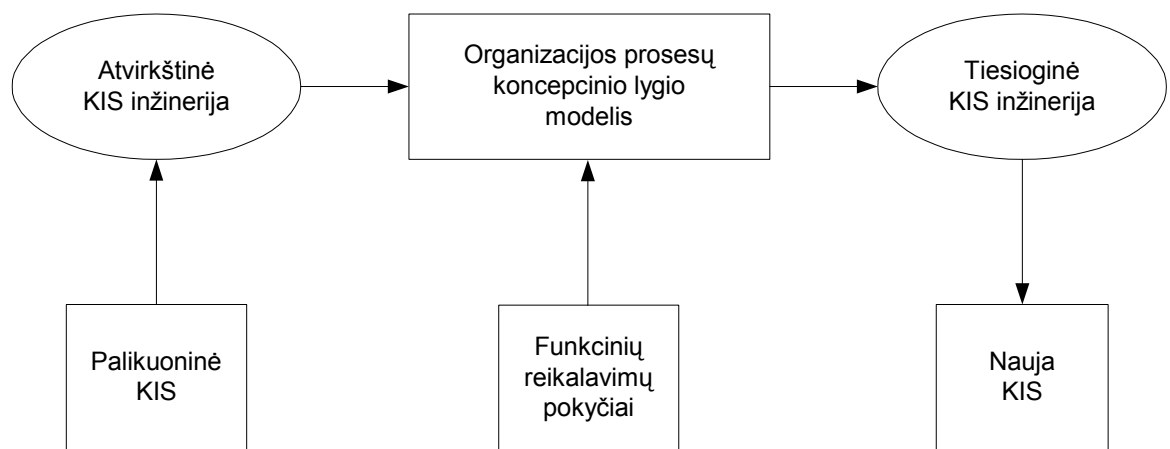
Elektroninio verslo procesams būdinga nuolatinė kaita. Keičiasi ne tik transakcijų sekos, bet ir paslaugų turinys. Paslaugos realizuojamos jas teikiančiose organizacijose ir skelbiamos visuotiniame interneto tinkle. Taigi verslo sistemose nuolat atsiranda naujų funkcinių reikalavimų, kuriems dengti reikia naujų duomenų struktūrų. Norint patenkinti pakitusius informacijos poreikius, naujas duomenų struktūras tikslinga integruoti su esamų palikuoninių duomenų bazių schemomis. Naujus funkcinius reikalavimus taip pat tikslinga apibrėžti komunikacinių veiksmų kilpomis, kurios integruojamos į vidinių ar bendrų verslo procesų modelius. Norint į esamą informacinę sistemą įjungti naujus reikalavimus, reikia identifikuoti palikuoninių duomenų bazių ir komunikacinių kilpomis aprašytų duomenų schemų sanklotas ir nustatyti jų struktūrinį ekvivalentumą. Nauji funkciniai poreikiai motyvuojami pragmatinėmis priklausomybėmis, tuo užtikrinant informacinės sistemos gyvybingumo ir informacijos susietumo kriterijus.

2.2. Hibridinės duomenų bazės

1.1 pav. pateikta kompiuterizuotosios informacijos sistemos (KIS) reinžinerijos schema, kurioje išskiriami tiesioginės ir atvirkštinės inžinerijos procesai. Nauji funkciniai reikalavimai vaizduojami bendrajame organizacijos procesų modelyje ir derinami su palikuoninės KIS procesais konceptualiajame lygmenyje. KIS atvirkštinės inžinerijos uždavinys yra gauti informaciją apie kompiuterizuotus duomenų apdorojimo procesus ir apibendrinti bei pateikti ją standartizuota verslo procesų aprašymo kalba. Toje

pačioje kalboje įvedami funkcinių reikalavimų papildymai. Iš visuotinio organizacijos procesų modelio sudaromas naujos KIS projektas, kurio svarbi sudėtinė dalis yra DB projektas.

Tradiciskai duomenų bazės loginio lygio (programuotojo arba DBVS lygio) schemų seka būtų generuojama automatiškai iš konceptualiojo lygio (vartotojo arba verslo procesų lygio) visuotinio duomenų modelio – EER diagramų arba klasių diagramos. Naujas visuotinis modelis projektuojamas naudojant palikuoninės KIS projektą, jeigu jis toks buvo ar išliko, ir įvertinant pokyčius organizacijos funkcinėje veikloje. KIS reinžinerija suprantama kaip nuoseklus perėjimas (2.2.1 pav.) nuo esamos KIS prie naujos: čia paliekami spęsti aktualūs uždaviniai, atitinkantys senus vartotojų poreikius, ir prie jų prijungiami nauji uždaviniai, reikalingi naujiems poreikiams tenkinti.



2.1.1 pav. KIS reinžinerijos procesas, pagrįstas veiklos procesų reinžinerija

Išplėstine komunikacine kilpa aprašytą verslo transakciją galima naudoti kaip funkcinių reikalavimų aprašymo vienetą. Naujų funkcinių reikalavimų įvedimas paprastai reikalauja papildomų duomenų struktūrų. Po kiekvieno naujo kompiuterizuojamo proceso įvedimo tenka spęsti klausimą, ar pakankami yra eksploatuojamos KIS DB informaciniai ištekliai. Jei informacinis turinys nepakankamas, tenka projektuoti papildomas duomenų struktūras. Naujos informacinės technologijos skatina kurti naujas DB, tačiau palikuoninėse bazėse saugomų duomenų išliekamoji vertė neleidžia jų atsisakyti. Todėl projektuojant KIS tenka numatyti, kad bus naudojamosi naujomis ir senomis duomenų bazėmis. Palikuonines duomenų bazes išsaugoti vertinga ir tuo požiūriu, kad galima palikti eksploatuoti nepakitusių uždavinius. Kelių DBVS (arba tos pačios, bet skirtingų versijų), kurios palaiko tokius pat ar skirtingus duomenų modelius, panaudojimas sąlygoja nevienalyčių DB projektavimo metodų atsiradimą. Pagrindinį vaidmenį tiesioginėje KIS inžinerijoje įgauna nevienalyčių DB komponentų integravimas. Nuolatinės funkcinių reikalavimų kaitos sąlygomis ypač aktualus tampa nevienalyčių DB komponentų projektavimo ir integravimo CASE priemonių sudarymas. Šios klasės CASE priemonės galima priskirti prie technologinių priemonių, skirtų sparčiam taikomųjų programų kūrimui užtikrinti. Jos naudojamos

KIS reinžinerijoje, tiesiogiai projektuojant naujus trūkstamus ar replikuojamus DB komponentus pagal informaciją apie funkcinių reikalavimų papildymus. Nevienalytės DB, turinčios tiek sąryšinių, tiek objektinių komponentų, vadinamos hibridinėmis duomenų bazėmis [3].

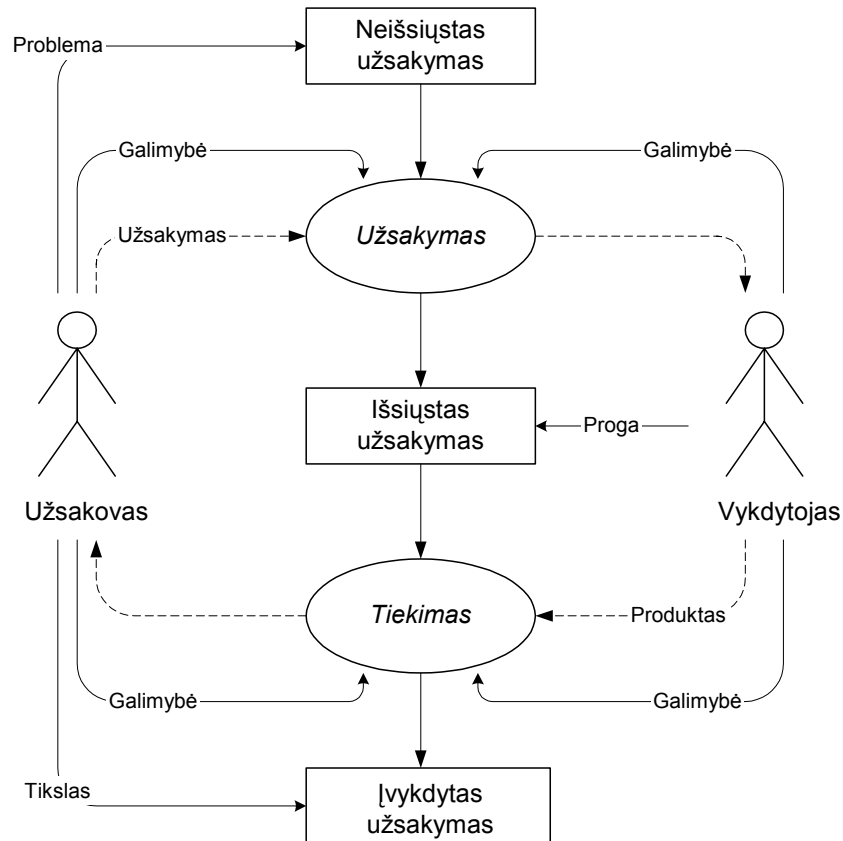
Hibridinės DB spartaus kūrimo metodą galima apibrėžti tokiais žingsniais:

- verslo transakcijų, apibrėžiančių naujus funkcinius reikalavimus, identifikavimas ir aprašymas tipinėmis pragmatiškai motyvuotomis komunikacinėmis kilpomis;
- komunikacinės kilpos konceptų (duomenų objektų ir jų atributų vardų) išskyrimas ir normalizuoto duomenų poschemio sudarymas;
- Poschemio (naujo hibridinės DB komponento), atitinkančio naujus informacinius poreikius, integravimas į visuotinę duomenų schemą:
 - nominalo ryšių tarp komponentinio poschemio ir visuotinės schemos elementų identifikavimas;
 - struktūrinių (dalies/visumos arba poaibio) priklausomybių išskyrimas;
 - objektų tipų konfliktų ir priklausomybių nesuderinamumo analizė ir likvidavimas;
 - poschemio ir visuotinės schemos duomenų objektų semantinės sanklotos (objektų duomenų sankaupos) identifikavimas ir visuotinės schemos papildymas naujais duomenų objektais;
- duomenų sankaupos (angl. *data cluster*) vaizdinio schemos sudarymas;
- naujos verslo transakcijos pavaizdavimas hibridinės DB transakcijomis.

2.3. Pragmatinės priklausomybės komunikacinių veiksmų kilpose

2.3.1 pav. pateiktas tipinės elementariosios komunikacinės kilpos pavyzdys, kurioje aktorius iniciatorius *Užsakovas* veiksmu (inicijuotu procesu) *Užsakymas* siunčia srautą *Užsakymas* kitam aktoriui *Vykdytojas*. Šis reakcijos veiksmas (reakcijos procesu) *Tiekimas* grąžina srautą *Produktas*. Inicijuotas procesas *Užsakymas* perkelia užsakymo objektą iš būsenos *Neišsiųstas užsakymas* į būseną *Išsiųstas užsakymas*, o reakcijos procesas *Tiekimas* – užsakymo objektą iš būsenos *Išsiųstas užsakymas* į būseną *Įvykdytas užsakymas*. Dviejų aktorių – *Užsakovas* ir *Vykdytojas* – komunikacinė kilpa bus pragmatiškai motyvuota, jei tenkins procesų gyvybingumo kriterijų [3]. Naudojantis pragmatinėmis priklausomybėmis, šį kriterijų galima formuluoti taip: elementarioji komunikacinė kilpa „**iniciatorius:Užsakovas** → **inicijuotas procesas:Užsakymas** → **recipientas:Vykdytojas** → **reakcijos procesas:Tiekimas** → **iniciatorius:Užsakovas**” bus gyvybinga, jei **recipientas** turi **progą reakcijos procesu** veiklos objekto **tarpinę būseną Išsiųstas užsakymas** perkelti į **galinę būseną Įvykdytas užsakymas** ir tokiais savo

reakcijos veiksmais jis teigiamai veikia **iniciatoriaus problemos** *Neišsiųstas užsakymas* sprendimą bei leidžia pasiekti **iniciatoriaus tikslą** *Ivykdytas užsakymas*. Abu aktoriai – **iniciatorius** *Užsakovas* ir **recipientas** *Vykdytojas*, vienoje kilpoje susieti bendrais **procesais** *Užsakymas* ir *Tiekimas*, turi galimybę šiuos procesus įgyvendinti.



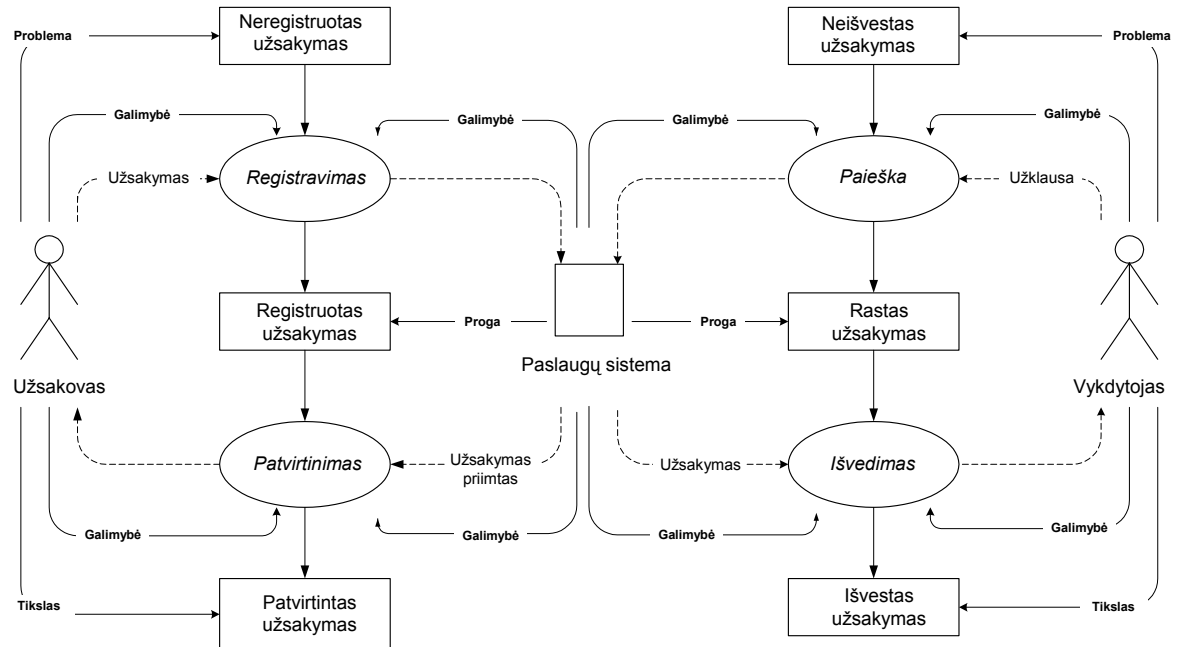
2.3.1 pav. Elementariosios tipinės komunikacinių veiksmų kilpos pavyzdys

Užsakovo siunčiamas srautas yra duomenų srautas ir procesas *Užsakymas* yra duomenų apdorojimo arba informacinis procesas. Sakykime, šis procesas yra kompiuterizavimo objektas ir sudaroma paslaugas teikianti sistema (2.3.2 pav.). Ji pasižymi tuo, kad:

- turi fizinę atmintį, gali atsiminti siunčiamą pranešimą ir kausti siunčiamus duomenis;
- sudaro sistemine programine ir technine duomenų priėmimo, kaupimo, persiuntimo aplinką;
- turi programine **galimybę** kartu su išoriniais aktoariais *Užsakovas* ir *Vartotojas* vykdyti vartotojų apibrėžtus duomenų apdorojimo procesus – *Registravimas*, *Patvirtinimas*, *Paieška*, *Išvedimas*.

Pavyzdyje parodyti keturi procesai sudaro kompiuterizuotai IS keliamą funkcinį reikalavimą. Šie procesai detalizuoja organizacijos funkcinį reikalavimą, kuris 1.3 paveiksle buvo pavadintas proceso vardu *Užsakymas*. Paslaugų sistema tampa aktoriumi tarpininku tarp *Užsakovo* ir *Vykdytojo*, o viena tipinė

komunikacinė kilpa pakeičiama dviem tipinėmis komunikacinėmis kilpomis. Šios kilpos taip pat turi būti pragmatiškai motyvuotos, t. y. užsakovas reikalauja iš KIS projektuotojo, o šis įsipareigoja, kad nauja informacijos sistema tenkins funkcinis reikalavimus.

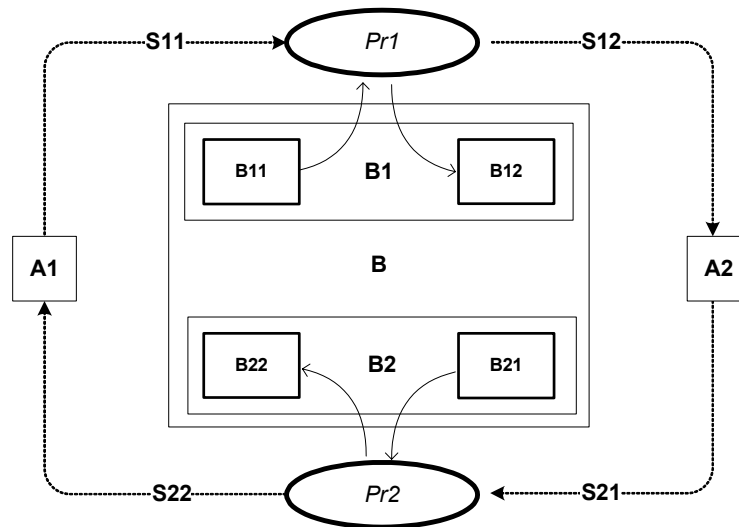


2.3.2 pav. Komunikacinių veiksmų kilpa su paslaugų sistema

Elementarioji komunikacinė kilpa procesų modelyje motyvuojama standartiniu būdu, t. y. susiejant aktorius su objektų būsenomis ir procesais pragmatinėmis priklausomybėmis, parodytomis 2.3.1 pav. KIS reinžinerijoje elementariosios kilpos procesas (vienas iš dviejų arba abu), nustatantis funkcinį reikalavimą, gali būti detalizuotas. Detalizavimo metu įvedamas aktorius tarpininkas, kuris jungiamas į dvi naujas standartinio tipo komunikacines kilpas, o pastarosios vėl motyvuojamos standartiniu būdu. Suprantama, kad standartinės pragmatinės priklausomybės po organizacijos procesų reinžinerijos, kai yra pasiektas organizacijos aktorių „balansas“, gali būti nespécifikuojamos. 2.3.2 pav. buvo parodytas proceso dekompozicijos pavyzdys, kai įvestas naujas aktorius tarpininkas yra kompiuterinė paslaugų sistema. Šis aktorius abiejose kilpose yra recipientas ir neturi nei tikslų, nei problemų. Ankstesnis aktorius recipientas Vykdytojas tampa aktoriumi iniciatoriumi tarpininko atžvilgiu. Gautų dviejų naujų kilpų pragmatinė motyvacija išliko standartinė. Paslaugų sistemą galima įjungti į kitas kilpas, kuriose ji gali teikti paslaugas kitiems aktoriams. Savo ruožtu ta pati paslaugų sistema gali būti įjungta į tokias naujas kilpas, kuriose ji traktuojama kaip paslaugas užsakanti sistema, kuri keisis pranešimais su kita paslaugas teikiančia sistema. Taigi procesams, kuriems taikomos šiuolaikinės paslaugų technologijos ir architektūros (pavyzdžiui, dotNET), galima duoti standartinę pragmatinę interpretaciją.

2.4. Išplėstųjų komunikacinių kilpų schemų sudarymas

2.4.1 pav. parodytas bendrasis išplėstosios komunikacinės kilpos atvejis. Kiekviena komunikacinė kilpa susideda iš dviejų aktorių – užsakovo A1 ir vykdytojo A2, srautų S11-S22 ir dviejų procesų – užsakovo veiksmų Pr1 ir vykdytojo veiksmų Pr2. Kilpos dinamika nustatoma dviem perėjimais iš būsenos B11 ir B21 į būsenas B12 ir B22. Komunikacinė kilpa vadinama išplėstąja, kadangi kilpos atžvilgiu ji turi papildomas duomenų struktūras bei priklausomybes (be abipusio apibendrinimo, dar naudojamos agregavimo, gyvavimo ir poaibio priklausomybės).



2.4.1 pav. Išplėstoji komunikacinių veiksmų kilpa su apibendrinimo ryšiais tarp būsenų B11, B12, B21 ir B22.

Jeigu kompiuterizuotos sistemos projektuojamos komunikacinių kilpų analizės metodu, galima užtikrinti tam tikrus semantinio integralumo kriterijus:

- Darnos (angl. *coherence*) kriterijus reiškia, kad galinės būsenos turi būti suderintos su aktorių tikslais.
- Pilnumo (angl. *completeness*). Nepilnumą galima identifikuoti iš probleminių būsenų, kurios rodo poreikius tobulinti kompiuterizuotą informacijos sistemą.
- Gyvybingumo (angl. *viability*) kriterijus išreiškiamas pragmatinėmis priklausomybėmis: komunikacinė kilpa yra gyvybinga, jei

$$A1 \xrightarrow{p} B11, A2 \xrightarrow{o} B22, A2 \xrightarrow{p} B21, A1 \xrightarrow{g} B12,$$

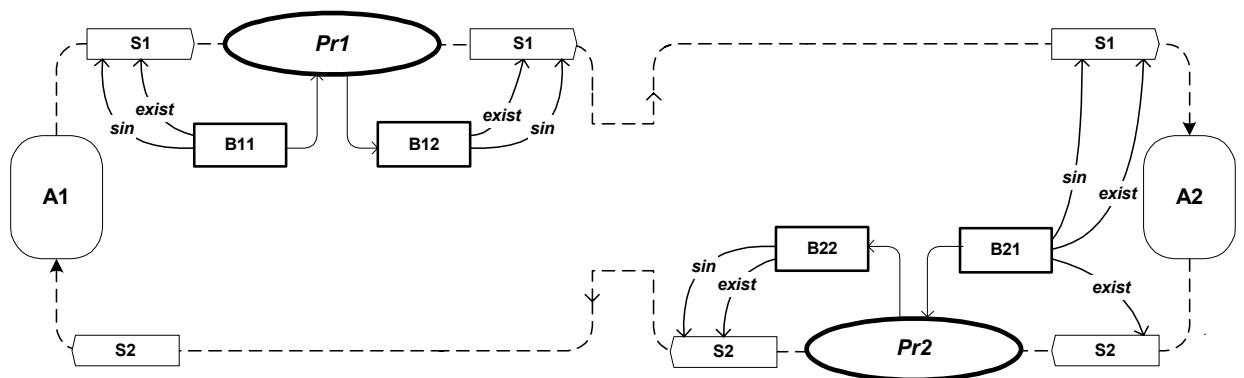
$$\text{tuomet } B11 \xrightarrow{-} B12, B22 \xrightarrow{+} B12, B22 \xrightarrow{-} B11, B12 \xrightarrow{-} B21;$$

čia simbolis \longrightarrow žymi pragmatines priklausomybes: \xrightarrow{g} reiškia tikslą, \xrightarrow{p} – problema, \xrightarrow{o} – proga, $\xrightarrow{+}$ – teigiamos įtakos priklausomybę ir $\xrightarrow{-}$ – neigiamos įtakos priklausomybę. Kitaip tariant, komunikacinė kilpa gyvybinga, jei recipientas problema neigiamai veikia iniciatoriaus problemą ir recipientas gali įvykdyti reakcijos veiksmą, kurio rezultatas teigiamai veikia tiek recipientą, tiek iniciatoriaus tikslus.

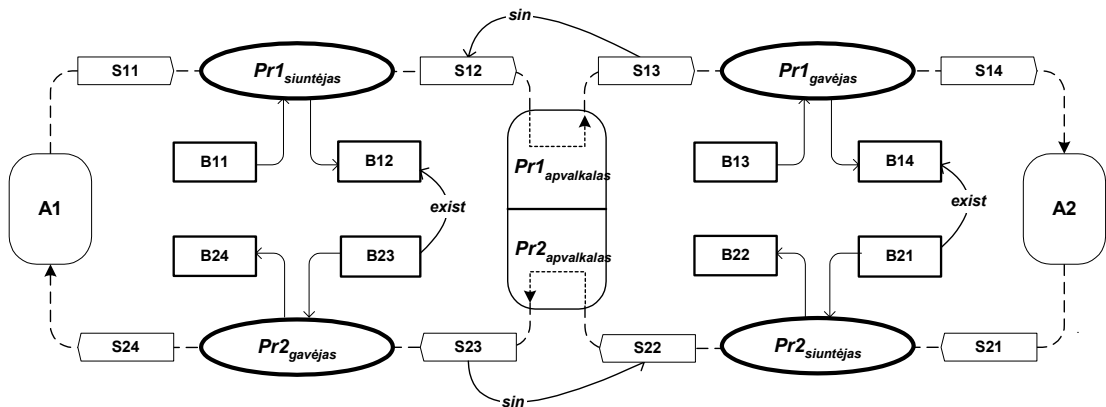
- Glaudumo (angl. *cohesion*) kriterijus nustato reikalavimus komunikacinių srautų valdomų būsenų atvaizdavimui duomenų bazėje. Ryšiai tarp būsenų ir komunikacinių srautų struktūros palaikomi atvaizduojant būsenų pokyčius į duomenų bazės egzempliorių atributų aibes.

2.4.2 pav. parodytas srauto S1 persiuntimas elementariojoje vienalytėje kilpoje, kurioje užsakovo siunčiami ir gavėjo gaunami srautai sutampa ir srauto gavėjas paprasčiausiai formuoja pranešimą apie srauto gavimą. Bendruoju atveju siunčiami ir gaunami srautai nesutampa.

Tarkime, nustatytas naujas funkcinis reikalavimas, kurį reikia realizuoti objektinėje aplinkoje ir jo informacinį turinį gali dengti palikuoninė sąryšinė DB. 2.4.3 pav. parodyta, kaip informaciniai procesai Pr1 ir Pr2 išskaidomi nevienalytėje kilpoje, kurioje užsakovas A1 sąveikauja su objektine aplinka, o vykdytojas A2 – su sąryšinės DB aplinka. Tiek užsakovo A1 dalyje Pr1_{siuntėjas}-Pr2_{gavėjas}, tiek vykdytojo A2 dalyje Pr1_{gavėjas}-Pr2_{siuntėjas} informacija apie verslo objektus per visą verslo transakcijos ciklą išsaugoma vietinėse duomenų bazėse. Pranešimų keitimą vykdo paslaugų sistema, kuri formuoja identiško turinio pranešimus: S12, S13 bei S22, S23.

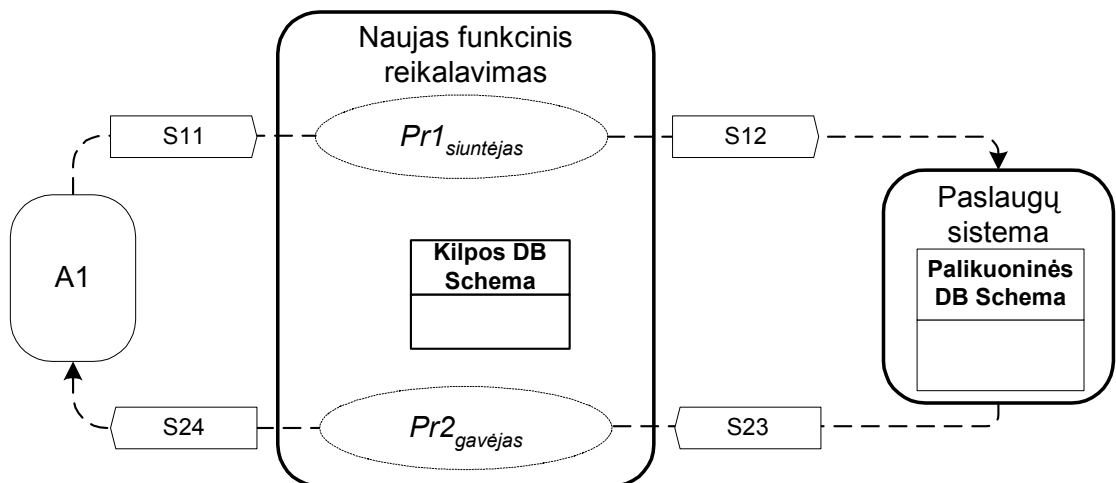


2.4.2 pav. Srauto S1 persiuntimas elementariojoje vienalytėje kilpoje; formuojamas pranešimas S2 „Srautą S1 gavau“



2.4.3 pav. Nevienalytė komunikacinė kilpa su paslaugų sistema

Schemiškai naujų funkcinių reikalavimų įvedimą galima pavaizduoti 2.4.4 paveikslu. Palikuoninė sąryšinė DB su objektiniu apvalkalu traktuojama kaip paslaugų sistema. Įvedant naują aktorius A1 funkcinį reikalavimą, jis pateikiamas sudėtinu procesu $Pr1_{siuntėjas} - Pr2_{gavėjas}$ ir kilpos objektinės DB schema; paslaugos sistema Pr3 praplečiama sudėtinu procesu $Pr1_{gavėjas} - Pr2_{siuntėjas}$, o palikuoninės sąryšinės duomenų bazės schema papildoma trūkstantomis struktūromis.



2.4.4 pav. Aktoriaus A1 naujas funkcinis reikalavimas, pavaizduotas sudėtinu procesu

Palikuoninės sąryšinės DB schema ir naują funkcinį reikalavimą atitinkanti objektinė duomenų schema IS reinžinerijos metu transformuojamos į visuotinio sąryšinio-objektinio modelio schemas. Kilpos DB ir palikuoninės DB schemų integravimo metu sudaroma duomenų objektų sanauja (angl. *cluster*) ir suformuojamos trūkstantos duomenų objektų struktūros. Integruotoje schemoje sankirtos duomenų sanaujos (angl. *cluster*) vaizdiniai turi sutapti.

3. XML ĮRANKIŲ APŽVALGA

3.1. XML srautų panaudojimo aktualumas

XML (angl. *eXtensible Markup Language*) buvo sukurta 1996 metais. Ją sukūrė XML darbo grupė, padedama W3C (angl. *World Wide Web Consortium*) konsorciumo.

XML aprašo duomenų objektų klasę, vadinamą XML dokumentu ir iš dalies aprašo šiuos dokumentus apdorojančių kompiuterinių programų elgesį. XML yra SGML (angl. *Standart Generalized Markup Language*) apribota forma. Savo struktūra XML dokumentai atitinka SGML dokumentus.

XML dokumentai yra sudaryti iš elementų, vadinamų esybėmis, kuriose yra išnagrinėti arba neišnagrinėti duomenys. Išnagrinėti duomenys yra sudaryti iš simbolių. Vieni simbolių junginiai sudaro duomenis, kiti junginiai sudaro žymes. Žymėmis yra koduojama dokumento sandara ir loginė struktūra. XML pateikia sandaros ir loginės struktūros apribojimų pritaikymo mechanizmą.

XML yra naudojama kaip žymių kalbų sudarymo šablonas:

- XML naudoja neribojamą žymių skaičių, t.y. skirtingai nei HTML nėra jokių apribojimų jų kiekiui – vartotojas pats gali susikurti žymes atsižvelgdamas į informacijos pobūdį;
- Kiekviena XML kalba yra skirta savo taikomajai sričiai, tačiau kalbos pasižymi bendromis savybėmis;
- Dokumentams apdoroti yra skirti bendri įrankiai.

Galima būtų išskirti šiuos pagrindinius XML kalbos bruožus:

- Atskirti sintaksę nuo semantikos. Tai yra reikalinga todėl, kad yra kuriamas bendras informacijos struktūrizavimo mechanizmas (naršyklės vaizduojamos semantikos yra apibrėžiamos stilių lentelėmis);
- Leisti kurti žymes bet kokiai įsivaizduojamai taikomųjų programų sričiai;
- Palaikyti internacionalizaciją ir nepriklausomybę nuo platformų;
- Ateityje vaizduoti struktūrizuotą informaciją, kuri apimtų ir duomenų bazes.

XML paskirtis ir privalumai:

- XML skirta keitimuisi duomenimis;
- XML skirta keistis finansine informacija, t. y. yra naudojama daugumoje „verslas verslui“ taikomųjų programų;
- XML gali būti naudojama duomenų paskirstymui;
- XML gali būti skirta duomenų saugojimui;

- XML gali padaryti duomenis labiau prieinamus.

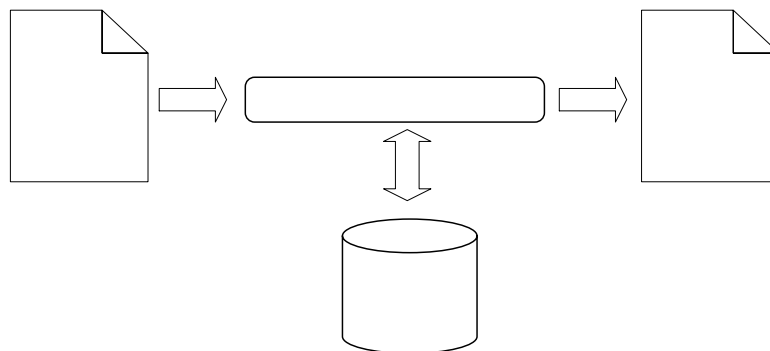
Remiantis aukščiau išvardintais XML požymiais galima daryti išvadą, jog XML kalba labiausiai tinka duomenų keitimuisi tarp heterogeninių duomenų bazių [9]. Tačiau šis procesas nėra toks paprastas, nes reikia atsižvelgti į įvairių sistemų funkcinius reikalavimus. Reikia sukurti bendrą šabloną, kuriuo remiantis būtų įmanoma atlikti keitimąsi duomenimis tarp įvairių sistemų. Naudojant XML šiuo šablonu patogiau naudoti XML schemas.

3.2. XML technologijos duomenų bazių valdymo sistemose

3.2.1. Oracle duomenų bazė ir XML panaudojimo galimybės

Oracle turi XML failų nagrinėjimo priemones (angl. *parsers*) Java, C, C++ ir PL/SQL kalboms, todėl programuojant šiomis kalbomis XML panaudojimas yra lengvesnis [8].

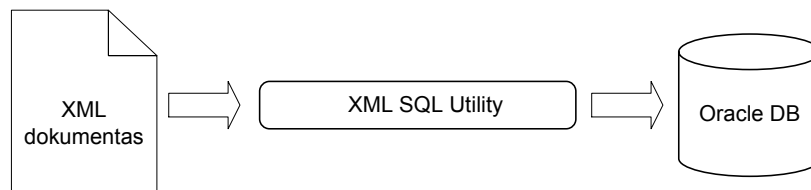
Naudojant XML SQL Utility galima vykdyti SQL užklausas ir gauti rezultatus XML dokumentą iš suformuotų užklausų rezultatų.



3.2.1 pav. XML dokumento formavimas naudojant XML SQL Utility

Gauto XML dokumento struktūra atitinka Oracle DB struktūrą, kuri buvo nurodyta formuojant SQL užklausą. Duomenų bazės stulpelių pavadinimai atitinka XML elementų grupių pavadinimus, o reikšmės yra priskiriamos vidiniams elementams.

Taip pat naudojantis XML SQL Utility galima ir surašyti XML duomenis į Oracle duomenų bazę.



3.2.2 pav. XML dokumento duomenų surašymas į Oracle DB

Saugant duomenis iš XML dokumento į Oracle duomenų bazę katu išsaugoma ir dokumento struktūra. XML dokumento elementų pavadinimai atitinka duomenų bazės lentelių ir lentelių stulpelių pavadinimus, o reikšmės yra įrašomos į atitinkamus lentelės laukus.

3.2.2. MS SQL Server duomenų bazė ir XML panaudojimo galimybės

MS SQL Server duomenų bazės duomenis į XML galima perduoti trimis būdais:

- Baigiant SELECT sakinį FOR XML;
- XPath queries priemonių pagalba, kurios komentuoja XML schemas;
- OpenXML funkcija procedūrose.

OpenXML funkcija dažniausiai naudojama atvaizduoti XML dokumento duomenis į sąryšinės bazės lenteles.

MS SQL Server yra galimos tokios XML savybės:

- HTTP pasiekiamumas (rašant SQL sakinį naršyklės adresu eilutėje);
- Formuoti XML galima SELECT sakinio pagalba (FOR XML);
- XML views (nurodoma kaip sąryšiniai duomenys turi būti atvaizduoti XML);
- XPath queries (nurodoma kaip duomenys turi būti paimti iš DB);
- OpenXML (nurodo, kaip duomenys turi būti atvaizduoti XML ir suteikia galimybę XML failo duomenis atvaizduoti duomenų bazėje lentelėse);
- OLE DB ir ADO pasiekiamumas (naudojama sąsajai su sąryšinėmis DB formuojant XML iš DB ir surašant XML duomenis į DB).

3.2.3. Oracle ir MS SQL Server teikiamų XML savybių palyginimas

3.2.3 lentelė

Oracle ir MS SQL Server teikiamos XML savybės

Savybės	SQL Server 2000	Oracle
Deklaratyvus žymėjimas	XML Views	Neturi specifinių savybių.

<p>laisvai susietų biznio sistemų (ši savybė suteikia dvikryptį žymėjimą tarp XML ir sąryšinių duomenų vaizdų).</p> <p>SQL Server 2000 turi integruotas XML savybes, kurios leidžia priėjimą prie sistemos su minimaliu programavimu.</p> <p>Oracle neturi integruotų XML specifinių savybių.</p>	<p>XPath support</p> <p>Transact-SQL FOR XML išplėtimai</p> <p>Transact-SQL OPENXML išplėtimai</p>	<p>Reikia rašyti papildomas programas.</p>
<p>Vietiniai XML išplėtimai skirti SQL ir leidžiantys vartotojui vykdyti aplikaciją greitai ir be papildomo programavimo.</p> <p>SQL Server 2000 teikia išplėtimus SQL kalbai, kurie gali būti panaudoti gražinti XML duomenis iš standartinių SQL užklausų.</p> <p>Oracle teikia serverio pagalbinius įrankius, kuriuos vykdant neišplečiama SQL ir reikalaujama papildomo programavimo gražinant XML duomenis.</p>	<p>Transact-SQL FOR XML išplėtimai</p> <p>Transact-SQL OPENXML išplėtimai</p>	<p>Neturi specifinių savybių. Reikia rašyti papildomas programas naudojant XSQL Utility.</p>
<p>Įvairūs XML prieigos</p>	<p>HTTP prieiga</p>	<p>HTTP prieiga</p>

<p>metodai, kurie palaiko lankstumą, leidžiantį programuotojams pasiekti XML duomenis per tinklo prisijungimus.</p> <p>SQL Server 2000 ir Oracle palaiko įvairius metodus leidžiančius pasiekti XML duomenis, esančius duomenų bazėse.</p>	<p>OLE DB/ADO prieiga JDBC</p>	<p>OLE DB/ADO prieiga JDBC</p>
<p>XML šablonai</p> <p>SQL Server ir Oracle palaiko išsaugotus XML užklausų šablonus serveryje su laisvai pasirenkamais parametrais.</p>	<p>URL/HTTP prieiga</p> <p>XML šablonai</p>	<p>URL/HTTP prieiga</p> <p>XSQL šablonai</p>
<p>Saugi Web aplikacijų prieiga prie XML duomenų</p> <p>SQL Server 2000 palaiko saugumą, kuris gali būti valdomas individualios lentelės lygyje.</p> <p>Oracle palaiko apribotą saugumą tik duomenų bazės vartotojo, bet ne individualios lentelės lygio.</p>	<p>Duomenų bazės vartotojo lygmens saugumas</p> <p>Duomenų bazės objekto lygmens saugumas</p>	<p>Duomenų bazės vartotojo lygmens saugumas</p>
<p>Galimybės detalesnei paieškai sudėtingiems XML dokumentams.</p>	<p>Pilna teksto paieška</p>	<p>Tarpinė teksto paieška su segmentų savybėmis.</p>

SQL Server 2000 ir Oracle teikia duomenų bazės palaikomą sudėtingų tekstinių dokumentų paiešką apimančią ir XML.		
XML failų nagrinėjimo priemonės (angl. <i>parsers</i>) (skirti programuotojams, kurie siekia didesnės galios ir lankstumo, kai kuriamos XML aplikacijos).	MSXML COM priėjimo parserį palaiko daugelis kalbų	Oracle parserius palaiko daugelis kalbų

Microsoft SQL Server 2000 siūlo XML savybes daugiausia skirtas Web ir duomenų bazių programuotojams. Web programuotojams nereikia mokintis duomenų bazių programavimo subtilybių norint pasinaudoti SQL Server 2000 XML privalumais, nes SQL Server teikia standartines XML konstrukcijas. Taip pat, duomenų bazių programuotojams nereikia išmanyti XML programavimo subtilybių, nes jie gali naudotis FOR XML ar OpenXML siekdami manipuluoti XML duomenimis.

Oracle labiau skirtas naudoti patyrusiems programuotojams. Norint pasiekti sprendžiamų uždavinių rezultatą tenka naudotis keliomis priemonėmis arba kurti specifinius įrankius.

3.2.4. Replikavimo priemonė MS SQL serveryje

MS SQL serverio replikavimas leidžia persiųsti duomenis iš vieno serverio į kitą serverį, net jei kitas serveris yra nutolęs. Yra daug programinių paketų, kurie leidžia persiųsti duomenis, tačiau pačio SQL serverio teikiama tokia galimybė, kai duomenis persiuntinėjami iš vieno SQL serverio į kitą SQL serverį, yra labiau patikima.

Replikavimo galimybė

Pagrindinės savybės, kurias reikia žinoti norint pasinaudoti SQL serverio teikiamu replikavimo įrankiu:

- sujungiamumas (angl. *connectivity*) (ar yra galimybė pasiekti kitą serverį);
- savarankiškumas (angl. *autonomy*) (kiek serveriai bus nepriklausomi vienas nuo kito);
- duomenys ir schema (angl. *data and schema needs*) (kokie duomenys bus siunčiami);

- vėlavimo laikas (angl. *latency*) (kiek dažnai reikia atnaujinti duomenis);
- lyderio nurodymas (angl. *determine the ladder*) (kaip duomenys bus atnaujinami jei tuo pačiu metu du vartotojai atnaujiną duomenis. Nurodoma, ar bus suteikiama kuriam nors vartotojui pirmenybė).

Replikavimo architektūra

Replikavimo architektūra yra paprasta. Aibė elementų, kuriuos replikuojame, yra vadinama publikacija (angl. *publication*). Galimi replikavimo būdai:

- visos lentelės (angl. *entire table*) (replikuojama visa lentelė);
- vertikalios padalijimo (angl. *vertical partition*) (pasirenkami lentelės stulpeliai, kuriuos norime replikuoti);
- horizontalios padalijimo (angl. *horizontal partition*) (leidžia replikuoti tik išrinktus duomenis);
- išsaugotos procedūros rezultatų (angl. *stored procedure results*) (replikuojama išsaugota procedūra);
- vaizdų (angl. *views*) (replikuojamas vaizdų turinys).

Galimi SQL serverio replikavimo tipai:

- „snapshot“ (leidžiamas didžiausias savarankiškumas);
- „merge“ (sudėtingos logikos bet turi galimybę atstatyti klaidas);
- transakcijų (siūlomi momentiniai atnaujinimai, neleidžiamas savarankiškumas).

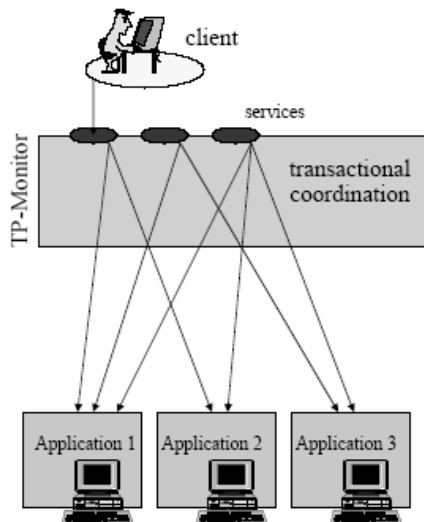
Šiame darbe dirbama su XML ir MS SQL Server ir Oracle duomenų bazėmis, o MS SQL siūloma replikavimo priemonė nenaudoja XML ir ji geriausiai tinkama ir labiausiai patikima, kai duomenys persiuntinėjami iš vieno SQL serverio į kitą SQL serverį, todėl reikia ieškoti papildomų sprendimų integruojant MS SQL Server ir Oracle duomenų bazes ir naudojant XML.

4. TRANSAKCIJŲ SERVERIAI

4.1. Transakcijų serverio apibrėžimas

Transakcijų serveriai arba transakcijų valdymo programos (angl. *transaction processing monitors*) yra tarpinė programinė įranga (angl. *middleware*) veikianti tarp klientų ir serverių. Labiausiai paplitę yra šie transakcijų serveriai: IBM CICS, Encina, Microsoft Transaction Server, BEA Tuxedo [11]. Transakcijų serverių funkcionalumas neapsiriboja vien tik paprastu tarpininkavimu tarp kliento ir serverio. Transakcijų serveris leidžia kurti sąsajas su daugeliu paskirstytų programų kartu palaikant transakcines

savybes (4.1.1 pav.). Jis išplečia transakcijų valdymo galimybes už duomenų bazės ribų ir kartu teikia priemones ir įrankius kurti taikomas programas, kuriose būtų užtikrintas sklandus transakcijų vykdymas.

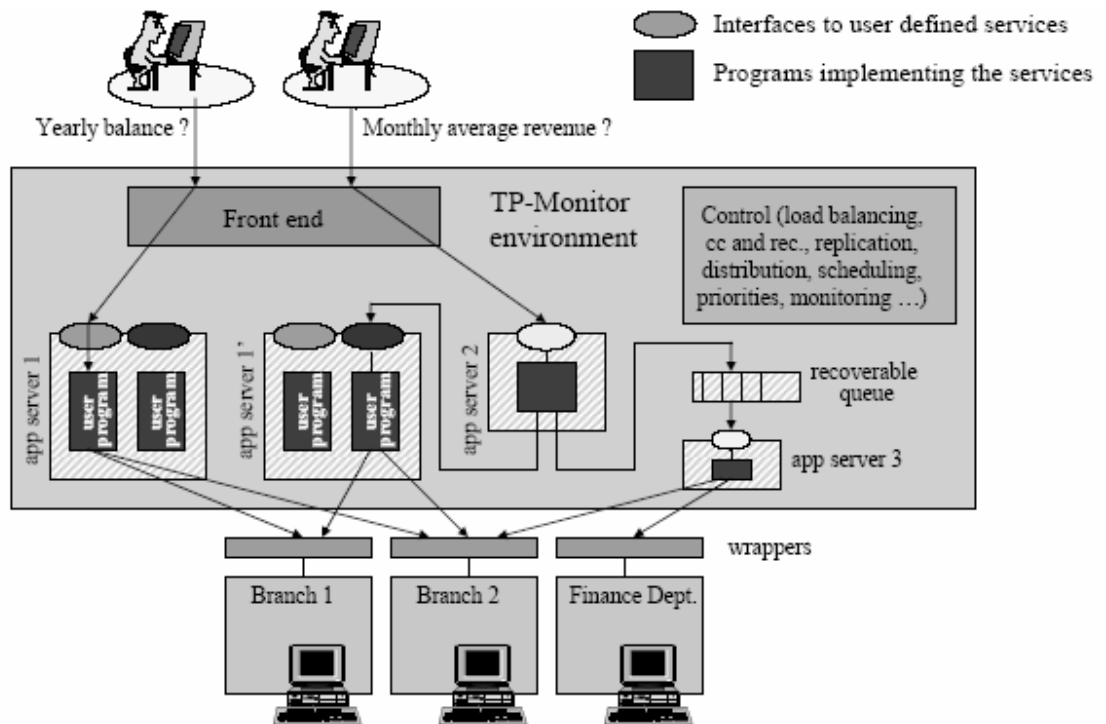


4.1.1 pav. Transakcijų serveris, jungiantis klientus su paskirstytais programomis

Transakcijų serverių atsiradimą sąlygojo tai, kad operacinės sistemos nebuvo skirtos transakcijų apdorojimui. Pažymėtina, kad bendras transakcijų serverių funkcionalumas nėra tiksliai apibrėžtas ir dažnai priklauso nuo konkrečios sistemos. Taip yra todėl, kad transakcijų serveriai buvo kuriami, kaip antro lygmens operacinės sistemos ant esamų operacinių sistemų. Bendru atveju transakcijų serveris – tai integravimo įrankis, kuris leidžia projektuotojams apjungti skirtingus sistemos komponentus, panaudojant įvairias pagalbines priemones. Panaudojant šias priemones integravimas tampa daug paprastesnis, nes didžiąją dalį reikalaujamo funkcionalumo palaiko pats transakcijų serveris.

Transakcijų serveris sprendžia tokias problemas, kaip daugelio vartotojų prieiga prie nevienalyčių paskirstytų duomenų šaltinių, tiksliai apibrėžtų sąsajų teikimas bei transakcinių savybių užtikrinimas. Transakcijų serveriai taip pat atlieka resursų rakinimo, resursų prieigos paskirstymo (angl. *scheduling*), įvykių žurnalo vedimo (angl. *logging*) bei atstatymo funkcijas.

Bendrinė transakcijų serverio architektūra pateikta 4.1.2 pav. Klientai per bendrą sąsają jungiasi prie transakcijų serverio, kuris koordinuoja daugelio taikomųjų programų serverių veiksmus.



4.1.2 pav. Bendrinė transakcijų serverio architektūra

Pagrindinės transakcijų serverio teikiamos paslaugos yra šios:

- Transakcinis nutolusių procedūrų iškviatimas (angl. *remote procedure call*).
- Transakcijų valdymas: atliekamas dviejų fazių įrašymas ir užtikrinama atstatymo galimybė.
- Įrašų žurnalo (angl. *log*) valdymas: fiksuojami visi transakcijų metu atlikti pakeitimai, tam, kad nesėkmės atveju sistemą būtų galima atstatyti.
- Rakinimo valdymas: prieigos prie bendrų duomenų reguliavimas.

Papildomos transakcijų serverių teikiamos paslaugos:

- Serverių valdymas ir administravimas: serverių paleidimas, sustabdymas, apkrovimo paskirstymas.
- Autentifikacija ir autorizacija: tikrinama ar vartotojas gali iškviesti paslaugą, iš tam tikro terminalo, tam tikru laiku ir su tam tikru parametru rinkiniu.
- Transakcijų eilės: asinchroninei sąveikai tarp komponentų.

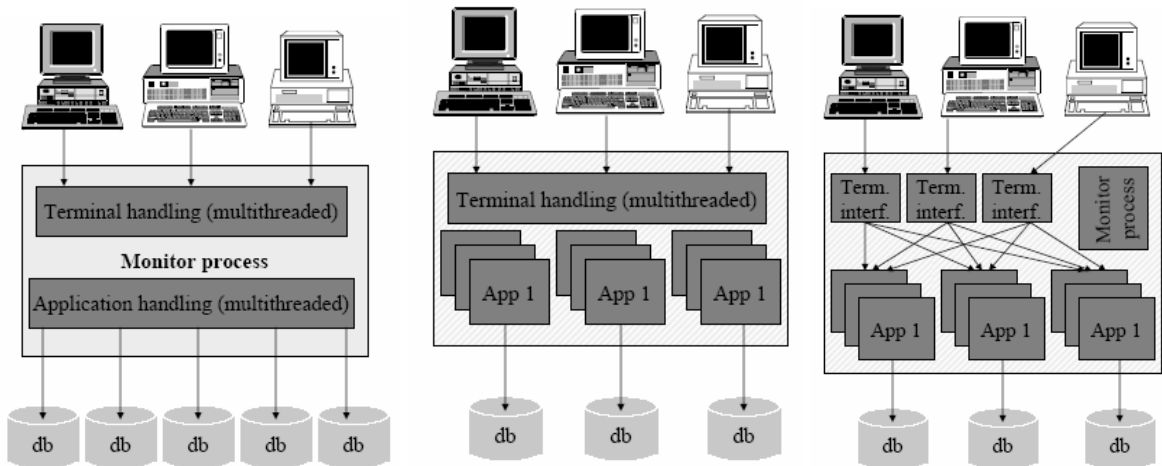
4.2. Transakcijų serverių struktūra

Tam kad užtikrintų efektyvų transakcijų vykdymą transakcijų serveriai perima dalį operacinės sistemos funkcijų. Transakcijų serverio struktūra priklauso nuo to, kokia operacine sistema jis paremtas.

Monolitinės struktūros transakcijų serveriuose (4.2.1 pav.) visas funkcionalumas yra realizuotas viename procese. Projektavimo požiūriu tokia sistema yra paprastesnė, tačiau kartu yra smarkiai apribota.

Dviejų lygių struktūroje funkcionalumas yra atskiriamas į skirtingus lygius. Vienas skirtas terminalų valdymui, kitas resursų valdymui. Projektavimo požiūriu sistema išlieka gana paprasta, ir kartu pasiekiamas didesnis našumas, bei lankstumas.

Multiprocesorinės struktūros atveju funkcionalumas padalinamas tarp daugelio nepriklausomų paskirstytų procesų.



4.2.1 pav. Monolitinė, dviejų lygių ir multiprocesorinė transakcijų serverių struktūros

Transakcijų serverio ir operacinės sistemos sąsaja parodytos 4.2.1 lentelėje.

4. 2. 1 lentelė

Transakcijų serverio ir operacinės sistemos sąsaja.

Transakcijų serverio komponentai bei vykdomi veiksmai pažymėti ryškiau.

	Skaičiavimai	Duomenys	Komunikacijos
Transakcijų serverio paslaugos	administravimo sąsaja, konfigūravimo įrankiai, apkrovimų paskirstymas, programavimo įrankiai	duomenų bazės, duomenų atstatymas, resursų valdymas	vardų serveris, vartotojų prieiga
Vidinės sisteminės transakcijų serverio paslaugos	transakcijų identifikatoriai serverio klases resursų paskirstymas autentifikacija	transakcijų valdymas, įrašų žurnalai, eilės, transakcijų failai	transakcinis nutolusių procedūrų iškvietimas, transakcinės sesijos, nutolusių procedūrų iškvietimas
Operacinė sistema	procesai-gijos, adresų erdvė,	saugykla, failų sistema,	sesijos, autentifikacija

	resursų paskirstymas	blokai, puslapiai, failų apsauga	
Aparatinė įranga	Procesorius	Atmintis	Tinklo įranga

5. KOMUNIKACINIŲ KILPŲ METODO TAIKYMAS

5.1. Verslo transakcijų realizavimas duomenų bazių transakcijomis

Verslo transakcija specifikuojama komunikacine kilpa. Informacija apie vykdomas verslo transakcijas saugoma duomenų bazėje. Verslo transakcijos identifikuojamos šiais identifikatoriais:

- kilpos identifikatoriumi,
- kilpos egzemplioriaus identifikatoriumi ir
- kilpos egzemplioriaus srauto S11 siuntimo laiko momentu.

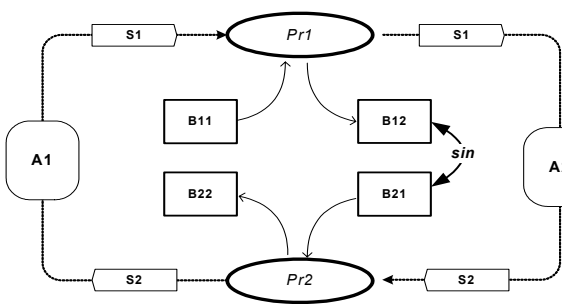
Verslo transakcijos egzempliorius gali būti realizuojamas keliomis DB transakcijomis. Duomenų bazėje turi būti išsaugota informacija apie su kiekviena verslo transakcija susietas būsenas B11, B12, B21, B22. Visos keturios būsenos turi būti atkurtos nesėkmės atveju, kad jų atžvilgiu būtų galima vykdyti kompensacinius veiksmus. Kitaip tariant, duomenų bazėje saugoma informacija apie verslo transakciją ir ją realizuojančias DB transakcijas. Galimas ir visiškas verslo transakcijos anulavimas. Tokiu atveju bus anuliuotos visos ją realizuojančios DB transakcijos [10].

2.4 skyrelyje buvo pateikti funkciniai reikalavimai, pateiktų binarinėmis komunikacinėmis kilpomis, semantinio nuoseklumo (integralumo) darnos, pilnumo, gyvybingumo ir glaudumo kriterijai. Komunikacinių kilpų pilnumo ir glaudumo kriterijai užtikrinami DB transakcijose realizuojamais duomenų apdorojimo procesais. DB transakcijų operacijų leistinėms sekoms apriboti darbe siūloma panaudoti dinamines *exist* [3] ir *coexist* priklausomybes, kurios yra specifikuojamos organizacijos analizės arba naujų funkciniai reikalavimų specifikuojimo etapuose. Paprasta komunikacinė kilpa [2] yra pavaizduota 5.1.1 pav., o išplėstinė komunikacinė kilpa [1] - 5.1.2 pav., kurioje:

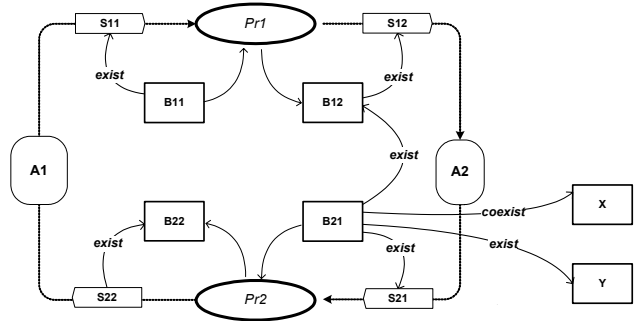
- įvestos dinaminės *exist* ir *coexist* priklausomybės [1,3];
- numatomi besiskiriantys įėjimo ir išėjimo srautai S11 ir S12 (proceso *Pr1* atžvilgiu) ir taip pat besiskiriantys įėjimo ir išėjimo srautai S21 ir S22 (proceso *Pr2* atžvilgiu).

Išplėstinių komunikacinių kilpų panaudojimas su besiskiriančiais procesų įėjimo-išėjimo srautais leidžia binarizuoti komunikacines kilpas, kai vienoje verslo transakcijoje dalyvauja daugiau negu trys aktoriai. Paprasta komunikacinė kilpa paprastesnė, kadangi aktoriaus pasiūstas srautas sutampa su srautu, kurį priima aktorius A2 (5.1.3 pav.). Jeigu verslo transakcijoje dalyvauja „trečiosios“ šalys, tai aktorius gali

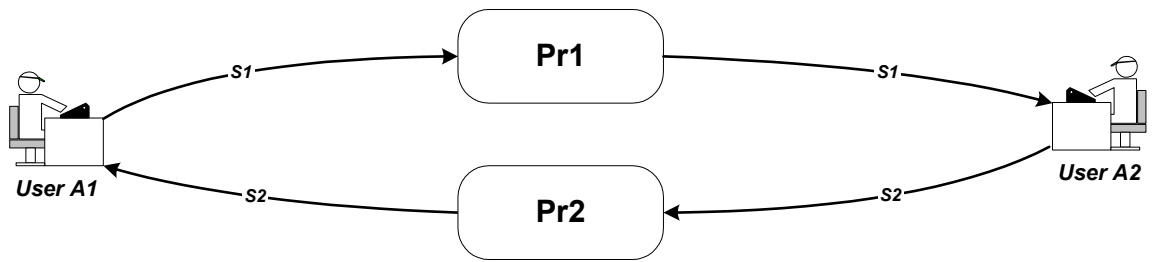
paruošti duomenis vienu metu dviems aktoriams A2 ir A2'. Sekančiame skyrelyje pateiksime *exist* priklausomybių griežtus apibrėžimus.



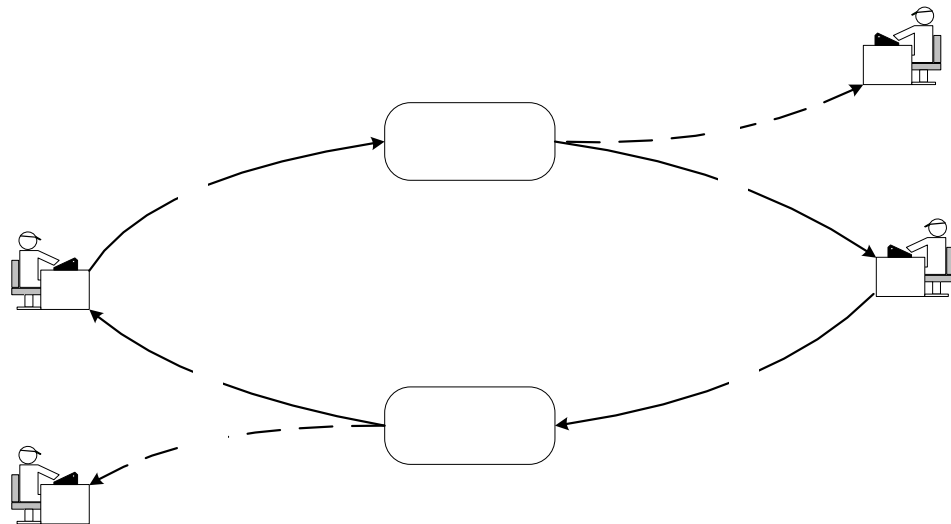
5.1.1 pav. Paprasta komunikacinė kilpa



5.1.2 pav. Išplėstinė komunikacinė kilpa



5.1.3 pav. Binarinė vartotojų bendradarbiavimo kilpa



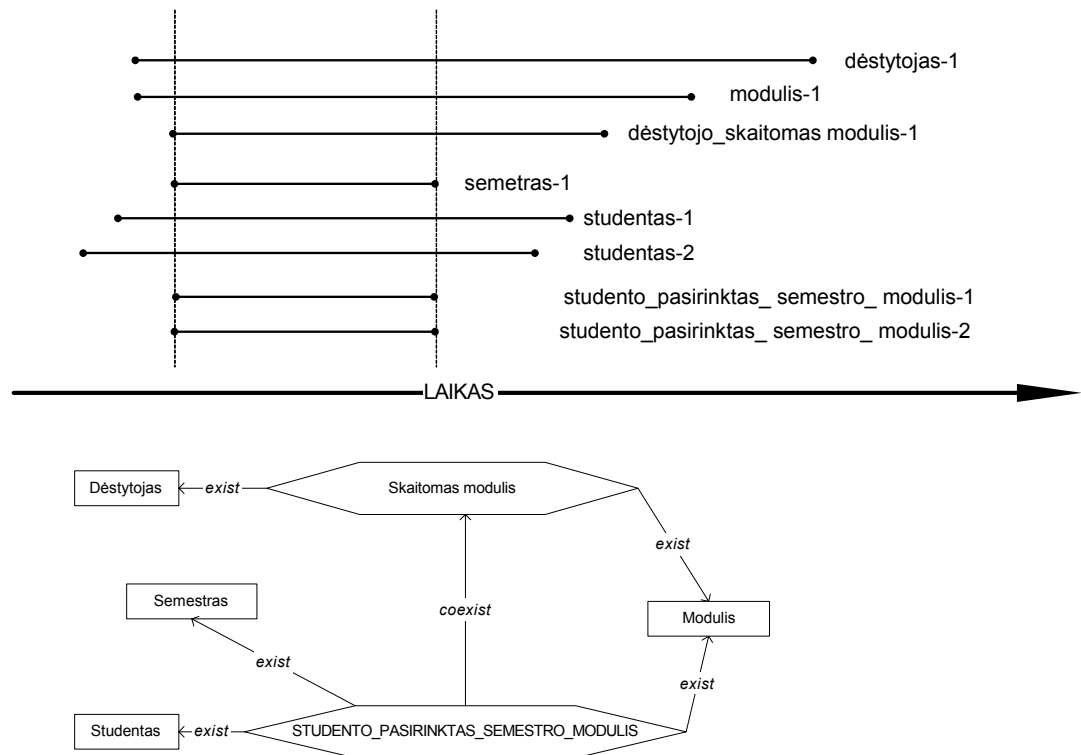
5.1.4 pav. Vartotojų bendradarbiavimo kilpa, kai siunčiami asinchroniniai pranešimai

5.2. Dinaminės priklausomybės ir jų panaudojimas DB transakcijoms sudaryti

Organizacijoje vykstančiai veiklai atvaizduoti vienas iš galimų būdų yra komunikacinių kilpų sudarymas. Veikla atvaizduojama komunikacinėmis binarinėmis kilpomis (angl. *Communicative Action Loop*), kurios yra susietos tarpusavyje semantinėmis ir pragmatinėmis priklausomybėmis. Kiekviena CAL susideda iš dviejų aktorių užsakovo A1 ir vykdytojo A2, srautų S11 - S22 ir dviejų procesų – užsakovo veiksmų Pr1 ir vykdytojo veiksmų Pr2. Kompiuterizuotos informacijos sistemos dinaminė situacijų kaita užduodama dviem perėjimais iš būsenos B11 ir B21 į būsenas B12 ir B22.

Apibrėžimas 1. Objektų tipas Y yra gyvavimo priklausomybėje nuo objektų tipo X, jei kiekvienas egzempliorius $y, y \in Y$, pastoviai atvaizduojamas tiksliai vienu ir tuo pačiu egzemplioriumi $x, x \in X$. Gyvavimo priklausomybėje Y *exist* X tipas Y vadinamas priklausomu tipu, o X tėviniu.

Apibrėžimas 2. Objektų tipas Y yra gyvavimo priklausomybėje nuo objektų tipo X, jei kiekvieno priklausomo egzemplioriaus $y (y \in Y)$ gyvavimo ciklą pilnai padengia tiksliai vieno tėvinio egzemplioriaus $x (x \in X)$ gyvavimo ciklas.



5.2.1 pav. Dalykinės srities objektų gyvavimo santykiai, kurienurodomi išplėstinėse komunikacinėse kilpose

Dėka dinaminių gyvavimo ciklo (*exist*) priklausomybių ir sambūvio (*coexist*) priklausomybių atsiranda galimybė parodyti komunikacinių kilpų duomenų objektų sudarymo ir panaudojimo nuoseklumo

apribojimus (5.2.1 pav.). Tokio tipo dinaminiai apribojimai iš funkcinių reikalavimų specifikacijos (komunikacinių kilpų) paverčiami į apribojimus DB transakcijų projektavimui:

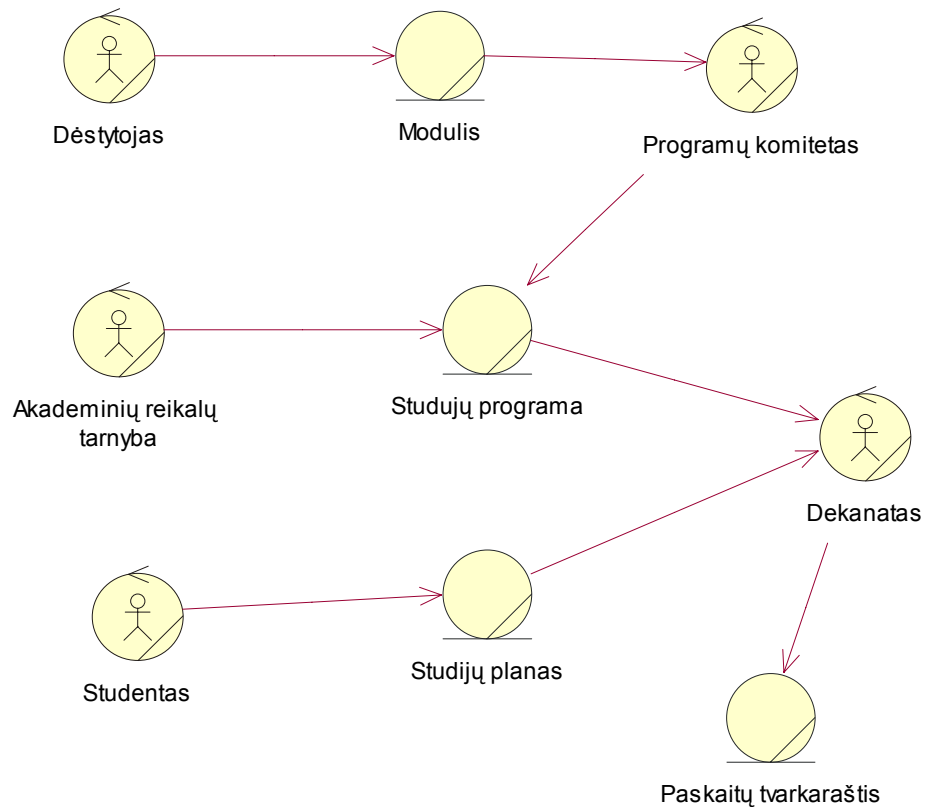
- binarinės kilpos *Pr1-Pr2* proceso *Pr2* prieš- būseną B21 turi būti paskaičiuota (žr. 5.1.2 pav.) iš proceso *Pr1* po- būsenos B12, srauto S21 ir kitų komunikacinių kilpų duomenų objektų X ir Y. Dinaminė priklausomybė *B21existY* reikalauja, kad DB transakcija, kurioje vykdomas *Pr2* procesas būtų pradama griežtai po to, kai bus baigtos DB transakcijos, kuriose vykdomi procesai *Pr1* ir *PrY*.
- dinaminė priklausomybė *B21coexistX* reikalauja, kad DB transakcijoje, kurioje vykdomas procesas *Pr2* prieš- būsenos B21 “vaikinių” objektų-egzempliorių aibės paskaičiavimui būtų jau sudarytas tipo X “tėvinių” objektų-egzempliorių aitinkamas poaibis, tačiau vaikinių ir tėvinių objektų skaičiavimas gali būti vykdomas operacijomis toje pačioje DB transakcijoje.

5.3. Akademinio posistemio funkcinių reikalavimų modeliavimas

Šiame skyrelyje pateiktos akademinio posistemio funkcinių reikalavimų specifikacijos, sudarytos panaudojant universaliąją UML kalbą ir komunikacines kilpas. Šių specifikacijų pagrindu sudaryta akademinio posistemio paskirstytų DB konceptualioji schema, projektuojami bakalaurinių darbų akademinio posistemio uždaviniai, kurie naudoja duomenų resursus, kaupiamus ir valdomus ORACLE ir MS SQL Server duomenų bazių valdymo sistemų aplinkose.

5.3.1. Veiklos sąveikų modelis

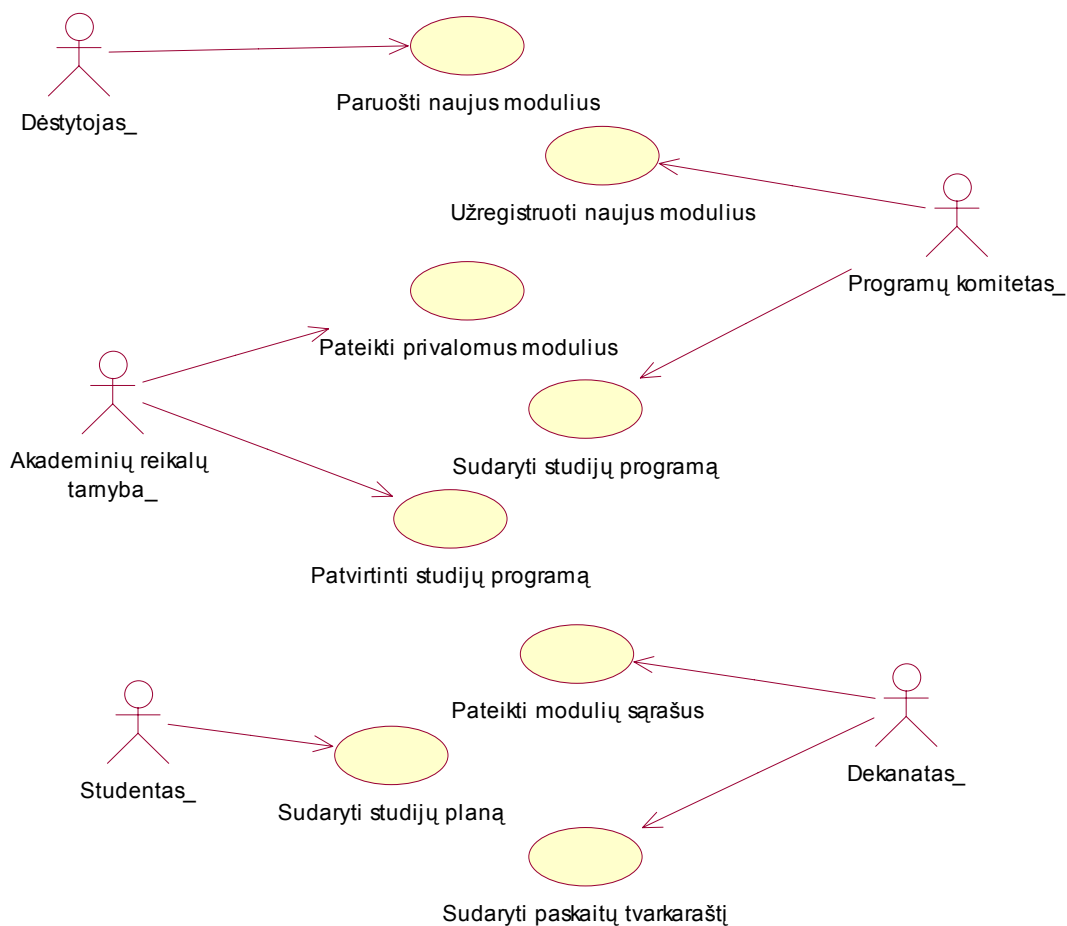
Pagrindiniai organizacijos veiklos dalyviai yra dėstytojas, dekanatas, programų komitetas, akademinų reikalų tarnyba, studentas. Dėstytojas paruošia modulius ir pateikia juos programų komitetui. Programų komitetas užregistruoja modulius ir sudaro studijų programą. Akademinų reikalų tarnyba patvirtina studijų programą ir perduoda ją dekanatui. Dekanatas pateikia studentams bazinius bei pasirenkamuosius modulius. Studentas sudaro savo studijų planą. Pagal studentų studijų planus dekanatas formuoja paskaitų tvarkaraštį.



5.2.1 pav. Objektų sąveikos modelis

5.3.2. Veiklos panaudojimo atvejų modelis

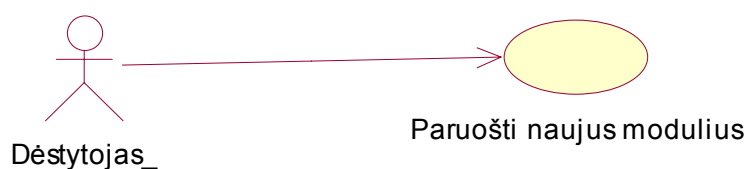
Iš veiklos panaudojimo atvejų diagramos (5.5.2 pav.) matome, kokius darbo procesus atlieka kiekvienas organizacijos veikėjas.



5.5.2 pav. Veiklos panaudojimo atvejų modelis

Dėstytojo veiklos panaudojimo atvejo modelis

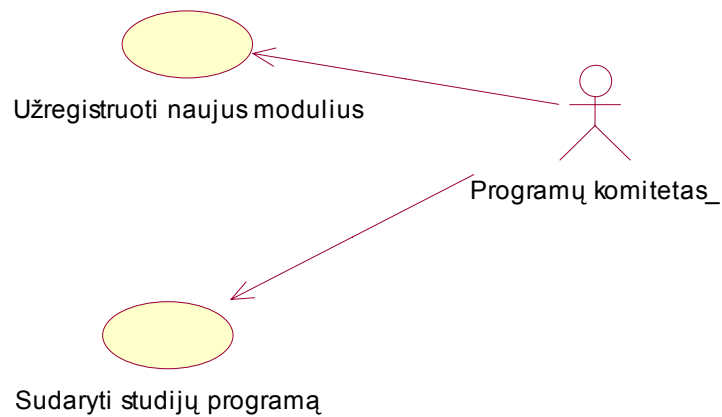
Dėstytojas ruošia naujus studijų modulius ir pateikia juos programų komitetui.



5.5.3 pav. Dėstytojo veiklos panaudojimo atvejo modelis

Programų komiteto veiklos panaudojimo atvejo modelis

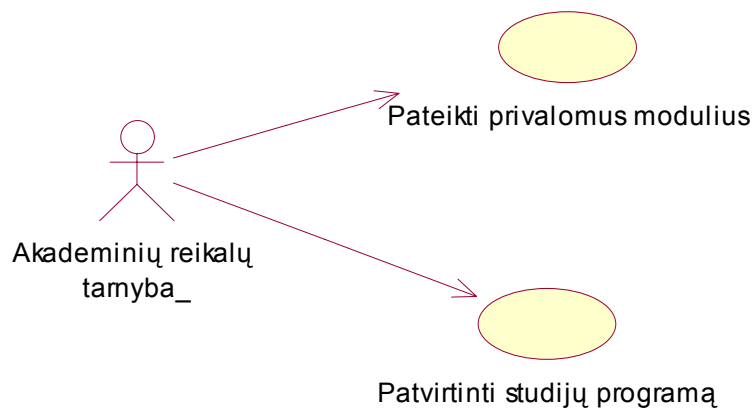
Programų komitetas užregistruoja dėstytojų sudarytus modulius, sudaro studijų programą ir pateikia ją akademinėms tarnyboms.



5.5.4 pav. Programų komiteto veiklos panaudojimo atvejo modelis

Akademinių reikalų tarnybos veiklos panaudojimo atvejo modelis

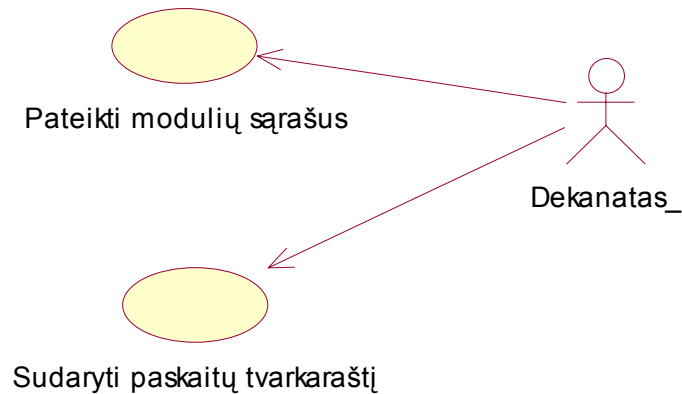
Akademinių reikalų tarnyba teikia programų komitetui privalomų modulių sąrašus, gauna iš programų komiteto sudarytą studijų programą ir ją patvirtina.



5.5.5 pav. Akademinėms tarnyboms veiklos panaudojimo atvejo modelis

Dekanato veiklos panaudojimo atvejo modelis

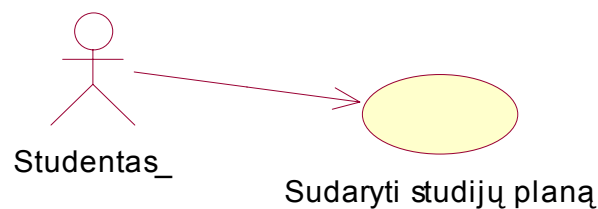
Dekanas pateikia studentams bazinių bei pasirenkamųjų modulių sąrašus, gauna iš studentų sudarytus studijų planus ir pagal juos formuoja paskaitų tvarkaraštį.



5.2.6 pav. Dekanato veiklos panaudojimo atvejo modelis

Studento veiklos panaudojimo atvejo modelis

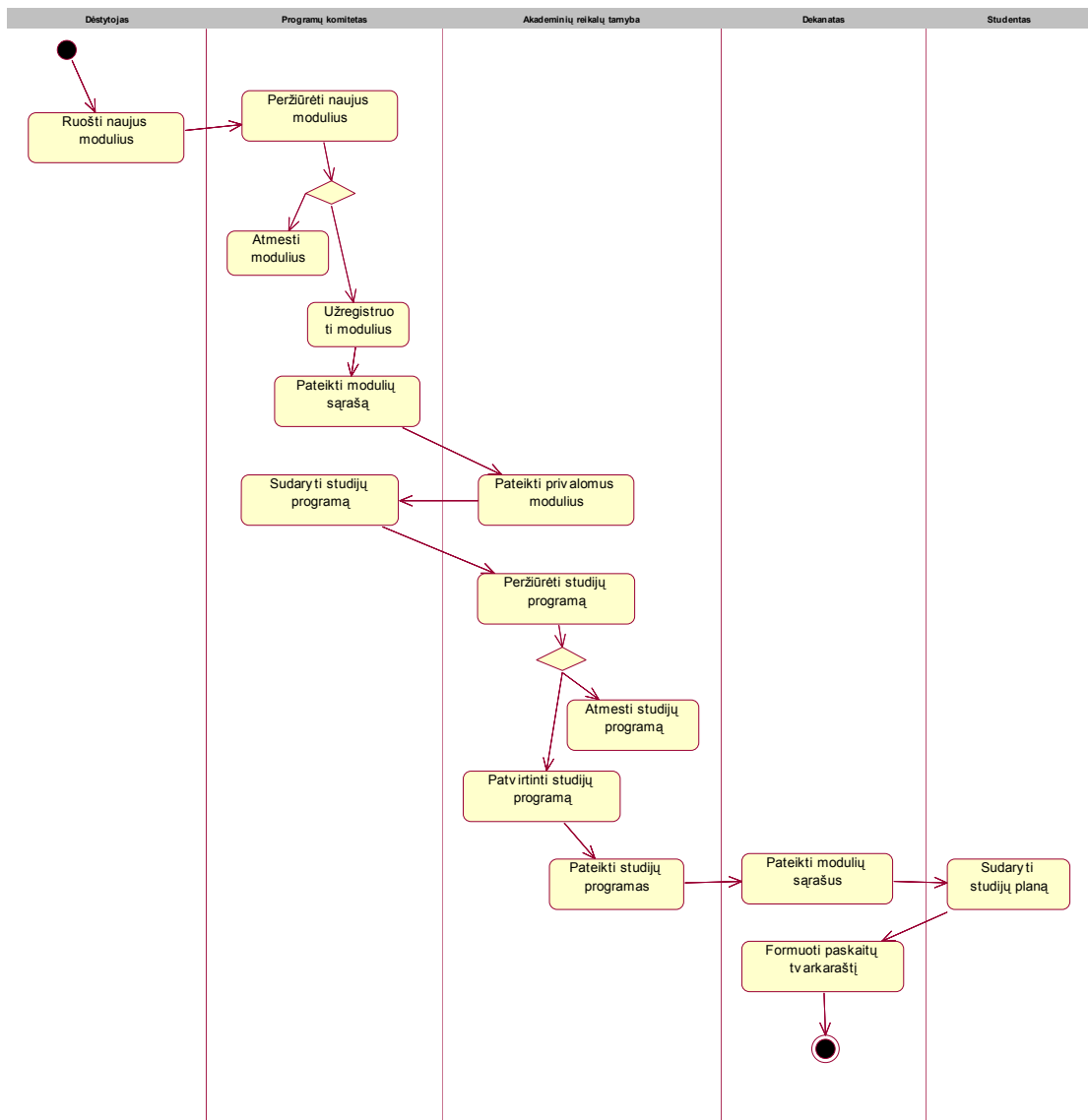
Studentas gauna iš dekanato pasirenkamųjų modulių sąrašus ir pagal juos sudaro savo studijų planą.



5.2.7 pav. Studento veiklos panaudojimo atvejo modelis

5.3.3. Procesų veiklos diagrama

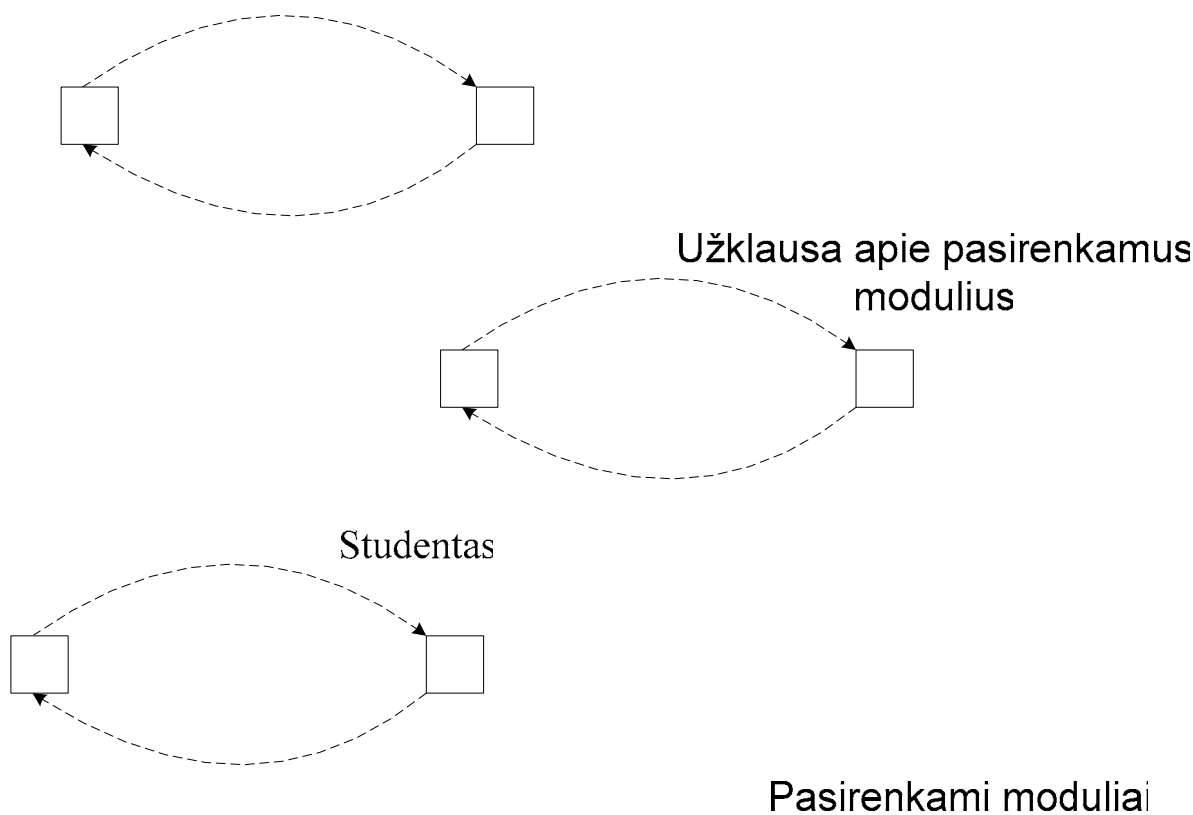
Iš procesų veiklos diagramos (5.2.3 pav.) matyti kaip yra sudaromi moduliai, studijų programos, studijų planai, formuojamas paskaitų tvarkaraštis.



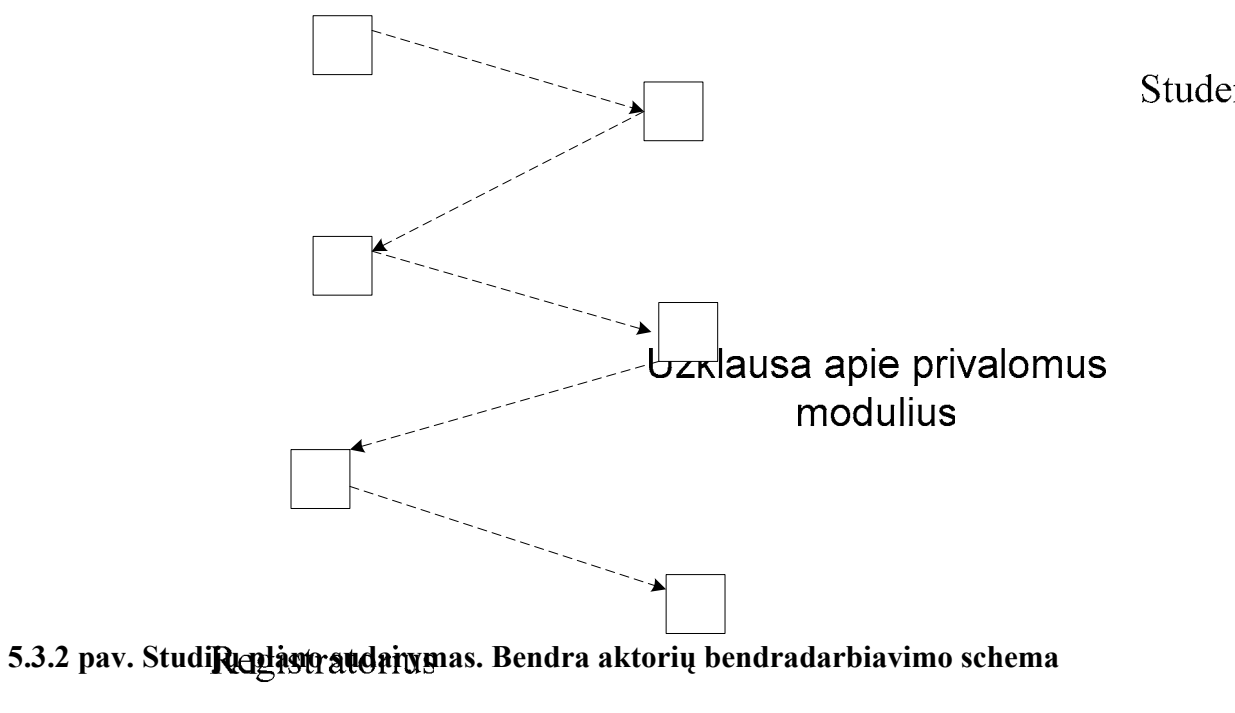
5.2.3 pav. Veiklos diagrama

5.4. Dalykinė sritis aprašyta komunikacinėmis kilpomis

Dalykinės srities modeliavimui be UML kalbos buvo naudotas ir komunikacinių kilpų modelis. Detalesniam modeliavimui buvo pasirinktas studijų plano sudarymas.



5.3.1 pav. Studijų plano sudarymas. Pagrindinės komunikacinės kilpos



5.3.2 pav. Studijų plano sudarymas. Bendra aktorių bendradarbiavimo schema

IŠVADOS

- Darbe išnagrinėta labiausiai išvystyta elektroninio verslo modeliavimo ebXML metodologija, kuri apima į kūrimą bei procesų valdymą orientuotą modeliavimą, remiasi ekonomine verslo verte ir aprašo verslo procesus moduliniu principu.
- ebXML procesų modeliavimo metodologija nėra išbaigta. Pažymėtina, kad procesas konstruojamas iš standartinių atominių elementų - transakcijų, bendradarbiavimo protokolų ir verslo dokumentų. Tai supaprastina proceso apibrėžimą, bet giluminės semantinės šių elementų priklausomybės lieka paslėptos.
- Sutrumpintas veiklos proceso apibrėžimo metodas užtikrina sintaksinį schemas korektiškumą, bet neužtikrina pragmatinio ir semantinio vientisumo bei veiksmų eigos korektiškumo.
- Darbe siūloma verslo transakcijas modeliuoti komunikacinėmis kilpų metodu, kadangi šiuo metodu galima užtikrinti tokius semantinio integralumo kriterijus:
 - darnos - galinės būsenos turi būti suderintos su aktorių tikslais;
 - pilnumo - nepilnumą galima identifikuoti iš probleminių būsenų, kurios rodo poreikius tobulinti kompiuterizuotą informacijos sistemą;
 - gyvybingumo - komunikacinė kilpa gyvybinga, jei recipiento problema neigiamai veikia iniciatoriaus problemą ir recipientas gali įvykdyti reakcijos veiksmą, kurio rezultatas teigiamai veikia tiek recipiento, tiek iniciatoriaus tikslus;
 - glaudumo - kriterijus nustato reikalavimus komunikacinių srautų valdomų būsenų atvaizdavimui duomenų bazėje.

Pragmatiškai motyvuotomis komunikacinėmis kilpomis, tenkinančiomis šiuos kriterijus, galima padidinti informacijos sistemos pakartotinės inžinerijos ilgalaikiškumą ir specifikuoti naujus funkcinius reikalavimus lokališkai, neperdarant globalinį verslo procesų modelį.

- Komunikacinių kilpų informacinius srautus siūloma realizuoti XML srautais. XML technologija pasirinkta todėl, kad tai specialiai internetui pritaikytas, lengvai išplečiamas, nuo platformos nepriklausantis ir plačiai taikomas standartas.
- Darbe apžvelgtos duomenų bazių valdymo sistemų priemonės XML srautams formuoti.
- Darbe apžvelgtos šiuo metu naudojamos verslo transakcijas vykdančios sistemos – transakcijų serveriai. Nustatyta, kad transakcijų serverių funkcionalumas neapsiriboja vien transakcijų tarp kliento ir serverio palaikymu. Tai sudėtinga įmonės verslo sistemų ir programų integravimo priemonė. Tačiau transakcijų serveriai nesuteikia paruoštų priemonių XML srautams formuoti.

- Nustatyta, kad persiunčiant tarp aktorių XML srautus, visos verslo procesų operacijos turi būti protokoluojamos DB žurnaluose. Norint nesėkmės atveju atstatyti verslo transakciją į pradinę būseną, duomenų bazėje turi būti išsaugojami tokie kilpos parametrai:
 - kilpos identifikatorius,
 - kilpos egzempliorių identifikatoriai,
 - kilpos egzempliorių pirminio srauto S11 siuntimo laiko momentai.
- Pateikti formalūs dinaminių exist- ir coexist- priklausomybių apibrėžimai. Parodyta, kaip sudėtinę verslo transakciją, pateikiamą keliomis izoliuotomis išplėstinėmis komunikacinėmis kilpomis, susieti tarpusavyje exist- ir coexist- priklausomybėmis. Jų panaudojimo privalumai yra:
 - sudėtingas verslo transakcijas galima detalizuoti;
 - pateikti jas paprastų binarinių kilpų rinkiniais;
 - šitaip susietose kilpose perduodami dinaminiai apribojimai sudedamųjų DB transakcijų programavimo stadijai ir tų transakcijų vykdymo grafikui kontroliuoti;
- Išnagrinėtam ir siūlomam naudoti komunikacinių kilpų metodui iliustruoti sudarytos universiteto akademinio posistemio naujų funkcinių reikalavimų specifikacijos dviem metodais – UML kalba ir komunikacinėmis kilpomis. Informacija apie dalykinės srities **statines** ir **dinamines** savybes, nustatoma:
 - panaudojimo atvejų,
 - objektų sąveikos,
 - objektų būsenų kaitos ir
 - veiklos diagramomis

gali būti pateiktos viena analogiško sudėtingumo komunikacinių kilpų diagrama, kurioje **statinės**, **dinaminės** ir **pragmatinės** priklausomybės yra tarpusavyje suderintos; komunikacinių kilpų diagramos pagrindu ISK projektuojamos mokomosios hibridinės paskirstytos objektinės, reliacinės ir XML duomenų bazės.

- Išnagrinėto funkcinių reikalavimų specifikavimo komunikacinėmis kilpomis metodo pagrindu reikia išpuoselėti:
 - naujų funkcinių reikalavimų detalizavimo, panaudojant išplėstines komunikacines kilpas, metodiką;
 - sudaryti taisykles duomenų srautų XML struktūroms nustatyti ir integruoti su reliacinėmis DB, kuriose yra saugojamos ryšio objektų būsenos;

- sudaryti taisykles kilpomis nustatytų verslo transakcijoms atvaizduoti paskirstytų DB transakcijomis, kuriose ryšių po- būsenos yra skaičiuojamos iš prieš- būsenų ir aktorių persiunčiamų duomenų srautų

LITERATŪRA

- [1] **Nemuraitė L., Paradauskas B., Salelionis L.** Extended Communicative Action Loop for Integration of New Functional Requirements, *Information technology and control. ISSN 1392-124X. Kaunas: Technologija*, 2002, Nr. 2(23), p. 18–26.
- [2] **Gustas R.** Semantic and pragmatic dependencies of information systems. *Monografija, Kaunas: Technologija*, 1997, p. 274.
- [3] **Snoeck M., Dedene G., Verhelst M., Depuydt A.M.** Object-oriented Enterprise Modeling with MERODE. Belgium: Leuven University Press, 1999.
- [4] **Nemuraitė L.** Elektroninio verslo procesų modeliavimo metodų tendencijos, *Informacijos mokslai. Vilniaus universitetas*, 2002, Nr. 21, p. 77-88.
- [5] **Paradauskas B., Šidlauskas K.** Duomenų dinaminių priklausomybių naudojimas modeliuojant verslo transakcijas. *Informacijos mokslai. Vilnius : Vilniaus universitetas*.
- [6] Business Transaction Protocol. Draft Specification, version 1.0. - Organization for the Advancement of Structured Information Systems (OASIS).
- [7] Grefen P., Vonk J. Apers P. Global transaction support for workflow management systems: from formal specification to practical implementation. *The VLDB Journal* N 10, 2001, p. 316–333.
- [8] Using XML in Oracle Database Applications. Prieiga per internetą: http://www.oracle.com/technology/tech/xml/info/htdocs/otnwp/about_oracle_xml_products.htm [žiūrėta. 2004-12-04]
- [9] XML and Databases. Prieiga per internetą: <http://www.rpbouret.com/xml/XMLDatabaseProds.htm> [žiūrėta. 2004-12-03]
- [10] **Kanaitis R., Paradauskas B.** Naujų funkcinių reikalavimų įvedimas informacijos sistemose. *Informacinės technologijos '05 : konferencijos pranešimų medžiaga, Kaunas 2005. Kauno technologijos universitetas. - Kaunas : Technologija*, 2005, p. 155-159.
- [11] Overview of TP Systems and Products. Prieiga per internetą: <http://www.ti5.tu-harburg.de/Lecture/99ws/TP/06-OverviewOfTPSystemsAndProducts/> . [Žiūrėta 2005-04-20].
- [12] Introduction to ebXML. Prieiga: <http://www-128.ibm.com/developerworks/edu/x-dw-xebxml-i.html>. [Žiūrėta 2005-03-12]
- [13] ebXML. Prieiga: <http://www.ebxml.org/>. [Žiūrėta 2004-04-14].

1 PRIEDAS. Straipsnis “Naujų funkcinių reikalavimų įvedimas informacijos sistemose”

Naujų funkcinių reikalavimų įvedimas informacijos sistemose

Ričardas Kanaitis, Bronius Paradauskas

Kauno Technologijos Universitetas, Informacijos sistemų katedra

Studentų g. 50, Kaunas

Darbe nagrinėjamas naujų funkcinių reikalavimų įvedimas jau sukurtai ir veikiančiai informacijos sistemai (IS). Vienas pagrindinių uždavinių įvedant naujus reikalavimus yra organizacijos veiklos procesų ir duomenų struktūrų vientisumo palaikymas. Naujų reikalavimų aprašymui tikslinga naudoti išplėstines komunikacines kilpas. Komunikacinė kilpa atitinka verslo transakciją, kuriai galima nustatyti formalius parametrus. Pažymėtina, kad IS yra sukurta hibridinių duomenų bazių pagrindu, todėl reikalinga papildoma programinė įranga (valdiklis), kuri užtikrintų sklandų verslo transakcijų vykdymą, duomenų bazių transakcijų pagrindu.

1. Įvadas

Norint garantuoti, kad verslo informacijos sistemose, ypač susijusiose su organizacijų darbų sekų valdymu bei verslas-klientui B2C (angl. Business-To-Customer) ar verslas-verslui B2B (angl. Business-To-Business) procesais, suderintai ir patikimai funkcionuotų taikomosios programos, reikia palaikyti verslo transakcijas. Kadangi vartotojų programos ir duomenys tokiose sistemose tendencingai atskiriamos, tai transakcijų mechanizmų svarba tokiomis sąlygomis labai padidėja. Paskirstytiesiems verslo procesams būdingos ilgos trukmės (palyginti su duomenų bazių transakcijų trukmėmis), jie realizuojami įvykdant daugelį DB transakcijų ir kaupiant duomenų srautus paskirstytosiose DB. Verslo procesus galima apibrėžti verslo transakcijų aibėmis.

Verslo transakcijų modelių lyginamoji analizė [1], [4] rodo, kad verslo transakcija įvairiuose elektroninio verslo modeliavimo metoduose suprantama skirtingai ir neturi tikslaus apibrėžimo. Kai kuriuose protokoluose ar kalbose (pavyzdžiui, BTP arba XLANG) visai neapibrėžiama, kokios veiksmų sekos gali būti pripažįstamos transakcijomis. ebXML bendradarbiavimas apibrėžiamas ekonominių įvykių pagrindu, t.y. transakcijos reiškia ekonominius įvykius, kurių metu vienas partneris perduoda kitam ekonominę vertę ar ekonominių išteklių kontrolę. Dažniausiai verslo transakcija apibrėžiama kaip universaliosios modeliavimo kalbos UML panaudojimo atvejis: veiksmų seka, kuri duoda tam tikrą reikšmingą rezultatą šią seką inicijavusiam aktoriui. Toks traktavimas sukelia problemų dėl panaudojimo atvejų detalizavimo laipsnio neapibrėžtumo. Darbe [1] verslo transakcijas siūloma apibrėžti komunikacinių veiksmų kilpomis, kurios sudaro organizacijos modeliavimo EM metodo [2] pagrindą.

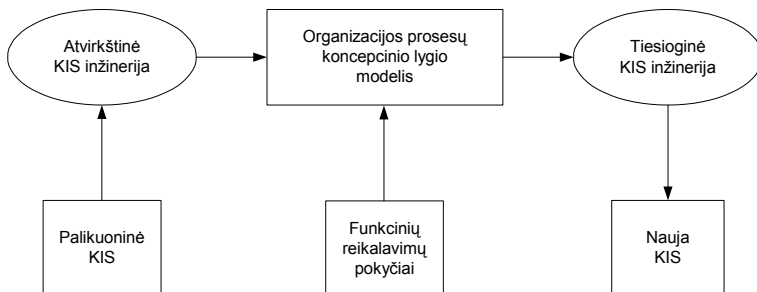
Elektroninio verslo procesams būdinga nuolatinė kaita. Keičiasi ne tik transakcijų sekos, bet ir paslaugų turinys. Paslaugos realizuojamos jas teikiančiose organizacijose ir skelbiamos visuotiniame interneto tinkle. Taigi verslo sistemose nuolat atsiranda naujų funkcinių reikalavimų, kuriems dengti reikia naujų duomenų struktūrų. Norint patenkinti pakitusius informacijos poreikius, naujas duomenų struktūras tikslinga integruoti su esamų palikuoninių duomenų bazių schemomis. Naujus funkcinius reikalavimus taip pat tikslinga apibrėžti komunikacinių veiksmų kilpomis, kurios integruojamos į vidinių ar bendrų verslo procesų modelius. Norint į esamą informacinę sistemą įjungti naujus reikalavimus, reikia identifikuoti palikuoninių duomenų bazių ir komunikacinių kilpomis aprašytų duomenų schemų sanklotas ir nustatyti jų struktūrinį ekvivalentumą. Nauji funkciniai poreikiai motyvuojami pragmatinėmis priklausomybėmis, tuo užtikrinant informacinės sistemos gyvybingumo ir informacijos susietumo kriterijus.

Darbe aptariamas naujų funkcinių reikalavimų specifیکavimas komunikacinių veiksmų kilpomis. Naujų reikalavimų realizavimui siūloma verslo transakcijas, specifikuotas komunikacinių veiksmų kilpomis, susieti su duomenų bazių transakcijomis. Taip būtų užtikrintas patikimas verslo transakcijų vykdymas.

2. Hibridinės duomenų bazės

1 paveiksle pateikta kompiuterizuotosios informacijos sistemos (KIS) reinžinerijos schema, kurioje išskiriami tiesioginės ir atvirkštinės inžinerijos procesai. Nauji funkciniai reikalavimai vaizduojami bendrajame organizacijos procesų modelyje ir derinami su palikuoninės KIS procesais konceptualiajame lygmenyje. KIS atvirkštinės inžinerijos uždavinys yra gauti informaciją apie kompiuterizuotus duomenų apdorojimo procesus ir apibendrinti bei pateikti ją standartizuota verslo procesų aprašymo kalba. Toje pačioje kalboje įvedami funkcinių reikalavimų papildymai. Iš visuotinio organizacijos procesų modelio sudaromas naujos KIS projektas, kurio svarbi sudėtinė dalis yra DB projektas.

Tradiciskai duomenų bazės loginio lygio (programuotojo arba DBVS lygio) schemų seka būtų generuojama automatiškai iš konceptualiojo lygio (vartotojo arba verslo procesų lygio) visuotinio duomenų modelio – EER diagramų arba klasių diagramos. Naujas visuotinis modelis projektuojamas naudojant palikuoninės KIS projektą, jeigu jis toks buvo ar išliko, ir įvertinant pokyčius organizacijos funkcinėje veikloje. KIS reinžinerija suprantama kaip nuoseklus perėjimas (1 pav.) nuo esamos KIS prie naujos: čia paliekami spręsti aktualūs uždaviniai, atitinkantys senus vartotojų poreikius, ir prie jų prijungiami nauji uždaviniai, reikalingi naujiems poreikiams tenkinti.

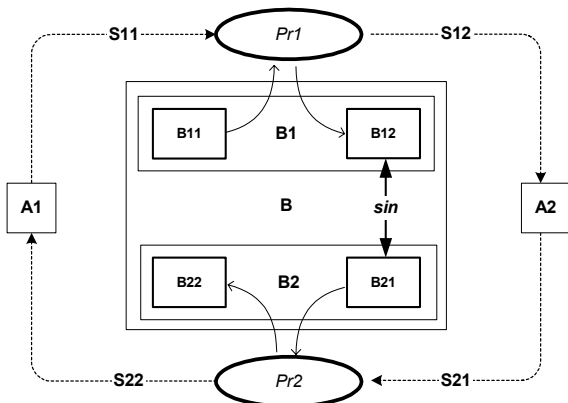


1 pav. KIS reinžinerijos procesas, pagrįstas veiklos procesų reinžinerija.

Išplėstine komunikacine kilpa aprašytą verslo transakciją galima naudoti kaip funkcinių reikalavimų aprašymo vienetą. Naujų funkcinių reikalavimų įvedimas paprastai reikalauja papildomų duomenų struktūrų. Po kiekvieno naujo kompiuterizuojamo proceso įvedimo tenka spręsti klausimą, ar pakankami yra eksploatuojamos KIS DB informaciniai ištekčiai. Jei informacinis turinys nepakankamas, tenka projektuoti papildomas duomenų struktūras. Naujos informacinės technologijos skatina kurti naujas DB, tačiau palikuoninėse bazėse saugomų duomenų išliekamoji vertė neleidžia jų atsisakyti. Todėl projektuojant KIS tenka numatyti, kad bus naudojamos naujomis ir senomis duomenų bazėmis. Palikuonines duomenų bazes išsaugoti vertinga ir tuo požiūriu, kad galima palikti eksploatuoti nepakitusius uždavinius. Kelių DBVS (arba tos pačios, bet skirtingų versijų), kurios palaiko tokius pat ar skirtingus duomenų modelius, panaudojimas sąlygoja nevienalyčių DB projektavimo metodų atsiradimą. Pagrindinį vaidmenį tiesioginėje KIS inžinerijoje įgauna nevienalyčių DB komponentų integravimas. Nuolatinės funkcinių reikalavimų kaitos sąlygomis ypač aktualus tampa nevienalyčių DB komponentų projektavimo ir integravimo CASE priemonių sudarymas. Šios klasės CASE priemonės galima priskirti prie technologinių priemonių, skirtų sparčiam taikomųjų programų kūrimui užtikrinti. Jos naudojamos KIS reinžinerijoje, tiesiogiai projektuojant naujus trūkstamus ar replikuojamus DB komponentus pagal informaciją apie funkcinių reikalavimų papildymus. Nevienalytės DB, turinčios tiek reliacinių, tiek objektinių komponentų, vadinamos hibridinėmis duomenų bazėmis [3].

3. Išplėstųjų komunikacinių kilpų schemų sudarymas

2 paveiksle parodytas bendrasis išplėtosios komunikacinės kilpos atvejis. Kiekviena komunikacinė kilpa susideda iš dviejų aktorių – užsakovo A1 ir vykdytojo A2, srautų S11-S22 ir dviejų procesų – užsakovo veiksmų Pr1 ir vykdytojo veiksmų Pr2. Kilpos dinamika nustatoma dviem perėjimais iš būsenos B11 ir B21 į būsenas B12 ir B22. Komunikacinė kilpa vadinama išplėstąja, kadangi kilpos atžvilgiu ji turi papildomas duomenų struktūras bei priklausomybes (be abipusio apibendrinimo *sin*, dar naudojamos agregavimo *composed of*, gyvavimo *exist* ir poaibio \subseteq priklausomybės).



2 pav. Išplėstoji komunikacinių veiksmų kilpa su apibendrinimo ryšiais tarp būsenų B11, B12, B21 ir B22.

Jeigu kompiuterizuotos sistemos projektuojamos komunikacinių kilpų analizės metodu, galima užtikrinti tam tikrus semantinio integralumo kriterijus:

- Darnos (angl. *coherence*) kriterijus reiškia, kad galinės būsenos turi būti suderintos su aktorių tikslais.

- Pilnumo (angl. *completeness*) kriterijus reiškia, kad nepilnumą galima identifikuoti iš probleminių būsenų, kurios rodo poreikius tobulinti kompiuterizuotą informacijos sistemą.
- Gyvybingumo (angl. *viability*) kriterijus išreiškiamas pragmatinėmis priklausomybėmis: komunikacinė kilpa yra gyvybinga, jei

$$A1 \xrightarrow{p} B11, A2 \xrightarrow{o} B22, A2 \xrightarrow{p} B21, A1 \xrightarrow{g} B12,$$

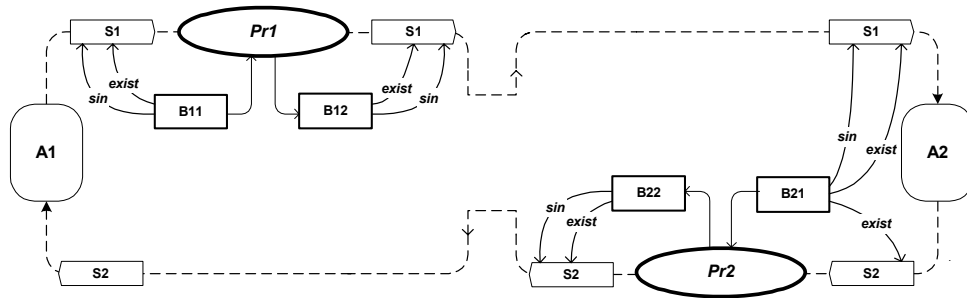
tuomet $B11 \xrightarrow{-} B12, B22 \xrightarrow{+} B12, B22 \xrightarrow{-} B11, B12 \xrightarrow{-} B21;$

čia simbolis \xrightarrow{g} žymi pragmatines priklausomybes: \xrightarrow{g} reiškia tikslą, \xrightarrow{p} – problemą, \xrightarrow{o} – progą, $\xrightarrow{+}$ – teigiamos įtakos priklausomybę ir $\xrightarrow{-}$ – neigiamos įtakos priklausomybę. Kitaip tariant, komunikacinė kilpa gyvybinga, jei recipientas problema neigiamai veikia iniciatoriaus problemą ir recipientas gali įvykdyti reakcijos veiksmą, kurio rezultatas teigiamai veikia tiek recipientą, tiek iniciatoriaus tikslus.

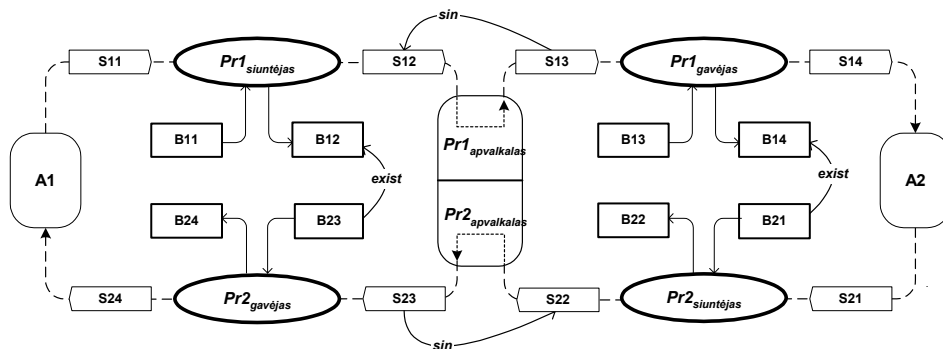
- Glaudumo (angl. *cohesion*) kriterijus nustato reikalavimus komunikacinių srautų valdomų būsenų atvaizdavimui duomenų bazėje. Ryšiai tarp būsenų ir komunikacinių srautų struktūros palaikomi atvaizduojant būsenų pokyčius į duomenų bazės egzempliorių atributų aibes.

3 paveiksle parodytas srauto S1 persiuntimas elementariojoje vienalytėje kilpoje, kurioje užsakovo siunčiami ir gavėjo gaunami srautai sutampa ir srauto gavėjas paprasčiausiai formuoja pranešimą apie srauto gavimą. Bendruoju atveju siunčiami ir gaunami srautai nesutampa.

Tarkime, nustatytas naujas funkcinis reikalavimas, kurį reikia realizuoti objektinėje aplinkoje ir jo informacinį turinį gali dengti palikuoninė RDB. 4 paveiksle parodyta, kaip informaciniai procesai Pr1 ir Pr2 išskaidomi nevienalytėje kilpoje, kurioje užsakovas A1 sąveikauja su objektine aplinka, o vykdytojas A2 – su reliacinės DB aplinka. Tiek užsakovo A1 dalyje Pr1_{siuntėjas}–Pr2_{gavėjas}, tiek vykdytojo A2 dalyje Pr1_{gavėjas}–Pr2_{siuntėjas} informacija apie verslo objektus per visą verslo transakcijos ciklą išsaugoma vietinėse duomenų bazėse. Pranešimų keitimą vykdo paslaugų sistema, kuri formuoja identišką turinio pranešimus: S12sinS13, S22sinS23. Išoriniai apribojimai – *exist* ir *sin* priklausomybės – yra savaime suprantami ir gali būti nerodomi.

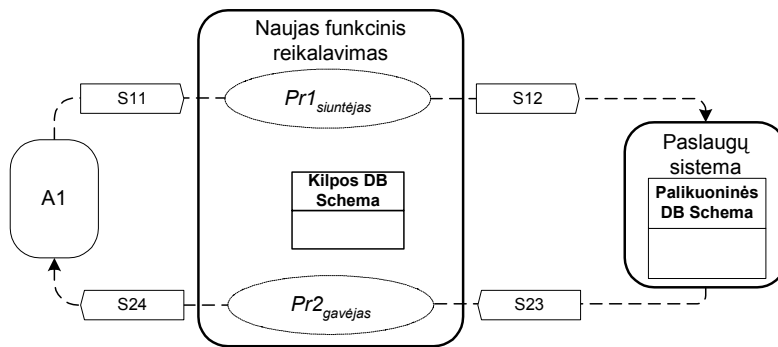


3 pav. Srauto S1 persiuntimas elementariojoje vienalytėje kilpoje; formuojamas pranešimas S2 „Srautą S1 gavau“.



4 pav. Nevienalytė komunikacinė kilpa su paslaugų sistema.

Schemiškai naujų funkcinių reikalavimų įvedimą galima pavaizduoti 5 paveikslu. Palikuoninė RDB su objektiiniu apvalkalu traktuojama kaip paslaugų sistema. Įvedant naują aktoriaus A1 funkcinį reikalavimą, jis pateikiamas sudėtinu procesu Pr1_{siuntėjas}–Pr2_{gavėjas} ir kilpos objektinės DB schema; paslaugos sistema Pr3 praplečiama sudėtinu procesu Pr1_{gavėjas}–Pr2_{siuntėjas}, o palikuoninės reliacinės duomenų bazės schema papildoma trūkstamomis struktūromis (5 pav.).



5 pav. Aktoriaus A1 naujas funkcinis reikalavimas, pavaizduotas sudėtinu procesu.

4. Verslo transakcijų realizavimas duomenų bazių transakcijomis

Verslo transakcija specifikuojama komunikacine kilpa. Informacija apie vykdomas verslo transakcijas saugoma duomenų bazėje. Verslo transakcijos identifikuojamos

- kilpos identifikatoriumi,
- kilpos egzemplioriaus identifikatoriumi ir
- kilpos egzemplioriaus srauto S11 siuntimo laiko momentu.

Verslo transakcijos egzempliorius gali būti realizuojamas keturiomis DB transakcijomis. Duomenų bazėje turi būti išsaugota informacija apie su kiekviena verslo transakcija susietas būsenas B11, B12, B21, B22. Visos keturios būsenos turi būti atkurtos nesėkmės atveju, kad jų atžvilgiu būtų galima vykdyti kompensacinius veiksmus. Kitaip tariant, duomenų bazėje saugoma informacija apie verslo transakciją ir ją realizuojančias DB transakcijas. Galimas ir visiškas verslo transakcijos anuliavimas. Tokiu atveju bus anuliuotos visos ją realizuojančios DB transakcijos.

5. Apibendrinimas

Norint garantuoti patikimą verslo informacijos sistemų funkcionavimą, reikia palaikyti verslo transakcijas. Verslo transakcijas galima apibrėžti komunikacinių veiksmų kilpomis, kurios sudaro organizacijos modeliavimo EM metodo pagrindą. Keičiantis verslo procesams informacijos sistemoms keliami nauji funkciniai reikalavimai, kuriuos taip pat tikslinga apibrėžti komunikacinių veiksmų kilpomis. Projektuojant KIS tenka numatyti, kad bus naudojamos tiek naujomis, tiek senomis (palikuoninėmis) duomenų bazėmis. KIS projektavime naudojant komunikacinių kilpų analizės metodą, galima užtikrinti semantinio integralumo kriterijus. Naujų funkcinų reikalavimų realizavime siūloma susieti verslo transakcijas su duomenų bazių transakcijomis. Tai užtikrintų patikimą verslo transakcijų vykdymą.

Literatūros sąrašas

- [1] Nemuraitė L., Paradauskas B., Salelionis L. Extended Communicative Action Loop for Integration of New Functional Requirements, *Information technology and control*. ISSN 1392-124X. Kaunas: Technologija, 2002, No. 2(23), pp. 18–26.
- [2] Gustas R. Semantic and pragmatic dependencies of information systems. *Monograph, Technologija*, 1997, pp. 274.
- [3] Paradauskas B., Nemuraitė L. Duomenų bazės ir semantiniai modeliai. *Monografija*. Kaunas: Technologija, 2002, pp. 227-243.
- [4] Nemuraitė L. Elektroninio verslo procesų modeliavimo metodų tendencijos, *Informacijos mokslai*. Vilniaus universitetas, 2002, No. 21, p. 77-88.

Integration of new functional requirements in information systems

In changing world of computerized information systems modeling of business transactions plays an important role. Business transaction can be defined as extended communicative action loop. Using such approach, business processes and new functional requirements may be represented as choreographed sets of extended communicative action loops with established pragmatic and semantic dependencies. In general case, a computerized information system may be of hybrid nature, that is, it may consist of object and relational databases and different versions of these databases.