

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INORMACIJOS SISTEMŲ KATEDRA

Verslo B2B transakcijų realizavimas panaudojant BTP protokolą

Magistro darbas

Recenzentas:
doc. dr. R.Micevičienė

Darbo vadovas:
doc. dr. B.Paradauskas

Atliko:
IFN-6/1 gr. studentas
Paulius Tuminas

Kaunas
2008

Business B2B Transactions Using BTP Protocol

SUMMARY

This work aim is to create information systems (IS), which can implement B2B transaction by using the Business Transaction Protocol (BTP). IS has only 50 years history, but without IS we can't imagine modern business company. Now computers are usable in most leading edge areas and their needs and power always grows. We use its in distance learning, organizing distance work (work-net), expeditiously developing electronic business, appointing a lot off attention to modernize medicine, new possibilities for electronic libraries. This work can be used in all mentioned regions implementing data exchange among computers regardless their actions in business. Many companies store data in database or data centers, so if we want to realize exchange data among computer – our programmers have to avail a lot of energy and workforce to realize exchange from remote systems or databases, especially, if some services and data are in different databases.

This work can realize data exchange simplification, which executes data exchange from unlimited numbers of business companies. It uses well-known BTP protocol from OASIS consortium to exchange data. All needed data we get from business participants using standard SQL queries. Thus people which are asking for data – in this work we can offer to get data from different databases platforms. To access system in different programming languages, data are implemented into standard SOAP1.1 protocol - where they will be transported in XML document.

This work was coded in JAVA with using ORACLE database. In this program there are about 8000 codes lines, was generated 36 class files (including 6 server class files), was realized 150 error tools (including 35 server errors) seeking to make software flexible and use simplest data's exchange among different autonomous servers.

Turinys

Turinys.....	3
Įvadas.....	4
1. Verslo transakcijų apžvalga	6
1.1. BTP protokolas	6
1.2. BPEL	7
1.3. WS-Transaction (WS-T)	8
1.4. Activity Service	8
1.5. Kitos transakcijų palaikymo sistemos.....	9
1.6. Transakcijų detalizacija.....	9
1.7. Cohesion.....	10
2. BTP transakcijų realizavimas.....	12
2.1. BTP transakcijų realizavimo pavyzdys.....	12
2.2. Transakcijų turinio pateikimo realizavimas	14
2.3. Transakcijų patvirtinimo realizavimas.....	15
2.4. Realizavimo įrankiai	18
2.4.1. Oracle duomenų bazė	18
2.4.2. Apache Tomcat	19
2.4.3. AXIS paslaugų paketas.....	20
2.4.4. SOAP protokolas.....	21
3. Produkto analizė	23
3.1. Produkto veiklos kontekstas.....	23
3.2. Veiklos padalinimas.....	23
3.3. Produkto veiklos sfera.....	24
3.4. Reikalavimų specifikacija	25
3.5. Sistemos išdėstymo (deployment) vaizdas.....	28
3.6. Sistemos suskaldymas į paketus	29
4. Paketų aprašymai.....	30
4.1. BTP paketo aprašymas.....	30
4.2. KOORD paketo aprašymas	31
4.3. PATEIK paketo aprašymas	32
4.4. ATSAKYTI paketo aprašymas.....	34
4.5. Duomenų bazės schema	35
5. Eksperimentinis tyrimas.....	40
5.1. Sistemos eksperimentinis tyrimas ir testavimas	40
5.2. Sistemos kriterijų vertinimas.....	46
5.3. Sistemos taikymo rekomendacijos.....	46
Išvados.....	48
Terminų žodynas	52
PRIEDAI.....	53

Ivadas

Dauguma šiandieninių verslo transakcijų atliekama įmonėse. Tam kad verslas būtų sėkmingai vystomas reikalinga atitikti kelis pagrindinius verslo kriterijus: laiko planavimas, greitas, paprastas, operatyvus prisitaikymas prie verslo pokyčių. Kadangi dauguma verslo įmonių yra kompiuterizuotos, todėl verslo transakcijas galime apibrėžti, kaip pastovią būseną, kuri gali pasikeisti vykstant verslo procesu bendradarbiavimui tarp vienos ar keletą šalių, kur kiekviena šalis turi skirtingą kompiuterinę sistemą, kuriai reikalingi įvairūs ir bendri duomenys, bei tarpusavio sąsajų suderinamumas.

Taigi verslas tampa neatsiejama duomenimis apsikaitimo dalis, todėl tampa ypač svarbu tinkamai transakcijas palaikyti verslo sistemose. Duomenys šiuo metu saugomi netik įmonių sukurtose bazėse, bet ir duomenų saugojimo centruose, todėl atsiranda poreikis šiuos duomenis standartizuoti ir leisti įmonėms apsikaiti esančiais duomenimis be specifinio savo modelio palaikymo. Tai galima realizuoti ir naudojant duomenų bazių standartines duomenų apsikaitimo transakcijų savybes, tačiau versle turi egzistuoti atitinkama verslo logika ir šiuo atveju tiesiogiai taikyti duomenų bazių ACID savybės netinka.

Verslo procesas neturėtų nutrūkti dėl ryšių nepastovumo, kai šiame verslo procese dalyvauja ne viena, o keletas organizacijų. Kadangi transakcija galėjo būti įvykdyta viename iš verslo proceso dalyvių, įmonei šis atvejis gali atnešti neprognozuojamų padarinių. Kai kuriais atvejais verslo transakcijos gali būti vykdomos net keletą parų, o kitais atvejais jos turi būti įvykdomos per keletą milisekundžių (Pvz. atliekant rezervacijas – kaip ilgalaikė, o trumpalaikė – kai reikalinga iš karto gauti dvi paslaugas, priešingu atveju verslo dalyvis rizikuoja, kad nesugebės pateikti pilną paslaugos paketą). Tačiau yra ir kita pusė, kartais gaunama informacija nekeičia verslo galutinio sandorio ir rizikos laipsnio, tačiau gali užtikrinti patikimumo laipsnį.

Siekiant įgyvendinti anksčiau paminėtus uždavinius, nuspręsta panaudoti BTP verslo transakcijų apsikaitimo protokolą, nes keitimas duomenis tarp skirtingų verslo sistemų turi būti standartizuotas ir tai neturėtų priklausyti nuo naudojamos OS ar tai, kokia programavimo kalba palaikomos įmonių valdymo sistemos. O taip pat norint

apsikeisti naujais duomenis tarp verslo vienetų, siekiama sutaupyti laiko, kuris yra skiriamas naujų programinių modulių sukūrimui siekiant apsikeisti papildomais duomenimis. Platforma aprašyta JAVA aukšto lygio programavimo kalba, nes tai tiksli, naujausia ir sintaksiškai labai stipri bei objektiškai orientuota kalba. Testavimui atlikti pasirinkta ORACLE duomenų bazė. Ši sukurta sistema pritaikyta taip, kad ją būtų galima išplėsti, siekiant ją naudoti su kitomis duomenų bazėmis, būtų galima palaikyti WEB aplinkoje, parengta kitų verslo protokolų realizacijai. Todėl šios sistemos naujumas ir reikalingumas yra neabejotinas, nes dauguma panašių sistemų yra pritaikyta konkreitiems programiniams modeliams realizuoti, su mažomis sistemos išplėtimo galimybėmis.

Atliekamas eksperimentinis tyrimas nebuvo detalizuotas į konkrečią dalykinę sritį. Darbe inicijuojami keletas verslo dalyvių, kurie dalyvauja duomenų apsikeitime vykdant transakcijas, inicijuojami interneto ryšio nutrūkimai ar laiko limitų išnaudojimai, blogų užklausų pateikimai.

1. Verslo transakcijų apžvalga

1.1. *BTP protokolas*

2002 metais gegužės mėnesį, OASIS Verslo transakcijų techninis komitetas (Business Transaction Technical Committee – BTTC) paskelbė BTP 1.0 detalizaciją, kuri apima transakcijų reguliavimą, sąryšį tarp programų. BTP 1.0 naudoja įvairios kompanijos (BEA, IBM, Sun, HP, Choreology, Oracle ir t.t.) siekdamos pakeisti verslo ir tradicines transakcijas[2].

BTP protokolas aprašytas taip, kad verslo transakcijose nėra atskirų dalių kontrolės visuose reikiamuose ištekliuose. Todėl tokioje aplinkoje šios dalys valdo atitinkamus išteklius, išskyrus jų koordinavimą, kur įprasti veiksmai naudojami, kaip siekiant įvykdyti užduotis pagal užduotas transakcijas. Servisų dalyviai gali nuspręsti ar priimti papildomą transakciją ar ne. Jei jie nusprendžia patvirtinti transakciją, jiems reikalingas specialus mechanizmas, kuris leidžia patvirtinti arba atšaukti transakcijas pagal užduotus papildomus veiksmus. Dažnai jie patys gali savarankiškai spręsti, kada ištekliai nėra blokuojami ir/arba kada pasinaudoti kompensacine transakcija, kurios pagalba galime sustabdyti perduodamus duomenis, atstatant transakciją į pradinę būseną.

BTP protokolo detalizacijoje nėra reikalavimų aukštesnių, nei tie, kurie sukurti WEB serviso architektūroje, bet jie yra suformuoti taip, kad galėtų palaikyti tarptautines transakcijas ir išlaikyti bendradarbiavimą tarp sisteminių stočių. Vis dėl to, autoriai realizavo išankstines ribas keičiantis tarpusavio protokolais.

BTP nustato atgalinius pranešimus ir pateikia juos kaip bendrą sąsają komunikaciniuose protokoluose. Tokiu atveju, mes turime galimybę koordinuoti transakcijas esančias savarankiškuose (atskiruose) servisuose ir naudoti XML pranešimus, kuriuos dalinai galėsime įsisavinti kuriant WEB serviso architektūrą. BTP suprojektuotas taip, kad ji galima susieti su bet koku kitu perdavimo protokolu.

Transakcijų saugumo aspektai, kurie naudojami taikomuosiose programose yra labai giminingi, bet BTP detalizacija nenurodo, kaip BTP transakcija turėtų būti integruota saugumo sistemoje, kadangi WEB servisų apsaugos standartai toliau plėtojami, o saugumo lygis yra palaikomas WEB servisų architektūroje.

1.2. BPEL

BPEL (BUSINESS PROCESS EXECUTION LANGUAGE) yra WEB servais pagrįsta XML kalba, suprojektuota taip, kad verslo užduotys pasiskirstytų tarp daugelio esančių verslo vienetų. Šį protokolą kūrė daugelį pasaulyje žinomų organizacijų, tokios kaip: BEA Systems, IBM ir Microsoft pakeitusios anksčiau naudotas kalbas: IBM – WSFL ir Microsoft XLANG specifikaciją. (BPEL yra kartais identifikuojamas kaip BPELWS arba BPEL4WS).

Daugiau kaip 35 kompanijos paskelbė tai ateities standartu vykdant ir teikiant kompiuterizuotas paslaugas[9].

Pasaulinio tinklo paslauga - pasaulinio tinklo taikomųjų programų standartizuotas integravimo būdas, besiremiantis paslaugomis grindžiama architektūra ir naudojant Interneto atvirusius standartus XML (duomenų žymėjimui), SOAP (duomenų persiuntimui), WSDL (paslaugų aprašymui), UDDI (paslaugų žinynų, viešųjų registrų organizavimui).

Kadangi tarpžinybiniai procesai turi būti griežtai standartizuoti bei valdomi, šio uždavinio realizavimui rekomenduotina naudoti specializuotus sprendimus paremtus BPEL (Business Process Execution Language) standartu. BPEL standartas skirtas aprašyti procesų vykdymo scenarijus, apjungiant paskirstytus kompiuterizuotus paslaugų teikimo (web-services) komponentus, kurie gali būti realizuoti skirtingose organizacijose. Tarpžinybinio bendradarbiavimo sprendimas, paremtas BPEL standartu, turi užtikrinti tokį funkcionalumą:

- Centralizuotai apibrėžti, valdyti ir koordinuoti tarpžinybinius paslaugų teikimo, duomenų mainų, konsolidacijos ir duomenų kokybės valdymo procesus;
- Apibrėžti kompleksinius paslaugų teikimo komponentus;
- Centralizuotai skirstyti standartizuotus pranešimus, skirtus paslaugų teikimo komponentams;
- Centralizuotai siųsti standartizuotus duomenų paketus šiuos paketus užsakiusioms institucijoms

Naudojant BPEL, programuotojas formaliai apibūdina verslo procesą kuris yra išplėtotas visame WEB tinkle tokiam būvyje, kur bet kurioje vietoje keletą žingsnių (gali būti ir vienas) vykdomas tas pats besitęsiantis kelias. Tai yra tiesiog BTP, WS-T ir WS-TXM specifikacijas atspindintis analogas. Kitaip tariant, tai kalba

leidžianti verslo procesus apibūdinti žymiai platesniu mastu, pasižyminti žymiai didesniu funkcionalumu.

1.3. WS-Transaction (WS-T)

WS-Transakcijos (WS-Transaction WS-TX) ir susijusi WS-Koordinavimas (WS-Coordination) specifikacijos yra išleistos 2002 rugpjūtį. Jie apdoroja mechanizmus skirtus koordinuoti individualius Web servisų užduotis į patikimas programas. Dvi detalizacijos, sukurtos bendrai su Microsoft, IBM ir BEA, priklauso arba sukurtos jau egzistuojantiems WEB servisams - WSDL, SOAP, ir WS-Security[3].

WEB servisų vykdymas yra pavaizduotas kaip serija veiksmų, kur kiekviena dalis gali būti įvykdyta atskirame WEB servise. WS-Coordination pasižymi mechanizmu, kuriame galime sukurti, registruoti, ir koordinuoti šiuos veiksmus skirtingiems WEB servisams. Kada WEB serviso vartotojas sukuria ir registruoja veiksmą, jis taip pat identifikuoja koordinatoriaus protokolą – būtent dėl to, įvyksta bendra sąsaja tarp esamų veiksmų. Detalizacija neaprašo koordinavimo protokolų, kuriuos galima būti panaudoti daugelyje sričių. Keletas koordinavimo protokolų gali būti transakciniais, kai kurie – ne.

1.4. Activity Service

“Activity Service” sistema priklauso OMG kompanijai. Ši sistema yra skirta papildomai valdyti struktūrinius mechanizmus panaudojant objektų transakcijų servisą (Object Transaction Service – OTS). OTS, suturtas pagal detalizacijų aprašymus, kurie paruošti COBRA transakcijų valdymui.[4] Activity Service detalizacijos aprašymas labai panašus į BTP detalizacijas, suskirstant į sudėtingas transakcijas, aprašant jų reikalavimus ilgalaikėms transakcijoms.

Lyginant su BTP protokolu galima pastebėti keltą trūkumų, kad siekiant naudoti XML ir Web servिसus, šis standartas visiškai pririštas prie OTS ir galime skaityti, kad WEB servिसai integruoti kaip bendrai susiję ir aiškiai nesikeis transakcijomis su servिसais užrašytais kitaip, galbūt net .NET platforma. Vis dėlto, kadangi dauguma esamų Java transakcijų vartotojai diegia JTS (tai yra Javos dalis OTS sistemoje), Activity Service yra tinkamas siekiant turėti kokybišką Java kalboje transakcijų dalies valdymą.

1.5. Kitos transakcijų palaikymo sistemos

JSR-95, J2EE Activity Service for Extended Transactions, pateikia nesudėtingą taikomosios programos interfeisą (application programming interface - API) J2EE konteineriams ir transakcijų dalyviams[5]. JSR-95 yra pasiskolintas iš OMG Activity Service ir įtakoja idėjas, kurios yra apibrėžtos JTA ir EJB detalizacijose.

JSR-156, XML Transactions for Java (JAXTX), taip pat pateikia palaikymą papildomoms transakcijoms J2EE sistemose[6]. Skirtingai nei JSR-95, JSR-156 pateikia galutines BTP specifikacijas, naudoti XML pranešimus siekiant valdyti ir koordinuoti transakcijas.

JOTM-BTP atvirojo kodo programinis modelis palaiko BTP protokolo standartą. Realizuojamas Tomcat 4.1.x serveryje ir AXIS 1.0 papildomame pakete, bei užkoduotas JAVA programavimo kalba. Programinis realizavimo modelis šiame darbe nepasirinktas todėl, kad JOTM nepalaiko apjungimo transakcijų[14].

Transakcijų paslauga JXTA platformai, kurioje naudojamas BTP protokolas. Šis projektas yra dar nebaigtas. Yra tik keletas BTP protokolo realizavimo variantų, kurie nepasižyminti stabilumu. Projektas paliktas tolesniam gerinimui, nes autorius Pavel Tarizek yra pasitraukęs iš šio projekto. Norint įgyvendinti šį projektą, reikia turėti žinių apie JXTA technologiją ir jos programavimo sąsają[15].

1.6. Transakcijų detalizacija

Išnagrinėjus pateiktus standartus, kuriuos taikome šiame transakcijų modelyje, pavyzdžiui: Business Transaction Protocol (BTP) iš OASIS kompanijos, WS-Transaction iš IBM ir Microsoft, bei Activity Service iš OMG. Nustačiau kad, šių transakcijų tikslas sujungti WEB servisus su atitinkamomis programomis ir gauti iš jų reikiamus rezultatus. Iš šių paminėtų transakcijų, BTP yra labiausiai naudojamas ir pakankamai tobulas. Todėl galime palyginti tradicines ir verslo transakcijas naudojant ACID savybes pavaizduotas 1 lentelėje.

1 lentelė. Tradicinių ir verslo transakcijų savybių palyginimas.

Savybės	Tradicinės transakcijos	Verslo transakcijos
Atominiškumas (Atomicity)	Būtinasis; viskas arba nieko	Neišskus; kartais pageidaujamas, kartais reikalinga tik dalis
Nuoseklumas (Consistency)	Būtinasis	Būtinasis; laikinas nesuderinamumo reguliavimas
Atskyrimas (Isolation)	Būtinasis; pasikeitimai nematomi, kol transakcija pilnai neįvykdoma	Neprivalomas; kiekvienas servisas kontroliuoja duomenų pateikimą
Ilgamžiškumas (Durability)	Būtinasis; veiksmas išlaikomas	Būtinasis, bet dažniau naudojama kaip atominė savybė; keletas dalių kintami.
Turinio pateikimas	Neprivalomas	Būtinasis

1.7. Cohesion

Tai pirmasis produktas, kuris realizuoja programinį modulio valdymą ir procesų sinchronizaciją šiandieninėje heterogeninėje įmonėje[23]. “Cohesions” yra pritaikytas taip, kad būtų žymiai pagerintas ir padidina efektyvumą tarp verslo sandorių, kurie koordinuoja daugialypes (įvairiapusiškas) programas. Šioje taikomojoje programoje yra panaudojamas greitas perėjimas nuo vienos būsenos prie kitos realiu laiku, vykdant partnerių ir verslo klientų verslo transakcijas realiame programiniame modulyje. Kai verslo procesai sąveikauja tarp sistemų, šis programinis modulis apima visą kompleksą įvairiapusių transakcijų (įskaitant ypatingai svarbius sandorius, kur neteisingai pateikti rezultatai gali įtakoti kainos padidėjimą, atitinkamą riziką ir užsakymus), dėl kurių gali sužlugti netik vidaus bet ir išorės operacijos, o tuo pačiu tai paliečia klientus ir palieka pėdsaką verslo reputacijoje.

Šiame programiniame modulyje (Cohesion) įgyvendintas OASIS Verslo transakcijos protokolas (BTP) ir Web Servisų Koordinavimas + Transakcijų palaikymas. Šis programinis modulis yra pritaikytas dirbti JAVA aplinkoje ir palaiko

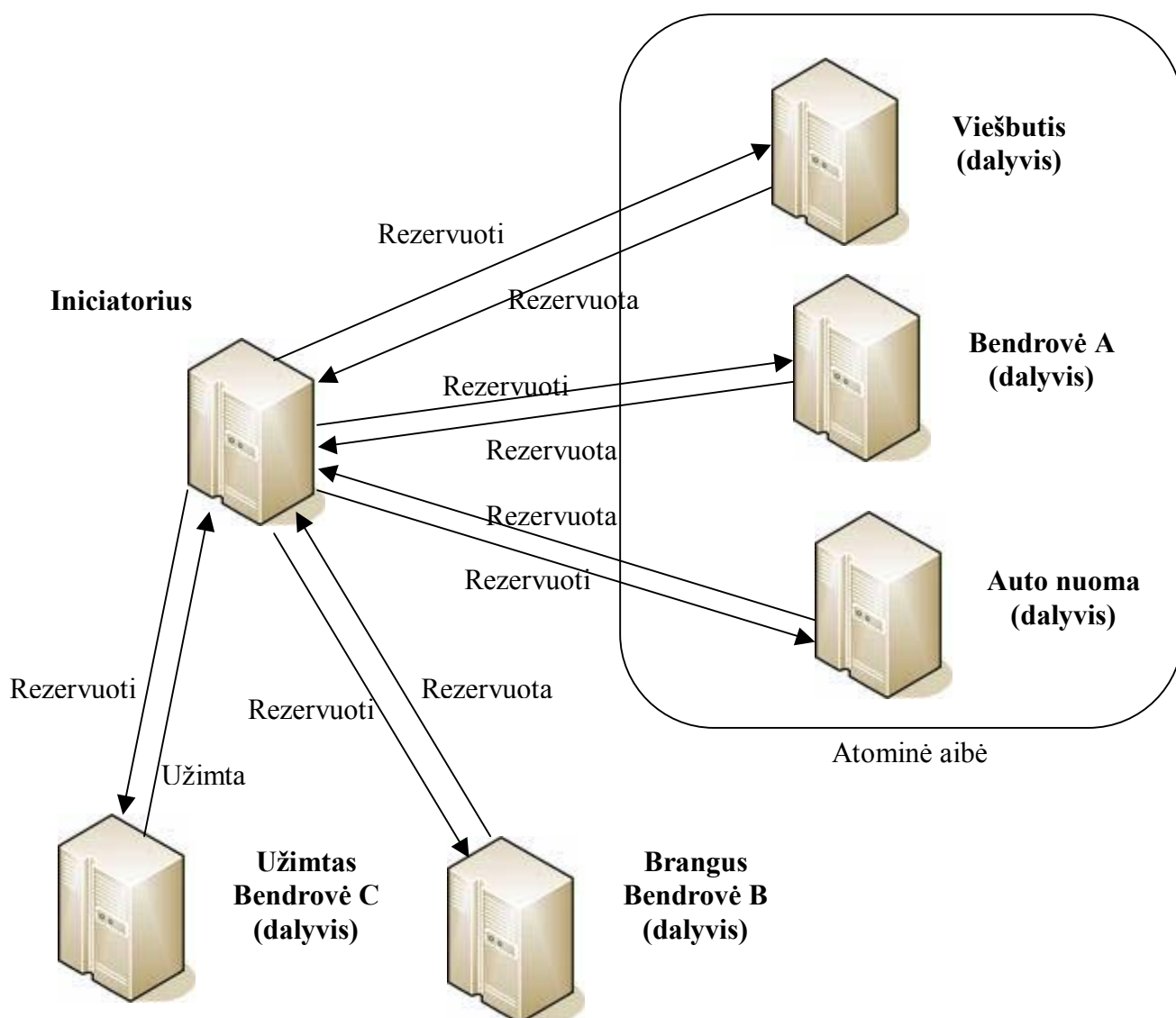
keletą skirtingų Web servisų priemonių, kurios yra paplitusios ir valdomos GRID servisų aplinkoje. “Cohesion” programiniai moduliai yra visiškai susiję su BPEL ir BTP pagrindiniais procesais, kurie supaprastina šių modelių valdymą ir automatinį kompensavimo mechanizmą bei atstatymo būseną. Joje taip pat yra realizuotas standartinis SOAP/HTTP protokolų palaikymas. Šis produktas yra komercinis ir norint jį plėtoti reikia turėti gerų Javos programuotojų įgūdžių, o taip pat įvairiapusių žinių apie reliacines duomenų bazes, Web servigus.

Programinio modelio realizacijai palaikyti reikalingas Tomcat 4.1.x serveris ir Axis papildomas paslaugų paketas SOAP protokolo palaikymui.

2. BTP transakcijų realizavimas

2.1. BTP transakcijų realizavimo pavyzdys

Pirmoje lentelėje mes apibendrinome transakcijos savybes atitikimą verslo transakcijoms. Visose verslo transakcijose, atskyrimo savybė yra neprivaloma. Keletas verslo transakcijų reikalauja, kad visa transakcija būtų pateikta kaip vientisa (atominė) operacija, kol kitos verslo transakcijų klasės pareikalaus atominės savybės, kurios taip pat taps neprivalomomis. BTP suderina abiejų rūšių transakcijų poreikius. BTP atominės verslo transakcijos, arba atomai, labiau vadinamos kaip tradicinės verslo transakcijos, kurios pasižymi tuo, kad turi neprivalomas transakcijų atskyrimo savybes. 1-me paveiklėlyje pavaizduotas bendras transakcijų veikimo mechanizmas.



1 pav. Kelionių agentūros atominė reakcija

Šią situaciją mes galime pavaizduoti taip: klientas užsisako keletą laisvų vietų iš vienos arba kelių oro skrydžių bendrovių (pavyzdyje jos pavaizduotos, kaip oro skrydžių bendrovės A, B ir C), išsinuomoja automobilį, užsisako viešbutį. Sakykime, kad kai kurios oro skrydžių bendrovių nusprendžia atsisakyti kliento transakcijų (skaitykime, kad nėra laisvų vietų), o tuo tarpu kitos transakcijos jau rezervuoja vietas pagal paklausimus (A ir B bendrovės rezervavo vietas ir patvirtino transakcijas).

Taigi C oro skrydžių bendrovė negali patvirtinti rezervacijos kaip ištisinę dėl vietų trūkumų, kadangi klientui reikalinga atominė aibė (t.y. užsakyti turi būti visi), siekiant įgyvendinti verslo logiką. Todėl galutinėje pasėkoje, atominė transakcija negali visiškai pasidengti ir tolimesnis transakcijos vykdymas nebetenka prasmės. Dar blogiau gali atsitikti taip, kad keletą oro skrydžių bendrovių jau rezervavo vietas (A ir B bendrovės), todėl tik viena bendrovė (šiuo atveju C) nerezervavo reikalingų vietų. Tai šiuo atveju matome, kad susilpnintas visos transakcijos efektas, kuri inicijavo kelionių agentūra. Tokiu atveju siekiant išspręsti šią problemą, programos veikimo logika turėtų būti tokia, kad sukuriama apgalvota atominė transakcija ir tuo pačiu sukuriama apjungimo transakcija, kuri patvirtins galutines transakcijas.

Tradicinėse transakcijose, transakcijų valdytojas atšauks (sugražins) transakcijas, jei bet kuris šaltinių valdytojas negalės patvirtinti arba pateikti transakcijos. BTP transakcijose nėra visai taip. Dalyviai visų pirma turi patvirtinti transakcijas ir tik tada jie gali užbaigti transakciją, kuri pavadinama 'patvirtinimas pateiktas'(transaction-confirmed). Tokiu atveju patvirtinimas turi būti pateiktas visų dalyvių, iš kurių yra reikalingi duomenys.

Atominės transakcijos tai verslo transakcijos, kuriose visi dalyviai turi sutikti su tuo, kad transakcija bus įvykdyta tik tuomet, kai visi dalyviai turės garantiją kad jie sėkmingai gali užbaigti transakciją. Jei bet kuris iš dalyvių negalės patvirtinti paklausimo, visa transakcija atšaukiama. Kadangi BTP transakcijoms nereikia tikslaus surišimo, tai kiekvienas serviso dalyvis gali nuspręsti kaip šią transakciją pritaikyti savoje sistemoje.

Apjungimo transakcijos yra tokios, kur visos atskiros dalys neprivalomai turi gauti sutikimą patvirtinant pakeitimus prieš tai pilnai patvirtinant transakciją. Tikrai daliai šios aibės transakcijų dalių reikia patvirtinimo. Daliai transakcijų, kurioms reikia transakcijų patvirtinimo, gali būti užbaigtos nelaukiant, kol apie patvirtinimą bus pranešta, aišku atsižvelgiant į verslo logiką.

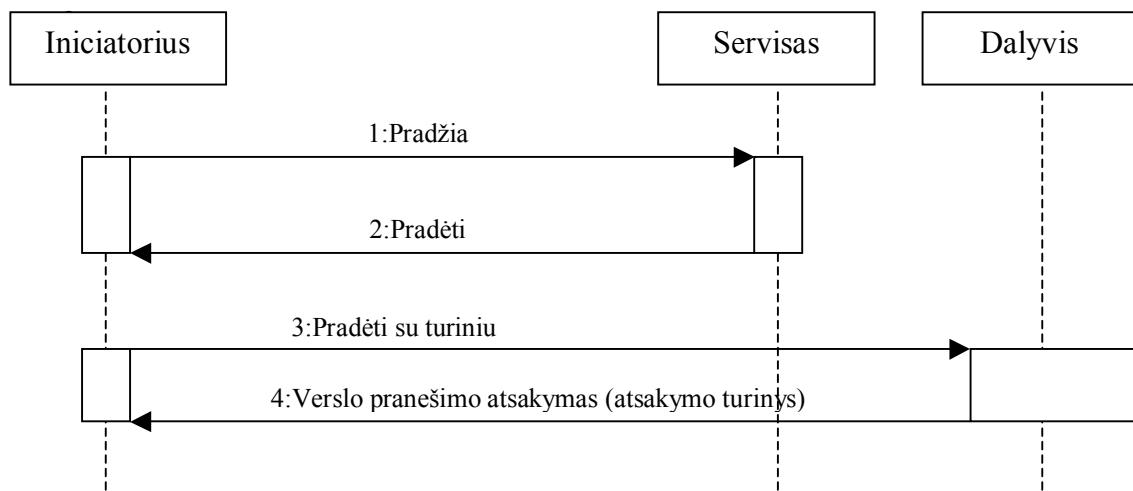
2.2. Transakcijų turinio pateikimo realizavimas

WEB servisų programos, kurios jungiasi su kitomis verslo programomis egzistuojančiose skirtingose organizacijose tarpusavyje keičiasi transakcijomis ir gali užtrukti tam tikrą laiko tarpą. Dauguma servisų turi įdiegę skirtingas technologijas skirtingose platformose (pvz. UNIX, Windows ir t.t.) ir kaupia duomenis įvairiarūšiuose duomenų bazėse. Kai transakcija paliečia šiuos skirtingus servišus, transakcijos turinys pereina iš vieno servišo į kitą. Pavyzdžiui viena iš sistemų yra vykdoma Java transakcijų servisų – JTS (JAVA Transaction Service) aplinkoje, JTS apibrėžia, kaip galima greitai keistis bei perduoti duomenimis iš vieno valdymo centro į kitą naudojantis IIOP (Internet Inter ORB Protocol). Keičiantis pranešimais, WEB transakcijų valdytojai turi prisiderinti prie XML transakcijų, jei mes nagrinėsime WEB servisų naudojamus protokolus.

Taigi, išnagrinėję apžvelgtas savybes pamatėme, kad norint išspręsti šias problemas reikalingas transakcijų modelis, kuris padėtų suvaldyti transakcijas. Modelyje transakcijas galime pavaizduoti kaip keletą verslo procesų (vienetų) keičiantis turiniu tarpusavyje, o turinys paskirstytas kiekvienam atliktam veiksmui. Šių veiksmų vykdymas turi būti koordinuotas, tačiau galimi ir individualūs sprendimai, kurie gali nuspręsti, ar vykdyti tolimesnę transakcijos seką ir pateikti ją kaip užbaigtą, ar palaukti kol turėsime visus duomenis ir ją užbaigsime pasižymėję kaip galutinę transakciją. Dažnai, transakcijos modeliai būna pritaikyti dideliame skaičiui WEB servisų programoms – kurios pasižymi plačiomis transakcijomis ir kurie atskiruose servišuose saugo duomenis vienoje duomenų bazėje arba naudojama duomenų bazė su analogiška programine įranga. Šiame modelyje panaudotos visos savybes, kurios yra suprantamos kaip ACID (nedalomumo, vientisumo, izoliavimo, išliekamumo (angl. Atomicity, Consistency, Isolation, Durability)). Šias savybes galime apibūdinti taip:

- Nedalumas - tai reiškia, kad transakcija turi būti atominė (viskas arba nieko) arba įvykdoma arba ne. Jei nors viena dalis nepavyko, kitos likusios transakcijos programos dalys negali būti įvykdytos. Jei dėl kokių nors priežasčių transakciją negali būti užbaigta, visi pakeitimai turi būti gražinti į ankstesnę būseną iki tol, kol buvo pradėta minima transakcija panaudojant transakcijos operacija „grąžinti“ (angl. rollback).

- Vientisumas – bendri transakcijos resursai turi išlikti vientisi. Transakcija turi pervesti sistemą iš vienos vientisos būsenos į kitą. Vientisumo būsenos prasmė yra ta, kad tam tikros verslo taisyklės buvo išpildytos.
- Izoliacija – kiekviena transakcija gali pasiekti resursus, jei nėra kitų konkurencinių transakcijų; transakcija įvykdoma be jokių įsikišimų iš kitų transakcijų ir turi išskirtinį priėjimą prie resursų. Transakcijos nuosekliai vykdomos viena paskui kitą ir jos nemato kitų transakcijų rezultatų tol, kol jos nepabaigiamos.
- Išliekamumas – transakciją baigus, visos būsenos ir atlikti pakeitimai turi išlikti duomenų saugykloje.

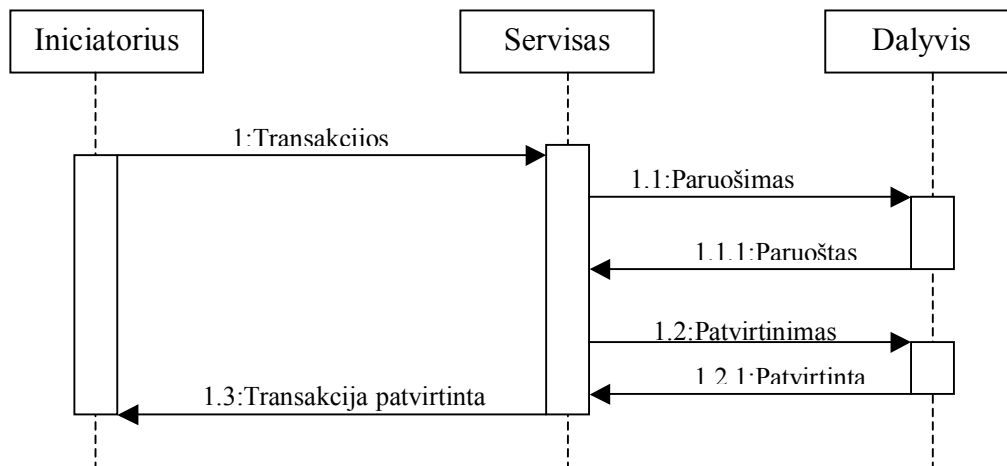


2 pav. BTP pranešimų ir turinio pateikimo apžvalga

2.3. Transakcijų patvirtinimo realizavimas

Jei transakcija vykdoma vietinėje ar paskirstytoje aplinkoje, privalo būti palaikomos ACID savybės. Tai sunkiau pasiekama, jei transakcijos resursai yra skirtingose autonominiuose mašinos. Kai transakcija turi būti patvirtinama, reikia įsitikinti, kad pakeitimai buvo atlikti abiejose sistemose. Pagrindinė problema, su kuria susiduriama vykdant transakcijas paskirstytoje aplinkoje, yra ta, kad bet kuris mazgas gali patirti nesėkmę dėl įvykusios klaidos, kadangi duomenys yra fiziškai skirtingose mašinos, nesėkmę patyrusio mazgo aptikimas yra sudėtingas. Pavyzdžiui, jeigu transakcija patvirtina pakeitimus vienoje sistemoje, bet kitoje sistemoje įvyko klaida, prieš patvirtinant transakciją ji turėtų būti pervesta į nepilnumo būseną. Tokiu atveju dalis darbo buvo atlikta sėkmingai, kita pusė – ne. Tai prieštarauja nedalumo

savybei, kuria apžvelgėme anksčiau. Šios problemos sprendimas būtų patvirtinimo (angl. Commit) protokolo panaudojimas. Dažniausiai pasitaikantis patvirtinimo protokolas transakcijų sistemose yra vadinamas dviejų fazių patvirtinimu (angl. Two phase commit), o jei reikia dar didesnio patikimumo naudojamas trijų fazių patvirtinimo protokolas.



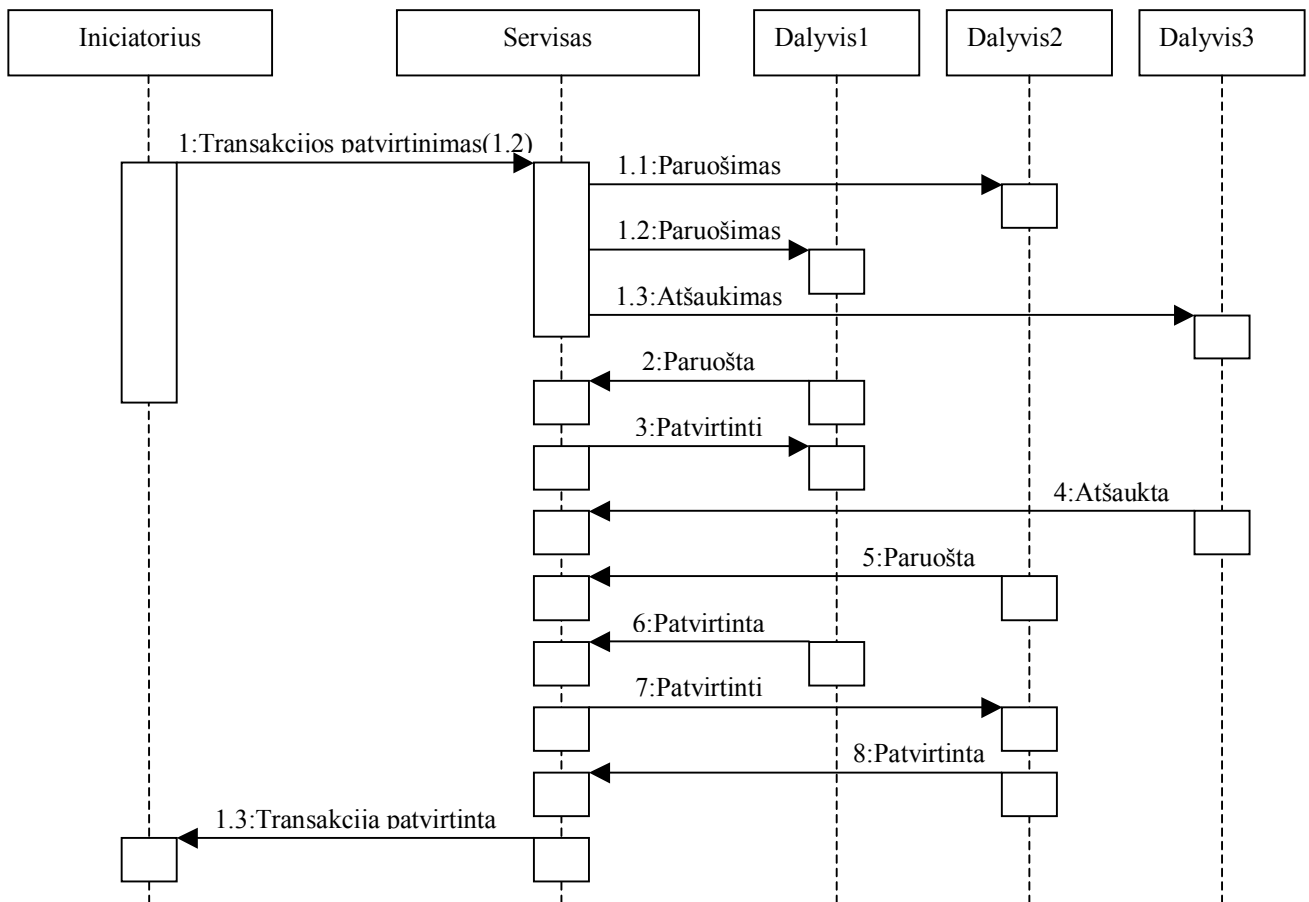
3 pav. Atominės transakcijos dvifazio patvirtinimo realizavimas.

Dviejų patvirtinimo fazių (kaip teigia pats pavadinimas) protokolas susideda iš dviejų fazių. Žemiau pateikiamas tipiškas jų aprašymas.

Pirmoji fazė. Pradedamas iniciatoriaus procesas, (dažniausiai ten, kur transakcija yra sukuriama pirmą kartą), įrašomas „Pradėti“ (angl. Begin) įrašas į peržiūros žurnalą, toliau siunčiamas „Pasiruošti“ (angl. Prepare) pranešimas dalyviams ir pereinama į laukimo būseną. Siunčiamas pranešimas taip pat gali turėti unikalų transakcijos identifikatorių (TID), kurį toliau naudos visi pranešimai. Kai dalyvis gauna „Pasiruošti“ (angl. Prepare) pranešimą, jis patikrina ar gali patvirtinti transakciją. Jeigu dalyvis gali patvirtinti transakciją, įrašo „Paruoštas“ (angl. Prepared) įrašą į savo peržiūros žurnalą, siunčia „Patvirtinti“ (angl. confirm). Kitu atveju, vienareikšmiškai nusprendęs nutraukti transakciją, dalyvis įrašo „atšaukti“ (angl. Cancel) įrašą į savo peržiūrų žurnalą ir siunčia „atšaukimas“ (angl. cancelled) pranešimą iniciatoriui. Jis toliau transakcijos neseka.

Antroji fazė – Iniciatorius gavęs pranešimus iš visų dalyvių, sprendžia ar transakciją patvirtinti, ar nutraukti ir įrašo savo sprendimą į savo peržiūrų žurnalą. Jeigu sprendimas yra pravirtinti, tuomet siunčiamas „transakcijos patvirtinimas“ (angl. Confirm-transaction) pranešimas visiems dalyviams, kitu atveju siunčiamas „atšaukti transakcijas“ (angl. cancel-transaction) pranešimas tiems dalyviams, kurie

pranešė, kad jie pasirengę patvirtinti transakciją. Galiausiai iniciatorius įrašo transakcijos pabaigą savo peržiūrų žurnale, o dalyviai pabaigia transakciją remdamiesi iniciatoriaus sprendimu ir įrašo rezultatus savo peržiūrų žurnaluose.



4 pav. Apjungimo transakcijos dvifazis realizavimas tarp kelių verslo dalyvių

Kaip diskutavome anksčiau, BTP nereikalauja, kad klientas sulauktų transakcijos patvirtinimo išlaikyti programos duomenis. Pasiruošęs vartotojas gali atlikti programinius pakeitimus duomenų bazėje ir pateikti juos kaip laikinai matomus (“laikinas efektas”). Tokiu atveju paruošiama kompensacinė transakcija (skaitliuko efektas), esant reikalavimui ją atšaukti.

Kadangi naudodami Web paslaugų transakcijas jos gali užtrukti ir pereiti į būseną kaip nepasiekusi serviso arba perkrauta, dalyvis ir jo iniciatorius gali tarpusavyje nepasikeisti reikiama informacija reikiamu metu. Programoje numatyta ši problema, kuri gali įvykti dėl nepastovaus ryšio. Dalyvis gali riboti įsipareigojimą savo iniciatoriui (informacijos siuntimu - ‘paruoštas’) vykdydamas savarankiškai atgalinį veiksma, naudodamas programos duomenų patvirtinimą arba atšaukimą. Tokie sprendimai labai dažnai priimami savarankiškai kaip euristiniai, kuomet

naudojamas standartinis dvifazis protokolas. Esminis skirtumas yra tas, kad kiti transakcijos modeliai tokius euristinius sprendimus svarsto labai retai, o BTP numano tokias problemas ir leidžia vartotojams pasirinkti, kaip ilgai jie bus pasiruošimo stadijoje.

Kas atsitiks, jei dalyvis informuotas laukti patvirtinimo, priims savarankišką sprendimą atšaukti, neskaitant to, kad negavo patvirtinimo numatytu laiku? O jeigu, po savarankiško patvirtinimo atšaukti transakciją, iniciatorius išsiunčia patvirtinimo pranešimą? Tokiu atveju programiniame modelyje tai numatyta, kad iniciatorius galiausiai suvoks ir įrašys pakeitimus, informuodamas likusius dalyvius, ir išsiūs jiems 'paneigimo' pranešimus.

BTP skirtumai tarp esamų ir kitų dviejų gali būti pateikiami taip:

- BTP transakcijose, programa patenka į būseną kaip patvirtinta
- BTP transakcijos pasirūpina programos priemonėmis "pritaikyti" protokolą, perduodant (laiko) apribojimus iniciatoriui. Šie apribojimai įtraukiami kaip standartiniai BTP apribojimai, kaip ilgai vartotojas turi laukti, prieš priimant savarankiškus sprendimus.
- BTP transakcijos nereikalauja blokavimo. Jos gali vadovautis paneigimais, jei dalyvis atlieka savarankiškus sprendimus. Skirtingai nuo kitų modelių, nagrinėjant euristinius sprendimus kaip retus įvykius, BTP numato tokias problemas ir bando pasirūpinti jomis.

2.4. Realizavimo įrankiai

2.4.1. Oracle duomenų bazė

Oracle technologija tai programinių priemonių ir sprendimų rinkinys, skirtas įmonės lygio informacinių sistemų kūrimui. Sudėtingų kompleksinių sprendimų įgyvendinimui yra skirtas visas spektras programinės įrangos, kurią galima suskirstyti į keletą kategorijų pagal jų funkcinę paskirtį.

Oracle duomenų bazių valdymo sistemos vystėsi nuo hierarchinės struktūros modelio per tinklinę į reliacinę struktūros modelį. Labiausiai pripažintas modelis yra reliacinis modelis. Oracle šio modelio pagrindu sukūrė objektinį - reliacinį DBVS modelį, kuris leidžia išsaugoti sudėtingus verslo modelius reliacinėje duomenų bazėje.

Oracle pastebėjo objektiškumo palaikymo reikalingumą savo duomenų bazėse, taigi pateikė kelias iniciatyvas tam įdiegti. Oracle pristatė Java Virtual Machine (Java

virtualiąją mašiną) duomenų bazės branduolyje; naujas duomenų konstrukcijas, tokias kaip Object Type (objektų tipai) arba User-Defined Type (UDT – vartotojo aprašyti tipai); ir, ko gero svarbiausia, objektų vaizdus (object view). Objektų vaizdai davė duomenų bazių administratoriams ir programuotojams galimybę kurti duomenų bazės vaizdus iš UDT, kuriuose rezultatų rinkinys yra objektų rinkinys vietoje reliacinių lentelių rinkinio.

Ši objektų vaizdų koncepcija buvo svarbiausia, todėl kad ji suteikė Oracle daliviams galimybę pakeisti jų naudotas programas, raportavimo įrangą ir sandėlius.

Objektai vis dar yra didelė Oracle strategijos dalis, jie yra visiškai palaikomi Oracle Database 10g programiniame pakete. Šiandien, industrija mažiau sukcentravusi dėmesį į objektų naudojimą (pagrindinė priežastis – ta, kad tai jau padaryta). Taip pat industrija reikalavo iš duomenų bazių objektų palaikymo devintam dešimtmety, kaip dabar reikalauja XML galimybių. Taip pat ši technologija yra pritaikyta kito verslo protokolo BPEL realizavimui. BPEL realizavimo grafiniai atvaizdai pateikti priede 29 ir 30 pav.

Šiame darbe ši DB realizuota duomenų saugojimui bei duomenų pasikeitimui tarp verslo dalyvių. Kad galėtumėme Oracle technologiją pritaikyti sukurtai BTP protokolų realizacijai reikalinga ojdbc14.jar klasė (biblioteka). Ši biblioteka leidžia palaikyti Oracle DB technologiją nuo 9-osios versijos.

2.4.2. Apache Tomcat

Apache Tomcat – tai Java parašytas daugiaplatforminis savarankiškas HTTP serveris, palaikanti servletus ir JavaServer Pages (JSP)[21]. Tomcat dirva įvairiose OS platformose: Linux, Windows, Solaris ir kitose. JSP puslapius į servletus kompiliuoja “Jasper” kompiliatorius. Įvairiuose testuose tarp serverių palaikančius servletus, Tomcat yra tarp lyderių. Tomcat yra oficiali pavyzdinė servletų ir JSP technologijų realizacija.

Apache palaiko šiuos WEB servisus: AXIS, AXIS2, WS-Commons, EWS, JaxMe, jUDDI, Kandula, Mirae, Muse, Pubscibe, Sandesha, Scout, SOAP, Synapse, TSIK, Tuscany, Woden, WSIF, WSRF, WSS4J, XML-RPC, o tai leidžia praplėsti atliekamo darbo galimybes/

Apache Tomcat taip pat apima įrankius konfigūracijai ir valdymui, be to gali suformuoti, redaguoti konfigūracijos bylas, kurios yra apibūdinamos XML standartais.

Šiame darbe realizuojamas duomenų perdavimas SOAP protokolu, be to kitos stambios kompanijos taip pat nusprendė pasirinkti Web servisų Apache realizaciją, todėl išnagrinėjus įmanomas Tomcat galimybes nuspręsta pasirinkti Apache Tomcat realizaciją Web servisų paslaugoms.

2.4.3. AXIS paslaugų paketas

Apache AXIS yra atvirojo kodo programa, kuri paremta XML web serviso paslauga. Šis paslauga derinasi prie JAVOS ir C++ programinių kodų SOAP servisui palaikyti bei sukurti papildomi įrankiai taikomosios programos kodavimui realizuoti diegiant Web servisu. Vartotojas vartojantis AXIS paslaugų paketą gali sukurti tarpinius skaičiavimus paskirstytoms skaičiavimo sistemoms.

Kai yra naudojama AXIS paslauga skirta JAVOS taikomajam kodui palaikyti diegiant web servisu, tai galima realizuoti dviem būdais. Vienas iš jų pasinaudoti JWS failais, kitas savarankiškas servisų realizavimas. Savarankiškas web servisų realizavimas yra paremtas pasinaudojant WSDD (Web Service Deployment Descriptor) sintaksės failu. Šių failų realizavimo WEB servise pavyzdys pateiktas priede 34 ir 35 pav.

Šiame darbe SOAP paslaugų realizavimas yra paremtas WSDD failais, kurie leidžia realizuoti Web servisu. Failai gali būti skirstomi sukuriant WSDD failą web servisu palaikyti ir diegiant atskiras javos klases (kitai vadinamomis javos bibliotekomis). Realizuojant šią sistemą pasinaudota šiomis AXIS klasėmis (bibliotekomis):

- saaj.jar – SOAP pranešimams sukurti
- jaxrpc.jar – realizuoti Web serviso paslaugą
- axis.jar – palaikyti SOAP transportinį protokolo lygį
- wsdl4j.jar – kurti WSDL dokumentus transportavimui paskirstyti
- activation.jar – duomenų failams perduoti SOAP protokolu
- xmlsec.jar – duomenų perdavimo saugumui užtikrinti

AXIS paslaugų realizavimo vaizdas pateiktas priede, 28 pav.

2.4.4. SOAP protokolas

SOAP (Simple Object Access Protocol) protokolas yra suprantamas kaip paprastas kreipties į objektus protokolas. SOAP yra XML kalba pagrįstas standartas, apibrėžiantis sąveiką tarp paslaugų ir jų vartotojų (kitais tariant, SOAP yra į paslaugas orientuotos architektūros protokolas). SOAP 1.2 versija buvo rekomenduotas W3C organizacijos. Kartais ši technologija yra sumaišoma su SOA arba į servisą orientuotą architektūrą, bet SOAP technologija yra kitokia nei palyginsime su SOA.

Šiame darbe mes galėjome pasirinkti ir kitą analogišką SOAP perdavimo paslaugą XML-RPC, kur SOAP susiformavimo iš XML-RPC protokolo, įgydamas žymiai sudėtingesnę sintaksę, tačiau didesnes galimybes. Tačiau atsižvelgiant į šio darbo specifiką ir pateikiamų duomenų įvairumą, nuspręsta panaudoti SOAP technologiją, nes SOAP yra W3C rekomendacija, skirtingai nuo XML-RPC, kuris nėra įgijęs standarto statuso. XML-RPC yra specifikuotas standartas, turintis paprastesnę sintaksę, tačiau nors SOAP yra sunkiau diegiamas nei XML-RPC, pagrindinis protokolų palyginimas pateiktas 2-oje lentelėje. Abu protokolai paremti metodų vardų, jų parametrų konvertavimu į XML užklausos žinutės suformavimą.

2 lentelė. XML-RPC ir SOAP protokolų palyginimas[10].

Savybė	XML-RPC	SOAP
Pagrindiniai kintamųjų tipai	Taip	Taip
Struktūros	Taip	Taip
Masyvai	Taip	Taip
Struktūrų ir masyvų vardai	Ne	Taip
Detalus klaidų apdorojimas	Taip	Taip
Paprastumas/greitas sintaksės išmokymas	Taip	Ne
Simbolių kodavimo apibrėžimas	Ne	Taip (US-ASCII, UTF-8, UTF-16)
Duomenų tipų apibrėžimas	Ne	Taip
Gavėjo apibrėžimas	Ne	Taip
Reikalauja kliento suderinamumo	Ne	Taip
Specifinės žinučių apdorojimo instrukcijos	Ne	Taip

SOAP naudoja taikomąjį internetinį (HTTP/SMTP) sluoksnio protokolą transportiniame lygmenyje. Ir SMTP, ir HTTP yra panaudoti transportuoti XML

pranešimus, tačiau HTTP yra plačiau naudojamas ir puikiai dera su šiandienine interneto struktūra, nes HTTP gerai dirba su naudojamomis tinklo ugniasienėmis (firewalls).

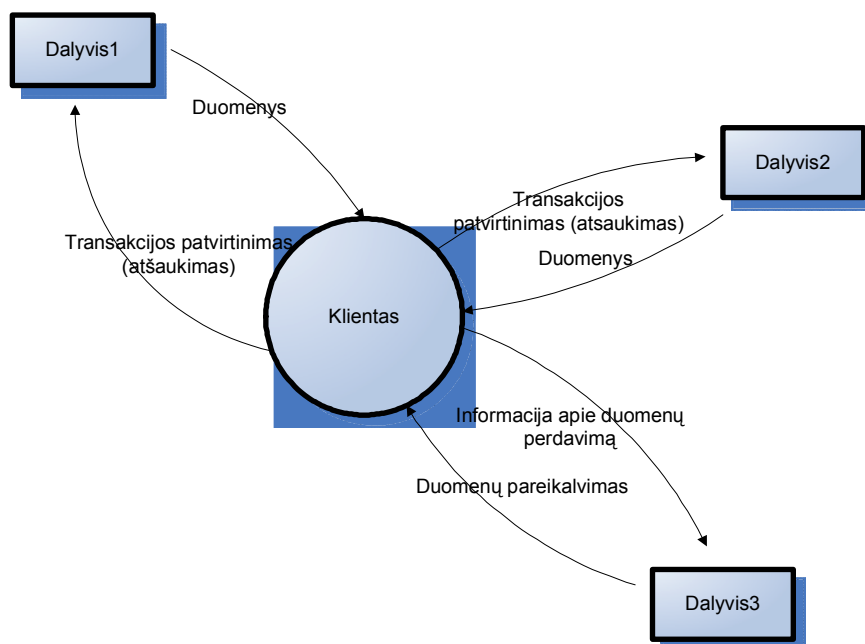
Viena didesnių galimybių, kad galime pasinaudoti HTTPS sluoksniu. HTTPS protokolas kaip ir HTTP yra taikomajame lygmenyje ir tuo pačiu naudoja užkoduotą (encrypt) transporto protokolą ar inicijuojant paprastą sujungimą ar inicijuojant abipusius sujungimus reikalaujančius autentiškumo nustatymo.

Norint panaudoti SOAP servisą jį reikia realizuoti su pasaulinio tinklo paslaugų aprašymo kalbos (WSDL) dokumentu [7]. Pagrindinis WSDL dokumento elementas yra "definitions", kuris turi į jį įtrauktus elementus "Types", "Message", "PortType", "Operation", "Binding", "Service", "Port", "Data Schema". WSDL dokumento realizavimo šiame darbe pavyzdys pateiktas priede 33 pav.

BTP pranešimas yra susietas su protokolu SOAP. BTP protokolo panaudojimas SOAP protokolu yra apibrėžtas OASIS komiteto specifikacijoje. Joje yra apibrėžti abstraktūs pranešimai, kurie prijungti prie komunikacijos protokolų. SOAP protokolo galėjimas koordinuoti sandorius per daugialypes autonomines paslaugas, bei XML pranešimų apsikeitimas ypač tiko realizuoti Web servisų paslaugų architektūrą. BTP protokolas yra suprojektuotas taip, kad galėtų būti pritaikytas prie bet kurio kito protokolo, be to daugelyje sistemų (naudojančioms BTP protokolą) yra realizuotas SOAP 1.1/HTTP sujungimas.

3. Produkto analizė

3.1. Produkto veiklos kontekstas



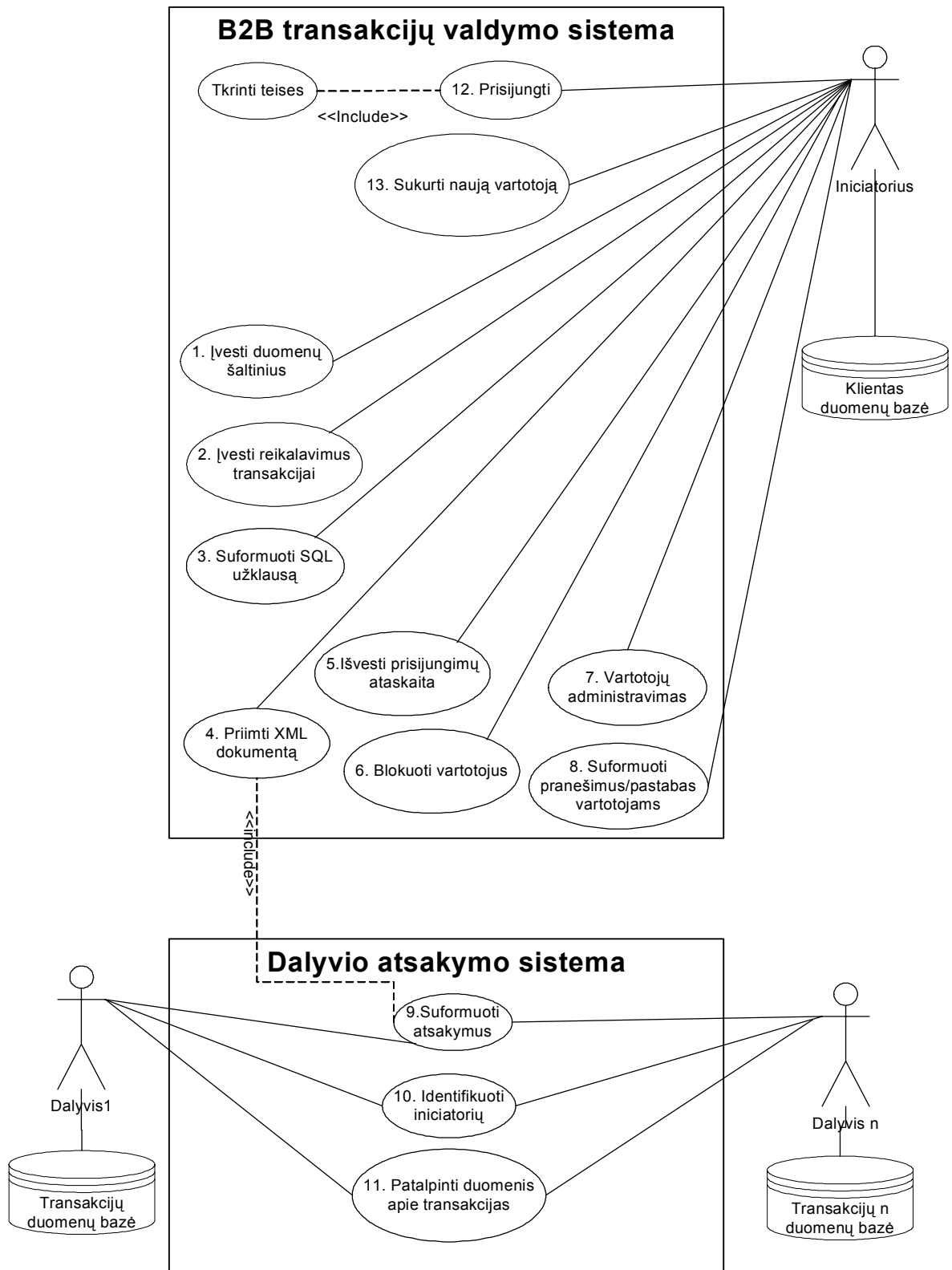
5 pav. Veiklos kontekstas

3.2. Veiklos padalinimas

Lentelė 3: Veiklos padalinimas

Eil. .	Įvykio pavadinimas	Įeinantys/išeinantys informacijos srautai
1.	Dalyvis3 perduoda duomenis transakcijos reikalavimui	Transakcijos duomenys (in)
2.	Klientas patvirtina apie galimą transakcijos vykdymą	Dalyvių pasiekiamumas (out)
3.	Prašymas leisti vykdyti transakciją	Transakcijos prašymas (in)
4.	Dalyviai (dalyvis1 ir dalyvis2) informuoja apie galimą transakcijos vykdymą	Transakcijų vykdymo-atšaukimo patvirtinimas (out)
5.	Dalyviai perduoda duomenis	Duomenų perdavimas (in)
6.	Informacija apie sėkmingą-nesėkmingą duomenų perdavimą	Sėkmingas-nesėkmingas perdavimas (out)
7.	Transakcijų įvykdymo kontrolė	Duomenys apie sėkmingą transakcijos įvykdymą (out)

3.3. Produkto veiklos sfera



6 pav. B2B transakcijų valdymo sistemos panaudojimo atvejų diagrama

3.4. Reikalavimų specifikacija

Lentelė 4: Panaudojimo atvejis “Įvesti duomenų šaltinius”

Nr.	1
Pavadinimas:	Įvesti duomenų šaltinius
Vartotojo/aktoriaus pavadinimas:	Iniciatorius
Aprašas:	Atidaromas langas, kuriame įvedami duomenų šaltiniai, priskiriami transakcijos reikalavimai
Prieš sąlyga:	Reikia turėti informacijos apie duomenų šaltinį.
Sužadinimo sąlyga:	Klientas įveda inicijuoja šaltinių įvedimą.
Po sąlyga:	Duomenys įtraukiami į DB, lange parodomi aktyvūs prisijungimai.

Lentelė 5: Panaudojimo atvejis “Įvesti reikalavimus transakcijai”

Nr.	2
Pavadinimas:	Įvesti reikalavimus transakcijai
Vartotojo/aktoriaus pavadinimas:	Iniciatorius
Aprašas:	Atidaromas langas, kuriame parenkami transakcijos laiko limitai.
Prieš sąlyga:	Reikia būti susipažinusiems su transakcijų tipais, galimais rizikos laipsniais. Turi būti įvesti šaltiniai ir parinkti transakcijos tipai
Sužadinimo sąlyga:	Klientas remdamasis verslo logika suformuoja transakcijos laiko limitą
Po sąlyga:	Duomenys įtraukiami į DB, ekrane pateikiami pranešimai apie įvestas transakcijas.

Lentelė 6: Panaudojimo atvejis “Suformuoti SQL užklausa”

Nr.	3
Pavadinimas:	Suformuoti SQL užklausa
Vartotojo/aktoriaus pavadinimas:	Iniciatorius
Aprašas:	Iniciatorius suformuoja SQL užklausoje tekstą reikalingiems duomenims, nurodant kiekvienam šaltiniui reikalingą SQL užklausa.
Prieš sąlyga:	Turi būti suvesti duomenų šaltiniai.
Sužadinimo sąlyga:	SQL užklausoje inicijavimo įvedimas.
Po sąlyga:	SQL užklausa patalpinama į DB.

Lentelė 7: Panaudojimo atvejis “Priimti XML dokumentą”

Nr.	4
Pavadinimas:	Priimti XML dokumentą
Vartotojo/aktoriaus pavadinimas:	Iniciatorius
Aprašas:	Iniciatorius inicijuoja transakcijos vykdymą.
Prieš sąlyga:	Turi būti nurodyti duomenų šaltiniai ir reikalavimai transakcijai, bei turi būti palaikomas ryšys su DB
Sužadinimo sąlyga:	Iniciatorius inicijuoja transakcijos vykdymą.
Po sąlyga:	Duomenys įtraukiami į DB, išvedami pranešimai apie transakcijos vykdymo eigą – transakcijų įvedimo lange. Duomenys siunčiami parinktiems klientams.

Lentelė 8: Panaudojimo atvejis “Išvesti prisijungimų ataskaitą”

Nr.	5
Pavadinimas:	Išvesti prisijungimų ataskaitą
Vartotojo/aktoriaus pavadinimas:	Iniciatorius
Aprašas:	Iniciatorius suformuoja prisijungimų suvestinę.
Prieš sąlyga:	Suformuoja išvestinės išvedimo parametrai.
Sužadinimo sąlyga:	Iniciatorius prašymas suformuoti ataskaitą.
Po sąlyga:	Iniciatorius pateikiama išvestinė.

Lentelė 9: Panaudojimo atvejis “Nepageidaujami šaltiniai”

Nr.	6
Pavadinimas:	Nepageidaujami šaltiniai
Vartotojo/aktoriaus pavadinimas:	Iniciatorius
Aprašas:	Iniciatorius įveda į DB nepageidaujamus šaltinius, kurie neturi būti pasiekiami transakcijos suformavimo metu.
Prieš sąlyga:	Pateikti klientai, kurie įvesti į DB.
Sužadinimo sąlyga:	Pažymėtas nepageidaujamas šaltinis.
Po sąlyga:	Į DB įvedami nepageidaujami šaltiniai, kurie neturėtų dalyvauti transakcijos formavimo procese.

Lentelė 10: Panaudojimo atvejis “Vartotojų administravimas”

Nr.	7
Pavadinimas:	Vartotojų administravimas
Vartotojo/aktoriaus pavadinimas:	Iniciatorius
Aprašas:	Redaguojami, šalinami, sistemos dalyviai.
Prieš sąlyga:	Dalyviai pasirašę sutartis, pasibaigusios sutartys.
Sužadinimo sąlyga:	Informacija apie vartotojo būsenos pakitimą.
Po sąlyga:	Dalyvis pašalintas, įtrauktas į IS.

Lentelė 11: Panaudojimo atvejis “Suformuoti pranešimus/pastabas vartotojams”

Nr.	8
Pavadinimas:	Suformuoti pranešimus/pastabas vartotojams
Vartotojo/aktoriaus pavadinimas:	Iniciatorius
Aprašas:	Iniciatorius išsiunčia pranešimus vartotojams (Pvz. apie numatomus atnaujinimo darbus)
Prieš sąlyga:	Įvykis, kuris svarbus sistemos vartotojams.
Sužadinimo sąlyga:	Įvykio paskelbimas.
Po sąlyga:	Dalyviai gauna pranešimą (-us).

Lentelė 12: Panaudojimo atvejis “Suformuoti atsakymus”

Nr.	9
Pavadinimas:	Suformuoti atsakymus
Vartotojo/aktoriaus pavadinimas:	Dalyvis
Aprašas:	Dalyvis suformuoja XML dokumentą, kuris siunčiamas iniciatoriui
Prieš sąlyga:	Turi būti nurodytos reikalingos SQL užklausos/BTP protokolo parametrai, identifikacijos duomenys.
Sužadinimo sąlyga:	Prašymas pateikti duomenis.
Po sąlyga:	Išsiunčiami atsakymo parametrai.

Lentelė 13: Panaudojimo atvejis “Identifikuoti iniciatorių”

Nr.	10
Pavadinimas:	Identifikuoti iniciatorių
Vartotojo/aktoriaus pavadinimas:	Iniciatorius
Aprašas:	Dalyvis identifikuoja iniciatoriaus teisę gauti duomenis
Prieš sąlyga:	Pateikiamas iniciatoriaus kodas, kuris nori gauti duomenis
Sužadinimo sąlyga:	BTP protokolo parametras siekiantis inicijuoti transakcijos vykdymą
Po sąlyga:	Iniciatoriai siunčiantys SQL užklausas šiam dalyviui, bus įtraukiami į DB ir pažymimi – kaip turintys teisę gauti duomenis.

Lentelė 14: Panaudojimo atvejis “Patalpinti duomenis apie transakcijas”

Nr.	11
Pavadinimas:	Patalpinti duomenis apie transakcijas
Vartotojo/aktoriaus pavadinimas:	Dalyvis
Aprašas:	Dalyvis įtraukia iniciatorius, kuriems buvo išsiūti duomenys į DB.
Prieš sąlyga:	Vartotojas identifikuotas
Sužadinimo sąlyga:	Vartotojo identifikacijos atitikimas
Po sąlyga:	DB atvaizduojami parametrai kurie buvo išsiūsti iniciatoriui

Lentelė 15: Panaudojimo atvejis “Prisijungti”

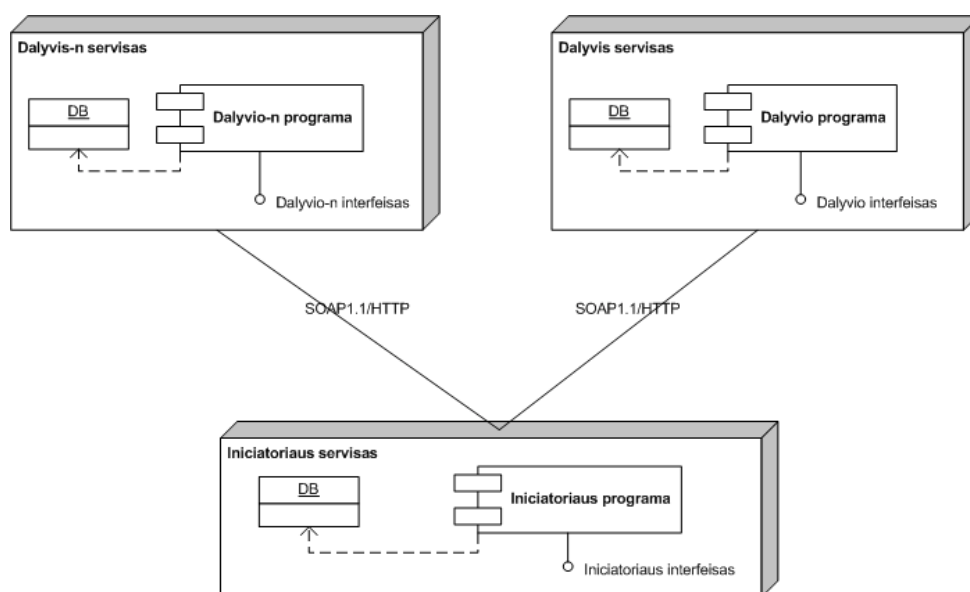
Nr.	12
Pavadinimas:	Prisijungti
Vartotojo/aktoriaus pavadinimas:	Iniciatorius
Aprašas:	Iniciatorius įveda prisijungimo duomenis, jei duomenys teisingi, leidžiama naudotis jam priklausančia DB.
Prieš sąlyga:	Programos paleidimas
Sužadinimo sąlyga:	Vartotojas suveda duomenis ir jungiasi prie sistemos
Po sąlyga:	Vartotojas patenka į pagrindinį programos interfeisą

Lentelė 16: Panaudojimo atvejis “Sukurti naują vartotoją”

Nr.	13
Pavadinimas:	Sukurti naują vartotoją
Vartotojo/aktoriaus pavadinimas:	Iniciatorius
Aprašas:	Iniciatorius įveda prisijungimo duomenis prie DB, suveda vartotojo vardą ir norimą slaptažodį, jei duomenys teisingi, leidžiama sukurti vartotoją.
Prieš sąlyga:	Naujo vartotojo sukūrimo inicijavimas
Sužadinimo sąlyga:	Iniciatorius suveda duomenis ir inicijuoja kurti naują vartotoją
Po sąlyga:	Iniciatoriui sukuriamas naujas iniciatorius.

3.5. Sistemos išdėstymo (deployment) vaizdas.

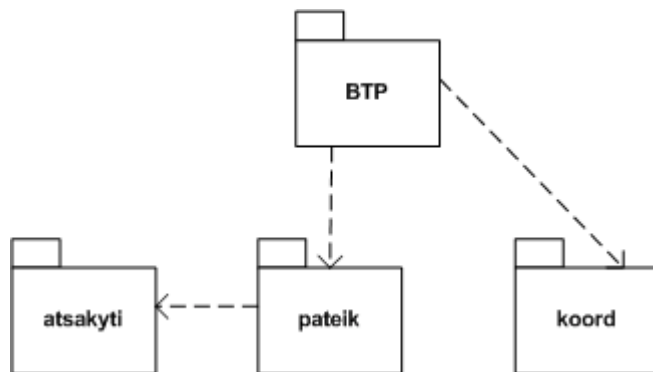
- ✓ Iniciatoriaus posistemė
- ✓ Dalyvio posistemė. Dalyvių posistemių skaičius gali būti neribotas, tai priklauso nuo iniciatoriaus techninių galimybių.



7 pav. Sistemos išdėstymo (deployment) vaizdas

3.6. *Sistemos suskaldymas į paketus*

Paketas “atsakyti” šioje realizacijoje yra pateiktas kaip sistemos dalis, tačiau realizuojant sistemą autonominiuose šaltiniuose, šis paketas talpinamas į atskirą serverį, kuris palaiko BTP protokolą aplankus.



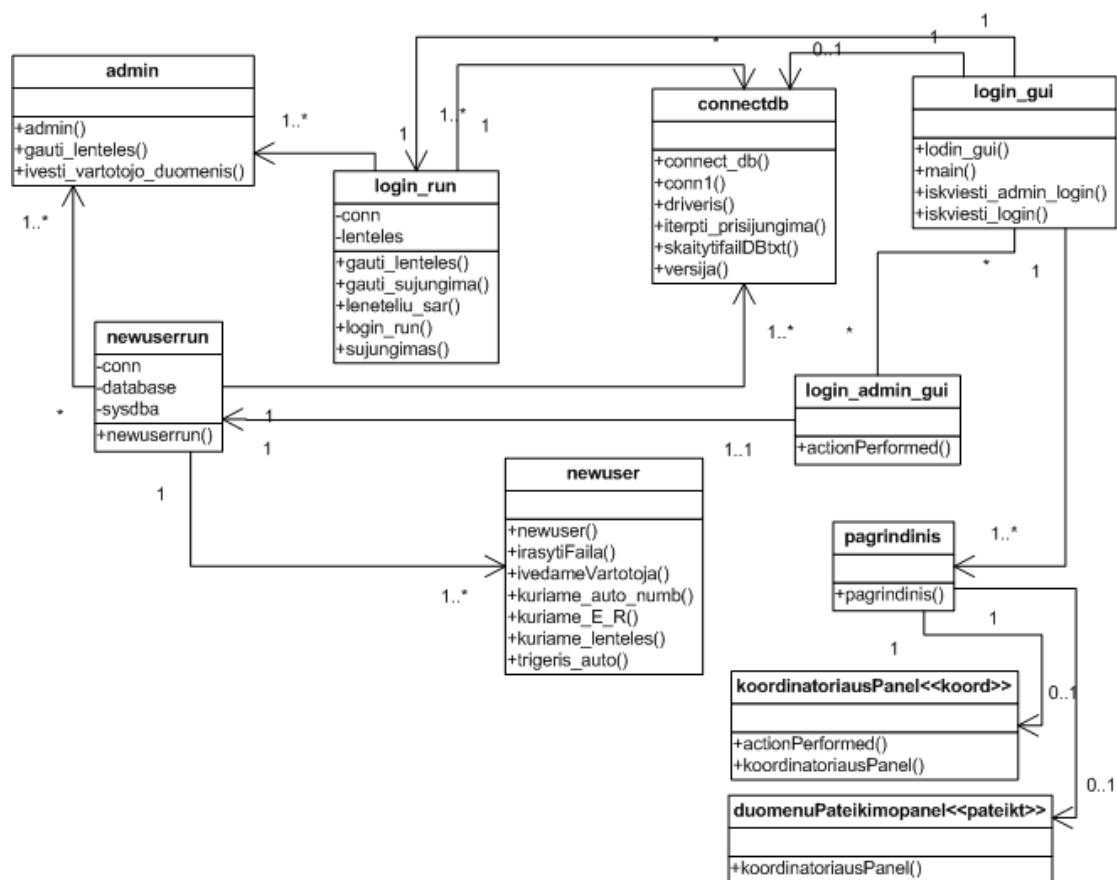
8 pav. Sistemos išdėstymo (deployment) vaizdas

4. Paketų aprašymai

4.1. BTP paketo aprašymas

17 lentelė. Paketo BTP aprašymas

Pavadinimas	BTP
Klasifikacija	paketas
Apibrėžimas	Paketas atitinka sistemos dalį, skirtą vartotojui norinčiam registruotis prie sistemos, taip pat vartotojui siekiančiam sukurti naujus vartotojus, kurie gali naudotis sistema jiems įvedant slaptažodžius ir duomenų bazės parametrus.
Atsakomybės	Komponentas realizuoja vartotojų įvedimą, lentelių E-R sukūrimą, jų E-R ryšių sudarymą, vartotojo identifikavimo tikrinimą. Sukuria grafinę vartotojo sąsają.
Apribojimai	Vykdomas prisijungimas prie DB.
Sąveikavimas	Komponentas prie DB jungiasi naudodamas TCP/IP protokolą. Taip pat naudojama ‚koord‘ paketo klasė ‚koordinatoriausPanel‘.
Resursai	JVM ne senesnė kaip 1.4 versijos
Skaičiavimai	-
Sąsaja/eksportas	Sąsaja pasiekama kaip grafinė vartotojo sąsaja (GUI), kurioje vartotojas įveda vartotojo identifikavimo vardą ir slaptažodį.

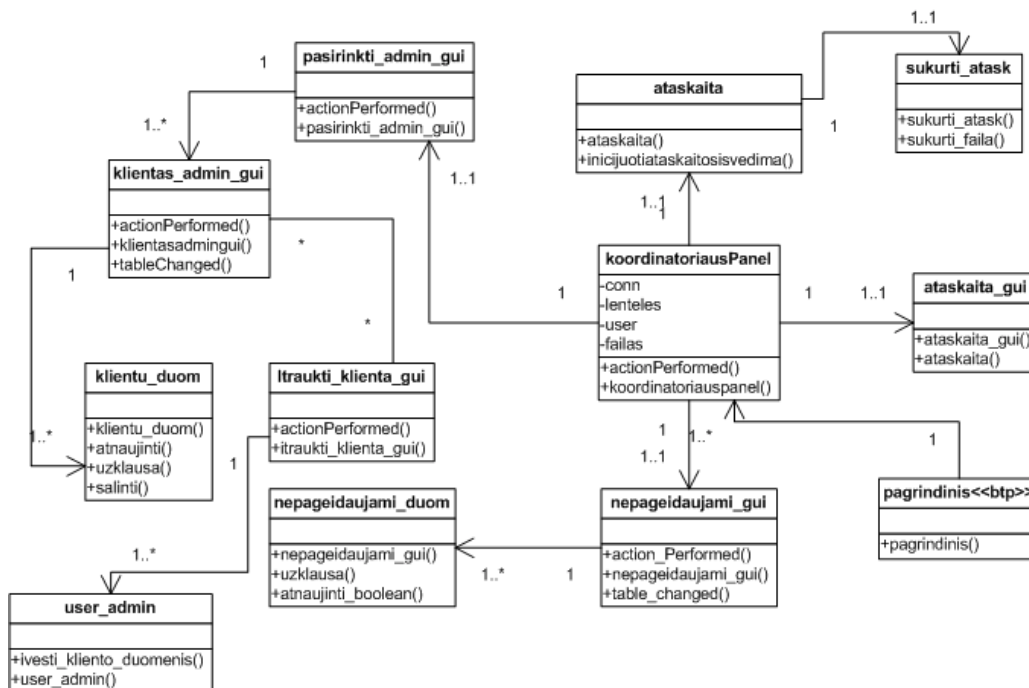


9 pav. BTP paketo klasių diagrama

4.2. KOORD paketo aprašymas

18 lentelė. Paketo “KOORD” aprašymas

Pavadinimas	KOORD
Klasifikacija	paketas
Apibrėžimas	Paketas atitinka sistemos dalį, skirtą vartotojui norinčiam administruoti vartotojus, kurie naudojami šia IS. Taip pat yra priemonės skirtos formuoti ataskaitas su galimybe jas išvesti. Vartotojas besinaudojantis šiuo paketu gali kontroliuoti nepageidaujamus šaltinius, nustatyti šaltinio apribojimo laiko limitus, išsiųsti pranešimus parenkant prioritetus, nustatant išsiuntimo laikus bei įvedant tekstą vartotojams, kurie pageidauja šiuos pranešimus gauti.
Atsakomybės	Komponentas realizuoja vartotojų įvedimą, redagavimą, šalinimą į DB. Formuoja nepageidaujamus šaltinius. Suformuoja ataskaitas pagal poreikius. Vykdo pranešimų išsiuntimą vartotojams.
Apribojimai	Vykdomas prisijungimas prie DB.
Struktūra	Komponentą sudaro klasės parašytos ir pavaizduotos 3 pav.
Sąveikavimas	Komponentas prie DB jungiasi naudodamas TCP/IP protokolą. Taip pat pateikia klases kitiems paketams t.y.: „BTP“;
Resursai	JVM ne senesnė kaip 1.4 versijos
Skaičiavimai	-
Sąsaja/eksportas	Sąsaja pasiekama kaip grafinė vartotojo sąsaja (GUI), kurioje vartotojas gali pagal atitinkamus kriterijus administruoti vartotojus, kurti pranešimus, išvesti ataskaitas. Duomenys vartotojams persiunčiami XML pavidalu.

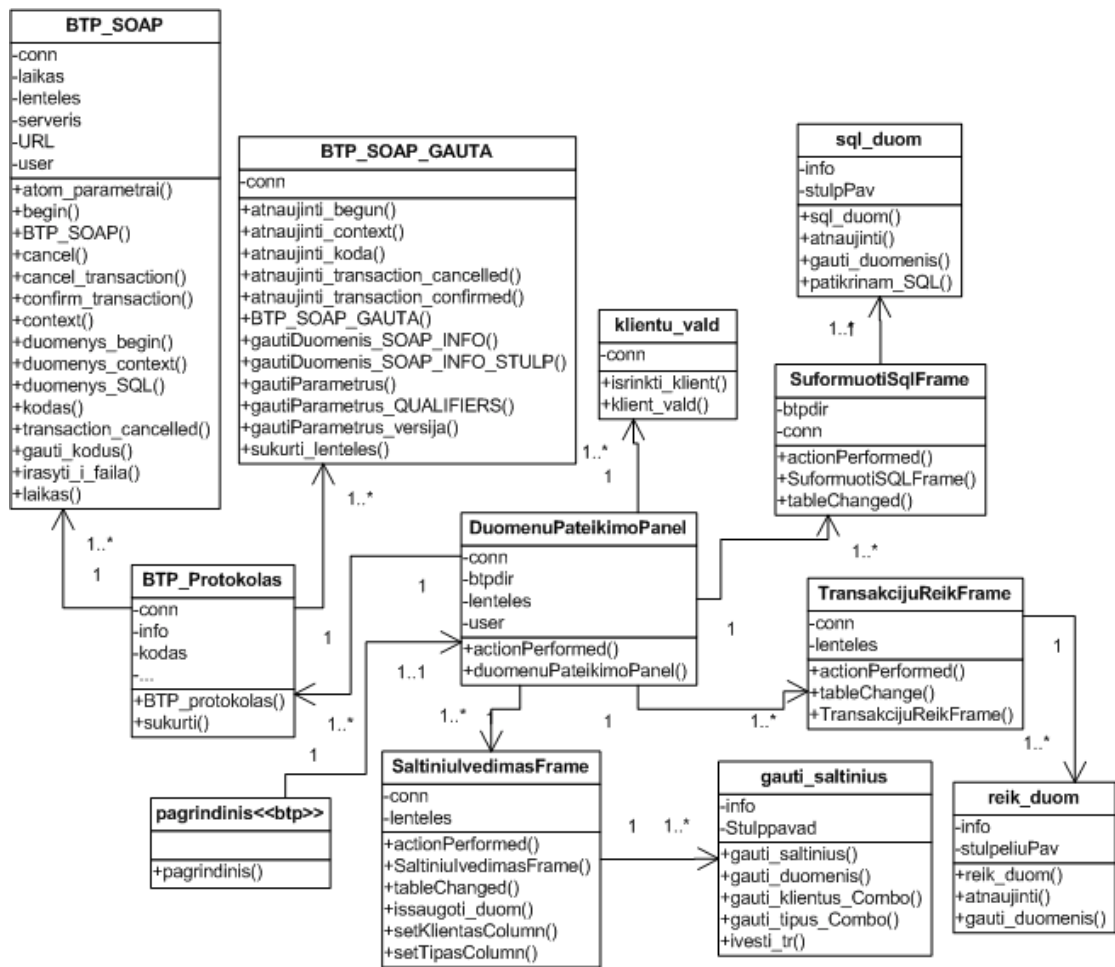


10 pav. Paketo “KOORD” klasių diagrama

4.3. PATEIK paketo aprašymas

19 lentelė. Paketo “PATEIK” aprašymas

Pavadinimas	pateik
Klasifikacija	paketas
Apibrėžimas	Paketas atitinka sistemos dalį, skirtą vartotojui norinčiam sukurti transakcijas pagal SQL užklausos sakinius tuo pačiu perimant duomenis iš vartotojų kurie pateikė duomenis vartotojui norinčiam gauti tuos duomenis. Taip pat yra priemonės skirtos administruoti transakcijas, bei tų transakcijų dalyvius.
Atsakomybės	Komponentas realizuoja, transakcijų tipų įvedimą ir jų informacijos išsaugojimą DB. Atlieka XML dokumentų sukūrimą. Vykdo XML dokumento patikrinimą gautą BTP protokolo pavidalu. Sukuriama grafinė vartotojo sąsaja.
Apribojimai	Vykdomas prisijungimas prie DB.
Sąveikavimas	Komponentas prie DB jungiasi naudodamas TCP/IP protokolą. Duomenų perdavimas/priėmimas vykdomas WEB serviso SOAP servisu, BTP protokolo pagrindu.
Resursai	JVM ne senesnė kaip 1.4 versijos
Skaičiavimai	-
Sąsaja/eksportas	Sąsaja pasiekama kaip grafinė vartotojo sąsaja (GUI), kurioje vartotojas gali pagal atitinkamus kriterijus įvesti URL šaltinį, vartotojus, parinkti transakcijos tipus, suvesti SQL sakinius. Duomenys vartotojui pateikiami XML pavidalu.

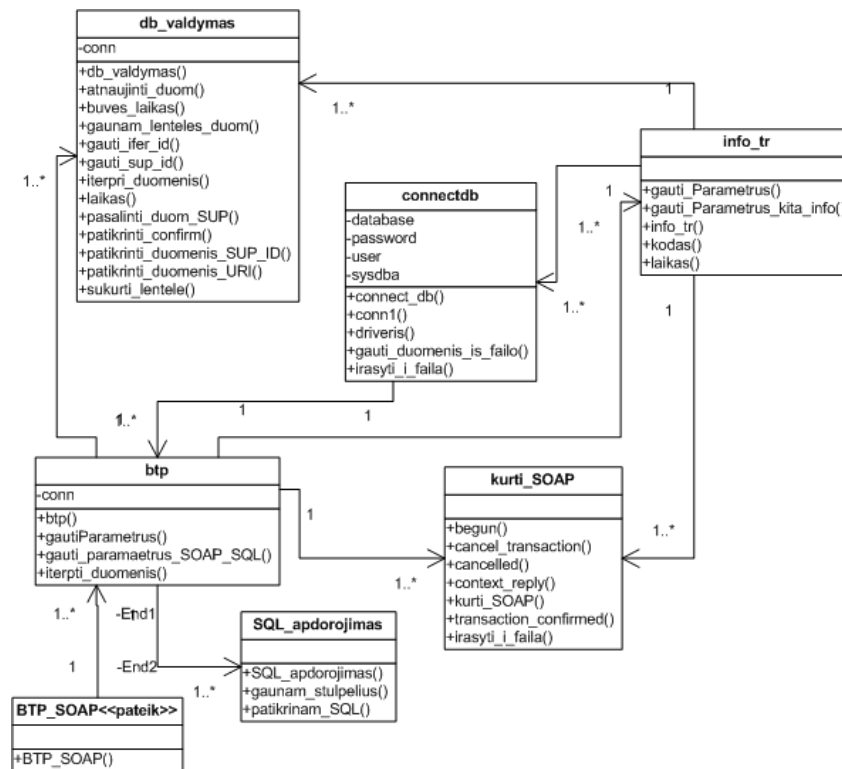


11 pav. Paketo "PATEIK" klasių diagrama

4.4. ATSAKYTI paketo aprašymas

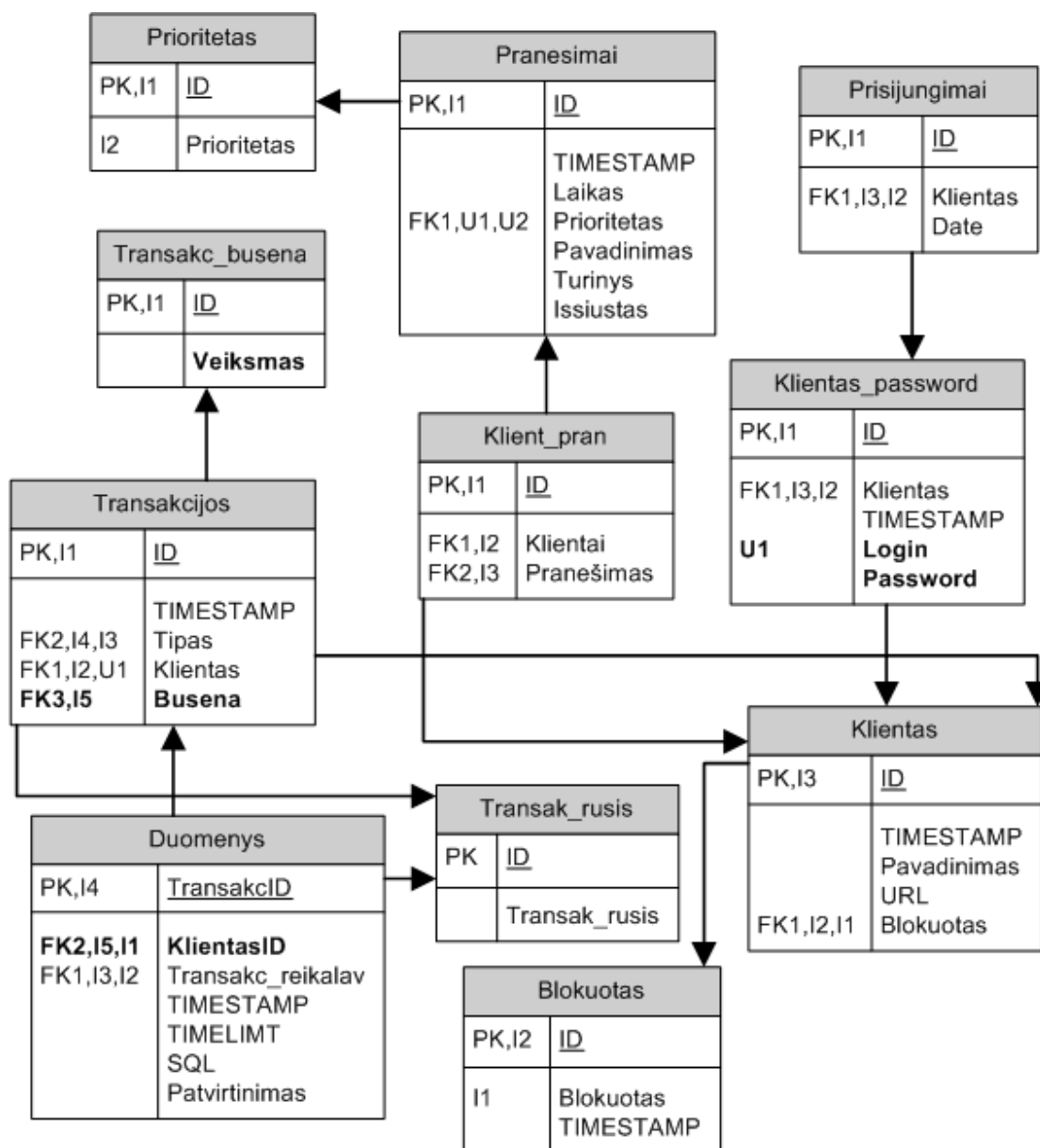
20 lentelė. Paketo “ATSAKYTI” aprašymas

Pavadinimas	atsakyti
Klasifikacija	paketas
Apibrėžimas	Paketas atitinka sistemos dalį, skirtą vartotojui norinčiam pateikti transakcijas pagal vartotojo SQL užklauso sakinius. Naudojantis šiuo paketu pateikiami duomenys vartotojams kurie pageidavo šių duomenų. Taip pat yra priemonės skirtos patalpinti duomenis apie sėkmingą/nesėkmingą transakcijų patalpiniimą.
Atsakomybės	Komponentas realizuoja vartotojų įvedimą, transakcijų informacijos eigos vykdymą bei išsaugojimą DB. Atlieka XML dokumentų sukūrimą. Vykdo XML dokumento persiuntimą BTP protokolo pavidalu.
Apribojimai	Vykdomas prisijungimas prie DB.
Sąveikavimas	Komponentas prie DB jungiasi naudodamas TCP/IP protokolą. Duomenų perdavimas/priėmimas vykdomas WEB serviso SOAP servisu, BTP protokolo pagrindu.
Resursai	JVM ne senesnė kaip 1.4 versijos
Skaičiavimai	-
Sąsaja/eksportas	Duomenų gražinimas SOAP protokolo pavidalu. Pagal gautus atitinkamus kriterijus išsiunčiami atitinkami XML dokumentai realizuojant BTP protokolo transakcijų tipus.



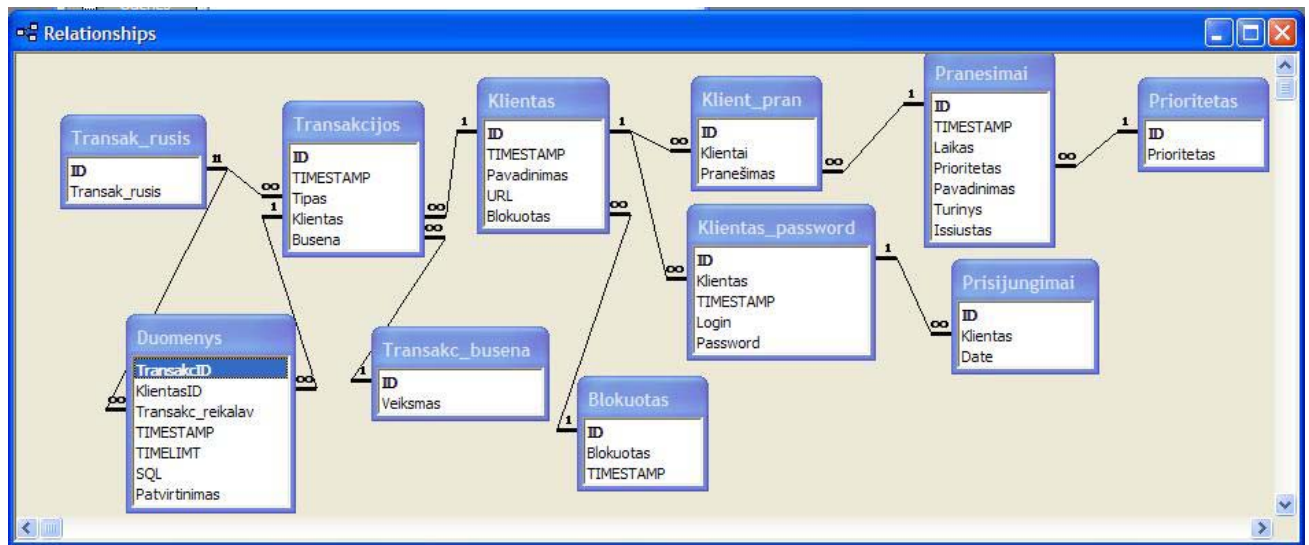
12 pav. Paketo “ATSAKYTI” klasių diagrama

4.5. Duomenų bazės schema



13 Pav. Duomenų bazės schema pateikta MSVisio pavidalu

PK- pirminio rakto stereotipas
 FK n -paveldinčio rakto stereotipas
 In-Indekso pavadinimas
 Un-Unikalaus rakto stereotipas



14 Pav. Duomenų bazės schema pateikta MSAccess pavidalu

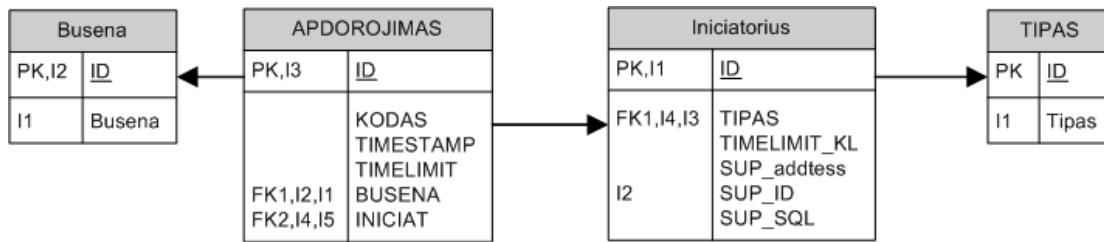
Detalesnis duomenų schemas aprašymas realizuotas ORACLE duomenų bazėje.

21 lentelė. Duomenų bazės lentelių aprašymas

Atributo pavadinimas	Ilgis/Tipas	Trumpas aprašas
Blokuotas		
ID	NUMBER	Įrašo identifikatorius
Blokuotas	NUMBER(1)	Fiksuojamas ar dalyvis blokuotas (Oracle duomenų bazėje neegzistuoja atributas BOOLEAN)
TIMESTAMP	DATE	Dalyvio blokavimo laikas
Klientas		
ID	NUMBER	Įrašo identifikatorius
TIMESTAMP	DATE	Fiksuojama iniciatoriaus įvedimo data
Pavadinimas	VARCHAR2(50)	Iniciatoriaus vardas (įmonės pavadinimas)
Blokuotas	NUMBER(1)	FK iš lentelės 'Blokuotas'
Klientas password		
ID	NUMBER	Įrašo identifikatorius
Klientas	NUMBER	FK iš lentelės 'Klientas'
TIMESTAMP	DATE	Fiksuojama iniciatoriaus slaptažodžio įvedimo/pakeitimo data
Login	VARCHAR2(10)	Vartotojo prisijungimo vardas
Password	VARCHAR2(10)	Vartotojo prisijungimo slaptažodis
Prisijungimai		
ID	NUMBER	Įrašo identifikatorius
Klientas	NUMBER	FK iš lentelės 'Klientas_password'
Date	DATE	Kliento prisijungimo data
Klient pran		
ID	NUMBER	Įrašo identifikatorius

Klientai	NUMBER	FK iš lentelės 'Klientas'
Pranešimas	NUMBER	FK iš lentelės 'Pranešimai'
Pranešimai		
ID	NUMBER	Įrašo identifikatorius
TIMESTAMP	DATE	Pranešimo įvedimo laikas
Laikas	DATE	Pranešimo išsiuntimo laikas
Prioritetas	NUMBER	FK iš lentelės 'Prioritetas'
Pavadinimas	VARCHAR2(50)	Pranešimo antraštės pavadinimas
Turinys	VARCHAR2(2000)	Laukelis skirtas turiniui saugoti
Issiustas	NUMBER(1)	Laukelis BOOLEAN tipo, skirtas pranešimui pažymėti kaip išsiustam.
Prioritetas		
ID	NUMBER	Įrašo identifikatorius
Prioritetas	VARCHAR2(10)	Skirtas prioriteto rūšims saugoti
Transakcijos		
ID	NUMBER	Įrašo identifikatorius
TIMESTAMP	DATE	Transakcijos įvedimo laikas
Tipas	NUMBER	FK iš lentelės 'Transakcijos rūšis'
Klientas	NUMBER	Įrašo identifikatorius lentelei 'Duomenys'
Busena	NUMBER	FK iš lentelės 'Transakcijos_busena'
Transakcijos_busena		
ID	NUMBER	Įrašo identifikatorius
Veiksmas	VARCHAR(10)	Laukelis skirtas transakcijos būsenai nurodyti ir aprašyti lietuvių kalba
Transakcijos_rūšis		
ID	NUMBER	Įrašo identifikatorius
Transak_rūšis	VARCHAR(8)	Laukelis skirtas transakcijos rūšiai nurodyti ir aprašyti lietuvių kalba
Duomenys		
TransakcID	VARCHAR(13)	Transakcijos identifikatorius identifikuojantis dalyviui, pagal šį numerį dalyvis inicijuoja duomenų priėmimą
KlientasID	NUMBER	FK iš lentelės 'Transakcijos'
Transakc_reikalav	NUMBER	FK iš lentelės 'Transakcijos_rūšis'
TIMESTAMP	DATE	Fiksuoja transakcijos suformavimo laiką
TIMELIMIT	DATE	Transakcijos laiko trukmė naudojama BTP protokole
SQL	VARCHAR2(2000)	Laukelis skirtas SQL sakiniui saugoti
Patvirtinimas	NUMBER(1)	Laukelis fiksuojanti apie sėkmingą visų transakcijų priėmimą

Serverinės dalies (dalyvių) duomenų bazė



15 Pav. Serverinės dalies duomenų bazės schema pateikta MSVisio pavidalu

Duomenų aprašymas ORACLE duomenų bazėje

22 lentelė. Serverinės dalies duomenų bazės lentelių aprašymas

Atributo pavadinimas	Ilgis/Tipas	Trumpas aprašas
APDOROJIMAS		
ID	NUMBER	Įrašo identifikatorius
KODAS	VARCHAR2(13)	Egzistuojantis dalyvio kodas nusiųstas klientui
TIMESTAMP	DATE	Duomenų prašymo laikas
TIMELIMIT	DATE	Įvestas atsakymo laiko limitas
BUSENA	NUMBER	FK iš lentelės 'Busena'
INICIAT	NUMBER	FK iš lentelės 'Iniciatorius'
Klientas		
ID	NUMBER	Įrašo identifikatorius
TIPAS	NUMBR	FK iš lentelės 'TIPAS'
TIMELIMIT_KL	DATE	Iniciatoriaus atsakymo į transakciją laiko ribos
SUP_address	VARCHAR2(2000)	Iniciatoriaus adresas kuriuo kreiptasi į dalyvi
SUP_ID	VARCHAR(13)	Iniciatoriaus identifikacinis kodas
SUP_SQL	VARCHAR(2000)	SQL užklausa, su kuria buvo prašyta duomenų.
BUSENA		
ID	NUMBER	Įrašo identifikatorius
Busena	VARCHAR(10)	Laukelis skirtas transakcijos būsenai nurodyti ir aprašyti lietuvių kalba
TIPAS		
ID	NUMBER	Įrašo identifikatorius
Tipas	VARCHAR(8)	Laukelis skirtas transakcijos tipui nurodyti ir aprašyti lietuvių kalba

Detalesnė architektūra, funkciniai/nefunkciniai reikalavimai, sistemos prototipo MS ACCESS variantų pavyzdžiai pateikti magistrinio tiriamojo darbo ataskaitoje (T000M102) 107 psl. 2008 m.

5. Eksperimentinis tyrimas

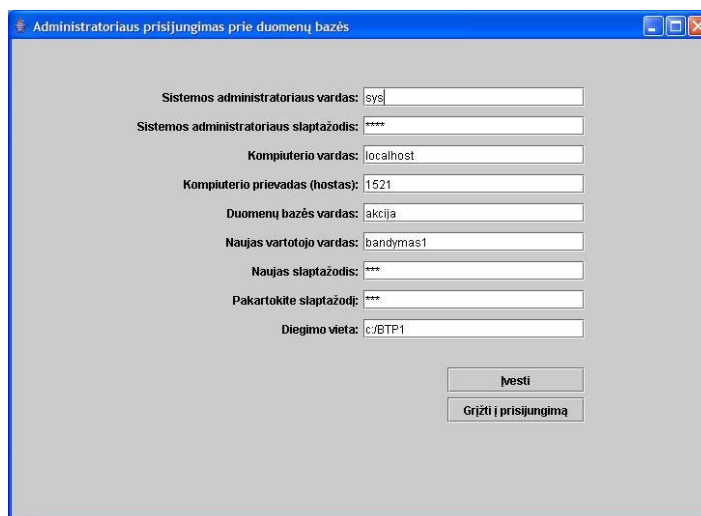
5.1. *Sistemos eksperimentinis tyrimas ir testavimas*

Šios sistemos nebuvo galimybės realizuoti tarp autonominių serverių, todėl pasirinktas sistemos testavimas ir eksperimentinis tyrimas buvo atliekamas vienoje darbo stotyje, realizuojant vietinį adresą 'localhost' arba '127.0.0.1'. Prieš jungiantis į sistemą (16 pav.) vartotojas privalo įvesti vartotojo identifikavimo duomenis arba realizuoti kitą panaudojimo atvejį kurti naują vartotoją. Siekiant informuoti vartotoją apie sistemos elgseną vienu arba kitu atveju yra sukurta būsenos juosta.



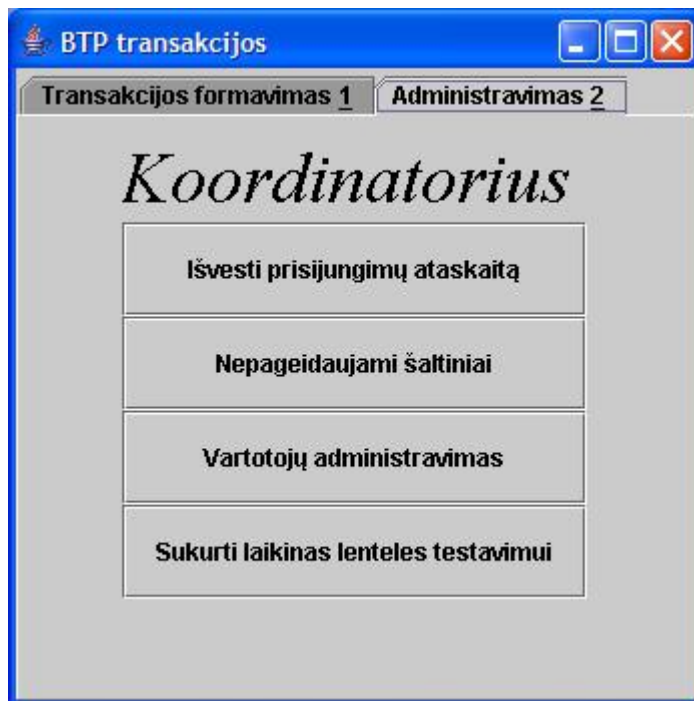
16 pav. Prisijungimas prie sistemos

Siekiant kuo paprasčiau vartotojui įdiegti sistemą ar sukurti naują vartotoją, programos naudotojui tereikia inicijuoti mygtuką ir yra sukuriamas GUI (17 pav.), kurioje vartotojui reikia įvesti tik pagrindinius parametrus.



17 pav. Naujo vartotojo sukūrimas

Vartotojui klaidingai įvedus sistemos duomenis yra pranešama apie egzistuojančius netikslumus, kitu atveju yra prisijungiama prie IS ir pateikiama sistemos administravimo kortelė, kurioje galima įvykdyti keletą panaudojimo atvejų nurodytus reikalavimų specifikacijoje.



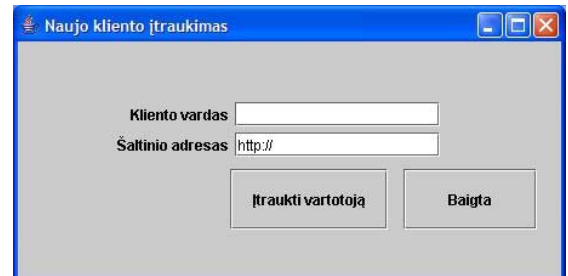
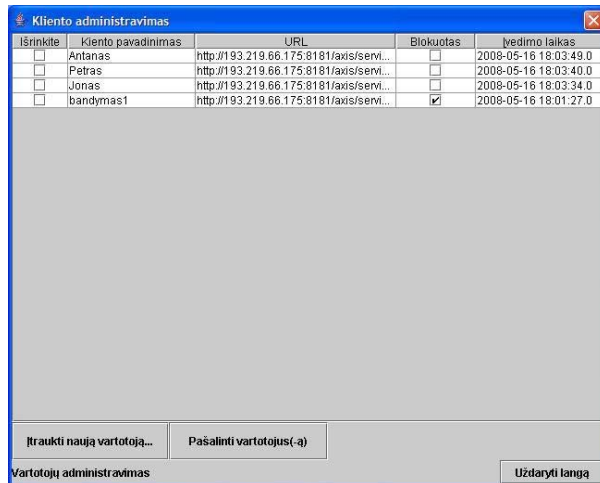
18 pav. Vartotojo administravimo GUI

Siekiant kad vartotojas galėtų matyti kaip buvo naudojama sistema, ar sistema nepasinaudojo pašaliniai asmenys - inicijuojamas prisijungimų ataskaitos išvedimas. Ši ataskaita yra formuojama iš HTML dokumento, tuo pačiu siekiant, kad administratorius (valdančiam šias transakcijas) galėtų matyti ataskaitas iš nutolusios darbo vietos, pasinaudojant interneto galimybėmis.



19 pav. Vartotojo ataskaitos išvedimas iš HTML dokumento

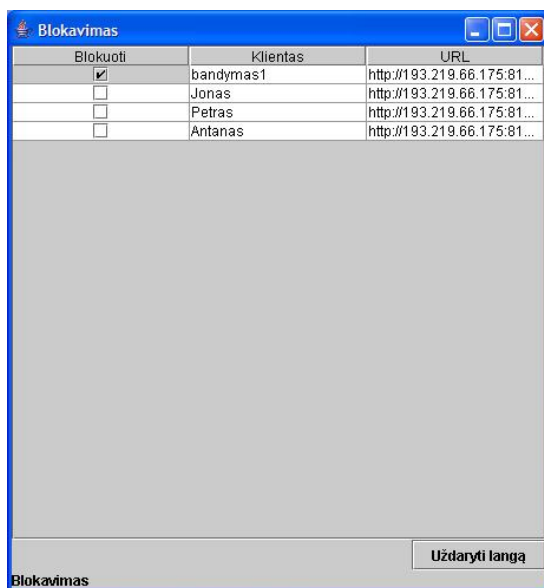
Sistemos administratorius administruojantis šią sistema gali įvesti klientus (dalyvius) iš kurių reikia duomenų. Įvedant sistemos dalyvius yra pateikiamas naujas GUI, kuriame administratorius įveda pagrindinius parametrus ir dalyvius. Jis taip pat paprasčiausiu būdu kelis/vieną vartotojus(-ą) gali pašalinti iš sistemos.



21 pav. Naujo kliento įtraukimo GUI

20 pav. Klientų administravimo GUI

Klientui atsisakius keletą vartotojų ar kurių sutarties terminas yra pasibaigęs, administratorius gali blokuoti sistemos dalyvius, juos nepašalinus iš sistemos tikintis šiuos duomenis (klientus) galės panaudoti vėliau. Šiam realizavimui yra sukurtas GUI 21 pav., kuriame administratorius gali blokuoti dalyvius.



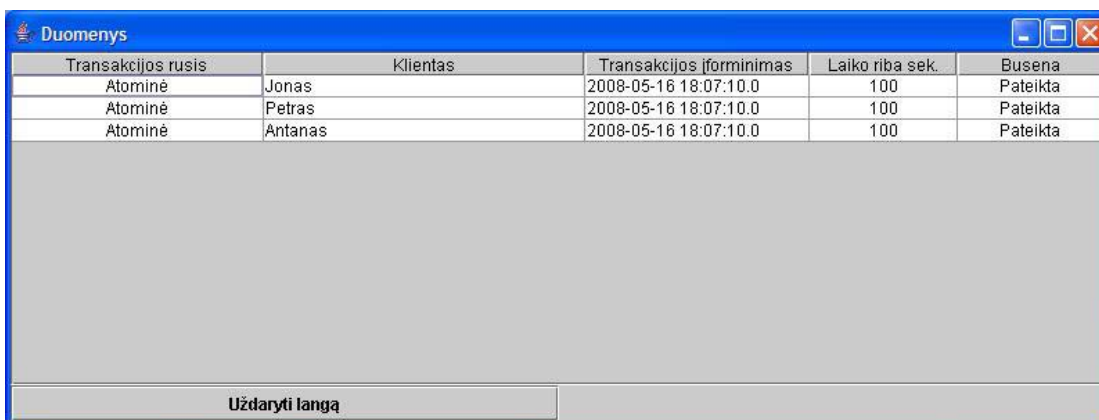
21 pav. klientų blokavimo GUI

Kitoje kortelėje ‘Transakcijos formavimas’ yra realizuojami panaudojimo atvejai, kurie skirti iniciatoriui valdyti transakcijas 22 pav. Iniciatorius prieš formuojant transakcijas turi suteikti klientams vykdomos transakcijos tipus. Remiantis verslo logika tas pats dalyvis vienu metu negali turėti skirtingų užklausų į tą patį servisą, todėl šis funkcinis reikalavimas yra apdorotas, kaip transakcijos tipo suformavimo klaida.



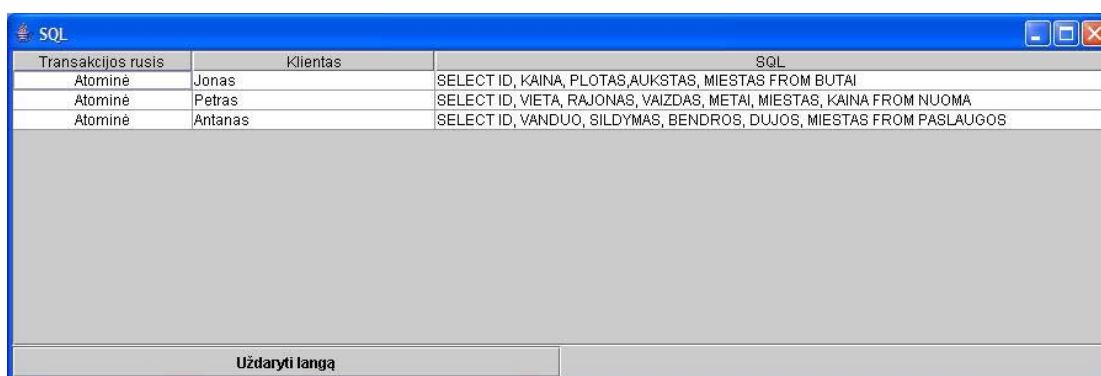
22 pav. Šaltinių priskyrimas verslo dalyviams (klientams)

Siekiant įgyvendinti vieną iš BTP protokolo transakcijų reikalavimų – iniciatorius suvedęs transakcijų reikalavimus įveda laiko apribojimus, laiko limitą per kurį turi būti įvykdyta transakcija 23 pav. Tai leidžia verslo dalyviams iš anksto planuoti užsakymus. Siekiant išplėsti sistemos funkcines galimybes, suteikta galimybė transakcijos iniciatoriui kritiniu atveju pakeisti laiko limitus transakcijų vykdymo metu. Ši galimybė numatyta ir dalyvių serverinėse klasėse. Šiame GUI, iniciatorius visada turi galimybę peržiūrėti informaciją apie tame laikotarpyje pateiktos transakcijos būseną, taip pat matyti likusį laiko limitą.



23 pav. Panaudojimo atvejo “Įvesti reikalavimus transakcijai” realizacija

Išplečiamumas - taip pat svarbus kriterijus siekiant sistemą plėtoti ir tobulinti. Šiam tikslui buvo panaudotos standartinės duomenų bazių SQL užklausos. Reikėtų paminėti, kad realizuojant standartines SQL užklausas susiduriama su problema: jeigu įrašai dubliuojasi, buvo atnaujinti arba buvo pašalinti. Kad galėtume sėkmingai atnaujinti arba pašalinti įrašus – tam buvo sukurtas specialus laukelis ‘ID’, kuris yra pažymėtas kaip unikalus įrašo identifikatorius. Vartotojo SQL įvedimo sąsajoje 24 pav. šis reikalavimas nurodytas. Neesant atributui ID užklausos įvedimas į IS atmetamas, taip pat siekiant sumažinti vartotojo klaidų tikimybę, atliekama SQL užklausos įvedimo sintaksės tikrinimas.



24 pav. SQL sintaksės įvedimo GUI.

Suformavus BTP transakcijos reikalavimus toliau yra vykdomas transakcijos siuntimas šios sistemos dalyviams. Kadangi yra tikimybė kad tarp paskirstytų sistemų gali įvykti neprognuojami netikslumai – ši problema yra numatyta. SOAP protokolas turi galimybę apdoroti netikslius duomenis, ir pranešti sistemos dalyviams apie egzistuojančią SOAP protokolo klaidą, todėl šioje sistemoje ši galimybė yra apdorota.

Testuojant sistemą buvo pasirinkti trys sistemos dalyviai (Jonas, Petras, Antanas) ir transakcijų iniciatoriai. Pasirinkti trys tariami duomenų modeliai atvaizduoti 23, 24 ir 25 lentelėse, kurios surašytos dalyvių ORACLE duomenų bazėje:

23 lentelė. Verslo dalyvio ‘Jonas’ lentelėje ‘BUTAI’ egzistuojantys duomenų vienetai

ID	KAINA	PLOTAS	AUKSTAS	MIESTAS
1	100000	54	3	Vilnius
2	800000	55	1	Kaunas
3	900000	80	6	Klaipėda
4	500000	200	11	Šiauliai
5	400000	345	2	Panevėžys

24 lentelė. Verslo dalyvio 'Petras' lentelėje 'NUOMA' egzistuojantys duomenų vienetai

ID	VIETA	RAJONAS	VAIZDAS	METAİ	MIESTAS	KAINA
1	CENTRAS	ANTAKALNIS	PUIKUS	1998	Vilnius	2500
2	UZMIESTIS	ANTAKALNIS	PRASTAS	2000	Vilnius	1500
3	SODAI	ZIRMUNAI	NORMALUS	2002	Vilnius	1300
4	SODAI	JUSTINISKES	NORMALUS	1990	Vilnius	1800
5	PAUPYS	PILIAKALNIS	PRASTAS	1975	Kaunas	900
6	PAUPYS	TABORAS	PUIKUS	1934	Vilnius	900
7	GYV.RAJONAS	VALAKAMPIAI	PUIKUS	1968	Vilnius	1600
8	CENTRAS	PILAITE	NORMALUS	1999	Vilnius	2100

25 lentelė. Verslo dalyvio 'Antanas' lentelėje 'PASLAUGOS' egzistuojantys duomenų vienetai

ID	VANDUO	SILDYMAS	BENDROS	DUJOS	MIESTAS
1	50	154	34	34	Vilnius
2	40	184	50	23	Kaunas
3	35	304	78	25	Šiauliai
4	78	145	35	12	Panevėžys
5	43	250	60	28	Elektrėnai
6	87	197	97	15	Palanga
7	32	187	84	43	Marijampolė
8	68	230	96	41	Alytus
9	28	240	59	21	Anykščiai
10	23	197	35	22	Šilutė

Iniciatorius siekdamas gauti šiuos duomenis inicijuoja užklausas pateiktas 26 lentelėje suteikdamas dalyviams atominės transakcijos. Pagal suformuotas transakcijas duomenys sėkmingai pasiekė iniciatorių ir duomenis patalpino į iniciatoriaus duomenų bazę.

26 lentelė. Iniciatoriaus inicijuotos užklaustos IS dalyviams

Vartotojas	Transakcijos tipas	SQL užklaustos sintaksės tekstas
Jonas	Atominė	SELECT ID, KAINA, PLOTAS,AUKSTAS, MIESTAS FROM BUTAI
Petras	Atominė	SELECT ID, VIETA, RAJONAS, VAIZDAS, METAİ, MIESTAS, KAINA FROM NUOMA
Antanas	Atominė	SELECT ID, VANDUO, SILDYMAS, BENDROS, DUJOS, MIESTAS FROM PASLAUGOS

Siekiant patikrinti perduotų duomenų taisyklingumą atliekant tyrimą, buvo pasinaudota trečiųjų šalių programine įranga, kuri patvirtino perduodamų taisyklingumą. Sistema papildomai buvo testuojama įvairiais transakcijų tipais, taip pat inicijuojant interneto ryšio sutrikimą, laiko limito pasibaigimą, blokuojant veikiančias gijas dalyvių ir iniciatoriaus sistemose. Visais šiais atvejais sistemos testavimas praėjo sėkmingai. Tačiau teigti, kad sistema veiks sklandžiai ir be surikimų negalima. Sistema nebuvo testuojama realiai nutolusiose sistemose, bei kompiuteriuose su skirtingais techniniais parametrais arba kita OS: Linux, Unix, Macintosh ir t.t. Realizuoto BTP protokolo, trečiųjų šalių programinės įrangos rezultatai ir pavyzdžiai pateikti šio darbo priede 25,26,27,31,32 pav.

5.2. Sistemos kriterijų vertinimas

Sistema pasižymi šiais kokybės kriterijais ir atitinka šiuos kokybės reikalavimus:

- ✓ Funkcionalumo – atliekamos pagrindinės šios sistemos funkcijos išvardintos reikalavimų specifikacijoje;
- ✓ Adaptyvumo – sistema duomenų keitimuisi tarp autonominių serverių naudoja visuotinai paplitusį SOAP ir BTP protokolą, keičiasi XML dokumentais, tokiu būdu sistemą galima suderinti su skirtingomis OS ir, sistemas palaikančias kitas programavimo kalbas, su galimybe realizuoti UTF-8, UTF-16, US-ASCII kalbų koduotes.
- ✓ Tolerancija klaidoms – sistemoje realizuotas platus klaidų apdorojimo mechanizmas, leidžia sumažinti vartotojo klaidos tikimybę, o tuo pačiu ir sistemos pažeidžiamumą;
- ✓ Plečiamumo – sistemoje palikta daug vietos plečiamumui. Galimybė pritaikyti kitas DB sistemas, serverio administravimo sąsajos realizavimas, platesnė duomenų kontrolė ir vartotojų administravimas.

5.3. Sistemos taikymo rekomendacijos

Sukurta sistema naudojama daugiašaliam bendradarbiavimui. Verslo transakcijose dalyvauja vienas transakcijų iniciatorius ir šios transakcijos dalyviai (atskirai kiekvienas dalyvis ar įmonė vykdo nepriklausomus verslo procesus). Dalyvių serveriuose gali registruotis neribotas kiekis dalyvių ir kiekvienas jų gali dalyvauti tarpusavio duomenų apsikeitimų procese.

Norint padidinti praktinį sistemos panaudojimą galima išplėsti sistemą, naudojantis tokiomis rekomendacijomis:

- ✓ Neribotas dalyvių kiekis.

Galimybė prijungti neribotą kiekį dalyvių yra realizuota esamoje sistemoje. Tereikia įdiegti serverį SOAP protokolo palaikymui ir nustatyti parametrus prisijungimui prie duomenų bazės.

- ✓ Skirtingos iniciatorių ir dalyvių duomenų struktūros

Sistemoje tiek transakcijų iniciatoriaus, tiek dalyvių duomenų struktūros gali sutapti, t.y. turi bendrą lentelių, kurios naudojamos verslo informacijai saugoti bei valdyti verslo transakcijas. Paprastai dalyvis turi savo duomenų bazę, kurios dalies, naudojamos verslo informacijai saugoti, struktūra skiriasi nuo klientų serveryje naudojamos struktūros. Siekiant netrukdyti verslo dalyviams keistis informacija rekomenduojama sukurti atskirą duomenų bazę transakcijų realizavimui BTP protokolu, sukuriant atitinkamas prieigos teises, suteikiant transakcijų dalyviams priejimą tik prie tam skirtų duomenų, kurie realizuojami pasirašant sutartį. Tai ypač svarbu, nes atviras duomenų laikymas gali turėti įtakos konfidencialiai įmonės informacijai.

- ✓ Skirtingo tipo paslaugas teikiančios įmonės

Prie sistemos gali prijungti dalyviai teikiantys įvairaus tipo paslaugas. Norint prijungti naujo tipo teikiamas paslaugas geriausia išeitis - kurti naujus vartotojus, tokiu būdu sistemos administratoriui bus paprasčiau šias transakcijas suskirstyti.

- ✓ Įvairaus tipo vartotojai

Informacine sistema rekomenduojama naudotis vartotojams, kurie naudojami pakankamai dažnai keliomis el. paslaugomis.

- ✓ Supaprastintas programuotojų darbas

Naudojantis šia sistema, yra sumažintas programinių modulių duomenų tikrinimas, sumažėja kaštai skirti programiniam įrangos palaikymui. O taip pat nekartojamas pakankamai brangus rankinis kodų rinkimas kuriant el. sistemas.

Išvados

1. Išanalizavus egzistuojančias sistemas ir atlikus literatūros apžvalgą pastebėti egzistuojančių sistemų išplečiamumo trūkumai, todėl iškelti nauji funkciniai ir nefunkciniai reikalavimai liudija apie naujos sistemos reikalingumą;
2. Šiame darbe sukurta sistema pagal pagrindinius iškeltus uždavinius, kurioje palikta daug erdvės sistemos išplečiamumui;
3. Šioje sistemoje realizuotas plačiai tarp Web servisų paslaugų paplitęs SOAP protokolas ir XML dokumentas leidžia teigti apie galimas sistemos panaudojimo galimybes įvairiose platformose;
4. Atliktas duomenų apsikeitimas tarp daugelių autonominių šaltinių, kurių pagalba galima atlikti duomenų mainus vienus metu tarp neriboto verslo įmonių skaičiaus, vienu metu realizuojant B2B ir B2C sąvokas;
5. Realizuotas OASIS konsorciumo sukurtas BTP protokolas leidžia panaudoti jau egzistuojančias taisykles atliekant verslo sandorius ar duomenų mainus internetu;
6. Sistema realizuota vienoje plačiausių tarp stambių verslo dalyvių naudojamų ORACLE duomenų bazėje, realizuojant SQL standartines užklausas leidžia teigti apie sistemos panaudojimo galimybes kitose duomenų bazėse: MySQL, Navision, MS SQL SERVER 2000 ir t.t.;
7. Ši sistema užkoduota JAVA programavimo kalba liudija apie sintaksės taisyklingumą ir šiuolaikinės objektinės kalbos realizavimą;
8. Sistemos prisijungimo ataskaitos išvedimai formuojami HTML dokumentu, tokiu būdu suteikiant galimybę ataskaitas peržiūrėti nuotoliniu būdu.
9. Sistema testuota realizuojant Apache Tomcat 4.2 serverį su Apache AXIS papildomu paketu, realizuojant imitacinį duomenų mainų modelį tarp skirtingų verslo dalyvių;
10. Parašyta apie 8000 kodų eilučių, sukurtos 36 klasės (iš jų 6 serverinės) siekiant programinės įrangos vartotojui sukurti sistemą kuo lankstesnę ir paprastesnę;
11. Atlikta 150 klaidų apdorojimo priemonių (tarp kurių 35 serverinės) tokiu būdu siekiant sukurti sistemą kuo mažiau pažeidžiamą;

12. Sistemos pritaikomumas galimas įvairiose verslo srityse kuriose vykdomi duomenų mainai tarp verslo dalyvių;
13. Sukurtas įdiegimo panaudojimo atvejis – suteikia galimybę naudotis sistema keliems vartotojams iš karto be papildomų programavimo žinių.
14. Atliktas SOAP protokolo tyrimas ir analizė keliomis taikomosiomis programomis, liudija apie SOAP protokolu perduodamų duomenų taisyklumą.
15. Sukurtas programinis modelis žymiai sumažina kainą, kurios reikalauja verslo transakcijų įgyvendinimas, priežiūra ir palaikymas.

Literatūra

1. Mandar Chitnis, Pravin Tiwari, & Lakshmi Ananthamurthy, UML Tools. Prieiga per internetą: <http://www.developer.com/design/article.php/1593811>
2. OASIS Technical Committee Advances Business Transaction Protocol (BTP) Specification CoverPages, 2004, December. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=business-transaction
3. William Cox, Felipe Cabrera, George Copeland, Tom Freund, Johannes Klein, Tony Storey, Satish Thatte, 2004, January. Prieiga per internetą: <http://dev2dev.bea.com/pub/a/2004/01/ws-transaction.html>
4. Catalog of OMG Specification. Prieiga per internetą: http://www.omg.org/technology/documents/spec_catalog.htm
5. JSR-000095 J2EETM Activity Service for Extended Transactions. Prieiga per internetą: <http://jcp.org/aboutJava/communityprocess/final/jsr095/index.html>
6. JSR 156: Java API for XML Transactions. Prieiga per internetą: <http://jcp.org/en/jsr/detail?id=156>
7. WSDL Tutorial. <http://www.w3schools.com/wSDL/default.asp>
8. Yannis Charalabidis, Vassilios Karakoidas, Stephanos Androutsellis-Theotokis, Diomidis Spinellis, Enabling B2B Transactions over the Internet through Application Interconnection: The PRAXIS Project, 2004. Prieiga per internetą: <http://www.spinellis.gr/pubs/conf/2004-eChallenge-Praxis/html/CKAS04.htm>
9. Oracle Integration B2B and Oracle BPEL Process Manager Interoperability, 2005, September. Prieiga per internetą: http://www.oracle.com/technology/products/integration/b2b/pdf/B2B_TU_001_B2B_BPEL.pdf
10. Kate Rhodes, XML-RPC vs. SOAP. Prieiga per internetą: http://weblog.masukomi.org/writings/xml-rpc_vs_soap.htm
11. A busy developers guide to SOAP 1.1. Prieiga per internetą: <http://www.software.org/bdg>
12. Gabsaga Tata, Simpaq Inc, Better Software Solutions. Prieiga per internetą: <http://www.simpaq.com/pages/4/index.htm>
13. Steven Holzner, InformIT: Sams Teach Yourself XML in 21 Days. Prieiga per internetą: http://www.informit.com/library/content.aspx?b=STY_XML_21days

14. Jeff Mesnil, JOTM-BTP: a BTP extension for JOTM, 2003, May. Prieiga per internetą: <http://jotm.objectweb.org/current/jotm-btp/doc/btp/btp.html>
15. Transaction service for the JXTA platform that uses BTP, Prieiga per internetą: <https://btp-jxta.dev.java.net/>
16. Elliott Rusty Harold, Processing XML with Java. Prieiga per internetą: <http://www.cafeconleche.org/books/xmljava/chapters/index.html>
17. WebServices – Axis. Prieiga per internetą: <http://ws.apache.org/axis/>
18. Tutorial: Axis: The next generation of Apache SOAP. Prieiga per internetą: <http://www.roseindia.net/software-tutorials/detail/3406>
19. Tarak Modi, Clean up your wire protocol with SOAP, 2001, January Prieiga per internetą: <http://www.javaworld.com/javaworld/jw-04-2001/jw-0427-soap.html?page=1>
20. The Apache + SSL on Win32 HOWTO Prieiga per internetą: <http://tud.at/programm/apache-ssl-win32-howto.php3>
21. Apache Tomcat. Prieiga per internetą: <http://tomcat.apache.org/>
22. SOAP specification Prieiga per internetą: <http://www.w3.org/TR/SOAP/>
23. Choreology. Prieiga per internetą: <http://www.choreology.com/usecases/index.htm>

Terminų žodynas

- **DB** – Oracle 10g reliacinė duomenų bazė (Oracle 10g Database);
- **PHP** – programavimo kalba aprašanti hipertekstinį procesorių (Hypertext Processor);
- **SOAP** – WEB serviso protokolas (Service Object Access Protocol);
- **LAN** – vietinis kompiuterinis tinklas (Local Area Network);
- **WAN** – Pasaulinis kompiuterių tinklas (Wide Area Network);
- **XML** – duomenų aprašymo standartas (eXtensible Markup Language);
- **HTML** – hipertekstinis dokumento aprašas (HyperText Markup Language);
- **BTP** – verslo transakcijos protokolas (Business Transaction Protocol);
- **JVM** – Javos virtuali mašina (Java Virtual Machine);
- **Verslo klientas** – asmuo besinaudojantis programos klientine dalimi;
- **Koordinatorius** – servisas, valdantis transakcijas;
- **Duomenų bazė** – kartu saugomų ir susijusių duomenų visuma, kuriai būdinga integruotumas, nepertekliškumas, nepriklausomumas

PRIEDAI

The screenshot shows the TCPMonitor application window. At the top, there's a title bar with the application name and standard window controls. Below that, there's a control panel with a 'Stop' button, 'Listen Port' set to 8181, 'Host' set to 127.0.0.1, 'Port' set to 8080, and a 'Proxy' checkbox. A table below displays a list of recent transactions with columns for status, time, request host, target host, and request details. The selected row shows a POST request to /axis/services/btp3 HTTP/1.0 from 192.168.1.100 to 127.0.0.1 at 2008-05-16 02:52:03. Below the table are 'Remove Selected' and 'Remove All' buttons. The main area is split into two panes showing the raw HTTP and SOAP messages. The top pane shows the request headers and XML body, and the bottom pane shows the response headers and XML body. At the bottom of the window, there are checkboxes for 'XML Format' (checked) and 'Numeric', along with 'Save', 'Resend', 'Switch Layout', and 'Close' buttons.

S...	Time	Request Host	Target Host	Request...
---	Most Recent	---	---	---
Done	2008-05-16 02:52:03	192.168.1.100	127.0.0.1	POST /axis/services/btp3 HTTP/1.0 ...
Done	2008-05-16 02:52:03	192.168.1.100	127.0.0.1	POST /axis/services/btp1 HTTP/1.0 ...
Done	2008-05-16 02:52:03	192.168.1.100	127.0.0.1	POST /axis/services/btp1 HTTP/1.0 ...
Done	2008-05-16 02:52:03	192.168.1.100	127.0.0.1	POST /axis/services/btp1 HTTP/1.0 ...
Done	2008-05-16 02:52:05	192.168.1.100	127.0.0.1	POST /axis/services/btp1 HTTP/1.0 ...
Done	2008-05-16 02:52:05	192.168.1.100	127.0.0.1	POST /axis/services/btp1 HTTP/1.0 ...
Done	2008-05-16 02:52:06	192.168.1.100	127.0.0.1	POST /axis/services/btp2 HTTP/1.0 ...
Done	2008-05-16 02:52:06	192.168.1.100	127.0.0.1	POST /axis/services/btp2 HTTP/1.0 ...
Done	2008-05-16 02:52:06	192.168.1.100	127.0.0.1	POST /axis/services/btp2 HTTP/1.0 ...

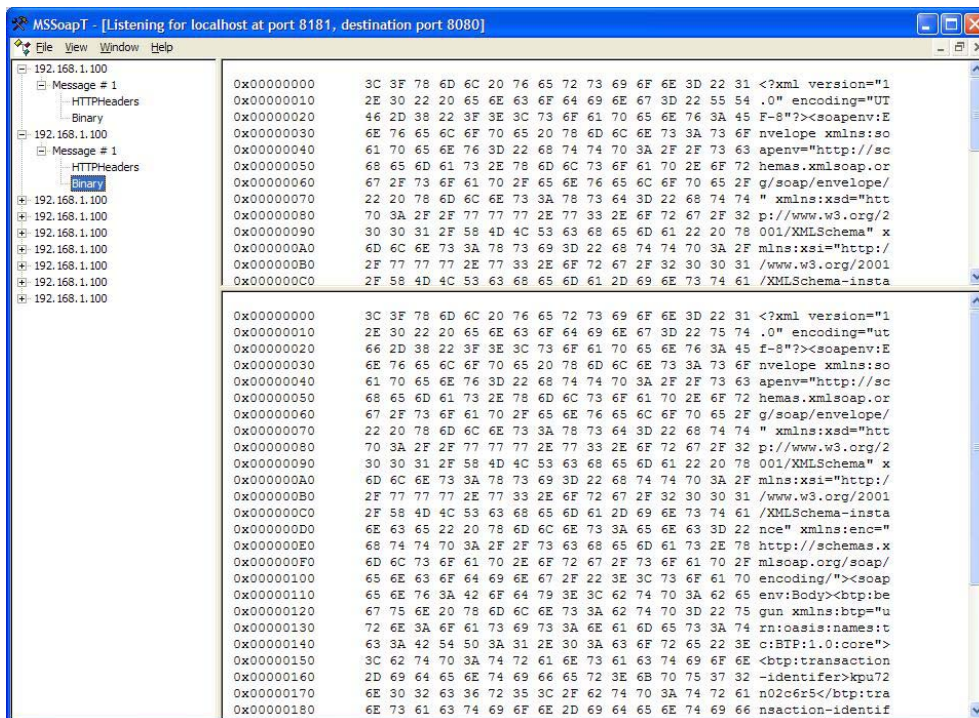
```
POST /axis/services/btp3 HTTP/1.0
Content-Type: text/xml; charset=utf-8
Accept: application/soap+xml, application/dime, multipart/related, text/*
User-Agent: Axis/1.4
Host: 127.0.0.1:8181
Cache-Control: no-cache
Pragma: no-cache
SOAPAction: ""
Content-Length: 523

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soapenv:Body>
    <btp:begin xmlns:btp="urn:oasis:names:tc:BTP:1.0:core">
      <btp.transaction-type>atom</btp.transaction-type>
      <btp:qualifiers>
        <btp.transaction-timelimit>3</btp.transaction-timelimit>
      </btp:qualifiers>
    </btp:begin>
  </soapenv:Body>
</soapenv:Envelope>

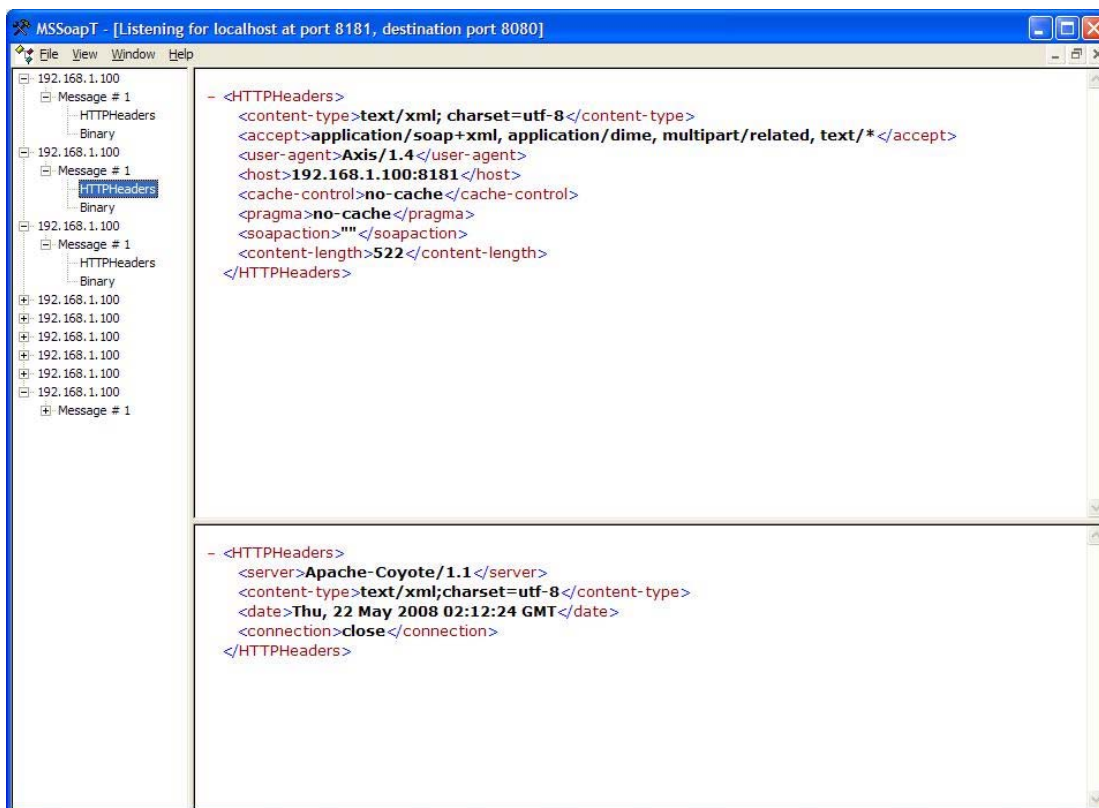
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/xml; charset=utf-8
Date: Thu, 15 May 2008 23:52:03 GMT
Connection: close

<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soapenv:Body>
    <btp:begin xmlns:btp="urn:oasis:names:tc:BTP:1.0:core">
      <btp.transaction-identifer>my4m4bs9az9d</btp.transaction-identifer>
      <btp:qualifiers>
        <btp.transaction-timelimit>3</btp.transaction-timelimit>
      </btp:qualifiers>
    </btp:begin>
  </soapenv:Body>
</soapenv:Envelope>
```

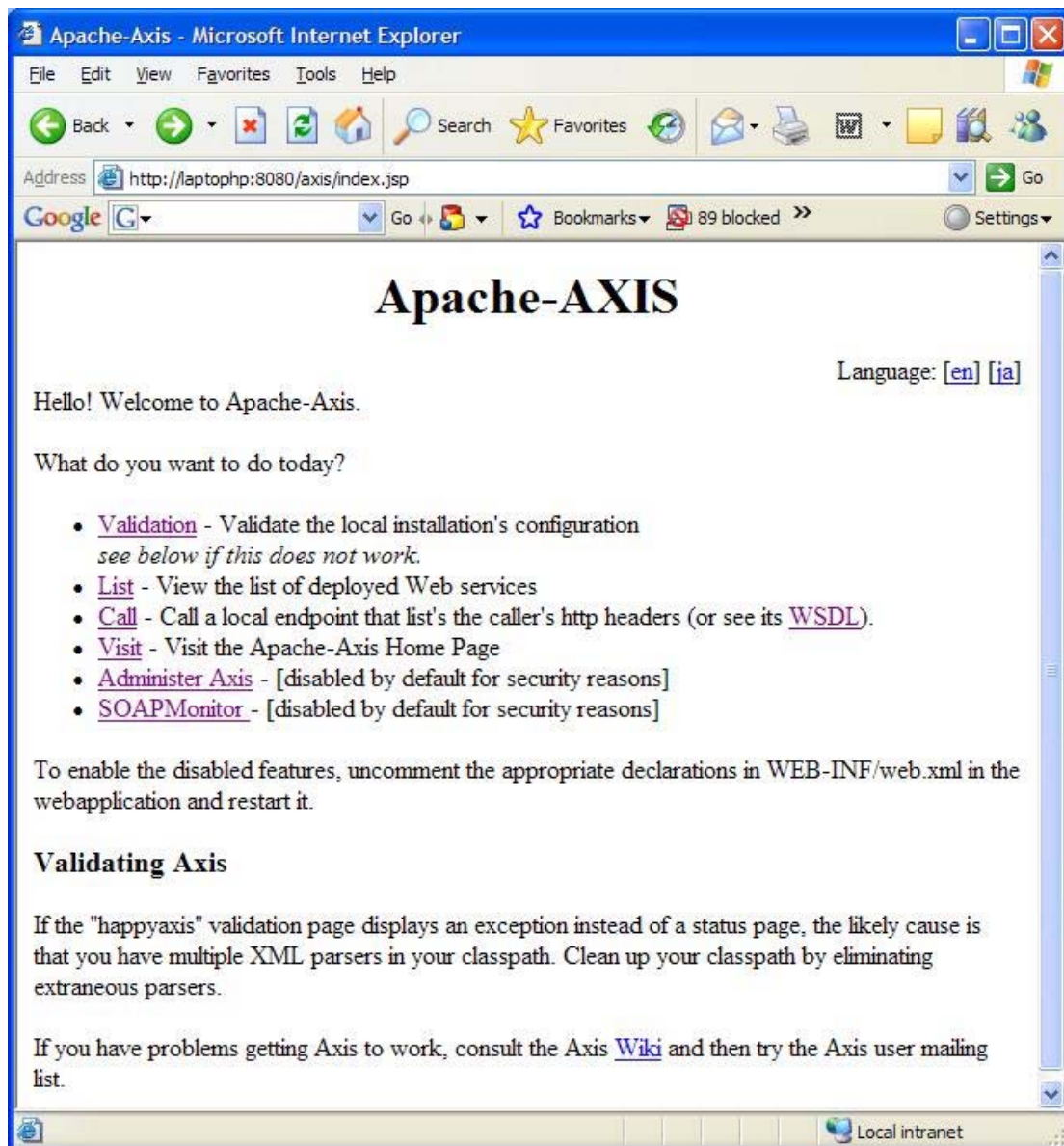
25 pav. SOAP protokolo tyrimas inicijuojant transakciją



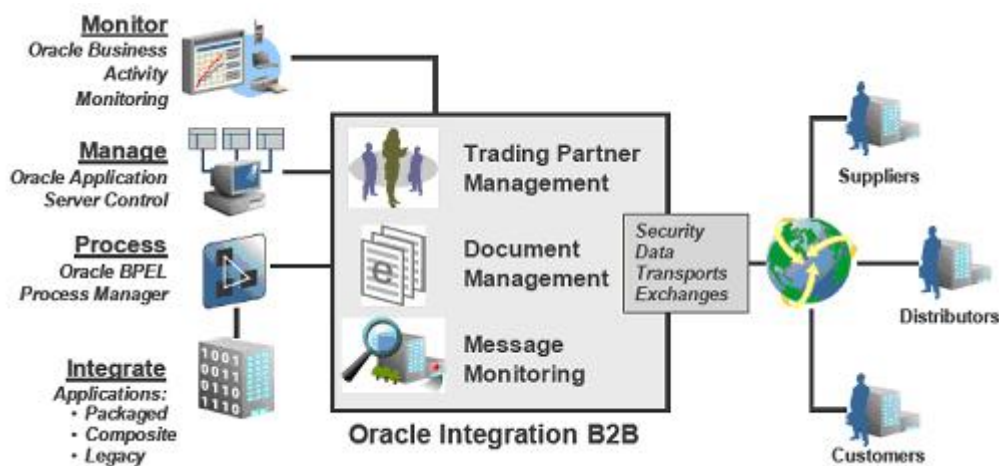
26 pav. SOAP protokolo tyrimas WINDOWS priemonėmis inicijuojant transakciją



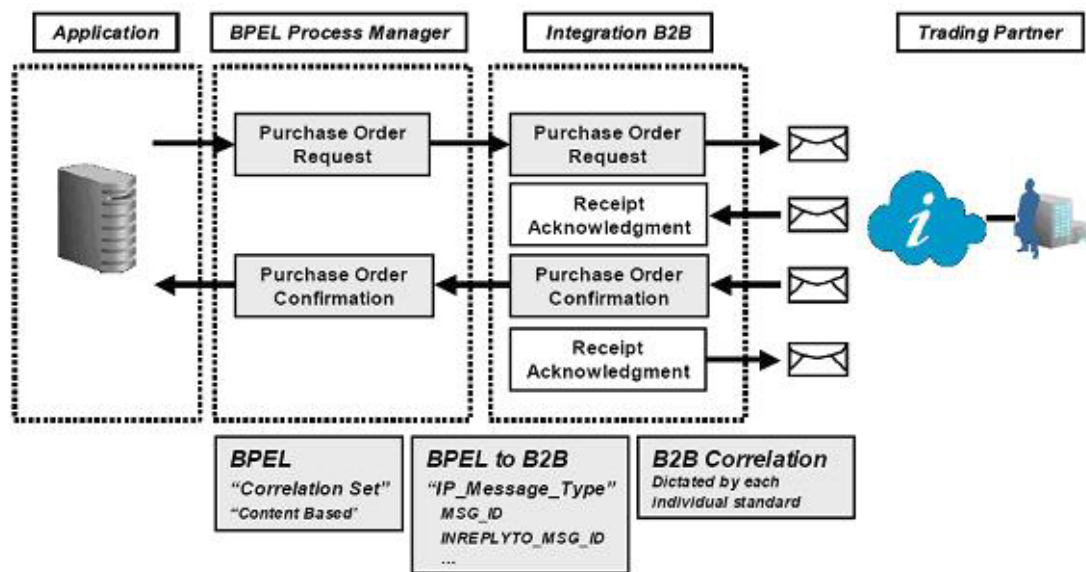
27 pav. SOAP protokolo tyrimas antraštinės dalies WINDOWS priemonėmis inicijuojant transakciją



28 pav. AXIS papildomo paketo realizacija TOMCAT serveryje.



29 pav. Oracle siūlomi sprendimai realizuojant B2B



30 pav. Oracle siūlomi sprendimai realizuojant BPEL verslo transakcijų protokolą

POST /axis/services/btp3 HTTP/1.0

Content-Type: text/xml; charset=utf-8

Accept: application/soap+xml, application/dime, multipart/related, text/*

User-Agent: Axis/1.4

Host: 127.0.0.1:8181

Cache-Control: no-cache

Pragma: no-cache

SOAPAction: ""

Content-Length: 522

<?xml version="1.0" encoding="UTF-8"?>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:enc="http://schemas.xmlsoap.org/soap/encoding/">

<soapenv:Body>

<btp:begin xmlns:btp="urn:oasis:names:tc:BTP:1.0:core">

<btp:transaction-type>atom</btp:transaction-type>

<btp:qualifiers>

<btp:transaction-timelimit>100</btp:transaction-timelimit>

</btp:qualifiers>

</btp:begin>


```
</soapenv:Body>
</soapenv:Envelope>
31 pav. BTP protokolo INICIATORIAUS užklauso realizavimo pavyzdys
```

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Content-Type: text/xml;charset=utf-8

Date: Thu, 22 May 2008 02:28:32 GMT

Connection: close

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:enc="http://schemas.xmlsoap.org/soap/encoding/">
```

```
<soapenv:Header>
```

```
<btp:context-reply soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="0" xmlns:btp="urn:oasis:names:tc:BTP:1.0:core">
```

```
<btp:superior-identifier>kh7b7s41f0hs</btp:superior-identifier>
```

```
<btp:completion-status>completed</btp:completion-status>
```

```
<btp:qualifiers>
```

```
<btp:transaction-timelimit>2</btp:transaction-timelimit>
```

```
</btp:qualifiers>
```

```
</btp:context-reply>
```

```
</soapenv:Header>
```

```
<soapenv:Body>
```

```
<lenteles_duomenys xmlns="">
```

```
<BUTAI>
```

```
<eilute>
```

```
<ID>1</ID>
```

```
<KAINA>100000</KAINA>
```

```
<PLOTAS>54</PLOTAS>
```

```
<AUKSTAS>3</AUKSTAS>
```

```
<MIESTAS>Vilnius</MIESTAS>
```

```
</eilute>
```

```
<eilute>
  <ID>2</ID>
  <KAINA>800000</KAINA>
  <PLOTAS>55</PLOTAS>
  <AUKSTAS>1</AUKSTAS>
  <MIESTAS>Kaunas</MIESTAS>
</eilute>
<eilute>
  <ID>3</ID>
  <KAINA>900000</KAINA>
  <PLOTAS>80</PLOTAS>
  <AUKSTAS>6</AUKSTAS>
  <MIESTAS>Klaipeda</MIESTAS>
</eilute>
<eilute>
  <ID>4</ID>
  <KAINA>500000</KAINA>
  <PLOTAS>200</PLOTAS>
  <AUKSTAS>11</AUKSTAS>
  <MIESTAS>Siauliai</MIESTAS>
</eilute>
<eilute>
  <ID>5</ID>
  <KAINA>400000</KAINA>
  <PLOTAS>345</PLOTAS>
  <AUKSTAS>2</AUKSTAS>
  <MIESTAS>Panevezys</MIESTAS>
</eilute>
</BUTAI>
</lenteles_duomenys>
</soapenv:Body>
</soapenv:Envelope>
```

32 pav. SOAP ir BTP protokolu atsakomi duomenys iniciatoriui

```

<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions
  targetNamespace="http://laptophp:8080/axis/services/btp3"
  xmlns:impl="http://laptophp:8080/axis/services/btp3"
  xmlns:intf="http://laptophp:8080/axis/services/btp3"
  xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:wsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns1="http://atsakyti.btp"
  xmlns:wsoap="http://schemas.xmlsoap.org/wsdl/">

- <wsdl:types>
- <schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://atsakyti.btp">
  <element name="process" type="xsd:anyType" />
</schema>
- <schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://laptophp:8080/axis/services
/btp3">
  <element name="processReturn" type="xsd:anyType"
  />
</schema>
</wsdl:types>
- <wsdl:message name="processRequest">
  <wsdl:part name="part" element="tns1:process" />
</wsdl:message>
- <wsdl:message name="processResponse">
  <wsdl:part name="processReturn"
  element="impl:processReturn" />
</wsdl:message>
- <wsdl:portType name="btp_tomcat">
- <wsdl:operation name="process">
  <wsdl:input name="processRequest"
  message="impl:processRequest" />
  <wsdl:output name="processResponse"
  message="impl:processResponse" />
</wsdl:operation>
</wsdl:portType>
- <wsdl:binding name="btp3SoapBinding"
  type="impl:btp_tomcat">
  <wsoap:binding style="document"
  transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="process">
  <wsoap:operation soapAction="" />
- <wsdl:input name="processRequest">
  <wsoap:body use="literal" />
</wsdl:input>
- <wsdl:output name="processResponse">
  <wsoap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:service name="btp_tomcatService">
- <wsdl:port name="btp3" binding="impl:btp3SoapBinding">

```

```

    <wsdlsoap:address
      location="http://laptop:8080/axis/services/btp
      3" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

33 pav. WSDL sukurto dokumento struktūra

```

<deployment name="btp3" xmlns="http://xml.apache.org/axis/wsdd/"
  xmlns:java="http://xml.apache.org/axis/wsdd/providers/java"
  xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance">
  <!-- note that either style="message" OR provider="java:MSG" both work -->
  <service name="btp3" style="message">
    <parameter name="className" value="btp.atsakyti.btp_tomcat" />
    <parameter name="allowedMethods" value="process" />
  </service>
</deployment>

```

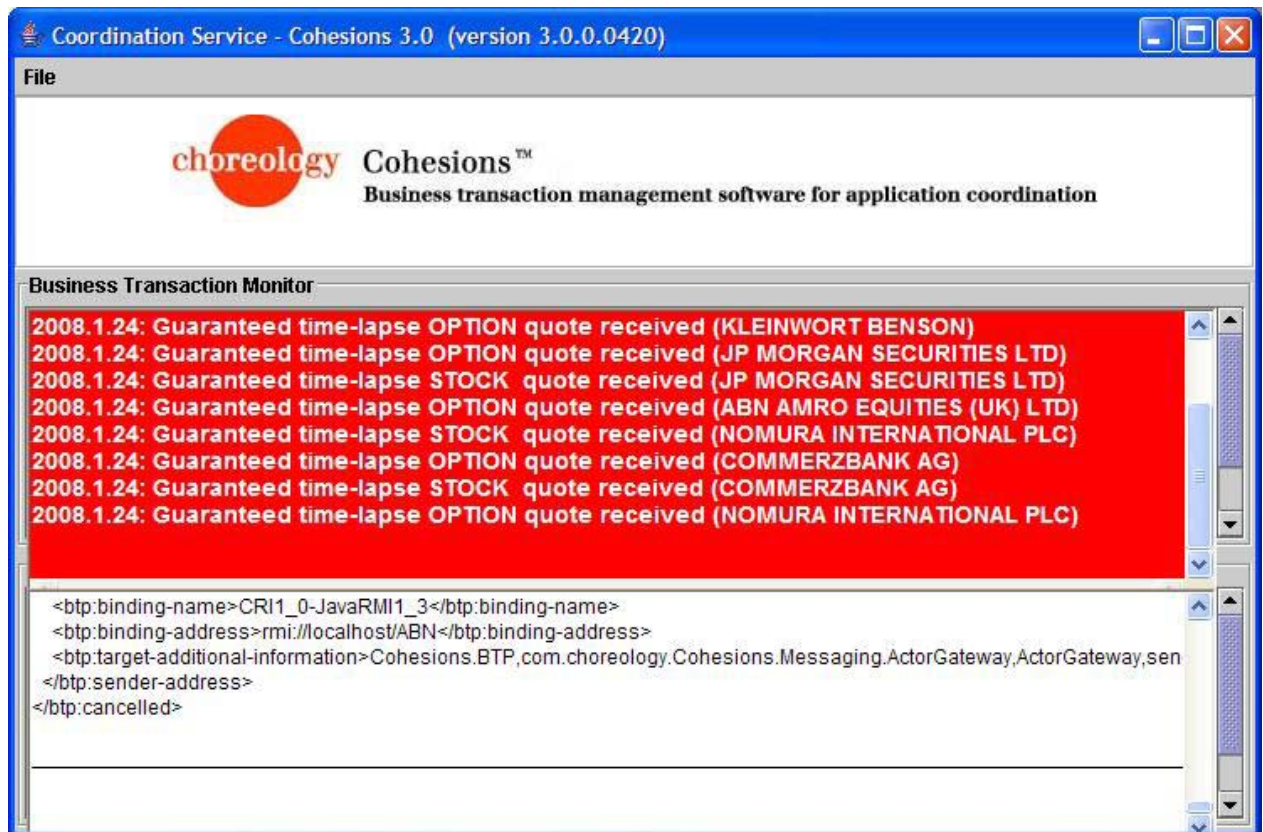
34 pav. WSDD failas, kuris reikalingas diegiant SOAP servisą kodo pavyzdys

```

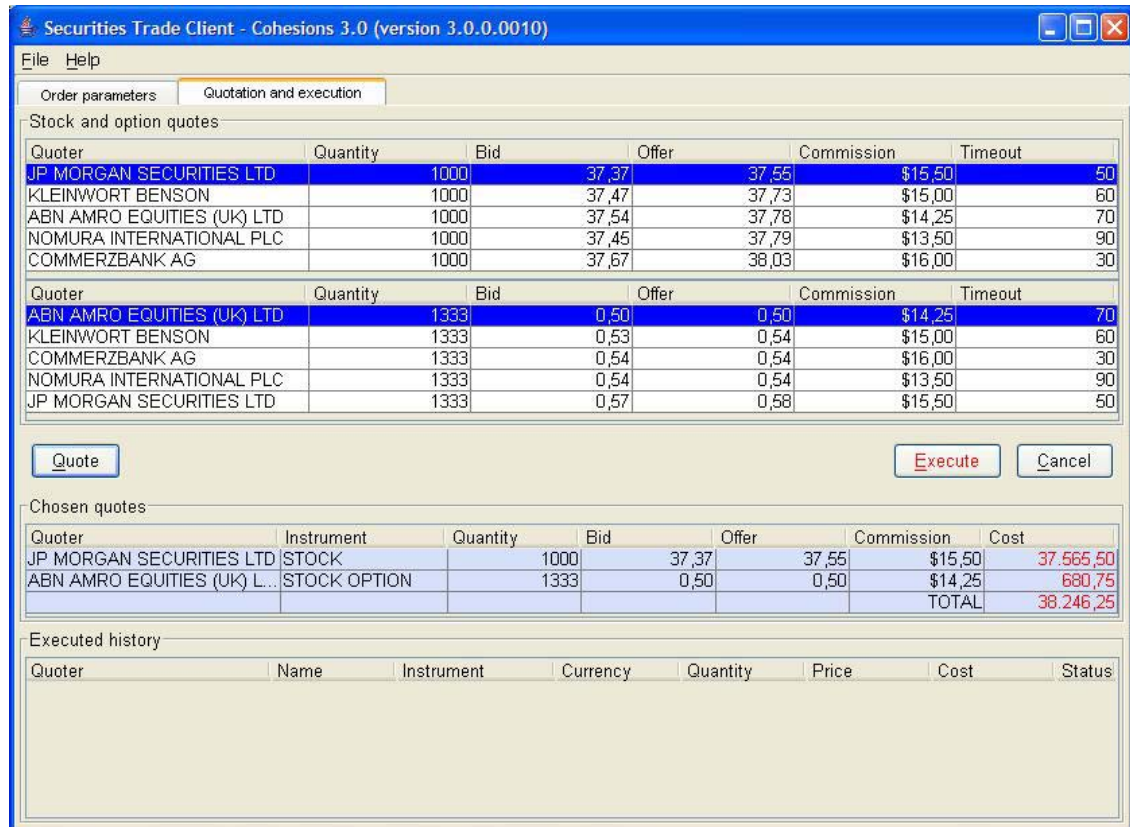
<undeployment name="btp3" xmlns="http://xml.apache.org/axis/wsdd/">
  <service name="btp3"/>
</undeployment>

```

35 pav. WSDD failas pašalinantis įdiegtą servisą



36 pav. Choreology kompanijos BTP protokolo realizacijos pavyzdys.



37 pav. Cohesions 3.0 verslo sistemos įgyvendinimo pavyzdys.