

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Andrius Čyvas

**Apsikeitimas paciento duomenimis, grįstas
paslaugų architektūra**

Magistro darbas

Darbo vadovas

doc. dr. R. Butleris

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Andrius Čyvas

**Apsikeitimas paciento duomenimis, grįstas
paslaugų architektūra**

Magistro darbas

Kalbos konsultantė

Lietuvių k. katedros
doc. dr. J. Mikelionienė

2006-05

Vadovas

doc. dr. R. Butleris

2006-05

Recenzentas

doc. dr. A. Ostreika

2006-05

Atliko

IFM-0/2 gr. stud.

Andrius Čyvas

2006-05

TURINYS

1	Įvadas	1
2	Apsikeitimo paciento duomenimis poreikių analizė.....	4
2.1	Informacijos srautų analizė	4
2.2	Keitimosi informacija problemos.....	6
2.2.1	Popierinis duomenų apskeitimo formatas	6
2.2.2	Informacijos fragmentacija organizacine prasme	8
2.2.3	Informacijos fragmentacija geografinė prasme.....	8
2.2.4	Skirtingi informacijos apie pacientą poreikiai	8
2.3	Panašių užsienio projektų analizė	9
2.4	Galimų keitimosi informacija modelių analizė	9
2.4.1	Centralizuotas modelis	10
2.4.2	Paskirstytasis modelis	11
2.4.3	Decentralizuotas modelis	12
2.5	Apibendrinimas	14
3	Paslaugų architektūros metodika	15
3.1	Metodikos principai.....	15
3.1.1	Laisvas sąryšis (angl. <i>Loose coupling</i>).....	15
3.1.2	Standartizuota sąsaja	16
3.1.3	Paslaugų autonomiškumas	17
3.1.4	Paslaugos lygmens abstrakcija.....	17
3.1.5	Pakartotinis panaudojamumas.....	18
3.1.6	Paslaugų suderinamumas	18
3.1.7	Būsenos atsisakymas	18
3.2	Palyginimas su komponentų architektūra	19
3.3	Žiniatinklio paslaugų technologija.....	21
4	Mišrus, paslaugomis grįstas duomenų apskeitimo modelis.....	24
5	Paslaugomis grįsto modelio realizacija.....	26
5.1	Tikslas	26
5.2	Sistemos ribos	26
5.3	Priemonės.....	27
5.3.1	Priemonių rinkimosi kriterijai	27
5.3.2	Sistemai kurti panaudotos technologijos ir jų pagrindimas	27

5.4	Posistemių funkcionalumas.....	29
5.4.1	ESIS portalas	29
5.4.2	Duomenų surinkimo brokeris.....	29
5.4.3	Gydytojų registras	31
5.4.4	Pacientų registras.....	32
5.4.5	Terminologijos registras.....	32
5.4.6	Vaistų registras	33
5.5	Posistemių loginis vaizdas	34
5.6	Posistemių duomenų vaizdas	35
5.6.1	Pacientų ligos istorijų posistemė.....	35
5.6.2	Diagnozių terminologijos posistemė.....	37
5.6.3	Vaistų registro posistemė	37
5.6.4	Gydytojų registro posistemė.....	37
5.6.5	Pacientų registro posistemė.....	38
5.7	Posistemių programinė įranga.....	38
5.8	Posistemių išdėstymo vaizdas	40
6	Paslaugų architektūra grįsto prototipo efektyvumo tyrimas	41
6.1	Prototipo kokybės tyrimas.....	41
6.2	Į žiniatinklio paslaugas ir serverio procedūras besikreipiančių interneto puslapių atsako laiko tyrimas	46
6.2.1	Tyrimo aprašymas	46
6.2.2	Tyrimui naudojamų serverių konfigūracijos:.....	46
6.2.3	Tyrimo rezultatai	47
6.3	Tyrimo rezultatų apibendrinimas	48
7	Išvados	49
8	Literatūra.....	50
9	Terminų ir santrumpų žodynas	52
10	Priedai.....	53
10.1	Priedas 1. Ligos istorijos įrašų duomenų bazės schema.....	53
10.2	Priedas 2. “Visual Studio 2005” paskirstytų sistemų diagramų notacija	54
10.3	Priedas 3. ISO/IEC 9126 kokybės modelio aprašymas.....	56
10.4	Priedas 4. Puslapio atsako laiko eksperimento rezultatai.....	59

Patient Data Sharing Based on Service Oriented Architecture

SUMMARY

Lithuanian healthcare is highly fragmented enterprise, with complex paper-based processes, that are inefficient, error prone, and often redundant. These shortcomings create formidable barriers to collaborative medical research. This paper is a research of Service Oriented Architecture solutions for solving interoperability problems when exchanging patient data between heterogeneous healthcare systems.

The Lithuanian healthcare information needs analysis was made to create the Service Oriented Architecture based model for exchanging patient data. The model prototype was developed to demonstrate the reusability and flexibility of service oriented solutions.

1 ĮVADAS

Remiantis Lietuvos elektroninės sveikatos strategijos projekte [2] pateiktais duomenimis, 80% gyventojų nėra patenkinti teikiamų sveikatos priežiūros paslaugų kokybe. Pagrindinės nurodytos priežastys: paciento sugaištas laikas gydymo įstaigoje, bei klaidingai nustatytos diagnozės, bei išrašyti vaistai. Viena iš silpnųjų grandžių teikiant sveikatos priežiūros paslaugas, yra gydytojo efektyvumas. Didžiąją laiko dalį gydytojas sugaišta informacijos apie pacientą paieškai, bei apžiūros rezultatų apiforminimui. AB "Alna" atlikto tyrimo[3] duomenimis, paciento apžiūrai ir diagnozės nustatymui sugaištama vos 30% gydytojo laiko. Informacijos paieška popierinėje paciento ligos istorijoje, neefektyvi ir paliekanti pakankamai didelę riziką gydytojui suklysti. Dėl šios priežasties popierinį keitimosi paciento duomenimis formatą pakeisti elektroniniu yra ypač aktualu, tačiau kartu ir problematiška.

Sveikatos priežiūros sektorius ypatingas savo konteksto apimtimi, informacijos fragmentacija, bei veiklos ir informacijos poreikių heterogeniškumu, todėl segmentų apjungimas į vieną bendradarbiaujančią sistemą yra ypač problematiškas. Didelis segmentų rišlumas smarkiai apsunkina tokios sistemos plečiamumą ir priežiūrą, kadangi keičiamas komponentas gali įtakoti su juo susijusius komponentus. Dėl šių priežasčių ypač dideli reikalavimai keliama suderinamumui, ir lankstumui. Vienas iš šiuos reikalavimus tenkinančių sprendimų yra skirtingų sveikatos priežiūros informacinių segmentų apjungimas grįstas paslaugų architektūra (angl. *Service Oriented Architecture*).

Paslaugų architektūra remiasi reikiama veiklos logikos apimtį inkapsuluojančia paslaugos sąvoka, bei akcentuoja atvirą, paplitusių duomenų apsikeitimo standartų panaudojimą. Dėl šių priežasčių paslaugomis grįsta architektūra pasižymi geromis suderinamumo, pakartotinio panaudojimo bei lankstumo savybėmis.

Šio darbo tikslas - įvertinti paslaugomis grįstos architektūros taikymą, kuriant apsikeitimui paciento duomenimis skirtas sistemas. Siekiant objektyvesnių įvertinimo rezultatų darbui keliama šie uždaviniai:

- Atlikti sveikatos priežiūros sektoriaus informacinių poreikių analizę ir identifikuoti potencialias kliūtis duomenų apsikeitimui
- Įvertinti paslaugų architektūros principus, jų pranašumus ir principų realizavimui tinkančias technologijas
- Sudaryti paslaugų architektūra grįstą keitimosi paciento duomenimis modelį

- Parinkti modeliui realizuoti reikalingas technologijas ir praktiškai realizuoti pagrindines modelio elementų funkcijas
- Įvertinti kūrimo metu kilusias problemas bei pastebėjimus

Darbą sudaro šios dalys:

Sveikatos priežiūros sektoriaus analizė. Gilinamasi į sveikatos priežiūros sektoriaus informacinius poreikius. Nagrinėjami sveikatos priežiūros įstaigos duomenų srautai, išskiriamos ir analizuojamos duomenų apsikeitimo problemos, apžvelgiami galimi informacijos keitimosi modeliai. Apibendrinime pateikiami pagrindiniai reikalavimai apsikeitimą paciento duomenimis realizuojančioms sistemoms.

Paslaugų architektūros projektavimo metodika. Nagrinėjami paslaugų architektūros principai ir jų teikiama nauda. Atliekamas palyginimas su komponentų architektūra. Tiriamos paslaugų architektūros realizacijai tinkamos technologijos. Labiausiai gilinamasi į “žiniatinklio paslaugų” (angl. *Web Services*) technologiją.

Paslaugų architektūra grįstas modelis. Pristatomas paslaugų architektūra paremtas sveikatos sektoriaus įstaigų bendradarbiavimo modelis.

Paslaugų architektūra grįsto modelio realizavimas. Apibrėžiamas modelio realizavimo tikslas. Pristatomas posistemių funkcionalumas, jų bendradarbiavimas ir architektūrinis vaizdas. Pagrindžiamos modeliui kurti pasirinktos priemonės ir technologijos.

Paslaugų architektūra grįsto prototipo efektyvumo tyrimas. Tiriama realizuoto modelio kokybė, apsikeitimo duomenimis poreikiu atitikimas. Nagrinėjamos modelio plėtimo ir taikymo galimybės. Pristatomi modelio realizacijos metu kilę sunkumai ir pastebėjimai.

Paslaugų architektūra – pakankamai nauja, ir vis dar besivystanti, tačiau smarkiai palaikoma didžiųjų rinkos dalyvių. Dėl šios priežasties, daugumoje literatūros šaltinių pateikdami paslaugų architektūros privalumai, o apie trūkumus nutylima. Darbo metu atlikta analizė padėjo įvertinti ir paslaugomis grįsto sistemos prototipo realizavimas padėjo įvertinti ne tik architektūros teikiamas naudas bet ir trūkumus bei pavojus.

Modelio prototipui realizuoti panaudotos modernios “ASP.NET 2”, “Microsoft SQL Server 2005” technologijos, bei “Microsoft Visual Studio 2005” programų kūrimo aplinka, leidusios per palyginti trumpą laiką realizuoti pakankamai nemažą posistemių funkcionalumą. Nors modelio prototipe realizuotos tik pagrindinės, dažniausiai pasitaikančios funkcijos,

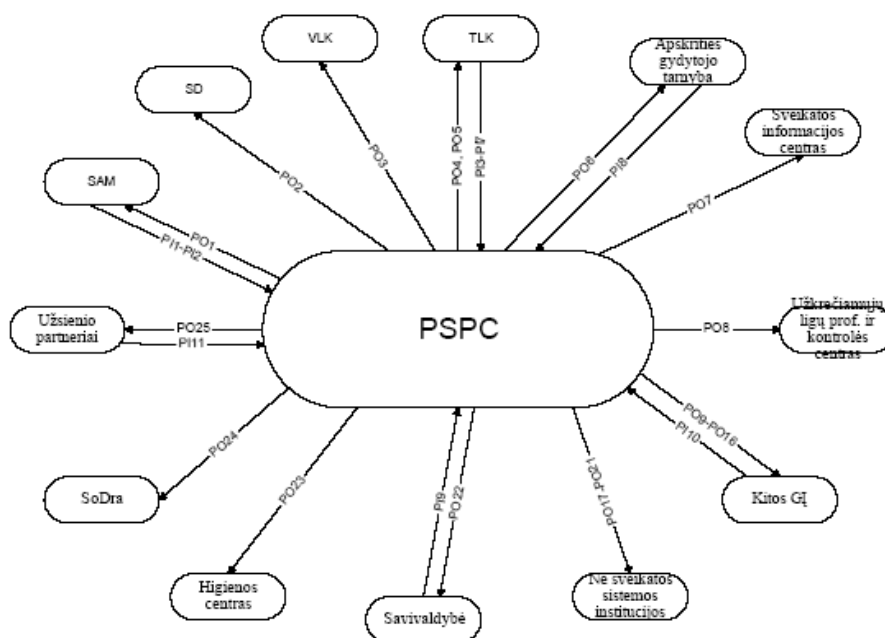
tačiau apibrėžta paslaugų sąsaja leidžia laisvai modifikuoti posistemas, neįtakojant jas naudojančių sistemos dalių. Sukurtas modelio prototipas gali būti ir toliau plėtojamas, jungiant naujas posistemas. Kai kurios posistemių teikiamos žiniatinklio paslaugos pvz. Vaistų registro jau dabar gali būti naudojamos kitų nutolusių sistemų.

2 APSIKEITIMO PACIENTO DUOMENIMIS POREIKIŲ ANALIZĖ

2.1 Informacijos srautų analizė

Informacijos srautų analizei naudotasi 2003 m. AB „ALNA“ vykdytos gydymo įstaigų apklausos duomenimis [3].

Dideli informacijos srautai vyksta ir antriniuose sveikatos priežiūros įstaigose, tačiau detaliau analizuosime pirminio sveikatos priežiūros centro duomenų srautus. Pirminiams sveikatos priežiūros centrams tenka didžiausias pacientų krūvis, todėl informacijos apsikeitimo problema čia yra aktualiausia. Pirminio sveikatos priežiūros centro informacijos srautai pateikti 1 paveiksle ir lentelėse 1 ir 2.



1 pav. Pirminio sveikatos priežiūros centro duomenų srautai[3]

Iš duomenų srautų diagramos matome, jog PSPC pagrindinę besikeičiamos informacijos dalį sudaro skirtingoms įstaigoms skirtos ataskaitos, o ne apsikeitimas paciento duomenimis su kitomis gydymo įstaigomis. Didelė ataskaitų dalis ruošiama remiantis įvairiai agreguota informacija apie pacientui suteiktas paslaugas.

Išeinantys pirminio sveikatos priežiūros centro duomenų srautai [3]

Kodas	Aprašymas	Istaiga
PO1	Ataskaitos	Sveikatos apsaugos ministerija
PO2	Ataskaitos	Statistikos departamentas
PO3	Ataskaitos	VLK
PO4	Ataskaitos	TLK
PO5	Pacientų registracija	TLK
PO6	Ataskaitos	Apskritis gydytojo tarnyba
PO7	Ataskaitos	Sveikatos informacijos centras
PO8	Ataskaitos	Užkrečiamųjų ligų profilaktikos ir kontrolės centras
PO9	Išrašas iš ligos istorijos	Kitos GĮ
PO10	Informacija apie klientų apsilankymą ir jiems suteiktas paslaugas	Kitos GĮ
PO11	Informacija apie prisirašiusius ligonius	Kitos GĮ
PO12	Skiepų ataskaita	Kitos GĮ
PO13	Sergamumas gripais, kataru	Kitos GĮ
PO14	Nedarbingumo pažymėjimų šaknelės	Kitos GĮ
PO15	Pacientų psichiatrinė informacija	Kitos GĮ
PO16	Darbo klausimai	Kitos GĮ
PO17	Atskirų profesijų darbuotojų darbo apmokėjimo ataskaita	Ne sveikatos sistemos institucijos
PO18	Darbo apmokėjimo ataskaita	Ne sveikatos sistemos institucijos
PO19	Darbo kainos ataskaita	Ne sveikatos sistemos institucijos
PO20	Sveikatos priežiūros ir socialinio darbo paslaugų ataskaita	Ne sveikatos sistemos institucijos
PO21	Labdaros, paramos gavimo ir paskirstymo ataskaita	Ne sveikatos sistemos institucijos
PO22	Ataskaitos	Savivaldybė
PO23	Ataskaitos	Higienos centras
PO24	Nedarbingumo pažymėjimai	Sodra
PO25	Žinutės, dokumentai	Užsienio partneriai

Išeinantys pirminio sveikatos priežiūros centro duomenų srautai [3]

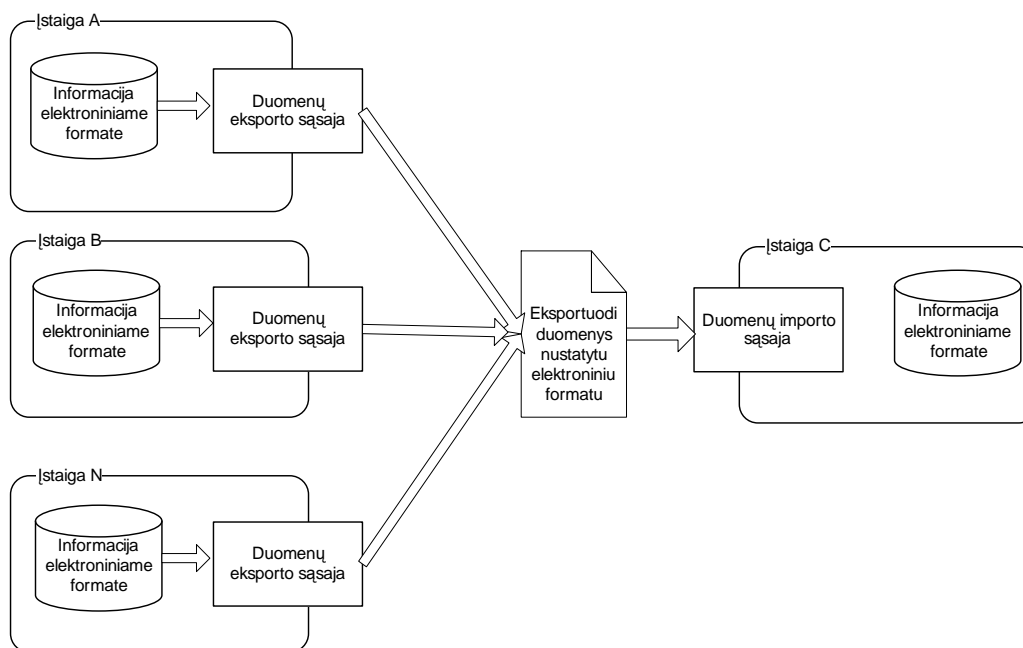
Kodas	Aprašymas	Istaiga
PI1	Statistinės ataskaitos	Sveikatos apsaugos ministerija
PI2	Dokumentai informaciniai pranešimai, įsakymai	Sveikatos apsaugos ministerija
PI3	Informacija apie prisirašiusių žmonių prirašymą prie gydymo įstaigos	TLK
PI4	Informacija apie pacientų gydymo vietą	TLK
PI5	Informacija apie išrašomus medikamentus	TLK
PI6	Ataskaitos	TLK
PI7	Informaciniai pranešimai	TLK
PI8	Raštai	Apskritis gydytojo tarnyba
PI9	Dokumentai	Savivaldybė
PI10	Psichiatrinė pacientų informacija	Kitos GĮ
PI11	Žinutės, dokumentai	Užsienio partneriai
PI12	Informacija apie vaistus	Farmacijos įmonės

2.2 Keitimosi informacija problemos

2.2.1 Popierinis duomenų apsikeitimo formatas

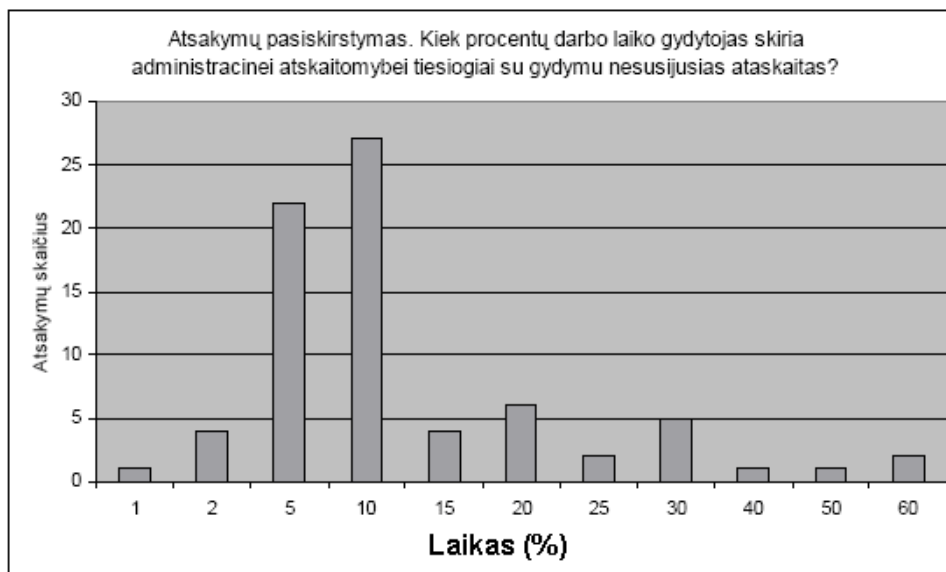
Šiuo metu informacijos mainai vyksta popieriniame formate. Tai ypač neefektyvus apsikeitimo būdas, tiek sugaišto laiko, tiek informacijos apdorojimo prasmėmis. Viena pagrindinių priežasčių, stabdančių perėjimą prie efektyvaus elektroninio duomenų apsikeitimo formato, yra skirtingų įstaigų naudojamos programinės įrangos heterogeniškumas. Informacijos formatas ir poreikiai keičiasi pakankamai dažnai, o tai lemia pakankamai dideles programinės įrangos adaptavimo išlaidas.

Paanalizuokime duomenų apsikeitimo scenarijų pagal 2 paveiksle pateiktą schemą. Įstaigos A, B ir N įstaigai C, teikia duomenis sutartu elektroniniu formate. Kiekvienoje įstaigoje į standartinį elektroninį formatą, duomenys paverčiami per specialią duomenų eksporto sąsają. Įstaiga C duomenis priima per duomenų importo sąsają. Problema iškyla pasikeitus įstaigos C informaciniais poreikiais. Pakeitus nustatytą duomenų apsikeitimo formatą, įstaigos A, B ir N privalės keisti savo duomenų eksporto sąsajas. Taigi, kiekvienas duomenų formato keitimas sukeltų dideles modifikavimo išlaidas duomenis teikiančioms įstaigoms. Turėkime omenyje analizuoto modelio paprastumą: duomenų srautas yra vienpusis, o duomenų apsikeitimo ryšio tipas – „daug - vienas“. Esant abipusiam „daug – daug“ apsikeitimo ryšiui, ši problema tampa ypač aktuali.



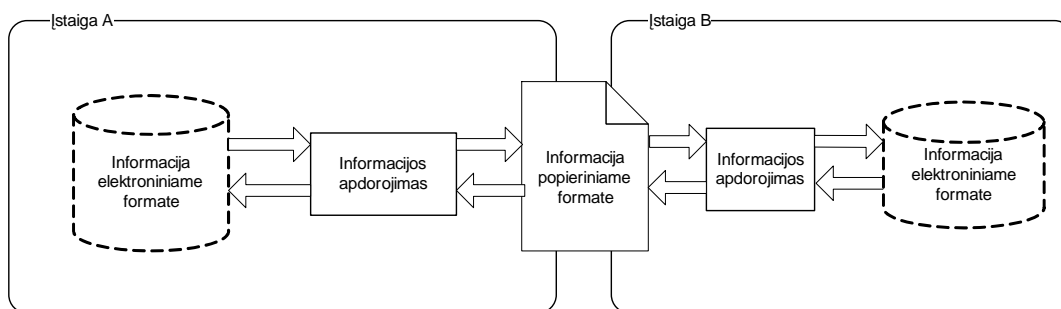
2 pav. Duomenų pasikeitimo elektroniniu formatu schema.

Daugumos gydymo įstaigų teikiamų ataskaitų duomenų šaltinis yra įvairūs, gydytojo pildomi atskaitomybės žurnalai, apie pacientui suteiktas paslaugas. Pagal AB „Alna“ 2003 metais atliktą gydytojų apklausą [3], apie 20% laiko gydytojai skiria su gydymu nesusijusių ataskaitų pildymui 3 pav. Didelė informacijos dalis, skirtingoms įstaigoms teikiamose ataskaitose, dubliuojasi. Kaupiant informaciją apie pacientams suteiktą informaciją elektroniniu formatu, šį darbą galima visiškai eliminuoti, o sutaupytą laiką, gydytojai galėtų skirti tiesioginėms savo pareigoms.



3 pav. Gydytojų laikas praleistas ruošiant tiesiogiai su gydymu nesusijusias ataskaitas [1]

Kai kurios gydymo įstaigos, pvz., Santariškių universitetinės klinikos jau kompiuterizavusios informacijos apie pacientą kaupimą. Į tai investuota pakankamai daug laiko ir pinigų. Tačiau ir laikanti paciento informaciją elektroninėje formoje gydymo įstaiga, dėl standartinio apskaitos duomenis formato nebuvimo, kitoms įstaigoms privalo teikti popieriniu formatu (4 pav.). Tai mažina informacijos apikeitimo proceso efektyvumą ir informacijos analizės galimybes. Be to popierinių ataskaitų formatas dažnai keičiamas, todėl elektronines sistemas turinčios įstaigos turi papildomų išlaidų adaptuojant sistemas naujam ataskaitų formatui.



4 pav. Įstaigų keitimosi informacija schema

2.2.2 Informacijos fragmentacija organizacine prasme

Gydymo įstaigos pačios gali nuspręsti, koku formatu koduoti ir klasifikuoti paciento ambulatorinę informaciją. Gydymo įstaigos, sudariusios sutartis su TLK, privalo pateikti informaciją TLK nustatytu elektroniniu formatu. Šiuo metu oficialus diagnozių klasifikatorius yra TLK-10 (Tarptautinis ligų klasifikatorius), o vaistų klasifikacijai naudojamas ATC klasifikatorius. Už TLK-10 klasifikatoriaus administravimą atsakinga Valstybinė ligonių kasa. Už vaistų klasifikatorių, ir vaistų sąrašą atsakinga Valstybinė vaistų kontrolės tarnyba. Už pacientų registrą atsakinga SODRA. Visos šios įstaigos informaciją administruoja nepriklausomai viena nuo kitos.

2.2.3 Informacijos fragmentacija geografine prasme

Paciento ambulatorine informacija keičiamasi tarp skirtingų, viena nuo kitos nutolusių įstaigų. Besikeičiančių informacija įstaigų naudojama tinklo ir programinė įranga smarkiai skiriasi, todėl pilnavertis keitimasis duomenimis galimas tik paplitusiu, daugumos programinės ir tinklo įrangos gamintojų pripažintu duomenų apsikeitimo formatu. Tarp apsikeitimo duomenimis taškų yra saugasiens, kurios sunkina apsikeitimą duomenimis.

2.2.4 Skirtingi informacijos apie pacientą poreikiai

Pirminiams sveikatos priežiūros centrams svarbi informacija apie pacientui nustatytas diagnozes, išrašytus vaistus ir tyrimų rezultatai.

Tyrimų laboratorijoms svarbios tyrimų užklausos, tyrimų eigos sekimas.

Vaistinėms svarbi vaistų registro informacija ir informacija apie pacientui išrašytus receptus.

Teritorinėms ligonių kasoms svarbi agreguota informacija apie gydymo įstaigos suteiktas paslaugas, nustatytas diagnozes ir išrašytus vaistus.

Informacijos apie pacientą poreikiai tarp įstaigų pasikartoja tik iš dalies. Informacija tarp įstaigų dažniausiai keičiamasi popieriniu formatu. Didelė dalis informacijos apsikeitimo dokumentuose pasikartoja. Be to kiekvienai įstaigai reikalinga specializuota programinė įranga. Perkant specializuotą programinę įrangą labiausiai kreipiamas dėmesys į kainos ir funkcionalumo santykį, o integracijos galimybėmis paprastai nėra domimasi. Todėl šiuo metu, skirtingose įstaigose naudojama heterogeninė programinė įranga, veikianti skirtingose platformose, kuri dažniausiai nėra pritaikyta tarpusavio integracijai. Kurti ar pirkti naują, tarpusavyje suderinamą programinę įrangą, būtų nelogiškai brangu, kadangi tektų dubliuoti į programinę įrangą įstaigų padarytas investicijas, be to išlieka rizika, jog bus nepatenkinti konkrečios gydymo įstaigos poreikiai funkcionalumui [1].

2.3 Panašių užsienio projektų analizė

Europoje vykdoma eilė elektroninių sveikatos priežiūros paslaugų projektų. Reikia turėti omenyje, jog Europos projektai, pagal apimtis problematiką, nėra tiesiogiai palyginami su Lietuvos situacija (tiek pagal šalių dydį, IT išsivystymą, jau esančių ir pilnai veikiančių sistemų kiekį), todėl 3 lentelėje pateiktos problemos ir atradimai, susiję tik su elektroniniu sveikatos informacijos apsikeitimu. [1]

3 lentelė

Užsienio šalių patirtis elektroninės sveikatos projektuose

Šalis/ Projektas	Problemos ir atradimai
Anglija/ „NHS“	Sunkiausia integruoti įstaigų vidaus veiklos procesus
Graikija/ „HYGEIAnet“	Dėl didelio gydymo įstaigų geografinio išsibarstymo, medicininės technikos skirtumų, esamų gydymo įstaigų pageidavimų skirtumų buvo nuspręsta nekurti monolitinės ir centralizuotos informacinės sistemos.
	Kadangi kai kurios aplikacijos jau yra rinkoje, taip pat kai kurios jau yra įdiegtos, buvo nuspręsta panaudoti esamus programinius produktus.
	Esamoms sistemoms buvo iškeltas reikalavimas, kad jos galėtų laisvai ir standartizuotai priimti ir perduoti informaciją, pasinaudojant atvira architektūra ir standartais.
Šveicarija/ “Triamun Project”	Įvykdžius projektą buvo prieita prie išvados, kad visų gydymo įstaigų, ir gyventojų priversti naudotis viena sistema nėra įmanoma. Kuriant nacionalines ar europines sistemas, reikėtų orientuotis į tai, kad nebus vieningos sistemos, tačiau atskiros sistemos pagal bendrus protokolus bendraus tarpusavyje.

2.4 Galimų keitimosi informacija modelių analizė

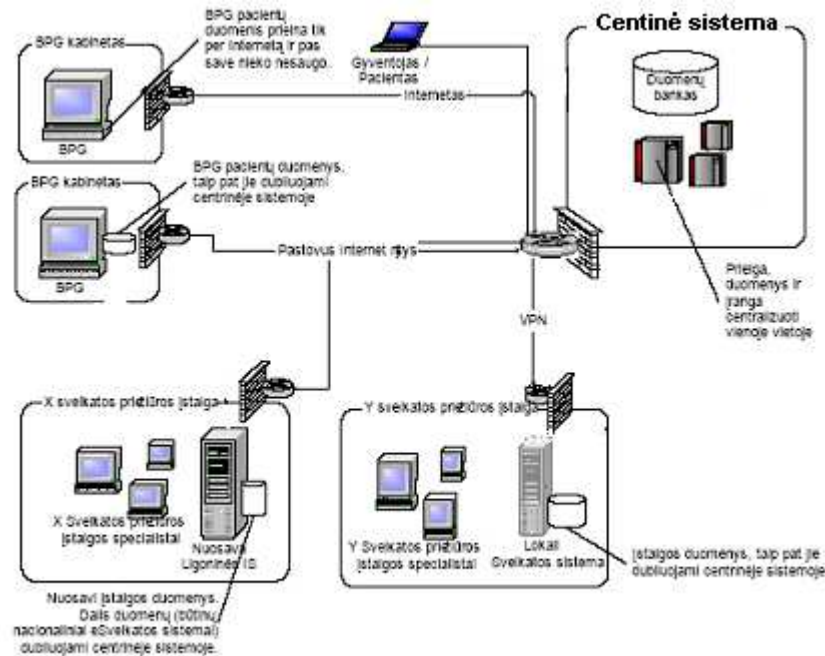
SAM galimybių studijoje [1] pateikti trys galimi keitimosi paciento duomenimis realizavimo modeliai:

- centralizuotas,
- paskirstytasis,
- decentralizuotas.

2.4.1 Centralizuotas modelis

Duomenys saugomi centralizuotai. Duomenimis keičiamasi per žiniatinklio sąsają.

Gydymo įstaigos, turinčios savo lokalias IS, duomenis pateikia per centralizuotą sistemą.



5 pav. Centralizuotas keitimosi paciento duomenimis modelis [1]

Privalumai:

- mažesnės sukūrimo sąnaudos,
- paprastesni apsikeitimo duomenimis mechanizmai,
- pigesnė eksploatacija,
- paprastesnis palaikymas,
- centralizuotas duomenų tvarkymas,
- didesnis informacijos saugumas,
- lengvesnis papildomu paslaugu įdiegimas,
- aiškus sistemos savininkas, sistemos palaikymas ir plėtra

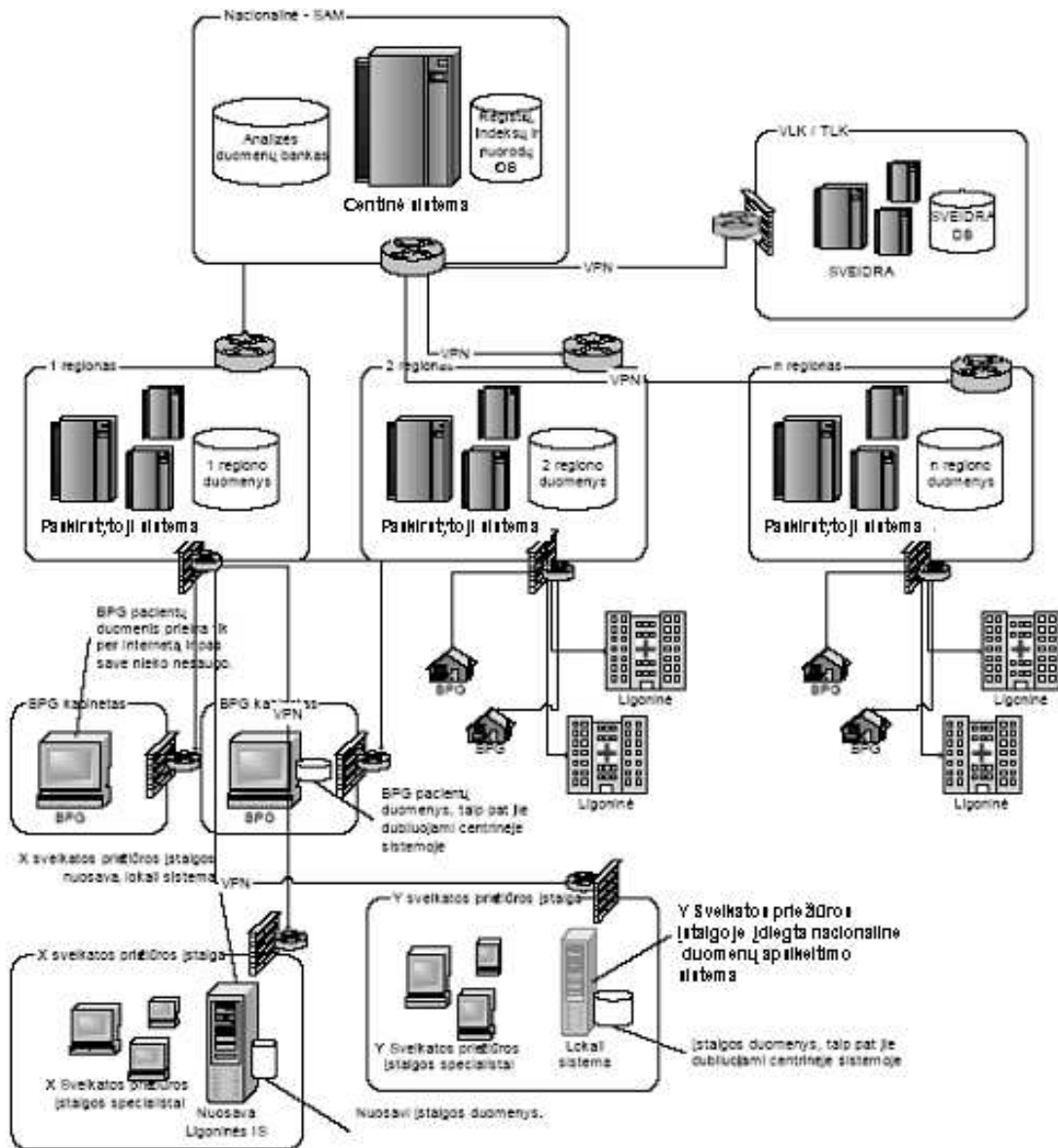
Trūkumai

- mažesnis prieinamumas gedimo atveju,
- dideli informacijos srautai į vieną tašką,
- ribotos galimybės diegti skirtingus sprendimus atskiriems regionams ir SPI

2.4.2 Paskirstytasis modelis

Centralizuotai teikiamos duomenų, indeksavimo ir nuorodų paslaugos regioninėms (paskirstytosioms) sistemoms.

Regioninės sistemos teikia paslaugas savo regiono sveikatos priežiūros įstaigoms, kurios neturi lokalių IS. Visi regioninių sistemų duomenys dubliuoti kaupiami taip pat centrinėje duomenų saugykloje analizės ir statistikos tikslams. Visi sistemos vartotojai dirba per žiniatinklio sąsają su savo regiono sistema. Įstaigos, kurios turi lokalias IS, duomenų pateikimą ir apsikeitimą su kitomis įstaigomis vykdo per regioninę sistemą, o regioninė – užtikrina informacijos apsikeitimą ir pateikimą Nacionalinei sistemai.



6 pav. Paskirstytasis apsikeitimo paciento duomenimis modelis [1]

Privalumai

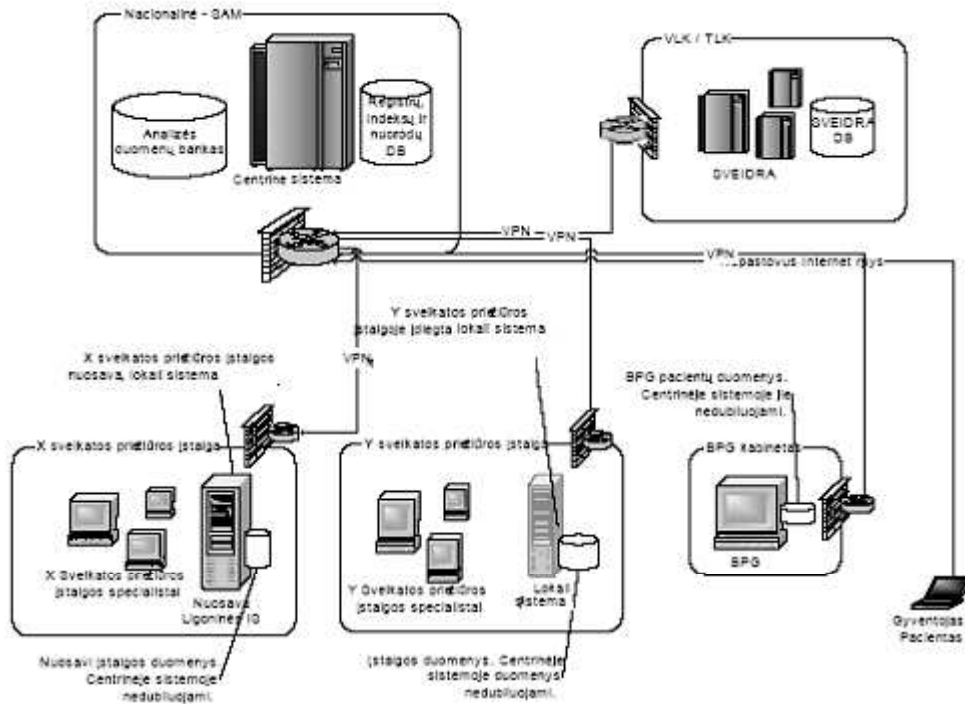
- greitesnė prieiga internetu,
- serverių apkrovos paskirstymas,
- geresnis paslaugos prieinamumas gedimo atveju,
- galimybė teikti specifines paslaugas regiono pacientams,
- galimybė atskiruose regionuose diegti skirtingas, bet suderinamas sistemas.

Trūkumai

- brangesnis kūrimas,
- brangesnis aptarnavimas,
- reikalauja sudėtingos duomenų replikavimo sistemos,
- informacijos vėlinimas dėl replikavimo lėtumo arba ryšio užimtumo ,
- sudėtingesnė bendrų papildomų paslaugų integracija.

2.4.3 Decentralizuotas modelis

Centrinė serverių grupė ir duomenų saugykla teikia duomenų indeksavimo paslaugas decentralizuotoms (SPI) sistemoms. Dalis sistemos duomenų kaupiami taip pat centrinėje duomenų saugykloje analizės ir statistikos tikslams. Visi sistemos vartotojai dirba su lokaliomis gydymo įstaigų sistemomis, kuriose ir saugomi visi įstaigos pacientų duomenys. Pacientai identifikuojami centrinėje sistemoje, kuri užtikrina pagal nuorodas į informacijos šaltinius jos surinkimą ir pateikimą. Įstaigos, kurios turi savo lokalias IS duomenų pateikimą ir apsikeitimą su kitomis įstaigomis vykdo, pasinaudamos centrinės sistemos teikiamomis indeksavimo ir nuorodų paslaugomis.



7 pav. Decentralizuotas apsikeitimo paciento duomenimis modelis [1]

Privalumai

- serverių apkrovos paskirstymas,
- geresnis paslaugos prieinamumas gedimo atveju

Trūkumai

- lėtesnė prieiga internetu,
- kiekviena SPĮ turi turėti bent vidutinio galingumo serverį,
- brangesnis aptarnavimas ir plėtimas,
- reikalauja sudėtingos duomenų prieigos ir replikavimo sistemos,
- informacijos vėlinimas dėl replikavimo lėtumo arba ryšio užimtumo,
- sudėtingesnė bendrų papildomų paslaugų integracija,
- didelė SPĮ atsakomybė už informaciją ir prieinamumą.

2.5 Apibendrinimas

Dėl skirtingų sveikatos priežiūros įstaigų informacinių poreikių, netikslinga kurti centralizuotos monolitinės elektroninio paciento įrašo sistemos. Tuo pačiu negalima atsakomybės už paciento įrašo saugojimą palikti kiekvienai sveikatos priežiūros įstaigai, kadangi nebus aiškus keitimosi paciento duomenimis savininkas, dėl šios priežasties standartizuotas apsikeitimas duomenimis taptų sunkiai įgyvendinamas. Smarkiai skiriasi įvairių sveikatos įstaigų informaciniai poreikiai

Didelė funkcionalumo dalis dubliuotų jau esamų sistemų funkcionalumą ir duomenis. Esamos infrastruktūros pakartotinis panaudojimas smarkiai sumažintų apsikeitimo paciento duomenimis sistemos kūrimo kaštus.

Įvertinus apsikeitimo paciento duomenimis poreikius bei galimus duomenų apsikeitimo modelius, galime išskirti pagrindinius reikalavimus paciento duomenų apsikeitimą realizuojančioms sistemoms. Svarbiausi reikalavimai pateikti 4 lentelėje.

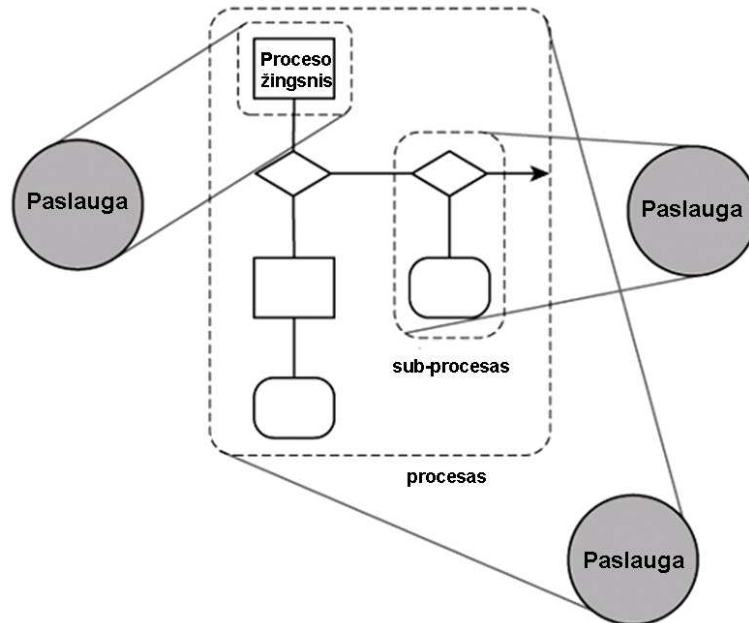
4 lentelė

Reikalavimai keitimosi paciento duomenimis sistemose

Reikalavimas	Svarba
Duomenų apsikeitimas tarp heterogeninių sistemų	Kritinis
Galimybė integruoti fiziškai nutolusias sistemas	Labai svarbus
Apsikeitimui duomenimis naudojamo formato palaikomumas, standartizacija ir taikymo lankstumas	Labai svarbus
Duomenų perdavimo sluoksnio paslėpimas	Labai svarbus
Plečiamumas	Kritinis
Palaikomumas	Kritinis
Pakartotinis panaudojamumas	Svarbus
Duomenų perdavimo saugumas	Kritinis
Integruojamų sistemų nepriklausomumas	Svarbus
Greitaveika	Labai svarbus
Prieinamumas	Kritinis

3 PASLAUGŲ ARCHITEKTŪROS METODIKA

Paslaugų architektūra (Angl. *Services Oriented Architecture*) yra paskirstytųjų sistemų kūrimo modelis. Paslaugų architektūra remiasi paslaugos sąvoka. Paslaugos koncepcija yra patogi tuo, jog paslauga galima apibrėžti įvairių veiklos logikos kieki (8 pav.).



8 pav. Paslaugų ir procesų sąsaja

3.1 Metodikos principai

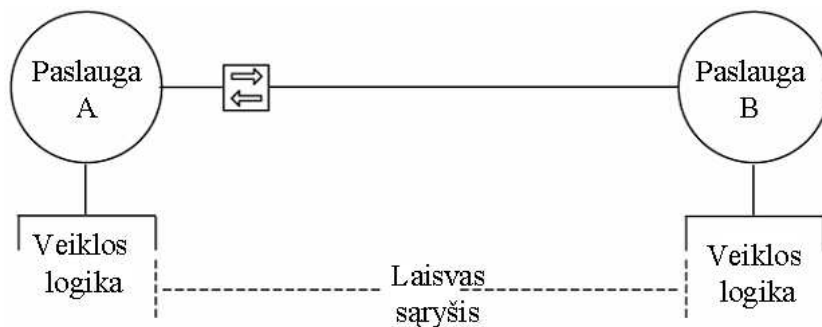
3.1.1 Laisvas sąryšis (angl. *Loose coupling*)

Paslaugomis grįstame, kaip ir tradiciniuose paskirstytosios architektūros modeliuose, atskiri programinės logikos vienetai integruojami spręsti tam tikras užduotis. Pagrindinė charakteristika, išskirianti paslaugomis grįstą architektūrą iš kitų paskirstytosios architektūros modelių, yra tamprus logikos vienetų sąryšis. Toks sąryšis pagrįstas logikos vienetų gebėjimu sužinoti, kaip panaudoti ir panaudoti kitus logikos vienetus, tuo pačiu liekant nuo jų nepriklausomais. Paveiksle 9 pavaizduota, kaip paslaugos A veiklos logika naudoja paslaugos B veiklos logiką. Paslaugos susietos tik aprašais, todėl paslaugos B logika gali būti pakeista alternatyvios paslaugos logika, nekeičiant nieko paslaugos A realizacijoje.

Laisvo sąryšio principai:

- Pakeitimai paslaugos tiekėjo realizacijoje neturi daryti įtakos paslaugos naudotojui.

- Paslaugos naudotojas gali pakeisti paslaugos tiekėją į alternatyvų, nekeisdamas programinio kodo.



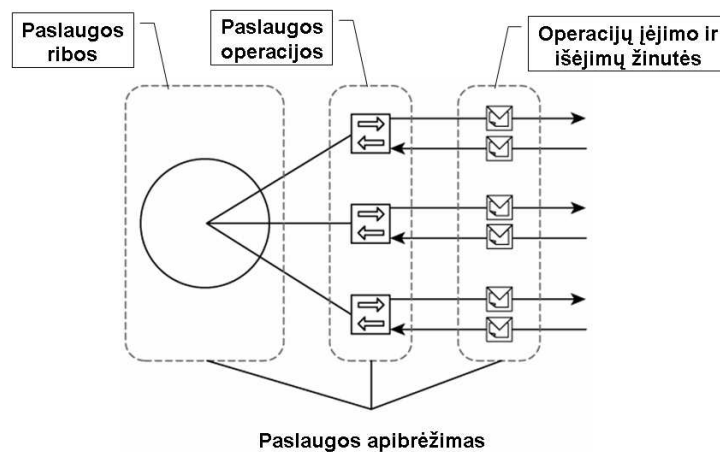
9 pav. Paslaugų nepriklausomumo principas [7]

3.1.2 Standartizuota sąsaja

Paslaugos turi turėti sutartą tarpusavio bendravimo būdą. Tai pasiekama naudojant sutartą paslaugų apibrėžimo semantiką.

Paslaugos apibrėžimas nurodo:

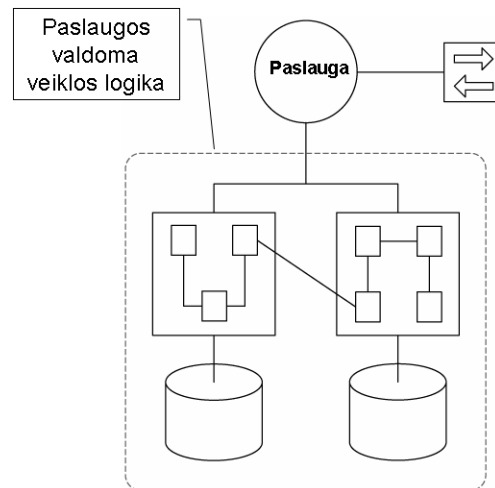
- paslaugos ribas,
- paslaugos operacijas,
- paslaugos operacijoms naudojamų įėjimo ir išėjimo žinučių apibrėžimus.



10 pav. Standartizuotos sąsajos principas [7]

3.1.3 Paslaugų autonomiškumas

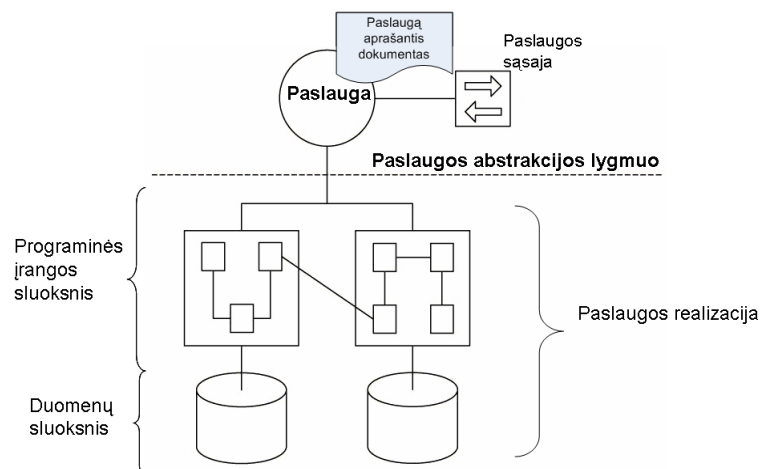
Paslaugos valdomos veiklos logikos ribos turi būti aiškiai apibrėžtos ir atskirtos. Paslaugos veikimo metu, veiklos logikos neturi įtakoti išoriniai veiksniai. Šis principas sumažina paslaugos priklausomybę nuo išorinių veiksnių, mažinančių paslaugos lankstumą.



11 pav. Paslaugos autonomiškumo principas [7]

3.1.4 Paslaugos lygmens abstrakcija

Šis principas leidžia paslaugą laikyti "juodąja dėže". Paslaugos veikimo detalės paslepamos nuo išorinio pasaulio. Paslaugos apibrėžime aprašoma paslaugos sąsaja, paslaugai reikalingi duomenys ir grąžinami rezultatai. Viena paslauga galima apibrėžti neribotą veikimo logikos kiekį. 12 paveiksle pateiktame pavyzdyje paslauga apima dviejų taikomųjų programų veikimo logiką.



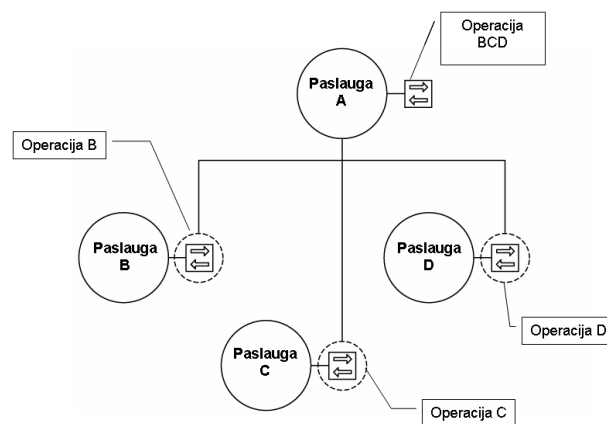
12 pav. Paslaugos lygmens abstrakcijos principas [7]

3.1.5 Pakartotinis panaudojumas

Veikimo logika suskirstoma į paslaugas su tikslu atskiras paslaugas pakartotinai panaudoti kitiems tikslams, nors kuriant programinę įrangą reikalavimai pakartotiniam panaudojimui nėra keliami. Paslaugos pakartotiniam panaudojimui didžiausią įtaką turi paslaugos apibrėžimo standarto paplitimas.

3.1.6 Paslaugų suderinamumas

Tai viena iš pakartotinio panaudojimo formų. Paslaugos projektuojamos taip, kad iš kelių paslaugų galima būtų sudaryti naują paslaugą 13 pav.



13 pav. Paslaugų suderinamumo principas [7]

3.1.7 Būsenos atsisakymas

Rekomenduojama paslaugas projektuoti ir realizuoti taip, jog paslaugos naudotojams nebūtų svarbi paslaugos būseną. Būsenos įtaka minimizuojama, kai bendravimui su paslauga naudojamos dokumento tipo žinutės. Mažesnė būsenos įtaka sąlygoja geresnes paslaugos prieinamumo charakteristikas.

3.2 Palyginimas su komponentų architektūra

Paslaugomis grįstos architektūra remiasi paslaugos sąvoka. Paslaugą galime laikyti kaip pakartotinai panaudojamą elementą iš kurio komponuojama sistema. Pakartotinai panaudojamo elemento sąvoka remiasi ir komponentų architektūra paremtos technologijos. Pagrindinės jų „Microsoft DCOM“ , „ Enterprise JavaBeans (EJB)“, ir „CORBA“. Visos šios technologijos išpildo apskaitimui tokius paciento duomenimis svarbius nefunkcinius reikalavimus kaip: dinaminis komponento iškvietimas, duomenų perdavimo tinklu paslėpimas ir paskirstyti skaičiavimai.

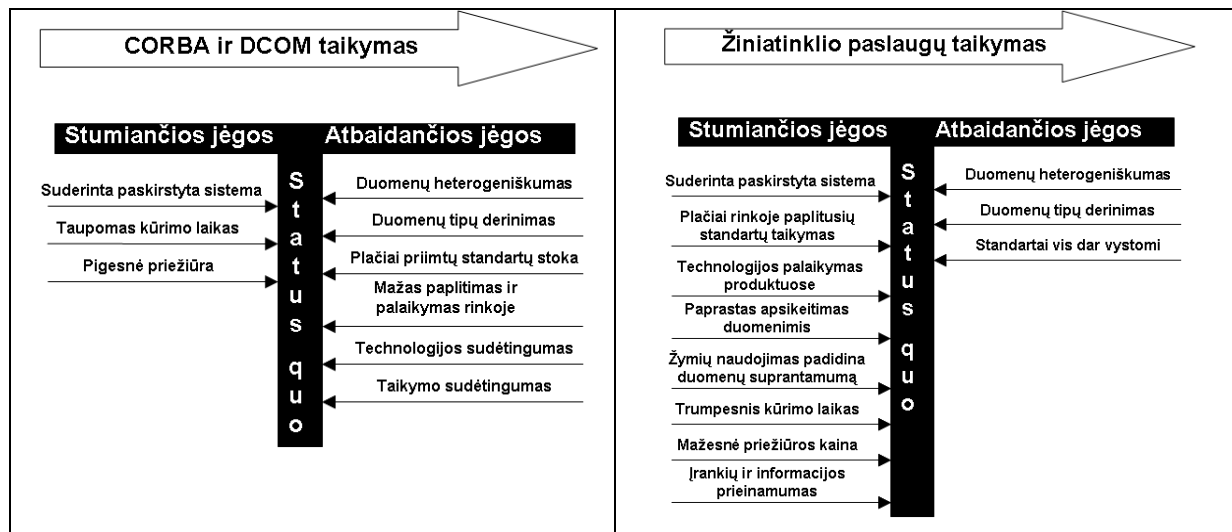
Tinkamai suprojektavus komponentus, jie gali tapti ir paslaugomis (pvz. „CORBA Trader“ technologija). Pagrindinis skirtumas tarp komponentų ir paslaugų architektūros yra elementų projektavimo integravimo būdai. Paslaugos aprašas visiškai atskiriamas nuo veikimo, tuo tarpu komponento veiklos logiką galima įtakoti per sąsajas. Paslaugos integruojamos vykdymo metu, o komponentų kombinavimas yra iš anksto apibrėžtas. Taigi paslaugų ir komponentų architektūros yra panašios, tačiau skirtos skirtingiems tikslams. Paslaugų architektūra naudinga, kai programos elementą reikia surasti, panaudoti ir pakeisti vykdymo metu, bei svarbus programos elementų nepriklausomumas. Komponentų architektūra naudinga tada, kai komponentų fizinė vieta yra aiškiai apibrėžta, programos elementų sąryšio stiprumas nesvarbus, naudojami transakcijų reikalaujantys veiksmai ir keliami dideli reikalavimai greitaveikai. Pagrindiniai architektūrų skirtumai pateikti 5 lentelėje.

Paslaugų ir komponentų architektūrų palyginimas

Architektūra Savybė	Komponentų architektūra	Paslaugų architektūra
Kūrimo modelis	Sistemos elementų kūrimas atskiriamas nuo sistemos kūrimo. Sistema surinkimas remiasi sukurtų komponentų kompozicija	Sistemos elementų kūrimas atskiriamas nuo sistemos kūrimo. Sistemos surinkimas remiasi abstrakčiais paslaugų aprašais.
Elemento sąvoka	Išorinė sąsaja ne visada atskiriama nuo realizavimo logikos. Komponentų egzemplioriai kuriami iš komponentų. Siekiant supaprastinti komponentų panaudojimą, komponentai pakuojami atskirais vienetais.	Paslaugos aprašas nuo realizacijos visiškai atskiriamas. Paslaugos tiekėjas sukuria paslaugos objektą. Paslaugos veiklos logikos ribos yra aiškios, tačiau paslaugą realizuojanti logika nėra fiziškai pakuojama
Komponavimas	Komponentai apjungiami konkretaus aprašo, kuris nurodo kaip komponentai yra apjungiami ir naudojami, pagalba. Galimybė vykdymo metu panaudoti naujus komponentus Struktūrinio projektavimo tendencijos	Paslaugų panaudojimas remiasi abstrakčiais paslaugų aprašais. Komponavimas vyksta vykdymo metu. Naudojant paslaugą reikia atsižvelgti į jos prieinamumą. Į vykdomą proceso aprašą orientuotos architektūros tendencijos.
Vykdymo aplinkos ypatumai	Už komponento egzemplioriaus sukūrimą ir sunaikinimą atsakingas klientas Nefunkcinių reikalavimų palaikymas	Už paslaugos egzemplioriaus sukūrimą atsakingas tiekėjas, paslaugos naudotojui žinomas tik paslaugos aprašas.
Transakcijų palaikymas	Dėl komponento būsenos išlaikymo savybių, puiki aplinka transakcijų reikalaujantiems veiksams.	Tarp paslaugos naudotojo ir tiekėjo bendravimas vyksta žinutėmis. Būsenos palaikymas įmanomas, tačiau tai labai pablogina paslaugos našumo charakteristikas.
Greitaveika	Iš anksto aprašytas komponavimas, dvejetainis duomenų apsikeitimas sąlygoja geras greitaveikos savybes	Abstraktus duomenų formatas, paslaugų aprašų skaitymas vykdymo metu padidina technikos ir tinklo apkrovimą

3.3 Žiniatinklio paslaugų technologija.

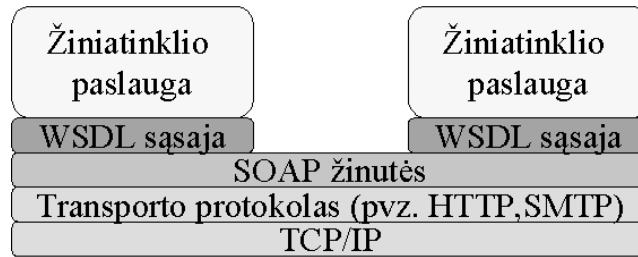
Nors paslaugomis grįstą architektūrą galima realizuoti paskirstytoms sistemoms realizuoti skirtomis CORBA ir DCOM technologijomis, labiausiai paplitusi yra žiniatinklio paslaugų (angl. *Web Services*) technologija. Pagrindiniai trūkumai atbaidantys nuo CORBA ir DCOM yra technologijų sudėtingumas, mažas paplitimas ir palaikymas rinkoje, bei taikymo sudėtingumas. DCOM technologija priklauso kompanijai „Microsoft“ ir veikia tik „Microsoft Windows“ platformose, todėl ji netenkina heterogeninių sistemų apjungimo poreikių. Pagrindiniai technologijų pasirinkimo kriterijai pateikti 14a ir 14b pav. Toliau paslaugomis grįstos architektūros savybes, tyrinėsime žiniatinklio paslaugų kontekste.



14a pav. DCOM ir CORBA technologijų taikymą veikiančios jėgos

14b pav. Žiniatinklio paslaugų technologijos taikymą veikiančios jėgos

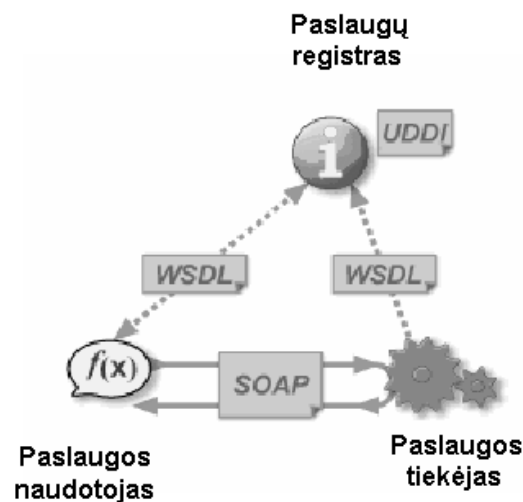
Pagal W3C konsorciumo apibrėžimą, žiniatinklio paslaugos (angl. *Web Services*), yra programų sistema suprojektuota palaikyti kompiuteris-kompiuteris sąveiką per tinklą. Žiniatinklio paslaugų sąsaja yra aprašoma kompiuteriui suprantama WSDL kalba. WSDL aprašą galima patalpinti UDDI registre, Išorinės sistemos su žiniatinklio paslaugomis bendrauja per SOAP žinutes. Aukštas SOAP žinučių abstrakcijos lygmuo leidžia žinučių perdavimą pririšti prie norimo žemesnio lygio protokolo. Žiniatinklio paslaugų bendravimo modelis pateiktas 15 paveiksle.



15 pav. Žiniatinklio paslaugų bendravimo modelis [10]

Žiniatinklio paslaugų technologija yra pagrįsta tik tokiais atvirais, plačiai paplitusiais ir priimtinais standartais, kaip XML. XML dokumento formato ir SOAP protokolo taikymas paslaugų bendravimui, leidžia skirtingomis programavimo kalbomis sukurtoms programoms, veikiančioms skirtingose aplinkose, apsikeisti duomenimis ir naudotis vienas kito per labiausiai paplitusius protokolus kaip HTTP, SMTP, FTP.

Koncepcinė žiniatinklio paslaugų schema pateikta 16 pav.



16 pav. Koncepcinė žiniatinklio paslaugų schema [10]

Žiniatinklio paslaugos savo charakteristikomis pilnai tenkina pagrindinius paslaugomų architektūros principus. Žiniatinklio paslaugos sąsaja aprašoma WSDL kalba, kuria aprašomos tik paslaugos panaudojimo detalės. Ši WSDL savybė tenkina paslaugos lygmens abstrakcijos, paslaugų suderinamumo ir silpno sąryšio reikalavimus. Plačiai taikomo XML dokumento formato panaudojimas duomenų apsikeitimui, smarkiai padidina paslaugų suderinamumą. XML dokumento žymės padidina dokumente esančių duomenų priklausomybę, tuo pačiu sumažindamas paslaugos naudotojo prisirišimą prie paslaugos tiekėjo. Galimybė paslaugos aprašą patalpinti UDDI registre, leidžia paslaugos naudotojui vykdymo metu dinamiškai rinktis paslaugos tiekėją, tai sumažina paslaugos naudotojo

prisirišimą prie paslaugos tiekėjo, o paslaugų paieškos UDDI registre galimybės padidina paslaugos pakartotinio panaudojimo galimybes. Šešto OSI lygmens SOAP duomenų perdavimo protokolas, leidžia duomenų perdavimą patikėti labiausiai paplitusiems penkto OSI lygmens protokolams pvz. HTTP. Duomenų SOAP žinutėmis vykstančio duomenų apsikeitimo paprastumo, paslaugos yra lengviau suderinamos, o tuo pačiu pakartotinai panaudojamos. SOAP žinutės pagrinde skirtos dokumentais paremtam bendravimui, verčia paslaugas projektuoti atsisakant paslaugos būsenos. Žiniatinklio paslaugų taikymo paslaugomis grįštoje architektūroje įvertinimo santrauka pateikta 6 lentelėje.

6 lentelė

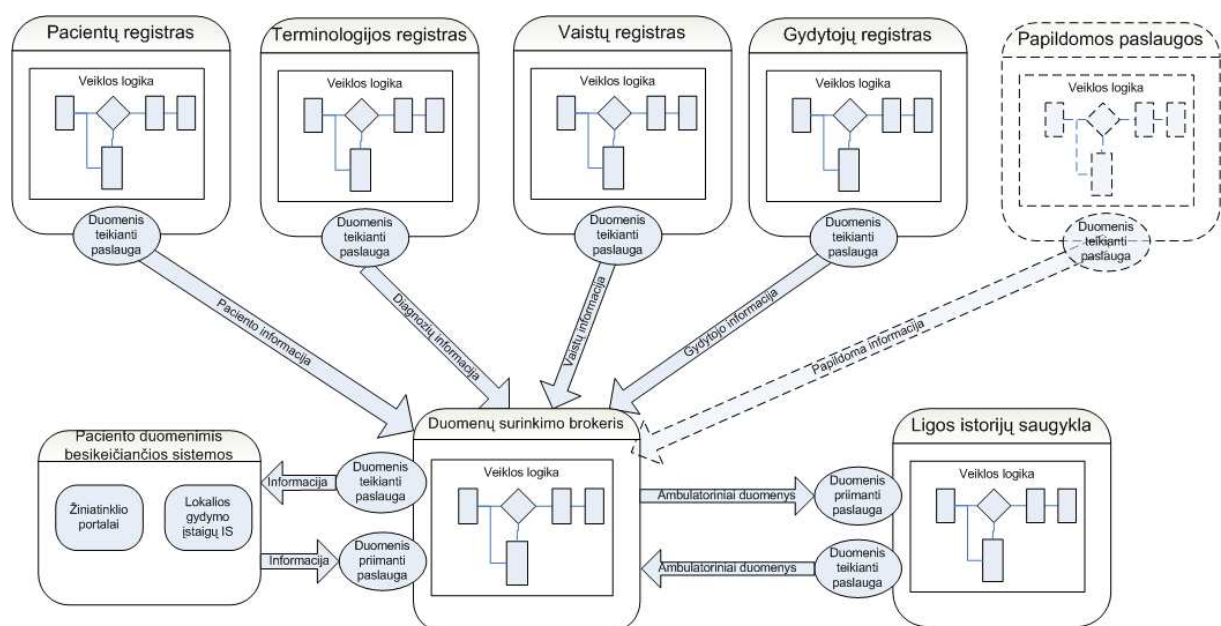
Žiniatinklio paslaugų technologijos tinkamumo paslaugų architektūros techninei realizacijai įvertinimas

Paslaugomis grįstos architektūros principas	Principą realizuojanti žiniatinklio paslaugų technologija	Technologijos tinkamumas
Silpnas sąryšis tarp paslaugų	WSDL, XML, UDDI	Labai tinka
Standartizuota sąsaja	WSDL	Tinka
Paslaugų autonomiškumas	Nerealizuotas. Priklauso nuo paslaugos suprojektavimo.	Neįvertintas
Paslaugos lygmens abstrakcija.	WSDL	Labai tinka
Pakartotinis panaudojamumas.	WSDL, UDDI, SOAP	Labai tinka
Paslaugų suderinamumas.	WSDL, XML, SOAP	Tinka
Būsenos atsisakymas.	SOAP	Tinka

4 MIŠRUS, PASLAUGOMIS GRĮSTAS DUOMENŲ APSIKEITIMO MODELIS

Įvertinus 2.3 skyriuje pateiktų modelių privalumus ir trūkumus, buvo nuspręsta pasirinkti mišrų centralizuotos ir decentralizuotos sistemos modelį. Paskirstytos sistemos modelio elementų atsisakyta dėl sudėtingos modelio realizacijos, priežiūros, plečiamumo ir valdomumo. Dėl didelių duomenų analizės poreikių, paciento duomenis nuspręsta dubliuotai kaupti centralizuotai. Įvertinus įeinančius pirminio sveikatos priežiūros centro duomenų srautus, nuspręsta į modelį įtraukti pagrindines priminiam sveikatos priežiūros centrui duomenis teikiančias įstaigas. Šis sprendimas leidžia deleguoti didelę dalį informacijos valdymo už ją tiesiogiai atsakingoms įstaigoms. Atsisakius dubliuojamo funkcionalumo turėtų smarkiai sumažėti sistemos kūrimas, bei supaprastėti jos priežiūra. Posistemių bendravimui pasirinktas brokerio modelis. Lygiaverčio (angl. *peer-to-peer*) bendravimo modelio atsisakyta, kadangi toks modelis yra sunkiai kontroliuojamas. Jis paliktas kaip alternatyvus įstaigoms, turinčioms poreikį tarpusavyje keistis specifine informacija.

Brokeris yra tarpinė programinė įranga (angl. *Middleware Application*) atsakinga už paciento informacijos pateikimą ir priėmimą, teikiant atitinkamas duomenų pateikimo ir priėmimo paslaugas. Brokeris taip pat atsakingas už vartotojų autentifikavimą ir autorizavimą, naudodamas savo arba trečių šalių saugumą užtikrinančią programine įrangą. Su brokeriu bendraujama tiesiogiai SOAP žinutėmis arba per žiniatinklio sąsają. Prie paciento saugomų ambulatorinių duomenų paslaugos gali prieiti tik duomenų surinkimo brokeris.



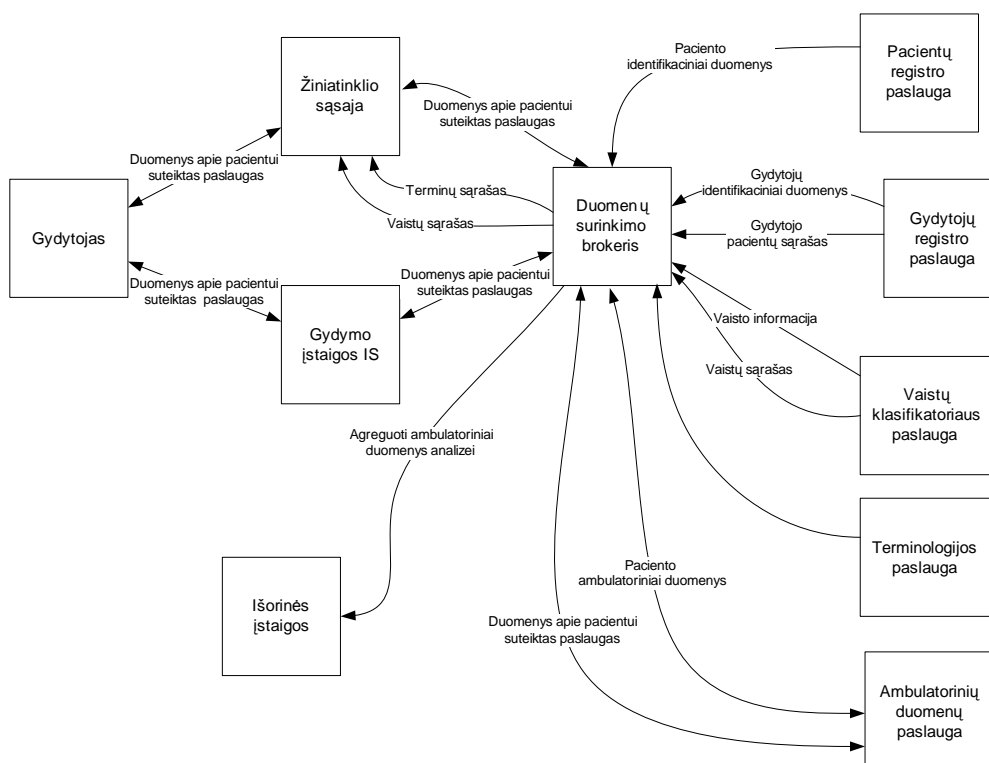
17 pav. Konceptinė mišraus apsikeitimo modelio schema

Modelio privalumai

- Sistemos lankstumas- kiekvieną posistemę galima patikėti skirtingoms įstaigoms, kurios pagal savo specializuotus poreikius gali vystyti posistemę, svarbu išlaikyti duomenų teikimo paslaugų sąsają.
- Universalumas – Paciento duomenų surinkimo brokeryje, gali būti realizuotos skirtingiems poreikiams teikiamų paslaugų sąsajos, pvz. agreguotų duomenų apie sergamumą teikimas statistikos departamentui.
 - – atskiros posistemės gali būti pakartotinai panaudotos kitiems tikslams, pavyzdžiui vaistų registro paslaugos gali būti teikiamos vaistinėms.
- Plečiamumas – pakankamai paprasta integruoti kitas posistemas. Tereikia kad nauja posistemė turėtų duomenis teikiančios/gaunančios paslaugos sąsają, ir realizuoti posistemės panaudojimą paciento duomenų surinkimo brokeryje.

Modelio trūkumai

- Organizacinės problemos – Sudėtingas duomenis teikiančių paslaugų keitimo derinimo procesas.
- Didelis informacijos srautas tenka paciento duomenų surinkimo brokeriui.



18 pav. Paslaugomis grįsto modelio duomenų srautų diagrama

5 PASLAUGOMIS GRĮSTO MODELIO REALIZACIJA

5.1 Tikslas

Realizacijos tikslas- pagal 4 skyriuje aprašytą modelį sukurti, paslaugomis grįstos apsiikeitimo paciento duomenimis prototipą. Teorinio modelio nagrinėjimas nėra pilnavertis, kadangi įvertinami ne visos įgyvendinimo grėsmės ir sunkumai. Prototipo realizacija bus šaltinis paslaugomis grįstos architektūros tinkamumo tyrimui. Kadangi darbe pagrindinis dėmesys skiriamas į paslaugas orientuotos architektūros taikymui, realizuojant modelį didžiausias dėmesys bus skiriamas posistemių integravimui. Realizuojant posistemas bus realizuoti tik pagrindiniai, sveikatos priežiūros sektoriuje, dažniausiai pasitaikantys panaudojimo atvejai. Dideli reikalavimai nefunkciniams greičio ir dizaino reikalavimams nėra keliami.

5.2 Sistemos ribos

Sistema bus sudaryta iš šių posistemių:

- **„ESIS portalas“**- Žiniatinklio portalas, skirtas vartotojo sąsajai su gydytoju realizuoti.
- **Duomenų surinkimo brokeris** – Žiniatinklio paslauga, skirta paskirtų duomenų surinkimui.
- **Pacientų registras** – Posistemė skirta pacientų byloms administruoti. Posistemėje realizuota žiniatinklio paslauga skirta teikti paciento informaciją išorinėms sistemoms.
- **Gydytojų registras** - Posistemė skirta gydytojų byloms administruoti. Posistemėje realizuota žiniatinklio paslauga skirta teikti gydytojo informaciją išorinėms sistemoms.
- **Vaistų registras** - Posistemė skirta vaistų registrui ATC klasifikatoriui administruoti. Posistemėje realizuota žiniatinklio paslauga, teikianti informaciją apie vaistus išorinėms sistemoms. Perspektyvoje sistema turėtų teikti vaistų kodų konvertavimo tarp skirtingų sistemų ir vaistų prieinamumo vaistinėse informacijos paslaugas
- **Terminologijos registras** – Posistemė skirta diagnozių terminologijos registrui administruoti. Posistemėje realizuota žiniatinklio paslauga, teikianti informaciją apie vaistus išorinėms sistemoms. Perspektyvoje sistema turėtų teikti diagnozių kodų konvertavimo tarp skirtingų sistemų, bei detalios informacijos apie diagnozės simptomus informacijos paslaugas.
- **Ligos istorijų saugykla** – Ši posistemė skirta pacientų ligos istorijų ir informacijos apie pacientams suteiktas paslaugas saugojimui. Posistemė teikia informacijos

priėmimo ir informacijos pateikimo žiniatinklio paslaugas. Perspektyvoje ši posistemė teiks agreguotą informaciją statistikos ir atskaitomybės ataskaitoms, bei analitinę informaciją medicinos mokslo įstaigoms.

5.3 Priemonės

5.3.1 Priemonių rinkimosi kriterijai

Pagrindiniai kriterijus renkantis prototipo realizavimui reikalingas technologijas:

- Priemonių efektyvumas kūrimo stadijoje
- Sukurtos sistemos plečiamumas
- Sukurtos sistemos priežiūra
- Sukurtos sistemos administravimas

5.3.2 Sistemai kurti panaudotos technologijos ir jų pagrindimas

7 lentelė

Sistemai kurti panaudotos technologijos

Priemonė	Technologija
Vartotojo sąsaja	“ASP.NET 2” žiniatinklio sąsaja
Žiniatinklio paslaugos	“ASP.NET 2 Web services”
Programavimo kalba	“C#”, “ASP.NET”
Duomenų priėjimas	“ADO.NET”
Programų kūrimo aplinka	“Microsoft Visual Studio 2005 Beta 2”
DBVS	“Microsoft SQL Server 2005 Express edition”
Modeliavimo įrankis	“Microsoft Visio 2005 for Enterprise architects” “Microsoft Visual Studio 2005 Beta 2”

5.3.2.1 Vartotojo sąsaja

Vartotojo sąsajai pasirinkta realizuoti pasirinkta žiniatinklio taikomoji programa. Žiniatinklio taikomosios programos yra žymiai pranašesnės už kompiuteryje įdiegtas programas diegimo, priežiūros ir suderinamumo atžvilgiu. Žiniatinklio taikomoji programa realizuota “ASP.NET 2” technologija. “ASP.NET 2” “Code Behind” technologija leidžia atskirti programinį kodą nuo vartotojo sąsajos kodo. Taip pat “Microsoft Visual Studio 2005” programų kūrimo aplinka jau turi iš anksto paruoštus vartotojos sąsajos elementus ir suteikia ASP greitojo

programų kūrimo (angl. *Rapid Application Development*) galimybes. Programuotojui nebereikia rūpintis, kaip realizuojami vartotojo sąsajos elementai, todėl didesnę dėmesį galima sutelkti į realizuojamus procesus.

5.3.2.2 Žiniatinklio paslaugos

Žiniatinklio paslaugos kuriamos “ASP.NET 2 Web services” technologija. ASP leidžia žiniatinklio paslaugas aprašyti kaip klases, o jų teikiamas funkcijas kaip klasių metodus, o vėliau iš jų generuoti WSDL failą. Taip žiniatinklio paslaugų kūrimas ypač supaprastinamas.

5.3.2.3 Duomenų bazių valdymo sistema

Kaip DBVS savo projektui pasirinkome “Microsoft SQL Server 2005 Express edition”. Tai nemokama DBVS versija, turinti 2GB duomenų bazės dydžio apribojimą. Modelio bandymams tokio dydžio duomenų bazės pakanka, o pagal poreikį šią DBVS versiją lengvai galima pakeisti galingesnėmis mokamomis “Microsoft SQL Server 2005” versijomis.

5.3.2.4 Bendravimas su DBVS

Bendravimui su DBVS pasirinkta “ADO.NET” technologija dėl supaprastinto objektinio duomenų apskaitimo modelio, bei didelio suderinamumo su ASP.NET programavimo kalba.

5.3.2.5 Programų kūrimo aplinka

Projektas buvo kuriamas “Microsoft Visual Studio 2005 Beta 2” aplinkoje. Ši kūrimo aplinka turi smarkiai programuotojo efektyvumą didinančius įrankius: sąsaja su DBVS, automatinio testavimo įrankiai, kodo refaktoringo įrankis, gereito programų kūrimo komponentai.

5.3.2.6 Modeliavimo įrankis

Panaudojimo atvejų, sekų ir esybių ryšių diagramų modeliavimui. Buvo naudojamas “Microsoft Visio 2005 for Enterprise architects” modeliavimo įrankis. Šio įrankio pagalba iš esybių ryšių diagramų buvo generuojamos duomenų bazės.

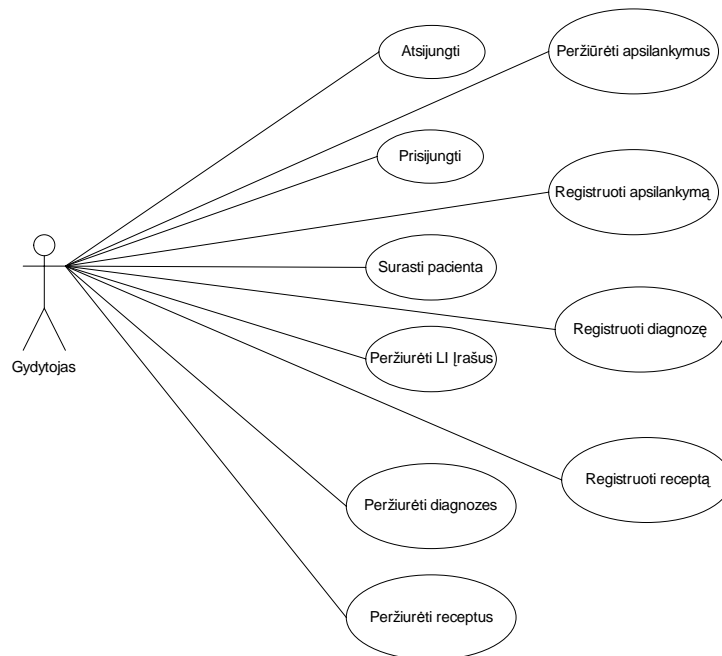
Paskirstytos sistemos ir klasių diagramų modeliavimui buvo naudojama “Microsoft Visual Studio 2005 Beta 2” aplinka. Iš paskirstytos sistemos diagramos galima generuoti posistemes realizuojančius projektus. O klasių modeliavimo įrankis turi tiesioginį ryšį su klase realizuojančiu kodu.

5.4 Posistemių funkcionalumas

Posistemių funkcijos pavaizduotos panaudojimo atvejų diagramomis. Svarbiausi duomenų surinkimo brokerio bendradarbiavimo su kitomis sistemomis elementai pateikti sekų diagramomis.

5.4.1 ESIS portalas

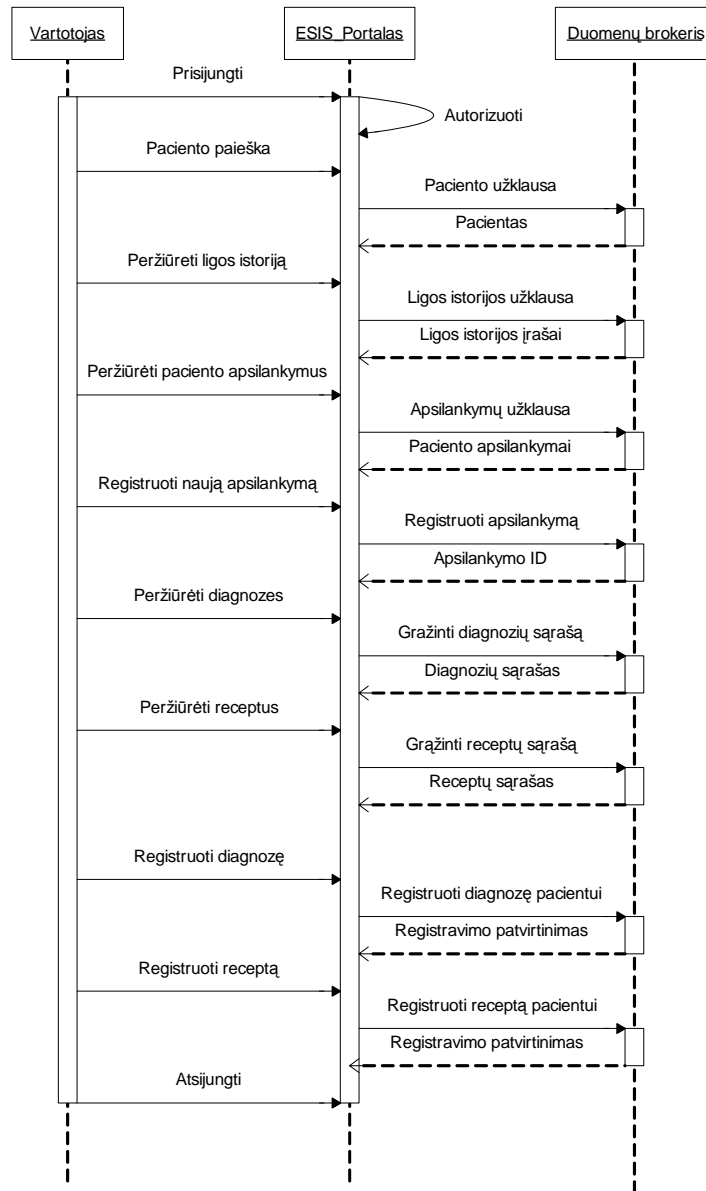
Elektroninės Sveikatos Informacinės Sistemos “ESIS” portalas skirtas gydytojams keistis paciento ambulatorine informacija. Posistemiėje realizuoti pagrindinės gydytojui reikalingos funkcijos. Realizuotas gydytojų autentifikavimas ir autorizavimas. Posistemiės panaudojimo atvejų diagrama pateikta 19 pav.



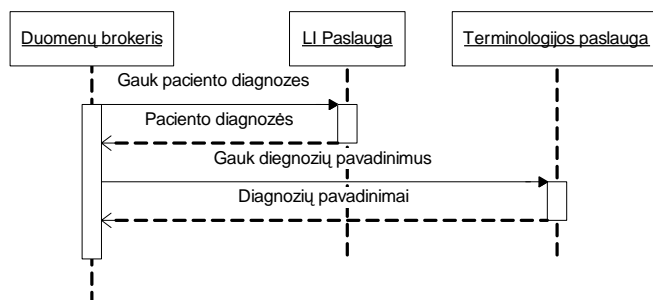
19 pav. ESIS portalų panaudojimo atvejų diagrama

5.4.2 Duomenų surinkimo brokeris

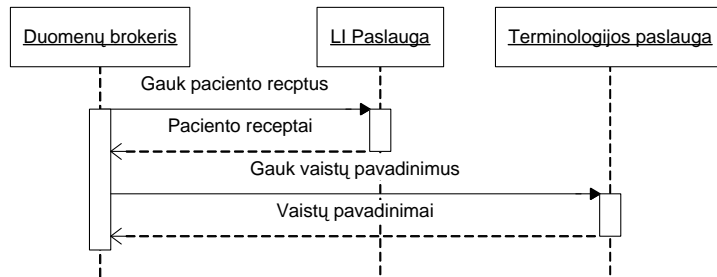
Tai tarpinė programinė įranga (angl *Middleware application*), skirta aptarnauti informaciją apie pacientą besikeičiančioms sistemoms. Duomenų surinkimo brokerio ir kitų sistemų bendradarbiavimas pavaizduotas sekų diagramomis (20, 21, 22 pav.).



20 pav. ESIS portalų funkcijų sekų diagrama



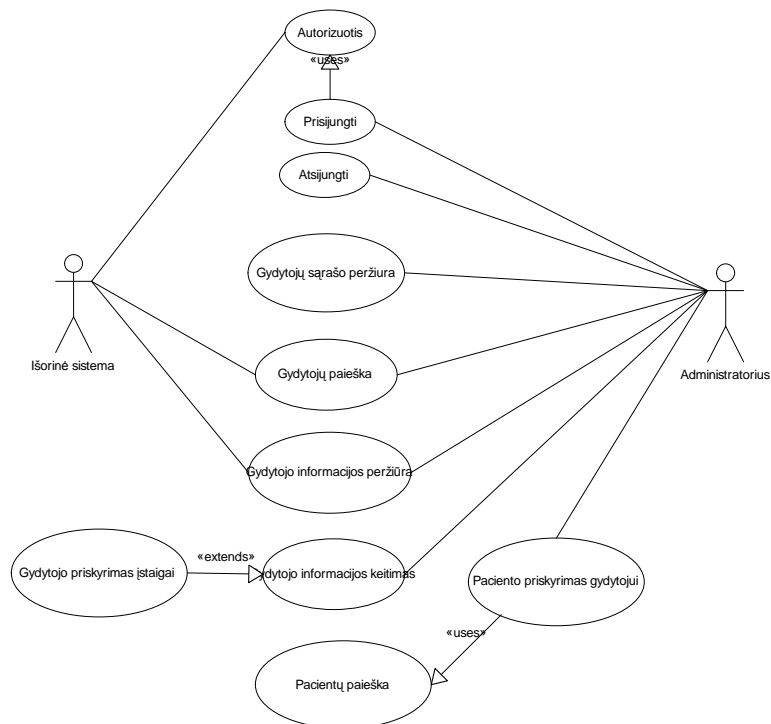
21 pav. Diagnozių pateikimo sekų diagrama



22 pav. receptų pateikimo sekų diagrama

5.4.3 Gydytojų registras

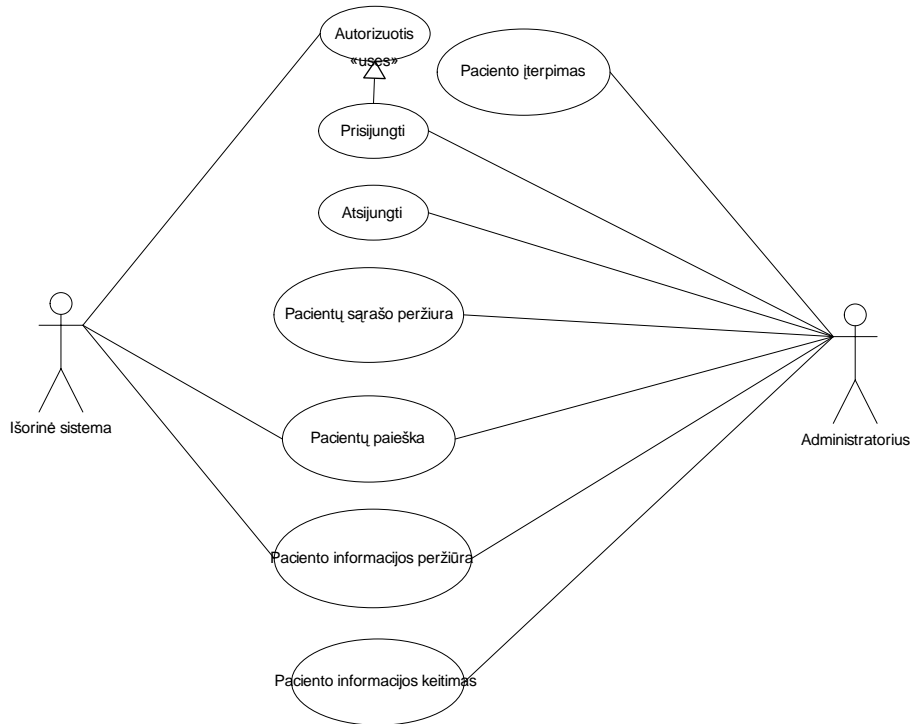
Gydytojų registras skirtas informacijai apie gydytojus administruoti, bei teikti informaciją apie gydytojus išorinėms sistemoms. Posistemės panaudojimo atvejų diagrama pateikta 23 pav.



23 pav. Gydytojų registro posistemės panaudojimo atvejų diagrama

5.4.4 Pacientų registras

Posistemė skirta informacijai apie pacientus administruoti, bei teikti pacientų informaciją išorinėms sistemoms. Posistemės panaudojimo atvejų diagrama pateikta 24 pav.



24 pav. Pacientų registro posistemės panaudojimo atvejų diagrama

5.4.5 Terminologijos registras

Terminologijos registras skirtas diagnozių kodavimui ir pavadinimams administruoti. Prototipe realizuotas tik “TLK-10” klasifikatoriaus informacijos administravimas, bei teikimas išorinėms sistemoms.

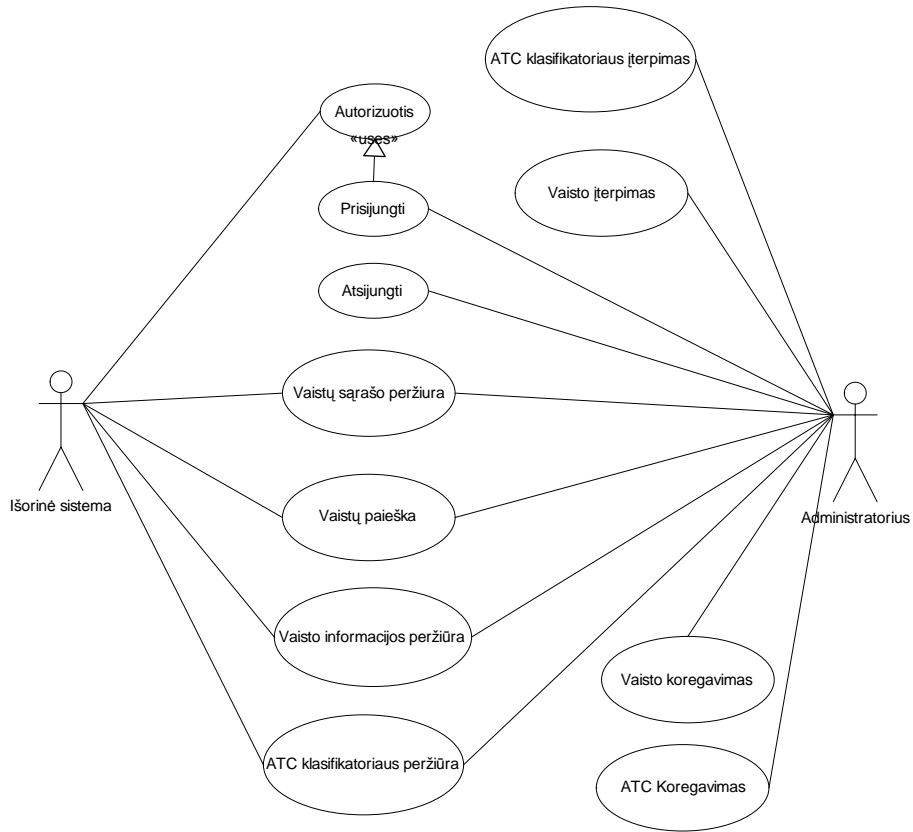
Posistemės panaudojimo atvejų diagrama pateikta 25 pav.



25 pav. Diagnostikų terminologijos registro posistemės panaudojimo atvejų diagrama

5.4.6 Vaistų registras

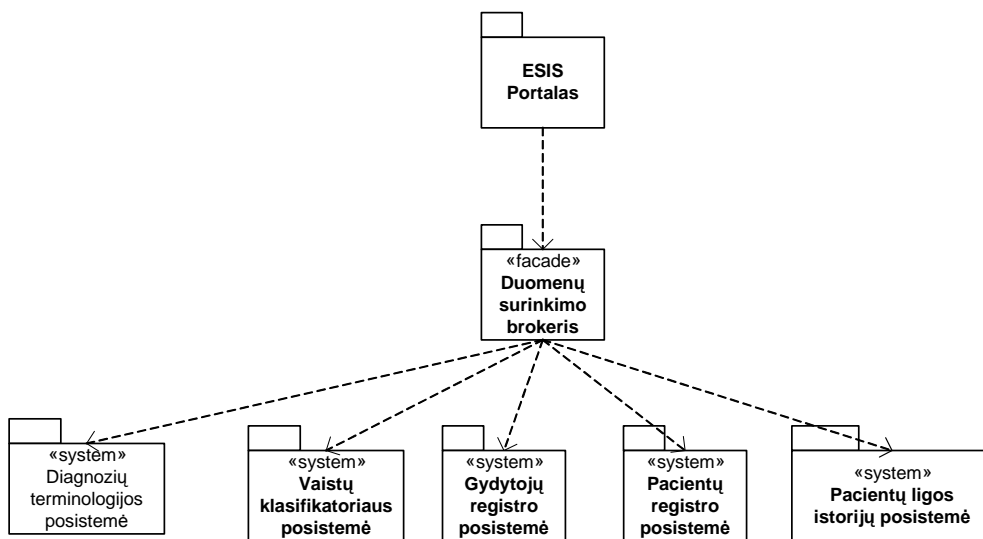
Vaistų registro posistemė skirta vaistų sąrašui, bei jo kodavimui ir klasifikatoriams administruoti. Išorinėms sistemoms, posistemė teikia vaistų, klasifikatoriaus paieškos paslaugas. Posistemės panaudojimo atvejų diagrama pateikta 26 pav.



26 pav. Vaistų registro posistemės panaudojimo atvejų diagrama

5.5 Posistemių loginis vaizdas

Prototipo posistemės suskirstytos į paketus. Posistemių paketų diagrama pateikta 27 paveiksle.



27 pav. Paketų diagrama

Kiekviena posistemė detalizuojama trimis sluoksniais:

- **Duomenų priėjimo.** Sluoksnis skirtas duomenų valdymą realizuojančioms klasėms.
- **Veiklos logikos.** Sluoksnis skirtas duomenų apdorojimą realizuojančioms klasėms.
- **Prezentacijos.** Sluoksnis skirtas informacijos pateikimą vartotojui ar išorinėms sistemoms realizuojančioms klasėms.

Toks funkcionalumo išskaidymas į sluoksnius padidina kodo skaitomumą ir palengvina sistemų, kadangi kiekvienas sluoksnis realizuoja skirtingą abstrakcijos lygmenį. Posistemių sluoksnių paketų diagrama pateikta 28 paveiksle.



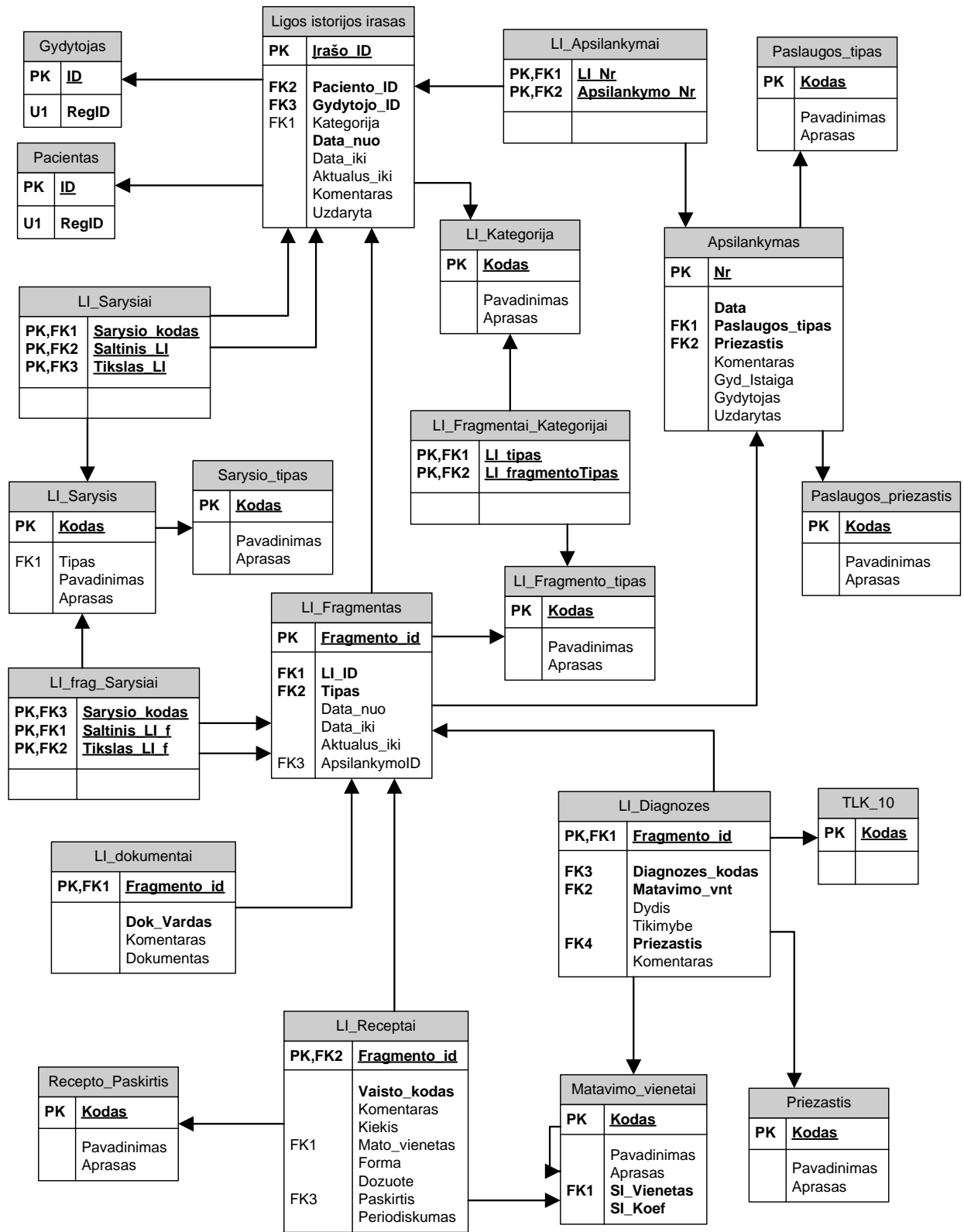
28 pav. Posistemės realizavimo sluoksnių paketų diagrama

5.6 Posistemių duomenų vaizdas

Šiame skyrelyje pateikiami posistemėms realizuoti reikalingų duomenų struktūrų aprašymai. Pagal pateiktas diagramas, automatiškai sugeneruotos duomenų bazės.

5.6.1 Pacientų ligos istorijų posistemė

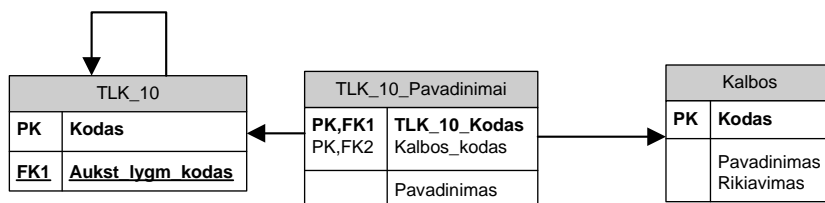
29 pav. pateikta supaprastinta pacientų ligos istorijoms ir informacijai pacientui suteiktas paslaugas sauguti reikalingos duomenų struktūros ER diagrama. Detali pacientų ligos istorijos posistemės ER diagrama pateikta 1 priede.



29 pav. Pacientų ligos istorijų saugyklos duomenų bazės schema

5.6.2 Diagnozių terminologijos posistemė

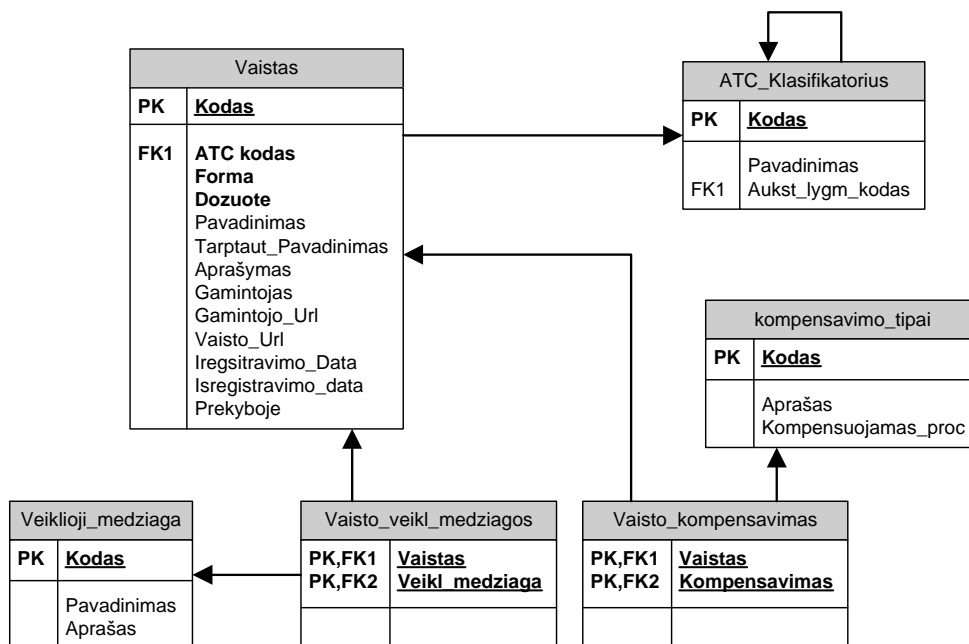
Kadangi nuspręsta kad prototipe terminologijos paslauga teiks tik “TLK-10” klasifikatoriuje koduojamus terminus, duomenų struktūra yra pritaikyta tik “TLK-10” klasifikatoriuje esantiems terminams administruoti. Duomenų struktūros ER diagrama pateikta 30 pav.



30 pav. Diagnozių terminologijos duomenų bazės schema

5.6.3 Vaistų registro posistemė

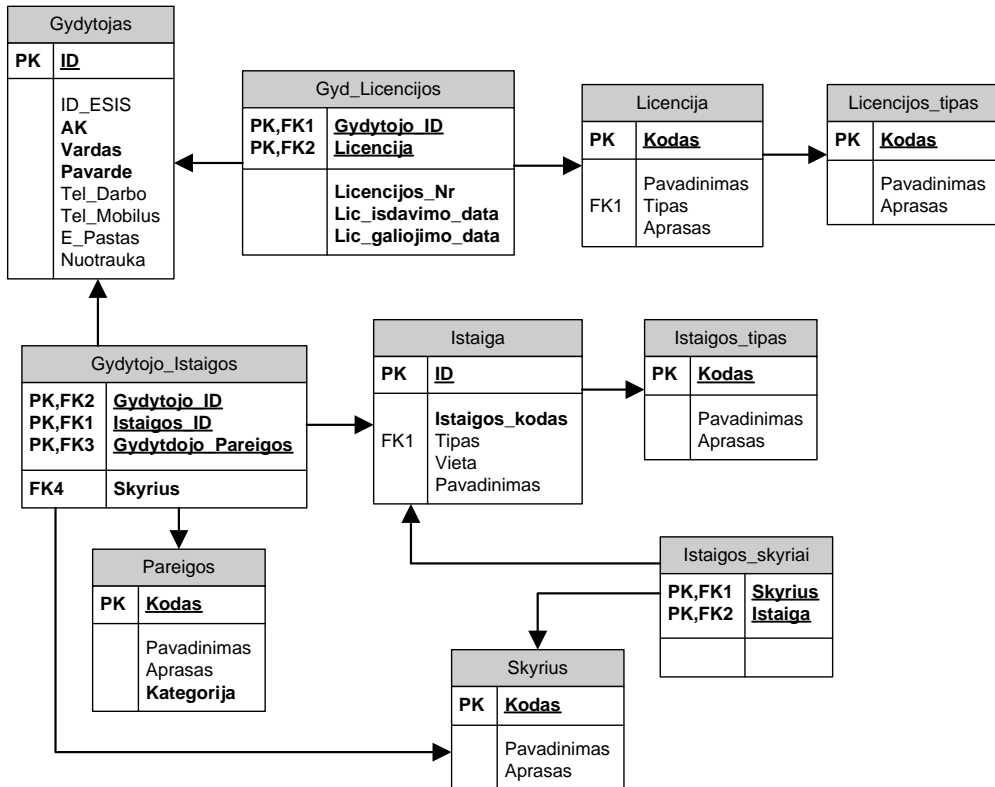
Duomenų struktūra realizuoja pagrindinės informacijos apie vaistą saugojimui ir kodavimui pagal ATC klasifikatorių saugojimą. Duomenų struktūros ER diagrama pateikta 31 pav.



31 pav. Vaistų registro duomenų bazės schema

5.6.4 Gydytojų registro posistemė

Duomenų struktūra realizuoja pagrindinės informacijos apie gydytojus, jų pareigas ir turimas licencijas saugojimą. Duomenų struktūros ER diagrama pateikta 32 pav.

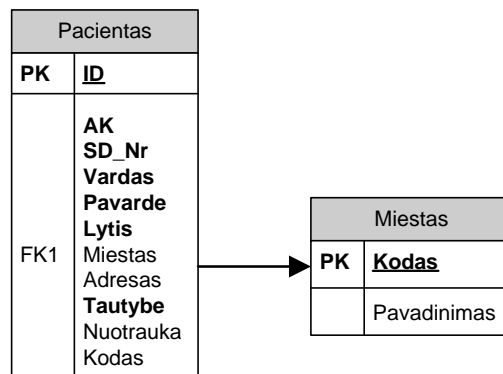


32 pav. Gydytojų registro duomenų bazės schema

5.6.5 Pacientų registro posistemė

Duomenų struktūra skirta elementarios informacijos apie pacientui saugojimui realizuoti.

Duomenų struktūros ER diagrama pateikta 33 pav.

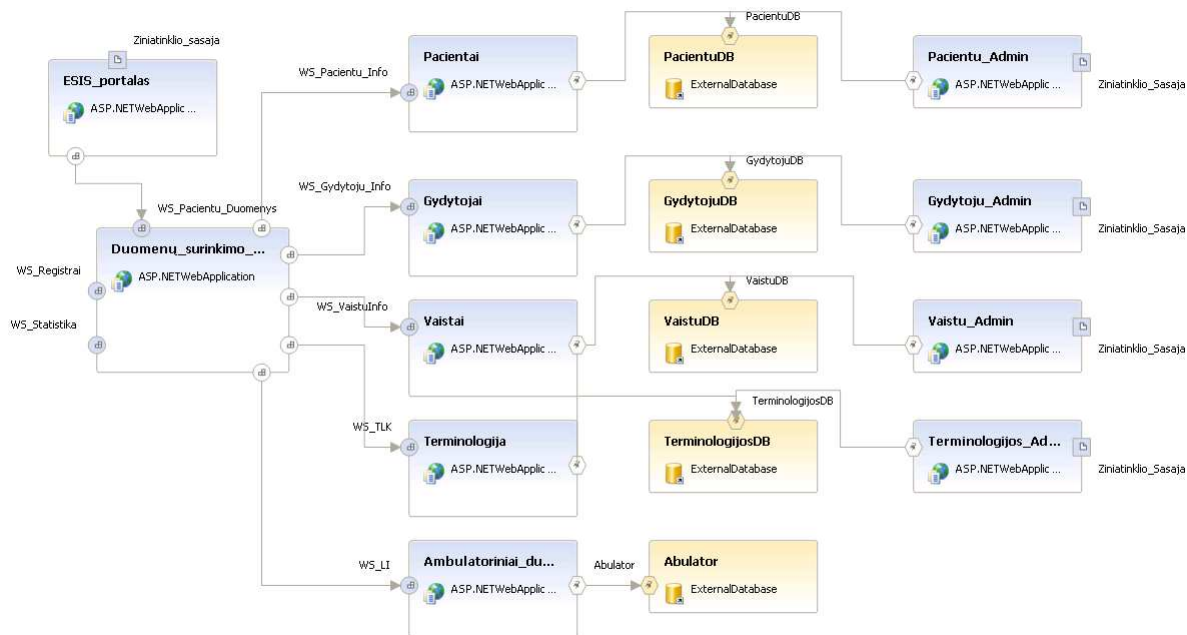


33 pav. Pacientų registro duomenų bazės schema

5.7 Posistemių programinė įranga

34 schemoje pateikta posistemes realizuojančios programinės įrangos sąryšių diagrama.

Diagramos notacijos paaiškinimai pateikti 3 priede. Pagal šią diagramą buvo automatiškai sugeneruoti programinės įrangos kūrimui reikalingi projektai ir pirminė jų konfigūracija.



34 pav. Programinės įrangos sąryšių diagrama

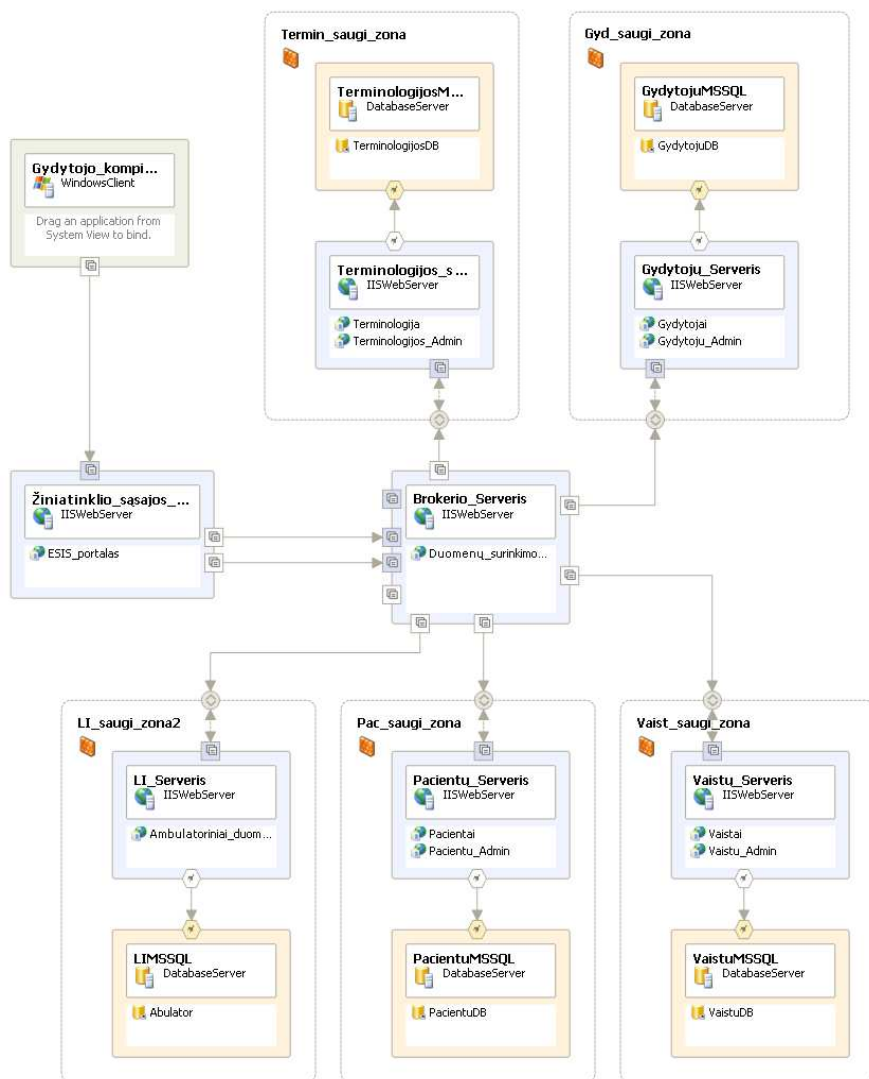
8 lentelė

Programinės įrangos paaiškinimai

Pavadinimas diagramoje	Programinės įrangos aprašymas
Pacientu_Admin	Žiniatinklio portalas skirtas pacientų sąrašo ir jų duomenų administravimui.
Gydytoju_Admin	Žiniatinklio portalas skirtas gydytojų sąrašo ir jų duomenų administravimui.
Vaistu_Admin	Žiniatinklio portalas skirtas vaistų klasifikatoriui administruoti.
Terminologijos_Admin	Žiniatinklio portalas skirtas medicininiais terminams administruoti.
PacientuDB	DBVS skirta pacientų sąrašui ir pagrindiniams duomenims saugoti.
GydytojuDB	DBVS skirta gydytojų sąrašui ir pagrindiniams duomenims saugoti.
VaistuDB	DBVS skirta vaistų sąrašui ir klasifikatoriui saugoti.
TerminologijosDB	DBVS skirta medicininiais terminams ir jų klasifikatoriui saugoti.
Pacientai	Žiniatinklio paslauga skirta pacientų paieškai ir informacijai apie pacientus teikti.
Gydytojai	Žiniatinklio paslauga skirta informacijai apie gydytojus teikti.
Vaistai	Žiniatinklio paslauga skirta vaistų paieškai, klasifikatoriaus sąrašui ir informacijai apie vaistus teikti.
Terminologija	Žiniatinklio paslauga skirta terminų paieškai, klasifikatoriaus sąrašui pateikti.
Ambulatoriniai_duom	Žiniatinklio paslauga paciento ambulatorinės informacijos priėmimui, pateikimui.
Duomenų_surinkimo_b	Tarpinė programinė įranga, skirta teikti reikiamas informacines paslaugas apie pacientą ir jo ligos istoriją.
ESIS portalas	Žiniatinklio portalas skirtas gydytojui peržiūrėti bei papildyti elektroninę paciento ligos istoriją.

5.8 Posistemių išdėstymo vaizdas

35 schemoje pavaizduotas posistemes realizuojančios programinės įrangos fizinis išdėstymas. Schemos notacijos paaiškinimai pateikti 3 priede. Posistemių programinę įrangą talpinantys serveriai apriboti saugiomis zonomis. Tai reiškia kad tarp šios techninės įrangos, turi būti užtikrintas saugus ryšys. Realizuojant prototipą, visa programinė įranga buvo patalpinta viename serveryje, tačiau projektuota, jog esant poreikiui, programinė įranga galėtų būti perkelta į kitus serverius.



35 pav. Posistemių išdėstymo diagrama

6 PASLAUGŲ ARCHITEKTŪRA GRĪSTO PROTOTIPO EFEKTYVUMO TYRIMAS

6.1 Prototipo kokybės tyrimas

Programinės įrangos įvertinimas buvo remiantis ISO/IEC 9126 standarte pateikiamu kokybės vertinimo modeliu. Modelio aprašymas pateiktas 2riede Standarte pateiktoms kokybės charakteristikoms buvo pasirinkta grupė subcharakteristikų iš rekomenduojamų subcharakteristikų sąrašo.

Pateikiama kokybės vertinimo lentelė su subcharakteristikų bendrais įvertinimais:

9 lentelė

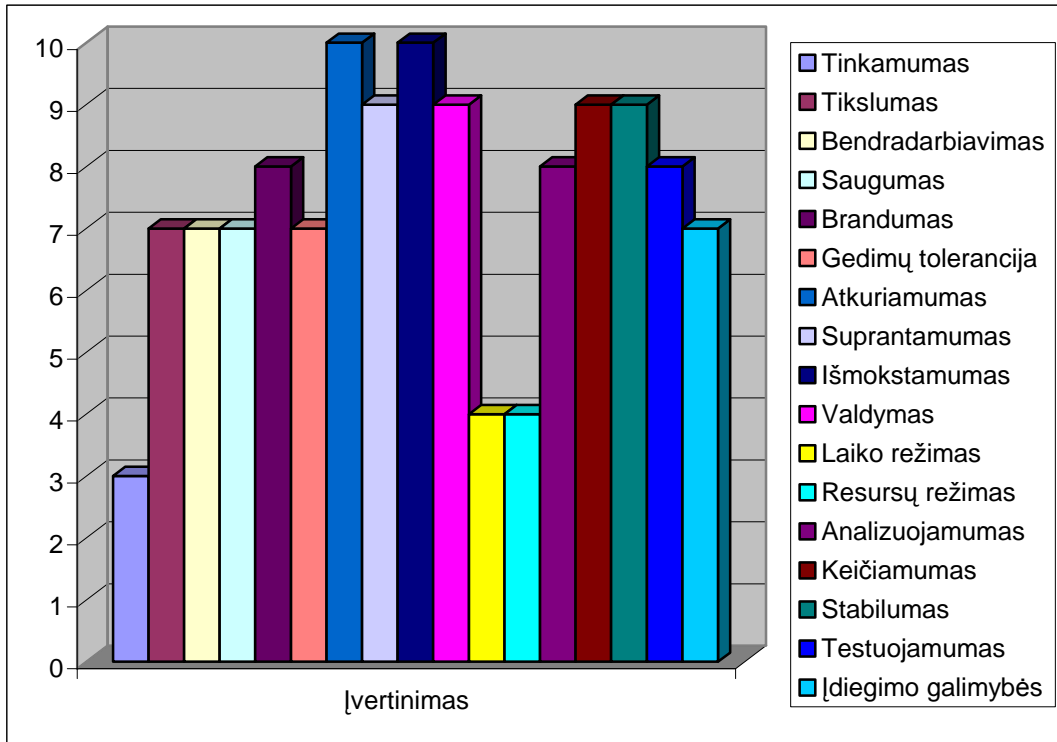
Prototipo kokybės įvertinimas

Subcharakteristika	Įvertinimas	Paaiškinimas
Tinkamumas	3	Produktas yra prototipo pobūdžio. Produkto teikiamos funkcijos (operacijos) nepilnai pateikia funkcionalumą, kuris yra reikalaujamas reikalavimų specifikacijos dokumente.
Tikslumas	7	Produkto pateikiami rezultatai ne taip detalčiai atitinka rezultatus, kurie yra reikalaujami pagal reikalavimų specifikacijos dokumentą.
Bendradarbiavimas	7	Produkto sistemų (bei posistemų) bendradarbiavimas nėra pilnai ištestuotas.

Saugumas	7	<p>Duomenys perduodami šifruotu kanalu.</p> <p>Duomenų bazės prisijungimas taip pat apsaugomas slaptažodžiu.</p> <p>Prisijungimas prie žiniatinklio tarnybų apsaugotas SOAP antraštėje perduodamu slaptažodžiu.</p> <p>Prisijungimas prie žiniatinklio sąsajos apsaugotas slaptažodžiu ir naudoja. Vartotojų autentifikavimas realizuotas "ASP forms authentication" technologija.</p> <p>standartinį ASP rolėmis pagystą saugumo modelį.</p> <p>Atviri žiniatinklio paslaugų komunikavimo standartai padidina mėginimo įsilaužti riziką.</p>
Brandumas	8	<p>Posistemių struktūra yra pakankamai sudėtinga, dėl to gedimas vienoje posistemėje gali įtakoti kelių posistemių nesėkmingą veikimą. Tačiau kitos posistemės įtakojamos tik tada, kai gedimas įtakoja posistemės teikiamas paslaugas.</p>
Gedimų tolerancija	7	<p>Yra galimi atvejai, kas sistemos komponento gedimas įtakos kitų sistemos komponentų veikimą. Sistema gali neteikti su sugedusiu komponentu susijusių funkcijų, tačiau sistema ir toliau veiks.</p>
Atkuriamumas	10	<p>Tyrimo metu nebuvo nustatyti programinės įrangos nekorektiško veikimo atvejai, kurių neigiama įtaka programinės įrangos funkcionalumui bei duomenų vientisumui nebūtų atstatyta.</p> <p>Esant poreikiui duomenys „Microsoft SQL Server 2005“ duomenų bazėje gali būti atstatyti iki reikiamo laiko momento.</p>

Suprantamumas	9	Programos kodas nėra sudėtingai skaitomas dėl naudojamo standartinio komponentų (klasių, atributų, metodų, kintamųjų) vardų suteikimo principo. Standartinių ASP komponentų panaudojimas leido smarkiai sumažinti programos kodo skaičių, todėl reikia susipažinti tik su veiklos logika susijusia informacija. Tačiau yra daug sąsajų tarp programos komponentų ir tai reikalauja daugiau vartotojo pastangų norint atpažinti bendrą loginį sistemos kontekstą.
Išmokstamumas	10	Programinė įranga pateikia intuityvią ir lengvai perprantamą vartotojo sąsają. Sistemos funkcionalumas nėra sudėtingas
Valdymas	9	Programinė įranga nereikalauja didelių vartotojo pastangų vykdant arba valdant pateikiamas operacijas.
Laiko režimas	4	Duomenų išgavimo iš centrinės duomenų saugyklos mechanizme yra naudojamos serverio pusėje vykdomos procedūros ir klientui gražinami duomenys atrinkti (filtruoti) pagal tam tikrus kriterijus. Laiko režimą gali įtakoti serverio veikimo sparta bei serverio darbinis apkrovimas. Duomenimis keičiamasi XML formato dokumentais, todėl žymiai išauga perduodamos informacijos kiekis, o tuo pačiu ir perdavimo laikas.

Resursų režimas	4	<p>Duomenų išgavimo iš centrinės duomenų saugyklos mechanizme yra naudojamos serverio pusėje vykdomos procedūros ir klientui gražinami duomenys atrinkti (filtruoti) pagal tam tikrus kriterijus. XML dokumentų apdorojimui reikia papildomų atminties ir procesoriaus resursų. Dėl XML formatu perduodamos informacijos kiekio apkraunami tinklo resursai. ASP žiniatinklio serveris reikalauja daugiau resursų, nei kitos panašios technologijos.</p> <p>Toks mechanizmas nereikalauja spartaus ryšio su centrine saugykla resursų, tačiau gali pritrūkti pačio serverio resursų.</p>
Analizuojamumas	8	<p>Dėl didelio ryšių tarp sistemos komponentų kiekio, gali būti sudėtinga nustatyti gedimo vietą arba modifikuojamą programos bloką. Klasių suskaidymas į duomenų, veiklos logikos ir prezentacijos sluoksnius, padidina kodo suprantamumą.</p>
Keičiamumas	9	<p>Dėl didelio ryšių tarp sistemos komponentų kiekio yra sudėtingiau atlikti pakeitimus programinėje įrangoje, užtikrinant pakeitimų pilnumą ir korektiškumą.</p>
Stabilumas	9	<p>Modifikacijos gali įtakoti susijusių komponentų korektišką veikimą, tačiau ne visos sistemos funkcionavimą.</p>
Testuojamumas	8	<p>Didelis ryšių kiekis tarp programinės įrangos komponentų reikalauja daug pastangų atliekant integravimo testavimą. Formalus paslaugų apibrėžimas leidžia paslaugą laikyti juoda dėže ir efektyviai taikyti automatizuotus testus. Sistema kurta .NET aplinkoje, todėl sistemos testavimui gali būti panaudoti efektyvūs “Visual Studio 2005” automatizuoto testavimo įrankiai.</p>
[diegimo galimybės	7	<p>Pakanka sistemą nukopijuoti į kompiuterį su įdiegta „IIS 5.0“, „ASP.NET“ ir „.NET framework 2“ programine įranga. Klientas turi turėti su XHTML 4 standartą palaikančią naršyklę. Veikia tik Windows</p>



36 pav. Prototipo kokybės vertinimo rezultatai

6.2 Į žiniatinklio paslaugas ir serverio procedūras besikreipiančių interneto puslapių atsako laiko tyrimas

6.2.1 Tyrimo aprašymas

Šio tyrimo tikslas – įvertinti žiniatinklio paslaugų panaudojimo duomenų keitimuisi įtaką puslapių atsako laikui. Tyrime lyginami du interneto puslapiai, gaunantys tą pačią informaciją skirtingais būdais: užklausa žiniatinklio paslaugai ir užklausa DBVS serverio procedūrai. Užklauskos rezultatas- 5000 įrašų. Tiriama puslapiai turintys po 1, 5 ir 10 tokių užklauskų. Pagrindinis kriterijus užklauskos efektyvumui įvertinti yra atsako laikas, kadangi didžioji resursų dalis sunaudojama puslapio formatavimui, išskirti sunaudojamą operatyvinės atminties ir procesoriaus resursų dalį, reikalingą užklauskų rezultatų apdorojimui yra sudėtinga.

Užklauskos generuojamos ir atsako laikas fiksuojamas “Microsoft Visual Studio 2005” apkrovos testavimo įrankiu.

6.2.2 Tyrimui naudojamų serverių konfigūracijos:

Žiniatinklio paslaugų ir DBVS serveris

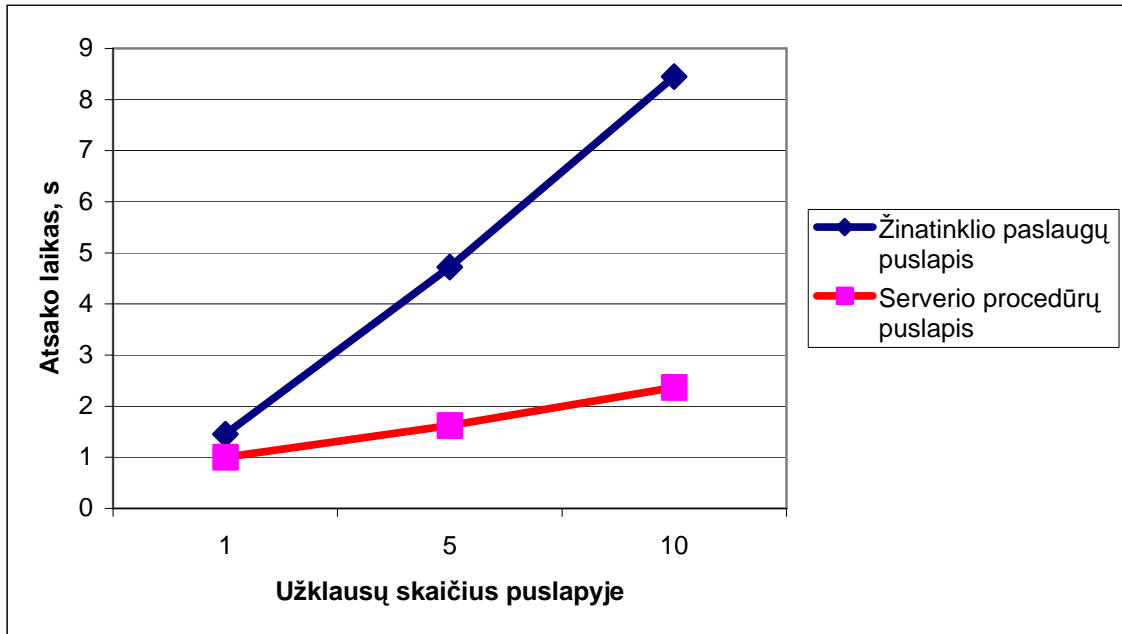
- Procesorius: Intel Celeron 1 Ghz
- Operatyvinė atmintis: 512 MB
- Kietasis diskas: 40GB 7200rpm
- Operacinė sistema: “Windows 2000 Profesional”
- DBVS: “Microsoft SQL Server 2005 Express Edition”
- Programinė įranga: “Microsoft IIS 5.0”, “Microsoft .NET 2.0 Framework”

Žiniatinklio puslapių serveris

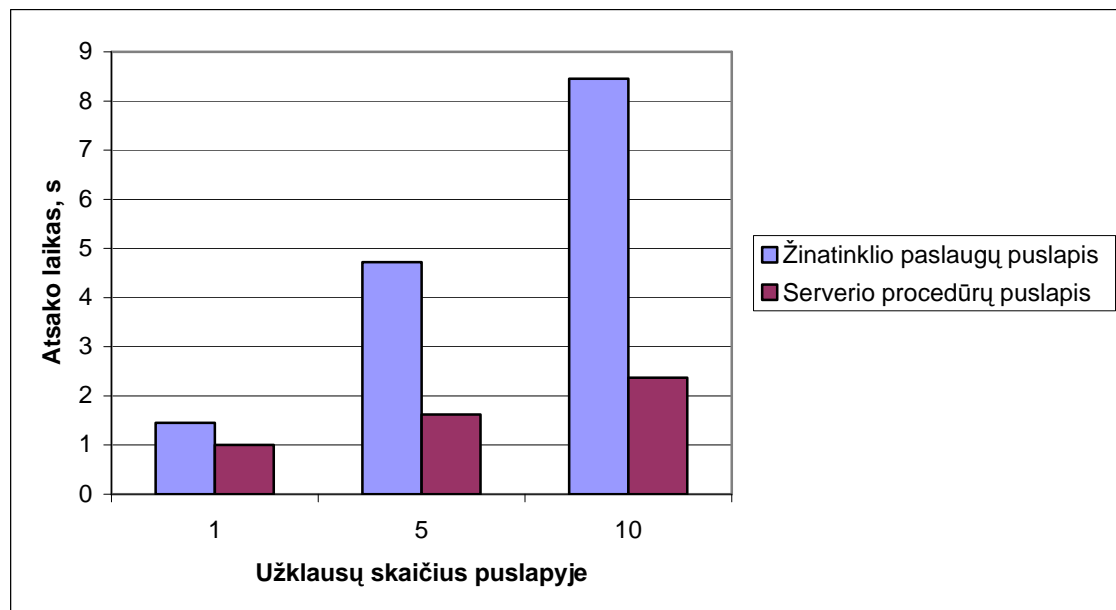
- Procesorius: AMD Athlon 1,8 GHz
- Operatyvinė atmintis: 704 MB
- Kietasis diskas: 40GB 5200rpm
- Operacinė sistema: “Windows 2000 Profesional”
- Programinė įranga: “Microsoft IIS 5.0”, “Microsoft .NET 2.0 Framework”

6.2.3 Tyrimo rezultatai

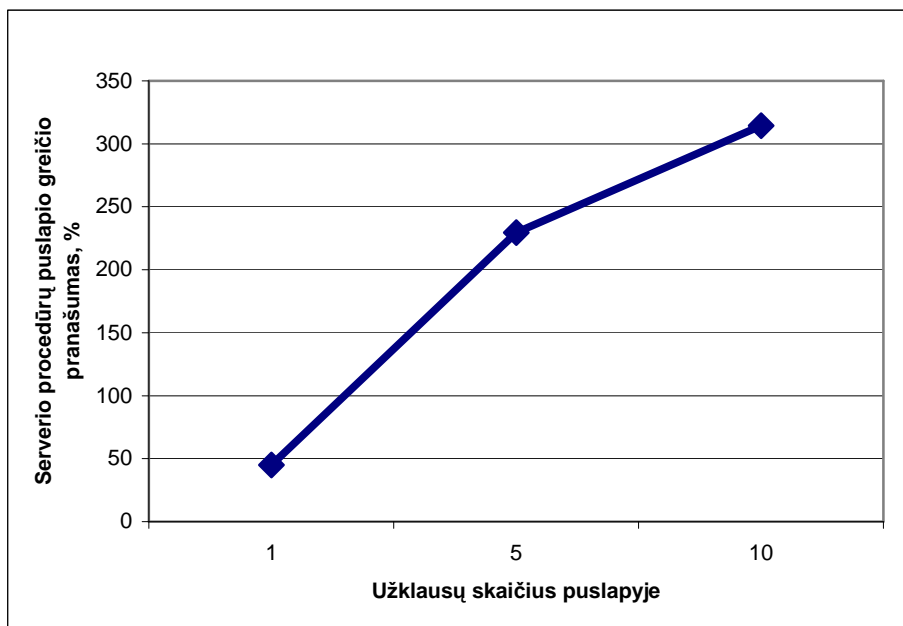
Pagal 4 priede pateiktus eksperimento rezultatus nustatytas vidutinis puslapių atsako laikas, pagal kurios įvertinta užklausų skaičiaus įtaka puslapio atsako laikui (37, 38 pav.). 39 pav. pavaizduotas tiesioginėmis užklausomis serverio procedūroms duomenis gaunančio puslapio, atsako greičio pranašumas prieš iš žiniatinklio paslaugų duomenis gaunančių puslapių.



37 pav. Atsako laiko priklausomybės nuo užklausų skaičiaus grafikas



38 pav. Atsako laikų palyginimo grafikas



39 pav. Atsako laikų santykinio palyginimo grafikas

6.3 Tyrimo rezultatų apibendrinimas

Pagal atlikto prototipo kokybės tyrimo rezultatus, silpniausios prototipo vietos yra prototipo tinkamumo, greičio ir resursų sunaudojimo charakteristikos. Kadangi prototipui tinkamumo reikalavimai nebuvo keliami, ši charakteristika nėra svarbi. Prastas greičio ir resursų sunaudojimo charakteristikas patvirtino ir atlikto eksperimento rezultatai. Didėjant puslapio užklausų skaičiui, žiniatinklio paslaugas naudojančio puslapio atsako laikas dramatiškai auga. Remiantis atlikto tyrimo rezultatais, galime teigti, jog duomenų apsikeitimui taikant žiniatinklio paslaugas, reikia įvertinti, dėl XML dokumente perduodamos papildomos informacijos padidėsiantį techninių ir tinklo resursų apkrovimą. Kita vertus XML dokumento formatas suteikia papildomas lankstumo ir suderinamumo galimybes. Dėl šios priežasties, rinkoje pasirodė įvairios, XML dokumentų perdavimą optimizuojančios, techninės ir programinės priemonės.

7 IŠVADOS

- Atlikto sveikatos priežiūros įstaigų informacinių poreikių analizės metu paaiškėjo, jog gydymo įstaigos turi skirtingus informacinius bei funkcionalumo poreikius keistis sveikatos priežiūros informacija, todėl negalima kurti centralizuotos, monolitinės sistemos. Tuo pačiu paciento ambulatorinės informacijos saugojimo negalima visiškai patikėti gydymo įstaigoms, nes nebebūtų aiškaus informacijos savininko, bei kiltų didesnis pavojus visiškam paciento duomenų praradimui, todėl pagrindinius paciento duomenis reikia kaupti centralizuotai. Papildomas centralizuotai kaupiamų duomenų privalumas - yra duomenų pakartotinis panaudojimas analizės paslaugoms teikti.
- Atlikus paslaugų architektūros metodikos analizę nuspėta, jog paslaugų architektūrai realizuoti labiausiai tinka žiniatinklio paslaugų technologija, kuria be papildomų pastangų galima realizuoti pagrindinius paslaugų architektūros principus.
- Žiniatinklio paslaugų naudojami atviri standartai, leidžia pasiekti puikų posistemų suderinamumo laipsnį, tačiau standartų atvirumas ir prieinamumas sumažina sistemos saugumą.
- Sistemų integravimas pagrįstas paslaugų architektūra, leidžia perimti reikiamą išorinių sistemų funkcionalumą ir taip leidžia smarkiai sutrumpinti sistemos kūrimo laiką.
- Kuriant paslaugų architektūrą grįsto duomenų apsikeitimo modelio prototipą paaiškėjo, jog dėl paslaugos lygio abstrakcijos principo laikymosi, nebereikia rūpintis paslaugos realizacijos keitimų įtaka išorinėms sistemoms.
- Žiniatinklio paslaugų teikiama dinaminės paslaugų paieškos ir panaudojimo galimybė (angl. *Loose Coupling*), suteikia didesnę lankstumą, tačiau jų realizacija ir testavimas nepaprastai imlūs laikui. Pasirinktame modelyje paslaugų tiekėjų funkcionalumas aiškiai apibrėžtas, todėl paslaugų registrą modelyje siūloma naudoti tik dinaminei paslaugų vietos paieškai.
- Remiantis atlikto prototipo kokybės tyrimo ir puslapių atsako laiko eksperimento rezultatais, galima teigti jog silpnoji paslaugų architektūros vieta, yra už lankstumą sumokama resursų sunaudojimo kaina. Dėl šios priežasties, taikant paslaugų architektūrą, reikia numatyti resursų apkrovimą mažinančias priemones. Efektyviausios priemonės šiai problemai spręsti – yra duomenų kešavimas bei specializuotos techninės įrangos panaudojimas.

8 LITERATŪRA

- [1] Lietuvos Respublikos Sveikatos apsaugos ministerija. *Esveikatos sistemos plėtros Lietuvos Respublikos sveikatos apsaugos sektoriuje projektas (galimybių studija)*. Vilnius, 2005.
- [2] Lietuvos Respublikos Sveikatos apsaugos ministerija. *PROJEKTAS, ELEKTRONINĖS SVEIKATOS STRATEGIJA 2005-2010 m.* Vilnius, 2004.
- [3] AB „Alna“, *Ataskaita B Informacijos poreikio ir informacinių technologijų sveikatos sektoriuje infrastruktūros vertinimo ataskaita*. Vilnius, 2003.
- [4] Stojanovic Z.; Dahanayake A. *Software- Oriented System Engineering: Chananges and practices*. Idea Group, 2005, ISBN 1-59140-428-2.
- [5] Krafzig D.; Banke K.; Slama D. *Enterprise SOA: Service-Oriented Architecture Best Practices*. Prentice Hall PTR, 2004, ISBN 0-13-146575-9.
- [6] Bieberstein N.; Bose S.; Fiammante M.; Jones K.; Shah R. *Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap*. IBM Press, 2005, ISBN 0-13-187002-5.
- [7] Erl T. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR 2005, ISBN 0-13-185858-0.
- [8] Erl T. *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*. Prentice Hall PTR 2005, ISBN 0-13-142898-5.
- [9] Pulier E.; Taylor H. *Understanding Enterprise SOA*. Manning Publications Co. 2006, ISBN 1-932394-59-1.
- [10] Douglas K. *Web Services and Service-Oriented Architecture: The Savvy Manager's Guide*. Morgan Kaufmann Publishers 2003, ISBN:1558609067.
- [11] Hartman B.; Flinn J.; Beznosov K.; Kawamoto S. *Mastering Web Services Security*. Wiley 2003, ISBN 0-471-36716-3.
- [12] Hogg J.; Smith D.; Chong F.; Taylor D. *Web Service Security Scenarios, Patterns, and Implementations Guidance for Web Service Enhancements (WSE) 3.0*. Microsoft Press 2005, ISBN 0735623147.
- [13] Apperly H.; Hofman R.; Latchem S.; Maybank B.; McGibbon B.; Piper D.; Simons.C. *Service- and Component-based Development*. Addison Wesley 2003, ISBN 0-321-15985-3.
- [14] Chatterjee S.; Webber J. *Developing Enterprise Web Services: An Architect's Guide*, Prentice Hall PTR, 2003 ISBN 0-13-140160-2.

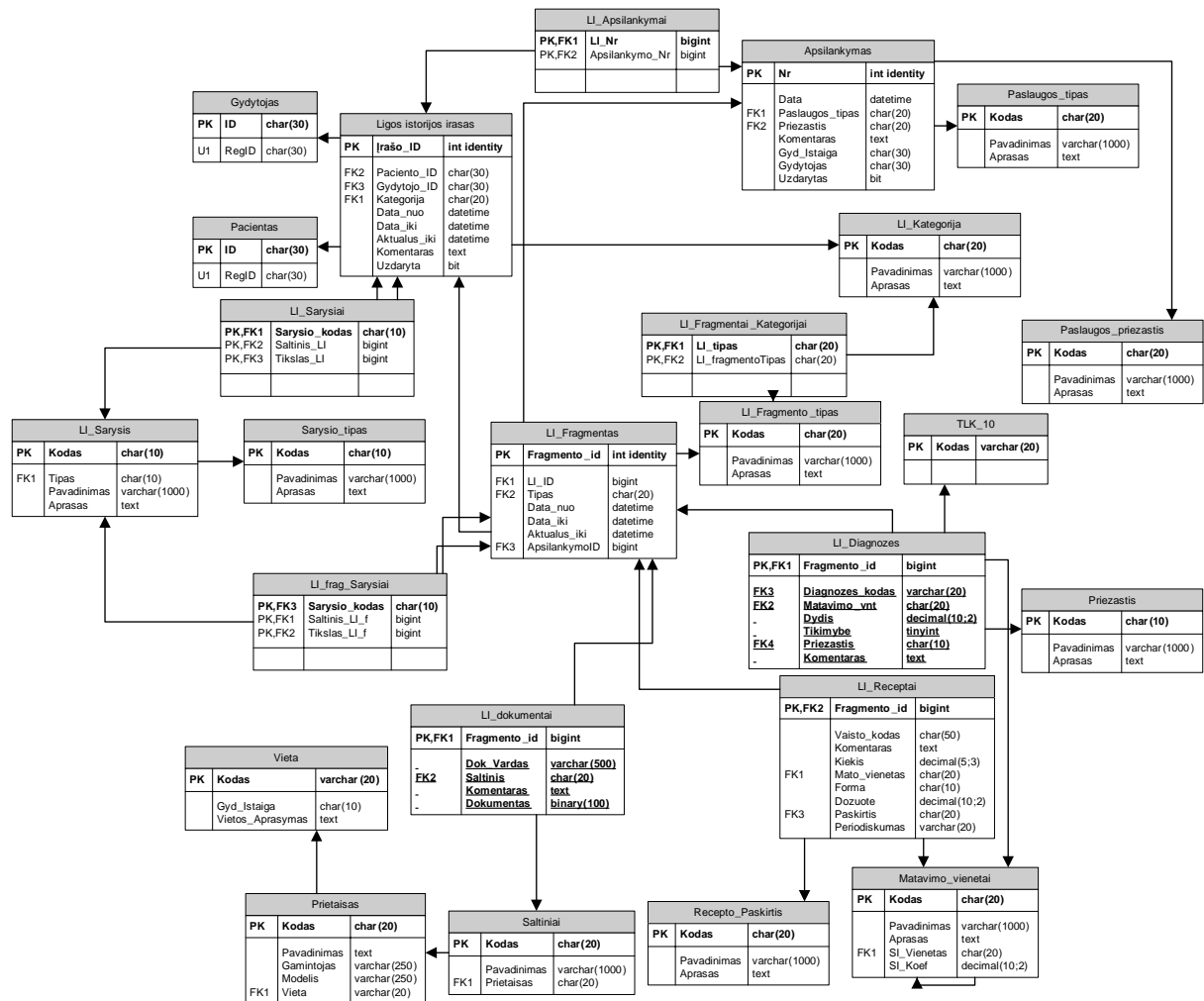
- [15] Nagel C. *Professional C# Web Services*, Wrox 2002 ISBN 1861004397.
- [16] Shepherd G. *Microsoft® ASP.NET 2.0 Step By Step*. Microsoft Press, 2005, ISBN 0-7356-2201-9.
- [17] Randell. B.; Lhotka R. *Visual Studio 2005 Bridge the Gap Between Development and Operations with Whitehorse*. [žiūrėta 2005.09.20]. Prieiga per internetą:
<http://msdn.microsoft.com/msdnmag/issues/04/07/Whitehorse/>
- [18] Burnett M.; Foster C. *Hacking the Code: ASP.NET Web Application Security*. Syngress Publishing, 2004, ISBN 1932266658.
- [19] Kittel M.; LeBlond G. *ASP.NET Cookbook*. O'Reilly, 2004, ISBN 0-596-00378-1.
- [20] Esposito D. *Programming Microsoft ASP.NET 2.0 Core Reference*. Microsoft Press 2005, ISBN 0735621764.

9 TERMINŲ IR SANTRUMPŲ ŽODYNAS


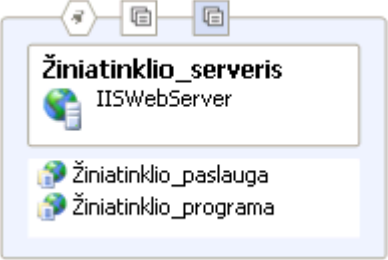



Terminas	Paiškinimas
SOA (Service Oriented Architecture)	Paslaugų architektūra
BPG	Bendrosios praktikos gydytojas
SPI	Sveikatos priežiūros įstaiga
PSPC	Pirminis sveikatos priežiūros centras
GI	Gydymo įstaiga
XML	Extended Markup Language
SOAP	Simple Object Access protocol
WSDL	Web Services Description Language
W3C	Pasaulinio tinklo konsorciūmas, tarptautinis interneto pramonės konsorciūmas (http://www.w3.org)
VLK	Valstybinė ligonių kasa Lietuvoje http://www.vlk.lt
TLK	Teritorinė ligonių kasa
HL-7	Sveikatos lygis 7. HL-7 yra standartinė sąsaja sveikatos duomenų mainams ir perdavimui tarp kompiuterinių sistemų. Tai taip pat yra pelno nesiekiančios ANSI akredituotos organizacijos pavadinimas. ANSI (Amerikos nacionalinis standartizacijos institutas) patvirtina visus nacionalinius standartus. ANSI X-12 komitetas nagrinėja su sveikatos priežiūra susijusius standartus, įskaitant pretenzijas ir perdavimus, kuriuos reglamentuoja HIPAA įstatymas. HL-7 kontroliuoja demografinius ir kitus pranešimų standartus. Naujausioje HL-7 versijoje 3.0 naudojama XML technologija. Iki šio dokumento paskelbimo ši versija dar nepasirodė. (Žr. HIPAA, XML.)
IS	Informacinės sistemos
LSIC	Lietuvos sveikatos informacijos centras (Lietuvos institucija) http://www.lsic.lt/
SSL	Saugi informacinė jungtis. Šifravimo protokolas (40 bitų arba 128 bitų algoritmas) sukurtas saugiai perdavimo jungčiai tarp vartotojo ir pagrindinės interneto svetainės bei leidžiantis atlikti E-verslo ar sveikatos priežiūros transakcijas. URL puslapiai, prasidedantys „https://“ gali reikšti, jog saugu pateikti kreditines korteles arba informaciją apie sveikatą.
TLK-10	Tarptautinė ligų klasifikacija
ATC	Vaistų klasifikatorius
Sveidra	Informacinė sistema, VLK sukurta apskaitos informacijos valdymui, kuria naudojasi Valstybinė ligonių kasa ir kitos institucijos.
Teisinis autentifikavimas	Įforminimo statusas, kai asmuo ranka arba elektroniniu parašu pasirašo dokumentą, už kurį jis yra teisiškai atsakingas. Tai paskutinis etapas darbo sraute.
DoS ataka	(angl. <i>Denyal of service</i>) tipo ataka. Atakos metu, siunčiamas didelis srautas užklausų, kuriomis apkrautas serveris, tampa neveiksnius.

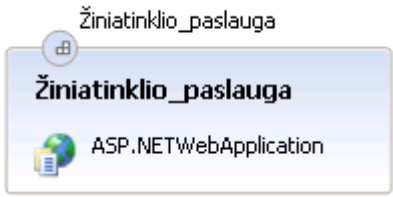
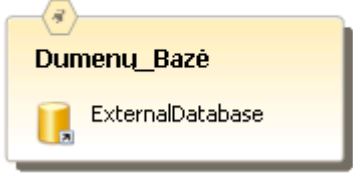




10 PRIEDAI

10.1 Priedas 1. Ligos istorijos įrašų duomenų bazės schema



10.2 Priedas 2. “Visual Studio 2005” paskirstytų sistemų diagramų notacija

	<p>Kliento kompiuteris</p>
	<p>IIS žiniatinklio serveris ir jame talpinamos interneto taikomosios programos.</p>
	<p>DBVS serveris ir jame talpinamos duomenų bazės.</p>
	<p>Zona, kurioje užtikrinamas techninės ir programinės įrangos saugumas</p>
	<p>ASP žiniatinklio programa</p>

	ASP žiniatinklio paslauga
	Duomenų bazė
	Duomenų bazės prisijungimas
	HTTP kliento prisijungimas
	Žiniatinklio puslapio sujungimas
	Žiniatinklio paslaugos

10.3 Priedas 3. ISO/IEC 9126 kokybės modelio aprašymas

Programinės įrangos kokybės įvertinimui galima panaudoti ISO/IEC 9126 standartą. Šis standartas pateikia programinės įrangos kokybės įvertinimo modelį, kuris gali būti taikomas įvairios programinės įrangos kokybės vertinimui. ISO/IEC 9126 standartas pateikia 6 programinės įrangos kokybės charakteristikas bei pateikia rekomenduojamas kokybės subcharakteristikas.

Pagal kokybės vertinimo modelį, kokybės vertinimas atliekamas tokiu metodu: nustatomos kokybės vertinimo charakteristikos; charakteristikoms nurodomos subcharakteristikos; subcharakteristikoms yra priskiriamos kokybės metrikos, kurių pagalba yra atliekamas kokybės matavimas.

ISO/IEC 9126 pateikiamos kokybės charakteristikos:

Charakteristika	Apibūdinimas
Funkcionalumas	Ar programinėje įrangoje yra prieinamos reikalaujamos funkcijos?
Patikimumas	Kaip patikima yra programinė įranga?
Panaudojamumas	Ar lengva naudoti programinę įrangą?
Efektyvumas	Kiek efektyvi yra programinė įranga?
Palaikomumas	Kaip lengva modifikuoti programinę įrangą?
Pernešamumas	Kaip lengva perkelti programinę įrangą į kitą veikimo aplinką?

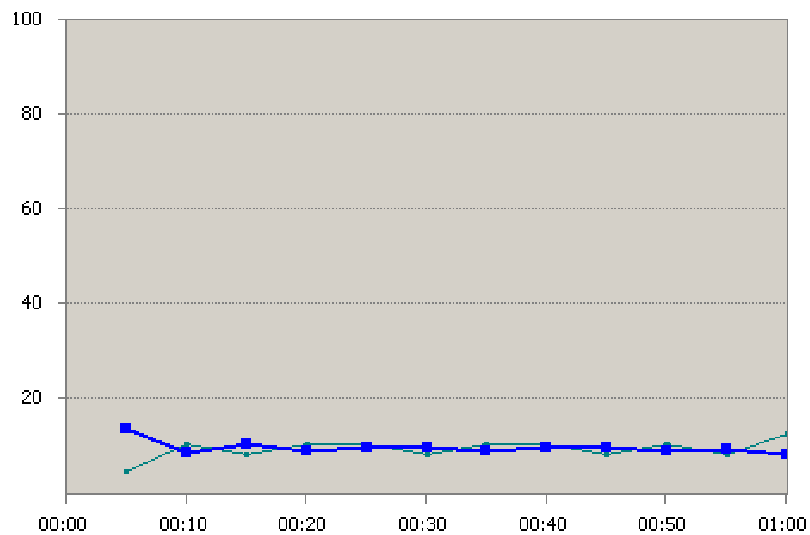
ISO/IEC 9126 standarte rekomenduojamos charakteristikų subcharakteristikos:

Charakteristika	Subcharakteristika	Apibūdinimas
Funkcionalumas	Tinkamumas	Produkto funkcijų, atliekančių reikiamas užduotis, pilnumas bei atitikimas reikalavimams.
	Tikslumas	Produkto veikimas pateikiant teisingus arba sutartus rezultatus.
	Bendradarbiavimas	Produkto sistemų tarpusavio bendradarbiavimo galimybės.
	Atitikimas standartams	Produkto atitikimas įvairiems standartams: PĮ kūrimo standartams, įstatymams ir pan.
	Saugumas	Produkto galimybės uždrausti neautorizuotą priėjimą prie programos arba programos duomenų.
Patikimumas	Brandumas	Nesėkmingų programos veikimo atvejų dėl gedimų dažnis.
	Gedimų tolerancija	Programos galimybė palaikyti nustatytą funkcionavimo lygį atsiradus tam tikriems gedimams.
	Atkuriamumas	Programos gebėjimas atstatyti funkcionavimo lygį bei prarastus duomenis nesėkmingo programinės operacijos atlikimo atveju nustatytoje laiko bei kaštų ribose.
Panaudojamumas	Suprantamumas	Vartotojo pastangos reikalingos programos loginio konteksto atpažinimui.
	Išmokstamumas	Vartotojo pastangos reikalingos siekiant išmokti dirbti su programa.
	Valdymas	Vartotojo pastangos reikalingos programos operacijų atlikimui bei jų valdymui.
Efektyvumas	Laiko režimas	Programos atsako bei veikimo laiko našumas.
	Resursų režimas	Programos naudojamų resursų apimtis bei resursų panaudojimo našumas.

Palaikomumas	Analizuojamumas	Pastangų apimtis, reikalinga programų trūkumų arba defektų analizei, arba modifikuojamų programos dalių nustatymui.
	Keičiamumas	Pastangų apimtis, reikalinga modifikacijoms, klaidų pašalinimui arba perėjimui prie kitos funkcionavimo aplinkos.
	Stabilumas	Rizikos dydis susijęs su nenuspėjamu funkcionavimu po programos modifikacijų.
	Testuojamumas	Pastangų apimtis, reikalinga atliekant programinės įrangos pripažinimą tinkančia.
Pernešamumas	Prisitaikymas	Programos galimybės prisitaikyti prie skirtingų funkcionavimo aplinkų
	Įdiegimo galimybės	Pastangų apimtis, reikalinga diegiant programinę įrangą nustatytoje funkcionavimo aplinkoje.
	Atitikimas standartams	Programinės įrangos atitikimas portatyvumo standartams.
	Pakeičiamumas	Galimybė panaudoti programinę įrangą vietoje kitos programinės įrangos jos funkcionavimo aplinkoje.

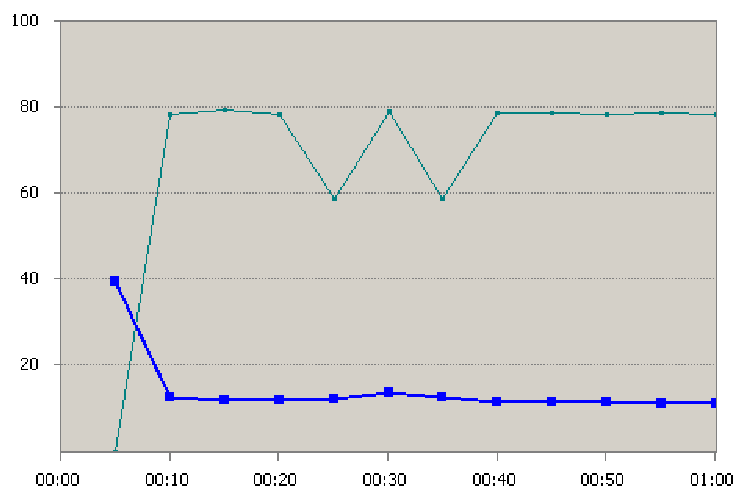
10.4 Priedas 4. Puslapio atsako laiko eksperimento rezultatai

Puslapis su 1 tiesiogine užklausa į serverio procedūrą



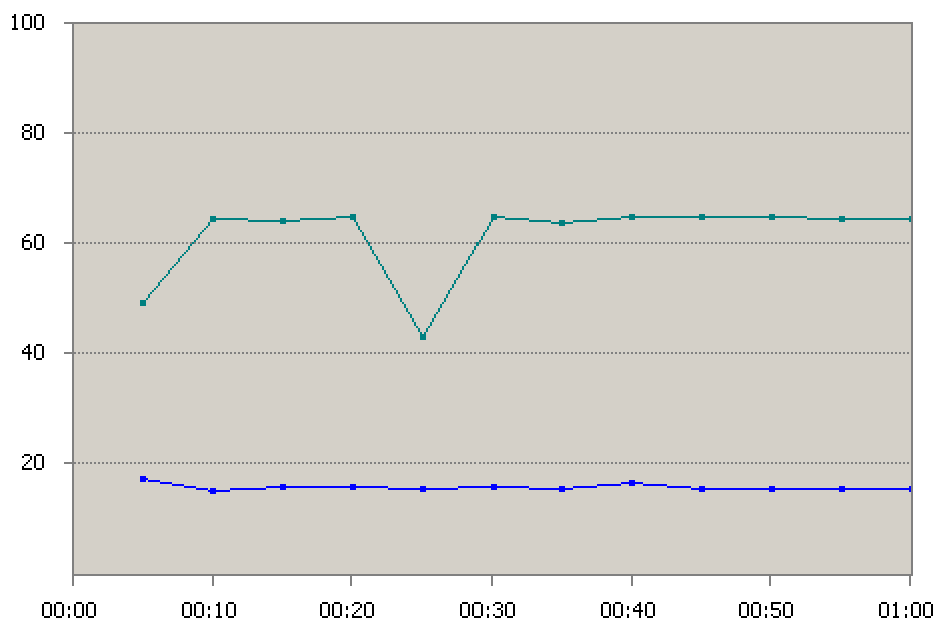
Parametras	Spalva	Mastelis	Min.	Max.	Vid.
Uždavimos/per sek.	■	1	0,60	1,00	0,90
Vid. Atsako laikas	■	10	0,87	1,41	0,98
Išsiųsti baitai/sek.	■	1E+07	479.823	1.296.279	982.096

Puslapis su 1 tiesiogine užklausa į žiniatinklio paslaugą



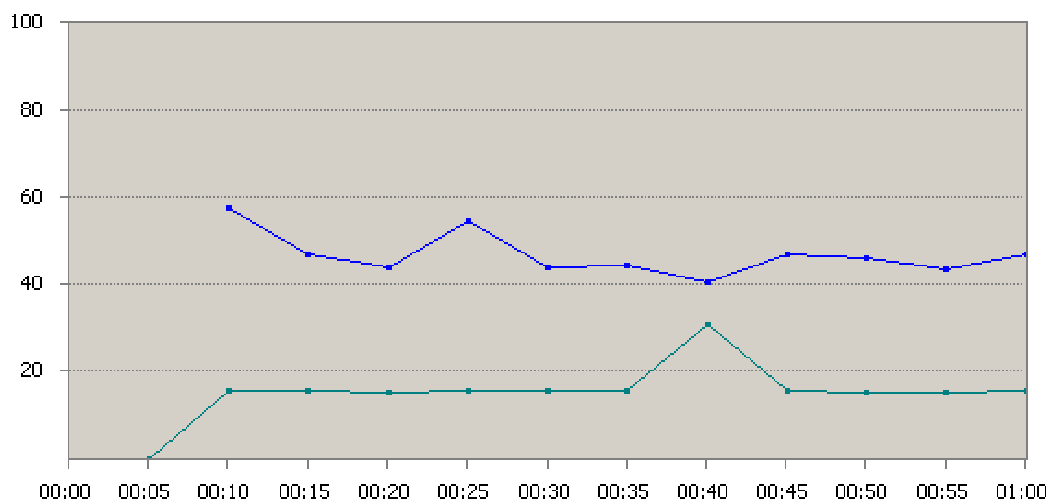
Parametras	Spalva	Mastelis	Min.	Max.	Vid.
Uždavimos/per sek.	■	1	0,20	0,80	0,72
Vid. Atsako laikas	■	10	1,13	3,97	1,28
Išsiųsti baitai/sek.	■	1000000	96	796.817	698.020

Puslapis su 5 tiesioginėmis užklausomis į serverio procedūrą



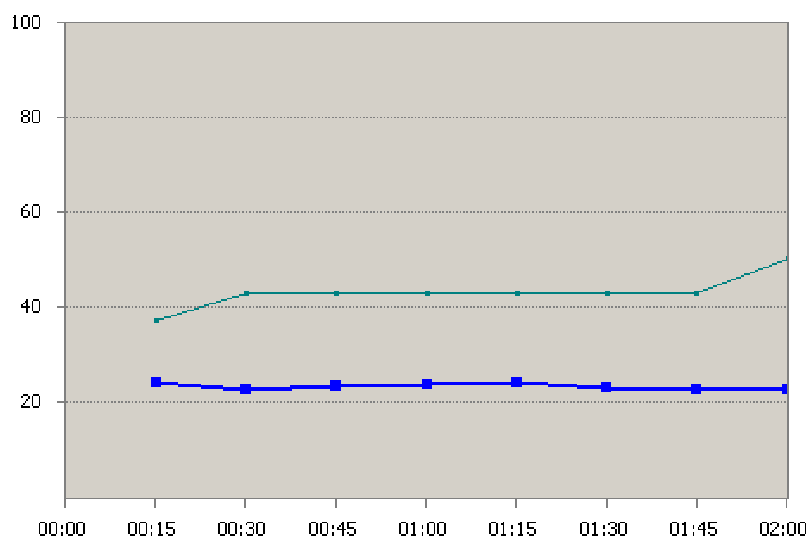
Parametras	Spalva	Mastelis	Min.	Max.	Vid.
Užklauso/per sek.	■	1	0,40	0,60	0,57
Vid. Atsako laikas	■	10	1,56	1,75	1,61
Išsiųsti baitai/sek.	■	1000000	435.778	654.095	620.897

Puslapis su 5 tiesioginėmis užklausomis į žiniatinklio paslaugą



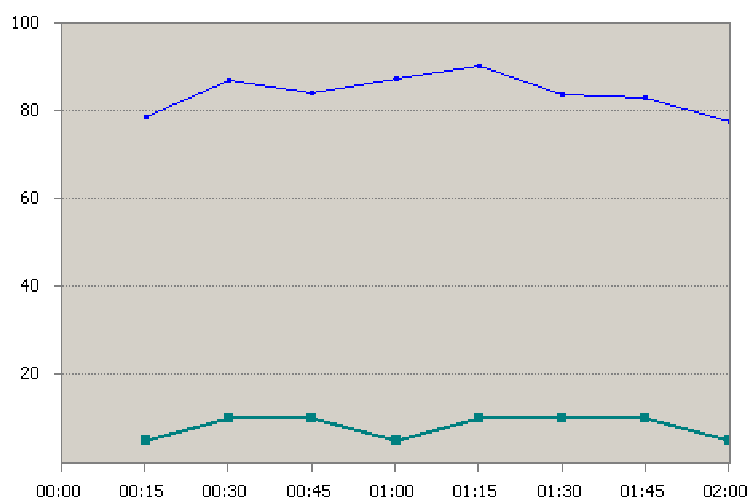
Parametras	Spalva	Mastelis	Min.	Max.	Vid.
Vid. Atsako laikas	■	10	4,09	5,78	4,70
Išsiųsti baitai/sek.	■	1000000	96	313.548	158.626
Užklauso/per sek.	■	1	0,00	0,40	0,20

Puslapis su 10 tiesioginių užklausų į serverio procedūrą



Parametras	Spalva	Mastelis	Min.	Max.	Vid.
Uždavimos/per sek.	■	1	0,33	0,47	0,40
Vid. Atsako laikas	■	10	2,30	2,47	2,37
Išsiųsti baitai/sek.	■	1000000	374.975	505.821	436.073

Puslapis su 10 tiesioginių užklausų į žiniatinklio paslaugą



Parametras	Spalva	Mastelis	Min.	Max.	Vid.
Uždavimos/per sek.	■	1	0,07	0,13	0,11
Vid. Atsako laikas	■	10	7,82	9,08	8,50
Išsiųsti baitai/sek.	■	1000000	52.333	105.184	85.468