

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Liudvikas Kreivis

**Grafinės vartotojo sąsajos generavimas pagal
tinklo paslaugos aprašą**

Magistro darbas

Darbo vadovas

prof. L. Nemuraitė

Kaunas, 2008

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Liudvikas Kreivis

**Grafinės vartotojo sąsajos generavimas pagal
tinklo paslaugos aprašą**

Magistro darbas

Recenzentas

dr. K. Kavaliauskas

2008-01-14

Vadovas

prof. L. Nemuraitė

2008-01-14

Atliko

IFM-2/4 gr. stud.
Liudvikas Kreivis

2008-01-14

Kaunas, 2008

Generation of graphical user interface from web service description

SUMMARY

Thanks to standards like web service Universal Description, Discovery, and Integration (UDDI), Web Services Description Language (WSDL), and SOAP, we can find web services and determine their interface and the message formats they understand. To actually call a web service, however, we must carefully study its documentation and then write special purpose code to handle it. As a further obstacle, no standards currently exist for making a service accessible through a web browser — doing so requires an ad hoc implementation.

To solve the problem, Web Service Graphical User Interface (WSGUI) approach is described that makes creating user-friendly, browser-based XForms GUIs for web services a straightforward task. Using this method, creating a GUI for a web service is as simple as writing a few XML documents.

WSGUI is a general method to augment web service interface and behavior descriptions with information about how to display certain aspects visually. The aim of the concept is to enable software application writers to integrate web service dialogs into their work independent of the structure of the web service.

This work presents the architecture, and XML documents required to deploy a WSGUI using abstract graphical interface descriptors and XForms technology. The implemented prototype is presented with examples of generated graphical user interfaces for different Web services with complex data types, different styles and different languages. Due to implementation of adapter pattern, the implementation could be easily applied with other GUI technologies and platforms

Keywords: Web service, WSDL, XML, XForms, WSGUI, XSD, XSLT

Turinys

1.	Įvadas.....	1
2.	Automatinio grafinės vartotojo sąsajos generavimo tinklo paslaugoms metodų analizė.....	4
2.1	Vartotojo sąsajos generavimo problemos analizė.....	5
2.2	Problemos sprendimo metodų literatūros šaltiniuose analizė.....	7
2.2.1	WSGUI – grafinės tinklo paslaugos sąsajos generavimo metodas.....	7
2.2.2	SOAPClient.com – visiškai automatinio sąsajos generavimo metodas.....	8
2.2.3	WSUI sąsajos generavimo metodas.....	8
2.2.4	Microsoft .NET platformoje naudojamas metodas.....	9
2.2.5	Automatinio grafinės sąsajos generavimo metodų analizės apibendrinimas.....	10
2.3	Grafinės sąsajos aprašymo metodų, formatų ir generavimo įrankių analizė.....	11
2.3.1	Abstraktūs GUI aprašymai.....	11
2.3.2	Nuo platformos priklausomos aprašymo kalbos.....	12
2.3.3	Įrankių komplektams skirti formatai.....	13
2.3.4	Vartotojo sąsajos generavimo įrankiai.....	14
2.3.5	Sąsajos aprašymo formatų ir generavimo įrankių analizės apibendrinimas.....	15
2.4	Architektūros ir galimų įgyvendinimo priemonių pasirinkimas.....	15
2.5	Darbo tikslas ir kokybės kriterijai.....	17
2.6	Reikalavimai duomenų struktūroms.....	18
2.7	Rizikos faktorių analizė.....	18
2.8	Analizės išvados.....	19
3.	Automatinio vartotojo sąsajos generavimo reikalavimų specifikacija.....	20
3.1	Funkciniai reikalavimai.....	20
3.2	Nefunkciniai reikalavimai.....	21
4.	Automatinio grafinės sąsajos generavimo sistemos projektas.....	22
4.1	Sistemos architektūra.....	22
4.2	Panaudojimo atvejų diagrama.....	23
4.3	Sekų diagrama.....	24
4.4	Grafinės sąsajos generavimo variklio veiklos diagrama.....	25
4.5	Komponentų diagrama.....	26
4.6	Sistemos elgsenos modelis.....	27
4.7	Klasių diagramos.....	29
4.8	Sąsajos duomenų struktūros.....	35
4.8.1	Tinklo paslaugos specifikacija (WSDL – Web Service Description Language).....	35
4.8.2	XSL schemas transformacijos (XSLT).....	39
4.8.3	Grafinės vartotojo sąsajos elementų išdėstymo aprašymas (GUIDD).....	40
4.9	Naudotų metodų, algoritmų, technologijų aprašas.....	44
4.10	WSGUI veilos diagrama.....	47
5.	Automatinio grafinės vartotojo sąsajos generavimo prototipo eksperimentinė realizacija ir įvertinimas.....	48
5.1	Automatiškai sugeneruotos svetainės pavyzdys.....	48
5.2	Vartotojo sąsajos generavimo prototipo įvertinimas.....	53
6.	Išvados.....	54
7.	Literatūra.....	55

1. Įvadas

Sparčiai vystantis tinklo paslaugų (web service) technologijoms bus reikalinga universali sąsaja (interface) skirtingiems jų panaudojimams. Skirtingi tinklo paslaugų ir portalų susiejimai yra vienas sudėtingiausių biznio procesų. Norint surasti lankstų sprendimą yra būtina dinaminė paslaugų integracija.

Norint pateikti paslaugą portale, reikalinga papildoma informacija apie grafinę vartotojo sąsają - nepakanka vien tik techninio tinklo paslaugos aprašo (WSDL). Automatinio vartotojo sąsajos generavimo idėja yra sukurti grafinę sąsają, kuo labiau suprantamą vartotojui, naudojant kiek įmanoma mažiau papildomos informacijos aprašančios pačią grafinę sąsają.

Naudojimąsi tinklo paslaugomis palengvina keletas jau egzistuojančių standartų. Tai tinklo paslaugų aprašymų aptikimas kataloguose naudojant UDDI (Universal Description, Discovery and Integration) standartą, jų struktūros bei teikiamų paslaugų nuskaitymas iš WSDL (Web Service Description Language) dokumentų, komunikavimas su tinklo paslaugomis naudojant XML (Extended Markup Language) pranešimus ir SOAP (Simple Object Access Protocol) protokolą. Toks tinklo paslaugų pristatymas palengvina informacijos tarp taikomųjų programų apsikeitimą. Atsiranda galimybė aptikti tinklo paslaugas, išsiaiškinti sąsają ir pranešimų formatus, kuriuos jos suprastų. Tačiau norint iš tikrųjų iškviešti tinklo paslaugą, reikia atidžiai studijuoti jos dokumentaciją ir parašyti specialiai tam pritaikytą kodą. Kita kliūtis yra ta, kad kol kas nėra standarto, kuris leistų paslaugą iškviešti per interneto naršyklę – norint tai atlikti, taip pat reikalinga šiam tikslui sukurta programinė įranga [1].

Paprastam vartotojui naudoti tinklo paslaugą be kokios nors specialiai tam sukurtos grafinės aplinkos yra itin sudėtinga, o kartais net ir neįmanoma. Labai dažnai tinklo paslaugose naudojami nesuprantami vartotojui, daugiau techninę prasmę turintys kintamųjų pavadinimai, taip pat sudėtingi duomenų tipai, neaiškūs galimų reikšmių apribojimai, nesuprantamos operacijos.

Šiame darbe problemą bandoma spręsti naudojant abstraktų grafinės sąsajos aprašą. Toks aprašas kartu su tinklo paslaugos aprašu (WSDL) leistų vartotojui vizualiai matyti galimas operacijas, reikalingus parametrus, suteiktų galimybę kviešti tinklo paslaugą ir gauti taip pat vizualizuotą rezultatą.

Naudojant abstraktų grafinės sąsajos aprašą, neprisirišama prie konkrečių grafinės sąsajos atvaizdavimo technologijų, platformų ir interneto naršyklių. Šiame darbe nuspręsta pagal abstraktų aprašą generuoti XForms tipo grafinę sąsają. Ši technologija pasirinkta todėl, kad XForms yra pagrįsta XML kalba, nuo 1.1 versijos yra galimybė naudoti SOAP užklausas, kurių pagalba galima iškviešti tinklo paslaugas. XForms atskiria duomenų prezentacijos elementus nuo turinio, suteikia pakartotinio panaudojimo (reuse) galimybę, yra nepriklausoma nuo platformos ir sumažina scenarijų (scripts) poreikį [1].

Šio tyrimo sritis - XML tinklo paslaugos, tinklo paslaugų architektūra, o tyrimo objektas - tinklo paslaugų komponentų vartotojo sąsajos generavimo interneto svetainėje procesas.

Šio darbo tikslas – pagal tinklo paslaugų aprašus (WSDL) sugeneruoti kuo suprantamesnę vartotojui grafinę sąsają, įgalinančią kviešti tinklo paslaugas ir peržiūrėti gautus rezultatus.

Darbo uždaviniai:

- Ištirti šiuo metu naudojamus įrankius ir metodus automatiniam tinklo paslaugų grafinės vartotojo sąsajos generavimui.
- Pasirinkti tinkamą metodą ir suprojektuoti automatinės vartotojo sąsajos generavimo sistemą.
- Realizuoti suprojektuotą sistemą ir eksperimentiškai išbandyti automatinio sąsajos generavimo veikimą.

Atliekant automatinio sąsajos generavimo metodų analizę, buvo remtasi literatūros šaltiniais [1], [16], kuriuose apžvelgiami skirtingi vartotojo sąsajos generavimo metodai ir technologijos. Pagal tai atliktas palyginimas ir pasirinktas WSGUI metodas, kuris pasirodė tinkamiausias darbui. Taip pat remtasi [5], [11], [12], [13], [14], [15] literatūros šaltiniais, kuriuose plačiau apibūdinami prieš tai paminėtuose darbuose aprašyti metodai: WSUI, SOAPClient, .NET ir kiti.

Atliekant realizavimo technologijų analizę, naudotasi [1] darbu. Pasirinkta XForms technologija plačiau aprašoma [3] ir [4] internetinėse svetainėse.

Literatūros šaltiniuose [2], [6], [7] aprašyta tinklo paslaugų architektūra, jų savybės, privalumai, trūkumai. Į tai atsižvelgta projektuojant automatinio sąsajos generavimo variklį ir jo veikimo principą.

Darbo struktūra:

Antrame skyriuje pateikiami atliktos analizės rezultatai. Atlikta problemos srities analizė, peržvelgti šiuo metu naudojami automatinio grafinės sąsajos generavimo metodai, įvertinti egzistuojantys įrankiai, naudojami tinklo paslaugų grafinei sąsajai generuoti. Atlikta vartotojų analizė. Išanalizuotas ir pasirinktas WSGUI metodas, kuris pasirodė tinkamiausias problemai spręsti. Nustatyti reikalavimai ir pasirinkti kokybės kriterijai, iš kurių svarbiausi yra sąsajos aiškumas ir patogumas paprastam vartotojui, kad naudojant kuo mažiau papildomos informacijos, būtų galima siųsti užklausas tinklo paslaugoms ir gauti rezultatus.

Trečioje dalyje nustatomi funkciniai ir nefunkciniai reikalavimai pasirinktos problemos sprendimui, apibrėžiamas automatinio vartotojo sąsajos generavimo tinklo paslaugoms funkcionalumas.

Ketvirtame skyriuje pateikiamas kuriamos sistemos projektas, detalizuojantis aprašytų reikalavimų realizaciją.

Penktojoje dalyje aprašoma sukurto automatinio grafinės vartotojo sąsajos generavimo realizacija. Pateikiami demonstraciniai pavyzdžiai. Trumpai aprašomas sistemos veikimas.

Šeštame syriuje pateikiamos darbo išvados.

2. Automatinio grafinės vartotojo sąsajos generavimo tinklo paslaugoms metodų analizė

Šioje dalyje pateikiami atliktos analizės rezultatai. Atlikta problemos srities analizė, peržvelgti šiuo metu naudojami automatinio grafinės sąsajos generavimo metodai, įvertinti egzistuojantys įrankiai, naudojami tinklo paslaugų grafinėi sąsajai generuoti. Atlikta vartotojų analizė. Išanalizuotas ir pasirinktas problemos sprendimo metodas, nustatyti reikalavimai ir pasirinkti kokybės kriterijai.

Analizės tikslas - susipažinti su jau sukurtais tinklo paslaugų kūrimo standartais; išanalizuoti jau egzistuojančius kūrimo metodus; apžvelgti jau naudojamus arba bandomus sukurti automatinio vartotojo sąsajos generavimo tinklo paslaugoms metodus, išsiaiškinti jų privalumus, trūkumus. Nustatyti, su kokiomis problemos susiduriama, kokios grėsmės slypi tinklo paslaugos grafinės sąsajos generavime ir, galiausiai, pasirinkti vartotojo sąsajos generavimo metodą.

Tyrimo sritis - XML tinklo paslaugos, tinklo paslaugų architektūra.

Tyrimo objektas - interneto svetainių vartotojo sąsajos generavimo metodai tinklo paslaugų komponentams.

Klausimai, į kuriuos turi atsakyti šis tyrimas:

- Kaip greitai atnaujinti vartotojo grafinę sąsają dažnai keičiantis teikiamoms tinklo paslaugoms;
- Kokie yra skirtingų automatinio sąsajos generavimo metodikų privalumai ir trūkumai ir kuris būdas yra tinkamiausias problemai spręsti.
- Kaip generuoti formas atvaizduojančias sudėtingus duomenų tipus;
- Kaip realizuoti generuojamos sąsajos daugiakalbiškumą;
- Kaip pagal tinklo paslaugos aprašymą realizuoti patogų duomenų įvedimą ir navigaciją po svetainę;

2.1 Vartotojo sąsajos generavimo problemos analizė

Pastaruoju metu IT infrastruktūrą charakterizuoja vis dažnesnis susijusių procesų atskyrimas. Procesai gali būti suprantami kaip programinės įrangos komponentai, arba kaip servais, kurie komunikuoja vienas su kitu per kompiuterių tinklą. Panaudojant labiau standartizuotas aprašymo kalbas, tokias kaip XML, sąsajų ir pranešimų keliaujančių tarp servisų aprašymui, buvo sukurta nauja servisų klasė pavadinta web servais. Remiantis jais, atsirado ir nauja į servisus orientuota architektūra (SOA), kadangi šie servais gali būti iškviešti ir susieti vienas su kitu.

Servisų iškvietimas dažniausiai atliekamas naudojantis specialiomis programomis, kurios vartotojo duomenis sugeba paduoti servisui ir konvertuoti gautą rezultatą į vartotojui suprantamą formą. Tačiau nėra priežasties, kodėl nebūtų galima savarankiškai kviešti servisų nesinaudojant specialiai tam pritaikyta programa. Išanalizavę architektūrą matome, kad apdorojus web serviso aprašymą (WSDL) galima automatiškai generuoti vartotojo sąsają, kuri leistų įvesti pradinis duomenis ir sugebėtų apdoroti ir pateikti gautą rezultatą. Susiduriame su grafinės vartotojo sąsajos generavimo pagal formalų modelį problema.

Funkcionalios ir intuityvios vizualios sąsajos taikomajai programai kūrimas (taip pat įskaitant ir servisų valdymą) tradiciškai atliekamas „rankomis“ ir yra taisomas ir koreguojamas kiekvienai kitai taikomajai programai. Pakeitus serviso parametą, greičiausiai reikės atnaujinti visą grafinę sąsają. Todėl kūrimo procese yra pageidaujama kuo daugiau automatizavimo, kas kartais gali būti atliekama už taikomosios programos ar tinklo paslaugos ribų.

Norint sukurti tokios sąsajos generavimą, reikia atlikti analizę ir išsiaiškinti, kokią platformą ir pagrindinę aprašymo kalbą pasirinkti. Sąsaja turi būti kuo lankstesnė, interaktyvi, turi būti įmanomas grįžtamasis ryšys.

Vartotojo ir taikomosios programos dialogas paprastai vyksta per konkrečiai tam sukurtą valdymo sąsają. Sąsajoje dažniausiai pateikiami fundamentalūs elementai teksto įvedimui ir išvedimui, dažnai komunikavimo dialogas papildomas grafiniais simboliais, pagalbiniais tekstais ir navigacijos elementais. Kiti elementai, tokie kaip aktyvūs pranešimai, akcentuojami neteisingai įvesti duomenys, ar automatiškas įvedimo užbaigimas (auto complete) nėra būtini, tačiau žymiai padidina suprantamumą, patogumą ir dėl to yra pageidautini. Taip pat negalima pamiršti ir modernios programinės įrangos ergonomiškumo kriterijų, tokių kaip internacionalizacija ir integracija į darbo aplinką.

Egzistuoja trijų tipų vartotojo sąsajos:

- Tekstinė vartotojo sąsaja (TUI – Text User Interface)
- Grafinė vartotojo sąsaja (GUI – Graphic User Interface)
- Kintamo vaizdo mastelio vartotojo sąsaja (ZUI – Zooming User Interface)

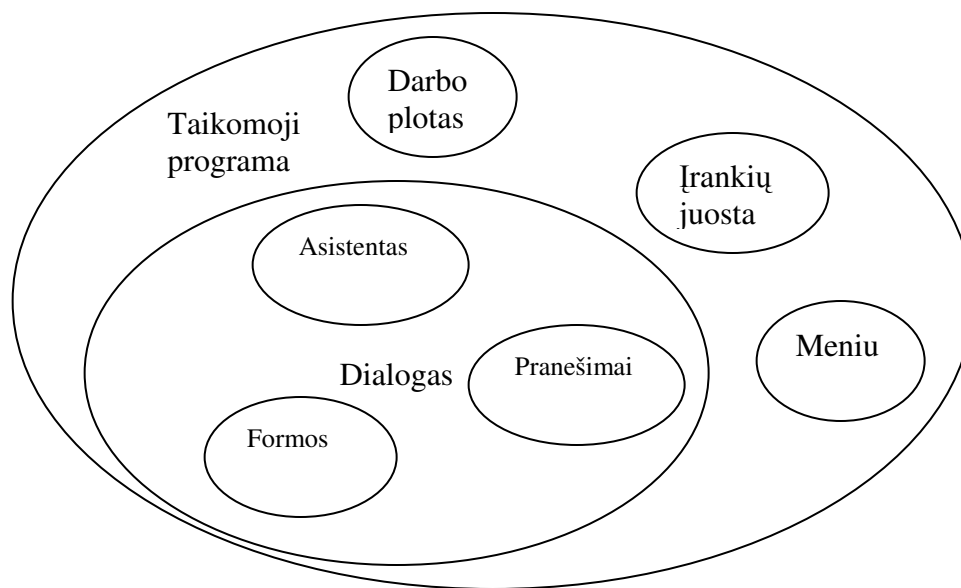
Tekstinė vartotojo sąsaja (Text User Interface arba Textual User Interface). Šis terminas sugalvotas atsiradus grafinei vartotojo sąsajai, kad būtų išvengta painiavos ir galima lengvai atskirti apie ką kalbama. TUI skiriasi nuo komandinės eilutės sąsajos, kadangi joje kaip ir GUI informacijos vaizdavimui naudojamas visas ekrano plotas ir nebūtinai informacija išvedama eilutė po eilutės. Vis tik TUI naudojama tik tekstas ir tipiškai terminalo simboliai, kai tuo tarpu GUI naudojamos didelės skiriamosios gebos grafiniai režimai. TUI pavyzdžiais gali būti MS-DOS ar Unix tipo operacinės sistemos.

Grafinė vartotojo sąsaja (Graphic User Interface) – tai dažniausias vartotojo sąsajos atvejis. Tokio tipo bendravimui su kompiuteriu skirta sąsaja, be teksto, informacijos atvaizdavimui gali turėti ir grafinių vaizdų ir kitų papildomų priemonių. Dažniausiai veiksmai ir atliekami tiesiogiai manipuluojant grafiniais elementais.

Kintamo vaizdo mastelio vartotojo sąsaja (Zooming User Interfaces) yra grafinė aplinka, kuri leidžia vartotojui pasiekti sistemos objektus. Tai grafinės vartotojo sąsajos tipas. ZUI gali sąsają atvaizduoti skirtingo dydžio skalės ir detalumo. Vartotojui leidžiama keisti skalės dydį peržiūrimam plotui, kad būtų parodytas detalesnis vaizdas.

Tokio tipo sąsajoje informacijos elementai yra parodomi tiesiogiai begaliniam virtualiame darbastalyje (dažniausiai sukurtame pasitelkiant vektorinę grafiką), nei tiesiog languose. Vartotojams leidžiama dvejų dimensijų režime peržvelgti panoraminį vaizdą ir pritraukti dominančius objektus. Pavyzdžiui, aptiktas teksto įvedimo laukas gali atrodyti kaip nedidelis taškas; pritraukus vaizdą – kaip mažas paveikslėlis; dar pritraukus būti atvaizduotas kaip pilnas puslapis, ir galų gale kaip padidintas puslapis.

Darbe automatinio generavimu bus bandoma sugeneruoti grafinę vartotojo sąsają (GUI). Tekstinė sąsaja jau yra pasenusi, o kintamo mastelio sąsaja perkrauta daugybe pagalbinių priemonių, kurios gali apsunkinti generavimą. Dar vienas ZUI sąsajos trūkumas tas, kad yra gan sudėtinga užtikrinti duomenų įvedimo nuoseklumą, kadangi vartotojas gali pasiekti bet kurią lango vietą. Grafinė sąsaja šiuo metu yra populiariausia ir suprantamiausia vartotojui, o jos funkcionalumo visiškai pakanka.



1 pav. Taikomosios programos GUI struktūra

Automatinės sąsajos generavimo sistemos palaikymą, web servisų kūrimą ir grafinės sąsajos generavimo priežiūrą turėtų atlikti programavimo ir sistemų administravimo patirties turintys asmenys.

Tinklo paslaugų per sugeneruotą grafinę sąsają vartotoju gali būti įvairūs asmenys. Tai labiau priklauso nuo to, kokia tinklo paslauga yra teikiama. Vienas iš pagrindinių darbo siekių – kad ir paprastas vartotojas galėtų naudotis tinklo paslauga per interneto naršyklę.

2.2 Problemos sprendimo metodų literatūros šaltiniuose analizė

Šioje dalyje apžvelgiami šiuo metu jau egzistuojantys, kuriami ar dar tik projektuojami metodai pasirinktai problemai spręsti. Analizuojami automatinio grafinės sąsajos generavimo metodai.

2.2.1 WSGUI – grafinės tinklo paslaugos sąsajos generavimo metodas

Prieš keletą metų pasiūlytas problemos sprendimo būdas pavadintas WSGUI (Web service GUI) paremtas XML aprašymais. Šis būdas leidžia kurti draugiškas vartotojui, interneto naršyklėms skirtas grafines sąsajas pagal web servisų aprašymą (WSDL).

Norint sugeneruoti grafinę sąsają web servisui pagal WSGUI schemą, reikia dviejų XML dokumentų. Pirmasis yra GUI realizacijos aprašymas (GUIDD – GUI Deployment Descriptor), kuris saugo abstraktų vartotojo sąsajos aprašymą. Antrasis yra XSLT transformacijos aprašymas, kuris iš abstraktaus aprašymo padeda sukurti konkrečią GUI sąsają. Galima susieti keletą transformacijos failų su vienu GUIDD dokumentu, taip suteikiant galimybę gauti skirtingas realizacijas pagal tą patį aprašymą.

XML dokumentui pritaikius XSLT transformaciją galime matyti duomenis draugišku vartotojui pavidalu, o pagal GUIDD gauname konkretų atvaizdavimą. Šie du dokumentai leidžia varikliui generuoti WSGUI [1], [16].

Šiuo sąsajos generavimo principu ir bus remiamasi sprendžiant problemas. Detaliau architektūra bus aprašyta tolimesniame 2.9.1 skyriuje.

2.2.2 SOAPClient.com – visiškai automatinio sąsajos generavimo metodas

SOAPClient.com pristatė papildomą WSDL aprašymu pagrįstą grafinės sąsajos generatorių. Šis generatorius naudoja tik WSDL aprašymą. Dėl to gaunama grafinė sąsaja yra paprasčiausia HTML forma turinti teksto įvedimo laukus, pavadintus taip pat, kaip WSDL aprašymo laukai, kuriuos jie vaizduoja. Tik paprasti duomenų tipai yra palaikomi. Leidžiama kviesti tinklo paslaugą per www.soapclient.com/soapclient tinklapį arba tiesiog nurodant WSDL failo aprašą kaip parametą http://www.soapclient.com/soapTest.html?SoapWSDL=WSDL_failo_adresas. Klientui galima paduoti ir sudėtingo tipo XML serializuotą objektą. Vistik visiškai automatiškai generuojami tik paprasti tipai ir vieno lygmens masyvai. SOAP Client svetainėje pateikiama ir keletas kitų naudingų įrankių darbui su tinklo paslaugomis. Tai SOAP saugumo įrankiai, SOAP pranešimų kūrimo įrankiai ir kiti [14].

2.2.3 WSUI sąsajos generavimo metodas.

WSUI principas labai panašus į WSGUI. Pagal WSDL sukuriama „component definition“ aprašas ir XSLT transformacija. „Component definition“ dokumente aprašomas ribotas kiekis kintamųjų, kurie paduodami kaip parametrai XSLT transformacijai. Kiekvienas sugeneruotos sąsajos laukas yra susietas su tam tikru kintamuoju. Norint iškviesti web servisą, yra rašomas skriptas, kuris surenka kintamųjų reikšmes ir sukuria web serviso pranešimą.

Kadangi kintamųjų skaičius yra visuomet fiksuotas, WSUI negali susidoroti su neriboto dydžio sąrašais duomenų įvedimui, o WSGUI tą atlikti sugeba. Be to, WSGUI dinamiškai sukuria web serviso pranešimą iš įvestų duomenų. Tai sumažina rūpesčius programuotojams, kadangi nereikia papildomai rašyti kodo pranešimo sukūrimui, kaip to reikalauja WSUI [1], [13].

2.2.4 Microsoft .NET platformoje naudojamas metodas

Microsoft kompanijos sukurtoje .NET platformoje parašytas tinklo paslaugas taip pat galima kviesti naudojant interneto naršyklę. Grafinės sąsajos generavimas yra visiškai automatinis. Tačiau palaikomi tik paprasti .NET kintamųjų tipai. Parametrų pavadinimai, galimų operacijų pavadinimai nustatomi pagal WSDL aprašą. Pasirinkus tinklo paslaugą su sudėtiniais tipais pasirodo pranešimas, kad paslauga negali būti įvykdyta [2], [6], [7], [9].

Šis metodas yra priklausomas nuo platformos ir naudojamas daugiau programuotojų Microsoft .NET platformoje tada, kai reikalinga patikrinti tinklo paslaugos veikimą ar klaidų taisymo metu. Pasirinktų problemų sprendimui tai nėra pats tinkamiausias variantas.

2.2.5 Automatinio grafinės sąsajos generavimo metodų analizės apibendrinimas

Apžvelgti automatinio grafinės sąsajos generavimo metodai pateikti 1-oje lentelėje. Atlikus metodų analizę nuspręsta pasirinkti WSGUI sąsajos generavimo metodą.

1 lentelė. Automatinio grafinės sąsajos generavimo metodų palyginimas

	WSGUI	SOAP Client	WSUI	.NET
Generavimo būdas	Pagal abstraktų aprašą	Pilnai automatinis	Pagal kintamųjų aprašą	Pilnai automatinis
Reikalavimai	GUI Deployment Descriptor, WSDL	WSDL	Component definition, WSDL	WSDL, palaikoma tik .NET platforma
Palaikomi kintamųjų tipai	Visi	Tik paprasti duomenų tipai ir vienos dimensijos masyvai	Fiksuotas kiekis kintamųjų. Neapdoroja neriboto dydžio sąrašų	Paprasti .NET tipai
Atsakymo formavimas	Pagal abstraktų aprašą	Automatinis	Pagal kintamųjų aprašą	Automatinis
Duomenų įvedimas keletu žingsnių	Palaikomas	Nepalaikomas	Nepalaikomas	Nepalaikomas

Pasirinkto metodo pagrindinis privalumas yra neribojamas duomenų tipų pasirinkimas. Kiekvienas duomenų tipas gali būti aprašytas abstraktaus aprašo faile ir atvaizduotas vartotojui. Taip pat WSGUI metodas gali būti realizuotas bet kurioje aplinkoje, nepriklausomai nuo pasirinktos platformos ir technologijos. Visiškai automatinis sąsajos generavimas apima ne visus problemos sprendimo būdus ir kartais gali pateikti neprognozuojamus rezultatus. Microsoft metodas taip pat netinkamas ir dėl priklausomybės nuo platformos.

2.3 Grafinės sąsajos aprašymo metodų, formatų ir generavimo įrankių analizė

Šiame skyriuje pateikiami įvairūs vartotojo sąsajos aprašymo formatai, realizavimo technologijos ir pateikiami jų palyginimai (2 lentelė). Taip pat pateikiamas išanalizuotų grafinės sąsajos generavimo įrankių analizė.

2.3.1 Abstraktūs GUI aprašymai

Abstraktus aprašymas leidžia aprašyti vartotojo sąsają, kuri gali būti skirtingai atvaizduota priklausomai nuo platformos. Tačiau visas bendravimas su vartotoju yra apribotas tais veiksmais, kurie gali būti atvaizduoti aprašymo formate. GUI elementai čia gali būti mygtukai, teksto įvedimo laukai, pasirinkimo laukai ir kiti panašūs įrankiai.

2.3.1.1 XForms technologija

XForms - formatas XML duomenų bei vartotojo sąsajos apdorojimo modeliui ir duomenų aprašymui. XForms buvo sukurtos kaip kitos kartos HTML / XHTML formos, tačiau gali būti naudojamos ir kaip atskiras formatas aprašyti kitų kalbų nei XHTML vartotojo sąsają ir veiksmų rinkinį duomenų apdorojimui.

XForms aprašymai negali funkcionuoti vieni. Reikalingi dokumentai, kurie sugrupuoja, sutvarko formos elementus. Elementus ši aprašymo kalba pateikia abstrakčia forma. Priklausomai nuo pagrindinio dokumento, tokio kaip XHTML ar SVG, elementai gali būti atvaizduoti GUI ar ZUI vartotojo sąsajoje.

XForms yra pagrįsta XML kalba. Tiek formų aprašymas, tiek įvestų duomenų perdavimas atliekamas XML formatu. Komunikavimas vyksta per HTTP POST OFFICE metodus, o nuo 1.1 versijos yra galimybė naudoti SOAP užklausas, kurių pagalba galima iškviešti web servisus. Egzistuoja keletas specialių XForms formų realizacijų, taip vadinamų procesorių. Viena iš tokių Mozilla grupės interneto naršyklės, kurios suteikia galimybę atvaizduoti formas įvairiose platformose.

XForms atskiria duomenų prezentacijos elementus nuo turinio, suteikia pakartotinio panaudojimo (reuse) galimybę, suteikia nepriklausomybę nuo platformos ir sumažina skriptų poreikį [3],[4].

2.3.1.2 UIML – vartotojo sąsajos aprašymo kalba

Vartotojo sąsajos aprašymo kalba (UIML – User Interface Markup Language).

Labai abstrakti kalba, kuri apibūdina GUI elementus ne kaip langus, mygtukus, bet pagal jų charakteristiką. Kiekvienas UIML failas yra padalijamas į keturias sekcijas: struktūra, stilius, turinys ir elgsena. Pagal šias sekcijas atlikus transformaciją, gauname vartotojo sąsają. UIML pagrindas yra XML kalba.

Kadangi UIML nesirūpina elementų sąsajoje išdėstymu, o tai yra svarbu, buvo sukurtas papildymas DI-UIML, nuo įrenginio nepriklausoma UIML (Device Independent UIML). Elementų išdėstymu rūpinasi algoritmas.

Kol kas ši programavimo kalba negali būti vadinama universalia. Jos realizaciją galime rasti .NET platformoje C# kalbos vartotojo sąsajos aprašyme. UIML kalboje slypi dar daug galimybių todėl ji gali būti plačiai naudojama ateityje [6].

2.3.2 Nuo platformos priklausomos aprašymo kalbos

Šio tipo kalbos vis dar abstrakčiai aprašo sąsajos elementus, tačiau negali būti naudojamos be papildomų programavimo kalbų komponentų. Dažniausiai šie komponentai pateikiami įvairių programavimo aplinkų.

2.3.2.1 XUL – vartotojo sąsajos aprašymo kalba

XML vartotojo sąsajos kalba (XML User Interface Language) sukurta Mozilla tipo interneto naršyklėms kaip GUI aprašymo kalba, kuri nepriklauso nuo platformos ir realizacijos kalbos.

XUL taikomosios programos negalimos be papildomų programavimo kalbų naudojimo. JavaScript yra naudojama dinamiškumui Web sąsajoje. Susiejus XUL su CATHEDRAL (dokumentų objektų modeliui) galima panaudoti standartizuotas struktūras XML dokumentų pavidalu, kartu su JavaScript modifikuojant XUL sąsają taikomosios programos vykdymo metu. Atvaizdavimo variantai nustatomi naudojant CSS (Cascading Style Sheets) failus. Apjungus

visus šiuos ir kitus komponentus (XPCOM, XBL ir kt.) gaunama XPFE nuo platformos nepriklausoma (Cross Platform Front End) architektūra, kurios pavyzdys yra Mozilla naršyklių Gecko variklis. XUL sąsajos be XPFE palaikymo yra labai retas atvejis [5].

2.3.2.2 XAML – vartotojo sąsajos aprašymo kalba

Po sėkmingo XUL kalbą naudojančių taikomųjų programų sukūrimo buvo pristatyta panaši kalba, pavadinta XAML (Extensible Application Markup Language). Ji orientuota į Microsoft Windows platformą ir tik sąlyginai gali būti naudojama kitose platformose. Planuotas nepriklausomumas nuo operacinės sistemos greičiausiai nebus pasiektas. Dėl daugybės panašumų į XUL ši kalba paminėta tik dėl to, kad toks formatas egzistuoja [8].

2.3.3 Įrankių komplektams skirti formatai

Paskutinė formatų grupė apibrėžia tuos formatus, kurie sukurti būtent konkrečiai bibliotekai, dažniausiai vėliau po jos sukūrimo, kaip funkcionalumo papildymai.

2.3.3.1 Qt UI – grafinės sąsajos aprašymo formatas

Formatas paremtas XML kalba, kuris aprašo taikomosios programos langus, meniu, įrankių juostas, statuso juostas, dialogo langus skirtingais variantais. Šio formato aprašyme leidžiama pateikti išdėstymo informaciją ir elementų pozicionavimą pikselių tikslumu. Taip pat galima nurodyti metainformaciją apie autorių, komentarus ir kt. Dinamiškumas pasiekiamas signalų ir gavėjų (Signals and Slots) aprašymo pagalba. Specialaus kompiliatoriaus pagalba šio formato aprašą galima gauti kaip C++ klasių realizaciją [1],[11],[15].

2.3.3.2 Glade – grafinės sąsajos aprašymo formatas

Kaip ir Qt UI, Glade taip pat yra XML kalba paremtas formatas. Jis tik nežymiai skiriasi nuo Qt UI. Gal būt didžiausi skirtumai yra signalų valdyme, kadangi šis formatas neleidžia atlikti dinamiškų veiksmų dialogo ribose [11],[15].

2.3.3.3 XSWT – Java programavimo kalbai skirtas grafinės sąsajos elementų aprašymo formatas

Java programavimo kalbai skirtas GUI elementų aprašymo formatas. Jis pasirodė po Toolkit SWT (Standard Widget Toolkit) vartotojo sąsajos įrankių rinkinio. Toolkit SWT – atviro kodo bazė skirta grafinės vartotojo sąsajos kūrimui Java kalba. Šis rinkinys sukurtas kaip alternatyva standartiniams Java įrankių rinkiniams (AWT, Swing). Norint atvaizduoti GUI elementus, SWT kreipiasi tiesiai į operacinės sistemos GUI bibliotekas naudojant JNI (Java Native Interface). Programos naudojančios SWT gali būti naudojamos įvairiose platformose, tačiau programinio kodo realizacija kiekvienoje platformoje skirsis, nors programavimo kalba visur bus Java [12].

2.3.3.4 RIB – Java programavimo kalbai skirtas grafinės sąsajos elementų aprašymo formatas

XSWT alternatyva RIB (Reflexive Interface Builder) turi keletą patobulinimų. Skirtingai nei XSWT, RIB palaiko tiek AWT, tiek Swing Java įrankius, kas suteikia aukštesnį abstrakcijos lygį [12].

2.3.4 Vartotojo sąsajos generavimo įrankiai

2.3.4.1 KWSDL – KDE aplinkai skirtas tinklo paslaugos grafinės sąsajos generavimo įrankis

Su KWSDL yra įmanoma generuoti web serviso parametrų reikšmių įvedimo dialogus. Sąsaja gali būti sugeneruota kaskart atliekant užklausą (Naudojant Qt XML GUI) arba sugeneruojama kompiliuojant su įrankiu kwsdl_compiler, kai transformuojama WSDL specifikacija. Tai leidžia taikomosioms programoms dinamiškai prisitaikyti prie pasikeitimų serverio pusėje. KWSDL yra kuriamas Unix platformai skirtoje KDE aplikoje [15].

2.3.4.2 Kung – KDE aplinkai skirta tinklo paslaugos grafinės sąsajos generavimo taikomoji programa

KWSDL principu pagrįsta darbastalio (desktop) programa, kuri leidžia dinamiškai naršyti po web servisų metodus. Ši programa palaiko paprastus web servisus (WSDL) įskaitant operacijų surinkimą, bendravimą (WSCL) ir, ateityje, žadama teikti GUI generavimą paremtą WSGUI [15].

2.3.5 Sąsajos aprašymo formatų ir generavimo įrankių analizės apibendrinimas

Išanalizuotų grafinės sąsajos aprašymo metodų ir formatų sąrašas pateikiamas 2-oje lentelėje. Pasirinkta XForms grafinės sąsajos aprašymo technologija, kadangi ji palaiko paprastą duomenų validaciją, palaiko SOAP protokolą ir taip žymiai supaprastina papildomos informacijos ir failų poreikį generuojant grafinę sąsają pasirinktu metodu (WSGUI).

Formatas	Loginiai elementai	Imperatyvūs elementai
XForms	Skaičiavimai ir veiksmai	Nėra
UIML	Nėra	Nėra
XUL	Nėra	Javascript
XAML	Nežinoma	Nežinoma
Qt UI	Signalai ir gavėjai	Nėra
Glade	Tik signalai	Nėra
XSWT	Nėra	Nėra
RIB	Apribojimai	Java/Python/Ruby

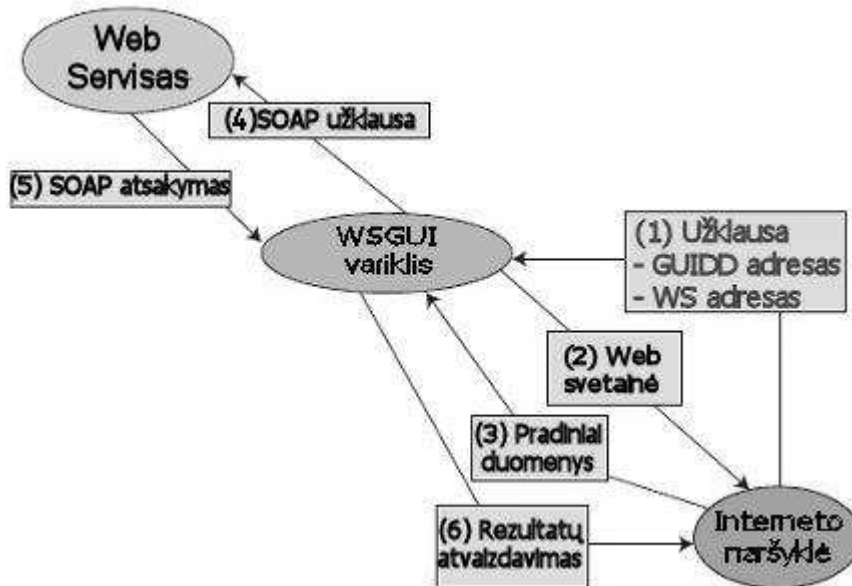
1 lentelė. Abstraktūs vartotojo sąsajos aprašymo formatai

Problemoms spręsti geriausi buvo pasirinkti aprašymo formatą nenaudojantį imperatyvių elementų, kadangi jie žymiai apsunkintų sąsajos generavimą. Taip pat netinka nuo platformos priklausančios technologijos.

2.4 Architektūros ir galimų įgyvendinimo priemonių pasirinkimas

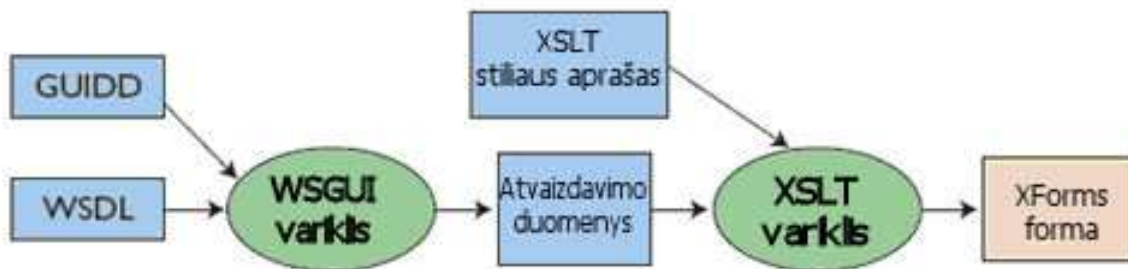
Šio darbo tikslams įgyvendinti pasirinkta WSGUI schema. Norint sugeneruoti grafinę sąsają web servisui pagal WSGUI schemą, reikia dvejų XML dokumentų. Pirmasis yra GUI realizacijos aprašymas (GUIDD – GUI Deployment Descriptor), kuris saugo abstraktų vartotojo sąsajos aprašymą. Antrasis yra XSLT transformacijos aprašymas, kuris iš abstraktaus aprašymo padeda sukurti konkrečią GUI sąsają. Galima susieti keletą transformacijos failų su vienu GUIDD dokumentu, taip suteikiant galimybę gauti skirtingas realizacijas pagal tą patį aprašymą.

XML dokumentui pritaikius XSLT transformaciją galima matyti duomenis draugišku vartotojui pavidalu, o pagal GUIDD gauname konkretų atvaizdavimą. Šie du dokumentai leidžia varikliui generuoti WSGUI [1], [16].



2 pav. WSGUI sistemos architektūra

Antrajame paveikslėlyje matome, kaip vyksta bendravimas su web servisu WSGUI sistemoje. Vartotojas nukreipia interneto naršyklę į WSGUI variklį, paduodamas GUIDD nuorodą ir pageidaujama pradinį puslapį. Antrame žingsnyje WSGUI variklis generuoja pradinę formą. Trečias žingsnis - vartotojas naudodamas sugeneruotą formą įveda informaciją ir siunčia duomenis naudodamas HTTP POST (arba GET) metodu. Sudėtingesnėse formose, pavyzdžiui tokiose, kur reikia įvesti sąrašą duomenų, gali reikalauti keleto POST arba GET metodų kvietimų. Jeigu duomenų padavimas ne galutinis, variklis gražina modifikuotą formą. Kada vartotojas baigia vesti duomenis, kviečiamas web servisas (ketvirtas žingsnis). Web servisas gražina informaciją (penktas žingsnis), kurią WSGUI variklis pateikia interneto naršyklei XForms formatu (šeštas žingsnis).



3 pav. HTML generavimas WSGUI sistemoje

Trečiajame paveikslėlyje pateikiama schema, kaip generuojamas HTML kodas. Pagal GUIDD ir WSDL aprašymus WSGUI variklis sugeneruoja abstrakčią vartotojo sąsajos informaciją, kuri pritaikius XSLT transformaciją perdaroma į XForms formą [1].

2.5 Darbo tikslas ir kokybės kriterijai

Kadangi analizė rodo, kad vartotojo sąsajos generavimas yra aktualus uždavinys, šio darbo tikslas – automatinis vartotojo grafinės sąsajos generavimas tinklo paslaugai, turint jos WSDL aprašymą.

Darbo rezultatui keliami kokybiniai kriterijai:

- Siekiama sukurti sistemą, kuri interneto svetainėje generuotų eiliniam vartotojui aiškia, suprantamą grafinę sąsają pagal tinklo paslaugų specifikaciją;
- Naudodamas šią sąsają, vartotojas gali siųsti užklausas tinklo paslaugoms ir gauti rezultatus;
- Sugeneruota grafinė sąsaja turi būti matoma su populiariausiom šiuo metu interneto naršyklėm;
- Sistema turi užtikrinti duomenų saugumą ir susidoroti su sudėtingais duomenų tipais;
- Sąsajos generavimo realizacija turi būti tokia, kad ją nesunkiai būtų galima pritaikyti įvairioms kalboms bei įvairioms vartotojo sąsajos realizavimo technologijoms.

Siekiama praktinė nauda:

- Didesnis svetainių lankstumas, dinamiškumas, greitesnis atnaujinimas, keičiantis tinklo paslaugoms.
- Mažiau darbo sistemų administratoriui, programuotojui.

2.6 Reikalavimai duomenų struktūroms

Norint automatiškai sugeneruoti grafinę sąsają tinklo paslaugai, būtini yra du failai. Pirmasis yra WSDL tinklo paslaugos aprašas, sudarytas pagal standartinę WSDL failams naudojamą XSD schemą. Antrasis – tai GUIDD aprašas. Jo struktūra aprašyta tolimesniuose šio darbo skyriuose. Turint šiuos du failus, galima generuoti abstrakčią tinklo paslaugos grafinę sąsają.

Papildomas failas yra XSLT transformacija, kurios pagalba iš abstrakčios sąsajos sukuriama grafinė sąsaja pasirinktos technologijos forma. Norint, kad sąsaja būtų malonesnė vartotojui, dar gali būti pritaikytas CSS stiliaus aprašymas. CSS failo adresas įrašomas GUIDD apraše.

Naudojami duomenų tipai priklauso nuo pačių web servisų. Pasirinkus XForms standartą, vartotojo sąsajos generavimui dalis vartotojų įvedamų duomenų validacijos gali būti atliekama duomenų įvedimo metu. Pagrindinis duomenų teisingumo tikrinimas turėtų būti atliekamas web serviso teikiamų metodų logikoje.

2.7 Rizikos faktorių analizė

Galimi rizikos faktoriai:

Sudėtingų duomenų tipų naudojimas web serviso metodų parametruose. Tokiu atveju paprasčiausias sprendimas būtų skaidyti web serviso parametrus į keletą paprastesnių. Jeigu to padaryti negalima, iškyla grėsmės, kad bus negalima sugeneruoti patogios ir suprantamos sąsajos, ar kad sugeneravus nepilną sąsają bus prarasti svarbūs duomenys ar įsivels klaidos.

Galimybių ar elementų trūkumas pasirinktuose įrankiuose, aprašymų formatuose, programavimo kalbose. Iškyla grėsmė, kad bus sugeneruota nepakankamai aiški ar nepilna vartotojo sąsaja.

Duomenų praradimas nutrūkus ryšiui su tinklo paslaugų serveriu.

2.8 Analizės išvados

Automatinių grafinės sąsajos generavimo metodų analizė parodė, kad pats funkcionaliausias ir labiausiai poreikius atitinkantis yra WSGUI grafinės sąsajos generavimo metodas. Tai grafinės sąsajos pagal WSDL ir abstraktų GUIDD aprašą generavimo metodas, pritaikytas būtent tinklo paslaugų sąsajai generuoti.

Išanalizavus įvairias grafinių vartotojo sąsajų technologijas ir aprašymo formatus, nuspręsta generuoti GUI tipo grafinę vartotojo sąsają, kurios atvaizdavimas interneto naršyklėse bus realizuotas XForms tipo formomis.

3. Automatinio vartotojo sąsajos generavimo reikalavimų specifikacija

Šioje dalyje nustatomi funkciniai ir nefunkciniai reikalavimai pasirinktos problemos sprendimui, apibrėžiamas automatinio vartotojo sąsajos generavimo tinklo paslaugoms funkcionalumas.

3.1 Funkciniai reikalavimai

1-oje lentelėje pateikiamas projektuojamos sistemos funkcinų reikalavimų sąrašas. Kiekvienas reikalavimas papildomai detalizuotas. Funkciniai reikalavimai kuriamai sistemai apibrėžia, kokios veiklos funkcijos turi būti kompiuterizuotos.

Reikalavimas	Aprašymas
Gauti tinklo paslaugų sąrašą	Vartotojui atlikus užklausą sistema turi gražinti prieinamų tinklo paslaugų sąrašą. Sąrašas gali būti pateikiamas paprasta HTML forma.
Gauti tinklo paslaugos operacijų sąrašą	Pasirinkęs tinklo paslaugą, vartotojas turi matyti tinklo paslaugos teikiamų operacijų sąrašą. Operacijų sąrašas saugomas WSDL apraše. Pateiktas sąrašas gali būti paprasta HTML forma arba XForms forma.
Pagal tinklo paslaugos operacijos aprašą generuoti vartotojo sąsają.	Pagal WSDL ir GUIDD aprašus pritaikius XSLT transformacijas generuojama XForms tipo vartotojo sąsaja
Vykdyti tinklo paslaugos operaciją	Vartotojui sugeneruotoje sąsajoje suvedus operacijai reikalingus duomenys turi būti sugeneruojama SOAP užklausa ir įvykdoma tinklo paslaugos operacija.
Atlikti duomenų įvedimo validaciją	Dalis vartotojo įvedamų duomenų validacijos gali būti atliekama duomenų įvedimo metu panaudojant Xforms komponentų galimybes. Pagrindinis duomenų teisingumo tikrinimas turėtų būti atliekamas web serviso teikiamų metodų logikoje.
Turi būti galimybė įvesti duomenis keletu žingsnių.	Jeigu operacijai įvykdyti reikalinga daug pradinių duomenų, duomenys gali būti paskirstyti į keletą įvedimo žingsnių ir tik suvedus visų žingsnių duomenis generuojama užklausa.
Gauti įvykdytos operacijos rezultatus.	Pabaigusi vykdyti tinklo paslaugos operaciją sistema turi gražinti rezultatą suprantama vartotojui forma.

1 lentelė. Funkciniai reikalavimai

3.2 Nefunkciniai reikalavimai

Nefunkciniai reikalavimai kuriamai sistemai nustato, kaip veiklos funkcijos turi būti kompiuterizuotos. Nefunkciniai reikalavimai apima reikalavimus, veikiančius funkcionalumą, patikimumą, patogumą, efektyvumą, priežiūros savybes, perkeliamumą.

Kuriamai sistemai keliami šie nefunkciniai reikalavimai:

- 1) Galimybė naudotis tinklo paslaugomis (web servais) nenaudojant specialios programos ar specialiai tam sukurtos svetainės. Web servisų metodai turėtų būti kviečiami tiesiogiai per sugeneruotą vartotojo sąsają.
- 2) Tinklo paslaugų metodų kvietimą reikia realizuoti per interneto naršyklės sąsają.
- 3) Reikalingas kuo didesnis vartotojo sąsajos generavimo sistemos lankstumas, kad pakitę web servisų metodai ar jų parametrai nesukeltų didelių problemų.
- 4) Daugiakalbiškumas. Vartotojui sąsaja gali būti generuojama jo pasirinkta kalba.
- 5) Turi būti galimas paprastų ir sudėtinių duomenų tipų apdorojimas

4. Automatinio grafinės sąsajos generavimo sistemos projektas

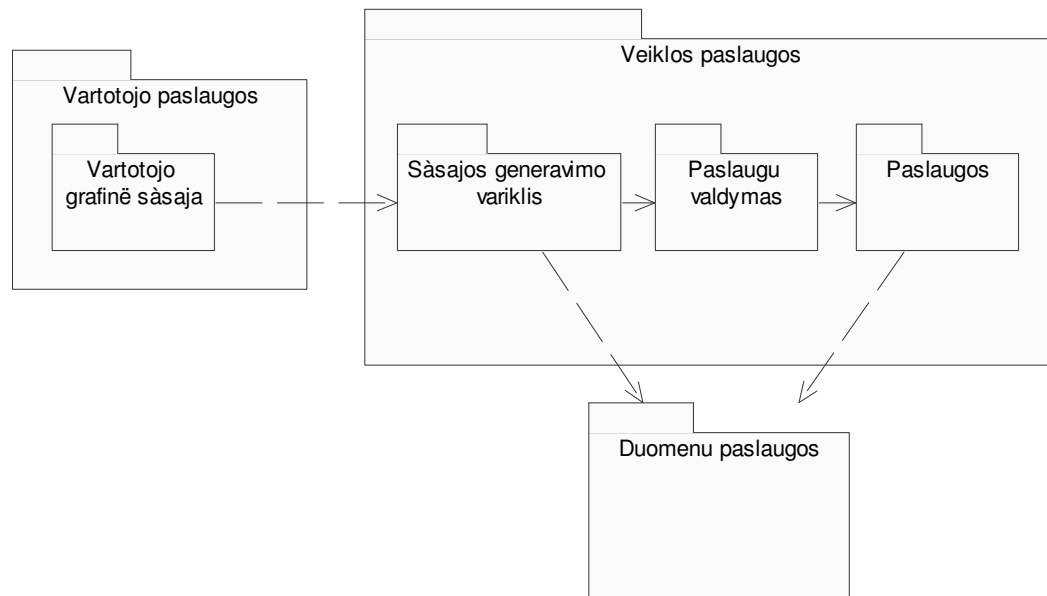
Šiame skyriuje pateikiamas kuriamos sistemos projektas, detalizuojantis, kaip bus realizuoti aprašyti reikalavimai.

Projekto tikslas – suprojektuoti automatinio vartotojo sąsajos generavimo pagal tinklo paslaugos specifikaciją sistemą, taikant CASE priemones.

Projekto modelyje apibrėžta kompiuterinės sistemos architektūra ir operacijų paskirstymas klasėms. Sistemos architektūra – sistemą sudarantys komponentai, jų tarpusavio sąsajos, išdėstymas vienas kito atžvilgiu. Loginė sistemos architektūra parodo kaip organizuoti sistemą sudarantys komponentai.

4.1 Sistemos architektūra

Vienas iš dažniausiai taikomų loginės architektūros variantų yra trijų lygių architektūra. Tokią architektūrą sudaro trys pagrindiniai lygiai – vartotojo interfeiso, veiklos ir duomenų.



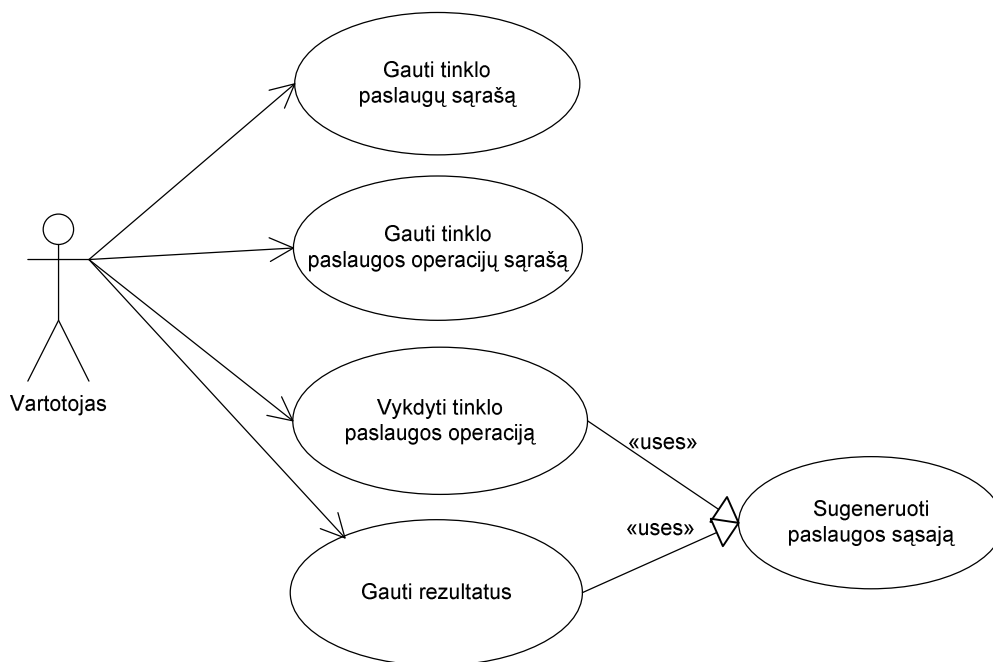
4 pav. Trijų lygių loginės architektūros schema

4-ame paveiksle matome sistemos architektūros schemą. Ją sudaro trys loginės sritys: vartotojo paslaugų, veiklos paslaugų ir duomenų paslaugų. Vartotojas sistemoje naudojami sugeneruota grafine sąsaja. Už sąsajos generavimą atsakingas sąsajos generavimo variklis esantis veiklos paslaugų lygyje. Variklis sąsajos generavimui naudoja tinklo paslaugos aprašą, gaunamą iš duomenų paslaugų lygio. Variklis, vartotojui nurodžius, gali vykdyti paslaugas.

Šioje dalyje pateikiamos konkrečios kuriamos sistemos projekto panaudojimo atvejų, sekų, komponentų ir klasių diagramos, pagal kurias bus vykdoma sistemos realizacija.

4.2 Panaudojimo atvejų diagrama

Panaudojimo atvejų diagramoje (5 pav.) vaizduojami veiksmai, kuriuos vartotojas gali atlikti naudodamasis sistema. Panaudojimo atvejai atitinka kuriamos sistemos funkcinius reikalavimus.

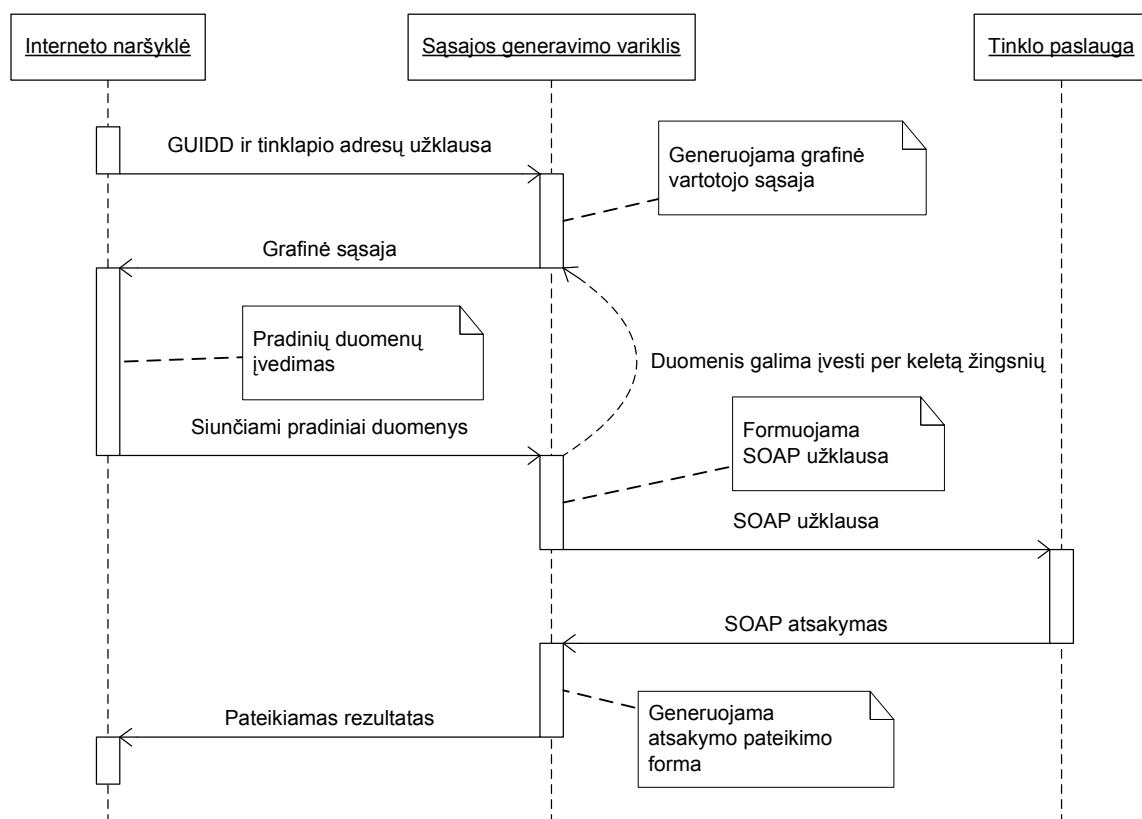


5 pav. Sistemos panaudojimo atvejai

Vartotojas, įvykdęs užklausą, turi gauti sistemos teikiamas tinklo paslaugas. Gavęs tinklo paslaugų sąrašą ir pasirinkęs iš jo konkrečią paslaugą, turi gauti pasirinktos paslaugos operacijų sąrašą. Vartotojui suteikiama galimybė įvykdyti pasirinktą operaciją, automatiškai sugeneruojant suprantamą grafinę sąsają. Po operacijos įvykdymo grafinėje sąsajoje pateikiami rezultatai.

4.3 Sekų diagrama

Sekų diagramoje, pavaizduotoje 6-ame paveiksle, pateikiama tipinė veiksmų seka, vartotojui norint įvykdyti tinklo paslaugos operaciją.



6 pav. Sąsajos generavimo sistemos sekų diagrama

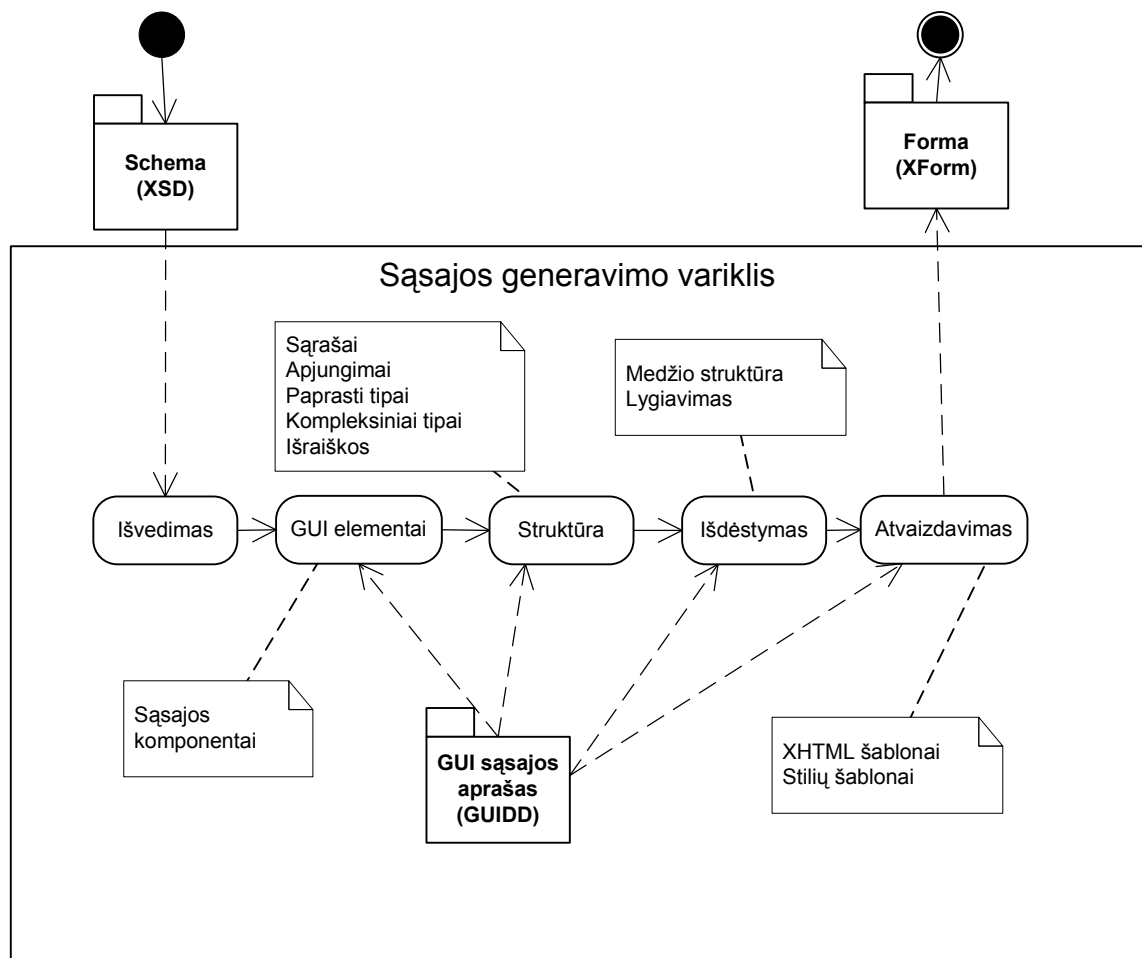
Vartotojui pasirinkus tinklo paslaugos operaciją, sąsajos generavimo varikliui pateikiama GUIDD aprašas. Pagal operacijos aprašą WSDL schemeje ir GUIDD schemą sugeneruojama grafinės sąsajos XML schema, kuri pritaikius XSLT transformaciją pateikia Xforms pavidalu.

Sekančiame žingsnyje vartotojas turi įvesti duomenis reikalingus sėkmingam operacijos įvykdymui. Priklausomai nuo GUIDD aprašo, vartotojo gali būti prašoma duomenis įvesti keletu etapų (Pavyzdžiui, pirmiausiai prašoma įvesti asmens duomenis, įvedus duomenis parodomas kitas dialogas, kuriame prašoma įvesti užsakomos prekės duomenis). Tik surinkus visus reikalingus duomenis, formuojama SOAP užklausa, kurios pagalba įvykdoma pasirinkta operacija.

Pasibaigui operacijos vykdymui, pateiktas rezultatas pritaikius atitinkamas XSLT transformacijas pateikiamas vartotojui.

4.4 Grafinės sąsajos generavimo variklio veiklos diagrama

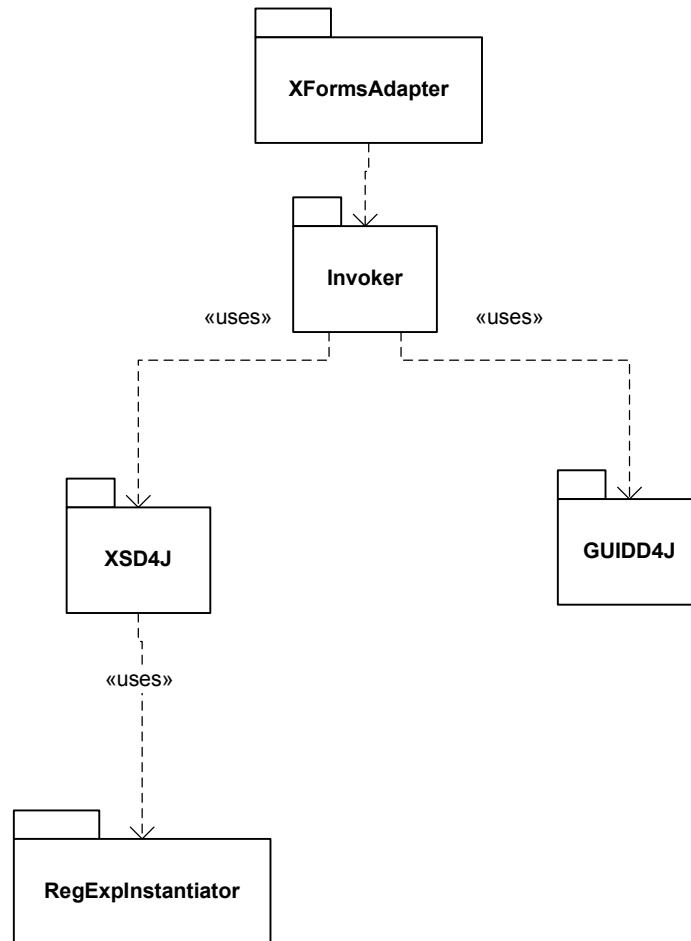
Grafinės sąsajos generavimo variklio procedūros matomos 8-ame paveiksle. Variklio veikimo principas pagrįstas WSGUI metodu aprašytu 2.10.1 ir 4.2.4.2 paragrafuose. Varikliui paduodamas tinklo paslaugos WSDL aprašas XSD schemas pavidalu. Komponentas išanalizuoja gautą WSDL schemą ir išveda pradinę reikalingą informaciją. Tolesnis veiksmas nustato kokie GUI elementai bus reikalingi. Sekančiame žingsnyje GUI elementams priskiriami konkretūs tipai ir atvaizdavimo forma. Turint šią informaciją, vykdomas elementų išdėstymo grafinėje sąsajoje algoritmas. Pabaigoje gavus GUI sąsajos struktūrą, pritaikoma XSLT transformacija ir gražinamas sugeneruotas rezultatas.



8 pav. Kuriamo sąsajos generavimo variklio procedūros

4.5 Komponentų diagrama

Komponentų diagramoje (9 pav.) pavaizduoti kuriamą sistemą sudarantys komponentai ir sąsajos tarp jų.



9 pav. Sistemos komponentų diagrama

Pagrindinis sąsajos generavimo variklio komponentas pavadintas „Invoker“ ir yra atsakingas už veiksmų koordinavimą. Jis naudoja XSD4J ir GUIDD4J komponentus, kurie apdoroja atitinkamai WSDL ir GUIDD failų turinį, reikalingą sąsajos generavimui. Jam padavus reikalingus duomenis kiekvienam žingsniui atlikti kviečiamas atitinkamas komponentas.

XSD4J yra WSDL schemas analizės komponentas skirtas tinklo paslaugos aprašo schemas analizei ir išvadų pateikimui.

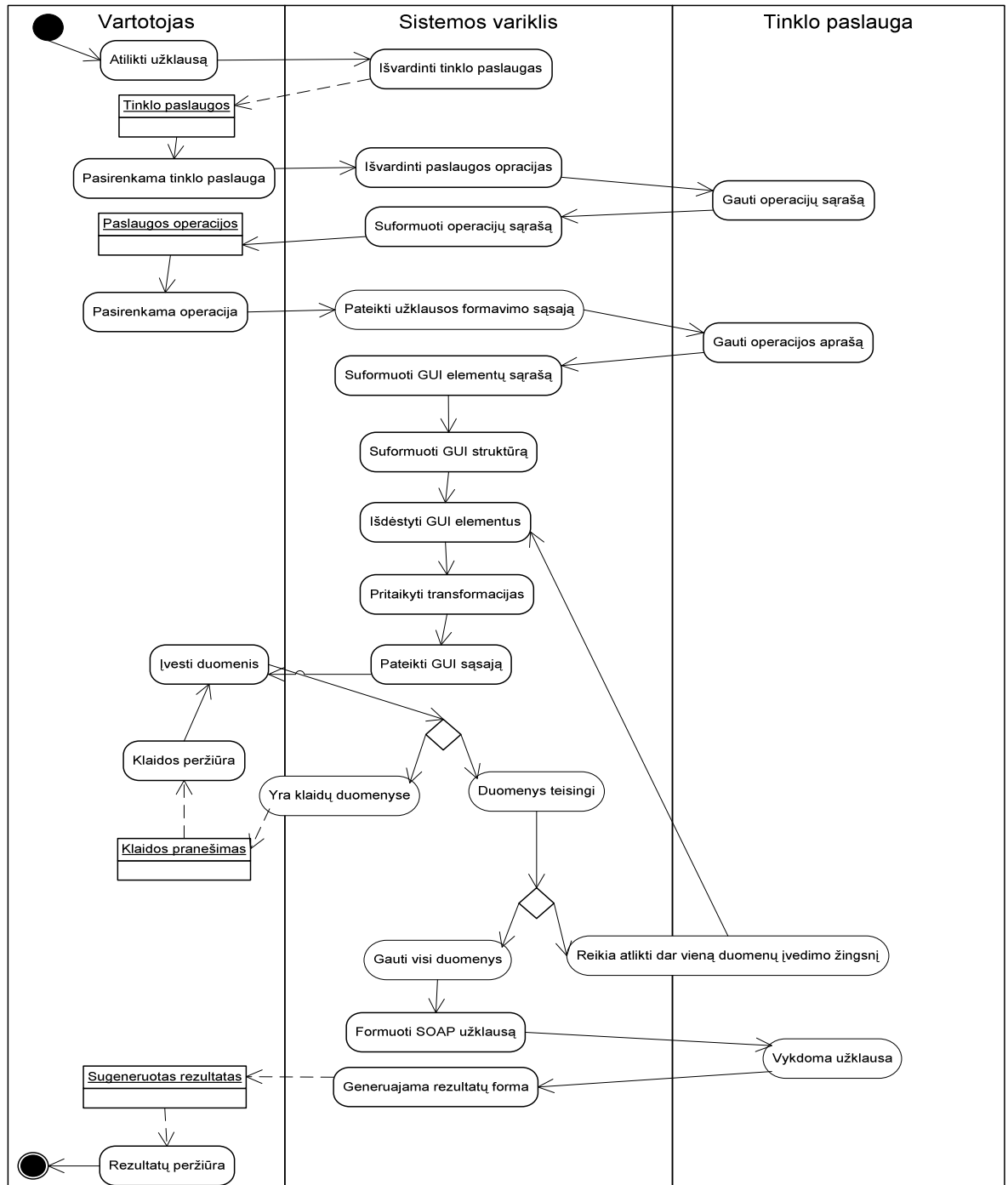
Pagal WSDL analizės komponento duomenis vykdomas GUI elementų kūrimo, struktūros sudarymo ir elementų išdėstymo komponentas GUIDD4J.

Pabaigoje, gavęs visą reikalingą informaciją, variklis suformuoja abstrakčią grafinę sąsają. Variklis pasiekiamas per transformacijų pritaikymo komponentą XFormsAdapter, kuris suformuoja galutinį rezultatą ir pateikia jį XForms tipo sąsają. XFormsAdapter klasė naudojama per sąsają IAdapter, todėl reikalui esant gali būti pakeista bet kokia kita klase, kuri galėtų generuoti kitokio tipo grafinę sąsają (XUL, XSWT ir kt.).

4.6 *Sistemos elgsenos modelis*

Sistemos elgsenos modelis vaizduoja vartotojo, sistemos ir tinklo paslaugos komunikavimą.

Vartotojas, atlikęs užklausą, gauna tinklo sąrašą. Pasirinkęs konkrečią tinklo paslaugą, pagal WSDL aprašą gauna operacijų sąrašą.



15 pav. Kuriamos sistemos elgsenos modelis

Pasirinkęs konkrečią operaciją vartotojas kviečia grafines sąsajos generavimo variklį, kuris pagal tinklo paslaugos specifikaciją atlieka sąsajos generavimą

Pateiktoje grafinėje sąsajoje vienu arba keliais etapais vartotojas turi suvesti operacijai reikalingus duomenis. Jeigu duomenys klaidingi, sugeneruojamas klaidos pranešimas. Praėjus visus duomenų suvedimo etapus generuojama SOAP užklausa, kurios pagalba vykdoma tinklo paslaugos operacija.

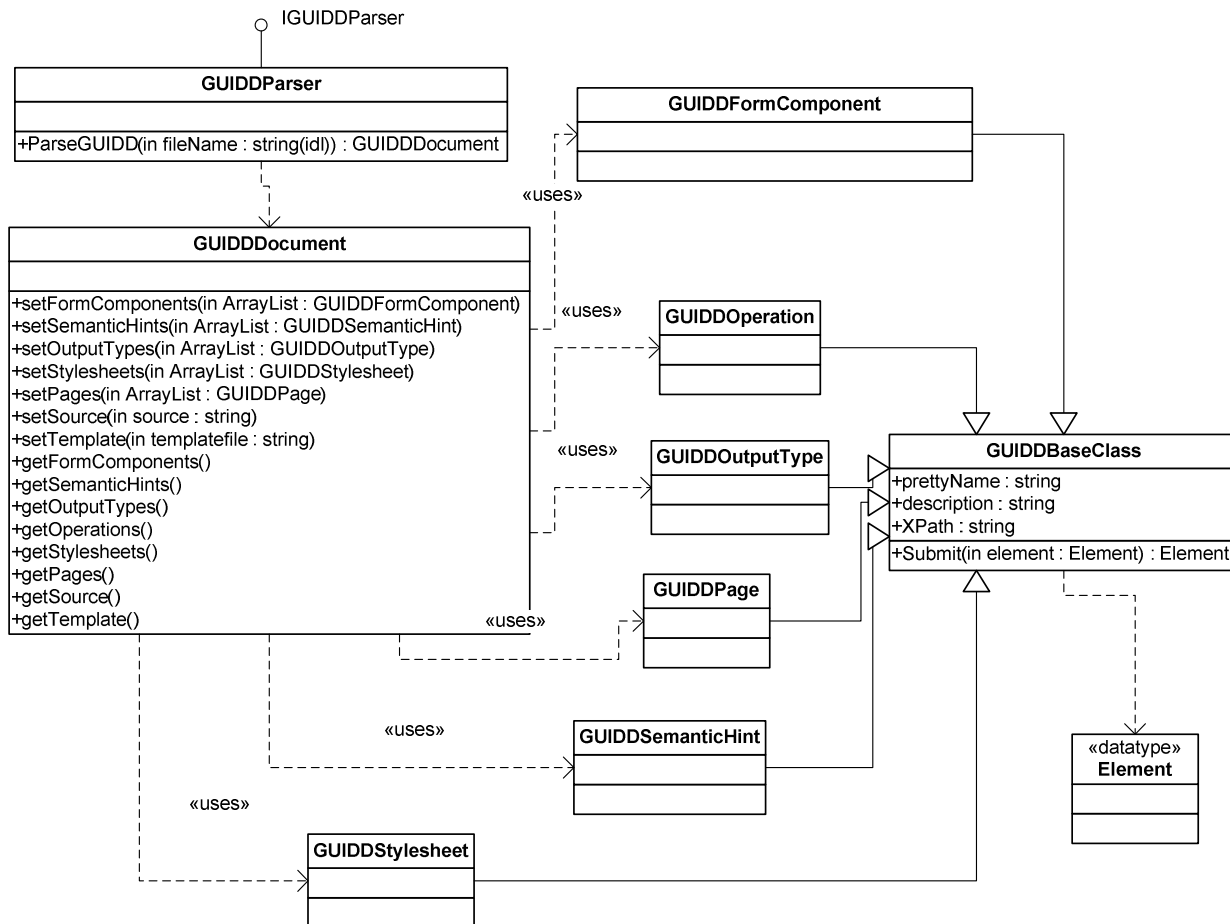
Gautas rezultatas suformuojamas ir pateikiamas vartotojui grafine forma.

4.7 Klasių diagramos

Šioje dalyje pateikiamos komponentų diagramoje pavaizduotų atskirų komponentų klasių diagramos su klasių aprašymais.

GUIDD4J klasių diagrama

GUIDD failai saugo grafinės vartotojo sąsajos abstraktų aprašą, kuris leidžia iš XML schemas kurti grafinę sąsają. Tai yra pagrindinis WSGUI metodu generuojamos sąsajos failas. GUIDD4J biblioteka leidžia apdoroti GUIDD failus, paimti iš jų visą svarbią meta informaciją ir pasiekti XML objektus atitinkančius XForms komponentus. Ši biblioteka naudojama kartu su XSD4J biblioteka. GUIDD4J komponento klasių diagrama pateikta 10-ame paveikslėlyje.



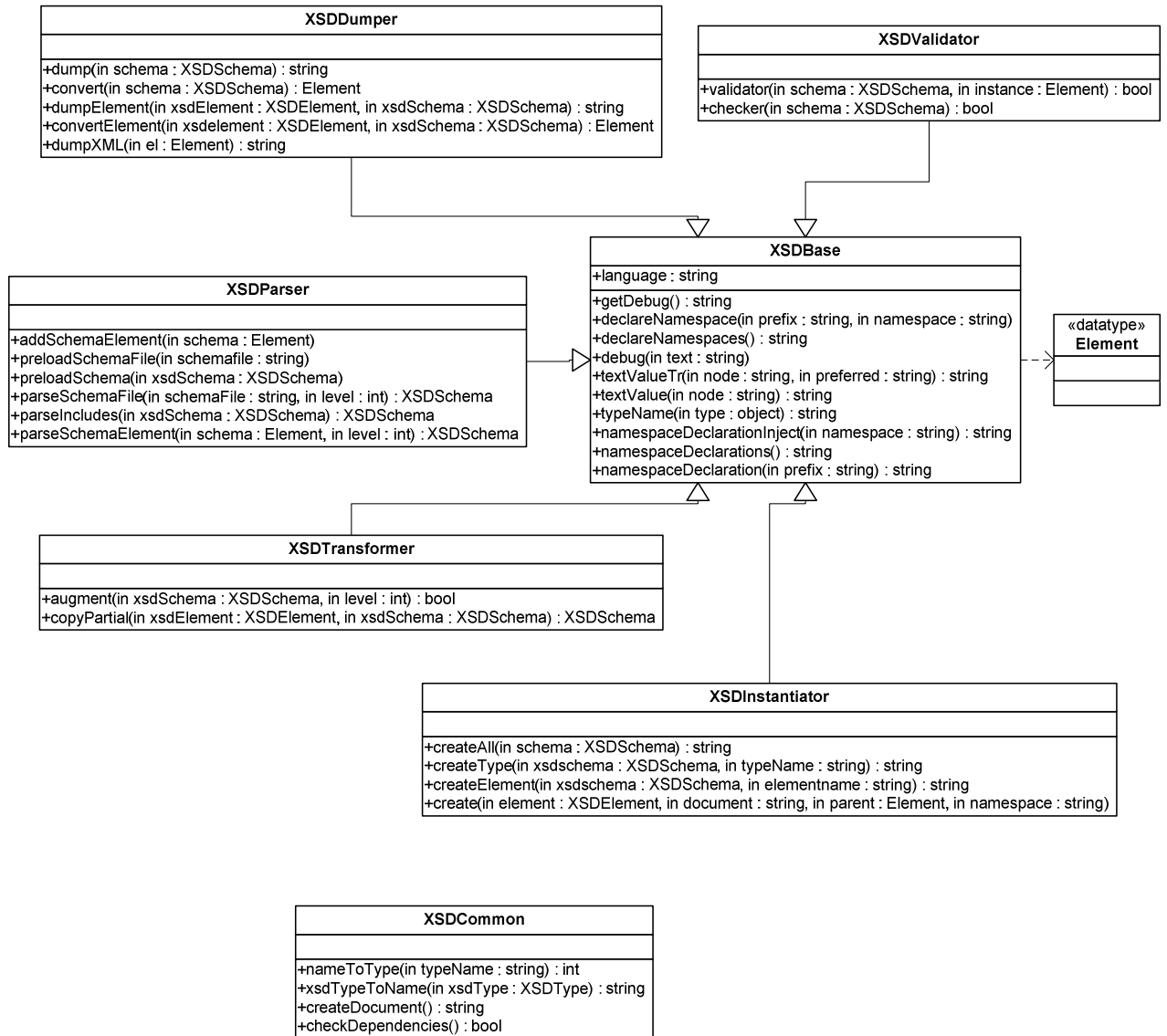
10 pav. GUIDD4J komponento klasių diagrama

GUIDD4J biblioteka leidžia taikomajai programai apdoroti GUIDD formato failus. GUIDD failai saugo WSDL failo paaiškinimus. Šis komponentas palengvina WSDL duomenimis paremtos grafinės sąsajos generavimą.

GUIDDParser objektas yra naudojamas GUIDDDocument objekto, reprezentuojančio abstraktaus grafinės sąsajos aprašymo failo turinį, sukūrimui. GUIDDDocument objektas gaunamas iškviečiant parseGUIDD metodą ir savo ruožtu turi GUIDDOperation, GUIDDOutputTypes ir GUIDDFormComponent objektus, kurie naudojami WSDL grafinės sąsajos papildymui galimomis tinklo paslaugos operacijomis, bei SOAP pranešimų turinio atvaizdavimui vartotojui suprantama forma.

XSD4J komponento klasių diagrama

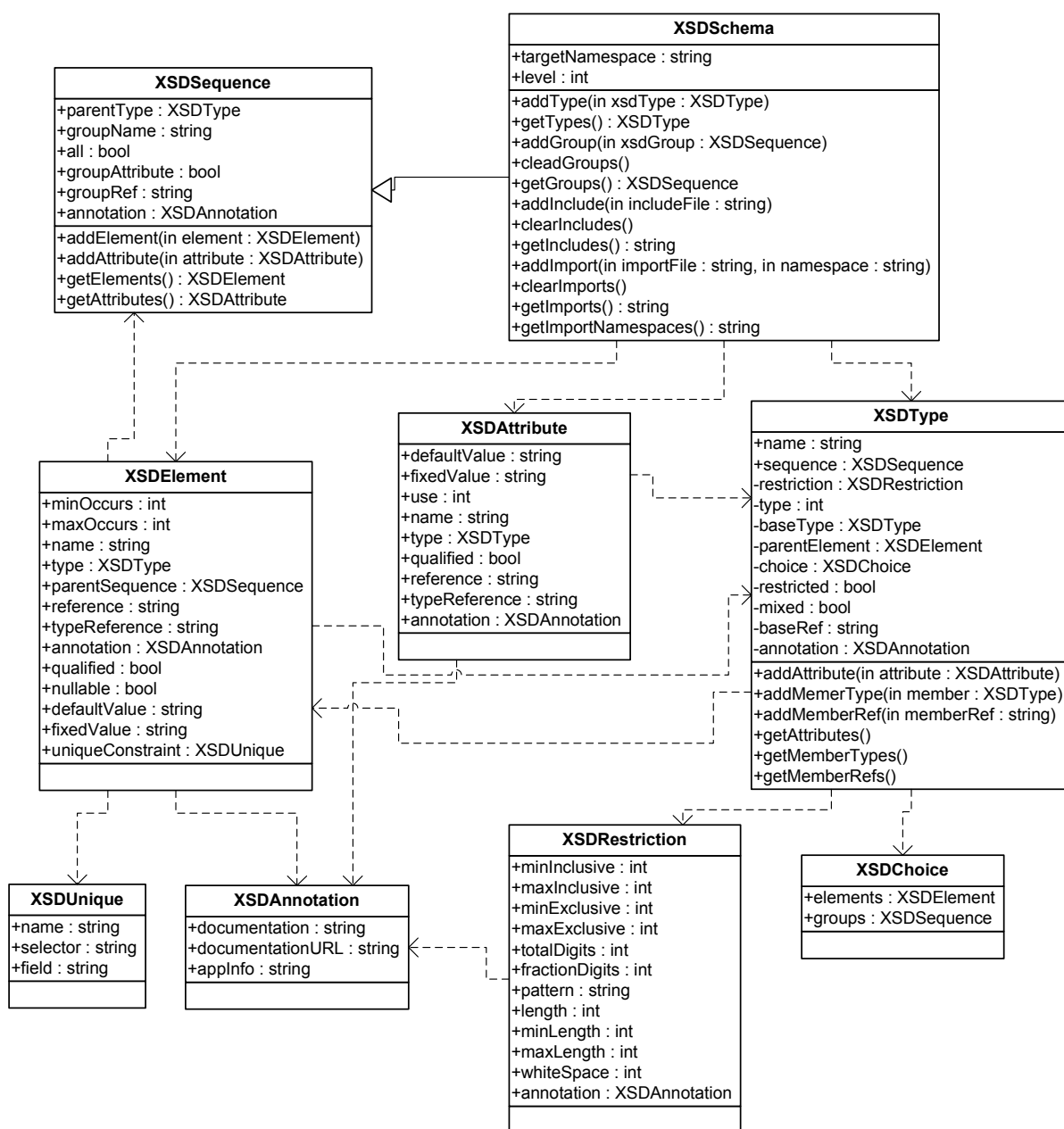
Biblioteka XSD4J yra naudojama XML schemų aprašų (XSD) nuskaitymui. Jos klasių diagrama pateikta 11-ame paveiksle. Schemas failo nuskaitymui yra naudojamas XSDParser objektas. Metodas parseSchemaFile nuskaitymo schemą ir grąžina XSDSchema objektą tolimesniam naudojimui, arba grąžina klaidos pranešimą, jeigu scheme buvo klaidų. Klaidos atveju getDebug metodo pagalba galima peržvelgti klaidos pranešimą.



11 pav. XSD4J komponento klasių diagrama

XSDSchema objektas saugo XSDElement, XSDAttribute ir XSDType objektų sąrašus. XSDSchema klasė ir jos ryšiai su kitais duomenų objektais yra pavaizduoda 12-ame paveiksle.

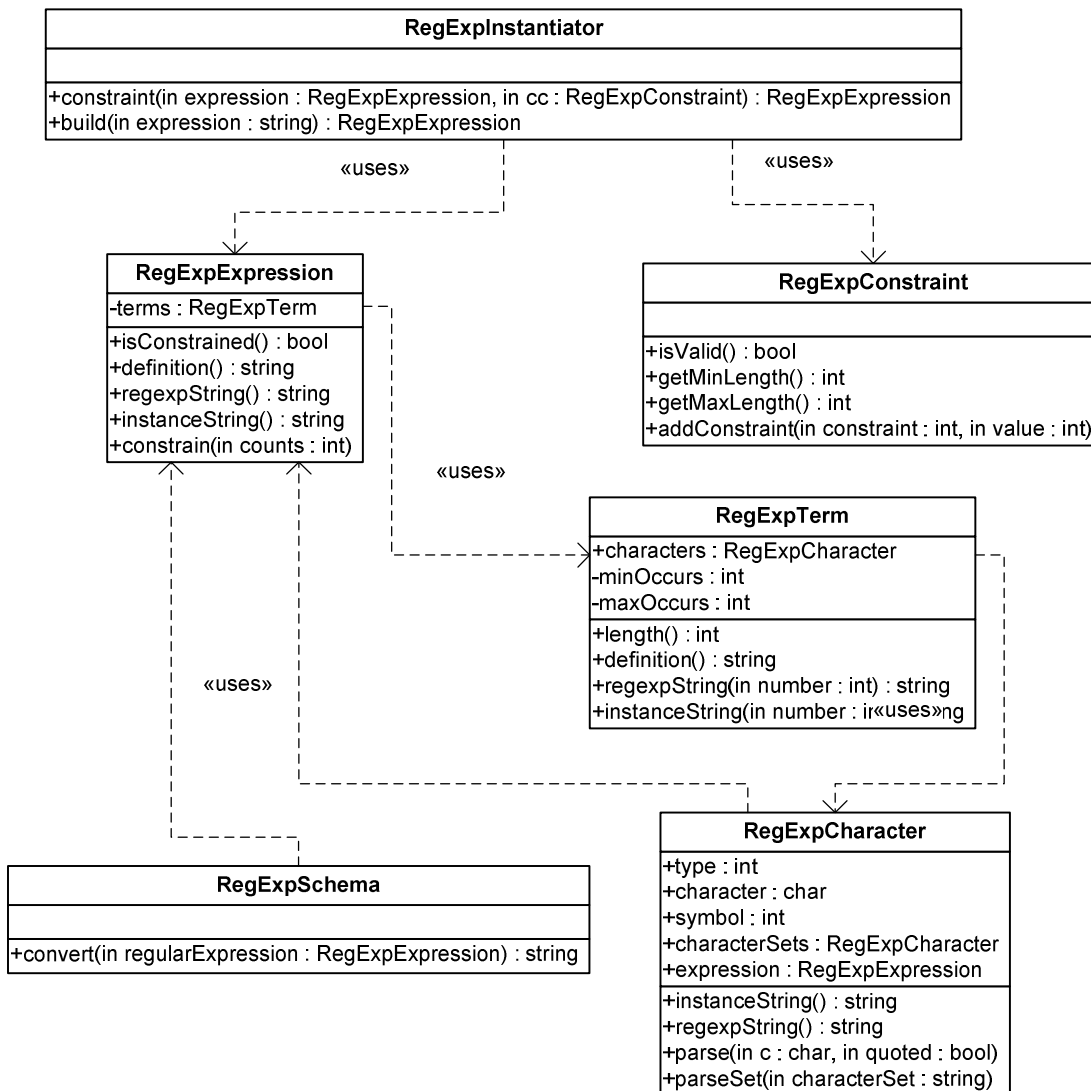
XSDSchema atvaizduoja medžio tipo objektų hierarchinę struktūrą. Jis gali būti naudojamas kaip bet kuris kitas XSDSequence objektas, gaunamas kviečiant getElement metoda. Metodo getType pagalba gaunamas XSDType objektas. Jeigu jis reprezentuoja sudėtinį tipą, tada jam prikabinama nauja XSDSequence objektų seka. Jeigu tai yra paprastas tipas, tokiu atveju galima gauti jo reikšmių apribojimus.



12 pav. XSDSchema klasė ir jos ryšiai

RegExpInstantiator komponento klasių diagrama

RegExpInstantiator yra nedidelė klasių biblioteka organizuojanti darbą su RegEx klase. RegEx (Regular Expressions) klasė palengvina dokumentų nuskaitymą. RegExpInstantiator saugo apribojimus, išraiškas ir metodus, kad būtų galima parastai konvertuoti išraiškas į XML schemas.



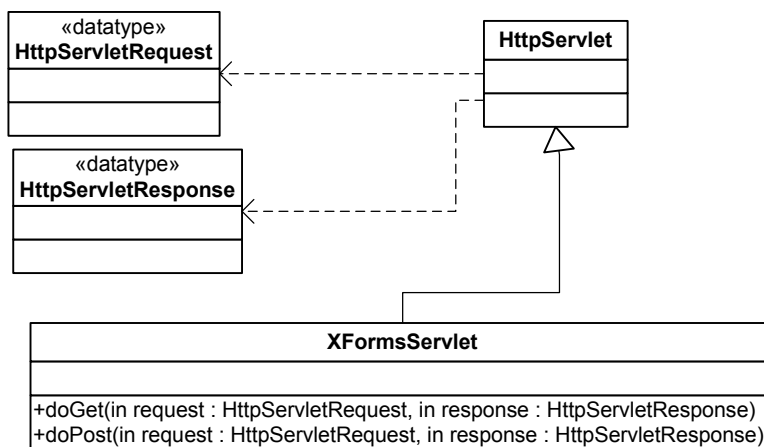
13 pav. RegExpInstantiator komponento klasių diagrama

RegExpParser objektas grąžina RegExpExpression objektą, kuris susideda iš RegExpTerm objektų masyvo. Jame kiekviena klasė saugo kiekybinę informaciją ir turi sąrašą RegExpCharacter klasių. RegExpCharacter klasės gali būti simboliai, arba simbolių rinkiniai.

XSD4J klasė naudoja RegExpInstantiator klasę pradiniais duomenimis sukurti.

XFormsServlet klasių diagrama

XFormsServlet yra pagalbinis komponentas darbui su XForms interneto svetainėmis ir tinklo paslaugomis.

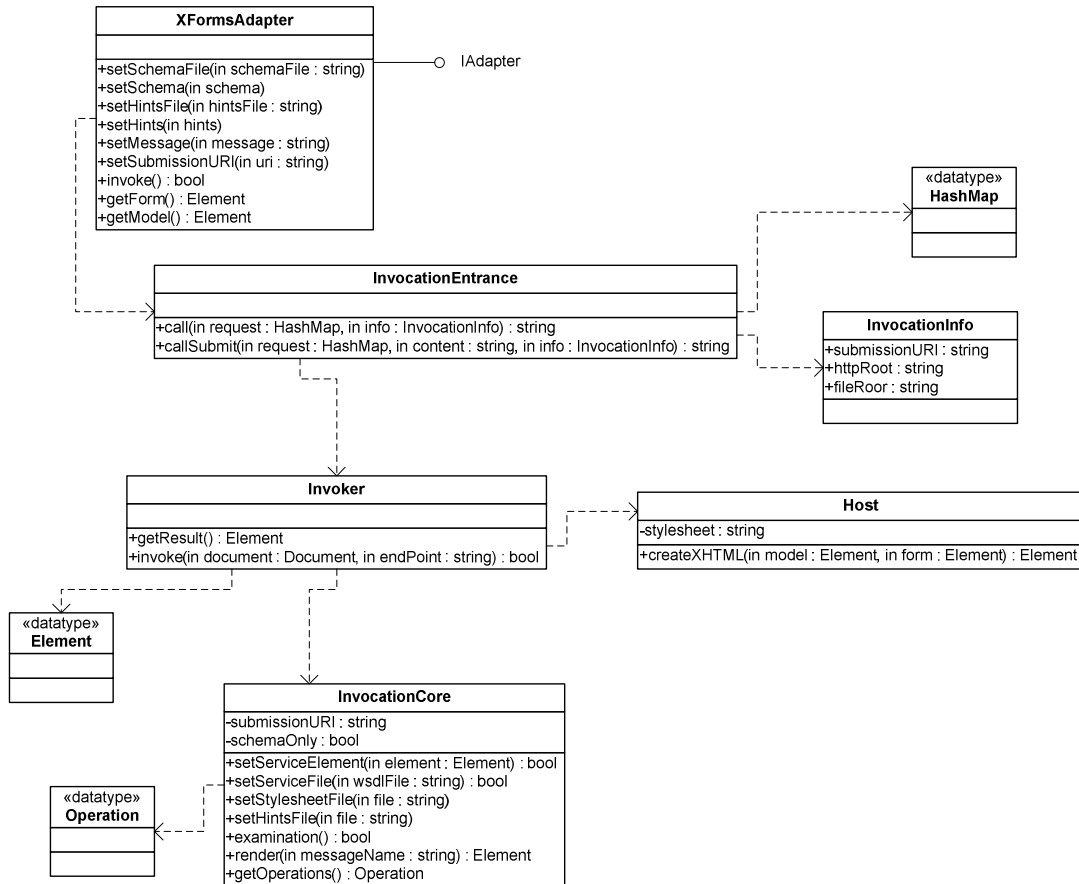


14 pav. XForms Java Servlet klasių diagrama

XFormsAdapter ir Invoker klasių diagramos

XFormsAdapter klasė pasiekama per Iadapter interfeisą. Ši klasė XForms grafinės sąsajos duomenis gali perduoti Invoker klasei ir iš šios klasės gautą abstraktų grafinės sąsajos modelį transformuoti į XForms formą. XFormsAdapter klasė gali būti pakeista kita klase, kuri naudotų skirtingą negu XForms technologiją grafinės sąsajos atvaizdavimui.

Invoker klasė iš XFormsAdapter gautus duomenis apdoroja, suformuoja SOAP pranešimą ir gali iškviesti tinklo paslaugą. Sąsajos, pradinių duomenų ir rezultatų apdorojimui naudojami XSD4J ir GUIDD4J komponentai. Invoker ir XformsAdapter komponentų klasių diagrama pavaizduota 15-ame paveiksle.



15 pav. XFormsAdapter ir Invoker klasių diagrama

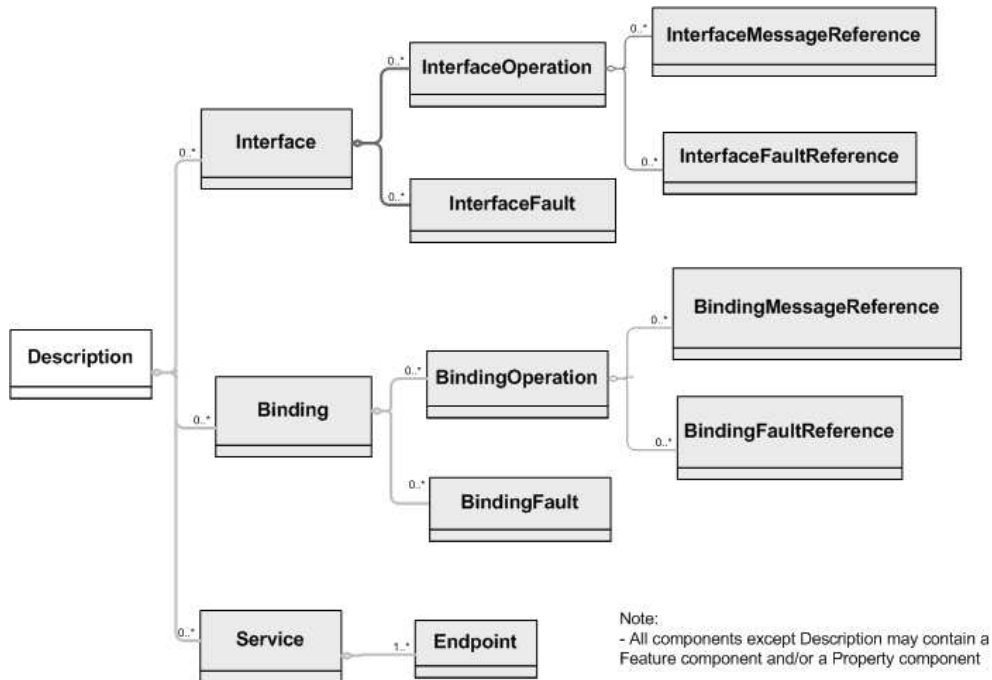
4.8 Sąsajos duomenų struktūros

Šiame skyriuje pateikiami automatinio grafinės sąsajos generavimo variklio apdorjami duomenų modeliai.

4.8.1 Tinklo paslaugos specifikacija (WSDL – Web Service Description Language)

Pagrindinis dokumentas pagal kurį sąsajos generavimo variklis žino kokias paslaugas galima iškviešti ir kokius parametrus reikia paduoti. Taip pat žino koks rezultatas bus grąžintas

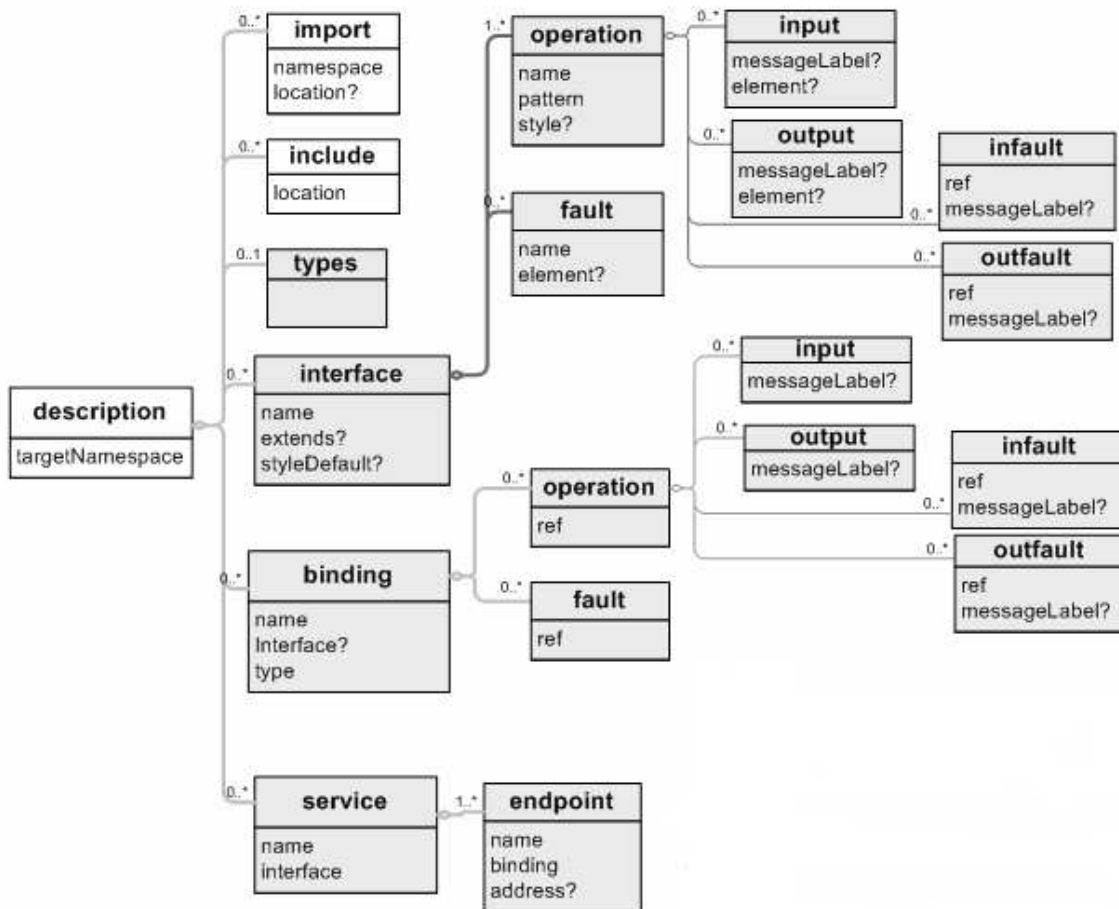
Žemiau (16 pav.) pateikiama WSDL schemas komponentų diagrama.



16 pav. WSDL schemas komponentų diagrama

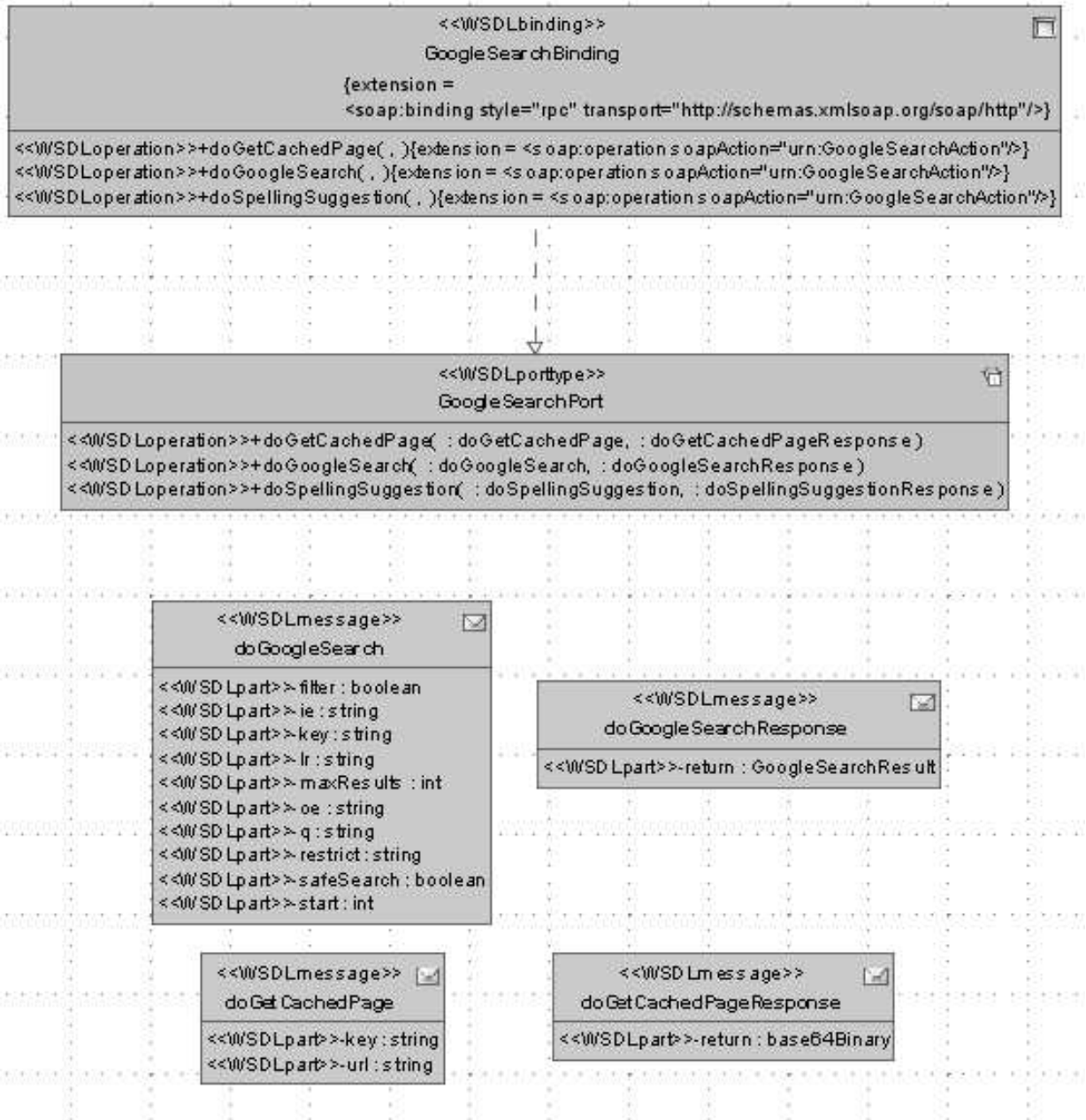
Schemoje matome kokią informaciją gali gauti sąsajos generavimo variklis apie aprašomą tinklo paslaugą

WSDL klasių diagrama (17 pav.) detaliau aprašo WSDL schemas struktūrą ir jos saugojamą informaciją.



17 pav. WSDL schemas klasių diagrama

WSDL schemas vizualizacijoje (18 pav.) aiškiai matyti, kokias operacijas galima atlikti aprašomai tinklo paslaugai, kokius pranešimus galima paduoti. Kaip pavyzdys pasirinktas populiaros Google paieškos tinklo paslaugos aprašas.



18 pav. Pavyzdinės WSDL schemas klasių vizualizacija

Iš klasių diagramos matyti, kad galimos operacijos yra doGetCachedPage, doGoogleSearch ir doSpellingSuggestion. Kiekvienai operacijai reikia paduoti atitinkamą duomenų klasę (pvz.: doGoogleSearch operacijai doGoogleSearch WSDL klasė). Atlikus operaciją tinklo paslauga gražina atsakymo klasę (pvz.: doGoogleSearchResponse). Bendravimas su tinklo paslauga vyksta SOAP pranešimais.

4.8.2 XSL schemas transformacijos (XSLT)

XSL schemas transformacijos aprašas (XSLT) yra oficialus W6C standartas apibūdinantis XML paremtą kalbą, XML dokumentų transformavimui į kitokius XML dokumentus, pavyzdžiui, HTML, XML ar PDF.

Žemiau pateiktas xsl transformacijos pavyzdys:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>Mano kolekcija</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th align="left">Pavadinimas</th>
        <th align="left">Atlikejas</th>
      </tr>
      <xsl:for-each select="catalog/cd">
        <tr>
          <td><xsl:value-of select="title"/></td>
          <td><xsl:value-of select="artist"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Xml dokumento pavyzdys:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="cdcatalog.xsl"?>
<catalog>
  <cd>
    <title>Stadium Arcadium</title>
    <artist>Red Hot Chilli Peppers</artist>
    <country>JAV</country>
    <company>Sony</company>
    <price>10.90</price>
    <year>2007</year>
  </cd>
  .
  .
  .
</catalog>
```

Pritaikę pavyzdyje pateiktą xsl transformaciją xml dokumentui gautume HTML formą pavaizduotą 14-ame paveiksle.

Mano kolekcija

Pavadinimas	Atlikėjas
Stadium Arcadium	Red Hot Chilli Peppers
Visu laiku topai	Hiperbole
Greatest Hits	Queen
Marso kanjonai	Andrius Mamontovas
Eros	Eros Ramazzotti
One night only	Bee Gees
Pilietis	Kostas Smoriginas

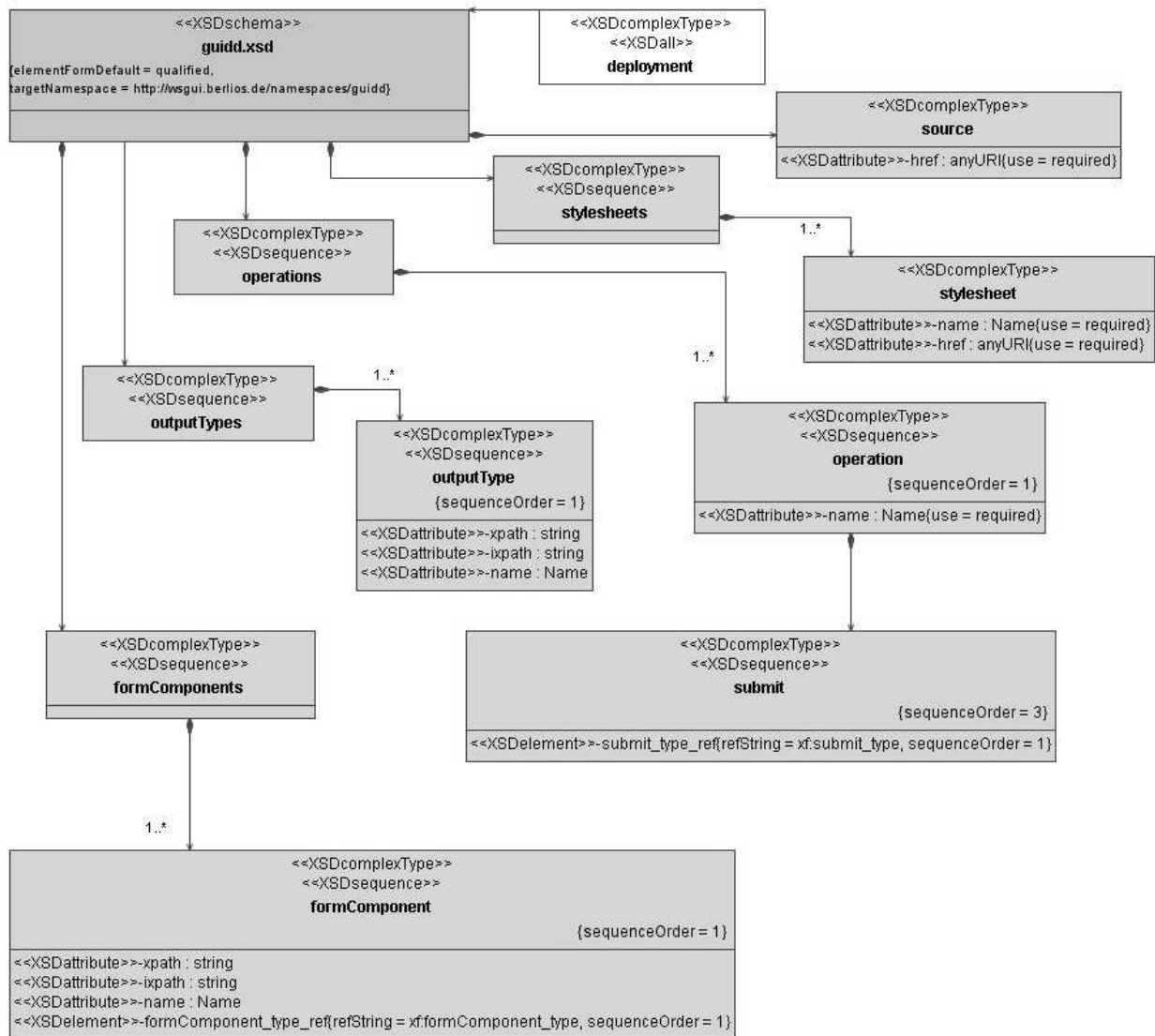
19 pav. Formos pavyzdys pritaikius XSL transformacija XML dokumentui

4.8.3 Grafinės vartotojo sąsajos elementų išdėstymo aprašymas (GUIDD)

XML kalba pagrįstas aprašas naudojamas suformuoti iš WSDL schemas grafinės sąsajos struktūrą. GUIDD – grafinės sąsajos išdėstymo aprašymas (GUI Deployment descriptor).

20-ame paveiksle pateikta pavyzdinio GUIDD aprašo klasių vizualizacija. Joje matoma kokią informaciją saugo struktūra.

Grafinės sąsajos generavimo variklis WSDL specifikacijai pritaikęs GUIDD duomenis sugeneruoja XML struktūrą aprašančią grafinę sąsają. Pritaikius XSLT transformaciją gaunama grafinė vartotojo sąsaja XForms pavidalu.



20 pav. GUIDD aprašo schemas vizualizacija

Žemiau pateiktas 20-ame paveiksle pavaizduoto GUIDD aprašo turinys:

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:guidd="http://wsgui.berlios.de/namespaces/guidd"
  xmlns:xf="http://www.w3.org/2002/xforms"
  targetNamespace="http://wsgui.berlios.de/namespaces/guidd"
  elementFormDefault="qualified">

<element name="deployment">
  <annotation>
    <documentation>

```

The top element of GUI Deployment Descriptor (*.guidd).

```
</documentation>
</annotation>
<complexType>
  <all>
    <element ref="guidd:source" />
    <element ref="guidd:stylesheets" />
    <element ref="guidd:operations" />
    <element ref="guidd:formComponents" />
    <element ref="guidd:outputTypes" />
  </all>
</complexType>
</element>

<element name="source">
  <complexType>
    <attribute name="href" type="xsd:anyURI" use="required" />
  </complexType>
</element>

<element name="stylesheets">
  <complexType>
    <sequence>
      <element name="stylesheet" minOccurs="1" maxOccurs="unbounded">
        <complexType>
          <attribute name="name" type="xsd:Name" use="required" />
          <attribute name="href" type="xsd:anyURI" use="required" />
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>

<element name="operations">
  <complexType>
    <sequence>
      <element name="operation" minOccurs="1" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element name="prettyName" type="xsd:string" />
            <element name="description" type="xsd:string" />
            <element name="submit">
              <complexType>
                <sequence>
                  <element ref="xf:submit_type" />
                </sequence>
              </complexType>
            </element>
          </sequence>
          <attribute name="name" type="xsd:Name" use="required" />
          <!--
            <attributeGroup ref="guidd:portTypeOperation" />
          -->
        </complexType>
      </element>
    </sequence>
  </complexType>
</sequence>
```

```

    </complexType>
</element>
<!--
<attributeGroup name="portTypeOperation">
  <attribute name="wsdlService" type="xsd:QName" use="required" />
  <attribute name="wsdlPort" type="xsd:NCName" use="required" />
  <attribute name="wsdlOperation" type="xsd:NCName" use="required" />
</attributeGroup>
-->
<element name="formComponents">
  <complexType>
    <sequence>
      <element name="formComponent" minOccurs="1" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element ref="xf:formComponent_type" />
          </sequence>
          <attribute name="xpath" type="xsd:string" />
          <attribute name="ixpath" type="xsd:string" />
          <attribute name="name" type="xsd:Name" />
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>
<element name="outputTypes">
  <complexType>
    <sequence>
      <element name="outputType" minOccurs="1" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <element name="mimetype" type="xsd:string" />
          </sequence>
          <attribute name="xpath" type="xsd:string" />
          <attribute name="ixpath" type="xsd:string" />
          <attribute name="name" type="xsd:Name" />
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>
</schema>

```

4.9 Naudotų metodų, algoritmų, technologijų aprašas

Pagrindinis sukurtos sistemos naujumas yra WSGUI motodo pritaikymas XForms grafinės sąsajos generavimui naudojant tinklo paslaugos WSDL aprašą. Šiame skyriuje aprašoma XForms pagrindinės savybės ir teikiami privalumai, bei pagrindinių WSGUI procedūrų diagrama.

XForms - formatas XML duomenų bei vartotojo sąsajos apdorojimo modeliui ir duomenų aprašymui. XForms buvo sukurta kaip kitos kartos HTML / XHTML formos, tačiau gali būti naudojamos ir kaip atskiras formatas aprašyti kitų kalbų nei XHTML vartotojo sąsają ir rinkinių veiksmų duomenų apdorojimui.

XForms aprašymai negali funkcionuoti vieni. reikalingi dokumentai, kurie sugrupuoja, sutvarko formos elementus. Elementus ši aprašymo kalba pateikia abstrakčia forma. Priklausomai nuo pagrindinio dokumento, tokio kaip XHTML ar SVG, elementai gali būti atvaizduoti GUI ar ZUI vartotojo sąsajoje.

XForms yra pagrįsta XML kalba. Tiek formų aprašymas tiek įvestų duomenų perdavimas atliekamas XML formatu. Komunikavimas vyksta per HTTP POST OFFICE metodus, o nuo 1.1 versijos yra galimybė naudoti SOAP užklausas, kurių pagalba galima iškviešti web servisus. Egzistuoja keletas specialių XForms formų realizacijų, taip vadinamų procesorių. Viena iš tokių Mozilla grupės interneto naršyklės, kurios suteikia galimybę atvaizduoti formas įvairiose platformose.

XForms atskiria duomenų prezentacijos elementus nuo turinio, suteikia pakartotinio panaudojimo (reuse) galimybę, yra nepriklausoma nuo platformos ir sumažina skriptų poreikį

Pagrindinis XForms naudojimo privalumas yra tas, kad galima kurti vizualiai tikslias formas. Ši savybė labai svarbi.

XForms privalumai: “Stiprūs” duomenų tipai (strong typing), pasikartojančios struktūros, XML duomenų modeliai, validacija kliento pusėje ir kt.

Automatinės sąsajos generavimui taip pat svarbios savybės tokios kaip dialogai, informaciniai pranešimai ir kitos vizualios priemonės palengvinančios, supaprastinančios ir paaiškinančios grafinę sąsają vartotojui.

Žemiau pateikiamas XForms failo pavyzdys:


```

<xforms>
<model>
<instance>
  <person>
    <fname/>
    <lname/>
  </person>
</instance>
<submission id="form1"
action="submit.asp"
method="get"/>
</model>
<input ref="fname"><label>First Name</label></input>
<input ref="lname"><label>Last Name</label></input>
<submit submission="form1"><label>Submit</label></submit>
</xforms>

```

Pagal pateiktą failo pavyzdį sugeneruojama grafinė sąsaja pateikta 21-ame paveikslėlyje

First Name

Last Name

21 pav. XForms grafinės sąsajos pavyzdys

Modalinio dialogo failo fragmentas ir dialogo pavyzdys (22 pav.):

```

<xf:model>
  <xf:message level="modal" ev:event="xforms-ready">Nurodykite tinklo paslaugos
  ir abstraktu grafines sasajos aprasus</xf:message>

```

Pagrindinis iskvietimo puslapis

Tinklo paslaugos adresas

GUIDD adresas

Pasirinkimas: sarasu valdymas

Pasirinkimas: Vartotojo-pakeitimas

[XForms]

Nurodykite tinklo paslaugos ir abstraktu grafines sasajos aprasus

Informacinio pranešimo aprašymas ir pavyzdys (23 pav.):

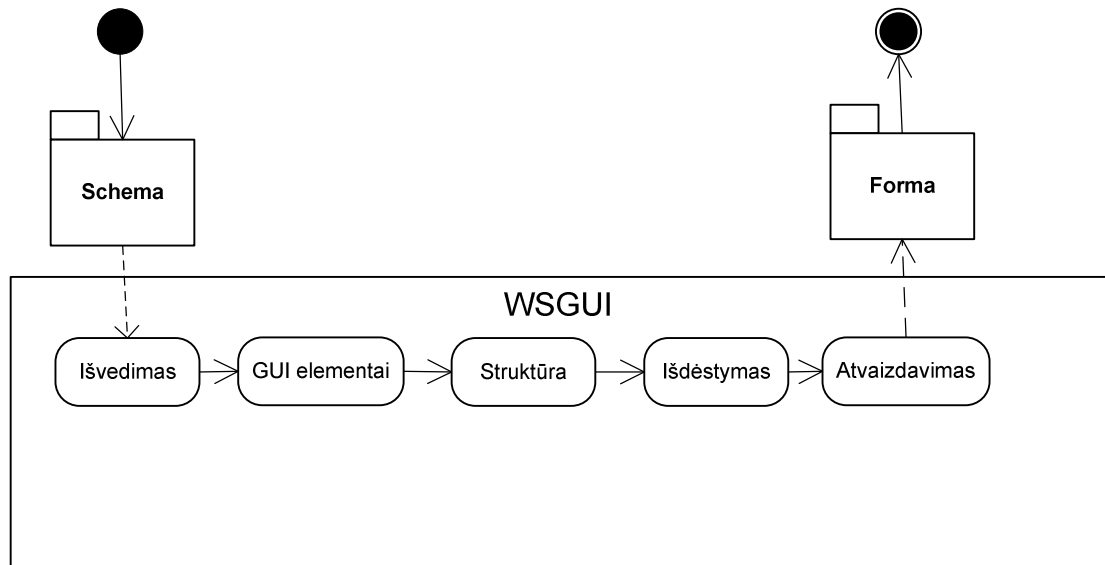
```
<secret ref="/login/password">  
  <label>Password</label>  
  <help>Have you forgotten your password? Simply call 1-900-555-1212 and have  
    a major credit card handy.</help>  
</secret>
```



23 pav. XForms informacinio pranešimo pavyzdys

4.10 WSGUI veilos diagrama

WSGUI metodas aprašytas analizės dalyje 2.10.1 skyriuje. Žemiau (24 pav.) pateikiamos pagrindinės WSGUI variklio procedūros veiklos diagramos pagalba.



24 pav. Pagrindinės WSGUI variklio procedūros

WSGUI varikliui pateikiama schema, kurios sąsają reikia sugeneruoti. Schema analizuojama, nustatomi reikalingi GUI elementai, sudaroma struktūra, išdėstomi elementai ir iš sugeneruoto aprašo sukuriama prezentacinė forma.

5. Automatinio grafinės vartotojo sąsajos generavimo prototipo eksperimentinė realizacija ir įvertinimas

Šioje dalyje aprašoma sukurto automatinio grafinės vartotojo sąsajos generavimo prototipo realizacija, pateikiami demonstraciniai pavyzdžiai. Panaudojant sukurtą prototipą, automatiškai sugeneruojama interneto svetainė, kurioje galima kviesti tinklo paslaugas.

5.1 Automatiškai sugeneruotos svetainės pavyzdys

Žemiau pateikiamas GUIDD, abstraktaus grafinės sąsajos aprašymo, failo turinys. Pagal šį failą naudojant automatinį sąsajos generavimo variklį buvo sugeneruota tinklo paslaugos iškvietimo svetainė.

```
<guidd xmlns:ev='http://www.w3.org/2001/xml-events'
xmlns:xsi='http://www.w3.org/1999/XMLSchema-instance'
xmlns:xsd='http://www.w3.org/2001/XMLSchema' xmlns:dyn='urn:Iskvietimas'
xmlns:xf='http://www.w3.org/2002/xforms' xmlns='http://www.w3.org/1999/xhtml'>
<head>
<link rel='stylesheet' type='text/css' href='iskvietimas.css' />
<xf:model>
<xf:instance>
<iskv:IskvietimasEntrance>
<iskv:url xsi:type='xsd:anyURI'>http://</iskv:url>
<iskv:guiddurl xsi:type='xsd:anyURI'>http://</iskv:guiddurl>
<iskv:optlists xsi:type='xsd:boolean'>>false</iskv:optlists>
<iskv:opttoggle xsi:type='xsd:boolean'>>false</iskv:opttoggle>
</iskv:IskvietimasEntrance>
</xf:instance>
<xsd:schema targetNamespace='urn:Iskvietejas'>
<xsd:element name='IskvietimasEntrance'>
<xsd:complexType>
<xsd:sequence>
<xsd:element name='url' type='xsd:anyURI' />
<xsd:element name='guiddurl' type='xsd:anyURI' minOccurs='0' />
<xsd:element name='optlists' type='xsd:boolean' />
<xsd:element name='opttoggle' type='xsd:boolean' />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
<xf:submission action='http://localhost/testas' method='post' id='default' />
</xf:model>
</head>
<body>
<div>
<xf:label xml:lang='en'>Invoking Entrance Page</xf:label>
<xf:label xml:lang='lt'>Pagrindinis iskvietimo puslapis</xf:label>
```

```

<hr />
<xf:input ref='//iskv:IskvietimasEntrance/iskv:url'>
<xf:label xml:lang='en'>URL of web service</xf:label>
<xf:label xml:lang='lt'>Tinklo paslaugos adresas</xf:label>
<xf:hint xml:lang='en'>WSDL files or XSD files can be given</xf:hint>
<xf:hint xml:lang='lt'>WSDL arba XSD failo adresas</xf:hint>
</xf:input>
<br />
<xf:input ref='//iskv:IskvietimasEntrance/iskv:guiddurl'>
<xf:label xml:lang='en'>URL of GUIDD</xf:label>
<xf:label xml:lang='lt'>GUIDD adresas</xf:label>
<xf:hint>
    This file contains optional GUI hints.
</xf:hint>
</xf:input>
<br />
<xf:input ref='//iskv:IskvietimasEntrance/iskv:optlists'>
<xf:label xml:lang='en'>Option: list handling</xf:label>
<xf:label xml:lang='lt'>Pasirinkimas: sarasu valdymas</xf:label>
</xf:input>
<br />
<xf:input ref='//iskv:IskvietimasEntrance/iskv:opttoggle'>
<xf:label xml:lang='en'>Option: User-toggle</xf:label>
<xf:label xml:lang='lt'>Pasirinkimas: Vartotojo-pakeitimas</xf:label>
</xf:input>
<br />
</div>
<xf:submit submission='default'>
<xf:label xml:lang='en'>Invoke WebService</xf:label>
<xf:label xml:lang='lt'>Iskviesti paslauga</xf:label>

</xf:submit>
</body>
</guidd>

```

Kaip matome, abstraktaus aprašo faile tekstai pateikiami dvejomis kalbomis: angliški ir lietuviški. Priklausomai nuo vartotojo pasirinkimo svetainė jam bus rodoma norima kalba. 25-ame paveiksle matome sugeneruotos sąsajos vaizdą.

Invoking Entrance Page

The screenshot shows a web form with the following elements:

- Input field: URL of web service | http://
- Input field: URL of GUIDD | http://
- Input field: Option: list handling | false
- Input field: Option: User-toggle | false
- Submit button: Invoke web service

25 pav. Automatiškai sugeneruota tinklo paslaugų iškvietimo svetainė.

Reikalingų pradinių duomenų (WSDL ir GUIDD) failų adresai nurodomi atitinkamai laukuose „URL of web service“ ir „URL of GUIDD“. Taip pat matoma papildomi pasirinkimai, ar taikomoji programa turi apdoroti sąrašus ir ar reikia jungtis kitu vartotoju.

Invoking Entrance Page

URL of web service

URL of GUIDD

This file contains optional GUI hints. list handling

user-toggle

26 pav. XForms sąsajos pagalbiniai elementai

Užvedę pelės žymeklį ant kurios nors antraštės, taip pat matome XForms informacinį tekstą pasirinkta kalba. GUIDD apraše nurodę stiliaus failą, galime gauti daug gražesnę ir suprantamesnę vartotojui rezultatą. 27-ame paveiksle matome svetainę sugeneruotą naudojant paprastą CSS stiliaus aprašą.

Invoking Entrance Page

URL of web service

URL of GUIDD

Option: list handling

Option: User-toggle

27 pav. Automatiškai sugeneruota sąsaja. Pritaikytas stiliaus failas

Pasirinkę kitą kalbą, gauname tą pačią sąsają, tik kita kalba. Jeigu kuris nors elementas bus neišverstas, tokiu atveju tekstas rodomas pagrindine kalba. 28-ame ir 29-ame paveiksle matome sulietuvintą sugeneruotą sąsają.

Pagrindinis iskvietimo puslapis

Tinklo paslaugos adresas	http://
GUIDD adresas	http://
Pasirinkimas: sarasu valdymas	false
Pasirinkimas: Vartotojo-pakeitimas	false

Iskviesti paslauga

28 pav. Lietuviškas sugeneruotos sąsajos pavyzdys

Pagrindinis iskvietimo puslapis

Tinklo paslaugos adresas	http://
GUIDD adresas	http://
Pasirinkimas: sarasu valdymas	false
Pasirinkimas: Vartotojo-pakeitimas	false

Iskviesti paslauga

WSDL arba XSD failo adresas

29 pav. Lietuviškas sugeneruotos sąsajos pavyzdys. Matomas informacinis pranešimas

Po to, kai nurodomas tinklo paslaugos WSDL failo adresas ir GUIDD failo adresas, galima iškviešti tinklo paslaugą. Pagal šiuos duomenis bus automatiškai sugeneruota interneto svetainė, kurioje išvardinamos galimos tinklo paslaugos operacijos. Operacijų sąrašas, kaip ir iškvietimo svetainė, pateikiamas XForms forma (30 pav.).

Operacijų sąrašas: testavimui

Vykdyti

- testavimui
- gauti irasus**
- nustatyti iraso reiksme

30 pav. Tinklo paslaugos operacijos pasirinkimo svetainė

Pasirinkęs operaciją iš pateikto sąrašo ir paspaudęs mygtuką „Vykdyti“, vartotojas yra nukeliamas į svetainę, kurioje galės įvesti operacijai reikalingus pradinis duomenis. Tokio lango pavyzdys pateiktas 31-ame paveikslėlyje.

gauti irasus	
Atlikejo pavadinimas	<input type="text" value="Įveskite pavadinimą"/>
<input type="button" value="Vykdėti operacija"/>	

31 pav. Operacijos įvykdymo svetainės pavyzdys

Svetainės viršuje parodomas operacijos pavadinimas. Žemiau išvardinami pradinių duomenų įvedimo laukai, pagal kurių reikšmes bus sukuriamas pranešimas tinklo paslaugai. Pateiktame pavyzdyje matome tik vieną lauką, susietą su paprastu duomenų tipu „string“. Įvedus reikšmę ir paspaudus mygtuką „Vykdėti operacija“, suformuojamas ir nusiunčiamas SOAP pranešimas tinklo paslaugai.

Sistema sugeba atvaizduoti ne tik paprastus, bet ir sudėtingus duomenų tipus. Tokio duomenų tipo aprašymo WSDL faile pavyzdys:

```
<xsd:complexType name="IrasoKonteineris">
  <xsd:all>
    <xsd:element name="pavadinimas" type="xsd:string"/>
    <xsd:element name="metai" type="xsd:int"/>
    <xsd:element name="tipas" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>
```

Matome, kad klasė `IrasoKonteineris` susideda iš trijų elementų: „pavadinimas“, „metai“ ir „tipas“. Kiekvienas elementas gali įgyti atitinkamo paprasto duomenų tipo reikšmę.

Pasirinkta operacija „gauti įrašus“ gražina „IrasoKonteineris“ tipo masyvą. Gautų rezultatų pavyzdys pateiktas 32-ame paveikslėlyje. Kiekvienas masyvo elementas pateikiamas naujoje eilutėje. Sudėtinio tipo elementai pateikiami toje pačioje eilutėje. Elementų pavadinimai rodomi suformuotos lentelės antraštėje.

gauti irasus	
Atlikejo pavadinimas - Sting	
Iraso pavadinimas	Metai Tipas
Brand new day	1999 CD
Send your love	2003 CD
Fragile	1988 MC
Grįžti į pagrindinį puslapį	

32 pav. Operacijos vykdymo rezultatas

Lentelės, kaip ir visos svetainės, atvaizdavimą galima keisti naudojant skirtingus stilių aprašo failus. Mygtukas „Grįžti į pagrindinį puslapį“ nukelia vartotoją atgal į tinklo paslaugos pasirinkimo svetainę.

5.2 Vartotojo sąsajos generavimo prototipo įvertinimas

Šiame poskyryje pateikiamas sukurto prototipo įvertinimas (lentelė)

Kriterijus	Kaip jis pasiekiamas
Sukurti sistemą, kuri generuotų eiliniam vartotojui aiškia, suprantamą grafinę sąsają pagal tinklo paslaugų specifikaciją	Vartotojas gali nesunkiai, jam patogiai kalba iškviešti tinklo paslaugą nesigilindamas į techninę dalį.
Naudodamas šią sąsają, vartotojas gali siųsti užklausas tinklo paslaugoms ir gauti rezultatus	Patvirtinta atliktu eksperimentu.
Sugeneruota grafinė sąsaja turi būti matoma su populiariausiom šiuo metu interneto naršyklėm	Sistema matoma su populiariausiom naršyklėm palaikančiom XForms standartą.
Sistema turi užtikrinti duomenų saugumą ir susidoroti su sudėtingais duomenų tipais	Naudojant abstraktų sąsajos aprašą galima apdoroti visus sudėtinius duomenų tipus
Sąsajos generavimo realizacija turi būti tokia, kad ją nesunkiai būtų galima pritaikyti įvairioms kalboms bei įvairioms vartotojo sąsajos realizavimo technologijoms	Sąsajos generavimas pasiekiamas per komponento interfeisą. Tokiu būdu nesunkiai galima pakeisti generuojamos sąsajos tipą.
Didesnis svetainių lankstumas, dinamiškumas, greitesnis atnaujinimas, keičiantis tinklo paslaugoms	Pasikeitus tinklo paslaugos aprašui nebūtina iš naujo perprogramuoti tinklo paslaugos sąsajos.

6. Išvados

1. Augantis tinklo paslaugų technologijų naudojimas praktikoje ir dažni paslaugų kitimai rodo, kad tikslinga automatiškai generuoti jų grafinę sąsają interneto svetainių vartotojams.

2. Automatinių grafinės sąsajos generavimo metodų analizė parodė, kad pats funkcionaliausias ir labiausiai poreikius atitinkantis yra WSGUI grafinės sąsajos generavimo metodas. Tai grafinės sąsajos pagal WSDL ir abstraktų GUIDD aprašą generavimo metodas, pritaikytas būtent tinklo paslaugų sąsajai generuoti.

3. Išanalizavus įvairias grafinių vartotojo sąsajų technologijas ir aprašymo formatus, nuspręsta generuoti GUI tipo grafinę vartotojo sąsają, kurios atvaizdavimas interneto naršyklėse realizuotas XForms tipo formomis.

3. Realizuoto grafinės sąsajos generavimo prototipo išbandymas tinklo paslaugų pavyzdžiams parodė, kad tyrimo tikslai pasiekti:

- Vartotojo sąsają galima sugeneruoti iš WSDL ir GUIDD aprašų, naudojant XSLT transformaciją į XForms tipo sąsają;
- Galima sugeneruoti ne tik paprastus, bet ir sudėtinius duomenų tipus;
- GUIDD aprašo pasirinkimo dėka galima generuoti sąsają įvairiomis kalbomis;
- Adapterio panaudojimas XForms sąsajos generavimo realizacijoje įgalina sukurtą specifikaciją pritaikyti ne tik XForms sąsajai, bet ir kitoms vartotojo grafinės sąsajos atvaizdavimo technologijoms ir platformoms.

5. Sukurto sąsajos generavimo prototipo pritaikymas praktikoje suteiktų galimybę interneto svetainių administratoriams nesirūpinti vartotojo sąsajos atnaujinimu, keičiantis tinklo paslaugoms.

6. Tolesnio plėtojimo metu būtų galima realizuoti duomenų, reikalingų tinklo paslaugos išskvietimui, surinkimą keletu žingsnių, duomenų patikrinimą prieš SOAP pranešimo formavimą, informacinių pranešimų rodymą.

7. Literatūra

- [1] Michael Kassoff, Daishi Kato, Waqar Mohsin, "Creating GUIs for Web Services," *IEEE Internet Computing*, vol. 07, no. 5, pp. 66-73, , 2003.
- [2] Scott Short, „*Building XML WebService for the Microsoft .NET platform*“, Microsoft
- [3] XForms 1.0 (Third Edition) W3C Recommendation. <http://www.w3.org/TR/xforms/>
- [4] XForms 1.1 W3C Candidate Recommendation <http://www.w3.org/TR/xforms11/>
- [5] XML User Interface Language (XUL) <http://www.mozilla.org/projects/xul/>
- [6] „*Developing XML Web Servines and server components with Microsoft visual basic .net and Microsoft visual C# .net*“, Microsoft press
- [7] Surbhi Malhotra, “*Microsoft .Net Framework Security*”, Thomson Course Technology
- [8] Niel M. Bornstein, “*.NET and XML*”, O’Reilly
- [9] Alex Ferrara, Matthew MacDonald, “*Programming .NET Web Services*”, O’Reilly
- [10] Robert Tabor, “*Microsoft .Net Xml Web Services*”, Sams Publishing
- [11] Apache Software foundation. <http://struts.apache.org/>
- [12] The JWIG Project. <http://www.brics.dk/JWIG/>
- [13] Barry & Associates, Inc. Web Services User Interface (WSUI) http://www.service-architecture.com/web-services/articles/web_services_user_interface_wsui.html
- [14] Generic SOAP Client <http://www.soapclient.com/>
- [15] KStuff.org Free services for desktop people. <http://www.kstuff.org/webservices/index.php>
- [16] Web service GUI. <http://wsgui.berlios.de/>
- [17] Google developer home. <http://code.google.com/>