

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERINIŲ TINKLŲ KATEDRA

Darius Zabiela

**Komponavimo paslauga paslaugoms orientuotai
mokymosi aplinkai**

Magistro darbas

Darbo vadovas

doc. dr. Vytautas Rėklaitis

Kaunas, 2008

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERINIŲ TINKLŲ KATEDRA

Darius Zabiela

**Komponavimo paslauga paslaugoms orientuotai
mokymosi aplinkai**

Magistro darbas

Recenzentas

2008-01-

doc. dr. Rimantas Kavaliūnas

Vadovas

2008-01-

doc. dr. Vytautas Rėklaitis

Atliko

2008-01-10

IFM-2/4 gr. stud.
Darius Zabiela

Kaunas, 2008

Turinys

ĮVADAS	4
1. E.MOKYMO SISTEMŲ ARCHITEKTŪRINIAI SPRENDIMAI	6
1.1 Tyrimo sritis, objektas ir problema	6
1.2 E.mokymo sistemos ir sprendimai	7
1.2.1 Mokymo objektai ir jų metaduomenys	7
1.2.2 Sprendimai ir pasiūlymai kurti šiuolaikines e.mokymo sistemas	10
1.2.3 Mokslo ir Studijų virtuali organizacija	12
1.3 POA technologijų analizė	13
1.3.1 Web paslaugos	15
1.3.2 Portalai	18
1.3.3 Tradicinės ir į paslaugas orientuotos architektūros palyginimas	20
1.4 Analizės išvados	22
2. MOKSLO IR STUDIJŲ IŠTEKLIAI IR KOMPONAVIMO UŽDAVINYS Į PASLAUGAS ORIENTUOTOJE ARCHITEKTŪROJE	23
2.1 Mokymo objektai paslaugoms orientuotoje architektūroje	23
2.2 Sudėtinės mokslo ir studijų paslaugos samprata	25
2.3 Sudėtinių paslaugų komponavimo būdai	26
2.4 Komponavimo posistemės komponentai	28
2.5 Resursų paskirstymo pavyzdys	29
2.6 Komponavimo veiklos procesas	31
3. KOMPONAVIMO POSISTEMĖS REIKALAVIMŲ ANALIZĖ	33
3.1 Komponavimo paslaugos funkcijos	33
3.2 Nefunkciniai reikalavimai	35
4. KOMPONAVIMO POSISTEMĖS PROJEKTAS	36
4.1 Komponavimo posistemės paslaugų identifikacija	36
4.1.1 Komponavimo paslaugos funkcinė-struktūrinė schema	36
4.1.2 Paslaugos tiekėjų ir paslaugos gavėjų identifikacija	38
4.1.3 Dalykinės srities modelis	38
4.2 Paslaugų specifikacija	41
4.3 Paslaugų realizacija	43
4.3.1 Panaudojimo atvejo „Sukurti Scenarijų“ sekų diagrama	43
4.3.2 Panaudojimo atvejo „Redaguoti Scenarijų“ sekų diagrama	44

4.3.3 Panaudojimo atvejo „Naikinti Scenarijų“ sekų diagrama.....	44
4.3.4 Panaudojimo atvejo „Pridėti Naują Objektą“ sekų diagrama.....	45
4.4.5 Panaudojimo atvejo „Šalinti Objektą“ sekų diagrama.....	45
4.3.6 Panaudojimo atvejo „Peržiūrėti Scenarijų“ sekų diagrama	47
4.3.7 Paslaugų klasių diagramos.....	51
4.4 Diegimo modelis.....	54
4.5 Paslaugų vartotojų sąsajos	55
5. PASLAUGOS EKSPERIMENTINIS PANAUDOJIMAS IR VERTINIMAS	62
REZULTATAI IR IŠVADOS	64
LITERATŪRA	65
TERMINŲ IR SANTRUMPŲ ŽODYNAS	67
PRIEDAS NR. 1, IEEE 1484.12.1 LOM METADUOMENŲ MODELIS.....	68
PRIEDAS NR. 2, INTERACTIVE COMPOSITION OF LEARNING WEB SERVICES.....	69
PRIEDAS NR. 3, THE ELEGI NOTIFICATION SERVICE: IMPLEMENTATION AND CASE STUDY	75

SUMMARY

Composition service for service oriented learning environment

The most important objective of e-learning systems is to provide learning content which should give required knowledge and competence for learners or scientists. Such content shouldn't be created from scratch. It should be interactively created from smaller/atomic learning objects and delivered to VO users. Learning object composition to seamless and more completed structure by learning approach is named learning service or learning scenario. Learning objects may vary from simple text document to sophisticated web services which are the most important part of VO. Therefore composition of learning service is quite complex task. However bringing on such implementation technologies like web services and portlets makes this task solvable.

Composed learning service should give learning content which is called learning scenario for VO users on some particular topic. This scenario can consist of description, task and theory, modeling, experimentation, assessment and other tools or web services what can be used for learning. These learning objects may change frequently. The main problem is creation of composition solution which could be able to deliver last versions of learning objects for learners.

The developed composition solution could be used for creation of modern e-learning systems.

Ivadas

Šiuolaikinės mokymo(si) sistemos dažniausiai realizuoja žinių perteikimo modelį kai naudojama aiškinamojo pobūdžio medžiaga. Paprastai ji apiforminama mokymo kursais LMS aplinkose. Tokie kursai besimokantiems suteikia galimybę įgyti žinias, bet nesuteikia gebėjimų.

Pastaraisiais metais e.mokymo srityje taikomi mokymo objektai. Mokymo objektai – tai, bet kokia skaitmeninio turinio medžiaga galinti pasitarnauti mokyme. Paprastai mokymo objektais laikomi vaizdinio, garsinio, tekstinio turinio failai. Tokiems mokymo objektams yra sukurtos saugyklos (angl. LO repositories) kuriose talpinami ir maži ir dideli objektai. Talpinami ne tik smulkūs mokymo objektai, bet ir aukščiau minėti mokymo kursai, sukurti Moodle ar Blackboard LMS aplinkoms, kadangi jie taip pat traktuojami kaip pedagoginiu požiūriu išbaigti mokymo objektai. Mokymo objektų saugyklų specifika tokia, kad ten dažniausiai laikomi mokymo objektai, kuriuos vartotojai parsisiunčia ir leidžia savo kompiuteryje. Visai neseniai pasirodžiusi IMS-LD specifikacija nusako kaip mokymo objektus aprašomus standartizuotais metaduomenimis, apjungti į kursą, kuris, savo ruožtu, yra specialus mokymo objektų rinkinio archyvas. Tas pats kursas gali būti panaudojamas daugelyje mokymo sistemų. Tačiau čia pasireiškia lankstumo problema, kadangi kursą sudarantys objektai gali pasikeisti, o kursai savyje saugo seną mokymo objekto versiją. Tenka kursą atnaujinti jam pridėdant atnaujintus mokymo objektus. Tuo pačiu ir kursų vykdymo aplinkos turi pakeisti seną kursą nauju.

Pastaruoju metu susiformavo naujas požiūris į mokymo objektą. Mokymo objektais pradedami vadinti ne tik failų pavidalo objektai vykdomi vartotojo kompiuteryje, bet ir sudėtingesnes funkcijas atliekančios programų sistemos. Tokios sistemos negali taip paprastai būti siunčiamos tinkle. Todėl nebetinka nusistovėję sprendimai, kaip IMS-LD ar SCORM supakuojantys kursą su visais mokymo objektais į vieną failą. Reikia kitokio sprendimo.

Patys naujausi mokymo sistemų sprendimai orientuojasi į paslaugoms orientuotą architektūrą (POA). Šiai kryptčiai atitinka ir KTU kuriama mokslo ir studijų virtuali organizacija (VO). Mokslo ir studijų ištekliai paskirstomi VO tinkle ir prieinami vartotojams kaip paslaugos. Tokia sistema sudaryta iš daugelio būdingų mokymo sričiai paslaugų ir iš jų turi būti galimybė sukomponuoti pedagoginiu požiūriu išbaigtą mokslo ir studijų paslaugą.

Šio darbo tikslas yra panagrinėti tradicinių e.mokymo sistemų privalumus ir trūkumus, bei pažvelgti į paslaugas orientuotos architektūros galimybes kuriant e.mokymo informacines sistemas. Siūloma taikyti į paslaugas orientuotą projektavimo modelį ir mokymosi paslaugas komponuoti iš tinkle paskirstytų e.mokymo išteklių – mokymo objektų. Darbe pateikiamas originalus komponavimo paslaugos sprendimas leidžiantis mokymosi scenarijų autoriams lanksčiai

komponuoti mokymosi paslaugas. Svarbu yra tai, kad į mokymosi scenarijų-paslaugą autoriai gali įkomponuoti ir aiškinamuosius ir vykdomuosius mokymo objektus. Tokių paslaugų gavėjai, tyrėjai ar studentai, gauna ne tik aiškinamąją mokymo medžiagą, bet ir skaičiavimo ar modeliavimo užduotis, kurioms atlikti būtinos pateiktos žinios. Tokiu būdu, komponavimo sprendimas tarnauja ne tik žinių įsisavinimui, bet ir įgūdžių formavimui, o tuo pačiu ir efektyvesniam mokymuisi. Darbą sudaro keturi skyriai.

Pirmajame skyriuje suformuluojama darbo sritis ir problema, bei analizuojami naujausi mokymo kursų komponavimo būdai ir jais paremti standartai..

Antrajame skyriuje aprašomas naujas požiūris į mokymo objektus. Pateikiamas sprendimas komponuoti mokslo ir studijų paslaugas.

Trečiajame ir ketvirtajame skyriuose aprašomas komponavimo posistemės detalus projektas.

Penktajame skyriuje atliekamas paslaugos eksperimentinis tyrimas įvertinant jos tinkamumą mokslo ir studijų virtualiajai organizacijai.

1. E.mokymo sistemų architektūriniai sprendimai

1.1 Tyrimo sritis, objektas ir problema

Pastaraisiais metais IT taikymas e.mokymo srityje įgavo gana kryptingą pobūdį. Labai išpopuliarėjo tokių sistemų kaip Blackboard-WebCT, Moodle, aTutor ir k.t. naudojimas studijose. Šiuo metu plačiai naudojamos mokymo kursų valdymo sistemos ar mokymosi valdymo sistemos (CMS, LMS) besimokantiems suteikia galimybę įgyti žinias ir gauti jų įvertinimą. Tačiau realiai mokymosi procesą sudaro ne tik žinių perteikimas, bet ir gilesnis dalyko supratimas, žinių taikymo įgūdžiai, gebėjimai analizuoti bei sintezuoti naujas žinias [1]. Šių dalykų kol kas nesuteikia šios sistemos. Iš kitos pusės jos yra programiniai produktai, palaikomi tik pačių gamintojų be aiškios galimybės vartotojams patiems plėsti sistemą pridėdant naujų galimybių funkcionalumo ir mokymo kokybės prasme.

Be to, mokslo ir studijų sistemoje yra labai daug mokymo objektų, tačiau nėra vieningos informacinės sistemos prieigai prie jų. Su mokymo objektais pateikiamais failų pavidalu susitvarko daugelis kursų ir mokymo valdymo sistemų, bet jos neturi aiškaus sprendimo, kaip panaudoti mokymo įstaigose esančius programinius produktus, kurie taip pat naudojami kaip mokymo objektai.

Šioje vietoje iškyla minėtųjų mokymo objektų valdymo problema – reikia būdo mokymo objektus komponuoti į stambesnius mokymo požiūriu vienetus. Kursų kūrimo sistemos leidžia kurti kursus. Tam plačiai naudojamas IMS-LD standartas, aprašantis kurso struktūrą ir mokymo objektų vykdymo eigą [2]. Toks kursas yra paprastas naudoti daugelyje sistemų, kadangi visas kursas susideda iš failų ir jis pats yra supakuotas į failą. E.mokymo sistemos palaikančios IMS-LD standartą gali tokius kursus panaudoti. Lengva tokiais kursais keistis. Yra netgi sukurtos saugyklos tokių kursų dalijimuisi. Tačiau tam, kad atitiktų visus Bloom taksonomijos lygius kursas privalo susidėti ne tik iš tekstinių, vaizdinių, garsinių failų, bet ir interaktyvesnių priemonių. Tai gali būti įvairios trimatės modeliavimo priemonės, realių sistemų valdymo įrankiai ir k.t. Problema yra apskritai tokių galimybių suteikimas e.mokymo sistemai, tačiau svarbesnė problema yra jų sukomponavimas į lankstų scenarijų. Lankstumą suprantamas dvejopai. Pirma, tai to paties sukomponuoto scenarijaus panaudojimas keliose mokymo aplinkose paliekant mokymo išteklius ten, kur jie yra patalpinti. Antra, tai nepriklausomas scenarijų sudarančių komponentų redagavimas kuomet pakeitimai atsispindi visose mokymo aplinkose neatliekant jose papildomų scenarijaus perkomponavimo ar kitų veiksmų. Deja, nėra sukurtų specifikacijų, kaip tai padaryti.

Šias problemas sprendžia Mokslo ir Studijų Virtuali Organizacija kuriama POA pagrindu. Šio projekto tikslas sukurti tokia sistema, kuri suteiktų platesnes mokymo galimybes, būtų lengvai plečiama pačių sistemos vartotojų, bei turėtų vieningą priėjimą prie mokymo išteklių.

Šio darbo uždaviniai:

- išanalizuoti e.mokymo sistemų pagrindines savybes ir naudojamus sprendimus;
- išanalizuoti naujausių technologinių sprendimų taikymą e.mokyme.
- išanalizuoti į paslaugas orientuotą architektūrą, ją sudarančius komponentus.
- pagal analizės rezultatus sukurti sprendimą leidžiantį dėstytojams kurti lanksčius mokymo scenarijus.
- Suprojektuoti ir realizuoti komponavimo posistemę, bei integruoti į Mokslo ir Studijų virtualią organizaciją.
- Eksperimentiškai įvertinti komponavimo posistemės funkcionalumą ir tinkamumą.

1.2 E.mokymo sistemos ir sprendimai

Sistemų skirtų mokymui yra pakankamai nemažai. Tokios sistemos yra Blackboard-WebCT, Moodle, aTutor ir kt.

Jų galimybės yra:

- *Vartotojų valdymas*
- *Sinchronines ir asinchronines bendravimo priemonės*
- *Testavimo priemonės*
- *Wiki priemonės*
- *Paštas*
- *Biblioteka*

Tačiau svarbiausios jų galimybės yra:

- *Kursų kūrimas ir valdymas*
- *Standartų palaikymas kursų išbaigtumui ir suderinamumui tarp skirtingų e.mokymo sistemų*

Visų paminėtų pagrindinių galimybių pagrindą sudaro mokymo objektai. Kursai kuriami iš mokymo objektų, bei patys laikomi mokymo objektais. Standartai nusako, kaip mokymo objektai turi būti aprašomi tam, kad būtų užtikrinamas suderinamumas tarp skirtingų sistemų. Toliau šiame skyrelyje aprašomi mokymo objektai, jų apibrėžimas ir panaudojimas.

1.2.1 Mokymo objektai ir jų metaduomenys

Tarptautinės organizacijos, kuriančios e.mokymo sistemų specifikacijas, apibrėžia mokymo objektą (MO) kaip skaitmeninį vienetą, galintį būti naudingai ar tikslingai panaudotą IT

grįstame mokyme [7]. Į taip apibrėžtą išteklių kategoriją patenka labai platus spektras įvairiausių formatų skaitmeninių objektų. Tai didžiulė ir plati objektų įvairovė, nepasiduodanti klasifikacijai. Įvairovė esamų duomenų formatų, augantys duomenų kiekiai bei sudėtingesni duomenų formatai, vis plėtėjanti mokomosios medžiagos tematika mažina galimybes rasti formatų apibendrinimus ir apibrėžti mokymo objektų klasifikacijos kriterijus. Todėl, kaip ir kitose srityse, pavyzdžiui įvairiems bibliotekų ištekliams aprašyti, mokymo objektams aprašyti pasiūlyta naudoti standartizuotus metaduomenis. Yra keletas neįžymiai besiskiriančių mokymo objektų metaduomenų standartų, tarp kurių bene žinomiausias ir dažniau taikomas yra LOM (Learning Object Metadata) standartas, patvirtintas IEEE 2002 metais.

LOM standartas yra skirtas mokymo objektų metaduomenims aprašyti. Jis yra sukonstruotas XML žymių kalbos pagrindu ir pateikia schemą, pagal kurią aprašomi mokymo objektai. LOM schemoje numatyta apie 70 skirtingų atributų, suskirstytų į 9 kategorijas [7]:

1. Bendroji dalis(General).
2. Gyvavimo ciklo atributai (Life cycle)
3. Metaduomenų metadata(Meta-metadata)
4. Techniniai atributai (Technical)
5. Pedagoginiai atributai(Educational)
6. Autorinių teisių dalis(Rights)
7. Ryšiai su kitais objektais (Relation)
8. Antacijos (Annotation)
9. Klasifikatorius (Classification)

Pilnas IEEE LOM standarto modelis ir atributų sąrašas pateikiamas schemoje (Priedas Nr. 1).

Metaduomenys tai yra aprašomojo pobūdžio duomenys apie objektą, pagal kuriuos kompiuteris gali surasti, išskirti iš kitų, objektus pagal metaduomenų atributų reikšmes. Metaduomenys skiriami ne tik paieškai. Jie yra tam tikra objekto charakteristika ir gali būti pateikiama žmogui kaip to išteklio aprašas.

Pagrindinis mokomųjų objektų metaduomenų standartizacijos tikslas yra kiek galima palengvinti objektų pakartotiną naudojimą. Taip yra daroma daugelyje e.mokymo sistemų. Vienos jų palaiko IMS-LD ir objektus pakuoja pagal turinio pakavimo informacinį modelį (angl. Content Packaging Information Model), o kitos remiasi SCORM ir objektus pakuoja į dalinamo turinio objektus (angl. Sharable Content Objects). Ir viename ir kitame standarte mokymo objektai pakuojami į zip archyvus ir tokiu būdu gali būti paskirstomi tinkle ar pernešami iš vienos aplinkos į kitą [8][9].

Mokymo objektai talpinami į specializuotas mokymo objektų saugyklas LOR (Learning Object Repository) [10]. Tuomet objektai būtinai aprašomi metaduomenimis, atsiranda galimybės atlikti objektų paiešką pagal metaduomenų laukus. Saugyklos naudojimas suteikia saugykloms būdingas priemones sekti įvairių mokymosi objektų kokybę, naujumą, populiarumą.

Pedagoginiu požiūriu išbaigti mokomieji objektai

Tai grupė failų apjungta į vieną modulį, pavyzdžiui pagal IMS-LD specifikaciją. Tokia grupė turi savo turinio lentelę, kuri nusako failų ryšius, turi LOM standartu apibrėžtus metaduomenis [2]. Tokius mokomuosius objektus galima įkelti į IMS standartą palaikančias virtualias mokymosi aplinkas ir aplinka atpažįsta mokymosi objektą bei jo vidinę struktūrą. Tokie objektai yra paruošti perdavimui tinklu ir pakartotiniam naudojimui mokymo standartus atitinkančiose vykdymo aplinkose be jokių keitimų ar modifikacijų. E.mokymo sistemose jie populiariai vadinami kursais.

Kursai kuriami iš saugyklose esančių mokymo objektų. Dėstytojas saugykloje suranda tinkamą objektą. Surasti objektai parsisiunčiami ir apjungiami į naują mokymosi kursą. Sukurtas kursas taip pat gali būti patalpintas į mokymo objektų saugyklą.

Sukurti kursai ar jį sudarantys smulkesni mokymo objektai, atsirandant naujoms idėjoms, moksliniams darbams, laiko bėgyje gali pasikeisti. Jie gali būti atnaujinami ir mokymo objektų saugyklose, tačiau to nepakanka, kad pakeistas kursas ar mokymo objektas būtų pakeistas ir jį naudojančiose mokymo sistemose. Pakeitimai e.mokymo sistemose turi būti atliekami rankiniu būdu. Tai daro kursus labai nelanksčius.

Nors šie kursai yra laikomi išbaigtais, tačiau jie besimokančiajam dažniausiai suteikia tik žinias ir iš dalies supratimą. Pagal Bloom yra išskirti tokie pažinimo lygmenys[1]:

1. Žinios
2. Supratimas
3. Taikymas
4. Analizė
5. Sintezė
6. Įvertinimas

Elementarūs tekstiniai, vaizdiniai ar garsiniai failai, esantys IMS kurse, iš principo negali suteikti žinių taikymo, analizės, sintezės ir įvertinimo gebėjimų.

1.2.2 Sprendimai ir pasiūlymai kurti šiuolaikines e.mokymo sistemas

Sistemos orientuotos į tiriamąjį mokymo požiūrį yra ELeGI ir Sakai.

Pirmoji sistema labiausiai verta dėmesio. Ji kuriama European Learning Grid Infrastructure projekto pavadinimu. Jos tikslas teorines mokymo idėjas pritaikyti praktikoje, panaudojant mokymosi kelius ir ontologijas. EleGI sistema yra suskirstyta į dvi mokymo platformas – formalią ir neformalią. Neformali platforma remiasi virtualios organizacijos vartotojų komunikavimu ir bendradarbiavimu. Formali platforma panaši į standartines mokymo sistemas dirbančias su mokymosi objektais, tačiau turi ir galimybę prieiti prie vykdomųjų organizacijos išteklių, kas praplečia šios sistemos praktinę naudą. Tačiau čia šie vykdomieji ištekliai nėra laikomi mokymo objektais ir yra integruoti į kuriamą sistemą be paprastos galimybės šiuos juos panaudoti kitose mokymo sistemose. Iš technologinės pusės projekte naudojamas IWT-GA (Intelligent Web Teacher-Grid Aware) portalas ir web paslaugos vykdomiesiems ištekliams realizuoti. Portalo paskirtis suteikti vieną prieigos tašką prie visų web paslaugų. Portalų technologijos leidžia dideles portalo aplinkos derinimo galimybes, tačiau ELeGI portalas atsižvelgdamas į paslaugas ir vartotojo vaidmenį pateikia iš anksto sukurtą vartotojo sąsają[3]. Kadangi ši sistema remiasi IMS-LD todėl ji neišsprendžia kursų lankstumo problemų. Pasikeitus vienam iš mokymo objektų stambesniį mokymo objektą (kursą) reikia atnaujinti rankiniu būdu peržiūrint pasikeitusius mokymo objektus. Taip pat neišsprendžiama ir kita problema – mokymo objektų naudojimas keliose mokymo aplinkose objektus paliekant vienoje vietoje. Tačiau vykdomųjų išteklių panaudojimas leidžia užtikrinti kokybiškesnį mokymąsi, kadangi jie gali suteikti aukštesnio lygmens (pagal Bloom[1]) gebėjimus.

Mokslinių darbų pateikiančių konkrečius sprendimus kaip kurti integruotą mokslo ir studijų sistemą leidžiančią naudoti heterogeninius resursus nėra. Tačiau yra nemažai pasiūlymų spręsti vienokias ar kitokias e.mokymo problemas.

X. Yang ir R. Allan mokymo išteklius siūlo kurti web paslaugų pagrindu ir juos pateikti vienoje vietoje – portale. Pasiūlyto metodo tikslas papildyti „Sakai“ e.mokymo informacinės sistemos branduolį pridedant standartų JSR-168, WSRP (Web Services For Remote Portlets) palaikymą kurie leistų pakankamai nesunkiai pridėti naujas galimybes į portalą. Naujos galimybės leidžiančios tyrėjams atlikti eksperimentavimo darbus realizuojamos web paslaugų pagrindu, o paslaugų vartotojo sąsajos komponentai (portletai) sukurti pagal standartus lengvai integruojasi į portalą [4]. WSRP panaudojimas leidžia panaudoti mokymo išteklius skirtingose e.mokymo aplinkose neatliekant jų kopijavimo. Tačiau čia nekalbama apie šių išteklių valdymą, bei sudėtinių paslaugų kūrimą.

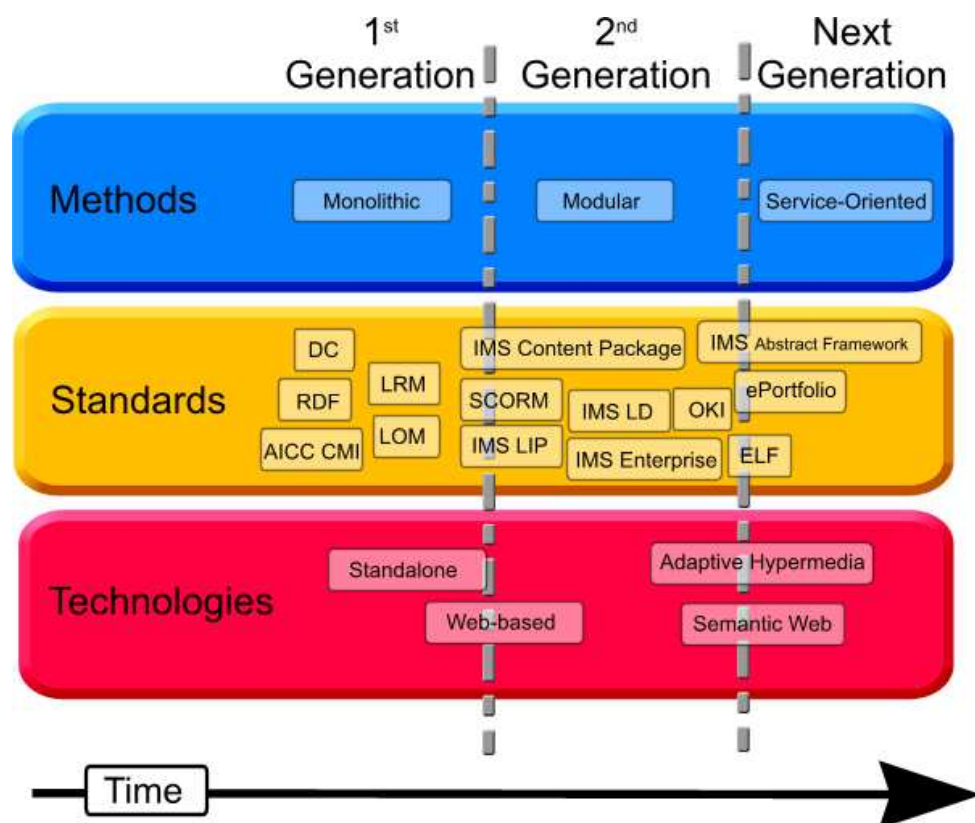
L. Pereira, R.Melo, F. Porto, B. Schulze nagrinėja darbų sekomis grindžiamą architektūrą realizuojamą grid technologijų pagrindu. Pagrindinis metodo tikslas pateikti kiekvienam studentui mokymosi objektus pagal studento darbų seką. Jų metodas remiasi mokymo objektų skirstymu į mokymosi programas. Studentui pasirinkus mokymosi programą jam pateikiami su ta programa susiję objektai. Kurias mokymosi programas rinktis nustato darbų sekos paskaičiavimo sistema. Šio metodo trūkumas yra tas, kad jis nesiūlo sprendimo darbui su paskirstytais objektais [5].

Panašų sprendimą siūlo ir D. Sampson, C. Karagiannidis, F. Cardinali. Skirtumas tas, jog čia studento mokymosi kelias valdomas ontologijų pagalba [6].

G.Vossen ir P.Westerkamp web servisuos naudoja mokymo objektų paėmimui. Mokymo objektas yra failas, tačiau jam perduoti naudojama web paslauga[18]. Toks metodas leidžia integruoti mokymo objektus keliose sistemose pačius mokymo objektus paliekant vienoje vietoje. Deja, ne visi mokymo ištekliai gali būti pateikiami failų pavidalu, tačiau autoriai tokio požiūrio nenagrinėja.

Daugelis autorių savo darbuose ir kuriamose sistemose naudoja web paslaugas, portalus kaip technologijas leidžiančias pasiekti didesnes mokymo galimybes, sprendžiančias integracijos ir lankstumo problemas.

D. Dagger, A. O'Connor, S. Lawless, E. Walsh ir V. Wade išskiria 3 mokymo sistemų kartas – monolitines, modulines ir paslaugoms orientuotas mokymo sistemas[19].



1 pav. E.mokymo sistemų evoliucija

Pirmajai kartai nuo 1993 iki 1999 metų priklauso pirmosios e.mokymo sistemos sukurtos, kaip monolitinės sistemos palaikančios pradinis mokymui sukurtus standartus[19]. Daugiausiai tai į atskirą kompiuterį diegiamos sistemos, tačiau šiai kartai galima priskirti ir pradinę WebCT versiją.

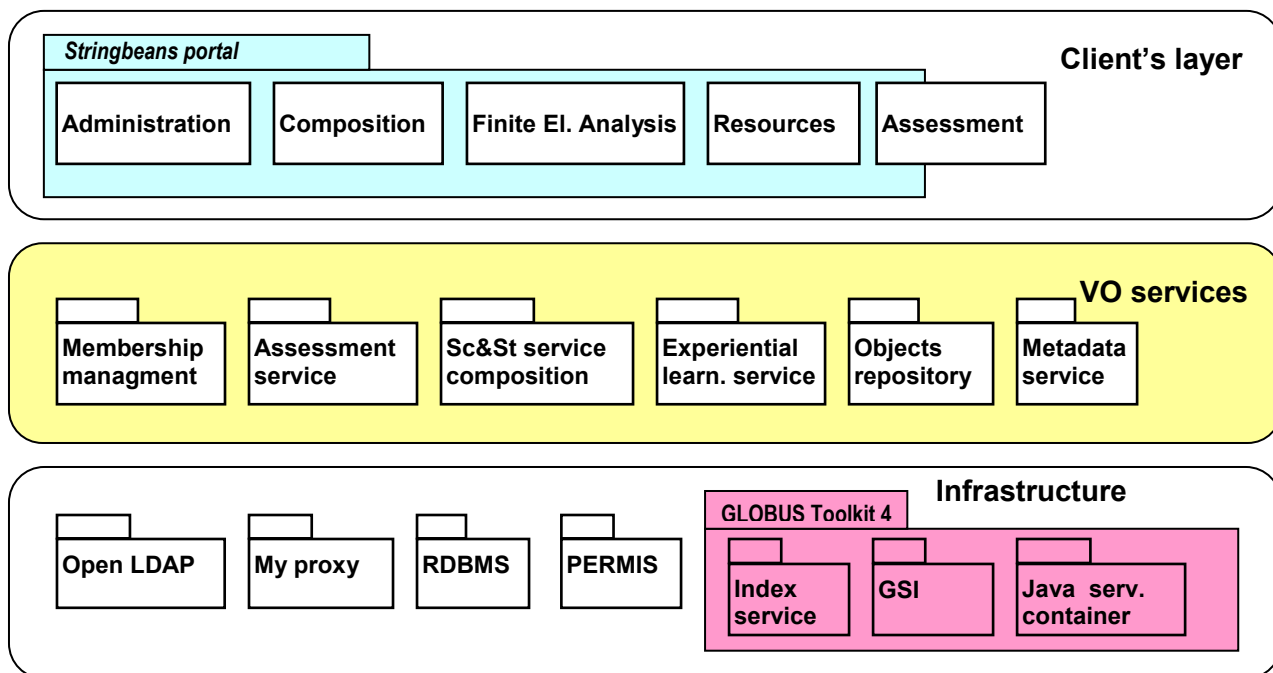
Antroji karta tęsiasi nuo 1999 iki dabar. Šiai kartai priklauso dabartinės populiaros mokymo sistemos kaip Blackboard/WebCT, Moodle, Sakai. Jos iš monolitinių sistemų tapo modulinėmis. Atsirado galimybė integruoti naujus modulius jau įdiegtai sistemai. Sakai sistema jau naudoja web paslaugas kaip priemonę išskaidyti atskirus sistemos komponentus. Atsirado nauji standartai kaip SCORM, IMD-LD leidžiantys sistemoms keistis ne tik smulkiais mokymo objektais, bet ir pedagoginiu požiūriu išbaigtais mokymo objektais.

Trečiosios kartos uždavinys sukurti sistemas panaudojančios ne tik pasyvius mokymo objektus, kuriuos besimokantysis naudoja tik žinių įgijimui, bet ir interaktyvius mokymo išteklius, kurie vartotojams suteiktų platesnes mokymosi galimybes. Taip pat jos apima žymiai platesnį spektrą informacijos, kuri gali būti dalinama tarp skirtingų e.mokymo sistemų. Pavyzdžiui, informacija apie vartotoją, jo žinias. Pagal tai sistema gali parinkti konkrečiam vartotojui reikalingiausią informaciją. Be to, vartotojas pagal savo poreikius gali pats spręsti ką mokytis. Šios kartos sistemos kuriamos paslaugoms orientuotos architektūros pagrindu, panaudojant naujausius technologinius sprendimus – web paslaugas, portalus, grid.

1.2.3 Mokslo ir Studijų virtuali organizacija

Naujausioms e.mokymo idėjoms atitinka ir KTU kuriama Mokslo ir Studijų VO realizuojama POA pagrindu. Mokslo ir studijų ištekliai paskirstomi VO tinkle ir prieinami vartotojams kaip paslaugos. Šiai paskirstytų išteklių sistemai sukurti taikomi moksliniuose darbuose aprašyti sprendimai ir technologijos.

Virtualios Organizacijos architektūra suskirstyta į 3 sluoksnius (2 pav). Architektūroje be VO dalyvių valdymo, baigtinių elementų analizės, žinių vertinimo paslaugų numatytos ir mokslo ir studijų paslaugų kūrimo uždaviniui reikalingi komponentai. Tai komponavimo paslauga ir portletas, metaduomenų paslauga, aiškinamųjų objektų saugyklos paslauga ir paslaugų registras (Index service).



2 pav. Mokslo ir Studijų VO architektūra

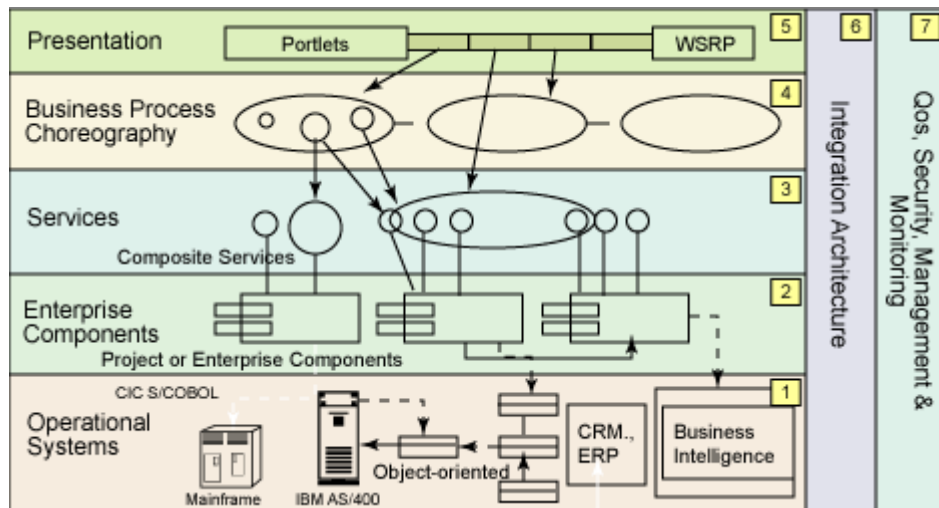
Mokslo ir Studijų VO svarbią vietą užima galimybė komponuoti mokymo scenarijų iš paskirstytų VO išteklių. Komponavimo paslauga yra šio darbo tikslas įrodantis komponavimo sprendimo tinkamumą mokymosi ištekliams valdyti.

1.3 POA technologijų analizė

Tiek apžvelgtoje literatūroje tiek kuriamose sistemose rekomenduojama taikyti POA naudojant dvi pagrindines technologijas – web paslaugas ir portalą. Šiame skyrelyje apžvelgiama POA ir kas lemia jos pasirinkimą ir taikymą.

Į paslaugas orientuota architektūra suteikia programinės įrangos kūrimui lankstumo. Jis nepasireiškia iš karto, kadangi pačios sprendimo architektūros projektavimas ir kūrimas užtrunka žymiai ilgiau negu komponentinės architektūros. Tačiau ilgainiui atsirandant naujoms galimybėms jos gali būti nesunkiai integruotos į egzistuojančią sistemą dėl mažo web paslaugų susietumo. Į paslaugas orientuota architektūra neturi nieko bendro su aparatūrinės, programinės įrangos naudojimu ar specialiais protokolais – tai požiūris į sistemų ir veiklos procesų projektavimą ir organizavimą suteikiantis lankstumą ir galimybę valdyti pasikeitimus. POA idėja iš esmės nepriklauso nuo realizacijos, šis architektūros stilius gali būti įgyvendinamas bet kokia tinkama technologija.

POA elementus galima suskirstyti į 7 sluoksnius. Sluoksniai pavaizduoti 3 paveikslėlyje [20].



3 pav. POA elementai

1 sluoksnis: Operacinių sistemų sluoksnis. Sudarytas iš ankščiau sukurtų sistemų vadinamų liktinėmis sistemomis. Daugiasluoksnė POA architektūra apima egzistuojančias sistemas, kurios gali būti integruojamos panaudojant web servisų technologijas.

2 sluoksnis: Įmonių komponentų sluoksnis. Šis sluoksnis apima tarpinę programinę įrangą, programų serverius užtikrinančius komponentų valdymą ir užduočių paskirstymą.

3 sluoksnis: Paslaugų sluoksnis. Paslaugų sąsajas, kurios gali būti surandamos arba statiškai susietos su juos naudojančiais klientais.

4 sluoksnis: Veiklos procesų kompozicijos ar choreografijos sluoksnis. Šį sluoksnį sudaro iš servisų sluoksnyje esančių servisų sudarytos grupės atliekančios tam tikrą veiklos procesą.

5 sluoksnis: Pateikimo sluoksnis. Nors šis sluoksnis tiesiogiai nesisieja su paslaugų architektūra, tačiau nauji standartai, kaip JSR-168 portletų specifikacija ir jo pagrindu sukurta WSRP specifikacija remiasi web paslaugomis tam, kad būtų galima atlikti sistemų integraciją vartotojo sąsajos lygmenyje.

6 sluoksnis: Integracijos sluoksnis. Integracija galima visuose sluoksniuose. Tam plačiai naudojama įmonės paslaugų magistralė.

7 sluoksnis: Paslaugų kokybė. Šiame sluoksnyje yra įrankiai skirti stebėti, valdyti paslaugų kokybei – saugumui, našumui ir pasiekiamumui.

Tokia sluoksninė architektūra gali būti taikoma ir e.mokymo sistemose. Sistemos sukurtos laikantis paslaugų architektūros principų gali turėti visas 1.2 skyrelio pradžioje išvardintas galimybes. Tačiau vienas pagrindinių POA privalumų suteikia galimybę tokioms sistemoms būti plečiamomis, leidžia naudoti paskirstytus programinius resursus, ko neleidžia įprastinės e.mokymo sistemos.

1.3.1 Web paslaugos

Pagal W3C web paslaugos tai programinės įrangos komponentai, sukurti tam, kad leistų kurti kintamo dydžio, laisvai susietas, nepriklausančias nuo platformos programas[21]. Esminės Web paslaugų charakteristikos yra tokios [13]:

Naudojamas XML

XML naudojimas duomenų perdavime leidžia eliminuoti bet kokias tinklo, operacinių sistemų, ar platformos nesuderinamumo problemas.

Silpnas susiejimas

Paslaugos gavėjas nėra pririštas prie web paslaugos tiesiogiai. Web paslaugos įgyvendinimas gali būti keičiamas visiškai nesugadinant galimybės gavėjui bendrauti su paslauga.

Mažas skaidymo lygis (angl. Coarse-grained)

Objektinėse technologijose kur yra didelis skaidymo lygis naudojama daug metodų. Ir kiekvieną iš tų metodų galima naudoti atskirai. O Web paslaugose reikalingi metodai yra apjungiami į vieną stambų metodą atliekantį tam tikrą veiklos funkciją, kurią gali vykdyti paslaugos gavėjas.

Gali būti sinchroninės arba asinchroninės

Sinchroniniame režime klientas vykdo web paslaugos operacijas ir nieko toliau nedaro kol negauna vykdymo rezultatų. Asinchroniniame režime klientas gali vykdyti paslaugos operacijas ir nesulaukus rezultatų vykdyti kitas operacijas, o paslaugos pateiktus rezultatus panaudoti vėliau.

Palaiko nuotolinį procedūrų iškvietimą

Web paslaugos leidžia iškviešti nuotolines procedūras, funkcijas ir metodus naudojant XML-inį protokolą. Nuotolinių procedūrų iškvietimas supaprastina sistemų integraciją.

Palaiko duomenų apsikeitimą

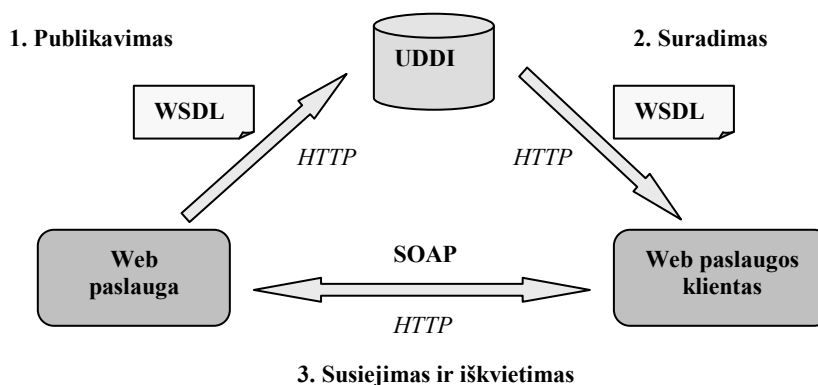
Vienas iš pagrindinių XML privalumų yra tas, jog XML gali atvaizduoti paprastus ir kompleksinius dokumentus. O web paslaugos leidžia skaidriai keistis šiais dokumentais.

Paslaugos paremtos tokiais standartais kaip SOAP, WSDL, UDDI:

- SOAP (Simple Object Access Protocol). SOAP tai XML protokolas, kuris supakuoja XML duomenis į specialios struktūros pranešimą persiuntimui web infrastruktūra.
- WSDL (Web Service Description Language). Aprašo web paslaugas pagal nustatytą XML schemą, t.y., nusako kaip su šia paslauga turėtų sąveikauti klientas.
- UDDI (Universal Description Discovery and Integration). UDDI – tai nepriklausantis nuo platformos, XML pagrįstas web paslaugų registras internete. Paslaugos teikėjas apibrėžia ir publikuoja paslaugos detales registre, tie kurie nori pasinaudoti paslauga užklausia registrą norint surasti paslaugos tiekėją, atliekamas paslaugų susiejimas pasinaudojant UDDI

teikiamom techninėm detalėm. UDDI priima SOAP užklaudas ir teikia priėjimą prie publikuotų WSDL dokumentų, apibūdinančių reikalingas protokolo sąsajas ir pranešimų formatus norint sąveikauti su paslaugomis užregistruotomis registre.

Žemiau pateiktame paveikslėlyje pateikiamas tipinis web paslaugų iškvietimo modelis (4 pav.).



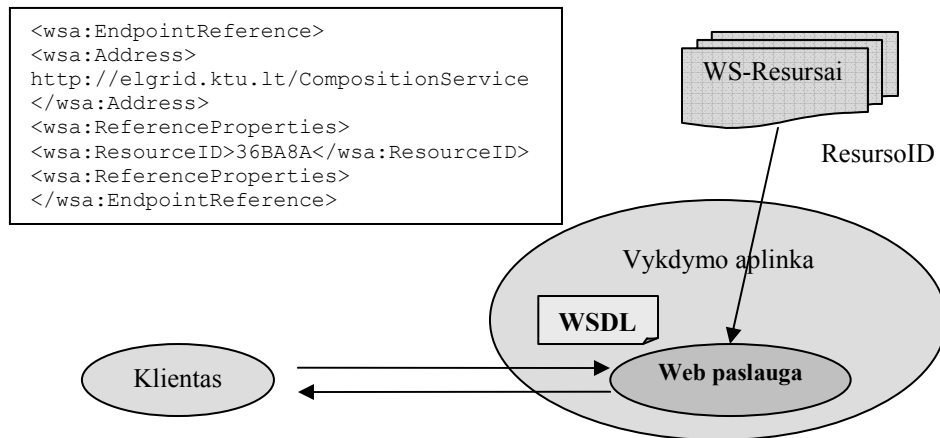
4 pav. Tipinis web paslaugų iškvietimo scenarijus

Šis modelis yra plačiai publikuojamas, tačiau retai naudojamas. Klientai dažniausiai turi būti interaktyvūs, todėl jie yra sukuriami iš anksto. O kai jie yra sukurti iš anksto tuomet nėra galimybės iškviešti bet kokią registre surastą paslaugą. UDDI galima naudoti, kai yra kelios paslaugos turinčios tokią pat sąsają. Tuomet klientas registre suradęs šias paslaugas gali pasirinkti tą, kuri, pavyzdžiui, turi geresnius paslaugos kokybės parametrus. Tačiau realiame gyvenime tokių atvejų pasitaiko gana retai.

Paslaugos su būsenomis

Dažnai į paslaugas orientuotoje architektūroje minimas Grid. Grid išpopuliarėjo kaip technologija leidžianti vykdyti sudėtingus skaičiavimus. Tačiau grid esmė yra ne tik paskirstyti uždavinius keliems kompiuteriams. Grid esmė yra paskirstyti bet kokius resursus – duomenų, skaičiavimo, informacijos. Su grid atsiradimu atsirado ir web paslaugų standartas WSRF, kurio tikslas praplėsti web paslaugas būsenos išsaugojimo galimybe. Ši specifikacija atskiria pačią paslaugą ir jos būseną. Būsenos informacija saugoma visiškai atskiroje esybėje dar kitaip vadinamoje resursu(5 pav.). Kiekvienas resursas turi atskirą identifikatorių, todėl kai tik norime atlikti sąveiką naudodami būseną, tiesiog turime nurodyti tam tikrai web paslaugai naudoti tam tikrą resursą, vykdant operacijas.

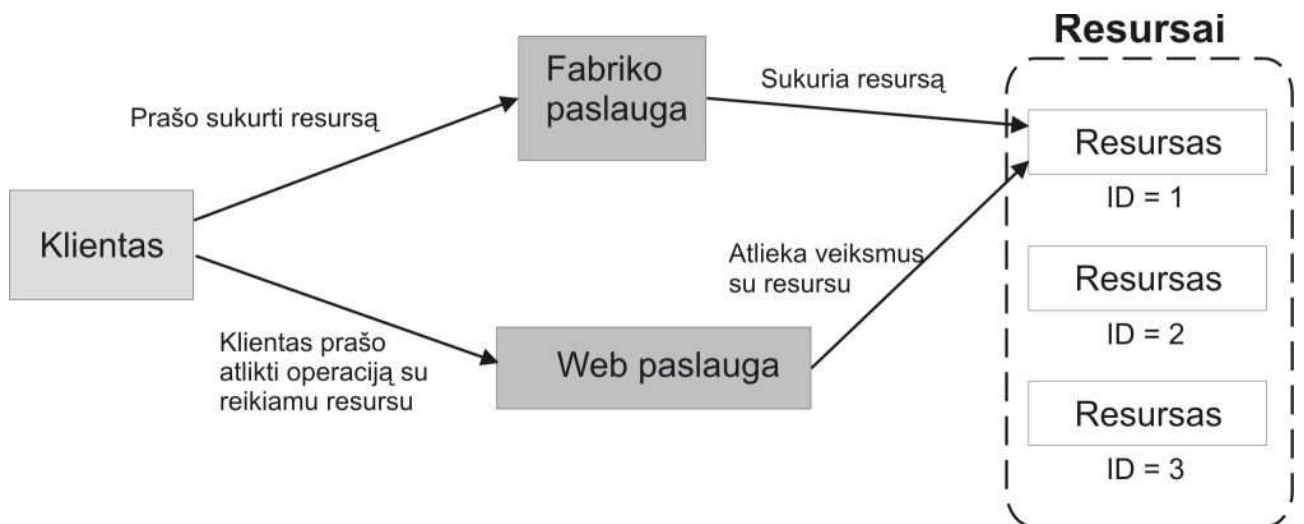
Paslaugos įėjimo taškas, susietas su konkrečiu resursu



5 pav. Paslauga atitinkanti WSRF standartą

WSRF esmę sudaro dvi web paslaugos. Pirmoji paslauga vadinama atskiro paslaugos atvejo paslauga ir įgyvendina reikalingas funkcijas prie pageidaujamo resurso, o kita paslauga vadinama fabriko paslauga atlieka pirmosios paslaugos atskiro atvejo kūrimą (6 pav). Sukurtasis paslaugos atvejis kitaip vadinamas web paslaugos resursu yra pilnai funkcionuojanti paslauga turinti pradines resursų savybes. Toms resurso savybėms galima suteikti norimas reikšmes. Kiekvienam sukurtam paslaugos atvejui suteiktos reikšmės gali būti skirtingos. Pavyzdžiui vienas vartotojas dirba su vienokiu paslaugos atveju naudodamas vienokias reikšmes, o kitas vartotojas kitokias. Ir kol paslaugos atvejis nėra sunaikintas tol vartotojai gali naudoti savo susikurtus resursus su norimomis reikšmėmis. Tai vadinama būsenos išsaugojimu [15]. Tai naudinga tuomet, kai norima sukurti tam tikrą dinamiškumą sąveikaujant su paslauga. Paslaugos teikiamos operacijos išlieka tos pačios, tačiau jos yra atliekamos su skirtingais resursais. Kiekvienas atskiras resursas gali turėti savo gyvavimo laiką, savo saugumo apribojimus ir pan. Kai klientas norės sukurti naują resursą, jis kreipsis į fabriko paslaugą, kuri bus atsakingas už naujo resurso sukūrimą ir inicializaciją.

Jeigu įprastinių web paslaugų adresą sudaro tik URL nurodantis kur yra paslauga, o WSRF atveju paslaugos adresą sudaro ne tik URL bet ir resurso raktas nurodantis konkretų paslaugos atvejį. Fabriko paslauga, sukūrus resursą, gražina WS-Resurso (paslauga + sukurtas resursas) adresą (endpoint reference). Šis adresas leidžia mums unikaliai identifikuoti vieną WS-Resursą – paslaugą sujungtą su atitinkamu resursu.



6 pav. Daugelio resursų paslauga

Įprastinės web paslaugos ir paslaugos saugančios būseną e.mokymo sistemose gali praversti realizuojant priėjimą prie mokymosi išteklių, o jeigu paslaugos su atskirais resursais tuomet galima suteikti POA sistemoms dinamiškumą. Tai mokymo sistemose naudinga, kadangi mokymo ištekliai dažnai keičiasi, su jais dirba ne vienas vartotojas.

1.3.2 Portalai

Kita į paslaugas orientuotoje architektūroje naudojama technologija yra portalai. Portalas yra Web programa teikianti tokias galimybes [14]:

Vienas prisijungimas

Leidžia kiekvieną kartą prisijungus matyti vartotojo poreikiams pritaikyta vaizdą. Vienam vartotojui informacija gali būti vienokia, o kitam kitokia. Joks kitas vartotojas negali pakeisti pirmo vartotojo portalo vaizdo konfigūracijos.

Turinio agregavimas

Turinio grupavimas leidžia atskiruose puslapiuose vaizduoti skirtingo turinio informaciją. Kiekvienas puslapis turi nuoroda, kuri leidžia iš karto patekti į kitą puslapį.

Portletų koncepcija

Leidžia kurti naujus portletus ir juos prijungti į portalą. Portletas tai vartotojo sąsajos elementas pateikiantis tam tikro turinio informaciją. Vartotojas iš daugelio portletų gali susikonfigūruoti norimą vaizdą.

Vartotojo sąsajos prisiderinimas

Leidžia portalo vartotojams realiame laike derinti portale pateikiamą informaciją

Portalo aplinkos derinimas

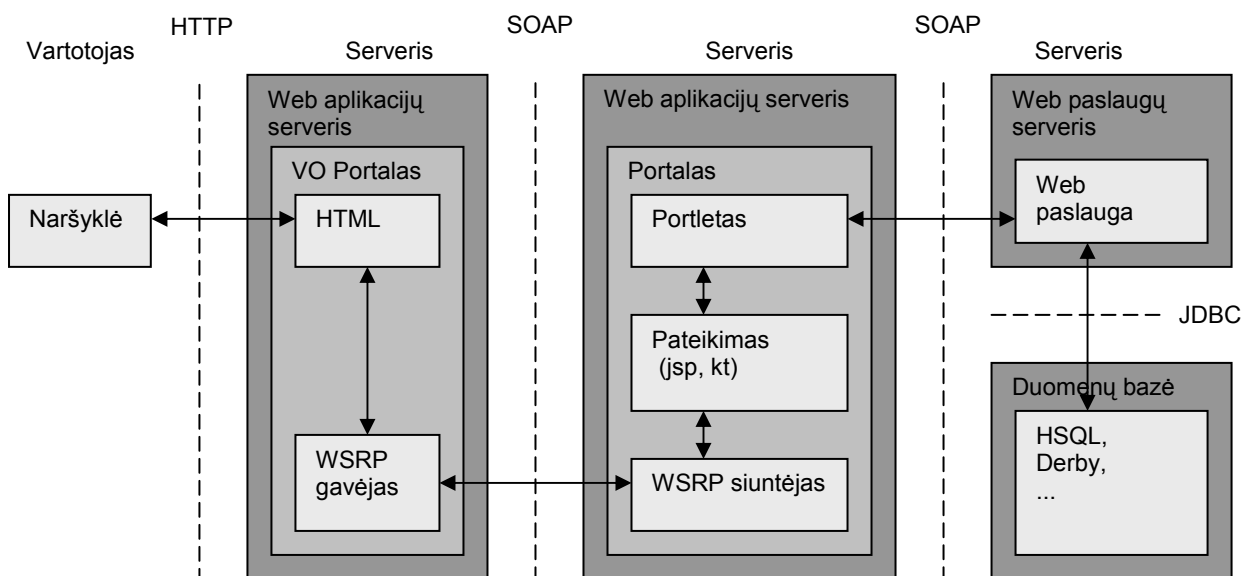
Leidžia portalo administratoriams portalo komponentus išdėstyti į norimą portalo struktūrą, o portalo vartotojai pradžioje matys tokį portalo vaizdą, kokį jiems sukuria administratorius.

Portletus labai gerai tinka naudoti kaip paslaugos klientus. Portletai yra nedideli, užima tik dalį viso portalo puslapio, yra dalinai nepriklausomi nuo jo, gali būti panaudoti keliuose puslapiuose, todėl juos galima naudoti atskiros web paslaugos iškvietimui ir duomenų atvaizdavimui. Tas pats portletas gali iškviešti kelias web paslaugas reikalingas įvykdyti norimą veiklą.

E.mokymo sistemose portalai gali pasitarnauti, kaip karkasas kurti apibendrintą vartotojo sąsają visiems informacijos ištekliams. Vartotojai prisijungę prie sistemos naudodami portalo teikiamą vieno prisijungimo galimybę gali nesunkiai pasinaudoti pageidaujama mokymo ištekliais, prisitaikyti aplinką savo poreikiams.

Įgyvendinimo požiūriu, panaudojus WSRP galima atlikti mokymo išteklių integraciją vartotojo sąsajos lygmenyje. WSRP specifikacija naudoja web servisų mechanizmą perduoti portleto HTML kodą kitam portalui[16]. Jeigu viename portale norima panaudoti kito portalo portletus tuomet pastarajame yra nurodoma kur iš tikrųjų yra reikalingi portletai. Ir į portalo puslapį įdėjus nutolusį portletą puslapyje atvaizduojamas tik HTML kodas, o veiklos logika apdorojanti vartotojo užklausas yra pirmajame portale, kur jis yra realiai įdiegta.

Architektūros variantas panaudojant WSRP pavaizduotas 7 paveiksle. Šiuo atveju naudojami du ar daugiau portalų. Pagrindinis VO portalas yra nekeičiamas, o portletai atliekantys veiksmus su paslauga pagrindiniam portalui perduodami WSRP mechanizmo pagalba.



7 pav. Architektūra naudojant WSRP

Šios architektūros privalumas tas, kad supaprastėja portletų derinimas ir diegimas. Sukūrus naują portletą nereikia keisti vidinės pagrindinio portalo konfigūracijos. Tokiu būdu išsprendžiama problema, kai portletai yra sukurti vienoje mokymo aplinkoje, o juos reikia panaudoti kitoje mokymo aplinkoje. Tačiau šio metodo esminis trūkumas – nenusistovėję standartai, ko pasekoje gali atsirasti nemažai nenumatytų problemų.

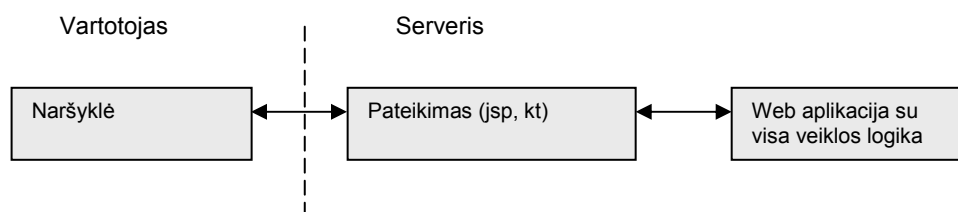
1.3.3 Tradicinės ir į paslaugas orientuotos architektūros palyginimas

Tradicinės architektūros leido išspręsti daugelį problemų ir vystėsi nuo didžiųjų kompiuterių, kliento – serverio, interneto architektūros modelių iki dabar pristatomo POA modelio. POA atėjimas reiškia monolitinių architektūrų ir jų palaikymo problemų pabaigą. Tačiau POA nereikalauja, kad būtų visiškai atsisakoma tradicinių architektūrų, POA gali pridėti papildomo lankstumo patenkinant iškylančius IT poreikius. Lentelėje pateikiame tradicinių ir į paslaugas orientuotų architektūrų palyginimą.

Tradicinės architektūros	SOA architektūros
Projektuojamos visam laikui	Projektuojamos pasikeitimam
Tvirtai susietos	Laisvai susietos, prisitaikančios, lanksčios
Nelanksčios	Sudarytos iš paslaugų
Ilgai kūrimo ciklai	Interaktyvūs ir iteratyvūs kūrimo ciklai
Palaiko homogenines technologijas	Palaiko heterogenines technologijas

1 lent. Tradicinės ir į paslaugas orientuotos architektūros palyginimas

Tradicinės web sistemos architektūra pavaizduota 8 paveiksle. Nors ji plačiai naudojama, tačiau turi vieną didelį trūkumą kai reikia integruoti vieną web programą su kita.

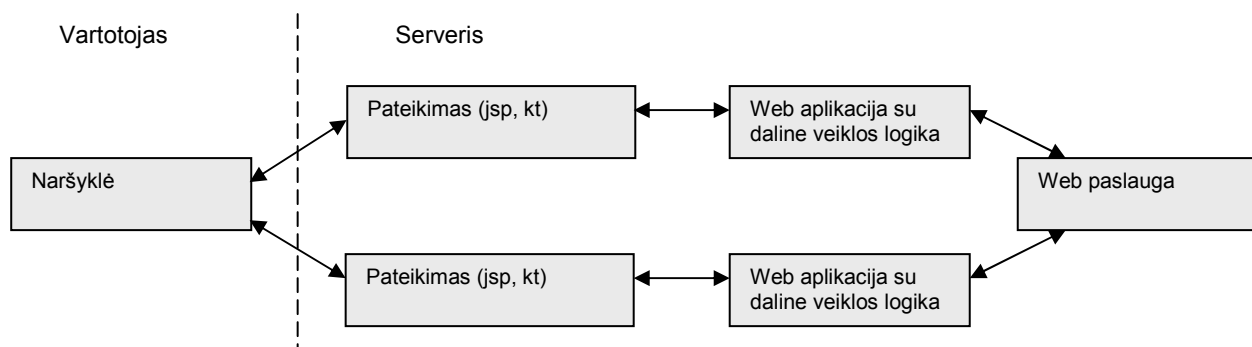


8 pav. Įprastinė web sistemų architektūra

Taikant tokią architektūrą visos funkcijos būtų realizuojamos web programoje. Šios architektūros pagrindinis minusas yra tas, kad web programa gali būti naudojama tame pačiame kompiuteryje kuriame yra portalas. Dėl šios priežasties ši web programa gali panaudota tik vienu

būdu, kadangi web programos neturi tiesioginių sąsajų kurios leistų jomis pasinaudoti iš išorės. Be to, tokiu atveju sudėtinga keliems programuotojams kurti didelę integruotą sistemą.

Žymiai geresnis būdas yra web programos logiką iškelti į web paslaugą. Tuomet atsiranda galimybė tą pačią paslaugą panaudoti keliuose veiklos atvejuose. O integravimas pasidaro nesudėtingas, kadangi web paslaugos turi standartizuotą bendravimo sąsają. Sistemos kūrėjai gali realizuoti paslaugas atliekančias skirtingas funkcijas, o po to jas apjungti į vientisą sistemą.



9 pav. Web paslaugų architektūra

9 paveiksle pavaizduota jog kelios web programos gali naudotis ta pačia web paslauga. Analogišką sprendimą gali suteikti ir Enterprise Java Beans, tačiau ne visi web programų serveriai šią technologiją palaiko. Taip pat gana sudėtinga realizuoti nuotoliniu būdu kviečiamus EJB.

1.4 Analizės išvados

1. Mokymo sistemose dominuoja pasyvūs mokymo objektai. Mokymo kursai surenkami iš objektų į kursą ir pateikiama besimokančiam, kuris savo ruožtu pasyviai skaito, įsisavina atsimena mokymo medžiaga. Tokie žemo interaktyvumo kursai pagal Bloom suteikia tik žinias, bet nesuteikia dalyko supratimo, žinių taikymo įgūdžių, gebėjimų analizuoti bei sintezuoti naujas žinias.
2. Nėra vieningos informacinės sistemos prieigai prie mokymo objektų. Su mokymo objektais pateikiamais failų pavidalu susitvarko daugelis kursų ir mokymo valdymo sistemų, bet jos neturi aiškaus sprendimo, kaip panaudoti mokymo įstaigose esančius programinius produktus, kurie taip pat naudojami kaip mokymo objektai.
3. Aiškinamiesiems objektams naudojami standartizuoti metaduomenys. Pagrindinis mokomųjų objektų metaduomenų standartizacijos tikslas yra kiek galima palengvinti objektų pakartotiną naudojimą. Metaduomenys gali būti panaudoti ir kitokio pobūdžio mokymo objektams.
4. Daugelis autorių savo darbuose ir kuriamose sistemose naudoja web paslaugas, portalus kaip technologijas leidžiančias pasiekti didesnes mokymo galimybes, sprendžiančias integracijos ir lankstumo problemas.
5. Problema yra mokymo objektų sukomponavimas į lankstų scenarijų. Lankstumas suprantamas dvejopai. Pirma, tai to paties sukomponuoto scenarijaus panaudojimas keliose mokymo aplinkose paliekant mokymo išteklius ten, kur jie yra patalpinti. Antra, tai nepriklausomas scenarijų sudarančių komponentų redagavimas kuomet pakeitimai atsispindi visose mokymo aplinkose neatliekant jose papildomų scenarijaus perkomponavimo ar kitų veiksmų. Deja, nėra sukurtų specifikacijų, kaip tai padaryti.

2. Mokslo ir studijų ištekliai ir komponavimo uždavinys į paslaugas orientuotoje architektūroje

Paslaugoms orientuotoje VO turi būti galimybė komponuoti ar surinkinėti mokslo ir studijų paslaugas iš VO prieinamų mokymo objektų. Mokymo objektu laikysime ne tik failų pavidale pateikiamą objektą, bet ir web paslaugas galinčias pasitarnauti mokymui. Tai naujas požiūris į mokymo objektus ir jų naudojimą. Mokymo objektai nėra kopijuojami norint sukurti pedagoginiu požiūriu išbaigtą objektą. Jie yra paskirstyti VO tinkle.

Pirmoje eilėje aptariami mokymo objektai kaip ištekliai sudėtinei paslaugai kurti ir pateikiamas loginis sudėtinės paslaugos neformalus modelis, apimantį plačią įvairovę mokslo ir studijų scenarijų. Taip pat apžvelgiama WS-BPEL paslaugų komponavimo specifikacija ją palyginant su pasiūlytu sprendimu komponuoti mokslo ir studijų paslaugą.

2.1 Mokymo objektai paslaugoms orientuotoje architektūroje

Mokymo objektai virtualioje organizacijoje yra dviejų tipų – aiškinamieji ir vykdomieji.

Aiškinamojo tipo objektai yra tokie, kuriuos apibrėžia vienokio ar kitokio formato failas. Tai gali būti įvairūs failai - doc, txt, pdf, jpg, bmp, mpg, avi, flash, exe ir k.t. Aiškinamojo tipo ištekliai turi būti surandami tinkle ir pateikiami sudėtinės paslaugos gavėjui kaip turinys, kitaip tariant tai pasyvus objektas – medžiaga studijoms. Aiškinamojo pobūdžio objektai plačiau analizuojami 1.2.1 skyrelyje.

Vykdomojo tipo mokomaisiais objektais laikomos virtualios organizacijos web paslaugos, kurios gali būti panaudotos mokymo tikslais. Požiūris į paslaugą, kaip mokymo objektą yra visiškai naujas. Paslaugos gavėjas tikisi tam tikro rezultato, kurį suskaičiuoja ir/arba suformuoja vykdomasis objektas, priėmęs paslaugos gavėjo užklausą.

Vykdomaisiais mokymo objektais iš dalies galima laikyti ir .exe tipo failus, tačiau pagal panaudojimo logiką jie šiame darbe laikomi aiškinamaisiais. Taip yra todėl, kad visi aiškinamojo tipo objektai bet kokių atveju yra parsiuočiami į vartotojo kompiuterį ir jame įvykdomi. O vykdomojo tipo objektai nėra ir negali būti parsiuočiami į vartotojo kompiuterį. Jie pagal vartotojo užklausas yra vykdomi tame serveryje, kur yra patalpinti.

Paslaugos yra paskirstytos VO tinklo serveriuose ir natūraliai saugomos ten, kur jos yra sukuriamos, atnaujinamos, palaikomos, t.y. ištekliai tampa paskirstyti VO tinkle.

Kaip ir aiškinamiesiems objektams paslaugoms gali būti sukuriama metaduomenys, kurie nusakytų kam ši paslauga yra skirta. Vykdomųjų objektų metaduomenys tai iš dalies supaprastinta LOM standarto versija (Priedas nr. 1) numetus atributus neturinčius ypatingos reikšmės virtualios

organizacijos tikslams pasiekti. Taip yra daroma todėl, kad kuriamoje VO nėra akcentuojamas objektų pernešamumas į kitą vykdymo aplinką. Juo labiau nėra poreikio to daryti, kadangi ištekliai yra paskirstyti.

Vykdomųjų objektų metaduomenų laukai ir jų aprašymas pateikiami 2 lentelėje. Prie atributo vardo yra parašytas numeris nurodantis atitikmenį tarp šių metaduomenų ir LOM standarto.

Įrašo pavadinimas	Tipas	Galima reikšmė	Aprašymas	Suteikia reikšmę
1. Bendroji dalis				
1.1. ID	String	Unikalus identifikatorius (UUID)	Unikalus identifikatorius. Identifikuoja metaduomenų įrašą. Pvz.: 20efcf14-6618-11dc-8314-0800200c9a66	Sistema
1.2. Vardas	String	Riboto ilgio tekstas	Dėstytojas suteikia paslaugai vardą pagal savo objektų vardų sistemą.	Dėstytojas
1.4. Aprašas	String	Riboto ilgio tekstas	Trumpas vykdomojo objekto paskirties, funkcionalumo (turinio) aprašas-pristatymas suradusiems tą objektą ir norintiems sužinoti "kas čia" ir kokiame kontekste galėtų būt naudojama.	Dėstytojas
1.5. Raktiniai žodžiai	String	Žodžiai atskirti kableliu	Žodžiai, nusakantys temą, dalyką, sritį, turinį ir pan., bet nenaudoti panaudojamumą nusakančių žodžių.	Dėstytojas
1.9. Tipas	String	Taikomoji Sisteminė Sudėtinė	Taikomoji, tai taikomoji mokslo ir studijų paslauga, turinti portletą, kurią galima naudoti kaip komponentą komponuojant sudėtinę paslaugą; Sisteminė, tai VO tarnybinė paslauga, prieinama VO administratoriui ar paslaugų kūrėjams; Sudėtinė, tai sukomponuota iš taikomųjų paslaugų ir pedagoginiu požiūriu išbaigta paslauga	Dėstytojas
4. Techniniai atributai				
4.1. Formatas	String	Teksto eilutė	Praktiškai bus diegiamos tik Java kalba realizuotos paslaugos	Dėstytojas
4.3. Lokacija (adresas)	String	URL	Paslaugos aprašo WSDL adresas, kuri gali būt naudinga ir reikalinga paslaugų kūrėjams. Tai URL tipo adresas.	Sistema arba Dėstytojas
End Point Reference	EPR	Tam tikrą struktūrą turintis paslaugos adresas	Viena dalis paslaugos URL adresas, kita paslaugos resursų priėjimo raktas.	Sistema
5. Pedagoginiai atributai				
5.1 Interaktyvumo tipas	String	vykdomasis	tai vykdomasis objektas (aktyvus), turintis portletą ir reikalaujantis/laukiantis VO dalyvio veiksmų (pvz., parametrų parinkimo modeliavimo režime).	Sistema
5.2 Turinio tipas	String	Pratimas Modeliavimas Testas Uždavinys	Nusako paslaugos pedagoginę prasmę.	Dėstytojas
5.5 Rolė	String	Dėstytojas Dalyvis Abu	Dėstytojas – paslaugą gali naudoti tik dėstytojas kaip komponentą sudėtinės paslaugos komponavime; Dalyvis – paslaugą gali naudoti VO dalyvis, pvz., tokia visada yra sudėtinė(sukomponuota) paslauga; Abu - tai objektas ir komponavimui ir teikimui.	

2 lent. Vykdomųjų objektų metaduomenys

Nors dauguma šių atributų ir turi atitikmenį LOM standarte, tačiau ne visi jie sutampa pagal prasmę. Šiame metaduomenų modelyje laukų prasmė parinkta tokia, kokia ji reikalinga virtualios organizacijos vykdomiesiems objektams aprašyti.

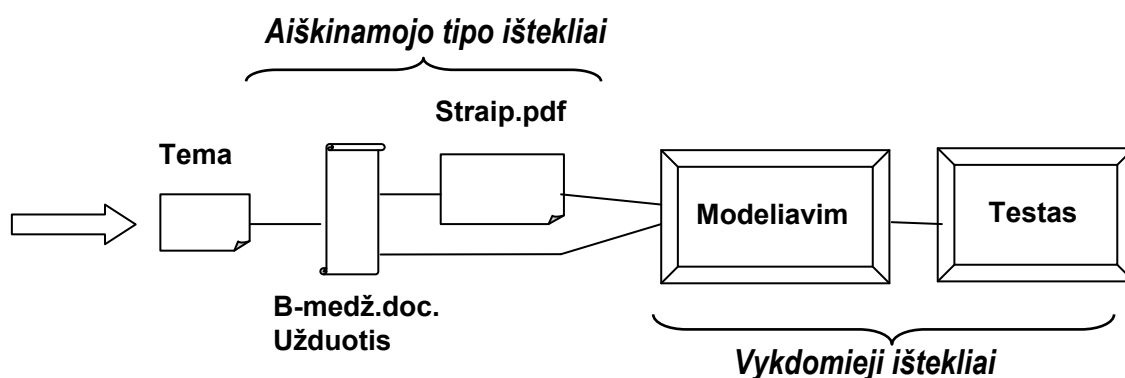
Vykdomuosius objektus nuo aiškinamųjų atskiria 5.1 atributas, pavadintas interaktyvumo lygiu. Jis nusako objekto esminę panaudojamumo galimybę.

Aiškinamųjų ir vykdomųjų objektų metaduomenys skiriasi atributu 4.8. Šis 4-tos kategorijos atributas nusako paslaugos adresą ir jos resurso unikalų raktą. Toks atributas nėra numatytas LOM standarte, tačiau paslaugoms iškviesti tai yra būtinas atributas.

2.2 Sudėtinės mokslo ir studijų paslaugos samprata

Virtualiojoje organizacijoje numatoma galimybė komponuoti sudėtines mokslo ir studijų paslaugas. Sudėtinė suprantama kaip paslauga susidedanti tiek iš aiškinamojo tipo, tiek ir vykdomo tipo išteklių ir vadinama mokymo scenarijumi. Taigi, į sudėtinę paslaugą gali būti įkomponuojami tiek mokymo medžiagą pateikiantys failai, tiek skaičiuojamojo pobūdžio ar modeliavimo uždavinių paslaugos, grindžiamos nuotoliniu būdu vykdomais programiniais moduliais. Schematiškai tai pavaizduota 10 paveikslėlyje. Tokia naujai sukurta paslauga registruojama paslaugų indekse, jai sukuriama metaduomenys, apibrėžiamas paslaugos gyvavimo laikas ir kitos paskirstytų resursų aplinkoje leidžiamos savybės.

Paslaugos vykdymo metu atskiri paslaugos komponentai iškviečiami pagal tos sudėtinės paslaugos gavėjo poreikimą.



10 pav. Neformalus sudėtinės paslaugos modelis

Mokymosi ar studijų atveju neformalus sudėtinės paslaugos modelis galėtų būti interpretuojamas kaip nuoseklus procesas, prasidedantis susipažinimu su tema ir mokomosios medžiagos studijavimu. Pirmiausia studentui suteikiama galimybė susipažinti su dėstytojo nuožiūra tikslingai atrinkta aiškinamąja mokymosi medžiaga bei praktinėmis užduotimis, o po to pateikiami

interaktyvesni vykdomieji mokymo objektai. Pav. 3 pavaizduotos dvi interaktyvios užduotys – modeliavimas ir testas. Modeliavimas paprastai skiriamas gebėjimams ugdyti taikant įsisavintas žinias. Mokymosi paslaugos atvejui yra svarbu gauti besimokančiojo veiklos įvertinimą, liudijanti tai, kad mokymo tikslas daugiau ar mažiau pasiektas. Vertinimo-testavimo paslaugos komponentas gali būti įkomponuojamas ir kitoje vietoje, tarkim prieš modeliavimą.

Mokymo scenarijus gali būti nuoseklus arba ne. Dėstytojas savo nuožiūra komponuodamas scenarijų pasirenka kaip tas scenarijus turėtų būti peržiūrimas. Jeigu scenarijus nuoseklus tuomet parenkama tvarka kuria mokymo objektai turėtų būti peržiūrimi. Na, o jeigu nenuoseklus tuomet studentas gali pats kuria tvarka peržiūrėti mokymo objektus.

Komponavimo paslauga turi įgyvendinti pakankamai sudėtingą veiklos modelį ir skiriama VO dalyviams, turintiems skirtingas roles bei teises. Skiriame tris VO dalyvių roles:

- dėstytojas,
- tyrėjas,
- autorius.

Dėstytojo rolei priskiriama sudėtinės studijų paslaugos komponavimo veikla. Tai pagrindinis uždavinys sukombonuoti paslaugą iš paskirstytų komponentų, nustatyti paslaugos gyvavimo ciklą ir užregistruoti ją VO paslaugų registre. Paskirstyti VO tinklo ribose aiškinamojo tipo objektai (duomenys, dokumentai) ir įvykdomi objektai (paslaugos) prieinami VO tinkle per išteklių valdymo paslaugas. Kadangi numatoma jau sukombonuotų scenarijų peržiūra, komponavimą dėstytojais gali derinti su paslaugos redagavimu, papildymu ar atnaujinimu naujais komponentais.

Tyrėjas, būdamas užregistruotas kaip VO dalyvis, gauna jam leistinas paslaugas ir gali jomis naudotis. Jei tai yra studijų paslauga, tai skaityti numatytus dokumentus, atlikti praktines užduotis ar testus, o jeigu tai specializuota modeliavimo paslauga, tai atlikti modeliavimą pagal formuluoto uždavinio sprendimo eigą.

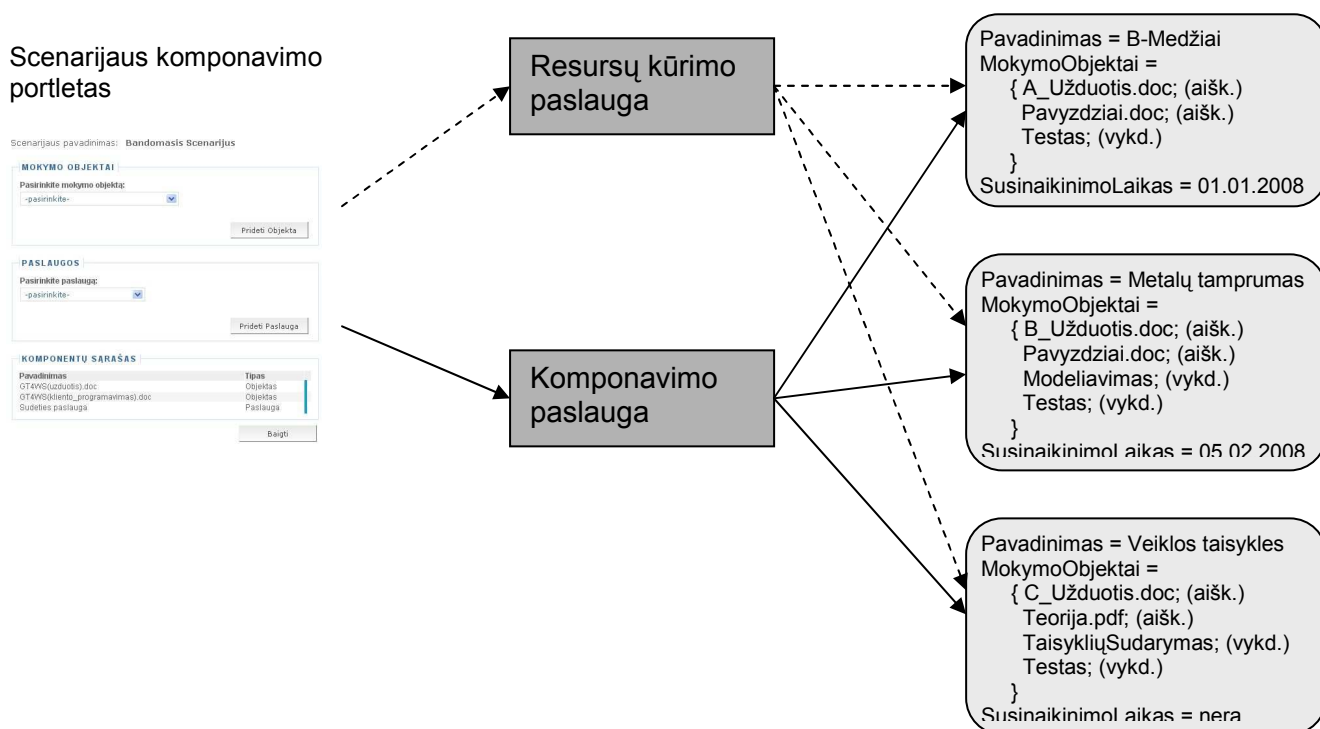
Į autoriaus funkcijas įeina aprašomojo ir vykdomojo tipo mokymo objektų sudarymas ir įkėlimas į VO aplinką, kartu jiems suformuojant metaduomenis.

2.3 Sudėtinių paslaugų komponavimo būdai

Paslaugų komponavimas gali būti suprantamas dvejopai. Šiame darbe paslaugų komponavimas remiasi pedagoginiu požiūriu. Čia komponavimas atliekamas iš kitų mokymo paslaugų į labiau išbaigtą pedagoginiu požiūriu paslaugą. Paslaugos kūrėjas pagal metaduomenų aprašus parenka mokslo ir studijų išteklius ir sukombonuoja į paslaugą savo tikslui pasiekti. Tačiau įgyvendinimo požiūriu nauja web paslauga nėra sukuriamas. Sukuriamas tik komponavimo

paslaugos taikymo atvejais, o paslauga lieka ta pati. Tai leidžia pasiekti anksčiau aprašytas WSRF mechanizmas. Kiekvienas naujas mokymosi scenarijus yra naujas paslaugos resursas. Ir tokia paslauga gyvuoja tol, kol jos nesunaikina jos dėstytojas arba kol nesibaigia jos gyvavimo laikas. Gyvavimo laiko funkcionalumą taip pat suteikia WSRF.

11 paveikslėlyje pavaizduota schema atitinka daugelio resursų paslaugos modelį aprašytą 1.3.1 skyrelyje. Resursų kūrimo paslauga (fabriko paslauga) atsakinga už naujo resurso sukūrimą. Paveiksle pavaizduoti sukurti 3 resursai. Resurso kūrimo paslauga sukūrusi resursą nesuteikia jam konkrečių reikšmių, o grąžina sukurtos paslaugos galinio taško nuorodą (angl. Endpoint reference), kurioje yra paslaugos adresas ir sukurto resurso raktas, paslaugos klientui (šiuo atveju tai portletas). Klientas gavęs galinio taško nuorodą jau nebesinaudoja resursų kūrimo paslauga, o kreipiasi tiesiai į sukurtąjį resursą. Taip resursui gali būti suteikiamos konkrečios reikšmės vadinamos resurso savybėmis. Konkrečiu komponavimo paslaugos atveju resurso savybės, tai naują mokslo ir studijų paslaugą nusakantys duomenys – paslaugos pavadinimas, susinaikinimo laikas ir ją sudarantys mokymo objektai (ne patys objektai, o nuorodos į juos). Norint pasinaudoti konkrečia mokslo ir studijų paslauga reikalinga galinio taško nuoroda, kuri leistų priėti prie pageidaujamo resurso.



11 pav. Mokymosi paslaugos komponavimas naudojant WSRF

Kitas komponavimo požiūris remiasi WS-BPEL specifikacija. Ši specifikacija numato atskirų web paslaugų komponavimą į vieną verslo procesą [11]. Toks sprendimas plačiai taikomas įmonėse apjungiant įmonės padalinių veiklas ar netgi skirtingų įmonių veiklas į bendrą procesą. Šiuo atveju yra sukuriama nauja web paslauga. Tačiau šis metodas netinka mokymo paslaugoms komponuoti, kadangi mokymo paslaugoje nėra poreikio web paslaugas jungti į procesą. Tačiau negalima teigti, kad šis metodas negali būti pritaikytas mokslo ir studijų virtualioje organizacijoje. Pagal WS-BPEL specifikaciją sukomponuotos paslaugos sudaro naują paslaugą, kuri gali būti įtraukta į mokymosi paslaugą, kaip vykdomasis mokymosi objektas, tačiau jis sudarytas ne iš vienos, o iš kelių vartotojui nepastebimų web paslaugų.

Sukomponuotos paslaugos peržiūra taip pat yra dvejopa. Pagal WS-BPEL specifikaciją sukomponuota paslauga yra nauja web paslauga, todėl jai iškviešti sukuriamas klientas. Ir toks klientas kuriamas kiekvienai sukomponuotai paslaugai. Yra keletas sprendimų web paslaugų klientams generuoti, pagal web paslaugų aprašus (WSDL). Vienas žinomiausių sprendimų yra Xydra [17]. Tačiau šis įrankis sugeba generuoti vartotojo sąsajas tik labai paprastoms paslaugoms. Žinant tai, kad mokymosi paslaugomis naudosis tyrėjai pravartu sukurti interaktyvesnes vartotojo sąsajas. Na, o paslaugai sukomponuotai į scenarijų gali būti sukurtas universalus klientas, kadangi scenarijaus komponavimui naudojama viena ir ta pati paslauga naudojanti skirtingus resursus saugančius atskirus mokymo scenarijus.

Jeigu paslauga būtų sukomponuota tik iš aiškinamųjų mokymo objektų pagal IMS-LD specifikaciją tuomet būtų galima pritaikyti jau egzistuojančius peržiūros įrankius. Vienas iš populiariausių įrankių yra CopperCore. Tai įrankis skirtas peržiūrėti mokymo objektus supakuotus pagal IMS-LD su galimybe mokymo scenarijui priskirti grupes ir vartotojus, bei valdyti jų mokymosi eigą [12]. Tačiau jis nepalaiko tokių scenarijų į kuriuos įkomponuotos web paslaugos, kadangi neturi priemonių joms peržiūrėti. Be to CopperCore įrankis dirba tik su lokaliais objektais. Jis neišspręstų lankstumo problemos, kuomet pasikeitus kuriam nors mokymo objektui turi būti pakeistos ir sukomponuotos mokymo paslaugos.

2.4 Komponavimo posistemės komponentai

Kadangi mokslo ir studijų virtuali organizacija yra kuriama į paslaugas orientuotos architektūros pagrindu, todėl visi be išimties veiklos komponentai realizuojami web paslaugomis, o vartotojo sąsajoms rekomenduojama naudoti portletus.

Komponavimo posistemėi galima išskirti kokius komponentus:

„*Komponavimo paslaugos fabrikas*“ ir „*Komponavimo paslauga*“ – tai paslaugos skirtos kurti mokymo scenarijus. Jos atsakingos už scenarijaus, kaip atskiro paslaugos resurso sukūrimą,

atnaujinimą ir naikinimą.

„*Vykdomųjų objektų metaduomenų paslauga*“ – tai web paslauga skirta valdyti vykdomųjų objektų metaduomenų sukūrimą, atnaujinimą ir sunaikinimą.

„*Aiškinamųjų objektų metaduomenų paslauga*“ – tai web paslauga skirta valdyti aiškinamųjų objektų metaduomenų sukūrimą, atnaujinimą ir sunaikinimą.

„*Objektų saugojimo paslauga*“ atlieka aiškinamųjų objektų patalpinimą į VO.

Vartotojo sąsajos užtikrina tokie portletai:

„*Komponavimo portletas*“

„*Peržiūros portletas*“

„*Vykdomųjų objektų metaduomenų portletas*“

„*Aiškinamųjų objektų metaduomenų portletas*“

„*Objektų saugyklos portletas*“

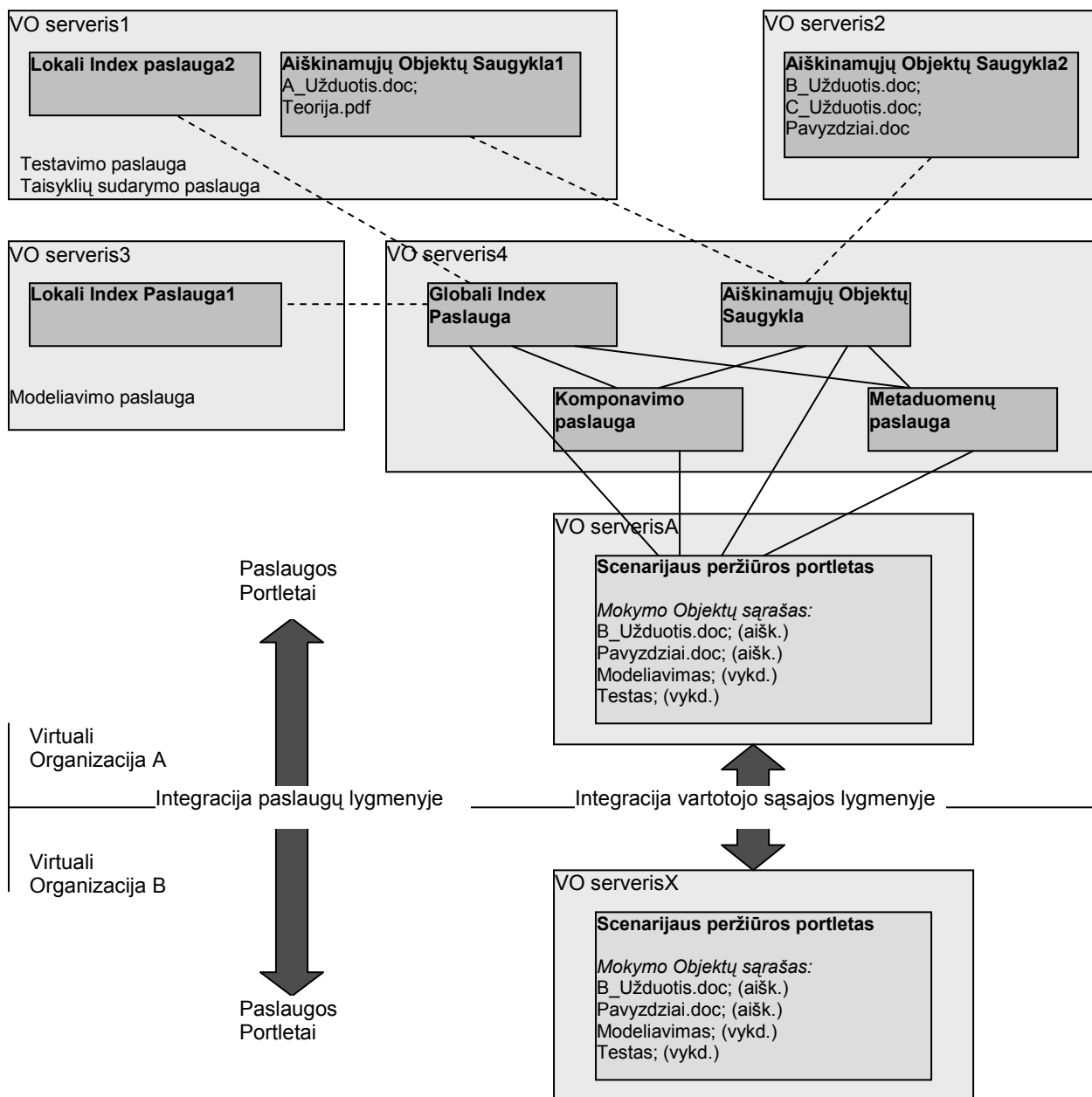
2.5 Resursų paskirstymo pavyzdys

Kaip minėta 1.1 skyrelyje šio darbo tikslas sukurti komponavimo sprendimą leidžiantį sukurti lanksčią mokymosi paslaugą, kuomet pasikeitus ją sudarantiems mokymo objektams nebereikia perkomponuoti pačios paslaugos. Šią savybę leidžia užtikrinti į paslaugas orientuota architektūra kurioje komponentai yra paskirstyti ir laisvai susieti. Objektų pasiskirstymą galima atvaizduoti schemeje (12 pav.). Šioje schemeje paimtas sukomponuoto scenarijaus pavyzdys ir jį sudarantys objektai yra paskirstyti. VO ServerisA yra portletas, kuriame pateikiamos sukomponuotos paslaugos. Jos yra surandamos paslaugų registro pagalba (Globali index paslauga). Šiame registre publikuojamos visos virtualios organizacijos paslaugos. Potlete pasirinkus pageidaujamą scenarijų yra pateikiami jį sudarantys mokymo objektai. Šių objektų sąrašas yra išgaunamas iš komponavimo web paslaugos. Kadangi naudojamas tik mokymo objektų sąrašas, o ne patys objektai, todėl objektus galima bet kada modifikuoti nekeičiant jų identifikatoriaus. Pavyzdžiui, serveryje VO Serveris2 pasikeitus mokymo objektų turiniui arba VO Serveris3 pasikeitus modeliavimo paslaugos įgyvendinimui, scenarijaus perkomponuoti nereikia. Vartotojas bet kuriuo momentu gali atsidaryti mokymo objektą. Jeigu tai aiškinamasis mokymo objektas tuomet vartotojui pagal objekto adresą iš aiškinamųjų objektų saugyklos bus atsiųsta naujausia objekto versija. Jeigu tai vykdomasis objektas tuomet pagal objekto adresą bus atidarytas atitinkamas portletas ir jis naudosis pakeista paslauga.

Taip pat šioje (12 pav.) schemeje pavaizduota ir integracija tarp kelių virtuliųjų organizacijų. Ji gali būti atliekama dviem būdais – vartotojo sąsajos lygmenyje ir web paslaugų lygmenyje. Vartotojo sąsajos lygmenyje integracija vykdoma WSRP mechanizmo pagalba.

Pateikiant Virtualios organizacijos B portalui reikiamų portletų vaizdą, šios VO dalyviai gali matyti Virtualios Organizacijos A mokymo paslaugas ir objektus.

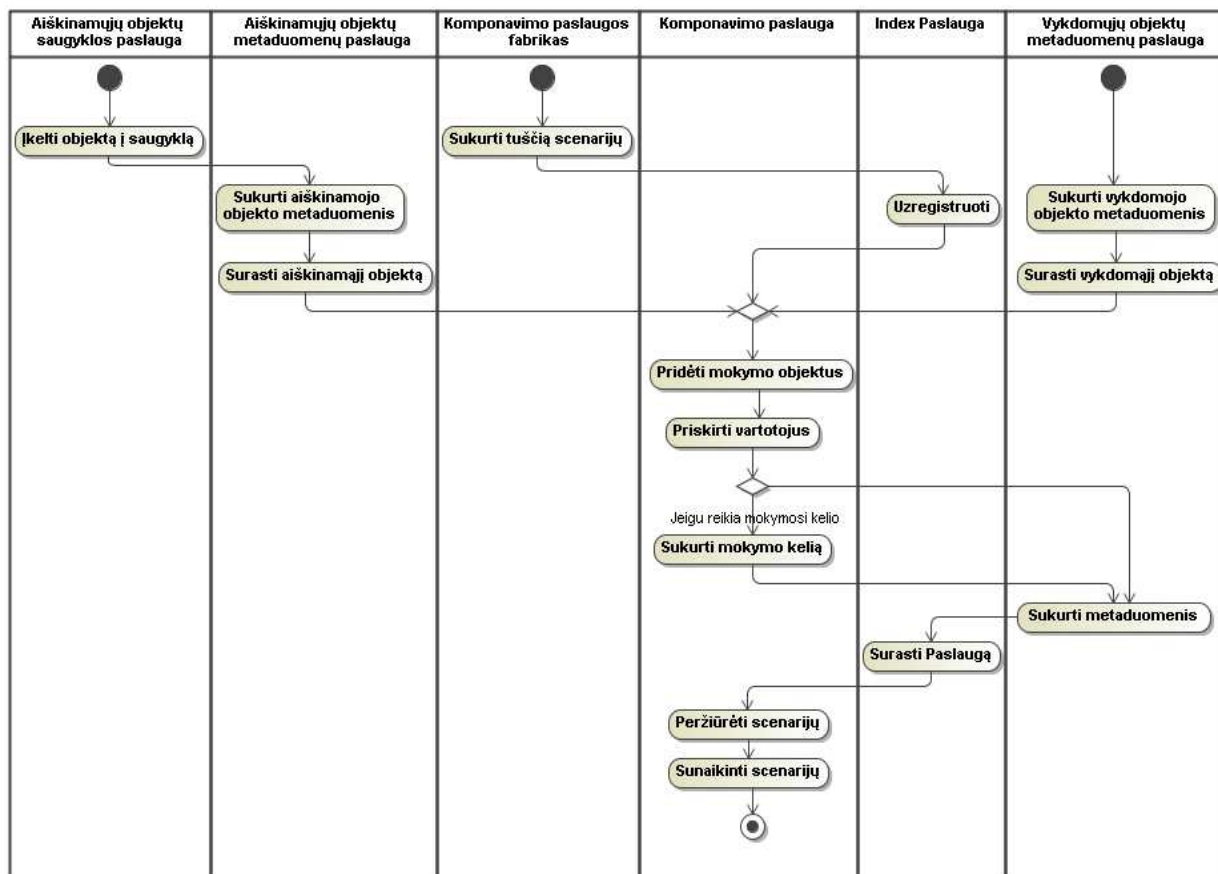
Kitokia integracija gali būti atlikta tiesiogiai kviečiant kitos VO paslaugas. Tačiau ši integracija sudėtingesnė, kadangi reikia kurti savus portletus. Iš kitos pusės ji suteikia didesnę laisvę pritaikant paslaugas savo poreikiams.



12 pav. Mokymo Objektų paskirstymo schema

2.6 Komponavimo veiklos procesas

Komponavimo veiklos procesas pavaizduotas diagramoje (13 pav.). Komponavimo paslaugai reikalingos funkcijos paskirstytos 5-ioms web paslaugoms, diagramoje vaizduojamoms stulpeliais. Tokiu būdu aiškiai parodoma, kokios web paslaugos už kokią veiklą yra atsakingos.



13 pav. Komponavimo paslaugos veiklos proceso diagrama

Pradžioje veiklos procesai šakojasi į 3 lygiagrečius procesus. Viename subprocese į objektų saugyklą įkeliami mokymo objektai ir aprašomi jų metaduomenys. Kitame subprocese aprašomi web paslaugų metaduomenys. Trečiajame subprocese sukuriamas mokymo paslaugos atskiras atvejis ir atliekama paslaugos registracija. Vėliau atliekama aiškinamųjų ir vykdomųjų objektų paieška. Kuris subprocesas bus vykdomas pirmiau, o kuris vėliau nusprendžia pats vartotojas ir bet kuriuo momentu gali persijungti nuo vieno proceso prie kito. Kai įvykdomi scenarijaus sukūrimo ir vieno ar kito objekto paieškos subprocesai, tada galima komponuojamam scenarijui priskirti mokymo objektus. Priskyrus objektus komponavimas nebaigiamas, tačiau atliekamas vartotojų priskyrimas scenarijui. Toks priskyrimas leidžia valdyti teises kokie vartotojai gales peržiūrėti sukomponuotą scenarijų. Sekantis žingsnis gali būti mokymo kelio sukūrimas. Šio

proceso metu objektai išrikiuojami pagal tai, kokia tvarka jie turi būti peržiūrėti. Tokiu atveju vartotojas peržiūrėdamas mokymo scenarijų iš pradžių matys tik vieną galimą peržiūrėti objektą, o sekantis objektas bus parodytas tik tada, kai bus peržiūrėtas pirmasis objektas. Jeigu mokymosi kelias nenaudojamas tada visi scenarijuje esantys objektai bus matomi iš karto. Svarbus žingsnis yra sukomponuotos paslaugos aprašymas metaduomenimis, be kurio scenarijaus peržiūra būtų negalima. Vartotojui peržiūrinčiam paslaugas yra pateikiamas sukomponuotų ir užregistruotų registre paslaugų sąrašas. Paslaugos peržiūros žingsnyje vartotojas peržiūri scenarijų ir į jį įkomponuotus objektus. Jeigu sukomponuotoje paslaugoje nustatytas mokymosi kelias, tada objektai peržiūrimi tik nustatyta seka, o jeigu nenustatytas, tada vartotojas gali peržiūrėti objektus bet kuria tvarka. Paskutiniame žingsnyje scenarijus yra sunaikinamas. Jis gali būti atliekamas rankiniu būdu arba atliktas automatiškai pasibaigus nustatytam laikui.

3. Komponavimo posistemės reikalavimų analizė

3.1 Komponavimo paslaugos funkcijos

Funkciniai reikalavimai komponavimo paslaugai formuluojami pagal dalyvių roles:

Dėstytojų priskiriamos funkcijos:

- Peržiūrėti egzistuojančias paslaugas (scenarijų sąrašą),
- Komponuoti studijų paslaugą (kurti paslaugos scenarijų),
- Redaguoti scenarijų,
- Šalinti scenarijų,
- Įkomponuoti aiškinamuosius ir vykdomuosius mokymo objektus,
- Priskirti vartotojų grupę sukomponuotam scenarijui,
- Sukurti mokymo kelią

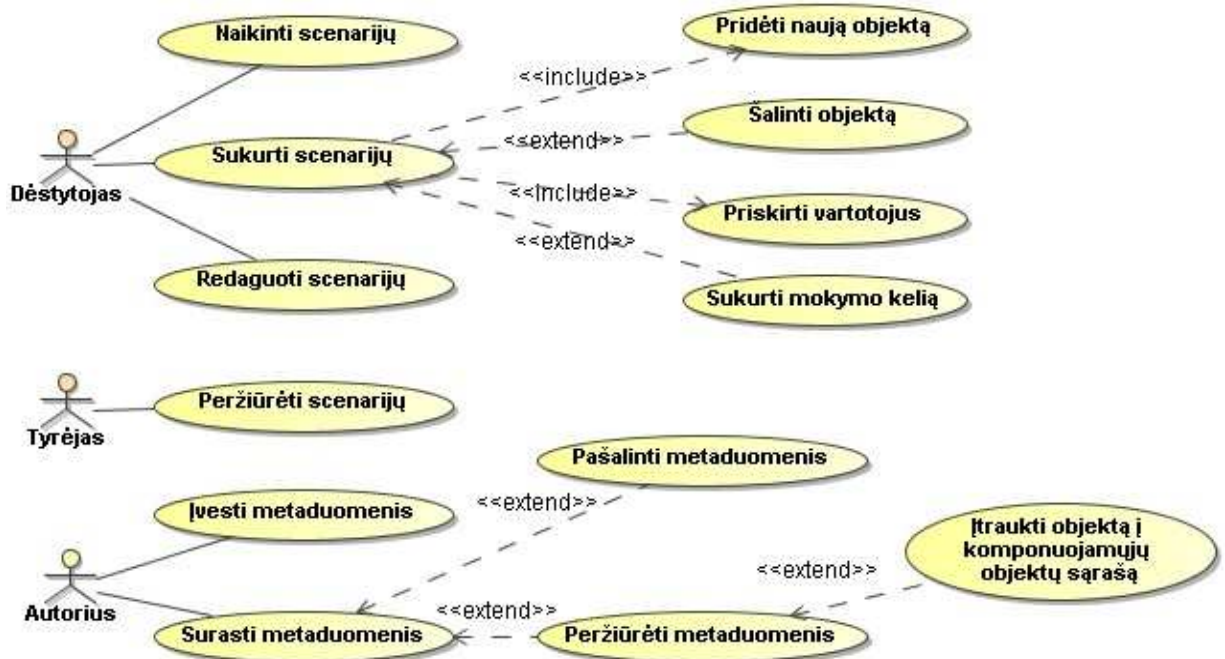
Tyrėjo funkcija:

- Pasinaudoti bet kuria jam leistina mokslo ir studijų paslauga

Autorius priskiriamos funkcijos:

- Sukurti metaduomenis aiškinamiesiems objektams,
- Sukurti metaduomenis web paslaugoms.
- Surasti metaduomenis
- Pašalinti metaduomenis
- Peržiūrėti metaduomenis

Šie funkciniai reikalavimai specifikuoti panaudojimo atvejų diagramoje (14 pav.).



14 pav. Panaudojimo atvejai

Panaudojimo atvejo „Sukurti Scenarijų“ vykdymo metu yra sukuriamas naujas mokymo paslaugos egzempliorius, suteikiant jam galiojimo laiką, jeigu tai laikoma reikalinga. Mokymo objektų įkomponavimą bei pašalinimą nusako panaudojimo atvejai „Pridėti Naują Objektą“ ir „Šalinti Objektą“. Į scenarijų pridedamas/šalinamas ne pats mokymo objektas, o nuoroda į objektą. Patys mokymo objektai šalinimo metu nepašalinami iš objektų saugyklos. Jeigu pridedama web paslauga, tuomet šio panaudojimo atvejo metu yra priskiriamas ir sąsajos portletas, kuris turėtų būti atvaizduojamas scenarijaus peržiūros metu. Panaudojimo atvejo „Priskirti vartotojus“ metu scenarijui yra priskiriami vartotojai ar visa jų grupė. Panaudojimo atvejis „Sukurti mokymo kelią“ leidžia sukurti mokymo kelią jeigu to reikia sukomponuotam scenarijui.

„Naikinti Scenarijų“ ir „Redaguoti Scenarijų“ yra išvestiniai panaudojimo atvejai suteikiantys papildomą scenarijaus valdymo funkcionalumą ir užtikrinantys jo gyvavimo ciklą. Panaudojimo atvejo „Peržiūrėti Scenarijų“ metu VO dalyviui leidžiama peržiūrėti sukomponuotą mokymo paslaugą. Įgyvendinimo požiūriu jis yra pats sudėtingiausias. Jo metu atliekama scenarijų filtracija pagal vartotojo teises, įgyvendinamas paslaugų peržiūros mechanizmas.

Panaudojimo atvejai „Įvesti metaduomenis“, „Pašalinti metaduomenis“, „Surasti metaduomenis“, „Peržiūrėti metaduomenis“ leidžia atlikti įprastinius metaduomenų valdymo veiksmus. Panaudojimo atvejis „Įtraukti į komponuojamųjų objektų sąrašą“ įtraukia pasirinktą objektą į komponuojamųjų objektų sąrašą, t.y. šis vykdomasis objektas tampa prieinamas

komponavimo portlete. Komponuojamų objektų sąrašas yra laikinas galiojantis tik vieno prisijungimo prie portalo metu. Atsijungus šis sąrašas panaikinamas.

Panaudojimo atvejis „Įkelti mokymo objektą į saugyklą“ priklauso VO išteklių valdymo paslaugoms, tačiau jis pavaizduotas ir prie komponavimo paslaugos, kadangi objektų įkėlimas į VO aplinką tam tikra prasme įeina į komponavimo veiklos procesą.

3.2 Nefunkciniai reikalavimai

Tai nefunkciniai reikalavimai komponavimo paslaugai, kurie nusako paslaugos įgyvendinimo aplinką VO paslaugų architektūroje infrastruktūroje:

- Scenarijus komponuojamas iš aiškinamojo tipo objektų, pateikiamų failų pavidale, ir iš paslaugų, pateikiamų vykdomų objektų pavidale.
- Aiškinamieji objektai gali būti įvairių formatų - .doc, .html, .mpeg, .jpg, .avi ir t.t.
- Komponavimo paslauga turi rasti ir įtraukti į sudėtinę paslaugą aiškinamuosius ir vykdomuosius objektus nepriklausomai nuo to, kokioje VO tarnybinėje stotyje jie yra saugomi ir įdiegti.
- Vykdomieji objektai-paslaugos turi būti talpinamos tarnybinių stočių GT4 Java paslaugų konteineryje ir užregistruotos paslaugų registre - GT4 IndexService.
- Objektams turi būti suformuoti metaduomenų atributai.
- Kiekvienas vykdomasis objektas-paslauga turi turėti bent vienas portletą.

4. Komponavimo posistemės projektas

Komponavimo posistemės projektas atliekamas laikantis į paslaugas orientuotos architektūros principo[20].

- **Identifikacija**
- **Specifikacija**
- **Realizacija**

Pirmiausia atliekama web paslaugų identifikacija ir dalykinės srities specifikuojimas. Apie tai buvo trumpai buvo aprašyta ankstesniuose skyriuose, tačiau čia apie tai rašoma plačiau. Specifikacijos dalyje detalizuojamos sąsajos parodant kokias konkrečias funkcijas jos atlieka. Realizacijos dalyje parodoma kokia tvarka vykdomos web paslaugų sąsajų funkcijos norint įvykdyti funkcinis reikalavimus.

4.1 Komponavimo posistemės paslaugų identifikacija

4.1.1 Komponavimo paslaugos funkcinė-struktūrinė schema

Sudėtinių paslaugų komponavimo posistemė turi būti natūraliai integruojama į VO programinės įrangos architektūrą, kurią patogų vaizduoti loginiais sluoksniais. Funkcinė-struktūrinė komponavimo posistemės schema, pavaizduota 15 paveikslėlyje, gauta priskiriant tam tikros paskirties komponavimo posistemės paslaugas atitinkamiems sluoksniams.

Apatinis sluoksnis tai tinklo ir grid infrastruktūros sluoksnis, kuriame yra sisteminė paslaugoms orientuotos architektūros platforma (GT4). Komponavimo paslauga naudojami Globus Index servisu, Java konteineriu, saugumo sistema, o taip pat šiame sluoksnyje yra VO dalyvių direktorija LDAP, reliacinė duomenų bazė. Nors scheme ji pavaizduota viena, tačiau įgyvendintos atskiros duomenų bazės aiškinamiesiems objektams, metaduomenims ir pačiai komponavimo paslaugai. Komponavimo paslaugos duomenų bazė saugo įkomponuotų mokymo objektų ir paslaugų pasiekiamumo informaciją. Index servise yra informacija apie VO esančias paslaugas.

Vidurinis tai paslaugų sluoksnis, kuriame pavaizduoti komponavimo paslaugos programiniai komponentai. Komponavimo paslaugos branduolį sudaro „Komponavimo paslaugos fabrikas“ ir „Komponavimo paslauga“. Komponuojami mokymo objektai nuskaitomi pasinaudojant objektų saugojimo paslauga, o paslaugos iš web paslaugų metaduomenų saugyklos. Sukomponuotam scenarijui galima priskirti vartotojų grupes (rolės poaibis). Šis priskyrimas nustato priėjimo teises prie scenarijaus.

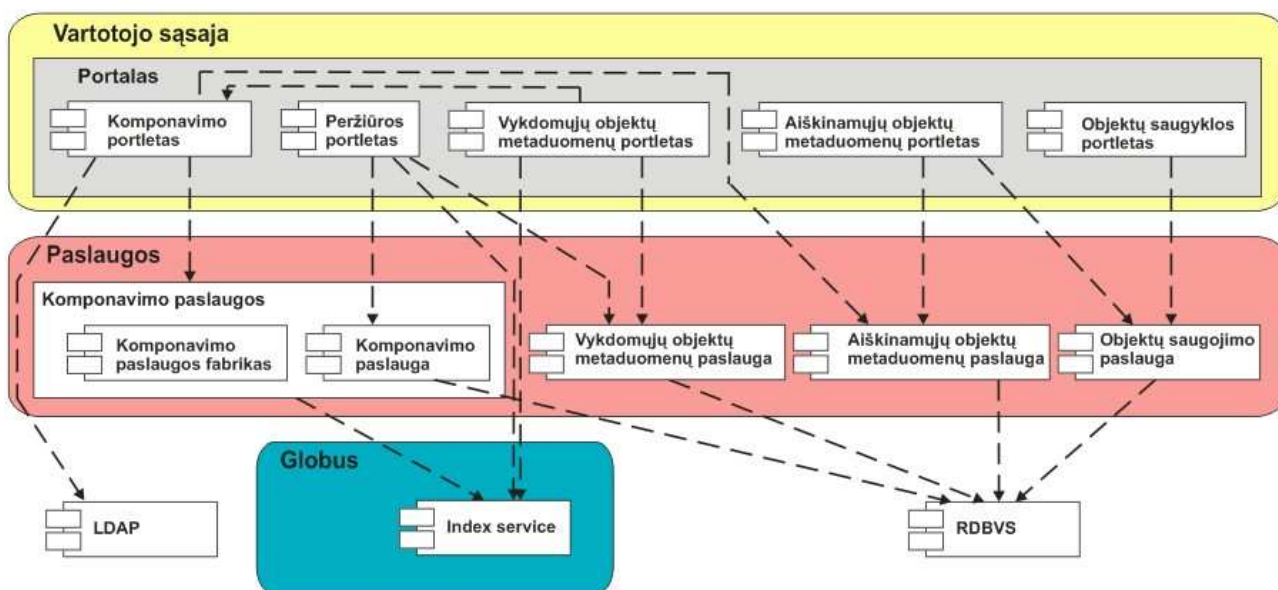
Šiam sluoksniui taip pat priklauso ir web paslaugų metaduomenų saugojimo paslauga

leidžianti metaduomenis išsaugoti, pašalinti, surasti ir peržiūrėti. Objektų paieška atliekama pagal objekto pavadinimą, aprašymą, raktinius žodžius ir edukacinius metaduomenų laukus.

Viršutinis yra taikymo sluoksnis, kuriame parodytas VO portalas su komponavimo sistemai reikalingais portletais, įgyvendinančiais VO dalyvių sąsajas į paslaugas. Komponavimo uždaviniui reikalingi 4 portletai.

Dėstytojai naudojami „Komponavimo portletu“ sudėtinės paslaugos sukūrimui. Komponuojant paslaugą pirmiausia yra sukuriama naujas komponavimo paslaugos atvejis – resursas. O į sukurtąjį resursą yra įkomponuojami mokymo objektai arba paslaugos. Sukomponuotam scenarijui priskiriami vartotojai. Šie vartotojai yra nuskaitomi iš LDAP duomenų bazės. Taip pat scenarijui gali sukurtas mokymo kelias.

„Metaduomenų kūrimo portletas“ vartotojams leidžia atlikti metaduomenų sukūrimą pasirinktai paslaugai, atlikti paslaugų paiešką pagal metaduomenis. Taip pat šis portletas leidžia įtraukti pasirinktą portletą į komponuojamųjų paslaugų sąrašą. Šitoks sprendimas leidžia integruoti metaduomenų paiešką ir scenarijaus komponavimą vartotojo sąsajos lygmenyje išnaudojant portalo galimybes. Integravimas vartotojo sąsajos lygmenyje leidžia supaprastinti komponavimo uždavinio įgyvendinimą.



15 pav. Komponavimo paslaugos struktūrinė schema

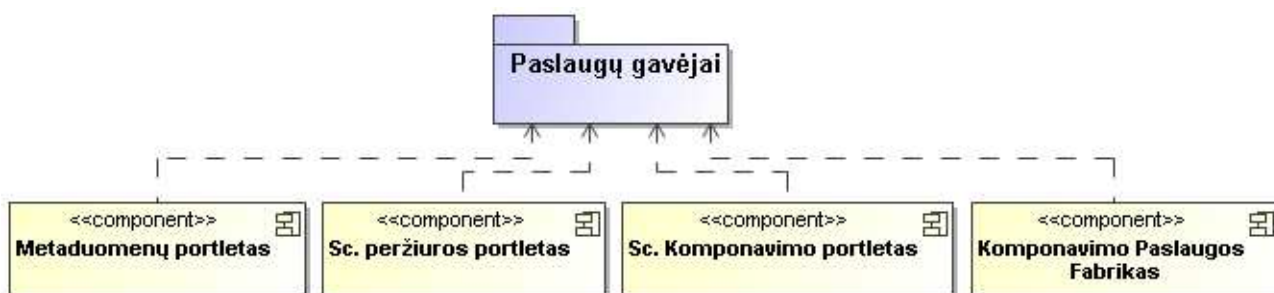
Kitas yra „Peržiūros portletas“ – sukomponuotos paslaugos (scenarijaus) peržiūrai. Peržiūros portletas pirmiausia pagal prisijungusio dalyvio grupę/es atrenka jam prieinamas paslaugas, dalyvis peržiūrai pasirenka vieną iš jų. Toliau peržiūrai naudojama „Komponavimo paslauga“ pateikia paslaugą sudarančių objektų sąrašą ir dalyvis gali peržiūrėti kiekvieną iš jų. Jeigu tai pasyvus mokymo objektas tuomet jis pasiunčiamas, jeigu tai paslauga, tuomet atidaromas tos paslaugos portletas.

„Objektų saugyklos portletas“ yra objektų saugyklos vartotojo sąsaja, leidžianti vartotojui atlikti mokymo objektų valdymo veiksmus. (Plačiau apie objektų saugyklą 1 skyriuje). Objektų saugojimą duomenų bazėje atlieka „Objektų saugojimo paslauga“, o patys objektai saugomi reliacinėje duomenų bazėje.

Sekančiame skyrelyje pateikiama komponavimo paslaugos vartotojų sąsajų (portletų) įgyvendinimo schema bei pateikiamas komponavimo paslaugos portleto pavyzdys VO portale.

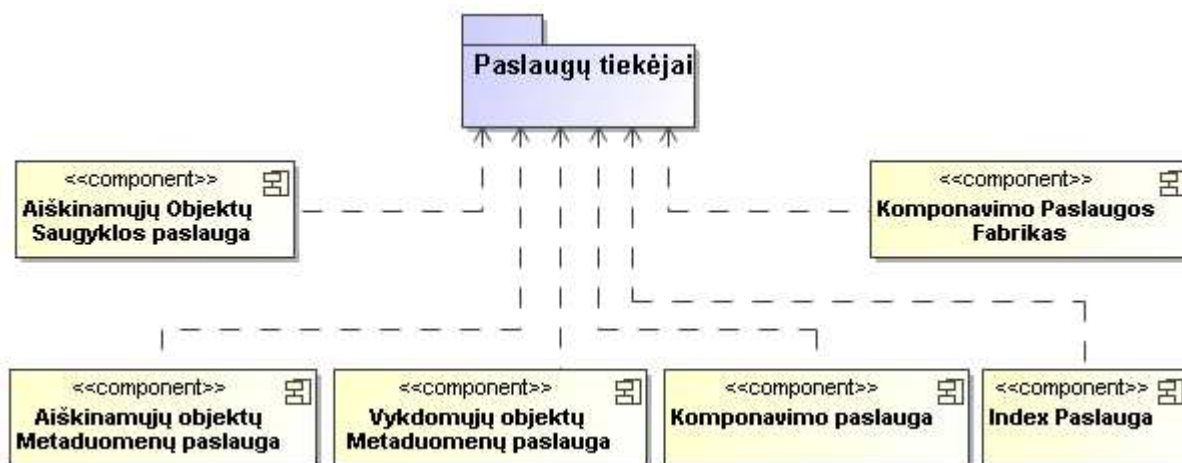
4.1.2 Paslaugos tiekėjų ir paslaugos gavėjų identifikacija

Numatyti keturi paslaugos gavėjai – metaduomenų portletas, Sc.peržiūros portletas, Sc. Komponavimo portletas ir Komponavimo paslaugos fabrikas. Pirmieji trys komponentai priklauso vartotojo sąsajai. Komponavimo paslaugos fabrikas yra paslauga, priklausanti ir tiekėjų ir gavėjų kategorijoms. Jis priklauso paslaugų gavėjams todėl, kad naudoja kitą paslaugą – Index paslaugą.



16 pav. Paslaugų gavėjai

17 paveikslėlyje pavaizduoti paslaugų tiekėjai.



17 pav. Paslaugų tiekėjai

4.1.3 Dalykinės srities modelis

Komponavimo paslaugos dalykinės srities modelyje išskirtos 4 duomenų klasės(18 pav):

Scenarijus – klasė atsakinga už informacijos saugojimą apie mokymo scenarijų. Šią esybę nusako trys atributai – resurso raktas, indikatorius ar scenarijus turi mokymosi kelią ir resurso gyvavimo laikas. Kiti galimi atributai kaip pavadinimas, aprašymas yra nenaudojami, kadangi ši informacija bus saugoma metaduomenyse.

ScenarijausKomponentas – saugo informaciją apie scenarijui priklausančius aiškinamuosius ir vykdomuosius objektus. Ši klasė turi tokius atributus:

Pavadinimas – Komponento pavadinimas.

Aprašymas – Komponento aprašymas.

Tipas – Indikatorius nusakantis, koks komponento tipas. Galimos dvi reikšmės – „objektas“ ir „paslauga“. Indikatorius „objektas“ parodo, kad tai aiškinamasis objektas, o „paslauga“ parodo, kad tai vykdomasis objektas.

Adresas – Komponento adresas. Jeigu tai aiškinamasis objektas, tuomet šio lauko reikšmė yra lygi objektu saugyklos adresui. Jeigu tai vykdomasis objektas, tuomet šio lauko reikšmė lygi paslaugos adresui.

VarduErdve – Jeigu paslauga turi daug resursu, tuomet šiame lauke yra išsaugoma tos paslaugos rakto vardu erdve.

LokalusVardas – Jeigu paslauga turi daug resursu, tuomet šiame lauke yra išsaugoma tos paslaugos rakto lokalus vardas.

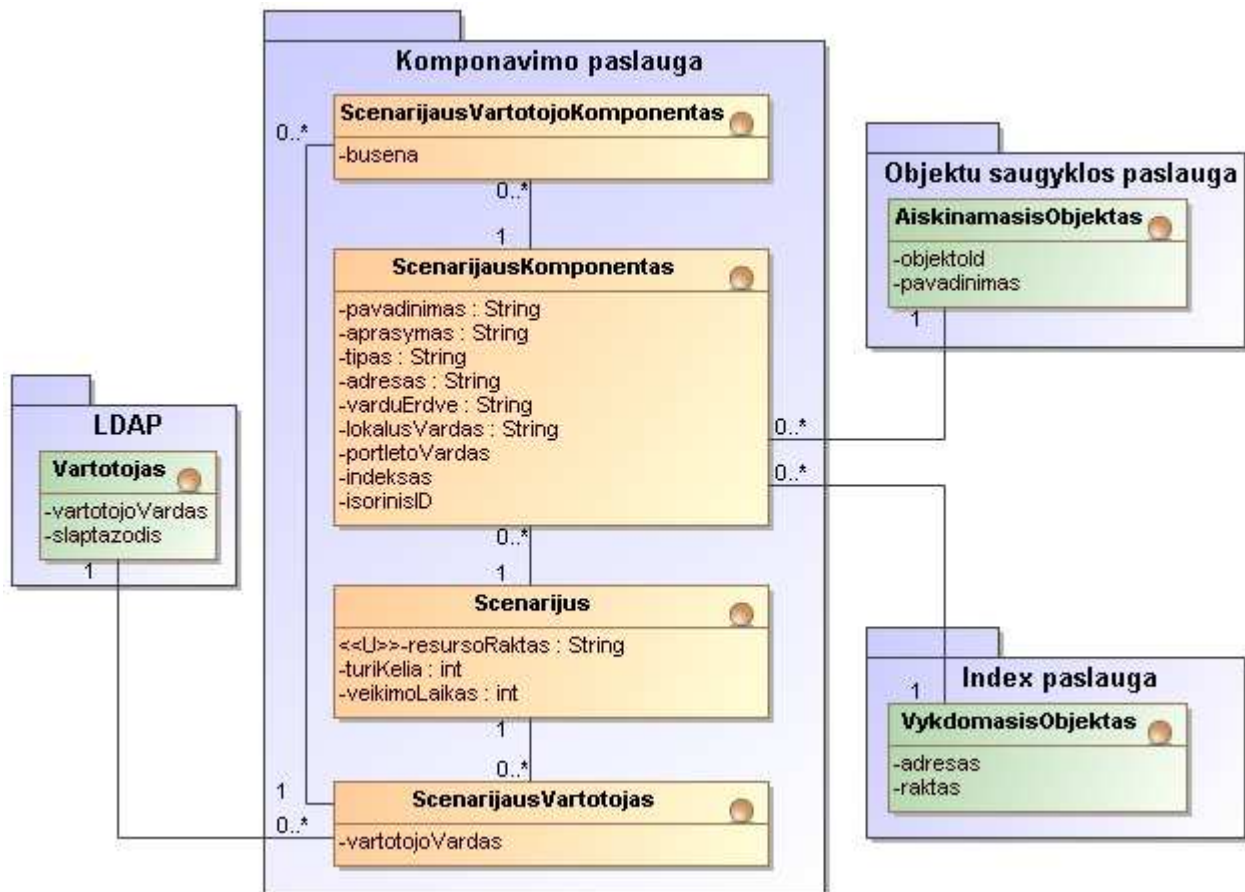
PortletoVardas – Portleto, kuris turėtų būti atidaromas paslaugos peržiūros metu, vardas.

Indeksas – Skaitinė reikšmė parodanti komponento vietą scenarijuje.

IsorinisId – Komponento išorinis identifikatorius. Jeigu tai aiškinamasis objektas, tuomet šio lauko reikšmė lygi to objekto identifikatoriui objektu saugykloje. Jeigu tai paslauga, tuomet ši reikšmė lygi resurso raktui arba paslaugos adresui, priklausomai nuo to ar paslauga daugiaresursė ar neturinti resursų.

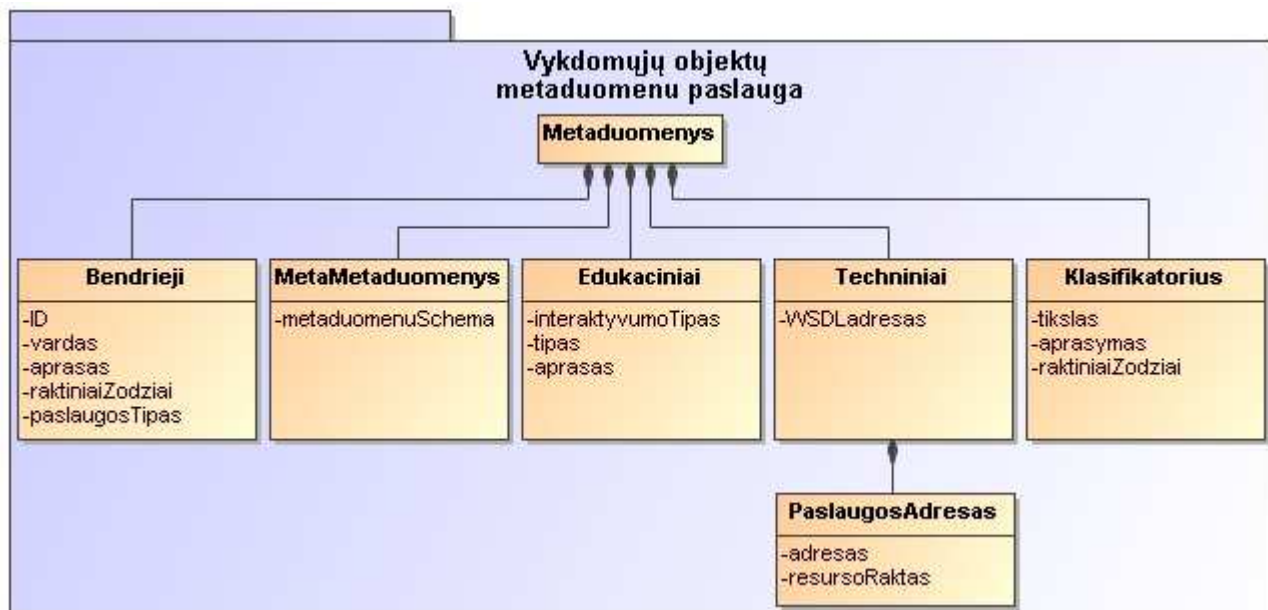
ScenarijausVartotojas – Saugo informacija apie vartotojus kurie turi teisę peržiūrėti tam tikrą scenarijų.

ScenarijausVartotojoObjektas – Tai ryšio lentelė susiejanti esybes „ScenarijausKomponentas“ ir „ScenarijausVartotojas“. Ji reikalinga tuo atveju, kai scenarijui sudarytas mokymosi kelias ir būtina sekti ar konkretus vartotojas scenarijui priklausantį objektą jau peržiūrėjo ar ne.



18 pav. Komponentavimo paslaugos dalykinės srities klasių diagrama

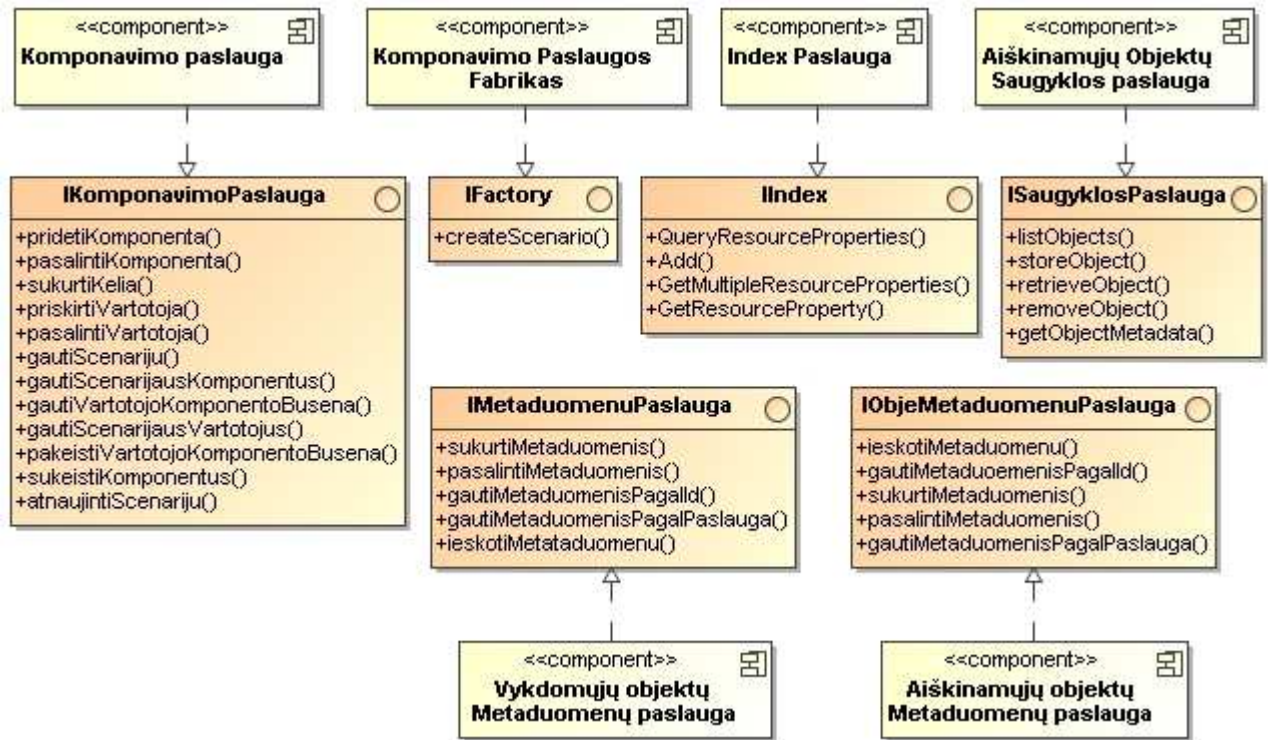
Sekančiame paveikslėlyje pavaizduotas virtualios organizacijos metaduomenų modelis. Esysbės tarpusavyje susietos 1:1 ryšiais, kadangi metaduomenys yra unikalūs kiekvienam VO mokymo objektui, tačiau išskaidyti į loginiu požiūriu atskiras dalis. Šis modelis atitinka vykdomųjų objektų metaduomenų Lentelę Nr. 2, žr. šio dokumento poskyrį 1.5.



19 pav. Metaduomenų modelis

4.2 Paslaugų specifikacija

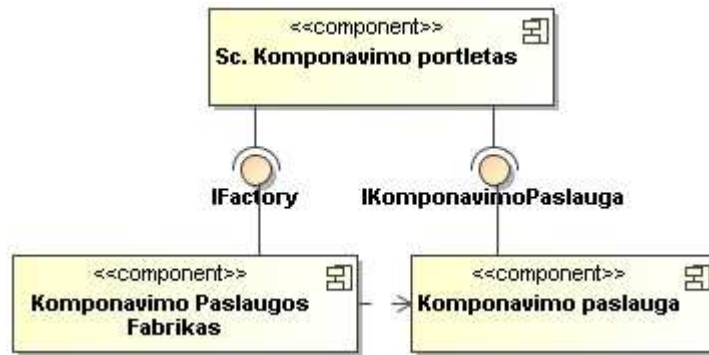
20 paveikslėlyje pavaizduota komponavimo paslaugos, bei vykdomųjų ir aiškinamųjų objektų metaduomenų paslaugų sąsajų specifikacija. T.y. konkrečios funkcijos, kurias teikia minėtosios paslaugos. Įvardinti metodai atsispindės ir web paslaugų aprašo failuose (WSDL).



20 pav. Paslaugų sąsajų specifikacija

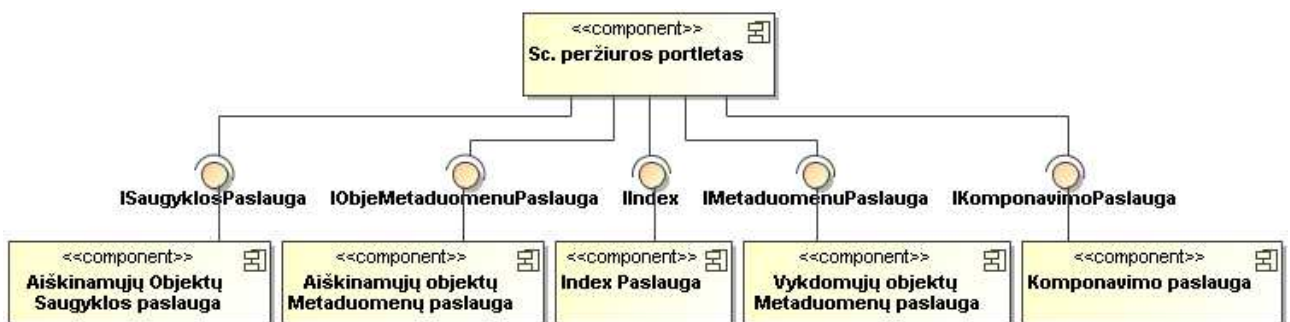
Sekančiuose paveiksluose pavaizduota, kokie paslaugų gavėjai kokius paslaugų tiekėjus naudoja. Tai viena iš pagrindinių projekto dalių parodanti sistemos realizuojamos paslaugomis komponentų ryšius. Šioje vietoje atsispindi į paslaugas orientuotos architektūros panaudojimas.

21 paveikslėlyje pavaizduota paslaugos gavėjo „Sc. Komponavimo portletas“ naudojamos paslaugos. Šio atveju naudojami tik du paslaugos tiekėjai. Komponavimo paslaugos fabrikas naudojamas sukurti naujam Komponavimo paslaugos atvejui. O komponavimo paslauga skirta komponuoti mokymo scenarijų. Komponavimo paslaugos fabrikas ir Komponavimo paslauga susieti priklausomybiniu ryšiu taip parodant, kad fabrikas sukuria naują komponavimo paslaugos atvejį.



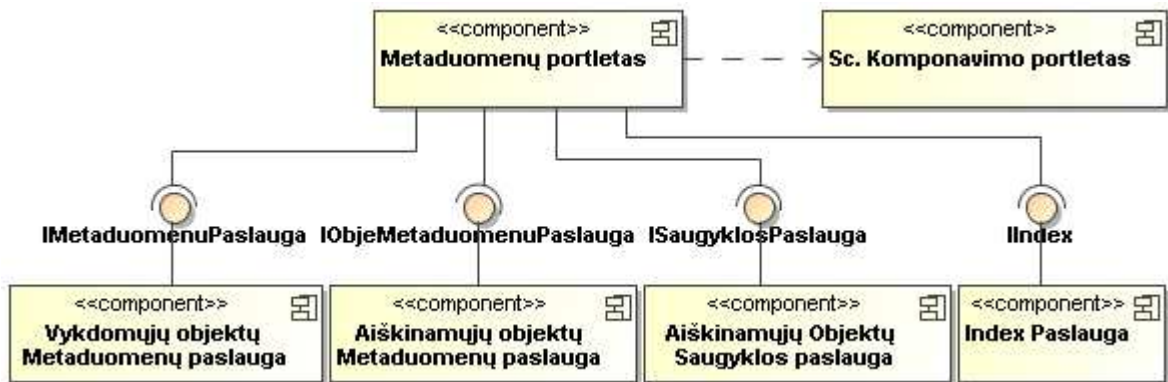
21 pav. Paslaugos gavėjo „Sc. komponavimo portletas“ naudojamos paslaugos

22 paveikslėlyje pavaizduoti paslaugos gavėjo „Sc. Peržiūros portletas“ naudojamos paslaugos. Komponavimo paslauga šiuo atveju skirta gauti sukomponuotą scenarijų duomenis. Index paslauga naudojama surasti sukomponuotoms mokymo paslaugoms. Aiškinamųjų ir Vykdomųjų objektų metaduomenų paslaugos naudojamos nuskaityti įkomponuotų mokymo objektų metaduomenims. Aiškinamųjų objektų saugyklos paslauga naudojama paimti aiškinamiesiems objektams.



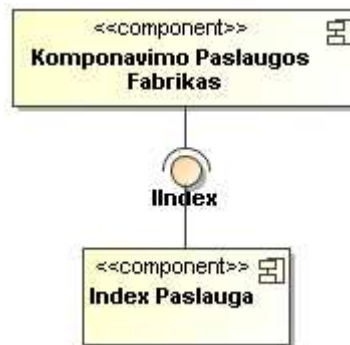
22 pav. Paslaugos gavėjo „Sc. Peržiūros portletas“ naudojamos paslaugos

Paslaugos gavėjo „Metaduomenų portletas“ naudojamos paslaugos pavaizduotos 23 paveikslėlyje. Šiuo atveju Index paslauga naudojama surasti visoms VO esančioms paslaugoms. Aiškinamųjų objektų saugyklos paslauga skirta surasti visiems virtualioje organizacijoje esantiems pasyviems mokymo objektams. Aiškinamųjų ir Vykdomųjų objektų metaduomenų paslaugos naudojamos išsaugoti ir surasti mokomųjų objektų metaduomenims.



23. Paslaugos gavėjo „Metaduomenų portletas“ naudojamos paslaugos

24 paveikslėlyje pavaizduotas paslaugos gavėjo „Komponavimo paslaugos fabrikas“ paslaugos atvejo registravimui naudojama Index paslauga.



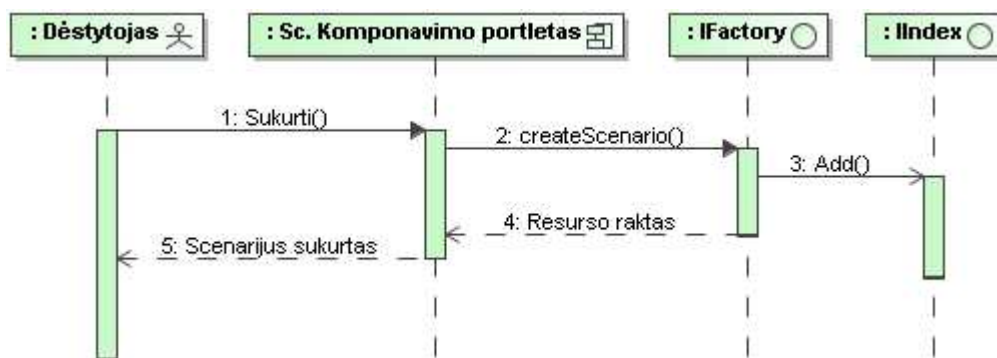
24 pav. Paslaugos gavėjo „Komponavimo paslaugos fabrikas“ naudojama paslauga

4.3 Paslaugų realizacija

Sekančiose diagramose pavaizduotos sekos tarp web paslaugų sąsajų vykdant išskirtus panaudojimo atvejus. Šios diagramos detaliai neparodo, kokios operacijos ir kaip jos vykdomos web servisų viduje. Jos parodo, kaip web servisų operacijos yra sukomponuojamos į loginę seką įvykdančią panaudojimo atvejį.

4.3.1 Panaudojimo atvejo „Sukurti Scenarijų“ sekų diagrama

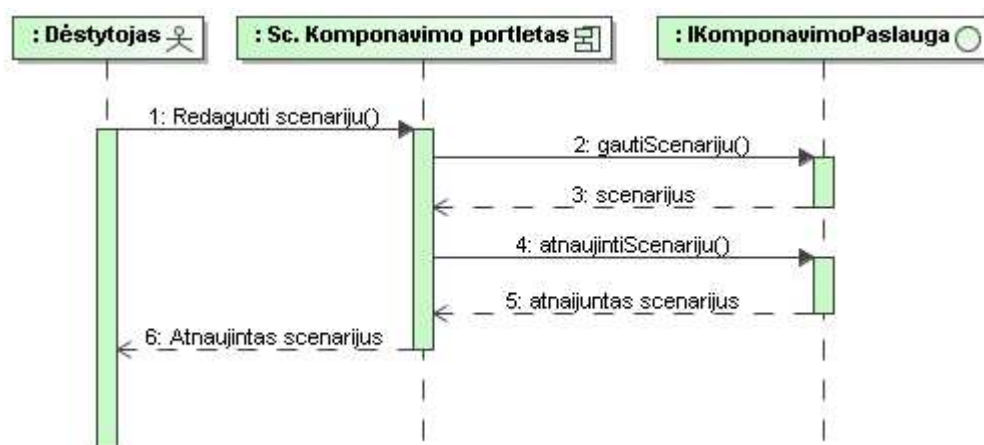
Į scenarijaus sukūrimo panaudojimo atvejį įeina panaudojimo atvejis „Pridėti naują objektą“ ir „Priskirti vartotojus“. Tai pat scenarijaus sukūrimo metu gali būti įvykdyti pasirenkami panaudojimo atvejai „Šalinti objektą“ ir „Priskirti vartotojus“.



25 pav. Panaudojimo atvejo „Sukurti Scenarijų“ sekų diagrama

4.3.2 Panaudojimo atvejo „Redaguoti Scenarijų“ sekų diagrama

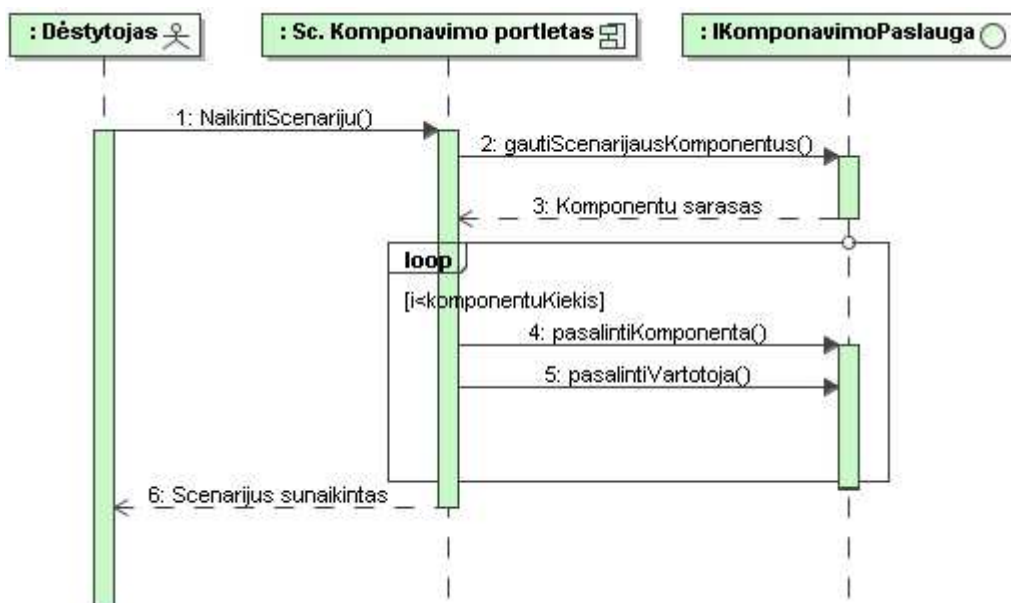
Scenarijaus redagavimo metu nėra kuriamas naujas resursas, o surandami pasirinkto resurso duomenys ir atnaujinamas esantis scenarijus.



26 pav. Panaudojimo atvejo „Redaguoti Scenarijų“ sekų diagrama

4.3.3 Panaudojimo atvejo „Naikinti Scenarijų“ sekų diagrama

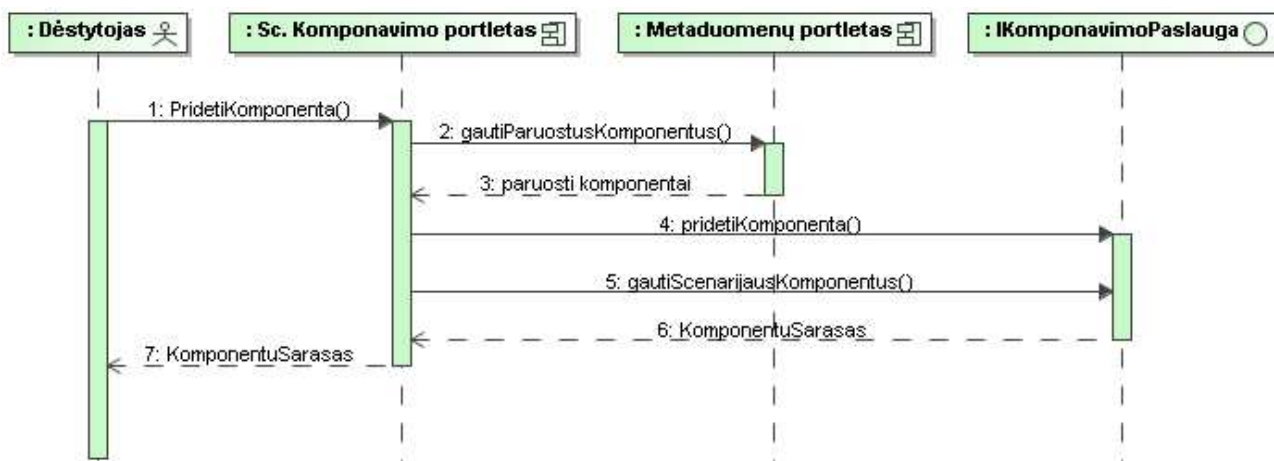
Scenarijaus naikinimo sekų diagramoje esminis epizodas yra tai, jog prieš naikinant scenarijų turi būti panaikinti visi scenarijui priklausantys komponentai.



27 pav. Panaudojimo atvejo „Naikinti Scenarijų“ sekų diagrama

4.3.4 Panaudojimo atvejo „Pridėti Naują Objektą“ sekų diagrama

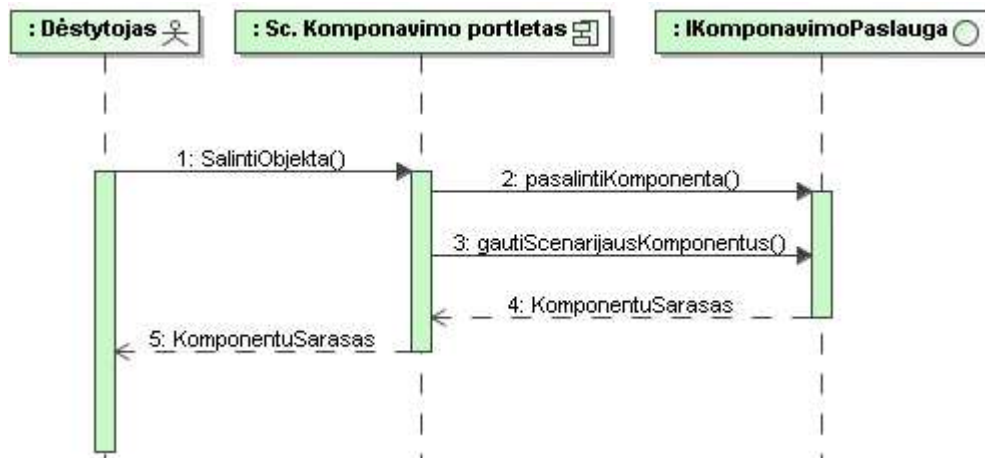
Komponentų ir interfeisų tarpusavio sąveika objektų pridėjimo metu pavaizduotos 28 paveikslėlyje. Šios sąveikos galimos tik tada, kai prieš tai buvo įvykdytas scenarijaus sukūrimo arba redagavimo veiksmas.



28 pav. Panaudojimo atvejo „Pridėti Naują Objektą“ sekų diagrama

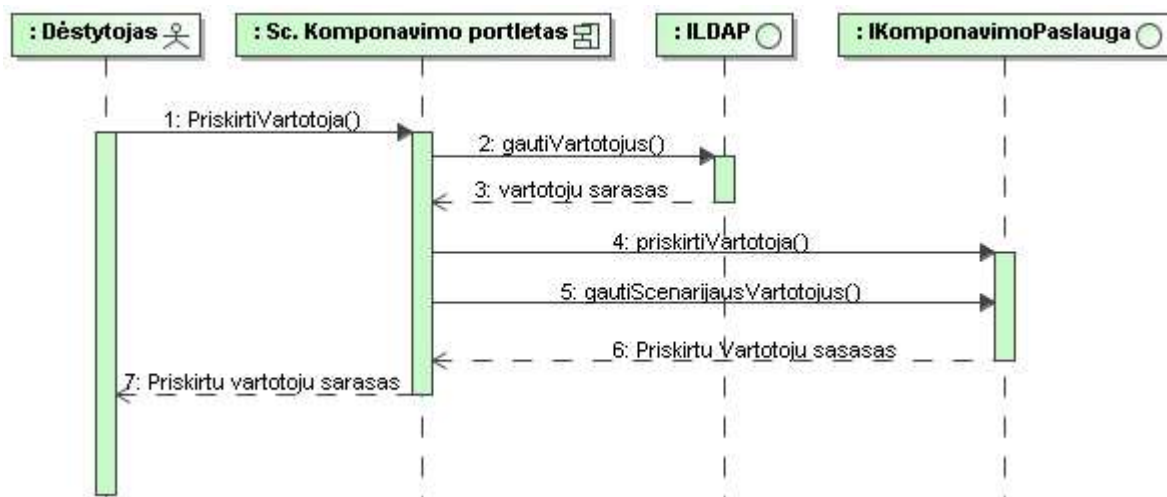
4.4.5 Panaudojimo atvejo „Šalinti Objektą“ sekų diagrama

Analogiškai objektų pridėjimui objektų šalinimo veiksmas gali būti atliekamas tik po scenarijaus sukūrimo ar redagavimo.



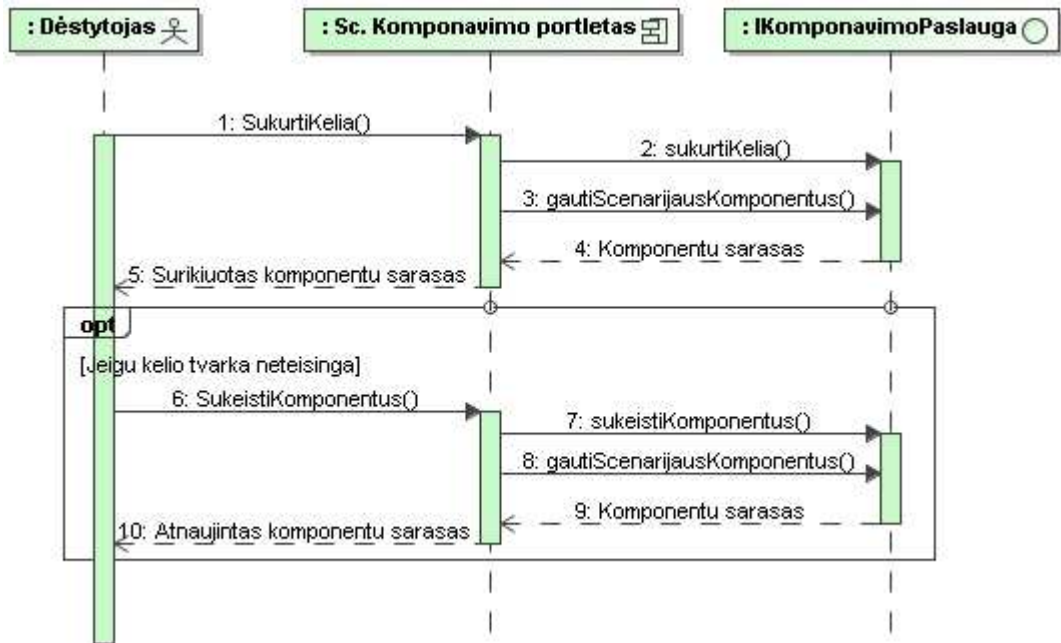
29 pav. Panaudojimo atvejo „Šalinti Objektą“ sekų diagrama

Sekančiame 30 paveikslėlyje pavaizduota vartotojų priskyrimo scenarijui seka. Pirmiausia jie yra nuskaitymi iš LDAP duomenų bazės, o po to jie priskiriami scenarijui.



30 pav. Panaudojimo atvejo „Priskirti Vartotojus“ sekų diagrama

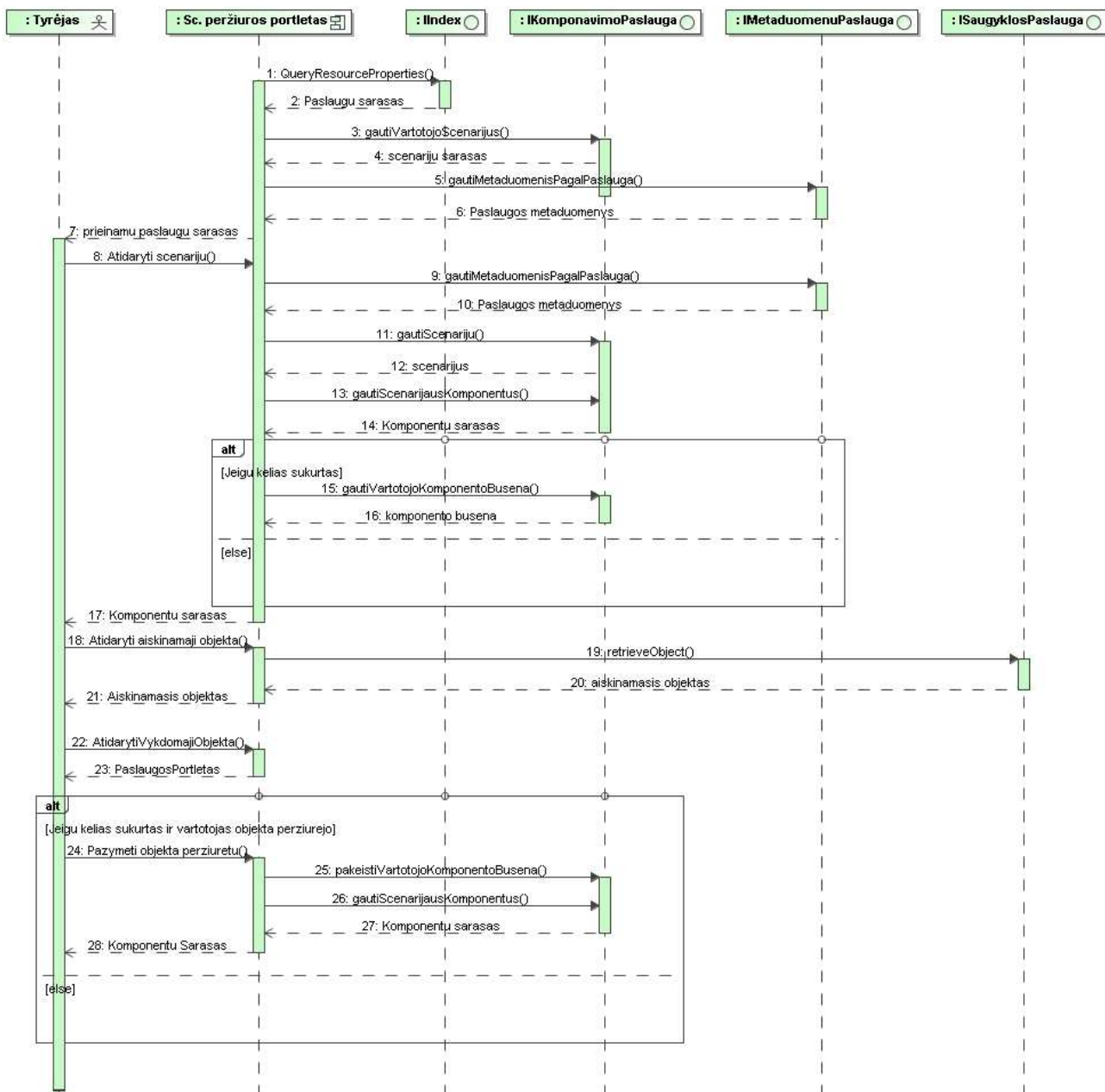
31 paveikslėlyje pavaizduotas mokymo scenarijaus sukūrimas. Pirmiausia jis sukuriamas suteikiant objektams pradines reikšmes, o po to tų komponentų vykdymo seka gali būti pakeista.



32 pav. Panaudojimo atvejo „Sukurti Mokymo Kelią“ sekų diagrama

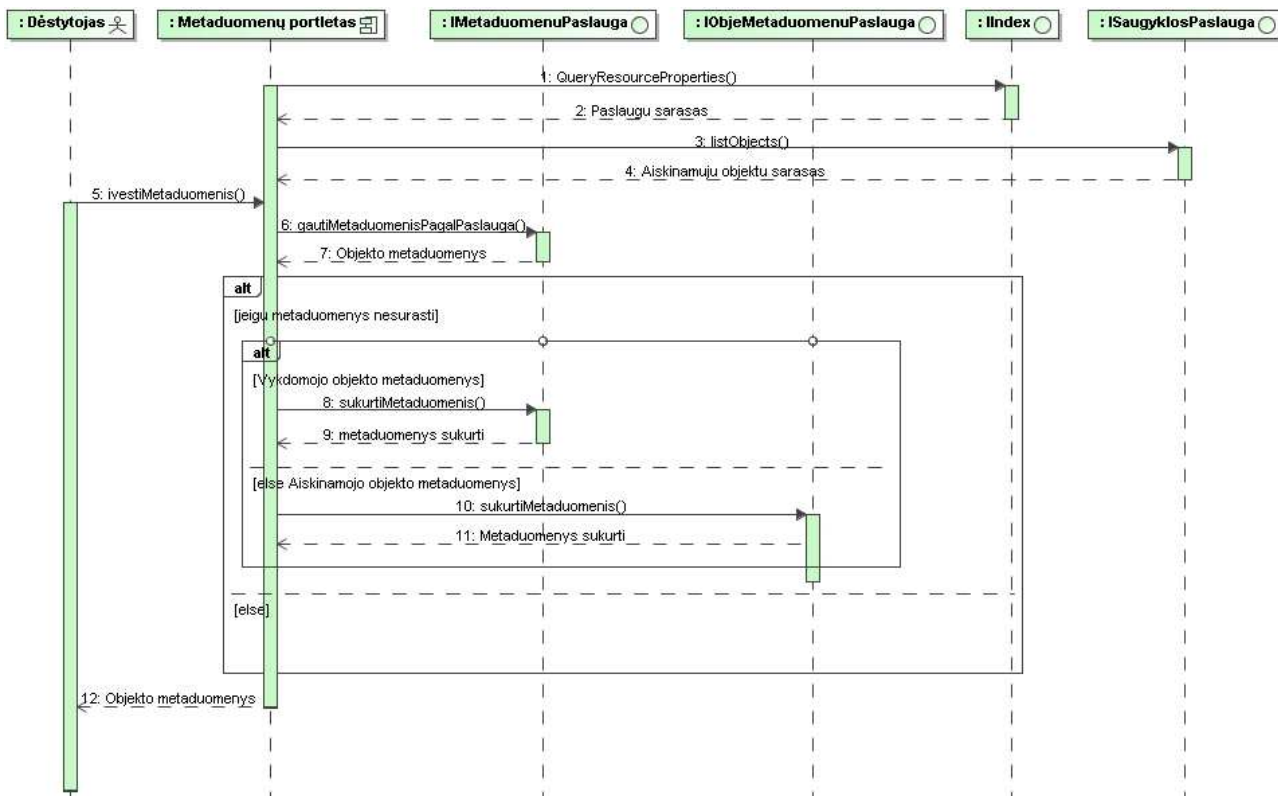
4.3.6 Panaudojimo atvejo „Peržiūrėti Scenarijų“ sekų diagrama

Scenarijaus peržiūros metu pirmiausia surandamos sukomponuotos paslaugos. Pasirinkus paslaugą surandami tos paslaugos metaduomenys ir jai priklausantys mokymo objektai. Toliau pagal vartotojo norą gali būti parsisiunčiami, peržiūrimi mokymo objektai arba atidaromi mokymo paslaugų portletai. Jeigu scenarijui yra sukurtas mokymosi kelias, tuomet vartotojui nėra iš karto parodomi visi sukomponuoti objektai, parodomas tik pirmasis įkomponuotas objektas. Vartotojui pažymėjus, kad objektas yra peržiūrėtas parodomas sekantis objektas. Ir taip toliau tol, kol neperžiūrimi visi objektai.



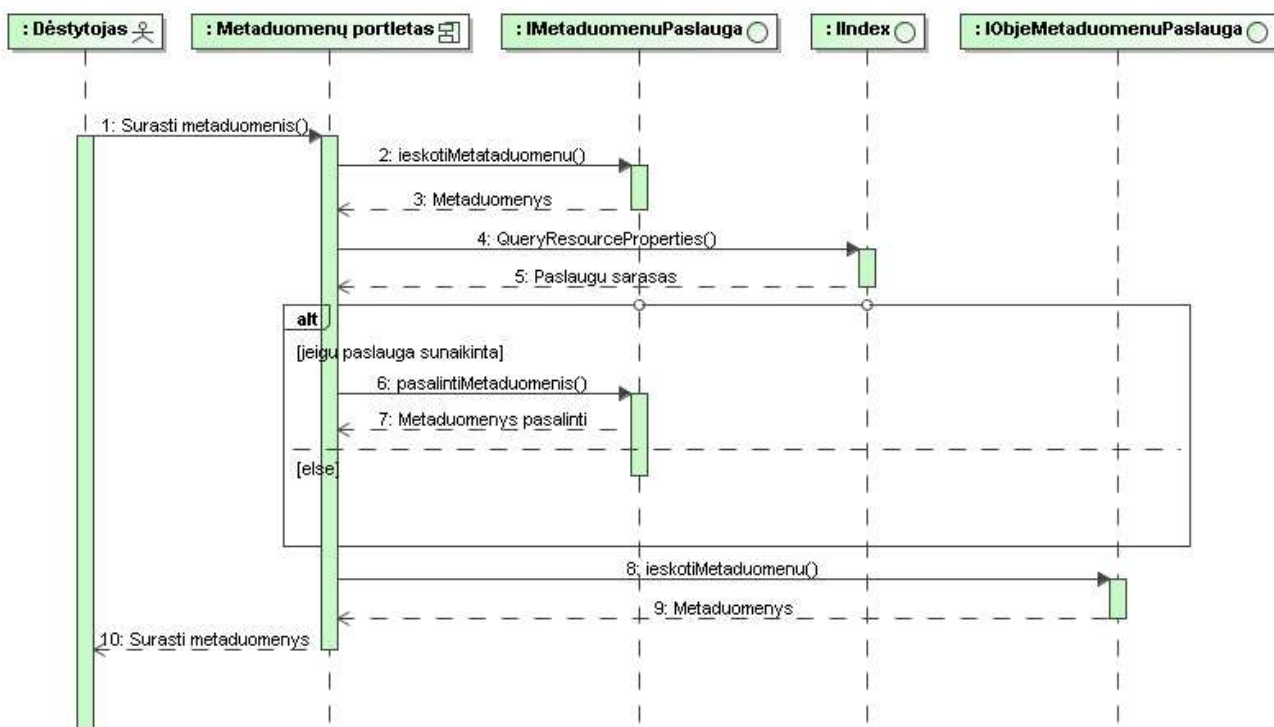
32 pav. Panaudojimo atvejo „Peržiūrėti scenarijų“ sekų diagrama

Sekančiame 33 paveikslėlyje panaudojimo atvejo „Ivesti metaduomenis“ sekų diagrama. Sekoje pirmiausia iš paslaugų registro nuskaitomos web paslaugos, o po to atliekamas aprašymas metaduomenimis, jeigu jie dar nebuvo aprašyti.



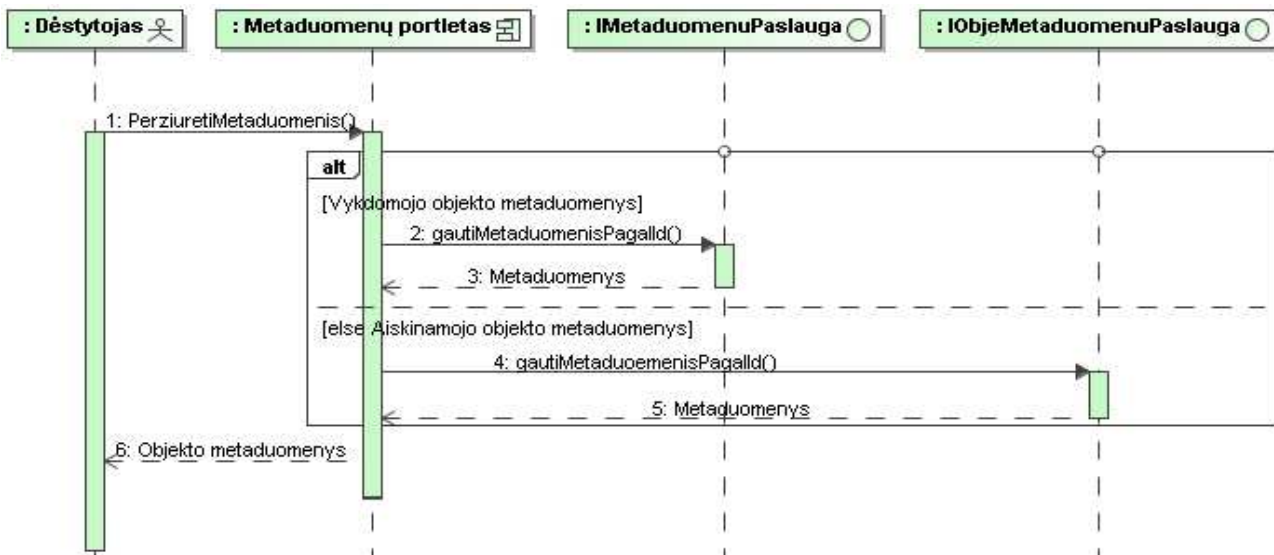
33 pav. Panaudojimo atvejo „Ivesti metaduomenis“ sekų diagrama

34 paveikslėlyje pavaizduota metaduomenų suradimo sekų diagrama. Ši seka ypatinga tuo, kad atlikus mokymo objektų paiešką metaduomenų saugykloje yra patikrinama ar surastos paslaugos yra gyvos. Jeigu paslaugos sunaikintos tuomet pašalinami ir jų metaduomenys. Ar paslauga gyva pasako Index servisas ir jeigu ji gyva tuomet ji būna užregistruota registre. Jeigu paslauga sunaikinama tuomet dingsta ir jos registracijos duomenys iš index serviso.



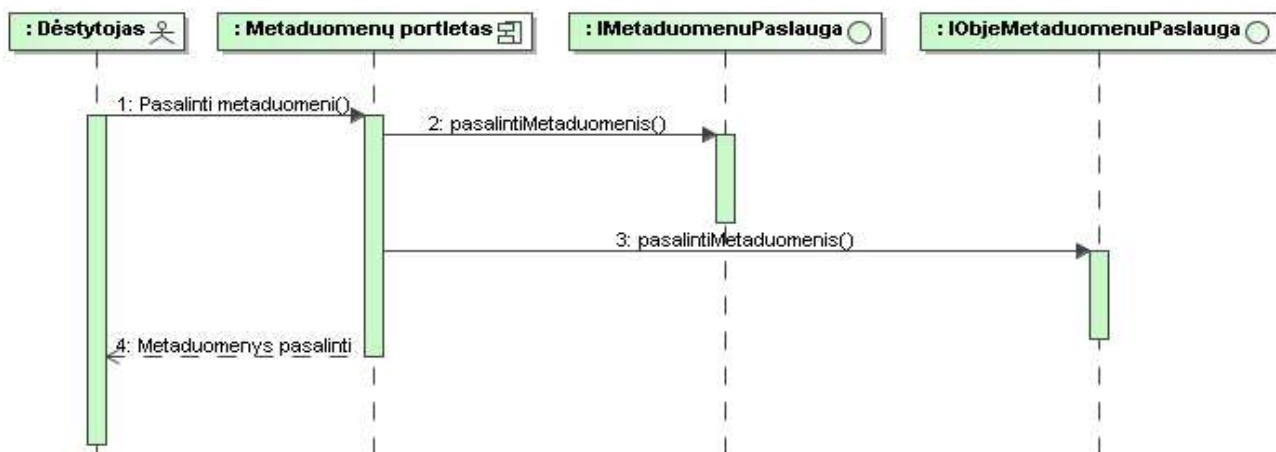
34 pav. Panaudojimo atvejo „Surasti metaduomenis“ sekų diagrama

Kitame 35 paveikslėlyje pavaizduota mokymo objekto peržiūrėjimo seka. Metaduomenys negali būti peržiūrimi kol jie nėra surasti.



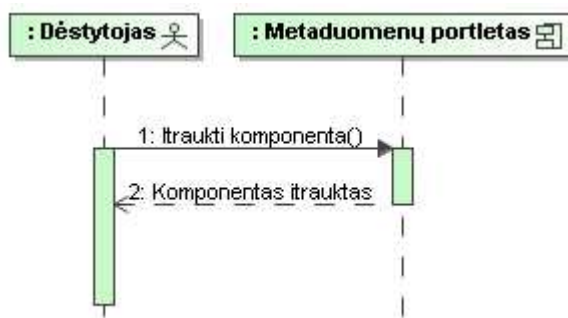
35 pav. Panaudojimo atvejo „Peržiūrėti metaduomenis“ sekų diagrama

36 paveikslėlyje pavaizduota metaduomenų šalinimo seka. Ji negali būti įvykdyta kol metaduomenys nesurasti.



36 pav. Panaudojimo atvejo „Pašalinti metaduomenis“ sekų diagrama

Paskutinėje sekų diagramoje parodyta, kaip yra atliekamas mokymo objekto įtraukimas į komponuojamųjų objektų sąrašą. Kai objektas įtraukiamas, jis pasidaro prieinamas mokymo scenarijaus komponavime (35 pav). Ir to pasekoje gali būti vykdomas panaudojimo atvejis „Pridėti naują objektą“.

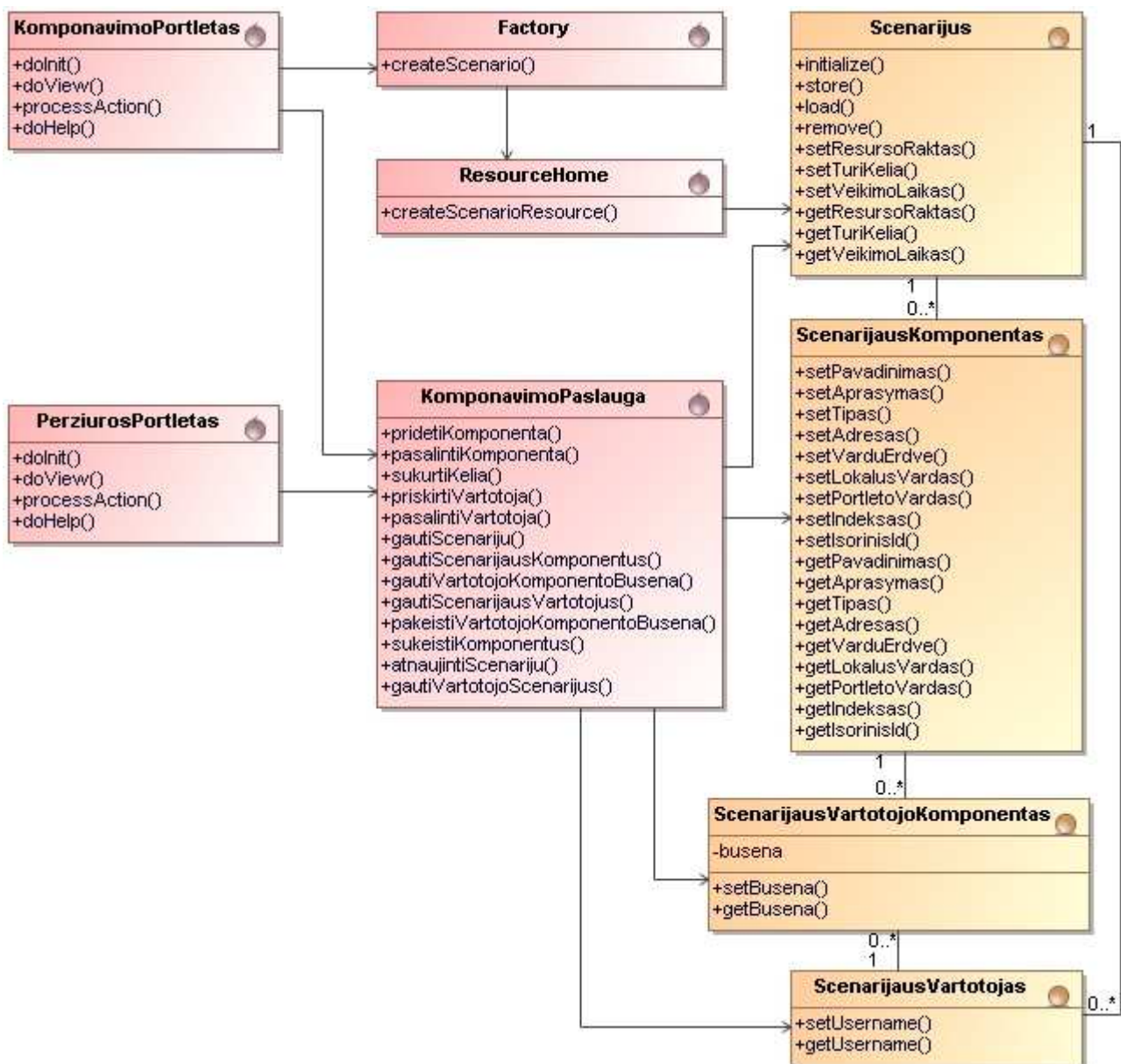


37 pav. Panaudojimo atvejo „Įtraukti objektą į komponuojamųjų objektų sąrašą“ sekų diagrama

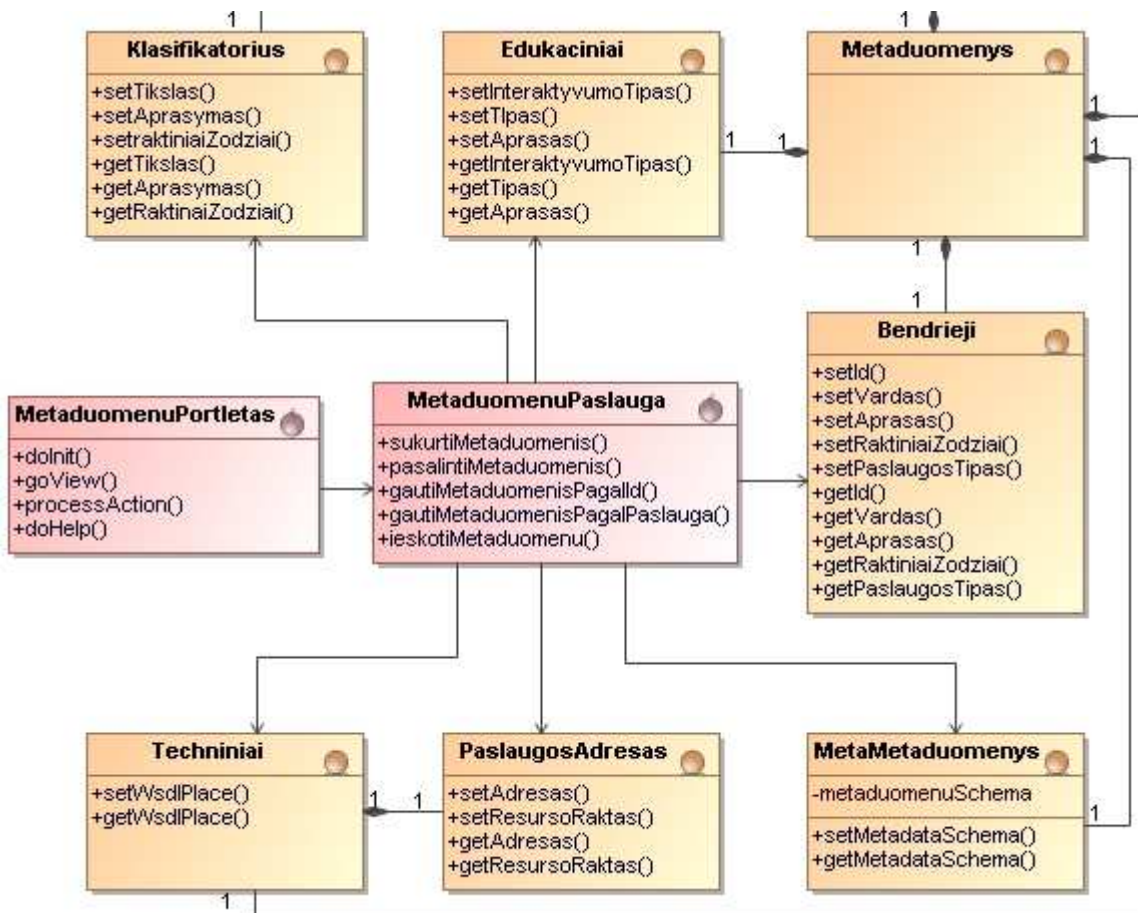
4.3.7 Paslaugų klasių diagramos

Paslaugas įgyvendinančios klasės pavaizduotos Komponavimo paslaugos (38 pav) ir Metaduomenų paslaugos (39 pav.) diagramose. Ši diagrama parodo klases esančias portletų komponentuose, bei klases įgyvendinančias paslaugų interfeisus.

Portletų klasių (komponavimo, priskyrimo, peržiūros) metodai kuriami pagal JSR-168 portletų specifikaciją. Klasių Factory ir Resource_home metodai taip pat kuriami pagal web paslaugų skirtą Globus Toolkit platformai rekomendacijas. Vaizdo klasės diagramoje nepavaizduotos, kadangi jos neturi realių metodų.

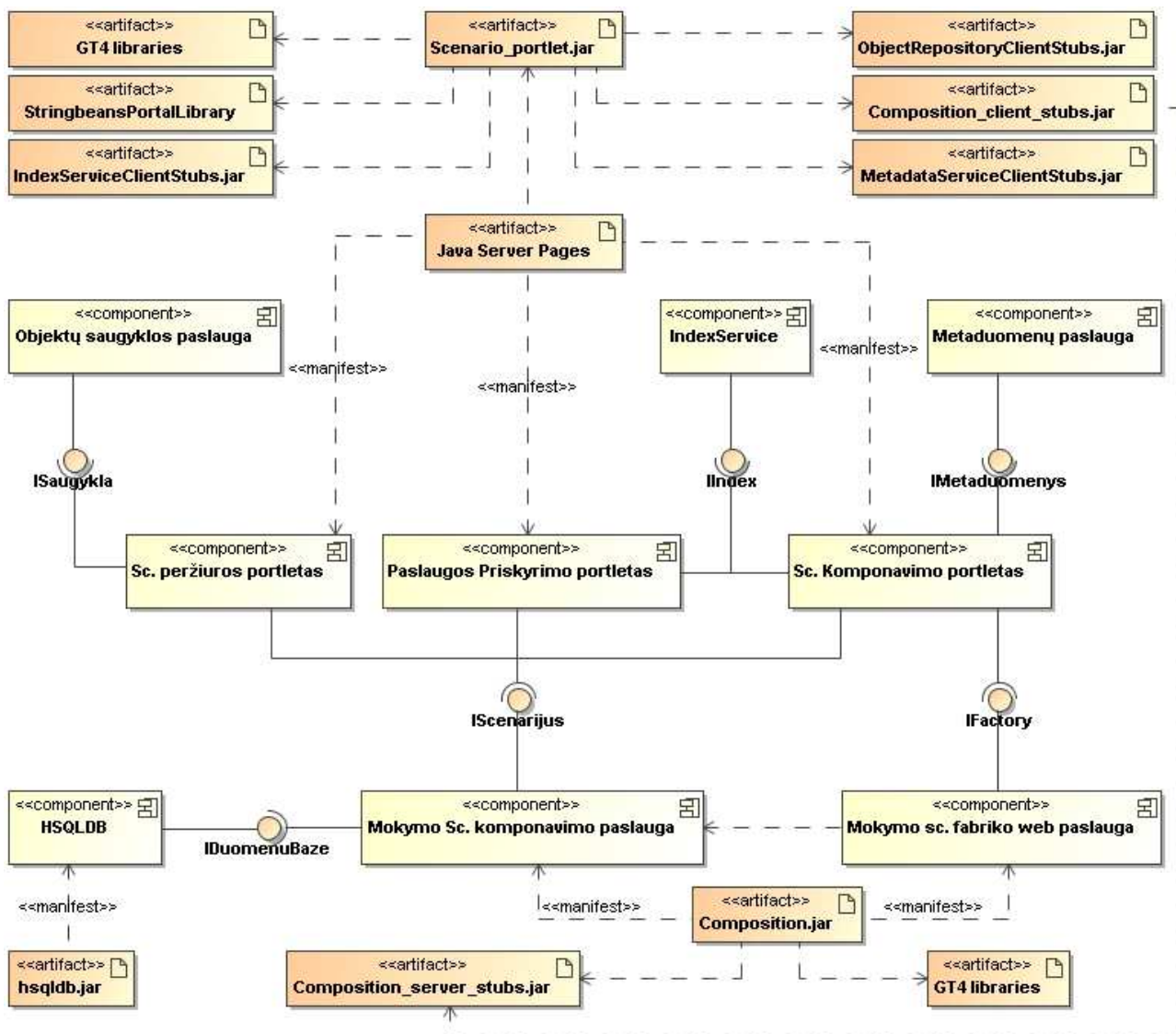


38 pav. Komonavimo paslaugos klasių diagrama



39 pav. Metaduomenų paslaugos klasių diagrama

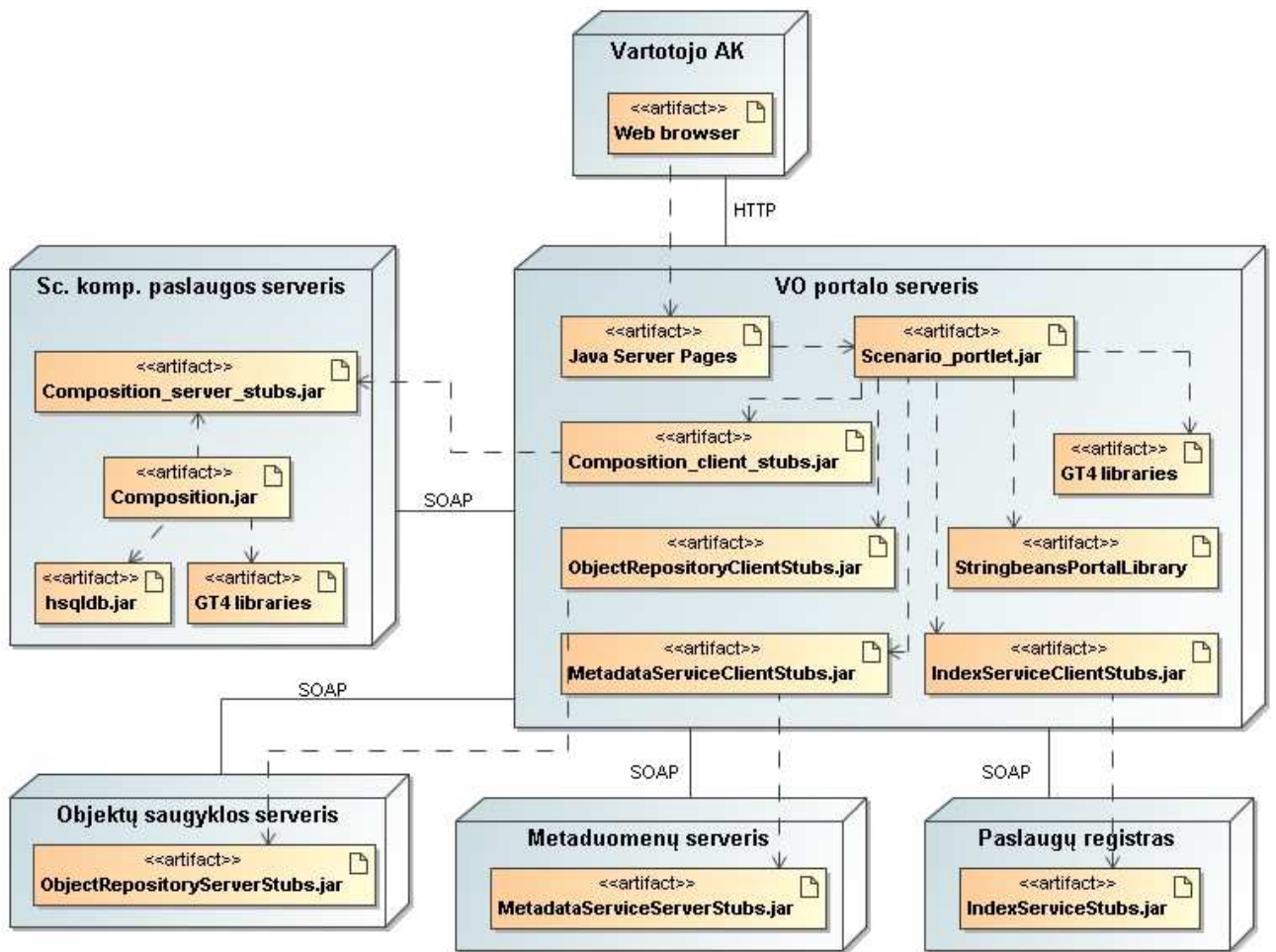
Sekanti komponentų diagrama vaizduoja kokie fiziniai realizacijos failai įgyvendina kiekvieną komponentą. Pagal tai diegimo diagramoje galima nustatyti kokie komponentai kokiam mazgui priklauso.



40 pav. Komponentų diagrama su artefaktais

4.4 Diegimo modelis

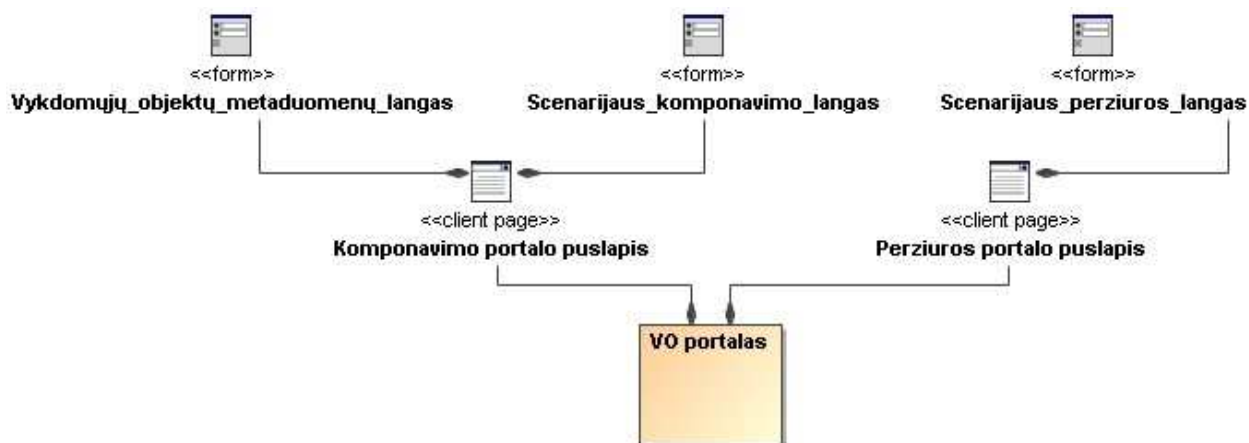
Kuriamoji sistema įgyvendinama web paslaugomis. Todėl paslaugų diegimo – paskirstymo diagrama yra viena iš aktualiausių. Idealiu atveju sistemai gali būti naudojami 5 serveriai. 4 serveriai naudojami web paslaugoms ir vienas serveris portalui. Portale esantys portletai su paslaugomis komunikuoja SOAP protokolu.



41 pav. Diegimo diagrama

4.5 Paslaugų vartotojų sąsajos

Komponavimo posistemės paslaugos VO dalyviams prieinamos per žiniatinklį, visos vartotojų sąsajos integruotos į VO portalą, kuris įgyvendinamas bazinės Stringbeans portalo programinės įrangos pagrindu. Komponavimo posistemėi reikalingi du web puslapiai – komponavimo puslapis ir scenarijaus peržiūros puslapis (42 pav.). Komponavimo puslapyje pateikiamas scenarijaus komponavimo langas ir metaduomenų paieškos langas, kurie įgyvendinami kaip portletai. Analogiškai peržiūros puslapyje pateikiamas scenarijaus peržiūros langas. Aiškinamųjų bei vykdomųjų objektų įkėlimo į VO aplinką paslaugos portletas šioje diagramoje nepavaizduotas.



42 pav. Vartotojo sąsajos modelis

Toliau pateikiami sąsajų pavyzdžiai su trumpais paaiškinimais.

Naujos sudėtinės paslaugos–scenarijaus kūrimo sąsaja pavaizduota 43 paveikslėlyje. Autorius įveda scenarijaus pavadinimą ir sukuriamas tuščias sudėtinės paslaugos ‚skeletas‘. Kitu atveju, galima redaguoti jau sukurtas sudėtines paslaugas arba naikinti pasirinktas.

Mokymosi paslaugos pavadinimas:	<input type="text"/>	<input type="button" value="Sukurti"/>
<hr/>		
Anksčiau sukurtos paslaugos:	<input type="text" value="-pasirinkite-"/>	<input type="button" value="Redaguoti"/>
	<input type="button" value="Šalinti"/>	

43 pav. Sudėtinės paslaugos komponavimo portletas (pradinė forma)

44 paveikslėlyje pavaizduotas sekantis komponavimo paslaugos portletas, t.y. aiškinamųjų ir vykdomųjų objektų pridėjimo portletas. Autorius gali pasirinkti objektą arba paslaugą ir jai atitinkantį portletą ir įtraukti į scenarijų. Portleto apačioje autoriui pateikiamas jau įtrauktų komponentų sąrašas. Ši sąsaja naudojama tiek sudėtinės paslaugos kūrimui tiek ir egzistuojančių paslaugų redagavimui.

Scenarijaus pavadinimas: **Bandomasis Scenarijus**

MOKYMO OBJEKTAI

Pasirinkite mokymo objektą:

-pasirinkite-

PASLAUGOS

Pasirinkite paslaugą:

-pasirinkite-

KOMPONENTŲ SĄRAŠAS

Pavadinimas	Tipas
GT4WS(uzduotis).doc	Objektas
GT4WS(kliento_programavimas).doc	Objektas
Sudeties paslauga	Paslauga

44 pav. Objektų pridėjimo forma

Dalyvių priskyrimas komponuojamai sudėtinei paslaugai atliekamas per sąsają-portletą pavaizduotą 45 paveikslėlyje. Priskiriant vartotojus leidžiama pasirinkti ar bus priskiriamas individualus vartotojas ar visa vartotojų grupė. Priskirtų vartotojų sąrašas čia pat matomas atskirame portleto lange.

Scenarijaus komponavimas

Pridėti objektus **Priskirti vartotojus** Sukurti kelias

Mokymosi paslaugos pavadinimas: **Bandomasis scenarijus**

Grupės:

- Adminai
- Useriai**

Vartotojai:

- All
- Darius Vardenis1 Pavardenis1
- admin Vardenis2 Pavardenis2**

Priskirti

Priskirti vartotojai:

Grupė:	GrupeA			x
Vardas	Pavardė	admin		x

45 pav. Scenarijaus komponavimo portletas (vartotojų priskyrimo forma)

Norint sukurti sudėtinę paslaugą su fiksuota paslaugų peržiūros seka, reikia sukurti scenarijaus peržiūros kelią. 46 paveikslėlyje pavaizduota kelio kūrimo forma, kurioje reikia nuspausti nuorodą „įjungti“ jeigu kelias reikalingas. Tokiu atveju visiems įkomponuotiems objektams suteikiamos pradinės reikšmės ir visi objektai atvaizduojami lentelėje.

Scenarijaus komponavimas

Pridėti objektus Priskirti vartotojus **Sukurti kelias**

Mokymosi paslaugos pavadinimas: **Bandomasis scenarijus**

Jeigu norite sukurti scenarijaus peržiūros kelią spauskite čia: [įjungti](#)

46 pav. Paslaugų peržiūros kelio kūrimo pradinė forma.

47 paveikslėlyje pavaizduotas portletas sudėtinės paslaugos objektų iškvietimo eiliškumui nustatyti. Objektų vykdymo eiliškumą galima pakeisti spaudžiant nuorodas „^“ ir „v“. Paspaudus nuorodą „^“ objektas kelyje bus perkeltas viena pozicija aukščiau. Analogiškai paspaudus nuorodą „v“ objektas bus perkeltas viena pozicija žemyn.

Scenarijaus komponavimas

Pridėti objektus Priskirti vartotojus **Sukurti kelia**

Mokymosi paslaugos pavadinimas: **Bandomasis scenarijus**

Jeigu norite panaikinti scenarijaus peržiūros kelią spauskite čia: [išjungti](#)

[1] Sumavimo web paslauga	Paslauga sudeda du skaičius.Paslauga sudeda du skaičius.Paslauga sudeda du skaičius.Paslauga sudeda du skaičius.Paslauga sudeda du skaičius.Paslauga sudeda du skaičius.Paslauga sudeda du skaičius.	V Δ
[2] Daugybės paslauga	Paslauga gali sudauginti du skaičius. Paslauga gali sudauginti du skaičius. Paslauga gali sudauginti du skaičius. Paslauga gali sudauginti du skaičius. Paslauga gali sudauginti du skaičius. Paslauga gali sudauginti du skaičius. Paslauga gali sudauginti du skaičius.	V Δ

47 pav. Peržiūros kelio kūrimo forma, kai kelias sukurtas

48 paveikslėlyje pateikiamas sudėtinės paslaugos peržiūros portletas. Jame kairėje pavaizduotas paslaugą sudarančių objektų sąrašas (Scenarijaus peržiūros portletas), o kairėje atidarytas sąrašė esančios sudėties paslaugos portletas.

+ New Page **Sc. peržiūra** * Sc. komponavimas TestTool Saugykla BEM Sudėtis

Scenarijaus peržiūros portletas

-pasirinkite-

Mokymosi objektai:

- [GT4WS\(WSDL_sudarymas\).doc](#)
- [GT4WS\(dieqimo_parametrai\).doc](#)
- [GT4WS\(uzduotis\).doc](#)
- [GT4WS\(kompiliavimas\).doc](#)
- [GT4WS\(logikos_programavimas\).doc](#)
- [GT4WS\(dieqimas\).doc](#)
- [GT4WS\(kliento_programavimas\).doc](#)

Paslaugos:

Sudeties paslauga

[Add Content](#)

SUDETIES PORTLETAS

Demuo A:

Demuo B:

Atsakymas:

Resurso savybes reiksme:

[Add Content](#)

48 pav. Sudėtinės paslaugos peržiūros portletas

Sekančiuose trijuose paveiksluose nuosekliai vaizduojamos 3 metaduomenų portleto formos. 49 paveikslėlyje pavaizduota metaduomenų atributų įvedimo forma vykdomiesiems objektams.

Web Servisu metaduomenys

[Ieškoti](#)

Pasirinkite paslauga

[1] http://127.0.0.1:8081/wsrf/services/SudetiesWebPaslauga

General

Pavadinimas: Sumavimo web paslauga

Raktazodziai: sudetis, paslauga

Paslaugos tipas: Mokymosi

Aprasymas:

Paslauga sudeda du skaicius.Paslauga sudeda du skaicius.Paslauga sudeda du skaicius.Paslauga sudeda du skaicius.Paslauga sudeda du skaicius.Paslauga sudeda du skaicius.Paslauga sudeda du skaicius.Paslauga sudeda du skaicius.

Educational

Mokymosi resurso tipas: Experimentavimas

Technical

Paslaugos apraso vieta: http://localhost:8081/wsrf/services/SudetiesWebPaslauga?wsdl

Issaugoti

49 pav. Metaduomenų įvedimo forma vykdomiesiems objektams

Joje galima pasirinkti paslaugą ir jai sukurti metaduomenis. Jeigu paslaugai metaduomenys jau sukurti tuomet pasirinkus paslaugą šie metaduomenys yra pavaizduojami atitinkamuose metaduomenų laukuose. Įvesti leidžiama tik tada kai metaduomenys dar nėra priskirti.

50 paveikslėlyje pavaizduota vykdomųjų objektų paieškos forma. Paiešką galima atlikti įvedus paieškos tekstą ir paspaudus mygtuką „Ieškoti“. Tokiu atveju bus surasti visi objektai atitinkantys paieškos frazę.

Web Servisu metaduomenys

[Ivesti metaduomenis](#)

Ieškoti

[1] [Sumavimo web paslauga](#)

Paslauga sudeda du skaicius.Paslauga sudeda du skaicius.Paslauga sudeda du skaicius.Paslauga sudeda du skaicius.Paslauga sudeda du skaicius.Paslauga sudeda du skaicius.Paslauga sudeda du skaicius.

[2] [Daugybos paslauga](#)

Paslauga gali sudauginti du skaicius. Paslauga gali sudauginti du skaicius. Paslauga gali sudauginti du skaicius. Paslauga gali sudauginti du skaicius. Paslauga gali sudauginti du skaicius. Paslauga gali sudauginti du skaicius. Paslauga gali sudauginti du skaicius.

50 pav. Vykdomųjų objektų paieškos forma

Paieška atliekama pagal tokius metaduomenų laukus kaip pavadinimas, aprašymas, raktiniai žodžiai, bei visus edukacinius laukus. Pavyzdžiui įvedus tekstą „sudėti“, bus surasti visi objektai, kurių metaduomenyse yra tekstas „sudėti“. Atlikus paiešką parodomas tik objekto pavadinimas ir aprašymas. Norint sužinoti visus objekto metaduomenis reikia paspausti ant objekto pavadinimo. Tuomet bus atidaryta objekto metaduomenų peržiūros forma (51 pav).

Web Servisu metaduomenys

[Ivesti metaduomenis](#) | [Ieskoti](#)

General

Identifikatorius	97f043a0-8a4e-11dc-8985-84b3f6d32e25
Pavadinimas	Sumavimo web paslauga
Aprašymas	Paslauga sudeda du skaičius.Paslauga sudeda du skaičius.Paslauga sudeda du skaičius.Paslauga sudeda du skaičius.Paslauga sudeda du skaičius.Paslauga sudeda du skaičius.Paslauga sudeda du skaičius.Paslauga sudeda du skaičius.
Raktiniai zodziai	sudetis, paslauga
Paslaugos tipas	Mokymosi

Educational

Mokymosi resurso tipas	Experimentavimas
------------------------	------------------

Technical

Web paslaugos apraso vieta	http://localhost:8081/wsrf/services/SudetiesWebPaslauga?wsdl
Web paslaugos adresas	http://127.0.0.1:8081/wsrf/services/SudetiesWebPaslauga
Resurso raktas	Neturi resurso rakto

[Itraukti si servisa komponuojamuju web servisu sarasa?](#) [Itraukti](#)

51 pav. Objekto metaduomenų peržiūros forma

Šioje formoje pateikiami visi pasirinkto objekto metaduomenų laukai ir jų reikšmės. Taip pat šioje formoje galima objektą įtraukti į komponuojamųjų objektų sąrašą. Šis sąrašas matomas komponavimo portlete objektų pridėjimo formoje paslaugų pasirinkimo iškrentančiame meniu ir jis išlieka tik einamosios sesijos metu. Todėl metaduomenų paieškos ir komponavimo portletai turėtų būti tame pačiame puslapyje.

5. Paslaugos eksperimentinis panaudojimas ir vertinimas

Sukurtoji komponavimo paslauga skirta naudoti Mokslo ir Studijų virtualiojoje organizacijoje. Paslaugos teikiamos galimybės buvo išbandytos Mokslo ir Studijų VO kursuose. Kursantai turėjo galimybę susipažinti su visomis scenarijaus komponavimui reikalingomis paslaugomis – aiškinamųjų objektų saugyklos paslauga, aiškinamųjų objektų metaduomenų paslauga, komponavimo paslaugos fabriko paslauga, komponavimo paslauga, index paslauga ir vykdomųjų objektų metaduomenų paslauga.

Pirmiausia į virtualiąją organizaciją buvo įkeliami aiškinamieji mokymo objektai naudojantis aiškinamųjų objektų saugyklos paslauga ir atitinkamu portletu. Po įkėlimo vartotojai aprašė jų metaduomenis naudojantis aiškinamųjų objektų metaduomenų paslauga ir portletu. Aiškinamųjų objektų įkėlimas šiame darbe neaprašytas, kadangi šiam tikslui pasiekti reikalingos paslaugos yra nepriklausomos nuo komponavimo paslaugos ir įgyvendintos atskirai.

Sekantis žingsnis tai virtualioje organizacijoje įdiegtų paslaugų galinčių pasitarnauti mokymui aprašymas metaduomenimis. Vartotojai turėjo įvesti dviejų skaičių sumavimo web paslaugos metaduomenis, naudojantis vykdomųjų objektų metaduomenų paslauga ir portletus. Portletas su įvedamais metaduomenimis pavaizduotas 49 paveikslėlyje. Mokymo objektus aprašius metaduomenimis iš jų galima sukompnuoti norimą mokymo paslaugą. Šiam tikslui buvo naudojamosi komponavimo paslaugos fabriko, komponavimo ir index web paslaugomis, bei komponavimo paslaugos portletu (43-47 pav.). Nors mokymo objektai ir buvo aprašyti metaduomenimis, tačiau jų negalima įkomponuoti, kol jie nebus surasti naudojantis metaduomenų paieška ir neįtraukti į komponuojamųjų objektų sąrašą (50-51 pav.). Ir tik įtraukus objektus į komponuojamųjų objektų sąrašą jie atsiranda objektų komponavimo formoje. Tai naudojamos portletų specifikacijos JSR-168 trūkumas, kuris neleidžia iš vieno portleto kreiptis į kitą ir taip įkomponuoti objektą tiesiai iš metaduomenų paieškos portleto. Nepaisant šio nepatogumo vartotojams pavyko sėkmingai įkomponuoti pageidaujamus mokymo objektus, priskirti vartotojus ir sukurti scenarijaus peržiūros kelią. Sekančiame žingsnyje buvo aprašyti mokymo paslaugos metaduomenys. Po metaduomenų aprašymo paslauga pasidaro prieinama komponavimo metu priskirtiems vartotojams. Peržiūrai buvo naudojamas mokymo paslaugų peržiūros portletas (48 pav.). Jo pagalba vartotojai peržiūrėjo įkomponuotus aiškinamuosius ir vykdomuosius mokymo objektus.

Peržiūrėję mokymo objektus vartotojai pakeitė kelių mokymo objektų turinį ir dar kartą peržiūrėjo sukompnuot scenarijų. Kaip ir reikėjo tikėtis vartotojams buvo pateiktos pakeistos mokymo objektų versijos.

Buvo naudojamas supaprastintas realaus VO panaudojimo atvejis. T.y viską turėjo atlikti vienas vartotojas. Realioje sistemoje aprašytame procese dalyvauja ne mažiau kaip 3 vartotojai – dėstytojas, autorius ir tyrėjas. Tačiau ir supaprastintame variante atsiskleidė pagrindinis tikslas – panaudojant 2 skyriuje aprašytą komponavimo sprendimą sukurtos komponavimo paslaugos pagalba galima scenarijų sudarančius komponentus keisti, o sukomponuota paslauga naudoja naujausius mokymo objektus. Scenarijų sudarantys komponentai yra atskiruose serveriuose ir vykdomi tame serveryje kur ir yra įdiegti (jeigu tai vykdomieji objektai), o aiškinamieji objektai parsiuoščiami į vartotojo kompiuterį pačiu paskutiniu momentu, kai jis nori šį objektą peržiūrėti.

Sistemos privalumai:

- Užtikrinamas lankstumas ir dinamiškumas. Pasikeitus mokymo objektų versijai, neboreikia perkomponuoti mokymo scenarijaus, kadangi sukomponuota paslauga nesaugo mokymo objektų kopijų, o saugo tik paslaugų adresus. Paslaugos perkomponuoti nereikia tol, kol nepasikeičia paslaugų adresai. Tai leidžia pasiekti viena pagrindinių web paslaugų savybių – silpnas susietumas.
- Paslaugos komponavimas gana paprastas, o automatinė paslaugos resurso sunaikinimo galimybė leidžia nebesirūpinti apie neboreikalingų mokymo scenarijų sunaikinimą. Jeigu Dėstytojas pastebi, kad scenarijaus mokymo objektai netinkami, objektų peržiūrėjimo eiliškumas netinkamas, ar priskirti ne tie vartotojai, jis bet kada gali scenarijų perkomponuoti.
- Tyrėjai patys sprendžia ką nori mokytis. Jie gali pasirinkti pageidaujamą scenarijų iš sukomponuotų paslaugų sąrašo. Taip pat galima panaudoti ontologijas tyrėjo gebėjimų medžio sudarymui. Tai leistų komponavimo posistemei atrinkti tik tas sukomponuotas paslaugas, kurios tinka pagal vartotojo gebėjimų medį.
- Komponavimo posistemei priklausantys portletai, naudojant WSRP gali būti panaudoti ir kitoje virtualioje organizacijoje. Tai leidžia kitai virtualiajai organizacijai prieiti prie laisvam naudojimui skirtų aiškinamųjų ir vykdomųjų mokymo objektų.

Rezultatai ir Išvados

1. E.mokymo sistemų architektūrinių sprendimų analizė parodė, kad nėra aiškaus sprendimo, kaip sukombonuoti mokymo objektus į lankstų scenarijų, kurio nereiktų perkombonuoti pasikeitus jį sudarančių mokymo objektų versijoms, todėl reikia sukurti savo komponavimo sprendimą.
2. Pagal siūlomų sprendimų apžvalgą komponavimo sprendimui įgyvendinti pasirinkta į paslaugas orientuota architektūra.
3. Mokymo objektu laikomas ne tik failų pavidale pateikiamas objektas, bet ir web paslaugos galinčias pasitarnauti mokymui. Tai naujas požiūris į mokymo objektus ir jų naudojimą.
4. Vykdomiesiems objektams aprašyti naudojami LOM standarto atributai pritaikyti vykdomiesiems objektams numetant nereikalingus atributus ir pridėdant naujus atributus.
5. Mokymosi ar studijų atveju neformalus sudėtinės paslaugos modelis galėtų būti interpretuojamas kaip nuoseklus procesas, prasidedantis susipažinimu su tema ir mokomosios medžiagos studijavimu, o po to pateikiami interaktyvesni vykdomieji mokymo objektai.
6. WSRF panaudojimas supaprastina mokymo paslaugų kūrimą ir valdymą.
7. Komponavimo posistemei priklausantys portletai, naudojant WSRP gali būti panaudoti ir kitoje virtualioje organizacijoje. Tai leidžia kitai virtualiajai organizacijai prieiti prie laisvam naudojimui skirtų aiškinamųjų ir vykdomųjų mokymo objektų.
8. Aprašytas komponavimo sprendimas pritaikytas realizuojant komponavimo posistemę Mokslo ir Studijų virtualiojoje organizacijoje.
9. Komponavimo posistemės panaudojimas kursuose parodė, kad tai yra tinkamas sprendimas kurti lanksčias e.mokymo sistemas.

Literatūra

- [1] Bloom B. S., *Taxonomy of Educational Objectives: The Classification of Educational Goals*, 1956, Susan Fauer Company, Inc; puslapiai. 201-207.
- [2] *IMS Learning Design Information Model*, prieiga internetu [žiūrėta 2007 01 23]
http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html
- [3] *EleGI projekto veiklos ataskaita*, prieiga internetu [žiūrėta 2007 07 12]
<http://www.elegi.org/elegi-publishable-final-activity-report-10.pdf>
- [4] Yang X., Allan R., *Realise e-Research through Virtual Research Environments*, 5th WSEAS Conference on e-Activities, 2006, prieiga internetu [žiūrėta 2007 02 12]
<http://esc.dl.ac.uk/Sakai/papers/conf.pdf>
- [5] Pereira L.A.M, Melo R.N., Porto F.A.M., Schulze B., *A workflow-based architecture for E-learning in the Grid*, IEEE International Symposium on Cluster Computing and the Grid (CCGrid'04), 2004, 208-209 puslapiai, prieiga internetu [žiūrėta 2007 11 24]
<http://arquivosweb.lncc.br/pdfs/relatorio-01-04.pdf>
- [6] Simpson D., Karagiannidis Ch., Cardinali F. (), 'An Architecture for Web-based e-Learning Promoting Re-usable Adaptive Educational e-Content', Educational Technology & Society, Vol. 5 No. 4, 2002, prieiga internetu [žiūrėta 2007 05 07]
http://ifets.ieee.org/periodical/vol_4_2002/sampson.pdf
- [7] Learning Technology Standards Committee, *IEEE Standard for Learning Object Metadata. IEEE Standard 1484.12.1*, Institute of Electrical and Electronics Engineers, New York, 2002, prieiga internetu [žiūrėta 2007 07 06]
http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf
- [8] *IMS Content Packaging Information Model*, prieiga internetu [žiūrėta 2007 05 05]
http://www.imsglobal.org/content/packaging/cpv1p1p4/imscp_infov1p1p4.html
- [9] *SCORM 2004 3rd Edition*, prieiga internetu [žiūrėta 2007 07 04]
<http://www.adlnet.gov/downloads/AuthNotReqd.aspx?FileName=SCORM.2004.3ED.ImpactsSummary.pdf&ID=161>
- [10] P. Polsani, *Use and Abuse of Reusable Learning Objects*, prieiga internetu [žiūrėta 2007 08 02]
<http://journals.tdl.org/jodi/article/view/jodi-105/88>
- [11] T. Andrews et al., *Business Process Execution Language*, 2003, prieiga internetu [žiūrėta 2007 10 02], <http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel.pdf>
- [12] *CopperCore Technical Documentation*, prieiga internetu [žiūrėta 2007 10 14],
http://coppercore.sourceforge.net/documentation/technical_documentation.shtml
- [13] Chappell D., Jewell T., *Java Web Services*, O'Reilly, First Edition, 2002, 276 puslapiai

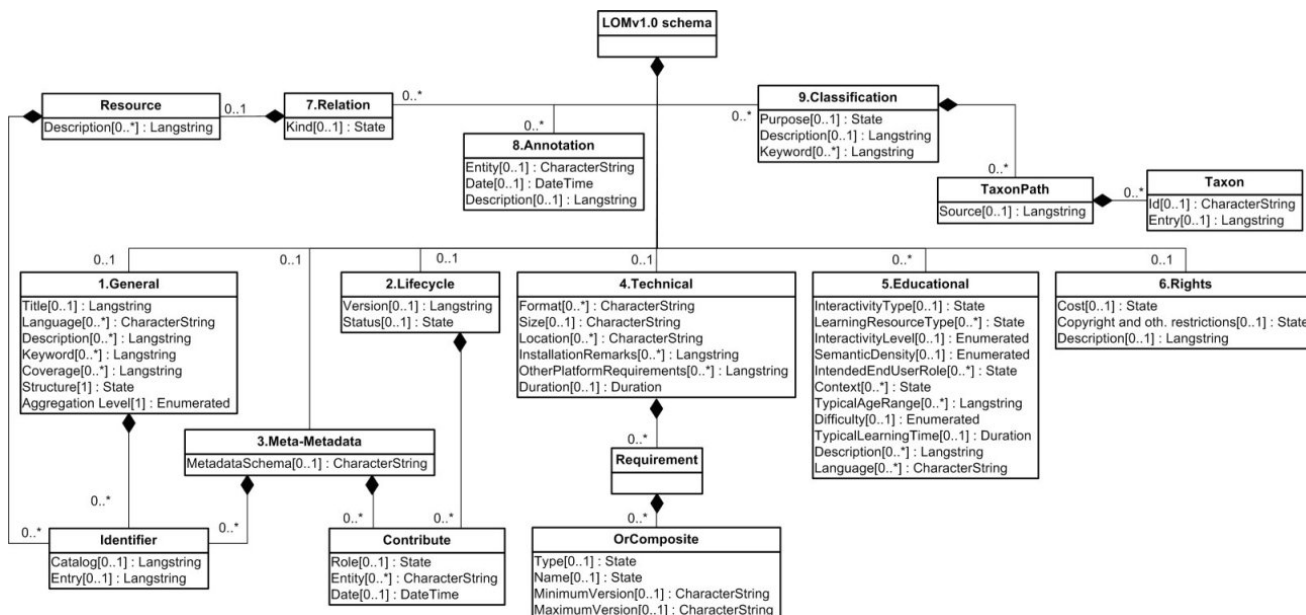
- [14] Hepper S. ir kt., *Portlets and Apache portals*, Manning, 2006, 437 puslapiai
- [15] Czajkowski K., Ferguson D., ir kt., *The WS-Resource Framework*, version 1.0, prieiga internetu [žiūrėta 2007 10 14] <http://www.globus.org/wsrp/specs/ws-wsrf.pdf>
- [16] A. Kropp ir kt., *Web Services for Remote Portlets Specification*, prieiga internetu [žiūrėta 2007 10 14] <http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf>
- [17] O. Chipara and A. Slominski., *Xydra—An automatic form generator for Web services*, prieiga internetu [žiūrėta 2007 08 23] <http://www.extreme.indiana.edu/xgws/xydra/>
- [18] G. Vossen, P. Westerkamp, *E-learning as a Web service*. 7th International Conference on Database Engineering and Applications (IDEAS), Hong Kong, China, IEEE Computer Society, 2003, puslapiai 242-249;
- [19] D. Dagger, A. O'Connor, S. Lawless, E. Walsh, V. Wade, *Service-Oriented E-Learning Platforms: From Monolithic Systems to Flexible Services*, IEEE Internet Computing, vol. 11, no. 3, 2007, puslapiai. 28-35
- [20] A. Arsanjani, *Service-oriented modeling and architecture*, prieiga internetu [žiūrėta 2007 07 25] <http://www.ibm.com/developerworks/library/ws-soa-design1/>
- [21] D. Booth, *Web Services Architecture*, W3C Working Group Note, 2004, prieiga internetu [žiūrėta 2007 07 25] <http://www.w3.org/TR/ws-arch/>

Terminų ir santrumpų žodynas

JAVA	Bendros paskirties, aukšto lygio, objektiškai orientuota, nuo platformos nepriklausoma programavimo kalba sukurta Sun Microsystems firmoje.
GRID	Paskirstyta sistema, kuri leidžia geografiškai nutolusių, autonomiškų resursų dinamišką dalinimąsi, pasirinkimą ir kaupimą.
HTML (<i>Hypertext Markup Language</i>)	Hiperteksto žymėjimo kalba. Tai kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete
HTTP (HyperText Transfer Protocol)	Pagrindinis protokolas pasiekti informaciją pasauliniame tinkle (WWW). Pradinė protokolo paskirtis - pateikti standartinį būdą HTML puslapių skelbimui ir skaitymui.
SOAP (Simple Object Access Protocol)	Protokolas XML pranešimų keitimuisi tinkle, dažniausiai naudojant HTTP.
VO	Virtualioji Organizacija
Web paslauga	Web paslaugos, tai programinės įrangos komponentai, sukurti tam, kad leistų kurti kintamo dydžio, laisvai susietas, nepriklausančias nuo platformos programas.
WSDL (Web Service DescriptionLanguage)	Web paslaugų aprašymo kalba
XML	Išplečiama žymių kalba (angl. eXtensible Markup Language)
WSRP (Web service for remote portlets)	Tai web paslaugomis paremtas protokolas leidžiantis portalui bendrauti su nutolusiais portletais.
IMS-LD (IMS Learning Design)	IMS konsorciumo specifikacija skirta mokymo procesų modeliavimui.
LOM (Learning Object metadata)	Mokymo objektų metaduomenų specifikacija
LMS	Mokymo valdymo sistema (angl. Learning Management System)
CMS	Kursų valdymo sistema (angl. Course Management System)
SCORM	Dalinamo turinio objektų ryšių modelis (angl. Sharable Content Object Reference Model)
POA	Į Paslaugas Orientuota Architektūra
LO	Mokymo Objektas (angl. Learning Object)
LOR	Mokymo objektų saugykla (angl. Learning Object Repository)
WS-BPEL	Web paslaugų verslo procesų vykdymo kalba (angl. Business Process Execution Language). Tai specifikacija nusakanti kaip jungti web paslaugas į procesus.

Priedas Nr. 1, IEEE 1484.12.1 LOM metaduomenų modelis

Šaltinis: http://upload.wikimedia.org/wikipedia/commons/1/14/LOM_base_schema.png, 2002m.



1 pav. IEEE LOM metaduomenų modelis

Priedas Nr. 2, Interactive composition of learning Web services

INTERACTIVE COMPOSITION OF LEARNING WEB SERVICES

Darius Zabiela, Gediminas Guginis

Kaunas University of Technology, Informatics Faculty, 50-416 Studentu Str., Kaunas, Lithuania

Abstract

The paper describes web service composition approach developed in the scope of the project “Creation, development and installation of science and studies virtual organization using Grid technologies” (BPD2004-ESF-2.5.0-03-05/0070). Service composition is a common issue in Service Oriented Architecture. Exists several composition specifications, but we present the design and usage of composition subsystem used specifically on the context of e-learning.

Introduction

The most important objective of e-learning systems is to provide learning content which should give required knowledge and competence for learners or scientists. Such content shouldn't be created from zero. It should be interactively created from smaller/atomic learning objects and delivered to VO users. Learning object composition to seamless and more completed structure by learning approach is named learning service or learning scenario. Learning objects may vary from simple text document to sophisticated web services which are the most important part of VO. Therefore composition of learning service is quite complex task. However bringing on such implementation technologies like web services and portlets makes this task solvable.

Different approaches of service composition

Web service composition is quite usual thing in service oriented architecture (SOA). There are several composition specifications like BPEL4WS [1], DAML-S [2], WSCI [3] and solutions based on these specifications. The most of these specifications describes web services operations connection to workflow. Each workflow has input and output without reference to passed web services of current execution. This approach is commonly used in SOA systems when web services are fine grained [4]. I.e. web services which operations are clearly defined and can't be simplified. Workflows can take results from one web service and transmit them to another web service input. There could be many such interactions in the workflow. This composition approach can be used to create sophisticated workflows and named learning service. Also for each workflow should be statically created or dynamically generated user interface. However problem arises when composable web services are not fine grained. I.e. web services take or return complex data types. In this case composition can be impossible if there are no web services which operate with the same complex data types. In the implementation point of view current composition approach is quite complicated, because most of work effort must be allocated to design web service architecture, define fine grained web services, and create or generate user interface by the workflow input and output data. There are some solutions like Xydra [5] which can generate user interface from web

service interface. Unfortunately Xydra can generate user interface only for very simple web services. However, for e-learning applications that are to be presented to the scientific users, it is often desirable to build a custom interface that is richer in its interactive features.

There is other composition solution where web services are not physically connected to workflows, but related by view point of learning. For example, in creation process of learning service about programming in java can be included such learning objects like the task, short theory about java programming, some video tutorial, assessment web service, experimentation web service and so on. All these components have no physical interconnection. They only related by view point of learning and can be used in different contexts – learning scenarios. In this case user interface problem is solved, because it hasn't to be created or generated to every composed scenario. Every web service has its own user interface which could be rendered at the moment of learning object preview. To implement this approach one possible solution is to use Service Data Elements which could be html file, java applet and java web start application [6]. However better solution to use portal and portlets as web service user interface. Use of portlets instead of several types of user interface simplifies overall implementation of the system. For previously mentioned example web service user interfaces could be text document preview portlet, video preview portlet, testing portlet, java source editing portlet and so on.

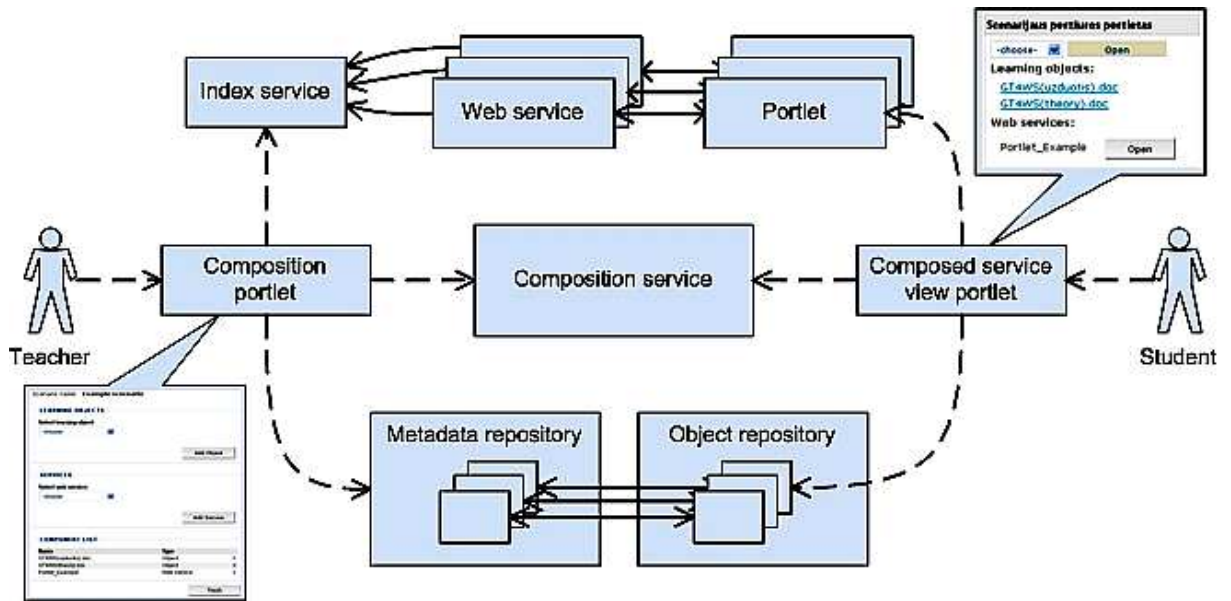
Learning web service composition

Composed learning service should give learning content which is called learning scenario for VO users on some particular topic. This scenario can consist of description, task and theory, modeling, experimentation, assessment and other tools or web services what can be used for learning. VO users should be able to use composed learning services what suits most for their needs. But at first, learning service should be composed. Next on this paper is described how these two tasks are achieved.

Composition scheme

Composition scheme is depicted in 1 picture. There are two distinct user roles – teacher and student. However they are not strictly specified. In place of teacher could be any VO user which is able to compose learning services and in place of student could be any VO user which is able to view composed learning services.

Composition starts when user able to compose learning service opens portal page where is “Composition portlet”. By the use of this portlet and “Composition service” he creates empty learning service. Next, he can compose this service by adding learning objects or web services. Teacher can select composable web service from the list of existing web services and add them to current learning service. The list of web services can be made from all web services of VO or only those web services which are suitable for the current learning scenario. In this case the list is made by searching for specific metadata value in “index service” which publishes learning web services metadata. Similarly, teacher can add learning object to the current learning service. With intention to do that he searches metadata repository for suitable learning object or selects from all available objects. To a previously created learning service not the real objects and web services are added, but their access information. Addition is made using “Composition service”.



1 picture. Learning service composition scheme

User who wants to use composed learning service opens portal page where is the “Composed service view portlet” and selects desirable learning service. “Composition service” lists learning objects and web services that belong to selected learning service. When user selects learning object or web service from the list the “Composed service view portlet” retrieves learning object from object repository and opens it or finds portlet for selected web service and opens it in the portal page.

Example

Considering the 1 picture are created several user interface forms. For the “Composition portlet” created two forms and for “Composed service view portlet” one form.

On a 2 picture is depicted possible user interface for creation of new learning service. In this form user can enter learning service title and description and click “create” button. By clicking this button is accessed “Composition service” that creates new empty learning service. Also this form allows viewing, editing and deleting already created learning services.

Learning service title:	<input type="text"/>
Description:	<input type="text"/>
	<input type="button" value="Create"/>
Existing services:	<input type="button" value="-choose-"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

2 picture. Learning service creation form

On the next picture depicted learning object and web service addition form (composition form). User can select object or web service and add it to empty scenario. Also in this form shows

already added components. As were mentioned before learning objects or web services are not included at this step itself. Only information about location of these components in the VO included. When user clicks button “Add Object” or “Add Service” then “Composition portlet” accesses “Composition service” who stores information about added object for current learning service. The same form is shown on learning service composition and editing.

Scenario name: **Example scenario**

LEARNING OBJECTS

Select learning object:

-choose-

SERVICES

Select web service:

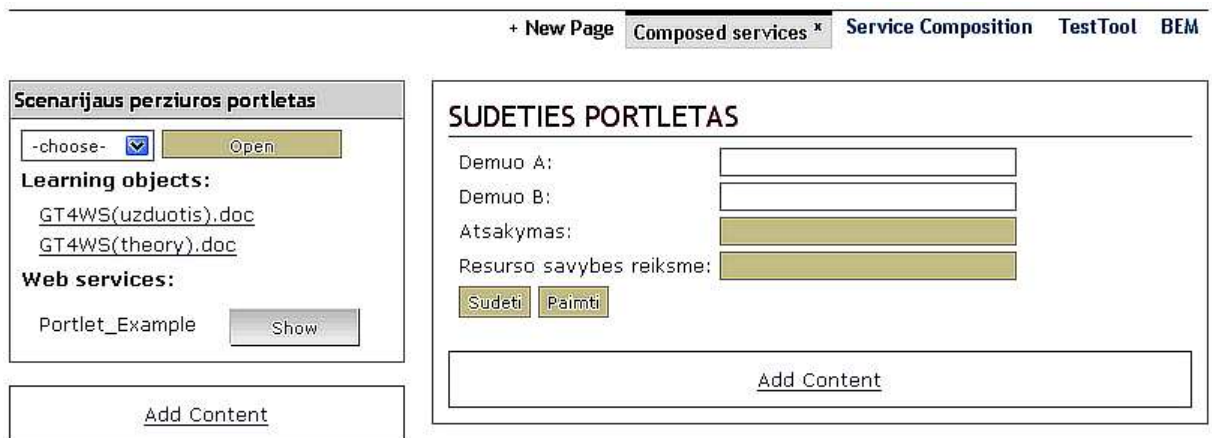
-choose-

COMPONENT LIST

Name	Type	
GT4WS(uzduotis).doc	Object	X
GT4WS(theory).doc	Object	X
Portlet_Example	Web Service	X

3 picture. Learning service composition form

On the 4 picture shown already composed learning service preview page. On the left is the “Composed service view portlet” form with opened learning service. When user selects desirable learning service and clicks “Open” then portlet retrieves content from “Composition service”. User can select either learning object or open web service. In case of selection of learning object “Composed service view portlet” retrieves selected object and opens it with the associated program. If user wants to use web service he clicks “Show” button, “Composed service view portlet” finds portlet associated with selected web service and opens it on the right side of the portal page. In the current picture shown “summation portlet” (“Sudeties Portletas”) used for summation web service.



4 picture. Learning service preview page

Requirements for virtual organization and web services

All previously described actions are possible only in fully functional VO when there are deployed and functional web services suitable for learning. Also when there is learning objects stored in object repository and described with metadata. All web services should have its metadata and they must be registered in Index service, because only in that case is possible to find web services suitable for one or another learning service.

Also there should be created at least one portlet for every web service used for learning, because users can see and work only with portlets and not with web services. There should be made a solution of portlet and web service assignment.

Conclusion

Composition approach described in this paper shows how web services of Virtual Organization can be composed to meet the requirements of learners, how web services can be fitted to e-learning. However this solution is not completed, because there is absence of such features like users group assignment to composed learning service, learning path setting and metadata description. Group assignment is needed in order that distinct VO users could see not all learning services, but only those that are allowed to see. Learning path allows setting in which order objects or web services of learning service could be shown. In that case next object in learning service would be enabled only when user sets what he had looked at the previous object. Metadata description let users to get not all available learning services, but only interesting for him by use of search for specific metadata value.

This composition approach when composed web services has no physical interconnection in comparison to composition solutions like BPEL4WS, etc. advantage is such what there is no need to strictly design system architecture. Web service interfaces hadn't to be fine grained.

This solution enables learning service composition more flexible. Flexible, because creation and editing of learning service is quite simple and there is no need to additional design of SOA, development or system configuration. Also there is possibility to add not only web services, but also learning objects.

Composed service is not tightly associated to all system thus this service could be temporary and used only on demand. Also implementation of this approach is simpler and suits most in Science and Studies Virtual Organization.

This composition approach is implemented by activity “Construction of virtual organization unified service system: set of services for new service implementers” as a part of EU project “Creation, development and installation of science and studies virtual organization using Grid technologies” (BPD2004-ESF-2.5.0-03-05/0070)

Literature

- [1] T. Andrews et al., <http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel.pdf>, Business Process Execution Language, 2003.
- [2] The DAML Services Coalition, <http://www.daml.org/services/daml-s/0.9/daml-s.pdf>, DAML-S: Semantic Markup for Web Services, 2003
- [3] A. Arkin et al., <http://www.w3.org/TR/wsci/>, Web Service Choreography Interface (WSCI) 1.0, 2002.
- [4] J. Hanson, http://articles.techrepublic.com.com/5100-22_11-5064520.html, Coarse-grained Interfaces Enable Service Composition in SOA, 2003
- [5] O. Chipara and A. Slominski., <http://www.extreme.indiana.edu/xgws/xydra/>, Xydra—An automatic form generator for Web services,
- [6] D. Gannon et al., Proceedings of the IEEE, Building Grid Portal Applications From a Web Service Component Architecture, 2005, VOL. 93, NO. 3

Priedas Nr. 3, The ELeGI Notification service: implementation and case study

The ELeGI Notification service: implementation and case study

Abstract. The paper describes a Web service-based notification system developed in the scope of the ELeGI¹ project. ELeGI is a Service Oriented Architecture for learning services provision and, due to its purpose, is different from Grid computing architectures. The notification service not only has to pass messages between services, but also to provide a convenient way to notify Virtual Learning Community (VLC) members. We present the design of the notification service and an experimental usage case in the context of ELeGI.

Keywords: notification service, WS notification, message oriented communication.

Introduction

Messaging is a fundamental method of communication between entities in distributed systems. A new impetus to evolve this method receives because the advances in Grid computing and Service Oriented Architectures (SOA), where message oriented communication is among the factors contributing to rapid growth and development of loosely coupled distributed systems. In Grid computing, event handling systems are being used to fix on changes or updates of resources status and those changes need to be disseminated among Grid parties. This happens in the form of event messages or notification messages (Pallickara, S., Fox, G., 2005).

ELeGI is a typical SOA grounded in the strategic advantages of Grid technology. According to its mission, ELeGI architecture differs from Grid computing architectures. Due to its orientation for learning and research services provision, ELeGI layered architecture is comprised of a great diversity of services, ELeGI deployment involves several categories of users with different roles. However, computational power and effective management, scheduling, monitoring of distributed resources are not the primary objectives for ELeGI. These differences should be taken into consideration in the design of the notification system. The event handling system, in parallel to passing of messages between services, should also provide a convenient way to notify end users.

The ELeGI notification system implements the application layer notification service. It is entirely dependent on Globus Toolkit (GT4) notification interfaces for SOAP messaging, which are realized in accordance with Web services BaseNotification specification (Graham, S. Murray, B., 2004). As such, a notification service is not a true Grid component and could be regarded as communication service. Actually it is a one way messaging service for outbound system generated messages and, obviously, it is not applicable as a real communication service. It is aimed at notifying ELeGI users about various occurrences in the system related to the changes in WS resources including notification of learners. But a notification service is not intended to be used as one way communication in learning.

In this paper we briefly discuss the publish/subscribe patterns being used when implementing notifications. Further we describe the notification service developed for the needs of the project, and present the architecture and implementation technology used in the ELeGI messenger. The realization and testing section describes an experimental case of the notification service in the

¹ – European Learning Grid Infrastructure (ELeGI),FR6 IST Integrated project No. 002205

context of ELeGI architecture. Lastly, possible extensions and modifications to meet the growing needs of ELeGI are discussed.

WS Notifications

The WS-Notification family of specifications define a notification model as a standard Web services approach to notification (Graham, S., Niblett, P., et.al. 2004). In the paper (Pallickara, S., Fox, G., 2005) an overview of the WS notification specifications is given and the far reaching implications these specifications may have on the development of asynchronous service-oriented architectures are discussed. Inherent in the WS-Notification specifications are the following categories (Myerson, J., 2005):

- NotificationProducer – a Web service that is ready to send notifications when a certain event occurs.
- NotificationConsumer – a Web service that receives notification messages.
- SubscriptionManager – a Web service that makes possible message exchange between the NotificationProducer and the NotificationConsumer.
- WS-Topics – topics to which a NotificationProducer can associate its messages.

According to the specifications there are two publish/subscribe patterns for the implementation of a notification service. The first corresponds to the notification model, where a service provider initiates messages based on a subscription or registration of interest from a service consumer. A NotificationProducer accepts incoming subscribe requests, each subscribe request identifies one or more topics of interest and a reference to a NotificationConsumer. Importantly, the NotificationProducer maintains a list of subscriptions and each subscription entry contains the topic(s) and a reference to the NotificationConsumer. The configuration whereby a NotificationConsumer subscribes directly to the NotificationProducer is referred to as the direct or point-to-point notification pattern.

The second is a brokered notification pattern which implies the deployment of message oriented middleware – an intermediary Web Service that decouples NotificationConsumers from Producers (Pallickara, S., Fox, G., 2005). In the brokered case the NotificationProducer plays simpler role of the publisher because it does not interact directly with NotificationConsumers. The publisher Web service preregisters with the NotificationBroker by sending a RegisterPublisher message, establishing with the broker the intention to publish messages on one or more topics. A NotificationBroker implements both the NotificationConsumer and NotificationProducer interfaces. Subsequent to registration a broker is able to get messages through the consumer's interface and disseminate them via the NotificationProducer interface to the subscribed NotificationConsumers. The NotificationBroker maintains a list of subscriptions and the subscription sequence is identical to the subscription in the non-brokered case.

The notification service designed for the ELeGI prototype actually implements the point-to-point notification pattern in a specific way for two reasons. First, the GT4 as an underlying grid platform is used at the KTU partner institution, but GT4 supports SOAP messaging interfaces only for the direct publish/subscribe pattern and does not provide API for brokered messaging. Second, due to the orientation of the application layer, the primary intention was to implement an infrastructure wide practical service aimed at notifying Virtual Learning Community (VLC) members rather than solely to disseminate messages among Web services. The function of the service therefore is to receive NotificationProducers' messages and to relay them to NotificationConsumers, in this case to VLC members. Moreover, consumers may have a preferred means to get notifications, for instance, e-mail, any instant messenger, notebook service or even SMS.

The Realization and Testing

In this paragraph we will discuss the design of the ELeGI notification service and explain a functionality of each component. Figure 1 shows a logical design of the notification service.

The three top layer rectangles to some extent are equivalent to application layer and represent communicating parties.

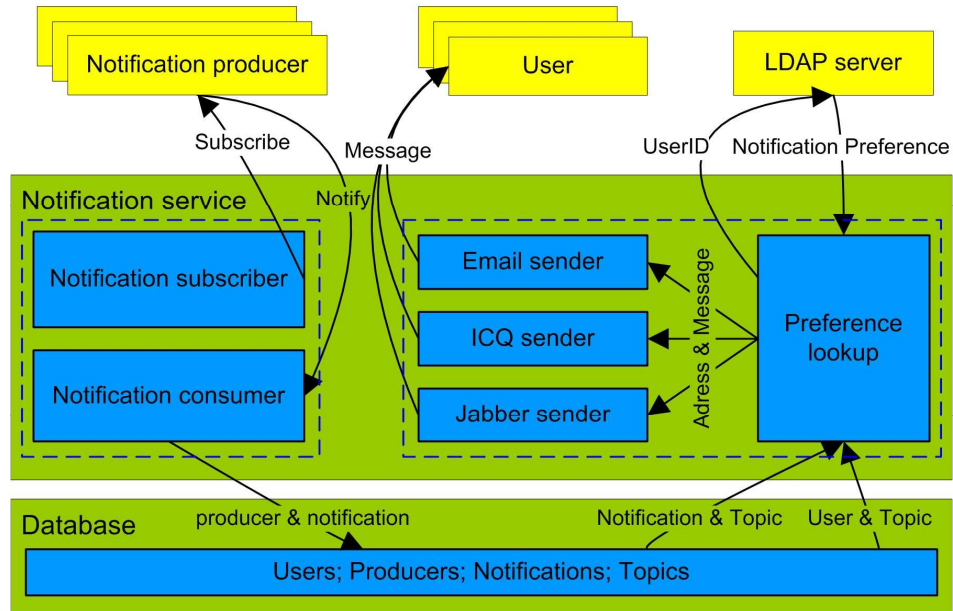


Figure 1. Notification service logical design

A Notification producer is intended to be any ELeGI architecture service. Typical examples could be the UserManagement and Membership services. The former, for instance, foresees the use case for a modification of the user profile. The modification of the WS resource invokes an event and the notification can be sent to the Virtual Community (VC) member. ELeGI communication and collaboration subsystem services are also designed to exploit notifications intensively. In the latter, the user is a VC member interested in receiving notifications about certain events.

The user can have the role of administrator, VC creator or any other ELeGI role, but rarely it is a learner. WS BaseNotification specification identifies NotificationConsumer as a Web service. In this case, however, in contrast to the entity of the Web service within the kernel of the notification service, at the application layer we have a real world entity-consumer – end user.

The LDAP Server maintains an ELeGI users' directory which is used for User, Role and Membership Management services. The directory contains a complete profile of every user, among its attributes is the specific attribute *elegiOrgPersonPreferredNotification*, which identifies the preferred means of notification, be that e-mail, instant messaging or other tools.

Notification service components

The middle layer actually is the kernel of the notification service. *Notification subscriber* implements the subscription of the producer to the *Notification consumer* and relates users to notification topics. *Notification consumer* receives WS notifications from producers in order to forward them to the final recipient. It should be anticipated, that *Notification consumer* is an

internal notification service component and not a real notification consumer in terms of WS BaseNotification specification.

Preference lookup retrieves user preferences from the LDAP directory. This component also distributes messages for users in accordance with their preferred message format. In this respect, the *Preference lookup* implements a function of the gateway, which changes the conventional WS SOAP messaging protocol into the preferred one:

- *Email sender* sends notification via SMTP protocol.
- *ICQ sender* sends notification via OSCAR protocol.
- *Jabber sender* sends notification via XMPP protocol.

Basically, the *Preference lookup* component along with a set of senders can be considered as a dispatcher of the notification messages.

Database

The Database is an intermedium component which serves two distinct purposes. Firstly, due to the dispatcher's gateway function it is convenient to have short term storage for a full record of notification data. The table *messages* (Figure 2) establishes the queue of incoming messages, which are directly sent off in sequence. In this respect the Database component facilitates the linkage of the *Notification consumer* with the dispatcher and reduces the failure probability in the process of dispatching. Secondly, the database component provides long term storage for the maintenance and recovery of the subscription data. For that purpose the tables *users* and *producers* are used.

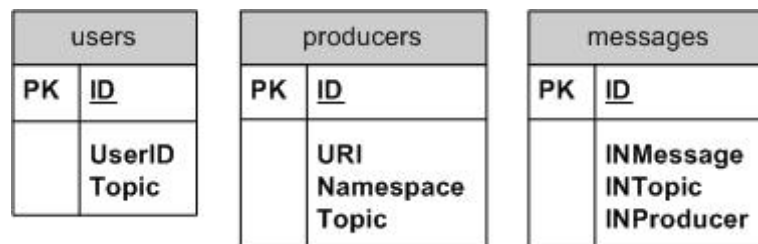


Figure 2. Database tables

Notification service interfaces

The components are realized and deployed as GT4 container at <https://elgrid.ktu.lt:8443/wsrp/services/notification/gateway>. Notification service interfaces are presented in the Table 1.

Operations	Description
addProducerDB	Implements subscription function. NotificationProducer endpoint reference and topic of interest are to be specified as parameters.
addUserDB	Establishes a topic and userID binding.
deleteUserDB	Destroys the topic-userID binding.
queryAll	Returns the list of active subscriptions in the form <topic, userID, endpoint reference to notification producer>.
subscribeAll	Enables the recovery of all active subscriptions using data stored in the database.
setParams	Sets parameters of the service like SMTP server, etc.

Table 1. Notification service operations

The first two operations are related to the setting up of a notification path between the notification producer and the notification consumer, in this case an ELeGI VC member. The interfaces being used to communicate with WS-producers are not included in the Table 1. Actually those interfaces are specified in the WS-BaseNotification specification and implemented within the Grid layer, in our case GT4. The next three operations are supporting functions, which can be used by system administrator for notification service maintenance. For instance, the list of subscriptions built by the *queryAll* method can be put into a convenient format and used when making a number of subscriptions with complicated endpoint references.

Testing

In a simple but revealing test experiment three ELeGI partners were involved. The membership service residing at University of St.Andrews was made NotificationProducer. Notification service components reside at the KTU node, whereas ELeGI LDAP server is hosted at Stuttgart. According to the WS BaseNotification specification, a notification consumer should ‘express the willingness’ to get notifications regarding certain events (topics) from a particular producer. That is, the subscription, which should be accomplished manually by the administrator using the *Notification subscriber* component. The administrator specifies the endpoint reference, topic and topic namespace of the producer. After subscription, the notification service starts listening for subscribed notifications.

This step of subscription sets up only the first half of the path, which is the SOAP messaging path between Web services. For the setting up of the second half of the path from the notification service to the end user a another subscription step has to be made. Actually this step is very much like the user registration to get messages, which belong to a certain topic of interest. Manually accomplished subscription and registration sets up a full route from notification producer to the end user, as it is shown on Figure 3.

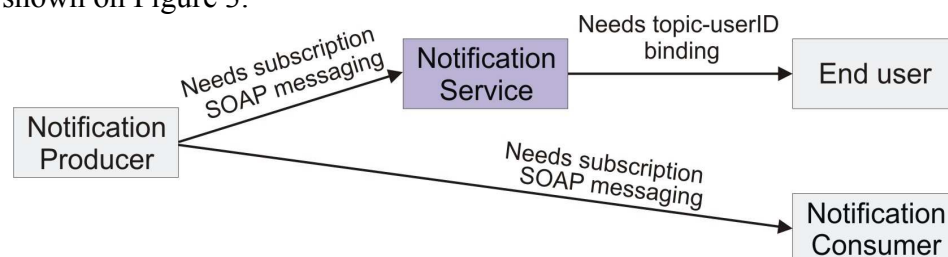


Figure 3. Notification service setting

The local Java client program was used for both subscription to notification producer and for user registration. In order to be able to establish the subscription, notification service and notification producer sites have to exchange certificates. We have been testing several certification schemes. However, authentication was successful and subscription became possible only when both sites have received each other’s CA certificates. Apparently, this circumstance may cause ‘the burden of subscriptions’ in architecture like ELeGI, where a number of distributed services can get a notification producer role. We are going to replace the current Java client application being used for subscription with a notification service portlet. Hopefully, this could facilitate the subscription process but it does not eliminate a burden problem.

Finally, the notification test was carried out. The *getAllGroups* method of the User Membership service was adopted for testing purposes in the following way. The purpose of the *getAllGroups* method originally was to query LDAP for all registered user groups and provide a list of groups to the user. For the experiment, the group ID was considered as a topic and the notification service

subscription was made only for one topic. Partners preferring to be notified via e-mail were registered to the same single topic. The *getAllGroups* method produce a notification for every group found in LDAP, but only one notification message appropriate to the subscription was sent off to the notification service consumer. For the received message preference lookup component retrieved the user's preference attribute and dispatched the message to all users subscribed for that topic.

The main result of the case study was not only the final fact of successful delivery of messages to the end user, but also, and no less important, was the experience gained in implementing and setting up the ELeGI notification service, including details of dealing with subscription interfaces and certificate authorities.

Discussion

The family of WS Notification specifications specify publish/subscribe notification patterns for SOAP messaging between Web services. The key idea behind the notification system for ELeGI is to notify Virtual Learning Community (VLC) members who are interested to be notified regarding the status of WS resources in selected cases. Thus, the ELeGI notification service receives SOAP messages from producers first, and then distributes them to VLC members in accordance with their preferred protocol. From this point of view, the notification service looks like messaging middleware and resembles WS Brokered Notification architecture. Actually in principle it is not because of the embedded WS BaseNotification subscription pattern.

The current version of the Notification service implements only the upper two parts of the route shown on the Figure 3. Obviously, if needed, any ELeGI service can be subscribed to any other notification producer bypassing the notification service. This could be facilitated considerably by having an implemented subscription portlet aimed at subscribing remote services.

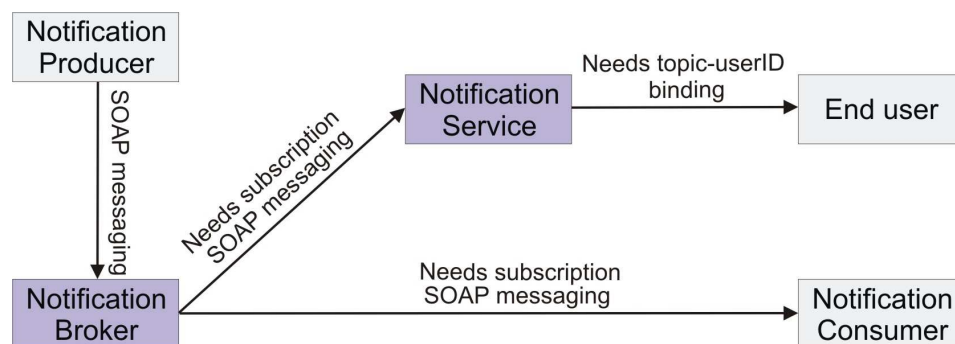


Figure 4. Notification Service and Notification Broker integration.

On the other hand, WS Brokered Notification architecture seems attractive due to the fact that the responsibility of subscriptions maintenance is transferred to the broker and the notification producer becomes more decoupled from the notification consumer. Moreover, a notification consumer has a single broker's endpoint reference to be used for subscriptions. Therefore, the prospective ELeGI notification subsystem could be comprised of the notification broker and of the notification service as it is shown in Fig. 4. The open source WS Messenger (Huang, Y., Slominski, A., Herath, C., Gannon, D., 2006) or the TrustCoM WSRF.Net implementation of the Notification Broker (Wilson, M. D., Arenas, A., Schubert, L., 2006) could be adopted. It should not be difficult to integrate a broker and the current version of the service because they should talk the SOAP protocol when communicating with each other. The adoption of a broker and making the ELeGI notification subsystem more flexible and convenient to use is the primary task for the near future.

Acknowledgments

The authors would like to thank ELeGI project partners Konrad Wulf from the University of Stuttgart and Stuart Purdie from the University of St.Andrews for the stimulating debate on WS-based notifications and their ongoing support.

References

Graham, S. Murray, B. (2004), '*Web Services Base Notification 1.2 (WS-BaseNotification) Working Draft*', Specification Document, Online, Available from: <http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-03.pdf>

Graham, S., Niblett, P., et.al. (2004), '*Publish-Subscribe Notification for Web Services Version 1.0 WS-Notification Whitepaper*', Specification Document , Online, Available from: <http://www.oasis-open.org/committees/download.php/6661/WSNpubsub-1-0.pdf>

Huang, Y., Slominski, A., Herath, C., Gannon, D. (2006) 'WS-Messenger: A Web Services-Based Messaging System for Service-Oriented Grid Computing', *Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)*, IEEE Computer Society, Conference Proceedings, pp. 166-173, Online, Available from: <http://www.extreme.indiana.edu/xgws/messenger/doc/HuangY-WSMessenger.pdf>

Myerson, J. (2005) 'Notify Web services and EAI's in heterogeneous SOAs' *Serie ,, Work with Web services in enterprise-wide SOAs*, part 8, Online, Available from: <http://www-128.ibm.com/developerworks/webservices/library/ws-soa-enter8/>

Pallickara, S., Fox, G. (2005), 'An Analysis of Notification Related Specifications for Web/Grid Applications', *International Conference on Information Technology: Coding and Computing - Vol. II*, IEEE Computer Society, Conference Proceedings, pp. 762-763, Online, Available from: <http://www.naradabroking.org/papers/WS-NotifyEventComparison.pdf>

Wilson, M. D., Arenas, A., Schubert, L. (2006), '*The TrustCoM Framework for trust, security and contract management of web services and the Grid*', Technical Report, Online, Available from: http://epubs.cclrc.ac.uk/bitstream/901/RAL_Tech_Rep_Trustcom_Framework.pdf