

# Internetinių paslaugų paieškos technologijų vertinimas jų tinkamumo internetinei prekybai požiūriu

## Albertas Čaplinskas

Vilniaus universiteto Matematikos ir informatikos instituto vyriausiasis mokslo darbuotojas, skyriaus vedėjas, profesorius, daktaras (HP)  
Vilnius University, Institute of Mathematics and Informatics, Principal Researcher, Head of Department, Professor, Doctor (HP),  
Akademijos g. 4, LT-08663 Vilnius  
El. paštas: albertas.caplinskas@mii.vu.lt

## Audronė Lupeikienė

Vilniaus universiteto Matematikos ir informatikos instituto vyresnioji mokslo darbuotoja, docentė, daktarė  
Vilnius University, Institute of Mathematics and Informatics, Senior Researcher, Assoc. Professor, Doctor  
Akademijos g. 4, LT-08663 Vilnius  
El. paštas: audrone.lupeikiene@mii.vu.lt

## Laima Paliulionienė

Vilniaus universiteto Matematikos ir informatikos instituto tyrėja  
Vilnius University, Institute of Mathematics and Informatics, Assistant researcher  
Akademijos g. 4, LT-08663 Vilnius  
El. paštas: laima.paliulioniene@mii.vu.lt

*Straipsnyje analizuojamos internetines paslaugas teikiančios sistemos, turinčios skirtingą architektūrą, akcentuojant prekių ir paslaugų reklamavimo ir radimo mechanizmus, įvertinami jų pranašumai ir trūkumai mažų ir vidutinių Lietuvos įmonių, norinčių pradėti teikti tokias paslaugas, požiūriu. Šiuo aspektu išnagrinėti paslaugų paieškos portalai ir pirkimų robotai, paslaugų paieška sistemose, veikiančiose atvirose išskirstytose aplinkose (SLP protokolas, CORBA, Jini technologija), paslaugų paieškos internete technologijos (UDDI, DNS, X.500, Ninja, P2P internetinės failų apsikeitimo technologijos), saityno paslaugų ir lokalizuotų paslaugų paieškos technologijos, pateiktas kiekvienos ekspertinis vertinimas balais pagal adekvatumo, universalumo, paprastumo, brandos, sąnaudų ir kitus kriterijus.*

## Įvadas

Kompiuterinio pardavimo ir paslaugų (paprastumo dėlei toliau vadinsime tiesiog paslaugomis, turėdami omenyje, jog tai gali būti ir pardavimo paslaugos) paieškos rinka yra perspektyvi ir sparčiai besiplečianti. Žmonės vis dažniau ieško prekių internetinėse parduotuvėse, ieško internete informacijos apie paslaugas ir jas užsako. Tačiau konkurencija čia yra labai didelė ir šioje rinkoje įsitvirtinti nelengva. Vienas iš svarbiausių paslaugų teikimo sistemų aspektų yra sistemos gebėjimas taip reklamuoti

savo paslaugas, kad potencialūs klientai galėtų paprastai ir greitai apie jas sužinoti. Pažymėtina, kad kompiuterinių paslaugų rinkoje šiuo metu matomos šios raidos tendencijos:

- paslaugų paieška naudojant mobiliuosius telefonus ir kitus nešiojamus mažo galimumo įrenginius, taip pat paieška mobiliuosiuose *ad hoc* tinkluose, kuriuose mobilūs esti ne tik klientai, bet ir paslaugų teikėjai;
- daiktų interneto teikiamų paslaugų paieška, ypač paslaugų paieška kompiuterizuotuose pastatuose (pažymėtina, kad

paslaugų ir daiktų internetui susiliejančią paslaugų paieškos technologijos tampa universalesnės);

- paslaugų ieškantys klientai, netgi tie, kurie ieško pirminių internetinėse parduotuvėse, pirmenybę teikia geografiškai arti esantiems paslaugų teikėjams;
- didėjantis saityno paslaugų išgalėjimas, Web 2.0 technologijų populiarėjimas, įskaitant saityno programas, sukomponuotas per viešąją dalykinių programų sąsają iš trečiųjų šalių teikiamų paslaugų, taip sukuriant naują paslaugą (angl.  *mashups*).

Šio straipsnio tikslas yra atlikti skirtingos architektūros kompiuterines (internetines) paslaugas teikiančių sistemų analizę, akcentuojant paslaugų reklamavimo ir radimo mechanizmus, įvertinti jų pranašumus ir trūkumus smulkių arba vidutinio dydžio įmonių, norinčių pradėti teikti tokias paslaugas, požiūriu\*. Straipsnyje pirmiausia apžvelgiamos dažniausiai šiam tikslui naudojamos technologijos – paslaugų paieškos portalai ir pirkimų robotai. Paskui nagrinėjamos sudėtingesnės technologijos – sistemos, veikiančios atvirose išskirstytose aplinkose (įskaitant SLP protokolą, CORBA, Jini technologiją), paslaugų paieškos internete technologijos (UDDI, DNS, X. 500, *Ninja*, P2P, internetinės failų apsikeitimo technologijos). Toliau aptariamos saityno paslaugų paieškos technologijos ir apžvelgiamos lokalizuotų paslaugų paieškos technologijos. Pabaigoje pateikiama visų šių technologijų ekspertinių įverčių lentelė.

## Paslaugų paieškos portalai ir pirkimų robotai

Paprasčiausi **integruotieji paslaugų paieškos portalai** vadovaujasi telefonų knygos metafora (pavyzdžiui, *MyGreenPages.com*, *VisaLietuva.lt*, *imones.lt*). Daugelyje šalių telefonų knygos turi trijų spalvų puslapius – baltus

(kuriuose spausdinami fizinių ir juridinių asmenų identifikaciniai duomenys, telefonų numeriai ir galbūt adresai), geltonus (juose spausdinama informacija apie minėtų juridinių ir fizinių asmenų teikiamas paslaugas, parduodamus produktus, klasifikuota pagal kokį nors standartinį klasifikatorių, pavyzdžiui, SIC, NAICS arba UNSPCS), žalius (juose nurodoma detali informacija apie tai, su kuo ir kaip susisiekti norint gauti kokią nors paslaugą ar įsigyti kokį nors produktą). Šią metaforą mėgdžioja paprasčiausi integruotieji paslaugų paieškos portalai, taip pat kai kurie internetiniai katalogai bei protokolai, pavyzdžiui, UDDI (protokolas, skirtas saityno paslaugoms ieškoti internete). Tiesa, daugelyje portalų geltonieji ir žalieji puslapiai yra sujungti. Sudėtingiausia problema šiuo atveju yra susitarti su paslaugų teikėjais, surinkti iš jų reikiamą informaciją ir ją reguliariai atnaujinti. Antra problema yra plačiai išreklamuoti portalą, jį išpopuliarinti. Be abejo, mažose šalyse, tokiose kaip Lietuva, šias problemas išspręsti yra paprasčiau. Pažymėtina, kad pats portalas taip pat yra paslaugų teikėjas. Viena vertus, jis teikia reklamos paslaugas kitiems paslaugų tiekėjams ir gaminių pardavėjams, kita vertus, teikia paslaugų paieškos paslaugas klientams. Ir viena, ir kita gali būti apmokestinta, nors paprastai klientai paslaugas gauna nemokamai.

Sudėtingesni yra portalai, kurie vieno langelio principu sukuria vadinamąjį fasadą, unifikuojantį skirtingų paslaugų teikėjų teikiamų paslaugų gavimo procedūras. Paprastai tokio tipo portalai yra skirti tik vieno tipo paslaugoms, pavyzdžiui, pardavinėti bilietus į skirtingų organizacijų organizuojamus renginius (pavyzdžiui, *bilietai.lt*) ar teikti skirtingų draudimo kompanijų teikiamas draudimo paslaugas. Naudojant vieno langelio principą, konkretūs paslaugų teikėjai tampa tiesiogiai nematomi klientui. Tokia architektūra reikalauja suvienodinti paslaugų gavimo, atsiskaitymo su klientu ir kitas procedūras, o tam savo ruožtu prireikia integruoti paslaugų teikėjų duomenis. Dažniausiai jų suvienodinti nepavyksta ir tenka kurti vadinamuosius preprocesorius ir postprocesorius, pertvarkančius paslaugų teikėjų duo-

\* Tyrimas atliktas pagal 2010 m. liepos 28 d. trišalę Mokslo, inovacijų ir technologijų agentūros, UAB „Fresh ID“ ir Matematikos ir informatikos instituto inovacinio čekio gavimo sutartį Nr. 31 V-80.

menis ir procedūras į portale naudojamą standartinį pavidalą ir atgal.

Sukurti vieno langelio principą realizuojanti portalą yra gerokai sudėtingiau negu portalą, grindžiamą telefono knygos metafora. Be jau minėtų duomenų ir procedūrų unifikavimo problemų, čia labai svarbi tampa duomenų aktualizavimo problema. Ji gali būti sprendžiama dviem būdais – naudojant kokią nors buferizavimo technologiją arba kokią nors *proxy* technologiją. Pirmuoju atveju portalas turi savo duomenų bazę, kurioje unifikuotu pavidalu kaupiami ir saugomi visi transakcijoms reikalingi paslaugų teikėjų duomenys. Transakcijos vyksta šioje bazėje, o paskui aktualizuojamos paslaugos teikėjo bazės. Tačiau tai trunka tam tikrą laiką ir, jeigu paslaugų teikėjas savo paslaugas teikia ne tik per šį portalą, bet ir kitais būdais, pavyzdžiui, tiesiogiai arba per kitus portalus, gali kilti kolizijų (pavyzdžiui, tą patį bilietą į renginį galima parduoti keliems pirkėjams), ir reikalingi sudėtingi transakcijų sinchronizavimo bei duomenų rezervavimo mechanizmai. Naudojant kokią nors *proxy* technologiją, portalas sukuria tiesioginį kliento ir paslaugos teikėjo sąveikos kanalą ir visos transakcijos vyksta paslaugų teikėjo bazėje. Kadangi lygiagrečiai gali vykti daug transakcijų, tai vienu metu portalui gali tekti kurti ir palaikyti daug kanalų. Be to, kanalai turi atlikti duomenų pertvarkymus (t. y. veikti kaip preprocesoriai ir postprocesoriai) ir unifikuoti sąveikos su skirtingais paslaugų teikėjais procedūras. Visam tam reikalingi laiko ir aparatiniai išteklių arba, kitaip tariant, susiduriama su efektyvumo problemomis.

**Pirkimų robotai** (angl. *shopping robots*, *shopbots*) yra skirti peržiūrėti didelį skaičių interneto parduotuvių, rasti tas, kuriose parduodamos kliento ieškomos prekės, ir surinkti, kokiomis kainomis kuri parduotuvė tas prekes parduoda (Montgomery ir kt., 2004). Tokio roboto arba, tiksliau, specializuotos paieškos mašinos pagrindu kuriami vadinamieji kainų palyginimo portalai (angl. *price comparison service*, *shopping comparison engine*). Tokie portalai (pavyzdžiui, nuo 1998 metų veikiantis *Streetprices.com*, lietuviškas portalas *Kainos.lt*)

patys nieko neparduoda, o pateikia klientui parduotuvių ar paslaugų teikėjų sąrašą, nurodydami prekių ar paslaugų kainas ir, klientui pasirinkus vieną iš to sąrašo eilučių, nukreipia jį į atitinkamą internetinę parduotuvę arba į atitinkamas paslaugas teikiančią portalą. Šiuo metu tokius portalus turi visi didieji paieškos rinkos žaidėjai (*Google*, *Yahoo* ir kiti) ir daugybė smulkių bendrovių, kurios apsiriboja kuria nors viena prekių ir paslaugų kategorija arba konkrečia geografine vietoje. Yra sukurta nemažai programinių priemonių kainų palyginimo portalams kurti. Kai kurie portalai tiek parduoda programines priemones savo kainų palyginimo portalui sukurti, tiek patys organizuoja informacijos iš pardavėjų teikimą ir atranką pagal kategorijas. Pavyzdžiui, *DataFeedFile.com* tam tikslui siūlo *JavaScripts*, XML, PHP/JSP/ASP API ir ROI duomenų sekimo įrankius (angl. *tracking tools*).

Viena iš palyginimo sistemų atmainų – mobilusis palyginimas (pavyzdžiui, *Amazon TextBuyIt*, *Google Product Search for Mobile*, *Google Shopper* paslaugos). Tai palyginimas naudojant mobiliuosius telefonus, kuris patogus, pavyzdžiui, norint greitai palyginti prekybos centre pamatytos prekės kainą su jos kainomis internetinėse parduotuvėse arba rasti artimiausias parduotuves, kuriose galima nusipirkti pasirinktą prekę.

Kuriant integruotus paslaugų paieškos portalus ir kainų palyginimo portalus reikia rinkti duomenis apie paslaugų teikėjus ir jų teikiamas paslaugas. Dažniausiai naudojamas duomenų rinkimas iš portalo nustatytos struktūros duomenų publikavimo failų (angl. *data feed file*). Tai gali būti CSV, XML (įskaitant RSS bei Atom) arba RDF failai. Kitas būdas – duomenų rinkimas iš laisvo formato tekstų, kuriuos tenka apdoroti rankiniu arba pusiau automatizuotu būdu naudojant informacijos išgavimo ir neraiškiosios logikos metodus. Dar vienas būdas – informacijos „iškirpimas“ tiesiai iš paslaugų teikėjų internetinių puslapių. Yra daug tam skirtų metodų (*text grepping and regular expression matching*; *HTTP programming*; *DOM parsing*; *HTML parsing*, *semantic annotation recognizing*) ir juos realizuojančių programinių produktų

(Mozenda\*, Freelancer\*\* ir kt.). Nėra iki galo aiškus šio metodo juridinis statusas, o ir kai kurie paslaugų teikėjai naudoja specialias apsaugos priemones, kurios neleidžia šitaip rinkti informacijos. Dar vienas duomenų rinkimo būdas – jų išgavimas iš minios (angl. *crowd-sourcing*). Šiuo atveju duomenis pateikia patys lankytojai, jie filtruojami bendromis lankytojų pastangomis, taikant dirbtinio intelekto metodus, arba tiesiog portalo darbuotojų rankomis.

Dažnai yra kuriami vadinamieji partnerystės tinklai (angl. *affiliate networks*). Jie jungia portalą, duomenis agreguojančius tarpininkus ir paslaugų teikėjus, publikuojančius savo teikiamų paslaugų aprašus (angl. *publisher*). Veikiama pagal partnerinės rinkotybės verslo modelį.

### **Paslaugų paieška sistemose, veikiančiose atvirose išskirstytose aplinkose**

Išskirstyta sistema – tai grupė savarankiškai veikiančių kompiuterių, kurie ne tik yra sujungti vienas su kitu, bet ir registruoti toje sistemoje, todėl gali tarpusavyje keistis pranešimais bei duomenimis. Galima sakyti, kad išskirstyta sistema virš realaus kompiuterių tinklo sukuria vienokį ar kitokį virtualų tinklą. Panašiai yra ir vieno langelio principu grindžiamų portalų kuriamuose tinkluose, tačiau tuose tinkluose pranešimų ir duomenų mainai vyksta tik tarp portalo ir paslaugų teikėjų. Portalas yra centrinis mechanizmas, koordinuojantis ir valdantis visos sistemos darbą. Išskirstytose sistemose tokio mechanizmo nėra. Jį keičia vienokia ar kitokia infrastruktūra, registruojanti paslaugas ir klientus bei sudaranti galimybes jiems susisiekti vienas su kitu. Paslaugų paieškai išskirstytose sistemose yra sukurta keletas reikšmingų protokolų. Aptarsime kai kuriuos iš jų.

**SLP** (*service location protocol*, sukurtas IETF) (Guttman ir kt., 1999; Penz, 2005) protokolas skirtas korporatyviniams TCP/IP kompiuterių tinklams. Protokolas yra pritaikytas dideliems korporatyviniams tinklams, bet gali

aptarnauti ir mažus *ad hoc* kompiuterių tinklus, įskaitant belaidžius. Formuliuodamas užklausas klientas privalo nurodyti pageidaujamas paslaugos kokybės charakteristikas. Pavyzdžiui, galima suformuluoti užklausa: *pateikti man visus greitus (ne mažiau kaip 12 puslapių per minutę) spalvotus lazerinius spausdintuvus, spausdinančius A4 formatu ir palaikančius PostScript* (Kerr, 2005). Be to, tai vienas iš nedaugelio protokolų, leidžiančių formuluoti užklausas skirtingomis natūraliomis kalbomis. SLP protokolas nėra susietas su jokia konkrečia programavimo kalba ar technologija ir gali būti naudojamas bet kokioms aparatūrinėms ar programinėms paslaugoms rasti. Kita vertus, paslaugos gali būti labai įvairios. Viena iš paprasčiausių yra valiutos konvertavimo paslauga. Jos kuriamas duomenų srautas yra labai menkas ir dėl jo perdavimo kompiuterių tinklu jokių problemų nekyla. Visai kitaip yra, pavyzdžiui, internetinės telefonijos atveju. Ši paslauga generuoja didžiulį duomenų srautą, kuris turi būti perduodamas tinklu be pertrūkių, nes kitaip pokalbis nevyks. SLP tinklas pats savaime šių problemų nesprendžia. Todėl kuriant SLP tinklą reikia turėti omenyje, kokio tipo paslaugoms jis skiriamas, ir galbūt naudoti papildomas priemones, sprendžiančias kokybiško paslaugų teikimo problemas. SLP technologiją išplėtė SDS (*Service Discovery Service*) technologija (Czerwinski ir kt., 1999), kurioje paslaugos aprašomos ir užklaustos formuluojamos XML kalba ir kuri užtikrina saugų, autentifikuojamą komunikavimą, sudaro prielaidas dirbti globaliuosiuose tinkluose, įskaitant internetą.

**CORBA** (*Common Object Request Broker Architecture*) (Chaffee, Martin, 1999) – OMG sukurtas standartas, kuris leidžia kurti išskirstytas sistemas iš komponentų, parašytų skirtingomis kalbomis ir veikiančių skirtinguose kompiuteriuose ir skirtingose operacijų sistemose. CORBA infrastruktūra pateikia sąsajas, per kurias galima pasinaudoti objektų valdymo funkcijomis arba, kitaip tariant, tos infrastruktūros teikiamomis paslaugomis. Paslaugos aprašomos objekto sąsajoje. Sąsajos aprašomos IDL (*Interface Definition Language*) kalba. Objektų, teikiančių reikalingas paslaugas, klientai gali

\* <http://www.mozenda.com/extraction-software>

\*\* <http://www.freelancer.com/projects/Web-Scraping/Product-locate-scrape-some-locations.html>

ieškoti pagal jų pavadinimus arba pagal pageidaujamas savybes. CORBA infrastruktūra teikia daug įvairių infrastruktūrinių paslaugų, įskaitant įvardijimo (angl. *directory naming*; CORBA DNS), t. y. baltųjų telefono knygos puslapių, ir biržos (angl. *trading*), t. y. geltonųjų telefono knygos puslapių, paslaugas. Buvo siūlymų (Seniwongse, Nanekrangsan, 2001) papildyti CORBA taip, kad paslaugos būtų aprašomos XML, sudarant klientams galimybę ieškoti paslaugų panašiai, kaip tai daroma naudojant universalias paieškos mašinas, lyginant XML dokumentus – kliento užklausa su paslaugų aprašais. Panašių siūlymų pateikta ir daugiau. Tačiau nepavyko rasti informacijos apie tai, ar šie pasiūlymai buvo realiai įgyvendinti. CORBA technologija naudojama elektroninės bankininkystės, e. komercijos, e. mokymosi, e. vyriausybės, e. pramogų ir kitoms išskirstytoms elektroninių paslaugų sistemoms kurti. Su šia technologija dirba dešimtys tūkstančių specialistų. Tačiau tai yra sudėtinga technologija, išskirstytoms CORBA sistemoms kurti reikalingi kvalifikuoti programuotojai. Yra ir daugiau panašių technologijų, pavyzdžiui, *Microsoft DCOM* ir *.Net*.

**Jini** (*Java Intelligent Network Infrastructure*) (Sun, 2003) – *Sun Microsystems* sukurta, *Java* kalba grindžiama išskirstytų sistemų technologija. Ji naudoja nutolusių *Java* metodų kvietimo protokolą (*Java Remote Method Invocation (RMI) Protocol*). *Jini* tinkluose klientai randa ir gauna paslaugas, kurios aprašomos *Java* kalba realizuojamomis parametrizuotomis sąsajomis, kurios gali būti papildytos atributais. Klientai gali būti asmenys, programos arba kitos paslaugos. Paslauga (tiksliau tai, kas ją kuria) gali būti bet kas – įrenginys, programa, asmuo. Paslaugos yra ieškomos naudojant hierarchinę informacinės paieškos paslaugą (angl. *hierarchical lookup service*), turinčią registrą, kuriame paslaugos registruojasi kartu su savo sąsajomis ir objektais. Informacinės paieškos paslaugai rasti naudojamas daigiaabonentis transliavimas (angl. *multicast*) – klientas siunčia užklausa jam žinomam adresu, ji išplatinama tinkle ir aktyvios informacinės paieškos paslaugos praneša klientui apie save. Klientui suradus reikiamą paslaugą ir nusprendus ją pasinaudoti, informacinės paieš-

kos paslauga sukuria lokalų kliento ir paslaugos sąveikos kanalą (angl. *proxy*), kuris yra CORBA technologijos objektų įgaliotinių ir klientų atstovų analogas. *Jini* neužtikrina teikiamų paslaugų lokalizavimo, t. y. klientas negali sužinoti, kiek nuo jo geografiškai yra nutolęs paslaugos teikėjas. Specialios užklausių kalbos *Jini* technologija neturi, o paslaugų šablonai yra *Java* stiliaus ir paprastam klientui nėra tinkami. Todėl reikalinga tarpininkavimo programa, kuri pagal kliento pateiktą informaciją užpildytų tokį šabloną. *Jini* tinklus galima kurti tik tuomet, kai fizinis kompiuterių tinklas turi fiksuotą latentiškumą, pakankamą dažnių juostos plotį (angl. *bandwidth*) ir tenkina tam tikrus kitus reikalavimus, priklausančius nuo paslaugų pobūdžio. Be to, yra tam tikrų apsaugos ir masto keitimo problemų. Technologija vis dar yra intensyviai tobulinama ir kai kurie trūkumai artimiausioje ateityje gali būti pašalinti.

Nors CORBA turi tą pranašumą, kad nėra susieta su jokia konkrečia programavimo kalba ir yra pramoninis standartas, tačiau *Jini* ir RMI naudotis yra paprasčiau, be to, *Java* kodas gali būti mobilus. Tai daro *Jini* patrauklia alternatyva. Kita vertus, CORBA tinkluose programuotojui reikia sukurti tik kliento ir serverio programinę įrangą, o visi *proxy* technologijos elementai generuojami automatiškai. *Jini* tinkluose tą turi padaryti pats programuotojas. Pagal *Sun Microsystems* sumanymą, *Jini* visų pirma yra skirta darbui su įrenginių ir įterptinių sistemų (angl. *embedded system*) teikiamomis paslaugomis ir yra plėtojama būtent šia kryptimi.

## Paslaugų paieška internete

Yra pasiūlyta gana daug metodų ištekliams, įskaitant paslaugas, internete ieškoti.

**DNS** (*domain name system*) (Smallegan, 2005) – tai sistema, sauganti lenteles, susiejančias tekstinius pavadinimus su konkrečiais kompiuterių tinklo adresais ir pateikus pavadinimą grąžinanti atitinkamą kompiuterių tinklo adresą. Kartais klaidingai manoma, kad DNS technologija tik tokiam susiejimui ir gali būti naudojama. Iš tiesų taip nėra, nes naudojama hierarchinė duomenų bazė, kurioje galima saugoti

beveik bet kokio pobūdžio duomenis. Pati DNS technologija riboja tik įrašų pavadinimų ilgį. Daroma prielaida, jog registruoti adresai yra retai keičiami ir gali būti traktuojami kaip statiniai. Nors siekiant užtikrinti gaunamos informacijos aktualumą galima naudoti dinaminį DNS, kuris aktualizuoja įrašus, apskritai DNS technologija nėra pritaikyta tokiam darbui. Ji yra pasiekusi didelę brandą, realizuota daugelio skirtingų firmų, gerai išstudijuota ir aprašyta, parengta daugelio specialistų, gerai išmanančių šią technologiją. Kitas svarbus šios technologijos pranašumas, jog ji globali ta prasme, kad panaudodama DNS serverių hierarchiją adresuoja visą internetą. Yra parengtas standarto „DNS grindžiama paslaugų paieška“ (DNS-SD) projektas (Cheshire, Krochmal, 2011). Tačiau standarto projektas vis dėlto buvo parengtas visų pirma galvojant apie tinklo paslaugas (spausdintuvai, saityno serveriai, skaitmeninės bibliotekos ir pan.), o ne verslo paslaugas. Todėl TXT įrašo ilgis kol kas ribojamas 1300 baitų (toks ilgis parinktas siekiant, kad informacija tilptų į vieną *Ethernet* paketą). Todėl verslo paslaugų kokybei aprašyti to dažniausiai nepakanka ir tokių paslaugų paieškos sistemose greta DNS bazių turi būti naudojamos kitos, papildomos, bazės.

**X.500** technologija (Weider, 1992) buvo sukurta siekiant teikti interneto masto darbo su katalogais paslaugas. Tai daugiau kaip 25 metus turinti technologija, kuri yra įteisinta oficialiu tarptautiniu standartu. Pagrindinis X.500 technologijos tikslas – palengvinti informacijos apdoravimo sistemų integravimą. Katalogu (angl. *directory*) šioje technologijoje vadinama katalogo tipo informacijos saugykla, kartu su programine įranga teikiančia darbo su šia saugykla paslaugas. Visi katalogo vartotojai – žmonės ir programos – gali skaityti kataloge saugomą informaciją ir ją keisti, jeigu jiems yra suteikta tokia teisė. Nors ir sudarantis loginę visumą, fiziškai katalogas yra išskirstytas. Kiekvienai katalogo fizinei daliai atstovauja vadinamasis katalogo sistemos agentas (angl. *directory service agent* – DSA). Jei gavęs užklausą DSA negali rasti reikiamos informacijos savo lokaliwoje informacijos bazėje, jis persiunčia tą užklausą kitiems DSA. Panaši į X.500 technologiją,

bet paprastesnė yra LDAP (angl. *Lightweight Directory Access Protocol*)\*.

**Ninja** technologija (Czerwinski ir kt., 1999; Leschinger, 2005) buvo sukurta siekiant sudaryti prielaidas naudotis patikimomis, keičiamo masto (plečiamomis), išskirstytomis internete paslaugomis, leidžiant prieigai prie tų paslaugų pasitelkti įvairių rūšių įrenginius. Paslaugų kūrimas ir priežiūra yra labai sudėtinga, nes jas realizuojanti programinė įranga turi būti keičiamo masto, labai patikima, nuolat prieinama, nepažeidžianti duomenų darnos. Kol kas nei gatavų komponentų, nei baigtų metodikų tokioms paslaugoms kurti nėra. Technologijoje naudojamos vadinamosios bazės, kurias sudaro galingų darbo stočių klasteriai. Klasteriai užtikrina didelį patikimumą, nes kai kuriems kompiuteriams sutrikus, kiti klasterio kompiuteriai nenustoja veikti. Jie taip pat yra lengvai keičiamo masto, nes klasterį galima papildyti naujais kompiuteriais. Tačiau kurti programinę įrangą darbo stočių klasteriams nėra paprasta. Todėl *Ninja* technologija numato infrastruktūrą *vSpace*, kuri perima paslaugų masto keitimo, komponavimo ir patikimumo rūpesčius, sudarydama prielaidas programuotojui telkti dėmesį į paslaugos funkcionalumo realizavimą. Viena iš infrastruktūros teikiamų paslaugų yra paslauga SDS (*Service Discovery Service*), naudojama paslaugoms registruoti ir registruotoms paslaugoms ieškoti. Ji turi hierarchinę struktūrą ir yra išskirstoma po savotiškas superviršūnes, vadinamas SDS serveriais. SDS ne tik suranda paslaugą, bet ir informuoja klientą, koku mastu verta pasitikėti paslauga, o paslaugos teikėją – kiek vertas pasitikėjimo klientas. Paslaugos aprašomos ir klientų užklausoje formuluojamos XML kalba arba transformuojamos į XML formatą iš įrenginio užklausoje kalbos (pavyzdžiui, naudodamasis mobiliuoju telefonu, klientas gali užklausoje formuluoti balsu).

**Napster** (Tyson, 2009), **Gnutella** (Gnutella, 2002; Brain, 2009), **KaZaA** (Watson, 2009), **BitTorrent** (Cohen, 2003; Cohen, 2008) yra internetinės failų mainų paslaugos, kuriose

\* <http://www.x500standard.com/index.php?n=Participate.LDAPX500>

duomenims persiųsti naudojama lygiarangių tiesioginio susijungimo technologija (angl. *peer-to-peer* – P2P). P2P technologija leidžia paslaugą išskirstyti, t. y. didžiąją dalį failų saugoti ne centralizuotoje saugykloje, bet pačių klientų kompiuteriuose, kuriuose turi būti įdiegta atitinkama klientinė programa. *Napster* buvo panašių sistemų pradininkas. Jis sukurtas 1999 metais ir buvo skirtas MP3 failų mainams tarp muzikos mėgėjų. *Napster* turėjo centrinį serverį, kurio pagrindinė užduotis buvo tvarkyti tuo metu aktyvių klientų indeksą ir sujungti juos vieną su kitu. Gnutelloje, sukurtoje 2000 metais, centrinio serverio atsisakyta. Paslaugos ieškoma naudojant „potvynio“ (angl. *flooding*) paieškos mechanizmą, t. y. persiunčiant užklausą iš bet kurios tinklo viršūnės visoms jos kaimyninėms viršūnėms, taip „užliejant“ tinklą. Taip vyksta tol, kol baigiasi užklauskos gyvavimo laikas (angl. *time to live* – TTL), arba tol, kol randama ieškoma paslauga. *Gnutella* technologija turi tą pranašumą, kad sistema veikia, jei veikia bent vienas tinklo kompiuteris. Tiesa, mažėjant aktyvių kompiuterių skaičiui, paprastai mažėja ir teikiamų paslaugų skaičius. Tačiau technologija turi ir didelių trūkumų, nes nėra garantijų, kad paslauga bus rasta per TTL ir nėra galimybių nurodyti ieškomos paslaugos kokybinių charakteristikų. KaZaA technologija sukurta 2001 metais ir primena Gnutellą, tačiau tinkle naudojamos specialios viršūnės, vadinamos superviršūnėmis (angl. *super-peers*). Užklausa platinama tik per šias viršūnes, dėl to gerokai sumažėja darbo sąnaudos. Superviršūnės tvarko joms priskirto tinklo fragmento indeksą. Kitaip tariant, KaZaA technologija sujungia *Napster* ir *Gnutella* technologijų pranašumus. KaZaA kūrėjai 2003 metais sukūrė ir populiarią *Skype* programą, kurioje taip pat naudojama P2P technologija. *BitTorrent* protokolas buvo sukurtas 2001 metais. Jis skiriasi nuo savo pirmtakų tuo, kad failai skaidomi į fragmentus ir dalimis siunčiami vienu metu iš kelių kompiuterių. Tai gerokai padidino atsisiuntimo greitį. Tuo metu, kai gaunami duomenys, jau turimus duomenis galima siųsti kitiems vartotojams.

**INS/Twine** technologija (Balazinska ir kt., 2002) buvo sukurta 2002 m. ir skirta naudoti

aplinkose, kuriose visos paslaugos yra lygiavertės. Nors autoriai mano, kad technologija galėtų būti naudojama viso interneto mastu, apskritai ji yra pritaikyta mažesniems masteliams, pavyzdžiui, paslaugoms teikti miesto mastu. Naudodami šią technologiją, gyventojai gali gauti orų prognozę, informaciją apie oro užterštumą bei vandens temperatūrą miesto paplūdimiuose, apie artimiausias paslaugų įmones ir pan. Užklauskos gali būti lokalizuotos, t. y. klientas gali prašyti pateikti informaciją apie kokias nors arčiausiai jo buvimo vietos teikiamas paslaugas. Kadangi dideliuose miestuose gali būti labai daug paslaugų ir klientų, technologija turi gebėti tvarkyti didelius tinklus. Teigiama (Balazinska et al., 2002), kad INS/Twine technologija gali palaikyti sistemas, teikiančias iki kelių milijardų paslaugų.

Dar galima paminėti projektus *Globus* ir *Condor*, skirtus skaičiuojamiesiems *Grid* tinklams kurti. *Grid* tinklas integruoja aparatūrinius, programinius, skaičiavimo bei informacinius po įvairias viena nuo kitos dideliais atstumais nutolusias organizacijas išbarstytus išteklius, kuriuos galima panaudoti bendram tikslui pasiekti. Paslaugos ten dažniausiai suprantamos kaip kompiuterinių išteklių teikimo paslaugos. Kadangi šios sistemos nėra tiesiogiai susijusios su mūsų tyrimo tematika, išsamiau jų neanalizuosime.

## Saityno paslaugų paieškos technologijos

Saityno paslaugą (angl. *web service*) galima apibrėžti kaip dalykinės programos sąsają, kuria galima pasinaudoti per kompiuterių tinklą (Dutta, 2008). Saityno paslaugų technologija pašalina du svarbius CORBA ir kitų išskirstytų sistemų trūkumus: (a) šiose sistemose pranešimų, kuriais keičiasi programos, struktūra priklauso nuo konkrečios technologijos, programavimo kalbos ar platformos, o saityno paslaugų technologijoje visi pranešimai rašomi XML kalba ir turi standartinę struktūrą; (b) CORBA turėjo problemų su ugniasienėmis, o saityno paslaugos naudoja HTTP protokolą, kuriam dauguma ugniasienių leidžia dirbti su 80-uju prievadu.

Pagrindinis saityno paslaugų technologijos trūkumas yra tas, kad labai sunku automatizuoti paslaugų paiešką ir komponavimą. Praktiškai būtinas vienoks ar kitoks žmogaus įsikišimas, dėl ko sudėtingose verslo sistemose tokios paslaugos tampa mažai naudingos.

Yra keletas saityno paslaugų paieškos technologijų. Standartine laikoma **UDDI** (*Universal Description Discovery Integration*) technologija. Būtent šią technologiją plėtoja W3C, *Oasis*, JCP ir OMG konsorciškai. Technologija numato centralizuotą paslaugų registrą, organizuojamą pagal integruotųjų portalų geltonųjų puslapių modelį. Organizacijos gali kurti ir eksploatuoti savo privačius registrus naudojamos tam skirtus įrankius, kuriuos teikia gamintojai (pvz., *Apache Axis*, *SOA Software*, *BEA*, *IBM* ir kt.). Vieši UDDI registrai paprastai kuriami programinės įrangos kompanijų, tokių kaip *IBM*, *Microsoft*, *Systinet*, *xMethods*. Saityno paslaugoms aprašyti UDDI naudoja **WSDL** (*Web Services Description Language*) kalbą, kuri yra **CORBA** technologijos **IDL** kalbos analogas. Tame apraše ne tik aprašomas paslaugos funkcionalumas, bet ir nurodoma, kaip ta paslauga pasinaudoti. Tačiau **WSDL** nesuteikia galimybės aprašyti paslaugų semantinių savybių. Ši trūkumą bandoma pašalinti semantinėse saityno paslaugose, kurias aptarsime vėliau.

Pranešimų perdavimas klasikinėje saityno paslaugų technologijoje vyksta pagal **SOAP** (*Simple Object Access Protocol*) protokolą. Yra ir kitokių, nenaudojančių **SOAP**, saityno paslaugų, pavyzdžiui, **RESTful** saityno paslaugos. **REST** (angl. *Representational State Transfer*) yra architektūros stilius (Fielding, 2000), naudojamas saityno sistemoms kurti. **REST** įgyvendinti taikomas ne objektinis požiūris į dalykinę sritį, o išteklių, t. y. sakoma, kad saitynas yra išteklių visuma. Taip sukuriama **REST** stiliaus saityno paslaugos, pagrindinį dėmesį skiriant įvairiose kalbose parašytų programų manipuliavimui išteklių, apimant darbą su išteklių būsenomis, jų adresavimą ir perdavimą pagal **HTTP** protokolą. **REST** turi tam tikrų ribojimų, tačiau jis leidžia paprasčiau realizuoti sąveiką, todėl šis saityno paslaugų modelis dabar naudojamas

vis dažniau (Rodriguez, 2008). Tai liudija **REST** principų taikymas kuriant **Web 2.0** paslaugas teikiančias sistemas, apimant *Yahoo*, *Google* ir *Facebook*.

Specifinė saityno paslaugų rūšis yra **Amazon saityno paslaugos** (angl. *Amazon Web Services*, **AWS**). Tai yra nutolusių paslaugų, internete teikiamų *Amazon.com*, visuma, įgyvendinanti patikimą ir galinčią keisti mastelį „debesies“ kompiuterinę platformą (Amazon, 2011). *Amazon* saityno paslaugos pasiekiamos per **HTTP**, naudojant **REST** ir **SOAP** protokolus. Apmokėjimo suma priklauso nuo naudojimosi, tačiau įvairių paslaugų įkainiai yra skirtingi. Teikiamos paslaugos padeda spręsti šias problemas: programų prieglobos, atsarginio kopijavimo ir saugojimo, turinio teikimo, duomenų bazių, e-komercijos, didelio našumo skaičiavimų ir kt.

**WSPDS** (*Web Services Peer-to-peer Discovery Service*) (Banaei-Kashani ir kt., 2004) yra technologija, įgyvendinanti saityno paslaugų paiešką kaip **P2P** paslaugą. Toks paslaugos realizavimo būdas panaikina būtinybę turėti tos paslaugos teikėją. Eksperimentinė **WSPDS** realizacija naudoja *Gnutella* protokolą, atlikta 2004 metais Pietų Kalifornijos universitete (**JAV**) ir panaudota geoinformacinių paslaugų sistemoje. Nors šis darbas iki šiol yra aptariamas mokslo spaudoje, kokių nors žinių apie tai, ar technologija buvo komercializuota, aptikti nepavyko.

Įvairių saityno paslaugų paieškos technologijų palyginimo ir vertinimo metodika pasiūlyta darbe (Bachlechner ir kt., 2006). Dažniausiai saityno paslaugų paieškos mašinos kaupia paslaugų duomenų bazes, kuriose savo teikiamas paslaugas gali registruoti bet kuris paslaugų teikėjas. Tačiau iš tiesų jų bazės vis dar labai nedidelės. Saityno paslaugų paieškai gali būti naudojamos ir kai kurios universalios paieškos mašinos, pavyzdžiui, *Google*.

**Semantinės saityno paslaugos** – tai tokios paslaugos, kuriose išreikštiniu būdu yra aprašyta jų semantika. Manoma, kad tai turėtų padidinti paieškos efektyvumą, nes atsiranda galimybės ieškoti paslaugų ir informacijos, atsižvelgiant į konkrečias prasmines situacijas. Manoma, jog perėjus prie semantinių saityno paslaugų iš es-

mės supaprastės paslaugų teikėjų lokalizavimas (t. y. geografinės jų buvimo vietos nustatymas) ir automatinis arba pusiau automatinis kooperavimasis su jais. Semantinių saityno paslaugų technologija grindžiama ontologijomis. Ontologijos numato priemones paslaugų semantiniams aprašams rengti. Kol kas dažniausiai naudojamos bendrosios ontologijos, tokios kaip OWL-S (Martin ir kt., 2004), WSDL-S (Akkiraju ir kt., 2005) ir WSMO (Domingue ir kt., 2005), bet pradedamos naudoti ir specifinės konkrečių dalykinių sričių ontologijos, ir tai sudaro prielaidas tiksliau aprašyti specialaus pobūdžio paslaugų semantiką. Paslaugų paieškos požiūriu, svarbiausioji aprašo dalis yra paslaugos profilis (angl. *service profile*). Jame aprašomos funkcinės ir nefunkcinės paslaugos savybės bei kontaktinė informacija. Kaip ir tradicinių saityno paslaugų atveju, semantinių paslaugų technologija numato paslaugų registrą OWL-S UDDI. Pagrindinis skirtumas yra tas, kad šiame registre saugomi semantiniai paslaugų aprašai. Paslaugų paieška registre vykdoma sugretinant klientų užklausas su paslaugų aprašais. Šiuo atveju procesas yra gana sudėtingas, grindžiamas samprotavimais deskriptyviosiose logikose.

## Lokaluotų paslaugų paieškos technologijos

Šiuolaikiniai mažo dydžio belaidžiai įrenginiai (mobilieji telefonai, delniniai kompiuteriai, išmanieji laikrodžiai, GSP imtuvai) leidžia teikti nuo buvimo vietos priklausomas paslaugas. **CoolTown** technologija (Kindberg, Barton, 2001) pateikia infrastruktūrą vadinamiesiems klajokliams skirtiems skaičiavimams (angl. *nomadic computing*). Klajokliai šiuo atveju yra žmonės, judantys tarp savo gyvenamųjų vietų, darboviečių, parduotuvių ir kitų jų kasdiniame gyvenime svarbių vietų. „Skaičiavimais“ vadinamos klajokliams skirtos paslaugos, ypač tos, kurių prireikia realiame materialiaame pasaulyje, kuriame klajoklis juda. Šios technologijos ypatumas yra tas, kad kiekvienas realaus pasaulio daiktas arba su juo susijusi paslauga turi savo

internetinį puslapį, tiksliau tariant, yra susieta su tam tikru saityno ištekliumi, arba saityno būtimi (angl. *web presence*), kuria žmogus gali patogiai pasinaudoti, būdamas netoli to daikto. Kitaip tariant, ši technologija skirta vadinamajam daiktų internetui (angl. *internet of things*). Daikto internetinio puslapio adresu galima sužinoti iš specialios prie daikto pritvirtintos etiketės su brūkšninio kodu, kurį gali nuskaityti, pavyzdžiui, į delninį kompiuterį įmontuotas skaitytuvas. **Splendor** yra viena iš daugelio eksperimentinių paslaugų paieškos technologijų, pritaikytų tokioms aplinkoms, kuriose paslaugas galima surasti, tačiau šiose aplinkose nei mobilūs klientai, nei mobilios paslaugos gali būti neregistruoti technologiją palaikančioje infrastruktūroje (Zhu et al., 2005). *Splendor* technologijoje ypatingas dėmesys skiriamas saugumui ir privatumui užtikrinti, tam naudojami įgaliojtieji serveriai (angl. *proxy*). Ši technologija kol kas yra eksperimentinio pobūdžio, jos brandos laipsnis nedidelis. **Agents2Go** (Ratsimor ir kt., 2001) – tai sistema, skirta lokalizuotų paslaugų paieškai mobiliosios elektroninės komercijos kontekste. Sistemą sudaro programa *PalmApp* delniniame kompiuteryje ir keli serveriniai komponentai – *Agent2Go* serveris, kuris apdoroja užklausas iš *PalmApp*, lokatorius, kuris nustato vietą pagal bazines stotis, informacijos saugyklą, kurioje laikomi statiniai duomenys (pavyzdžiui, restoranų adresai, telefonai, darbo laikas) ir brokeriai, pateikiantys dinaminę informaciją (pavyzdžiui, restorano meniu ir taikomas nuolaidas). Kadangi *Agents2Go* architektūra grindžiama centriniu serveriu, sistema gali apimti tik tam tikro dydžio geografines teritorijas. Lietuvai tai neturėtų būti esminis ribojimas. Tačiau vienu serveriu grindžiama architektūra nėra patikima, nes, sutrikus tam serveriui nustoja veikti visa sistema. Kitas trūkumas yra tas, kad ši architektūra numato tik paslaugų paiešką, bet nenumato kitų svarbių funkcinių galimybių – transakcijų sudarymo, paslaugų teikties stebėsenos, paslaugų komponavimo, veikimo atkūrimo po trikties ir kt. Be to, nepavyko rasti konkrečios informacijos apie komercinę šios technologijos realizaciją.

## Paslaugų paieškos technologijų vertinimas

Išanalizuotų paslaugų paieškos technologijų ekspertinis vertinimas jų tinkamumo mažoms ir vidutinėms Lietuvos įmonėms požiūriu pateikiamas lentelėje.

Vertinimas atliktas penkių balų sistema pagal septynis vertinimo kriterijus. Technologijos *adekvatumas* nusako, kiek technologija yra pritaikyta eiliniams klientams skirtoms bendro pobūdžio paslaugoms ieškoti. Kai kurios technologijos, nors ir gali būti pritaikytos tokiems tikslams, šiuo metu yra naudojamos aparatūri-

nių paslaugų (tinkluose arba kompiuterizuotuose pastatuose), korporacinėms sistemoms reikalingų paslaugų, pramogų, mokslinių paslaugų ar kitokių paslaugų paieškai, ir nėra visiškai aišku, kiek pastangų reikėtų, kad jos būtų pritaikytos bendro pobūdžio paslaugoms ieškoti. *Universalumas* reiškia, kad technologija naudojama ne vien konkrečios rūšies bendro pobūdžio paslaugų, pavyzdžiui, restoranų paslaugų, paieškai. Nors tikėtina, kad tokios technologijos gali būti universalizuojamos, tačiau jei tokia praktika nežinoma, jos priskirtinos prie specializuotų technologijų. *Brandos* kriterijus nusako, kiek

Lentelė. Technologijų vertinimas mažų ir vidutinių Lietuvos įmonių požiūriu

| Technologijos pavadinimas           | Adekvatumas | Universalumas | Brandas | Paprastumas | Paplitimas Lietuvoje | Perspektyvumas | Šaunumas | Σ  |
|-------------------------------------|-------------|---------------|---------|-------------|----------------------|----------------|----------|----|
| Paslaugų paieškos portalai          | 5           | 5             | 5       | 5           | 5                    | 2              | 3        | 30 |
| Pirkimų robotai                     | 5           | 2             | 4       | 4           | 5                    | 2              | 4        | 26 |
| SLP                                 | 2           | 5             | 3       | 3           | 1                    | 3              | 2        | 19 |
| CORBA                               | 2           | 5             | 5       | 3           | 2                    | 3              | 2        | 22 |
| Jini                                | 3           | 5             | 5       | 3           | 3                    | 3              | 2        | 31 |
| DNS                                 | 2           | 2             | 5       | 4           | 3                    | 1              | 4        | 21 |
| X.500                               | 2           | 2             | 5       | 4           | 1                    | 1              | 4        | 19 |
| Ninja/SDS                           | 4           | 5             | 4       | 3           | 1                    | 3              | 2        | 22 |
| Napster                             | 3           | 1             | 5       | 3           | 2                    | 1              | 4        | 19 |
| Gnutella                            | 3           | 1             | 5       | 3           | 2                    | 1              | 4        | 19 |
| KaZaA                               | 3           | 3             | 5       | 3           | 2                    | 2              | 4        | 22 |
| BitTorrent                          | 2           | 3             | 5       | 3           | 2                    | 3              | 4        | 22 |
| INS/Twine                           | 5           | 5             | 3       | 2           | 1                    | 4              | 1        | 21 |
| Saityno paslaugų paieškos mašinos   | 5           | 5             | 4       | 3           | 1                    | 3              | 2        | 23 |
| UDDI                                | 4           | 5             | 5       | 3           | 1                    | 4              | 2        | 26 |
| REST saityno paslaugos              | 4           | 5             | 3       | 3           | 1                    | 4              | 2        | 22 |
| Amazon saityno paslaugos            | 5           | 5             | 5       | 4           | 2                    | 3              | 4        | 28 |
| WSPDS                               | 5           | 5             | 3       | 3           | 1                    | 4              | 3        | 24 |
| Semantinių saityno paslaugų paieška | 5           | 5             | 2       | 2           | 1                    | 4              | 1        | 20 |
| CoolTown                            | 2           | 2             | 4       | 2           | 2                    | 5              | 1        | 18 |
| Splendor                            | 2           | 2             | 2       | 2           | 1                    | 5              | 1        | 15 |
| Agents2Go                           | 5           | 1             | 3       | 2           | 2                    | 3              | 1        | 19 |

seniai ir kiek plačiau naudojama technologija. Technologijos *paprastumo* kriterijumi įvertinama, ar pakankamai lengva yra sukurti, įdiegti ir prižiūrėti technologijos infrastruktūrą, ar tam reikia labai aukštos kvalifikacijos specialistų. *Paplitimo Lietuvoje* kriterijumi įvertinama, kiek Lietuvoje yra paplitusios tai technologijai reikalingos priemonės. Juk kai kurios paslaugų specifikuojimo ir publikavimo priemonės Lietuvoje yra beveik nežinomos, kitos gerai žinomos tik akademinėje aplinkoje, dar kitos, pavyzdžiui, HTML, naudojamos visuotinai. Pagal *perspektyvumo* kriterijų vertinama, koku mastu technologija atitinka šiuolaikines IT raidos tendencijas, ar ji nėra senstelėjusi ir sunkiai modernizuojama. *Sąnaudų* kriterijaus balas tuo aukštesnis, kuo mažesnės yra darbo ir kitos sąnaudos, reikalingos technologijai įdiegti (programavimas, diegimas, priemonių išsigijimas, organizacinė veikla buriant paslaugų teikėjus, duomenų rinkimas ir aktualizavimas, infrastruktūros administravimas ir kt.).

Visi lentelėje pateikti vertinimai yra eksperimentiniai. Skaičiuojant suminius vertinimus visi kriterijai buvo laikomi lygiaverčiais, jokie specialūs agregavimo metodai nebuvo naudojami vertinimams pagal konfliktuojančius kriterijus agreguoti. Todėl lentelė duoda tiksliai apytikrą vaizdą, tačiau sudaro gerą pagrindą atlikti tikslesnius vertinimus.

## LITERATŪRA

ADJIE-WINOTO, W.; SCHWARTZ, E.; BALAKRISHNAN, H.; LILLEY J. (1999). The design and implementation of an intentional naming system. In *Proc. ACM Symposium on Operating Systems Principles*, p. 186–201. Prieiga per internetą: <<http://nms.csail.mit.edu/papers/ins-sosp99.pdf>>.

AKKIRAJU, Rama et al. (2005). *Web Service Semantics – WSDL-S*. W3C Member Submission 7 November 2005. Prieiga per internetą: <<http://www.w3.org/Submission/WSDL-S/>>.

*Amazon Web Services* (2011). Amazon Web Services LLC, 2011. Prieiga per internetą: <<http://aws.amazon.com/>>.

BACHLECHNER, D.; SIORPAES, K.; FENSEL, D.; TOMA I. (2006). *Web Service Discovery – A Reality Check*. Technical Report. DERI – Digital Enter-

## Išvados

Mūsų nuomone, paslaugų paieškos verslą norinčioms pradėti mažoms ir vidutinėms Lietuvos įmonėms tikslinga rekomenduoti pirkimų robotų technologiją, užtikrinant prieigą prie paslaugų paieškos portalo iš mobiliųjų telefonų ir lokalizuotų paslaugų (t. y. geografiškai arti esančių paslaugų teikėjų teikiamų paslaugų) paiešką. Pradėjus tokį verslą pamažu reikia rengtis pereiti prie saityno paslaugų paieškos technologijų. Tačiau semantinių saityno paslaugų technologija dar nėra pakankamai brandi. Be to, net ir paprastų saityno paslaugų technologija reikalauja, kad paslaugų teikėjai specifikuotų savo teikiamas paslaugas WSDL kalba. Tačiau Lietuvoje kol kas ši technologija dar mažai žinoma ir bandymas kurti saityno paslaugų paieškos verslą, mūsų nuomone, būtų per ankstyvas. Toks perėjimas yra prasmingas tik pavykus suburti su paieškos portalu glaudžiai bendradarbiaujančių paslaugų teikėjų bendruomenę ir įtikinus ją tokio perėjimo tikslingumu.

## Padėka

Dėkojame už malonų bendradarbiavimą UAB „Fresh ID“ atstovams Arnoldui Vasiliauskui ir Mantui Viliui.

prise Research Institute, January 17, 2006. Prieiga per internetą: <<http://www.deri.ie/fileadmin/documents/DERI-TR-2006-01-17.pdf>>.

BALAZINSKA, M.; BALAKRISHNAN, H.; KARGER, D. (2002). INS/Twine: A Scalable Peer-to-Peer Architecture for Intentional Resource Discovery. In *Proceedings of the First International Conference on Pervasive Computing. LNCS 2414*, Springer-Verlag, p. 195–210. Prieiga per internetą: <<http://nms.csail.mit.edu/papers/index.php?detail=38>>.

BANAEI-KASHANI, F.; CHEN, CH.-CH.; SHAHABI, C. (2004). WSPDS: Web Services Peer-to-peer Discovery Service. In *Proceedings of the International Symposium on Web Services and Applications (ISWS'04)*, Las Vegas, USA, 21–24 June 2004.

BRAIN, Marshall (2009). *How Gnutella Works* [interaktyvus]. Prieiga per internetą: <<http://computer.howstuffworks.com/file-sharing.htm>>.

CHAFFEE, A.; MARTIN, B. (1999). *Introduction to CORBA*. JGuru Training by the Magelang Institute.

CZERWINSKI, S.; ZHAO, B.; HODES, T.; JOSEPH, A.; KATZ, R. (1999). An Architecture for a Secure Service Discovery Service. In *Proc. ACM/IEEE MOBICOM*, August 1999, p. 24–35. Prieiga per internetą: <<http://iceberg.cs.berkeley.edu/papers/Czerwin-Mobicom99/>>.

CHESHIRE, S.; KROCHMAL, M. (2011). *DNS-Based Service Discovery*. Internet-Draft (submitted to IETF), Apple Inc. [interaktyvus]. Prieiga per internetą: <<http://files.dns-sd.org/draft-cheshire-dnsextdns-sd.txt>>

COHEN, Bram (2003). Incentives build robustness in BitTorrent. In *Proc. of First Workshop on Economics of Peer-to-Peer Systems*, Berkeley, USA, June 2003. Prieiga per internetą: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.14.1911&rep=rep1&type=pdf>>.

COHEN, Bram (2008). *The BitTorrent Protocol Specification*. Prieiga per internetą: <[http://www.bit-torrent.org/beps/bep\\_0003.html](http://www.bit-torrent.org/beps/bep_0003.html)>.

DOMINGUE, J.; ROMAN, D.; STOLLBERG, M. (2005) Web services modeling ontology (WSMO) – An ontology for semantic web services. In *W3C Workshop on Frameworks for Semantics in Web Services*, June 2005, Innsbruck, Austria. Prieiga per internetą: <[http://www.w3.org/2005/04/FSWS/Submissions/1/wsmo\\_position\\_paper](http://www.w3.org/2005/04/FSWS/Submissions/1/wsmo_position_paper)>.

DUTTA, Biswanath (2008). *Semantic Web Services: A Study of Existing Technologies, Tools and Projects*. *DESIDOC Journal of Library & Information Technology*, vol. 28, no. 3, p. 47–55. Prieiga per internetą: <<http://publications.drdo.gov.in/ojs/index.php/djlit/article/viewFile/176/87>>.

FIELDING, Roy Thomas (2000). *Architectural Styles and the Design of Network-based Software Architectures*: Dissertation. University of California, Irvine.

*Gnutella RFC* (2002). Prieiga per internetą: <<http://rfc-gnutella.sourceforge.net/>>.

GUTTMAN, E.; PERKINS, C. et al. (1999). Service Location Protocol, Version 2. RFC 2608, June 1999. Prieiga per internetą: <<http://ietf.org/rfc/rfc2608.txt>>.

KINDBERG, T.; BARTON, J. (2001). A Web-Based Nomadic Computing System. *Computer Networks*, vol. 35, no. 4, p. 443–456. Prieiga per internetą: <<http://web.archive.org/web/20030429031043/www.cooltown.com/dev/wpapers/nomadic/nomadic.asp>>.

KERR, C. (2005) *Service Discovery & Advertisement Techniques*. Internetworking (TELE402), University of Otago, Dunedin, New Zealand 2005. Prieiga per internetą: <<http://www.telecom.otago.ac.nz/tele402/files/servdisc.pdf>>.

LESCHINGER, B. (2005). *Mobile Frameworks: Ninja & OSGI*. Institute of Computer Science, University of Innsbruck, January, Prieiga per internetą: <[http://www.sti-innsbruck.at/fileadmin/documents/teaching\\_archive/acsp0405/18\\_Leschinger\\_Ausarbeitung.pdf](http://www.sti-innsbruck.at/fileadmin/documents/teaching_archive/acsp0405/18_Leschinger_Ausarbeitung.pdf)>.

MARTIN, D. et al. (2004). *OWL-S: Semantic Markup for Web Services*. W3C Member Submission 22 November 2004. Prieiga per internetą: <<http://www.w3.org/Submission/OWL-S/>>.

MONTGOMERY, A. L.; HOSANAGAR, K.; KRISHNAN, R.; CLAY, K. B. (2004). Designing a Better Shopbot. *Management Science*, vol. 50, no. 2, p. 189–206. Prieiga per internetą: <<http://www.andrew.cmu.edu/user/alm3/papers/shopbot%20design.pdf>>.

PENZ, S. (2005). SLP based Service Management for Dynamic Adhoc Networks. In *3rd International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC05) Proceedings*. Prieiga per internetą: <<http://middleware05.objectweb.org/WSPProceedings/MPAC05/a18-penz.pdf>>.

RATSIMOR, O.; KOROLEV, V.; JOSHI, A.; FININ, T. (2001). Agents2Go: An Infrastructure for Location-Dependent Service Discovery in The Mobile Electronic Commerce Environment. In *WMC'01 Proceedings of the 1st international workshop on Mobile commerce*, July 2001, p. 31–37. Prieiga per internetą: <<http://www.cs.umbc.edu/~finin/papers/amcw01/a2go.pdf>>.

ROMPOTHONG, Pornpong; SENIVONGSE, Twittie (2003). A query federation of UDDI registries. In *Proc. of the 1st International Symposium on Information and Communication Technologies*. Trinity College Dublin, p. 561–566.

ROWSTRON, Antony I. T.; DRUSCHEL, Peter (2001). Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware '01 Proceedings of the IFIP/ACM*

*International Conference on Distributed Systems Platforms*. Heidelberg–London: Springer-Verlag, p. 329–350.

SENIWONGSE, T.; NANEKRANGSAN, W. (2001). An extension to a CORBA trader to support XML service descriptions. In K. Zielinski, K. Geihs, A. Laurentowski (eds). *New Developments in Distributed Applications and Interoperable Systems*. Kluwer Academic Publisher, p. 85–98.

SMALLEGAN, R. (2005). *DNS Based Service Discovery (DNSSD)*. Smallegan.com, 2005. Prieiga per internetą: <<http://www.smallegan.com/blog/2005/06/09/dns-based-service-discovery-dns-sd/>>.

STOICA, Ion; MORRIS, Robert; KARGER, David; KAASHOEK, M. Frans; BALAKRISHNAN, Hary (2001). Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM '01 Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for compu-*

*ter communications*, ACM, New York, p. 149–160. Prieiga per internetą: <[http://pdos.csail.mit.edu/papers/chord:sigcomm01/chord\\_sigcomm.pdf](http://pdos.csail.mit.edu/papers/chord:sigcomm01/chord_sigcomm.pdf)>.

Sun Microsystems Inc. (2003). *Jini Architecture Specification*. Version 2.0, June 2003.

TYSON, Jeff (2009). *How the Old Napster Worked* [interaktyvus]. Prieiga per internetą: <<http://computer.howstuffworks.com/napster.htm>>.

WATSON, Stephanie (2009). *Next Page How KaZaA Works* [interaktyvus]. Prieiga per internetą: <<http://computer.howstuffworks.com/kazaa.htm>>.

WEIDER, C.; REYNOLDS, J.; HEKER, S. (1992). *Technical Overview of Directory Services Using the X.500 Protocol*. March 1992. RFC 1309.

ZHU, F.; MUTKA, M. W.; NI, L. M. (2005). Service Discovery in Pervasive Computing Environments. *Pervasive computing*, vol. 4, issue 4, p. 81–89.

## EVALUATION OF SUITABILITY OF INTERNET SERVICE SEARCH TECHNOLOGIES FOR E-COMMERCE

Albertas Čaplinskas, Audronė Lupeikienė, Laima Paliulionienė

### Summary

The paper analyzes Internet service systems of different architecture, paying special attention to the mechanisms of advertizing and finding services. Their pros and cons are discussed in the context of small and medium enterprises (SME) that want to use the technology for e-commerce in Lithuania. The paper overviews the technologies of online shops, shopbots, service search in open distributed proces-

sing (SLP protocol, CORBA, Jini), search in the Internet (UDDI, DNS, X.500, Ninja, peer-to-peer file sharing services), web services, and nomadic computing. A quantitative expert evaluation of each technology is performed taking into account its adequacy for the task under consideration, universality, simplicity, maturity, cost and other criteria.