VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Magistro baigiamasis darbas

Portalo kūrimas, naudojant kitų svetainių duomenis

Creating a portal using data from other sites

Atliko: magistro 2 kurso studentas
Mindaugas Žemaitis

Darbo vadovas:
Lekt. Tadas Savičius

Recenzentas:
Asist.. Mindaugas Žilinskas

Vilnius
2008

Santrauka

Duomenų importas reiškia taikomųjų programų panaudojimą, siekiant gauti kuo daugiau informacijos iš duomenų šaltinių, nuskaityta informacija turi būti pritaikyta naudojamai sistemai ir paruošta saugojimui.

Duomenų šaltiniai – priemonės, kurių pagalba duomenys perduodami portalui. Vienas sunkiausių uždavinių – kaip perduoti duomenis tinkama forma kitai sistemai, kuri galėtų juos nuskaityti ir išsaugoti bendroje duomenų bazėje, užtikrinti nuolatinį informacijos atnaujinimą. Tam nagrinėjami įvairūs duomenų šaltinių metodai, ieškoma naujų alternatyvų. Paprastai kartu su duomenų perdavimo, nagrinėjami ir jų gavimo metodai, būdai. Šioje srityje naudojamos kompiuterinės programos (taikomosios programos), skirtos duomenų gavimui ir paruošimui saugoti iš šaltinio. Dažniausiai importo sistemos naudoja tik joms tinkamą duomenų struktūros standartą, todėl šaltiniai turi pateikti duomenis, suformuotus laikantis jame nustatytų taisyklių. Jei duomenys kitokios struktūros, nei reikalaujama importo sistemoje, jie netinkami. Darbe išnagrinėjama minėta duomenų struktūrų nesuderinamumo problema, siekiama, kad taikomosios programos būtų suderinamos su daugeliu informacijos apsikeitimo formų, veiktų patikimai ir greitai.

Summary

Data import means creation of applications to get as much as possible information from data sources, collected information must be fitted for the using system and be prepared for saving.

Data sources – tools that supply data for portals. One of the most difficult problems are how to supply fitted data for other system that could collect and save it in main database, ensure regular update of information. For this reason in this work methods of data sources are analyzed, searching for new alternatives. Simply data supply and receive methods are analyzed in one time. In this case usually usable computer programs (applications) that gather and tailor data from data sources. Mostly import systems use one standard of data structure that fits for them, so data sources must supply data using that rules of data structure. If data is in inappropriate form, they are unacknowledged. In work this problem of different data structure that unacknowledged in other systems is analyzed, to make applications be appropriate with many forms of data. These applications must work reliable and fast.

Turinys

Įvadas

Šiais laikais internetas – viena pagrindinių paieškos priemonių norint rasti reikiamą informaciją. Jos kiekis milžiniškas, panašių šaltinių gausu, todėl vis populiarėja internetiniai portalai, kurie pateikia specifikuotą informaciją vienoje vietoje.

Apskritai, portalas yra suvokiamas kaip svarbiausia pradžios vieta, kai vartotojai prisijungia prie pasaulinio tinklo, nes suteikia galimybes susiorientuoti teikiamos informacijos, žinių ir paslaugų visumoje. Nuo portalų atsiradimo, jų naudojimas sparčiai plito, nuolat auga iki šiol, pakito ir samprata: nuo sistemos, užtikrinančios vieningą informacijos prieigą internete, prie daugiafunkcinės sistemos, skirtos įvairioms vartotojų grupėms verslo ar institucijų portaluose, atskiroms profesijoms, gyventojų grupėms ar jų interesams.

Viena svarbiausių sričių portaluose yra importo sistemos, kurios atlieka duomenų surinkimo bei atnaujinimo funkcijas. Vis didėjant įvairiausių portalų skaičiui, informacijos kiekiui, daug dėmesio skiriama importo sistemų nagrinėjimui, naudojamų duomenų surinkimo priemonių, metodų tobulinimui, stengiamasi išplėsti jų galimybes, rasti naujų, greitesnių, patikimesnių būdų kaip surinkti informaciją iš daugelio šaltinių ir apjungti ją bendroje sistemoje.

Duomenų šaltiniais laikome priemones, kurių pagalba duomenys perduodami portalui. Yra jau sukurtų priemonių duomenims perduoti, paprastai jie pagrįsti interneto standarto paslaugomis, taip pat naudojami kiti būdai, susiję su užklausų formavimu duomenų bazėms, turinio valdymo sistemomis. Darbe siekiame išnagrinėti kuo daugiau galimų duomenų šaltinių būdų, siekiant išplėsti duomenų gavimo galimybes importo metu.

Paprastai kartu su duomenų perdavimo, nagrinėjami ir jų gavimo metodai, būdai. Šioje srityje paprastai naudojamos taikomosios programos, kitaip tariant kompiuterinės programos, skirtos duomenų gavimui ir paruošimui saugoti iš šaltinio. Dažniausiai importo sistemose laikomasi tam tikro, tik joms tinkamo duomenų pateikimo standarto, kitaip tariant šaltinis turi pateikti duomenis, suprantama forma, laikantis juos gaunamos sistemos nustatytų duomenų sudarymo taisyklių. Jei duomenys kitokios struktūros, nei reikalaujama importo sistemoje, jie netinkami. Darbe plačiai išnagrinėjama minėta duomenų struktūrų nesuderinamumo problema, siekiama, kad taikomosios programos būtų suderinamos su daugeliu informacijos apsikeitimo formų, veiktų patikimai ir greitai.

Taigi bendru atveju duomenų importas reiškia taikomųjų programų panaudojimą, siekiant gauti kuo daugiau duomenų, nuskaityta informacija turi būti pritaikyta naudojamai sistemai ir paruošta saugojimui. Vėliau atliekami reikalingi išskyrimo, apjungimo, grupavimo ir kiti veiksmai, tenkinantys jiems keliamus reikalavimus išvedant, pateikiant. Duomenų gavimas, veiksmų su jais uždaviniai gali būti atliekami daugeliu būdų, metodų, daugiausiai nagrinėjamų

interneto technologijų srityje. Importo sistema kuriama remiantis informacinės sistemos teikiamomis paslaugomis, kurios yra pagrįstos šiuolaikiniais interneto standartais, interneto naršymo programų paradigmomis.

Taigi dėl duomenų struktūros skirtumų bei būdų kaip juos pateikti, gauti, duomenų apsikeitimas yra sudėtingas, interneto standartų paslaugų (*angl. Web services*) ir kitų technologinių žinių reikalaujantis uždavinys, tačiau atsiveriančios plačios informacijos ir komunikacijos technologijų taikymo perspektyvos, sudaro galimybes pateikti įvarius su duomenų apsikeitimu susijusius sprendimus, kurie gali būti įgyvendinami portaluose.

Darbo tikslas:

- Ištirti ir išplėsti importo sistemos veikimą, skirtą duomenų gavimui iš išorinių šaltinių, siekiant padidinti suderinamumo galimybes su daugiau galimų duomenų šaltinių, negu tai atliekama su jau naudojamomis, sukurtomis sistemomis.

Darbo uždaviniai:

- Atrinkti geriausias esamų importavimo sistemų veikimo savybes, praktikas, papildyti jas naujomis savybėmis, išplečiančiomis jų galimybes.
- Išnagrinėti kaip perduoti duomenis tinkama forma juos gaunančiai sistemai, kuri galėtų juos nuskaityti, apjungti bendroje sistemoje, užtikrinti nuolatinį atnaujinimą.
- Išnagrinėti duomenų perdavimo problemą, kai šaltinis turi reikiamos informacijos portalui, tačiau neturi galimybės jos perduoti. Išanalizavus duomenų perdavimo būdus, metodus, pateikti priemones problemai spręsti.
- Išanalizavus duomenų gavimo būdus, priemones, pateikti sprendimus problemai, kylančiai kuomet šaltinio perduodami duomenys nėra pritaikyti juos gaunančiai sistemai ir jis prarandamas.

1. Problemos formuluotė

Šio darbo tikslas: Ištirti ir išplėsti importo sistemos veikimą, skirtą duomenų gavimui iš išorinių šaltinių, siekiant padidinti suderinamumo galimybes su daugiau galimų duomenų šaltinių negu tai atliekama su jau naudojamomis, sukurtomis sistemomis.

Darbe nagrinėjamos importo sistemos, naudojamos portaluose. Visų portalų pagrindinė funkcija yra apjungti ir susisteminti informaciją iš daugelio išorinių šaltinių ir suteikti galimybę ją pasiekti vienoje vietoje arba nurodyti kur jos ieškoti. Vienas sunkiausių uždavinių – kaip perduoti duomenis tinkama forma kitai sistemai, kuri galėtų juos nuskaityti ir išsaugoti bendroje duomenų bazėje, užtikrinti nuolatinį informacijos atnaujinimą. Kitaip šis uždavinys laikomas kaip duomenų apsikeitimo problema, nagrinėjama importo sistemų, užtikrinančių duomenų gavimą iš šaltinių bei jų atnaujinimą, srityje.

Tipiška duomenų apsikeitimo problemą skaidyti į dvi dalis:

1. Duomenų perdavimo problema. Šaltinis turi reikiamos informacijos portalui, tačiau neturi galimybės jos perduoti (galimų šaltinių problema).
2. Duomenų gavimo problema. Šaltinio perduodami duomenys nėra pritaikyti juos gaunančiai sistemai, todėl nepriimami (duomenų struktūrų nesuderinamumo problema).

Detaliau nagrinėjame antrąją duomenų apsikeitimo problemą, kadangi iš jos kyla kitos, ne mažesnės problemos. Importo sistemą sudaro dvi pagrindinės dalys – duomenų šaltiniai, kurie perduoda informaciją ir taikomosios programos, kurios ją nuskaito ir paruošia saugojimui. Duomenų šaltiniais laikome priemones, kurių pagalba duomenys perduodami kitai sistemai, mūsų atveju portalui, o taikomosios programos, tai kompiuterinė programa ar programų paketas, skirtas duomenų gavimui ir paruošimui saugoti iš duomenų šaltinio. Vienos didžiausių problemų šiose dalyse yra duomenų struktūrų nesuderinamumas, kuris kyla dėl dviejų priežasčių:

- Naudojamos taikomosios programos dažniausiai nėra pritaikytos įvairioms duomenų formoms, laikomasi vieno šablono, pagal tam tikrą standartą suformuotų duomenų struktūros, tik tokie duomenys laikomi tinkamais nuskaitymui ir saugojimui.
- Duomenų šaltinių duomenys dažnai skirtingai suformuoti, pritaikyti savitai specifinei saugojimo formai, tad neatitinka taikomosioms programoms priimamo standarto.

Taikomųjų programų negalėjimas nuskaityti turinčio mums reikiamos informacijos šaltinio duomenų, neišvengiamai reiškia to šaltinio praradimą, todėl ypač svarbu turėti tokią importo sistemą, gebančią priimti daugelio šaltinių pateikiamus duomenis ir sėkmingai surinkti

informaciją, šaltinio neprarandant. Į šią problemą ir jos sprendimą galime žvelgti taip: taikomosios programos priima jai pritaikytą duomenų pateikimo standartą, siekiame atvirkščiai, jas pritaikyti prie įvairių duomenų formų, kuriuos tik gali pateikti šaltinis.

Keliamų problemų aktualumas priklauso nuo portalų, kurie naudoja importo sistemas, rūšies, kokio tipo informaciją jie pateikia, tam atskleisti, sekančiame skyriuje plačiau nagrinėjame portalus.

2. Portalų tipai

Portalai naudojami įvairiose srityse, šioje dalyje siekiame jas išskirti, plačiai išnagrinėti kokia jų paskirtis, teikiami privalumai.

Bendrąja prasme portalai neretai yra apibūdinami kaip interneto vartų, kurie pateikia nuorodų į kitus interneto informacijos išteklius sankaupas ir užtikrina vartotojui vieningą darbo su ištekliais pradinę vietą, sinonimas [WP08]. Portalų pagalba vartotojai gali nesunkiai rasti, kur ieškoti informacijos arba rasti daug specializuotos, susistemintos, sugrupuotos informacijos vienoje vietoje, be ko būtų sunku susiorientuoti didžiuliame informacijos kiekyje internete.

Išskiriamos dvi portalų rūšys:

- Vertikalūs portalai arba specializuotos informacijos nišų (pvz., žinių portalai, elektroninio verslo portalai, korporatyviniai portalai). Tai portalai, kurių vartotojai užsiima konkrečia veikla ar juos sieja bendri interesai. Tokio portalo turinys priklauso nuo konkrečios srities ir jis turi tenkinti tos srities specialistų poreikius.
- Horizontalūs portalai arba bendrosios paskirties (pvz., paieškos portalai kaip google, yahoo) - tai portalai, skirti visiems interneto vartotojams ir kuriuose dažniausiai teikiamos tokios paslaugos kaip informacijos ir žinių paieška, naujienų skelbimas, rekomendacinės nuorodos, bendravimo tinkle priemonės ir pan.

Abiejų rūšių portalai tarpusavyje susiję tuo, kad horizontalūs laikomi tarsi atskaitos pradžia, nuo kurios vartotojas pradeda ieškoti informacijos, šie portalai gali vesti į kitus, horizontalius ar vertikalius portalus.

Mažiau naudojamas, tačiau portalų skirstymas taip pat galimas į:

- Vidinius arba kitaip intraneto (pvz., įmonės informacijos, darbų valdymo portalai)
- Išorinius arba ekstraneto (pvz., elektroninio verslo portalas, asmeninis portalas)
- Viešas (pvz., pramoninis portalas)

Internete informacijos kiekis milžiniškas, problema atsiranda, kai nėra žinoma kurioje vietoje jos ieškoti. Horizontalių portalų pateikiama informacija pagrinde yra nuorodų pavidalu, vedančių į interneto svetaines. Vartotojas nurodo kriterijus, pagal kuriuos nori rasti susijusią informaciją, jam grąžinamas lankytinų nuorodų sąrašas. Privalumas tas, kad vartotojui nereikia prisiminti begalės nuorodų, kur yra informacija, jam užtenka žinoti vieną nuorodą, pavyzdžiui paieškos sistemos, kurioje nukreipiamas atitinkamai pagal reikiamos informacijos tematiką. Jei panašių šaltinių, su vienodos tematikos informacijos daug, išryškėja vertikalių portalų privalumas, kadangi jie skirti ne tik nuorodoms į kitas svetaines pateikti, juose saugoma informacija platesnė, apima paslaugų, prekių pilnus aprašymus ir pan. Taupydami laiką,

vartotojai, gali nelankyti paprastai labai didelio sąrašo skirtingų šaltinių, o viską rasti (pvz., paslaugą, prekę) vienoje vietoje, kur daugelio šaltinių informacija saugoma bendroje sistemoje.

Portalų naudojimo privalumai pavaizduoti 1. lentelėje.

Problema	Sprendimas
Nėra žinoma kur ieškoti reikiamos informacijos.	Horizontalūs portalai
	Pateikiamas sąrašas su šaltiniais, galinčiais suteikti reikalingą informaciją
Panašios tematikos šaltinių internete labai daug, todėl atsirinkimas tinkamiausios, geriausios informacijos užima daug laiko, kaskart reikia naujai perprasti šaltinio struktūrą, informacijos paieškos principą, atrasti kur pateikiama informacija.	Vertikalūs portalai
	Informacija laikoma bendroje sistemoje, nereikia aplankyti įvairių šaltinių, užtenka perprasti portalo struktūrą, paieškos principą vieną kartą.

1. lentelė. Portalų naudojimo privalumai.

Portalai sprendžia didelio kiekio informacijos sisteminimo, paskirstymo problemas, kas leidžia nepasimesti jos gausoje, atsirinkti kas reikalinga. Kuo informacijos portale daugiau, tuo didesnė tikimybė, kad vartotojas ras ko ieško, tam pasiekti importo sistema turi gebėti surinkti kuo daugiau duomenų iš išorinių šaltinių.

Taigi, pagrinde portalai skirstomi į horizontalius ir vertikalius, skirstymas priklauso nuo pateikiamos tematikos, atliekamos funkcijos. Visų rūšių portalai naudoja šaltinius, iš kurių surenkama informacija, pagrindinės čia kylančios problemos:

- Šaltinis turi reikiamos informacijos portalui, tačiau neturi galimybės jos perduoti
- Šaltinio perduodami duomenys nėra pritaikyti juos gaunančiai sistemai ir jis prarandamas.

Nustatyti, kokių rūšių portalams minėtos problemos aktualiausios, skirtas sekantis skyrius.

3. Tyrimas. Nagrinėjamoji sritis

Praeitame skyriuje išskirta, kuriose srityse naudojami portalai, kokia jų atliekama paskirtis, funkcija. Šioje dalyje siekiama išsiaiškinti, kuriose portalų naudojimo srityse keliamos problemos aktualiausios, dažniausiai kylančios. Atliktas tyrimas, naudojantis duomenimis iš statistikos duomenų šaltinių, tokių kaip: www.hey.lt; www.stats.lt; www.google.com/analytics.

Tyrimo pagrindinis tikslas: nustatyti, kurioje portalų srityje darbe nagrinėjamos problemos aktualiausios, kitaip tariant, rasti *nagrinėjamą sritį*, kurioje bus atliekamas tolesnis tyrimas. Tolesnio tyrimo metu siekiama galutinai įvertinti darbe nagrinėjamų problemų aktualumą, svarbumą, sužinoti kokie duomenų perdavimo būdai naudojami jau sukurtose, veikiančiose sistemose, kodėl pasirenkamas būtent tam tikras metodas, kurie jų paplitę daugiau, kurie mažiau, kokie sunkumai dažniausiai sprendžiami duomenų apsikeitimo procese. Taip pat siekiame įsitikinti, kad darbe pateikti sprendimai, susiję su duomenų perdavimo ir gavimo metodais veikia, jų pagalba pasiekiamas viso darbo tikslas, juos realizuojant.

Nagrinėjamai sričiai rasti, nustatome kriterijus portalams, kur probleminių situacijų pasitaikymo tikimybė didžiausia:

1. Svarbu, kad portale būtų kuo daugiau šaltinių, kurie pateiktų didelio kiekio informaciją. Visuomet sunkiau atlikti įvairius veiksmus su daug informacijos, didesnių problemų gali kilti juos nuskaitant, apdorojant, be to, aktualesnė procesams skirto laiko, kitų resursų taupymo problema, kad sistema veiktų greitai ir patikimai.
2. Svarbu, kad informacijai būtų vykdomas nuolatinis atnaujinimo procesas. Priešingu atveju tyrimo metu nebūtų išsiaiškinta kokie sunkumai kyla šio proceso metu.
3. Aktualu, kad kuo mažiau informacija būtų suvedama rankiniu būdu. Jei šaltiniai yra pavieniai asmenys, kurie suveda, koreguoja savo informaciją, nėra svarbūs įvairūs duomenų perdavimo būdai, metodai, kas svarbu vykdant duomenų importą automatiniu būdu
4. Taip pat svarbus faktorius, ar portalas populiarus. Jei jis lankomas, naudojamas, plačiai paplitęs, vadinasi šaltiniai suinteresuoti pateikti juose savo informaciją, kad vartotojai ją lengviau, dažniau pasiektų, taigi didėja tikimybė, kad šaltinių skaičius tokiuose portaluose didelis, jie pateikia daug informacijos, yra daugiausiai, dažniausiai spręstos duomenų apsikeitimo problemos, kas padės įvertinti darbe nagrinėjamų duomenų apsikeitimo problemos aktualumą.

Visų pirma išskirtos portalų sritys, tenkinančios ketvirtą kriterijų. Populiariausių portalų paieškos tyrimo metu, buvo naudojami minėti statistikos duomenų šaltiniai (hey.lt, stats.lt, google analytics), gauti didžiausią lankomumą turintys portalai:

- Pažinčių (bendravimo) portalai
- Naujienų portalai
- Skelbimų portalai

Toliau nagrinėjame kiekvienus išskirtus specializuotų portalų variantus:

- Pažinčių portaluose automatinės importavimo sistemos paprastai netaikomos, kadangi pateikiama informacija talpinama pavienių asmenų rankiniu būdu, nereikalingas ir nuolatinis senos informacijos atnaujinimas, todėl ji neatitinka mums svarbių kriterijų.
- Naujienų portaluose duomenys gali būti pateikiami iš daugelio šaltinių, su didelio kiekio informacija, tačiau naujienoms nėra svarbu, kad senosios būtų kas kart atnaujinamos, taigi ši sritis atitinka pirmą kriterijų, tačiau neatitinka reikalavimų antrajam.

Pasirinkti skelbimų portalai. Pasirinkimą lėmę faktoriai:

- Skelbimų portaluose naudojamų šaltinių kiekis didžiausias (lyginant su prieš tai minėtais), pateikiamos informacijos daug. Atitinka pirmą kriterijų.
- Šios srities portaluose skelbimai paprastai turi informacijos, kuri reikalauja nuolatinio atnaujinimo. Pats skelbimas laikomas objektu, juos dažniausiai sudaro detalios produkto ar paslaugos ypatybės, savybės, visa arba dažniausiai tik kai kuri dalis nuolat kinta. Pavyzdžiui, reikalinga atnaujinti automobilio, buto ar namo skelbimo kainą. Tam vykdomas nuolatinis informacijos atnaujinimo ir išsaugojimo procesas, todėl išpildomas antras kriterijus.
- Yra įvairios tematikos skelbimų portalų. Vienuose informacija vedama daugiau rankiniu būdu, kituose automatiniu. Šioje vietoje reikia papildomo tyrimo išsiaiškinti, kurios tematikos skelbimų portaluose dažniausiai informacija kaupiama ir atnaujinama automatiniu būdu. Naudojantis jau minėtais statistikos duomenų šaltiniais, išskiriame trijų tematikų populiariausius skelbimų portalus: darbo, automobilių ir nekilnojamojo turto. Juose siekiame rasti kuo daugiau įmonių, skelbiančių savo informaciją, o ne fizinių asmenų, kadangi pavienių asmenų informacijos kiekiai paprastai nedideli, įvedami ir atnaujinami rankiniu būdu, tuo tarpu įmonių informacijos paprastai daug, rankiniu būdu ją suvesti ir atnaujinti būtų daug laiko ir darbo reikalaujantis procesas, todėl mažai tikėtina, kad atliekamas. Daugiausiai įmonių, skelbiančių savo informaciją rasta nekilnojamojo turto portaluose. Todėl šios tematikos portalus laikome pagrindiniais, naudingiausiais tyrimo objektais, labiausiai atitinkančiais trečią kriterijų.

Nagrinėjamai sričiai labiausiai kriterijus atitinka skelbimų portalai, kurie priklauso vertikalių portalų rūšiai. Taigi nustatėme, kad darbe nagrinėjamos, iškeltos problemos aktualiausios, dažniausiai sprendžiamos vertikaliuose skelbimų (elektroninio verslo) portaluose, ypatingai nekilnojamojo turto portaluose.

Pastebėkime, kad horizontalios grupės portalai atitiktų visus kriterijus, tačiau tyrime nenagrinėjami, nes, kaip įsitikinsime vėliau darbe, horizontaliuose portaluose duomenų apsikeitimo problema, bent jau šiuo metu, nėra neaktuali, esančių priemonių pilnai užtenka, nėra poreikio naujiems būdams, metodams, taigi šios rūšies portalai darbe mažiausiai aktualūs.

Šis tyrimas naudingas ne tik problemų aktualumo sričiai nustatyti, bet ir tuo, kad sužinome, kurioje srityje geriausia atlikti sekančią tyrimo dalį, galime gauti daugiausiai informacijos, siekiant sužinoti kokie duomenų perdavimo būdai naudojami esamose sistemose, kodėl pasirenkamas naudojamas metodas, kokie sunkumai dažniausiai sprendžiami duomenų apsikeitimo procese. Prieš pateikiant sekančią tyrimo dalį, remiantis literatūros šaltiniais, nagrinėjama importo sistema, kuri yra svarbiausia portalo dalis, atliekanti pagrindinius veiksmus su duomenimis.

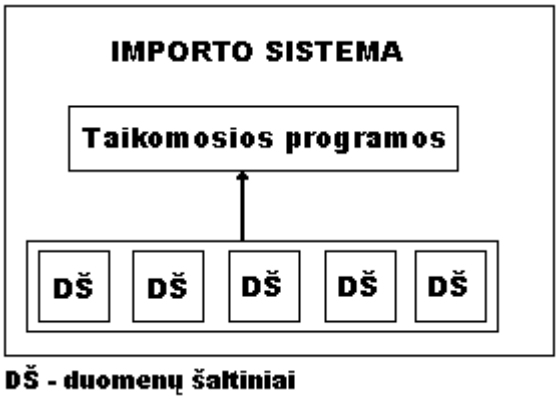
4. Importo sistema

Šioje dalyje siekiame ištirti ir išplėsti importo sistemos veikimą, skirtą duomenų gavimui iš išorinių šaltinių, kad padidinti suderinamumo galimybes su daugiau galimų duomenų šaltinių negu tai atliekama su jau naudojamomis, sukurtomis sistemomis.

Importo sistemos yra duomenų pakeitimo į informaciją priemonės ir būdai. Ji surenka, apdoroja, saugo, analizuoja ir paskirsto informaciją, turinčią konkrečią paskirtį kuriai nors veiklai. Kitaip tariant, importo sistema tiesiog apdoroja įeigą (duomenis) ir suformuluoja išeigą (veiksmų su duomenimis rezultatas).

Sistema yra efektyvi tik tuo atveju, kai joje talpinama informacija yra nepasenusi, bei nuolatos atnaujinama. Pavyzdžiui, nedidelės internetinės svetainės, su mažu kiekiu duomenų dažniausiai yra atnaujinamos rankiniu būdu, t.y. atsakingas žmogus, paprastai naudodamas turinio valdymo programas, atlieka visus pakeitimus. Portalams šis būdas nepriimtinas, nes duomenų kiekiai dideli, juos reikia greitai, nuolat atnaujinti, kad informacija būtų tiksli, teisinga. Dėl šios priežasties nagrinėjamos duomenų importo sistemos, kurios atlieka automatinius duomenų gavimo iš šaltinių bei atnaujinimo veiksmus.

Importo sistemos sudedamosios dalys pavaizduojamos schema (1 pav.).



1 pav. Importo sistemos sudedamosios dalys

Taigi importo sistema susideda iš dviejų dalių:

- Duomenų šaltiniai
- Taikomosios programos.

Duomenų šaltinių dalyje nagrinėjama duomenų perdavimo problema:

- Dažniausiai kylančios problemos, kokiu būdu šaltiniui perduoti savo duomenis portalui. Tarkime šaltinis nori, kad jo informacija patektų į portalą, tačiau neturi priemonių kaip tai padaryti. Nagrinėjami būdai, metodai kaip apsikeisti duomenimis

tarp dviejų skirtingų sistemų. Ši problema paprastai aktuali tiek portalui, tiek šaltiniams, tačiau pačių priemonių ar metodų pritaikymas atliekamas šaltinių pusėje, plačiau apie tai nagrinėjama sekančiame (4*.1 Duomenų perdavimas*) skyriuje.

Taikomųjų programų dalyje, nagrinėjama duomenų gavimo problema:

- Prarandame šaltinius, kurie turi būdą kaip perduoti informaciją, tačiau jie nėra tinkami, suprantami importo sistemai ir atmetami. Kitaip ši problema įvardinama - duomenų struktūrų nesuderinamumo problema. Yra sukurti standartai, priemonės, leidžiančios duomenims migruoti tarp įvairių sistemų, siekiame jas išnagrinėti, ištirti, pasiūlyti naujus.

4.1. Duomenų perdavimas

Šioje dalyje nagrinėjama duomenų perdavimo problema, pateikiami sprendimai kaip ją spręsti. Tarkime šaltinis nori, kad jo informacija patektų į portalą, tačiau neturi priemonių kaip tai padaryti. Todėl nagrinėjami būdai, metodai kaip apsikeisti duomenimis tarp dviejų skirtingų sistemų, jie nuolat tobulinami, ieškoma naujų alternatyvų.

Duomenys - kažkokia turima informacija. Informacija - tai objektyviai egzistuojantis pasaulio reiškinys. Tai žinios, kurias žmogus gauna, įsimena, perduoda. Abstraktesnis apibrėžimas: tai žmogaus suvoktas objekto turinys. Informacija, kuri įtakoja žmonių santykius, visuomenės gyvenimą, ji atsiskleidžia žmonių bendravime – komunikacijoje. Duomenų šaltinis - visa, iš ko galime gauti informacijos. Tokiu šaltiniu galime laikyti žmogų, knygą, televiziją, kompiuterinę laikmeną. Informacijos šaltiniai kaupiami informacijos saugyklose, tokiose kaip bibliotekos, muziejai, duomenų bazės.

Iki šiol darbe *šaltiniais* laikėme tai, iš ko gauname tam tikrą informaciją (paslaugos, prekių aprašymas, nuorodos į svetaines), iš kurios susideda portalo turinys. *Duomenų šaltiniais* laikome priemones, kurių pagalba duomenys perduodami kitai portalui.

Vienas pirmųjų priemonių kaip spręsti duomenų apsikeitimo problemą, buvo sukurtos interneto standarto paslaugos (*angl. web services*), leidžiančios duomenims migruoti tarp įvairių sistemų. Šios priemonės plačiai naudojamos tiek horizontaliuose, tiek vertikaliuose portaluose. Dar vienas būdas perduoti informaciją kitai sistemai yra duomenų pateikimas tiesiogiai iš naudojamos informacijos saugyklos, paprastai duomenų bazės. Tai mažiau naudojamas, tačiau šis būdas taip pat suteikia duomenų perdavimo galimybę.

Nagrinėjami informacijos perdavimo portalams būdai:

- Interneto standarto paslaugos (XML, SOAP, WSDL, UDDI)
- Užklausos, skirtos pateikti duomenis iš duomenų bazių (SQL)
- Interneto svetainės (HTML – tam tikra dalis priklauso ir interneto standartui)

Visų duomenų šaltinių paskirtis, atliekama funkcija yra perduoti informaciją kitai sistemai.

	Vertikalūs	Horizontalūs
Interneto standarto paslaugos	+	+
Duomenų bazių duomenų šaltinis	+	-
Interneto svetainės	-	+

2. lentelė. Naudojamos duomenų perdavimo priemonės skirtingų tipų portaluose. Lentelės duomenys pagrindžiami sekančiame (4*.1.1 Interneto standarto paslaugos*) skyriuje.

Darbe siekiama rasti daugiau galimų duomenų šaltinių portalams. Tai galima padaryti papildant jų skaičių, kur nėra naudojami, pritaikyti.

Interneto svetainės nėra naudojamos kaip duomenų šaltinis vertikaliuose portaluose (2. lentelė). Tai įgyvendinus, tikslas būtų pasiektas, tačiau tam reikia rasti priemonių kaip išskirti, paimti tik tam tikrą dalį turinio, kur yra reikalinga informacija, kitaip tariant turime rasti priemones kaip gauti nestandartinėje HTML dalyje esančiai informacijai.

Tolesnėse duomenų šaltinių dalyse siekiame:

- Plačiai išnagrinėti duomenų perdavimo priemones
- Išnagrinėti, kodėl vienoms portalų rūšims domenų šaltiniai taikomi, o kitoms ne (pvz, interneto svetainės naudojamos horizontaliuose portaluose, o vertikaliuose ne).
- Išskirti duomenų perdavimo priemonių privalumus ir trūkumus
- Išnagrinėti interneto svetainių pritaikymo galimybes duomenims perduoti vertikaliems portalams, rasti priemones kaip tai padaryti.

4.1.1. Interneto standarto paslaugos

Paslaugų standartus žiniatinkliui išleido W3C konsorciumas (*angl., World Wide Web Consortium*). Jam vadovauja Tim Berners-Lee, sukūręs URL (*Uniform Resource Locator*), HTTP (*HyperText Transfer Protocol*) ir HTML (*HyperText Markup Language*), interneto

technologinį pagrindą [WWW08]. Interneto standarto paslaugos (*angl. web services*) – priemonės, leidžiančios duomenims migruoti tarp įvairių sistemų [WS08].

Interneto standartų paslaugų priemonės:

- HTML (tik tam tikra dalis)
- XML
- SOAP
- WSDL
- UDDI.

4.1.1.1. HTML

HTML (*Hypertext Markup Language* „Hiperteksto žymėjimo kalba“) – tai kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete. Pagrindinis HTML kalbos vienetas yra elementas. HTML elementas turi vardą ir gali turėti bet kokį skaičių atributų. Elemento viduje gali būti tekstas, bei kiti elementai. Tiek tekstas, tiek ir kiti (dukteriniai) elementai paprastai gali kartotis ir sekti bet kuria tvarka.

Prie interneto standarto priemonių galime priskirti ir HTML, tačiau tik iš dalies. Svetainės puslapio turinį sudaro HTML struktūra. Standartizuota dalis, kurioje aprašomas puslapio pavadinimas, svarbiausi raktažodžiai, trumpas aprašymas, kitaip tariant tik ta dalis, kuri prasideda elementu „<html>“, baigiasi „<body>“, kur ir laikomi minėti aprašymai.

Siekiame atsakyti į klausimą, kodėl ši priemonė naudojama horizontaliuose portaluose, bet netinkama vertikaliuose.

Atsakymas paprastas - standartas pritaikytas būtent horizontaliems portalams. Iš horizontalių portalų apibrėžimo - tai paieškos, įvairių direktorijų portalai. Jų pateikiama informacija yra nuorodos, kurios veda į kitus informacijos šaltinius. Paprastai kartu su nuoroda pateikiamas ir aprašymas apie joje esančio puslapio informacijos temą, trumpas pavadinimas. Standartinėje HTML dalyje yra visa reikalinga informacija, kitaip tariant, pateikiamos informacijos pilnai užtenka horizontaliems portalams (plačiau nagrinėjama duomenų gavimo iš interneto svetainių dalyje). Vertikaliems portalams šis būdas nėra naudojamas, kadangi HTML standarto laikomasi tik tam tikroje, mažoje dalyje. Šie portalai paprastai pateikia su tam tikra veikla susijusią, specifikuotą informaciją, kuriai reikia pilno paslaugos ar prekės aprašymo, ši informacija pateikiama HTML dalyje, kurioje standarto nebesilaikoma.

HTML standarto pilnai užtenka horizontaliems portalams, todėl jiems nėra aktualūs kiti metodai, šio tipo portalai darbe probleminės srities nesudaro. Sekančiame skyriuje nagrinėjama interneto standarto paslauga, kur, priešingai nei HTML, standarto laikomasi visur.

4.1.1.2 XML

XML (*extensible Markup Language*) – tai duomenų struktūrų bei jų turinio aprašomoji kalba. W3C patvirtino XML standartą 1998 metais. Iš esmės, tai seniai pramonėje vartojamos kalbos SGML poaibis, specialiai pritaikytas naudojimui internete. Norima struktūra išgaunama naudojantis žymėmis („tags“). XML standartas nustato bendras jų sudarymo taisykles, bet neapibrėžia konkretaus žymių rinkinio ar jų reikšmės. Tai apibrėžia kiti XML šeimos standartai [WXM07].

Interneto paslaugos pateikia pasaulinio tinklo naudotojams įvairias naudingas funkcijas, besiremdamos standartiniu interneto protokolu. Dažniausiai yra naudojamas paprastas kreipties į objektus protokolas (SOAP). SOAP yra XML kalba pagrįstas standartas, apibrėžiantis sąveiką tarp paslaugų ir jų vartotojų (kitaip tariant, SOAP yra į paslaugas orientuotos architektūros protokolas).

Interneto paslaugos turi jų interfeisų pakankamai detalaus aprašymo galimybes, įgalinančias vartotojus kurti klientines taikomąsias programas. Šis aprašas paprastai yra pateikiamas vadinamame interneto paslaugų aprašymo kalbos (WSDL) dokumentu. Pagrindinis WSLD dokumento elementas yra <definitions>, kuris turi į jį įtrauktus elementus „Types“, „Message“, „PortType“, „Operation“, „Binding“, „Service“, „Port“, „Data Schema“.

Interneto paslaugos yra registruojamos taip, kad potencialūs naudotojai galėtų juos lengvai rasti. Tai yra atliekama naudojant universalų atradimo aprašymo ir integravimo (UDDI) žinyną. Geri UDDI registrų pavyzdžiai yra IBM, Microsoft ir Hewlett-Packard firmų registrai.

Lengvesniam duomenų keitimuisi tarp skirtingo tipo sistemų užtikrinti, naudojama XML ir jo pagrindu pagrįstos kitos interneto standartų paslaugos kaip SOAP, WSDL, UDDI. Visos interneto paslaugos remiasi XML, todėl šį struktūrizavimo metodą galima pavadinti „kalba kitoms kalboms aprašyti“ ir toliau naudojame ir nagrinėjame tik šią interneto paslaugų priemonę.

XML pagalba galime gauti reikiamus duomenis iš išorinių šaltinių. Tai plačiausiai, daugiausiai naudojamas metodas. Pagrindinis XML kalbos vienetas yra elementas. Elementas visada turi vardą ir, be jo, gali turėti:

- Norimą skaičių atributų. Atributas turi savo vardą bei reikšmę.
- Kitus (dukterinius) šio elemento viduje esančius elementus.
- Su elementu susijusį tekstą.

XML priemone informacijos perdavimas paprastai vyksta internetu. Sukuriama programa, kuri duomenis struktūrizuoja ir pateikia reikiamu pavidalu. Pavyzdys galėtų būti kai duomenys gaunami iš dažniausiai naudojamų informacijos saugojimo priemonių - duomenų bazių. Tuomet duomenų bazės lentelės lauko pavadinimas atitinka elemento vardą, o lauko reikšmė elementą. Rezultatai pateikiami faile, kuris patalpinamas serveryje ir pasiekiamas per www protokolą (failo

vietą apibrėžia nuoroda). Failą atidarant, kas kart sugeneruojama duomenų struktūra, programa grąžina tuo metu gautus duomenis. Jų apsaugai, tam, kad failo vietą sužinojusiam pašaliniam asmeniui nebūtų išvedami programos rezultatai, paprastai reikalaujami prisijungimo duomenys. Tik tokiu atveju, jei gaunami teisingi prisijungimo duomenys, grąžinami rezultatai. XML priemonės pagalba duomenų perdavimas rekomenduojamas didžiulę patirtį portalo kūrime turinčios kompanija kaip IBM [BU05].

XML naudojimo pagrindiniai privalumai:

- Šiuose dokumentuose informacija talpinama tam tikromis XML failams nustatytomis taisyklėmis, aiškioje struktūroje, todėl užtikrinamas lengvesnis duomenų keitimasis tarp skirtingo tipo sistemų.
- Greitas, paprastas duomenų nuskaitymas, reikalingos informacijos paėmimas.
- Lengvas prieinamumas. Tarkime atidarant failą (failui pasiekti dažniausiai naudojama nuoroda, vedanti į tam tikrą svetainės vietą serveryje), duomenys kas kart sugeneruojami iš duomenų bazės ir pateikiami XML struktūra. Prireikus failas gali būti bet kada nuskaitomas su naujausiais duomenimis.

Trūkumai:

- Suformuoti tam tikros struktūros duomenis, reikia specifinių žinių, dažnai šaltinis arba neturi tokio asmens, kuris galėtų sukurti tokį duomenų pateikimo XML struktūra galimybę arba tai yra per brangu.
- Įvairių problemų gali kilti dėl to, kad esant kažkokiai klaidai XML struktūroje, ją likviduoti gali tik asmuo, kuris ją kūrė.
- Negalime būti užtikrinti, kad gauname tikrai korektiškus, visus duomenis.

XML plačiai naudojamas kaip duomenų šaltinis importo sistemai, laikantis nustatyto jam standarto, duomenis nesunkiai galima struktūrizuoti, tam kad pritaikyti kitai sistemai. Nors šioje dalyje nagrinėjame duomenų perdavimo būdus, tačiau jau čia pasireiškia ir duomenų gavimo problemos aktualumas. Dažnai sistema pritaikyta priimti tik iš anksto numatytos, suformuotos struktūros duomenis, taigi šaltinis turi laikytis tam tikro šablono, kad importo sistema duomenis suprastų. Būtent čia dažniausiai ir pasireiškia XML minėtas pirmasis trūkumas.

Nagrinėjamas dar vienas duomenų perdavimo šaltinis – duomenų bazė.

4.1.2. Duomenų bazės šaltinis

Duomenų bazėje įrašai saugomi tam tikra tvarka, kad kompiuterinė programa galėtų jais naudotis ir atsakyti į užklausas. Dažniausiai geresniam duomenų ištraukimui ir rūšiavimui įrašai išdėstyti kaip duomenų elementų rinkiniai, eilutės. Užklausų rezultatai tampa informacija, kuria

remiantis galima perduoti duomenų bazėje esančius duomenis, tokiu būdu ši organizuotos informacijos laikymo priemonė gali būti panaudojama kaip duomenų šaltinis. Didžiausia šio būdo kaip šaltinio panaudojimo problema ta, kad duomenų gavimas vyksta tiesiogiai iš duomenų bazės duomenų gavėjo, o ne šaltinio pusėje. Tiesioginiam duomenų nuskaitymui būtini prisijungimo duomenys, kuriuos patikint, duomenų bazėje esanti informacija lieka visiškai neapsaugota, gali būti pažeista duomenų bazės struktūra, pakeisti joje esantys įrašai, trumpai tariant gali būti padaryta žala, dėl ko sutriktų duomenis naudojančios svetainės, administravimo programos ir pan. Jei prisijungimo duomenys gaunami, tuomet išnagrinėjus duomenų bazės struktūrą, išanalizavus kokie duomenys reikalingi, nesudėtinga suformuoti užklausas reikalingai informacijai gauti.

Duomenų gavimo turint prisijungimo duomenis prie duomenų bazės privalumai:

- Lengvas prieinamumas. Galime bet kada (kai įmanomas ryšys) pasiekti mums reikiamus joje saugomus duomenis.
- Nesunkus, greitas duomenų gavimas. Tereikia suformuoti užklausą, kuri iš duomenų bazės grąžintų tam tikrą, mums reikiamą informaciją.
- Duomenys talpinami tam tikru tipu apibrėžtuose laukuose, todėl rečiau reikia juos papildomai apdoroti.

Trūkumai:

- Dažnai būna nepaprastai sunku, o kartais ir beveik neįmanoma perprasti ne savo kurtos, svetimos duomenų bazės struktūrą ir paruošti duomenis perkėlimui į kitą sistemą, išauga tikimybė gauti nevisus, nekorektiškus duomenis.
- Praktiškai tik labai išskirtiniais atvejais patikimi prisijungimo duomenys prie duomenų bazės.

Aptarėme priemones, kurių pagalba sprendžiamas uždavinys, kaip duomenis perduoti kitai sistemai. Anksčiau darbe iškėlėme siekį: pritaikyti papildomą duomenų perdavimo priemonę vertikaliems portalams – interneto svetaines. Sekančioje dalyje nustatoma kaip tai pasiekti, kad siūlomas papildomas duomenų šaltinis būtų galimas.

4.1.3. Interneto svetainės šaltinis

Duomenų perdavimo dalyje apsibrėžėme ne tik išnagrinėti duomenų perdavimo priemones, išskirti jų privalumus ir trūkumus, taip pat siekiame išnagrinėti interneto svetainių pritaikymo galimybes kaip duomenų šaltinio vertikaliems portalams.

Literatūroje aprašomos importo sistemos daugiausiai naudoja XML duomenų struktūros failus kaip duomenų šaltinį, jos skiriasi tik XML nuskaitymo, duomenų apdorojimo, saugojimo principu, tačiau jei šio būdo šaltinis neturi galimybių pateikti, jis prarandamas. Sprendžiant galimų duomenų šaltinių būdus, dažnai ieškoma alternatyvų, kuomet neišeina surinkti duomenų vienu ar kitu būdu. Svetainės labai populiarios šiais laikais, todėl galėdami jas naudoti kaip duomenų šaltinį, sumažėtų prarandamų šaltinių skaičius.

Šį šaltinį jau nagrinėjome HTML dalyje, išsiaiškinome, kad vertikalūs portalai interneto svetainių kaip duomenų šaltinio nenaudoja, kadangi HTML standartas jiems nepritaikytas. Vertikalūs portalai paprastai pateikia su tam tikra veikla susijusią, specifikuotą informaciją, kuriai reikia pilno paslaugos ar prekės aprašymo, ši informacija pateikiama HTML dalyje, kurioje standarto nebesilaikoma. Kad šaltinis būtų galimas naudoti, reikia rasti būdą, priemones kaip atrinkti informaciją iš nestandartizuotos HTML turinio dalies, išskirti reikiamą informaciją, prireikus ją apdoroti, paruošti saugojimui, kad būtų tinkama importo sistemai. Radus tam priemones, interneto svetainės taptų galimu duomenų šaltiniu vertikaliems portalams. Kol kas daroma prielaida, kad šaltinis galimas, o jos patvirtinimas nagrinėjamas sekančiame duomenų gavimo skyriuje.

Interneto svetainės duomenų šaltinio privalumai:

- Duomenys pasiekiami bet kada, kai mums to reikia (išskyrus, kai sutrikusi svetainės veikla).
- Pasiekiami naujausi duomenys.
- Interneto svetainės plačiai naudojamos
- Norint pateikti šaltinį, nereikia turėti formuojamos savo duomenų XML struktūros.
- Nereikia turėti prisijungimo informacijos prie šaltinio duomenų bazės.
- Esame nepriklausomi nuo kito asmens, jo galimų klaidų ir sistemos geras darbas priklauso tik nuo mūsų pačių.

Trūkumai:

- Svetainės puslapių HTML struktūros turinys paprastai labai didelis, kuriant programas, kurios nuskaitytų duomenis iš daugelio puslapių, reiktų dirbti su didelio kiekio informacija, tai užima daug laiko, naudojami dideli kompiuterio resursai.
- Jai svetainės atnaujinamos (pakinta dizainas, ar duomenų pateikimo išdėstymas), pakinta puslapių HTML struktūra, programa nuskaitanti duomenis turi būti pritaikoma iš naujo.

Apibendrinant duomenų perdavimo dalį, išnagrinėtos duomenų perdavimo priemonės, kurių pagrindinės: XML, interneto svetainės, duomenų bazės. Horizontalių portalų poreikius pilnai tenkina HTML standartas, nėra poreikio naujiems metodams, duomenų perdavimo problema neaktuali. Iškėlėme uždavinį rasti priemones kaip atrinkti informaciją iš nestandartizuotos HTML turinio dalies, išskirti reikiamą informaciją, prireikus ją apdoroti, paruošti saugojimui, kad būtų tinkama importo sistemai ir interneto svetainių duomenų šaltinis taptų pritaikomas vertikaliems portalams.

Turint duomenų šaltinį, sekanti sąlyga, kad jame informacija būtų tinkama, suprantama kitos sistemos, priešingu atveju šaltinis prarandamas. Ši problema vadinama duomenų struktūrų nesuderinamumo problema, nagrinėjama duomenų gavimo dalyje.

4.2. Duomenų gavimas

Praeitoje dalyje išnagrinėjome galimus duomenų šaltinių būdus, priemones kaip perduoti informaciją kitai sistemai, tačiau šaltinio turėjimas nereiškia kad jis jau išsprendžia duomenų pasikeitimo problemą. Perduodama informacija turi būti suprantama ir pritaikoma kitai sistemai, kuri juos siekia nuskaityti ir išsaugoti.

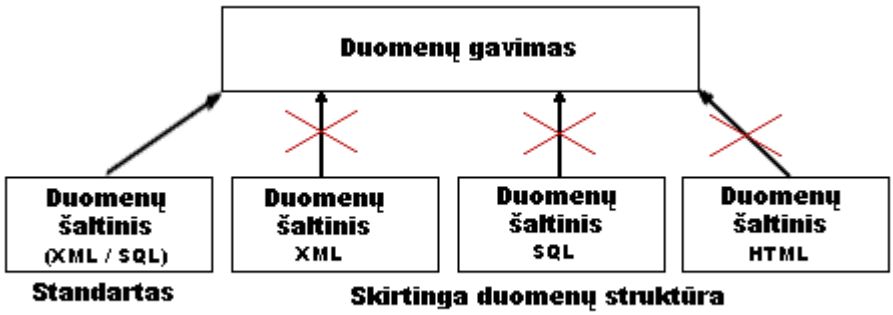
Šioje dalyje tiriamos duomenų gavimo priemonės bei metodai, išskiriami jų naudojimo privalumai ir trūkumai, nagrinėjama ir sprendžiama duomenų struktūrų nesuderinamumo problema, kuomet prarandame šaltinius, kurie turi būdą kaip perduoti informaciją, tačiau jie nėra tinkami, suprantami juos gaunamai sistemai ir atmetami.

Paprastai importo sistemos naudoja vieną duomenų struktūros standartą, pritaikytą duomenų nuskaitymui ir saugojimui. Pavyzdžiui, horizontalūs portalai naudoja HTML dalį, kur naudojamas standartas visur vienodas. Sukurta viena programa, kuri nuskaito reikiamus duomenis ir paruošia saugojimui vienam šaltiniui, tinkama ir kitiems šaltiniams. Šiame pavyzdyje duomenų šaltiniai suderinti su juos gaunančia sistema. Gerai, jei šaltinių daug, jų pilnai pakanka visai reikimai informacijai gauti, tačiau dėl standarto laikymosi vertikaliuose portaluose, daug duomenų šaltinių prarandama.

Dažnai nors ir turimas duomenų šaltinis, tačiau dėl duomenų struktūrų nesuderinamumo, jis netinkamas. Pavyzdžiui, šaltinis turi XML struktūros duomenų failą, tačiau jis neatitinka standarto duomenis gaunančios sistemos ir prarandamas. Kitas pavyzdys, jei duomenys nuskaitomi iš duomenų bazės - suformuotos užklausos gali atitikti mums reikalingą informaciją, tačiau jei duomenų bazių struktūra nevienoda, pritaikius programą duomenų nuskaitymui vienai duomenų bazei, ji netinkama kitai. Taip pat, matyt, sunkiai rastume interneto svetainę, turinčią vienodą HTML struktūrą (toje dalyje, kur nesilaikoma standarto) dėl skirtingo duomenų išdėstymo, stiliaus skirtumų ir pan., taigi vienos programos duomenų nuskaitymui iš įvairių

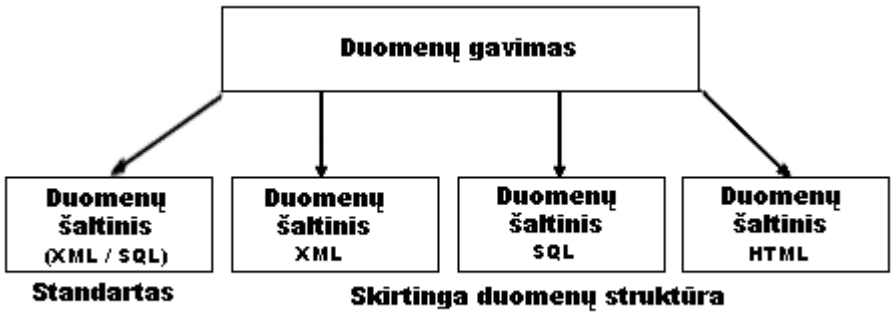
svetainių kūrimas, praktiškai neįgyvendinamas uždavinys.

Apibendrinant, šioje dalyje nagrinėjama problema, kad galimi šaltiniai tik tie, kurių duomenys specialiai paruošti kitai sistemai, tik jai suprantama, tinkama forma. Taip prarandame kitus duomenų šaltinius, kuriuose gali būti reikiama portalui informacija, tačiau jis nepritaikytas perdavimui, nesuprantamos formos, skirtingos struktūros nei reikalaujama gaunamos sistemos (2. pav.).



2. pav. Duomenų struktūrų nesuderinamumo problema. Šaltinio perduodami duomenys nėra pritaikyti juos gaunančiai sistemai, todėl netinkami.

Siekiame įgyvendinti, kad duomenis gaunanti sistema galėtų prisitaikyti prie skirtingai suformuotų duomenų šaltinių ir juos priimtų kaip tinkamus. Tai įgyvendinus, išplečiamas importo sistemos veikimas, skirtas duomenų gavimui iš išorinių šaltinių, padidinant suderinimo galimybes su daugiau galimų duomenų šaltinių (3 pav.).



3 pav. Duomenų struktūrų nesuderinamumo problemos siūlomas sprendimo būdas

Nagrinėti galimi duomenų šaltiniai: XML; SQL - duomenų bazės; HTML - interneto svetainės. Toliau nagrinėjama, kokiu būdu jie nuskaitomi, kokiomis priemonėmis, metodais, kaip gaunama iš jų informacija, kokie taikomų būdų privalumai ir trūkumai. Siekiame įrodyti, kad duomenis gaunanti sistema gali prisitaikyti prie skirtingai suformuotų duomenų šaltinių ir juos priimti saugojimui.

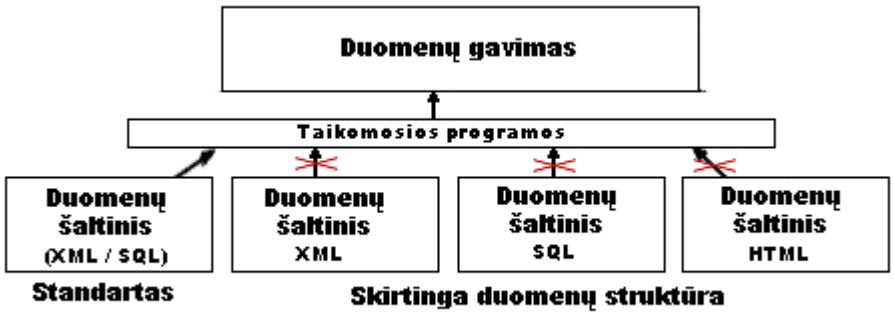
4.2.1. Taikomosios programos

Taikomąja programa laikome – kompiuterinę programą ar programų paketą, skirtą duomenų gavimui ir paruošimui saugoti iš šaltinio. Šios programos kuriamos, naudojant duomenų aprašymo ir manipuliavimo jais kalbą, kitaip vadinamą programavimo kalba. Bendru atveju, taikomosios programos skirtos atlikti tam tikras funkcijas vartotojams.

Dažnai reikalingiems veiksmams su duomenimis ir tų veiksmų sekoms atlikti, naudojami kreipiniai į vadinamąsias standartines funkcijas, bibliotekos. Yra labai daug įvairiausios paskirties funkcijų - tai matematinės funkcijos, simbolių ir tekstų apdorojimo funkcijos, datos funkcijos ir t. t. Be standartinių sudaromos ir vadinamosios vartotojo funkcijos [PKS08]. Jų pagalba kuriamos taikomosios programos duomenų gavimui ir paruošimui saugoti iš šaltinio.

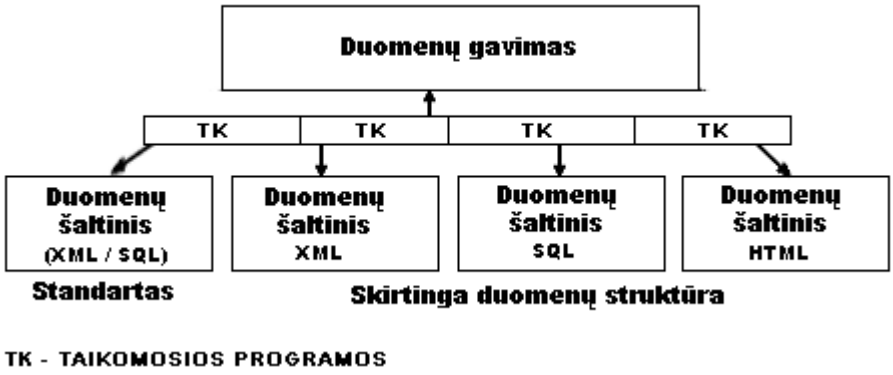
Jei importo sistema priima tik tam tikra struktūra suformuotus duomenis, laikantis tam tikro nustatyto šablono, tuomet pagal jį sukuriama viena taikomoji programa, kuri kaskart naudojama duomenų nuskaitymui iš turimų duomenų šaltinių. Tokiu atveju kitokios struktūros duomenys netinkami.

Pridėjus taikomąsias programas, duomenų struktūrų nesuderinamumo problema pavaizduota (4 pav.)



4 pav. Duomenų šaltiniai pritaikomi taikomajai programai

Problemai spręsti, siekiama kurti ne vieną taikomąją programą, kuri priimtų tik tam tikrą duomenų struktūros šabloną, o taikomąsias programas kurti atskirai kiekvienam šaltiniui. Kitaip tariant, ne šaltinių duomenys būtų pritaikomi taikomajai programai, o taikomosios programos pritaikomos prie duomenų šaltinių (5 pav.).



5 pav. Duomenų struktūrų nesuderinamumo problemos siūlomas sprendimo būdas, taikomąsias programas pritaikant prie duomenų šaltinių

Taikomosios programos kuriamos visiems galimiems duomenų šaltiniams:

- Interneto standarto paslaugos (XML),
- Duomenų bazės (SQL),
- Svetainės (HTML).

Siūlomo sprendimo atveju sugaištama daugiau laiko, kadangi programos kaskart kuriamos atskiros kiekvienam duomenų šaltiniui, sunaudojama daugiau žmogaus darbo resursų, tačiau svarbiausia, kad jis neprarandamas.

Tam, kad galėtume taikyti pateikto sprendimo būdą, išnagrinėjama, kokiu būdu taikomųjų programų pagalba nuskaitomi duomenys iš duomenų šaltinių, bei kaip pritaikomi reikiama forma saugojimui bendroje sistemoje. Kai kurios importo sistemos naudoja arba XML, arba duomenų bazės, arba interneto svetainės (horizontaliuose portaluose) duomenų šaltinį. Geriausiam būdui kaip gauti reikiamą informaciją iš įvairios struktūros duomenų šaltinių rasti, siekiame išnagrinėti esamas taikomųjų programų kūrimo praktikas, prireikus jomis pasinaudoti, kad kuriamos programos veiktų greitai ir patikimai. Daugiausiai dėmesio skiriama taikomųjų programų radimui interneto svetainių duomenų šaltiniams vertikaliems portalams, bei jų analizei.

4.2.1.1. XML

XML apibrėžėme duomenų perdavimo dalyje, todėl iš kart pereiname prie jo nuskaitymo galimybių. XML nuskaitymui galima rasti begalę literatūros, kadangi tai plačiausiai naudojamas būdas kaip duomenų šaltinis, daugiausia duomenų importo sistemų sukurtos šiuo pagrindu. Didžiausias jo privalumas: greitas, paprastas duomenų nuskaitymas, reikalingos informacijos paėmimas. XML elementai gali būti automatiškai randami pagal elemento vardą arba pagal

kelią, jei elementas priklauso kitam elementui. Šiuo metu yra daug bibliotekų dirbti su XML, nagrinėjame keletą jau sukurtų, aprašytų literatūroje bibliotekų veikimo principus:

- SAX
- DOM

SAX modelį autoriai laiko kaip “auksiniu standartu” [WS08], kol kas labiausiai išbaigtu ir tiksliausiu metodu. SAX klasėje nagrinėjamoji funkcija nuskaito visą dokumentą, patikrina ar XML suformuotas taisyklingai (visi elementai turi pradžią, pabaigą), atpažįsta elementus ir susieja turinčių kažkokią reikšmę elementų pavadinimus su jiems priklausančia reikšme.

Kitas metodas – DOM (*Document Object Model*) [WDO98]. Jo pagrindas - medis. Šis metodas veikia panašiai kaip SAX nagrinėjamoji funkcija - nuskaitomas dokumentas, patikrinama ar XML suformuotas be klaidų, atpažįstami elementai, tik vietoje to, kad grąžinamas rezultatas yra sudarytas iš mažų fragmentų, DOM metodas sudaro XML hierarchijos DOM medį, kuris turi viską kaip originalus XML dokumentas, t.y. visi elementai, komentarai, tekstinė informacija, yra patalpintos medžio objekte kaip lapai, imant viršūnę kaip patį XML dokumentą. Kai visa informacija patalpinta atmintyje, reikiamus duomenis galime išrinkti, keisti.

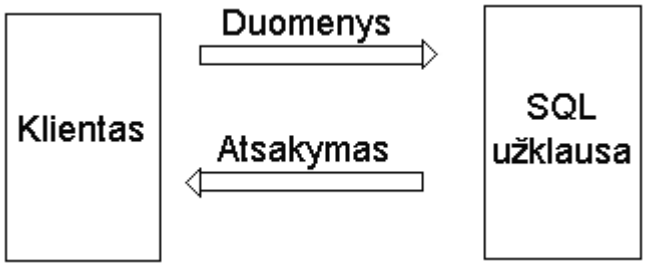
Pateikta keletas priemonių, jas galima naudoti kuriant taikomąsias programas XML duomenų šaltiniui, tačiau minėtos bibliotekos dažniausiai atlieka daug mums nereikalingų tyrimų, veiksmų, tiksliau kurie nereikalingi sprendžiamoms užduotims pasiekti, tad sunaudojama daugiau procesams įvykti reikalingo laiko. XML - tekstinė informacija, suformuota tam tikra struktūra, laikantis interneto standarto nustatytų taisyklių. Darbui su tekstine informacija programavimo kalbos turi standartinių, taip pat tam sukurta daug papildomų funkcijų, kurių pagalba galima atpažinti XML elementus ir susieti turinčių kažkokią reikšmę elementų pavadinimus su jiems priklausančia reikšme. Taigi, greičiausiam veikimui, naudojant standartines, vartotojo sukurtas funkcijas, pasirenkame kurti taikomąsias programas, nesinaudojant bibliotekomis. Tokiu būdu programos veikimo greitis, patikimumas priklauso tik nuo mūsų pačių programavimo specifikos, galima užtikrinti, kad atliekami tik reikalingi veiksmai duomenų nuskaitymui ir paruošimui saugoti.

4.2.1.2. Duomenų bazės

Siekiama pritaikyti taikomąsias programas duomenų nuskaitymui iš duomenų bazės. Tam formuojamos užklausos, pagal kurias grąžinami reikiami duomenys, kitaip tariant naudojamos į duomenų bazę orientuotos programos. Visų šių programų veikimo principas toks pat visoms duomenų bazėms.

Jei duomenų bazės struktūra yra sudėtinga, siekiant paruošti duomenis perkėlimui į kitą

sistemą, sudėtingėja ir užklausų formavimas, taip pat nuo užklausų sudarymo specifikos, priklauso ir duomenų gavimo greitis, todėl reikia gerai žinoti duomenų bazės, iš kurios imami duomenys, užklausų programavimo kalbą. Dažniausiai tai nesukelia problemos, kadangi praktiškai visos šiuolaikinės duomenų bazės naudoja SQL užklausų programavimo kalbą. Tiesa, skirtingose duomenų bazėse gali skirtis SQL užklausų sudarymo specifika, tačiau principas lieka toks pat, nesunku prisitaikyti.



6 pav. Į duomenų bazę orientuotos taikomosios programos

Taigi, informacijos išgavimui iš nutolusios duomenų bazės, kuriamos į duomenų bazę orientuotos taikomosios programos (6 pav.)

4.2.1.3. Interneto svetainės

Šioje dalyje nagrinėjamos duomenų gavimo galimybės, metodai iš interneto svetainių, siekiant išsiaiškinti jų pritaikomumą reikiamos informacijos gavimui iš HTML dalies, kurioje nesilaikoma interneto paslaugų standarto. Kadangi svetainių duomenų šaltinis nėra taikomas vertikaliems portalams, reikalinga atrasti papildomas priemones kaip iš jų gauti reikiamą informaciją.

Svetainės puslapių turinį sudaro HTML struktūra. Duomenų nuskaitymui iš HTML naudojamos interneto naršymo programos, kurios automatizuotai atlieka tam tikrus veiksmus, kuriuos atliktų su kompiuteriu dirbantis žmogus [PSM04].

Interneto naršymo programos (*angl. Web crawler*) - viena pagrindinių sudedamųjų paieškos sistemų (minėtų apibrėžiant horizontalius portalus) dalių. Trumpas veikimo aprašymas: surenkamas sąrašas nuorodų, kurias reikia aplankyti, jas aplankius, identifikuojamos kitos nuorodos puslapiuose ir pridedamos prie lankytinų nuorodų sąrašo. Vėliau tam tikra tvarka nuorodos iš to sąrašo rekursiškai aplankomos, surenkama reikiama informacija. Minėtas interneto naršymo programas galima laikyti taikomosiomis programomis, kurios naudojamos horizontaliuose portaluose. Jos pritaikytos HTML standartizuotai daliai, kitaip tariant naudojama viena taikomoji programa visiems šaltiniams. Turbūt sunkiai rastume interneto svetainę, turinčią vienodą HTML struktūrą toje dalyje, kur nesilaikoma standarto, taigi vienos programos duomenų

nuskaitymui iš įvairių svetainių kūrimas, praktiškai neįgyvendinamas uždavinys. Tiesa, naujausios paieškos sistemų interneto naršymo programos, kažkokiu būdu atrenka reikalingus duomenis ir iš nestandartinės HTML dalies, tačiau jų veikimo principas griežtai saugomas kaip komercinė paslaptis, literatūros šaltinių apie tai nepateikiama, todėl mūsų rastas būdas galėtų būti pavyzdys, paaiškinantis tokių programų veikimo principą.

Pagrindinis šios dalies siekis - rasti būdą, kaip surinkti duomenis iš interneto svetainių vertikaliems portalams, kur reikiama informacija yra nestandartinėje HTML dalyje.

Duomenų gavimas iš svetainių nagrinėjamas ir atliekamas dviem etapais:

1. Detaliai išnagrinėjamas interneto naršymo programų veikimo principas
2. Randamas būdas, kaip surinkti reikiamą informaciją iš nestandartinės HTML dalies

Svetainė susideda iš nuorodų, informaciją galima gauti nuskaičius jų turinį, kuriame pateikiama HTML struktūra suformuoti reikalingi duomenys. Toliau nagrinėjama, kokie svetainių duomenys yra reikalingi.

Anksčiau darbe nustatyta, kad aktualiausia darbe nagrinėjamoji sritis yra vertikaliems priklausantys (nekilnojamojo turto) skelbimų portalai. Skelbimus galime laikyti objektais, kuriuos norime gauti iš interneto puslapių. Objektus sudaro sudėtinė informacija, sudaryta iš juos apibūdinančių požymių. Pavyzdžiui, nekilnojamojo turto objektas gali būti butas, kurio požymiai yra jo plotas, kaina, vietovė ir pan. Objektai, kartu su požymiais, laikomi reikalingais duomenimis, kuriuos siekiame surinkti iš interneto svetainių ir paruošti saugojimui bendroje duomenų saugykloje. (Duomenų paruošimas saugojimui, nagrinėjamas vėliau darbe). Toliau nagrinėjame, kaip interneto naršymo programų pagalba, pasiekiami reikalingi duomenys.

Minėtos interneto naršymo programos yra svarbi paieškos sistemų dalis, kurių visas veikimo principas slepiamas kaip komercinė paslaptis, kad apsaugoti nuo programų kopijavimo. Nemokamos versijos turi mažiau galimybių nei komerciniai atitikmenys, tačiau pagrindiniai architektūros modeliai, kurie svarbūs detaliam interneto naršymo programų išnagrinėjimui, yra panašūs.

Remiantis literatūros šaltiniu [BC02], pagrindiniai interneto naršymo programų architektūros modeliai yra keturi:

- Surinkimo
- Prioriteto
- Prisijungimo
- Perdavimo

Toliau atskirai nagrinėjame kiekvieno modelio paskirtį.

Surinkimo modelio pagalba, pridedamos nuorodos į specialiai suplanuotą greitam naudojimui struktūrą ir nusprendžiama ar interneto puslapis jau matytas anksčiau, kitaip tariant, sudaromas unikalių nuorodų sąrašas, kurį naudoja sekantis prioriteto modelis.

Prioriteto - šis modelis žiūri kiekvieno lankytino puslapio (nuorodos) naujumą ir nusprendžia, kurie puslapiai turi būti naršomi dabar, o kurie vėliau. Jei puslapių labai daug, sudarant tam tikrą pirmumo eilę, sutaupomi serverio resursai. Prioritetams sudaryti, pagal tam tikras formules, skaičiuojami specialūs dydžiai. Nustatoma riba, kurios neperžengę puslapiai atnaujinami, kiti - ne. Realizavimo būdas: gerumo parametrą pažymime *score*, tai vertė tarp 0 ir 1. Kuo aukštesnė vertė, tuo puslapis įvertinamas kaip geresnis. Gerumas apibendrina meta-duomenis apie puslapį vienu dydžiu. Šis dydis priklauso nuo įvairių puslapio savybių, pavyzdžiui, ar prisijungimas greitas, tekstinė informacija gali būti lengvai kategorizuojama ir pan. Kitas naudojamas parametras, tai naujumo parametras, pažymėkime *freshness*. Kuomet aplankome puslapį, jo vertė lygi 1, mažėja, jei puslapis neaplankomas t.y. praleidžiamas. Formulė puslapio prioritetui (priority) apskaičiuoti: priority = score * (1 - freshness). Nustačius atnaujinamus puslapius, jie naudojami prisijungimo modelyje.

Prisijungimo - šis modelis atidaro internetinius puslapius ir gali nustatyti prisijungimo uždelsimą. Paprastai nuskaitant kiekvieną puslapį, naudojamas bent 15 sekundžių uždelsimas. Tai reikalinga, kad nebūtų naudojama pernelyg daug resursų ir išvengiama didžiulio interneto puslapių apkrovimo. Toliau seka perdavimo modulis.

Perdavimo - šis modelis nuskaito internetinių puslapių turinį, surenka reikiamą informaciją, taip pat randa naujas nuorodas, kurias nusiunčia į surinkimo modulį. Šioje vietoje modelis panaudojamas tik iš dalies, kadangi paieškos sistemos priklauso horizontaliems portalams, reikiamos informacijos surinkimas čia vykdomas tik iš standartinės HTML dalies.

Minėti modeliai padeda apibrėžti duomenų gavimo principą, būdus kaip sutaupyti kompiuterio resursus, nuskaitant svetainių duomenis, tačiau neišsprendžia duomenų gavimo iš nestandartinės HTML dalies problemos.

Toliau nagrinėjamas antras duomenų gavimo iš interneto svetainių etapas, siekiama papildyti perdavimo modulį papildomomis savybėmis, rasti priemones informacijos surinkimui ne tik iš standartinės HTML dalies.

HTML galime rasti daug panašumų su XML. Pagrindinis HTML kalbos vienetas yra elementas. Kaip ir XML, HTML elementas turi vardą ir gali turėti bet kokį skaičių atributų. Elemento viduje gali būti tekstas bei kiti elementai. Tiek tekstas, tiek ir dukteriniai elementai paprastai gali kartotis ir sekti bet kuria tvarka. XML turi savybę dokumentų turinį vaizduoti

įvairiais formatais, galime ją panaudoti XML generuoti į HTML, taigi galime ir atvirkščiai. Turint XML, duomenų gavimui pasirinktas būdas XML taikomųjų programų dalyje. Būtų patogu šį būdą taikyti, tačiau susiduriame su rimta problema: beveik visada interneto svetainės neišlaiko validumo testo (pvz., validumą galima pasitikrinti, svetainėje „http://validator.w3.org/“, tiesiog nurodant puslapio internetinį adresą). Tai reiškia, kad savo struktūroje puslapiai turi klaidų, pavyzdžiui, neuždarytas HTML elementas, jų kai kur trūksta ar per daug. Tokiu atveju generavimas iš HTML į XML negalimas, todėl šis būdas atkrenta.

Papratai svetainės puslapio turinį sudaro labai didelio kiekio informacija, o darbas su tokia informacija yra sudėtingas ir daug kompiuterio resursų reikalaujantis uždavinys, reikalingi įvairūs duomenų išskyrimo, paėmimo tik reikiamos dalies veiksmai, standartinių programavimo kalbos funkcijų neužtenka, reikalinga kurti papildomas vartotojo funkcijas, dažnai naudotis reguliariais reiškiniais informacijos radimui ir apdorojimui. XML atveju greičiausiam veikimui, pasirinktas būdas taikomąsias programas kurti naudojant programavimo kalbos standartines ir vartotojo sukurtas funkcijas, nesinaudojant bibliotekomis. Šis būdas pritaikomas ir duomenų gavimui iš HTML struktūros duomenų, taigi pasirenkame naudoti atskirai kuriamas interneto naršymo programas, kurioms nurodžius kokių duomenų mums reikia, prisijungiama prie reikiamų puslapių, surenkama reikiama informacija ir paruošiama saugojimui. Šiam tikslui naudojamos standartinės programavimo kalbos funkcijos, neapsieinama ir be vartotojo funkcijų. Pavyzdžiui, funkcija, kuriai nurodžius kažkokio teksto dalies pradžią, pabaigą, gaunama tarpinė dalis tarp jų. Taigi, jeigu tekstas yra tarp kažkokių HTML elementų, nurodžius pirmą ir antrą elementus, funkcija grąžina tarp jų esančią informaciją, kurią dažnai reikia papildomai apdoroti, (tai nagrinėjama *4.3 Duomenų apdorojimas* skyriuje).

Pasirinkta kurti atskirai kuriamas interneto naršymo programas, tokiu būdu programos veikimo greitis, patikimumas, kaip ir XML atveju, priklauso tik nuo mūsų pačių programavimo specifikos, galima užtikrinti, kad atliekami tik reikalingi veiksmai duomenų nuskaitymui ir paruošimui saugoti.

Kadangi taikomųjų programų kūrimas svetainės duomenų šaltiniui vertikaliems portalams metodas naujas, reikalingas papildomas nagrinėjimas, kokiais atvejais kyla sunkumų jas kuriant:

- Dažnai atidarant svetainės puslapius, jiems siunčiami įvairūs parametrai (kintamieji), nuo kurių priklauso išvedami duomenys. Jei parametrai nurodomi nuorodoje, šis metodas vadinamas „GET“. Tokiu atveju problemos nėra, nes reikiamus parametrus galime nurodyti, perduoti nuskaitant nuorodą ir gauti reikiamus duomemis. Kintamieji gali būti perduodami „POST“ metodu. Šiuo atveju parametrai siunčiami serveriui atskirai nuo adreso. Jei žinome kokie parametrai perduodami POST metodu,

juos nusiųsti galime pasinaudojant papildomomis funkcijomis (dažniausiai naudojant „curl“ funkcijas), kitu atveju šaltinį prarandame.

- Javascript programavimo kalbos pagrindu sukurtos svetainės. Java Script yra pilnai perduodamas kliento mašinai (kompiuteriui) ir atliekamas klientinėje dalyje. Paprastai JavaScript kalbos kodas įtraukiamas į HTML (*Hypertext Markup Language* „Hiperteksto žymėjimo kalba“ – kompiuterinė žymėjimo kalba, naudojama pateikti turinį internete) puslapius, tokiu būdu išplečiant statinius HTML puslapius dinaminiu skripto funkcionalumu – galimas anketų parametrų tikrinimas, naujų langų atidarymas, suskleidžiamos hierarchinės struktūros rodymas, išsiskleidžiantis meniu ir daug kitų interaktyvumo formų. Tarkime Javascript funkcija, įtraukta į HTML kodą, atidarant puslapį, nurodytoje vietoje suformuoja tam tikrą HTML dalį su portalui reikiama informacija. Šios dalies nuskaičius turinį nematysime, kadangi tai atlikta klientinėje pusėje. Tačiau tam, kad klientinėje pusėje būtų išvedami kažkokie duomenys, jie jau turi būti laikomi atmintyje (užkrauti) atidarius nuorodą, kad juos galėtų naudoti funkcija, taigi jie yra, radus kurioje vietoje, galime nuskaityti.
- Viena pagrindinių svetainių kūrimo metodologijų, kuri remiasi JavaScript kalba – AJAX. Tradicinių svetainių interaktyvumas kuriamas kaskart kreipiantis į serverį. Pavyzdžiui, užpildžius anketą (ar formą), kreipiamasi į serverį ir užkraunamas naujas puslapis. Tokiu būdu bereikalingai iššvaistomi resursai, nes didelė dalis informacijos nepakinta ir siunčiama kaskart iš naujo. Taip pat tokiu būdu neįmanoma pasiekti tokio interaktyvumo, kokį gali suteikti ne internetinės aplikacijos. AJAX svetainės gali siųsti užklausas serveriui, gauti atsakymą tam tikra apsibrėžta forma, bei naudojant JavaScript programavimą atnaujinti tik reikiamą puslapio dalį. Tokiu būdu sutaupomi tinklo resursai, nes žymiai sumažinami duomenų srautai. Taip pat taupomi ir serverio procesoriaus resursai. Nuskaičius nuorodą, tos dalies pasikeitimo iškviesti negalime, vėlgi dėl to, kad esame klientinėje pusėje, o joje gali ir būti pateikiama mums reikalinga informacija. Iš pradžių tai atrodo labai rimta problema, dėl sparčiai populiarėjančio Ajax naudojimo, galėtume prarasti vis daugiau svetainių, tinkančių mūsų sistemai kaip duomenų šaltinis.

  Rastas sprendimas išnagrinėjus detaliau AJAX veikimo principą. Trumpai jį galime aprašyti taip: funkcija atidaro failą, perduodama jam GET ar POST metodu reikiamus parametrus (kintamuosius), imituoja jo būsenos pasikeitimą, koks būna užkraunant puslapį, taip faile atliekami užprogramuoti veiksmai, pavyzdžiui, iš duomenų bazės pagal gautus parametrus išrenkami ir grąžinami duomenys. Šie rezultatai išvedami funkcijoje nurodytoje, dažniausiai kažkokioje HTML elemento dalyje pažymėjus jį

„id“, vietoje. Atskirai perduodant funkcijos atidaromam failui reikiamus parametrus, rezultatus galime išsivesti jį atsidarius kaip atskirą svetainės puslapį ir nuskaičius jo turinį. Tuomet grąžinamas HTML turinys su reikiama informacija. Taigi sparčiai populiarėjantis Ajax taikymas šaltiniui nepakenks, duomenis vis tiek galime gauti, tik tai užima šiek tiek daugiau laiko. Tačiau svarbiausia šaltinio neprarandame, papildomas laiko sugaišimas minimalus.

- Svetainės, kurios naudoja Flash programas. Jei duomenys formuojami ir pateikiami Flash aplikacijose, tuomet turinyje jų niekaip nematysime ir negausime. Tačiau dažnai tokių aplikacijų naudojama duomenų bazė - XML failas. Svetainėje aplikacijos įdėjimo kode XML šaltinio vieta nurodyta, taigi duomenis galime gauti iš ten. Kai kurios aplikacijos pateikia duomenis gautus GET metodu, tad šiuos duomenis turinyje taip pat rasime. Jei Flash aplikacijos naudoja savo vidinę duomenų bazę, tokiu atveju šaltinis atkrenta.

Apibendrinant, svetainės duomenų šaltiniai prarandami, kai jie sukurti flash principu su vidine duomenų baze arba svetainės puslapio reikiamiems duomenims išvesti naudojami nežinomi POST metodu perduodami kintamieji.

Išskiriame privalumus atskirai kuriamų interneto naršymo programų:

- Kuriant savo taikomąsias programas, nesinaudojant jau sukurtomis priemonėmis, bibliotekomis, galime užtikrinti, kad atliekami veiksmai tik tie, kurie reikalingi, nėra vykdomi papildomi tikrinimai, tyrimai dėl ko sugaištama daugiau laiko procesams įvykti, resursų ir laiko dydis priklauso tik nuo mūsų pačių gerai atlikto darbo;
- Programa pritaikoma prie bet kokios HTML struktūros turinio. Kadangi nurodoma kuriuos duomenis paimti, iš kurios vietos, nėra svarbu ar HTML sudaryta be klaidų, kitaip tariant praeina validumo testus.

Taikomųjų programų paskirtis duomenų šaltiniuose:

1. duomenų gavimas
2. paruošimas saugoti (duomenų apdorojimas)

Taigi**,** duomenims gauti naudojame programavimo kalbos turimas standartines ir papildomas (vartotojo) funkcijas XML ir HTML duomenų šaltiniams, duomenų bazės šaltiniui naudojamas į duomenų bazes orientuotos taikomosios programos, pagrįstos užklausų formavimu. Sekančioje dalyje nagrinėjama antroji taikomųjų programų paskirtis.

4.3. Duomenų apdorojimas

Gaunamiems duomenims dažnai reikia papildomo apdorojimo, tokio kaip nereikalingos informacijos pašalinimas, pavertimas tam tikro tipo duomenimis, kad bendroje sistemoje informacija būtų laikoma tvarkinga, vėliau būtų galima atlikti reikalingus išskyrimo, apjungimo, grupavimo ar kitus reikalingus veiksmus, tenkinančius jiems keliamus reikalavimus išvedant, pateikiant. Duomenų apdorojimas - atliekamas reguliarių reiškinių pagalba, tai pat jie naudojami sunkiai tiksliai nusakomų šablonų aprašymui ir jų paieškai tekste.

Knygoje *„Automata Theory, Languages and computation“* [HMU06] reguliarūs reiškiniai traktuoti kaip programavimo kalba, naudojama tekstų paieškai, kompiliatorių konstravimui ir kituose taikymuose. Reguliarūs reiškiniai suteikia būdą, kaip deklaratyviai išreikšti kalbos eilutes, todėl naudojami kaip įėjimo kalba daugelyje sistemų, dirbančių su eilutėmis, pavyzdžiui, interneto naršyklių bei teksto redaktorių programų paieškos komandose ieškomiems šablonams (angl. *pattern*.) išreikšti. Šablonais (angl., *patterns*) vadiname teksto fragmentus (kitaip sakant, ženklų eilutes). Išties, įvairiose paieškos sistemose ieškomą šabloną aprašo reguliarūs reiškiniai. Bendru atveju, galime apibrėžti reguliarius reiškinius kaip tarpininką tarp teksto pavyzdžių, kuriuos norime rasti ir paieškos programų [Dič07]. Nurodžius reguliarų reiškinį, jis būtų konvertuojamas, gaunama programa, atliekanti reikiamą paiešką. Knygoje reguliarūs reiškiniai su šablonais siejami taip: reguliarių reiškinių technologiją galima panaudoti sunkiai tiksliai nusakomų šablonų aprašymui ir jų paieškai tekste.

Privalumas: jei informacijos paieška ar apdorojimas suprogramuotas naudojantis standartinėmis ar vartotojo sukurtomis funkcijomis, bet nesinaudojant reguliariais reiškiniais, tai programos kodo taisymas užtrunka ilgiau. Taigi, reguliarūs reiškiniai išties patogus būdas tekste atrinkti, palikti reikiamą informaciją, ją reikiamai struktūrizuoti.

Nuskaitymas iš duomenų šaltinio turi būti kiek galima greitesnis, sunaudoti mažiau resursų, kadangi esant dideliam šaltinių skaičiui, procesui skirtas laikas gali smarkiai išaugti ir būti per ilgas, kad nuolat duomenis atnaujinti, laikyti naujausius. Tam svarbu pasirinkti tinkamą programavimo kalbą, kuri turėtų pakankamai galimybių realizuoti pasirinktus taikomųjų programų kūrimo būdus.

4.4. Programavimo kalbos

Svarbiausi reikalavimai, kad minėtos programavimo kalbos turėtų tokias savybes, paslaugas, reikalingas taikomosioms programoms kurti:

- Technologijos, įgalinančios kurti web programas
- Plačios galimybės atlikti įvairius manipuliavimo veiksmus su tekstine informacija
- Galėjimas iškviesti kitas programas tam tikrų užduočių atlikimui, pavyzdžiui, prieiti

prie duomenų bazės.

- Daugiaplatforminė aplinka (Papildomas privalumas)
- Veikia su dauguma interneto serverių (Papildomas privalumas)

Internetui skirtos programavimo kalbos, tokios kaip [PKS08]:

- **PHP** (*PHP Hypertext Preprocessor*) – plačiai paplitusi dinaminė interpretuojama programavimo kalba, sukurta 1997 m. ir specialiai pritaikyta interneto svetainių kūrimui, nors šiuo metu su ja galima programuoti ne tik internetui. Į HTML puslapį galima įterpti PHP kodą, kuris bus vykdomas kiekvieną kartą aplankant tą puslapį. PHP kodas interpretuojamas Web-serverio ir generuoja HTML arba kitokį išvedimą, matomą vartotojui. PHP kalba yra atviro kodo ir tai yra viena priežasčių, dėl ko kalba yra nors ir nesudėtinga, bet gana lanksti – veikia daugumoje operacinių sistemų, palaiko nemažai reliacinių duomenų bazių bei veikia su dauguma interneto serverių – Apache, CGI, FastCGI, ISAPI ir kitais protokolais. Jį galima laisvai keisti, naudoti ir platinti kitoms organizacijoms ar vartotojams. Dabartiniu metu naujausia versija yra PHP5. PHP sintaksė panaši į daugelį struktūrinių kalbų, ypatingai į C bei Perl, todėl žinant šias kalbas, nėra sunku pradėti programuoti PHP. Nors ir PHP yra dažniausiai naudojama interneto puslapių kūrimui, PHP yra labai galingas įrankis atlikti kitas funkcijas komandinėje eilutėje.
- Programavimo kalba **JAVA**, sukurta 1991 metais Sun Microsystems inžinierių. Java yra objektiškai orientuota, ji beveik nepriklausoma nuo naudojamos platformos. Savyje kalba turi priemones ir bibliotekas komunikacijai tinklu, kalba turi būti suprojektuota taip, kad kodas iš nutolusio šaltinio būtų vykdomas saugiai. Vienos iš dalių, sudarančių Java platformą: J2SE – šioje platformoje yra pateikiamos visos bazinės bibliotekos ir įrankiai, kurie naudojami komandinės eilutės ir vizualių programų kūrimui; J2EE – ši dalis skirta informacinių verslo sistemų kūrimui. Prie J2SE yra pridedamos įvairios technologijos, įgalinančios kurti interneto programas. Sistema tarpinį kodą paprastai prieš vykdydamos kompiliuoja, todėl vykdymo greitis panašus ar tik nežymiai mažesnis. Java nesunku jungti su esančiomis C, C++ ar FORTRAN bibliotekomis. Dažniausiai to prireikia jei būtina naudoti šiomis kalbomis parašytas matematines ar kitokias bibliotekas. Šiuo metu esama tiek komercinių, tiek ir atviro. Atviro kodo modelis, pateikia ne visas java programose esančias galimybes.
- **ColdFusion** taikomoji programinė įranga, naudojama kurti kompiuterinių programų pagrindus ir dinamines interneto svetaines. Šiuo atžvilgiu, ColdFusion yra panašus produktas į ASP.NET, Java Enterprise Edition ar PHP. Pirminis skiriamasis

ColdFusion bruožas yra asocijuota skripto kalba - ColdFusion Markup Language (žymima CFML), kuri lyginama su JSP, ASP.NET, ar PHP ir savo sintakse yra panaši į HTML. ColdFusion dažniausiai naudojama duomenų perdavimui interneto svetainėse ar intranetu, tačiau taip pat gali būti naudojama sugeneruoti nuotolinį valdymą, tokį kaip SOAP, svetainių priežiūrai ar Flash valdymui. ColdFusion užtikrina kelias paslaugas: transformacija iš HTML į PDF ir FlashPaper; platforma - nepriklausoma duomenų bazė, pateikianti užklausas per ODBC; sesijos, kliento ir aplikacijų valdymas; XML nagrinėjimas, užklausimas ir patvirtinimas; Serverių grupavimas; Užduočių planavimas. Kitos CFML priemonės siūlomos panašaus ar išplėsto funkcionalumo, tokios kaip .NET aplinka ar paveikslėlių manipuliacija.

- **ASP** (*Active Server Pages – Aktyvūs serverio tinklalapiai*) technologija buvo sukurta Microsoft kompanijos. Ji pagrinde yra naudojama IIS (Internet Information Server) serveryje Web-aplikacijų aptarnavimui, galima naudoti Apache. ASP palaiko keletą programavimo kalbų, tokių kaip: VBScript, JavaScript, C#, C, labiausiai naudojama yra VBScript. ASP scenarijai – tai HTML dokumentai su specialiais „tagais“ (tagas – HTML elementas), kurie interpretuojami serverio kaip vykdomasis kodas. ASP puslapiai gali iškviesti kitas programas tam tikrų užduočių atlikimui, pavyzdžiui, prieiti prie duomenų bazės. Šiuo metu Microsoft išleido naujos kartos ASP patobulinamą, kuris įeina į .NET produktų šeimą.
- **PERL**. Ši kalba paveldėjo labai daug kitų programavimo kalbų savybių. Perl kalba buvo pradėta kurti tekstinių failų apdorojimui, lengvam reikalingos informacijos paėmimui ir vizualiam pateikimui, kaip įrankis, galintis pilniau ir patogiau atlikti sh, awk, sed ir kitų UNIX įrankių darbus. Vėliau išsivystė į pilnateisę bendros paskirties programavimo kalbą su išvystytomis teksto ir masyvų apdorojimo galimybėmis. Kalba lengva naudoti, efektyvi, kompaktiška. Sintaksė primena C kalbos sintaksę. Nuo 5 versijos Perl visados įrašoma į visas Linux distribucijas kaip standartinė skriptinimo kalba administravimo užduotis spręsti. Perl kalba buvo parašyta didžioji dalis pirmųjų dinaminių interneto svetainių kodo, kai dinaminiam turiniui kurti populiariausia buvo CGI technologija. Vėliau šios pozicijos buvo užleistos labiau specializuotai PHP. Perl (Perlscript), panašiai kaip ir Javascript, galima panaudoti skriptų rašymui, kurie interpretuojami vartotojo naršyklėje.
- **PYTHON** yra interpretuojama, interaktyvi programavimo kalba, kuriama kaip atviro kodo projektas. Python yra daugiaparadigmė programavimo kalba – ji leidžia naudoti keletą programavimo stilių: objektinį, struktūrinį, funkcinį, aspektinį. Python naudoja dinaminį tipų tikrinimą. Privalumai: Dėl indentacijos, galimas praktiškai vienintelis

būdas (skiriasi tik tarpų/tabuliacijos ženklų vartojimas) parašyti kodą, todėl lengva dirbti grupėse; Kodas gali būti kompiliuojamas į vidinę formą, kas leidžia greičiau įkrauti daug kartų naudojamus modulius ir pan.; Greitas programuotojų darbo ciklas; Multi platforminė (tinka Win/Lin/Mac OS).

Visos minėtos programavimo kalbos skirtos internetiniams projektams kurti. Svarbiausia kokios programavimo kalbos labiausiai tinka taikomosioms programoms, kurios skirtos nuskaityti duomenis iš duomenų šaltinio ir prireikus juos papildomai apdoroti saugojimui, kurti.

Anksčiau sudarytus kriterijus atitinka visos programavimo kalbos, galima tik išskirti, kad labiausiai orientuotos, pritaikytos dinaminių interneto projektų kūrimui laikomos PHP ir ColdFusion, tačiau šis išskyrimas pasirinkimui nėra naudingas, todėl reikalinga išplėsti reikalavimus.

Papildomi kriterijai:

- Atviro kodo
- Daugiausiai naudojamas

Atviro kodo programavimo kalbos turi didžiulį privalumą, nes jos nemokamos, tačiau pasirinkimui ne mažesnis prioritetas skiriamas ir kalbos paplitimui, kadangi prireikus naujam programuotojui perprasti taikomųjų programų specifiką, būtų didelė tikimybė, kad jis tą kalbą mokės, greičiau perpras, nereiks išmokti naujos.

Programavimo bendruomenės „TIOBE“ svetainėje [TPC08], pateikiamas programavimo kalbų naudojimo populiarumas. Duomenys pateikiami remiantis didžiausių paieškos sistemų (Google, MSN, Yahoo, YouTube) skaičiavimais. Remiantis minėtos programavimo bendruomenės pateikiamais duomenimis, iš nagrinėjamų programavimo kalbų, labiausiai naudojamos: pirmoje vietoje JAVA, antroje – PHP ir trečioje - Perl .

Į atviro kodo kategoriją patenka PHP, Python, JAVA (3 lentelė).

Atviro kodo	Labiausiai naudojamos
PHP	JAVA
Python	**PHP**
JAVA	Perl

3 lentelė. Papildomi kriterijai, renkantis programavimo kalbą taikomosioms programoms.

Sudarius papildomų kriterijų 3 lentelę, abu kriterijus atitinka PHP ir JAVA. Java atviro kodo modeliai pateikia ne visas šiose programose esančias galimybes, skirtingai nei PHP, kuris yra pilnai atviro kodo, todėl prioritetas skiriamas šiai kalbai.

Išnagrinėti papildomi PHP privalumai, lyginant su kitomis kalbomis:

- Norint pradėti programuoti ASP kalba, pirmiausiai reikia įsisavinti objektinio programavimo metodiką. ASP.NET - tai viena geriausių platformų internetiniams projektams realizuoti, tačiau serverių pasiūla tokiems projektams talpinti maža, kainos pakankamai didelės. Tai viena iš pagrindinių priežasčių, kodėl PHP gerokai populiaresnis nei ASP.NET.
- PHP sintaksė panaši į daugelį struktūrinių kalbų, ypatingai į C bei Perl, todėl žinant šias kalbas, nėra sunku pradėti programuoti PHP.
- Perl kalba buvo parašyta didžioji dalis pirmųjų dinaminių interneto svetainių kodo, kai dinaminiam turiniui kurti populiariausia buvo CGI technologija. Vėliau šios pozicijos buvo užleistos labiau specializuotai PHP.

Dėl minėtų svarbių kriterijų atitikimo ir papildomų privalumų lyginant su kitomis programavimo kalbomis, taikomosioms programoms kurti, pasirenkame PHP programavimo kalbą.

Taigi, duomenų gavimo dalyje iškėlėme siekį įgyvendinti, kad duomenis gaunanti sistema galėtų prisitaikyti prie skirtingai suformuotų duomenų šaltinių ir juos priimtų kaip tinkamus. Tai įgyvendinus, galimi tampa tiek įvairiai suformuoti XML, duomenų bazės, duomenų šaltiniai, priimtinas ir svetainės duomenų šaltinis vertikaliems portalams.

Pritaikant importo sistemą su kuo daugiau šaltinių, taikomosios programos kuriamos kiekvienam šaltiniui atskirai, duomenims gauti naudojant programavimo kalbos turimas standartines ir vartotojo funkcijas XML ir HTML duomenų šaltiniams, duomenų bazės šaltiniui naudojamas į duomenų bazes orientuotos taikomosios programos, pagrįstos užklausų formavimu, papildomas teksto apdorojimas vykdomas reguliarių reiškinių pagalba, taikomųjų programų veikimo greitumas ir patikimumas priklauso nuo pačių gerai atlikto darbo, programavimo specifikos, funkcijų naudojimo ir pan.

5. Tyrimas, rezultatai

Pirmoje tyrimo dalyje išsiaiškinta, kuriose portalų naudojimo srityje darbe keliamos problemos aktualiausios, dažniausiai kylančios, taigi žinome, kur geriausia atlikti sekantį tyrimą ir gauti naudingiausios informacijos. Pirmo tyrimo rezultatas: darbe nagrinėjama probleminė sritis aktualiausia vertikaliems priklausantiems nekilnojamojo turto portalams.

Šio tyrimo tikslas:

- Galutinai įvertinti darbo probleminės srities aktualumą

- Įvertinti kiek išplečiamas importo sistemos panaudojamumas, pritaikant ne šaltinių duomenis taikomajai programai, o taikomąsias programas pritaikant prie galimų duomenų šaltinių

Tyrimo uždaviniai:

- išsiaiškinti, kokie duomenų perdavimo metodai labiausiai paplitę, kurie mažiausiai
- išsiaiškinti, kas įtakoja duomenų perdavimo metodo pasirinkimą
- išsiaiškinti, kokios aktualiausios problemos kyla duomenų apsikeitimo procese.

Tyrimo metu bendrauta su įmonių atstovais, galinčiais atsakyti į techninius klausimus, pagrinde susijusius su duomenų importu, duomenų perdavimo ir gavimo metodais, būdais. Iš viso tyrime sutiko dalyvauti 38 nekilnojamojo turto agentūros, susidūrusios su informacijos pateikimo portalams procesu.

Įgyvendinant pirmąjį tyrimo tikslą, visų pirma tiriama kokie duomenų perdavimo būdai naudojami, kokius duomenų šaltinius įmonės pateikia portalams. Gauti rezultatai 4 lentelėje.

XML	Informacija suvedama rankiniu būdu	Duomenų bazės šaltinis
7	31	0

4. Duomenų šaltinių pateikiami būdai

Remiantis 4 lentelės duomenimis, XML failu duomenų struktūrą pateikia beveik penktadalis, duomenų bazės šaltinis nenaudojamas, daugiausia įmonės portaluose informaciją suveda rankiniu būdu, kiti būdai nenaudojami.

Apklausos metu paaiškėjo, kad duomenų bazės šaltinio nei viena įmonė nepateikia, kad apsaugotų įmonės duomenis, taigi pasitvirtina darbe minėtas šio duomenų šaltinio trūkumas, kad beveik niekada įmonės nepatiki prisijungimo duomenų ir jis labai retai naudojamas kaip duomenų perdavimo galimybė.

Sekančiame etape nagrinėta, kodėl 31-rios įmonės naudoja rankinį informacijos suvedimą, o nepateikia kitokio duomenų perdavimo būdo. Paaiškinimai buvo dvejopi. Pirmasis, kad suvedimas rankiniu būdu joms yra priimtinas, talpinamas nedidelis kiekis informacijos, todėl tai nesukelia sunkumų. Antrasis paaiškinimas, kad savo duomenų XML struktūrą sukurti brangu arba nėra asmens, kuris tai atliktų. Šioje vietoje išryškėja darbe sprendžiama duomenų perdavimo problema, kuomet šaltinis turi reikiamos informacijos portalui, tačiau neturi galimybės jos perduoti. Tokiu atveju teigėme, kad šaltiniai prarandami, tačiau taip nėra, jie suveda informaciją rankiniu būdu. Tai išties stebėtina, kadangi esant nemažam kiekiui objektų,

sugaištama daug laiko juos įvedant naujus, keičiant esamus, trinant senus. Rankiniu būdu informaciją suvedančios įmonės, dažniausiai nepatenkintos tokia sistema, kadangi tai užima daug laiko, didelis darbuotojų apkrovimas, tačiau priverstos tai daryti, kad sulauktų kuo daugiau lankytojų dėmesio, jų informacija būtų lengviau pasiekiama, daugiau randama. Įdomu, kad kai kuriais atvejais naudojamas būdas suvesti objektus į kažkurį portalą rankiniu būdu, o po to šią informaciją naudoti savo svetainėje, taip sutaupant laiko nevedant tos pačios informacijos kelis kartus.

Suvedimas rankiniu būdu neautomatizuotas, užima daug laiko, žmoniškųjų resursų, todėl darbe ir nenagrinėtas, be to, didelė tikimybė, kad tokia informacija nėra dažnai atnaujinama, taigi nekorektiška, netiksli.

XML struktūros failą turėto didžiausią informacijos kiekį turinčios įmonės. Šiuo būdu perduodant informaciją nusiskundimų nebuvo, todėl galima daryti išvadą, kad plačiausiai naudojamas, paplitęs XML duomenų perdavimas išties geras, patikimas duomenų šaltinio būdas.

Toliau tyrėme atvejį, kuomet įmonės savo XML duomenis pateikia keliems portalams. Šiuo atveju siekiame išsiaiškinti, ar šaltinio perduodami duomenys visuomet pritaikomi juos gaunančiai sistemai ir priimami. XML keliems portalams pateikė keturos įmonės. Paaiškėjo, kad kiekvienam portalui pateikiami atskiri XML struktūros failai, pritaikyti būtent jiems. Taigi, pasireiškia duomenų struktūrų nesuderinamumo problema, įsitikiname, kad naudojamos taikomosios programos dažniausiai nėra pritaikytos įvairioms duomenų formoms, laikomasi vieno šablono, pagal tam tikrą standartą suformuotų duomenų struktūros, tik tokie duomenys laikomi tinkamais nuskaitymui ir saugojimui. Mūsų siūlomas sprendimas kurti ne vieną taikomąją programą, kuri priimtų tik tam tikrą duomenų struktūros šabloną, o taikomąsias programas kurti atskirai kiekvienam šaltiniui pasiteisina, pavyzdžiui, šio tyrimo atveju, tinkamais tampa visi XML duomenų šaltiniai.

Tolesnis tyrimas atliekamas siekiant įvertinti ar išplečiamas darbe naudojamos importo sistemos panaudojamumas, pritaikant ne šaltinių duomenis taikomajai programai, o taikomąsias programas pritaikant prie galimų duomenų šaltinių. Naudojami duomenų šaltinių būdai: XML, duomenų bazė, interneto svetainės.

Apklausus visas įmones kokiu būdu galėtų pateikti duomenis atliekamam tyrimui, XML duomenų šaltinį pateikė tos pačios 7-ios įmonės, duomenų bazės šaltinį patikėjo 1-na įmonė. Paklausus ar sutiktų suvesti objektus rankiniu būdu, dėl per didelio darbuotojų apkrovimo, sutiko labai maža dalis įmonių. Pasiūlius alternatyvą duomenis importuoti iš svetainės, kurie būtų automatiškai atnaujinami, visuomet pasirinkta ši alternatyva.

Apklausos metu sistemai pateikti galimi duomenų šaltinių būdai, pavaizduoti 5 lentelėje (Duomenų šaltinių skaičius išaugo, nes skaičiuojami visi galimi būdai, pavyzdžiui, kad duomenys būtų nuskaityti iš svetainių, sutiko visos įmonės).

XML	Duomenų bazės šaltinis	Interneto svetainė	Informacija suvestų rankiniu būdu
7	1	38	2

5 lentelė. Duomenų šaltinių pateikiami būdai

Gavus duomenų šaltinius, siekiama įsitikinti, kad galima kurti taikomąsias programas kiekvienam galimam duomenų šaltiniui, taikyti papildomą svetainės duomenų šaltinį vertikaliems portalams, palyginti programų ruošimo ir veikimo trukmę, atitinkamai pagal duomenų šaltinio rūšį, metodų naudojimo sprendimai priimti teisingi.

Anksčiau darbe taikomąsias programas XML duomenų šaltiniams pasirinkome kurti naudojant programavimo kalbos standartines, vartotojo sukurtas funkcijas, nesinaudojant bibliotekomis. Tyrimo metu išbandytas SAX metodas, jo metu duomenų nuskaitymas truko ilgiau. SAX metodas kol kas laikomas kaip labiausiai išbaigtas, todėl teigiame, kad pasirinkimas atliktas teisingai. Privalumas, kad kas kart atliekant duomenų gavimą, taupomas laikas programos veikime.

Duomenų bazės šaltinio atveju ilgiausiai truko perprasti duomenų bazės struktūrą, tai didžiausias šio duomenų šaltinio trūkumas. Duomenų bazė naudojo SQL programavimo kalbą, tad užklausos sukūrimas nesukėlė jokių problemų, perpratus duomenų bazės struktūrą, nesunkiai ir greitai gauti visi reikiami duomenys.

Interneto svetainės šaltiniai buvo visi tinkami, nei viena svetainė nebuvo sukurta Flash principu su vidine duomenų baze ar perduodami nežinomi kintamieji POST metodu, tačiau darbe atlikti papildomi nagrinėjimai ypač padėjo, kadangi vienas svetainės šaltinis buvo sukurtas flash principu su XML duomenų baze. Šiam atvejui duomenims gauti, darbe sprendimas rastas. Tiesa, sukūrus taikomąją programą vienai svetainei, jos struktūra pakito, todėl taikomąją programą teko pritaikyti iš naujo, tačiau tai neužėmė daug laiko, tereikėjo pakoreguoti tik tą programos dalį, kuri naudojo pakitusią HTML struktūros dalį, atlikus pakeitimus duomenų gavimas toliau vyko sklandžiai.

Taip pat taikomosios programos kurtos siekiant palyginti programų ruošimo ir veikimo trukmę, atitinkamai pagal duomenų šaltinio rūšį.

Išvados tokios: ilgiausiai trunka taikomosios programos kūrimas svetainės šaltiniui, toliau XML, o greičiausias duomenų gavimo būdas – duomenų bazė. Tyrimo atveju galima buvo gauti

duomenis naudojant vien tik svetainių duomenų šaltinius, tačiau jei yra pasirinkimas, kad importo sistema veiktų kuo greičiau, taikomosios programos duomenų šaltiniams kuriamos tokia prioritetine tvarka:

1. Duomenų bazės, kadangi duomenims gauti, tai greičiausiais, lengviausias būdas.
2. Duomenų gavimas XML lėtesnis, sunkesnis už duomenų bazių, tačiau greitesnis už svetainių.
3. Svetainės šaltinis. Nors ir lėčiausias, sudėtingiausias būdas, tačiau svarbiausia šaltinis galimas, nėra sugaištama pernelyg daug laiko, resursų

Apibendrinant, tyrimo metu nustatyta, kad nagrinėjamos darbe problemos aktualios, kadangi dažniausiai naudojamuose sistemose galimi šaltiniai tik tie, kurių duomenys specialiai paruošti kitai sistemai, jai suprantama, tinkama forma. Duomenų šaltiniai, kuriuose yra reikiama portalui informacija, tačiau jie nepritaikyti perdavimui, skirtingos struktūros nei reikalaujama gaunamos sistemos, suveda informaciją rankiniu būdu, kas yra didelis apsunkinimas šaltiniams ir šis būdas neišsprendžia duomenų apsikeitimo problemos.

Vertikalūs portalai plačiausiai naudoja XML interneto standarto paslaugą, tačiau šis būdas nėra visuomet galimas, dėl sunkumų jį pasigaminant. Duomenų bazės atveju, kartais gali būti sunku perprasti jos struktūrą, o tai svarbu, kad užklausos būtų teisingai suformuotos korektiškiems duomenims gauti. Vis tik duomenų bazės šaltinis patogus, greitas duomenų gavimas. Dažniausias duomenų perdavimo būdas – XML duomenų šaltinis. Išsiaiškinta, kad duomenų apsikeitimo procese yra aktualiausia duomenų struktūrų nesuderinamumo problema. Taikomąsias programas pritaikius prie galimų duomenų šaltinių, gauti visų įmonių duomenys automatiniu būdu, tai atlikus, importo sistemos panaudojamumas išplėstas ir tyrimo siekiai, tikslai galutinai įvykdyti.

Išvados

Darbe ištirta importo sistemų duomenų gavimo iš išorinių šaltinių, naudojant taikomąsias programas, problematika ir pateikti sprendimai, kurie gali būti naudojami, siekiant padidinti jos suderinamumo galimybes su daugiau galimų duomenų šaltinių. Išnagrinėtos dvi pagrindinės problemos: duomenų perdavimo problema, kuomet šaltinis turi reikiamos informacijos, tačiau neturi galimybės jos perduoti ir duomenų struktūrų nesuderinamumo problema, kuomet šaltinio perduodami duomenys nėra pritaikyti juos gaunančiai sistemai, todėl nepriimami.

Jei importo sistema priima tik tam tikra struktūra suformuotus duomenis, laikantis tam tikro nustatyto šablono, tuomet pagal jį sukuriama viena taikomoji programa, kuri kaskart naudojama duomenų nuskaitymui iš turimų duomenų šaltinių. Tokiu atveju pasireiškia duomenų struktūrų nesuderinamumo problema ir duomenų šaltiniai prarandami. Siekiant padidinti importo sistemos suderinimo galimybes su daugiau galimų duomenų šaltinių, nuspręsta nesilaikyti vieno suderinto standarto, o pritaikyti taikomąsias programas kiekvienam galimam duomenų šaltiniui. Nors kaskart duomenų gavimui, pritaikant programą prie savitos duomenų struktūros, sunaudojama daugiau laiko, žmoniškųjų darbo resursų, tačiau duomenis reikia atnaujinti, o paruošta programa to šaltinio nuskaitymui naudojama nuolatos, taip taupant laiką programos veikime, sistema naudoja mažiau kompiuterio resursų. Šį pasirinkimą lemia ir tai, kad importo sistema tuomet pritaikoma prie bet kokios struktūros duomenų šaltinio, taip jo neprarandant. Tokiu būdu išsprendžiama ir duomenų perdavimo problema, nes duomenų šaltiniai tampa galimais, tinkamais.

Jei yra galimybė, taikomoji programa kuriama ir duomenys gaunami iš duomenų bazės, kadangi tai greičiausias ir lengviausias būdas, kitu atveju naudojamas XML šaltinis, kurio taikomųjų programų kūrimas yra sudėtingesnis už užklausų formavimą duomenų bazei, tačiau lengvesnis už nuskaitymą iš interneto svetainės. Pasirenkant šaltinius, neprisirišant prie vieno būdo, įgyvendinama viena svarbiausių importo sistemos savybių, kad ji veiktų kuo greičiau, patikimiau.

Papildoma duomenų šaltinio alternatyva vertikaliems portalams – interneto svetainės, tačiau dėl ilgiausiai trunkančio taikomųjų programų kūrimo, ilgiausio jų veikimo, naudojamos tuomet, kai nėra kitų duomenų gavimo būdų. Šiuo duomenų šaltiniu išplečiamas galimų šaltinių kiekis. Tyrimų metu nustatyta, kad svetainės duomenų šaltinis galimas, išskyrus tais atvejais, kai ji sukurta Flash pagrindu su vidine duomenų baze, veiksmams su duomenimis kintamieji perduodami POST metodu yra nežinomi. Kadangi papildytas galimų, tinkamų šaltinių skaičius, išplečiamas importo sistemos panaudojimo galimybės didesniam duomenų kiekiui gauti.

Taikomosios programos kuriamos naudojant programavimo kalbos turimas standartines ir

vartotojo funkcijas XML ir HTML struktūros duomenų šaltiniams, duomenų bazės šaltiniui naudojant į duomenų bazes orientuotos taikomosios programas, pagrįstas užklausų formavimu, papildomas teksto apdorojimas duomenų paruošimui saugoti, pilnai gali būti atliktas reguliarių reiškinių pagalba.

Apibendrinant galima teigti, kad pagal apsibrėžtus tikslus ir uždavinius, importo sistema tenkina jai keliamus reikalavimus ir galima padidinti sistemos suderinamumo galimybes su daugiau duomenų šaltinių, gauti daugiau duomenų negu tai atliekama su jau naudojamomis, sukurtomis sistemomis. Tai leidžia manyti ir tyrimo metu, kuriant taikomąsias programas duomenų šaltiniams, apsibrėžtus tikslus ir uždavinius gauti patvirtinantys rezultatai.

Literatūros sąrašas

[BC02] B.Yates, R. Castillo. Balancing volume, quality and freshness in web crawling. In: Santiago, Chile. IOS Press Amsterdam, 2002, p.565-572. [žiūrėta 2008-04-21]. Prieiga per internetą: <http://www.dcc.uchile.cl/~ccastill/papers/baeza02balancing.pdf>

[BU05] Anthony Bernal, Ian Uriarte. IBM, Creating a new portal. 2005

[žiūrėta 2008-04-03]. Prieiga per Internetą: <www.ibm.com/developerworks/websphere/library/techarticles/0508_bernal/0508_bernal-intro.html>

[Dič07] V. Dičiūnas. Diskrečios struktūros. Vilnius, 2007, p.165. [žiūrėta: 2008-04-23]. Prieiga per internetą: <http://www.mif.vu.lt/~valdas/DISKRECIOS_STRUKTUROS/Tutorial/ds_konspektai.pdf>

[HMU06] John E. Hopcot, R. Motwani, Jeffrey D. Uleman. Introduction to Automata Theory, Languages, and Computation (3rd Edition), 2006. Prieiga per internetą: <http://portal.acm.org/citation.cfm?id=1177300>

[PSM04] G. Pant, P. Srinivasan, F. Menczer. Crawling the web. 2004, p.1-25. [žiūrėta 2008-04-22]. Prieiga per internetą: <http://www.informatics.indiana.edu/fil/Papers/crawling.pdf>

[PKS08] Programavimo kalba. Straipsnis iš Vikipedijos, laisvosios enciklopedijos, 2008. [žiūrėta 2008-05-28]. Prieiga per internetą: <http://lt.wikipedia.org/wiki/Programavimo_kalba>

[TPC08] Tiobe Programming Community Index for May 2008. [žiūrėta 2008-05-27]. Prieiga per internetą: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

[WDO98] W3C. Document Object Model (DOM) Level 1 specification. W3C recommendation, 1998 [žiūrėta 2008-05-28]. Prieiga per internetą: <http://www.w3.org/TR/REC-DOM-Level-1/>

[WP08] Web Portal. Wikipedia, the free encyclopedia, 2008. [žiūrėta 2008-05-10]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Web_portal>

[WWW08] World Wide Web Consortium**.** Wikipedia, the free encyclopedia, 2008. [žiūrėta 2008-05-27]. Prieiga per internetą: <http://en.wikipedia.org/wiki/World_Wide_Web_Consortium>

[WS08] Web service. Wikipedia, the free encyclopedia, 2008. [žiūrėta 2008-05-28]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Web_services>

[WXM07] Wikibooks. XML - Managing Data Exchange/Introduction to XML, 2007. <http://en.wikibooks.org/wiki/Programming:XML>