

---

IŠ DALIES EUROPOS SOCIALINIO FONDO LĖŠOMIS FINANSUOJAMAS PROJEKTAS "LIETUVOS GEOGRAFINĖS INFORMACIJOS  
VALDYTOJŲ KVALIFIKACIJOS KĖLIMAS" (NR. BPD2004-ESF-2.2.0-02-05/0143)

Valstybės tarnautojų mokymo programa

GEOGRAFINĖS INFORMACIJOS INFRASTRUKTŪROS NUOTOLINIS MOKYMAS

Mokomoji knyga

# WEB PROGRAMAVIMAS GEOGRAFINĖS INFORMACIJOS INFRASTRUKTŪRAI

## GII-09

Vilnius, 2008

Mokomoji knyga „WEB programavimas geografinės informacijos infrastruktūrai“ (GII-09)

Autoriai

1 skyrius - Frank Fucile

2 skyrius - Michael O'Neal Campbell

3, 4, 5 skyriai - Michael Govorov ir dr. Gennady Gienko

Recenzavo ir redagavo

prof. Romualdas Baušys (Vilniaus Gedimino technikos universitetas)

Recenzavo

prof. habil. dr. Genadijus Kulvietis (Vilniaus Gedimino technikos universitetas)

Iš anglų kalbos vertė ir redagavo

UAB Astraneta

Valstybės tarnautojų nuotolinio mokymo programos „Geografinės informacijos infrastruktūros nuotolinis mokymas“, patvirtintos Valstybės tarnybos departamento prie LR Vidaus reikalų ministerijos direktoriaus įsakymu „Dėl valstybės tarnautojų mokymo programų tvirtinimo“ Nr. 27V-72 (2007-04-17), mokymo kursų medžiaga parengta vykdant iš dalies Europos Sąjungos lėšomis finansuojamą projektą „Lietuvos geografinės informacijos valdytojų kvalifikacijos kėlimas“ (Nr. BPD2004-ESF-2.2.0.-02-05/0143)

Projekto vykdytojas Nacionalinė žemės tarnyba prie Žemės ūkio ministerijos.

Projekto rangovas UAB HNIT-BALTIC, Vilniaus Gedimino technikos universitetas

© Autoriaus išimtinės turtinės teisės priklauso Nacionalinei žemės tarnybai prie Žemės ūkio ministerijos

© Autoriaus neturtinės teisės priklauso Malaspina University-College

## Turinys

1.	Interneto technologija, HTML ir CSS .....	6
1.1.	Kaip veikia internetas? .....	7
1.2.	HTML žymos .....	12
1.3.	Nuorodos .....	26
1.4.	Paveikslai .....	30
1.5.	Lentelės .....	40
1.6.	Rėmeliai .....	46
1.7.	HTML padalijimai .....	49
1.8.	CSS .....	51
1.9.	Formos .....	57
2.	Interneto kalbos vartotojo sąsajos, ryšio, duomenų ir schemų aprašams .....	64
2.1.	DHTML, W3C standartai, HTML dokumento objektų modelis .....	65
2.2.	Skriptų rašymo pagrindai .....	72
2.3.	XML kodavimo pagrindai .....	96
2.4.	Išvados .....	113
3.	Interneto žemėlapiai ir susijusios technologijos .....	117
3.1.	Daugiapakopė architektūra .....	118
3.2.	„AJAX“ technologija .....	124
3.3.	Interneto žemėlapių programinės įrangos architektūra .....	125
3.4.	WMS paslaugos .....	129
3.5.	WFS paslaugos .....	132
3.6.	Interneto aprėpties paslauga .....	141
3.7.	Filtro kodavimas .....	142
3.8.	Sluoksnių stiliaus deskriptorius ir simbolikos kodavimas .....	143
3.9.	Išvados .....	146
4.	XML tipo kalbos žemėlapiams ir SOA .....	150
4.1.	XML kalbos .....	151
4.2.	GML kalba .....	152
4.3.	ESRI ArcXML kalba .....	162
4.4.	KML kalba .....	168
4.5.	SVG kalba .....	170
4.6.	Interneto žemėlapių paslaugos ir į paslaugas orientuota architektūra (SOA) ....	172
4.7.	Išvados .....	177
5.	ESRI interneto programų architektūra .....	182
5.1.	Kas yra ESRI ArcIMS ir „ArcGIS Server“? .....	183
5.2.	ESRI interneto komponentai: serverio pusė .....	185
5.3.	ESRI ArcIMS vartotojo sąsajos komponentai: serverio pusė .....	191
5.4.	ESRI ArcIMS komponentai: kliento pusė .....	195
5.5.	ESRI ArcIMS ryšio srautas .....	196
5.6.	ESRI ArcIMS diegimas .....	199
5.7.	ESRI ArcIMS ir „ArcGIS Server“ adaptavimo galimybės .....	202
5.8.	ESRI ArcIMS papildomos savybės .....	204
5.9.	Išvados .....	206

## Įžanga

Interneto taikomoji programa leidžia turinį iš serverio perduoti klientui naudojant vaizdinę vartotojo sąsają, dažnai interneto naršyklėje. Didžioji dalis verslo logikos, duomenų ir kitų procesų yra valdomi serverio pusėje naudojant ne tik turtingą programavimo aplinką (C#, „Visual Basic.NET“, „Java“ ir k. t.), bet ir skriptų kalbas – PHP, „Python“, „JavaScript“, „VBScript“. Kadangi naudotojas interneto taikomąją programą mato interneto naršyklėje, kliento pusės programa yra kuriama naudojant interneto standartus, užtikrinančius vartotojo sąsają, pavyzdžiui, HTML, CSS ir skriptų kalbas (pvz., „JavaScript“). Ryšiai tarp kliento ir serverio bei duomenų kodavimui naudojami kiti interneto standartai, pavyzdžiui, XML arba SOAP. Be to, interneto taikomosios programos gali būti realizuojamos įvairiose aplinkose (daugiapakopėje architektūroje, SOA) ir naudojant įvairias technologijas (.NET, J2EE arba AJAX).

Šiame mokymo kurse apžvelgiamos interneto technologijos ir programavimo aplinkos, skirtos interneto žemėlapių programoms kurti, o kai kurios šių technologijų aptariamos išsamiau. Pirmoje paskaitoje supažindinama su interneto ištakomis ir jo funkcijomis, susijusiomis su tinklalapių kūrimu ir programavimu. Studijuojantieji susipažins su pagrindiniais interneto struktūriniais elementais, susijusiais su HTML kodais, tiksliau – kaip naudoti lenteles, nuorodas, rėmelius, paveikslus ir HTML formas. Jie taip pat išmoks sudėtingesnio programavimo pasitelkdami pakopinio stiliaus lapus (CSS), kurie padeda formuoti tinklalapių išvaizdą, kurią nuotaiką ir išdėstymą. Šias žinias galima pritaikyti kuriant interneto žemėlapių taikomąsias programas, kuriems reikalingi HTML ir CSS įgūdžiai.

Antroje paskaitoje aptariamos skriptų kalbos ir susijusios technologijos – DHTML, HTML dokumento objektų modeliai (DOM) ir „JavaScript“. Nemažai interneto žemėlapių taikomųjų programų naudoja „JavaScript“ įvairioms kliento pusės funkcijoms atlikti, pavyzdžiui, žemėlapių spragtelėjimo sąveikai ir duomenų atsisiuntimui iš serverio per atgalinį kreipinį (angl. *callback*). Be to, paskaitoje analizuojamos XML ir XSL kalbos, kurios yra labai svarbios interneto užklausų ir atsakymų protokolams, erdvinių duomenų ir taikomųjų schemų kodavimui bei duomenų atvaizdavimui. Šioje paskaitoje supažindinama su šiomis interneto taikomųjų programų technologijomis praplečiant temas, kurios buvo nagrinėjamos 1 paskaitoje, ir apibūdinant, kaip naujaisi interneto taikomųjų programų technologijos laimėjimai prisidėjo prie interneto technologijų veiksmingumo didinimo.

Trečioje paskaitoje pristatomos interneto žemėlapių ir kitos susijusios technologijos bei specifikacijos. Pirmojoje temoje bus aptariama daugiapakopė architektūra, sudaranti interneto žemėlapių paslaugų aplinką. Taip pat apžvelgiama ir interneto technologija „AJAX“, kuri dažnai naudojama kliento pakopos interneto žemėlapių taikomose programose. Toliau analizuojama triapakopė architektūra, naudojama interneto žemėlapių paslaugoms, ir sistemos, esančios WMS (Web Mapping Server) ir „storajame“ WFS (Feature Server), taip pat pristatomi interneto žemėlapių klientai. Toliau smulkiai aptariamos OGC standartų realizacijos specifikacijos, apibrėžiančios „Web Mapping Server“, „Web Feature Server“ ir „Web Coverage Server“ funkcines galimybes. Šiomis specifikacijomis reikia naudotis kuriant suderintas paskirstytas interneto žemėlapių sistemas. Šioje paskaitoje pritaikomos ankstesnėse paskaitose įgytos studentų žinios apie HTTP, XML, XSL, DTD, XSD ir k. t., ir mokomasi, kaip kurti interneto žemėlapių architektūras.

Ketvirtoje paskaitoje nagrinėjamos geoerdviniams duomenims skirtos kalbos, kurias galima realizuoti XML. Joms priskiriamos GML (*Geographic Markup Language*), ArcXML (*ESRI Arc Extensible Markup Language*), KLM (*Google Keyhole Markup Language*), SVG (*Scalable Vector Graphics*), GeoRSS (*Geographic Really Simple Syndication*) ir daugelis kitų kalbų. Be to, XML kartais naudojama kaip šių taikomųjų programų kalbų specifikacijų kalba. Šioje paskaitoje aptariamos kelios XML taikomųjų programų gramatikos, naudojamos erdvinių duomenų mainų kodavimui, interneto žemėlapių paslaugos ryšio ir žemėlapių apibrėžtis, interneto paslaugų naudojimas ir hibridinimas (angl. *mashuping*) į paslaugas orientuotoje architektūroje ir vektorinės grafikos atvaizdavimas kliente. Šios XML taikomųjų programų schemas gali būti naudojamos duomenų modeliavimui ir mainams įvairiose viešosiose ir komercinėse srityse – kadastro ir žemėtvarkos, eismo ir transporto, telekomunikacijų, aplinkosaugos ir kitose. Taip pat aptariamos temos, susijusios su į paslaugas orientuota architektūra, interneto žemėlapių paslaugomis bei susijusiomis technologijomis.

Ankstesnėse paskaitose buvo aptarta interneto žemėlapių architektūra, jos komponentai ir vartotojo sąsajai, ryšiui, duomenims skirtos interneto kalbos bei schemų aprašai. Paskutinėje paskaitoje anksčiau nagrinėtos temos pristatomos pasitelkiant praktinio įgyvendinimo pavyzdžius – ESRI ArcIMS ir ArcGIS Server programinę įrangą. ESRI buvo interneto žemėlapių programinės įrangos kūrimo lyderė. Bendrovės produktus klientai rinkosi analizei atlikti, kadangi produktų dokumentaciją lengva naudoti. Viso pasaulio miestų tarybos, savivaldos institucijos, įmonės ir kitos organizacijos publikuoja, randa ir dalijasi geoerdvine informacija naudodamos ArcIMS.

# 1. Interneto technologija, HTML ir CSS

Šioje paskaitoje susipažinsite su interneto ištakomis ir jo funkcijomis, susijusiomis su tinklalapių kūrimu ir programavimu. Susipažinsite su pagrindiniais interneto struktūriniais elementais, susijusiais su HTML kodu, tiksliau – kaip naudoti lenteles, nuorodas, rėmelius, paveikslus ir HTML formas. Taip pat išmoksite sudėtingesnio programavimo pasitelkdami pakopinio stiliaus lapus, kurie padeda formuoti tinklalapių išvaizdą, kuriamą nuotaiką ir išdėstymą.

1 paskaitoje bus aptariamos šios temos:

## Paskaitos planas

- Kaip veikia internetas?
- HTML žymos
- Saitai
- Paveikslai
- Lentelės
- Rėmeliai
- Padalijimai
- CSS (pakopinio stiliaus lapai)
- Formos

## 1.1. Kaip veikia internetas?

### 1.1.1 Kas yra internetas?

Internetas – tai technologija, sujungianti viso pasaulio kompiuterių tinklus. Interneto ryšys pasiekia net Antarktidą ir Tarptautinę kosminę stotį. Internetą, kuriam naudojamas TCP/IP (perdavimo valdymo protokolas/interneto protokolas), ARPANET tinklo pagrindu praėjusio amžiaus aštuntojo dešimtmečio pradžioje sukūrė JAV vyriausybė. TCP/IP protokolas leidžia siųsti ir priimti visų rūšių skaitmeninę informaciją.

### 1.1.2 Interneto sluoksniai

Septyni atvirųjų sistemų jungties (OSI) modelio duomenų perdavimo lygmenys (šaltinis – [http://lt.wikipedia.org/wiki/OSI\\_modelis](http://lt.wikipedia.org/wiki/OSI_modelis)):

1. Fizinis (10Base-T, ISDN, RS-232)
  - Fizinis sluoksnis apibrėžia elektrines ir fizines įrenginių specifikacijas.
2. Ryšio (kanalinis) (Ethernet, Wi-Fi, PPP, „Token Ring“...)
  - Ryšio (kanalinis) sluoksnis sudaro funkcines ir procesines galimybes perduoti duomenis iš vieno tinklo mazgo į kitą, taip pat aptikti ir galbūt ištaisyti klaidas, kurios gali atsirasti fiziniu lygmeniu.
3. Tinklo (IPV4, IPV6, ICMP, IGMP, ARP...)
  - Tinklo sluoksnis sudaro funkcines ir procesines galimybes viename arba keliuose tinkluose perduoti kintamojo ilgio duomenų sekas iš šaltinio į paskirties vietą išlaikant paslaugos kokybę, nurodytą transporto sluoksnyje. Transporto (TCP, UDP...)
    - Transporto sluoksnyje duomenys skaidriai perduodami tarp galutinių vartotojų teikiant patikimas duomenų perdavimo paslaugas aukštesniems sluoksniams.
5. Sesijos (RPC)
  - Sesijos sluoksnis valdo kompiuterių dialogus ir sujungimus (sesijas).
6. Duomenų pateikimo (prezentacijos) (XDR)
  - Duomenų pateikimo (prezentacijos) sluoksnyje sukuriamas kontekstas taikomojo sluoksnio subjektams, kuriame aukštesnio sluoksnio subjektai gali naudoti įvairią sintaksę ir semantiką, jeigu tik jas supranta Duomenų pateikimo paslauga, ir jų atvaizdavimą (angl. *mapping*).
7. Taikomasis (NFS)
  - Taikomajame sluoksnyje vyksta tiesioginė sąveika su taikomųjų programų procesais ir jiems teikiamos bendros taikomųjų programų paslaugos; jis taip pat teikia užklausas duomenų pateikimo (prezentacijos) sluoksniui.

Siunčiama informacija yra suskaidoma ir performatuojama į daugelį TCP/IP paketų. Skirtingos rūšies informacijai ir (arba) skirtingoms programoms naudojami skirtingų rūšių paketai.

## FTP (failų perdavimo protokolas)

FTP, arba failų perdavimo protokolas, naudojamas duomenims perduoti iš vieno kompiuterio į kitą internetu arba tinkle.

FTP dažniausiai naudojamas keitimosi failais protokolas visuose tinkluose, kurie palaiko TCP/IP protokolą (pavyzdžiui, internete ar intranete). Perduodant failus dalyvauja du kompiuteriai – serveris ir klientas. FTP serveris, kuriame įdiegta FTP serverio programinė įranga, tinkle laukia ryšio užklausų iš kitų kompiuterių. Kliento kompiuteris, kuriame įdiegta FTP kliento programinė įranga, pradeda jungtis prie serverio. Prisijungęs, klientas gali atlikti įvairius manipuliavimo failais veiksmus, pavyzdžiui, įkelti juos į serverį, atsisiųsti failus iš serverio, pakeisti failų vardus arba juos ištrinti iš serverio ir pan. Bet kuri programinės įrangos bendrovė arba programuotojas gali sukurti FTP serverio arba kliento programinę įrangą, nes protokolas yra atviro standarto. Praktiškai kiekviena kompiuterinė platforma palaiko FTP protokolą. Todėl bet kuris kompiuteris, įjungtas į TCP/IP pagrindu veikiančią tinklą, gali manipuluoti failais kitame į tą tinklą įjungtame kompiuteryje, kad ir kokios operacinės sistemos būtų naudojamos (jeigu kompiuteriai leidžia FTP prieigą). Yra sukurta daug FTP kliento ir serverio programų. FTP serverius galima pateikti visuose žaidimų serveriuose, balso perdavimo serveriuose, interneto paslaugų tiekėjo mazguose ir kituose fiziniuose serveriuose.

## HTTP (hiperteksto perdavimo protokolas)

Hiperteksto perdavimo protokolas (HTTP) yra ryšių protokolas, naudojamas informacijai perduoti arba teikti intranete ir pasauliniame internete. Jis buvo sukurtas siekiant sudaryti galimybes hiperteksto puslapiams skelbti ir ieškoti. HTTP kūrimą koordinavo W3C (Pasaulinio interneto konsorciumas) ir IETF (Interneto inžinierių darbo grupė), o darbą vainikavo paskelbti RFC standartai, visų pirma RFC 2616 (1999 m. birželio mėn.), apibrėžiantis HTTP/1.1 – plačiai naudojamą HTTP versiją.

HTTP – tai užklausų ir atsakymų protokolas tarp kliento ir serverio. Klientas, teikiantis HTTP užklausą (internetu naršyklė, paieškos serveris (angl. *spider*) arba kita galutinio vartotojo priemonė), vadinamas vartotojo programa (angl. *user agent*). Atsakantis serveris, kuriame saugomi arba kuriami tokie ištekliai, kaip HTML failai ir paveikslai, vadinamas siunčiančiuoju serveriu (angl. *origin server*). Tarp vartotojo programos ir siunčiančiojo serverio gali būti keletas tarpininkų – tarpinių serverių, tinklo sietuvų ar tunelių (angl. *tunnels*). HTTP naudoja ne tik TCP/IP ir jo pagalbinius sluoksnius, nors tai yra populiariausias protokolo naudojimo būdas internete. Iš tiesų HTTP galima „naudoti kartu su bet kuriuo kitu protokolu internete arba kituose tinkluose. HTTP tiesiog laikomas patikimu perdavimo būdu; gali būti naudojamas bet kuris protokolas, kuris teikia tokias pačių garantijų“.

Paprastai HTTP klientas teikia užklausą prisijungdamas perdavimo valdymo protokolo (TCP) ryšiu prie konkretaus pagrindinio kompiuterio prievado (numatytasis yra 80 prievadas; žr. TCP ir UDP prievadų numerių sąrašą). HTTP serveris, kuris klausosi šio prievado, laukia, kol klientas išsiųs užklausos pranešimą.

Gavęs užklausą, serveris išsiunčia būsenos eilutę, pavyzdžiui, „HTTP/1.1 200 OK“, ir savo pranešimą, kurio pagrindinę dalį tikriausiai sudaro prašomas failas, klaidos pranešimas arba kita informacija.



Ištekliai, kuriais galima naudotis pasitelkiant HTTP, yra identifikuojami naudojant universaliuosius išteklių identifikatorius (URI) (arba, tiksliau, universaliuosius adresus (URL)) naudojant http: arba https URI schemas.

### **Telnetas (terminalo emuliacijos protokolas)**

Terminalo emuliacijos, terminalo taikomoji programa, terminalas, arba sutrumpintai tty, yra programa, kuri pamėgdžioja „nebylų“ (angl. „*dumb*“) vaizdo terminalą kitoje vaizdavimo architektūroje. Nors sąvoka „terminalas“ paprastai yra sinonimiška komandinės eilutės apvalkalui (angl. *command line shell*) arba tekstiniam terminalui, ji apima visus nutolusius terminalus, įskaitant ir grafines sąsajas. Terminalo emuliacijos grafinėje vartotojo sąsajoje dažnai vadinamas terminalo langu.

Terminalo langas leidžia vartotojui pasiekti tekstinį terminalą ir naudotis visomis jo taikomosiomis programomis – komandinės eilutės sąsajomis (CLI) bei tekstinėmis vartotojo sąsajomis taikomosiomis programomis, kurios gali veikti tame pačiame arba kitame kompiuteryje per telnetą, ssh protokolą arba skambinant. „Unix“ tipo operacinėse sistemose dažnai vienas arba keli terminalo langai yra sujungti su vietiniu kompiuteriu.

Terminalai paprastai palaiko kaitos sekų grupę, kuria nustatoma spalva, žymeklio padėtis ir pan., pavyzdžiui, terminalo valdymo sekų standartus ECMA-48, ANSI X3.64 arba ISO/IEC 6429. „xterm“, populiarus terminalo emuliacijos, sukurtas X11.

Bankuose, draudimo bendrovėse ir valdžios įstaigose – vienuose pirmųjų kompiuterinių technologijų naudotojų – terminalo emuliacijos dažnai naudojamos ir dabar. Jų didžiuosiuose kompiuteriuose paprastai veikia dešimtmečių senumo taikomosios programos. Senieji „nebylūs“ vaizdo terminalai, kurie buvo naudojami prisijungti prie didžiojo kompiuterio, jau seniai atgyveno, tačiau taikomosios programos didžiuosiuose kompiuteriuose vis dar naudojamos. Dažnai terminalo emuliacijos yra vienintelė galimybė vartotojui pasiekti taikomas programas, veikiančias šiuose senesniuose kompiuteriuose.

### **SMTP (paprastas pašto perdavimo protokolas)**

SMTP – tai gana paprastas tekstinis protokolas, kuriame vienas ar keli el. laiško gavėjai yra nurodomi (ir daugeliu atveju patvirtinama, kad jie egzistuoja) kartu su el. laiško tekstu ir galimai kitais užkoduotais objektais. Tada kliento ir serverių užklausų ir atsakymų būdu el. laiškas perduodamas nuotoliniam serveriui. SMTP kliento funkcijas gali atlikti galutinio vartotojo el. pašto klientas (pašto skaitymo programa) (angl. *MUA (Mail User Agent)*) arba persiuntimo serverio pašto siuntimo programa (angl. *MTA (Mail Transport Agent)*).

El. pašto klientas išsiunčiamo pašto SMTP serverį atpažįsta pagal jo sąranką. Persiuntimo serveris paprastai nustato, prie kurio SMTP serverio prisijungti, patikrindamas kiekvieno gavėjo domeno vardo (el. pašto adreso dalis, esanti į dešinę nuo simbolio @) MX („Mail Exchange“) DNS įrašą. Jeigu MX įrašo nėra, reikalavimus atitinkančios (ne visos) pašto skaitymo programos grįžta prie paprasto A įrašo. Kai kurios dabartinės pašto persiuntimo programos taip pat naudoja SRV įrašus (bendresnio tipo MX), tačiau šie nėra labai paplitę. (Persiuntimo serverius taip pat galima konfigūruoti taip, kad būtų naudojamas išmanusis serveris (angl. *smart host*).

SMTP klientas pradeda TCP jungimąsi prie serverio 25 prievado (nebent kitaip nurodyta sąrankoje). Naudojant telneto programą yra gana paprasta patikrinti SMTP serverį (žr. toliau).

SMTP yra „stūmimo“ protokolas, neleidžiantis pagal pareikalavimą „ištraukti“ laiškų iš nuotolinio serverio. Tam pašto klientas turi naudoti POP3 arba IMAP. Kitas SMTP gali nurodyti pristatyti paštą SMTP protokolu naudojant ETRN.

### **„Bittorrent“ (duomenų protokolas, naudojamas filmams)**

„BitTorrent“ – tai lygiaverčių tinklo kompiuterių (angl. *peer-to-peer*) (P2P) dalijimosi failais ryšio protokolas. Šis protokolas naudojamas plačiai dalytis dideliais duomenų kiekiais, o failą pradėjusiam platinti įrenginiui tenka tik dalis aparatinės įrangos, prieglobos (angl. *hosting*) ir juostos pločio (angl. *bandwidth*) sąnaudų. Iš tiesų, kai duomenys yra platinami naudojantis „BitTorrent“ protokolu, kiekvienas gavėjas duomenų dalis siunčia naujesniems gavėjams, taip sumažindamas kiekvienam individualiam šaltiniui tenkančias sąnaudas ir našta, užtikrindamas didesnius sistemos išteklius ir sumažindamas priklausomybę nuo failą pradėjusio platinti įrenginio.

Šį protokolą 2001 m. balandžio mėn. sukūrė ir 2001 m. liepos 2 d. jo pirmąją versiją išleido programuotojas Bram Cohen. Dabar protokolo priežiūrą atlieka jo įkurta bendrovė „BitTorrent, Inc.“.

Nors protokolas naudojamas perduoti duomenims, kurie sudaro nemažą interneto srautą, tikslų jo dydį nustatyti nėra lengva.

Yra daugybė darančių „BitTorrent“ klientų, parašytų pačiomis įvairiausiomis programavimo kalbomis ir veikiančių su pačiomis įvairiausiomis kompiuterijos platformomis.

### **1.1.3 Struktūrinis elementas – TCP/IP**

TCP/IP modelis, arba interneto pamatinis modelis, kartais vadinamas DoD modeliu (angl. „DoD, Department of Defence“ – JAV gynybos ministerija) arba ARPANET pamatiniu modeliu, yra kelių sluoksnių abstraktusis ryšio ir kompiuterinių tinklų protokolų kūrimo aprašas. Jį praėjusio amžiaus aštuntajame dešimtmetyje sukūrė DARPA interneto protokolams rengti. TCP/IP modelis ir dabar labai artimai atspindi interneto struktūrą.

Pirminį TCP/IP pamatinį modelį sudaro keturi sluoksniai, tačiau kai kurie autoriai teigia, kad dabar modelis jau turi penkis sluoksnius, o jo žemiausiasis sluoksnis (tinklo prieigos sluoksnis) yra padalytas į fizinį ir ryšio (kanalinį) sluoksnius. Tačiau nėra tokio IETF standartų dokumento (angl. *standards-track document*), kuriame būtų patvirtintas penkerių sluoksnių modelis. Taip yra tikriausiai dėl to, kad IETF nėra standartizavusi fizinio ir ryšio (kanalinio) sluoksnių protokolų. IETF dokumentuose nepritariama griežtam sluoksnių atskyrimui. Kadangi institucija, atsakinga už protokolų rinkinio techninių charakteristikų tvirtinimą, nėra pripažinusi penkių sluoksnių modelio, nėra netikslinga penkių sluoksnių pristatymus laikyti mokomosiomis priemonėmis, sudarant galimybes kalbėti apie ne IETF protokolus fiziniame sluoksnyje.

Šis modelis buvo sukurtas anksčiau nei OSI pamatinis modelis, tad Interneto inžinierių darbo grupė (IETF), kuri yra atsakinga už modelį ir jam kuriamus protokolus, niekuomet

nejautė įsipareigojimo laikytis OSI reikalavimų. Nors bazinis OSI modelis dažnai naudojamas mokymams, septynerių sluoksnių OSI modelis neatspindi realaus pasaulio protokolo architektūros (RFC 1122), kuri naudojama vyraujančioje interneto aplinkoje.

Atnaujintame IETF architektūros dokumente [1] netgi yra dalis, pavadinta „Sluoksniavimas laikomas žalingu“. Sluoksniavimas, kaip pagrindinė architektūros savybė, akcentuojamas ne TCP/IP, o OSI modelyje. Daug painiavos atsiranda tada, kai OSI tipo sluoksniavimą bandoma primesti architektūrai, kuri minimizuoja sluoksnių naudojimą.

Išsamesnis TCP/IP modelio paaiškinimas pateiktas šiame tinklalapyje: [http://en.wikipedia.org/wiki/TCP/IP\\_model](http://en.wikipedia.org/wiki/TCP/IP_model) (anglų kalba).

#### 1.1.4 Interneto plėtra

Galima drąsiai teigti, kad per pastaruosius keletą metų interneto plėtra vyko geometrine progresija. Naujausi statistiniai duomenys rodo, kad 20 proc. pasaulio gyventojų naudojami internetu. Dar svarbiau yra tai, kad likę 80 proc. pamažu įsilieja į interneto bendruomenę, kadangi vis daugiau šalių sukuria infrastruktūrą, būtiną interneto prieigai užtikrinti.

Ypač svarbu paminėti, kad internetas iš lėto pradeda plėtoti ir besivystančiose šalyse, kadangi vis akivaizdžiau, kad stipri ekonomika yra neatsiejama nuo interneto prieigos, reikalingos verslui, švietimui, pramogoms ir valdžios institucijoms, masto ir kokybės.

Norėdami išsamiau susipažinti su interneto statistika, apsilankykite šiame tinklalapyje: <http://www.internetworldstats.com/> (anglų kalba).

#### 1.1.5 Populiariausias interneto naudojimas

Internetas – tai pasaulinė, viešai prieinama tarpusavyje sujungtų kompiuterių tinklų visuma, kurioje duomenys perduodami paketinio duomenų siuntimo būdu standartiniu interneto protokolu (IP). Tai „tinklų tinklas“, sudarytas iš milijonų mažesnių buitinių, akademinų, verslo ir valdžios institucijų tinklų, kurie kartu teikia įvairiausios informacijos ir paslaugų – elektroninį paštą, interneto pokalbius, perduoda failus, taip pat tarpusavyje susietus tinklalapius ir kitus pasaulinio interneto išteklius.

Pastaruosius keletą metų internetas vis dažniau naudojamas pramogoms ir bendradarbiavimui. Tai liudija tokių interneto socialinių tinklų (angl. *social networking websites*), kaip „My Space“ ir „Facebook“ populiarumas. Naujausi IBM paskelbti statistiniai duomenys rodo, kad žmonės daugiau laiko skiria internetui nei televizijai žiūrėti: <http://broadcastengineering.com/iptv/ibm-internet-overtaking-tv-0828/> (anglų kalba).

## 1.2. HTML žymos

### 1.2.1 Pagrindinės žymos

**Pagrindinės** HTML žymos yra gana nesudėtingos. Visų pirma svarbu suprasti, kaip veikia žymėtojo hiperteksto kalbos (HTML) sintaksė ir kaip ji tvarko tekstą.

- HTML žymą iš abiejų pusių supa kampiniai skliaustai (simboliai mažiau nei „<“ ir daugiau nei „>“).

Tipišką žymą <title> sudaro trys dalys:

1. < = pradžios kampinis skliaustas
  2. title = žymos pavadinimas
  3. > = pabaigos kampinis skliaustas
- Daugelis žymų naudojamos poromis, kurios supa jų žymimą tekstą. Pirmoji žyma pradeda, o antroji – pabaigia žymėjimą.

```
<title>Tinklalo pavadinimas</title>
```

- Antroji žyma, kurią mes pavadinsime „išjungikliu“, pradeda **dešiminiu brūkšniu** „/“ ir žymi veiksmo pabaigą.
  - Pavyzdžiui, paryškinimo žyma atrodo taip: <b>paryškina šį tekstą</b>.
- Kaip ir visose programavimo kalbose, yra šios taisyklės išimčių. Pavyzdžiui, žyma <br> sukuria eilutės lūžį, bet neturi „išjungiklio“. Žyma <br> tiesiog į tekstą įterpia eilutės lūžį.
- Be HTML žymų reikia atsižvelgti ir į požymius. Požymiai žymų nurodymams gali taikyti vertes. Pavyzdžiui, žyma <p> pradeda naują pastraipą. Pridėjus lygiuotės požymį, pakeičiama naujos pastraipos padėtis. Šiame pavyzdyje tekstą centruojame:

```
<p align="center">Centre lygiuojamos pastraipos pavyzdys</p>
```

Viskas bus daug aiškiau, kai patys pradėsite rašyti kodą. Tam visų pirma turite atverti teksto rengyklę. Šioje paskaitoje naudosime kompiuteryje įdiegtą „Windows“ teksto rengyklę „Notepad“. Programą „Notepad“ rasite meniu Programos/Reikmenys. Darbalaukyje atlikite šiuos veiksmus:

1. Spragtelėkite mygtuką „Start“ (Pradėti)
2. Pasirinkite „Reikmenys“
3. Spragtelėkite ant „Notepad“

Turėtumėte pamatyti atvertą „Notepad“ langą. Šią rengyklę naudosite HTML kodui rinkti nagrinėjant kitą temą.

### 1.2.2 HTML išdėstymas

Kadangi rengyklė „Notepad“ jau atidaryta, laikykitės šios temos nurodymų ir rinkite toliau pateikiamą HTML kodą.

Visi tinklalapiai turi prasidėti dokumento tipo apibrėžties (DTD) kodu, esančiu pačiame tinklalapio viršuje. Kiekviename tinklalapyje, kurį kuriate, turite nurodyti jo **DTD**, kaip parodyta toliau:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

Kitoje kodo eilutėje naršyklė ieškos žymos **<html>**.

Naršyklė pradės skaityti tinklalapį po to, kai ras tokią žymą **<html>**

```
<html>
```

Dokumento pabaigoje turite užbaigti HTML kodą surinkdami pabaigos žymą

```
</html>
```

### 1 žingsnis

„Notepad“ dokumente pradėkite rinkti kodą, kuris turėtų atrodyti taip

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
</html>
```

**HTML** dokumentą sudaro dvi dalys:

1. **antraštė**
2. **pagrindinė dalis**

**Antraštė** naudojama konkrečiai informacijai, susijusiai su interneto tinklalapiu, rodyti, o **pagrindinėje dalyje** nurodomas turinys, kurį norite pateikti skaitytojams.

### 2 žingsnis

Iškart po pradžios žymos **<html>** įterpkime **antraštės** sekciją, kaip parodyta toliau:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
</html>
```

### 3 žingsnis

Dabar suteikime tinklalapiui **pavadinimą** kaip parodyta toliau:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head><title>Lietuvos technologijos institutas</title>
```

```
</html>
```

### 4 žingsnis

Dabar baikite **antraštės** sekciją ir pradėkite **pagrindinės dalies** sekciją:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head><title>Lietuvos technologijos institutas</title></head>
```

```
<body>
```

```
</html>
```

## 5 žingsnis

Surinkite pirmąjį tinklalapio pastraipą:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head><title>Lietuvos technologijos institutas</title></head>
```

```
<body>
```

```
Lietuvos technologijos institutas yra geriausia GIS mokykla šalyje.
```

```
</html>
```

## 6 žingsnis

Norėdami užbaigti tinklalapį, **pagrindinės dalies** sekciją turime baigti žyma **</body>**:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head><title>Lietuvos technologijos institutas</title></head>
```

```
<body>
```

```
Lietuvos technologijos institutas yra geriausia GIS mokykla šalyje.
```

```
</body>
```

```
</html>
```

Atlikus visus šešis žingsnius, jūsų HTML kodas dabar turėtų atrodyti taip:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head><title>Lietuvos technologijos institutas</title></head>
```

```
<body>
```

```
Lietuvos technologijos institutas yra geriausia GIS mokykla šalyje.
```

```
</body>
```

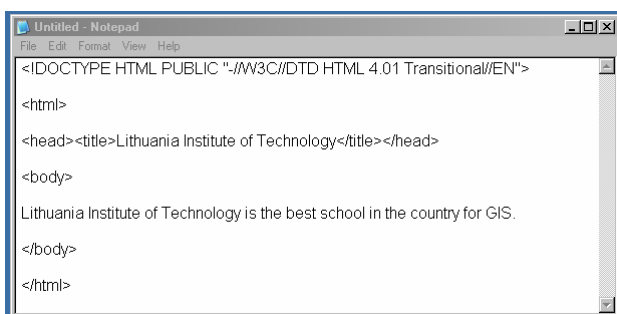
```
</html>
```

### Atkreipkite dėmesį:

Tekstas buvo **paryšintas**, tarpai tarp eilučių ir protarpiai buvo įterpti į pirmiau nurodytą kodą tik tam, kad jį būtų patogiau perskaityti.

### 1.2.3 HTML dokumento įrašymas ir peržiūrėjimas

Jūs ką tik sukūrėte savo pirmąjį paprastą tinklalapį, tačiau procesui užbaigti būtina įrašyti failą. Rengyklėje „Notepad“ turėsite surinkti tokį kodą:



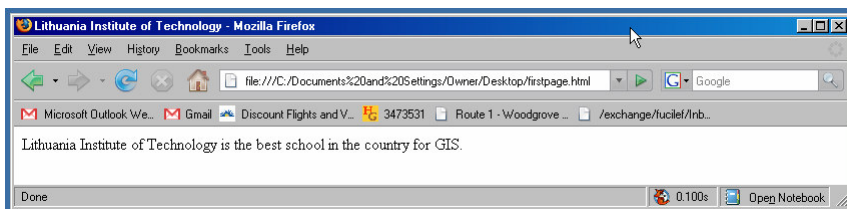
Norėdami įrašyti failą, laikykitės šių nurodymų:

- Viršutinėje parankinėje spragtelėkite parinktį **File (Failas)** ir pasirinkite **Save As (Įrašyti kaip)**.
- Failą įrašykite kaip „tekstinį“ dokumentą ir suteikite jam vardą: **firstpage.html**
- Įrašykite jį į „Windows“ **darbalaukį**.



- Atverkite naršyklę („Internet Explorer“, „Netscape“, „Mozilla“ ar bet kurią kitą turimą naršyklę). Šiame pavyzdyje mano interneto naršyklė bus „Firefox“.
- Viršutinėje parankinėje spragtelėkite parinktį **File (Failas)** ir pažymėkite **Open File (Atverti failą)**.
- Kitame ekrane pasirinkite failą **firstpage.html** darbalaukyje:

Dabar turėtumėte pamatyti, kad jūsų tinklalapis naršyklėje atrodo taip:



Panagrinėkime keletą jūsų pirmojo paprasto tinklalapio aspektų:

- Naršyklės **pavadinimo juostoje pavadinimas** yra per aukštai tinklalapio viršuje – Lietuvos technologijos institutas
- Vienintelis dalykas, esantis svetainės **pagrindinėje dalyje**, yra vienas sakiny, kurį surinkote
  - Lietuvos **technologijos institutas yra geriausia GIS mokykla šalyje**.
- Vietos juostoje rodomas jūsų kompiuterio, kuris yra vietinis pagrindinis kompiuteris, adresas, kadangi failas yra jūsų kompiuterio darbalaukyje

## 1.2.4 Kitos žymos

Tęsdami pradėtą darbą, tinklalapį papildykite kitomis žymomis, kurios jį padarys įdomesnį ir lengviau skaitomą.

**Antraštės** žyma

**Antraštės** žyma nurodo naršyklei padidinti ir paryškinti šriftą. Dauguma naršyklių gali suprasti 1–6 antraštės žymas. Po žymos <body> ir prieš sakinio pradžią

įterpkite šias antraštės žymas:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
```

```

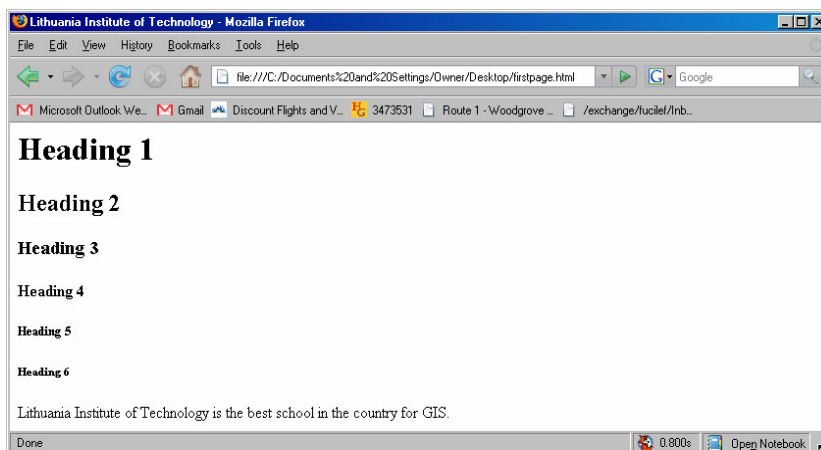
    <head><title>Lietuvos technologijos institutas</title></head>
    <body>
    <h1>1 antraštė</h1>
    <h2>2 antraštė</h2>
    <h3>3 antraštė</h3>
    <h4>4 antraštė</h4>
    <h5>5 antraštė</h5>
    <h6>6 antraštė</h6>
```

Lietuvos technologijos institutas yra geriausia GIS mokykla šalyje.

```
</body>
</html>
```

Norėdami peržiūrėti naujas antraštės žymas, įrašykite failą **firstpage.html**. Grįžkite į naršyklę ir spragtelėkite mygtuką „Refresh“ (Atnaujinti) arba „Reload“ (Įkelti iš naujo) ekrano viršuje, kaip parodyta paveiksle.

Dabar turėtumėte matyti tokį tinklalapį. Atkreipkite dėmesį į santykinį antraštės žymų dydį – antraštė H1 yra didžiausia, o antraštė H6 – mažiausia.



## Paryškinimo ir kursyvo žymos

Atskirus žodžius arba frazes galite akcentuoti apgaubdami juos žymomis **<b>** arba **<i>**. Pamėginkite šias žymas įrašyti iš abiejų kai kurių pastraipos žodžių pusių:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
```

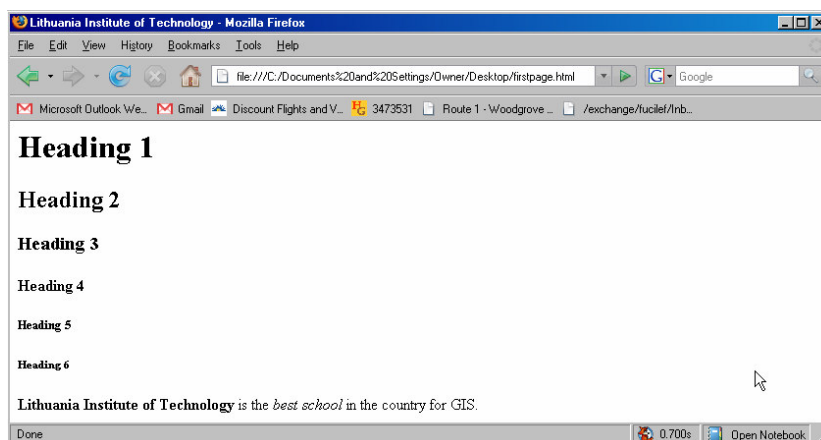
```

    <head><title>Lietuvos technologijos institutas</title></head>
    <body>
    <h1>1 antraštė</h1>
    <h2>2 antraštė</h2>
    <h3>3 antraštė</h3>
    <h4>4 antraštė</h4>
    <h5>5 antraštė</h5>
    <h6>6 antraštė</h6>
```



```
<b>Lietuvos technologijos institutas</b> yra <i>geriausia</i> GIS
<i>mokykla</i> šalyje.
```

```
</body>
</html>
```



## Pastraipos ir horizontaliosios juostos žymos

Tinklalapį apačioje papildykime horizontaliaja juosta. Visų pirma abiejose sakinio pusėse įrašykite pastraipos žymas **<p>**, taip atsiras daugiau vietos aplink sakinį. Pabaigoje įrašykite horizontaliosios juostos žymą **<hr>**. Horizontaliosios juostos žymai nereikia naudoti pradžios ir pabaigos žymų.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
```

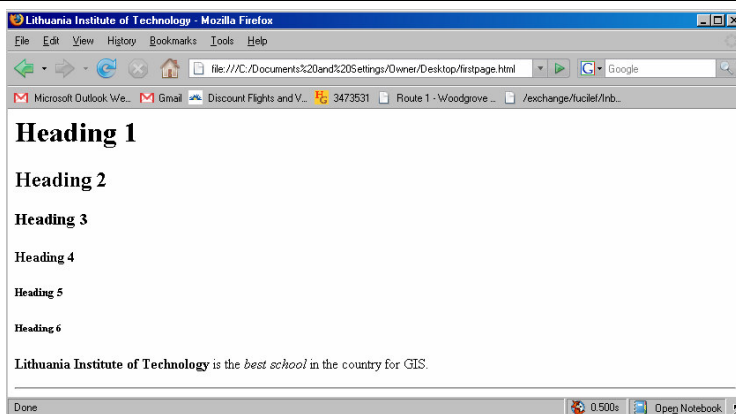
```

    <head><title>Lietuvos technologijos institutas</title></head>
    <body>
    <h1>1 antraštė</h1>
    <h2>2 antraštė</h2>
    <h3>3 antraštė</h3>
    <h4>4 antraštė</h4>
    <h5>5 antraštė</h5>
    <h6>6 antraštė</h6>
```

```
<p><b>Lietuvos technologijos institutas</b> yra <i>geriausia</i> GIS
<i>mokykla</i> šalyje.</p>
```

```
<hr>
</body>
</html>
```

Norėdami peržiūrėti naujas žymas, įrašykite failą **firstpage.html**. Grįžkite į naršyklę ir spragtelėkite mygtuką „Refresh“ (Atnaujinti) arba „Reload“ (Įkelti iš naujo) ekrano viršuje, kaip parodyta paveiksle.



## 1.2.5 Sudėtingesnės žymos

Ankstesniuose pavyzdžiuose susipažinote su keliomis pagrindinėmis žymomis ir jų naudojimu. Yra daugybė kitų HTML žymų, kurias galite naudoti kurdami tinklalapius ir svetaines. Visai nebūtina atsiminti arba mintinai išmokti visas žymas, kurias galite naudoti. Galite tiesiog pasinaudoti žymų „rodykle“, kurią sudarė W3C – <http://www.w3.org/> (anglų kalba).

Pasaulinio interneto konsorciumas (W3C) kuria sąveikias technologijas (specifikacijas, gaires, programinę įrangą ir priemones), kurios leidžia išnaudoti visą interneto potencialą. W3C – tai informacijos, bendravimo, ryšių ir kolektyvinių žinių forumas. Toliau pateikta nuoroda į visas W3 organizacijos HTML žymas, kurias galite naudoti kurdami tinklalapius.

<http://www.w3.org/TR/html401/index/elements.html>

## 1.2.6 Žymų požymiai

Norint išplėsti žymų galimybes, joms galima priskirti tam tikrus **požymius**. Žymų požymiai įrašomi skliaustuose ir nustatoma konkreti jų reikšmė; požymiai rašomi kabutėse.

### Lygiavimo požymiai

Lygiavimo požymiai nurodo teksto padėtį konkrečioje tinklalapio vietoje. Pavyzdžiui:

- `<p align="center">` – pastraipa lygiuojama tinklalapio centre.
- `<p align="right">` – pastraipa lygiuojama ekrano dešinėje.

Atkreipkite dėmesį į požymių formatą:

1. Pirma – rašomas požymio pavadinimas: `<p align="center">`
2. Antra – dedamas lygybės ženklas: `<p align="center">`
3. Trečia – kabutėse įrašoma norima reikšmė: `<p align="center">`

Atkreipkite dėmesį, kad **NĖRA** jokių tarpelių: `align="center"`.

Tačiau tarpelis paliekamas tarp žymos ir požymio: `<p align="center">`

### Pločio ir dydžio požymiai

Tam tikroms žymoms galima priskirti pločio ir dydžio požymius.

Pavyzdžiui, horizontaliosios juostos dydį galite pakeisti priskirdami pločio ir dydžio

požymius. Atkreipkite dėmesį, kad visos požymių reikšmės rašomos kabutėse:

```
<hr align="center" width="400" size="5">
```

Šiuo atveju ekrane eilutė bus 400 vaizdo taškų pločio (15 colių ekranas yra maždaug 700 vaizdo taškų pločio) ir 5 vaizdo taškų storio.

Įvairūs požymiai gali būti priskiriami žymoms bet kuria tvarka:

```
<hr align="center" width="500" size="20"> arba
```

```
<hr width="500" align="center" size="20">
```

Pabandykite priskirti požymius. Atverkite teksto rengyklę ir faile **firstpage.html** išbandykite įvairius derinius.

Pamėginkite nurodyti skirtingus horizontaliosios juostos dydžius ir lygiuotes:

```
<p><b>Lietuvos technologijos institutas</b> yra <i>geriausia</i> GIS  
<i>mokykla</i> šalyje.</p>
```

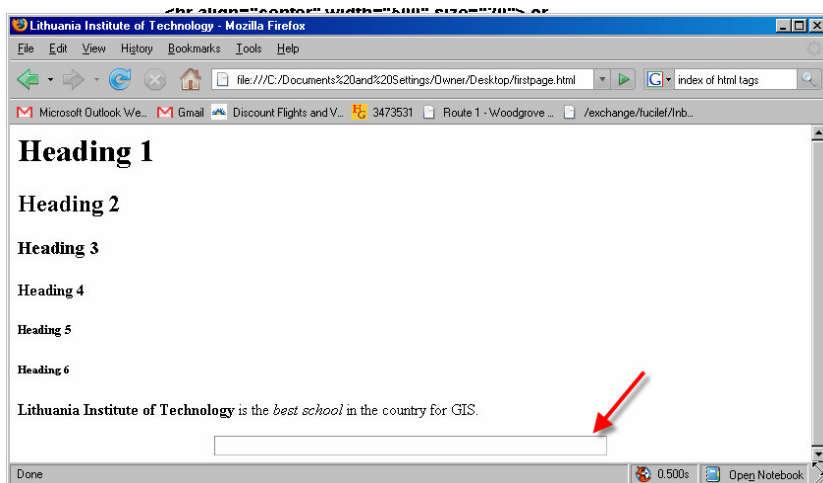
```
<hr align="center" width="400" size="20">
```

```
</body>
```

```
</html>
```

Atsiminkite, kad norėdami pamatyti, kaip pasikeitė tinklalapis pakeitus kodą, turite:

1. **Įrašyti** failą **firstpage.html**.
2. **Atverti** naršyklę („Explorer“, „Netscape“, „Mozilla“) ir **atverti** failą **firstpage.html**



## Spalvos požymiai

Naudodami požymius taip pat galite parinkti tinklalapio spalvas. Visų pirma turite susirasti specialius kodus, vadinamuosius **šešioliktainius kodus**, kurie žymi atitinkamas spalvas. Šioje svetainėje išvardyti visi šešioliktainiai spalvų kodai: [http://www.webmonkey.com/reference/color\\_codes/](http://www.webmonkey.com/reference/color_codes/).

Turėdami visų kodų sąrašą, galite tiesiog nukopijuoti ir įdėti tą kodą, kurį norite naudoti.

Hex Code	Color	Hex Code	Color	Hex Code	Color
#FFFFFF		#CCFFFF		#99FFFF	
#FFFFCC		#CCFFCC		#99FFCC	
#FFFF99		#CCFF99		#99FF99	
#FFFF66		#CCFF66		#99FF66	
#FFFF33		#CCFF33		#99FF33	
#FFFF00		#CCFF00		#99FF00	
#FFCCFF		#CCCCFF		#99CCFF	
#FFCCCC		#CCCCCC		#99CCCC	
#FFCC99		#CCCC99		#99CC99	
#FFCC66		#CCCC66		#99CC66	
#FFCC33		#CCCC33		#99CC33	

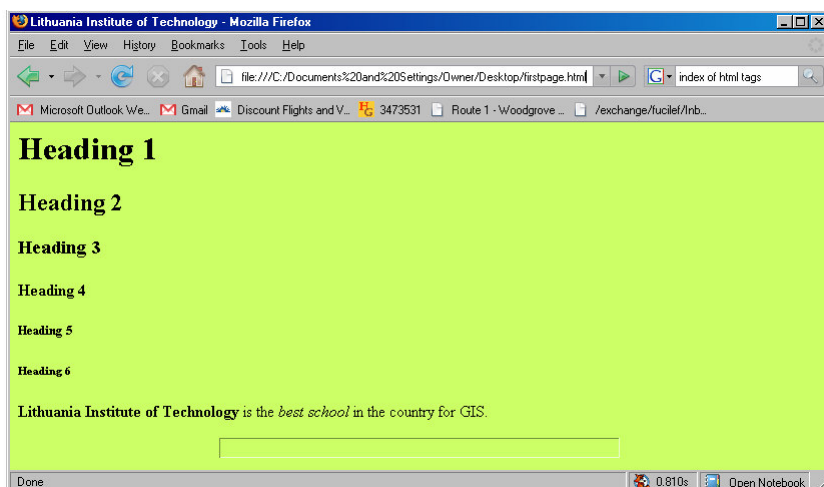
## Fono spalva

Tinklalapio fonui reikia naudoti neryškia šviesią spalvą, kad tekstas būtų aiškiai matomas ir lengvai įskaitomas. Pavyzdžiui, #CCFF66 yra neryški šviesiai geltona spalva, ant kurios aiškiai matomas tamsus tekstas.

```
<body bgcolor="#CCFF66">
```

Atkreipkite dėmesį, kad tarp požymio **bgcolor**, lygybės ženklo ir reikšmės nepaliekami tarpeliai. Kaip visuomet, reikšmė nurodoma kabutėse. Jeigu įterpiamas **spalvos** požymis, jo reikšmė visuomet prasideda skaičiaus simboliu (#).

Pamėginkite atitinkamai pakeisti žymą **<body>**. **Įrašykite** tinklalapį ir **įkelkite jį iš naujo** į naršyklę. Patikrinkite, ar jums patinka fono spalva.

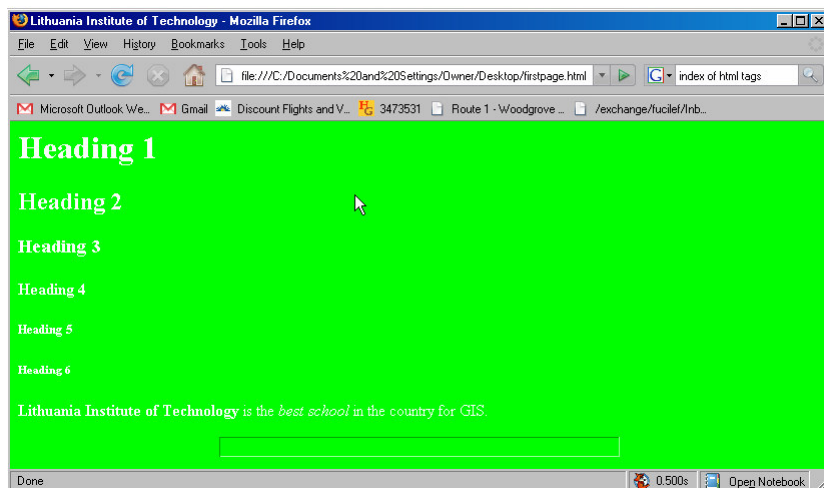


## Teksto spalva

Naudodami tuos pačius kodus galite keisti ir teksto spalvas:

```
<body bgcolor=#00ff00 text="#ffffff">
```

Vėl pasikeis fono spalva, taip pat pasikeis ir viso teksto spalva. Norėdami pamatyti šiuos pakeitimus, **įrašykite failą ir jį įkelkite iš naujo.**



## Nuorodos spalva

Taip pat galite keisti hiperteksto nuorodų spalvą, o visą šią informaciją nurodyti toje pačioje žymoje **<body>**, kaip parodyta toliau:

```
<body bgcolor="#FFCCCC" text="#000000" link="#990000" alink="#009900" vlink="#330033">
```

Ką nurodo žymos **<body>** požymiai?

- **text** = pagrindinio teksto spalva tinklalapyje
- **link** = nuorodos teksto spalva tinklalapyje
- **vlink** = aplankytos nuorodos spalva
- **alink** = kol vyksta aktyvinimas, nuorodos tekstas mirksi kita spalva.

Pagal pateikto pavyzdžio kodą:

- bus parinkta juoda teksto spalva
- nuorodos bus šviesiai raudoni
- aktyvinta nuoroda bus tamsiai žalia
- aplankyta nuoroda bus tamsiai violetinė.

Kadangi šiuo metu tinklalapyje nėra nė vienos nuorodos, šių nuorodų spalvų pasikeitimų nepamatysite. Su nuorodomis išsamiau susipažinsime 3 temoje.

Be to, visai nesvarbu, ar spalvos reikšmes nurodysite DIDŽIOSIOMIS ar mažosiomis raidėmis; svarbiausia nepamišti simbolio **#** ir kabučių iš abiejų pusių. Pavyzdžiui:

```
bgcolor="#FFCCCC" yra tas pats, kas ir bgcolor="#ffcccc"
```

O ką daryti, jeigu nenorite keisti viso teksto spalvos? Jūs norite pakeisti tik pirmosios antraštės spalvą – galbūt pasirinkti ryškiai raudoną.

Visų pirma, pakeiskite žymos **<body>** spalvą į neryškia ir ištrinkite **teksto** požymį:

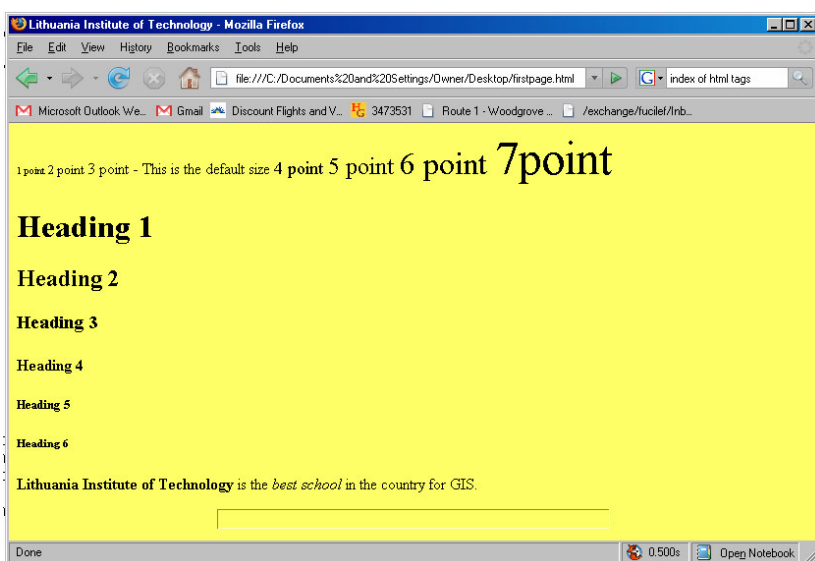
```
<body bgcolor="#ffff66">
```

### Šrifto požymiai

Norint pakeisti teksto dydį, reikia naudoti šrifto požymius. Naujausiose HTML versijose (HTML 4 ir XHTML) atsisakoma pirminio būdo šrifto dydį nustatyti naudojant šrifto žymą. HTML elementų išdėstymo ir rodymo savybėms apibrėžti bus naudojami stiliaus lapai (CSS), su kuriais išsamiau susipažinsite 8 temoje.

```
<font size="1">1 punktas</font>
<font size="1">2 punktai</font>
<font size="1">3 punktai</font>
<font size="1">4 punktai</font>
<font size="1">5 punktai</font>
<font size="1">6 punktai</font>
<font size="1">7 punktai</font>
```

Failo firstpage.html kodą papildykite šrifto dydžio požymiais ir pažiūrėkite, kaip tai atrodo. Pamatysite, koks yra santykinis šrifto dydis.



### 1.2.7 Sudarykime sąrašą

Žmonės tinklalapius skaito kitaip nei laikraščio ar žurnalo straipsnius. Interneto skaitytojai greitai akimis permeta tinklalapį ir nesigilina į informaciją. Akys pastebi didesnes paryškintas antraštes ir punktais suskirstytus temų sąrašus. Taip pat svarbu paminėti, kad dauguma žmonių kompiuterio ekrane skaito 33–66 proc. lėčiau nei skaitytų popieriaus lapą. Jie perskaito išsiskiriantį tekstą, kadangi jiems taip lengviau. Tai paaiškina, kodėl yra tiek daug skirtingų būdų sąrašams programuoti.

- Yra suženklintų sąrašų kaip šis.
- Kiekvienas punktas pradedamas nedideliu simboliu.

- Tai yra nerūšiuotas sąrašas = `<ul>`
  1. Yra sunumeruotų sąrašų kaip šis.
  2. Kiekvienas punktas pradedamas skaitmeniu arba raide.
  3. Tai yra rūšiuotas sąrašas = `<ol>`

Taip pat yra apibrėžti sąrašai kaip šis. Kiekvienas punktas yra atitrauktas, tačiau nežymimas jokių simboliu ar skaičiumi.

Tai yra apibrėžtas sąrašas = `<dl>`

### Nerūšiuotas sąrašas

Nerūšiuoto sąrašo kode punktai išdėstomi iš eilės ir vėliau ženklelio paveikslui galite priskirti įvairius tipo požymius.

HTML kodas	Naršyklė
<pre>&lt;ul&gt;   &lt;li&gt;Pirmas punktas.&lt;/li&gt;   &lt;li&gt;Antras punktas.&lt;/li&gt; &lt;/ul&gt;</pre>	<ul style="list-style-type: none"> <li>• Pirmas punktas</li> <li>• Antras punktas</li> </ul>

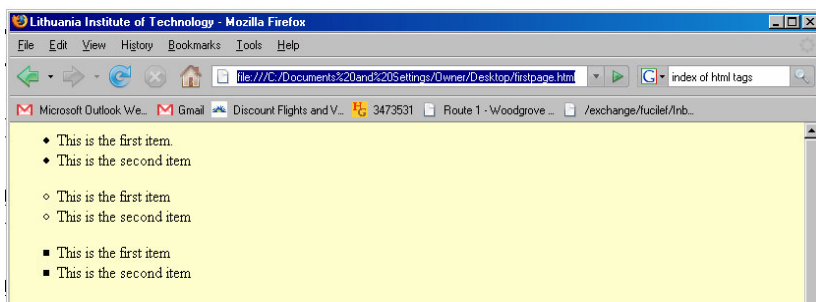
Atkreipkite dėmesį, kad visų pirma atveriamą žymą `<ul>`, paskui – žymą `<li>`, o pabaigoje užveriamą žymą `</li>`, paskui – žymą `</ul>`. Tvarka visada yra tokia pati – **pirmas į sistemą padėtas elementas paimamas paskutinis** (angl. *first in, last out*). Šios žymos vadinamos **įdėtinėmis žymomis**.

Juodą apvalų ženklelį galite pakeisti į kvadratą arba tuščiaidurį apskritimą **sąrašo** žymą papildydami **tipo** požymiu:

HTML kodas	Naršyklė
<pre>&lt;ul type="circle"&gt;   &lt;li&gt;Pirmas punktas&lt;/li&gt;   &lt;li&gt;Antras punktas&lt;/li&gt; &lt;/ul&gt;</pre>	<ul style="list-style-type: none"> <li>○ Pirmas punktas</li> <li>○ Antras punktas</li> </ul>

HTML kodas	Naršyklė
<pre>&lt;ul type="square"&gt;   &lt;li&gt;Pirmas punktas&lt;/li&gt;   &lt;li&gt;Antras punktas&lt;/li&gt; &lt;/ul&gt;</pre>	<ul style="list-style-type: none"> <li>□ Pirmas punktas</li> <li>□ Antras punktas</li> </ul>

Nerūšiuoto sąrašo kodą surinkite faile firstpage.html ir peržiūrėkite jį naršyklėje. Jis turėtų atrodyti taip:



## Rūšiuotas sąrašas

Rūšiuoto sąrašo kode punktai išdėstomi tam tikra seka, taip pat naudojami skirtingų rūšių ženkleliai. Pavyzdžiui:

HTML kodas	Naršyklė
<pre>&lt;ol&gt;   &lt;li&gt;Miltus suberkite į dubenį&lt;/li&gt;   &lt;li&gt;Į dubenį su miltais įpilkite vandens&lt;/li&gt;   &lt;li&gt;Išmaišykite laikydamiesi nurodymų&lt;/li&gt; &lt;/ol&gt;</pre>	<ol style="list-style-type: none"> <li>1. Miltus suberkite į dubenį</li> <li>2. Į dubenį su miltais įpilkite vandens</li> <li>3. Išmaišykite laikydamiesi nurodymų</li> </ol>

Skaičių galite pakeisti raidėmis **sąrašo** žymą papildę **tipo** požymiu:

HTML kodas	Naršyklė
<pre>&lt;ol type="A"&gt;   &lt;li&gt;Miltus suberkite į dubenį&lt;/li&gt;   &lt;li&gt;Į dubenį su miltais įpilkite vandens&lt;/li&gt;   &lt;li&gt;Išmaišykite laikydamiesi nurodymų&lt;/li&gt; &lt;/ol&gt;</pre>	<ol style="list-style-type: none"> <li>A. Miltus suberkite į dubenį</li> <li>B. Į dubenį su miltais įpilkite vandens</li> <li>C. Išmaišykite laikydamiesi nurodymų</li> </ol>

HTML kodas	Naršyklė
<pre>&lt;ol type="i"&gt;   &lt;li&gt;Miltus suberkite į dubenį&lt;/li&gt;   &lt;li&gt;Į dubenį su miltais įpilkite vandens&lt;/li&gt;   &lt;li&gt;Išmaišykite laikydamiesi nurodymų&lt;/li&gt; &lt;/ol&gt;</pre>	<ol style="list-style-type: none"> <li>i. Miltus suberkite į dubenį</li> <li>ii. Į dubenį su miltais įpilkite vandens</li> <li>iii. Išmaišykite laikydamiesi nurodymų</li> </ol>

Jeigu norite sąrašą pertraukti ir įterpti pastabas arba nurodymus, sąrašą galite pradėti nuo pasirinkto skaičiaus arba raidės. Tam naudojamas **pradžios** požymis. Atkreipkite dėmesį, kad norėdami nurodyti trečią abėcėlės raidę turite rašyti skaičių **3**.



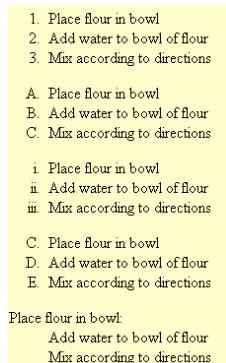
HTML kodas	Naršyklė
<pre>&lt;ol type="A" start="3"&gt;   &lt;li&gt;Miltus suberkite į dubenį&lt;/li&gt;   &lt;li&gt;Į dubenį su miltais įpilkite vandens&lt;/li&gt;   &lt;li&gt;Išmaišykite laikydamiesi nurodymų&lt;/li&gt; &lt;/ol&gt;</pre>	<p>C. Miltus suberkite į dubenį  D. Į dubenį su miltais įpilkite vandens  E. Išmaišykite laikydamiesi nurodymų</p>

## Apibrėžtas sąrašas

Apibrėžto sąrašo kodas atrodo taip:

HTML kodas	Naršyklė
<pre>&lt;dl&gt;   &lt;dt&gt;Miltus suberkite į dubenį:   &lt;dd&gt;Į dubenį su miltais įpilkite vandens&lt;/dd&gt;   &lt;dd&gt;Išmaišykite laikydamiesi nurodymų&lt;/dd&gt; &lt;/dl&gt;</pre>	<p>Miltus suberkite į dubenį:    Į dubenį su miltais įpilkite vandens    Išmaišykite laikydamiesi nurodymų</p>

Šį sąrašo kodą surinkite faile **firstpage.html** ir pažiūrėkite, kaip jis atrodo naršyklėje. Jis turėtų atrodyti taip:



## Kodėl naudojami sąrašai

1. Sąrašus lengviau naudoti, nei kurti sąrašų imitacijas patiems įrašant skaičius, kadangi jūs bet kuriuo metu sąrašą galite papildyti dar vienu punktu, o naršyklė automatiškai iš naujo sunumeruos visus punktus.
2. Jeigu yra sąrašas, tai reiškia, kad kiekvieno sąrašo punkto tekstas yra **atitrauktas** nuo paraštės. Jeigu sąrašas sudaromas ne automatiškai, ilgi sakiniai arba pastraipos apgaubia skaičių ir neišlieka atitraukti nuo paraštės.
3. Kai **pradžios** požymis naudojamas su rūšiuotais sąrašais, galite bet kurioje vietoje pertraukti sąrašą jį sustabdydami, įterpdami komentarus ir vėl pradėdami sąrašą nuo atitinkamo skaičiaus arba raidės.
4. Sąraše galite naudoti kitas **HTML** žymas – **pastraipos** žymas norėdami atskirti sąrašo punktus arba **paryškimo** ir **kursyvo** žymas norėdami akcentuoti tekstą.

## 1.3. Nuorodos

### 1.3.1 Nuorodų formatas

Internetas yra toks populiarus dėl to, kad yra naudojamos nuorodos, sukuriančios „hiperteksto“ pasaulį. Jie didina interneto efektyvumą ir palengvina naudojimąsi juo. Nuorodoms naudojamos **pririšimo** žymos. Norėdami sukurti nuorodą į svetainę, iš abiejų nuorodos teksto pusių įterpkite **pririšimo** žymas. Taikykite tokią formulę:

- Pradėkite ir baikite pagrindine pririšimo žyma: **<a>..... </a>**
- Pradžios žymą papildykite požymiu **href** – hipersaito nuoroda. Kabutėse įrašyta reikšmė nurodys URL, prie kurio norite jungtis

**<a href="http://www.maps.lt/">..... </a>**

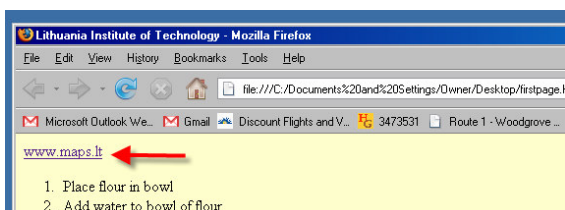
- Tarp pradžios ir pabaigos žymų surinkite žodžius, kurie bus rodomi kaip hipersaitas. Pavyzdžiui:

**<a href="http://www.maps.lt/">www.maps.lt</a>**

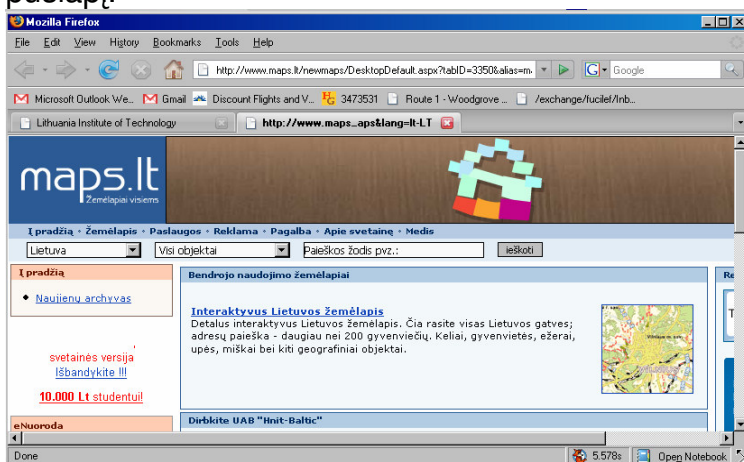
Atkreipkite dėmesį, kad tarp požymio **href** ir paskutinio pabaigos žymos **</a>** skliausto nėra palikta jokių tarpelių.

Be to, URL nurodomas kabutėse kaip ir visi kiti požymiai. Tinklapyje firstpage.html surinkite nuorodos kodą.

Sukursite nuorodą puslapyje, kuris atrodys taip: **www.maps.lt**. Paprastai tekstas bus mėlynos spalvos (jeigu nebuvo pakeista nuorodos spalva) ir pabrauktas.



Pelės mygtuku spragtelėjus ant nuorodos, naršyklė eis į <http://www.maps.lt/> pradinį puslapį.



### 1.3.2 Nuorodų sąrašai

Kaip parodyta toliau, sąrašo punktus galite papildyti nuorodomis iš abiejų sąrašo punktų pasiūlydami pririšimo žymą `<a>`. Pateikiame keletą pavyzdžių:

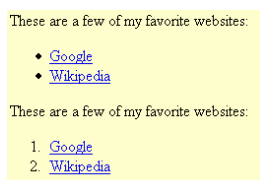
#### Nerūšiuotas nuorodų sąrašas

HTML kodas	Naršyklė
<pre>&lt;p&gt;Štai keletas mano mėgstamiausių svetainių: &lt;ul&gt;   &lt;li&gt;&lt;a href="http://google.com"&gt;Google&lt;/a&gt;&lt;/li&gt;   &lt;li&gt;     &lt;a href="http://www.wikipedia.org"&gt;Wikipedia&lt;/a&gt;&lt;/li&gt; &lt;/ul&gt; &lt;/p&gt;</pre>	<p>Štai keletas mano mėgstamiausių svetainių:</p> <p><a href="http://google.com">Google</a></p> <p><a href="http://www.wikipedia.org">Wikipedia</a></p>

#### Rūšiuotas nuorodų sąrašas

HTML kodas	Naršyklė
<pre>&lt;p&gt;Štai keletas mano mėgstamiausių svetainių:   &lt;ol&gt;   &lt;li&gt;&lt;a href="http://google.com"&gt;Google&lt;/a&gt;&lt;/li&gt;   &lt;li&gt;     &lt;a href="http://www.wikipedia.org"&gt;Wikipedia&lt;/a&gt;&lt;/li&gt;   &lt;/ol&gt; &lt;/p&gt;</pre>	<p>Štai keletas mano mėgstamiausių svetainių:</p> <p><a href="http://google.com">Google</a></p> <p><a href="http://www.wikipedia.org">Wikipedia</a></p>

Šį kodą surinkite faile firstpage.html, įrašykite failą ir iš naujo įkelkite į naršyklę. Turėtumėte matyti tokį nuorodų sąrašą:



### 1.3.3 Paskirtys

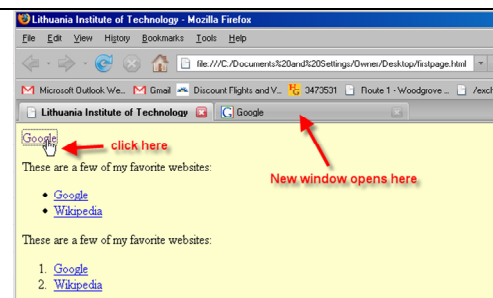
Dar vienas požymis, kurį galite priskirti žymoms `<a>`, yra `target="_blank"`

Šiuo atveju naršyklė atvers naują puslapį naujame lange. Pirminis puslapis liks atidarytas pirmajame lange. Visas kodas yra toks:

```
<a href="http://www.google.com/" target="_blank">Google</a>
```

Vienas tarpelis paliktas tarp URL pabaigos kabutės ir žodžio „target“.

Nurodytą kodą surinkite faile **firstpage.html**, įrašykite failą ir iš naujo įkelkite į naršyklę. Dabar turėtumėte matyti nuorodą į „Google“, o spragtelėjus ant nuorodos „Google“, puslapis turėtų būti atvertas kitame lange.



### 1.3.4 Prieraišai

Nuorodas taip pat galima susieti su konkrečiomis svetainės arba tinklalapio vietomis. Jeigu puslapiai yra labai ilgi, tokios nuorodos skaitytojui padeda patogiau judėti po puslapį.

1. Prieraišai perkelia į kitą to paties puslapio vietą,
2. nuorodos perkelia į vietą už puslapio ribų,
3. tačiau jiems abiemis naudojama žyma <a>.

Prieraišui sukurti reikalingi du žingsniai:

- vienas – nuorodai su konkrečia vieta sukurti
- kitas – tai vietai dokumente pažymėti.

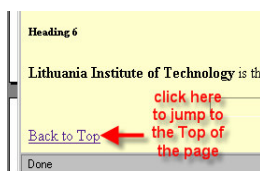
Vienas iš populiariausių pririšimo nuorodų sukurtas tam, kad galėtumėte greitai peršokti į puslapio „viršų“. Visų pirma jį reikia pažymėti skaičiaus simboliu, kaip parodyta toliau pateiktame kode. Šį kodą įterpkite prieš pagrindinės dalies pabaigos žymą:

```
<a href="#top">Grįžti į viršų</a>
</body>
```

Tada žymą <a name="top"> naudokite tai vietai pažymėti. Žymą </a> užverkite iškart po to, kai ją atvėrėte, kadangi nenorite susieti jokių žodžių. Taip pažymima tik ta vieta, į kurią turi vesti nuoroda. Šį kodą įterpkite iškart po žymos <body>:

```
<body>
<a name="top"></a>
```

Šį kodą surinkite faile firstpage.html, įrašykite failą ir iš naujo įkelkite į naršyklę. Turėtumėte matyti tokį vaizdą. Slinkite žemyn iki puslapio apačios ir pamatysite nuorodą „Grįžti į viršų“:



### 1.3.5 Santykiniai ir absoliutieji nuorodos

Kai kuriate nuorodas į puslapius, esančius kitoje svetainėje, visuomet naudojate absoliučiąją nuorodą ir nurodote visą kito puslapio URL arba interneto adresą. Tačiau nuorodas į kitą tos pačios svetainės puslapį galite kurti dvejopai.

Pavyzdžiui, nuorod į kontaktų puslapį tiesiog nurodoma kaip **contactus.html**. Kodas užrašomas taip:

```
<a href="firstpage.html">Pirmasis puslapis</a>
```

Tai nėra visas URL arba interneto adresas ir nuoroda veikia tik iš jūsų kompiuterio, kadangi šiuo metu tinklalapis yra jūsų kompiuterio darbalaukyje. Naršyklė jį supranta kaip komandą surasti dokumentą, pavadintą **firstpage.html**, kuris yra tame pačiame kataloge kaip ir ta vieta, kurioje dabar esate. Tokia nuoroda vadinama **santykine**.

Norėdami nurodyti absoliutųjį saitą, turėtumėte rašyti visą tinklalapio adresą, kaip parodyta toliau:

```
<a href="http://www.yourwebsite.com/firstpage.html">Pirmasis puslapis</a>
```

Toks saitas būtų vadinamas **absoliučiaja** nuoroda, kadangi ji nurodo absoliučią, išsamią ir visą URL arba interneto adresą. Naršyklė jį supranta kaip komandą surasti interneto adresą, pavadintą **yourwebsite.com**, o šioje svetainėje rasti dokumentą **firstpage.html**.

Ši mintis jums bus aiškesnė, kai atliksite visą 1 paskaitos užduotį.

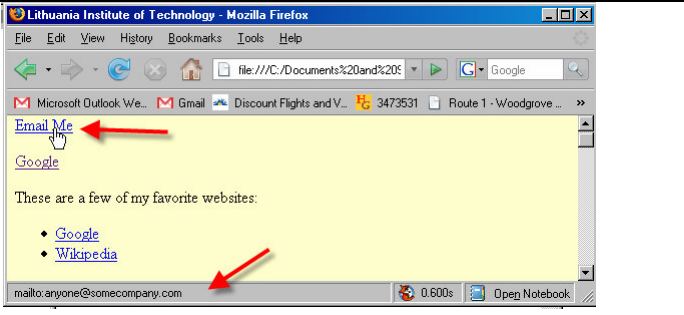
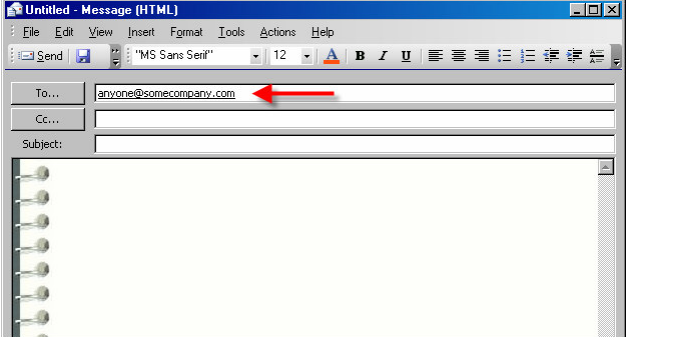
### 1.3.6 Elektroninis paštas

Nuoroda gali nukreipti ne tik į kitas svetaines, bet ir į el. pašto formą, kurios adresato eilutėje jau įrašytas gavėjo el. pašto adresas. Kodas atrodo taip pat, tik:

- vietoje **http://** rašoma **mailto:** (be dešiminių brūkšnių), o
- vietoje URL įrašomas el. pašto adresas.

Štai kaip tai atrodo:

```
<a href="mailto: anyone@somecompany.com">Siųsti man el. laišką</a>
```

<p>Jeigu tokia eilutė būtų įrašyta dokumente, žodžiai <b>Siųsti man el. laišką</b> būtų paryškinti ir susieti su el. pašto forma, kurios eilutėje būtų nurodytas atitinkamas adresas. Prašome naudoti tikrą el. pašto adresą. Nurodytas el. pašto adresas yra išgalvotas ir pateikiamas tik kaip pavyzdys.</p>	
<p>Spragtelėjus ant nuorodos „Siųsti man el. laišką“, bus atverta skaitytojo el. pašto programa, o programos adresato eilutėje bus įrašytas el. pašto adresas, kaip parodyta paveikslėlyje.</p>	

## 1.4. *Paveikslai*

Sakoma, kad vaizdai iškalbingesni už žodžius, tačiau vaizdus tinklalapiui reikia rinktis atidžiai. Vaizdams įkelti reikia daug daugiau laiko nei žodžiams, o tyrimai parodė, kad tais atvejais, kai tenka laukti ilgiau nei 10 sekundžių, žmonės linkę rinktis kitą svetainę.

Žinodami šią informaciją, puslapį papildykite vaizdais. Visų pirma jums reikalingas pats vaizdas. Internete yra keletas paveikslų galerijų, siūlančių nemokamus paveikslus ir grafiką. Pradžioje keletas naudingų svetainių:

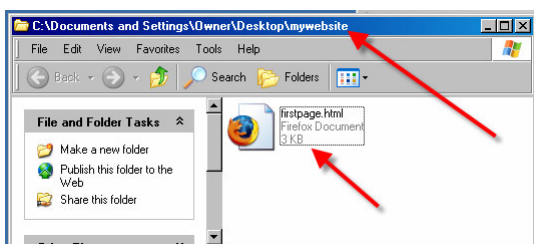
- <http://images.google.com/>
- <http://www.freedigitalphotos.net/>

Atlikdami greitą paiešką „Google“ sistemoje visada rasite daugiau paveikslų svetainių.

### 1.4.1 Paveikslų atsisiuntimas

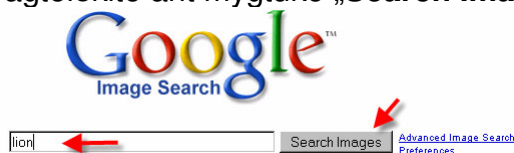
Tam, kad paveikslą galėtumėte įdėti į tinklalapį, turite jį visų pirma atsisiųsti į savo kompiuterį. Prieš atsisiųsdami paveikslus, turite susikurti aplanką, kuriame laikysite visus **HTML** dokumentus ir paveikslų failus.

- Darbalaukyje sukurkite naują aplanką (išskleiskite parinktį „File“ (Failas) viršutinėje horizontaliojoje meniu juostoje ir pasirinkite „New Folder“ (Naujas aplankas). Aplanką pavadinkite mywebpage.
- Į šį aplanką įdėkite dokumentą **firstpage.html**.

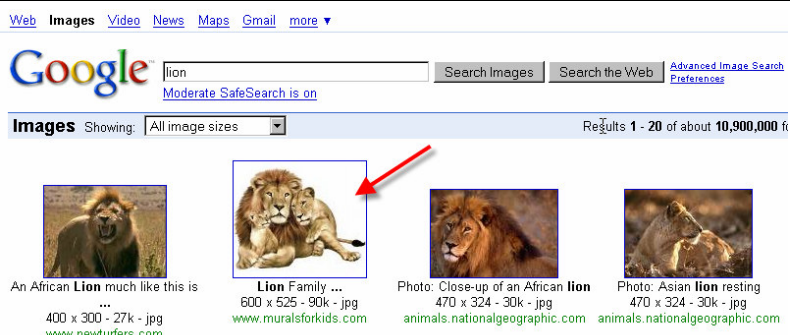


Pasinaudokime „Google Images“ galimybėmis ir susiraskime paveikslą, kurį atsisiųsime. Naršyklėje surinkite tokį svetainės adresą: <http://images.google.com/>.

Šiam pavyzdžiui ieškosiu paveikslo su liūtu. Paieškos teksto langelyje surinkite „liūtas“ ir spragtelėkite ant mygtuko „**Search Images**“ (Ieškoti paveikslų):

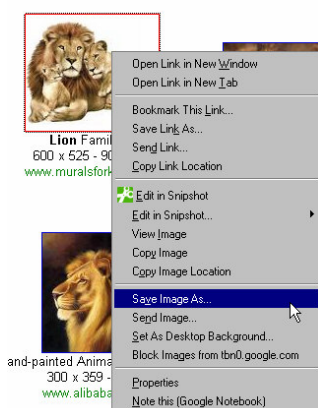


Pamatysite ilgą sąrašą paveikslų, kurie buvo surasti duomenų bazėje. Atsisiųsiu antrąjį iš toliau parodytų paveikslų.

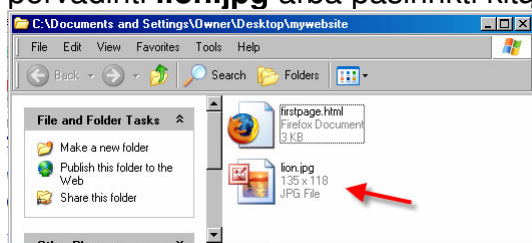


Norėdami atsisiųsti paveikslą, atlikite šiuos veiksmus:

- Dešiniu pelės mygtuku spragtelėkite ant paveikslo.
- Pasirinkite vieną iš parinkčių „**Save Picture As**“ (Išsaugoti vaizdą kaip) arba „**Save Image As**“ (Išsaugoti paveikslą kaip).



- Iškilusiam dialogo lange pasirinkite naują aplanką **mywebsite** failui įrašyti.
- Paveikslą, kurį įrašote, pavadinkite taip, kad lengvai atsimintumėte jo vardą. Kai kurių atsisiųstų paveikslų vardai bus panašūs į 178563Aht.gif, todėl juos geriau pervadinti **lion.jpg** arba pasirinkti kitą žodį, susijusį su paveikslu.



## 1.4.2 Failų vardai ir formatai

Visų paveikslų failų vardai turėtų baigtis .jpg arba .gif. Šiuos du formatus supranta naršyklės. Jeigu jūsų failai yra kito formato, paleiskite juos grafinėje programoje, pavyzdžiui, „Photoshop“ ir pakeiskite formatą.

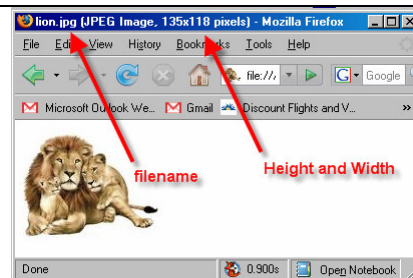
## 1.4.3 Paveikslo dydžio požymiai

Vienas svarbiausių paveikslo požymių yra jo dydis, kurį nusako paveikslo aukštis ir plotis. Šią informaciją galite nesunkiai sužinoti atverdami paveikslą naršyklėje.

Naršyklėje spragtelėkite parinktį **File (Failas)** viršutinėje parankinėje ir pažymėkite **Open File (Atverti failą)**. Tada pasirinkite norimo paveikslo failą. Šiame pavyzdyje atversiu liūto paveikslą, kurį minėjau ankstesniame pavyzdyje.



Pats paveikslas pasirodys naršyklėje, o pavadinimo juostoje bus rodomas paveikslo vardas bei jo matmenys



#### 1.4.4 Paveikslo lygiuotės požymiai

Kadangi paveikslą jau atsisiuntėme, jį įdėkite į failą firstpage.html naudodamiesi toliau nurodytu kodu. Kodą įterpkite po pagrindinės dalies pradžios žymos <body>. Norėdami palikti vietas, po paveikslo įrašykite keletą žymų <p>.

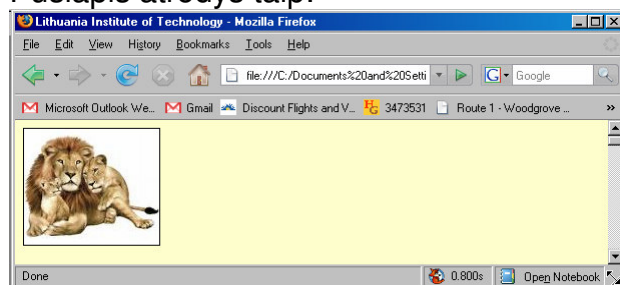
```

```

```
<p>
```

```
<p>
```

Puslapis atrodys taip:

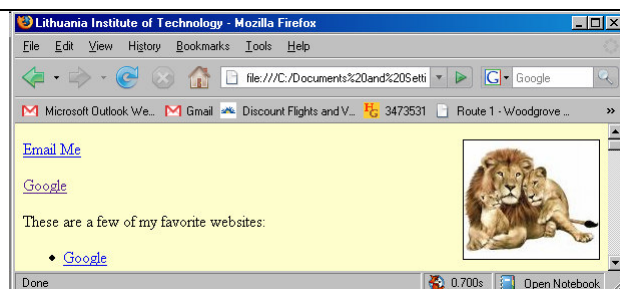


Žymai galite priskirti dešininės lygiuotės požymį – liūtas atsiras dešinėje pusėje:

```

```

Puslapis atrodys taip. Atkreipkite dėmesį, kad paveikslą lygiuojant dešinėje, po juo buvęs tekstas atsiranda šalia paveikslo kairėje.



Kaip matote iš pateikto pavyzdžio, lygiuotės požymis nėra labai patikimas ir itin apsunkina puslapio išdėstymo tvarkymą. Supažindindami jus su lentelėmis parodysime daug veiksmingesnį metodą, kaip lygiuoti paveikslus tinklalapyje.

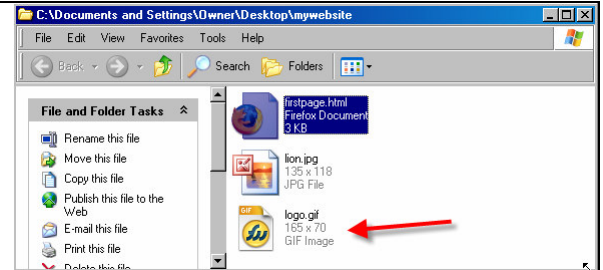
#### 1.4.5 Paveikslų susiejimas

Paveikslus galima naudoti ir kaip nuorodas. Viskas, ką reikia padaryti, tai abiejose paveikslo žymos pusėse įrašyti pririšimo žymas <a>.

Tam pademonstruoti pasitelksime „GIS-Centro“ logotipą. Šį paveikslą naudosime kaip nuorodą į svetainę.



Visų pirma apsilankykime svetainėje <http://www.gis-centras.lt/>, kurios viršutiniame kairiajame kampe yra logotipas. Atsisiųskite logotipo paveikslą į aplaką **mywebsite** laikydamiesi 1.4.1 punkto nurodymų. Logotipo paveikslas turėtų atsirasti čia:



„GIS-Centro“ logotipo kodas atrodo taip:

```

```

Norint sukurti nuorodą, reikia įterpti pririšimo žymas <a>:

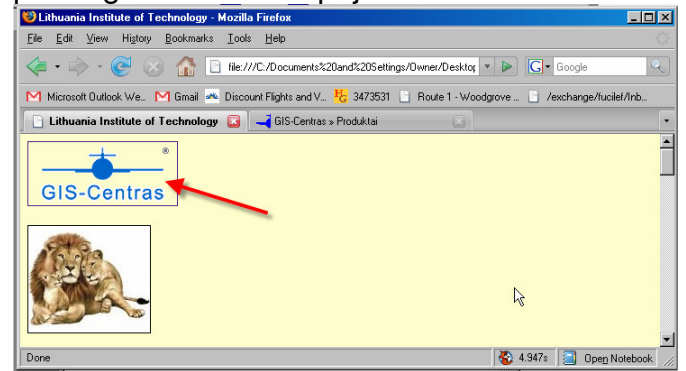
```
<a href="http://www.gis-centras.lt/" target="_blank">
```

```

```

```
</a>
```

Dabar logotipas yra nuoroda, grąžinanti į „GIS-Centro“ pradinį puslapį. Aplink daugumą susietų paveikslų yra kraštinės, kurias automatiškai sukuria naršyklė, atpažinusi žymas <a>. Jeigu jūsų paveikslas toks, kaip liūto paveikslas žemiau, jums gali nepatikti šios kraštinės, nes pradingsta slankumo pojūtis.



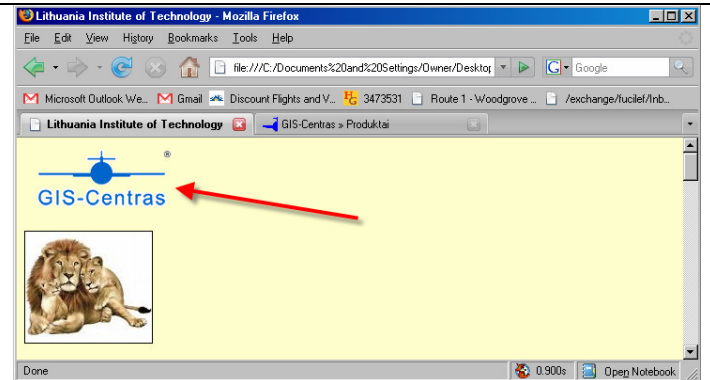
Jeigu norite, kad paveikslas atrodytų slankus, reikia kraštinių požymiui priskirti reikšmę „0“.

```
<a href="http://www.gis-centras.lt/" target="_blank">

```

```
</a>
```





Liūto paveikslo kraštinės reikšmė vis dar yra „1“. Išsamiau su paveikslų kraštinėmis susipažinsime kitoje potemėje.



### 1.4.6 Paveikslo kraštinės

Jeigu nenudysite kraštinės požymio, dauguma naršyklių paveikslus, kurie yra nuorodos, automatiškai įrėmims. Kraštinės požymis yra toks kaip ir visi kiti požymiai, su kuriais jus supažindinome – prieš požymį ir po jo paliekami tarpeliai, pats požymis yra be tarpelių, su lygybės ženklų ir reikšme. Kaip visuomet, reikšmė nurodoma kabutėse.

Kartais, norėdami paveikslui suteikti daugiau išbaigtumo, galite jį įrėminti. Paveikslui praplatinti galite naudoti kraštinės požymį ir pločio vaizdo taškais storio reikšmę. Toliau pateikti paveikslėliai iliustruoja paveikslo kraštinių storumą ir plonumą.

Border="0"	border="3"	border="5"	border="7"
			

### 1.4.7 Fonas iš paveikslų

Paveikslus galima naudoti ir kaip foną. Fono spalvą galite keisti naudodami požymį bgcolor – **<body bgcolor="#ffffff">**

Kodas **<body bgcolor="#ffffff">** sukuria tuščią vienspalvį foną. Tačiau galite paimti ir nedidelį paveikslą bei nurodyti naršyklei dėti jį ekrane tol, kol užsipildys visas ekranas (kaip darbalaukio fonas).

Fono paveikslų nemokamai galite atsisiųsti iš šios svetainės:  
**[http://www.boogiejack.com/free\\_graphics.htm](http://www.boogiejack.com/free_graphics.htm)**  
 I. Maždaug ties svetainės viduriu yra „viso puslapio“ stiliaus fono paveikslų, kuriuos galite atsisiųsti, sąrašas.

Full Page Style Background Tiles and Textures

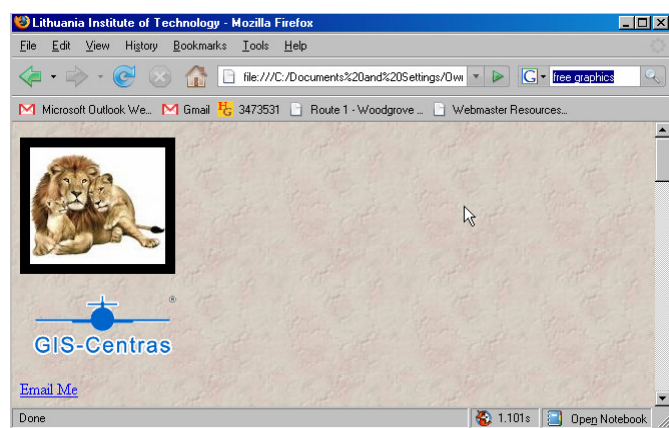
Full Page Backgrounds Page 01	Full Page Backgrounds Page 02
Full Page Backgrounds Page 03	Full Page Backgrounds Page 04
Full Page Backgrounds Page 05	Full Page Backgrounds Page 06
Full Page Backgrounds Page 07	Full Page Backgrounds Page 08
Full Page Backgrounds Page 09	Full Page Backgrounds Page 10
Full Page Backgrounds Page 11	Full Page Backgrounds Page 12
Full Page Backgrounds Page 13	Full Page Backgrounds Page 14
Full Page Backgrounds Page 15	Full Page Backgrounds Page 16
Full Page Backgrounds Page 17	Full Page Backgrounds Page 18

Spragtelėkite ant „**Full Page Backgrounds Page 01**“ ir į aplanką **mywebsite** atsisiųskite antrąjį paveikslą viršutiniame kairiajame kampe **tile\_002.jpg**.

Šį paveikslą įterpkime į dokumentą firstpage.html. Surinkite šį kodą:

**<body background=" tile\_002.jpg">**

Dabar tinklalapis atrodo taip:



Fonus naudoti reikia labai atsargiai. Teksto spalva ir dydis turi būti pakankami, kad tekstas aiškiai matytųsi fone, kitaip gali būti labai sunku įskaityti tekstą ant jūsų pasirinkto fono. Dar vienas populiarus fono išklotinės stilius – tai „kairiosios kraštinės stilius“. Kairioji kraštinė bus kartojama iki pat puslapio apačios. Nepamirškite, kad tekstą ir kitus paveikslus būtina atitraukti nuo kraštinės.

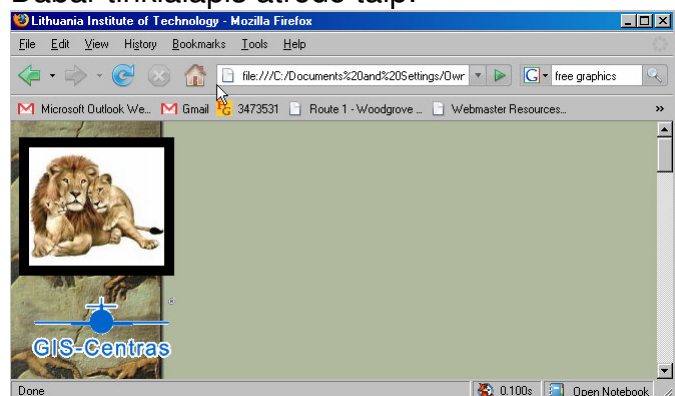
Iš nurodytos svetainės atsisiųskite „kairiosios kraštinės“ paveikslą:

[http://www.boogiejack.com/left\\_border\\_backgrounds\\_page02.html](http://www.boogiejack.com/left_border_backgrounds_page02.html)

Kaip pavyzdį naudosis pirmąjį fono paveikslą bjack006.jpg

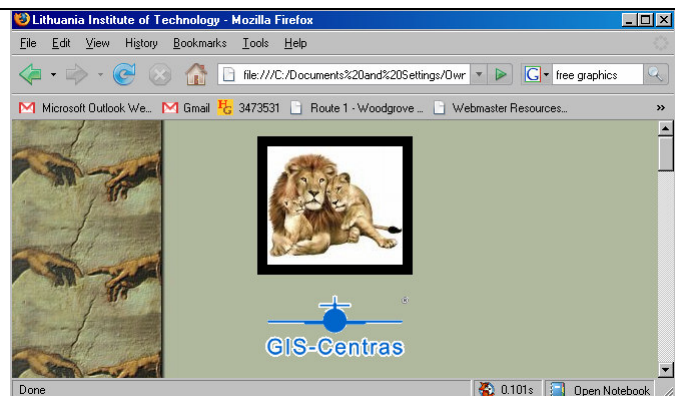
```
<body background=" bjack006.jpg ">
```

Dabar tinklalapis atrodo taip:



Jeigu ketinate naudoti „kairiosios kraštinės“ stilių, rekomenduojame visą tekstą ir objektus lygiuoti tinklalapio centre. Tam tereikia centro žymą <center> įterpti iškart po pagrindinės dalies žymos:

```
<body background=" bjack006.jpg ">
<center>
```









#### 1.4.8 Paveikslai suženklintiems sąrašams

Ankstesnėje temoje sužinojote, kaip kurti sąrašus su iš anksto nustatyta rūšies ženkleliais. Sąrašuose ženklelius galite pakeisti paveikslais naudodami toliau nurodytą kodą.

Visų pirma atsisiųskime ženklelio paveikslą iš čia: <http://www.boogiejack.com/bullet01.html>

Šiame pavyzdyje naudosis „mėlynos rodyklės“ paveikslą **arrow2.gif**

Šiame pavyzdyje nenaudosis fono paveikslo. Patarimas: jeigu norite, kad naršyklė praleistų kodą, prieš žymą arba požymį įrašykite šauktuką !. Šiame pavyzdyje naršyklė praleis fono paveikslą, todėl fonas turėtų būti baltos spalvos.

<pre>&lt;body !background=" bjack006.jpg "&gt; &lt;dl&gt; &lt;dt&gt;Kas žinotina apie paveikslus: &lt;dd&gt;&lt;img alt=" arrow " border="1" src=" arrow2.gif " width="32" height="32" /&gt;Stenkitės, kad paveikslai būtų maži ir greitai įkeliami.&lt;/dd&gt; &lt;dd&gt;&lt;img alt=" arrow " border="1" src=" arrow2.gif " width="32" height="32" /&gt;Naršyklės supranta tik gif ir jpg formatus.&lt;/dd&gt; &lt;dd&gt;&lt;img alt=" arrow " border="1" src=" arrow2.gif " width="32" height="32" /&gt;Naudodami pločio ir aukščio požymius paspartinkite puslapio įkėlimą.&lt;/dd&gt; &lt;/dl&gt;</pre>	<p>Dabar tinklalapis atrodo taip:</p> <p>Important things about images:</p> <ul style="list-style-type: none"> <li> Keep your images small and fast-loading.</li> <li> Only gif and jpg formats can be read by browsers.</li> <li> Use width and height attributes to make your page load faster.</li> </ul>
<p>Jeigu norite, kad sąrašas atrodytų dar gražiau, kraštinės požymį pakeiskite į „0“ ir atrodys, kad paveikslo ženkleliai puslapyje yra slankūs.</p> <pre>&lt;dl&gt; &lt;dt&gt;Kas žinotina apie paveikslus: &lt;dd&gt;&lt;img alt="arrow" border="0" src=" arrow2.gif " width="32" height="32" /&gt;Stenkitės, kad paveikslai būtų maži ir greitai įkeliami.&lt;/dd&gt; &lt;dd&gt;&lt;img alt=" arrow " border="0" src=" arrow2.gif " width="32" height="32" /&gt;Naršyklės supranta tik gif ir jpg formatus.&lt;/dd&gt; &lt;dd&gt;&lt;img alt=" arrow " border="0" src=" arrow2.gif " width="32" height="32" /&gt;Naudodami pločio ir aukščio požymius paspartinkite puslapio įkėlimą.&lt;/dd&gt; &lt;/dl&gt;</pre>	<p>Important things about images:</p> <ul style="list-style-type: none"> <li> Keep your images small and fast-loading.</li> <li> Only gif and jpg formats can be read by browsers.</li> <li> Use width and height attributes to make your page load faster.</li> </ul>

### 1.4.9 Paveikslų miniatiūrų kūrimas


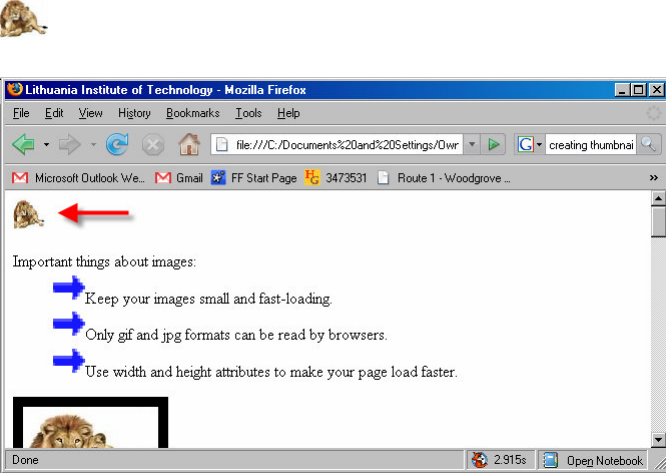
Paveikslų miniatiūros – tai labai maži (vaizdo ir failo prasme) paveikslai, t. y. sumažinti paveikslai, kurie ne tik atrodo maži, bet ir yra mažo dydžio failai. Tikros miniatiūros **NĖRA** tiesiog paveikslai, kurie atrodo mažesni, kadangi jų dydis keičiamas naudojant žymos `img` pločio ir aukščio požymius.

Tikrų miniatiūrų privalumas tas, kad labai greitai įkeliami nedideli „peržiūros“ paveikslai. Tačiau taip yra tik tuo atveju, jeigu failas yra mažas. Miniatiūrai sukurti netikslinga naudoti paveikslo žymos, kadangi failo dydis bus toks pats, kaip ir nesumažinto paveikslo.

Daugelis renkasi savo mėgstamiausią grafinę programą miniatiūroms sukurti. Sukurti miniatiūrą „nespecializuota“ grafine programa visai nesudėtinga:

1. Grafinėje programoje (pavyzdžiui, „Photoshop“, „Paint Shop Pro“, „Gimp“ ir pan.) atverkite nesumažintą paveikslą
2. Pasinaudokite programos dydžio keitimo funkcija „Resize“. Įsitikinkite, kad yra nustatyta parinktis „Preserve aspect ratio“
3. Standartinė miniatiūra yra maždaug 200 x 200 arba mažesnė, o failo dydis yra mažesnis nei 3–10k (atsižvelgiant į pirminę skiriamąją gebą ar optimizavimą)
4. Miniatiūrą įrašykite NAUJU vardu pasinaudodami „Save As“ (Įrašyti kaip). Paprastai prie paveikslo miniatiūros vardo pridama **\_tn**

Pavyzdyje pateikiama liūto paveikslo miniatiūra. Paveikslą sumažinau programa „Paint Shop Pro X“.

<p>Normalus dydis = 135 X 118</p> <p>Failo vardas: lion.jpg</p>	<p>Miniatiūros dydis = 50 X 44</p> <p>Failo vardas: lion_tn.jpg</p>
 <p>Norėdami sukurti paveikslo miniatiūrą su nuoroda, surinkite tokį kodą:</p> <pre>&lt;a href="http://www.gis-centras.lt/" target="_blank"&gt;</pre> <pre>&lt;img alt="Lion Thumbnail" border="0" src=" lion_tn.jpg" width="32" height="32" /&gt;</pre> <pre>&lt;/a&gt;</pre>	

#### 1.4.10 Paveikslo planų kūrimas

Paveikslo planas – tai toks paveikslas su nuorodomis, kuris leidžia skirtingas sritis susieti su skirtingomis svetainėmis. Norėdami sukurti paveikslo planą, tinklalapį turite papildyti dviem kodų grupėmis:

1. Visų pirma – plano sekcija, kurioje yra sričių apibrėžtys
2. Antra – specialia žyma img su papildiniu, nurodančiu naudoti planą.

Kai dokumente sukuriate paveikslo planą, privalote sukurti ir tekstinius nuorodas, papildančias paveikslo nuorodas. Tada vartotojai, kurie naudoja negrafines naršykles arba tie, kurie išjungia paveikslų rodymą, turės prieigą prie viso puslapio. Tekstinės nuorodos naudingi tuo, kad rodomi ir tada, kai paveikslas nėra įkraunamas.



Toliau pateikti aprašymai paaiškina, kaip veikia šios žymos. Tačiau praktikoje naudojama programinė įranga „ArcMap“, kuri automatiškai sukuria paveikslų planus.

## Plano žyma

Čia paveikslo planui suteikiate vardą, kurį turėsite naudoti vėliau žymai <b>img</b> . Šis vardas – tai privalomas plano žymos parametras, be kurio planas neveikia. Plano žyma atrodo taip. Atkreipkite dėmesį, kad plano žyma turi būti užverta pabaigoje.	<pre>&lt;map name="Map of Lithuania"&gt;</pre> <p>Čia rašomos sričių žymos...</p> <pre>&lt;/map&gt;</pre>
--	---

## Srities žyma

Čia priskiriate atskirų paveikslo sričių nuorodas. Srities žyma turi tris privalomus parametrus: figūros, koordinačių ir parametras href. Jeigu srities žymai trūks bent vieno parametro, jūsų planas neveiks! Pateikiame srities žymos pavyzdį:	<pre>&lt;area shape="čia nurodykite figūrą" coords="čia nurodykite koordinates" href="put URL here"&gt;</pre>
--	---

## Figūros ir koordinatės

Paveikslo planui galima naudoti <b>tris</b> figūras: <b>apskritimą</b> , <b>stačiakampį</b> ir <b>daugiakampį</b> . Koordinačių parametre kiekvienai figūrai turi būti nurodytas skirtingas skaitmenų skaičius. Labai svarbu laikytis srities žymų rašymo tvarkos. Pirmoji srities žyma, kurioje minima sritis, yra ta, kuri yra naudojama, todėl visų pirma turite nustatyti mažas sritis, o vėliau – didžiąsias (kurios gali apgaubti mažąsias).	
--	--

## Apskritimas

Jeigu pasirenkate <b>apskritimą</b> , turite nurodyti tris skaičius: apskritimo centro X ir Y koordinatės bei spindulį.	<pre>&lt;area shape="circle" coords="284,131,60" href="link.htm"&gt;</pre>
---	--

## Stačiakampis

Parametras <b>rect</b> sukuria nuorodas stačiakampio formos plote. Tam reikia nurodyti keturis skaičius: viršutinio kairiojo kampo X ir Y koordinatės bei apatinio dešiniojo kampo X ir Y koordinatės.	<pre>&lt;area shape="rect" coords="91,148,194,262" href="link.htm"&gt;</pre>
--	--

## Daugiakampis

<p><b>poly</b> – tai paskutinis srities žymos tipas. Ši žyma naudojama norint sukurti daugiakampį, ant kurio galima spragtelėti. Būtina nurodyti visų daugiakampio kraštinių taškų (viršūnių) X ir Y koordinatas. Kai naudojami sudėtingi daugiakampiai, koordinatų parametras gali būti labai ilgas, todėl yra lengva suklysti. Paprastai rekomenduojama rinktis ne daugiau kaip 8 taškų daugiakampes sritis.</p>	<pre>&lt;area shape="poly" coords="203,213,203,387,325,336" href="link.htm"&gt;</pre>
--	---

## href arba nohref

<p>Tai yra nuorodos paskirties vieta, kuri nurodoma lauke href lygiai taip pat, kaip ir žymos &lt;a&gt; parametras href. Be to, jeigu norite nurodyti sritį, kuri neturi nuorodos, parametras href galite pakeisti parametru nohref.</p>	<p>nohref pavyzdys:</p> <pre>&lt;area shape="default" nohref&gt;</pre> <p>Todėl sritis, kuri yra neapibrėžta (nohref), nereaguos, kai ant jos spragtelėsite.</p>
--	--

## Žyma img

<p>Čia naudojama žyma &lt;img&gt; yra tokia pati kaip ir įprasta žyma, bet turi vieną papildinį – ją reikia papildyti parametru usemap, kuris žymi dokumente anksčiau sukurtą planą. Pavyzdžiui, paveikslas planas, kuris naudoja planą, pavadintą „navigation“, atrodytų taip:</p>	<pre>&lt;img src="navigation.gif" alt="Navigation" usemap="#navigation"&gt;</pre>
---	---

## 1.5. Lentelės

### 1.5.1 Paprastos lentelės sukūrimas

Lentelės buvo sukurtos tam, kad jose būtų galima pateikti lentelinę informaciją. Vėliau jos tapo pagrindine tinklalapių kūrimo priemone, turinį paskirstančia visame ekrane. Ar atkreipėte dėmesį, kad HTML kodas yra linkęs viską grūsti prie kairiosios paraštės? Lentelės padeda nustatyti puslapio paraštes – labai svarbų „baltą tarpą“ kuriant tinklalapius.

Kurdami lentelę naudosite šias pagrindines žymas:

<code>&lt;table&gt;</code>	atveria lentelę
<code>&lt;th&gt;</code>	atveria ir rodo lentelės antraštę
<code>&lt;/th&gt;</code>	užveria antraštės eilutę
<code>&lt;tr&gt;</code>	atveria eilutę
<code>&lt;td&gt;</code>	atveria langelį
<code>&lt;/td&gt;</code>	užveria langelį
<code>&lt;/tr&gt;</code>	užveria eilutę
<code>&lt;/table&gt;</code>	užveria lentelę

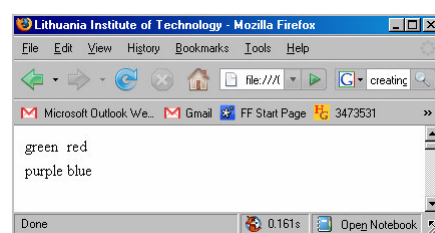
Pateikiame paprastos lentelės pavyzdį:

```
<table>
<!First Row>
<tr>
<td>green</td>
<td>red</td>
</tr>

<!Second Row>
<tr>
<td>purple</td>
<td>blue</td>
</tr>

</table>
```

Ši lentelė atrodytų taip:





## 1.5.2 Lentelės požymiai

Paprasta lentelė, kurią sukūrėme, nebuvo labai įdomi. Pažiūrėkime, kaip ją galime keisti papildydami lentelę požymiais:

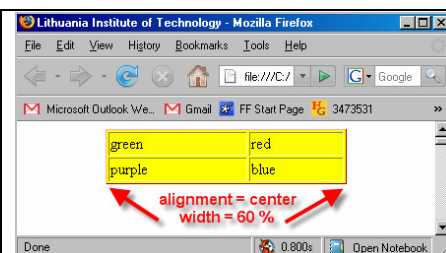
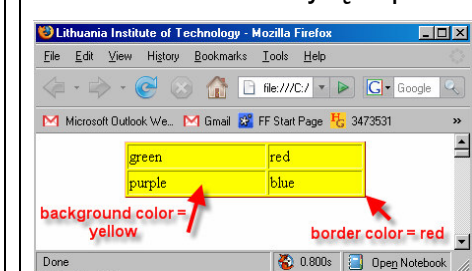
```
<table width="60%" border="1" align="center" bordercolor="#FF0000"
bgcolor="#FFFF00">

<tr><td>green</td><td>red</td></tr>

<tr><td>purple</td><td>blue</td></tr>

</table>
```

Dabar lentelė atrodytų taip:

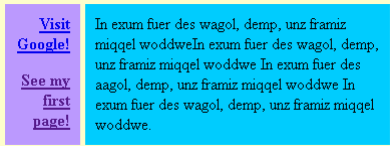


Atsižvelgiant į tai, kur įterpiate požymį bgcolor, galite iškart nuspalvinti visą eilutę, visą lentelę arba vieną langelį. Atkreipkite dėmesį, kad žymoje požymius galite įrašyti bet kuria tvarka, o jų formatas turi būti toks:

- Požymio pavadinimas su lygybės ženklu (bgcolor= align= width=).
- Jo reikšmė, įrašyta kabutėse ("center" "#CC99FF" "60%").
- Tarp požymio pavadinimo ir paskutinės pabaigos kabutės nepalieinama jokių tarpelių.

### 1.5.3 Nuorodos lentelėse

Naudodami lenteles, stulpelyje galite sukurti naršymo nuorodas:

Kodo paaiškinimas	Lentelės kodas
<p>Pirmoji eilutė atveria lentelę, kuri bus centre ir užims apie 80 proc. ekrano (iš kiekvienos pusės lieka po 10 proc. tuščios vietos), taip pat turės tamsiai violetinę fono spalvą. Požymis <b>cellpadding</b> turinį atitraukia 10 vaizdo taškų nuo langelio kraštinės, o požymis <b>cellspacing</b> išlaiko 5 tarpų atstumą tarp atskirų langelių.</p> <p>Tada atveriamą eilutę. Pirmojoje eilutėje atveriamas pirmasis langelis, kuris turi užimti 20 proc. lentelės pločio. Šio langelio fonas bus šviesiai violetinės spalvos, o turinys bus langelio viršuje dešinėje pusėje.</p> <p>Toliau pateikiamas dviejų aktyvių nuorodų, vieno į „Google“, o kito – į jūsų pirmąjį puslapį, sąrašas.</p> <p>Visos nuorodos yra pirmajame langelyje vienas virš kito.</p> <ul style="list-style-type: none"> <li>• Tada langelis yra užveriamas.</li> <li>• Tačiau eilutė dar neužverta. Paskui atversite antrąjį langelį, nustatysite, kad jis užimtų 80 proc. lentelės pločio, suteiksite jam šviesiai mėlyną foną ir turinį lygiuosite prie viršutinio kairiojo kampo.</li> <li>• Tada eina teksto pastraipa, o pasibaigus pastraipai užveriate antrąjį langelį, po to – eilutę ir visą lentelę.</li> </ul>	<pre>&lt;table width=80% align="center" bgcolor="#ffffcc" cellpadding="10" cellspacing="5"&gt;  &lt;tr&gt;  &lt;td width="20%" align="right" valign="top" bgcolor="#BB99ff"&gt; &lt;p&gt;&lt;b&gt;&lt;a href="http://www.google.com/"&gt;Visit Google!&lt;/a&gt;&lt;/b&gt;&lt;/p&gt; &lt;p&gt;&lt;b&gt;&lt;a href="firstpage.html"&gt;See my first page!&lt;/a&gt;&lt;/b&gt;&lt;/p&gt;&lt;/td&gt;  &lt;td width="80%" align="left" valign="top" bgcolor="#HHccff"&gt; &lt;p&gt;In exum fuer des wagol, demp, unz framiz miqqel woddweIn exum fuer des wagol, demp, unz framiz miqqel woddwe In exum fuer des aagol, demp, unz framiz miqqel woddwe In exum fuer des wagol, demp, unz framiz miqqel woddwe.&lt;/p&gt;&lt;/td&gt;  &lt;/tr&gt;  &lt;/table&gt;</pre> <p>... ir lentelė atrodys taip:</p> 

### 1.5.4 Požymiai cellpadding ir cellspacing

Požymis **cellpadding** sukuria daugiau baltos erdvės tarp langelių kraštų ir turinio.

Požymis **cellspacing** padidina atstumą tarp langelių. Abiejų požymių reikšmės nurodomos vaizdo taškais.

Abu šie požymiai priskiriami žymai <table>.

Daugiau šių dviejų požymių naudojimo pavyzdžių pateikta šiose svetainėse:

Cellpadding

[http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_table\\_cellpadding](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_cellpadding)

Cellspacing - [http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_table\\_cellspacing](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_cellspacing)

Dar vienas pavyzdys, kaip naudoti lenteles. Apibrėžtą sąrašą, kurį sukūrėme ankstesnėje temoje, paverskime lentele:

```
<table align="center" cellpadding="5" cellspacing="5" width="100%">
<tr><td colspan="2" align="center"><h2> Kas žinotina apie paveikslus:</h2></td></tr>
<tr>
<td width="20%" align="right"></td>
<td width="80%">Stenkitės, kad paveikslai būtų maži ir greitai įkeliami.</td></tr>
<tr>
<td align="right"></td>
<td>Naršyklės supranta tik gif ir jpg formatus.</td>
</tr>
<tr>
<td align="right">
</td>
<td> Naudodami pločio ir aukščio požymius paspartinkite puslapio įkėlimą. </td>
</tr>
</table>
```

**Important things about images:**

- ➔ Keep your images small and fast-loading.
- ➔ Only gif and jpg formats can be read by browsers.
- ➔ Use width and height attributes to make your page load faster.

Dabar papildykime kraštinės požymiu:


```
<table align="center" cellpadding="5" cellspacing="5" width="100%" border="1">
```

... ir lentelė turėtų atrodyti taip:


Important things about images:	
➔	Keep your images small and fast-loading.
➔	Only gif and jpg formats can be read by browsers.
➔	Use width and height attributes to make your page load faster.

### 1.5.5 Spalvų ir paveikslų naudojimas lentelėse

Lentelės galite naudoti norėdami tekstą pateikti kairėje esančiame fone, kaip parodyta pavyzdyje. Nustatyta, kad pirmasis langelis užimtų 25 proc. lentelės pločio ir jame nebūtų teksto, o antrasis – 75 proc. lentelės pločio, o jame būtų tekstas, kuris matomas dešinėje:

<pre>&lt;table width="100%" cellpadding="5" cellspacing="5"&gt; &lt;tr&gt; &lt;td width="25%"&gt;&amp;nbsp;&lt;/td&gt; &lt;td width="75%" align="left"&gt; &lt;h1&gt;Džiunglėse&lt;/h1&gt; &lt;h4&gt;Liūtai&lt;/h4&gt; &lt;h6&gt;Drambliai&lt;/h6&gt; &lt;/td&gt; &lt;/tr&gt; &lt;/table&gt;</pre>	 <div style="display: inline-block; border: 1px solid black; padding: 5px; background-color: #c0c0c0;"> <p><b>In the Jungle</b></p> <p>Lions</p> <p>Elephants</p> </div>
--	---

Lentelės taip pat galite naudoti kurdami tinklalapių grafinį dizainą ir nuspalvindami kiekvieną langelį. Pavyzdžiui:

<pre>&lt;table width="100%" border="0" cellpadding="5" cellspacing="0"&gt; &lt;tr&gt; &lt;td bgcolor="#6600CC" width="12%"&gt;&amp;nbsp;&lt;/td&gt; &lt;td bgcolor="#FF0000" width="13%"&gt;&amp;nbsp;&lt;/td&gt; &lt;td bgcolor="#FFFF00" width="75%"&gt;&amp;nbsp;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td bgcolor="#6666FF" width="12%"&gt;&amp;nbsp;&lt;/td&gt; &lt;td bgcolor="#FF9966" width="13%"&gt;&amp;nbsp;&lt;/td&gt; &lt;td bgcolor="#FFFF00" width="75%"&gt;&lt;h1&gt;GIS- Centras&lt;/h1&gt;&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td bgcolor="#0066FF" width="12%"&gt;&amp;nbsp;&lt;/td&gt;</pre>	<p>Tinklalapis atrodytų taip:</p> <div style="display: flex; justify-content: center; align-items: center; gap: 10px;">  <p>GIS-Centras</p> </div>
--	--

```
width="12%">&nbsp;</td>
<td bgcolor="#CC0000"
width="13%">&nbsp;</td>
<td bgcolor="#FFFF00"
width="75%">&nbsp;</td>
</tr>

</table>
```

## 1.6. Rėmeliai

### 1.6.1 Supažindinimas su rėmeliais

Naudodami rėmelius tame pačiame naršyklės lange galite rodyti daugiau nei vieną tinklalapį. Pavyzdžiui, galite pasirinkti **vertikaliają** arba **horizontaliąją rėminę konstrukciją**.

Naudodami rėmelius tame pačiame naršyklės lange galite rodyti daugiau nei vieną HTML dokumentą. Kiekvienas HTML dokumentas vadinamas rėmeliu, o kiekvienas rėmelis priklauso nuo kitų rėmelių.

Rėmelių naudojimo trūkumai:

- tinklalapio kūrėjui reikia stebėti daugiau HTML dokumentų
- sunku išspausdinti visą puslapį

Šiai temai naudosime pavyzdžius iš svetainės „W3schools“, kurie geriau iliustruos įvairius rėmelių naudojimo aspektus.

[http://www.w3schools.com/html/html\\_frames.asp](http://www.w3schools.com/html/html_frames.asp) (anglų kalba)

### 1.6.2 Rėmelių žymos

#### Rėminės konstrukcijos žyma

Žyma `<frameset>` apibrėžia, kaip langą padalyti į rėmelius. Kiekviena rėminė konstrukcija apibrėžia eilučių arba stulpelių grupę. Eilučių arba stulpelių reikšmės nurodo, kiek ekrano ploto užims kiekviena eilutė arba stulpelis.

#### Rėmelio žyma

Žyma `<frame>` apibrėžia, kurį HTML dokumentą patalpinti kiekviename rėmelyje.

Toliau pateiktame pavyzdyje yra rėminė konstrukcija, sudaryta iš dviejų stulpelių. Nustatyta, kad pirmasis stulpelis užimtų 25 proc. naršyklės lango pločio. Antrasis stulpelis turi užimti 75 proc. naršyklės lango pločio. HTML dokumentas `frame_a.htm` yra pateiktas pirmajame stulpelyje, o HTML dokumentas `frame_b.htm` – antrajame.

```
<frameset cols="25%,75%">
  <frame src="frame_a.htm">
  <frame src="frame_b.htm">
</frameset>
```

### 1.6.3 Rėmelių požymiai

Jeigu rėmelis turi matomas kraštines, vartotojas gali keisti rėmelio dydį tempdamas kraštinę. Norėdami nuo to apsisaugoti, galite žymą `<frame>` papildyti `noreize="noreize"`.

Naršyklėms, kurios nepalaiko rėmelių, įterpkite `<noframes>`.

**Atkreipkite dėmesį:** žymų `<body></body>` negalima naudoti kartu su žymomis `<frameset></frameset>`! Tačiau, jeigu naršyklėms, kurios nepalaiko rėmelių, įterpsite žymą `<noframes>` su tekstu, šį tekstą turėsite apgaubti žymomis `<body></body>`! Kaip tai daroma, parodyta pirmame pavyzdyje toliau.

### Kaip naudoti žymą `<noframes>`

Šiame pavyzdyje parodyta, kaip naudoti žymą `<noframes>`.

[http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_noframes](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_noframes)

### Mišri rėminė konstrukcija

Šiame pavyzdyje parodyta, kaip sukurti trijų dokumentų rėminę konstrukciją ir kaip juos derinti eilutėse ir stulpeliuose.

[http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_frame\\_mix](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_frame_mix)

### Rėminė konstrukcija su `noreize="noreize"`

Šiame pavyzdyje parodyta, kaip naudojamas požymis `noreize`. Rėmelių dydžio keisti negalima. Pele veskite per rėmelių kraštines ir pamatysite, kad kraštinių judinti negalima.

[http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_frame\\_noreize](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_frame_noreize)

### Naršymo rėmelis

Šiame pavyzdyje parodyta, kaip sukurti naršymo rėmelį. Naršymo rėmelyje pateikiamas nuorodų sąrašas, o antrasis rėmelis nurodytas kaip paskirtis (angl. *target*). Faile `tryhtml_contents.htm` yra trys nuorodos. Nuorodų išėtinis kodas:

```
<a href="frame_a.htm" target="showframe">Frame a</a><br>
```

```
<a href="frame_b.htm" target="showframe">Frame b</a><br>
```

```
<a href="frame_c.htm" target="showframe">Frame c</a>
```

Antrajame rėmelyje bus rodomas susietas dokumentas.

[http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_frame\\_navigation](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_frame_navigation)

### Įterptasis rėmelis

Šiame pavyzdyje parodyta, kaip sukurti įterptąjį rėmelį (rėmelį HTML puslapio viduje).

[http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_iframe](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_iframe)

### Peršokti į nurodytą rėmelio sekciją

Šiame pavyzdyje parodyti du rėmeliai. Viename iš rėmelių yra šaltinis į nurodytą failo sekciją. Faile `link.htm` nurodyta sekcija žymima `<a name="C10">`.

[http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_frame\\_jump](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_frame_jump)

### Peršokti į nurodytą sekciją naršymo rėmelyje

Šiame pavyzdyje parodyti du rėmeliai. Naršymo rėmelyje (`content.htm`) kairėje pateikiamas nuorodų sąrašas, o antrasis rėmelis (`link.htm`) naudojamas kaip paskirtis. Antrajame rėmelyje rodomas susietas dokumentas. Vienas iš nuorodų naršymo rėmelyje yra susietas su nurodyta paskirties failo sekcija. HTML kodas

faile content.htm atrodo taip: <a href = "link.htm" target = "showframe">Link without Anchor</a><br><a href = "link.htm#C10" target = "showframe">Link with Anchor</a>.

Rėmelių žymos

Žyma Apibūdinimas

<frameset> Apibrėžia rėminę konstrukciją

<frame> Apibrėžia lango dalį (rėmelį)

<noframes> Apibrėžia berėmę sekciją naršyklėms, kurios nepalaiko rėmelių

<iframe> Apibrėžia įterptąją lango dalį (rėmelį)

[http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_frame\\_navigation2](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_frame_navigation2)

### 1.6.4 Vertikalieji rėmeliai

Šiame pavyzdyje parodyta, kaip sukurti trijų skirtingų dokumentų vertikaliąją rėminę konstrukciją

([http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_frame\\_cols](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_frame_cols)).

### 1.6.5 Horizontalieji rėmeliai

Šiame pavyzdyje parodyta, kaip sukurti trijų skirtingų dokumentų horizontaliąją rėminę konstrukciją

([http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_frame\\_rows](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_frame_rows)).

### 1.6.6 Naršymo rėmelis

Šiame pavyzdyje parodyta, kaip sukurti naršymo rėmelį. Naršymo rėmelyje pateikiamas nuorodų sąrašas, o antrasis rėmelis nurodytas kaip paskirtis (angl. *target*). Faile tryhtml\_contents.htm yra trys nuorodos.

Saitų išeitinis kodas:

```
<a href = "frame_a.htm" target = "showframe">Frame a</a><br>
```

```
<a href = "frame_b.htm" target = "showframe">Frame b</a><br>
```

```
<a href = "frame_c.htm" target = "showframe">Frame c</a>
```

Antrajame rėmelyje bus rodomas susietas dokumentas

([http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_frame\\_navigation](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_frame_navigation)).



## 1.7. HTML padalijimai

Žyma `<div>` apibrėžia loginius padalijimus, nustatytus tinklalapyje. Ji veikia labai panašiai kaip ir pastraipos žyma, tačiau puslapį padalija į didesnes sekcijas.

Žyma `<div>` taip pat leidžia nustatyti ištisu HTML sekcijų stilių. Galite puslapio sekciją apibrėžti kaip paaiškinimą, o šiai sekcijai suteikti stilių, kuris skirsis nuo aplinkinio teksto stiliaus.

Žyma `<div>` leidžia suteikti vardą tam tikroms dokumentų sekcijoms, todėl jas galėtumėte formuoti pasinaudodami stiliaus lapais arba dinamine HTML.

Vienas dalykas, kurį reikia atsiminti naudojant žymą `<div>` yra tai, kad ji pertraukia pastraipas. Ji atitinka pastraipos pabaigą arba pradžią, todėl šia žyma pažymėtose sekcijose pastraipos yra galimos, tačiau pastraipos viduje žymos įterpti negalima.

Svarbiausi žymos `<div>` požymiai yra šie:

- „style“ – stiliaus
- „class“ – klasės
- „id“ – atpažinimo

Žyma `<div>` **negali** pakeisti žymos `<p>`. Žyma `<p>` naudojama tik pastraipoms, o žyma `<div>` apibrėžia bendresnio pobūdžio dokumento padalijimus. Žymų `<p>` nekeiskite žymomis `<div>`.

Įterpiant žymas `<div>` į dokumentą rekomenduojama jas pažymėti. Pavyzdžiui, jeigu apibrėžiate svetainės pagrindinę turinio sritį, turėtumėte žymą DIV pavadinti taip: „maincontent“. `<div id="maincontent">`

Rekomenduojama žymas `<div>` visada užverti iškart po to, kai jas atveriate. Po to turinį patalpinkite elemente.

Jeigu naudojate įdėtines žymas `<div>`, patikrinkite, ar žinote, kur įrašote turinį (kitaip tariant, kurios žymos dalimi jis turėtų būti).

### Stiliaus padalijimas

```
<div style="color : #ff0000">raudonas tekstas</div>
```

### Abipusiai lygiuotas tekstas

```
<div align="justify">Toliau pateiktas tekstas bus iš abiejų pusių lygiuotas per naršyklės plotį. Toks padalijimas gali nepasitvirtinti kai kuriose naršyklėse. Jam apibrėžti taip pat galima naudoti stiliaus žymas. Lorem ipsum sit dolor amet. Lorem ipsum sit dolor amet. Lorem ipsum sit dolor amet. Lorem ipsum sit dolor amet. Lorem ipsum sit dolor amet.</div>
```

- Kadangi tai yra elemento lygmens žyma, dauguma naršyklių prieš elementą ir po jo įterps papildomų tarpų (panašiai kaip ir pastraipos elemento atveju).
- Šis elementas dažnai naudojamas tinklalapiams kurti naudojant stiliaus lapus.

Norėdami išsamiau susipažinti su žymos <div> naudojimu, apsilankykite šioje svetainėje: [http://www.w3schools.com/tags/tag\\_div.asp](http://www.w3schools.com/tags/tag_div.asp) (anglų kalba).

## 1.8. CSS

### 1.8.1 Supažindinimas su CSS

Pakopinio stiliaus lapai (CSS) jums leidžia supaprastinti tinklalapių kūrimą. Vienas pagrindinių CSS privalumų yra tas, kad jis leidžia nuosekliai išlaikyti tinklalapių išvaizdą ir stilistiką. Be to, CSS jums leidžia valdyti daugiau tinklalapio išdėstymo ir kompozicijos aspektų. Daugiau, nei leidžia HTML kodas.

Panagrinėkime vieną iš populiariesnių HTML taikomų stilių – teksto spalvą ir dydį. Rašydami HTML kodą, raudoną tekstą `<h1>` kurtumėte taip:

<code>&lt;font color="#FF0000"&gt;&lt;h1&gt;Raudonas tekstas&lt;/h1&gt;&lt;/font&gt;</code>	<b>Red Text</b>
---	-----------------

Tačiau jeigu norėtumėte, kad visos jūsų antraštės `<h1>` būtų raudonos, turėtumėte šias šrifto žymas įterpti prieš kiekvieną žymą `<h1>` ir po jos.

Tam galite sugaišti labai daug laiko, taip pat padidėtų tikimybė įvelti klaidų programuojant.

Naudodami CSS galite sukurti stiliaus lapą, kuris visas antraštes `<h1>` nuspalvins raudonai. Toks stiliaus lapas atrodys taip:

```
h1 {color: #FF0000;}
```

Sukūrus šį stilių, visos žymos `<h1>`, kurios naudoja šį stilių, automatiškai virs raudonomis.

Stiliaus lapo struktūrą panagrinėkime išsamiau:

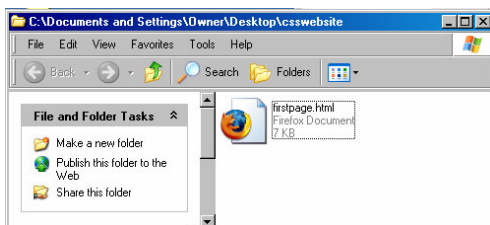
1. Pasirinkite elementą arba žymą, kuriems bus taikomas stiliaus lapas. Pavyzdžiui, **h1**
2. Įterpkite pradžios riestinį skliaustą {
3. Įvardykite savybę ir reikšmę, jas atskirdami dvitaškiu. Pavyzdžiui, spalva #FF0000 (color: #FF0000)
4. Įterpkite kabliataškį ;
5. Ir stiliaus žymą užverkite pabaigos riestiniu skliaustu }

Savybių arba reikšmių porų skaičius nėra ribojamas. Po kiekvienos poros reikia įrašyti kabliataškį, o abu elementus poroje atskirti dvitaškiu. Yra daugybė smulkmenų, bet atsiminkite: stiliaus kodą tereikia parašyti vieną kartą ir jis bus taikomas visoms žymoms <code>&lt;h1&gt;</code> .	<pre>h1 {color: #FF0000;}</pre>
Galime tęsti toliau ir stilių papildyti šrifto dydžio bei šrifto storio požymiais.	<pre>h1 {color: #FF0000; font-size: 24px; font-weight: bold; }</pre>

## 1.8.2 Pirmasis tinklalapis naudojant CSS

Prieš pradėdami kurti pirmąjį CSS tinklalapį, visų pirma turite sukurti tinklalapį, kuris bus susijęs su CSS failu.

Darbalaukyje sukurkite naują aplanką ir jį pavadinkite **csswebsite**. Į šį aplanką įdėkite failo **firstpage.html** kopiją.



Atverkite failą **firstpage.html**, ištrinkite visą kodą ir jį pakeiskite tokiu kodu:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>Mano pirmasis CSS tinklalapis</title></head>
```

```
<body>
```

```
<h1>GIS-CENTRAS</h1>
```

```
<p>In exum fuer des wagol, demp, unz framiz miqqel woddweIn exum fuer des wagol, demp, unz framiz miqqel woddwe In exum fuer des aagol, demp, unz framiz miqqel woddwe In exum fuer des wagol, demp, unz framiz miqqel woddwe.</p>
```

```
<p>In exum fuer des wagol, demp, unz framiz miqqel woddweIn exum fuer des wagol, demp, unz framiz miqqel woddwe In exum fuer des aagol, demp, unz framiz miqqel woddwe In exum fuer des wagol, demp, unz framiz miqqel woddwe.</p>
```

```
</body>
```

```
</html>
```

Teksto rengyklėje „Notepad“ atverkite naują dokumentą ir nukopijuokite toliau nurodytą kodą.

```
body { background-color: #FFFFFF;
margin-top: 2%;
margin-right:10%;
margin-bottom: 2%;
margin-left: 10%; }
```

```
h1 { font-family: Georgia, Times, serif;
```

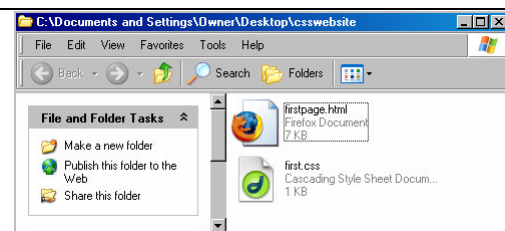
```
font-size: 24px;
font-weight: bold;
color: #000033; }
```

**Pagrindinės dalies** stiliuje fono spalva nustatyta balta (#FFFFFF), paraštės nustatytos aplink visą puslapį.

Stiliuje h1 buvo nustatytas šrifto tipas, storis ir spalva.

Šį puslapį įrašykite kaip failą **first.css** ir įsitikinkite, kad jis yra tame pačiame aplanke (**csswebsite**).

Dabar šiame aplanke turėtų būti du failai – **first.css** ir **firstpage.html**. Atkreipkite dėmesį, kad „Windows“ atpažino failo tipą CSS ir šį failą pažymėjo kaip pakopinio stiliaus lapų dokumentą.



Dabar **HTML** failą reikia susieti su **CSS** failu ir tik paskui jį bus galima naudoti.

Dar kartą atverkite failą **firstpage.html** ir tarp pradžios bei pabaigos žymų <head> </head> įterpkite žymą <link>.

```
<head>
```

```
<link href="first.css" rel="stylesheet" type="text/css">
```

```
<title>Mano pirmasis CSS tinklalapis</title></head>
```

- Bet kurioje žymoje <a> požymis href yra toks pats ir nurodo jūsų stiliaus lapo URL.
- Požymis rel nurodo, kad siejama su stiliaus lapu.
- Tipo požymis nurodo stiliaus lapo, su kuriuo siejate, tipą.

Įrašykite failą **firstpage.html** ir pamėginkite paleisti jį naršyklėje. Turėtumėte pamatyti tokį vaizdą:

### GIS-CENTRAS

In exum fuer des wagol, demp, unz framiz miqqel woddweIn exum fuer des wagol, demp, unz framiz miqqel woddwe In exum fuer des aagol, demp, unz framiz miqqel woddwe In exum fuer des wagol, demp, unz framiz miqqel woddwe.

In exum fuer des wagol, demp, unz framiz miqqel woddweIn exum fuer des wagol, demp, unz framiz miqqel woddwe In exum fuer des aagol, demp, unz framiz miqqel woddwe In exum fuer des aagol, demp, unz framiz miqqel woddwe In exum fuer des wagol, demp, unz framiz miqqel woddwe.

Pakeiskime tinklalapį įterpdami žymos <p> stilių. Failą **first.css** papildykite šiuo kodu:

```
p { color: #FF00FF;
font-size: 18px;
}
```

Dabar tinklalapis turėtų atrodyti taip:  
**GIS-CENTRAS**

In exum fuer des wagol, demp, unz framiz miqqel woddweIn exum fuer des wagol, demp, unz framiz miqqel woddwe In exum fuer des aagol, demp, unz framiz miqqel woddwe In exum fuer des wagol, demp, unz framiz miqqel woddwe.

In exum fuer des wagol, demp, unz framiz miqqel woddweIn exum fuer des wagol, demp, unz framiz miqqel woddwe In exum fuer des aagol, demp, unz framiz miqqel woddwe In exum fuer des wagol, demp, unz framiz miqqel woddwe.

### 1.8.3 Naršymo sąrašai

Naršymo meniu galima vertinti tiesiog kaip nuorodų sąrašą. Kad ir koks būtų naršymo meniu – horizontali eilutė viršuje ar vertikali eilutė šone – tai vis tiek yra tik sąrašas. Jeigu naršymą programuosite naudodami XHTML su CSS, galite sukurti meniu, kuris greitai įkeliamas (XHTML) ir lengvai adaptuojamas (CSS).

Tam, kad pradėtumėte kurti naršymo sąrašą, jums reikalingas sąrašas. Tam naudojamas standartinis nerūšiuotas sąrašas, kurio kiekvienas punktas yra susietas su žymomis <a>.

Toliau nurodytą kodą įterpkite į puslapį **firstpage.html**, esantį aplanke **csswebsite**. Kodą rašykite iškart po pradžios žymos <body>.

```
<ul>
<li><a href=http://www.google.com/>Google</a></li>
<li><a href=http://www.wikipedia.org/>Wikipedia</a></li>
</ul>
```

Nenaudojant CSS stiliaus, toks meniu nėra labai patrauklus, bet pakanka vos kelių CSS stilių ir galite sukurti meniu, kuriuo tikrai didžiuositės. Čia buvo panaudotas tuščiasis arba # nuorodas, todėl meniu punktai atrodo kaip nuorodos, kai pele užlipama ant teksto, tačiau jie nėra susieti su jokiais puslapiais.

- ◆ [Google](http://www.google.com/)
- ◆ [Wikipedia](http://www.wikipedia.org/)

### 1.8.4 Padalijimų įterpimas

Atverkite stiliaus lapą **first.css** ir įterpkite toliau nurodytą stiliaus kodą. Puslapis atrodys geriau sukomponuotas. Puslapis bus skaidomas į du padalijimus: vieną kairėje pusėje esančiam naršymo skydeliui ir kitą pagrindinei daliai arba puslapio centrui.

Pirmąjį padalijimą pavadinkite `.navigation` ir nustatykite langelio padėtį pradėdami nuo absoliučiosios padėties (0,0 x-y ašyje, pradinis taškas viršuje kairėje), nurodykite, kad viršutinės paraštės dydis yra 10 vaizdo taškų, dešinėsios šoninės 1 vaizdo taško pločio kraštinės spalva yra šviesiai žalia. Visas langelis užims 20 proc. ekrano pločio.

Antrajame padalijime `.centerdoc` nurodykite langelio atitraukimą. Naudodami tik skaičius nurodykite viršutinės, dešinėsios, apatinės ir kairiosios kraštinės vaizdo taškus tokia tvarka:

viršutinė = 0 vaizdo taškų dešinioji = 0 vaizdo taškų apatinė = 20 vaizdo taškų kairioji = 200 vaizdo taškų

Būtent dėl šių 200 vaizdo taškų naršyklė praleis pirmąjį padalijimą `.navigation` ir puslapyje pagrindinę dalį parodys atitrauktą per 200 vaizdo taškų.

#### **.navigation**

```
{position: absolute;
margin-top: 10px;
border-right: 1px solid #C6EC8C;
```

```
width: 15%;}
```

### .centerdoc

```
{padding: 0 0 20px 200px;
margin-top: 10px;
z-index: 20;}
```

Įterpkite nurodytą kodą į dokumentą **first.css** ir būtinai **įrašykite failą**. Dabar atverkite failą **firstpage.html** ir įterpkite toliau nurodytus **padalijimus**:

```
<div class="navigation">
```

```
<ul>
```

```
<li><a href=http://www.google.com/>Google</a></li>
```

```
<li><a href=http://www.wikipedia.org/>Wikipedia</a></li>
```

```
</ul>
```

```
</div>
```

```
<div class="centerdoc">
```

```
<h1>GIS-CENTRAS</h1>
```

```
<p>In exum fuer des wagol, demp, unz framiz miqqel woddweIn exum fuer des wagol, demp, unz framiz miqqel woddwe In exum fuer des aagol, demp, unz framiz miqqel woddwe In exum fuer des wagol, demp, unz framiz miqqel woddwe.</p>
```

```
<p>In exum fuer des wagol, demp, unz framiz miqqel woddweIn exum fuer des wagol, demp, unz framiz miqqel woddwe In exum fuer des aagol, demp, unz framiz miqqel woddwe In exum fuer des wagol, demp, unz framiz miqqel woddwe.</p>
```

```
</div>
```

Stiliaus lapą papildykime dar vienu stiliumi faile **first.css** įterpdami šį kodą:

```
li { list-style-type: none;
display: block;
text-align: left;
background-color: #C6EC8C;
border-top: solid 1px #cc9933; }
```

Įrašykite failą **first.css** ir atverkite **firstpage.html** naršyklėje. Turėtumėte pamatyti tokį puslapį:

[Google](#)  
[Wikipedia](#)

## GIS-CENTRAS

In exum fuer des wagol, demp, unz framiz miqqel woddweIn exum fuer des wagol,  
demp, unz framiz miqqel woddwe In exum fuer des aagol, demp, unz framiz miqqel  
woddwe In exum fuer des wagol, demp, unz framiz miqqel woddwe.

In exum fuer des wagol, demp, unz framiz miqqel woddweIn exum fuer des wagol,  
demp, unz framiz miqqel woddwe In exum fuer des aagol, demp, unz framiz miqqel  
woddwe In exum fuer des wagol, demp, unz framiz miqqel woddwe.

Norėdami toliau eksperimentuoti ir sužinoti daugiau apie pakopinio stiliaus lapus, apsilankykite nurodytoje svetainėje. Joje pateikta daugiau kaip 70 CSS pavyzdžių ir nuorodų.

<http://www.w3schools.com/css/default.asp> (anglų kalba)



## 1.9. Formos

Forma – tai tiesiog tinklalapis, galintis surinkti informaciją apie skaitytojus ir ją išsiųsti el. pašto adresu. Tokią informaciją galima surinkti atliekant apklausą, teikiant kontaktų formą, užsakymo formą arba klientų atsiliepimų formą.

### 1.9.1 Formos žymos

Prieš pradėdami kurti formas tinklalapio kodą turite papildyti tam tikromis eilutėmis. Šios kodo eilutės tvarkys formos duomenis ir rezultatus siųs jūsų el. pašto adresu. Šiame pavyzdyje naudosisime išgalvotą tinklalapio adresą. Kad jūsų forma veiktų, tinklalapio administratorius jums turi suteikti reikalingą informaciją (toliau paryškinta juodai).

```
<body>

<form action="http://www.somecompany.com/cgi-bin/formmail.cgi" method="post">
<input type="hidden" name="emailto" value="anyone@somecompany.com">
<input type="hidden" name="nextpage"
value="http://www.somecompany.com/thanks.html">
<input type="hidden" name="subject" value="Information Needed">
</form>

</body>
```

Pirmoji žyma `<form>` atveria formą. Puslapio pabaigoje nepamirškite įterpti pabaigos žymos `</form>`.

Veiksmo požymis **action="http://www.somecompany.com/cgi-bin/formmail.cgi" method="post"** naršyklei nurodo, kur siųsti į formą įvestus duomenis. Šiuo atveju duomenys turėtų būti siunčiami skriptui **cgi**, kurį svetainės administratorius sukūrė jūsų svetainėje. Šis skriptas **cgi** nuskaitys įvestus duomenis iš `<form>` ir juos formatuos. CGI reiškia bendrąjį tinklų sietuvą.

Kitos eilutės **<input type="hidden" name="emailto" value="anyone@somecompany.com">** vartotojas nematys, tačiau ji nurodys naršyklei rezultatus siųsti el. pašto adresu, kuris nurodytas **reikšmės** požymyje.

Kitos dvi eilutės taip pat skirtos įvedamiems duomenims ir yra paslėptos:

```
<input type="hidden" name="nextpage"
value="http://www.somecompany.com/thanks.html">
```

Naršyklei nurodoma, kad spragtelėjus ant mygtuko „**Submit**“ (Išsiųsti), reikia įkrauti kitą puslapį – **http://www.somecompany.com/thanks.html**. Šiame puslapyje svetainės lankytoji nurodoma

```
<input type="hidden" name="subject" value="Information Needed">
```

Kita paslėpta įvedama informacija **name="subject"** naršyklei nurodo, ką norite matyti el. laiško temos eilutėje.

Dabar žinosite, kad visi laiškai, kurių temos eilutėje nurodyta „**Information Needed**“, atkeliauja iš šios formos.

Kiekvienai žymai `<input>` reikia priskirti:

1. Tipą, kuris gali būti „hidden“, „radio“, „checkbox“, „text“, „textarea“, „submit“ ir (arba) „reset“
2. Vardą, kuris atkeliaus kaip žymena el. laiške.
3. Reikšmę, kuri yra turinys, vartotojo įrašomas į pateiktą formą. Jeigu naudojamos parinktės akutės ir žymimieji langeliai, jų reikšmė bus nurodyta kode. Jeigu pasirinkamas tekstas ir tekstinės sritys, reikšmes surenka vartotojas.

### 1.9.2 Parinktės akutės

Kiekvienam formos įvediniui reikės nurodyti tipą, vardą ir reikšmę. Pateikiame parinktės akučių žymų `<input>` sąrašą:

<pre>&lt;p&gt;Kokius ledus labiausiai mėgstate&lt;/p&gt; &lt;p&gt; &lt;input type="radio" name="ice cream" value="Chocolate"&gt;Šokoladinius? &lt;input type="radio" name="ice cream" value="Vanilla" checked="checked"&gt;Vanilinius? &lt;input type="radio" name="ice cream" value="Strawberry"&gt;Braškinius? &lt;/p&gt;</pre>	<p>Naršyklėje tinklalapis atrodys taip:</p> <p>What is your favourite ice cream</p> <p><input type="radio"/> Chocolate? <input checked="" type="radio"/> Vanilla? <input type="radio"/> Strawberry?</p>
---	---

Požymių sąrašą papildantis požymis `checked="checked"` nurodo, kad yra iš anksto pasirinktas būtent šis pasirinkimas.

### 1.9.3 Išsiuntimo mygtukas

Užpildęs formą, svetainės lankytojas turi spragtelėti ant išsiuntimo mygtuko. Išsiuntimo mygtuko kodas yra toks:

```
<input type="submit" value="Submit">
```

Šiuo atveju vardo nurodyti nereikia. Pakanka nurodyti naršyklei, kad tai yra signalas šias reikšmes siųsti cgi-bin adresu, nurodytu žymoje `<form>`.

Reikšmės požymis parodo tai, kas bus matoma mygtuke, kai dokumentas bus įkrautas į naršyklę.

Submit

Jeigu reikšmę pakeistumėte į `<input type="submit" value="Send Now">`, mygtukas atrodytų taip:

Send Now

### 1.9.4 Žymimieji langeliai

Parinkties akutės formą pildančiam asmeniui leidžia tik vieną pasirinkimą. Bandant spragtelėti daugiau nei ant vienos parinkties akutės, anksčiau spragtelėta akutė pasidarys tuščia.

Žymimieji langeliai leidžia formą pildančiam asmeniui pasirinkti keletą parinkčių.

Kiekvienai žymai <input> reikia nurodyti tipo, vardo ir vertės požymį.

```
<p>Jūs mėgstamiausias vaisius?</p>
<p>
<input type="checkbox" name="fruit1" value="banana">Bananas
<input type="checkbox" name="fruit2" value="orange">Apelsinas
<input type="checkbox" name="fruit3" value="apple">Obuolys
<input type="checkbox" name="fruit4" value="pear">Kriaušė
</p>
```

Atkreipkite dėmesį, kad visi žymimųjų langelių **vardai** turi būti skirtingi (kitais atvejais nei parinkties akučių). Taip yra todėl, kad parinkties akutės leidžia tik vieną atsakymą, o žymimieji langeliai – kelis.

Štai kaip tai atrodys naršyklės lange:

Pick your favourite fruit?

Banana  Orange  Apple  Pear

Send Now

### 1.9.5 Įvesties žyma

Tam, kad surinktumėte tekstinę informaciją, jums reikia vietos, kur svetainės lankytojai galėtų įvesti informaciją. Vienas iš taikomų metodų yra žyma „input“.

Štai kaip atrodys kodas, parašytas keliems tekstiniams langeliams, kuriuose klientai įrašys savo vardą, pavardę ir telefono numerį, taip pat užsakomų sėklų pakelių skaičių:

```
<p>Vardas: <input type="text" name="first_name"></p>
```

```
<p>Pavardė: <input type="text" name="last_name"></p>
```

```
<p>Telefono numeris: <input type="text" name="phone"></p>
```

Forma turėtų atrodyti taip:

First Name: Last Name: Telephone Number: 

### 1.9.6 Tekstinė sritis

Tekstinės sritys – tai didesni formos langeliai, kuriuose skaitytojai gali surinkti keletą eilučių pastabų arba klausimų. Šiems teksto langeliams nenaudojamos žymos `<input>`, todėl požymiai turi priskirti papildomą informaciją, nustatančią tekstinės srities eilučių ir stulpelių skaičių. Kitoje eilutėje atverkite 44 simbolių pločio ir 5 eilučių ilgio tekstinę sritį.

`<p>`Šiame langelyje įrašykite papildomas pastabas:

`<textarea name="comments" rows="4" cols="45"></textarea>`

`</p>`

Ji atrodys taip:

Please provide additional comments in the area below:



### 1.9.7 Grįžties mygtukas

Jeigu kuris nors lankytojas nurodo visą informaciją, bet vėliau apsigalvoja, galite numatyti grįžties mygtuką „Reset“. Vienu spragtelėjimu dings visi į formą įvesti duomenys. Kodas yra toks:

`<input type="reset" value="Reset">`

Kaip ir išsiuntimo mygtuko atveju, vardo nurodyti nereikia. Tereikia nurodyti naršyklei, kad tai yra signalas visiems įvestiems duomenims ištrinti ir pradėti iš naujo.

Reikšmės požymis parodo tai, kas bus matoma mygtuke, kai dokumentas bus įkrautas į naršyklę. Štai taip:

Jeigu reikšmę pakeistumėte į

`<input type="reset" value="Try Again">`,

mygtukas atrodytų taip:

## Formų pavyzdžiai iš svetainės „W3 Schools“

Šioje svetainėje pateikiama daugybė formų pavyzdžių:

### Forma su įvesties laukais ir išsiuntimo mygtuku

Šiame pavyzdyje parodyta, kaip puslapį papildyti forma. Formoje yra du įvesties laukai ir išsiuntimo mygtukas:

[http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_form\\_submit](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_submit)

### Forma su žymimaisiais langeliais

Formoje yra du žymimieji langeliai ir išsiuntimo mygtukas:

[http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_form\\_checkbox](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_checkbox)

### Forma su parinkties akutėmis

Formoje yra dvi parinkties akutės ir išsiuntimo mygtukas:

[http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_form\\_radio](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_radio)

### Siųsti el. laišką iš formos

Šiame pavyzdyje parodyta, kaip iš formos siųsti el. laišką:

[http://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_form\\_mail](http://www.w3schools.com/html/tryit.asp?filename=tryhtml_form_mail)

## Nuorodos (anglų kalba)

- <http://www.wikipedia.org/>
- <http://www.w3.org>
- <http://www.w3schools.com/>
- <http://www.htmlcodetutorial.com/>
- <http://www.htmlgoodies.com/>
- <http://free-html-tutorials.com/>
- [http://webdesign.about.com/od/htmlxhtmltutorials/HTML\\_Tutorials\\_Web\\_Design\\_Tutorials\\_Beginning\\_HTML\\_and\\_Web\\_Design.htm](http://webdesign.about.com/od/htmlxhtmltutorials/HTML_Tutorials_Web_Design_Tutorials_Beginning_HTML_and_Web_Design.htm)
- <http://www.mcli.dist.maricopa.edu/tut/lessons.html>

## Paskaitos savarankiško mokymosi klausimai:

1. Kokie yra septyni interneto sluoksniai ir kokias funkcijas jie atlieka?
2. Apibūdinkite interneto TCP/IP struktūrinį elementą ir paaiškinkite, kaip jis sąveikauja su FTP ir HTTP.
3. Paaiškinkite HTML žymų ir HTML požymių paskirtį.
4. Nurodykite 3 priežastis, dėl kurių tinklalapyje pateiktumėte sąrašą, ir paaiškinkite, kuo rūšiuotas sąrašas skiriasi nuo nerūšiuoto.
5. Į ką reikia atkreipti dėmesį atsisiunčiant paveikslus iš interneto?
6. Paaiškinkite, kokia yra pagrindinė paveikslo plano paskirtis ir naudojimas.
7. Koku tikslu buvo sukurtos lentelės?
8. Kokie yra du pagrindiniai rėmelių naudojimo trūkumai?
9. Koks yra pagrindinis CSS naudojimo privalumas?
10. Kokias funkcijas formos atlieka tinklalapyje?

## Vartojamieji terminai:

- TCP/IP
- FTP
- HTTP
- Telnetas
- SMTP
- „Bittorrent“
- ARPANET
- OSI
- HTML
- Tinklapis
- Žymos
- Požymiai
- DTD
- W3C
- Naršyklė
- Nuorodos
- Sąrašai
- Paskirtis
- Prieraišas
- Santykinis
- Absoliutusias
- El. paštas
- Atsisiųsti
- Paveikslas
- Paveikslo planas
- Lentelė
- Langelių atitraukimas (angl. *cellpadding*)
- Langelių protarpiai (angl. *cellspacing*)
- Rėmeliai
- Rėminė konstrukcija
- Padalijimai
- CSS
- Formos

## **2. Interneto kalbos vartotojo sąsajos, ryšio, duomenų ir schemų aprašams**

Skriptų kalbos ir susijusios technologijos – DHTML, HTML, HTML dokumento objektų modeliai (DOM), „JavaScript“, XML, XSL ir XSLT – yra labai svarbios, ypač interneto kūrimo plėtrai ir taikomosioms programoms. Šioje paskaitoje supažindinama su šiomis interneto taikomųjų programų technologijomis praplečiant temas, kurios buvo nagrinėjamos 1 paskaitoje, ir apibūdinant, kaip naujausi interneto taikomųjų programų technologijos laimėjimai prisidėjo prie interneto technologijų veiksmingumo didinimo. Aptariami metodai, susiejantys DHTML DOM, W3C standartai, „JavaScript“ metodai, XML kodavimas, XSL ir XSLT metodai, taip pat jų privalumai ir trūkumai.



## 2.1. DHTML, W3C standartai, HTML dokumento objektų modelis

### 2.1.1 Dinaminė HTML arba DHTML

**Dinaminė HTML**, arba DHTML, yra interaktyviųjų ir animacinių svetainių kūrimo metodų visuma. Šie metodai leidžia keisti tinklalapių išvaizdą ir turinį (todėl tokie puslapiai gali būti vadinami interaktyviaisiais arba animaciniais). DHTML naudoja keletą kalbų (pavyzdžiui, HTML), kliento pusės skriptų kalbą (pvz., „JavaScript“), aprašomąją turinio kalbą (angl. *presentation definition language*) (pakopinius stiliaus lapus, CSS) ir dokumento objektų modelį. DHTML tinklalapyje skriptai keičia turinio aprašomosios kalbos kintamuosius. Taip keičiama HTML puslapio turinio išvaizda ir veikimas. DHTML dažnai naudojama tinklalapiuose kuriant paveikslų kaitos mygtukus arba išskleidžiamuosius meniu.

Be to, dinaminis tinklalapis yra platesnė sąvoka. Tai bet kuris tinklalapis, kuris kiekvienam vartotojui, apkrovai ar konkrečioms kintamųjų reikšmėms kuriamas skirtingai. Tokie tinklalapiai gali būti sukurti ir kliento pusės, ir serverio pusės skriptų (pvz., PHP arba „Perl“), kai interneto serveris turinį kuria prieš išsiųsdamas jį klientui.

DHTML trūkumai:

- Kartais skriptai nevienodai veikė skirtingose interneto naršyklėse, todėl buvo pasitelkti naujesni metodai (pvz., „JavaScript“ kodavimas ir DOM skriptai). Šiais naujesniais metodais tapo lengviau naudotis pradėjus taikyti progresinį gerinimą (angl. *Progressive Enhancement*) (internetu projektavimo strategija, kurioje akcentuojamas prieinamumas bei išorinių stiliaus lapų ir skriptų technologijos).
- Sunkumai kuriant ir derinant tinklalapius, kadangi skirtingos interneto naršyklės palaiko skirtingas technologijas. Kadangi ekranų dydžiai skiriasi, DHTML metodai gali būti tinkami tik tam tikriems naršyklių ir ekrano dydžių deriniams. Naujesnėse naršyklėse, pavyzdžiui, „Internet Explorer 5.0+“, „Mozilla Firefox 2.0+“ ir „Opera 7.0+“ DHTML paprastai derinama su bendroju dokumento objektų modeliu.

## Paprastai DHTML sukurto tinklalapio struktūra yra tokia:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>DHTML example</title>
    <script type="text/javascript">
      function init() {
        myObj = document.getElementById("navigation");
        // .... more code
      }
      window.onload=init;
    </script>
  </head>
  <body>
    <div id="navigation"></div>
  </body>
</html>

```

Dažnai kodas yra išoriniame faile; tai daroma susiejant failą, kuriame yra „JavaScript“.

Toks būdas padeda, kai tą patį skriptą naudoja keli puslapiai:

```

</pre>
<script type="text/javascript" src="myjavascript.js"></script>

```

Toliau pateiktame pavyzdyje bus rodomas papildomas teksto blokas. Kodas iliustruoja dažnai naudojamą funkciją. Papildoma tinklalapio dalis bus rodoma tik tuo atveju, jeigu naudotojas to paprašys.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Test</title>
    <style type="text/css">
      h2 {background-color: lightblue; width: 100%}
      a {font-size: larger; background-color: goldenrod}
      a:hover {background-color: gold}
      #example1 {display: none; margin: 3%; padding: 4%; background-color:
limegreen}
    </style>
    <script type="text/javascript">
      function changeDisplayState (id) {
        d=document.getElementById("showhide");
        e=document.getElementById(id);
        if (e.style.display == 'none' || e.style.display == "") {
          e.style.display = 'block';
          d.innerHTML = 'Hide example';
        } else {
          e.style.display = 'none';
          d.innerHTML = 'Show example';
        }
      }
    </script>
  </head>
  <body>

```

```

<h2>How to use a DOM function</h2>
<div><a id="showhide" href="javascript:changeDisplayState('example1') ">Show
example</a></div>
<div id="example1">
  This is the example.
  (Additional information, which is only displayed on request)...
</div>
<div>The general text continues...</div>
</body>
</html>

```

## 2.1.2 Pasaulinio interneto konsorciumas (W3C)

Pasaulinio interneto konsorciumas (**W3C**) – tai pagrindinė tarptautinė pasaulinio interneto (trump. WWW arba W3) standartų organizacija. W3C buvo įkurtas siekiant programų suderinamumo ir interneto kūrėjų pritarimo tvirtinant naujus standartus. Konsorciumas buvo įkurtas siekiant paskatinti visus pardavėjus susitarti dėl pagrindinių principų ir komponentų, kuriems visi pritarė, pvz., skriptų. Tačiau W3C nėra parengęs specifikacijos, kurioje būtų formaliai apibrėžta DHTML. Dauguma gairių gali tiktai tais atvejais, kai naudojama DHTML, tačiau toliau išvardytose gairėse akcentuojama skriptų ir stiliaus lapų problematika. W3C ir (arba) IETF standartams (iš interneto protokolų komplekto) priskiriama: CSS, CGI, DOM, GRDDL, HTML, OWL, RDF, SVG, SOAP, SMIL, SRGS, SSML, VoiceXML, WSDL, XACML, XHTML, XML, „XML Events“, XForms, XML ir XSLT. 1998 m. W3C paskelbė 1 lygmens DOM specifikaciją, sudariusią galimybes pasiekti ir valdyti kiekvieną HTML puslapio elementą. Ši rekomendacija buvo įdiegta visose naršyklėse, todėl DOM suderinamumo problemų beveik nebeliko.

Šiam kursui svarbios W3C technologijos:

- Pakopinio stiliaus lapai (CSS), kurie buvo aptarti pirmojoje paskaitoje, yra stiliaus lapų kalba, naudojama ženklinimo kalba parašyto dokumento pateikimui apibūdinti. Ji dažniausia naudojama suteikiant stilių tinklalapiams, parašytiems HTML arba XHTML (<http://lt.wikipedia.org/wiki/XHTML>), tačiau kalbą galima taikyti visiems XML dokumentams, įskaitant ir SVG (<http://en.wikipedia.org/wiki/SVG> – anglų kalba).
- Bendrasis tinklų sietuvas (CGI) – tai standartinis išorinės taikomosios programinės įrangos ir informacijos serverio (paprastai interneto serverio) sąveikos protokolas. Informacijos serveris turi atsakyti į užklausas (internetu serverių atveju – į kliento interneto naršyklių užklausas) ir siųsti rezultatus. Kai tik gaunama nauja užklausa, serveris analizuoja, ko yra prašoma, ir siunčia atitinkamus rezultatus.
- Dokumento objektų modelis (DOM) – tai nuo kompiuterių platformos ir kalbos nepriklausomas standartinis objektų modelis, skirtas HTML arba XML ir susijusiems formatams pateikti. Interneto naršyklei nebūtina naudoti DOM, kad ji galėtų pateikti HTML dokumentą. Tačiau DOM reikalingas „JavaScript“ skriptams, kurie nori dinamiškai patikrinti arba pakeisti tinklalapį. Kitaip tariant, dokumento objektų modelis leidžia „JavaScript“ pamatyti HTML puslapį ir naršyklės būseną.
- Dvimatė vektorinė grafika (*Scalable Vector Graphics*) (SVG) – tai XML specifikacija ir failų formatas, skirtas statinei ir animacinei dvimatei vektorinei grafikai aprašyti. SVG gali būti tik deklaratyvi arba su skriptais. Paveikslai gali būti su hipersaitais, kuriems naudojami išsiunčiamos paprastos XLink nuorodos.
- SOAP – tai protokolas, skirtas keistis XML pranešimais kompiuterių tinkluose, paprastai naudojant HTTP ir (arba) HTTPS. SOAP yra interneto paslaugos dėklo

pagrindo sluoksnis, užtikrinantis pagrindinę pranešimų siuntimo (angl. *messaging*) struktūrą, ant kurios galima kurti abstrakčiuosius sluoksnius.

- Interneto paslaugos aprašomoji kalba (*Web Services Description Language*) (WSDL) – tai XML kalba, numatanti interneto paslaugos aprašymo modelį. WSDL paslaugas apibūdina kaip tinklo galinių taškų, arba prievadų, rinkinius. Šiuo tikslu WSDL specifikacijoje numatytas dokumentų XML formatas.
- XHTML (*Extensible HyperText Markup Language*) kalba – tai ženklinimo kalba, turinti tokių pačių išraiškos galimybių kaip ir HTML, tačiau atitinkanti XML sintaksę. HTML naudojama SGML (*Standard Generalized Markup Language*) – itin universaliai ženklinimo kalbai, o XHTML – XML kalbai, daugiau apribojimų numatančiam SGML poaibiui. Kadangi jie turi būti struktūriškai tvarkingi, teisingi XHTML dokumentai leidžia atlikti automatinį duomenų apdorojimą naudojant standartines XML priemones, o HTML reikalingas gana sudėtingas, lankstus ir dažniausiai specialiai sukurtas analizatorius. XHTML daugeliu aspektų galima vertinti kaip HTML ir XML samplaiką, kadangi tai yra HTML performulavimas XML formatu.
- XML (*Extensible Markup Language*) kalba yra plačios paskirties ženklinimo kalba [1]. Ji laikoma išplečiama kalba, kadangi naudotojai gali apibrėžti savo pačių žymas. Jos pagrindinė paskirtis – palengvinti dalijimąsi struktūriniais duomenimis skirtingose informacinėse sistemose, visų pirma internete.

Šios W3C technologijos bus aptariamose šioje ir vėlesnėse kurso paskaitose.

### 2.1.3 Dokumento objektų modelis (DOM)

**Dokumento objektų modelis** (DOM) – tai nuo kompiuterių platformos ir kalbos nepriklausomas standartinis objektų modelis, skirtas HTML arba XML ir susijusiems formatams pateikti. DOM yra suskirstytas į atskiras dalis (pagrindinę dalį, XML ir HTML) bei skirtingus lygmenis (1, 2, 3 DOM lygmenys). Dokumento objektų modelis leidžia pasiekti visus HTML dokumento elementus, taip pat numato metodus ir savybes, kaip papildyti, perkelti, pakeisti arba pašalinti HTML elementus. Interneto naršyklei nebūtina naudoti DOM, kad ji galėtų pateikti HTML dokumentą. Tačiau DOM reikalingas [JavaScript](#) skriptams, kurie nori dinamiškai patikrinti arba pakeisti tinklalapį. Kitaip tariant, dokumento objektų modelis leidžia „JavaScript“ pamatyti HTML puslapį ir naršyklės būseną. Kad galėtų ką nors pakeisti puslapyje, „JavaScript“ skriptui reikalinga prieiga prie visų HTML dokumento elementų. Naudodami „JavaScript“ galite pakeisti viso HTML dokumento struktūrą. Galite įterpti, pašalinti, keisti arba pertvarkyti puslapio objektus. „JavaScript“ taip pat gali DOM naudoti HTML, XHTML bei XML dokumentams skaityti ir keisti.

Kadangi DOM palaiko navigaciją bet kuria kryptimi (pvz., tėvo ir ankstesnio brolio) bei leidžia daryti pakeitimus savo nuožiūra, įgyvendinant būtiną dokumentą, kuris iki tol buvo nuskaitytas (arba tam tikrą jo išanalizuotą formą), bent jau perkelti į buferį. Todėl DOM tikriausiai labiausiai tinka tais atvejais, kai dokumentą reikia pasiekti pakartotinai arba ne eilės tvarka. Jeigu puslapis bus naudojamas tik nuoseklia prieiga ir tik vieną kartą, SAX modelis (<http://en.wikipedia.org/wiki/SAX> – anglų kalba) būtų greitesnis sprendimas, reikalaujantis mažiau atminties. Be to, tiesioginės (angl. *non-extractive*) XML analizės modeliai, pavyzdžiui, VTD-XML (<http://en.wikipedia.org/wiki/VTD-XML> – anglų kalba), yra naujas mažiau atminties reikalaujantis variantas.

W3C pradėjo kurti DOM praėjusio amžiaus dešimtojo dešimtmečio viduryje. Nors W3C neparengė DOM 0 specifikacijos, šis modelis buvo iš dalies dokumentuotas ir įtrauktas į HTML 4 specifikaciją. Iki 1998 m. spalio mėn. buvo išleista pirmoji DOM (DOM 1) specifikacija. 2000 m. lapkričio mėn. buvo išleista DOM 2 specifikacija, kurioje buvo pateiktos stiliaus lapų objektų modelio ir manipuliavimo stiliaus informacija ypatybės. 2004 m. balandžio mėn. buvo išleista DOM 3, kuri ir yra naujausia paskelbta DOM specifikacija.

W3C parengtos DOM specifikacijos yra suskirstytos lygmenimis, kurių kiekviename yra privalomieji ir pasirenkamieji moduliai. Galima teigti, kad taikomoji programa palaiko tam tikrą lygmenį tik tuomet, jeigu ji įgyvendina visus šio lygmens, taip pat ir žemesnių lygmenų, reikalavimus. Taikomoji programa taip pat gali palaikyti konkretaus gamintojo plėtinius, kurie atitinka W3C standartus. Nuo 2005 m. 1 lygmuo, 2 lygmuo ir kai kurie 3 lygmens moduliai tapo W3C rekomendacijomis, t. y. jie įgavo galutinį pavidalą.

### 0 lygmuo

Taikomoji programa palaiko tarpinį DOM variantą, egzistavusi iki DOM 1 lygmens sukūrimo. Pavyzdžiai – *DHTML objektų modelis* arba „Netscape“ tarpinis DOM. 0 lygmuo – tai ne formali W3C paskelbta specifikacija, o daugiau santrauka, kurioje apibūdinama tai, kad egzistavo iki standartizacijos. DOM 0 lygmenį „Netscape“ išrado tuo pat metu, kai buvo išrastas ir „JavaScript“, o pirmąjį kartą lygmuo buvo panaudotas „Netscape 2“. Jis užtikrina prieigą prie kelių HTML elementų, formų ir paveikslų. Siekiant užtikrinti suderinamumą su nusistovėjusiomis sistemomis, naujesnės ir pažangesnės naršyklės, įskaitant palaikančias ir DOM 1 lygmenį, taip pat palaiko senesnę DOM 0 lygmenį, nes kitaip neveiktų dauguma bendrų skriptų.

### 1 lygmuo

DOM (HTML arba XML) dokumento (medžio struktūros) navigacija ir manipuliavimas turiniu (įskaitant elementų įterpimą). Taip pat yra įtraukti HTML elementai. Taip ne tik pateikiamas tikslus viso HTML (arba XML) dokumento modelis, bet ir sudaroma galimybė labiau keisti dokumentą, pavyzdžiui, pastraipas arba lenteles.

### 2 lygmuo

2 lygmens specifikaciją sudaro šešios skirtingos specifikacijos:

- „DOM2 Pagrindinė dalis“
- „Rodiniai“
- „Įvykiai“
- „Stilius“
- „Analizė ir blokai“
- „2 lygmens HTML“

Specifikacija „2 lygmens pagrindinė dalis“ papildo „1 lygmens pagrindinės dalies“ funkcionalumą. Joje taip pat pateiktos specializuotos sąsajos, skirtos XML lygmeniui palaikyti. Specifikacija „2 lygmens rodiniai“ leidžia programoms ir skriptams dinamiškai pasiekti ir atnaujinti dokumento pateikimo turinį. Specifikacija „2 lygmens įvykiai“ programoms ir skriptams nurodo bendrą įvykių sistemą. Joje pristatomos tokios įvykių srauto sąvokos, kaip paėmimas (angl. *capture*), iškėlimas (angl. *bubbling*) ir nutraukimas (angl. *cancellation*). CSS 2 lygmuo, arba „2 lygmens stilius“, leidžia programoms ir skriptams dinamiškai pasiekti ir atnaujinti stiliaus lapų turinį. Numatytos sąsajos stiliaus lapams ir pakopiniams stiliaus lapams. Specifikacija „2 lygmens analizė ir blokai“ leidžia programoms ir skriptams dinamiškai išanalizuoti ir atpažinti dokumento turinio bloką.

Specifikacija „2 lygmens blokas“ leidžia kurti, įterpti, keisti ir ištrinti dokumento turinio bloką. Specifikacija „2 lygmens HTML“ leidžia programoms ir skriptams dinamiškai pasiekti ir atnaujinti HTML dokumentų turinį ir struktūrą. Ji išplečia 1 lygmens HTML aprašytas sąsajas išnaudojant „2 lygmens pagrindinės dalies“ galimybes.

**3 lygmens** specifikaciją sudaro 6 skirtingos specifikacijos:

- „DOM 3 lygmens pagrindinė dalis“
- „DOM 3 lygmens įkėlimas ir įrašymas“
- „DOM 3 lygmens XPath“
- „DOM 3 lygmens rodiniai ir formatavimas“
- „DOM 3 lygmens reikalavimai“
- „DOM 3 lygmens patikrinimas“, leidžianti dar geriau išnaudoti DOM galimybes

DOM3 specifikacija „Pagrindinė dalis“ papildo DOM1 ir DOM2 specifikacijų „Pagrindinė dalis“ funkcionalumą naujais metodais. DOM3 specifikacija „Įkėlimas ir įrašymas“ leidžia programoms ir skriptams dinamiškai įkelti XML dokumento turinį į DOM dokumentą bei DOM dokumentą paversti XML dokumentu. DOM3 specifikacija „XPath“ numato paprastus funkcionalumus DOM medžiui pasiekti naudojant „XPath 1.0“. 3 lygmens specifikacijoje „Rodiniai ir formatavimas“ daugiausia nagrinėjamas puslapių formatavimas ir sudarymas, taip pat šių puslapių susiejimas su kitais puslapiais ir funkcijomis. 3 lygmens specifikacija „Reikalavimai“ užtikrina sąsajas su kitomis sistemomis, kurios paspartina funkcijas kuriant ir naudojant tinklalapius. 3 lygmens specifikacija „Patikrinimas“ leidžia programoms ir skriptams dinamiškai atnaujinti dokumentų turinį ir struktūrą tuo pat metu užtikrinant, kad dokumentas išlieka teisingas, arba užtikrinant, kad dokumentas taps teisingas.

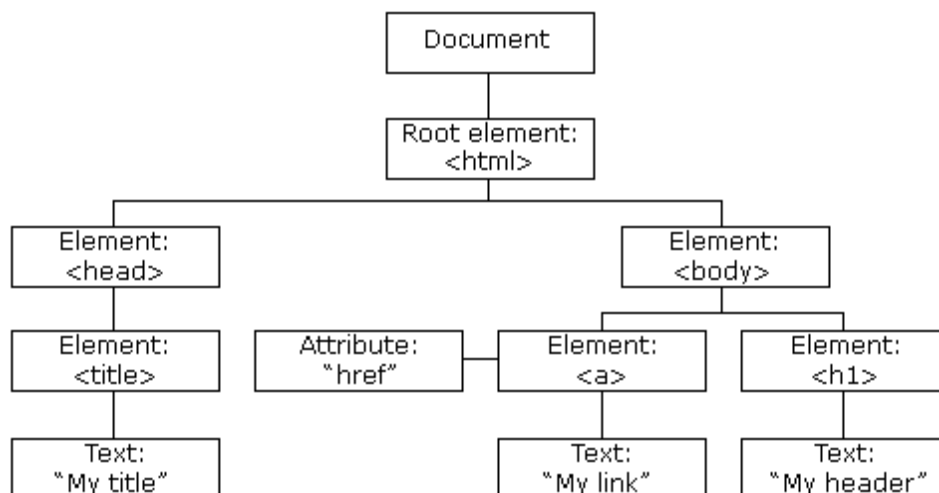
DOM struktūra gali būti perteikta medžio schemoje. Taip yra todėl, kad pagal DOM technologiją viskas HTML dokumente yra laikoma viršūne – visas dokumentas yra dokumento viršūnė, kiekviena HTML žyma yra elemento viršūnė, HTML elementuose esantys tekstai yra teksto viršūnės, kiekvienas HTML požymis yra požymio viršūnė, o komentarai yra komentarų viršūnės. Toliau pateiktas HTML tekstas iliustruoja šių viršūnių sąveiką.

```
<html>
  <head>
    <title>DOM Tutorial</title>
  </head>
  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>
</html>
```

Šiose tekstinėse eilutės visos viršūnės yra susijusios su kitomis viršūnėmis, o kiekviena viršūnė turi tėvą (išskyrus dokumento viršūnę). Pavyzdžiui, viršūnių <head> ir <body> tėvai yra viršūnė <html>, o teksto viršūnės „Hello world!“ tėvas yra viršūnė <p>. Elementų viršūnės gali turėti ir vaikų. Pavyzdžiui, viršūnė <head> turi vieną vaiką – viršūnę <title>. Viršūnė <title> taip pat turi vieną vaiką – teksto viršūnę „DOM Tutorial“. Tais atvejais, kai viršūnės turi bendrą tėvą, jas galima laikyti broliais. Šiame pavyzdyje viršūnės <h1> ir <p> yra broliai, kadangi bendras jų tėvas yra viršūnė <body>. Šia prasme viršūnės gali turėti palikuonių ir protėvių. Palikuonys – tai visos viršūnės, kurios yra viršūnės vaikai, arba tų vaikų vaikai ištisinėje sekoje. Todėl visos teksto viršūnės gali būti viršūnės <html> palikuonys. Pirmosios teksto viršūnės taip pat yra viršūnės <head> palikuonys. Protėviai



yra viršūnės, kurios yra tam tikros viršūnės tėvai, arba šio tėvo tėvai ištiesinėje sekoje. Pavyzdžiui, visų teksto viršūnių protėvis yra viršūnė <html>. Viršūnės tarpusavyje susijusios ir hierarchiniais santykiais. Visos HTML dokumento viršūnės sudaro dokumento medį (arba viršūnių medį). Medis prasideda nuo dokumento viršūnės ir toliau šakojasi, kol pasiekia visas teksto viršūnes žemiausiu medžio lygmeniu. Toliau parodyta šių viršūnių medžio schema.



Document	Dokumentas
Root element: <html>	Šaknies elementas: <html>
Element: <head>	Elementas: <head>
Element: <body>	Elementas: <body>
Element: <title>	Elementas: <title>
Attribute: <href>	Požymis: <href>
Element: <a>	Elementas: <a>
Element: <h1>	Elementas: <h1>
Text: "My title"	Tekstas: „My title“
Text: "My link"	Tekstas: „My link“
Text: "My header"	Tekstas: „My header“

**Pav. HTML dokumento viršūnės sudaro dokumento medį (arba viršūnių medį). Šaltinis:**  
[http://www.w3schools.com/html/dom/dom\\_nodes.asp](http://www.w3schools.com/html/dom/dom_nodes.asp)

## 2.2. Skriptų rašymo pagrindai

### 2.2.1 Supažindinimas su skriptų kalbomis

Skriptų kalbos – tai programavimo kalbos, valdančios taikomąsias programas. Jos atlieka daugelį funkcijų, iš kurių viena svarbiausių yra tinklalapių kūrimas, jų veikimo priežiūra ir keitimas. Toliau išvardytos kelios pagrindinės jų savybės.

- Jos paleidžiamos tiesiai iš jų išeitinio kodo (bet kuri sakinių ir (arba) aprašų, parašytų kuria nors žmogui suprantama kompiuterine programavimo kalba, seka), paprastai tekstinių failų, kuriuose yra kalbai būdingo ženklavimo, ir todėl skiriasi nuo programuotojų (ženklinimas apibrėžiamas kaip teksto ir papildomos informacijos, pvz., struktūrą ir išdėstymą atitinkančių simbolių, derinys).
- Jos naudojamos jau egzistuojantiems komponentams sujungti ir naujai susijusiai užduočiai atlikti. Jos dažnai naudojamos kuriant grafines vartotojo sąsajas (pvz., dinaminis tinklalapis) arba vykdančias daug komandų, kurias kitaip reikėtų įvesti interaktyviai klaviatūra, sulaukus komandos raginimo.
- Kai kurios skriptų kalbos leidžia galutiniams naudotojams rašyti ir derinti trumpas, paprastas ir dažnai konkrečiam domenui skirtas programas (pavyzdžiui, kintamo formato tinklalapius).
- Kitas skriptų kalbos tikslas – padėti su domenu susipažinusiam naudotojui (inžinieriui, statistikui, ekonomistui ir pan.) atlikti konkrečias užduotis (pvz., sukurti atitinkamus tinklalapius).
- Skriptai paprastai rašomi kaip grynasis tekstas ir yra kaskart interpretuojami arba kompiliuojami prieš pat juos naudojant (galima naudoti ir raides, ir simbolius, pvz., < arba >).
- Kai kurios skriptų kalbos yra skirtos konkrečiam domenui arba funkcijai, tačiau dažnai yra galimybė tam tikra kalba rašyti bendresnes programas (pavyzdžiui, „JavaScript“ galima naudoti kelioms interneto taikomosioms programoms).
- Skriptų kalbą ir žemesnio lygmens programavimo kalbą galima naudoti kartu, išnaudojant kiekvienos jų stipriąsias puses konkrečioms problemoms spręsti.
- Skriptų kalbos gali užtikrinti spartesnę programavimą. Kai kurios skriptų kalbos gali būti vykdomos daug lėčiau ir naudoti daugiau atminties, tačiau dažniausia, ypač esant nedideliems skriptams, laikas, sutaupyta rašant skriptą, atperka jo vykdymo trukmę.

Yra daugybė skriptų kalbų tipų. Viena svarbi taikomosioms programoms skirtų skriptų kalbų grupė yra naudojama tinklalapiams kurti. Šiai grupei priskiriamos kalbos „JavaScript“, „JScript“, „VBScript Python PHP“ ir „Perl“.



Kai kurios skriptų kalbos naudojamos kuriant kliento pusės programas, kitos – serverio pusės programas. Kliento pusės skriptus vykdo klientas, kadangi gali reikėti informacijos arba funkcionalumo, kuriuos gali užtikrinti tik klientas, o ne serveris, kadangi naudotojui reikia juos stebėti ar teikti pradinius duomenis arba serveriui trūksta duomenų apdorojimo pajėgumų atliekant operacijas už susijusius klientus. Be to, jeigu klientai operacijas atlieka nesiųsdami duomenų tinkle, jie gali veikti sparčiau, naudoti mažiau juostos pločio ir būti saugesni.

Serverio pusės skriptus tiesiogiai vykdo pats interneto serveris arba plėtinių moduliai. Populiariausi kliento pusės skriptai yra „JavaScript“ ir „VBScript“, tačiau abu jie gali būti naudojami ir serverio pusės skriptams. „Perl“, PHP ir serverio pusės „VBScript“ naudojami serverio pusės skriptams rašyti.

- **„JavaScript“** – tai skriptų kalba, dažniausia vartojama kliento pusės žiniatinkliui kurti. „JavaScript“ paprastai vartojama rašant funkcijas, kurios yra įstatytos HTML puslapiuose arba į juos įtrauktos, bei užtikrinant sąveiką su dokumento objektų modeliu (DOM) puslapyje. „JavaScript“ galima vartoti norint atverti naują langą, kurio dydis, padėtis ir išvaizda bus pakeista (t. y. ar bus matomi meniu, parankinės ir pan.). Kalbą taip pat galima vartoti norint patikrinti į interneto formą įvedamas reikšmes prieš išsiunčiant jas į serverį. „JavaScript“ taip pat leidžia keisti paveikslus, kai per juos vedama pelės žymekliu, siekiant atkreipti naudotojo dėmesį į svarbias nuorodas, kurie rodomi kaip grafiniai elementai. Kadangi „JavaScript“ gali veikti lokaliai naudotojo naršyklėje (o ne nuotoliniame serveryje), ji greitai reaguoja į naudotojo veiksmus. Kalba taip pat gali atpažinti naudotojo veiksmus, pavyzdžiui, atskirus klavišų paspaudimus (kitais nei HTML). „JavaScript“ gali aktyvinti tokias taikomas programas kaip „ArcIMS“, kadangi didžioji naudotojo sąsajos logikos dalis yra parašyta „JavaScript“ kalba, kuri serveriui išsiunčia informacijos užklausas (pvz., nurodyti koordinatas ypatybei atpažinti).
- **„JScript“** – tai aktyvusis skriptų modulis. Jį galima susieti su bet kuria „Windows Script“ palaikančia taikomąja programa, pavyzdžiui, „Internet Explorer“, „Active Server Pages“ (<http://en.wikipedia.org/wiki/ASP.NET> – anglų kalba) ir „Windows Script Host“. Bet kuri „Windows Script“ palaikanti taikomoji programa gali naudoti keletą kalbų – „JScript“, „VBScript“, „Perl“ ir kitas. „JScript“ (ir kitas kalbas) galima naudoti ir paprastoms (pvz., pelei perkelti tinklalapiuose), ir sudėtingoms užduotims (pvz., duomenų bazei atnaujinti arba registravimosi į „Windows NT“ skriptams paleisti).
- **„VBScript“** (*Visual Basic Scripting Edition*) – tai bendrovės „Microsoft“ sukurta aktyvioji skriptų kalba. „VBScript“ skriptas turi būti vykdomas kompiuterio aplinkoje, kurių keletas įtraukta į standartinį „Microsoft Windows“ įdiegimo paketą („Windows Script Host“, „Windows Internet Explorer“). Kai „VBScript“ naudojamas programoje „Microsoft Internet Explorer“, jis veikia panašiai kaip „JavaScript“, t. y. kaip kalba, kuria rašomos funkcijos, įstatytos HTML puslapiuose arba į juos įtrauktos, ir kuri užtikrina sąveiką su dokumento objektų modeliu (DOM) puslapyje bei atlieka užduotis, kurių negalima atlikti naudojant tik HTML. „VBScript“ taip pat galima naudoti kuriant taikomas programas, veikiančias asmeniniame kompiuteryje, kuriame įdiegtas „Microsoft Windows“ paketas. Paprasčiausias pavyzdys – tai skriptas, naudojantis „Windows Script Host“ (WSH) aplinką. Toks skriptas dažniausia yra savarankiškas failas su plėtiniu .vbs. „VBScript“ užtikrina

pagrindinius datos ir laiko, eilučių manipuliavimo, matematinius, naudotojo sąveikos, klaidų apdorojimo ir reguliariusius reiškinius.

- **„Python“** – tai daugiaparaigmė programavimo kalba (leidžianti funkcinį, objektinį ir imperatyvųjį programavimą), kuri naudoja automatinį atminties tvarkymą. Daugiaparaigmė programavimo kalba yra tokia programavimo kalba, kuri palaiko daugiau nei vieną programavimo paradigma. Programavimo paradigma – tai pagrindinis programavimo stilius, numatantis, kaip programavimo kalboje turi būti formuluojami problemų sprendimai. „Python“ kūrėjai siekė sukurti lengvai skaitomą kalbą su paprastu vaizdiniu išdėstymu ir vietoje kitose kalbose naudojamų skyrybos ženklų vartojo anglų kalbos reikšminius žodžius. Kalbai „Python“ reikia mažiau stereotipinio teksto (bet kurio teksto, kuris naudojamas pakartotinai arba gali būti naudojamas naujuose kontekstuose arba programose su nedideliais pakeitimais) nei tradicinėms statiškai renkamos struktūrinėms kalboms (pvz., C). Joje, palyginti su kitomis skriptų kalbomis, taip pat yra mažiau sintaksės išimčių ir ypatingų atvejų. Sakinių blokams atskirti „Python“ kalboje naudojami ne riestiniai skliaustai ar reikšminiai žodžiai, o įtraukos ir matomi tarpai. Po tam tikrų sakinių įtraukos padidinamos, o įtraukų sumažėjimas žymi dabartinio bloko pabaigą.
- **PHP** (*Hypertext Preprocessor*) – tai kompiuterinė programavimo kalba, skirta dinamiams tinklalapiams kurti ir dažniausia vartojama serverio skriptams rašyti. Ją galima vartoti komandinės eilutės sąsajoje arba savarankiškose grafinėse programose. PHP yra populiarus plačios paskirties skriptų kalba, tinkama žiniatinkliui kurti, kurią galima įstatyti į HTML. PHP paprastai veikia interneto serveryje, PHP kodą naudoja kaip pradinis duomenis ir kuria tinklalapius kaip rezultatą. Tačiau ją galima naudoti ir komandinės eilutės skriptams bei kliento pusės grafinės sąsajos programoms. PHP kodą analizuoja tik tarp skirtukų.
- **„Perl“** (*Practical Extraction and Reporting Language*) – tai plačios paskirties programavimo kalba, kuri buvo sukurta manipuluoti tekstu, o dabar naudojama pačioms įvairiausioms užduotims, įskaitant sistemų administravimą, interneto kūrimą, tinklo programavimą ir bendrą vartotojo sąsajos kūrimą. Jos ypatybėms priskiriamas skirtingų programavimo paradigmu (procedūrinio, objektinio ir funkcinio stilių) palaikymas, įtaisytais teksto apdorojimo palaikymas ir didelė kitų asmenų sukurtų modulių kolekcija. Kitaip nei „Python“ kalboje, „Perl“ akcentuojamas bendrų taikomosioms programoms skirtų užduočių palaikymas, pvz., naudojant įtaisytuosius reguliariusius reiškinius, failų skenavimo ir ataskaitų generavimo funkcijas. Todėl ji labiau tinka taikomosioms programoms ir taikoma šiek tiek siauresnėje srityje. „Python“ kalboje akcentuojamas bendrų programavimo metodikų, pavyzdžiui, duomenų struktūrų projektavimo ir objektinio programavimo, palaikymas.

## 2.2.2 HTML ir DOM

Kaip jau buvo aptarta 1 paskaitoje, HTML yra populiariausia tinklalapių ženklinimo kalba. MIME tipo teksto ar html turinys taip pat dažnai vadinamas HTML. Platesne prasme HTML yra bendrinis terminas, vartojamas ir XML pagrindu sukurtoms kalboms (pvz., XHTML 1.0 ir jos vėlesnes versijas), ir pagal SGML sukurtoms kalboms nusakyti (pvz., HTML 4.01 ir ankstesnes versijas). Yra keletas HTML versijų. Naujausia ir galutinė HTML versija yra HTML 4.01, kuri buvo paskelbta kaip W3C rekomendacija ir ISO/IEC 15445:2000 (2008 m. sausio mėn.).

HTML apibrėžia keletą elementų turinio duomenų tipų – skriptų duomenis ir stiliaus lapų duomenis – bei daugybę požymių reikšmių tipų, įskaitant ID, vardus, URI, skaičius, ilgio vienetus, kalbas, terpės deskriptorius, spalvas, ženklų kodavimą, datas ir laikus ir pan. Visi šie duomenų tipai yra ženklų duomenų specializacijos. Dokumento objektų modelis (DOM) leidžia pasiekti visus HTML dokumento elementus, taip pat metodus ir savybes, skirtas HTML elementams papildyti, perkelti, pakeisti arba pašalinti.

HTML dokumento pradžioje gali būti **dokumento tipo aprašas** (neoficialiai – „DOCTYPE“). Dokumento tipo apraše, kurį žymi DOCTYPE, yra kompiuteriui suprantama gramatika, nurodanti, koks tokį aprašą atitinkantis dokumento turinys yra leidžiamas ir draudžiamas. Naršyklės nebūtinai nuskaito dokumento tipo aprašą. Populiariausios grafinės naršyklės DOCTYPE aprašus (arba jų nebuvimą) ir kitus kodo pradžioje pateikiamus duomenis naudoja nustatyti, kurią veikseną pasirinkti.

Pavyzdžiui:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

Šiame apraše nurodytas HTML 4.01. aprašas „Strict“, kuriame nėra tokių pateikimo elementų kaip <font>, o formatuojama naudojant pakopinius stiliaus lapus ir žymas <span> ir <div>. SGML tikrinimo įrankiai skaito dokumento tipo aprašą tam, kad išanalizuotų dokumentą ir atliktų patikrinimą. Šiuolaikinėse naršyklėse HTML 4.01 dokumento tipo aprašas „Strict“ aktyvina standartų pateikimo veikseną, skirtą CSS, o ne suderinamumo veikseną (angl. *quirks mode*).

HTML 4.01 taip pat galimi „Transitional“ ir „Frameset“ dokumento tipo aprašai. Dokumento tipo aprašas „Transitional“ buvo skirtas pakeitimams, kurie buvo padaryti apraše „Strict“, pamažu įvesti, o aprašas „Frameset“ buvo skirtas dokumentams su rėmeliais. HTML dokumentus galima pateikti tomis pačiomis priemonėmis, kaip ir bet kurį kitą kompiuterinį failą, tačiau HTML dokumentai dažniausiai pateikiami viena iš dviejų formų: per HTTP serverius arba el. paštu.

Pasaulinį žiniatinklį daugiausia sudaro HTML dokumentai, kurie iš interneto serverio perduodami į interneto naršyklę naudojant hiperteksto perdavimo protokolą (HTTP). Tačiau HTTP galima naudoti ne tik HTML, bet ir paveikslams, garsui ir kitam turiniui perduoti. Tam, kad interneto naršyklė žinotų, kaip apdoroti gautą dokumentą, kartu su perduodamu HTML dokumentu būtina nurodyti ir jo failo formatą.

### 2.2.3 „JavaScript“ pagrindai

Šioje dalyje apibūdinama „JavaScript“ kalba. Medžiaga suskirstyta į tokius poskyrius:

- „JavaScript“ apibūdinimas
- „JavaScript“ ir interneto naršyklės
- „JavaScript“ sakiniai
- „JavaScript“ komentarai
- „JavaScript“ kintamieji
- Pagrindiniai „JavaScript“ duomenų tipai

- Operacijų ženklai
- Sakiniai
- Funkcijos
- „JavaScript“ įvykiai

Po šios dalies dvi atskiros dalys bus skirtos „JavaScript“ objektų hierarchijai ir „JavaScript“ objektų, naudojamų „JavaScript“ paveikslų planams, pavyzdžiams.

„JavaScript“ apibūdinimas. „JavaScript“ – tai skriptų kalba, dažniausia naudojama kliento pusės žiniatinkliui kurti. Ją sudaro vykdomojo kompiuterinio kodo eilutės. Kalba išpopuliarėjo dėl jos vartojimo tinklalapiuose (kaip kliento pusės „JavaScript“), tačiau ji vartojama ir skriptų prieigai prie objektų, įstatytų kitose taikomosiose programose. Nuo 2006 m. naujausia kalbos versija yra „JavaScript 1.7“.

Kadangi „JavaScript“ veikia ne serveryje, o kliente, ji gali greitai reaguoti į naudotojo veiksmus, todėl susidaro įspūdis, kad programa reaguoja geriau. Be to, „JavaScript“ gali atpažinti naudotojo veiksmus, ko negali HTML, pavyzdžiui, atskirų klavišų paspaudimus. Ši savybė išnaudojama tokiose taikomosiose programose, kaip „ArcIMS“: didelė dalis vartotojo sąsajos logikos yra parašyta „JavaScript“, kuri serveriui išsiunčia informacijos užklausas (pvz., nurodyti koordinatas ypatybei atpažinti).

**„JavaScript“ ir interneto naršyklės.** „JavaScript“ taip pat naudojama puslapio kompozicijai gerinti, formoms patvirtinti, naršyklėms atpažinti, slapukams kurti ir pan. Ji veikia su visomis pagrindinėmis interneto naršyklėmis, pvz., „Internet Explorer“, „Mozilla“, „Firefox“, „Netscape“ ir „Opera“. „JavaScript“ buvo sukurta HTML puslapių interaktyvumui padidinti ir funkcijoms, kurios įstatytos arba įtrauktos į HTML puslapius, kurti bei sąveikauti su puslapio DOM. Keletas paprastų tokio naudojimo pavyzdžių:

- Atverti arba iškelti naują langą, kurio dydis, padėtis ir „išvaizda“ bus pakeista (t. y. ar bus matomi meniu, parankinės ir pan.) programinėmis priemonėmis.
- Interneto formos pradinėms reikšmėms patikrinti siekiant įsitikinti, kad jos bus priimtos prieš perduodant į serverį.
- Paveikslams keisti pelės žymekliu siekiant atkreipti naudotojo dėmesį į svarbias nuorodas, kurie rodomi kaip grafiniai elementai.

HTML kūrėjai „JavaScript“ naudoja kaip programinę priemonę dinaminiam tekstui į HTML puslapį įterpti. „JavaScript“ gali reaguoti į įvykius: galima nustatyti, kad skriptas suveiktų atlikus tam tikrą veiksmą, pavyzdžiui, kai įkraunamas puslapis arba naudotojas spragteli ant HTML elemento. „JavaScript“ gali skaityti, rašyti ir taip keisti HTML. „JavaScript“ galima naudoti formos duomenims patikrinti prieš perduodant juos serveriui, tad šiam nereikia jų apdoroti. „JavaScript“ gali atpažinti lankytojo naršyklę ir įkrauti konkrečiai naršyklei skirtus puslapius. „JavaScript“ taip pat galima naudoti slapukams kurti, informacijai saugoti ir ieškoti lankytojo kompiuteryje.

Populiariausia „JavaScript“ veikimo aplinka (angl. *host environment*) – interneto naršyklė. Interneto naršyklės paprastai naudoja viešąsias API sąsajas „kompiuterio objektams“ (angl. *host objects*), kurie turi atspindėti DOM „JavaScript“ skripte, sukurti. Kita veikimo aplinka – interneto serveris. „JavaScript“ interneto serveris parodo kompiuterio objektus, atitinkančius HTTP užklausas ir atsako objektus (angl. *response objects*), kuriuos „JavaScript“ programa gali naudoti kurdamą tinklalapius.

Paprastas tinklalapio, kuriame yra „JavaScript“ (naudojant HTML sintaksę), pavyzdys būtų toks:

```
<!DOCTYPE html>
<html>
  <head><title>simple page</title></head>
  <body>
    <script>
      document.write('Hello World!');
    </script>
    <noscript>
      Your browser does not support JavaScript.
    </noscript>
  </body>
</html>
```

Kalbant apie tai, kur turi būti „JavaScript“, puslapyje „JavaScript“ skriptai yra vykdomi puslapį įkėlus į naršyklę. Kartais naudotojas skriptą paleidžia įkėlus puslapį arba naudotojui sukėlus įvykį. Jeigu skriptai turi būti vykdomi juos iškvietus arba pradėjus įvykį, jie rašomi antraštės sekcijoje. Taip užtikrinama, kad skriptas įkraunamas anksčiau, nei panaudojamas. Jeigu skriptai turi būti vykdomi įkeliant puslapį, jie yra pagrindinės dalies sekcijoje, kuri generuoja puslapio turinį. Skriptai taip pat gali būti ir pagrindinės dalies, ir antraštės sekcijoje, jeigu tokios funkcijos yra naudojamos kartu.

„JavaScript“ sintaksę sudaro taisyklės, nustatančios, kas „JavaScript“ kalboje laikytina teisinga programa. Kintamiesiems tipai nepriskiriami, o bet kuri reikšmė gali būti priskirta bet kuriam kintamajam. Kintamuosius galima paskelbti naudojant komandą „var“. Šie kintamieji turi statines galiojimo sritis, todėl, nurodytą kintamąjį galima pasiekti bet kurioje funkcijos vietoje, kurioje jis yra paskelbtas. Kintamieji, paskelbti ne funkcijoje, ir kintamieji, pirmą kartą panaudoti funkcijose jų nepaskelbus su komanda „var“, yra globalieji. Toliau pateikiamas kintamųjų paskelbimo ir globaliųjų reikšmių pavyzdys:

```
x = 0; // A global variable
var y = 'Hello!'; // Another global variable

function f(){
  var z = 'foxes'; // A local variable
  twenty = 20; // Global because keyword var is not used
  return x; // We can use x here because it is global
}
// The value of z is no longer available
```

„JavaScript“ skriptai skiria didžiąsias ir mažąsias raides. Objektų vardai dažnai prasideda didžiąja raide, o funkcijų arba kintamųjų – mažąja. Tarpai, tabuliavimo žymės ir eilučių skirtukai už eilučių konstantų ribų yra vadinami matomais tarpais. Kitaip nei C kalboje, matomi tarpai „JavaScript“ pirminiame tekste gali tiesiogiai paveikti semantiką. Dėl vadinamojo „kablitaškojo įterpimo“ metodo, bet kuris sakinytis, kuris yra struktūriškai tvarkingas analizuojant eilučių skirtuką, bus laikomas užbaigtu (tarsi kablitaškis būtų įterptas prieš pat eilutės skirtuką). Programuotojams rekomenduojama aiškiai nurodyti

sakinius užbaigiančius kabliataškius, nes skriptą bus lengviau skaityti, taip pat bus sumažinta automatinio kabliataškio įterpimo tikimybė.

Nereikalingi matomi tarpai, tarpų ženklai, kurių nereikia teisingai sintakse, gali padidinti nereikalingą erdvę ir atitinkamai failų .js dydį. Failų dydžio problemą lengviausia spręsti nurodant serveriui naudoti zip pakavimą. Toks pakavimas bus tinkamas labiau nei bet kuris matomų tarpų analizatorius ir sumažins visų kitų failų, kuriuos įkelia serveris, dydį. Šis metodas gali būti taikomas su kabliataškiais arba be jų.

**„JavaScript“ sakiniai.** „JavaScript“ sakiniai – tai komandos, nurodančios naršyklei, ką daryti. Pavyzdžiui, toliau pateiktu formatu naršyklei nurodoma parašyti „Sveikas“.

```
document.write("Hi There");
```

Paprastai sakinio pabaigoje rašomas kabliataškis, tačiau jis nėra privalomas, kadangi naršyklė eilutės pabaigą supranta kaip sakinio pabaigą.

**„JavaScript“ komentarai.** „JavaScript“ komentarų sintaksė atitinka C++ sintaksę, kaip parodyta šiame pavyzdyje:

```
// single line  
  
/* multi-line  
   comment */
```

Komentarus galima naudoti norint palengvinti kodo skaitymą arba paaiškinti „JavaScript“. Vienos eilutės komentarai pradedami „//“. Pavyzdžiui:

```
<script type="text/javascript">  
// This will write a header:  
document.write("<h1>This is a header</h1>");  
// This will write two paragraphs:  
document.write("<p>This is a paragraph</p>");  
document.write("<p>This is another paragraph</p>");  
</script>
```

Kelių eilučių komentarai pradedami „/\*“ ir baigiami „\*/“. Pavyzdžiui:

```
<script type="text/javascript">
/*
The code below will write one header and two paragraphs
*/
document.write("<h1>This is a header</h1>");
document.write("<p>This is a paragraph</p>");
document.write("<p>This is another paragraph</p>");
</script>
```

Komentarus galima naudoti ir tada, kai norima, kad nebūtų vykdoma viena arba kelios kodo eilutės. Komentarus galima rašyti ir eilutės pabaigoje.

**„JavaScript“ kintamieji.** Juos galima apibūdinti kaip informacijos saugojimo „talpyklas“. Pavyzdžiui:

```
x = 8; length =79.40; truckname = "Ford"
```

Kintamojo reikšmę galima keisti skripte. Naudotojas, norėdamas perskaityti arba pakeisti reikšmę, gali nurodyti kintamojo pavadinimą. Kintamųjų pavadinimuose skiriamos didžiosios ir mažosios raidės. Jie turi prasidėti raide arba pabrauktu ženklų. Kintamieji naudojami duomenims saugoti. Pavyzdžiui:

```
<html>
<body>

<script type="text/javascript">
var name="Hege";
document.write(name);
document.write("<br>");
name="Tove";
document.write(name);
</script>

<p>The script above declares a variable, assigns a value to it, change the
value, and displays the value again.</p>
</body>
</html>
```

Kintamąjį galima sukurti naudojant sakinį, pavyzdžiui:

```
var strname = some value
```

Kintamajam reikšmė gali būti priskiriama keliais būdais. Pavyzdžiui, toliau pateiktame pavyzdyje kintamojo „strname“ reikšmė yra „Hege“.



```
var strname="Hege"; //or
strname="Hege";
```

## Pagrindiniai „JavaScript“ duomenų tipai

Skaičiai. Skaičiai „JavaScript“ kalboje pateikiami dvejetainėje sistemoje kaip IEEE-754 dvigubo tikslumo skaičiai, todėl užtikrinamas maždaug 14 arba 15 reikšminių skaitmenų tikslumas. Kadangi skaičiai yra dvejetainiai, jie ne visuomet tiksliai atitinka dešimtainės sistemos skaičius, ypač trupmenas. Todėl kyla sunkumų formatuojant skaičius išvediniams, kuriems „JavaScript“ neturi įtaisytųjų metodų. Pavyzdžiui:

```
alert(0.94 - 0.01); // displays 0.9299999999999999
```

Todėl skaičius, formatuojamus išvediniui, reikėtų apvalinti.

Skaičius galima nurodyti bet kuriuo iš šių formatų:

```
345; // an "integer", although there is only one numeric type in JavaScript
34.5; // a floating-point number
3.45e2; // another floating-point, equivalent to 345
0377; // an [[octal]] integer equal to 255
0xFF; // a [[hexadecimal]] integer equal to 255, the letters A-F may be upper-
or lowercase
```

Kai naudojamas kaip konstruktorius, yra sukuriamas skaitmeninis apgaubiamasis objektas (angl. wrapper object) (žr. objekto tipo aprašymą toliau):

```
myNumericWrapper = new Number( 123.456 );
```

Masyvai. Masyvas – tai duomenų struktūra, sudaryta iš elementų grupės, pasiekiamos indeksuojant. Kiekvienam elementui gali būti priskirtas toks pats duomenų tipas, o masyvas užima ištisinę atminties sritį. Daugelyje programavimo kalbų yra įtaisytasis masyvo duomenų tipas. „JavaScript“ kalboje masyvams priskiriama ilgio savybė, kuri yra didesnė nei didžiausias sveikųjų skaičių indeksas, naudojamas masyve. Ji yra automatiškai atnaujinama, jeigu sukuriama savybė su dar didesniu indeksu. Jeigu rašomas mažesnis ilgio savybės skaičius, pašalinami didesnieji indeksai. Ši ilgio savybė yra vienintelė ypatinga masyvų savybė, kuri juos išskiria iš kitų objektų.

Masyvų elementus galima pasiekti naudojant paprastą objekto savybės prieigos formatą:

```
myArray[1];
myArray["1"];
```



Abu pateikti pavyzdžiai yra lygiaverčiai. Alternatyviai vaizduojant skaičių negalima naudoti „dot“ formato arba eilučių:

```
myArray.1;      // syntax error
myArray["01"]; // not the same as myArray[1]
```

Masyvui paskelbti galima naudoti masyvo literalą arba masyvo konstruktorių.

```
myArray = [0,1,,,4,5]; // array with length 6 and 4 elements
myArray = new Array(0,1,2,3,4,5); // array with length 6 and 6 elements
myArray = new Array(365); // an empty array with length 365
```

Masyvai kuriami taip, kad atmintį naudotų tik apibrėžti elementai; tai vadinamieji „reti masyvai“. Jeigu nustatoma, kad `myArray[10] = 'someThing'` ir `myArray[57] = 'somethingOther'`, bus naudojama vieta tik šiems dviem elementams, kaip ir bet kurio kito objekto atveju. Masyvo ilgis vis tiek bus nurodomas kaip 58.

Objekto aprašo literalą galima naudoti kuriant objektus, kurie veikia kaip asociatyvieji masyvai kitose kalbose:

```
dog = {"color":"brown", "size":"large"};
dog["color"]; // this gives you "brown"
```

Objekto ir masyvo aprašo literalus galima naudoti greitai kuriant masyvus, kurie yra asociatyvieji, daugiamačiai arba ir asociatyvūs, ir daugiamačiai.

```
cats = [{"color":"brown", "size":"large"},
        {"color":"black", "size":"small"}];
cats[0]["size"]; // this gives you "large"

dogs = {"rover":{"color":"brown", "size":"large"},
        "spot":{"color":"black", "size":"small"}};
dogs["spot"]["size"]; // this gives you "small"
```

**Eilutės.** „JavaScript“ kalboje eilutes sudaro ženklų seka. Eilutes galima kurti ženklų sekas įrašant tarp dvigubų arba viengubų kabučių, pavyzdžiui:

```
var greeting = "Hello, world!";
var another_greeting = 'Greetings, people of Earth.';
```

„Mozilla“ tipo naršyklėse atskirus ženklus eilutėje galima pasiekti (kaip eilutes, kuriose yra vienintelis ženklas) per tą patį formatą kaip ir masyvuose:

```
var h = greeting[0]; // Now h contains 'H' - Works in Mozilla based browsers
```

Tačiau naršyklėje „Internet Explorer“ atskirus ženklus galima pasiekti taikant metodą `charAt()` (numatytą eilučių klasėje). Šiam atskirų ženklų pasiekimo metodui teikiama pirmenybė, kadangi jis tinka ir „Mozilla“ tipo naršyklėms.

```
var h = greeting.charAt(0); // Now h contains 'H' - Works in both Internet Explorer
                               // and Mozilla based browsers
```

Tačiau „JavaScript“ eilutės yra nekintamos, t. y. sukūrus jų nebegalima keisti, nes gali atsirasti klaidos pranešimas:

```
greeting[0] = "H"; // ERROR
```

Lygybės ženklą („==“) dviem eilutėms galima taikyti tik tada, jeigu šių eilučių turinys yra toks pats, t. y. jos yra to paties ilgio ir raidės yra to paties registro. Todėl:

```
var x = "world";
var compare1 = ("Hello, " + x == "Hello, world"); // Now compare1 contains true
var compare2 = ("Hello, " + x == "hello, world"); // Now compare2 contains false
since the first characters of both operands are not of the same case
```

**Objektai.** Tipai paprastai skirstomi į *primityvus* ir *objektus*. Objektai yra tapatybę turintys vienetai (jie yra lygūs tik sau patiems) ir susieja savybių pavadinimus su reikšmėmis („lizdai“ prototipais grindžiamo programavimo terminologijoje). „JavaScript“ objektai dažnai klaidingai apibūdinami kaip asociatyvieji masyvai arba maišos funkcijos, tačiau taip nėra.

„JavaScript“ kalboje yra keletas įtaisytyjų objektų – masyvas, loginis kintamasis, data, funkcija, matematinė operacija, skaičius, objektas, reguliarusis reiškinytis ir eilutė. Kiti objektai yra „kompiuterio objektai“ (angl. host objects), kuriuos apibrėžia ne kalba, o veikimo laiko aplinka. Pavyzdžiui, naršyklėje tipiniai kompiuterio objektai priklauso DOM (langas, forma, nuorodos ir pan.).

Objektus galima kurti naudojant aprašo, inicijavimo (angl. initializer) arba konstruktoriaus funkciją:

```
// Declaration
var anObject = new Object();

// Initialiser
var objectA = {};
var objectB = {'index1': 'value 1', 'index2': 'value 2'};
```

```
// Constructor (see below)
```

Patys svarbiausi objektai „JavaScript“ kalboje atlieka žodynų funkcijas. Šiuose žodynuose bet kurio tipo reikšmei gali būti priskiriamas raktas, taip sukuriant eilutę. Objektus su priskirtomis reikšmėmis galima kurti tiesiogiai naudojant objekto literalo formatą:

```
var o = {name: 'My Object', purpose: 'This object is utterly without purpose.',
answer: 42};
```

Naudojant įprastą taško („.“) formatą arba panašią į masyvų sintaksę galima kurti, nustatyti ir skaityti atskiras objektų savybes:

```
var name = o.name; // name now contains 'My Object'
var answer = o['answer']; // answer now contains 42
```

Naudojant objekto literalus ir masyvo literalus galima kurti lanksčias duomenų struktūras:

```
var myStructure = {
  name: {
    first: "Mel",
    last: "Smith"
  },
  age: 33,
  hobbies: [ "chess", "jogging" ]
};
```

Tai yra JSON (JavaScript Object Notation, <http://en.wikipedia.org/wiki/JSON> – anglų kalba), paprasto formato, kuriame duomenims perduoti naudojama į „JavaScript“ panaši sintaksė, pagrindas.

**„JavaScript“ operacijų ženklai:**

***Aritmetiniai*** dviečių operacijų ženklai:

```
+      Addition
-      Subtraction
*      Multiplication
/      Division (returns a floating-point value)
%      Modulus (returns the integer remainder)
```

Operacijos ženklas + yra perkrautas, t. y. jis naudojamas eilutėms sujungti, skaičiams sudėti ir eilutėms paversti į skaičius. Be to, naudojamas reguliariajame reiškinyje, jis turi ypatingą prasmę. Pavyzdžiui:

```

// Concatenate 2 strings
var a = 'This';
var b = ' and that';
alert(a + b); // displays 'This and that'

// Add two numbers
var x = 2;
var y = 6;
alert(x + y); // displays 8

// Adding a string and a number results in concatenation
alert( x + '2'); // displays 22

// Convert a string to a number
var z = '4'; // z is a string (the digit 4)
alert( z + x); // displays 42
alert( +z + x); // displays 6
    
```

### Aritmetiniai vienviečių operacijų ženklai:

```

-    Unary negation (reverses the sign)
++   Increment (can be prefix or postfix)
--   Decrement (can be prefix or postfix)
    
```

### „JavaScript“ *priskyrimo* operacijų ženklai:

```

=     Assign
+=    Add and assign
-=    Subtract and assign
*=    Multiply and assign
/=    Divide and assign

var x = 1;
x *= 3;
document.write( x ); // displays: 3
x /= 3;
document.write( x ); // displays: 1
x -= 1;
document.write( x ); // displays: 0
    
```

### „JavaScript“ *palyginimo* operacijų ženklai:

```

==     Equal
!=     Not equal
>     Greater than
>=    Greater than or equal to
<     Less than
<=    Less than or equal to

===    Identical (equal and of the same type)
    
```

```
!== Not identical
```

„JavaScript“ **loginio palyginimo** operacijų ženklai. „JavaScript“ kalba turi tris loginių operacijų ženklus: && (loginis IR), || (loginis ARBA), ir ! (loginis NE):

```
&&    and  
||    or  
!     not (logical negation)
```

Kai atliekama loginė operacija, visos „JavaScript“ reikšmės apskaičiuojamos kaip teisingos, nebent reikšmė atitinka pačią loginę operaciją false, skaičių 0, nulinio ilgio eilutę arba vieną iš specialiųjų reikšmių null, undefined arba NaN. Loginę funkciją galima naudoti būtent šiam konvertavimui atlikti:

```
Boolean( false ); // returns false
Boolean( 0 ); // returns false
Boolean( 0.0 ); // returns false
Boolean( "" ); // returns false
Boolean( null ); // returns false
Boolean( undefined ); // returns false
Boolean( NaN ); // returns false
```

Vienvietės operacijos ženklas ! visų pirma apskaičiuoja savo operandą loginėje aplinkoje ir tada grąžina priešingą loginę reikšmę:

```
var a = 0;
var b = 9;
!a; // evaluates to true, same as (Boolean( a ) == false)
!b; // evaluates to false, same as (Boolean( b ) == true)
```

Ankstyvose „JavaScript“ ir „JScript“ versijose operacijų ženklai && ir || veikė kaip ir jų atitikmenys kitose C pagrindu sukurtose programavimo kalbose, t. y. visuomet buvo grąžinama loginė reikšmė:

```
x && y; // returns true if x AND y evaluate to true: (Boolean( x ) == Boolean( y ) == true), false otherwise
x || y; // returns true if x OR y evaluates to true, false otherwise
```

Naujesnėse versijose šie operacijų ženklai grąžina vieną iš savo operandų:

```
expr1 && expr2; // returns expr1 if it evaluates to false, otherwise it returns expr2
expr1 || expr2; // returns expr1 if it evaluates to true, otherwise it returns expr2
```

Šis naujoviškas veikimas nėra gerai žinomas net ir patyrusiems „JavaScript“ programuotojams, todėl gali kilti problemų, jeigu tikimasi gauti loginę reikšmę.

Supaprastintose loginėse operacijose reiškiny vertinamas iš kairės į dešinę tol, kol randamas atsakymas. Pavyzdžiui, a || b yra automatiškai teisinga, jeigu a yra teisinga, o a && b yra neteisinga, jeigu a yra neteisinga. Todėl visiškai nebūtina apskaičiuoti b.

**Bitinių dviviečių operacijų ženklai:**

```

&      And
|      Or
^      Xor

<<    Shift left  (zero fill)
>>    Shift right (sign-propagating); copies of the leftmost bit (sign bit) are
shifted in from the
      left.
>>>  Shift right (zero fill)

      For positive numbers, >> and >>> yield the same result.

```

**Bitinių vienviečių operacijų ženklas:**

```

~      Not (inverts the bits)

```

**Eilučių operacijų ženklai:**

```

=      Assignment
+      Concatenation
+=     Concatenate and assign

```

**Pavyzdžiai:**

```

str = "ab" + "cd"; // "abcd"
str += "e";        // "abcde"

```

**„JavaScript“ sakiniai:**

**Sąlyginiai sakiniai.** „JavaScript“ kalboje jie naudojami tada, kai skirtingiems sprendimams reikalingi skirtingi veiksmai:

- sakinys `if` leidžia vykdyti tam tikrą kodą tik tada, jeigu tenkinama nurodyta sąlyga
- sakinys `if...else` leidžia vykdyti vieną kodą tada, jeigu sąlyga tenkinama, ir kitą kodą, jeigu sąlyga netenkinama
- sakinys `if...else if...else` leidžia pasirinkti vieną iš daugelio kodo bloką, kuris turi būti vykdomas

### Šių sakinių pavyzdžiai:

```
<script type="text/javascript">
//Write a "Good morning" greeting if the time is less than 10var d=new Date();
var time=d.getHours();
if (time<10)
{
document.write("<b>Good morning</b>");
}
</script>
```

Jeigu yra lyginami kintamieji, būtina vieną šalia kito naudoti du lygybės ženklus (==).

### Sakinio if...else sintaksė:

```
if (condition)
{
code to be executed if condition is true
}
else
{
code to be executed if condition is not true
}
```

### Sakinio if...else pavyzdys:

```
<script type="text/javascript">
//If the time is less than 10, you will get a "Good morning" greeting.
//Otherwise you will get a "Good day" greeting.var d = new Date();

var time = d.getHours();
if (time < 10)
{
document.write("Good morning!");
}
else
{
document.write("Good day!");
}
</script>
```



Sakinio `if...else if...else` sintaksė:

```

if (condition1)
{
code to be executed if condition1 is true
}
else if (condition2)
{
code to be executed if condition2 is true
}
else
{ code to be executed if condition1 and condition2 are not true
}

```

Sakinio `if...else if...else` pavyzdys:

```

<script type="text/javascript">
var d = new Date()
var time = d.getHours()
if (time<10)
{
document.write("<b>Good morning</b>");
}
else if (time>10 && time<16)
{
document.write("<b>Good day</b>");
}
else
{
document.write("<b>Hello World!</b>");
}
</script>

```

Perjungimo sakiny ( **switch** ) leidžia pasirinkti vieną iš daugelio kodo blokų, kuris turi būti vykdomas. Jo sintaksė:

```

switch (expr)
{
case VALUE:
statements;
break;
case VALUE:
statements;
break;
default:
statements;
break;
}

```

Nors parametras `break` nėra privalomas, dažniausia rekomenduojama jį naudoti, kitaip kodas bus vykdomas iki kito bloko `case` pagrindinės dalies. Paskutinio varianto pabaigoje įrašykite lūžio sakinį, kuris bus tarsi apsaugos priemonė, jeigu vėliau bus įterpta papildomų

variantų. Variantinių sakinių reikšmėms galima naudoti eilutes. Būtina naudoti skliaustus {}.

Sakinio **for loop** su reiškiniu sintaksė yra tokia:

```
for (initial-expression; cond-expression; expression evaluated after each loop-  
round)  
{  
  statements;  
}
```

Pavyzdžiui, *for loop* galima naudoti norint spausdinti skaičius nuo 1 iki 10 įspėjimo lauke:

```
var msg = "";  
  
for (var x = 1; x <= 10; x++)  
{  
  msg = msg + x + "\n";  
}  
  
alert(msg);
```

Sakinio **for ... in loop** sintaksė yra tokia:

```
for (var property-name in object-name)  
{  
  statements using object-name[property-name];  
}
```

**For ... in loop** kartojamas visose išvardijamose objekto savybėse arba masyvo indeksų objektuose. Pavyzdžiui, šį sakinį galima naudoti norint atpažinti interneto naršyklės tipą. Įvairios interneto naršyklės rodo skirtingas savybes, jeigu yra naudojamas sakiny *for ... in loop*. Teoriškai tai kontroliuoja vidinės būklės savybė, kurią apibrėžia ECMA skripto standartas „DontEnum“, tačiau praktiškai kiekviena naršyklė grąžina šiek tiek skirtingas savybes.

Sakinio **while loop** sintaksė yra tokia:

```
while (cond-expr)  
{  
  statements;  
}
```

Pavyzdžiui, *while loop* galima naudoti norint spausdinti skaičius nuo 1 iki 10 įspėjimo lauke:

```
var msg = "";
var x = 1;
while (x <= 10)
{
    msg = msg + x + "\n";
    x++;
}
alert(msg);
```

Sakinio **do ... while** sintaksė yra tokia:

```
do {
    statements;
} while (cond-expr);
```

Sakinio **with** sintaksė yra tokia:

```
with(document) {
    var a = getElementById('a');
    var b = getElementById('b');
    var c = getElementById('c');
};
```

Atkreipkite dėmesį, kad prieš kiekvieną `getElementById()` nėra įrašyta `document`. Sakinys *with* nurodo objektą, kuris bus konkrečiai naudojamas pagrindinėje sakinio dalyje. Sakinys *with* yra patogus, kadangi nereikia pakartotinai rinkti to paties objekto pavadinimų. Naudojant sakinį *with*, nuorodą į objektą galima sutrumpinti taip:

```
with (document)
{
    write("Hello from JavaScript");
    write("<<br />>");
    write("You can write what you like here");
}
```

**„JavaScript“ funkcijos.** Funkcija – tai blokas su (galimai tuščiu) parametų sąrašu, kuriam paprastai suteikiamas pavadinimas. Funkcija gali gražinti reikšmę. „JavaScript“ funkcijos sintaksė:

```
function function-name(arg1, arg2, arg3)
{
  statements;
  return expression;
}
```

Taip pat galimos anoniminės funkcijos, pavyzdžiui:

```
var fn = function(arg1, arg2)
{
  statements;
  return expression;
};
```

Pavyzdžiui, originalusis Euklido algoritmas, skirtas dviejų skaičių didžiausiam bendram dalikliui surasti, yra geometrinis sprendimas, kai iš ilgesnio segmento atimamas trumpesnis:

```
function gcd(segmentA, segmentB) {
  while (segmentA != segmentB) {
    if (segmentA > segmentB)
      segmentA -= segmentB;
    else
      segmentB -= segmentA;
  }
  return segmentA;
}
```

Funkciją iškviečiant nurodytų argumentų (pvz., segmentA, segmentB) skaičius nebūtinai sutampa su argumentų skaičiumi funkcijos apibrėžime; jame nurodytas argumentas, neturintis atitikmens kreipinyje, turės reikšmę undefined.

Pačioje funkcijoje argumentus taip pat galima pasiekti per sąrašą arguments; taip galima pasiekti visus argumentus, naudojant indeksus (pvz., arguments[0], arguments[1], ... arguments[n]), įskaitant ir tuos, kurie nėra nurodyti. Atkreipkite dėmesį, kad nors argumentų sąrašas turi savybę .length, tai nėra masyvas. Sąrašas neturi tokių metodų kaip .slice(), .sort() ir pan. Pagrindiniams duomenų tipams (eilutėms, sveikiems skaičiams ir pan.) yra naudojamos *reikšmės*, o objektams – *nuorodos*.

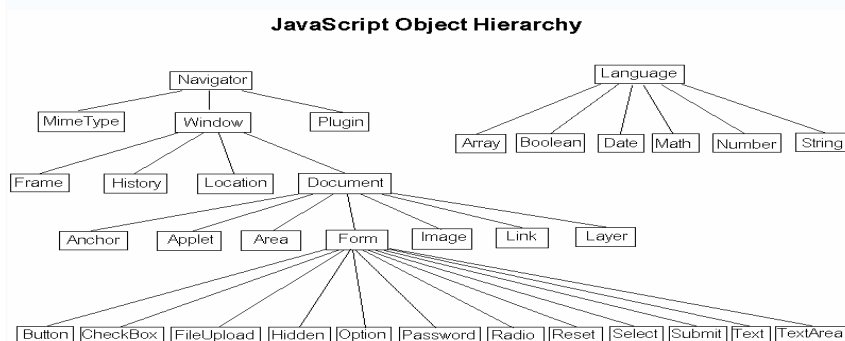
**„JavaScript“ įvykiai.** Be objektinio programavimo, įvykiai yra antra svarbiausia „JavaScript“ savybė, palaikanti įvykių valdomą programavimą. Įvykiai – tai veiksmai, kuriuos gali atpažinti „JavaScript“. Kiekvienas tinklalapio elementas yra susijęs su tam tikrais įvykiais, kurie gali iškviešti „JavaScript“ funkcijas. Įvykių pavyzdžiai: spragtelėjimas pele, tinklalapio arba paveikslo įkėlimas, pelės vedimas ant nuorodos tinklalapyje,

duomenų įvedimo lauko pasirinkimas HTML formoje, HTML formos siuntimas arba klavišo paspaudimas.

## 2.2.4 „JavaScript“ objektų hierarchija

Daugelis „JavaScript“ objektų yra vienas kitame. „JavaScript“ objektai yra susiję sudėtinio ir atskiro rodinio, o ne klasės ir poklasio santykiais. Vieno tipo objekto savybės nėra perduodamos kito tipo objektui. „JavaScript“ objektai yra dviejų pagrindinių tipų.

- Kalbos objektai – jie numatomi kalboje ir nepriklauso nuo kitų objektų.
- Žvalgiklis – kliento naršyklėje numatyti objektai. Visi šie objektai yra pavaldūs žvalgiklio objektui.



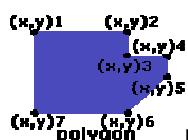
Navigator	Žvalgiklis
Mime Type	MIME tipas
Window	Langas
Plugin	Papildinys
Frame	Rėmelis
History	Praeitis
Location	Vieta
Document	Dokumentas
Anchor	Prieraišas
Applet	Programėlė
Area	Sritis
Form	Forma
Image	Paveikslas
Link	Nuoroda
Layer	Sluoksnis
Button	Mygtukas
CheckBox	Žymimasis langelis
FileUpload	Failo įkėlimas
Hidden	Paslėptasis
Option	Parinktis
Password	Slaptažodis
Radio	Parinkties akutė
Reset	Atstatyti
Select	Pažymėti
Submit	Išsiųsti
Text	Tekstas
TextArea	Teksto sritis
Language	Kalba
Array	Masyvas

Boolean	Loginė operacija
Date	Data
Math	Matematinė operacija
Number	Skaičius
String	Eilutė

Pav. „JavaScript“ objektų hierarchija (<http://www.w3schools.com/default.asp> – anglų kalba).

## 2.2.5 „JavaScript“ paveikslo plano pavyzdys

Paveikslo planas – tai su konkrečiu paveikslu susijęs koordinacių sąrašas, sukurtas atskiroms paveikslo sritims susieti su įvairiomis paskirties vietomis (kitaip nei paprastas paveikslo nuoroda, kai visas paveikslo plotas yra susietas su viena paskirties vieta). Pavyzdžiui, pasaulio žemėlapyje kiekviena šalis gali turėti hipersaitą į papildomą informaciją apie tą šalį. Paveikslo planas leidžia nesunkiai susieti įvairias paveikslo dalis paties paveikslo nedalijant į atskirus paveikslo failus. Daugiau informacijos pateikta 1 paskaitos dalyje „Paveikslų planai“, skirtoje HTML žymai `<map>` ir atitinkamiems parametrams.



Daugiakampio `<area shape="poly">` paveikslo žemėlapyje HTML sintaksė (argumentai paaiškinti 1 paskaitoje):

```
<map name="poly">
<area shape="poly" coords="x1, y1, x2, y2, x3, y3, x4, y4, x5, y5, x6, y6, x7,
y7" href="page.html" alt="Alternate Text Here">
</map>
```

Šį pavyzdį galima pagerinti naudojant „JavaScript“. Paveikslo žemėlapyje žymose `<area>` galima įterpti įvykius, kurie iškvies „JavaScript“. Žyma `<area>` palaiko įvykius `onClick`, `ondblclick`, `onmousedown`, `onmouseup`, `onmouseover`, `onmousemove`, `onmouseout`, `onkeypress`, `onkeydown`, `onkeyup`, `onfocus` ir `onblur`.

## Tas pats pavyzdys, papildytas „JavaScript“:

```
<html>
<head>
  <script type="text/javascript">
    function writeText(txt)
    {
      document.getElementById("desc").innerHTML=txt;
    }
  </script>
</head>

<body>



<map id ="SamplePolygon" name="SamplePolygon">
<area shape ="poly" coords =" x1, y1, x2, y2, x3, y3, x4, y4, x5, y5, x6, y6,
x7, y7" onMouseOver="writeText('Sample of clickable polygon.')" href
="Code_Sample.htm" target ="_blank" alt="Code_Sample" />
<p id="desc"></p>

</body>
</html>
```

Daug „JavaScript“ pavyzdžių pateikiama internetinėje mokomojoje priemonėje ([http://www.w3schools.com/js/js\\_examples.asp](http://www.w3schools.com/js/js_examples.asp) – anglų kalba). Studijuojantieji gali apsilankyti šioje svetainėje ir išsamiau susipažinti su „JavaScript“ kalba.

## 2.3. XML kodavimo pagrindai

### 2.3.1 XML apibrėžimas

**XML (*Extensible Markup Language*)** – tai plačios paskirties ženklinimo kalba. Ji įvardijama kaip išplečiamoji kalba, kadangi naudotojai gali patys apibrėžti žymas. Pagrindinė jos paskirtis – palengvinti struktūrinių duomenų dalijimąsi skirtingose informacinėse sistemose, visų pirma internetu. Ji naudojama ir dokumentams *koduoti*, ir duomenims *nuoseklinti*. XML rekomenduoja Pasaulinis interneto konsorciumas (W3C). Tai nemokamas atvirasis standartas. W3C rekomendacijoje nurodyta XML leksinė gramatika ir analizės reikalavimai. Kalba iš pradžių buvo kuriama kaip SGML (*Standard Generalized Markup Language*) porūšis siekiant, kad ji būtų gerai suprantama žmogui. Įterpus semantinius apribojimus, XML galima realizuoti taikomųjų programų kalbas, kurioms priskiriama XHTML (*XML pagrįsta HTML*), GML (*Geographic Markup Language*), ArcXML (*ESRI Arc Extensible Markup Language*), KLM (*Google Keyhole Markup Language*) ir daugelis kitų. Kartais XML vartojama kaip šių taikomųjų programų kalbų specifikacijų kalba. ISO ir OGC standartų organizacijos taip pat rekomenduoja XML kalbą geografinės informacijos schemoms ir duomenims koduoti.

XML pranašumai:

- Tai tekstinė kalba. Jai naudojami grynojo teksto failai, numatantys mažiau apribojimų nei kiti nuosavybiniai dokumentų formatai.
- XML palaiko Unikodą, todėl galima perduoti beveik visą informaciją visomis rašytinėmis žmogaus kalbomis.
- Kalba gali pateikti bendrąsias kompiuterijos mokslo duomenų struktūras: įrašus, sąrašus ir medžius. Hierarchinė struktūra tinkama daugumai (nors ir ne visiems) dokumentų tipams.
- Kadangi formatas yra aprašomasis, kalba apibūdina struktūrą ir laukų vardus bei konkrečias reikšmes.
- Dėl griežtų sintaksės ir analizės reikalavimų būtini analizės algoritmai yra ypač paprasti, veiksmingi ir pastovūs.
- XML yra labai plačiai naudojamas dokumentų saugojimo ir tvarkymo prijungties ir atjungties būsenos formatais.
- Kalba grindžiama tarptautiniais standartais.
- Galimi jos prieauginiai (incrementally) atnaujinimai.
- Patikrą galima atlikti naudojant tokias schemų kalbas, kaip XSD ir DTD, palengvinančias vienetų testavimą, užkardų naudojimą, funkcinių testavimą, sutartinių reikalavimų nustatymą ir programinės įrangos kūrimą. Gana paprasta išlaikyti perspektyvinį ir atgalinį suderinamumą nepaisant DTD arba schemas pasikeitimų.
- Kalba nepriklauso nuo kompiuterinės platformos, todėl yra iš dalies nepriklausoma nuo technologinių pokyčių.
- XML pirmtakė SGML buvo naudojama nuo 1986 m., tad yra sukaupta daug patirties, sukurta nemažai programinės įrangos.
- Struktūriškai tvarkingo XML dokumento elemento fragmentas taip pat yra ir struktūriškai tvarkingas XML elementas.

XML trūkumai:



- XML sintaksė yra perteklinė arba didelė, palyginti su dvejetainiu panašių duomenų pateikimu. Dėl to gali nukentėti kalbos vartojimo veiksmingumas, kadangi išauga duomenų saugojimo, perdavimo ir apdorojimo sąnaudos.
- XML sintaksė yra daugiažodė, ypač žmonėms, lyginant su kitais alternatyviais tekstiniais duomenų perdavimo formatais.
- Hierarchinis pateikimo modelis yra ribotas, palyginti su objektiniu grafu. Papildomų pastangų reikalauja persiklojančių (ne hierarchinių) viršūnių santykio išraiška.
- Sudėtinga naudoti XML vardų erdves, o jų palaikymą gali būti sunku teisingai realizuoti XML analizatoriuje.
- XML dažnai vaizduojama kaip „save dokumentuojanti“ kalba, tačiau taip neatsižvelgiama į esminius dviprasmiškumus.
- Kai kurie mano, kad turinio ir požymių skirtumai XML nėra natūralūs, todėl sunkiau kurti XML duomenų struktūras.
- XML dokumentams susieti reikia naudoti „XLink“ nuorodas, o tai yra sudėtinga lyginant su hipersaitais.
- Sunku rasti XML analizatorių, kuris būtų išbaigtas, teisingas ir veiksmingas.

XML palaiko tiesioginį beveik visų Unikodo ženklų (išskyrus turinčius ypatingą simbolinę prasmę XML kalboje, pavyzdžiui, kampinį pradžios skliaustą „<“) naudojimą elementų varduose, požymiuose, komentaruose, simboliniuose duomenyse ir apdorojimo komandose. Todėl toliau pateiktas XML dokumentas yra struktūriškai teisingas, nors jame yra ir kinų hieroglifų, ir kirilicos abėcėlės ženklų:

```
<?xml version="1.0" encoding="UTF-8"?>
<俄語>Данные</俄語>
```

XML dokumentuose nėra informacijos apie tai, kaip rodyti duomenis.

### 2.3.2 XML rašymas

XML dokumento teisingumas apibrėžiamas dviem aspektais:

- **Struktūriškai tvarkingas** XML dokumentas atitinka visas XML sintaksės taisykles. Pavyzdžiui, jeigu pradžios žyma neturi atitinkamos pabaigos žymos, dokumentas nėra struktūriškai tvarkingas. Dokumentas, kuris nėra struktūriškai tvarkingas, nelaikomas XML; atitinkamam analizatoriui neleidžiama tokio dokumento analizuoti.
- **Teisingas** XML dokumentas papildomai atitinka kai kurias semantines taisykles. Jas nustato naudotojas arba jos numatytos XML schemeje ar DTD. Pavyzdžiui, jeigu dokumente yra neapibrėžtas elementas, tada dokumentas nėra *teisingas*, o patikros analizatoriui neleidžiama tokio dokumento analizuoti.

### 2.3.3 XML sintaksės taisyklės

Jeigu reikalingas tik struktūrinis tvarkingumas, XML yra bendra struktūra, tinkama bet kokiam teksto arba duomenų, kurių struktūrą galima pavaizduoti kaip medį, kiekiui saugoti. Vienintelis privalomas sintaksės reikalavimas – dokumentas turi turėti tik vieną *šakninį elementą* (dar vadinamą *dokumento elementu*). Tai reiškia, kad dokumentas turi būti pateiktas tarp šaknies pradžios žymos ir atitinkamos pabaigos žymos. Pateikiamas „struktūriškai tvarkingas“ XML dokumentas:

```
<book>This is a book.... </book>
```

Prieš šakninį elementą gali būti neprivalomas XML aprašas. Šiame elemente nurodoma, kuri XML versija yra naudojama (paprastai 1.0); taip pat gali būti pateikiama informacija apie simbolių koduotę ir išorinę priklausomybę.

```
<?xml version="1.0" encoding="UTF-8"?>
```

W3C specifikacijoje numatytas *reikalavimas*, kad XML rengyklės palaikytų Unikodo ženklų koduotes UTF-8 ir UTF-16 (UTF-32 nėra privaloma). Labiau ribotų koduočių, pavyzdžiui, pagrįstų ISO/IEC 8859, naudojimas yra pripažintas, paplitęs ir palaikomas.

Komentarai gali būti rašomi bet kurioje medžio vietoje, net ir tekste, jeigu elemento turinį sudaro tekstas arba #PCDATA (#PCDATA yra mišraus turinio elemento aprašas: simboliniai duomenys arba simboliniai duomenys kartu su kitais elementais).

XML komentarai pradedami <!-- ir baigiami -->. Komentaro tekste negali būti dviejų brūkšnių (--).

```
<!-- This is a comment. -->
```

Daugumoje taikomųjų programų papildomas ženklavimas naudojamas norint XML dokumento turiniui suteikti struktūrą. Tarp šaknies žymų esančiame tekste gali būti pasirenkamas XML elementų skaičius. Pagrindinė vieno elemento sintaksė yra tokia:

```
<name attribute="value">content</name>
```

Čia abu `name` yra atitinkamai vadinami pradžios ir pabaigos žymomis. Pavyzdyje `content` yra koks nors tekstas, kuriame taip pat gali būti XML elementų. Todėl paprastame XML dokumente yra medžio pavidalo duomenų struktūra. Toliau pateikiamas struktūrinio XML dokumento pavyzdys:

```
<recipe name="bread" prep_time="5 mins" cook_time="3 hours">
  <title>Basic bread</title>
  <ingredient amount="3" unit="cups">Flour</ingredient>
  <ingredient amount="0.25" unit="ounce">Yeast</ingredient>
  <ingredient amount="1.5" unit="cups" state="warm">Water</ingredient>
  <ingredient amount="1" unit="teaspoon">Salt</ingredient>
  <instructions>
    <step>Mix all ingredients together.</step>
    <step>Knead thoroughly.</step>
    <step>Cover with a cloth, and leave for one hour in warm room.</step>
    <step>Knead again.</step>
    <step>Place in a bread baking tin.</step>
  </instructions>
</recipe>
```

```

<step>Cover with a cloth, and leave for one hour in warm room.</step>
<step>Bake in the oven at 350°F for 30 minutes.</step>
</instructions>
</recipe>

```

Požymių reikšmės visuomet būtina nurodyti su viengubomis arba dvigubomis kabutėmis, o kiekvieno požymio vardas elemente gali būti minimas tik vieną kartą.

Reikalaujama, kad elementai XML dokumente būtų įdedami teisingai (jie negali persikloti). Pavyzdžiui, toliau pateiktame pavyzdyje yra struktūriškai netvarkingas XML kodas, kadangi persikloja elementai *em* ir *strong*:

```

<!-- WRONG! NOT WELL-FORMED XML! -->
<p>Normal <em>emphasized <strong>strong emphasized</em> strong</strong></p>

<!-- Correct: Well-formed XML. -->
<p>Normal <em>emphasized <strong>strong emphasized</strong></em>
<strong>strong</strong></p>

<p>Alternatively <em>emphasized</em> <strong><em>strong emphasized</em>
strong</strong></p>

```

XML numatyta speciali sintaksė, vaizduoti elementą su tuščiu turiniu. Vietoje parašytos pradžios žymos, paskui kurią iškart seka pabaigos žyma, dokumente gali būti tuščio elemento žyma. Tuščio elemento žyma yra panaši į pradžios žymą, tačiau prieš pat kampinį pabaigos skliaustą rašomas pasvirasis brūkšnys. Toliau pateikti trys pavyzdžiai yra lygiaverčiai XML kode:

```

<foo></foo>
<foo />
<foo/>

```

Tuščias elementas gali turėti požymių:

```

<info author="John" genre="science-fiction" date="2009-Jan-01" />

```

**Esybių nuorodos.** XML dokumente *esybė* – tai nurodytas duomenų, dažniausia teksto, vienetas. Esybės dažnai naudojamos atskiriems simboliams, kuriuos sunku įvesti klaviatūra, nurodyti. Esybės taip pat naudojamos vaizduojant standartinio (stereotipinio) teksto dalis, esančias daugelyje dokumentų, ypač jeigu reikia numatyti, kad toks tekstas gali būti keičiamas tik vienoje vietoje.

Specialius simbolius galima vaizduoti naudojant esybių nuorodas arba skaitmenines nuorodas. „&#x20AC;“ yra skaitmeninės nuorodos pavyzdys. Naudojant Unikodo šešioliiktinį kodą čia nurodytas euro simbolis.

Esybės nuoroda yra konkrečią esybę atitinkantis vietaženklis. Jį sudaro esybės vardas, prieš kurį yra įrašytas ampersendas („&“), opo kurio eina kabliataškis („;“). XML turi penkias iš anksto nustatytas esybes:

&amp;	&	ampersendas
&lt;	<	mažiau nei
&gt;	>	daugiau nei
&apos;	'	apostrofas
&quot;	"	kabutė

Toliau pateiktame pavyzdyje pavadinimas „AT&T“ pavaizduotas naudojant iš anksto nustatytą XML esybę:

```
<company_name>AT&T</company_name>
```

Papildomas esybes (be iš anksto nustatytųjų) galima rašyti dokumento tipo apibrėžties (**DTD**) arba XML schemas (**XSD**) dokumentuose. Toliau pateikiamas pavyzdys, kaip tai daroma minimalioje vidinėje DTD. Aprašytos esybės gali apibūdinti atskirus simbolius arba teksto dalis, taip pat gali būti nuorodomis viena į kitą.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE example
[
  <!ENTITY copy "&#xA9;">
  <!ENTITY copyright-notice "Copyright &copy; 2006, XYZ Enterprises">
]>
<example>
  &copyright-notice;
</example>
```

Žiūrint tinkamoje naršyklėje, pirmiau pateiktas XML dokumentas atrodo taip:

```
<example> Copyright © 2006, XYZ Enterprises </example>
```

**Skaitmeninės nuorodos.** Skaitmeninės nuorodos atrodo kaip esybių nuorodos, tik vietoje vardo jas sudaro ženklas # ir skaičius po jo. Skaičius (dešimtainis arba šešioliktainis su priešdėliu „x“) atitinka Unikodo poziciją. Kitaip nei esybių nuorodos, jos nėra iš anksto aprašytos ir jų nereikia aprašyti DTD arba XSD dokumentuose. Paprastai jos naudojamos simboliams, kuriuos sunku koduoti, pavyzdžiui, arabiškiems simboliams dokumente, kuris parengtas europiniu kompiuteriu. „AT&T“ pavyzdyje taip pat galima atsisakyti ampersendo (dešimtainis skaičius 38 ir šešioliktainis skaičius 26 atitinka ženklo „&“ Unikodo poziciją):

```
<company_name>AT&#38;T</company_name>
<company_name>AT&#x26;T</company_name>
```

**Struktūriškai tvarkingi dokumentai.** Struktūriškai tvarkingam dokumentui, be kitų, taikomos šios taisyklės:

- Ne tuščius elementus apgaubia pradžios ir pabaigos žymos.
- Tuščius elementus galima žymėti tuščio elemento žyma (kuriai nereikia pabaigos žymos), pavyzdžiui, `<IAmEmpty />`. Tą patį reiškia ir `<IAmEmpty></IAmEmpty>`.
- Visos požymių reikšmės įrašomos su viengubomis (') arba dvigubomis (") kabutėmis. Vienguboms kabutėms naudojamas viengubų, dviguboms – dvigubų kabučių ženklas.
- Žymos gali būti įdėtinės, bet negali persikloti. Kiekvienas ne šakninis elementas turi būti visiškai apgaubtas kito elemento.
- Dokumentas turi atitikti jo aprašytą simbolių kodavimą. Kodavimas gali būti aprašytas arba numatytas išoriškai, pavyzdžiui, antraštėse "Content-Type", jeigu dokumentui perduoti naudojamas HTTP, arba pačiame dokumente, aiškiai ženklinant pačioje dokumento pradžioje. Jeigu tokio aprašo nėra, daroma prielaida, kad naudojama Unikodo koduotė, kurią nustato Unikodo baitų eiliškumo ženklas, esantis prieš pirmąjį dokumento simbolį. Jeigu tokio ženklo nėra, daroma prielaida, kad naudojama UTF-8 koduotė.

Elementų varduose būtina naudoti tą patį raidžių registrą. Pavyzdžiui, tokia pora yra struktūriškai tvarkinga:

```
<Step> ... </Step>
```

O ši – ne:

```
<Step> ... </step>
```

Kruopščiai parinkti XML elementų vardai perteikia ženklinimo duomenų prasmę. Todėl kodą lengviau perskaityti žmonėms, nors jis yra gana tikslus programinės įrangos analizei. Kai pasirenkami prasmingi vardai, skaitytojas (žmogus) supranta elementų ir požymių semantiką nesinaudodamas išoriniais dokumentais. Tačiau tai skatina daugiažodiškumą, dėl kurio programų kūrimas sudėtingėja, o failo dydis auga.

**Automatinis tikrinimas.** Patikrinti, ar XML dokumentas yra struktūriškai tvarkingas arba teisingas, yra gana paprasta, kadangi XML struktūrinio tvarkingumo ir patikros taisyklės numato priemonių perkeliamumą. Bet kuri priemonė, skirta darbui su XML failais, turėtų tikti visiems XML failams, parašytiems bet kuria XML kalba (arba XML programoms). Toliau pateikiamas savarankiškos priemonės naudojimo pavyzdys:

- Įkelkite XML dokumentą į XML palaikančią naršyklę, pavyzdžiui, „Firefox“ arba „Internet Explorer“
- Pasirinkite priemonę, pavyzdžiui, `xmlwf` (paprastai pateikiama su „Expat“, srautinio (angl. *stream-oriented*) XML 1.0 analizatoriaus biblioteka, parašyta C kalba)
- Pradėkite dokumento analizę, pavyzdžiui, „Ruby“ kalba:

```
irb> require "rexml/document"
irb> include REXML
irb> doc = Document.new(File.new("test.xml")).root
```

**XML semantika ir patikra.** Kadangi XML kalboje galima laisvai pasirinkti vardus, leidžiamą hierarchiją ir elementų bei požymių prasmę naudojant individualiai pritaikytą schemą arba DTD, XML sintaksė yra specialios paskirties XML tipo ženklinimo kalbų

kūrimo pagrindas. Šių kalbų bendroji sintaksė yra griežta – dokumentai turi atitikti bendrąsias XML taisykles ir užtikrinti, kad visa XML palaikanti programinė įranga gali bent jau perskaityti ir suprasti dokumentuose pateiktos informacijos struktūrą.

Kalbos schema, kurioje numatyti tam tikri apribojimai, tiesiog papildo sintaksės taisykles. Schemos paprastai numato elementų ir požymių vardų bei jų leidžiamų turinio hierarchijų apribojimus, pavyzdžiui, elemente 'birthday' gali būti tik vienas elementas 'month' ir vienas elementas 'day', o kiekviename jų gali būti tik simboliniai duomenys. Schemos apribojimai taip pat gali numatyti duomenų tipo priskyrimus, lemiančius, kaip bus apdorojama informacija; pavyzdžiui, elemento 'month' simboliniai duomenys pagal tam tikros schemų kalbos taisykles gali būti apibrėžiami kaip mėnuo. Tai gali reikšti, kad duomenys ne tik turi būti tam tikro formato, bet ir negali būti apdorojami kaip kito tipo duomenys.

Dokumentas, kuris atitinka tam tikros schemos arba DTD taisykles ir yra struktūriškai tvarkingas, laikomas *teisingu*.

XML schema – tai XML dokumento tipo apibrėžimas, kuriame paprastai nurodomi tokio tipo dokumento struktūros ir turinio apribojimai, kurie yra griežtesni nei pagrindiniai pačios XML kalbos apribojimai. Buvo sukurta daug XML schemų kalbų, skirtų formaliai tokių schemų išraiškai, o kai kurios šių kalbų buvo sukurtos remiantis XML.

Kol nebuvo apibendrintų duomenų aprašymo kalbų (pvz., SGML arba XML), programinės įrangos kūrėjai turėjo sukurti specialius failų formatus arba nedideles kalbas, kurios leistų programoms keistis duomenimis. Tam reikėjo rašyti detalias specifikacijas, specialios paskirties analizatorių ir tekstų rašymo programų (angl. *writers*).

Dėl taisyklingos XML struktūros ir griežtų analizės taisyklių programinės įrangos kūrėjai analizei gali naudoti standartines priemones. Kadangi XML numato bendrą duomenų modelių struktūrą taikomosioms programoms, kurios skirtos kalboms kurti, programinės įrangos kūrėjai gali visą dėmesį sutelkti į duomenims taikomų taisyklių kūrimą gana aukštu abstrakcijos lygmeniu.

Yra gerai išbandytų priemonių, atliekančių XML dokumento patikrą pagal schemą. Priemonė automatiškai patikrina, ar dokumentas atitinka schemeje nustatytus apribojimus. Kai kurios patikros priemonės yra įtrauktos į XML analizatorius, o kitų paketai pateikiami atskirai.

Schemos naudojamos ir kitoms paskirtims. Pavyzdžiui, XML tekstų rengyklės gali naudoti schemas redagavimui (pasiūlomi teisingi elementų ir požymių vardai ir pan.).

### 2.3.4 XML schema (DTD ir XSD)

Seniausias XML schemas formatas yra dokumento tipo apibrėžtis (*Document Type Definition*, DTD). Nors DTD palaiko beveik visos programos, nes DTD yra įtraukta į XML 1.0 standartą, ji turi keletą trūkumų:

- DTD nepalaiko naujesnių XML ypatybių, visų pirma vardų erdvių.
- Jai trūksta išraiškos priemonių. Kai kurių formaliųjų XML dokumento aspektų neįmanoma išreikšti DTD.
- Schemai aprašyti DTD naudoja specialią ne XML sintaksę, perimtą iš SGML.



- DTD vis dar plačiai naudojama, kadangi ši kalba laikoma lengviausiai skaitoma ir rašoma.

**Dokumento tipo apibrėžtis (DTD)** yra viena iš kelių SGML ir XML schemų kalbų; šis terminas vartojamas ir dokumentui, sukurtam DTD kalba, apibrėžti. DTD daugiausia naudojama schemai išreikšti naudojant aprašus, kurie atitinka tam tikrą ženklavimo sintaksę ir aprašo SGML arba XML dokumentų klasę, arba *tipą*, ir kuriuose nurodomi tokių dokumentų struktūros apribojimai. DTD taip pat gali būti aprašomos konstrukcijos, kurios ne visuomet reikalingos dokumento struktūrai nustatyti, bet gali turėti įtakos kai kurių dokumentų interpretavimui.

DTD yra kilusi iš SGML ir XML specifikacijų. Jai atsiradus buvo sukurtos kitos specifikacijų kalbos, pavyzdžiui, „XML Schema“ (XSD) ir „RELAX NG“, kurios užtikrina papildomą funkcionalumą.

Kadangi DTD yra schemos išraiška, joje nurodoma SGML arba XML „taikymo“, pvz., išvestinių kalbų HTML arba XHTML, sintaksė.

DTD dokumentų klasės struktūra aprašoma elementų ir požymių sąrašų aprašuose. Elementų aprašuose nurodoma, kurie elementai leidžiami dokumente, kaip kiekviename elemente turi būti pateikiami aprašyti elementai ir simbolinių duomenų sekos, jeigu tai yra leidžiama. Požymių sąrašų aprašuose nurodoma, kurie požymiai gali būti priskirti kiekvienam aprašytam elementui, įskaitant kiekvienos požymio reikšmės tipą, o gal net ir tikslų teisingų reikšmių kompleksą.

DTD su XML dokumentu susieti naudojamas dokumento tipo aprašas – žyma, įterpiama netoli XML dokumento pradžios. Apraše nustatoma, kad dokumentas atitinka nurodytame DTD aprašytą dokumento tipą.

DTD aprašai skirstomi į vidinio poaibio ir išorinio poaibio aprašus. Vidinio poaibio aprašai įstatomi pačiame dokumente esančiame dokumento tipo apraše. Išorinio poaibio aprašams naudojamas atskiras tekstinis failas. Nuorodai į išorinio poaibio aprašą galima naudoti viešuosius identifikatorius ir (arba) sisteminius identifikatorius.

Pateikiamas dokumento tipo aprašo pavyzdys, kuriame naudojami ir viešieji, ir sisteminiai identifikatoriai:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Šiame pavyzdyje dokumento tipo aprašas apima vidinį poaibį, sudarytą iš vienos esybės aprašo:

```
<!DOCTYPE foo [ <!ENTITY greeting "hello"> ]>
<!DOCTYPE bar [ <!ENTITY greeting "hello"> ]>
```

Visi HTML 4.01 dokumentai turėtų atitikti vieną iš trijų SGML dokumento tipo apibrėžčių. Šių DTD viešieji identifikatoriai yra pastovūs:

- `-//W3C//DTD HTML 4.01//EN`
- `-//W3C//DTD HTML 4.01 Transitional//EN`
- `-//W3C//DTD HTML 4.01 Frameset//EN`

Šių DTD sisteminiai identifikatoriai, jeigu jie nurodyti dokumento tipo apraše, yra URI nuorodos. Sisteminiai identifikatoriai gali būti įvairūs, tačiau turėtų nurodyti konkrečius aprašus, esančius nustatomoje vietoje. SGML leidžia viešuosius identifikatorius susieti su sisteminiais identifikatoriais *kataloguose*, kurie gali būti prieinami URI vardų išsprendimo priemonėmis (angl. *resolver*), naudojamoms dokumento analizės programinės įrangos.

Dažnai klaidingai manoma, kad patikros neatliekantys (angl. *non-validating*) XML analizatoriai neturi skaityti DTD, nors iš tiesų vis tiek reikia patikrinti, ar DTD sintaksė, esybių ir numatytųjų požymių aprašai yra teisingi. Tačiau patikros neatliekantis analizatorius gali nuspręsti neskaityti *išorinių esybių*, įskaitant išorinį DTD poaibį. Jeigu XML dokumentas priklauso nuo aprašų, kurie yra tik išorinėse esybėse, dokumento XML apraše turi būti nurodyta `standalone="no"`.

Pateikiamas labai paprasto XML DTD, aprašančio asmenų sąrašą, pavyzdys:

```
<!ELEMENT people_list (person*)>
<!ELEMENT person (name, birthdate?, gender?, socialsecuritynumber?)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT birthdate (#PCDATA)>
<!ELEMENT gender (#PCDATA)>
<!ELEMENT socialsecuritynumber (#PCDATA)>
```

Analizuodami kiekvieną eilutę matome, kad:

- 1.) `people_list` yra teisingas elemento vardas, o tokiaime elemente yra bet koks asmens elementų skaičius. Žvaigždutė (\*) parodo, kad elemente `people_list` gali būti nulis arba daugiau asmens elementų.
- 2.) `person` yra teisingas elemento vardas, o tokiaime elemente yra vienas elementas `name`, paskui eina vienas elementas `birthdate` (neprivalomas), tada elementas



`gender` (taip pat neprivalomas) ir elementas `socialsecuritynumber` (taip pat neprivalomas). Klausukas (?) parodo, kad elementas nėra privalomas. Kadangi nuorodoje į pavardės elementą `name` nėra klausuko, asmens elemente *privalo* būti pavardės elementas.

- 3.) `Name` yra teisingas elemento vardas, o tokiam elemente yra „išanalizuoti simboliniai duomenys“ (#PCDATA).
- 4.) `Birthdate` yra teisingas elemento vardas, o tokiam elemente yra simboliniai duomenys.
- 5.) `Gender` yra teisingas elemento vardas, o tokiam elemente yra simboliniai duomenys.
- 6.) `Socialsecuritynumber` yra teisingas elemento vardas, o tokiam elemente yra simboliniai duomenys.

Toliau pateikiamas XML failo, kuris naudoja šią DTD ir atitinka jos taisykles, pavyzdys. Daroma prielaida, kad DTD galima atpažinti pagal atitinkamą URI nuorodą "example.dtd":

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE people_list SYSTEM "example.dtd">
```

```
<people_list>
  <person>
    <name>Fred Bloggs</name>
    <birthdate>27/11/2008</birthdate>
    <gender>Male</gender>
  </person>
</people_list>
```

XML palaikančioje naršyklėje (pvz., „Internet Explorer 5“ arba „Mozilla“) tai galima pavaizduoti šį DTD komponentą įdedant ir įrašant tekstiniame faile *example.dtd*, o XML failą įrašant kitaip pavadintame faile, tada atveriant XML failą naršyklėje. Abu failus reikia įrašyti į tą patį katalogą. Tačiau daugelis naršyklių netikrina, ar XML dokumentas atitinka DTD taisykles. Jos turi tik patikrinti, ar DTD sintaksė yra teisinga. Saugumo sumetimais naršyklės taip pat gali neskaityti išorinės DTD.

Nors daugelis XML priemonių palaiko DTD, nes ji yra įtraukta į XML 1.0 standartą, dėl pirmiau išvardytų priežasčių DTD laikoma ribota kalba. Yra naujesnių XML schemų kalbų, kurios yra galingesnės ir kurioms teikiama pirmenybė, pavyzdžiui:

- „XML Schema“ kalbai, taip pat vadinamai XSD (*XML Schema Definition*), buvo suteiktas W3C rekomendacijos statusas.
- „RELAX NG“, kuri taip pat yra DSDL dalis, yra ISO tarptautinis standartas.
- DSD (*Document Structure Description*) kalboje derinamos schemų išraiškos galimybės ir naudojimo paprastumas.

„**XML Schema**“ (XSD) yra naujesnė XML schemų kalba, kurią W3C apibūdina kaip DTD tąsą ir kuri neoficialiai vadinama XSD (*XML Schema Definition*). XSD turi daugiau galimybių nei DTD aprašyti XML kalbas. Kalboje naudojama turtinga duomenų tipų sistema, galima numatyti detalesnius XML dokumento loginės struktūros apibrėžimus, jai apdoroti reikalinga griežtesnė patikros struktūra. XSD taip pat naudoja XML grindžiamą

formatą, leidžiantį naudoti įprastas XML priemones, kurios padeda apdoroti XSD, nors XSD įgyvendinti reikia daugiau nei vien gebėjimo skaityti XML.

XSD buvo pirmosios W3C rekomenduojamos XML schemas, siūlančios vardų erdvių ir duomenų tipų palaikymo alternatyvą XML dokumento tipo apibrėžtims (DTD).

XSD kalba kritikuojama dėl šių aspektų:

- Jos specifikacija yra labai didelė, todėl ją sunku suprasti ir realizuoti.
- Dėl XML sintaksės atsiranda daugiažodiškumas schemų aprašuose, todėl XSD sunkiau skaityti ir rašyti.
- Schemų patikra gali būti brangus XML analizės papildymas, ypač didelės apimties sistemose.
- Modeliavimo galimybės yra labai ribotos, požymiams neleidžiama daryti įtakos turinio modeliams.
- Tipų išvedimo modelis yra labai ribotas; tipų išvedimas naudojant plėtinį retai pasirodo naudingas.

Ir visas XML schemų kalbas, ir „XML Schema“ galima naudoti schemai išreikšti. Vadovaujantis ta schema, XML dokumentas laikomas *teisingu*, jeigu jis atitinka joje nustatytas taisykles. XSD apibrėžia XML dokumento tipą pagal galimų elementų ir požymių apribojimus, jų tarpusavio santykius, galimus duomenų tipus ir t. t. Kartu su patikros programine įranga, „XML Schema“ galima naudoti norinti patikrinti, ar konkretus XML dokumentas yra reikiamo tipo, bei parengti PSVI (*Post-Schema Validation Infoset*) aprašą, kuriame kiekvienam elementui, požymiui ir visoms XML dokumento viršūnėms yra priskirtas tipas pagal XML schemą.

Tačiau kitaip nei daugelis kitų schemų kalbų, „XML Schema“ buvo kuriama siekiant, kad atlikus dokumento patikrą būtų surinkta informacija, atitinkanti konkrečius duomenų tipus. Tokia po patikros surinkta informacija gali būti naudinga kuriant XML dokumentų apdorojimo programinę įrangą, tačiau schemas kalbos priklausomybė nuo konkrečių duomenų tipų susilaukė kritikos.

„XML Schema“ pavyzdys yra XSD (XML schemas apibrėžtis), kuri rašoma „XML Schema“ kalba ir kurios failo vardo plėtinys paprastai yra `.xsd`. Kartais pati kalba neoficialiai vadinama XSD. Buvo pasiūlymų, kad WXS (W3C rekomenduojama „XML Schema“) yra tinkamesnis sutrumpinimas, nors šis akronimas nėra labai populiarus, o W3C darbo grupė jį atmetė. XSD taip pat vartojamas kaip „XML Schema Datatypes“, „XML Schema“ duomenų tipų dalies, santrumpa.

Yra nurodyta 19 primitiviųjų *duomenų tipų*. Buvo pasiūlyti keli sprendimai, kaip dvejetainius duomenis įstatyti į xml (pavyzdžiui, paveikslus). Dvejetainius duomenis galima laikyti neskaidrių (angl. *opaque*) duomenų tipu. Toks pavadinimas atsirado dėl to, kad duomenų modeliui nieko nėra žinoma apie vidinį tipo atvaizdavimą. Pateikiamas paprastas „XML Schema“ apibrėžties pavyzdys, kuriame apibūdinama šalis.

```

<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="country" type="Country"/>
  <xs:complexType name="Country">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="population" type="xs:decimal"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

Toliau pateikiamas XML dokumento, kuris atitinka šią schemą, pavyzdys (daroma prielaida, kad schemas failo vardas yra `country.xsd`):

```

<country
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="country.xsd">
  <name>France</name>
  <population>59.7</population>
</country>

```

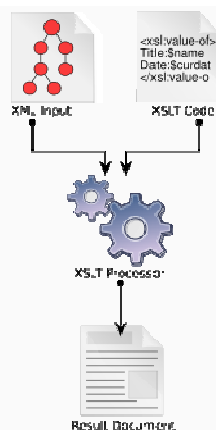
XSD dar neturi įregistruoto unikalaus interneto terpės tipo, todėl reikia naudoti "application/xml" arba "text/xml", kaip numatyta RFC 3023 (RFC (*Request for Comments*) – tai serija dokumentų, aprašančių naujus tyrimus, naujoves ir metodikas, taikomas interneto technologijoms; skaičius 3023 nurodo XML ir XSD standartus).

### 2.3.5 XSL ir XSLT

XML dokumentuose  *nėra* informacijos apie tai, kaip rodyti duomenis. Jeigu nenaudojama CSS arba XSL (*eXtensible Stylesheet Language* – transformavimo kalbų šeima, leidžianti aprašyti, kaip turi būti formatuojami arba transformuojami XML standartu užkoduoti failai), dauguma interneto naršyklių paprastą XML dokumentą pateikia kaip neapdorotą XML tekstą. Kai kurios naršyklės dokumentą pateikia su „rankenėlėmis“ (pvz., simboliais + ir – parašėje), kurios leidžia pelės spragtelėjimu išskleisti arba suskleisti struktūros dalis. Tam, kad naršyklėje XML dokumentas būtų vaizduojamas tam tikru stiliumi, XML dokumente turi būti nuoroda į stiliaus lapą.

**XSL (*eXtensible Stylesheet Language*)** – transformavimo kalbų šeima, leidžianti aprašyti, kaip turi būti formatuojami arba transformuojami XML standartu užkoduoti failai. XSL buvo sukurta taip, kad būtų valdoma duomenų, ir labai skatina naudoti valdymo inversijos projektavimo šabloną (angl. *control of inversion design pattern*). Šeimoje yra keletas kalbų, kurios tapo W3C rekomendacijomis. XSL šeimai priskiriamos šios kalbos:

- **XSLT (*XSL Transformations*)** – tai XML dokumentų transformavimo kalba. Šiuo metu yra daug XSLT realizacijų. Kelios interneto naršyklės, įskaitant „Internet Explorer“, „Firefox“, „Mozilla“, „Netscape“ ir „Opera“, palaiko XML transformavimą į HTML naudojant XSLT. Paveikslėlyje pateikiamas transformavimo pavyzdys.



Pav. Pagrindiniai XSLT elementai ir procesų srautai (šaltinis: [http://en.wikipedia.org/wiki/XSL\\_Transformations](http://en.wikipedia.org/wiki/XSL_Transformations) – anglų kalba)

XSLT kuria W3C konsorciumas. Nuo 2007 m. plačiau naudojama ir realizuojama versija yra XSLT 1.0, nors naujausia versija yra XSLT 2.0, kuriai W3C rekomendacijos statusas buvo suteiktas 2007 m. sausio 23 d. XSLT atsirado kaip dalis W3C konsorciumo 1998–1999 m. vykdyto XSL kūrimo projekto. Įgyvendinus šį projektą, taip pat buvo sukurta objektų formatavimo kalba „XSL Formatting Objects“ ir kalba „XML Path Language“ (XPath).

Kaip kalbai, XSLT turi įtakos funkcinės kalbos ir tekstinės šablonų sutapdinimo kalbos, pavyzdžiui, SNOBOL ir awk. Ši kalba buvo sukurta remiantis DSSSL, t. y. kalba, kuri SGML kalboje atliko tą pačią funkciją, kaip XSLT atlieka XML. XSLT taip pat galima laikyti šablonų rengykle.

- **XSL-FLO (XSL Formatting Objects)** – tai kalba, nustatanti XML dokumento vaizdinį formatavimą, kuri įgauna vis didesnę populiarumą.
- **XPath (XML Path Language)** – tai XSL šeimos kalba, kurią XSLT vartoja XML dokumento navigacijai.
- **XQuery (XML Query Language)** – tai dar vienas W3C projektas, kuriuo siekiama sukurti panašias XML dokumentų užklausos galimybes naudojant „XPath“.

XSL galima naudoti keičiant XML duomenų formatą į HTML formatą arba kitus „žmonėms suprantamus“ formatus, kuriuos gali atvaizduoti naršyklė. Pirminis dokumentas nėra keičiamas, nes, remiantis esamo dokumento turiniu, yra sukuriamas naujas. Naują dokumentą rengyklė gali nuoseklinti standartinė XML sintakse arba kitu formatu, pavyzdžiui, HTML arba grynojo teksto formatu.

XSLT dažniausia naudojama duomenų formatui keisti pagal skirtingas XML schemas arba XML duomenų formatui keisti į interneto naršyklėms skirtus HTML arba XHTML dokumentus, sukuriant dinaminį interneto puslapį, arba į tarpinį XML formatą, kurį galima paversti PDF dokumentais. Šiems formatams keisti XSLT rengyklė kaip orientyrą gali naudoti XSL stiliaus lapą.

XSLT apdorojimo modelis apima:

- vieną arba kelis XML pirminius dokumentus
- vieną arba kelis XSLT stiliaus lapų modulius
- XSLT šablono apdorojimo rengyklę
- vieną arba kelis rezultatų dokumentus

Paprastai XSLT rengyklei reikia dviejų pirminių dokumentų – XML pirminio dokumento ir XSLT stiliaus lapo, iš kurių ji parengia vieną rezultatų dokumentą. XSLT stiliaus lape, kuris pats yra XML dokumentas, yra XSLT programos tekstas (arba išėtinis kodas kitose kalbose). Jame apibūdinamos šablono taisyklės – komandos ir kiti nurodymai rengyklei, kaip parengti rezultatų dokumentą.

XSLT kalba yra deklaratyvioji. Šablono taisyklėse vietoje privalomos veiksmų sekos nustatoma, kaip apdoroti viršūnę, kuri atitinka konkretų „XPath“ tipo šabloną, jeigu rengyklė tokią suranda, o šablonų turinyje nurodyti funkciniai reiškiniai, kurie tiesiogiai atitinka savo įvertintą formą, t. y. rezultatų medį, kuris yra rengyklės pateikiamo rezultato pagrindas.

Rengyklė taiko nustatytą algoritmą. Darydama prielaidą, kad stiliaus lapas jau buvo perskaitytas ir parengtas, rengyklė suformuoja *pirminį medį* pagal pirminį XML dokumentą. Paskui rengyklė pradeda apdoroti pirminio medžio šaknies viršūnę, stiliaus lape surasdama šiai viršūnei tinkamiausią šabloną ir įvertindama šablono turinį. Kiekviename šablone esančios komandos paprastai nurodo rengyklei sukurti rezultatų medžio viršūnes arba daugiau pirminio medžio viršūnių apdoroti taip pat kaip ir šaknies viršūnę. Rezultatas parengiamas pagal rezultatų medį.

XSLT rengyklės realizavimas skirstomas į dvi pagrindines kategorijas: serverio pusės ir kliento pusės. Norint nurodyti kliento pusės XSLT, XML turi būti tokia duomenų apdorojimo komanda:

```
<?xml-stylesheet type="text/xsl" href="myTransform.xslt"?>
```

XSLT rengyklės gali būti pateikiamos kaip atskiri produktai arba priklausyti kitai programinei įrangai, įskaitant interneto naršykles, taikomųjų programų serverius, tokias struktūras kaip „Java“ ir „.NET“, netgi operacines sistemas. Pavyzdžiui, „Windows XP“ pakete yra MSXML3 biblioteka, kuri turi XSLT rengyklę. Dauguma senesnių XSLT rengyklių buvo interpretatoriai.

Daugelis interneto naršyklių su XSLT rengyklėmis palaiko kliento pusės XSLT. Norint nepriklausyti nuo galutinio naudotojo naršyklės gebėjimų, galima naudojant XSL keisti XML formatą į vaizduojamąjį formatą *serveryje*. Galutinis naudotojas nežino, kas įvyko „užkulisiuose“, ir mato tik struktūriškai tvarkingus rodomus duomenis.

Gaunamo XML dokumento pavyzdys:

```
<?xml version="1.0" ?>
<persons>
  <person username="JS1">
    <name>John</name>
    <family_name>Smith</family_name>
  </person>
  <person username="MI1">
    <name>Morka</name>
    <family_name>Ismincius</family_name>
  </person>
</persons>
```

XML formato keitimo į XML formatą su XSLT stiliaus lapu, kuriame yra XML dokumento transformavimo šablonai, pavyzdys:

```
<?xml version="1.0" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="xml" indent="yes"/>

<xsl:template match="/">
  <root> <xsl:apply-templates/> </root>
</xsl:template>

<xsl:template match="//person">
  <name username="{@username}">
    <xsl:value-of select="name" />
  </name>
</xsl:template>

</xsl:stylesheet>
```

Įvertinus sukuriamas naujas XML dokumentas, turintis kitą struktūrą:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <name username="JS1">John</name>
  <name username="MI1">Morka</name>
</root>
```

## XML transformavimo į XHTML su XSLT stiliaus lapų pavyzdys:

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:template match="/persons">
    <html xmlns="http://www.w3.org/1999/xhtml">
      <head> <title>Testing XML Example</title> </head>
      <body>
        <h1>Persons</h1>
        <ul>
          <xsl:apply-templates select="person">
            <xsl:sort select="family_name" />
          </xsl:apply-templates>
        </ul>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="person">
    <li>
      <xsl:value-of select="family_name"/>,
      <xsl:value-of select="name"/>
    </li>
  </xsl:template>

</xsl:stylesheet>

```

XSLT vartoja W3C kalbą „**XPath**“ pirminio dokumento medžio poaibiams nustatyti ir skaičiavimams atlikti. „XPath“ taip pat užtikrina įvairias funkcijas, kurias toliau papildo pati XSLT. Naudojama kartu su „XPath“, XSLT kalba tampa veiksmingesnė ir lankstesnė.

XSLT 2.0 versija veikia su „XPath“ 2.0 versija, o abi specifikacijos buvo paskelbtos tą pačią dieną. Analogiškai XSLT 1.0 versija veikia su „XPath“ 1.0 versija.

„XPath“ – tai išraiškos kalba, skirta XML dokumentų dalims arba apskaičiuotoms reikšmėms (pavyzdžiui, eilutėms, skaičiams arba loginėms reikšmėms), atsižvelgiant į XML dokumento turinį. „XPath“ kalba grindžiama XML dokumento medžio pateikimu. Ji užtikrina navigaciją aplink medį. Viršūnės pasirenkamos pagal keletą kriterijų. Bendriausias „XPath“ reiškiny (po kurio nurodoma „XPath“) yra kelio reiškiny. Kodas rašomas kaip seka žingsnių, norint nuo vienos XML viršūnės (dabartinio konteksto viršūnės) pereiti prie kitos viršūnės arba viršūnių. Žingsniai atskiriami „/“ (t. y. kelio) simboliais. Kiekvieną žingsnį sudaro trys komponentai:

- Ašies ženklas
- Viršūnės testas
- Predikatas

Pavyzdžiui, jeigu XML dokumente yra bent jau:

```
<A><B><C></C></B></A>
```

Paprasčiausias „XPath“ yra tokios formos:

/A/B/C

Atrenkami C elementai, kurie yra B elementų, A elemento vaikų, vaikai. A elementas yra daugiausia nutolęs XML dokumento elementas.



## 2.4. Išvados

Šioje paskaitoje buvo nagrinėjamos skriptų kalbos ir kelios susijusios technologijos, įskaitant DHTML, HTML dokumento objektų modelius (DOM), „JavaScript“, XML, XSL ir XSLT. Su jomis buvo susipažįstama kaip su tinklalapių kūrimo, priežiūros ir keitimo priemonėmis. Šios temos grindžiamos ankstesnių paskaitų medžiaga, ypač temomis, skirtomis HTML, paprastoms interneto kūrimo ir GIS priemonėms. Ši paskaita papildo ankstesnių paskaitų temas, kadangi interneto kūrimas yra vienas svarbiausių informacijos tvarkymo aspektų ir esminis informacijos infrastruktūros tarptautiniu, nacionaliniu, vietiniu bei individualiu lygmeniu aspektas.

Šioje paskaitoje išdėstytoms temoms visiškai įsisavinti būtina atkreipti dėmesį į keletą pagrindinių dalykų:

- Skriptų kalbos nuolat keičiasi, o daugiausia dėmesio skiriama jų tikslumui, naudojimo paprastumui, galimybėms taikyti kliento ir serverio programoms, lankstumui ir rafinuotumui.
- Šie metodai susiję ir su programinės įrangos kūrimu, ir standartais. Standartai tai pat yra svarbūs dėl saugumo ir vartotojų poreikių, todėl susipažįstant su programinės įrangos kūrimu, būtina akcentuoti nustatytų taisyklių svarbą.
- Nors skriptų kalbos gali pasirodyti gana sudėtingos, jos grindžiamos paprasta tinklalapių kūrimo logika, kuri užtikrina tam tikrą lankstumą ir leidžia pasireikšti naudotojo kūrybiškumui.
- Nė viena skriptų kalba nėra tobula, tačiau, kaip matyti iš šios paskaitos, kalbų pranašumai ir trūkumai yra įvairūs, todėl, norint šias kalbas suprasti, jas reikia vertinti kritiškai.
- „JavaScript“ yra viena svarbiausių skriptų kalbų, galinti veikti ir didesnėje aplinkoje, reikalaujančioje mokėti kitų su žiniatinkliu susijusių metodų bei kalbų, pavyzdžiui, HTML ir DHTML.
- XML kalba yra taip pat svarbi, nes yra grindžiama tekstiniu formatu, kuris numato mažiau apribojimų nei kiti dokumentų formatai, ji beveik visapusiškai išnaudoja „UNICODE“ ir turi naudingą hierarchinę struktūrą, paprastus ir pastovius algoritmus bei prieauginio atnaujinimo galimybes. Šios savybės yra svarbios neatsižvelgiant į tokius trūkumus, kaip daugiažodiškumas, sunkumai susiejant su dvejetainiais pateikimais bei hierarchinio modelio ribotumas pateikimo atžvilgiu. Todėl, kaip ir „JavaScript“ atveju, būtina šią kalbą vertinti kritiškai.

### **Paskaitos savarankiško mokymosi klausimai:**

1. Apibūdinkite, kuo DHTML skiriasi nuo HTML. Kaip manote, ar DHTML yra veiksmingesnė tinklalapių valdymo ir kūrimo priemonė nei HTML?
2. Apibūdinkite pagrindines Pasaulinio interneto konsorciumo (W3C) charakteristikas. Koks yra pagrindinis šios organizacijos vaidmuo kuriant internetą?
3. Apibūdinkite dokumento objektų modelio funkcijas. Kurios specifikacijos ir punktai įtraukti į 0, 1, 2 ir 3 lygmenis?
4. Kokios yra pagrindinės „JavaScript“ funkcijos programuojant žiniatinklį ir kuriant svetaines? Pateikdami pavyzdžių, apibūdinkite pagrindinius „JavaScript“ sakinių principus.
5. Kokie yra pagrindiniai XML pranašumai ir trūkumai? Apibūdinkite, kuo skiriasi XML schemas (DTD ir XSD).

### **Siūlomieji skaitiniai:**

- „JavaScript“ mokymosi programa: <http://www.w3schools.com/js/>
- XML mokymosi programa: <http://www.w3schools.com/xml/>
- XSL mokymosi programa: <http://www.w3schools.com/xsl/>

### Nuorodos (anglų kalba):

- W3Schools, <http://www.w3schools.com/default.asp>
- Flanagan, D., JavaScript: The Definitive Guide, 5th Edition, O'Reilly & Associates, 2006
- Duffy, S., How to do Everything with JavaScript, Osborne, 2003
- Vander Veer, Emily A., JavaScript For Dummies, 4th Edition, Wiley Pub, 2004
- Powell, T., Schneider, F., JavaScript: The Complete Reference. McGraw-Hill Companies, 2001
- XML on Open Directory Project, [http://www.dmoz.org/Computers/Data\\_Formats/Markup\\_Languages/XML/](http://www.dmoz.org/Computers/Data_Formats/Markup_Languages/XML/)
- van der Vlist, E., Van Comparing XML Schema Languages, 2001, <http://www.xml.com/pub/a/2001/12/12/schemacompare.html>
- XHTML2 Working Group Home Page, W3C, <http://www.w3.org/MarkUp/>
- Ousterhout, John K. Scripting: Higher Level Programming for the 21st Century, <http://home.pacbell.net/ouster/scripting.html>
- Prechelt, L., Are Scripting Languages Any Good? A Validation of Perl, Python, Rexx, and Tcl against C, C++, and Java, 2002, [http://page.mi.fu-berlin.de/prechelt/Biblio/jccpprt2\\_advances2003.pdf](http://page.mi.fu-berlin.de/prechelt/Biblio/jccpprt2_advances2003.pdf)
- The Extensible Stylesheet Language Family (XSL), W3C, <http://www.w3.org/Style/XSL/>
- XSL, The Cover Pages Web Site, <http://xml.coverpages.org/xsl.html>

### Vartojamieji terminai:

- Dinaminiai parametriniai tinklalapiai
- Kliento pusės skriptai
- DHTML
- Dokumento objektų modelis (DOM)
- Dokumento tipo aprašas (DTD)
- Kodavimas
- „JavaScript“
- „Jscript“
- Ženklimas
- Objektų hierarchija
- Objektinis programavimas
- „Perl“
- PHP
- „Python“
- Skriptų kalbos
- Serverio pusės skriptai
- Struktūriniai komponentai
- „Vbscript“
- W3C rekomendacija
- W3C standartai
- XHTML
- XML
- XSD
- XSL
- XSLT

### 3. Interneto žemėlapiai ir susijusios technologijos

Interneto žemėlapių technologija atvėrė naujus kelius, kaip pasauliniuose ir vietiniuose tinkluose skleisti, dalintis, atvaizduoti ir apdoroti erdvinę informaciją. Interneto technologija pagrįsti sprendimai suteikia galimybę nebrangiai ir efektyviai pateikti žemėlapius naudotojams. Šioje paskaitoje aptariamos interneto žemėlapių ir su kitos susijusios technologijos bei specifikacijos. Pirmiausia bus aptariama daugiapakopė architektūra, sudaranti interneto žemėlapių paslaugų aplinką. Taip pat apžvelgiama ir interneto technologija „AJAX“, kuri dažnai naudojama interneto žemėlapių taikomosiose programose kliento pakopoje. Toliau analizuojama tripakopė architektūra, naudojama interneto žemėlapių paslaugoms, taip pat pristatomos sistemos, veikiančios su „plonaisiais“ (Web Mapping Server - WMS) ir „storaisiais“ (Web Feature Server - WFS) interneto žemėlapių klientais. Toliau smulkiai aptariamos OGC standartų realizacijos specifikacijos, apibrėžiančios „Web Mapping Server“, „Web Feature Server“ ir „Web Coverage Server“ funkcinės galimybes. Toliau smulkiai aptariamos OGC standartų realizacijos specifikacijos, apibrėžiančios „Web Mapping Server“, „Web Feature Server“ ir „Web Coverage Server“ funkcinės galimybes. Šiomis specifikacijomis reikia naudotis kuriant suderintas paskirstytas interneto žemėlapių sistemas. Šios specifikacijos turi būti naudojamos gamintojų kuriant tarpusavyje suderinamas paskleistas internetinio atvaizdavimo sistemas. OGC interneto žemėlapių paslaugų specifikacija apibrėžia rinkinį funkcijų, kurias klientai gali naudoti bendraudami su WMS/WFS/FCS teikėjais (serveriais).

#### Paskaitos planas

- Daugiapakopė architektūra
- „AJAX“ technologija
- Interneto žemėlapių programinės įrangos architektūra
- Interneto žemėlapių paslaugos (Web Mapping Services – WMS)
- Interneto geoobjektų paslaugos (Web Feature Services – WFS)
- Interneto aprėpimo paslaugos
- Filtro kodavimas
- Sluoksnių stiliaus aprašymas ir simbolių reikšmių kodavimas
- Išvada

### 3.1. Daugiapakopė architektūra

Programinės įrangos architektūra, kuri apibrėžiama kaip strateginė programinės įrangos konstrukcija, pasižymi architektūriniais stiliais. Egzistuoja keletas programinės įrangos architektūrinių (strateginių) stilių: autonominis, kliento-serverio, lygiarangių įrenginių ir t.t. Apskritai, autonominė, vienpakopė taikomoji programa - tai programa, kuri veikia vien kliento kompiuteryje.

Kliento-serverio arba dvipakopė architektūra - tai tokia architektūra, kai klientas yra atskirtas nuo serverio ir ją galima realizuoti per kompiuterių tinklą. Paprasčiausio tipo kliento-serverio architektūroje naudojami tik dviejų tipų mazgai: klientai ir serveriai. Kliento-serverio taikomosios programos sudarytos iš kliento, kuris turi naudotojo sąsają ir serverio, kuris atlieka duomenų saugojimą, licenzijavimą, didelio kiekio duomenų apdorojimą ir kt.

Kiekviena konkreiti kliento programinė įranga gali siųsti užklausas į vieną arba daugiau sujungtų serverių. Serveriai gali priimti šias užklausas, apdoroti jas ir nusiųsti klientui jo prašytą informaciją. Šį principą galima pritaikyti daugeliui skirtingų taikomųjų programų. Kliento-serverio architektūros principai išlieka tokie patys. Dabar tipinis klientas – tai interneto naršyklė ir interneto, duomenų bazių bei elektroninio pašto serveriai.

Kai kurių tipų kliento-serverio architektūros konstrukcijos gali būti sudėtingesnės ir susidaryti iš eilės mazgų. Daugiapakopėje kliento-serverio architektūroje naudojamos trys (**tripakopė**) arba daugiau kaip trys pakopos (**daugiapakopė** arba *n-pakopė architektūra*), ir kiekviena iš jų gali naudotis keliomis taikomosiomis programomis arba paslaugomis. Daugiapakopę taikomąją programą vykdo daugiau negu vienas individualus programinės įrangos agentas, galintis būti ant kelių įrangos ir programinės įrangos platformų. Pavyzdžiui, į tai gali būti *klientai, taikomųjų programų serveriai*, kurie apdoroja duomenis klientams, ir *duomenų bazių serveriai*, kurie saugo duomenis taikomųjų programų serveriams. Plačiausiai naudojama tripakopė architektūra.

*N-pakopių* architektūrų privalumas tame, kad jos pasižymi žymiai didesnėmis *išplėtimo* galimybėmis, kadangi jos išlygina ir paskirsto darbo apkrovą tarp daugelio, dažnai perteklinių, specializuotų serverių mazgų. Todėl padidėja visos sistemos našumas ir patikimumas, kadangi tuo pačiu metu galima atlikti daugiau skaičiavimo darbų.

Programinės įrangos mazgai arba konkretūs **lygiaverčių įrenginių** (angl. **peer-to-peer (P2P)**) architektūros pavyzdžiai vienu metu gali veikti ir kaip klientas, ir kaip serveris, o atsakomybė ir statusas abiem atvejais vienodi. Šiuo metu plačiai naudojamos ir kliento-serverio ir P2P architektūros.

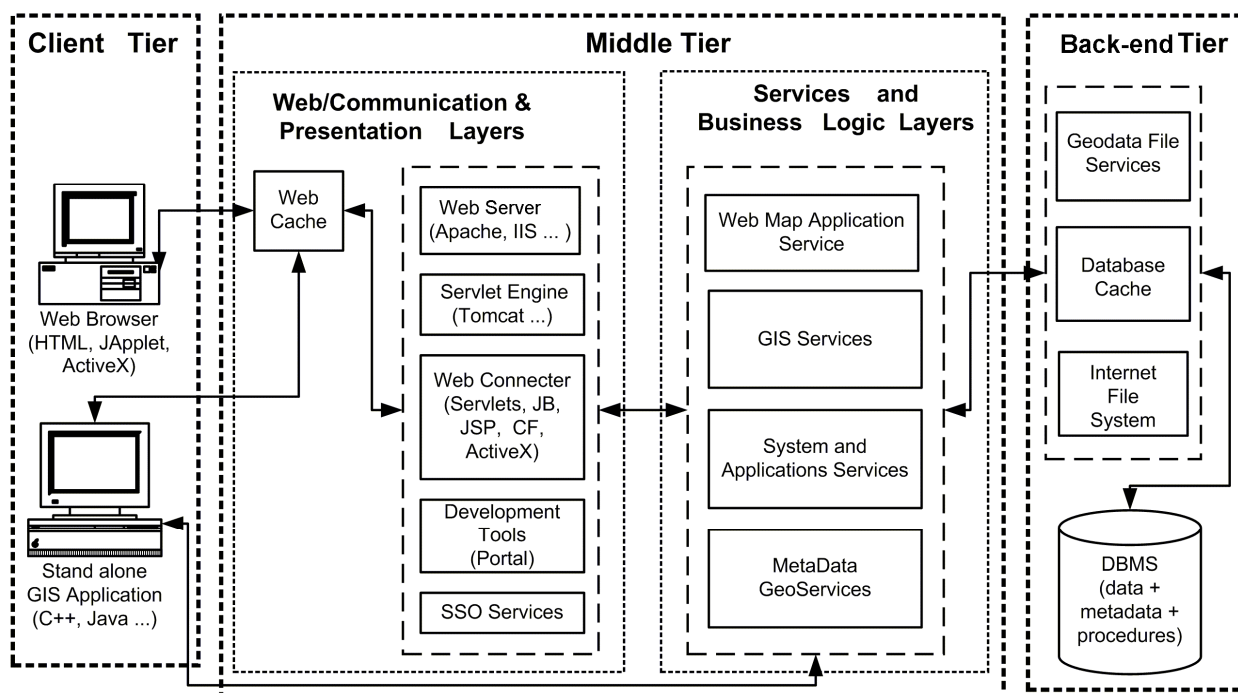
Viena iš pagrindinių paskatų vystyti GIS technologiją – tai noras pritaikyti IT technologiją, kuri tiesiogiai naudoja daugiapakopę programinės įrangos architektūrą operacinei sistemai organizuoti. Daugelis erdviųjų sprendimų prasidėjo 1990-ojo dešimtmečio pabaigoje perėjimu prie žiniatinkliu grindžiamos paskirstytosios ir atviros n-pakopės architektūros. Tačiau dauguma esamų interneto žemėlapių taikomųjų programų leidžia atlikti tik kartografinius vaizdavimus ir paprastą analizę, o dauguma skaičiavimo funkcijų vis dar yra kliento pakopoje. Dauguma dabartinių interneto žemėlapių serverių - tai supaprastintas variantas visas funkcines galimybes turinčių taikomųjų programų serverių, esančių tipakopės pramoninio standarto architektūros viduryje.

Žemėlapių programinės įrangos serveriai leidžia naudotojams ir taikomosioms programoms per internetą/intranetą su saugia prieiga nuotoliniu būdu surasti GIS duomenis naudojant paprastą „nurodyk ir spustelk“ sąsają. Taikomųjų programų serveris suteikia galimybę šiomis žemėlapių paslaugomis:

- Erdvinių duomenų suradimas ir vizualizacija
- Nuotolinis erdvinio duomenų apdorojimo užsakymas
- Dinaminis užduoties būsenos tikrinimas
- Erdvinė analizė
- Duomenų gavimo ir perdavimo proceso saugumo valdymas
- Duomenų iš heterogeninių šaltinių integravimas
- Senos programinės įrangos pritaikymas

Šias žemėlapių paslaugas galima išdėstyti skirtinguose n-pakopės architektūros sluoksniuose ir paskirstyti po tinklus. Šiuo metu naudojami du pagrindiniai taikomųjų programų kūrimo modeliai (API) – tai „Java 2 Enterprise Edition“ (J2EE, [http://en.wikipedia.org/wiki/Enterprise\\_application](http://en.wikipedia.org/wiki/Enterprise_application) - anglų kalba) ir .NET (<http://en.wikipedia.org/wiki/.NET> – anglų kalba). Abu šie modeliai naudojami kuriant n-pakopės žemėlapių taikomasias programas. Jie sukurti panašiais architektūriniais principais ir abu remiasi daugiapakopės architektūros koncepcija. Žemiau pateiktame paveiksle parodyta galima n-pakopės žemėlapių architektūros schema.

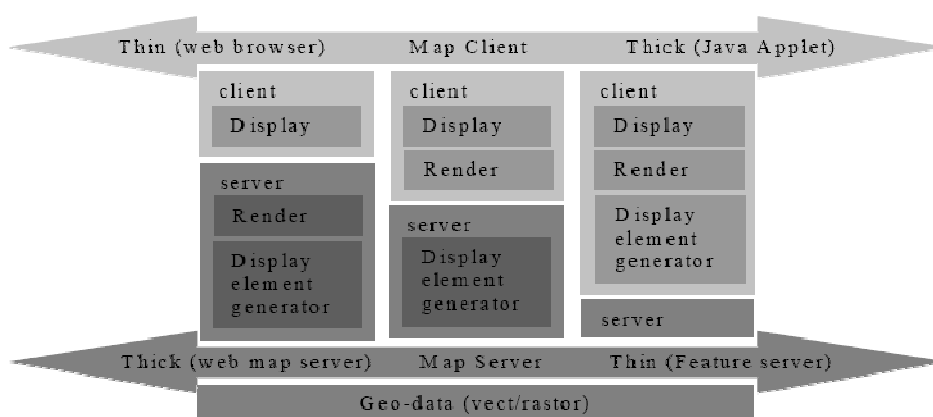
GIS funkcinės galimybės, duomenys ir metaduomenys visame tinkle gali būti priskirti skirtingoms pakopoms (kurios gali būti suskaidytos į sluoksnius) ir juos galima rasti serverio pusėje (viename arba daugiau tarpinių vidurinės įrangos sluoksnių), vidinėje ar kliento pusėje. Visas pakopas galima konfigūruoti nepriklausomai pagal naudotojų poreikius ir plėsti atsižvelgiant į būsimus poreikius.



Pav.: Galima GIS n-pakopė architektūra

<b>Client tier</b>	<b>Kliento pakopa</b>
Web Browser	Interneto naršyklė
Stand alone	Pavienis kompiuteris
Middle tier	Vidurinė pakopa
<b>Web/Communication &amp; Presentation layers</b>	<b>Interneto/komunikacijos ir pateikimo sluoksniai</b>
Web Cache	Interneto podėlis
Web Server	Interneto serveris
Servlet Engine	Serverio programėlių modulis
Web Connector	Interneto jungtis
Development Tools	Kūrimo įrankiai
SSO Services	SSO paslaugos
<b>Services and Business Logic Layers</b>	<b>Paslaugos ir verslo logikos sluoksniai</b>
Web map application service	Interneto žemėlapių taikomųjų programų paslauga
GIS services	GIS paslaugos
System and application services	Sisteminės ir taikomųjų programų paslaugos
MetaData GeoServices	Metaduomenų geopaslaugos
<b>Back-end tier</b>	<b>Vidinė pakopa</b>
Geodata File Services	Geoduomenų failų paslaugos
Database cache	Duomenų bazės podėlis
Internet file system	Interneto failų sistema
DBMS (data+metadata+procedures)	DBMS (duomenys+metaduomenys+procedūros)

Tripakopėje paskirstytoje kliento-serverio architektūroje (kaip parodyta pav.) yra naudotojo sistemos sąsaja *kliento pakopa*, kurioje yra naudotojo paslaugos. Kliento pakopa gali būti arba vien tik interneto naršyklės („plonasis“ klientas), arba interneto naršyklė su atvaizdavimo papildiniais, pavyzdžiui SVG („vidutinis“ klientas), arba „storaisiais“ klientas, pavyzdžiui Java programėlės, ActiveX komponentai (<http://en.wikipedia.org/wiki/ActiveX> - anglų kalba), GIS programos arba belaidis įrenginys (gali būti „plonasis“ klientas). Paskirstytosios sistemos su interneto naršyklėmis gali pasitelkti interneto serverius naudodamos HTTP, HTTPS arba SOAP protokolus. Interneto serveris persiunčia užklausą į pateikimo arba verslo logikos pakopą per atitinkamas jungtis, galinčias palaikyti CGI ([http://en.wikipedia.org/wiki/Common\\_Gateway\\_Interface](http://en.wikipedia.org/wiki/Common_Gateway_Interface) - anglų kalba), NSAPI ir ISAPI sąsajos metodus. „Storasis“ klientas gali naudotis Object Request Brokers (ORB, liet. *Objektų Užklauskos tarpininkais*, [http://en.wikipedia.org/wiki/Object\\_request\\_broker](http://en.wikipedia.org/wiki/Object_request_broker) - anglų kalba) programine įranga, grindžiama CORBA, RMI arba SOAP. Ryšiui su taikomosios programos serveriu naudojama bet kuri iš DCOM specifikacijų.



Pav.: trijų rūšių žemėlapių serveriai ir atitinkami žemėlapių klientai (Zhang S. and Goddard S., 2003)



Thin (web browser)	„Plonasis“ (interneto naršyklė)
client	klientas
Display	Rodymas
server	serveris
Render	Atvaizdavimas
Display element generator	Rodomų elementų generatorius
Thick (web map server)	„Storasis“ (interneto žemėlapių serveris)
AP Client	Žemėlapių klientas
Map Server	Žemėlapių serveris
Thick (Java Applet)	„Storasis“ (JAVA programėlė)
Thin (feature server)	„Plonasis“ (geoobjektų serveris)
Geo-data (vect/rastor)	Geo-duomenys (vektoriniai/rastriniai)

**HTTP/HTTPS (Hypertext Transfer Protocol/Secure Hypertext Transfer Protocol – liet. hiperteksto perdavimo protokolas)** yra standartinis užklauso-atsako protokolas, naudojamas tarp kliento ir vidurinių pakopų interneto serverio. Interneto paslaugos dažniausiai grindžiamos HTTP protokolu. Kiti protokoliai sluoksniais išdėstyti ant jo (pvz. HTTP protokolu perduodami SOAP pranešimai). Klientui pateikus HTTP užklausa, atsakantis serveris išsaugo ir sukuria *išteklis*, tokius kaip HTML failai ir vaizdai. Tarp naudotojo programos ir siunčiančiojo serverio (*angl. origin server*) gali būti keletas tarpinių grandžių – įgaliojusių serverių serveriai, tinklo sietuvų ir tunelių. HTTP nebūtinai turi naudoti vien TCP/IP protokolą ir jo pagalbinis sluoksnius (žr. 1-ąją paskaitą). HTTP galima naudoti kartu su bet kuriuo kitu protokolu internete arba kitame tinkle.

Paprastai HTTP klientas teikia užklausa prisijungdamas TCP (Transmission Control Protocol – liet. Perdavimo Valdymo Protokolas) protokolo pagalba prie konkretaus pagrindinio kompiuterio prievado (numatytasis yra 80 prievadas). HTTP serveris, sekantis šį prievada, laukia kol klientas išsiųs užklauso pranešimą. Gavęs užklausa, serveris išsiunčia būsenos eilutę, pavyzdžiui, „HTTP/1.1 200 OK“, ir savo pranešimą, kuriame gali būti prašomas failas, klaidos pranešimas arba kokia nors kita informacija.

Ištekliai, kuriais galima naudotis pasitelkiant HTTP, yra identifikuojami naudojant universaliuosius išteklių identifikatorius (URI - Uniform Resource Identifiers) arba universaliuosius adresus (URL - Uniform Resource Locators) naudojant http: arba https URI schemas.

*Vidurinioji pakopa* dažnai yra padalinama į dvi arba daugiau posistemių arba sluoksnių, atliekančių skirtingas funkcijas ir pasižyminčių skirtingomis saugumo savybėmis. Serverio pakopą gali sudaryti interneto serveris su atitinkamomis jungtimis, tokiomis kaip JSP (<http://en.wikipedia.org/wiki/JSP> - anglų kalba), Servlets (*liet. Programėlės*) (<http://en.wikipedia.org/wiki/Servlets> - anglų kalba) arba ASP (<http://en.wikipedia.org/wiki/ASP> - anglų kalba), ir sisteminiai, taikomųjų programų ir operacijų (pvz. Enterprise JavaBeans arba .NET) komponentai. „Storasis“ klientas, naudodamas RMI-JRPM arba CORBA-IIOP standartus (<http://en.wikipedia.org/wiki/CORBA> - anglų kalba), gali tiesiogiai kreiptis į EJB komponentus aplenkdamas interneto serverį.

Viduriniojoje pakopoje suteikiamos proceso valdymo paslaugos, pavyzdžiui, protokolų valdymas, HTML/XML srautinis duomenų siuntimas, sisteminės, administracinės ir taikomųjų programų paslaugos (pvz. apkrovimo išlyginimas, priegos prie duomenų valdymas, apmokėjimas ir kt.) bei atliekamos operacijos, kuriomis naudojasi daug

taikomųjų programų. Interneto žemėlapių paslaugos vykdo konkrečias žemėlapių apdorojimo ir GIS funkcijas, tokias kaip kartografinis atvaizdavimas, erdviųjų geoobjektų duomenų srautinis siuntimas, geokodavimas ir maršruto parinkimas, žemėlapių metaduomenų publikavimas ir peržiūrėjimas, skirtingo formato erdviųjų duomenų integravimas, duomenų atsiuntimas, erdvinė ir kartografinė generalizacija (*angl. spatial and cartographic generalization*) ir erdvinės užklauskos – visa tai galima integruoti kaip dalį stambesnio taikomųjų programų serverio.

Komunikacijos ir pateikimo sluoksnio logika valdo taikomosios programos sąsają su naudotoju – ji apdoroja užklauskas, generuoja atsako turinį, jį formatuoja bei pateikia naudotojui (vaizdo žemėlapiai, duomenų srautai ir t.t.). Servlet, JSP arba ASP programinė įranga apdoroja gaunamas užklauskas ir paruošia atsaką. Verslo logika paprastai yra perkeliama į EJB arba .NET verslo komponentus, o rezultatai paprastai perkeliama į JSP arba ASP puslapius.

Verslo logikos sluoksnyje, EJB arba .NET architektūroje, verslo komponentai iškviečiami Servlet, JSP arba ASP programomis ir tvarko daugumą taikomosios programos verslo logikos ir duomenų apdorojimo. Komponentai leidžia verslo logikai išlikti stabiliai visiems kreipiniams ir jie sukurti taip, kad galėtų glaudžiai dirbti su JDBC API (<http://en.wikipedia.org/wiki/JDBC> - anglų kalba) arba ADO.NET (<http://en.wikipedia.org/wiki/ADO> - anglų kalba) operacijose su duomenų baze. Verslo logika apibrėžia veiksmus, kurie susiję su konkrečiu turinio generavimu: duomenų saugojimas ir paėmimas bei operacijų su duomenimis atlikimas. Šio sluoksnio komponentas tvarko gaunamų erdviųjų užklauskų paskirstymą ir katalogizuoja, kurios paslaugos atliekamos su kuriais erdviniais komponentais. Erdviniai komponentai gali paimti ir pertvarkyti žemėlapius bei duomenis į tinkamą formatą prieš siunčiant juos atgal klientui. Šie komponentai gali palaikyti įvairias funkcines galimybes: generuoti vaizdo žemėlapius ir srautiniu būdu nusiųsti vektorinius erdvinis duomenis klientui, atsiųsti požymių duomenis erdvinėms ir lentelinėms užklauskoms, atlikti geokodavimą ir maršruto nustatymo funkcijas, išarchyvuoti ir pateikti erdvinis duomenis reikiamu formatu, erdviųjų metaduomenų saugyklose ieškoti dokumentų, susijusių su erdviniais duomenimis ir paslaugomis, atlikti erdvinę ir kartografinę generalizaciją.

Duomenų valdymo sluoksnis valdo duomenų išsaugojimą duomenų bazėje ir jų paėmimą. Komponentai gali naudoti JDBC API arba ADO.NET, kaip ir visais API, nors operacijos su duomenų bazėmis paprastai valdomos EJB arba .NET komponentų. Prieigos prie duomenų logika apibrėžia operacijas su duomenų baze. Duomenų paėmimas paprastai atliekamas kaip verslo logikos funkcija. Kadangi daugelis erdviųjų duomenų iki šiol saugomi kaip failai, šių duomenų valdymą galima ženkliai patobulinti juos saugant įmonių duomenų bazių sistemose.

Klasikinėje triapakopėje architektūroje trečioji *vidinė pakopa* yra viena arba keletas duomenų bazių, failų serveris arba kitas vidinis įmonės informacijos šaltinis – visi pasižymi duomenų bazių ir failų tvarkymo funkcinėmis galimybėmis. Kai duomenų bazės valdymo komponentai suteikia galimybes saugoti ir tvarkyti erdvinės informacijos failus vienoje saugykloje, galima siūlyti hibridinius sprendimus. Duomenų valdymo komponentai užtikrina duomenų vientisumą, neprieštarumą ir saugumą, ypač kai tai liečia paskirstytą aplinką.

Duomenų bazės sluoksnyje yra rinkinys sąryšinių duomenų bazių lentelių, įskaitant LOB objektus (<http://en.wikipedia.org/wiki/BLOB> - anglų kalba), laikomus plokščiuosius failus,

metaduomenų lenteles, ir išsaugotas procedūras. Paskutiniu atveju visos funkcijos ir procedūros gali greitai ir efektyviai pasiekti duomenų bazėje saugojamą informaciją.

Kitokiu būdu vidinė pakopa gali būti sukurta taip, kad galėtų atlikti didelio našumo apskaičiavimus ((angl. High Performance Computing), pvz., geoduomenų apdorojimui, žr. [http://en.wikipedia.org/wiki/High\\_Performance\\_Computing](http://en.wikipedia.org/wiki/High_Performance_Computing)). Tai klasikinė triapakopė architektūra, išskyrus tai, kad vidinė pusė suteikia priėjimą prie tokių operacijų, kaip topologinis „išvalymas“, generalizacija, perdangos, duomenų konvertavimas, transformavimas bei kitos didelio našumo reikalaujančių geoduomenų apdorojimo paslaugos, todėl darbo procesas tampa efektyvesnis. Naudojantis senų sistemų integravimo metodika žiniatinkliu paremtuose sprendimuose, žemėlapių taikomųjų programų serveriai leidžia lanksčiai integruoti senas sistemas į bet kokias naujai sukurtąsias.

### 3.2. „AJAX“ technologija

Kita interneto technologija, dažnai naudojama interneto žemėlapių taikomiosiose programose – tai **AJAX** – Asinchroninis „JavaScript“ plius XML. (<http://www.adaptivepath.com/ideas/essays/archives/000385.php> - anglų kalba).

Internetas buvo sukurtas naršyti tarp HTML dokumentų, o klasikiniame interneto taikomųjų programų modelyje priimta „spustelk, palauk ir atnaujink“ naudotojo sąveikos paradigma bei sinchroninis užklauso-atsako komunikavimo mechanizmas. Tokia naudotojo sąveikos paradigma nulemia tai, kad interneto taikomiosios programos yra lėtos, nepatikimos, nenašios ir neefektyvios.

AJAX metode „spustelk, palauk ir atnaujink“ vartotojo sąveikos modelis pakeistas į „*dalinį ekrano atnaujinimą*“. AJAX grindžiamose taikomiosiose programose atnaujinami tik tie naudotojo sąsajos elementai, kuriuose yra naujos informacijos, o likusioji naudotojo sąsajos dalis ekrane neatnaujinama.

Sinchroninis užklauso-atsako komunikavimo mechanizmas pakeistas *asinchroniniu* komunikacijos modeliu. AJAX grindžiamose taikomiosiose programose užklausa-atsakas tarp naudotojo ir kliento naršyklės bei tarp naršyklės ir serverio gali būti kaip asinchroninis. Todėl, kol kliento programa fone užklausia informaciją iš serverio, naudotojas gali tęsti darbą su taikomąja programa. Atėjus atsakui iš serverio, atnaujinama tik pakitusi sąsajos dalis.

Tačiau, nepaisant sutrumpintos reikšmės (Asinchroninis „JavaScript“ + XML metodas), AJAX nėra priklausoma nuo kurios nors konkrečios programavimo kalbos, apsikeitimo duomenimis formato ar tinklo komunikacijos objekto. AJAX – tai tik interneto taikomasis modelis, naudojantis *dalinį ekrano atnaujinimą* ir *asinchroninę komunikaciją*. AJAX kliento programai sukurti galima naudoti tris technologijas, iš kurių dažniausiai naudojami DHTML/JavaScript (Asinchroninis JavaScript + XML), Java (asinchroninė Java + XML) ir Flash (asinchroninis ActionScript + SWF).

Nuo interneto daugiapakopės architektūros programinės įrangos AJAX metodas skiriasi **tu**, kad jame tarp naudotojo sąsajos ir serverio kaip tarpinis sluoksnis patalpinamas kliento pusės modulis. Naudotojas bendrauja su kliento pusės moduliu, o kliento pusės modulis bendrauja su serveriu.

AJAX technologija pritaikyta interneto žemėlapių programoms (pvz. „Google Map“). Todėl padidėja našumas, kadangi kai naudotojas paprašo atnaujinti žemėlapi, atnaujinamas tik paties žemėlapio vaizdas, o likusios dalies atnaujinti nereikia. Naudojant AJAX, interneto žemėlapių taikomoji programa greitai braižo ir keičia mastelį, sklandžiai slenka vaizdą gali būti išplėsta, kad rodytų plačią informacijos įvairovę.

### 3.3. Interneto žemėlapių programinės įrangos architektūra

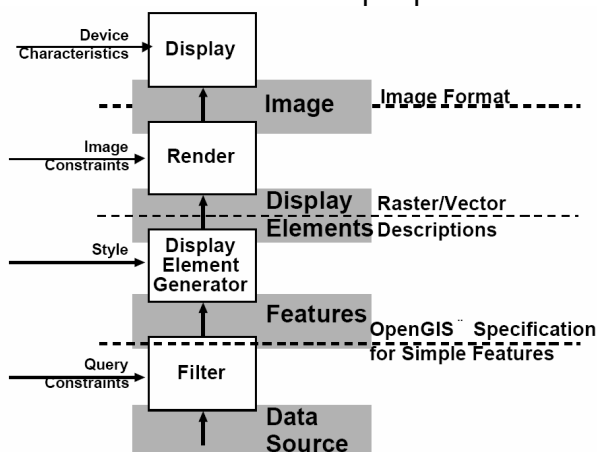
Pirmoji žinoma geografinė interneto svetainė, Xerox, buvo atidaryta 1993-aisiais metais adresu <http://mapweb.parc.xerox.com/map>. Šioje svetainėje buvo patalpintas pasaulio žemėlapis su daugeliu lygių mastelio keitimu. Iš anksto buvo sukurtas didelis žemėlapių atvaizdų skaičius kiekvienam mastelio ir išplėtimo scenarijui. Tai nebuvo tikras interneto žemėlapių serveris, kaip jis suprantamas šiandien, tačiau konceptuali idėja buvo panaši.

Pirmąją vidurinėsios pakopos interneto žemėlapių taikomąją programą „ArcView Internet Map Server“ sukūrė ESRI. Nuo tada ESRI sukūrė keletą naujų savo IMS programos kartų ir šiuo metu siūlo „ArcGIS Server“ taikomąją programą, kuri turi ne tik interneto žemėlapių galimybes, tačiau gali apdoroti geografinius duomenis, geokoduoti, veikia tinkle ir t.t. Didžiulis interneto populiarumas ir prieinamumas, įgautas per paskutinį dešimtmetį, pritraukė GIS įrangos pardavėjus iš viso pasaulio. Kiti pagrindiniai GIS įrangos pardavėjai ir jų interneto žemėlapių programos, tokie kaip „MapInfo“ su „MapExtrem“, „Autodesk“ su „Map Guide“ ir „Intergraph“ su „GeoMedia WebMap“. Be to, kūrėjams ir naudotojams savo gaminius, pvz. „MapServer“, „GeoServer“ ir kt., siūlo atvirosios programinės įrangos gamintojai ir nekomercinės organizacijos.

Kalbant iš esmės, interneto žemėlapių taikomosios programos nėra GIS. Dabartinių IMS funkcinės galimybės yra žymiai mažesnės lyginant su stacionariomis GIS sistemomis. Tik ESRI kompanijos „ArcServer“ programą galima būtų laikyti internetine programine įranga, galinčia atlikti tikros GIS sistemos darbą.

Bendrai kalbant, geo-erdvinių duomenų pateikimą (atvaizdavimą) galima būtų padalinti į keturis procesus (Cuthbert, 1997):

1. Rengiamų pateikti geo-erdvinių duomenų filtravimas arba atrinkimas.
2. Vaizdo elementų generavimas iš atrinktų geo-erdvinių duomenų.
3. Vaizdo elementų surinkimas į atvaizduojamą žemėlapi.
4. Atvaizduoto žemėlapio pateikimas naudotojui.



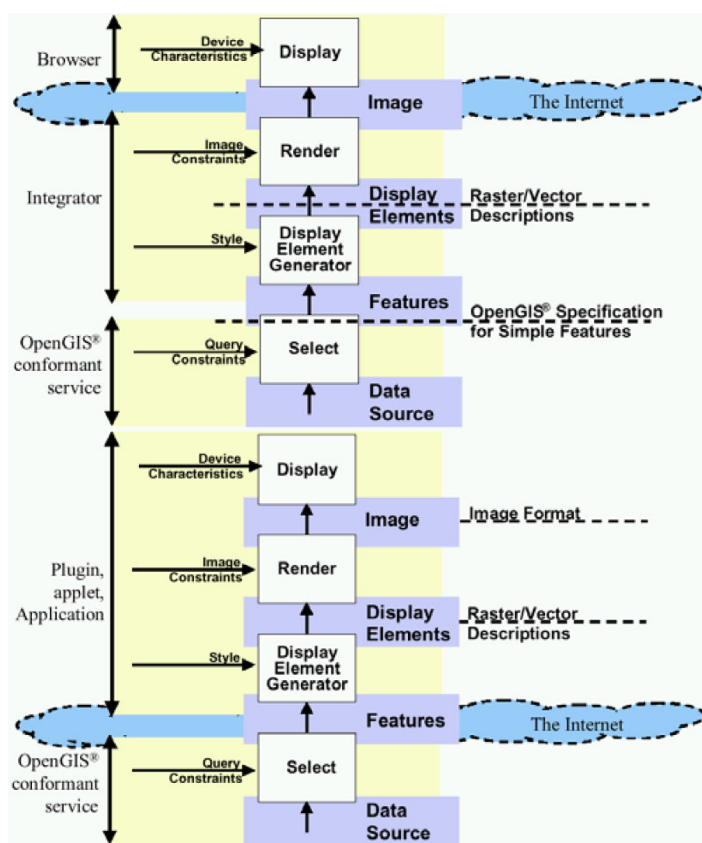
Pav.: OGS atvaizdavimo modelis Atvaizdavimas – tai informacijos pateikimas žmonėms, pvz. žemėlapiu.

Device characteristics	Įrenginio charakteristikos
Display	Pateikimas
Image	Vaizdas
Image	Vaizdo formatas
Image Constraints	Vaizdo apribojimai
Render	Atvaizdavimas
Display elements	Vaizdo elementai
Raster/Vector	Rastriniai/Vektoriniai
Description	Aprašymai
Style	Stilius
Display Element Generator	Vaizdo elementų generavimas
Features	Geoobjektai
OpenGIS Specification for Simple Features	OpenGIS specifikacija paprastiems geoobjektams
Query constraints	Užklauskos apribojimai
Filter	Filtravimas
Data Source	Duomenų šaltinis

Interneto žemėlapių taikomosiose programose šis atvaizdavimo modelis įgyvendinamas kaip daugiapakopė architektūra. Ruošiamų pateikti geo-erdvinių duomenų filtravimas ir atrinkimas realizuojamas *vidinės pusės* (*ang. back-end*) arba *duomenų saugojimo* pakopoje, kurį sudaro failų serveriai, saugojantys erdvinius duomenis (pvz. iš formos ir vaizdo failų, o taip pat iš serverio, kuriame veikia Oracle arba kitokia RDBMS, ir SDE). Atrinkimo procesas inicijuojamas iš *vidurinio* arba *verslo* pakopos, kurią sudaro serverio pusės komponentai, įskaitant interneto serverį, taikomosios programos serverį, erdvinių duomenų serverį ir kt. Ši pakopa - tai vieta, kur apdorojamos visos užklauskos ir atliekamas serverio administravimas.

Atvaizduoto žemėlapio pateikimas naudotojui realizuojamas kliento kompiuteryje. *Kliento* arba *pateikimo pakopą* sudaro kliento pusės komponentai, kurie naudojami užklauskoms į serverį siųsti ir žemėlapiams bei duomenims peržiūrėti. Žiūryklės sudaro HTML/DHTML žiūryklės, Java, kitos savarankiškos žiūryklės arba XML klientai. HTML/DHTML žiūryklės yra įkeliamos interneto naršyklės (pvz. su „JavaScript“). Java žiūryklės gali būti nepriklausomos programos arba jas galima realizuoti naršyklėje panaudojant programėlę.

Antrasis ir trečiasis atvaizdavimo procesai (vaizdo elementų generavimas iš atrinktų geo-erdvinių duomenų ir jų surinkimas atvaizduojamą žemėlapi) gali būti realizuojami *viduriniojoje pakopoje* arba *kliento pakopoje*. Tai priklauso nuo to, koks realizuojamų paslaugų tipas (WMS – Web Map Service (*liet. Interneto Žemėlapių Paslaugos*) arba WFS – Web Feature Service (*liet. Interneto Geoobjektų Paslaugos*)). WMS atitinka „plonąjį“ klientą, o WFS atitinka „storąjį“ kliento architektūrą.



Pav.: „plonojo“ ir „storojo“ klientų palyginimas atvaizduojant geobjektus per internetą (OGC Reference Model ORM versija 0.1.3).

Browser	Naršyklė
Integrator	Integratorius
OpenGIS conformant service	Su OpenGIS suderinamos paslaugos
Device Characteristic	Įrenginio charakteristikos
Display	Pateikimas
Image	Vaizdas
Image format	Vaizdo formatas
Image Constraints	Vaizdo apribojimai
Render	Atvaizdavimas
Display elements	Vaizdo elementai
Raster/Vector	Rastriniai/Vektoriniai
Description	Aprašymai
Style	Stilius
Display Element Generator	Vaizdo elementų generavimas
Features	Geobjektai
OpenGIS Specification for Simple Features	OpenGIS specifikacija paprastiems geobjektams
Query constraints	Užklauskos apribojimai
Select	Atrinkimas
Data Source	Duomenų šaltinis
Plug-in applet, application	Ppildoma programėlė, taikomoji programa

WMS arba vaizdavimo paslauga žemėlapius išleidžia kaip rastrinius vaizdus, kurie viduriniojoje pakopoje būna jau iš anksto paruošti pateikimui, o WFS arba geobjektų paslauga viduriniojoje pakopoje sutvarko geografinius objektus arba vektorinius duomenis



ir atvaizduoja juos klientui pateikti. WMS ir WFS paslaugos yra kaip tarpininkai žemėlapių ir geobjektų duomenų bazių. Atvaizduotas žemėlapis – tai žemėlapių serverio darbo rezultatas, o geobjektai gaunami iš geobjektų serverio.

WMS atsiunčia žemėlapių turinį į kliento taikomąją programą kaip JPEG, PNG arba GIF formato vaizdus. Naujas žemėlapių atvaizdas sugeneruojamas kiekvieną kartą klientui paprašius naujos informacijos. WFS atsiunčia žemėlapių turinį naudotojui kaip geobjektų duomenų srautą arba kaip XML/GML koduotą geometriją. Geobjektų geometrijos duomenys siunčiami srautiniu būdu ir išlieka tik tol, kol klientas yra atviras. WFS užklauso siunčiamos į erdvinių duomenų serverį tik tuomet, kai prireikia papildomų duomenų.

Jeigu interneto žemėlapių klientas gali tik peržiūrėti informaciją, klientui bus atsiųsti vaizdo elementų (rastriniai) duomenys. Tai gali visos standartinės interneto naršyklės. Jeigu žemėlapių klientas turi žemėlapių atvaizdavimo ir vaizdo elementų generavimo galimybes, tuomet žemėlapių klientui bus atsiųsti geobjektai iš geobjektų serverio (*angl. feature server*). Palyginus su WFS, vaizdo elementų siuntimo privalumas tame, kad jis mažiau užima tinklą, kadangi vektoriniai geo-duomenys siunčiami tik vieną kartą, ypač *didinimo/mažinimo* ir *slinkimo* (*angl. pan*) *operacijose* kliento pusėje. Trūkumas tame, kad šio proceso reikalavimai vektorinių duomenų žemėlapių klientui yra daug didesni, negu reikalavimai vaizdo elementų duomenų žemėlapių klientui. Tipiniam vektorinių duomenų žemėlapių klientui reikalinga Java programėlė, Active-X komponentas arba papildinys, skirtas apdoroti vektorinius duomenis.



### 3.4. WMS paslaugos

Pagal „OpenGIS Web Map Server“ realizacijos specifikacija (2006), WMS (Web Map Service – *Interneto žemėlapių paslauga*) iš geografinės informacijos dinamiškai kuria erdvinės atskaitos duomenų žemėlapius.

WMS sukurti žemėlapiai atvaizduojami PNG, GIF arba JPEG formato **vaizdais**, arba kaip vektoriniai grafiniai elementai SVG (Scalable Vector Graphics) arba webCGM (WEB Comuper Graphics Metatfile) formatais.

OGC (Open Geospatial Consortium - Atvirojo Geografinio Konsorciumo) specifikacija apibrėžia, kad OGC interneto žemėlapių serveris realizuoja tris funkcijas arba *operacijas*: *GetCapabilities*, *GetMap* ir *GetFeatureInfo*. *GetCapabilities* funkcija pateikia klientui žemėlapių serverio paslaugų metaduomenis, aprašančius jo galimybes. *GetMap* funkcija aprašo gerai-apibrėžtos žemėlapių užklauskos geografinius ir matmenų parametrus, leidžiančius klientui paprašyti vaizdo žemėlapių. Galiausiai, *GetFeature* funkcija leidžia klientui paprašyti daugiau informacijos apie geobjektus konkrečioje žemėlapių vietoje. *GetCapabilities* ir *GetMap* yra privalomos operacijos, o *GetFeature* operacija yra pasirinktina.

Tipiškoje OGC WMS kliento-serverio sąveikoje klientas iš žemėlapių serverio pirmiausia iškviečia *GetCapabilities* operaciją, kad galėtų nustatyti, ką serveris gali atlikti ir kokius žemėlapius gali pateikti. Po to, kad galėtų gauti žemėlapių vaizdą, klientas iškviečia *GetMap* operaciją, kuri pateikia informaciją apie žemėlapių serverio galimybes. Galiausiai, norėdamas gauti daugiau informacijos apie geografinį objektą, nustatęs žemėlapyje konkretų tašką klientas gali iškviešti *GetFeatureInfo* operaciją.

WMS Operations
+ <i>GetCapabilities</i> ( : <i>MetadataRequest</i> ) : <i>MetadataResponse</i>
+ <i>GetFeatureInfo</i> ( : <i>FeatureInfoRequest</i> ) : <i>FeatureInfoResponse</i>
+ <i>GetMap</i> ( : <i>MapRequest</i> ) : <i>MapResponse</i>

#### Pav.: OGC WMS funkcijos/operacijos

WMS pagrindinis kompiuteris turi palaikyti HTTP protokolą paskirstytoje platformoje. HTTP palaiko du užklauskos būdus: GET ir POST ([http://en.wikipedia.org/wiki/HTTP#Request\\_methods](http://en.wikipedia.org/wiki/HTTP#Request_methods) – anglų kalba). WMS serveris gali pasiūlyti vieną arba abu šiuos būdus. GET metodo palaikymas yra privalomas, o POST metodo palaikymas neprivalomas. Kaip parodyta žemiau, įvairioms operacijoms ir parametrų URL koduoti HTTP GET atveju naudojamos raktinio žodžio-reikšmės poros (KVP, *keyword-value pairs*). HTTP POST atveju, POST dokumento pagrindinėje dalyje kaip kodavimo kalba naudojama XML.

WMS operacijas galima iškviešti standartinės interneto naršyklės pagalba nusiunčiant WMS pagrindiniam kompiuteriui GET užklauską HTTP universalios adreso (URL, Uniform Resource Locators) forma. HTTP URL tinkle suranda išteklių kiekvienos operacijos, kurią palaiko paslaugos teikėjas. Tokių URL turinys priklauso nuo kviečiamos operacijos. Žemėlapių prašymo URL nurodomas žemėlapių turinys, geografinė žemėlapių apimtis, norima koordinacijų sistema ir pateikiamas žemėlapių atvaizdo dydis.

Kiekvienas URL turi atitikti IETF RFC 2616 (dalis 3.2.2 „HTTP URL“) apibrėžimą, bet kitais atžvilgiais jie priklauso nuo realizacijos. OGC WMS specifikacijoje apibrėžiama tik užklauso dalis, sudaranti patį paslaugos prašymą.

URL eilutę sudaro protokolas („http“ arba „https“), pagrindinio kompiuterio (serverio) vardas arba IP adresas, pasirenkamo prievado numeris, kelias, privalomas klausuko ženklas „?“ (skiriamasis ženklas, nurodantis užklauso eilutės pradžią), ir laisvai pasirenkama eilutė su viena arba keletu nuo serverio priklausančių parametru, pasibaigiančių „&“ ženklu (skiriamasis ženklas tarp užklauso eilutės parametru). URL eilutė sudaroma pagal IETF RFC 2616. Tokiu būdu, URL eilutė nurodo tinklo adresą konkretaus serverio, į kurį bus siunčiami konkrečios operacijos užklauso pranešimai.

WMS OGC specifikacija apibrėžia, kaip pateikti URL eilutės parametrus arba užklauso dalis, norint sudaryti užbaigtą užklauso pranešimą. Visos WMS operacijos turi keletą privalomų arba neprivalomų užklauso parametru. Kiekvienas parametras apibrėžiamas vardu. Kiekvienas parametras gali turėti vieną arba kelias reikšmes „name=value&“ forma. Reikšmės apibrėžiamos pagal specifikaciją arba klientas jas pasirenka remdamasis paslaugos metaduomenis.

**Lentelė: URL eilutė, skirta užbaigti WMS GET užklauso pranešimą**

URL komponentas	Apibrėžimas
http://host[:port]/path[?{name[=value]&}]	Paslaugos operacijos URL prievardis. [ ] nurodo 0 arba 1 neprivalomos dalies egzempliorius; { } nurodo 0 arba daugiau egzemplioriu ( <i>angl. occurrences</i> ).
name=value&	Viena arba daugiau standartinių užklauso parametru vardo/reikšmės porų, kaip apibrėžta kiekvienai <i>operacijai</i> WMS OGC specifikacijoje.

Gavęs tinkamai suformuotą užklausą, serveris siunčia atsaką kaip failą, kuriame gali būti tekstas arba žemėlapio vaizdas. Kartu su šiais atsakais siunčiamas atitinkamas „Multipurpose Internet Mail Extensions“ (MIME) (IETF RFC 2045) tipo pranešimas.

Atsakydamas į *GetCapabilities* užklausą, OGC interneto žemėlapiu serveris sukuria XML dokumentą su interneto žemėlapiu serverio paslaugu metaduomenimis, nurodančiais visas jo palaikomas operacijas ir informaciją apie turimus žemėlapius. Kliento taikomoji programa turi išanalizuoti XML galimybiu dokumentą ir pasiimti žemėlapiu užklausiai reikiama informaciją.

Kartu su informacija apie galimybes, klientas gali paprašyti žemėlapiu vaizdo iš žemėlapiu serverio. *GetMap* užklausoje paprastai yra keletas privalomų parametru, tokiu kaip Layers (sluoksniai), Styles (stiliai), Bounding Box (BBox) (rėmelis), erdvinės atskaitos sistemos (SRS – Spatial Reference System), pateikiamo žemėlapiu plotis (Width) ir aukštis (Height) ir vaizdo formatas (Format). Sluoksniai nurodo kokia informacija bus rodoma žemėlapyje (pvz. keliai, upės, miestai ir t.t.). Stiliai naudojami sluoksniams aprašyti, pavyzdžiui, linijos reiškia kelius, apskritimai reiškia miestus ir t.t. BBox - tai keturiu koordinatiniu reikšmiu rinkinys (minX, minY, maxX, maxY), apibrėžiantis ant žemės paviršiaus stačiakampį plotą, kurio žemėlapiu ketinama sudaryti. SRS - tai naudojama projekcinu koordinacių sistema

(smulkesnis aprašymas pateiktas OGC specifikacijoje adresu <http://www.opengeospatial.org/standards/wms> - anglų kalba).

```
http://yourfavoritesite.com/WMS?  
VERSION=1.1.0&  
REQUEST=GetMap&  
BBOX=0.0,0.0,1.0,1.0&  
LAYERS=Rivers,Roads,Houses&  
STYLES=CenterLine,CenterLine,Outline
```

**Pav.: nepilnas pavyzdys HTTP GET užklauso žemėlapiui, kuriame žemėlapijo atvaizdavimo informacija apibrėžiama Layers ir Styles patametrals.**

Panaudojant vaizdo formatus, kurie palaiko skaidrius fonus (pvz. GIF arba PNG), tame pačiame pateikiamo žemėlapijo vaizde gali persikloti vienas arba daugiau žemėlapių. Šie žemėlapijai gali būti paprašyti iš skirtingų paskirstytų žemėlapių serverių.

Paprasciausias WMS duomenis saugo kaip „Sluoksnius“ („Layers“) – iš serverio paprašomas bazinis geografinės informacijos vienetas ir jam pasiūlomas tam tikras skaičius iš anksto nustatytų „Stilių“ („Styles“), kuriais tie sluoksniai turi būti atvaizduoti. Vienas arba keletas sluoksnių, sudarančių kompiuterio ekrane tinkamą pateikti skaitmeninį vaizdą, vadinami „Žemėlapiu“ („Map“).

OGC WMS žemėlapiui ir sluoksniams gali palaikyti pagrindines koordinačių sistemas. Atvaizdavimo operacijos metu, WMS gali konvertuoti geografinius duomenis iš sluoksnių arba duomenų šaltinio koordinačių sistemos į žemėlapijo koordinačių sistemą. Sluoksniams gali turėti susijusias vertikalią, laikinąją arba kitokią koordinačių sistemą.

Gavęs žemėlapij, *GetFeatureInfo* funkcijos pagalba klientas gali paprašyti konkretaus žemėlapijo taško geobjekto duomenų. Atsakas iš WMS serverio būtų siunčiamas vienu iš trijų išvesties formatų: GML (Geography Markup Language) failu, paprastu tekstiniu failu arba HTML failu. GML – tai XML grindžiamas kodavimo tipas, nurodantis informaciją apie geobjektus. Į geobjektus įeina taškai, linijos, daugiakampiai ir kitkas, kas apibrėžta OGC paprastų geobjektų (Simple Features) specifikacijoje (<http://www.opengeospatial.org/standards/sfa> - anglų kalba). Kliento taikomoji programa išanalizuoja GML ir pateikia interneto naršyklėje informaciją apie geobjektus.

### 3.5. WFS paslaugos

OGC interneto žemėlapių paslauga (WMS - Web Map Service) leidžia pateikti žemėlapių vaizdus iš daugelio WMS, esančių kliento kompiuteryje, tuo tarpu OGC interneto geobjektų paslaugų (WFS - Web Feature Service) leidžia dirbti su **geobjektais** arba vektorinėmis geoerdvinių duomenų išraiškomis, užkoduotomis Geography Markup Language (GML) kalba iš daugelio WFS.

OGC WFS specifikacija apibrėžia sąsajas ir jų funkcijas arba operacijas, susijusias su duomenų paėmimu ir manipuliavimu geografiniais objektais naudojant HTTP paskirstytose skaičiavimo platformose. OGC WFS palaiko *insert* (įterpti), *update* (atnaujinti), *lock* (užrakinti), *query* (užklausti) ir *discovery* (atidengimas) operacijas. Šias operacijas galima panaudoti kiekvienai atskirai savybei. Galima sukurti naują geobjektą, jį panaikinti, atnaujinti ir užklausti, atsižvelgiant į erdvinius ir neerdvinius apribojimus.

**Geobjektas** - tai realaus pasaulio reiškinių abstrakcija. Pagal OGC, *geobjektas* tai dalykas, kuris aprašytas savybių rinkiniu, kur kiekviena savybė gali būti nurodoma kaip **kortežas** (*angl. tuple*) {vardas, tipas, reikšmė}. Kiekvienos geobjekto savybės vardas ir tipas apsprendžiamas jos tipo apibrėžimo. Geobjektas turi bent vieną savybę, kuri geometriškai susijusi ir susieta su vieta Žemėje. Tačiau, geobjektai gali visai neturėti geometrinių savybių. ISO nustato, kad konkretus *geobjektas* turi turėti geobjekto požymius, geobjekto tarpusavio sąryšius ir geobjekto operacijas (apibrėžti matematiniai veiksmai, skirti apdoroti informaciją apie geobjektus). Atskiri geobjektai skirstomi į keturis tipus, pasižyminčius bendromis charakteristikomis.

OGC GML (Geography Markup Language) specifikacijoje aprašomas geobjektų kodavimas XML kalba, skirtas geografinių duomenų saugojimui, perdavimui, apdorojimui ir transformavimui.

Todėl WFS reikalauja, kad jų sąsaja būtų apibrėžta XML kalba. Kaip XML išplėtimą, GML būtina naudoti norint išreikšti geobjektus pateikimui ir atvaizdavimui skirtoje sąsajoje. būtina naudoti XML kalboje atrinkimui reikia naudoti SQL filtravimo kalbą. Išėities duomenys gali būti pasiekiami tik per WFS sąsaja ir nematomi kliento taikomosioms programoms. Xpath išraiškų poaibis gali būti naudojamas atskaitos savybėms nurodyti.

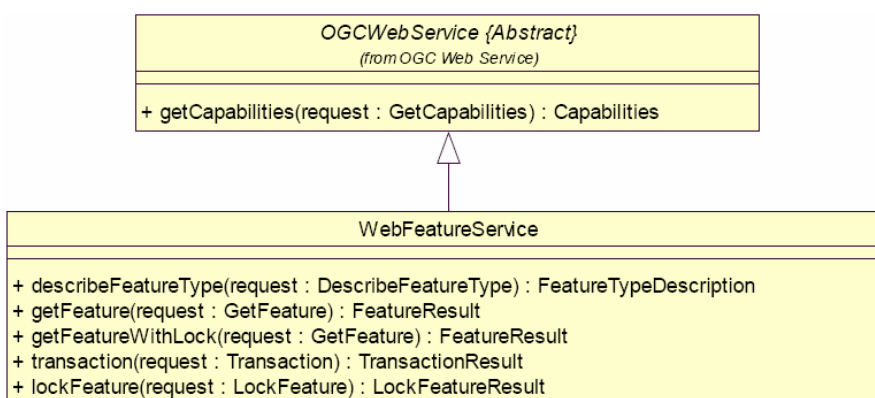
WFS galima apibūdinti tokiu pat būdu kaip WMS, apibrėžiant užklauskos ir atsako keliavimą iš kliento į WMS serverį ir atgal. Užklausa gali būti apdorojama tokia tvarka:

1. Pirmiausia kliento taikomoji programa paprašo dokumento apie galimybes, kuriame yra visų WFS palaikomų operacijų aprašymas ir sąrašas visų geobjektų tipų, kuriuos sistema gali aptarnauti.
2. Toliau, jeigu reikia, kliento taikomoji programa gali paprašyti informacijos apie vieną arba daugiau geobjektų arba elementų tipų, kuriuos gali aptarnauti WFS.
3. Remiantis geobjektų tipo (-ų) apibrėžimu, kliento taikomoji programa sugeneruoja užklausa, kuri siunčiama į interneto serverį.
4. Iškviečiama WFS, ji perskaito ir aptarnauja užklausa. Pavyzdžiui, WFS gali atrinkti konkrečius geobjektus su tam tikromis savybėmis pagal erdvines ir/arba neerdvines užklauskas.
5. Pabaigus apdoroti WFS užklausa, WMS sugeneruoja būsenos ataskaitą ir siunčia ją atgal klientui. Įvykus klaidai, tai matysis būsenos ataskaitoje.

Panašiai kaip WMS, WFS gali atsakyti į GET (raktinio žodžio-reikšmės poros) ir/arba POST/SOAP (XML) HTTP užklauso būdus, tačiau URL užklauso sudarymas į WMS kiekvienu atveju skiriasi. URL sudarymas WFS GET užklausiai yra panašus į aprašytą aukščiau WMS atvejui. Be to, WFS (Web feature services) gali būti pasiūlytos panaudojant HTTPS (<http://en.wikipedia.org/wiki/Https> - anglų kalba) protokolą. HTTPS yra HTTP protokolas per saugų ryšio kanalą, kas leidžia tinklu perduoti šifruotą informaciją.

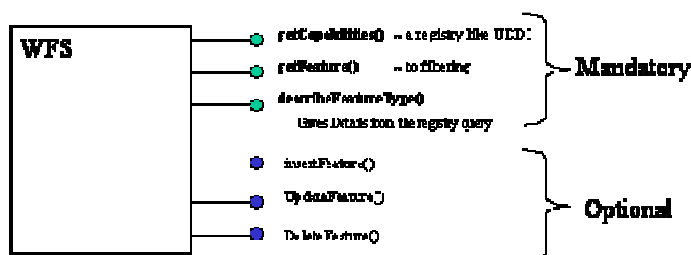
Atsakas į kiekvieną GET metodu koduotą užklausa, yra identiškas atsakams, kurie generuojami XML būdu koduotas užklausoms (POST metodas).

OGC WFS užklausa gali palaikyti tokias operacijas:



**Pav.: WFS sąajos UML diagrama. WebFeatureService klasė gauna *getCapabilities* operaciją iš abstrakčios OGCWebService klasės, kuri yra bendra visoms OGC interneto paslaugoms (įskaitant WMS). WebFeatureService klasė papildomai gali turėti *DescribeFeatureType*, *GetFeature*, *getFeatureWithLock*, *Transaction* ir *LockFeature* metodus.**

Leiskite trumpai apžvelgti šias WFS operacijas: *getCapabilities* (privaloma), *DescribeFeatureType* (privaloma), *GetFeature* (privaloma), *GetGmlObject*, *Transaction – Insert*, *Update* ir *Delete* (neprivalomos) ir *LockFeature* (Ineprivaloma):



**Pav.: OGC WFS sąajos modelis (iš [http://en.wikipedia.org/wiki/Web\\_Feature\\_Service](http://en.wikipedia.org/wiki/Web_Feature_Service) - anglų kalba)**

WFS **GetCapabilities** paslauga aprašo WFS galimybes ir taip nurodo galimus WFS geobjektų tipus ir WFS palaikomas operacijas. Faktiškai ši paslauga sugrąžina paslaugos metaduomenis ir turi palaikyti HTTP GET užklauso metodą (KVP kodavimas).

Šiame fragmente pateiktas pavyzdys GetCapabilities GET URL užklauso:

```
http://www.someWFSserver.com/wfs.cgi?
VERSION=1.1.0&
SERVICE=WFS&
REQUEST=GetCapabilities
```

**Pav.: GET URL užklausa galimybių dokumentui iš WFS.**

Atsako apie galimybes dokumente yra tokios dalys:

- *Paslaugų identifikavimo* dalis pateikia informaciją apie pačias WFS paslaugas, kaip aprašyta OWS bendrosios realizavimo specifikacijos 7.4.3 skyriuje. (<http://www.opengeospatial.org/standards/common>).
- *Paslaugų teikėjo* dalyje pateikiami metaduomenys apie organizaciją, kuri valdo WFS serverį, kaip aprašyta OWS bendrosios realizavimo specifikacijos 7.4.4 skyriuje.
- *Operacijos Metaduomenų* dalyje pateikiami metaduomenys apie WFS operacijas. Operacijų metaduomenų dalies turinys aprašytas OWS bendrosios realizavimo specifikacijos 7.4.5 skyriuje. Šiuose metaduomenyse yra surašyti kiekvienos operacijos parametrai ir apribojimai.
- *Geoobjektų tipų sąrašo (angl. FeatureType list)* dalyje apibrėžiamas sąrašas geoobjektų tipų, išvestų iš gml:AbstractFeatureType ir operacijos su kiekvienu geoobjektų tipu, kurias galima atlikti naudojantis WFS. Papildoma informacija, tokia kaip numatytoji erdvinė atskaitos sistema (SRS – Spatial Reference System), bet kokios kitos palaikomos SRS, arba kokios bebūtų ne SRS (neerdvinių geoobjektų tipams), WFS užklausoms yra pateikiama kiekvienam geoobjektų tipui.
- *ServesGMLObjectType list* dalyje išvardinami GML objektų tipai, kurie neišvedami iš gml:AbstractFeatureType, tačiau prieinami interneto paslaugoms, kurios palaiko GetGMLObject operaciją. Šiuos tipus galima apibrėžti pagrindinėje GML schemeje arba taikomosios programos schemeje panaudojant jos pačios vardų sritį.
- *SupportsGMLObjectType list* dalyje apibrėžiamas sąrašas GML objektų tipų, kuriuos WFS serveris pajėgtų aptarnauti, jeigu jis būtų pritaikytas aptarnauti duomenis kaip aprašyta taikomosios programos schemeje, kuri arba tiesiogiai naudojo tuos GML objektų tipus (ne abstraktiems tipams) arba apibrėžė išvestinius tipus pagal tų tipų pagrindą.
- *Filtravimo galimybių (Filter capabilities)* neprivaloma dalis, kurioje apibrėžiamas sąrašas WMS palaikomų filtravimo operacijų. Jeigu dalis neapibrėžta, tuomet serveris palaiko tik minimalų numatytų filtravimo operacijų rinkinį, kaip apibrėžta OGC filtro kodavimo realizacijos specifikacijoje (<http://www.opengeospatial.org/standards/filter> - anglų kalba). Dalies schema taip pat apibrėžiama toje pačioje specifikacijoje.

Smulkų atsako elementų aprašymą šiai ir kitoms WMS operacijoms galima rasti OGC WMS specifikacijoje, kaip ir smulkius aprašymus GET URL WFS užklauso parametrų, naudojant standartinį CGI stiliaus raktinio žodžio–reikšmės poros kodavimą WFS operacijoms, taipogi aprašytoms OpenGIS Web Feature Service (WFS) realizacijos specifikacijoje ([http://portal.opengeospatial.org/files/?artifact\\_id=8339](http://portal.opengeospatial.org/files/?artifact_id=8339) – anglų kalba).

The WFS **DescribeFeatureType** paslauga apibūdina struktūrą kiekvieno geoobjekto tipo, kuriuos gali aptarnauti WFS. Operacija sukuria schemą, kuri apibrėžia, kaip WFS realizacija nori, kad įvedime būtų koduojamas konkretus geoobjektas (per Insert ir Update

užklausas) ir kaip geobjektai bus generuojamos išėjime (atsakant į GetFeature ir GetGmlObject užklausas).

Žemiau pateiktas DescribeFeatureType GET URL užklauso fragmentas:

```
http://www.someserver.com/wfs.cgi?
SERVICE=WFS&
VERSION=1.1.0&
REQUEST=DescribeFeatureType&
TYPENAME=Buildings_100K
```

**Pav.: GET URL užklausa scheminio aprašymo Buildings\_100K geobjekto tipo.**

Vienintelė privaloma išvestis yra GML3 taikomosios programos schema, aprašyta naudojant XML schemą ir turi būti sugeneruota. Kitas schemų kalbas taip pat galima naudoti geobjekto tipui aprašyti, kadangi MIME tipo reikšmė outputFormat požymiui nurodoma galimybių dokumente.

Atsakant į aukščiau aprašytą DescribeFeatureType užklausa, WFS gali sugeneruoti XML schemas dokumentą, kuris yra tokio pavidalo:

```
<?xml version="1.0" ?>
<xsd:schema
  targetNamespace="http://www.someserver.com/myns"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" version="0.1">

  <xsd:import
    namespace="http://www.opengis.net/gml" schemaLocation="../../../gml/3.1.1/base/gml.x
    sd"/>

  <!-- =====
  define global elements
  ===== -->
  <xsd:element name="Buildings_100K"
    type="myns:Buildings_100K_Type"
    substitutionGroup="gml:_Feature"/>

  <!-- =====
  define complex types (classes)
  ===== -->
  <xsd:complexType name="Buildings_100K_Type">
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractFeatureType">
        <xsd:sequence>
          <xsd:element name="shpGeom"
            type="gml:PolygonPropertyType" nillable="false"/>
          <xsd:element name="id" nillable="true" minOccurs="0">
            <simpleType>
              <xsd:restriction base="integer">
                <xsd:totalDigits value="10"/>
              <xsd:/restriction>
            <xsd:/simpleType>
          <xsd:/element>
        <xsd:/sequence>
      <xsd:/extension>
    <xsd:/complexContent>
  </xsd:complexType>

```

```

        <xsd:element name="mailAddress" type="myns:AddressType"
            nillable="true">
            <xsd:/sequence>
        <xsd:/extension>
    <xsd:/complexContent>
<xsd:/complexType>
<xsd:complexType name="AddressType">
    <xsd:sequence>
        <xsd:element name="streetName" nillable="true">
            <xsd:simpleType>
                <xsd:restriction base="string">
                    <xsd:maxLength value="50"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="streetNumber" nillable="true">
            <xsd:simpleType>
                <xsd:restriction base="string">
                    <xsd:maxLength value="10"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="city" nillable="true">
            <xsd:simpleType>
                <xsd:restriction base="string">
                    <xsd:maxLength value="30"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="municipality" nillable="true">
            <xsd:simpleType>
                <xsd:restriction base="string">
                    <xsd:maxLength value="30"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

**Pav.: XML schemas dokumento pavyzdys (xsd), sukurtas atsakant į DescribeFeatureType užklausą.**

WFS **GetFeature** paslauga pasiima geobjektus su nurodytomis savybėmis. GML naudojamas kanoniniu būdu atvaizduoti geobjektus. Dėl šios priežasties būtina suprasti GML modelius naudojamus geobjektams, kas bus aptarta sekančioje paskaitoje.

GML formate geobjektas atvaizduojamas kaip XML `element`. Feature elemento vardas nurodo geobjekto tipą (pvz. `Buildings_100K` arba `gml:Buildings_100K`) ir paprastai nurodomas UpperTitleCase. Feature elemente gali būti rinkinys antrinių elementų, kurie aprašo geobjekto savybių rinkinį. Savybės elemento vardas nurodo savybės tipą, pvz. `streetName` arba `gml:boundedBy`, ir paprastai nurodomas lowerCamelCase.

Savybės reikšmė gali būti duota toje pačioje eilutėje, pavyzdžiui `<xsd:element name="location" type="gml:PointPropertyType" nillable="true"/>`, arba kaip nuoroda kaip vertė šaltinio, nurodyto su nuoroda, esančia kaip XML atributas ypatybės elemento. Jeigu naudojama eilutės forma, tuomet turinys gali būti struktūrizuotas naudojant XML elementus. Pavyzdžiui:



```
<xsd:element name="municipality" nillable="true">
  <xsd:simpleType>
    <xsd:restriction base="string">
      <xsd:maxLength value="30"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

Žemiau pateiktuose fragmentuose pateiktas GetFeatureType GET URL užklauso pavyzdys.

```
http://www.someserver.com/wfs.cgi&
SERVICE=WFS&
VERSION=1.1.0&
REQUEST=GetFeature&
TYPENAME=Buildings_100K,Roads_100K
```

**Pav.: GET URL užklausa užklausa visų savybių visų Buildings\_100K and Roads\_100K geobjektų.**

Privalomas `typeName` požymis naudojamas nurodyti vardą vieno arba daugiau geobjektų tipų klasių, kurios bus užklausiamos. Jų reikšmės turi būti sąrašė vardo erdvė-tinkami vardai (XML Schema tipas `Qname`, pvz. `myns:AddressType`) ir sutapti su vienu iš geobjektų tipų iš WFS galimybių dokumento. Nurodant daugiau kaip vieną tipo vardą reiškia, kad bus atlikta jungtinė operacija.

```
http://www.someserver.com/wfs.cgi&
SERVICE=WFS&
VERSION=1.1.0&
REQUEST=GetFeature&
PROPERTYNAME=Buildings_100K/buildingId&
TYPENAME= Buildings_100K
```

**Pav.: GET URL prašymas užklausiai buildingId savybei visiems Buildings\_100K tipo geobjektams.**

Kiekvieno `wfs:PropertyName` elemento reikšmė, pvz. `buildingId`, turi būti vardui tinkamas vardas (XML Schema tipas `Qname`, pvz. `myns:Buildings_100K`) ir sutampa su vienu iš savybių elementų vardų GML vaizdavime atitinkamo geobjekto. Savybių elementų vardus galima gauti pateikus `DescribeFeatureType` užklausą.

```
http://www.someserver.com/wfs.cgi&
SERVICE=WFS&
VERSION=1.1.0&
REQUEST=GetFeature&
PROPERTYNAME= Buildings_100K/buildingId&
FEATUREID= Buildings_100K.1366
```

**Pav.: GET URL prašymas užklausiai buildingId savybės Buildings\_100K tipo geobjektų su 1366 identifikatoriumi.**

```

http://www.someserver.com/wfs.cgi&
SERVICE=WFS&
VERSION=1.1.0&
REQUEST=GetFeature&
TYPENAME=InWaterA_1M&
FILTER=<Filter><Within><PropertyName>Buildings_100K/wkbGeom<PropertyName>
<gml:Envelope><gml:lowerCorner>80 80<gml:lowerCorner>
<gml:upperCorner>200 200</gml:upperCorner></gml:Envelope>
</Within></Filter>

```

**Pav.: GET URL prašymas užklausiai visų savybių Buildings\_100K tipo geobjektų su *erdviniu* filtru. XML koduoto filtro parametro reikšmė nustatyta OGC filtro kodavimo realizacijos specifikacijoje.**

WMS GetFeature funkcijoje gali būti vienas arba daugiau Query apibrėžimų, kurie yra sujungti, kad sukurtų galutinį užklauskos rinkinį. Filter elementas naudojamas aprašyti užklauskos apribojimus. Ir erdviniai ir neerdviniai apribojimai gali būti nurodyti filtro kodavimo specifikacijoje.

Jeigu interneto geobjektų paslauga turi įdiegtą Xlink traversal, WFS klientas gali naudoti traverseXlinkDepth ir traverseXlinkExpiry požymius norėdamas prašyti šios lizdinės savybės Xlink nurodantį elemento lokatoriaus požymį (href) Xlink nuorodos, jeigu įmanoma traversuojami ir išsprendžiami. Xlink susiejimo elementas yra dalis XML Path Language (XPath) specifikacijos, kuri nurodo į XML dokumento dalis, aptartas ankstesniojoje paskaitoje.. WFS realizacija turi palaikyti tik poaibį XPath kalbos, kuri iš anksto apibrėžta OGC WMS specifikacijoje.

Jeigu WMS GetFeature outputFormat požymis nustatomas į text/gml; subtype=gml/3.1.1 (tai yra numatytoji reikšmė), tuomet klientas gaus atsaką iš WMS GetFeature užklauskos kaip galiojantį GML3 dokumentą.

Atsakydama į GetFeature užklausa, WFS gali arba generuoti pilną atsako dokumentą arba gali paprasčiausiai pateikti geobjektų skaičių. Pasirinktinis resultType požymis naudojamas valdyti WFS atsako tipą.

Šiame pavyzdyje pavaizduota GetFeature užklausa geobjekto su 1366 identifikatoriumi, naudojant Towns\_100K/townId kaip typeName/PropertyName elementus.

```

<wfs:GetFeature
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:myns="http://www.someserver.com/myns"
  xsi:schemaLocation="http://www.opengis.net/wfs ../wfs/1.1.0/WFS.xsd"
  version="1.1.0"
  service="WFS"
  outputFormat="text/xml; subtype=gml/3.1.1">
  <Query typeName="myns:Towns_100K">
    <wfs:PropertyName>myns:townId</wfs:PropertyName>
    <wfs:PropertyName>gml:directedNode</wfs:PropertyName>
    <ogc:Filter>
      <ogc:GmlObjectId myns:townId="1366"/>
    </ogc:Filter>
  </myns:Query>
</wfs:GetFeature>

```

**Pav.: POST XML prašymas užklausi `townId` savybės `Towns_100K` tipo geobjektui su 1366 identifikatoriumi.**

Atsake yra `xlink:href`, bet ji gražinama kaip:

```
<?xml version="1.0" ?>
<wfs:FeatureCollection
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.someserver.com/myns PopulatedPlaces.xsd
  http://www.opengis.net/wfs ../wfs/1.1.0/WFS.xsd">
  <gml:boundedBy>
    <gml:Envelope
      srsName="http://www.opengis.net/gml/srs/epsg.xml#63266405">
      <gml:lowerCorner>90 100</gml:lowerCorner>
      <gml:upperCorner>190,200</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <gml:featureMember>
    <Towns_100K myns:townId="1366">
      <myns:name>Vilnius</myns:name>
      <gml:directedNode orientation="+" xlink:href="#n1"/>
    </Town>
  </gml:featureMember>
</wfs:FeatureCollection>
```

**Pav.: POST GML atsakas į užklausą `townId` savybės `Towns_100K` tipo geobjektui su 1366 identifikatoriumi.**

Nebūtina WMS **GetGmlObject** operacija leidžia geobjektus ir elementus iš WMS paimti pagal jų ID. Kaip atsakas į šią operaciją klientui atsiunčiamas XML dokumento fragmentas. WFS GetGmlObject paslauga gali apdoroti užklausą ir paimti konkrečius elementus traversuojant Xlink nuorodas, kurie nurodo į savo XML ID.

GetGmlObject prašo WFS atsiųsti elementą su `gml:id` požymio reikšme, nustatyta `ogc:GmlObjectId` elemento viduje. `gml:id` požymio reikšmė naudojama kaip unikalus raktas, norint paimti sudėtinį elementą su `gml:id` požymio reikšme. Paprašytas elementas gali būti bet kuris GML duomenų, kuriuos aptarnauja WFS, identifikuotas elementas, pavyzdžiui geobjektas, geometrija, topologija arba kompleksinis požymis.

Kiekvienas konkretus geobjektas WFS realizacijoje turi būti unikaliam identifikuojamas. Geoobjekto identifikatorius yra unikalus kiekvienam serveriui ir gali būti naudojamas pateikiant nuorodą į tą patį konkretų geobjektą. Geoobjekto identifikatorius koduojamas kaip aprašyta OGC filtro kodavimo specifikacijoje. Geoobjekto identifikatorių galima naudoti bet kada, kai tik prireikia nuorodos į geobjektą (pvz. GetGmlObject operacijai).

WFS **operacijų** paslauga vykdo operacijas, kurios modifikuoja geobjektus: kuria, atnaujina ir panaikina konkrečius geobjektus. WFS sugeneruos XML atsako dokumentą, kurią bus nurodyta operacijos užbaigimo būseną.

Transaction elemente gali būti vienas arba daugiau Insert, Update arba Delete elementų, kurie aprašo operacijas sukurti, modifikuoti arba panaikinti konkrečius geobjektus. WFS apdoroja Insert, Update ir Delete elementus jų pasirodymo operacijos užklausoje tvarka.

`Insert` elementas naudojamas kurti naujiems geobjektams. Numatyta pradinė sukurto geobjekto būseną išreiškianti GML3 forma ir turi būti sutinkanti su GML3 taikomosios programos schema, sukurta `DescribeFeatureType` operacijos.

`Update` elementas aprašo vieną atnaujinimo operaciją, kuri naudojama vienam geobjektui arba jų to paties tipo rinkiniui. Vienoje operacijos užklausoje gali būti daug `Update` operacijų.

`Delete` elementas naudojamas pašalinti vieną arba daugiau geobjektų. Žemiau nurodytame fragmente parodytas operacijos `Delete` GET URL užklauso pavyzdys:

```
http://www.someserver.com/wfs.cgi?  
SERVICE=WFS&  
VERSION=1.1.0&  
REQUEST=Transaction&  
OPERATION=Delete&  
TYPENAME=Buildings_100K&  
BBOX=10,10,20,20
```

**Pav.: GET URL užklausa WMS operacijai, kurios metu būtų panaikinti visi `Buildings_100K` tipo geobjektai nurodyto rėmelio `BBOX` viduje.**

Išreiškiant ilgas `INSERT` arba `UPDATE` užklausas, nepatogu naudoti raktinio žodžio reikšmės poros kodavimą.

WFS ***LockFeature*** paslauga užrakina vieną arba daugiau geobjektų operacijos laikui, tokiu būdu išlaikydama operacijų serijškumą.

Žemiau pateiktuose fragmentuose aprašytas pavyzdys `LockFeature` GET URL užklauso

```
http://www.someserver.com/wfs.cgi?  
SERVICE=WFS&  
VERSION=1.1.0&  
REQUEST=LockFeature&  
TYPENAME=Buildings_100K
```

**Pav.: GET URL užklausa užrakinti visus geobjektus, kurių tipas `Buildings_100K`**

### 3.6. Interneto aprėpties paslauga

**Aprėptis** yra geoobjektas, kuris veikia kaip funkcija, skirta paimti reikšmes iš jų diapazono kiekvienai tiesioginei pozicijai jos erdvinės-laikinės srities (<http://www.opengeospatial.org/standards/wcs>). Aprėptis yra tolydus fenomenas, kadangi vektoriniai arba atskiri geoobjektai atitinka diskrečius objektus taškų, linijų arba daugiakampių forma. Yra skirtingi aprėpties tipai – rinkinys daugiakampių, reikšmių tinklelis, matematinė funkcija arba jų derinys (<http://www.opengeospatial.org/standards/gc>).

Tinklelinė aprėptis yra reikšmė (skaliarinę arba vektorinę) tinkelio taškuose. Kiekvieno taško aprėpties pateikiama reikšmė yra ta reikšmė, kuri yra arčiausia tinkelio taško. Dažnai tinkelio aprėpties sritis sudaroma iš taisyklingai išsidėsčiusių vietų išilgai 0,1, 2 arba 3 ašių erdvinių koordinačių atskaitos sistemoje. Reikšmė tinkelio taške gali būti skaliarinė, tokia kaip aukštis (viena juosta) arba vektorinė, tokia kaip šviesumo reikšmės skirtingose dalyse elektromagnetinio spektro (multiple bands).

OGC interneto aprėpties paslaugos (WCS) specifikacija aprašo, kaip tinkle palaikomas elektroninis geoerdvinių duomenų gavimas kaip daugiareikšminės „aprėptys“. Dabartinė OGC WCS versija apsiriboja tik tinklinės aprėpties aprašymu.

Kaip OGC WMS ir WFS, WCS leidžia klientams paimti dalimis serverio erdviniu duomenis remiantis erdviniais apribojimais. Tačiau, kitaip negu WMS, kuris atvaizduoja erdvinius duomenis kaip žemėlapių atvaizdą, WFS sugražina geoerdvinius duomenis kartu su jų savybėmis. Šie duomenys, kaip ir WFS atveju, gali būti apdoroti, o ne tik atvaizduoti. Kitaip negu WFS, kuri pateikia diskrečius geoerdvinius objektus, WMS pateikia daugiadimensines aprėptis, atspindinčias priklausymo nuo erdvės reiškinių.

WCS atlieka tris operacijas: *GetCapabilities*, *DescribeCoverage* ir *GetCoverage*. *GetCapabilities* operacija pateikia XML dokumentą aprašančią paslaugą ir trumpus aprašymus aprėpčių, kurių klientas gali užklausti. *DescribeCoverage* operacija leidžia klientui užklausti pilną aprašymą vienos arba keletos aprėpčių, aptarnaujamų konkretaus WCS serverio. *GetCoverage* operacija pateikia aprėptį, užkoduotą populiariu aprėpties formatu. *GetCoverage* sintaksė ir semantika turi kažkiek panašumo su WMS *GetMap* ir WFS *GetFeature* užklausomis, tačiau keletas išplėtimų palaiko aprėpčių paėmimą, o ne tik statinius žemėlapius arba diskrečius geoobjektus. *GetCoverage* operacija paprastai eina po *GetCapabilities* ir *DescribeCoverage* operacijų atsakai parodo, kurios užklauskos yra leidžiamos ir kokie duomenys prieinami.

### 3.7. Filtro kodavimas

OGC Filtro kodavimo realizacijos specifikacija (<http://www.opengeospatial.org/standards/filter> - anglų kalba) iš pradžių buvo dalis Web Feature Server (WFS) versijos 0.0.10 specifikacijos. Ji buvo perkelta į atskirą dokumentą, kadangi filtro kodavimas gali būti naudojamas daugelyje paslaugų (WFS, WCS, Gazetteer, Web Registries), kurioms reikia galimybės išreikšti užklausos apribojimą XML.

*Filtrai* naudojamas aprašyti rinkinį geoobjektų, kuriuos pagal tai reikia apdoroti. Operacijų rinkinį gali sudaryti vienas arba daugiau sunumeruotų geoobjektų arba geoobjektų rinkinys, aprašytas nustatant erdvinius arba neerdvinius geoobjekto tipo geometrinių arba skaliarinių savybių apribojimus. *Filtrų išraiškos* naudoja objekto tipo savybių reikšmes apribojimams sudaryti.

OGC filtro kodavimo specifikacija apibūdina XML kodavimą OGC Common Catalog Query Language (CQL) kaip užklausos predikato sistemą. Naudojant XML įrankius, XML filtras gali būti patvirtinamas, analizuojamas ir po to transformuojamas į kitokią užklausų kalbą (pvz. WHERE punktas SQL SELECT sakinyje arba XPath išraiška išgauti informaciją iš XML dokumentų), kuri reikalinga norint paimti arba modifikuoti konkrečius objektus duomenų bazėse.

Ši specifikacija apibrėžia išraišką šakniniame elemente `<Filter>`. `<Filter>` elementas apibrėžiamas žemiau parodytu XML schemas fragmentu:

```
<xsd:element name="Filter" type="ogc:FilterType"/>
  <xsd:complexType name="FilterType">
    <xsd:choice>
      <xsd:element ref="ogc:spatialOps"/>
      <xsd:element ref="ogc:comparisonOps"/>
      <xsd:element ref="ogc:logicOps"/>
      <xsd:element ref="ogc:_Id" maxOccurs="unbounded"/>
    </xsd:choice>
  </xsd:complexType>
```

**Pav.: OGC XSD schema `<Filter>` elementui.**

Elementai `logicOps`, `comparisonOps`, ir `spatialOps` yra pakaitalų grupės loginiams, erdviniams ir palyginimo operatoriams. Kad nustatytų geometrinius kintamuosius, kurie tenkina topologines sąlygas, erdvinis operatorius naudoja erdvinis sąryšius.

Ši OGC specifikacija taipogi apibrėžia pagrindinių aritmetinių veiksmų ir funkcijų kodavimą.

Žemiau pateiktuose fragmentuose apibrėžtas filtras, naudojantis `IsBetween` operatorių. Filtras atpažįsta visus geoobjektus, kurių `FLOOR` yra tarp 2 ir 5.

```
<Filter>
  <PropertyIsBetween>
    <PropertyName>FLOOR</PropertyName>
    <LowerBoundary><Literal>2</Literal></LowerBoundary>
    <UpperBoundary><Literal>5</Literal></UpperBoundary>
  </PropertyIsBetween>
</Filter>
```

**Pav.: OGC filtro palyginimo operatorius.**

### 3.8. *Sluoksnių stiliaus deskriptorius ir simbolikos kodavimas*

WMS sukuria žemėlapių vaizdą kaip atsakymą į iš kliento kompiuterio gautą naudotojo užklausą. Naudotojams gali reikėti vizualizuoti arba atvaizduoti iš serverio gautus geobjektų duomenis žemėlapyje su naudotojo nustatytais simbolikos stiliais. Naujasis OGC sluoksnių stiliaus deskriptorius (Styled Layer Descriptor) profilis iš WMS realizacijos specifikacijos (<http://www.opengeospatial.org/standards/sld> - anglų kalba) ir Simbolikos kodavimo realizacijos specifikacija (<http://www.opengeospatial.org/standards/symbol> - anglų kalba) išplečia OGC WMS specifikaciją ir tokiu būdu įgalina vartoti naudotojo nustatytą simboliką geobjektams ir aprėpties duomenims.

Dabartinė OGC WMS specifikacija palaiko galimybę nustatyti pačias paprasčiausias stiliaus pasirinktis, panaudojant minimalų rinkinį galimų vizualinių stilių duomenų rinkiniams (žr. pavyzdį aukščiau su `STYLES=CenterLine,CenterLine,Outline` parametrais, WMS naudoja iš anksto nustatytą pagrindinį stilių kiekvienam stiliaus kodui). Naudotojas negali nustatyti savų stiliaus parinkimo taisyklių. Tam reikalinga stiliaus aprašymo kalba, kurią galėtų suprasti ir klientas ir serveris. OGS apibrėžia tokią kalbą ir ji vadinama **Simbolikos kodavimu**. SE kalba aprašyta Simbolikos Kodavimo realizacijos specifikacijoje. Šią kalbą galima naudoti ne tik WMS, tačiau taip pat atvaizdavimui WFS ir WCS sistemose.

Yra įvairių kartografinės simbolikos metodikų ir technikų, o (žr. GII-06 6-oje paskaitoje). Neskaitant SE, OGC siūlo naujas operacijas OGC paslaugoms OGC **Styled Layer Descriptor** (SLD) profilio specifikacijoje. Šios operacijos palaiko kai kurias kartografinio vizualizavimo metodikas. Todėl, *FeatureTypeStyle* elementas SE dokumente gali būti naudojamas pritaikant tą patį simbolikos stilių visiems vieno tipo geobjektams arba viename sluoksnyje. Toks būdas atitinka *vieno taško simbolio* metodikas teminėje kartografijoje. Pavyzdžiui, visos hidrografijos sluoksnių linijiniai ir daugiakampiai geobjektai (upės, upeliai, ežerai, tvenkiniai, vandenynai ir t.t.) gali būti vizualizuojami šviesiai mėlyna spalva daugiakampiams nuspalvinti ir tamsesne mėlyna spalva visų daugiakampių kraštinėms ir visoms linijoms.

Išsamesnės kokybinės arba kiekybinės kartografinio simbolizavimo metodikos simbolizavimui naudoja geobjektų požymių reikšmes. Pavyzdžiui, skirtingų klasių keliai gali būti vaizduojami skirtinga simbolika pagal požymių reikšmes Class lauke. SLD profilis leidžia vykdyti pasirenkamą operaciją *DescribeLayer*, kuri pateikia užklausoje nurodytame sluoksnyje arba sluoksniuose esančių geobjekto tipus, o požymiai gali būti surasti naudojant WFS sąsajos *DescribeFeatureType* operaciją arba WCS sąsajos *DescribeCoverageType* operaciją. Tai būdas, kuriuo klientas gali gauti informaciją apie geobjekto/aprėpties tipą WMS įvardintam sluoksniui.

Paskutinis skaičius WMServices dalyje parodo pavyzdį stiliaus parametrų trims sluoksniams `LAYERS=Rivers,Roads,Houses`, naudojant HTTP GET užklausą. SLD profilis WMS apibrėžia papildomus parametrus, leidžiančius klientui pasiekti, Get užklausoje sluoksniai būtų atvaizduojami pagal kažkokį konkretų stilių. SLD pateikia keturis papildomus parametrus, kuriuos galima panaudoti GetMap GET užklausoje.

Tas pats stilius gali būti aprašytas panaudojant XML Styled-Layer Descriptor (SLD) dokumentą, kaip parodyta žemiau pateiktame paveiksle. **Sluoksnių stiliaus deskriptorius (Styled-Layer Descriptor)** dokumentas aprašo žemėlapių išvaizdą, vartojant naudotojo nustatytą XML kodavimą. SLD dokumente yra *StyledLayerDescriptor* XML elementas, kuriame yra eilė stiliaus-sluoksnių apibrėžimų. Šiuos stiliaus-sluoksnių

apibrėžimus gali panaudoti WMS įvardinti arba naudotojo apibrėžti sluokniai ir stiliai. SLD dokumentas apibrėžiamas kaip seka sluoksnių su stiliaus apibūdinimu.

```
<StyledLayerDescriptor version="1.0.0">
  <NamedLayer>
    <Name>Rivers</Name>
    <NamedStyle>
      <Name>CenterLine</Name>
    </NamedStyle>
  </NamedLayer>
  <NamedLayer>
    <Name>Roads</Name>
    <NamedStyle>
      <Name>CenterLine</Name>
    </NamedStyle>
  </NamedLayer>
  <NamedLayer>
    <Name>Houses</Name>
    <NamedStyle>
      <Name>Outline</Name>
    </NamedStyle>
  </NamedLayer>
</StyledLayerDescriptor>
```

**Pav.: sluoksnių stiliaus deskriptoriaus pavyzdys.**

SLD XML dokumentas gali tapti sudėtinu su vartotojo nustatytais stiliais.

OGC siūlo 3 skirtingus SLD kodo iš WMS dokumentų naudojimo būdus:

- WMS HTTP GET užklausoje gali būti nuoroda į nutolusį SLD XML failą.
- SLD XML dokumentas gali būti įtrauktas viduje HTTP GET užklausoje.
- SLD XML dokumentas gali būti įterptas WMS HTTP POST GetMap užklausoje.

Šiame fragmente pateiktas pirmojo būdo pavyzdys:

```
http://yourfavoritesite.com/WMS?
VERSION=1.0.5&
REQUEST=GetMap&
SRS=EPSG%3A4326&
BBOX=0.0,0.0,1.0,1.0&
SLD=http%3A%2F%2Fmyclientsite.com%2FmySLD.xml&
WIDTH=400&
HEIGHT=400&
FORMAT=PNG
```

**Pav.: SLD identifikuojamas naudojant URL. URL paruoštam SLD dokumentui yra <http://myclientsite.com/mySLD.xml>**

Simbolikos Kodavimo (SE, Symbology Encoding) kalba apibrėžia keletą pagrindinių elementų, naudojamų pačios SE, tačiau taip pat ir SLD. SE kalba apibrėžia elementus, naudojamus geobjektų ir aprėpties vaizdavimui. Simbolikos kodavimo šakninis elementas tokiu būdu yra *FeatureTypeStyle* arba *CoverageStyle*. *FeatureTypeStyle* apibrėžia stilių, kuris turi būti taikomas vienam geobjekto tipui. *CoverageStyle* apibrėžia stilių, kuris turi būti taikomas aprėpties duomenų poaibiui.



Daugiau apie SLD ir WMS-SLD operacijas, WMSSLD įvardintus ir vartotojo apibrėžtus sluoksnių ir stilių apibrėžimus, formatą SE žemėlapių stiliaus suteikimo kalbos, skirtos sudaryti georeferencinius žemėlapius su vartotojo apibrėžtu stiliumi, SLD simbolizavimo tipus (apibrėžiančius kaip geobjektas turi atrodyti žemėlapyje), žemėlapių aprašo ir mastelio kūrimas su *GetLegendGraphic* operacija, skaitykite šiais adresais:  
<http://www.opengeospatial.org/standards/sld> ir  
<http://www.opengeospatial.org/standards/symbol>.

### 3.9. Išvados

Interneto žemėlapių paslaugos realizuojamos paskirstytoje skaičiavimų platformoje (DCP – Distributed computing platform). Yra daug technologijų, pavyzdžiui J2EE, .NET, „Common Gateway Interface“ (CGI), „Active Server Pages“ (ASP), „JavaServer Pages“ (JSP), „AJAX“ ir kt., kurios naudojamos kuriant interneto žemėlapių taikomąsias programas. Šios technologijos realizuojamos daugiapakopėje architektūroje. Žemėlapių moduliai gali būti realizuojami skirtingose architektūros pakopose.

Geo-erdviniams duomenims vaizduoti skirtinguose daugiapakopės architektūros sluoksniuose gali būti realizuojami keturi procesai. Todėl, interneto žemėlapių serveriui (Web Mapping Server) klientas yra standartinė interneto naršyklė, galinti siųsti GetMap užklausas ir pateikti atsakus žemėlapių vaizdų forma. Žemėlapių moduliai yra daugiapakopės architektūros serverio pusėje.

Interneto geoobjektų serveris (WFS – Web Feature Server) dirba su vektoriniais duomenimis GML kalboje apibrėžtu formatu. WFS serveriai, atsakydami į kliento užklausą, tik atrenka ir siunčia klientui diskrečius geoobjektų duomenis. Klientas turi galėti pateikti ir apdoroti erdvinis duomenis, ko dabartinės standartinės interneto naršyklės padaryti negali. „Storasis“ turi įsidiegti priemones, kurios galėtų, pavyzdžiui, pateikti, sukurti, modifikuoti ir pašalinti geoobjektus jų serveryje, naudojant HTTP protokolą. Interneto aprėpties serveris, panašiai kaip WFS, gali žiniatinkliu teikti tinklelinius arba rastrinius duomenis.

Siekiant išspręsti geopaslaugų suprantamumo problemas tarp skirtingų interneto žemėlapių sistemų, OpenGIS konsorciumas (OGC) pasiūlė standartinės specifikacijas, įskaitant geo-duomenų pateikimą ir geo-paslaugų sąsajas. Siekiant standartizuoti geoduomenų formatą, buvo pasiūlytos Simple Feature (SF) (OGC-SF, 1998) ir Grid/Coverage (GC) (OGC-GC, 2001) realizacijos specifikacijos, skirtos vektoriniams ir rastriniams duomenims, kurie buvo atnaujinti į naują atitinkamą specifikaciją. Siekiant standartizuoti geopaslaugų tarpusavio sąveiką, buvo pasiūlytos Web Mapping Server (WMS) (OGC-WMS, 2006 dabartinė versija) ir Feature Server (WFS) (OGC-WFS, 2006 dabartinė versija) realizacijos specifikacijos žemėlapių (žemėlapių vaizdo) pateikimui ir geoobjektų (vektorinių duomenų) tvarkymui per naudojantis žiniatinkliu.

OGS taip pat siūlo susijusias specifikacijas geoobjektų filtravimui (atrinkimui) ir atvaizdavimui WMS, WFS ir WCS paslaugose. Yra du būdai, kaip suteikti simboliką geoobjektams naudojant OGC interneto paslaugas: tai naudoti LAYERS (sluoksniai) ir STYLES (stiliai) parametrus su kintamaisiais arba naudoti Sluoksnio stiliaus deskriptoriaus (SLD - Styled-Layer Descriptor) dokumentą WMS apibrėžtam arba naudotojo nustatytam stiliui.

### **Klausimai savarankiškam mokymuisi:**

1. Kokios yra pagrindinės WMS ir WFS išvestys?
2. Koks WMS ir WFS pagrindinis skirtumas?
3. Koks yra esminis skirtumas tarp WMS ir WCS?
4. Koks yra pagrindinis WFS ir WCS skirtumas?
5. Kas yra vektorinis geoobjektas ir ką jis apima? Koks yra pagrindinis skirtumas tarp geoobjekto ir aprėpties?
6. Kas yra filtras interneto paslaugų kontekste? Kokius tris filtrų tipus palaiko OGC?
7. Kuo skiriasi SLD ir SE?

### **Rekomenduojama literatūra:**

- The OpenGIS Web Map Server Cookbook, <http://www.opengeospatial.org/resource/cookbooks>
- OpenGIS Web Map Service (WMS) Implementation Specification, <http://www.opengeospatial.org/standards/wms>
- OpenGIS Web Feature Service (WFS) Implementation Specification, <http://www.opengeospatial.org/standards/wfs>
- Section 3, System Design Strategies An ESRI ® Technical Reference Document, 2007, <http://www.esri.com/library/whitepapers/pdfs/sysdesig.pdf>

## Naudota literatūra

- Zhang, S., and Goddard, S., OpenGIS Conforming Map-Feature Server Implementation Specifications in Component-Based Distributed Systems, *Proceedings of the 2003 International Society for Environmental Information Sciences Conference*, Regina, Canada, July 2003, <http://www.cse.unl.edu/~goddard/Papers/Conference/ISEIS03ZhangGoddard.pdf>
- OpenGIS Web Map Service (WMS) Implementation Specification, <http://www.opengeospatial.org/standards/wms>
- OpenGIS Web Feature Service (WFS) Implementation Specification, <http://www.opengeospatial.org/standards/wfs>
- OpenGIS® Web Coverage Service (WCS) Implementation Specification, <http://www.opengeospatial.org/standards/wcs>
- OpenGIS® Filter Encoding Implementation Specification, <http://www.opengeospatial.org/standards/filter>
- OpenGIS® Styled Layer Descriptor Profile of the Web Map Service Implementation Specification, <http://www.opengeospatial.org/standards/sld>
- OpenGIS Symbology Encoding Implementation Specification, <http://www.opengeospatial.org/standards/symbol>

## Vartojamieji terminai

- Kliento-serverio architektūra
- Daugiapakopė architektūra
- Lygiarangiai įrenginiai
- HTTP/HTTPS protokolas
- „AJAX”
- HTTP GET ir raktinio žodžio-reikšmės poros
- HTTP POST
- Interneto žemėlapių paslaugos (WMServices)
- Interneto geoobjektų paslaugos (WFServices )
- Interneto aprėpties paslaugos (Web Coverage Service)
- Filtro kodavimas (Filter Encoding)
- Simbolikos kodavimas (Symbology Encoding)
- Sluoksniu stiliaus deskriptorius (Styled Layer Descriptor)

## 4. XML tipo kalbos žemėlapiams ir SOA

XML (*Extensible Markup Language*) – tai plačios paskirties ženklavimo kalba. Įterpus semantinius apribojimus, XML galima realizuoti taikomųjų programų kalbas, kurioms priskiriamos GML (*Geographic Markup Language*), ArcXML (*ESRI Arc Extensible Markup Language*), KLM (*Google Keyhole Markup Language*), SVG (*Scalable Vector Graphics*), GeoRSS (*Geographic Really Simple Syndication*) ir daugelis kitų kalbų. Be to, kartais XML naudojama kaip šių taikomųjų programų kalbų specifikacijų kalba.

Šioje paskaitoje aptariamos kelios XML taikymo gramatikos, naudojamos erdvinių duomenų mainų kodavimui, interneto žemėlapių paslaugų protokolai ir žemėlapių apibrėžtis, interneto paslaugų naudojimas ir hibridinimas (angl. *mashuping*) į paslaugas orientuotoje architektūroje ir vektorinės grafikos atvaizdavimas kliente. Šios XML taikomųjų programų schemas gali būti naudojamos duomenų modeliavimui ir mainams įvairiose viešosiose ir komercinėse srityse – kadastro ir žemėtvarkos, eismo ir transporto, telekomunikacijų, aplinkosaugos ir kitose. Taip pat aptariamos temos, susijusios su į paslaugas orientuota architektūra, interneto žemėlapių paslaugomis bei susijusiomis technologijomis.

### Paskaitos planas

- XML kalbos
- GML kalba
- ESRI ArcXML kalba
- KML kalba
- SVG kalba
- Interneto žemėlapių paslaugos ir į paslaugas orientuota architektūra (SOA)

## 4.1. XML kalbos

Praėjusioje paskaitoje sužinojome, kad XML (*Extensible Markup Language*) kalba leidžia naudotojams patiems nustatyti žymas ir duomenis saugoti žmonėms suprantamu formatu, todėl XML naudojama kaip daugelio taikomųjų programų kalbų specifikacijų kalba. Įterpus semantinius apribojimus, XML galima realizuoti taikomųjų programų kalbas, įskaitant CML (*Chemical Markup Language*), SBML (*Systems Biology Markup Language*), RSS (*Really Simple Syndication*), GML (*Geographic Markup Language*), ArcXML (*ESRI Arc Extensible Markup Language*), KLM (*Google Keyhole Markup Language*) ir daugelį kitų. Taip pat galima atkreipti dėmesį į XML revoliuciją duomenų ir duomenų schemų kodavimo srityje.

Tarptautinė standartų organizacija (ISO, <http://www.iso.org> – anglų kalba) bei OpenGIS konsorciumas (OGC, <http://www.opengeospatial.org> – anglų kalba) taip pat rekomenduoja XML kalbą geografinės informacijos schemoms, duomenims ir interneto paslaugoms koduoti. OGC XML užklausų ir atsakymų dokumentai, apibrėžti OGC „XML Schema“ kalboje, šiuo metu naudojami geoerdviniams interneto paslaugoms (daugiau informacijos žr. 3 paskaitoje).

Todėl standarte ISO 19118:2005 „Geografinė informacija. Kodavimas“ yra nustatyti kodavimo taisyklių, naudojamų geografinių duomenų mainams, apibrėžimo reikalavimai pagal ISO 19100 standartus. Kodavimo taisyklė leidžia taikomosiose schemose ir standartinėse schemose apibrėžtą geografinę informaciją koduoti pagal nuo sistemos nepriklausomą duomenų struktūrą, tinkamą perdavimui ir saugojimui. XML kodavimo taisyklė skirta neutralių duomenų mainams ir yra pagrįsta XML bei ISO/IEC 10646 simbolių rinkinių standartais (daugiau informacijos žr. GII-03 4 paskaitoje).

Tačiau OGC „GML“ bendruomenė geoerdviniams duomenims koduoti nepriėmė ISO 19118 XML. Buvo susitarta dėl naujo ISO/TC211 darbinio pasiūlymo sukurti naują suderintą kodavimo versiją – GML (*Geography Markup Language*) ir ISO 19136.

Šioje paskaitoje bus aptariama GML kalba, kurią kaip erdvinių duomenų kodavimo ir mainų standartą rekomenduoja ISO ir OGC, taip pat kitos dvi geografinių informacinių sistemų srityje naudojamos XML kalbos (ArcXML ir KML).

## 4.2. GML kalba

GML (*Geography Markup Language*) kalbos **kodavimą** galima naudoti **perduodant** ir **saugant** geografinę informaciją, įskaitant geometriją (vektorinio formato geoobjektus) ir savybes, kurios yra objektų erdvinės ir neerdvinės savybės. GML kalbą taip pat galima naudoti kaip geografinės informacijos **modeliavimo** kalbą ir geoobjektų interneto paslaugų projektavimo priemonę.

Atkreipkite dėmesį, kad *GML naudojama geografiniam turiniui koduoti, o ne stiliui suteikti ar duomenims atvaizduoti*. Stiliui arba legendai kurti skirta XSLT (*XML Transformation Language*) kalba.

GML buvo sukurta kaip OpenGIS realizavimo specifikacija. Vėliau GML tobulino ISO/TC 211 ir 2007 m. pirmą kartą paskelbė kaip ISO 19136 standartą „Geografinė informacija. GML (*Geography Markup Language*) kalba“. GML yra pagrįsta XML technologijomis (W3C) ir palaiko XML, „XML Namespaces“, „XML Schema“ ir „Xlinks“. GML yra išplečiamoji kalba ir palaiko visų GML galimybių profilių apibrėžtis (teisingus poaibius). GML 3.2.1 versija pasirodė 2008 m. sausio mėn.

Pagrindinės GML savybės:

- palaiko geoerdvinių *taikomųjų schemų, skirtų informacijos bendruomenėms, aprašus*
- leidžia kurti ir tvarkyti *susietas* geografines taikomas schemas ir duomenų rinkinius
- palaiko taikomųjų schemų ir duomenų rinkinių *perdavimą ir saugojimą*
- išplečia organizacijų galimybes *dalytis* geografinėmis taikomosiomis schemomis ir informacija, kurią jos aprašo
- leidžia projekto vykdytojams nuspręsti, kaip naudoti GML – taikomųjų schemų ir duomenų rinkinių saugojimui GML kalba ar tik schemų ir duomenų perdavimui
- pateikia GML schemas, kurios gali būti naudojamos kaip bendros konstrukcijos ir konceptai, tinkantys pačioms įvairiausioms taikomųjų programų sritims
- apibrėžia kitų OGC plėtojimų standartinių technologijų ir protokolų pagrindą. Viena svarbi technologija yra WFS (*Web Feature Service*), užtikrinanti sąveikos su geoerdvinių duomenų baze mechanizmą naudojant GML
- siūlo „projektavimo šabloną“, kuris yra viena svarbiausių XML schemų projektavimo proceso naujųjų savybių

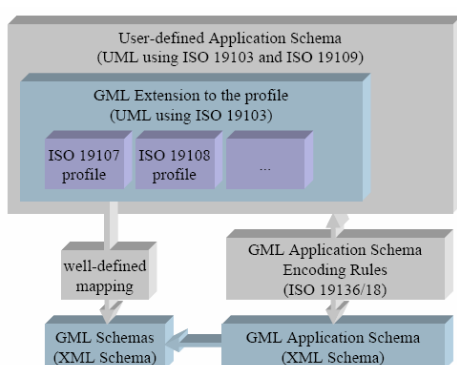
Remiantis ISO standartu 19136:2007, GML yra XML gramatika, parašyta „XML Schema“ kalba, skirta geografinėi informacijai perduoti ir saugoti bei naudotojo nustatytoms taikomosioms schemoms modeliuoti. GML yra XML kodavimo kalba, atitinkanti ISO standarto 19118 reikalavimus, skirta geografinės informacijos, sumodeliuotos pagal konceptualaus modeliavimo sistemą, perdavimui ir saugojimui, taip pat naudojama ISO 19100 serijoje (geomatikos standartai) ir OpenGIS abstrakčiųjų duomenų specifikacijoje. GML gali apdoroti geografinių objektų erdvinės ir neerdvinės savybes.

Pagrindinė ISO 19100 standartų serijos idėja, kuri vėliau buvo panaudota ir GML, yra modeliuoti pasaulį naudojant objektus, kurie yra realaus pasaulio reiškinių (pvz., kelio, upės, žmogaus, automobilio, administracinio vieneto, įvykio ir pan.) abstrakcijos. Šie objektai gali būti geografiniai arba susiję su Žemės vietovėmis. Objekto būklę apibrėžia tam tikros savybės, o kiekvieną savybę sudaro trys komponentai {vardas, tipas, reikšmė}.



Objektas gali turėti keletą savybių, įskaitant ir geometriją. Objektus galima sugrupuoti į rinkinius, kuriems taip pat galima priskirti geoobjektų tipus.

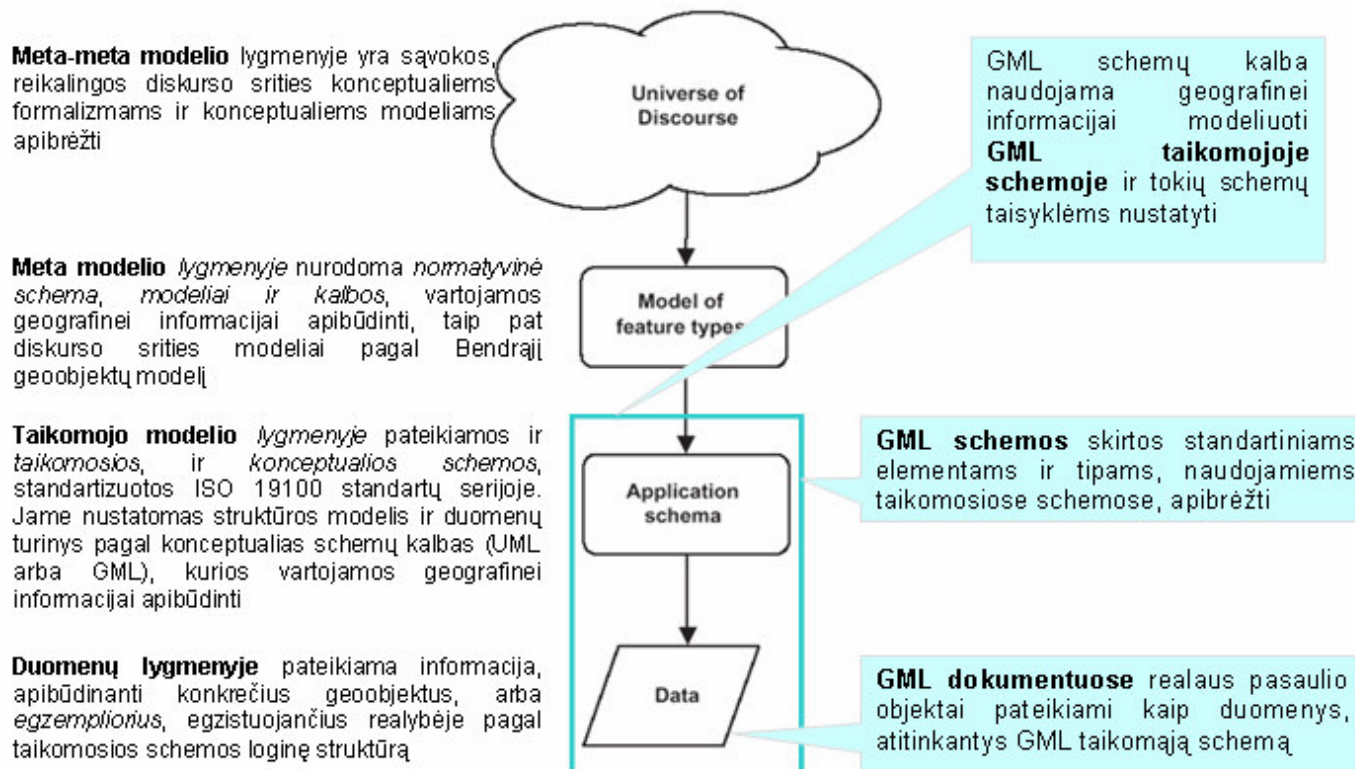
Pagrindinis ISO 19100 standartų serijos tikslas yra erdvinio pasaulio konceptualios logikos sumodeliavimas į tam tikrą konceptualią taikomąją schemą, kurioje struktūra ir duomenų turinys yra apibrėžiamas vartojant UML (*Unified Modeling Language*) kalbą. Be to, šis UML taikymo modelis gali būti koduojamas GML kalba ir naudojamas mainams tinkamų duomenų perdavimui ir saugojimui. ISO 19109 standartas „Taikomosios schemos taisyklės“ nustato taisykles, kaip kurti konkrečių objektų arba jų rinkinių taikomasias schemas laikantis kitų ISO geomatikos standartų. „XML Schema“ kalbą, numatytą ISO standarte 19136 pagal ISO standartą 19118, reikia naudoti perkeliant objekto taikomąjį modelį UML kalboje į GML kalbą. Taikomąją schemą galima sukurti ir tiesiogiai laikantis „XML Schema“ kalboje nustatytų GML taisyklių.



**Pav. UML ir „XML Schema“ susiejimas bei santykis su tarptautinių ISO standartų 19100 serija (ISO 19136)**

ISO 19109 reikalavimus atitinančios UML taikomosios schemos susiejimas su atitinkama GML taikomąja schema atliekamas remiantis nustatytomis kodavimo taisyklėmis. Schemos kodavimo taisyklės grindžiamos mintimi, kad klasės apibrėžtys taikomojoje schemoje yra susiejamos su „XML Schema“ tipų ir elementų aprašais, kad objektus egzemplioriaus modelyje būtų galima susieti su atitinkamomis elementų struktūromis XML dokumente.

Šiame paveiksle parodyta, kaip taikomosios schemos kuriamos visiškoje GML aplinkoje nenaudojant UML:



Universe of Discourse	Diskurso sritis
Model of feature types	Objektų tipų modelis
Application schema	Taikomoji schema
Data	Duomenys

**Pav. Taikomąją schemą galima sukurti ir tiesiogiai laikantis „XML Schema“ kalboje nustatytų GML taisyklių (daugiau informacijos žr. ISO 19101 „Pamatinis modelis“)**

GML modelyje galimas nesudėtingas atvaizdavimas naudojant UML profilį. Todėl geobjektai yra vaizduojami:

- UML – kaip objektai, kur geobjekto tipo vardas naudojamas kaip objektų klasės vardas
- GML egzemplioriuose – kaip XML elementai, kur geobjekto tipo vardas naudojamas kaip elemento vardas

Geobjektų savybės vaizduojamos:

- UML – susiejant funkcijas su geobjektų tipų klasėmis ir geobjektų tipų klasių požymiais, kur savybių semantiką nurodo ryšio funkcijos vardas arba požymio vardas
- GML egzemplioriuose – geobjektų paelemenčiuose (vadinamuose savybių elementais), kur savybių semantiką nurodo savybės elemento vardas.

Savybės reikšmei priskiriamas tipas, kurį nurodo:

- UML – ryšio tikslo klasė arba požymio duomenų tipas

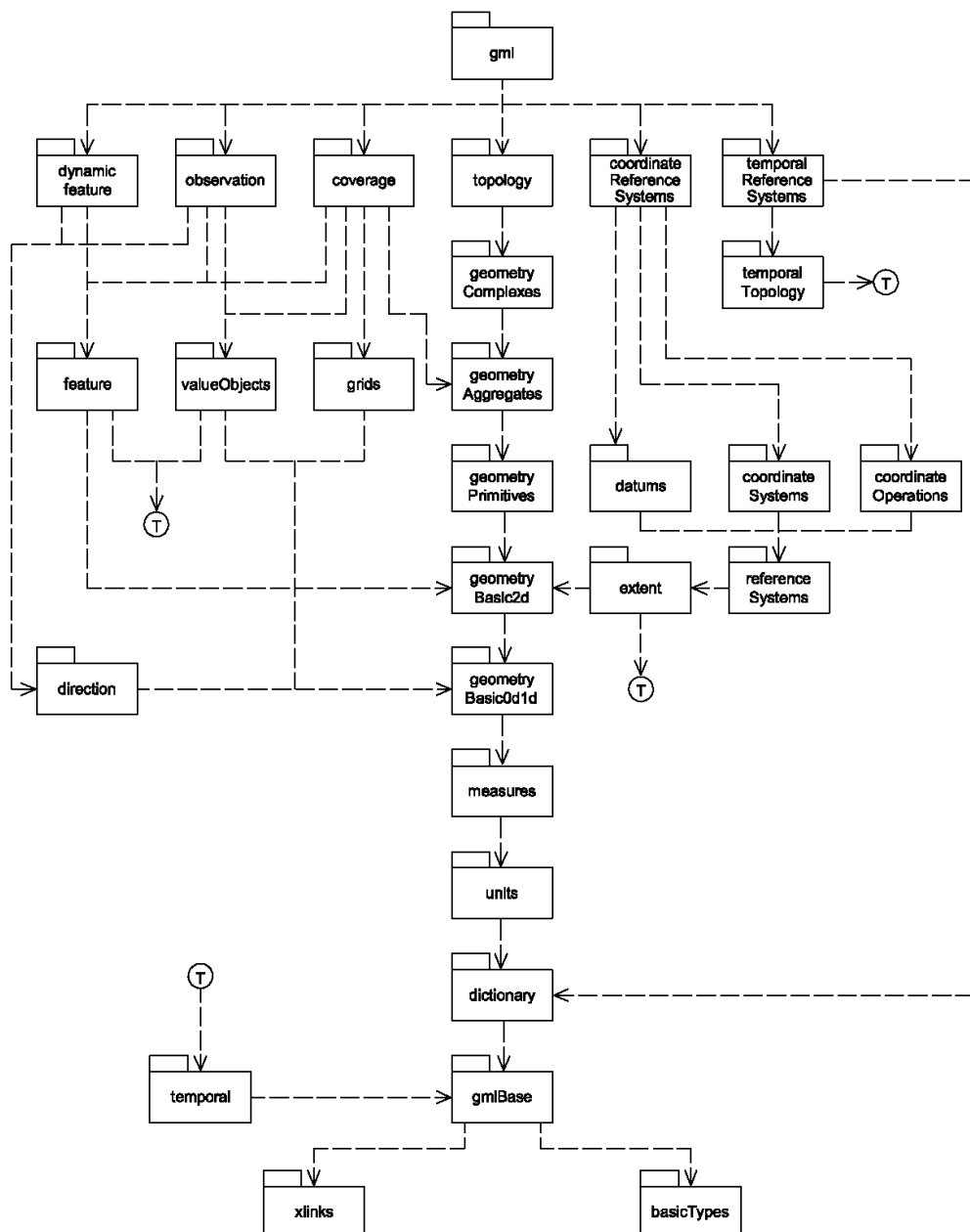
- GML – objekto elemento, esančio savybės elemente, vardas, kai savybėms priskiriamos kompleksinės reikšmės, arba tiesioginės reikšmės tipas, neturintis įstatytojo XML ženklavimo, kai savybei priskirta paprasta reikšmė.

GML įvairios esybės, pavyzdžiui, geobjektai, geometrijos, topologijos ir pan., apibrėžiamos pasitelkiant GML objektų tipų hierarchiją. Normatyvinė GML schema kuriama naudojant šiuos objektų tipus. GML taikomojoje programoje turi būti pakartotinai panaudojami pagrindiniai GML geobjektai, apibrėžti konceptualiame modelyje. Geobjektas koduojamas kaip XML elementas su geobjekto tipo vardu. Kiti atpažįstami objektai koduojami kaip XML elementai su objekto tipo vardu. Kiekvienas geobjekto požymis ir geobjekto ryšio funkcija yra geobjekto savybė. Geobjekto savybės yra koduojamos XML elemente.

Savybės elemento reikšmė gali būti užkoduota kaip paties elemento turinys arba nuoroda į jo reikšmę gali būti pateikiama kaip paprastas „XLink“ nuoroda. Savybės reikšmė gali būti paprasta arba geobjektas ar kitas sudėtingas objektas.

Todėl ISO standarte 19136 apibūdinamos normatyvinės XSD schemas, išaiškinamas jų turinys, struktūra ir priklausomybės, taip pat apibrėžiama, kaip koduoti geografinę informaciją, susijusią su:

- pagrindiniais duomenų tipais, kurie naudojami GML schemoje
- pagrindine geometrija (0d, 1d, 2d)
- papildomais geometriniais primityvais (0d, 1d, 2d, 3d)
- geometriniais kompozitais
- geometriniais agregatais
- koordinačių atskaitos sistemomis
- topologija
- laikiniais informacijos ir dinaminiais geobjektais
- apibrėžtimis ir žodynais
- vienetais, matais ir reikšmėmis
- kryptimis
- stebėjimo rezultatais
- tinkleliais ir aprėpiamais plotais
- numatytoju stiliumi



**Pav. GML schemas su priklausomybėmis (ISO 19136)**

GML schemą sudaro ISO standarte 19136 apibūdinti komponentai (XML elementai, požymiai, paprasti tipai, sudėtingi tipai, požymių grupės, grupės ir pan.).

Norėdami apibrėžti jų naudojimo sričiai tinkamus tipus, konkrečių GML taikomųjų schemų kūrėjai gali išplėsti arba apriboti GML schemoje apibrėžtus tipus. Jeigu nereikalingi jokie pakeitimai, neabstraktieji elementai, požymiai ir tipai iš GML schemas gali būti tiesiogiai naudojami taikomojoje schemoje.

Jeigu kūrėjas modeliuoja tam tikrą objekto klasę, jam bus reikalingas konkretus XML schemas dokumentas (XSD). Pavyzdžiui, jeigu modeliuojami geografiniai objektai, bus reikalingas dokumentas feature.xsd. Jeigu geobjektai turi savybių, kurioms naudojami matavimo vienetai, reikės importuoti vieną iš šių dokumentų: basicTypes.xsd, measure.xsd arba valueObjects.xsd. Jeigu naudojama erdvinė atskaitos sistema, reikės dokumento coordinateReferenceSystems.xsd. Jeigu modelio erdviniams primityvams taikoma

topologija, reikės importuoti dokumentą `topology.xsd`. Jeigu kuriami aprėpiami plotai (pvz., nuotolinio atpažinimo vaizdai, aeronuotraukos, dirvožemio išsidėstymas ir skaitmeniniai reljefo modeliai), reikės aprėpiamo ploto schemų ir pan. Kiekvienas modelis tenkina atitinkamo ISO standarto reikalavimus (pvz., GML geometrijos modelis tenkina ISO standarto 19107 „Erdvinė schema“ reikalavimus). Tame pačiame dokumente gali būti keletas schemų. GML schemų dokumentai pateikiami internete.

Panagrinėkime keletą GML XSD schemų elementų ir atitinkamus XML duomenų kodavimo rezultatus.

Elementas `gml:AbstractFeature` aprašomas taip:

```
<element name="AbstractFeature" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="gml:AbstractGML"/>
```

`gml:AbstractFeature` galima laikyti bet kuo, kas yra GML geoobjektas, ir jį galima naudoti kintamiesiems arba šablonams, kuriuose GML savybės reikšmė yra koks nors geoobjektas, apibrėžti.

Pagrindinį geoobjektų modelį nurodo `gml:AbstractFeatureType`, kuris schemoje apibrėžiamas taip:

```
<complexType name="AbstractFeatureType" abstract="true">
<complexContent>
<extension base="gml:AbstractGMLType">
<sequence>
<element ref="gml:boundedBy" minOccurs="0"/>
<element ref="gml:location" minOccurs="0"/>
</sequence>
</extension>
</complexContent>
</complexType>
```

`gml:AbstractFeatureType` turinio modelis `gml:AbstractGMLType` apibrėžtą turinio modelį papildo dviem specialiomis savybėmis, tinkamomis geografiniams objektams. `gml:boundedBy` savybės reikšmė apibūdina voka, kuris apima visą geoobjekto egzempliorių, ir ji visų pirma naudinga palaikant greitą geoobjektų, esančių konkrečioje vietoje, paiešką. Savybės `gml:location` reikšmė apibūdina geoobjekto mastą, padėtį arba santykinę vietą.

Savybės `gml:locationName` elementas yra patogumo savybė, kurios tekstinė reikšmė apibūdina geoobjekto buvimo vietą. Šis elementas apibrėžiamas taip:

```
<element name="locationName" type="gml:CodeType"/>
```

Pavyzdys, kaip nurodytos vietos duomenų egzempliorius gali atrodyti tekstinėje eilutėje:

```
<Feature>
  <gml:locationName>Klaipeda town of residence</gml:locationName>
</Feature>
```

Savybės `gml:locationReference` elementas yra patogumo savybė, kurios tekstinė reikšmė, žymima požymiu `xlink:href`, apibūdina geoobjekto buvimo vietą. Šis elementas apibrėžiamas taip:

```
<element name="locationReference" type="gml:ReferenceType"/>
```

Toliau pateikiamas pavyzdys, kaip gali atrodyti kitos paslaugos nurodytos vietos duomenų egzempliorius:

```
<Feature>
  <gml:locationReference
xlink:href="http://www.somesite.lt/bin/gazm01?placename=kliapeda&placetype=R&pro
vince=PA+"/>
</Feature>
```

Remiantis GML 3.2 specifikacijomis, GML taikomiosiose schemose reikia pasirinkti taikomajai programai būdingus geometrinių savybių vardus. Savybių vardai turi būti pasirenkami taip, kad jie išreikštų atitinkamų reikšmių semantiką. Savybėms, įskaitant ir geometrines savybes, pavadinti dažniausia pasirenkami konkrečiai taikomajai programai tinkami vardai.

Pavyzdžiui, išorinei taikomajai programai skirto geoobjekto tipo `<ext:Tower>` vardas gali būti *vieta*, kai gražinama taškų geometrija jo vietai taške nustatyti. Geoobjekto `Tower` tipą naudotojas gali apibrėžti naudodamas tokią schemą:

```
<element name="Tower" type="ext:TowerType" substitutionGroup="gml:_Feature"/>
<complexType name="TowerType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element ref="gml:pointProperty"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

GML schemoje yra iš anksto apibrėžti savybių tipai (pvz., tipas `PointProperty`), kuriuos galima naudoti kaip geometrinių savybių elementų tipus.

Pavyzdyje parodyta, kaip gali atrodyti egzemplioriaus `Tower` duomenų egzempliorius:

```
<ext:Tower>
<gml:pointProperty>
  <gml:Point srsName="urn:x-ogc:def:crs:EPSG::4326">
    <gml:pos>2478112.85645389 6695151.20645945</gml:pos>
  </gml:Point>
</gml:pointProperty>
</ext:Tower>
```

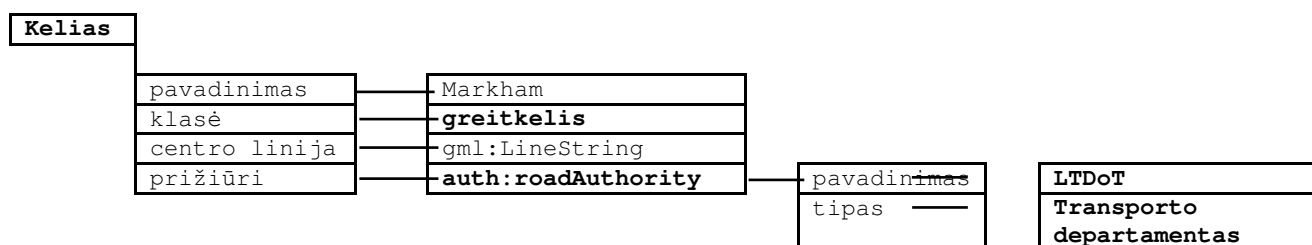
Šiame pavyzdyje naudojamas GML geometrinėse schemose apibrėžto tipo `PointProperty` objektas `gml:Point`. Tą patį savybės elementą galima naudoti vietai pažymėti pateikiant nuorodą:

```
<ext:Tower xlink:href="http://my.somesite.org/locations/point96"/>
```

Čia `http://my.somesite.org/locations/point96` yra taškas (elementas `gml:Point`), kurį pateikia nurodyta paslauga. Objektas yra savybės antrinis elementas (angl. *child element*) arba apibrėžiamas požymio `xlink:href` savybės elemente. Požymis `xlink:href` suprantamas taip: savybės reikšmė yra nuorodoje nurodytas objektas. Objektas gali būti tame pačiame GML dokumente arba bet kur internete arba intranete.

Bet kurį geometrinį elementą, kuriam perduodama `gml:AbstractGeometryType` semantika, galima laikyti tiesioginių padėčių (angl. *direct positions*) rinkiniu `<gml:pos>`. Tiesioginių padėčių egzemplioriuose yra X ir Y koordinatės (pvz., `<gml:pos>2478112.85645389 6695151.20645945</gml:pos>`), nurodančios padėtį tam tikroje koordinatinių atskaitos sistemoje (KAS) (pvz., KAS 4326, kurią apibrėžia `urn:x-ogc:def:crs:EPSG::4326`). Visos klasės, sukurtos pagal `gml:AbstractGeometryType`, iš požymių grupės `gml:SRSReferenceGroup` požymio `srsName` perima neprivalomą ryšį su koordinatinių atskaitos sistema. Visos tiesioginės padėtytys turi turėti tiesioginį arba netiesioginį ryšį su koordinatinių atskaitos sistema per požymį `srsName`.

Panagrinėkime kelio objekto su požymiais, kurio struktūra apibrėžta pateiktame paveiksle, pavyzdį:



Pav. Geoobjekto `Kelias` taikomoji schema

Pavyzdžiui, naudotojai gali taip apibrėžti modeliuojamą kelio tiesinio objekto tipą `Road` su požymiais `road_name`, `class`, ir `maintainer`, kuris skirtas ne GML priklausančiai taikomajai programai:

```
<element name="Road" type="ext:RoadType" substitutionGroup="gml:_Feature"/>
<complexType name="RoadType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="road_name" minOccurs="0">
          <simpleType>
            <restriction base="string">
              <maxLength value="50"/>
            </restriction>
          </simpleType>
        </element>
        <element name="class" minOccurs="0" type="string"/>
        <element ref="auth:roadAuthority"/>
        <element ref="gml:centerLine" minOccurs="0"/>
        <element ref="gml:multiCurveProperty" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Pavyzdyje parodyta, kaip gali atrodyti egzemplioriaus `Road` duomenų egzempliorius:

```
<ext:Road gml:id="id6df7af28-9cf1-4f0c-8858-7eaed738b544">
  <ext:name>Markham</ext:name>
  <ext:class>Highway</ext:class>
  <auth:maintainer gml:id="o.1f32a3">
    <name>LTDoT</name>
    <type>DepartmentOfTransportaion</type>
  </auth:maintainer>
  <gml:centerLine>
    <gml:LineString srsName="urn:x-ogc:def:crs:EPSG::4326">
      <gml:posList>2233796.19000493 6508921.45806473 2379332.41231563
6536523.15539952 2477192.97559352 6591726.5500691 2554979.57717337
6619328.24740389 2670404.85693703 6654457.68037544 2815941.07924774
6659476.17079994 2956458.81113393 6659476.17079994</gml:posList>
    </gml:LineString>
  </gml:centerLine>
</ext:Road>
```

Elementas `gml:AbstractCurve` yra abstraktus pakeitimo grupės (angl. *substitution group*) pagrindinis elementas visiems nenutrūkstamos kreivės elementams. Savybė `curveProperty` yra iš anksto nustatyta savybė, kuri gali būti naudojama GML taikomosiose schemose, jeigu GML geobjektas turi savybę su reikšme, kurią gali pakeisti `gml:AbstractCurve`. `gml:LineString` yra speciali kreivė, sudaryta iš vienintelio segmento su tiesine interpoliacija. Ją apibrėžia du arba daugiau koordinatų kortežai, tarp kurių yra tiesinė interpoliacija.



Kitame pavyzdyje parodyta, kaip gali atrodyti išoriniam tipui `ext:School`, kurį atvaizduoja daugiakampis GML elementas, skirtas duomenų egzempliorius:

```
<ext:School gml:id="idc8feb418-6934-42d6-ab01-436b49e8e111">
  <gml:surfaceProperty>
    <gml:Surface srsName="urn:x-ogc:def:crs:EPSG::4326">
      <gml:patches>
        <gml:PolygonPatch>
          <gml:exterior>
            <gml:LinearRing>
              <gml:posList>2522359.38941408 6516449.19370149 2522359.38941408
6544050.89103628 2562507.31281013 6544050.89103628 2562507.31281013
6516449.19370149 2522359.38941408 6516449.19370149</gml:posList>
            </gml:LinearRing>
          </gml:exterior>
        </gml:PolygonPatch>
      </gml:patches>
    </gml:Surface>
  </gml:surfaceProperty>
</ext:School>
```

#### Pav. Daugiakampis geobjektas GML dokumente

`gml:Polygon` yra ypatingas paviršius, kurį apibrėžia vienas paviršiaus fragmentas `<gml:PolygonPatch>`. Šio fragmento riba yra vienoje plokštumoje, o poligono viduje naudojama plokštuminė interpoliacija. Elementai `gml:exterior` ir `gml:interior` apibūdina poligono paviršiaus ribą. Paviršiaus ribą sudaro keletas apskritimų. Vienas iš šių įprasto dvimačio objekto apskritimų laikomas išorine riba. `gml:LinearRing` apibrėžia keturi arba daugiau koordinatinių kortezai, tarp kurių yra tiesinė interpoliacija; pirmoji ir paskutinė koordinatės turi sutapti. Elementas `gml:posList` leidžia kompaktiškai nurodyti padėčių koordinates, jeigu visos padėtys yra toje pačioje koordinatinių atskaitos sistemoje.

GML modeliavimas ir kodavimas padeda programinės įrangos kūrėjams rengti conceptualius taikomųjų schemų projektus. Taikomųjų schemų kūrėjai gali naudotis taikomųjų schemų taisyklėmis ir jomis vadovautis modeliuodami duomenų struktūras. Programinės įrangos kūrėjai gali naudotis iš anksto parengta GML specifikacija kaip darbo su XML dokumentais ir jų apdorojimo gairėmis.

ESRI plėtinys „ArcGIS Data Interoperability“ palaiko ESRI formatų eksportą ir importą į GML naudojant formatą „Simple Feature“.

### 4.3. ESRI ArcXML kalba

Kitaip nei GML, kurioje akcentuojamas turinys, ArcXML (**Arc Extensible Markup Language**) kalba naudojama ArcIMS **žemėlapių sąrankos failams**, **užklausų** ir **atsakymų** pranešimų turiniui bei ESRI ArcIMS paslaugos **protokolui** apibrėžti. ArcXML yra išvestinė XML kalbos, SGML kalbos poaibio, kalba.

ArcXML kalboje naudojama hierarchinė žymų ir jų poaibių sistema. ArcXML kalboje yra penkios pagrindinės žymos (angl. *root tags*):

- <ARCXML> naudojama kiekviename ArcXML dokumente ir aprašo, koks jis yra
- <CONFIG> naudojama konfigūruojant žemėlapių paslaugą ir nustatant, kurie duomenys bus įtraukti bei kaip ji atrodo
- <REQUEST> naudojama kliento užklausiai, siunčiamai serveriui
- <REQUEST> naudojama atsakymui, kurį erdvinių duomenų serveris siunčia klientui
- <MARKUP> naudojama pakeitimams, kurie padaryti kliente, aprašyti.

Visi ArcXML sakiniai kuriami naudojant elementus ir požymius, kurie sudaro hierarchinę struktūrą. Šiai struktūrai priklausantys elementai yra skirstomi į pirminius elementus (angl. *parent elements*) ir antrinius elementus (angl. *child elements*). Antriniai elementai yra įstatyti pirminiuose elementuose. Dauguma elementų turi vieną arba daugiau požymių. Požymiai – tai vardų-reikšmių poros, kurios aprašo elementą. Kai kuriems ArcXML elementams požymiai yra būtini. Jeigu žemėlapių sąrankos faile elemento požymis nėra nurodytas, naudojama numatytoji reikšmė, jeigu požymis turi numatytąją reikšmę. ArcXML failų struktūra atitinka W3C XML rekomendacijas.

Rašant elementus ir požymius būtina laikytis griežtų taisyklių, kitaip ArcIMS negalės sukurti žemėlapių paslaugos. Elementai, nustatantys aprašomo turinio pobūdį, turi būti rašomi didžiosiomis raidėmis, turėti tinkamus pradžios ir pabaigos ženklus – mažiau nei (<) ir daugiau nei (>). Požymių vardai turi būti rašomi mažosiomis raidėmis, o jiems priskiriamos reikšmės turi būti įtrauktos į dvigubas kabutes. Vienu metu galima apibrėžti tik vieną požymio reikšmę.

**Žemėlapių sąrankos failai** yra naudojami kaip pradiniai ArcIMS paslaugų duomenys. Tokiame faile yra informacija apie žemėlapių savybes ir nurodymai, kaip pateikti žemėlapius kaip ESRI „Image“ (WMS) ir „Feature“ (WFS) paslaugas. Šiame faile yra elementai, kurie apibrėžia vietinę aplinką (kalbą ir pan.) ir žemėlapių elementus (žemėlapių savybes, darbo erdves ir sluoksnius).

Žemėlapių sąrankos failas yra tekstinis failas su .axl plėtiniu, turintis pagrindinę žymą <CONFIG>. Žemėlapių sąrankos failo struktūra yra tokia:

```
<?xml version="1.0" encoding="UTF-8"?><ARXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>...</ENVIRONMENT>
    <MAP>
      <PROPERTIES>...</PROPERTIES>
      <WORKSPACES>...</WORKSPACES>
      <LAYER>...</LAYER>
    </MAP>
  </CONFIG>
</ARXML>
```

- Sekcijoje `PROPERTIES` nurodoma pradinė žemėlapių apimtis, žemėlapių vienetai ir dabartinė projekcija, taip pat papildomi nurodymai „Image Services“ paslaugai
- Sekcijoje `WORKSPACES` nurodyta visų duomenų, kurie naudojami žemėlapių sluoksniams kurti, vieta
- Elementas `LAYER` naudojamas visiems žemėlapių sluoksniams, jame pateikiama informacija apie tai, kaip turi būti vaizduojami duomenys.

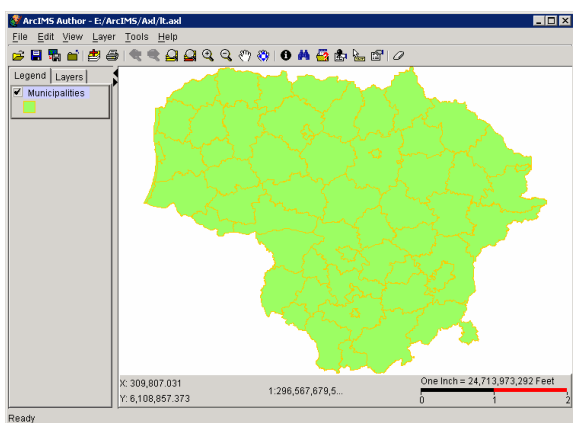
Taigi matome, kad ESRI `.axl` faile yra ne faktinių duomenų turinys, o tik nurodomas kelias į naudojamus duomenis (pvz., kelias iki *shape* failų, vaizdų, ArcSDE duomenų bazių). Todėl ArcXML kalbos paskirtis yra visiškai kita nei GML.

Toliau pateiktame žemėlapių sąrankos failo pavyzdyje naudojami pirmiau aptarti elementai ir kai kurie papildomi antriniai elementai, apibūdinti ArcXML programuotojo vadove (<http://edndoc.esri.com/arcims/9.2/> – anglų kalba). Antrinių elementų `</MAP>` paskirtis – nurodyti duomenų šaltinį ir nurodyti, kaip pateikti duomenis.

```
?xml version="1.0" encoding="UTF-8"?>

<ARXML version="1.1">
  <CONFIG>
    <ENVIRONMENT>
      <LOCALE country="US" language="en" variant="" />
      <UIFONT color="0,0,0" name="SansSerif" size="12" style="regular" />
      <SCREEN dpi="96" />
    </ENVIRONMENT>
    <MAP>
      <PROPERTIES>
        <ENVELOPE minx="306455.4687500001" miny="5973368.499999999"
maxx="680107.6249999998" maxy="6257754.0" name="Initial_Extent" />
        <MAPUNITS units="meters" />
      </PROPERTIES>
      <WORKSPACES>
        <SHAPEWORKSPACE name="shp_ws-0" directory="E:\ArcIMS\Data\LT" />
      </WORKSPACES>
      <LAYER type="featureclass" name="Municipalities" visible="true" id="0">
        <DATASET name="savivaldybes" type="polygon" workspace="shp_ws-0" />
        <SIMPLERENDERER>
          <SIMPLEPOLYGONSYMBOL boundarytransparency="1.0" filltransparency="1.0"
fillcolor="153,255,102" boundarycuptype="round" boundarycolor="255,200,0" />
        </SIMPLERENDERER>
      </LAYER>
    </MAP>
  </CONFIG>
</ARXML>
```

Šiame .axl faile numatyta, kad ESRI interneto žemėlapių paslauga naudoja daugiakampio formos failą savivaldybes, esantį aplanke E:\ArcIMS\Data\LT, vienam sluoksniui Municipalities pateikti naudojant stilius boundarytransparency="1.0", filltransparency="1.0", fillcolor="153,255,102" boundarycuptype="round" ir boundarycolor="255,200,0" SIMPLEPOLYGONSYMBOL. Paveiksle parodyta, kaip tai atrodo.



**Pav. Failui .axl redaguoti galima naudoti „ArcIMS Author“.**

Failą .axl taip pat galima naudoti konkrečių sluoksnių projekcijoms apibrėžti (pvz., <COORDSYS id="54030"), DBF failams sujungti su žymomis <SPATIALQUERY joinexpression ...>, papildomoms vaizdavimo ir žymėjimo parinktimis priskirti, duomenų rinkinius taikyti objektų ženklavimui ir pan.

ArcXML kalba taip pat naudojama kliento ir ESRI interneto žemėlapių serverių, verslo logikos pakopos ir ArcIMS serverių ryšio protokolams. ArcXML taip pat naudojama kaip protokolas, skirtas bendravimui su „ArcIMS Spatial Server“. ArcIMS kliento ir serverio komponentų bendravimas išsamiau paašškintas 5 paskaitoje.

ArcXML apibrėžia toliau išvardytų klientų *užklausų* turinį:

- GET\_IMAGE – prašymas ArcIMS arba „ArcGIS Server“ sukurti žemėlapių vaizdą
- GET\_FEATURES – prašoma, kad „ArcIMS Feature Server“ geobjektus pateiktų kaip suglaudintą dvejetainį srautą ESRI formatu, arba prašoma, kad „ArcIMS Query“ arba „ArcMap Server“ pateiktų požymių duomenis ArcXML formatu
- GET\_GEOCODE – prašymas, kuriame yra adreso informacija, skirta „Geocode Server“
- GET\_EXTRACT – prašymas „Extract Server“ turinio sluoksnius ištraukti „shapefile“ formatu, sudėti į suspaustą (zip) failą ir išsiųsti klientui
- GET\_SERVICE\_INFO – prašoma pateikti informacijos apie kiekvieną ArcIMS paslaugos sluoksnį. Kai naudojamos vaizdų ir geobjektų paslaugos „Image“ ir „Feature“, atsakant į užklausą gali būti teikiama informacija apie laukus, vokus, plėtinius arba vaizdavimo priemones (angl. *renderers*). Kai naudojama paslauga „ArcMap Image“, atsakant į užklausą gali būti teikiama informacija apie laukus, vokus, duomenų rėmelius ir turinį.
- GET\_RASTER\_INFO – prašymas nurodyti paveikslo vaizdo taškų reikšmę konkrečioje x, y koordinatų vietoje nurodytame sluoksnyje
- GET\_LAYOUT – prašymas pateikti ArcMap išdėstymą
- GET\_PROJECT – prašymas perprojektuoti geometriją iš vienos koordinatų sistemos į kitą
- GET\_METADATA – prašymas siųsti užklausas ESRI paslaugai „Metadata Service“
- PUBLISH\_METADATA – prašymas administruoti ir publikuoti metaduomenis

Kiekvieną ArcXML *užklausos* tipą atitinka „ArcIMS Spatial Server“ *atsakymo* tipas:

- atsakymas į GET\_EXTRACT yra EXTRACT
- atsakymas į GET\_FEATURES yra FEATURES
- atsakymas į GET\_GEOCODE yra GEOCODE
- atsakymas į GET\_IMAGE yra IMAGE
- atsakymas į GET\_LAYOUT yra LAYOUT
- atsakymas į GET\_METADATA yra METADATA
- atsakymas į GET\_PROJECT yra PROJECT
- atsakymas į GET\_RASTER\_INFO yra RASTER\_INFO
- atsakymas į GET\_SERVICE\_INFO yra SERVICEINFO
- atsakymas į PUBLISH\_METADATA yra METADATA\_ACTION

Užklauso <GET\_IMAGE> pavyzdys:

```
<ARXML version="1.1">
<REQUEST>
<GET_IMAGE>
<PROPERTIES>
<ENVELOPE minx="429196.520177882" miny="5445070.86354896"
maxx="429996.733754915" maxy="5445803.33475179" />
<IMAGE_SIZE width="500" height="350" />
</PROPERTIES>
</GET_IMAGE>
</REQUEST>
</ARXML>
```

ArcIMS atsakymas, kuris vietos dalį gražina į žemėlapių vaizdą (<http://tree.mala.bc.ca/output/Municipalities.jpg>) ir vizualizuoja paveikslą [Municipalities.jpg](http://tree.mala.bc.ca/output/Municipalities.jpg) kliento interneto naršyklėje, galėtų būti toks:

```
<?xml version="1.0" encoding="UTF-8"?>
<ARXML version="1.1">
<RESPONSE>
<IMAGE>
<ENVELOPE minx="429073.433250091" miny="5445070.86354896"
maxx="430119.820682706" maxy="5445803.33475179" />
<OUTPUT url="http://tree.mala.bc.ca/output/Municipalities.jpg" />
</IMAGE>
</RESPONSE>
</ARXML>
```

Be to, ArcIMS ir „ArcGIS Server“ palaiko OGC WMS ir WSF serverius per jungtis ESRI „WMS Connector“ ir „WFS Connector“. Esant tokiems scenarijams, kliento ir serverio ryšio protokolas palaiko toliau išvardytas OGC XML POST užklauso ir atsakymus:

- **GetCapabilities** – prašymas teikia informaciją apie WMS arba WFS paslaugą
- **GetMap** – prašymas suranda žemėlapių WMS svetainėje
- **GetFeatureInfo** – prašymas teikia geobjekto informaciją atpažindamas žemėlapių tašką pagal vaizdo taškų vietą
- **DescribeFeatureType** – prašymas sukuria schemas aprašą geobjektų tipams, kurie naudojami realizuojant WFS
- **GetFeature** – prašymas suranda geobjektus paslaugoje „Web Feature Service“. Atsakymas **GetFeature** gražina geometrinius duomenis, užkoduotus GML kalba

Su ArcXML 1.1 galima naudoti ArcXML duomenų tipo apibrėžtį (*Data Type Definition*, DTD). Tokia apibrėžtis ArcXML kalba apibrėžia elementų ir požymių struktūros taisyklės, patikrina, ar dokumentas yra teisingas. Jeigu kūrėjas nori atlikti ArcXML patikrą, jis gali naudoti <http://support.esri.com/index.cfm?fa=downloads.samplesUtilities.viewSample&PID=16&etalD=1242>.

Šiai DTD tinka daugelis XML rengyklių, pavyzdžiui, „XML Spy“ (<http://www.xmlspy.com>) arba „Xeena“ (<http://alphaworks.ibm.com/tech/xeena>). Kai naudojama DTD su ArcXML, elementų vardai, požymiai ir išvardytos požymių reikšmės turi būti rašomos tokiomis raidėmis (didžiosiomis ar mažosiomis), kurios nurodytos DTD. ArcXML elementų vardai visada rašomi didžiosiomis raidėmis. Požymiai – visada mažosiomis. Išvardytos požymių reikšmės dažniausia rašomos mažosiomis raidėmis.

Kai žemėlapių sąrankos faile, užklausoje arba atsakyme naudojamas sakiny `DOCTYPE`, jis turi būti antras po XML aprašo eilutės. Sakinyje `DOCTYPE` yra informacija apie DTD vietą. Pavyzdžiui:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE ARCXML SYSTEM "<dtd_location>/arcxml.dtd">  
<ARCXML version="1.1">  
...  
</ARCXML>
```

Kitoje paskaitoje išsamiau susipažinsime su `.axl` ir ArcXML ryšio funkcijomis.

#### 4.4. KML kalba

KML (*Google Keyhole Markup Language*) kalba yra XML ženklavimo kalba, skirta interneto žemėlapių paslaugai „Google Maps“ vartoti ir papildomiems naudotojo nustatytiems objektams atvaizduoti „Google Earth“, „Google Maps“, „Google Maps for mobile“ ir kitose programose, įskaitant ESRI „ArcGIS Explorer“, NASA „WorldWind“, „Adobe Photoshop“, AutoCAD, „Flickr“ ir „Yahoo Pipes“.

KML buvo sukurta specialiai „Google Earth“. Kadangi šis projektas iš pradžių buvo vadinamas „Keyhole“ (kaip ir bendrovė, kuri ėmėsi šio darbo), susijusi ženklavimo kalba buvo pavadinta atitinkamai. Kai 2004 m. „Google“ įsigijo „Keyhole“, ji įgijo teises į projektą ir vėliau jį pervadino „Google Earth“.

„Google Maps“ – tai žemėlapių paslauga, kuri kurdama interaktyvius interneto žemėlapius apjungia kelias skirtingas žemėlapių technologijas ir kitas „Google“ paslaugas. Interneto „Google Maps“ paslauga gali būti laikoma paslaugos teikėju paslaugos prašytojui *į paslaugas orientuotoje architektūroje (SOA)*. Interneto paslaugos – tai eilė protokolų, leidžiančių publikuoti, atrasti ir naudoti paslaugas nuo technologijų nepriklausančiu standartiniu formatu. SOA – tai verslo taikomųjų programų, kurios naudoja bendras paslaugas verslo funkcijoms palaikyti, kūrimo metodas, aptariamas paskutinėje šios paskaitos dalyje. Po to, kai interneto paslauga publikuojama į paslaugas orientuotoje architektūroje, ja gali naudotis įvairios kliento taikomosios programos ir procesai.

Pavyzdžiui, paslaugos prašytojas naudoja KML kalbą „Google Maps“ interneto žemėlapių paslaugai iškviešti ir gautame žemėlapyje *hibridina* papildomus naudotojo apibrėžtus objektus. Terminas „hibridinimas“ naudojamas apibūdinant interneto taikomas programas, kurios iš kelių skirtingų šaltinių gautus duomenis ir turinį integruoja į bendrą vaizdinį. Kaip pavyzdį galima paminėti tokį duomenų hibridinimą, kai kartografiniai arba nuotolinių tyrimų duomenys iš „Google Maps“ papildomi nekilnojamojo turto objektų vietos informacija, paimta iš internete esančio nekilnojamojo turto objektų sąrašo. Taip sukuriama naujos interneto paslaugos, kurių neteikia nė vienas iš šių šaltinių.

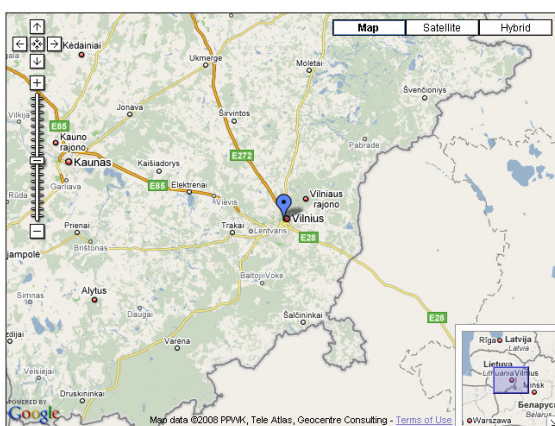
Panašiai kaip GML ir ArcXML, KML kalboje, sukurtoje pagal XML standartą, naudojama žymų struktūra su įdėtiniais elementais ir požymiais. Visos žymų raidės turi būti tokios pačios (didžiosios arba mažosios) kaip nurodyta KML 2.2 žinyne. Žinyne nurodyta, kurios žymos yra neprivalomos. Konkrečiame elemente žymos turi būti rašomos tokia tvarka, kuri nurodyta žinyne ([http://code.google.com/apis/kml/documentation/kml\\_tags\\_beta1.html](http://code.google.com/apis/kml/documentation/kml_tags_beta1.html) – anglų kalba).

KML dvimatėse „Google Maps“ arba trimatėse „Google Earth“ interneto taikomoseiose programose leidžia papildomai įterpti anotacijas ir persidengimo ypatybes. KML faile nurodomos įvairių objektų vaizdavimo specifikacijos. KML objektų rinkiniui priskiriamos vietos žymės (angl. *placemarks*), tekstiniai aprašai, paveikslai, antžeminės perdangos, keliai, poligonai ir pan. (daugiau žr. [http://code.google.com/apis/kml/documentation/kml\\_tags\\_beta1.html](http://code.google.com/apis/kml/documentation/kml_tags_beta1.html) – anglų kalba). Kiekvienai vietai priskiriami ilgumos ir platumos duomenys (KML palaiko tik WGS84 sistemos geografines koordinates ir altitudes, t. y. aukštį virš žemės paviršiaus metrais) bei su rodymu susiję duomenys – antraštė, altitudė, o pokrypį galima numatyti norint apibrėžti vadinamąjį geoerdvinių duomenų „kameros rodinį“. KML kalboje kai kurios geometrijos gramatikos taisyklės atitinka GML taisykles.



Pateikiamas objekto „point“ vietos žymės „placemark“ su susijusia geometrija pavyzdys:

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.0">
  <Placemark>
    <description> Vilnius City</description>
    <name>Vilnius</name>
    <Point>
      <coordinates>25.258197, 54.680445, 0</coordinates>
    </Point>
  </Placemark>
</kml>
```



**Pav. Taško vietos žymė atitinka piktogramą, kuri žymi Žemės tašką. Norėdami šį kodą peržiūrėti programoje „Google Map“, eikite: [http://www.thechrisoshow.com/display\\_kml/](http://www.thechrisoshow.com/display_kml/).**

KML dokumentai įrašomi kaip tekstiniai failai su plėtiniais .kml arba .kmz. KML dokumentai gali būti platinami kaip .kmz failai, t. y. suspausti KML dokumentai. Paprastai .kmz faile yra vienas KML dokumentas (pavadintas „doc.kml“) bei perdangų paveikslai ir piktogramos, kurias gali naudoti dokumentas. KML failus galima kurti naudojant „Google Earth“ vartotojo sąsają, XML arba tekstų rengyklę.

Dabartinė KML specifikacija yra 2.2. Ji buvo perduota OpenGIS konsorciui, kuris turėtų paruošti atvirą standartą, tinkamą visoms geografinėms naršyklėms. Kadangi konsorciui jau atsakingas už GML standartizavimą, toks žingsnis buvo protingas.

KML kalba plačiai naudojama daugelyje taikomųjų programų, todėl svarbu užtikrinti, kad KML failai atitiktų KML standartus. KML patikrą galima atlikti dviem būdais: naudoti internete pateikiamas patikros priemones, pavyzdžiui, „Feedvalidator“ ([feedvalidator.org](http://feedvalidator.org)), arba atjungties būsenos patikrą atlikti su XML KML schema (<http://code.google.com/apis/kml/schema/kml21.xsd>) naudojant XML analizatoriaus rengyklę, kurios leidžia atlikti schemų patikrą (pvz., „Xerces“).

## 4.5. SVG kalba

SVG (Scalable Vector Graphics) kalba yra XML gramatikos kalba, naudojama dvimatei keičiamo stiliaus vektorinei grafikai rodyti su tinkamomis XML vardų erdvėmis. SVG 1.1 yra W3C rekomendacija, suderinama su XML 1.0 rekomendacija, ir laikoma šiuo metu kuriamų SVG variantų pagrindu. SVG formatu galima aprašyti ir statinę, ir animacinę vektorinę grafiką. SVG gali būti tik deklaratyvi arba su skriptais. Paveikslai gali būti su hipersaitais, kuriems naudojami išsiunčiamieji paprasti XLink nuorodos.

Keičiamo dydžio reiškia, kad galima didinti arba mažinti vienodumą. Keičiamo dydžio grafika yra tokia, kurios vaizdo taškų dydis gali kisti. SVG grafikos dydį galima keisti pritaikant skirtingoms ekrano skiriamosioms geboms. Ta pati SVG grafika gali būti pateikiama skirtingais dydžiais tame pačiame tinklalapyje, o skirtinguose puslapiuose galima pakartotinai naudoti jos skirtingus dydžius. SVG grafiką galima išdidinti iki labai stambaus mastelio. Kitaip nei taškinės grafikos (JPEG, GIF, PNG), SVG grafikos mastelį galima didinti arba mažinti pagal poreikį.

Vektorinėje grafikoje pateikiami geometriniai objektai, pvz., tiesės ir kreivės. Todėl šis formatas užtikrina daugiau lankstumo nei rastriniai formatai (PNG arba JPEG), kurie turi saugoti informaciją apie kiekvieną grafikos vaizdo tašką. Paprastai vektoriniuose formatuose galima integruoti rastrinius vaizdus ir juos derinti su vektorine informacija. SVG turi tokių pačių galimybių.

SVG taip pat skirta naudoti kaip vienas komponentas daugybinių vardų erdvių XML programoje. Taip išplečiamos kiekvienos naudojamos vardų erdvės galimybės, galima kurti naujovišką turinį. Pavyzdžiui, SVG grafiką galima įtraukti į dokumentą, kuriame naudojama į tekstą orientuota EXL vardų erdvė, įskaitant ir XHTML. SVG yra geras, plačios paskirties komponentas, tinkamas bet kuriai daugybinių vardų erdvių gramatikai, kuri turi naudoti grafiką.

SVG galimi trijų rūšių grafiniai objektai:

- vektorinė grafika
- rastrinė grafika
- tekstas

Kadangi naudojama XML kalba, SVG failas yra paprastas tekstinis failas, kurį galima peržiūrėti ir redaguoti kaip bet kurį kitą ženklimą. Pavyzdžiui, toliau pateiktas SVG ženklimas tinkamoje žiūryklėje atrodo kaip žvaigždė ★ :

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="12cm" height="4cm" viewBox="0 0 1200 400" version="1.1"
xmlns="http://www.w3.org/2000/svg" preserveAspectRatio="xMidYMid meet"
zoomAndPan="magnify">
  <desc>Example - star </desc>
  <!-- Show outline of canvas using 'rect' element -->
  <rect x="1" y="1" width="1198" height="398" fill="none" stroke="blue"
stroke-width="2"/>
  <polygon fill="red" stroke="blue" stroke-width="10" points="350,75 379,161
469,161 397,215 423,301 350,250 277,301 303,215 231,161 321,161"/>
</svg>
```

Šiame pavyzdyje žyma `<polygon>` naudojama kuriant grafinį vaizdą, turintį ne mažiau kaip tris kraštus. SVG failus reikia įrašyti su plėtiniu `.svg`, tačiau SVG failai gali būti įstatomi į HTML puslapius.

ESRI ArcGIS palaiko savo vektorinių formatų eksportą į SVG grafiką.

Tam, kad SVG būtų galima naudoti internete, reikia turėti SVG palaikančią naršyklę. Šiuo metu visos pagrindinės naršyklės, išskyrus „Internet Explorer“, turi tam tikrą SVG palaikymą. Nuo 2007 m. SVG turinį „Internet Explorer“ naršyklėje galima peržiūrėti įdiegus papildinį. Populiariausias darbalaukio SVG papildinys yra sukurtas „Adobe Systems“ ir palaiko didžiąją dalį SVG 1.0/1.1.

Formalų kiekvieno SVG abstraktaus modulio apibrėžimą kaip DTD modulį galima rasti: <http://www.w3.org/TR/SVG11/svgdtd.html> (anglų kalba). DTD nurodyti visi leistini SVG elementai. Daugiau informacijos (anglų kalba) apie SVG rasite: <http://www.w3.org/TR/SVG11/> arba <http://www.w3schools.com/svg/>.

## 4.6. Interneto žemėlapių paslaugos ir į paslaugas orientuota architektūra (SOA)

Į paslaugas orientuota architektūra (Service-Oriented Architecture, SOA) – tai paslaugų rinkinys. Šios paslaugos tarpusavyje bendrauja. Toks bendravimas gali apimti paprastą duomenų perdavimą arba dviejų ar daugiau paslaugų veikimo koordinavimą.

Paslauga – tai gerai apibrėžta, savarankiška funkcija, kuri nepriklauso nuo kitų paslaugų konteksto arba būklės. Paslauga yra ryšio galutinis taškas. Į paslaugas orientuotoje architektūroje paslaugos vartotojas siunčia paslaugos prašymo pranešimą paslaugos teikėjui. Paslaugos teikėjas atsakymo pranešimą išsiunčia paslaugos vartotojui. Prašymo ir vėlesnio atsakymo jungtys apibrėžiamos ir paslaugos vartotojui, ir paslaugos teikėjui suprantamu būdu. Paslaugos teikėjas gali būti ir paslaugos vartotojas.

Interneto paslaugas galima naudoti į paslaugas orientuotai architektūrai realizuoti. Interneto paslaugose didžiausias dėmesys skiriamas tam, kad funkciniai teikėjo elementai būtų prieinami naudojant standartinius interneto protokolus, kurie nepriklauso nuo platformų ir programavimo kalbų. Šios paslaugos gali būti visiškai naujos programos arba įdiegiamos senose veikiančiose sistemose, kad pastarosios galėtų dirbti tinkle.

Kiekvienas SOA struktūrinis elementas gali atlikti vieną arba kelias iš trijų funkcijų:

- *Paslaugos teikėjas* sukuria interneto paslaugą ir jos sąsają bei prieigos informaciją tikriausiai publikuoja paslaugų registre.
- *Paslaugos tarpininkas*, arba paslaugų registras, atsako už tai, kad interneto paslaugos sąsaja ir realizavimo prieigos informacija būtų prieinama visiems potencialiems paslaugos prašytojams. Su viešaisiais tarpininkais galima susisiekti internetu. Kai kurie tarpininkai tvarko kitų tarpininkų katalogus. **UDDI** (*Universal Description Discovery and Integration*) specifikacija apibrėžia, kaip turi būti publikuojama ir atrandama informacija apie interneto paslaugas.
- *Paslaugos prašytojas* (arba interneto paslaugos klientas, vartotojas) įrašus tarpininkų registre suranda atlikdamas įvairius paieškos veiksmus ir tada prisijungia prie paslaugos teikėjo norėdamas pasinaudoti viena iš teikiamų paslaugų.

Reikalinga tam tikra priemonė, kuri sujungtų paslaugas. Sujungimo technologija yra į paslaugas orientuotos architektūros technologija. Tvirtam ryšiui sukurti interneto paslaugos naudoja XML. Ši kalba labai plačiai naudojama į paslaugas orientuotoje architektūroje kuriant duomenis, kurie yra beveik išsamioje aprašo talpykloje. Pačioms paslaugoms aprašyti paprastai naudojama **WSDL** (*Web Services Description Language*) kalba, o ryšio protokolams – **SOAP** (*Simple Object Access Protocol*) protokolas. Todėl **internetu paslauga** yra *programinės įrangos komponentas, aprašytas WSDL kalba, pasiekiamas SOAP protokolu per HTTP*.

**WSDL** (*Web Services Description Language*) – tai interneto paslaugos sąsajos aprašymo formatas, kuriuo aprašomos paslaugos ir tai, kaip jas susieti su konkrečiais tinklo adresais. WSDL sudaro trys dalys:

- *Apibrėžtys* paprastai išreiškiamos XML kalba ir apima duomenų tipo apibrėžtis bei pranešimo apibrėžtis. Šios apibrėžtys paprastai parengiamos naudojant kokį nors suderintą XML žodyną, pavyzdžiui, OGC interneto paslaugų specifikacijas.

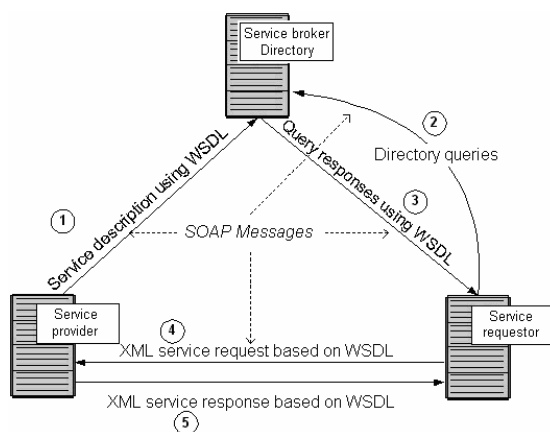
- *Operacijos* apibūdina, kokie veiksmai atliekami su pranešimais, kuriuos palaiko interneto paslauga. Operacijos gali būti keturių rūšių:
  - vienkrypčiai pranešimai, kuriuos išsiuntus nereikalingas atsakymas
  - prašymų ir atsakymų pranešimai, į kuriuos turi atsakyti gavėjas
  - atsakymo prašymai, kurie yra prašymai atsakyti
  - informaciniai pranešimai, kurie vienu metu siunčiami daugeliui gavėjų

Operacijos skirstomos į prievadų tipus (angl. *port types*), kurie nustato interneto paslaugos palaikomas operacijas.

- Paslaugų susiejimai (angl. *bindings*) prievadų tipus sujungia su prievadu. Prievadas apibrėžiamas susiejant tinklo adresą su prievado tipu. Prievadų rinkinys apibrėžia paslaugą. Toks susiejimas paprastai sukuriamas naudojant SOAP, tačiau galimos ir kitos formos.

WSDL yra interneto paslaugų pagrindas. Paslaugos teikimo ir vartojimo žingsniai:

1. Paslaugos teikėjas apibrėžia paslaugą vartodamas WSDL kalbą. Ši apibrėžtis publikuojama paslaugų kataloge; katalogas gali naudoti UDDI.
2. Paslaugos prašytojas pateikia prašymą paslaugų tarpininkų katalogui surasti paslaugą ir nustatyti, kaip su ta paslauga bendraujama.
3. Dalis WSDL, kurią parengė paslaugos teikėjas, yra perduodama paslaugos prašytojui. Taip paslaugos vartotojas susipažįsta su paslaugos teikėjo prašymais ir atsakymais.
4. Paslaugos prašytojas WSDL vartoja siųsdamas prašymą paslaugos teikėjui.
5. Paslaugos teikėjas laukiamą atsakymą pateikia paslaugos prašytojui.



#### Pav. WSDL vartojimo žingsniai

Paslaugų tarpininkų katalogas gali būti UDDI registras. UDDI teikia aprašą atitinkančių paslaugų rinkinio apibrėžtį ir užtikrina įmonių, organizacijų bei kitų interneto paslaugų teikėjų, jų teikiamų interneto paslaugų ir techninių sąsajų, kurios gali būti naudojamos prieigai prieš šių paslaugų, suradimą. UDDI yra grindžiamas įprastais šios veiklos srities standartais, įskaitant HTTP, XML, „XML Schema“ ir SOAP.

Pagrindinis tikslas – „surasti“ organizacijas ir organizacijų siūlomas paslaugas (tarsi naudojantis telefonų knyga ar telefonų sąrašu). UDDI registre kontaktinės informacijos ir įvairioms organizacijoms teikiamų interneto paslaugų paiešką galima atlikti įvairiais būdais.

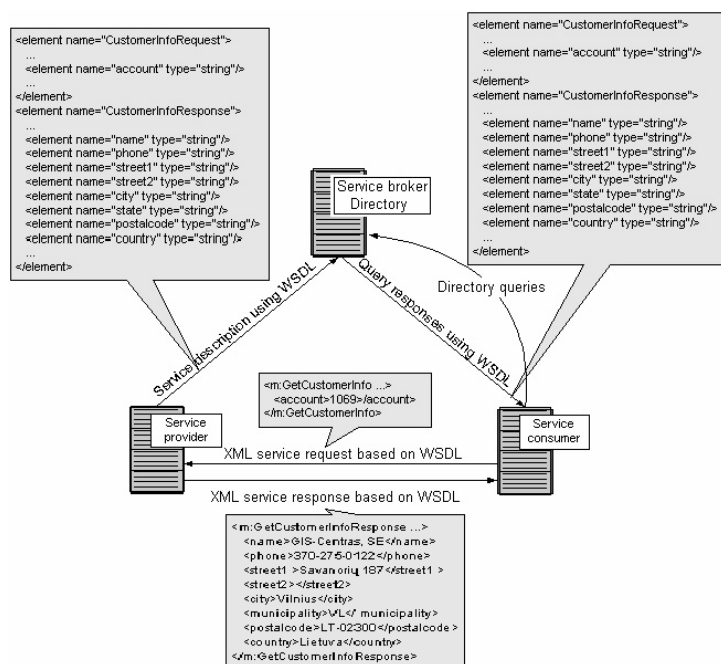
Registro sistemą sudaro trys katalogai: baltieji puslapiai, geltonieji puslapiai ir žalieji puslapiai.

Visi pranešimai į paslaugas orientuotoje architektūroje siunčiami naudojant SOAP protokolą (<http://www.w3.org/TR/2007/REC-soap12-part0-20070427/> – anglų kalba), kuris yra vienas iš W3C konsorciūmo parengtų standartų. SOAP leidžia kompiuteriams bendrauti HTTP arba XML protokolais neatsižvelgiant į operacinę sistemą ar platformą. SOAP – tai metodas XML (Extensible Markup Language) kalbai vartoti pranešimų ir RPC (Remote Procedure Call) technologijos protokoluose. Iš esmės SOAP yra savarankiška vienkrypčių pranešimų mainų paradigma. SOAP 1.2 versijoje pateikta XML informacijos, kurią lygiaverčiai įrenginiai gali naudoti struktūrinės ir tipizuotos informacijos mainams siųsdami pranešimus decentralizuotoje ir išskirstytoje aplinkoje, apibrėžtis. Formaliai SOAP pranešimas parengiamas naudojant XML gramatiką su SOAP elementais (pvz., `<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">`), kurioje nurodomas abstraktus turinys.

SOAP protokole naudojamas vokas `<env>`, kuriame siunčiami interneto paslaugų pranešimai. SOAP dažniausia naudojamas pranešimams siųsti HTTP protokolu, tačiau galima naudoti ir kitas ryšio priemones. SOAP gali būti naudojamas keičiantis visais dokumentais arba iškviečiant nuotolinę procedūrą.

SOAP vokas yra sudarytas iš dviejų dalių (daugiau žr. <http://www.w3.org/TR/2007/REC-soap12-part0-20070427/#intro> – anglų kalba):

1. Neprivalomos antraštės, kurioje nurodoma tapatybės nustatymo informacija, duomenų kodavimas arba tai, kaip SOAP pranešimo gavėjas turėtų apdoroti pranešimą.
2. Pagrindinės dalies, kurioje yra pranešimas. Šiuos pranešimus galima apibrėžti naudojant WSDL specifikaciją.

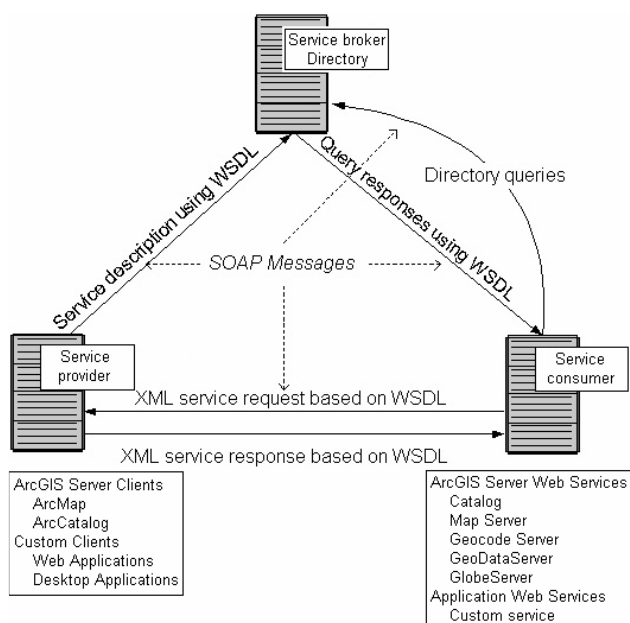


**Pav. Pranešimų siuntimas į paslaugas orientuotoje architektūroje naudojant interneto paslaugas** (<http://www.service-architecture.com/> – anglų kalba)



Panašios technologijos (WSDL, SOAP su HTTP, UDDI) gali būti naudojamos interneto žemėlapių paslaugoms realizuoti ir prižiūrėti. Šios paslaugos gali užtikrinti prieigą prie plataus spektro žemėlapių ir GIS galimybių. Pavyzdžiui, ESRI paslauga „ArcWeb Services“ leidžia naudotojams į paslaugas orientuotoje architektūroje kliento taikomas programas papildyti žemėlapių ir erdvinėmis funkcijomis. Šios paslaugos gali apimti duomenų vizualizavimą, surastus duomenis, įskaitant maršrutus, vietų ir adresų suradimą, erdviųjų užklausų paieškas ir verslo analizės atlikimą (angl. *business intelligence performance*) (<http://www.arcwebservices.com/v2006/help/index.htm> – anglų kalba).

ESRI „ArcGIS Server“ taip pat sudaro galimybes į paslaugas orientuotoje aplinkoje naudoti geoerdvinę informaciją. „ArcGIS Server“ siūlo erdviųjų duomenų vizualizavimo, erdviųjų duomenų analizės ir duomenų tvarkymo paslaugas. Šias paslaugas taip pat galima iškviešti ir integruoti į kitas interneto paslaugas naudojant standartinius interneto paslaugų protokolus, pavyzdžiui, SOAP ir XML (<http://webhelp.esri.com/arcgisserver/9.2/> – anglų kalba).



#### Pav. ESRI paslauga „ArcWeb Services“

Nauji interneto technologijų patobulinimai, pavyzdžiui, vadinamasis „**Web 2.0**“, taip pat naudojami geoerdviųjų duomenų ir paslaugų kūrimui. „Web 2.0“ išplečia nekeičiamųjų svetainių galimybes ir numato socialinį elementą, kuriame naudotojai kuria ir platina turinį.

„Web 2.0“ reiškia „antrosios kartos“ svetaines, kuriose lankytojai gali teikti bendradarbiavimui ir dalijimuisi skirtą informaciją. „Web 2.0“ taikomosios programos naudoja interneto paslaugas ir gali siūlyti „Ajax“, „Flash“ (<http://labs.adobe.com/>) arba „JavaFX“ (<http://www.sun.com/software/javafx/index.jsp>) vartotojo sąsajas, interneto informacijos apjungimą (angl. web syndication), RSS srautus (žr. dalį „Išvada“), tinklaraščius (angl. blogs) ir vikius (angl. wikis). Interneto informacijos apjungimas reiškia tai, kad svetainėje sukurti nauji kanalai (angl. web feeds), kurie kitus žmones supažindina su glausta informacija apie svetainės turinio atnaujinimus. Dažnai naujių kanalų yra XML dokumentas, kuriame yra turinio punktai su nuorodomis į ilgesnes versijas. Tinklaraštis arba interneto dienoraštis – tai tokia svetainė, kurioje galima daryti įrašus tvarkant arba papildant turinį, o įrašai paprastai pateikiami atvirkštine chronologine

tvarka. Vikis – tai programinė įranga, kuri leidžia naudotojams nesunkiai kurti, redaguoti ir susieti tinklalapius. Vikiai dažnai naudojami kuriant bendradarbiavimo svetaines bei gerinant bendruomenių svetainių turinį.

Nors „Web 2.0“ standartai nėra nustatyti, technologija papildo esamą interneto serverių architektūrą ir naudoja paslaugas. Todėl galima manyti, kad „Web 2.0“ yra būdingos kai kurios SOA charakteristikos.

Yra manančių, kad hibridai (angl. mashups) yra „Web 2.0“ technologijos dalis. Šiuo metu yra trijų rūšių hibridų: vartotojų hibridai, duomenų hibridai ir verslo hibridai. Geriausiai žinomas vartotojų hibrido pavyzdys yra „Google Maps“ programos. Vartotojų hibriduose derinami iš skirtingų šaltinių gautų duomenų elementai, paslėpti po paprasta suvienodinta grafine sąsaja. Populiarūs yra ir „duomenų hibridai“ bei „įmonių hibridai“. Duomenų hibride derinami iš skirtingų šaltinių gauti panašaus tipo duomenys (pvz., „Yahoo Pipes“), pavyzdžiui, duomenys iš skirtingų RSS kanalų derinami viename kanale, turinčiame grafinę sąsają. Įmonės hibride dažniausiai integruojami iš vidinių ir išorinių šaltinių gauti duomenys. Pavyzdžiui, derinant išorinį visų rinkoje parduotų namų sąrašą su vidiniais duomenimis apie agentūros parduotus namuose, galima sukurti nekilnojamojo turto ataskaitą. Verslo hibridas – tai visų išvardytų hibridų derinys, kuriame akcentuojamas duomenų kaupimas ir pateikimas, taip pat bendradarbiavimo funkcijos, sukuriant galutinį rezultatą, kuris tinka naudoti kaip verslo taikomoji programa.



## 4.7. Išvados

XML (*Extensible Markup Language*) vartojama taikomosioms kalboms, kurios atlieka skirtingas funkcijas, realizuoti. Todėl geoerdvinėse taikomosiuose programose gali būti vartojamos tokios XML gramatikos kalbos:

- GML (*Geographic Markup Language*) kalba vartojama:
  - koduojant geografinę informaciją
  - perduodant ir saugant geografinę informaciją
  - modeliuojant erdvinių duomenų taikomąsias schemas
- ESRI ArcXML (*Arc Extensible Markup Language*) kalba vartojama:
  - apibrėžiant žemėlapiu turinio vaizdavimą, erdvinių duomenų vietą ir vietinę aplinką ESRI žemėlapių paslaugoms
  - koduojant užklausų ir atsakymų pranešimus ESRI ArcIMS architektūroje
- KML (*Google Keyhole Markup Language*) kalba vartojama:
  - kreipiantis į „Google Maps“ interneto žemėlapių paslaugą
  - hibridinant naudotojų apibrėžtus objektus „Google“ žemėlapyje
- SVG (*Scalable Vector Graphics*) kalba vartojama:
  - vaizduojant dvimatę vektorinę grafiką, įskaitant ir interneto naršyklėje
- SOAP (*Simple Object Access Protocol*) pranešimai naudojami:
  - keičiantis ištisais dokumentais arba iškviečiant interneto taikomųjų programų nuotolinę procedūrą (pvz., į paslaugas orientuotoje architektūroje)
- WSDL (*Web Services Description Language*) kalba vartojama:
  - apibrėžiant interneto paslaugas, įskaitant duomenų tipo ir pranešimų apibrėžtis, interneto paslaugų operacijas ir susiejimus
- UDDI (*Universal Description Discovery and Integration*) specifikacijos programavimo sąsaja grindžiama XML
- GeoRSS (*Geographic Really Simple Syndication*) formatas naudojamas:
  - RSS arba „naujienų kanalas“ – tai XML formatų šeima, skirta keistis naujienomis, ypač naujienomis apie tinklalapius ar kitą interneto turinį (<http://www.rssboard.org/rss-specification> – anglų kalba). Daugelyje dinaminių tinklalapių, ypač „*tinklaraščiuose*“, yra naujo arba pakitusio turinio RSS „kanalai“. Jiems naudojami paprasti struktūriniai XML formatai, kuriuose yra tik svarbiausieji aprašomieji elementai – autorius, data, pavadinimas, apibūdinimas ir hiperteksto nuoroda, kuri padeda skaitytojui arba RSS „kaupimo“ paslaugai nuspręsti, su kuria pirmine medžiaga verta susipažinti išsamiau.
  - GeoRSS (<http://www.georss.org/>) – tai paprastas RSS kanalas, kuriuos aprašo vieta arba geografinės žymos, pasiūlymas. GeoRSS kanalus galima naudoti geografinės informacijos paieškai ir kaupimui. Naudojant GeoRSS, vieta aprašoma suderinamu būdu, todėl taikomios programos gali prašyti, kaupti, dalytis ir susieti geografiškai pažymėtus kanalus. Naudojant GeoRSS

galima ne tik gauti informacijos apie konkretų miestą arba pašto indeksą, bet ir ieškoti pagal pačius įvairiausias geografinius kriterijus.

- Šiuo metu yra dvi GeoRSS koduotės: „GeoRSS-Simple“ ir „GeoRSS-GML“. „GeoRSS Simple“ palaiko pagrindines geometrijas (tašką, tiesę, langelį, daugiakampį) ir apima tipinius naudojimo koduojant vietas atvejus. „GeoRSS GML“ yra formalusis GML taikomasis profilis ir palaiko daugiau objektų bei koordinatinių atskaitų sistemų, ne tik WGD-84 platumos ir ilgumos koordinatas. Abu formatai skirti naudoti su „Atom 1.0“ ([http://en.wikipedia.org/wiki/Atom\\_%28standard%29](http://en.wikipedia.org/wiki/Atom_%28standard%29) – anglų kalba), „RSS 2.0“ ir „RSS 1.0“, nors tinka naudojimui ir ne RSS XML kodavimui.

„GeoRSS Simple“ formatu užkoduotas taškas atrodo taip:

```
<georss:point>25.258197 54.680445</georss:point>
```

„GeoRSS GML“ formatu:

```
<georss:where>  
  <gml:Point>  
    <gml:pos>25.258197 54.680445</gml:pos>  
  </gml:Point>  
</georss:where>
```

### **Paskaitos savarankiško mokymosi klausimai:**

1. Kokia yra GML kodavimo esmė?
2. Kuo skiriasi GML ir ArcXML?
3. Kokia ryšio kalba naudojama OGC WMS?
4. Kaip erdviniai duomenys koduojami OGC WFS?
5. Kaip erdviniai duomenys koduojami ESRI „ArcIMS Feature Server“?
6. Kokie yra trys SOA struktūriniai elementai? Kaip SOA tarpusavyje susiejamos paslaugos?
7. Kas yra „hibridinimas“ ir kaip jis naudojamas „Web 2.0“ žemėlapių technologijoje?

### **Rekomenduojamieji skaitiniai (anglų kalba)**

- OpenGIS Geography Markup Language (GML) Encoding Standard, [http://portal.opengeospatial.org/files/?artifact\\_id=20509](http://portal.opengeospatial.org/files/?artifact_id=20509)
- Ho, M., Extending ArcIMS MapServices with ArcXML, ESRI Educational Services, <http://www.esri.com/news/arcuser/0102/extend1of3.html>
- SVG Tutorial, W3 School, <http://www.w3schools.com/svg/>
- SOAP Tutorial, W3 School, <http://www.w3schools.com/soap/>
- WSDL Tutorial, W3 School, <http://www.w3schools.com/wSDL/>
- Web Services Tutorial, W3 School, <http://www.w3schools.com/webservices/>
- Web Services and Service-Oriented Architectures, <http://www.service-architecture.com/>

## Nuorodos (anglų kalba)

- OpenGIS Geography Markup Language (GML) Encoding Standard, [http://portal.opengeospatial.org/files/?artifact\\_id=20509](http://portal.opengeospatial.org/files/?artifact_id=20509)
- *ArcXML Programmer's Reference Guide for ArcIMS 9.2*, <http://edndoc.esri.com/arcims/9.2/>, <http://webhelp.esri.com/arcims/9.2/general/> or <http://support.esri.com/index.cfm?fa=knowledgebase.webHelp.arcIMSGateway>
- KML in Google Maps, <http://code.google.com/apis/kml/documentation/mapsSupport.html>
- Scalable Vector Graphics (SVG), <http://www.w3.org/Graphics/SVG/>
- Web Services Description Language (WSDL), <http://www.w3.org/2002/ws/desc/>
- SOAP, <http://www.w3.org/TR/2007/REC-soap12-part0-20070427/>
- **OASIS UDDI Specification TC**, [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=uddi-spec](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec)

## Vartojamieji terminai

- XML gramatikos taikymas
- GML kalba
- Kodavimas
- ArcXML kalba
- AXL žemėlapių sąrankos failas
- KML kalba
- Į paslaugas orientuota architektūra (SOA)
- Hibridinimas
- SVG kalba
- Keičiamo dydžio
- Paslauga
- UDDI
- WSDL kalba
- SOAP protokolas
- Interneto informacijos apjungimas
- Tinklaraščiai
- Vikiai
- „Web 2.0“
- GeoRSS

## 5. ESRI interneto programų architektūra

Ankstesnėse paskaitose buvo aptarta interneto žemėlapių architektūra, jos komponentai ir vartotojo sąsajai, ryšiui, duomenims skirtos interneto kalbos bei schemų aprašai. Šioje paskaitoje anksčiau nagrinėtos temos pristatomos pasitelkiant praktinio įgyvendinimo pavyzdžius – ESRI ArcIMS ir „ArcGIS Server“ programinę įrangą. Kaip jau buvo minėta 3 paskaitoje, ESRI buvo interneto žemėlapių programinės įrangos kūrimo lyderė. Bendrovės produktus klientai rinkosi analizei atlikti, kadangi produktų dokumentaciją lengva naudoti.

ESRI ArcIMS naudoja IMS (*Internet Map Server*) programinę įrangą interneto žemėlapių programoms kurti, projektuoti, publikuoti ir administruoti. ArcIMS yra dinaminių žemėlapių ir GIS duomenų bei paslaugų teikimo internetu sprendimas. ArcIMS leidžia tarpusavyje bendrauti interneto klientams, žemėlapių serveriams, duomenų serveriams ir interneto serveriui.

Šis naujasis ESRI produktas („ArcGIS Server“) taip pat išplečia interneto žemėlapių programų galimybes. Be to, jis per interneto sąsają palaiko daugybę geoduomenų apdorojimo galimybių. ESRI „ArcGIS Server“ taip pat sudaro galimybes į paslaugas orientuotoje aplinkoje naudoti geoerdvinę informaciją.

### Paskaitos planas

- Kas yra ESRI ArcIMS ir „ArcGIS Server“?
- ESRI interneto komponentai: serverio pusė
- ESRI ArcIMS vartotojo sąsajos komponentai: serverio pusė
- ESRI ArcIMS komponentai: kliento pusė
- ESRI ArcIMS ryšio srautas
- ESRI ArcIMS diegimas
- ESRI ArcIMS ir „ArcGIS Server“ adaptavimo galimybės
- ESRI ArcIMS papildomos savybės
- Išvada

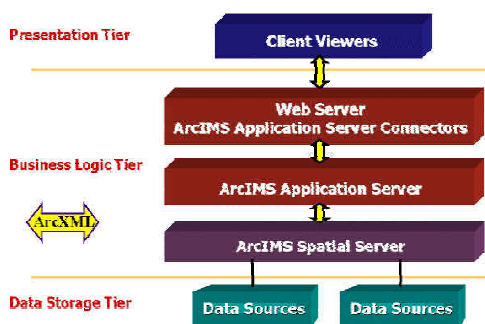
## 5.1. Kas yra ESRI ArcIMS ir „ArcGIS Server“?

Kaip jau buvo minėta 3 paskaitoje, ESRI buvo interneto žemėlapių programinės įrangos kūrimo lyderė. Šiuo metu ESRI interneto žemėlapių ir GIS paslaugas palaiko ArcIMS ir „ArcGIS Server“ programinė įranga. Kaip jau buvo minėta 3 paskaitoje, interneto paslaugos yra publikuojamos interneto serveryje, o kliento programos, kreipdamosi į interneto žemėlapių serverius, gali prašyti publikuotų paslaugų galimybių ir jas naudoti. Klientai ir interneto serveriai susijungia laisvai, o kiekvienas bendravimas su klientu sudaro atskirą transakciją. Transakciją sudaro užklausa, siunčiama atitinkamam interneto žemėlapių arba GIS serveriui, ir klientui grąžinamas atsakymas.

**ArcIMS** – tai interneto žemėlapių sprendimas ir sistema, skirta žemėlapių (ir pagrindinių GIS) galimybių platinimui internete. Kaip duomenų teikimo technologija, ArcIMS apima kelias ESRI *serverio* pusės technologijas (žemėlapių modulius), kurios palaiko žemėlapių ir metaduomenų paslaugų teikimą internete. Kaip publikavimo technologija, ArcIMS gali dirbti su pačiais įvairiausiais *klientais*. ArcIMS taip pat gali teikti OGC WMS ir WFS paslaugas.

„**ArcGIS Server**“ gali teikti ne tik interneto žemėlapių ir metaduomenų paslaugas (ESRI ir WMS), bet ir įvairios geografinės informacijos apdorojimo ir kitas paslaugas, todėl „ArcGIS Server“ galima priskirti ne tik interneto žemėlapių ir geografinės informacijos kodavimo serverio, bet ir GIS interneto serverio kategorijai. „ArcGIS Server“ galima naudoti interneto architektūroje arba kaip LAN/WAN tinklo paslaugą darbalaukiui klientams. „ArcGIS Server“ taip pat naudojamas su „išmaniaisiais klientais“ (angl. „*smart clients*“), t. y. laisvai susijungiančiu, lengvu rankiniu arba staliniu kompiuteriu, palaikančiu ilgalaikes duomenų podėlio ir atsijungusio GIS kliento operacijas. Kliento programų taikymą ir duomenų sinchronizaciją valdo „ArcGIS Server“ pirminės paslaugos.

ArcIMS ir „ArcGIS Server“ komponentai gali būti realizuojami daugiapakopėje programinės įrangos architektūroje, kuri buvo aptarta 3 paskaitoje. Šie komponentai turi skirtingus pavadinimus ir galimybes, tačiau logine prasme atlieka panašias funkcijas ir juos palaiko bendros platformos konfigūravimo strategijos. Šios programos yra sudarytos iš trijų pakopų: kliento arba pateikimo pakopos, verslo logikos pakopos ir duomenų saugojimo pakopos. Šios pakopos apibūdina įvairių taikomųjų programų komponentų galimybių loginį suskirstymą ir nebūtinai atitinka jų buvimo vietą.

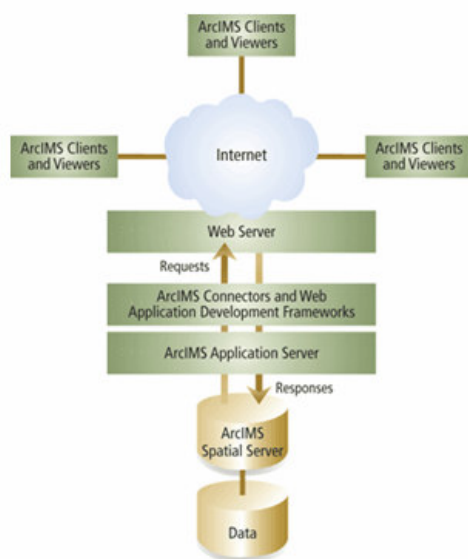


Pav. ArcIMS sudaro trys pakopos (<http://www.esri.com/software/arcgis/arcims/> – anglų kalba)

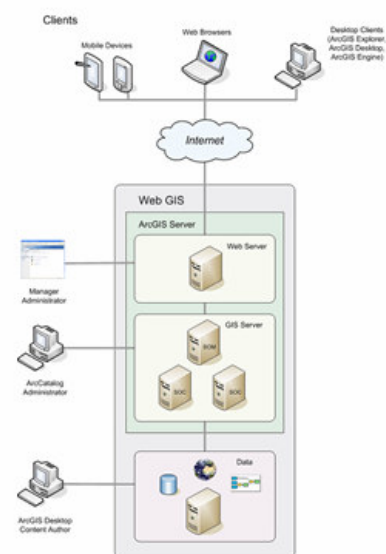
*Pateikimo pakopą* sudaro kliento pusės komponentai, naudojami siunčiant užklausas serveriui ir peržiūrint žemėlapius bei duomenis.

*Verslo logikos pakopą* sudaro serverio pusės komponentai. Abiejuose programinės įrangos sprendimuose yra interneto taikomųjų programų, paslaugų valdymo ir erdvinio serverio sluoksniai, kurie gali būti taikomi skirtingų platformų deriniuose. Šioje pakopoje apdorojamos užklausos ir atliekamas serverio administravimas. Įvairių programinės įrangos komponentų vieta ir pasirinkta programinės įrangos sąranka gali turėti įtakos sistemos darbui.

*Duomenų saugojimo pakopą* sudaro failų serveriai, kuriuose laikomi „shape“ failai ir vaizdų failai, bei serveris, kuriame įdiegta „Oracle“ (arba kita RDBMS) ir SDE. Erdvinį serverį taip pat galima laikyti duomenų saugojimo pakopos dalimi.



**Pav. ArcIMS komponentai**  
[\(http://webhelp.esri.com/arcims/9.2/general/\)](http://webhelp.esri.com/arcims/9.2/general/)

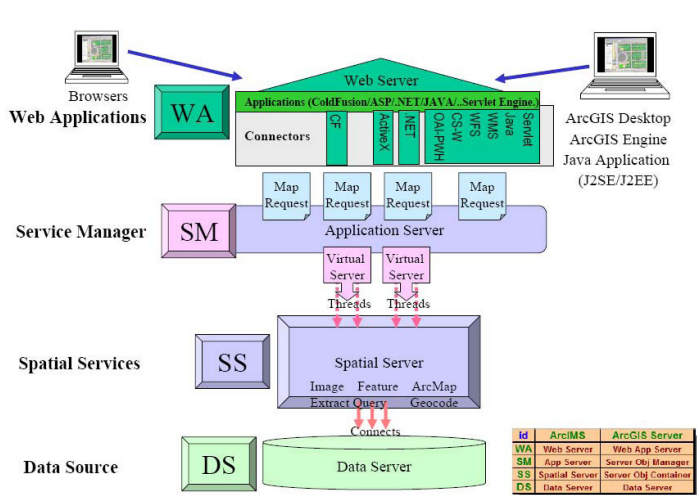


**Pav. „ArcGIS Server“ komponentai**  
[\(http://webhelp.esri.com/arcims/9.2/general/\)](http://webhelp.esri.com/arcims/9.2/general/)

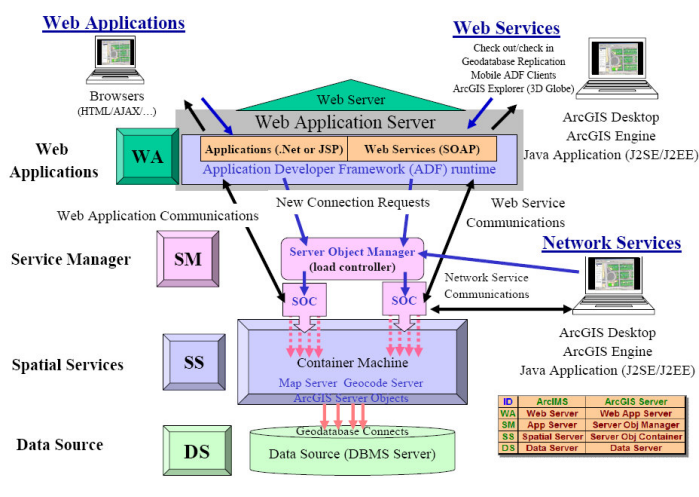


## 5.2. ESRI interneto komponentai: serverio pusė

Šioje dalyje bus aptariami ESRI interneto žemėlapių programos, pavyzdžiui, ArcIMS, komponentai. Kaip jau buvo minėta, logine prasme „ArcGIS Server“ interneto žemėlapių programa atlieka panašias funkcijas kaip ir ArcIMS.



Pav. ArcIMS komponentai  
<http://www.esri.com/library/whitepapers/pdfs/sysdesiq.pdf>



Pav. „ArcGIS Server“ komponentai  
<http://www.esri.com/library/whitepapers/pdfs/sysdesiq.pdf>

Kaip bendrauja žiūryklė ir serveris? ArcIMS klientas sukuria užklausą, kuri siunčiama interneto serveriui „Web Server“, per jungtis „Connector“ patenka į taikomųjų programų serverį „Application Server“, po to į virtualųjį serverį „Virtual Server“, kuris ją perduoda vienam iš erdvinių serverių „Spatial Server“. Visos užklaustos ir atsakymai yra rašomi ArcXML arba verčiami į ArcXML. Atsižvelgiant į šį srautą, ArcIMS komponentai yra:

- ArcIMS interneto programų slunksnyje yra interneto HTTP serveris ir interneto programų serverio jungtis.
- „Web Server“ – tai komponentas, kuris tiesiogiai bendrauja su klientu ir kuris būtinas ArcIMS programinės įrangos veikimui. Tai kompiuterinė programa, kuri saugo ir pateikia prašomus HTML puslapius arba failus. Interneto serveris turi palaikyti hiperteksto perdavimo protokolo (HTTP) užklausas. Pavyzdžiui, interneto naršyklė yra klientas, kuris interneto serveriams teikia HTML failų prašymus.

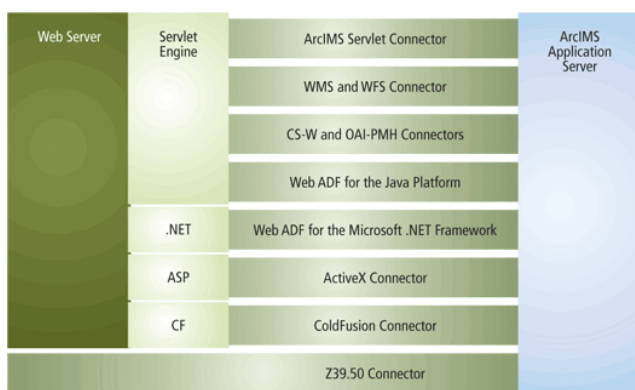
Interneto serveriai yra išoriniai programinės įrangos komponentai, neįtraukti į ArcIMS diegimo paketus. Galima naudoti įvairius interneto serverius – „Microsoft Internet Information Server“ (IIS), „Netscape Enterprise Server“, „iPlanet“ arba „Apache“.

- „ArcIMS Application Server“ jungtys „Connectors“ leidžia interneto serveriui bendrauti su ArcIMS taikomųjų programų serveriu „Application Server“, taip pat užtikrina ryšio kanalą tarp interneto serverio arba trečiosios šalies taikomųjų programų serverio (pvz., „ColdFusion“) ir „ArcIMS Application Server“. Jos turi

užklausa kita kalba išversti į ArcXML kalbą ir tik po to ją išsiųsti į „Application Server“. Jas galima naudoti vartotojo darbo srautams ir pateikimui kliento ekrane tvarkyti ir optimizuoti (adaptuojant).

Kadangi ArcIMS „Application Server“ ir „Spatial Servers“ serverių bendravimui naudojama ArcXML, reikalingas mechanizmas, kuris leistų užklausas iš HTTP/XML standartų tipo protokolų arba kitų kūrimo aplinkų (pvz., „ColdFusion“ puslapių, dinaminių parametrinių tinklalapių) išversti į ArcXML. Tam naudojamos „Application Server“ jungtys ir interneto taikomųjų programų kūrimo platformos (*Web Application Development Framework*) (internetu ADF). Jungtys ir internetu ADF veikia tarp internetu serverio su serverio programėlės moduliu „Servlet Engine“ ir ArcIMS „Application Server“ arba tarp trečiosios šalies taikomųjų programų serverio (pvz., „ColdFusion“) ir ArcIMS „Application Server“.

ArcIMS palaiko keletą kūrimo aplinkų ir gali apdoroti užklausas iš kai kurių plačiai naudojamų standartinių protokolų. ArcIMS turi keletą jungčių (žr. paveikslą toliau). „ArcIMS Servlet Connector“ yra numatytoji ArcIMS jungtis. Visos serverio jungtys „Server Connectors“ turi būti įdiegtos tame pačiame kompiuteryje kaip ir internetu serveris „Web Server“.



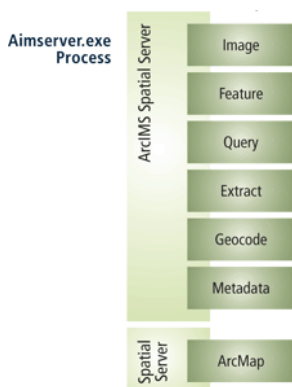
Pav. ArcIMS serverio „Application Server“ jungtys (<http://webhelp.esri.com/arcims/9.2/> – anglų kalba)

ArcIMS taip pat turi jungčių, kurios atitinka „Open Geospatial Consortium Inc.“ specifikacijas: „WMS Connector“, „WFS Connector“ ir „CS-W Connector“ (OGC standartinė katalogo paslauga 2.0 ArcIMS metaduomenų paslaugoms). Konvertavimo modulis gauna OGC standarto WMS užklausas ir jas konvertuoja į ArcXML užklausas, kurias persiunčia publikuotoms ArcIMS paslaugoms. ArcXML atsakymai vėl konvertuojami į OGC standarto atsakymus ir gražinami užklausas pateikusiems klientams. „WFS Connector“ leidžia objektų ir požymių informaciją siųsti klientams.

- ArcIMS „**Spatial Server**“ turi bendrauti su „Application Server“, analizuoti ArcXML užklausas skaidydamas jas į komponentus, perskaityti „shape“ failų, vaizdų ir SDE duomenis iš kitų serverių ir apdoroti gautą užklausa. Gavęs užklausa, „Spatial Server“ atlieka vieną arba kelias funkcijas:

- *Vaizdo* funkcija sukuria žemėlapiu vaizdo failą pagal žemėlapi, sukurtą naudojant „ArcIMS Author“ (logine prasme atitinka WMS)

- *ArcMap vaizdo funkcija* sukuria žemėlapio vaizdo failą pagal žemėlapi, sukurtą naudojant „ArcMap“ (logine prasme atitinka WMS)
- *Objekto funkcija* siunčia žemėlapio objektus (logine prasme atitinka WFS)
- *Užklauso funkcija* ieško objektų, kurie atitinka paieškos kriterijus
- *Geokodo funkcija* atlieka adresų sutapdinimo operacijas
- *Išgavimo funkcija* sukuria „shape“ failus iš pasirinktų žemėlapio objektų
- *Metaduomenų funkcija* publikuoja ir ieško metaduomenų
- *Maršruto funkcija* apskaičiuoja maršrutus tarp dviejų arba daugiau sustojimų, jeigu naudojamas pasirenkamas ESRI „RouteServer“ išplėtimas
- *SDCGeocode funkcija* atlieka „RouteServer“ duomenų adresų sutapdinimo operacijas



Pav. ArcIMS „Spatial Server“ (aimserver.exe) ir jo komponentai (<http://webhelp.esri.com/arcims/9.2/> – anglų kalba)

- Naudodamos ArcIMS programinę įrangą, žemėlapio turinį internete arba intranete publikuoja viena arba kelios ArcIMS paslaugos. ArcIMS „**Services**“ – tai procesai, nuolat vykstantys ArcIMS erdviname serveryje ir iš anksto nustatytu metu arba reaguodami į tam tikrus įvykius atliekantys numatytą operaciją. ArcIMS paslauga per interneto serverį naudotojams pateikia erdvinius duomenis ir (arba) metaduomenis.

ArcIMS paslaugos yra keturių rūšių: „Image Service“ ir „ArcMap Image Service“ (logine prasme atitinka OGC WMS lengvam „plonajam“ klientui), „Feature Service“ (logine prasme atitinka OGC WFS – objektų srautas siunčiamas laikinu ESRI suglaudintu formatu arba GML formatu „storajam“ klientui) ir „Metadata Service“ (leidžia dalytis metaduomenimis).

- Keliuose kompiuteriuose gali būti įdiegti keli „Spatial Servers“, todėl šiems erdviniams serveriams ir juose veikiančioms paslaugoms valdyti reikalingas mechanizmas. ArcIMS „**Virtual Server**“ yra mechanizmas, sugrupuojantis vieną arba daugiau erdvinių serverių. Jo paskirtis – administruoti ir valdyti apkrovimo paskirstymą tarp erdvinių serverių.

„Virtual Server“ iš tiesų nėra serveris. ArcIMS paslauga „Service“ yra priskiriama „Virtual Server“: kai paleidžiama paslauga, ji turi būti priskirta „Image Server“, „ArcMap Server“, „Feature Server“, „Metadata Server“, „Virtual Server“ arba kitam „Virtual Server“, o ne tiesiogiai konkrečiam ArcIMS „Spatial Server“. Paslauga paleidžiama visiems egzemplioriams, esantiems grupėje „Virtual

Server“. ArcIMS „Virtual Server“ naudoja svetainei valdyti, todėl ArcIMS paslaugos viename arba keliuose erdviniuose serveriuose gali būti susietos tarpusavyje ir vienu kartu perduotos klientui.

Virtual Server Name	Description	Access Type
ExtractServer1	ExtractServer	PRIVATE
FeatureServer1	FeatureServer	PUBLIC
GeocodeServer1	GeocodeServer	PRIVATE
ImageServer1	ImageServer	PUBLIC
ImageServerArcMap1	ArcMap ImageServer	PUBLIC
MetadataServer1	MetadataServer	PUBLIC
QueryServer1	QueryServer	PRIVATE

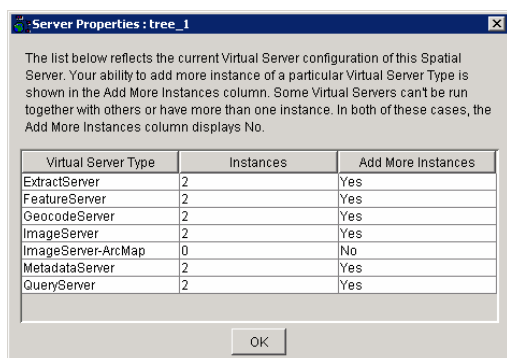
#### Pav. Virtualiųjų serverių tipai

ArcIMS erdviųjų serverių grupavimas yra svarbus ne tik dėl administravimo, bet ir dėl patikimumo. Jeigu ArcIMS „Spatial Server“ nustoja veikęs, gaunamas užklausa gali tvarkyti kiti erdviniai serveriai, priskirti tam pačiam virtualiajam serveriui.



Pav. „Feature Virtual Server“ sugrupuoja dviejų „Spatial Server“ serverių „Feature Server“ egzempliorius. „Image Virtual Server“ analogiškai sugrupuoja dviejų „Image Server“ egzempliorius.

- „Spatial Server“ – tai toks procesas, kuris suaktyvina ir valdo vieną arba daugiau serverio **egzempliorių** (gijų). Numatyta, kad ArcIMS vienam virtualiojo serverio procesui priskiria du egzempliorius.



Pav. Numatyta, kad ArcIMS vienam „Virtual Server“ procesui priskiria du egzempliorius. Išimtis taikoma erdviniams serveriams, kurie teikia ArcMap paslaugas „Image Services“. Šiems erdviniams serveriams priskiriama tik po vieną egzempliorių.

Egzempliorių skaičius yra lygus užklauskų, kurios gali būti apdorojamos vienu metu, skaičiui. Pavyzdžiui, jeigu virtualųjį serverį sudaro du erdviniai serveriai, kurių kiekvienas turi po du egzempliorius, toks virtualusis serveris vienu metu gali apdoroti keturias užklausas.

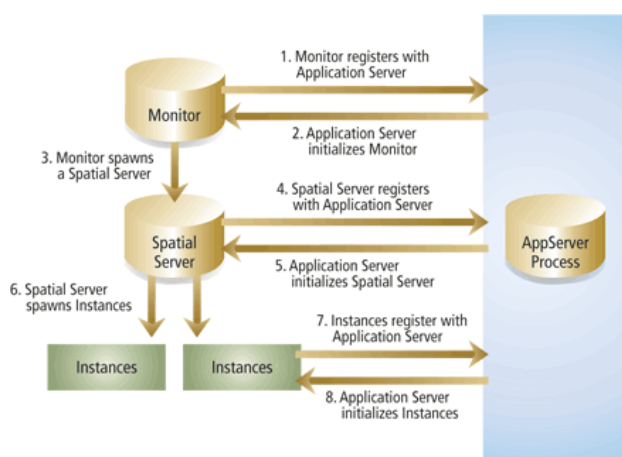
- ArcIMS „**Service Manager**“ komponentai atlieka paslaugos valdymo funkciją, kuri palaiko gaunamų žemėlapių paslaugos užklauskų eiles („Virtual Servers“) ir

sukonfigūruotus jungimusis prie ArcIMS viešųjų paslaugų modulių („Image“, „ArcMap“, „Feature“, „Extract“). Gaunamos užklausos nukreipiamos į turimus paslaugos egzempliorius apdorojimui. Taikomųjų programų serverio funkcijoms palaikyti reikalingas santykinai nedidelis apdorojimas.

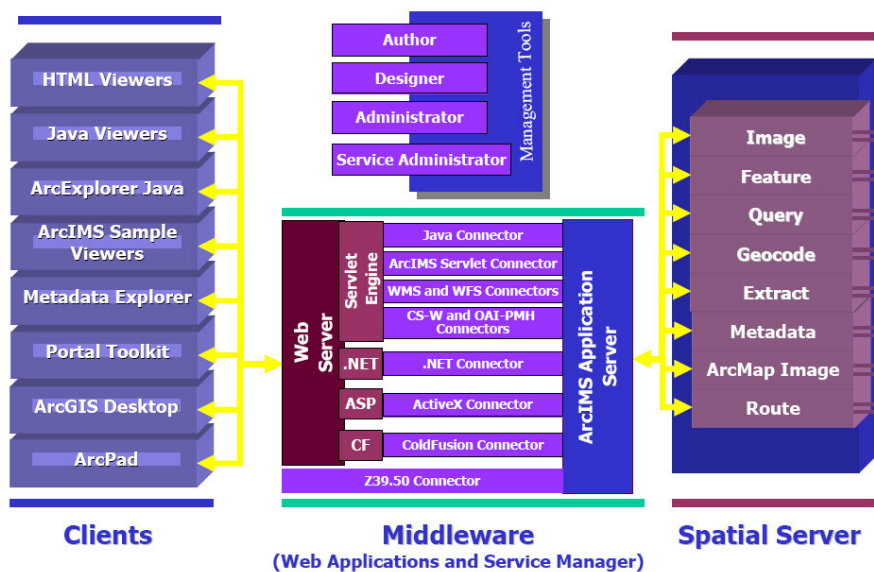
- ArcIMS „**Application Server**“ naudojamas kaip foninis procesas, stebintis, kurios paslaugos veikia konkrečiame virtualiajame serveryje. Jis paskirsto gaunamų užklausų apkrovą. Kiekviename virtualiajame serveryje taikomųjų programų serveris stebi, kurie erdvinio serverio egzemplioriai yra susieti su virtualiuoju serveriu. Naudodamas šią informaciją, taikomųjų programų serveris gaunamą užklausą išsiunčia atitinkamam erdvinio serverio egzemplioriui. Kai užklausų skaičius viršija turimų egzempliorių skaičių, užklausos saugomos taikomųjų programų serverio eilėje. Kai tik kuris nors egzempliorius atsilaisvina, taikomųjų programų serveris jam nusiunčia užklausą.

„Application Server“ nebūtina įdiegti į tą kompiuterį, kuriame yra „Web Server“.

- ArcIMS „**Monitor**“ – tai foninis procesas, kuris stebi erdvinio serverių būklę ir paleidžia arba atkuria ArcIMS paslaugas. Taikomųjų programų serverio ir erdvinio serverio procesai tiesiogiai bendrauja tarpusavyje. Kai erdvinio serverio procesas nustoja bendrauti su taikomųjų programų serveriu, taikomųjų programų serveris „Monitor“ nurodo iš naujo paleisti erdvinio serverio procesą. Esant standartinei sąrankai, kiekviename kompiuteryje, kuriame įdiegtas vienas ar daugiau „Spatial Server“ procesų, yra vienas „Monitor“.
- ArcIMS „**Tasker**“ – tai foninis procesas, kuris stebi erdvinio serverio sukurtus išvedinių failus ir periodiškai ištrina išvedinių vaizdų failus naudotojo nurodytais intervalais.



Pav. „Application Server“, „Monitor“ ir „Spatial Server“ procesai veikia kartu bendraudami, paleidžia paslaugas ir apdoroja užklausas (<http://webhelp.esri.com/arcims/9.2/> – anglų kalba)



Pav. ArcIMS 9.2 komponentų architektūros vaizdas (<http://www.esri.com/library/whitepapers/pdfs/sysdesiq.pdf> – anglų kalba)

- „**Data Source**“ yra failų serverio duomenys. Jame saugomi GIS duomenys. ArcSDE duomenų šaltinis palaiko užklausų apdorojimo funkcijas. Šiame lygmenyje taip pat yra standartiniai GIS vaizdų arba failų duomenų šaltiniai.



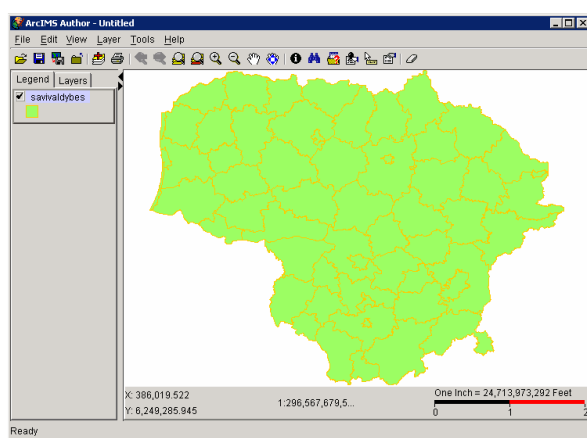
Pav. GIS duomenų šaltinio pakopos pavyzdys



### 5.3. ESRI ArcIMS vartotojo sąsajos komponentai: serverio pusė

Vienas ArcIMS pranašumas yra tas, kad programinė įranga suteikia interneto taikomųjų programų kūrimo ir administravimo išteklius bei priemones. Yra sukurtos tokios ArcIMS priemonės:

- Savarankiška ArcIMS „**Author**“ taikomoji programa yra naudojama žemėlapių .axl sąrankos failui, kuris atitinka žemėlapi, sukurti. Šis AXL failas naudojamas kaip „Image Service“ arba „Feature service“ pradiniai duomenys. „Author“ leidžia įterpti duomenų sluoksnius, nustatyti sluoksnių simboliką, sukurti sukauptas užklausas ir nustatyti geokodavimo sluoksnius .axl sąrankos faile. Sąrankos faile yra nurodymai, kurie duomenys bus pateikiami, faktinis kelias iki tų duomenų ir prisijungimo informacija, taip pat tiksli tokių duomenų išvaizda.



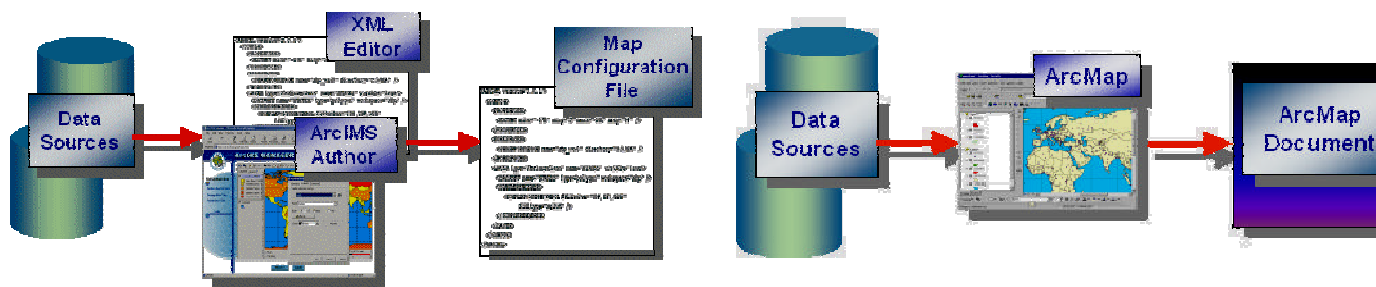
Pav. ArcIMS „Author“

Žemėlapių sąrankos failai yra rašomi ArcXML kalba, kuri yra XML išplėtimas, skirtas ArcIMS (daugiau informacijos žr. 4 paskaitoje). „Author“ galimybės neleidžia į žemėlapių sąrankos failą įterpti visų ArcXML žymų, kurias galima naudoti. Tačiau .axl failus galima redaguoti kaip visus kitus XML dokumentus numatant papildomas galimybes (pvz., ranka galima įrašyti žemėlapių projekcijų informaciją).

ArcMap paslauga „Image Service“ ArcMap dokumento (.mxd) failą, sukurtą ArcMap, naudoja kaip pradinis duomenimis.

Lentelė. Žemėlapių sudarymas ArcIMS procesams

Procesas	Paslaugos tipas			
	Image	Feature	ArcMap Image	Metadata
Žemėlapių kūrimo produktas	Author	Author	ArcMap	XML Editor
Sąrankos failo formatas	ArcXML	ArcXML	Dvejetainis ArcMap dokumentas .mxd arba .pmf	ArcXML
Klientai	„HTML Viewer“, Java žiūryklės „ArcMap“, „ArcPad“, „Custom HTML“ programa	Java žiūryklės, „ArcExplorer“, „ArcMap“	„HTML Viewer“, Java žiūryklės „ArcMap“, „ArcPad“	„Metadata Viewer“, „ArcCatalog“



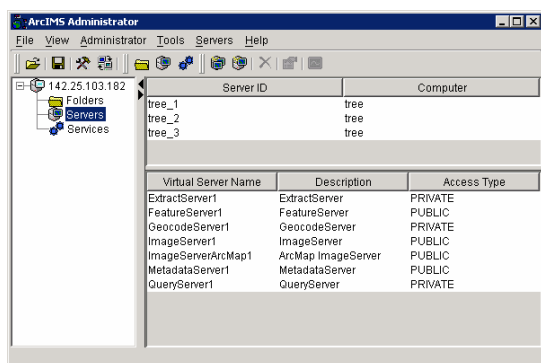
Pav. Žemėlapio failo kūrimo paslaugoms „Image Service“ ir „Feature Service“ procesai

Pav. Sąrankos failo kūrimo paslaugoms „ArcMap Image Service“ procesas



Pav. Metaduomenų sąrankos failų kūrimas paslaugai „Metadata Service“

- ArcIMS „**Administrator**“, kaip savarankiška taikomoji programa, arba „**Service Administrator**“, kaip interneto taikomoji programa, arba komandinės eilutės administravimas, gali būti naudojamas ArcIMS paslaugoms, serveriams ir aplankams kurti ir tvarkyti.



Pav. ArcIMS „Administrator“

„Administrator“ arba „Service Administrator“ kuria, paleidžia, sustabdo, įrašo ir pašalina paslaugas. Programa taip pat gali nustatyti ir keisti paslaugos savybes, pavyzdžiui, „Virtual Server, „Spatial Servers“ arba aplankų priskyrimo „Virtual Servers“. Pavyzdžiui, „Spatial Server“ procesai tvarkomi paleidžiant ar sustabdant „Monitor“ arba naudojant „Administrator“ ar „Service Administrator“.

Failas .axl naudojamas ArcIMS žemėlapių paslaugai sukurti. Nors ArcIMS paslauga ir failas .axl yra glaudžiai susiję, jie yra visiškai savarankiški. Vienas .axl gali apibrėžti kelias ArcIMS paslaugas, o sukūrus ArcIMS paslaugą, failą .axl galima ištrinti arba pakeisti, ir tai neturės jokios įtakos ArcIMS paslaugai.

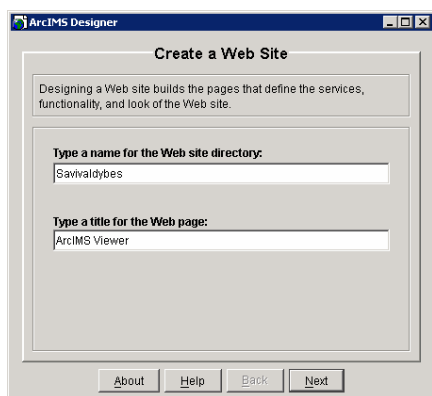


- ArcIMS „**Web Manager**“ ir „**Designer**“ galima naudoti kuriant tinklalapį, t. y. aplanką, kuriame yra HTML, DHTML ir „JavaScript“ failai. ArcIMS svetainėje gali būti teikiama viena arba kelios paslaugos, kurios buvo sukurtos naudojant „Administrator“. Joje taip pat gali būti legenda, žemėlapio navigacijos priemonės, mastelio skalė ir peržiūros žemėlapis.

ArcIMS „Designer“ – tai vediklio valdoma taikomoji programa, kuri padeda sukurti svetainės projektą. „Designer“ sukuria tris žiūrykles: „HTML Viewer“ standartinei naršyklei, kuri atitinka WMS, „Java Custom Viewer“ ir „Java Standard Viewer“ standartinei naršyklei su ESRI papildiniu, kuri atitinka WFS.

Naujajame ArcIMS 9.2 „Web Manager“ taip pat pateikiami išsamūs nurodymai, kurie padeda sukurti svetainę. Jis turi daugiau galimybių lyginant su „Designer“. Projektuojant galima pasirinkti ne tik „Image Service“, „Feature Service“, bet ir „ArcGIS Server“ paslaugas, „ArcWeb“ paslaugas ir WMS paslaugas. „Web Manager“ gali sujungti kelias vartotojo pasirinktas paslaugas. Be to, „Web Manager“ leidžia pasirinkti daugiau programos galimybių, pavyzdžiui, įterpti suradimo užduotį (adresų programa arba objektų paieška), įterpti tokius žemėlapio elementus kaip turinys arba peržiūros žemėlapis, antraštės puslapio elementus pritaikyti svetainei ir pan.

Interneto žemėlapių programa, sukurta naudojant „Web Manager“, naudoja .NET, „Java“ ir „AJAX“ technologiją, kuri apima daug naujų savybių – tolydų kadravimą (angl. *seamless panning*), mastelio keitimą pelės ratuku (angl. *scroll-wheel zoom*), klaviatūros navigaciją ir funkcionalų turinį su atkarpomis.



**Pav. ArcIMS „Designer“**

Be to, ArcIMS siūlo įvairių išteklių ir galimybę kurti svetaines naudojant šablonus. Todėl HTML žiūryklių šablonai „Web Sites“ automatiškai įdiejami su ESRI ArcIMS. Jie nustato kiekvieno naudotojo ArcIMS svetainės grafinę išvaizdą ir funkcionalumą. Žiūryklės naudojamos įvairiems svetainės grafiniams elementams – žemėlapiui, parankinei, legendai, peržiūros žemėlapiui ir k. t. Jeigu svetainei kurti naudojama programa „Web Manager“, „Web Mapping Application“ šabloną galima įkelti į „Web Manager“.

Toliau pateiktoje lentelėje apibendrinta, kaip naudojant ArcIMS „Manager“ taikomas programas sukurti interneto žemėlapių taikomųjų programų svetainę.

Lentelė. ArcIMS svetainės kūrimo žingsniai (<http://webhelp.esri.com/arcims/9.2/> – anglų kalba)

1	Map	Žemėlapis ir atitinkamas žemėlapių sąrankos failas, skirtas interneto taikomajai programai, kuriami naudojant „Author“, „ArcMap“ arba rankiniu būdu. Žemėlapių sąrankos failą galima optimizuoti, norint užtikrinti gerą veikimą.
2	Service	Naudojant „Administrator“ arba „Service Administrator“, sukurtą žemėlapių reikia publikuoti k paslaugą.
3	Web App	Pasitelkiant turimas paslaugas, interneto taikomųjų programų svetainė sukuriama naudojant „Designer“ arba „Web Manager“. Galima taikyti adaptavimą.
4	Tune	Siekiant, kad sukurtas svetainė atitiktų efektyvumo ir saugumo reikalavimus, ją galima derinti stebėti.

Norint kurti ir administruoti svetaines nuotoliniu būdu, ArcIMS tvarkytuves „Author“, „Designer“ ir „Administrator“ galima įdiegti kitame kompiuteryje, kuriame nėra ArcIMS. Be to, „Service Administrator“ programą galima naudoti keliems kompiuteriams administruoti nuotoliniu būdu. „Service Administrator“ bus naudojamas administruojant kelias „ArcIMS Application Server“ programas.

#### **5.4. ESRI ArcIMS komponentai: kliento pusė**

ArcIMS „Client Viewer“ yra ArcIMS komponentas, galintis teikti konkrečias užklausas serveriui ir gauti norimą atsakymą, pavyzdžiui, žemėlapio vaizdą arba užklausos lentelės rezultatus. Klientas taip pat turi žinoti, kaip elgtis su serverio ArcXML atsakymu (pvz., formatuoti ArcXML srautą į HTML požymių lentelę).

ArcIMS palaiko šiuos pagrindinius žiūryklių tipus:

- Lengvus „plonuosius“ klientus – HTML arba DHTML interneto naršykles (pvz., „Internet Explorer“ arba „Netscape“)
- „Storuosius“ klientus, kurie gali būti naudojami kaip savarankiškos programos (pvz., ESRI „ArcExplorer“) arba kaip interneto naršyklės programėlės (pvz., ESRI „Java Viewer“ programėlė, kurią į kliento kompiuterį galima išsiųsti tik vieną kartą po pirmojo atsisiuntimo iš interneto žemėlapių svetainės. Be to, kliento kompiuteryje turi būti įdiegta „Java Runtime Environment“ (JRE)).
- Užsakomuosius „storuosius“ klientus, parašytus COM reikalavimus atitinkančia kalba (pavyzdžiui, VB, .NET, J2EE ir pan.).
- „Storuosius“ klientus, pavyzdžiui, ArcGIS taikomasias programas („ArcMap“ ir „ArcPad“).

## 5.5. ESRI ArcIMS ryšio srutas

Kliento ir serverio „ArcIMS Server“ ryšys yra laisvas, todėl kiekvienas kliento prisijungimas atitinka vieną transakciją. Transakciją sudaro užklausa, siunčiama atitinkamam serveriui „ArcIMS Server“, ir klientui gražinamas atsakymas.

Toliau lentelėje ir paveiksle parodyta ArcIMS kliento ir serverio bendravimo seka:

Lentelė. ArcIMS kliento ir serverio ryšio ciklo etapai

Etapai	Veiksmai
2.	<p>ArcIMS klientas paruošia prašymą ArcIMS svetainei. Jis siunčiamas kaip užklausa paslaugai „ArcIMS Service“. Užklausų pavyzdžiai: naujo kitokio mastelio žemėlapio prašymas, geoobjekto požymių informacijos gražinimas, sluoksnio vaizdavimo pakeitimas, sluoksnių įjungimas ir išjungimas ir pan.</p> <p>Prašymą ArcIMS serveriui galima siųsti naudojant <code>POST</code> ir <code>GET</code> HTTP užklausų metodus.</p>
3.	<p>Interneto serveris gauna prašymą ir jį perduoda „Server Connector“ arba trečiosios šalies taikomųjų programų serveriui, kuris prašymą perduoda jungčiai.</p>
4.	<p>Jungtis atveria kelią „ArcIMS Application Server“ atsakyti, o prašymas iš jungties perduodamas į „Application Server“.</p>
5.	<p>„Application Server“ prašymą siunčia atitinkamam „Spatial Server“, priklausančiam „Virtual Server“ grupei. „ArcIMS Spatial Server“ gauna ArcXML dokumentą <code>&lt;REQUEST&gt;</code>, kuriame gali būti toliau išvardytos ArcIMS ir (arba) OGC užklausos (žr. ArcXML programuotojo žinyną, skirtą ArcIMS 9.2 <a href="http://webhelp.esri.com/arcims/9.2/general/">http://webhelp.esri.com/arcims/9.2/general/</a> – anglų kalba):</p> <ul style="list-style-type: none"> <li>▪ <code>GET_IMAGE</code></li> <li>▪ <code>GET_FEATURES</code></li> <li>▪ <code>GET_GEOCODE</code></li> <li>▪ <code>GET_EXTRACT</code></li> <li>▪ <code>GET_SERVICE_INFO</code></li> <li>▪ <code>GET_RASTER_INFO</code></li> <li>▪ <code>GET_LAYOUT</code></li> <li>▪ <code>GET_Project</code></li> <li>▪ <code>GET_METADATA</code></li> <li>▪ <code>PUBLISH_METADATA</code></li> <li>▪ <b>GetCapabilities</b></li> <li>▪ <b>GetMap</b></li> <li>▪ <b>GetFeatureInfo</b></li> <li>▪ <code>GetCapabilities</code></li> <li>▪ <code>DescribeFeatureType</code></li> <li>▪ <code>GetFeature</code></li> </ul> <p>Prašymas gali atrodyti taip:</p> <pre>&lt;?xml version="1.0" encoding="UTF-8" ?&gt; &lt;ARCXML version="1.1"&gt; &lt;REQUEST&gt;     &lt;GET_IMAGE&gt;         &lt;PROPERTIES&gt;             &lt;ENVELOPE minx="-125" miny="25" maxx="- 67" maxy="50" /&gt;         &lt;/PROPERTIES&gt;     &lt;/GET_IMAGE&gt; &lt;/REQUEST&gt;</pre>

6.	<p>&lt;/ARCXML&gt;</p> <p>„Spatial Server“ sukuria atsakymą, kurį sudaro XML eilutė (pvz., užklaustos rezultatai arba vaizdo vieta), jeigu prašoma vaizdo ar WMS, arba duomenų srautas, jeigu užklausa yra „Feature“ arba WFS.</p> <p>Kiekvieną ArcXML <i>užklaustos</i> tipą atitinka „ArcIMS Spatial Server“ <i>atsakymo</i> tipas. Pavyzdžiui:</p> <ul style="list-style-type: none"> <li>▪ GET_EXTRACT response is EXTRACT</li> <li>▪ GET_FEATURES response is FEATURES</li> <li>▪ GET_GEOCODE response is GEOCODE</li> <li>▪ GET_IMAGE response is IMAGE</li> <li>▪ Etc</li> </ul> <p>Jeigu siunčiama užklausa GET_IMAGE, „Spatial Server“ naudoja atitinkamo vaizdų serverio ArcXML &lt;CONFIG&gt; .axl dokumentą ir rezultatų kataloge sukuria žemėlapio vaizdą (pvz., gif formatu). Tada klientui išsiunčia ArcXML &lt;RESPONSE&gt; dokumentą su to vaizdo pavadinimu ir URL, per kurį klientas gali pasiekti vaizdą. Atsakymas gali atrodyti taip:</p> <pre>&lt;?xml version="1.0" encoding="UTF-8" ?&gt; &lt;ARCXML version="1.1"&gt;   &lt;RESPONSE&gt;     &lt;IMAGE&gt;      &lt;ENVELOPE minx="-87.5" miny="30.0" maxx="-59.5" maxy="50.0" /&gt;     &lt;OUTPUT url="http://tree.mala.bc.ca/output/ savivaldybes5681824311.jpg" /&gt;     &lt;/IMAGE&gt;   &lt;/RESPONSE&gt; &lt;/ARCXML&gt;</pre> <p>Toliau pateikiamas žemėlapio sąrankos failo pavyzdys. Šiame faile nustatomas pradinis žemėlapio vaizdo mastas, naudojamų duomenų žemėlapio vienetai, kelias iki „shape“ failo, po to – „shape“ failo vardas ir atvaizdavimo simbolika.</p> <pre>&lt;ARCXML version="1.0.1"&gt;   &lt;CONFIG&gt;     &lt;MAP&gt;       &lt;PROPERTIES&gt;         &lt;ENVELOPE minx="1205720.8" miny="21567.0" maxx="1585746.9" maxy="307341.899475" name="Initial_Extent"/&gt;         &lt;MAPUNITS units="METERS"/&gt;         &lt;BACKGROUND color="0,0,210" transcolor="255,255,255"/&gt;       &lt;/PROPERTIES&gt;     &lt;WORKSPACES&gt;       &lt;SHAPEWORKSPACE name="Transport" directory="//website\data\shapes "/&gt;     &lt;/WORKSPACES&gt;     &lt;LAYER type="featureclass" name="Roads" visible="true" id="8"&gt;       &lt;DATASET name="roads" type="line" workspace="Transport"/&gt;       &lt;SIMPLERENDERER&gt;         &lt;SIMPLELINESYMBOL width="2" color="156,50,235" type="solid" captive="round"         &lt;/SIMPLERENDERER&gt;     &lt;/LAYER&gt;   &lt;/MAP&gt; &lt;/CONFIG&gt; &lt;/ARCXML&gt;</pre>
----	--



## 5.6. ESRI ArcIMS diegimas

Kartais interneto žemėlapių serverio įdiegimas yra sudėtingas. Reikia įdiegti keletą komponentų, minėtų ankstesnėse paskaitos dalyse. Todėl ESRI parengė labai išsamią ne tik ESRI komponentų, bet ir trečiųjų šalių komponentų (pvz., interneto serverio su serverio programėlės modulių) įdiegimo instrukciją. Paprastai tinklalapis yra viename kompiuteryje, tačiau naudotojams vis tiek reikia pasirinkti geriausią svetainės sąrankos sprendimą, galbūt net įdiegti ArcIMS paskirstytojoje sistemoje.

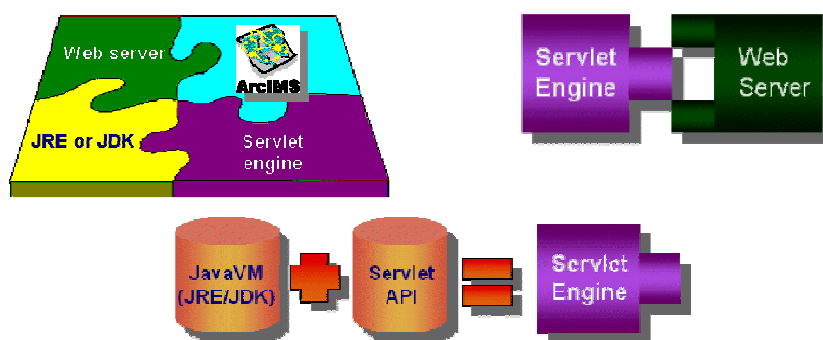
ESRI ArcIMS įdiegimo instrukcijoje daroma prielaida, kad veikia interneto serveris ir serverio programėlės modulis, naudotojas žino, kaip administruoti interneto serverį, sustabdyti ir paleisti paslaugas bei tarnybas, kurti virtualiuosius katalogus. Taip pat daroma prielaida, kad naudotojas išmano interneto technologijas ir žino susijusių terminiją.

ArcIMS įdiegti ir paleisti reikia atlikti penkis įdiegimo veiksmus. Šie veiksmai yra tokie:

1. Patikrinimas, ar svetainė atitinka ArcIMS sistemos reikalavimus.

ArcIMS galima įdiegti keliose operacinių sistemų platformose – „Windows“, „Sun Solaris“ ir pan.

ArcIMS reikalinga tokia sisteminė programinė įranga:



**Pav. Naudotojas turi pasirinkti ir įdiegti „Java 2 Platform Standard Edition“ platformos „Java Runtime Environment“ aplinką (J2SE JRE), interneto serverį „Web Server“ ir modulį „Servlet Engine“ (daugiau informacijos pateikta ESRI ArcIMS diegimo instrukcijoje).**

ESRI teikia informacijos apie tai, kaip savo svetainėje įdiegti „Web Server“ ir „Servlet Engine“ (pvz.,

<http://support.esri.com/index.cfm?fa=knowledgebase.techarticles.articleShow&d=31218> – anglų kalba).

ArcIMS „Spatial Server“ ir ArcIMS „ArcMap Server“ reikalingas ESRI prieigos failas.

2. ArcIMS svetainės sąrankos planavimas

Šis etapas yra sudėtingiausias, kadangi ArcIMS svetainė yra viename ar keliuose kompiuteriuose, taip pat yra ArcIMS pagrindinis kompiuteris (ArcIMS vartotojo sąsajos

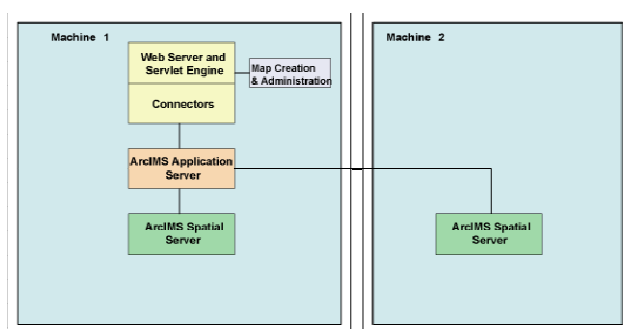
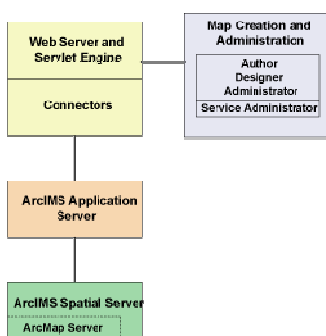
komponentai – „Author“, „Designer“, „Administrator“ ir „Service Administrator“, „Connectors“ ir „ArcIMS Application Server“), keli ArcIMS „Spatial Server“ ir duomenų bazės.

Yra daug galimų ArcIMS svetainių sąrankų. Kai kurių svetainių visi ArcIMS komponentai yra viename kompiuteryje, o kitas svetaines sudaro keletas kompiuterių. Sąranka priklauso nuo turimų kompiuterių išteklių bei reikalingo teikiamų žemėlapių paslaugų efektyvumo. Reikia atsakyti į keletą klausimų. Kiek paslaugų bus kuriama vienu metu? Kaip vyksta sudėtingos duomenų apdorojimo funkcijos? Kiek naudotojų kasdien galės aplankyti svetainę tuo pat metu?

ESRI teikia ArcIMS svetainės sąrankos planavimo rekomendacijas, pavyzdžiui:

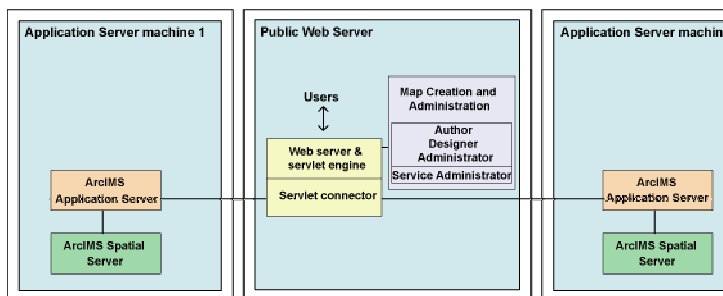
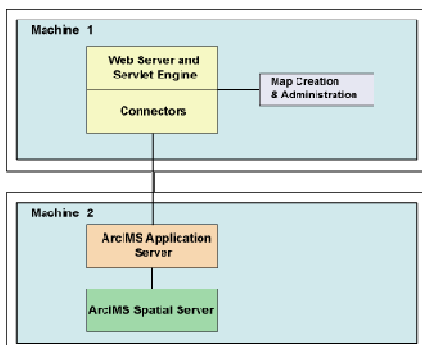
- „Connectors“ reikia įdiegti kompiuteryje, kuriame yra interneto serveris
- ArcIMS „Servlet Connector“ turi būti įdiegta tik kompiuteryje, kuriame yra interneto serveris
- „Author“, „Designer“ ir „Administrator“ galima įdiegti bet kuriame kompiuteryje
- „Spatial Server“ galima įdiegti viename arba keliuose kompiuteriuose
- „Application Server“ galima įdiegti viename arba keliuose kompiuteriuose

Toliau parodyti galimi įvairių paskirstytųjų įdiegimo komponentų sąrankos įvairiose kompiuterių platformose scenarijai. ArcIMS komponentai, kurie įdiegimo sąrankos diagramoje pavaizduoti ta pačia spalva, turi būti įdiegiami tame pačiame kompiuteryje (žr. ESRI ArcIMS diegimo instrukcija).



Pav. ArcIMS svetainė viename kompiuteryje

Pav. Du ArcIMS „Spatial Server“ keliuose kompiuteriuose



Pav. „Web Server“ skirtas kompiuteris

Pav. Keli ArcIMS „Application Server“

### 3. ArcIMS diegimas

ArcIMS 9.2 diegimas apima diegimo sąranką ir sąranką po diegimo.



- Diegiant programinę įrangą yra įdiegiamos pasirinktos funkcijos. Visas procesas išsamiai aprašytas ESRI ArcIMS diegimo instrukcijoje.
- Sąranka po diegimo yra konfigūravimo veiksmai, skirtas ArcIMS diegimui baigti. Atliekant konfigūraciją po diegimo, atsižvelgiant į pasirinktas funkcijas, galima parengti ArcIMS naudojimui, sukongūruoti ArcIMS, sukongūruoti „ArcSDE Services Monitor“, sukongūruoti interneto serverio ir serverio programėlės modelį ir sukongūruoti sistemas JRE.

Šiame etape įdiegiant kiekvieną „Spatial Server“ reikia nurodyti, kuriame kompiuteryje įdiegtas „Application Server“. Vienas „Spatial Server“ gali dirbti tik su vienu „Application Server“, tačiau vienas „Application Server“ gali dirbti su keliais „Spatial Server“.

Diegiant kiekvieną „Application Server“ reikia nurodyti interneto serverio vardą. Įdiegus interneto serverį kiekvienam „Application Server“ reikia nurodyti „Web Server“ vardą, galima redaguoti svetainės parinktį ir diagnostikos savybių failus.

#### 4. Interneto serverio konfigūravimas

Šiame etape reikia patikrinti „Web Server“ ir „Servlet Engine“ ir įsitikinti, kad programos tinkamai bendrauja tarpusavyje. Jų konfigūravimas gali būti automatinis arba rankinis.

#### 5. ArcIMS konfigūravimas

Šiame etape galima nustatyti ArcIMS aplinkos parametrus.

## 5.7. ESRI ArcIMS ir „ArcGIS Server“ adaptavimo galimybės

Vienas iš ArcIMS ir „ArcServer“ pranašumų yra tas, kad skirtingoms programinės įrangos platformoms numatytos kelios adaptavimo parinktys, įskaitant ir neprogramuojamas parinktis. Adaptuoti galima:

- Serverių svetaines ir šiuos jų komponentus:
  - Tinklapius: svetaines, kurios buvo sukurtos naudojant „Designer“ arba „Web Manager“, galima adaptuoti atliekant HTML, CSS, „JavaScript“ arba „VBScript“ pakeitimus. Tokiam adaptavimui gali nereikėti programavimo.

„ArcIMS“ naudoja žiūryklės šablonus ir apibrėžia, kaip galutiniai naudotojai matys svetaines bei kuriomis žemėlapių tyrinėjimo galimybėmis jie galės naudotis. ArcIMS numato dviejų rūšių žiūrykles: HTML žiūryklė naudojama ESRI vaizdų paslaugoms ir jai reikalingas tik interneto naršyklės palaikymas, o „Java“ žiūryklė leidžia siųsti objektų geometriją, atitinka ESRI objektų paslaugas ir jai reikalingas „storasis“ klientas. „HTML Viewer“ naudoja HTML, „JavaScript“ ir kai kurias dinamines HTML (DHTML). Žiūryklė naudojama įvairiems svetainės grafiniams elementams – žemėlapiui, parankinei, legendai, peržiūros žemėlapiui ir k. t. Adaptavimą galima atlikti serverio pusėje redaguojant HTML ir „JavaScript“ kodus (daugiau žr. [http://webhelp.esri.com/arcims/9.2/general/mergedProjects/Books/Customizing the HTML Viewer.pdf](http://webhelp.esri.com/arcims/9.2/general/mergedProjects/Books/Customizing%20the%20HTML%20Viewer.pdf) ir [http://webhelp.esri.com/arcims/9.2/general/mergedProjects/Books/Customizing the Java Viewer.pdf](http://webhelp.esri.com/arcims/9.2/general/mergedProjects/Books/Customizing%20the%20Java%20Viewer.pdf) – anglų kalba).

Jeigu svetainė buvo sukurta naudojant „Web Manager“ su .asp technologija, galima įterpti arba pašalinti sluoksnius ir užduotis, keisti žemėlapių ir puslapių elementus naudojant „Web Manager“ ([http://edndoc.esri.com/arcobjects/9.2/NET Server Doc/manager/applications/intro web apps mgr.htm](http://edndoc.esri.com/arcobjects/9.2/NET%20Server%20Doc/manager/applications/intro%20web%20apps%20mgr.htm) – anglų kalba). Be to, interneto žemėlapių taikomąją programą, sukurta naudojant „Web Manager“, galima atverti ir redaguoti integruotoje kūrimo aplinkoje (pvz., „MS Visual Studio .NET“). Taigi „Web Manager“ programa sukuriamas „ArcIMS Web ADF“ (taikomųjų programų kūrimo struktūroje) ir galima naudoti visas interneto valdymo priemones bei API, susijusias su ADF (pvz., ASP.NET valdymo priemones). Programuotojas gali keisti valdymo priemonių savybes arba parašyti naują kodą, jeigu reikalingas sudėtingesnis adaptavimas ([http://edndoc.esri.com/arcobjects/9.2/NET Server Doc/developer/ADF/extending manager web apps.htm](http://edndoc.esri.com/arcobjects/9.2/NET%20Server%20Doc/developer/ADF/extending%20manager%20web%20apps.htm) – anglų kalba).

Interneto taikomosioms programoms kurti „ArcGIS Server“ naudoja ArcGIS „Server Manager“. Interneto žemėlapių svetainę, sukurta naudojant „Manager“, galima adaptuoti keičiant interneto valdymo priemonių savybes bei atliekant paprastus HTML pakeitimus. Taip pat galima naudoti programavimą J2EE ir .NET taikomųjų programų programavimo sąsajose (API) (žr. <http://webhelp.esri.com/arcgisserver/9.2/> – anglų kalba). „ArcGIS Server“ pakete yra ADF su priemonėmis, skirtomis GIS interneto taikomosioms programoms kurti. Programos kūrėjas interneto ADF valdymo priemones gali pelenuti į interneto formą ir greitai sukurti taikomąją programą, kuri leidžia kurti žemėlapius, redaguoti, koduoti ir apdoroti

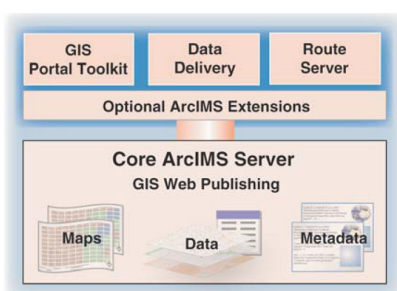
geografinius duomenis ir pan. Galima naudoti ADF „Designer“ bibliotekas. Interneto ADF taip pat leidžia keletą ArcIMS ir „ArcWeb Services“ paslaugų integruoti viename žemėlapyje, numato šių paslaugų rūšių programavimo pasitelkiant .NET arba „Java“ įėjimo taškus. Įdiegimo metu interneto ADF automatiškai integruojama į įdiegtą kūrimo aplinką, pavyzdžiui, „MS Visual Studio“, „Eclipse“ arba „Creator“.

- Jungtis: ArcIMS galima adaptuoti ir išplėsti naudojant standartines interneto kūrimo aplinkas, pavyzdžiui, JSP, ASP, .NET ir k. t. (žr. [http://webhelp.esri.com/arcims/9.2/general/mergedProjects/Books/Using\\_ActiveX\\_Connector.pdf](http://webhelp.esri.com/arcims/9.2/general/mergedProjects/Books/Using_ActiveX_Connector.pdf), [http://webhelp.esri.com/arcims/9.2/general/mergedProjects/Books/Using\\_ColdFusion\\_Connector.pdf](http://webhelp.esri.com/arcims/9.2/general/mergedProjects/Books/Using_ColdFusion_Connector.pdf) – anglų kalba)
- ArcXML sąrankos failą: ArcIMS klientai ir serveriai bendrauja naudodami ArcXML – standartinės XML kalbos ESRI išplėtimą. Keičiant ArcXML galima adaptuoti ArcIMS taikomas programas.
- Išorinį taikomųjų programų serverį (pvz., „ColdFusion“).
- „ArcGIS Server“ užtikrina „ArcObjects“ programinės įrangos serverio kūrimo aplinką, kurioje galima naudoti GIS serverio ArcGIS taikomas programas ir paslaugas. „ArcGIS Server“ užtikrina turtingą taikomųjų programų kūrimo struktūrą J2EE ir .NET aplinkose.
- Kliento pusės programos ir šiuos jų komponentus:
  - Specialias taikomas programas (pvz., „ArcMap“)
  - Interneto naršyklės programėlę, papildinį ir pan. (pvz., naudojant „Java“ programavimą galima adaptuoti ESRI „Java Viewer“ kliento programėlę).
- DBMS lygmenį.

## 5.8. ESRI ArcIMS papildomos savybės

### 5.8.1 Pasirenkami ESRI išplėtimai

GIS svetainės publikavimo internete galimybes galima išplėsti taikant įvairius pasirenkamus ArcIMS produktų išplėtumus. ESRI taip pat siūlo šiuos ArcIMS išplėtumus. „ArcIMS Data Delivery“ (<http://www.esri.com/software/arcgis/arcims-datadelivery/index.html> – anglų kalba) išplėtimą galima naudoti erdvinių duomenų publikavimui skirtingais erdviniais formatais, kurie naudojami GIS sistemose. „ArcIMS Route Server“ (<http://www.esri.com/software/arcgis/arcims-routeserver/index.html> – anglų kalba) išplėtimas ArcIMS tinklalapius papildo maršrutų kūrimo galimybę.



Pav. Pasirenkami ESRI išplėtimai

### 5.8.2 Saugumo klausimai

ArcIMS turi standartinių ir papildomų mechanizmų, skirtų duomenų ir svetainių apsaugai, pavyzdžiui:

- Žemėlapių paslaugų naudotojų tapatybės atpažinimas, leidžiantis nustatyti, kurie naudotojai arba klientai (pagal IP adresą) gali naudotis GIS duomenimis. Galima nustatyti papildomus apribojimus – iki kurios dienos ir kuriuo laiku naudotojas gali naudotis paslaugomis, konkrečių ArcXML elementų naudojimą ir pan.
- Saugus hiperteksto perdavimo protokolas ir saugus programinių jungčių sluoksnio protokolas svetainės saugumui valdyti.
- Keletas įtaisytųjų būdų riboti prašomų geobjektų skaičių, kai pateikiamos užklausos GET\_FEATURES.
- Atsakydamos į ArcMap kliento užklausas, ArcIMS vaizdų paslaugos gražina ribotą geometriją.
- „Servlet Connector“ savybių faile nustatomi parametrai, ribojantys užklausas, siunčiamas „Spatial Server“.

Be to, naudotojo tapatybės nustatymą, duomenų užšifravimą, prieigą prie aplankų galima riboti naudojant standartinius operacinių sistemų, interneto serverių, trečiųjų šalių taikomųjų programų serverių ir DBMS apsaugos mechanizmus. (Daugiau informacijos pateikta „ArcGIS Enterprise Security: Delivering Secure Solutions“, ESRI baltoji knyga, 2005 m. <http://www.esri.com/library/whitepapers/pdfs/arcgis-security.pdf> – anglų kalba). Papildomus apsaugos mechanizmus galima sukurti adaptuojant taikomųjų programų serverio jungtis.

### 5.8.3 ArcIMS metaduomenų paslauga

ArcIMS turi „Metadata Service“ metaduomenų paslaugos teikimo mechanizmą, kuris leidžia klientams teikti duomenis paslaugai ir atlikti paiešką paslaugoje. ArcIMS 9.2 versijoje yra interneto taikomoji programa „Metadata Explorer“. Ji įtraukta į ArcIMS interneto ADF, skirtą „Java“ platformai, ir leidžia naudotojams iš interneto naršyklės naršyti metaduomenų paslaugos turinyje. „Metadata Explorer“ panašus į sąsają, skirtą prieigai prie „Geography Network“.



Pav. „Geography Network“ sąsaja, skirta erdvinių duomenų ir paslaugų paieškai naudojant metaduomenis

### 5.8.4 ArcIMS derinimo patarimai

ESRI taip pat teikia keletą nurodymų, kaip užtikrinti veiksmingą svetainės veikimą. Akcentuojami šie aspektai:

- .axl failė nustatomas nuo mastelio priklausantis atvaizdavimas
- naudojama paprasta simbolika ir žymės
- erdviniai duomenys apibendrinami
- naudojami konkretūs vaizdų tipai ir dydžiai
- dirbama su žemėlapių projekcijomis
- sukuriama erdviniai indeksai

Su šiais ir kitais nurodymais galima susipažinti svetainėje <http://webhelp.esri.com/arcims/9.2/general/> (anglų kalba) ir (arba) peržiūrint nemokamą ESRI interneto mokomąjį kursą „Introduction to ArcIMS“ [http://campus.esri.com/acb2000/showdetl.cfm?&DID=6&Product\\_ID=833&CATID=84&CFID=4506767&CFTOKEN=37230417](http://campus.esri.com/acb2000/showdetl.cfm?&DID=6&Product_ID=833&CATID=84&CFID=4506767&CFTOKEN=37230417) (anglų kalba).

## 5.9. Išvados

ArcIMS leidžia dinامينius žemėlapius ir duomenis teikti internete, nesunkiai kurti taikomas programas, kurios naudoja geografinį turinį, kurti specialioms poreikiams pritaikytas taikomas programas naudojant standartines interneto kūrimo aplinkas, dalytis duomenimis ir kurti GIS portalus. ArcIMS užtikrina keičiamo dydžio struktūrą, skirtą interneto žemėlapiams publikuoti įmonių intranete ir pasauliniame internete. ArcIMS paslaugas gali naudoti įvairiausi klientai, įskaitant specialias interneto taikomas programas, „ArcGIS Desktop“, mobiliuosius ir bevielius įrenginius. ArcIMS taip pat galima adaptuoti. ArcIMS turi „Application Server“ jungtis, kurios užtikrina OGC WMS ir WFS specifikacijų reikalavimų tenkinimą.

Lyginant su ArcIMS, „ArcGIS Server“ palaiko įvairesnes GIS ir žemėlapių paslaugas. Šioms paslaugoms priskiriama: „ArcGIS Mapping“, OGC WMS, „Mobile Data Access“, KLM, „Geodata Access“ duomenų gavimas, „Access“ funkcija, „Geoprocessing“, „Network Analysis“, „Globe“ ir „Geocoding“ (žr. <http://webhelp.esri.com/arcgisserver/9.2/dotNet/> – anglų kalba). Šioms paslaugoms gali būti naudojama interneto naršyklė arba speciali taikomoji programa. ArcGIS taikomosios programos, pavyzdžiui, „ArcMap“ ir „ArcGlobe“, gali būti naudojamos ir kaip GIS paslaugų klientai.

Tačiau ArcIMS ir ArcGIS serveriai nėra nemokama programinė įranga. Jeigu naudotojui arba organizacijai nepakanka lėšų ArcIMS įsigyti, jie gali naudoti atvirąją arba atviros kodo programinę įrangą. Tokiu atveju galima naudoti toliau išvardytą programinę įrangą:

- „GeoServer“, <http://docs.codehaus.org/display/GEOS/Home> (atviros kodo programinė įranga)
- „MapServer“, <http://mapserver.gis.umn.edu/> (atviros kodo programinė įranga)
- „Autodesk“ programinę įrangą „MapGuide“, <http://usa.autodesk.com/adsk/servlet/index?id=7171990&siteID=123112> (atviros kodo programinė įranga)
- „OpenLayers, API: Javascript“, <http://openlayers.org/> (atviros kodo programinė įranga)
- kita

ESRI dokumentuose išsamiai apibūdinama ArcIMS architektūra ir jos komponentai. ESRI interneto žemėlapių programinės įrangos veikimo principus galima nagrinėti siekiant suprasti kitos interneto žemėlapių programinės įrangos, pavyzdžiui, atvirosios programinės įrangos, kurios pagalbos sistema paprastai yra ne tokia išsami, galimybes.

### **Paskaitos savarankiško mokymosi klausimai:**

1. Kokios yra pagrindinės ESRI „Application Server“ funkcijos?
2. Kuris ArcIMS komponentas atitinka žemėlapių modulį?
3. Kas yra ESRI „Virtual Server“ ir kodėl jis buvo sukurtas?
4. Kokios yra „Spatial Server“ ir „Application Server“ tarpusavio palaikymo ypatybės?
5. Kuo skiriasi ESRI paslaugos „Image Service“ ir „Feature Service“? Koks yra „Image Service“ sukuriamo vaizdo formatas?
6. Išnagrinėkite failą .axl <CONFIG>, pateiktą dalyje „ArcIMS ryšio srautas“ ir pamėginkite atsakyti į šiuos klausimus:
  - a. Kuri ArcXML versija naudojama?
  - b. Kur yra saugojami šio projekto duomenys?
  - c. Kurie žemėlapių vienetai yra naudojami?
7. Kodėl ArcIMS architektūra yra keičiamo dydžio?
8. Koks yra esminis ESRI ArcIMS ir „ArcGIS Server“ skirtumas?

### **Rekomenduojamieji skaitiniai (anglų kalba)**

- Section 4, 5 and 6, System Design Strategies An ESRI Technical Reference Document, 2007, <http://www.esri.com/library/whitepapers/pdfs/sysdesig.pdf>
- Introduction to ArcIMS at the [http://campus.esri.com/acb2000/showdetl.cfm?&DID=6&Product\\_ID=833&CATID=84&CFID=4506767&CFTOKEN=37230417](http://campus.esri.com/acb2000/showdetl.cfm?&DID=6&Product_ID=833&CATID=84&CFID=4506767&CFTOKEN=37230417), free online ESRI virtual campus course

## Nuorodos (anglų kalba)

- ArcIMS Online Main Help <http://webhelp.esri.com/arcims/9.2/general/> or <http://support.esri.com/index.cfm?fa=knowledgebase.webHelp.arcIMSGateway>
- The ArcGIS Server Web Help, <http://support.esri.com/index.cfm?fa=knowledgebase.webHelp.agServer>
- Getting Started with ArcIMS, <http://downloads.esri.com/support/documentation/ims /1010Getting Started with ArcIMS.pdf>
- ArcXML Programmer's Reference Guide for ArcIMS 9.2, <http://webhelp.esri.com/arcims/9.2/general/>
- System Design Strategies An ESRI ® Technical Reference Document, 2007, <http://www.esri.com/library/whitepapers/pdfs/sysdesig.pdf>
- ArcGIS Enterprise Security: Delivering Secure Solutions, An ESRI White Paper, 2005, <http://www.esri.com/library/whitepapers/pdfs/arcgis-security.pdf>



## Vartojamieji terminai

- ArcIMS
- „ArcGIS Server“
- „Web Server“
- „Application Server Connector“
- „Spatial Server“
- Paslaugos
- „Virtual Server“
- Egzempliorius
- Gija
- „Application Server“
- „Monitor“
- „Tasker“
- „Author“
- „Administrator“
- „Service Administrator“
- „Web Manager“
- „Designer“
- „Client Viewer“
- AXL sąrankos failas
- „Java 2 Platform Standard Editon“ platformos „Java Runtime Environment“ (J2SE JRE)
- „Servlet Engine“
- Taikomųjų programų kūrimo struktūra (ADF)
- „Java Viewer“
- „HTML Viewer“