

**ŠIAULIŲ UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA**

Mindaugas GEDVILAS
Informatikos magistro studijų programos studentas

**IMITACINIO MODELIAVIMO
PRITAIKYMAS NUOTOLINIAM
KOMPIUTERIŲ TINKLŲ MOKYMIUI BEI
ŽINIŲ PATIKRINIMUI**

**(SIMULATION ADAPTATION FOR COMPUTER NETWORK E-LEARNING AND
KNOWLEDGE EXAMINATION)**

MAGISTRO BAIGIAMASIS DARBAS

Darbo vadovas:
Prof. L. Sakalauskas

Šiauliai, 2008

TURINYS

I. IMITACINIO MODELIAVIMO, NUOTOLINIO MOKYMO BEI KOMPIUTERINIŲ TINKLŲ TEORINIAI ASPEKTAI.....	6
1.1. Imitacinio modeliavimo esmė	6
1.1.1. Imitacinio modeliavimo aplinka DESMO-J.....	8
1.1.2. Imitacinio modeliavimo įrankis AnyLogic.....	9
1.1.3. Imitacinio modeliavimo įrankis OMNeT++.....	11
1.2. Nuotolinio mokymosi (e-mokymosi) sistemos.....	12
1.3. Pagrindiniai kompiuterių tinklų aspektai.....	16
1.3.1. Duomenų bloko - paketo koncepcija.....	16
1.3.2. Kompiuterių tinklų topologijos.....	17
1.3.3. Tinklo sąsajos įrenginiai.....	20
II. PROJEKTINĖ DALIS.....	23
2.1. Darbinės srities modelis.....	23
2.2. Įrankių ir priemonių pasirinkimas, kuriant sistemą NetLab.....	24
2.3. Projekto NetLab eigos aprašymas.....	26
2.4. Projekto NetLab sistemos aplinka.....	28
2.5. Projekto NetLab struktūra.....	35
IŠVADOS.....	37
ANOTACIJA.....	39
LITERATŪRA.....	40
PRIEDAI.....	41

ILIUSTRACIJOS

1 pav. DESMO-J modelis vaizduoja sunkvežimių simuliaciją konteinerių terminale.....	9
2 pav. Sant Peterburgo metro	10
3 pav. Telefonų stočių tinklas.....	10
4 pav. Traukinių stotis.....	10
5 pav. Oro apsaugos sistemos.....	10
6 pav. TokenRing tinklas.....	12
7 pav. Maršrutų parinkimas tinkle.....	12
8 pav. Pagrindinės protokolo TCP/IP operacijos	17
9 pav. Žvaigždinis tinklas ir jo veikimas panaudojus koncentratorių.....	18
10 pav. Žvaigždinis tinklas ir jo veikimas panaudojus komutatorių.....	18
11 pav. Magistralinis tinklas su įvykusia kolizija.....	19
12 pav. Žiedinis tinklas.....	19
13 pav. Apibendrinta maršrutizatoriaus architektūra.....	22
14 pav. Darbinės srities modelis (schema).....	23
15 pav. Užregistruotų Moodle sistemų skaičius 2003-2008 metais.....	24
16 pav. Populiariausios duomenų bazių valdymo sistemos tarp IT specialistų.....	25
17 pav. Darbų atlikimo (eigos) diagrama.....	26
18 pav. Apibendrintas projekto vaizdas.....	27
19 pav. Tinklo objekto kortelė.....	29
20 pav. Tinklo objekto savybės kortelė.....	29
21 pav. Skriptų rašymo aplinka.....	30
22 pav. Objekto savybių keitimo langas.....	32
23 pav. Kabelio savybių keitimo langas.....	32
24 pav. Paketų generavimo nustatymo langas.....	32
25 pav. Eksperimento vykdymo langas.....	32
26 pav. Modeliavimo aplinkos darbo vaizdas.....	33
27 pav. Modeliavimo aplinkos darbo vaizdas.....	34
28 pav. Modeliavimo aplinkos darbo vaizdas.....	34
29 pav. Gijų ryšys su kitais tinklo objektais.....	35

IVADAS

Šiuolaikinėje informacinėje visuomenėje žinios, informacija ir įgūdžiai yra ypač vertingi bei reikalingi. Tačiau dėl nuolat didėjančio gyvenimo tempo ne visada atsiranda galimybių bei laiko lankyti tam tikras mokymo įstaigas. Puiki to išeitis – nuotolinis mokymasis. Nuotolinis mokymasis, įgalina suderinti galimybę tobulėti su patogumu, kadangi įmanoma studijuoti savarankiškai, patiems pasirinkus laiką bei vietą. Nuotolinis mokymasis suteikia galimybę mokytis dirbantiems žmonėms, neįgaliesiems, moksleiviams, studentams ir kt. Pasitelkus šiuolaikines technologijas ir komunikavimo priemones, t.y. elektroninį pašta, diskusijų forumus internete, pokalbius internete realiame laike, virtualią mokymosi terpę, garso ir vaizdo konferencijas tai padaryti visai nesudėtinga. Pastaruoju metu nuotolinio mokymosi metodai vystosi ir diegiami ypatingai sparčiai, todėl verta šią sritį dar labiau plėsti bei tobulinti, siekiant mokymosi procesą padaryti prieinamesnį bei paprastesnį.

Aktualumas. Gerinant studijų kokybę, diegiant naujas technologijas bei mokymosi būdus, tobulinamas ir pats studijų procesas. Spartėjanti informacinė plėtra paveikia ir šios srities specialistus, informatikos studentus bei dėstytojus ir kt. Studijuojant vieną iš informatikos krypčių, t.y. kompiuterių tinklus, matyti, jog šiai sričiai reikalingas vaizdas, eksperimentinis mokymas, vykdomas nuotoliniu būdu. Dėl to studijų procesas būtų spartesnis bei efektyvesnis. Tai gali būti pasiekta pritaikius naujausias technologijas. Šiuo tikslu yra kuriamos įvairios nuotolinio mokymosi sistemos, leidžiančios mokytis per atstumą. Todėl svarbu nuolat tobulinti šį mokymosi būdą, kurti naujas bei geresnes sistemas, jas nuolat atnaujinti bei papildyti naujomis funkcijomis.

Magistro darbo **tikslas** – sukurti imitacinio modeliavimo priemonę, skirtą nuotoliniam kompiuterių tinklų mokymui bei žinių patikrinimui.

Darbo tikslui pasiekti išskelti tokie **uždaviniai**:

- Atlikti nuotolinio mokymo, imitacinio modeliavimo bei kompiuterių tinklų analizę teoriniu aspektu.
- Pasirinkti reikiamas priemones nuotolinio mokymosi sistemai NetLab sukurti.
- Integruoti kuriamą sistemą NetLab į Moodle nuotolinio mokymosi sistemą.
- Aptarti sukurtos sistemos NetLab veikimo principus bei panaudojimo galimybes.

Darbo **objektas** – imitacinis modeliavimas ir nuotolinis mokymasis.

Rengiant magistro darbą panaudota įvairi mokslinė literatūra, straipsniai bei internetinių šaltinių medžiaga, cituojant bei referuojant autorių mintis ir teiginius. Kuriant sistemą NetLab panaudota Java programavimo kalba, Eclipse aplinka, duomenų bazių valdymo sistema MySQL, skriptavimo kalba JavaSkript. Taip pat projektui įgyvendinti pasitelktas lygiagretusis programavimas, gijų sinchronizavimas, skriptų integravimas ir kt. Integruojant NetLab sistemą į Moodle sistemą, panaudota Java Web Start technologija bei PHP skriptavimo kalba. Sukurta sistema buvo analizuojama bei pateikiami apibendrinimai.

Sukurta NetLab sistema gali būti panaudojama praktiškai, integravus ją į Moodle sistemą. Šios sistemos dėka, NetLab technologija gali būti pritaikoma nuotoliniam kompiuterių tinklų mokymui bei žinių patikrinimui. NetLab sistema yra surišta su imitaciniu modeliavimu ir kompiuterių tinklų mokymu. Atlikti sukurtos sistemos bandymai parodė, jog šis produktas gali būti taikomas mokymo tikslams dėl savo vaizdaus bei lankstaus panaudojimo.

I. IMITACINIO MODELIAVIMO, NUOTOLINIO MOKYMO BEI KOMPIUTERINIŲ TINKLŲ TEORINIAI ASPEKTAI

Siekiant išsamiai ištirti imitacinio modeliavimo pritaikymą nuotoliniam kompiuterių tinklų mokymui bei žinių patikrinimui, visų pirma reikia aptarti šios srities teorinius aspektus, suvokti jų esmę. Įgyvendinant šį projektą, svarbu susipažinti su imitaciniu modeliavimu, įvairiais jo pritaikymo ir panaudojimo aspektais, taip pat įsigilinti į nuotolinio mokymo bei kompiuterių tinklų teorinius niuansus, tam, kad tinkamai būtų įgyvendintas šis projektas ir pagrįstas imitacinio modeliavimo pritaikomumas nuotoliniam mokymui.

1.1. Imitacinio modeliavimo esmė

Imitacinis modeliavimas gali būti pritaikomas daugelyje sričių (pvz., tam tikri veiksmų planai, gamybos valdymo modelis, statomo namo fizinis modelis ir daugelis kitų). Naudojant imitacinį modeliavimą kur kas lengviau priimti tam tikrus sprendimus, atlikti analizę, įvairius bandymus, tyrimus ir panašiai. Taigi imitacinio modeliavimo pritaikomumas ir panaudojimas platus, todėl kalbant apie jį, verta išsamiau ištirti jo esmę ir savybes. Išsamesnis imitacinio modeliavimo apibrėžimas pateiktas kitoje pastraipoje.

Imitacinis modeliavimas – tai analizės, prognozavimo ir sprendimų priėmimo būdas, kuris, lyginant su tradiciniais statiniais ir analitiniais modeliavimo būdais, yra daug išsamesnis ir tikslesnis, leidžiantis detaliau įvertinti modeliuojamą sistemą ir išorinius veiksnius. Imitacinis modeliavimas leidžia aprėpti atsitiktinę daugelio reiškinių prigimtį, jų neapibrėžtumą ir sudėtingą tarpusavio sąveiką realiame pasaulyje. [5]

Daugeliu atveju imitacinis modeliavimas yra vienintelis būdas rasti teisingą problemos sprendimą. Tuo pačiu imitacija leidžia paprastai ir greitai išmėginti kuriamą sistemą, ją vizualizuoti ir patikrinti jos elgesį esant įvairiausioms besikeičiančiomis aplinkybėms, net realiai neturint tos sistemos fizinio prototipo, taigi ir visai nerizikuojant. Modelis veikia tik taip, kaip jam leidžia įvestos matematinės ir fizikos formulės, todėl tokia modeliavimo sistema gali būti ir virtualiosios gamybos įrankiu ir mokymo priemone.[5] .

Taigi, kaip buvo minėta, imitacinio modeliavimo pritaikomumas labai platus. Sukūrus atitinkamą modelį, jį galima panaudoti ne tik įvairiuose tyrimuose, atliekant prognozes, priimant sprendimus, bet ir mokyme. Šios sistemos panaudojimas priklauso nuo tam tikrų imitacinio modeliavimo priemonių.

Imitacinio modeliavimo priemonių yra daug ir įvairių. Vienos (bibliotekos pobūdžio) skirtos programuotojams, kuriems padeda greičiau kurti imitacinio modeliavimo sistemas ir kurie turi didelę interpretavimo laisvę. Kitos (programinio pobūdžio) skirtos pvz., analitikams, tyrėjams, mokiniams ir kt., kurie naudoja jau sukurtą imitacinio modeliavimo aplinką ir jiems tereikia su sukurtais įrankiais kurti norimus modelius ir analizuoti gautus rezultatus. Tokių priemonių yra daug ir įvairių.

Keletas imitacinio modeliavimo priemonių:

1. Imitacinis modeliavimas pasinaudojus sukurtomis bibliotekomis:

- *C++SIM* – C++ imitacinio modeliavimo paketas, skirtas diskretinių įvykių sistemoms analizuoti.
- *JavaSIM* – Java versija C++SIM;
- *DESMO-J* – paketas skirtas diskretinių įvykių modeliavimui ir imitavimui, orientuotas į programuotojus kuriant imitacinius modelius Java aplinkoje;

2. Daugiafunkcinis imitacinis modeliavimas:

- *AnyLogic* - imitacinio modeliavimo programinė įranga, sukurta su Java kalba ir integruota Elipse aplinkoje;

3. Imitacinis modeliavimas orientuotas į kompiuterių tinklus:

- *OMNeT++* – diskretinių įvykių imitavimo aplinka sukurta su C+ kalba;
- *ns2* – atviro kodo diskretinių įvykių stimulatorius, pritaikytas kompiuterių tinklų analizei. Ši aplinka parašyta C++ kalba ir kompiliuojama Unix ir Windows platformose;
- *NetSim* – skirtas mokomiems tikslams; kompiuterių tinklų imitacinio modeliavimo sistema;
- *Ir kt.*

Siekiant plačiau susipažinti su įvairiomis imitacinio modeliavimo priemonėmis, pateiktos trys iš jų (skirtingo pobūdžio), taip norint išsamiau apibrėžti jų paskirtį, panaudojimo galimybes.

1.1.1. Imitacinio modeliavimo aplinka DESMO-J

Pirmoji pasirinkta imitacinio modeliavimo priemonė – tai bibliotekų paketas DESMO-J. Siekiama nusakyti jo funkcijas, panaudojimo galimybes bei kam šis įrankis skirtas.

DESMO-J yra objektiškai orientuota aplinka, skirta programuotojams kurti imitacinius modelius. Akronimas „DESMO-J“ - šifruojamas "Discrete-Event Simulation and Modelling in Java".[4]

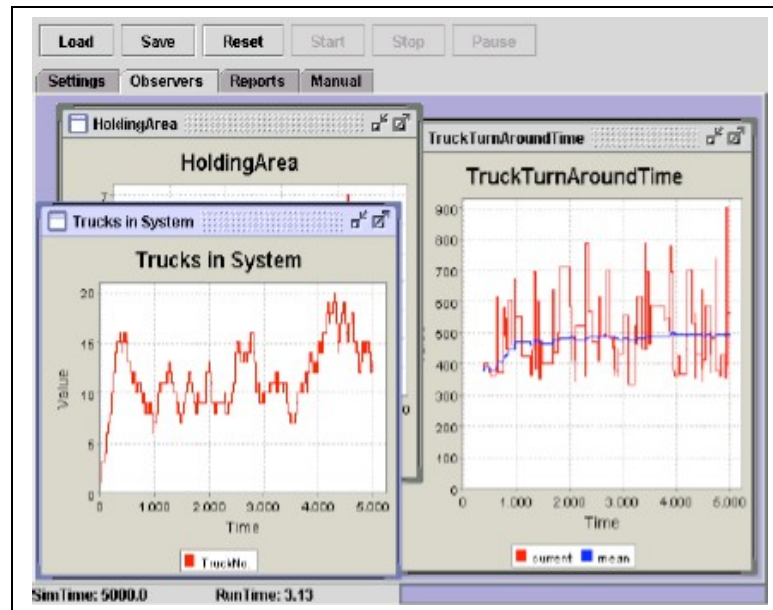
DESMO-J projektas jau daugelį metų yra aktyviai tobulinamas Hamburgo universiteto. Vienas iš svarbiausių šios idėjos motyvų buvo tinkamo įrankio poreikis, mokant studentus kurti diskretinių įvykių modelius. Nors DESMO-J negali būti vertinamas kaip e-mokymo griežtas komponentas, jis yra pripažintas labai naudinga simuliacijos tobulinimo įrankiu. Čia pagrindinė imitacinio modeliavimo koncepcija turi remtis praktiniu darbu modeliuojant ir įgyvendinant įvairius modelius [4].

DESMO-J yra paremtas Java. DESMO-J aplinka turi kai kurių techninių privalumų naudojant Java kaip simuliacijos įrankių bazę, nes sukurti imitaciniai modeliai pasidaro nepriklausomi nuo operacinės sistemos ir lengvai (tinklo pagalba) yra pasiekiami nuotoliniu būdu. DESMO-J siūlo visapusę simuliacijos aplinką, o ne paprastą simuliacijos kalbą [4].

Norint pradėti kurti sugalvotus imitacinius modelius, studentai turi suprasti DESMO-J branduolio pagrindines koncepcijas ir sistemos architektūrą. Čia jie susiduria su diskretinių įvykių imitacinio modeliavimo svarbiausiomis idėjomis: įvykių sąrašais, įvykių klasėmis ir metodais, lygiagrečių įvykių naudojimu, skirtingomis statistikos rūšimis, modelio naudojimu ir pan [6].

DESMO-J mokymo idėja yra talpinti įvairius modelius interneto laboratorijose. Prisijungęs studentas pagal savo kursą gali matyti modelių medį, jam skirtas užduotis, paaiškinimus ir t.t. Vartotojas priklausomai nuo jam suteiktų teisių gali administruoti visus šiuos modelius, ir kitą svarbią informaciją. Tam, kad pavaizduoti kaip veikia DESMO-J sukurtas imitacinis modelis, pateiktas šis pavyzdys.

DESMO-J imitacinio modelio pavyzdys. Sukurti imitaciniai modeliai leidžia konfigūruoti modelio parametrus, vykdyti modelio imitacijas ir tiesiogiai iš tinklo puslapio analizuoti rezultatus. Tokie modeliai gali būti įgyvendinti DESMO-J aplinkoje. 1 paveiksle parodyti konteinerių terminale kraunamų sunkvežimių įvykdytos imitacijos rezultatai.



1 pav. DESMO-J modelis vaizduoja sunkvežimių simuliaciją konteinerių terminale
 Šaltinis: A Framework for Web-based E-Learning of Discrete Event Simulation Concepts [4]

Įdėjus DESMO-J modelius į interneto puslapį, puslapio lankytojai gali keisti imitacinio modelio parametrus ir gauti informaciją apie modelio būseną (žr. 1 pav.). Nėra jokių tiesioginių modelių kodo pakeitimų. Tai yra svarbi prielaida, vykdant eksperimentus su modelių prieinamumu didelei vartotojų grupei.

1.1.2. Imitacinio modeliavimo įrankis AnyLogic

AnyLogic – naujos kartos imitacinio modeliavimo programinė įranga, paremta pastarojo dešimtmečio modeliavimo teorijos bei informacinių technologijų pasiekimais [5].

Lyginant su tradiciniais modeliavimo instrumentais, AnyLogic leidžia tyrėjams gauti daug tikslesnius ir išsamesnius rezultatus, tuo pačiu įdedant daug mažiau pastangų kuriant ir tobulinant kompiuterinį modelį. AnyLogic sistemos vartotojai gali:

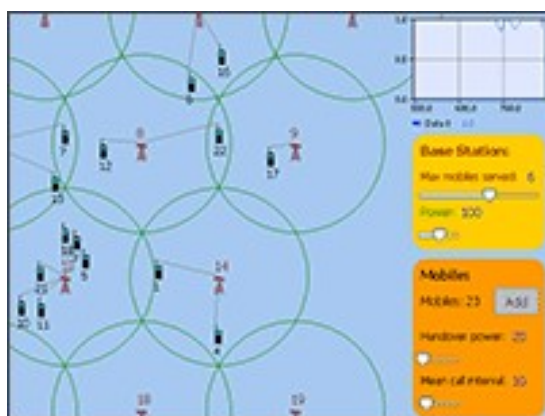
- modeliuoti naudojant vizualius, išplečiamus daugkartinio panaudojimo aktyvius objektus, tiek standartinius, pateikiamus su programa, tiek paties sukurtus naudojant Java programavimo kalbą;
- modeliuoti daug tiksliau, įtraukiant daugiau veiksmų ir sąlygų, juos apjungiant bei derinant ir taip pritaikant konkrečiai užduočiai spręsti;
- pailginti modelio naudojimo trukmę, nes AnyLogic suteikia galimybę jį koreguoti, plėsti ir keisti, derinant prie besikeičiančių sąlygų;
- panaudoti efektyvius programoje esančius analizės ir optimizavimo įrankius;

- paprastai ir efektyviai integruoti AnyLogic atviros architektūros modelius su kita programine įranga - elektroninėmis lentelėmis, duomenų bazėmis, ryšių su klientais ir projektų valdymo sistemomis ir k.t.;
- vaizdžiai pademonstruoti modelį ir jo darbo rezultatus, sukuriant jo interaktyvią animaciją, ją publikuoti internete.[5]

Žemiau pateikiami demonstraciniai AnyLogic pavyzdžiai (žr. 2,3,4 ir 5 paveikslus), tam, kad būtų galima geriau įsivaizduoti šio produkto panaudojimo galimybes.



2 pav. Sant Peterburgo metro



3 pav. Telefonų stočių tinklas



4 pav. Traukinių stotis



5 pav. Oro apsaugos sistemos

Šaltinis: XJ Technologies. Simulation Software and servines [13]

„Sant Peterburgo metro“ pavyzdys imituoja traukinių judėjimą ir jų apkrovimus piko metu bei kitu laiko momentu. Šio modelio pagalba galima padaryti išvadas, ar traukinių skaičius pakankamas atitinkamu laiko momentu, ar pakankamai išvystytas traukinių maršrutų tinklas, ar užtektinai yra traukinių stočių.

„Telefonų stočių“ modelis imituoja žmonių judėjimą ir telefonų prieraišumą atitinkamai telefonų stočiai. Galima didinti ar mažinti stočių signalo stiprumą, apkrovimą ir žiūrėti kaip telefonų tinklas susitvarko su savo darbu.

„Traukinių stoties“ pavyzdys parodo kaip sistema susitvarko su traukinių takelių paskirstymu. Šiame modelyje galime stebėti traukinių atvykimo ir išvykimo laikus.

„Oro apsaugos sistemos“ modelis imituoja miestelio puolimą iš oro. Šio modelio pagalba galima padidinti arba pamažinti puolimo intensyvumą, nustatyti radarų ir apsaugos sistemų parametrus, taip išsiaiškinant, ar apsaugos sistemos susitvarkys su joms pateiktomis užduotimis.

1.1.3. Imitacinio modeliavimo įrankis OMNeT++

OMNeT++ yra diskretinių įvykių imitavimo aplinka sukurta su C+ kalba. Andras Varga yra pagrindinis autorius ir puoselėtojas.

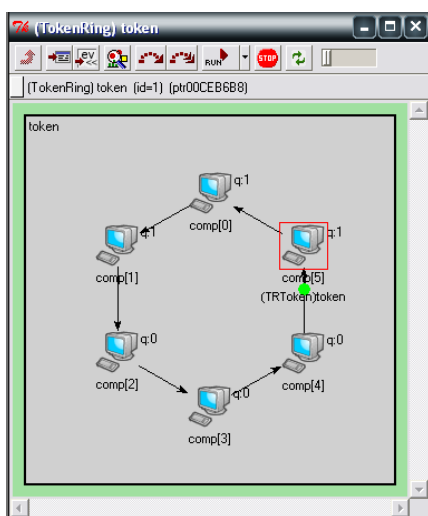
Tai pirmiausiai yra programa, skirta kurti kompiuterių tinklų komunikavimo imitacijas, tačiau šis įrankis yra pakankamai apibendrintas ir lankstus, taigi jį galima sėkmingai naudoti ir kitose srityse kaip sudėtingų IT sistemų, eilių tinklų arba aparatinės įrangos imitavimui [14].

OMNeT++ palaiko komponentinę architektūrą, kuri leidžia programuoti komponentus C++ kalba. Šie komponentai sudaro kitus didelius komponentus, taip plečiant programos galimybes ir kartu imitacinio modeliavimo lankstumą. OMNeT++ turi platų grafinį vartotojo interfeisą (GUI) palaikymą ir suteikia vartotojui galimybę įtraukti modelius į kitas sistemas [14].

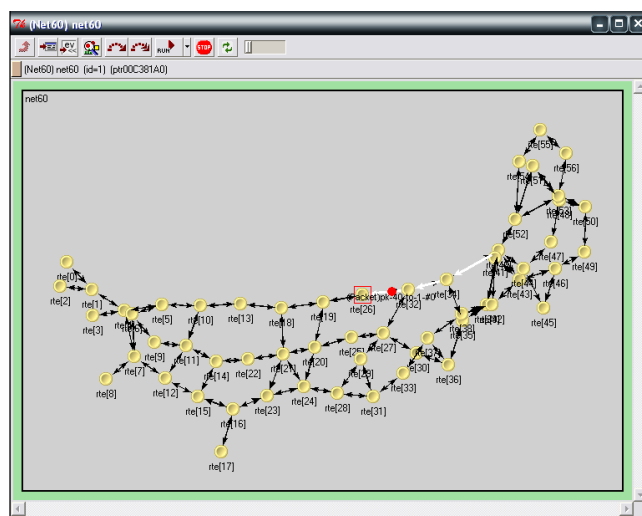
Šios aplinkos sudedamosios dalys yra:

- imitacijos bibliotekos branduolys;
- topologijos apibūdinančios kalbos kompiliatorius (nedc);
- tinklo redaktorius (GNED);
- grafinė aplinka imitaciniam modeliavimui vykdyti (Tkenv);
- komandinės eilutės vartotojo aplinka imitacinio modeliavimo vykdymui (Cmdenv);
- grafinės išvesties diagramos kūrimo įrankis (Plove);
- stulpelinės diagramos kūrimo įrankis (Scalars);
- modelio dokumentacijos kūrimo įrankis (opp_neddoc);
- atsitiktinių skaičių generavimo įrankis;
- dokumentacija, imitacinių modelių pavyzdžiai ir t.t.

6 ir 7 paveiksluose kompiuterių tinklų imitaciniai modeliai, sukurti OMNeT++ aplinkoje. Šie modeliai skirti imituoti TokenRing tinklui bei maršrutų parinkimui tinkle.



6 pav. TokenRing tinklas



7 pav. Maršrutų parinkimas tinkle

Šaltinis: paimta iš OMNeT++ pateikiamų pavyzdžių

TokenRing tinklo pavyzdyje yra imituojamas loginis žiedo veikimas. Čia vaizdžiai parodoma, kaip žymė keliauja tinklu ir perduoda leidimus siusti informaciją. Gavęs šią žymę kompiuteris, persiunčia informaciją nurodytam gavėjui. Perdavus visus duomenis, žymė siunčiama žiedo kaimynui. Šiame pavyzdyje tai pat matome, kokios eilės susidaro pvz., keičiant kompiuterių skaičių tinkle ir panašiai.

Antrame pavyzdyje demonstruojamas statinis trumpiausio kelio parinkimas. Maršrutų lentelės yra užpildomos prieš pradėdant imitavimą. Čia kiekvienas mazgas daro užklausimus ir siunčia atsitiktiniais intervalais paketus kitiems tinklo mazgams, parinkdami iš maršrutų lentelės trumpiausią kelią prie kitų tinklo mazgų, taip pat žiūrėdami per kiek mazgų yra nutolęs ieškomas mazgas.

1.2. Nuotolinio mokymosi (e-mokymosi) sistemos

Laikui bėgant keičiasi ne tik mokymo metodika, bet ir mokymo procese naudojamos techninės priemonės.

Iš pradžių mokymui buvo ir tebėra naudojami: balsas, gestai, primityvios aplinkos objektai ir kt. Vėliau, laikui bėgant atsirado kreida, skaidrės, televizija, radijas ir galiausiai

kompiuteriai bei jų tinklai (internetas). Būtent su kompiuteriais bei tinklais (internetu) nuotolinis mokymas ir susijęs.

E-mokymas yra internetinis mokymas. Jo komponentai gali apimti turinio pristatymą įvairiais formatais, mokymo žinių valdymą ir besimokančiųjų administravimą. E-mokymas teikia greitesnį mokymąsi, mažesniais kaštais, padidina mokymosi galimybę ir aišku, visų mokymo procesų dalyvių atsakingumą [4].

E-mokymas naudoja elektroninę multi-mediją, kad patobulintų mokymą ir mokymąsi. Vaizdus mokymas, naudojant virtualias laboratorijas, įvairius Java imitacinius modelius, simuliacijas, padeda studentams, moksleiviams ir kitiems besidomintiems geriau suprasti nagrinėjamą sritį ir jos praktinį problemų sprendimą. Modeliai ir jų vaizdavimas padeda paaiškinti procesų kompleksą, iliustruoja abstrakčias žinias, parodo jungtis ar abipusę priklausomybę, skatina eksperimentinį mokymąsi ir ugdo žinių pritaikymą tam tikroje sferoje.[4]

Mokymui nuotoliniu būdu, naudojant kompiuterių tinklus reikalinga speciali programinė įranga - nuotolinio mokymosi aplinka (NMA), dar kitaip vadinama nuotolinio mokymosi sistema (NMS). Šiuo metu plačiai pasaulyje žinomos ir Lietuvos aukštosiose mokyklose dažniausiai naudojamos šios nuotolinio mokymosi aplinkos:

- Lotus LearningSpace;
- WebCT;
- FirstClass;
- Moodle [1].

Nuotolinis mokymasis šiuo metu labai paplitęs ir jo panaudojimo galimybės yra labai didelės. Dažniausiai išskirtinos šios:

- Mokyklose;
- Aukštesniosiose m-klose;
- Universitetuose;
- Valstybės įstaigose;
- Verslo įstaigose;
- Prekybos asociacijose;
- Ligoninėse;
- Bibliotekose;
- Įdarbinimo agentūrose.

Plačiau aptariamos trys labiausiai Lietuvoje paplitusios nuotolinio mokymosi sistemos, t.y. Lotus LearningSpace, WebCT bei Moodle.

Moodle sistema. Moodle (Modular Object-Oriented Dynamic Learning Environment) – modulinė, objektinė nuotolinio mokymosi aplinka. Tai „atviro kodo“ nuotolinio mokymosi sistema, kuriama ir tobulinama nuo 1998 m.

Moodle sistema yra tinklinė sistema. Norėdami ja dirbti dėstytojai ir studentai neturi įsidiesti jokios papildomos programinės įrangos.

Šios sistemos privalumai:

- lengvas ir lankstus kursų kūrimas;
- reikalauja minimalaus administravimo;
- pedagoginių priemonių įvairovė;
- paprastas ir saugus vartotojų administravimas;
- Moodle yra atviro kodo sistema, o tai leidžia lanksčiai ją adaptuoti prie kiekvieno poreikių;
- programa nemokama.

Reikalavimai Moodle sistemos serveriui:

- www serveris (Apache), palaikantis PHP bei MySQL;
- pageidautinas GD-lib (grafikos modulis – apklausų suvestinių grafikams);

Lotus LearningSpace. Lotus LearningSpace yra vienas iš Lotus ir IBM firmų produktų, skirtų žinioms gilinti ir švietimo procesui valdyti. LearningSpace programinė įranga leidžia kurti ir publikuoti kursus internete. Mokymuisi tokiuose kursuose paprastai užtenka naršyklės (dėstytojas, gali integruoti tokius elementus, kuriems peržiūrėti reikės papildomos programinės įrangos) [2].

Pagrindinės Lotus LearningSpace panaudojimo galimybės:

- mokomosios medžiagos išdėstymas;
- bendravimo organizavimas;
- vertinimo organizavimas;
- sistemos administravimas [2].

Vienas iš teigiamų šios sistemos bruožų – universalumas. Sistema leidžia įkelti ir naudoti kitomis priemonėmis sukurtą mokamąją medžiagą ar net ištikus mokymui skirtus modulius. Lotus LearningSpace nėra savarankiška. Ji veikia kaip kitos, grupiniam darbui skirtos, programos Lotus Notes priedas. Tam, kad mokomas kursas būtų pasiekiamas kitų asmenų, jis turės būti patalpintas į Lotus Domino tarnybinę stotį (serverį) [2].

WebCT. WebCT buvo sukurta britų Kolumbijos universitete, 1996 metais ir pirmiausiai pristatyta penktojoje tarptautinėje World Wide Web konferencijoje Paryžiuje. WebCT žymi tuo, jog yra pirmoji sėkminga komercinė nuotolinio mokymosi aplinka. Ši sistema yra virtualaus mokymosi aplinka, parduodama koledžams ir kitoms institucijoms ir yra naudojama įvairiuose universitetų miesteliuose nuotoliniam mokymuisi. Vadovai gali pridėti tokius savo WebCT įrankius: diskusijų lentą, elektroninio pašto sistemas ir gyvus pokalbius kartu su įvairiais dokumentais bei internetiniais puslapiais [15].

WebCT siūlo:

- Diskusijų forumus, kur užduodami klausimai ekspertams ar aptariami esminiai rezultatai su savo kolegomis.
- Konkrečiai mokymo disciplinai būdingus Web puslapio katalogus su savalaikė ir tiesiogine informacija visai mokymo klasei.
- Mokymo disciplinai reikalingą Web puslapių paiešką (pagal kurso temą ar discipliną), kuri sumažina bereikalingos paieškos rezultatus.
- Resursų bibliotekas, padedančias sukurti WebCT kursus ar pasidalinti kažkuo, tuo ką esate suprojektavę.
- Erdvę, kur dėstytojai, kursų projektuotojai ir studentai mokosi apie WebCT, žiūri WebCT sukurtus produktus ir jų palaikymo ypatybes.
- Pastovią saugyklą, kur galima įsigyti WebCT kursų medžiagą [16].

WebCT automatiškai patikrina, ar naršyklė palaiko JAVA ir ar yra įdiegtos reikalingos Java Virtual Machine (JVM) versijos. Patikrinimas yra atliekamas visada, kai naudojami įrankiai, tokie kaip HTML Creator, kuriam reikalinga JAVA aplinka [17].

1.3. Pagrindiniai kompiuterių tinklų aspektai

Aptariant kompiuterių tinklus, svarbu suvokti jų esmę, paskirtį bei panaudojimą. Kompiuterių tinklai jungia tarpusavyje nutolusius įrenginius, kurie keičiasi informacija. Vienas iš efektyviausių informacijos perdavimo būdų – yra jos skaidymas į paketus, kuris leidžia pigesniais kaštais sukurti tinklą, nes nereikalauja tiesioginio fizinio tarpusavyje nutolusių įrenginių sujungimo. Svarbu aptarti kompiuterių tinklų jungimo būdus, nes nuo to priklauso ryšio kokybė, reikiamų įrengimų panaudojimas ir kiti veiksniai. Darbe taip pat aprašomi įrenginiai, kurie naudojami kuriant kompiuterių tinklus.

1.3.1. Duomenų bloko - paketo koncepcija

Žemiausio lygio aparatinė įranga (*hardware*) persiunčia bitus laidais, radijo bangomis ar stiklo pluoštu. Dauguma tinklų tiekia patogesnę sąsają (*interface*), kuri leidžia kompiuteriui persiųsti duomenų bitus tinklu be jų individualaus valdymo ir be žinojimo, kaip aparatūra koduoja bitus signale [9].

Dauguma kompiuterių tinklų nesiunčia reikalingą informacijos kiekį ištaisai. Tinklų sistema padalina duomenis į mažus blokus, vadinamus paketais (*packets*) ir juos siunčia individualiai. Kompiuterių tinklai dažnai vadinami paketų tinklais (*packets networks*) arba paketų perjungimo (komutavimo) tinklais (*packet switching networks*), kadangi naudoja paketų technologiją [9].

Pagrindiniai paketų panaudojimo motyvai yra tokie:

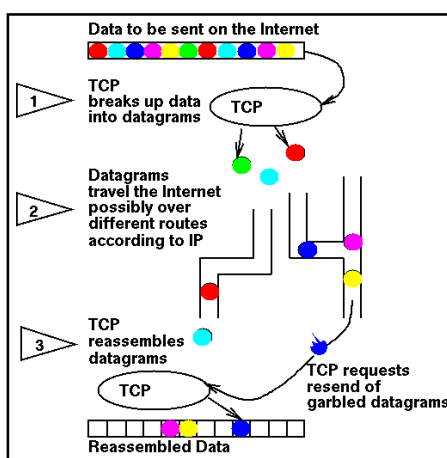
- siuntėjas ir gavėjas turi koordinuoti perdavimą. Perdavimo metu įvykus klaidai, duomenys gali būti prarasti. Duomenų padalinimas į smulkius paketus padeda siuntėjui ir gavėjui nustatyti, kuris blokas atėjo sugadintas ir kuris ne.
- komunikavimo grandinės ir su jomis susijusieji modemai yra brangūs, tai dauguma kompiuterių bendrai naudojami toliau esančiais aparatūra ir perdavimo linijomis. Norint, kad visi kompiuteriai galėtų priimti informaciją, reikia, kad jie naudotų komunikavimo linijas palyginti trumpai ir nekliudytų kitiems kompiuteriams kreiptis į kitus.

Nežiūrint į tai, kad terminas paketas susijęs su bendra mažų duomenų blokų koncepcija, nėra bendro susitarimo apie tikslų paketų formatą. Kiekvienas techninis įrenginys tiksliai apibrėžia

paketo, kuris gali būti persiųstas šiuo įrenginiu, detales. Taip pat tiksliai yra nurodoma, kaip persiunčiamas kiekvienas bitas [9].

Siekiant geriau suprasti, paketo būseną tinkle (žr. 8 pav.), apžvelgti TCP protokolo suformuoti paketai:

1. TCP protokolas padalina duomenis į mažus blokus paketus (datagramas).
2. Paketai keliauja tinkle įvairiais kanalais, kol pasiekia adresatą.
3. TCP protokolas surenka paketus ir perduoda informaciją kitiems protokolams. Gali būti situacijų, kai paketas atkeliauja sugadintas, tada prašoma pakartoti šio paketo siuntimą.



8 pav. Pagrindinės protokolo TCP/IP operacijos

Šaltinis: Units of Analysis for Internet Communication [10]

1.3.2. Kompiuterių tinklų topologijos

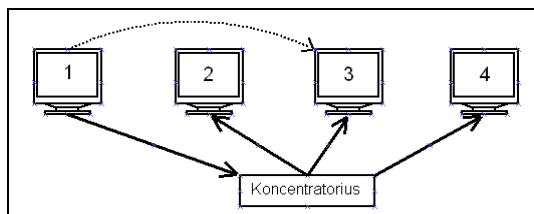
Tinklo topologija yra fizinis kompiuterių išsidėstymas vienas kito atžvilgiu, pagal kurį galima klasifikuoti įvairius tinklus. Praktikoje naudojami trys pagrindiniai topologijų tipai: magistralė, žvaigždė ir žiedas. Išnagrinėjus juos, apžvelgta, kaip gali judėti informacijos paketas.

Žvaigždinis tinklas (Star network). Ši topologija dažnai naudojama kompiuterių tinklams jungti. Kompiuteriai jungiami prie tam tikro įrenginio, vadinamo tinklo mazgu (nodes). Mazgu gali būti kompiuteris arba specialus įrenginys, pvz. koncentratorius (hub), komutatorius (switch).

Panaudojus *koncentratorių* paketas yra perduodamas visiems kompiuterių tinkle esantiems įrenginiams, išskyrus siunčiantįjį. Koncentratorius realizuoja loginę magistralę, nes

paketai yra perduodami visiems kompiuteriams ir einamu laiku paketą tinkle siunčia tik vienas kompiuteris.

Kai kurie koncentatoriai turi tarpinio duomenų saugojimo (store and forward) paslaugą. Jos esmė ta, kad koncentratoriuje yra laikino paketų saugojimo buferiai, kurie vėliau perduodami per atitinkamas jungtis [11]. 9 paveiksle matyti grafinis koncentratoriaus panaudojimo pavaizdavimas.

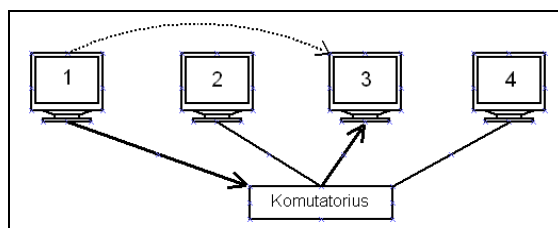


9 pav. Žvaigždinis tinklas ir jo veikimas panaudojus koncentratorių

Šaltinis: sudaryta darbo autoriaus

Iš 9 paveikslo matyti kaip pirmas kompiuteris nori perduoti paketą ketvirtam kompiuteriui ir paketas yra persiunčiamas visiems tinklo kompiuteriams, o ne konkrečiai adresuotam.

Jungiant tinklą žvaigždės topologija su *komutatoriumi*, informacinis paketas perduodamas tik adresuotam kompiuteriui. Tokia technologija leidžia bendrauti keliems kompiuteriams esamuju laiku ir taip padidina tinklo našumą. Komutatoriaus veikimas pavaizduotas 10 paveiksle.

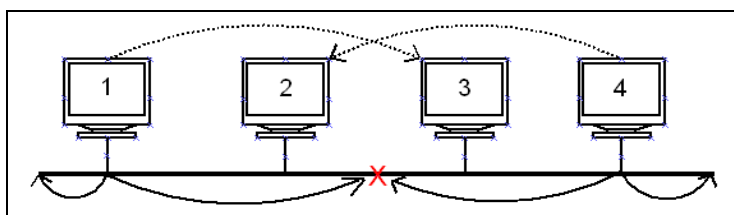


10 pav. Žvaigždinis tinklas ir jo veikimas panaudojus komutatorių

Šaltinis: sudaryta darbo autoriaus

Magistralinis tinklas (Bus network). Šiame tinkle kompiuteriai yra sujungti vienu kabeliu, jungiančiu vieną kompiuterį su kitu. Paketas šiame tinkle siunčiamas į abi puses ir susinaikina pasiekęs kompiuterio tinklo galus. Paketas yra pradedamas siųsti, tik tada, kai

kompiuteris užfiksuoja, jog tinklas yra laisvas. Gali atsitikti ir taip, kad vienu metu keli kompiuteriai užfiksuoja, kad tinklas laisvas ir pradeda paketo siuntimą, šiuo atveju įvyksta paketų kolizija (žr. 11 pav.)

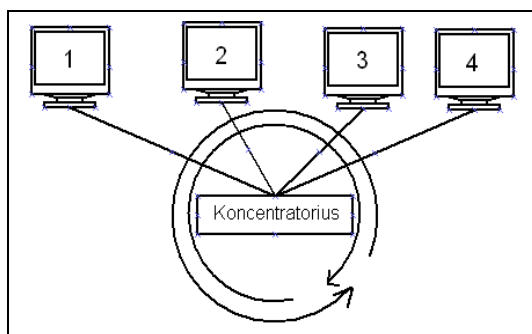


11 pav. Magistralinis tinklas su įvykusia kolizija

Šaltinis: sudaryta darbo autoriaus

11 paveiksle pavaizduota, kaip du kompiuteriai užfiksuoja, jog tinklas laisvas ir pirmas bei ketvirtas kompiuteriai pradeda paketų siuntimą, ko pasekoje įvyksta paketų kolizija.

Žiedinis tinklas (Ring network). Šioje topologijoje ratu – arba į viena pusę arba atvirkščiai, nuolatos keliauja signalas (žr. 12 pav.). Žiediniame tinkle paketų kolizijos klausimas yra išspręstas, nes kompiuteris gali pradėti siųsti paketą, tik gavęs leidimą iš vaikščiojančio tinkle signalo.



12 pav. Žiedinis tinklas

Šaltinis: sudaryta darbo autoriaus

Praktikoje naudojami keli žiedinio tinklo variantai. Populiariausieji jų – Kembridžo žiedas (Cambridge ring) ir žymėtasis žiedas (Token ring) [3].

Kembridžo žiedo tinklu keliauja tik vienas fiksuoto dydžio informacijos mini paketas, atliekantis voko vaidmenį. Kiekvienas žiedo kompiuteris, gavęs tokį voką, patikrina nurodytą gavėjo adresą. Jeigu informacija adresuota jam, ji perskaitoma. Jeigu tas kompiuteris turi siunčiamos informacijos, ji „įdedama“ į voką ir užrašomas naujas gavėjo adresas, jeigu ne – žiedo kaimynui

perduodamas tuščias vokas. Visais atvejais informaciją galima persiųsti tik turint tuščią voką. Voko judėjimą kontroliuoja specialus kompiuteris, vadinamas monitoriumi, kuris ir pašalina „niekieno“ laiškus informuodamas apie tai siuntėjui [3].

Žymėtame žiede keliauja ne vokas, o žymė – leidimas perduoti informaciją. Tik gavus šią žymę, informacija gali būti siunčiama gavėjui. Perdavus visus duomenis, žymė siunčiama žiedo kaimynui. Panašiai kaip Kembridžo žiede, vienas kompiuteris čia seka žymės judėjimą ir naikina nesančių adresatų informaciją [3].

Visi išvardinti būdai gali būti kombinuojami pvz., žvaigždutės ir magistralės jungimo būdas naudojamas, kai keli tinklo mazgai, panaudojus specialius prievadus, sujungiami į bendrą magistralę, o prie jų pagal žvaigždutės principą jungiami kompiuteriai. Žvaigždutės ir žiedo jungimo būdas – tai, kai kiekviena kompiuterių grupė yra jungiama prie mazgo žvaigždutės principu, o šie mazgai jungiami prie įrenginio, kuris naudoja žiedo topologiją [3].

Praktikoje yra naudojami ir kiti tinklo jungimo būdai, pvz., jungimas visiškai susietu tinklu, kai visi tinklo įrenginiai turi tiesioginį ryšį. Jungimas ląstelėmis - kai tinkle yra keletas kanalų, kuriais paketas gali pasiekti adresatą ir kt.

1.3.3. Tinklo sąsajos įrenginiai

Norint įgyvendinti pasirinktą tinklo topologiją ar plėsti vietinius tinklus, reikia pasirinkti tinkamus tinklo sąsajos įrenginius. Šie įrenginiai padeda kurti vietinius tinklus ar sujungti keletą jau esančių, padidinti atstumus tarp jungiamų taškų ir pan. Analizuojant tinklo sąsajos įrenginius, pirmiausiai aptarti kartotuvai.

Kartotuvai (Repeater). Kartotuvai skirti signalui stiprinti, nes perduodamas kabeliu, jis nuolat silpnėja. Atskirai šie įrenginiai retai aptinkami, jie dažniausiai yra įmontuoti į kitus įrenginius – koncentratorius(hub) arba komutatorius(switch) [3].

Toks įrenginys dirba fiziniame lygmenyje, o tai reiškia, kad neskaito siunčiamos informacijos turinio, o tiesiog ją pakartoja.

Koncentratoriai (Hub). Koncentratoriai, naudojami Ethernet tinkluose ir kartais vadinami daugiajungčiais kartotuvais, kadangi patys savaime sugeba sustiprinti signalą [11].

Koncentratoriai labiausiai paplitę žvaigždutės tipo tinkluose. Čia kiekvienas kompiuteris jungiamas prie koncentratoriaus atskiru kabeliu, o iš bet kurio prievado perduotas signalas sustiprinamas ir siunčiamas kitiems.

Komutatoriai (Switch). Komutatoriai yra panašūs į koncentratorius. Komutatoriai taip pat statomi tinklo šerdyje ir jungia tarpusavyje tinklus, tačiau skirtingai nei koncentratoriai, kurie

persiunčia tinklo duomenis visiems prie jo prijungtiems mazgams, komutatoriai “peržiūri” kiekvieną atėjusį paketą ir persiunčia jį tik konkrečiam adresatui pagal informaciją, esančią kiekvieno paketo antraštėje. Kad išskirti konkretų perdavimą nuo kitų, komutatorius sukuria laikiną ryšį tarp šaltinio ir tikslo, ir jį nutraukia, kai perdavimas baigiasi. Dėl to komutatoriai mažiau apkrauna tinklus ir duomenų perdavimas vyksta greičiau [11].

Tiltai (Bridge). Tiltai – tai įrenginiai, susiejantys tinklo segmentus. Komutatoriai iš esmės yra sudėtiniai tiltai viename įrenginyje. Tiltas, kaip ir kartotukas, atlieka signalo sustiprinimo funkciją, bet tuo pačiu sugeba pasirinktinai filtruoti paketus pagal adresus [11].

Tiltai padeda išplėsti tinklą, maksimaliai padidinant kompiuterių skaičių jame. Skirtingai nei koncentratoriai, tiltai padeda išskaidyti tinklą į keletą segmentų, taip izoliuodami srauto dalį.

Tiltai veikia OSI modelio duomenų perdavimo linijos sluoksnyje. Jie gali sujungti skirtingų rūšių tinklus, pavyzdžiui, Ethernet ir Token Ring. Tiltai sujungia ir skirtingas fizines aplinkas, pavyzdžiui, vytos poros ir koaksialinį kabelį.

Tiltai yra „intelektualus“ įrenginys, nes nagrinėja, kur reikia siųsti duomenis. Kai duomenys perduodami per tiltą, kompiuterių adresai išsaugomi tiltų operatyvinėje atmintyje. Jis naudoja tuos duomenis maršrutizavimo lentelės kūrimui. Priimdamas paketą, tiltas ieško siuntėjo adreso maršrutizavimo lentelėje. Jei adreso nėra, jis įrašomas. Po to tiltas tikrina paskirties adresus su maršrutizavimo lentelės duomenų baze [11].

Paketai ateinantys į tilto įėjimą, praleidžiami į išėjimą tik tuo atveju, jei jie adresuojami kompiuteriui, esančiam kitoje tilto pusėje, t.y., jei adresatas yra viename segmente su šaltiniu, paketas atmetamas. Tokia filtracija sumažina tinklo srautą ir izoliuoja tinklo segmentus.

Jeigu adresatas ir šaltinis yra skirtinguose segmentuose, tiltas perduoda paketą adresatui per atitinkamą jungtį. Jeigu gavėjo adreso nėra, tiltas perduoda paketą per visas savo jungtis, išskyrus tą, per kurią paketas buvo priimtas [11].

Maršrutizatoriai (Router). Aplinkoje, jungiančioje keletą tinklo segmentų su skirtingais protokolais ir architektūromis, tiltai ne visada užtikrina greitą ryšį su visais segmentais. Tokiam sudėtingam tinklui reikalingas maršrutizatorius (router), t.y. įrenginys, kuris ne tik žino kiekvieno segmento adresą, bet ir nustato patogiausią maršrutą duomenų perdavimui.

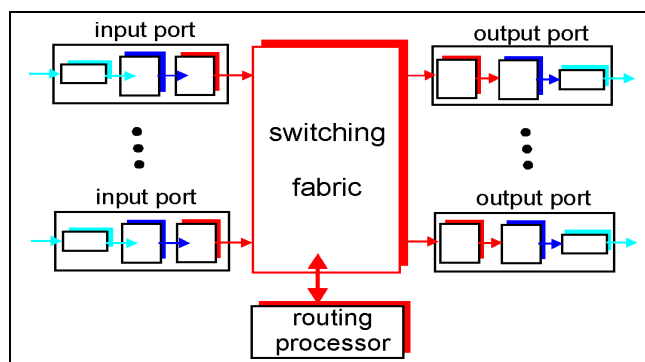
Maršrutizatoriai nuskaityti pakete tinklo adreso informaciją bei papildomus duomenis ir siunčia gavėjo tinklui. Jeigu gavėjas yra tolimame tinkle, maršrutizatorius persiunčia paketus kitam maršrutizatoriui. Pagal tinklo kelio žemėlapi, vadinamą “maršrutizavimo lentelė”, maršrutizatoriai užtikrina, kad paketai keliautų efektyviausiais keliais link paskirties vietų. Maršrutizavimo lentelėje rankiniu būdu arba automatiškai įrašomi tinklų adresai ir kita informacija. Gautus paketus prieš siųsdamas maršrutizatorius lygina su turima informacija [11].

Maršrutizatoriai gali filtruoti ir izoliuoti srautą, keistis duomenimis apie maršrutų būklę ir remdamiesi šia informacija, apeiti lėtus arba sutrikusius ryšio kanalus. Kadangi jie gali perduoti duomenis keliais kanalais, tai net sugedus kuriam nors maršrutizatoriui, siunčiantysis maršrutizatorius gali nustatyti alternatyvų maršrutą siųsti duomenis.

Maršrutizatoriai nustato ir tai, ką reikia perduoti, ir tai kur reikia perduoti. Jis pažįsta ne tik adresą, kaip tiltas, bet ir protokolo tipą. Be to, maršrutizatorius gali nustatyti kitų maršrutizatorių adresus ir nuspręsti, kokius paketus kokiems maršrutizatoriams peradresuoti. Tiltas atpažįsta tik vieną kelią. Maršrutizatorius iš kelių galimų kelių nustato esamu momentu geriausią.

Išskiriamos dvi pagrindinės maršrutizatoriaus funkcijos:

- Maršrutų parinkimas;
- Įeinančių paketų nukreipimas į reikiamą prievadą. Grafiškai ši supaprastinta maršrutizatoriaus architektūra pavaizduota 13 paveiksle.



13 pav. Apibendrinta maršrutizatoriaus architektūra

Šaltinis: Computer Networks [12]

Kaip matyti iš pateikto paveikslo, maršrutizatorius parenka maršrutus analizuodamas maršrutų lenteles ir suradęs optimaliausią variantą nukreipia paketus į reikiamą prievadą. Maršrutizatoriaus panaudojimas pagerina kompiuterių tinklų našumą ir yra neatsiejamas kompiuterių tinklų komponentas.

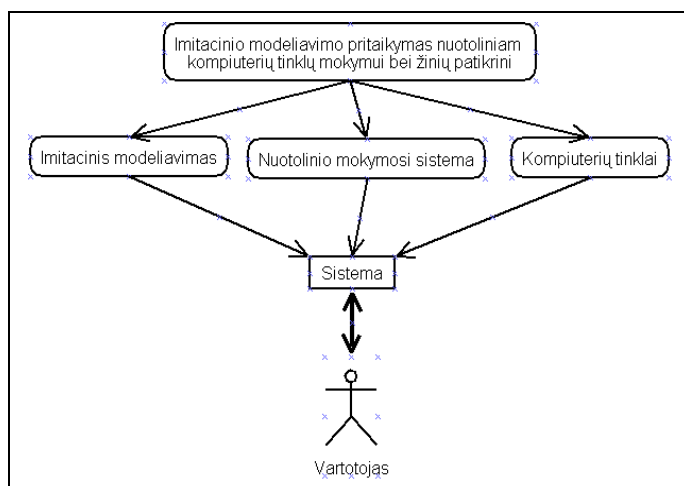
II. PROJEKTINĖ DALIS

Norint pavaizduoti kaip imitacinis modeliavimas gali būti pritaikomas nuotoliniam kompiuterių tinklų mokymui, buvo sukurta sistema (NetLab), kuri gali būti nuotolinio mokymosi sistemos Moodle dalis ir kuri leidžia praplėsti mokymosi galimybes kompiuterių tinklų srityje. Projektinėje dalyje analizuojamas šio įrankio sukūrimas, panaudojimo galimybės ir kiti aspektai.

Prieš pradėdant analizuoti sukurta imitacinio modeliavimo, mokymo įrankį, pateikiamas darbinės srities modelis, kuris pateikia grafinę viso darbo struktūrą (žr. 14 pav.).

2.1. Darbinės srities modelis

Šis darbinės srities modelis skirtas pavaizduoti projekto grafinei struktūrai. Jame vaizdžiai išdėstytos pagrindinės sudedamosios darbo dalys.



14 pav. Darbinės srities modelis (schema).

Šaltinis: sudaryta darbo autoriaus

Kuriama sistema turi apimti tris pagrindines sudedamąsias dalis: imitacinį modeliavimą, nuotolinio mokymosi sistemą bei kompiuterių tinklus.

Sistemos imitacinis modeliavimas turi būti orientuotas į kompiuterių tinklus, kuri padės vaizdžiai analizuoti kompiuterių paketų srautų judėjimą, leis vartotojui keisti tinklo modelio parametrus ir matyti įvykstančias pasekmes.

Sistema turi nuolat augti, pilnėti, ne tik atliktų modelių gausa, bet ir funkcionavimu, todėl turi būti įgyvendintas naujų tinklo objektų kūrimas.

Kuriama sistema turi būti tam tikros nuotolinio mokymosi sistemos dalis, nes savarankiška sistema turėtų labai ribotas galimybes.

Sukurta sistema turi atpažinti prisijungusį vartotoją, pagal kurį automatiškai prisitaikys vartotojo interfeisą (pvz. studentas galės modeliuoti ir imituoti kompiuterių tinklą, o dėstytojas be viso to, peržiūrėti studentų darbus, kurti užduotis, įvertinti studentų sukurtus tinklo modelius, kurti tinklo objektus ir t.t.).

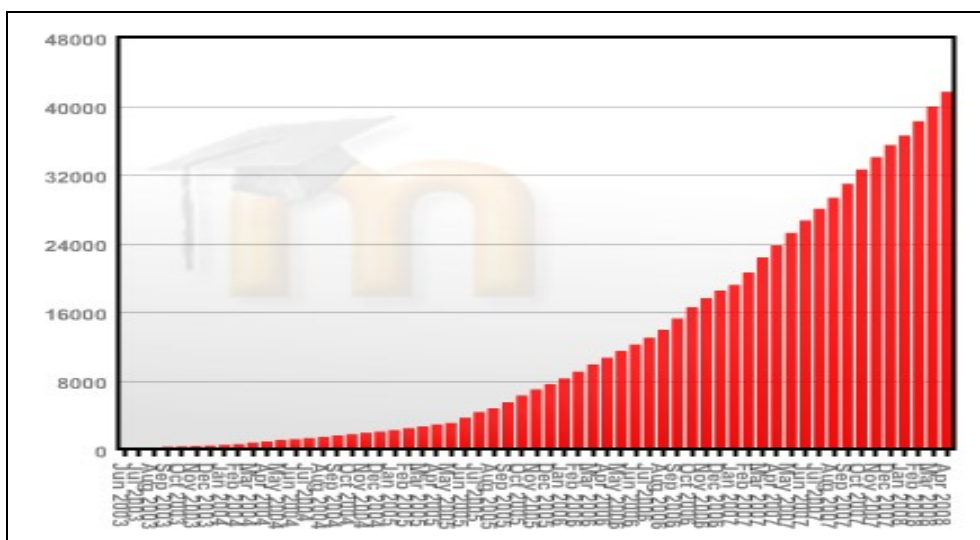
2.2. Įrankių ir priemonių pasirinkimas, kuriant sistemą NetLab

Kuriant nuotolinio mokymosi sistemą NetLab, buvo panaudoti šie įrankiai ir priemonės:

- Prie kurtos sistemos taikyta nuotolinio mokymosi sistema Moodle;
- Pasirinkta programavimo kalba: Java;
- Programavimo aplinka: Eclipse;
- JNLP(Java Network Launch Protocol) failo kūrimui naudotas: PHP;
- Pasirinkta duomenų bazės valdymo sistema: MySQL;
- Naudota skriptavimo kalba tinklo objektams aprašyti: JavaScript;

Dabartiniu metu nuotolinio mokymosi sistemos yra labai plačiai taikomos mokymo tikslams. Studentai mokosi šiuo būdu įvairių sričių dalykus. Atsiranda poreikis dar labiau plėsti ir tobulinti ir didinti tokių sistemų panaudojimą.

Šiam projektui įgyvendinti pasirinkta Moodle nuotolinio mokymosi sistema, nes ji yra atviro kodo produktas – tai leidžia modifikuoti sistemą ir prisitaikyti savo reikmėms; yra nemokoma bei greitai populiarėjanti. 15 paveiksle pavaizduota, kaip kito užregistruotų Moodle sistemų skaičius, t.y. kiek jų buvo panaudota kiekvienais metais (2003-2008 m.). Duomenys pateikiami kas mėnesį.

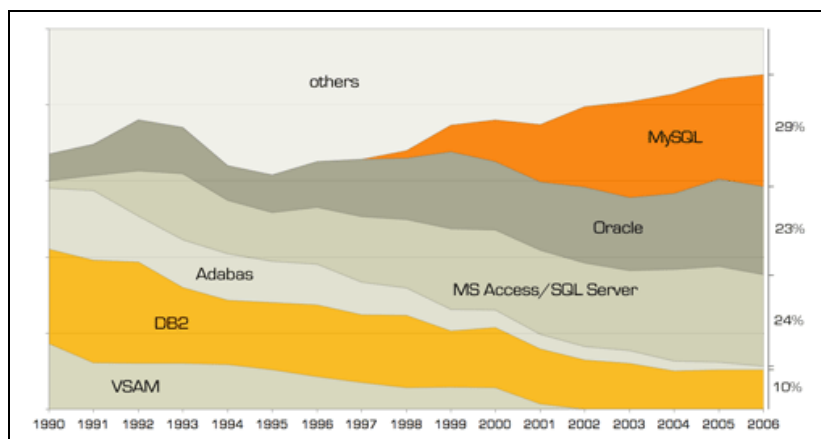


15 pav. Užregistruotų Moodle sistemų skaičius 2003-2008 metais

Šaltinis: sudaryta „Moodle Statistics“ [7]

Kaip matyti iš 15 paveikslo, nuo 2003 iki 2008 metų sparčiai išaugo Moodle nuotolinio mokymosi sistemos panaudojimas. Galima daryti prielaidą, jog ši tendencija išsilaikys, nes nuolat populiarėja nuotolinis mokymasis ir tokių sistemų paklausa auga.

Nors Moodle palaiko ne tik MySQL, tačiau projektui įgyvendinti pasirinkta būtent ši duomenų bazių valdymo sistema. Ši sistema yra populiariausia atviro kodo duomenų bazių valdymo sistema ir dažniausiai pasirenkama IT specialistų. 16 paveiksle pavaizduota, kokias dar duomenų bazių valdymo sistemas pasirenka informacinių technologijų specialistai, kurios iš jų populiariausios.



16 pav. Populiariausios duomenų bazių valdymo sistemos tarp IT specialistų

Šaltinis: sudaryta JoinVision E-Services GmbH [8]

Aptariant 16 paveikslą, galima matyti, jog atviro kodo MySQL duomenų bazių valdymo sistema populiarumu nenusileidžia tokioms komercinėms sistemoms kaip Oracle, SQL Server, DB2 ir kai kuriais atvejais ši sistema (MySQL) populiarumu bei panaudojimu net lenkia minėtas sistemas.

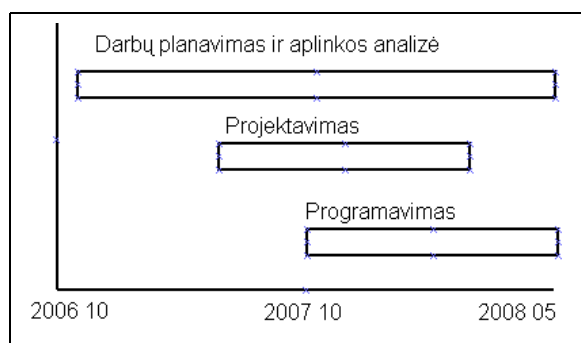
Java programavimo kalba pasirinkta todėl, kad ji yra vykdoma virtualios mašinos, kuri turi Java Web Start technologiją, leidžiančią iš tinklo vykdyti Java programas. Tarkime norima prisijungti prie Moodle sistemos. Prisijungiant prie tokių sistemų, įvedamas vartotojo vardas ir slaptažodis. Šitaip sistema atpažįsta prisijungusį asmenį ir jam suteikia numatytas teises. Vartotojas jungdamasis per minėtą sistemą prie Java programos formuoja individualų programos paleidimo scenarijų ir pateikia jį Java Web Start technologijai.

2.3. Projekto NetLab eigos aprašymas

Kuriant projektą, pirmiausiai buvo numatyta tam tikra jo eiga, t.y. etapai, kuriais jis bus vykdomas ir sistemingai įgyvendinamas. Šie etapai yra tokie:

- **Darbų planavimas ir aplinkos analizė.** Šiame etape sudaromas pradinis projekto modelis, suformuojama darbinė sritis ir nustatomas laikas darbinei sričiai įgyvendinti. Pasirenkamos atitinkamos priemonės projektui įgyvendinti. Šiam etapui skiriami aštuoni mėnesiai.
- **Projektavimas.** Nusistačius darbinę sritį bei priemones projektui įgyvendinti, toliau sekantis etapas – projektavimas. Čia sudaromas detalus modelis, kaip pasiekti (su pasirinktomis priemonėmis) galutinį tikslą. Šiam etapui įgyvendinti yra skiriami penki mėnesiai.
- **Projekto įgyvendinimas.** Šis etapas gali prasidėti ir nepasibaigus projektavimo etapui. Čia realizuojami projekte numatomi uždaviniai. Tai – programavimas ir kiti darbai. Šiam etapui skiriami aštuoni mėnesiai.

Aprašant projekto eigą, buvo nusistatyti tam tikri laikotarpiai, per kuriuos bus įgyvendintas kiekvienas projekto etapas. Kuriant NetLab sistemą, paaiškėjo, jog nusistatyto darbų eigos grafiko laikytis sudėtinga, tačiau nepaisant to, toks darbų planavimas suteikia disciplinos. Žemiau pateiktame 17 paveiksle, pavaizduotas projekto (kiekvieno etapo) įgyvendinimo laikas, kuris skiriasi nuo suplanuoto.



17 pav. Darbų atlikimo (eigos) diagrama.

Šaltinis: sudaryta darbo autoriaus

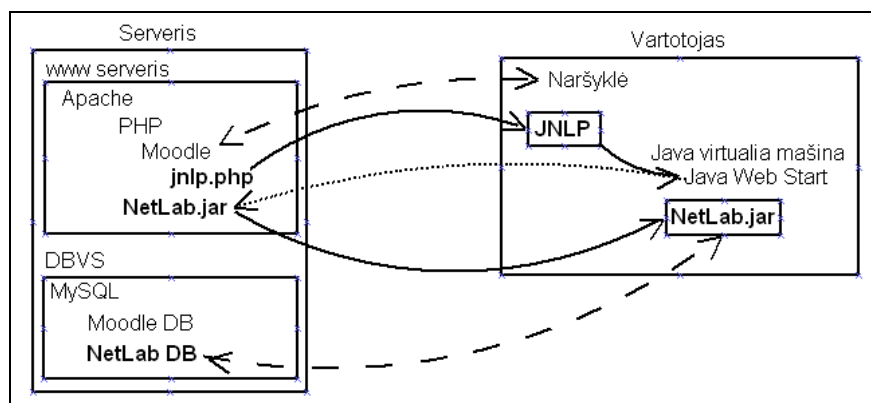
Darbų atlikimas suskaidytas į tris etapus: darbų planavimą ir aplinkos analizę, projektavimą ir programavimą. Diagramoje pavaizduota, kiek laiko ir kada kiekvienas etapas buvo vykdomas. Kaip matyti, darbų planavimas ir aplinkos analizė vyko visą projekto

įgyvendinimo laiką, projektavimas ir programavimas užėmė mažiau laiko. Pastebėtina, jog tam tikru metu, visi etapai buvo vykdomi kartu.

Vykdamas šį projektą, reikėjo surišti nuotolinio mokymosi sistemą (Moodle) su kuriama NetLab sistema. Tai yra, Moodle sistema turi perduoti reikiamus duomenis apie vartotoją, jo teises ir pan. Java programai galima perduoti norimus duomenis per JNLP failą. JNLP failai turi būti kuriami dinamiškai, nes kiekvienam vartotojui bus kuriamas skirtingas failas ir ši informacija yra nuolat kintanti. Nenorint „šiukšlinti“ serverio, pseudo failas bus generuojamas PHP skriptu ir siunčiamas per HTTP header'į, dėl to JNLP failai serveryje neegzistuos. Taip pat reikia sukonfigūruoti web serverį (Apache), kad JNLP failas būtų sėkmingai vykdomas.

Viskas būtų gerai, išskyrus saugumą. Vartotojas gali perskaityti JNLP failą arba vykdomą programą, kuriuose nurodytas prisijungimas prie duomenų bazės ir kita neviešintina informacija. Internetinis puslapis yra patrauklus tuo, jog duomenų bazė yra pakankamai apsaugota. Tai yra duomenų bazės valdymo sistemą galima sukonfigūruoti taip, kad priėjimą prie DB turėtų tik konkretus web serveris. Kuriamai sistemai duomenų bazės priėjimas turi būti atviras. Iš to seka, kad vartotojas gali gauti prisijungimo duomenis prie duomenų bazės ir sėkmingai joje šeiminkauti. Ši problema liko neišspręsta. Saugumo stoka privertė reikiamas sistemos NetLab duomenų bazės lenteles talpinti atskiroje duomenų bazėje, taip apsaugant Moodle sistemą.

Pasirinkus projekto įgyvendinimui atitinkamus įrankius ir susidarius darbų vykdymo planą, sudaryta apibendrinta projekto schema (žr. 18 pav.).



18 pav. Apibendrintas projekto vaizdas.

Šaltinis: sudaryta darbo autoriaus

Aptariant pateiktą projekto schemą, matyti, jog vartotojas naršyklės pagalba prisijungia prie nuotolinio mokymo sistemos Moodle, tada šioje sistemoje jis nurodytoje vietoje duoda komandą paleisti NetLab programą(sužadinama nuoroda), tuomet PHP pagal jnlp.php skriptą generuoja vartotojui individualų pseudo failą JNLP, o Java Web Start technologija perskaito

JNLP, kuriame nurodyta parsisiųsti NetLab programą ir perduoda jai reikiamus parametrus apie vartotoją ir duomenų bazę. Galiausiai vykdoma programa NetLab, kuri bendrauja su duomenų baze NetLab.

Sukurta sistema NetLab turi dvi darbinės aplinkas, t.y eksperto, dėstytojo darbo aplinka bei mokinio, studento darbo aplinka. Šios aplinkos turi skirtingas paskirtis.

Eksperto, dėstytojo darbo aplinka:

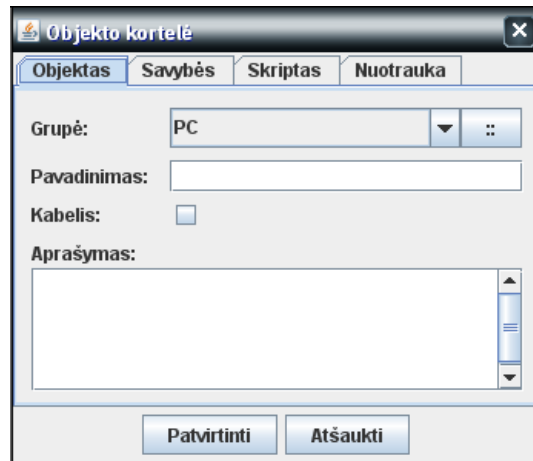
- Kuria tinklo objektus;
- Kuria objektų parametrus;
- Skriptų pagalba aprašo tinklo objektų veikimo principus;
- Kuria studentams užduotis;
- Peržiūri studentų darbus;
- Rašo įvertinimą.

Mokinio, studento darbo aplinka:

- Panaudodamas eksperto sukurtus objektus modeliuoja tinklus;
- Derina tinklo objektus pagal eksperto nustatytus reikalavimus;
- Nustato vykdomos imitacijos parametrus;
- Vykdo tinklo imitaciją ir analizuoja gautus rezultatus.

2.4. Projekto NetLab sistemos aplinka

Norint įgyvendinti nuotolinius mokymus NetLab sistemoje, visų pirma reikia susikurti tinklo objektus, tam, kad studentas arba dėstytojas galėtų jais modeliuoti ir imituoti kompiuterių tinklus. Norint kurti tinklo objektus, yra sukurta atitinkama aplinka, kurioje aprašomi objektų parametrai bei veikimo principai ir kita reikalinga informacija. Pirmas žingsnis kuriant objektus yra objekto kortelės užpildymas, kuri sudaryta iš keturių dalių – bendra tinklo objekto informacija, savybės, skriptas bei nuotraukos (žr. 19 pav.).



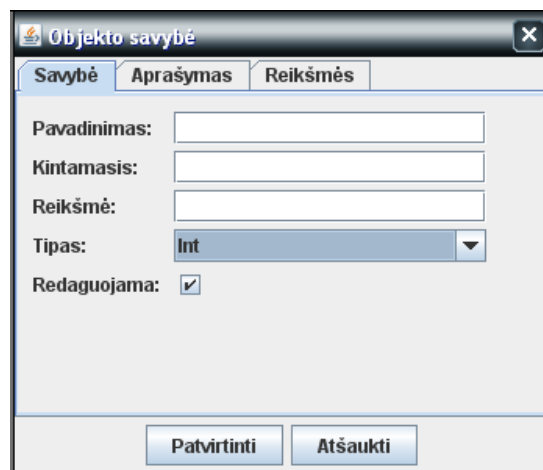
19 pav. Tinklo objekto kortelė

Šaltinis: programos NetLab darbo lango vaizdas

Kaip buvo minėta, kuriant tinklo objektus yra keturios pagrindinės dalys:

Bendra tinklo objekto informacija – čia objektui priskiriama grupė (pvz., kompiuteriai, serveriai, maršrutizatoriai, komutatoriai, kabeliai ir t.t.). Toliau užrašomas objekto pavadinimas ir nurodomas, kokios prigimties yra tinklo objektas (kabelis ar įrenginys). Suformuotą objekto aprašymą matys studentai, taip gaudami visą informaciją apie objekto panaudojimą ir funkcionavimą.

Antra svarbi skiltis yra **objekto savybės**. Atsidarius savybių kortelę, matosi jų sąrašas. Kuriant arba keičiant objekto savybę (parametrą), atsidaro šis programos langas (žr. 20 pav.).



20 pav. Tinklo objekto savybės kortelė

Šaltinis: programos NetLab darbo lango vaizdas

Objekto savybių arba parametrų kūrimas susideda iš penkių pagrindinių etapų:

- savybės pavadinimo formulavimas;
- kintamojo vardo parinkimas – jis bus įtrauktas į skriptą;
- kintamojo reikšmės parinkimas – nutylėta kintamojo reikšmė;
- parametro tipo parinkimas – galimi parametrų tipai yra: *Int* (*sveikasis skaičius*), *Float* (*realusis skaičius*), *String* (*simbolių eilutė*), *ComboBox* (*vienas iš galimų*), *CheckBox* (*pažymėti*). Nurodžius tipą *ComboBox*, reikės priskirti reikšmių sąrašą. Tipo parinkimas apsaugo objektą nuo netinkamo tipo įvedimo;
- Objekto savybės yra redaguojamos ir neredaguojamos. Redaguojamas skiltis studentai galės keisti darant tinklo modelį, o neredaguojamas tik peržiūrėti.

Neatskiriamas nuo objektų kūrimo yra ir skriptų rašymas. Nuo šio skripto kodo priklauso sukurtų objektų veikimas atliekant tinklo imitacijas. Skriptavimo aplinka pavaizduota 21 paveiksle.

21 pav. Skriptų rašymo aplinka

Šaltinis: programos NetLab darbo lango vaizdas

Tinklo objekto veikimas yra aprašomas *JavaScript* kalba. Ekspertas turi aprašyti šešias funkcijas, kurias sužadina vykdomo modelio imitacija. Šios funkcijos yra:

- *get_paketo_perdavimo_trukme()* – šią funkciją ir tiksliai ją naudoja kabelio prigimties objektai. Ji gražina sugaištą laiką, kurį paketas turi praleisti kabelyje;

- *siusti_paketa()* – šioje funkcijoje aprašomas paketas, kuris yra „neribotas“ masyvas. Visi paketo (masyvo) įrašai yra laisvi, išskyrus pirmą, kuris nurodo paketo spalvą; taip pat ši funkcija grąžina paketo siuntimo būseną (1- leidžiama siųsti paketą, 0 – draudžiama);
- *priimti_paketa()* – ši funkcija tikrina, ar priims objektas siunčiamą paketą (1 – priima, 0 – atmeta);
- *tikrinti_adresata()* – šioje funkcijoje aprašoma, ar tai galutinis taškas, ar reikia bandyti paketą persiųsti kitiems tinklo objektams.
- *get_proc()* – ši funkcija grąžina, paketo apdorojimo laiką tinklo objekte.
- *atsakyti_paketui()* – ši funkcija naudojama, norint atsakyti siuntusiam objektui paketais.

Sukūrus objektus, ekspertas turi sudaryti užduotis, pagal kurias vėliau studentai darys modelius, analizuos tinklų sistemas. Užtai jie gaus atitinkamus įvertinimus.

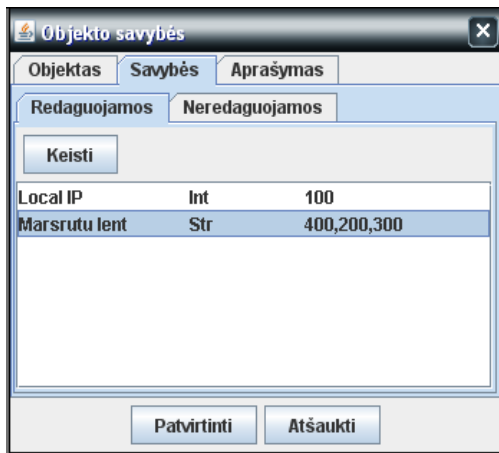
Siekiant pamatyti studento darbo aplinką, pateikiamas šis pavyzdys.

Tarkime ekspertas sukūrė keturis tinklo objektus:

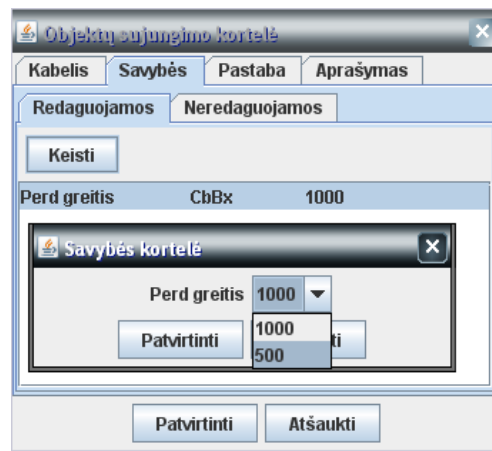
- Kompiuterį – kuris kurs ir siųs nurodytu adresu paketus. Paketo apdorojimo trukmė nustatyta pastovi – 200 ms;
- Maršrutizatorių – kuris rūpinsis, kad paketas pasiektų adresatą. Vartotojas turės sudaryti maršrutų lentelę. Jei maršrutas turi keletą kanalų, tai kelias parinks automatiškai iš galimų variantų. Paketo apdorojimo laikas yra pastovus -1000 ms;
- Serverį – jo funkcionavimas sutampa su kompiuterio. Eksperimento metu jis priims paketus ir siųs atsakymus kompiuteriams;
- Kabelį, kurio greitį vartotojas galės pasirinkti – 1000 ms arba 500 ms (paketo sugaištas laikas keliaujant kabeliu); kabelio ilgis modeliui įtakos neturi.

Kai tinklo objektai sukurti, studentas arba mokinys gali tais objektais disponuoti, keisti jų savybes, suteikti jiems pavadinimus, pasirašyti sau komentarus ar pastabas. Objektų savybių langas pateiktas 22 paveiksle. Objektus galima įkelti iš objektų medžio, kuris yra kairėje programos lango pusėje. Objektai keliami „paimk ir mesk“ (DragDrop) pagalba. Objektai tarpusavyje jungiami kabeliais (ryšiais). Kabelių sujungimui reikia pasirinkti objektą, vėliau išsirinkti norimą kabelio rūšį. Visa tai vyksta „iššokančio“ lango pagalba. Toliau pasirenkama su kuriuo objektu norima susijungti. Kiekvienas kabelis taip pat turi savo tam tikras savybes (žr. 23

pav.). Pasirinkus kabelį, pažymėjus jį, galima keisti jo įvairius parametrus. Taip modeliuojamas kompiuterių tinklas.



22 pav. Objekto savybių keitimo langas



23 pav. Kabelio savybių keitimo langas

Šaltinis: sukurtos sistemos NetLab programos langai

Po tinklo modeliavimo bei derinimo, vartotojas gali imituoti šį tinklą. Pirmiausiai jis turi objektams parinkti paketų kūrimo (generavimo) šiame objekte intensyvumą. Paketą generuoti galima dviem būdais: pasirinkus pastovų (vykstantį vienodais laiko tarpais) paketų kūrimą arba nurodžius intervalą, iš kurio kas kart atsitiktinai parenkamas paketo generavimo intensyvumas. 24 paveiksle pateiktas paketų generavimo langas. Šiame lange taip pat galima nustatyti, kad šis objektas eksperimento metu paketų nekurtų. Žinoma, kad paketų kūrimas priklausys ir nuo eksperto sukurto skripto. Jei ekspertas yra uždraudęs kurti šiam objektui paketus arba yra netinkamai parinkti objekto parametrai ir sujungti ryšiai ir kita, paketų generavimas nebus vykdomas, nors to pageidauja vartotojas. Suderinus visą modelį, galima atlikti eksperimentą, kuris yra sužadinamas pasirinkus pagrindiniame meniu „Pradėti“. Prieš vykdant imitaciją, programa paprašys įvesti eksperimento trukmę milisekundėmis. Tai pavaizduota 25 paveiksle.



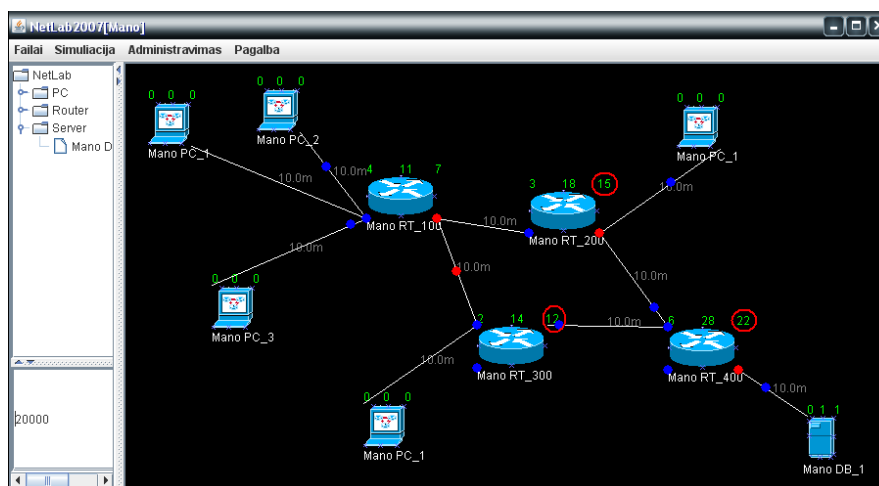
24 pav. Paketų generavimo nustatymo langas



25 pav. Eksperimento vykdymo langas

Šaltinis: sukurtos sistemos NetLab programos langai

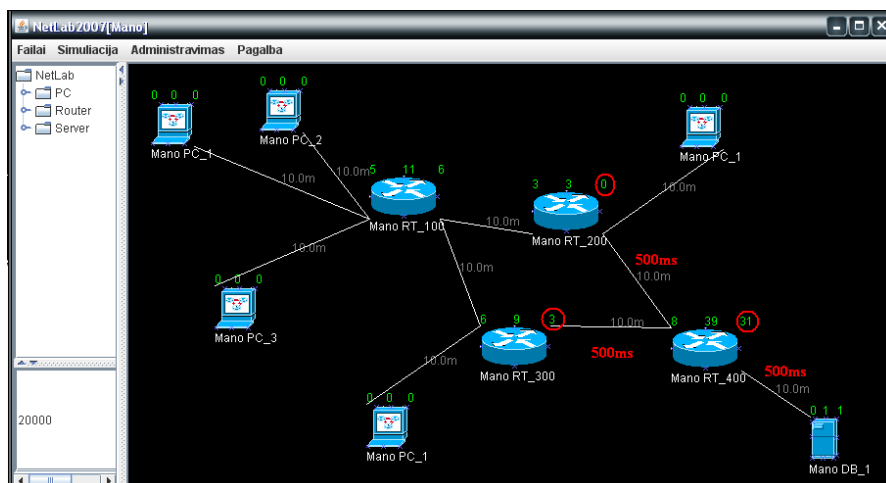
Norint geriau suprasti modelio sukurto NetLab sistemoje veikimo esmę, reikia įvykdyti keletą eksperimentų keičiant modelio parametrus. Tarkime, jog visi eksperimentai vykdomi 20 sekundžių ir visi tinklo kompiuteriai kas 1000ms siunčia paketus duomenų bazės serveriui. Serveris atsako į siunčiamus paketus vienu paketu siuntėjui ir visi tinklo ryšiai aptarnauja paketą 1000 ms greičiu. Vykdamt imitaciją, įvairiais kanalais keliauja paketai, kurie gali būti pažymėti atitinkamomis spalvomis – mėlynu rutuliuku žymimas paketas keliantis link tikslo, o raudonas rutuliukas reiškia atsakymą į atsiustą paketą. Sukurto modelio ir modeliavimo aplinkos darbo vaizdas pateiktas 26 paveiksle.



26 pav. Modeliavimo aplinkos darbo vaizdas

Šaltinis: programos NetLab darbo lango vaizdas

Bandymo metu paaiškėjo, kad tinklo mazguose *Mano RT_200*, *Mano RT_300* ir *Mano RT_400* susidarė nemažos paketų išėjimo eilės. Bendra jų suma siekia 49 paketai (26 paveiksle apibraukti raudonai), t.y. susidarė paketų išėjimo kamščiai. Siekdami geresnių rezultatų, pakeistas kabelių paketų perdavimo greitis į 500ms mazguose tarp *Mano RT_200*, *Mano RT_300*, *Mano RT_400* ir *Mano DB_1* (žr. 27 pav.).

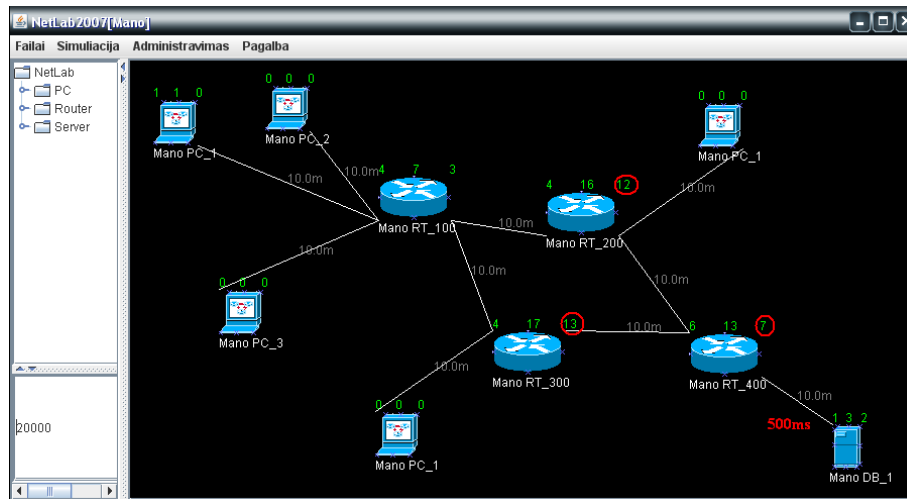


27 pav. Modeliavimo aplinkos darbo vaizdas

Šaltinis: programos NetLab darbo lango vaizdas

Matome, kad tinklo mazguose *Mano RT_200*, *Mano RT_300* eilės išnyko (27 paveiksle apibraukta raudonai), tačiau labai išaugo *Mano RT_400* mazge. Eilių suma tarp šių mazgų - 34 paketai. Tinklo būseną pagerėjo lyginant su anksčiau tirtu.

Norint pasiekti, kad eilės tolygiau pasiskirstytų tarp sistemos mazgų, pakeistas paketų perdavimo greitis tik *Mano RT_400* ir *Mano DB_1* kanale - į 500ms.



28 pav. Modeliavimo aplinkos darbo vaizdas

Šaltinis: programos NetLab darbo lango vaizdas

Eksperimentas parodė, kad rezultatai yra labai panašūs į prieš tai tirtame modelyje. Šiuo atveju paketų eilės beveik tolygiai pasiskirsto mazguose *Mano RT_200*, *Mano RT_300*, *Mano RT_400* (žr. 28 pav.). Bendra šių mazgų eilės suma 32 paketai.

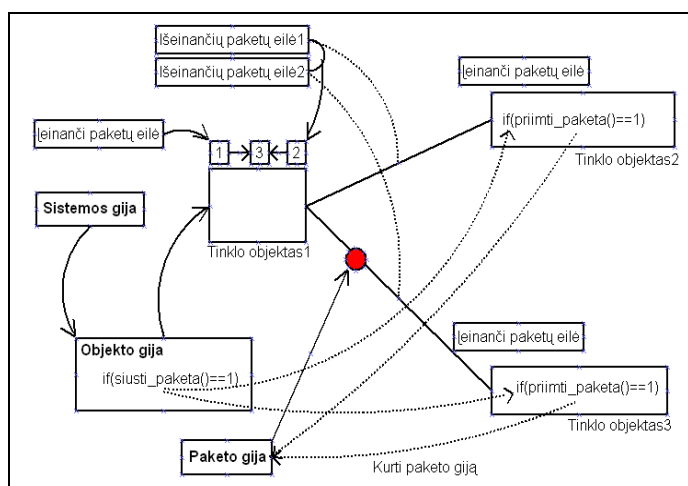
Apibendrinant gautus rezultatus, galima teigti, jog pirmiausiai apsimoka investuoti į šio kanalo spartinimą, nes jo dėka pagerėja tinklo našumas. Taip pat matoma, kad kurių kanalų vystymą neverta investuoti.

2.5. Projekto NetLab struktūra

Kuriant imitacinius modelius, lygiagrečiai turi būti vykdomi skaičiavimai, kuriuos aptarnauja procesai (gijos) ir kurie skirti aplinkos objektų įvykiams aprašyti.

Modelio imitaciją aptarnauja trijų tipų gijos:

- *Sistemos gija* – atsakinga už sistemos laiką. Sistemos gija yra gyvybinga viso eksperimento metu. Pasibaigus eksperimento laikui, ji stabdo visas gijas ir pati baigia darbą.
- *Objekto gija* – atsakinga už kuriamus paketus.
- *Paketo gija* – kai sukuriama paketo gija, ji pirmiausiai patikrina, ar yra eilių prie procesoriaus ir ar objekto procesorius laisvas. Jei užimtas, ji yra stabdoma ir padedama į įeinančių paketų eilę. Atsilaisvinusi gija pažadina pirmą į eilę atsistojusią giją (FIFO). Paskui gija tikrina, ar yra laisvas kanalas su kitu objektu. Jei kanalas užimtas, gija stabdoma ir statoma į reikiamo kanalo eilę. Atlaisvinusi kanalą gija pažadina pirmą to kanalo laukiančią giją (FIFO). Šis gijų ryšys su kitais objektais pateiktas 29 paveiksle.



29 pav. Gijų ryšys su kitais tinklo objektais

Šaltinis: sudaryta darbo autoriaus

Visą projektą sudaro klasių rinkinys, kurios atsakingos už projekto langų ir kitų sistemos komponentų bei objektų vaizdavimą, duomenų bazės prisijungimą, įvairius skaičiavimus ir kt. Visas klases galime suskirstyti į tris grupes: langų pobūdžio – kurios naudojamos langams vaizduoti, projekto pobūdžio – kurios atsakingos už vykdomo modeliavimo objektų funkcionavimą, sistemos pobūdžio – kurios atsakingos už langų bei objektų valdymą.

Langų pobūdžio klases dar galima suskirstyti į dvi grupes: sistemos administravimui, reikalingus langus ir kompiuterių tinklų modeliavimui naudojamus langus. Visą klasių hierarchiją galima pamatyti 2 priede.

Informacijos saugojimui yra naudojama duomenų bazė NetLab, kurią sudaro dešimt tarpusavyje surištų duomenų bazės lentelių. Joje saugoma visa informacija apie aprašytus sistemos objektus, eksperto sudarytas užduotis, sukurtus tinklo modelius ir kt. Visą duomenų bazės struktūrą galima suskirstyti į dvi dalis: sistemai aprašyti reikalingas duomenų bazės lenteles ir sukurtų modelių saugojimui reikalingas lenteles. Duomenų bazės struktūra pateikta 1 priede.

IŠVADOS

Atlikta nuotolinio mokymo, imitacinio modeliavimo bei kompiuterių tinklų literatūros šaltinių analizė parodė, jog naudojant imitacinį modeliavimą kur kas lengviau priimti tam tikrus sprendimus, atlikti analizę, įvairius bandymus, tyrimus ir panašiai. Šios sistemos panaudojimas priklauso nuo tam tikrų imitacinio modeliavimo priemonių, kurių yra daug ir įvairių. Vienos jų skirtos programuotojams, kitos – analitikams, tyrėjams, mokiniams ir kt. Programuotojams skirtos imitacinio modeliavimo priemonės įgalina juos kurti, tobulinti, papildyti naujomis funkcijomis bei atnaujinti šias imitacinio modeliavimo sistemas, jas dar labiau pritaikant mokymosi procesui. Nuotolinio mokymosi tobulėjimas leidžia panaudoti naujausias technologijas studijuojant kompiuterių tinklus dėl vaizdaus bei eksperimentinio mokymosi būdo. Imitacinio modeliavimo galimybių pritaikymas nuotolinio mokymosi procesui teigiamai įtakoja kompiuterių tinklų studijas.

Pasirenkant reikiamas priemones NetLab sistemai sukurti buvo atsižvelgiama į tai, jog sukurta sistema turi veikti kaip kitos nuotolinio mokymosi sistemos dalis. Dėl šios priežasties buvo pasirinkta nemokama, atviro kodo Moodle nuotolinio mokymosi sistema. Panaudota duomenų bazės valdymo sistema MySQL, nes šią sistemą visiškai palaiko Moodle sistema; ji taip yra nemokama bei atviro kodo ir IT specialistų ypač gerai vertinama. Java programavimo kalbos pasirinkimą lėmė tai, jog ji turi Java virtualią mašiną (JVM), kuri turi technologiją Java Web Start. Šios technologijos pagalba galima vykdyti nuotolinį prisijungimą prie NetLab sistemos. Sistemos objektų įvykiams aprašyti pasitelkta Skriptavimo kalba JavaSkript, nes ją palaiko daugelis interneto naršyklių, todėl praktikoje su šia skriptavimo kalba yra susidūrę daugelis informacinių technologijų specialistų.

Savarankiška NetLab nuotolinio mokymosi sistema būtų labai ribotų galimybių, nes ji yra orientuota į kompiuterių tinklų mokymą imitacinio modeliavimo pobūdžiu ir neturi daugelio funkcijų, kurias turi kitos didelės nuotolinio mokymosi sistemos. Todėl NetLab sistema buvo integruota į Moodle nuotolinio mokymosi sistemą. Svarbiausios priežastys, dėl ko pasirinkta ši sistema yra tai, jog ji plačiai taikoma, nemokama ir atviro kodo. Dėl to, lengva integruoti į ją sukurta nuotolinio mokymosi sistemą NetLab.

Sukurta sistema NetLab skaidoma į dvi dalis. Pirmoji – sistemos administravimas, į kurią įeina tinklo objektų kūrimas ir aprašymas, užduočių sudarymas bei žinių patikrinimas. Antroji dalis skirta tinklų modeliavimui ir tinklo imitavimui bei gautų rezultatų analizei. NetLab sistema yra pasiekama interneto naršyklės pagalba ir norint naudotis ja, turi būti tenkinama tik viena sąlyga – vartotojo kompiuteryje turi būti įdiegta Java virtuali mašina. Nepaisant to, kad sukurtas produktas neapima visų kompiuterių tinklų veikimo principų imitavimo, šią sistemą galima naudoti nuotoliniame kompiuterių tinklų mokyme.

ANOTACIJA

Magistro baigiamajame darbe nagrinėjama nuotolinio mokymosi svarba studijuojant kompiuterių tinklus bei šiam tikslui pritaikant imitacinį modeliavimą. Nepaliaujamai spartėjant technologinei pažangai, svarbu tobulinti ir nuotolinio mokymosi sistemas, tam, kad kompiuterių tinklų mokymasis vyktų efektyviai bei sklandžiai. Šiuo tikslu sukurta nuotolinio mokymosi sistema NetLab, kuri yra Moodle sistemos dalis bei leidžia imituoti kompiuterių tinklus, juos vaizdžiai pateikti bei palengvinti studentams mokymosi procesą.

Siekiant sukurti šią programą, buvo atlikta išsami nuotolinio mokymo, jo pritaikymo studijuojant kompiuterių tinklus literatūros analizė. Taip pat apžvelgti imitacinio modeliavimo teoriniai aspektai, kurių pagalba sukurta sistema NetLab ir ji pritaikyta nuotoliniam kompiuterių tinklų mokymui bei žinių patikrinimui.

Išbandžius nuotolinio mokymosi sistemą NetLab, paaiškėjo, kad ji gali būti pritaikoma mokymosi procese, nes jos panaudojimas platus ir kartu nesudėtingas.

SUMMARY

In master's final paper there is analysing e-learning importance during computer network studies and for this purpose adopting simulation. During unceasing rapidity of technological advance, also it is important to improve e-learning systems. This should be done because of trying that computer network e-learning studies would be effective and fluent. For this purpose was created e-learning system - NetLab, which is a part of the Moodle system and allow to simulate computer networks. Also this program presents computer networks very ostensive and help students to relieve the learning process.

With a view to create this program, it was done a comprehensive e-learning, it's adoption during computer network studies literature analysis. Also it was reviewed theoretical aspects of simulation, with which help it was created NetLab system and this system was fitted for computer networks e-learning and knowledge examination.

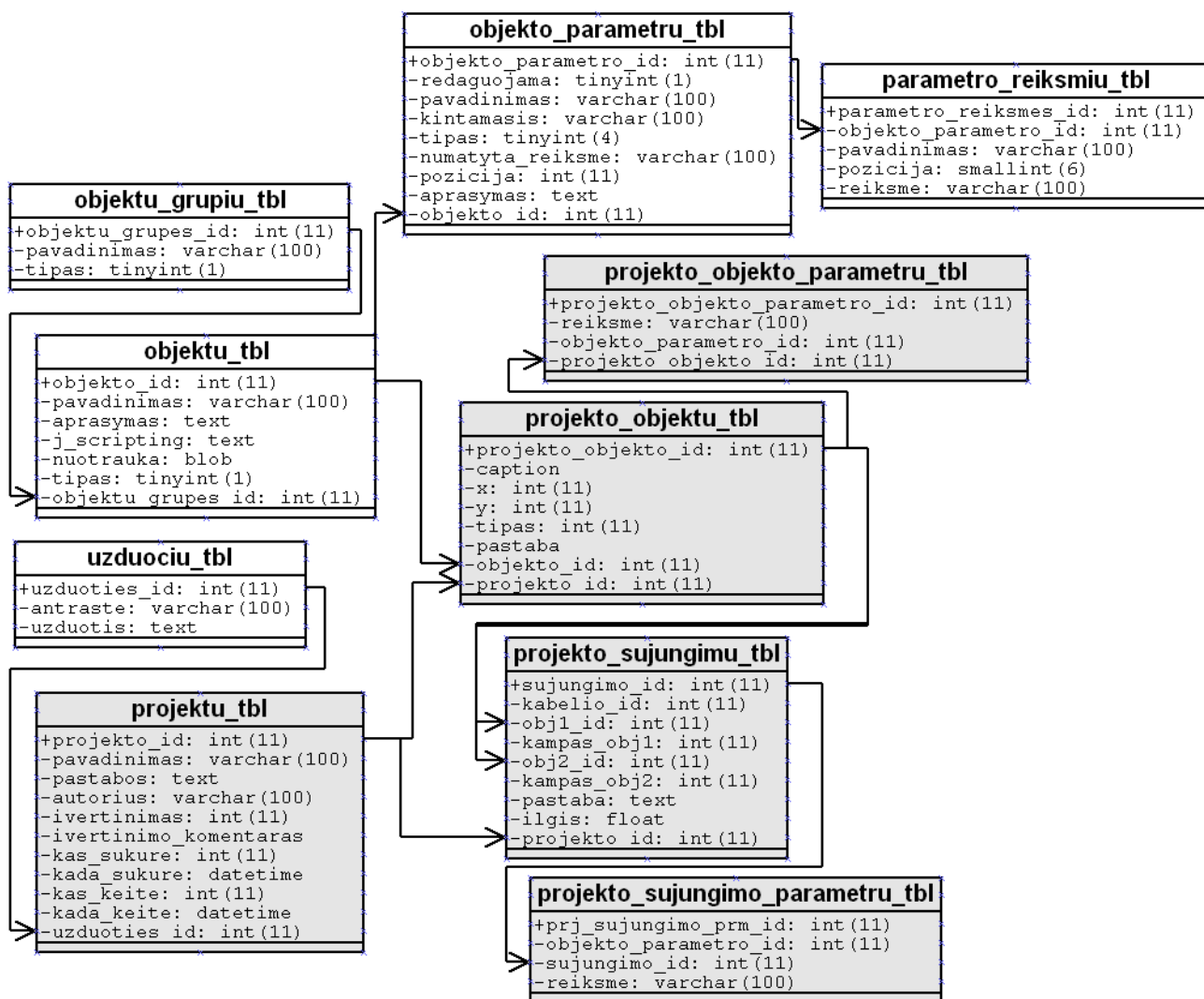
When this system was tried, emerged that it can be applicable to learning process, because it's practise is wide and simple.

LITERATŪRA

1. Giedrimas V., Giedrimienė L. (2007). *Nuotolinių studijų kursų rengimas Moodle sistemoje*. Vilnius: Vaga.
2. Giedrimas V., Tankeliavičienė L. (2004). *Nuotolinių studijų kursų rengimas Lotus LearningSpace Forum aplinkoje*. Šiauliai: VšĮ Šiaulių universiteto leidykla.
3. Kazlauskas L. (2003). *Kompiuterių tinklai. 1 dalis*. Šiauliai: Šiaulių universiteto leidykla.
4. Kreutzer W. A Framework for Web-based E-Learning of Discrete Event Simulation Concepts. Prieiga per internetą: <<http://www.simvis.org/Tagung2006/proceedings/4.3.pdf>>.
5. AnyLogic – imitacinio modeliavimo programinė įranga mokslui, gamybai bei verslui. Prieiga per internetą:
<http://www.ibn.lt/lit/anylogic/AnyLogic_imitacinis_modeliavimas_straipsnis.htm>.
6. DESMO – J. *University of Hamburg*. Prieiga per internetą: <<http://asi-www.informatik.uni-hamburg.de/themen/sim/forschung/Simulation/Desmo-J/overview.html>>.
7. Total known Moodle sites. *Moodle Statistics*. Prieiga per internetą: <<http://moodle.org/stats/>>.
8. Market Share. *MySQL*. Prieiga per internetą:
<<http://www.mysql.com/why-mysql/marketshare/>>.
9. Paketų koncepcija. Prieiga per internetą:
<http://techno.su.lt/~bernotas/kt/pak_duom_perd.htm>.
10. December J. Units of Analysis for Internet Communication. *Polytechnic Institute*. Prieiga per internetą: <<http://jcmc.indiana.edu/vol1/issue4/december.html>>.
11. Šegždaitė V. Tinklo sąsajos įrenginiai. *Vilniaus universitetas*. Prieiga per internetą:
<<http://www.kf.vu.lt/~public/ivs1/vitalija/Tinklo%20sasajos%20irenginiai.htm>>.
12. Router Architecture Overview. Prieiga per internetą: <<http://www.cs.umd.edu/~shankar/417-F01/Slides/chapter4b-aus/sld023.htm>>.
13. AnyLogic 6 Professional. Prieiga per internetą: <<http://www.xjtek.com/>>.
14. What is OMNeT++? Prieiga per internetą: <<http://www.omnetpp.org/external/whatis.php>>.
15. WebCT. *Vikipedija*. Prieiga per internetą: <<http://en.wikipedia.org/wiki/WebCT>>.
16. Prieiga per internetą: <http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2004~D_20040909_160030-86007/DS.005.0.02.ETD>.
17. Pagrindinių dalykų kūrimas ir mokymas WebCT Vista 4. Prieiga per internetą:
<http://webct.liedm.lt/index_files/webct_blackboard_vista_vadovas.pdf>.

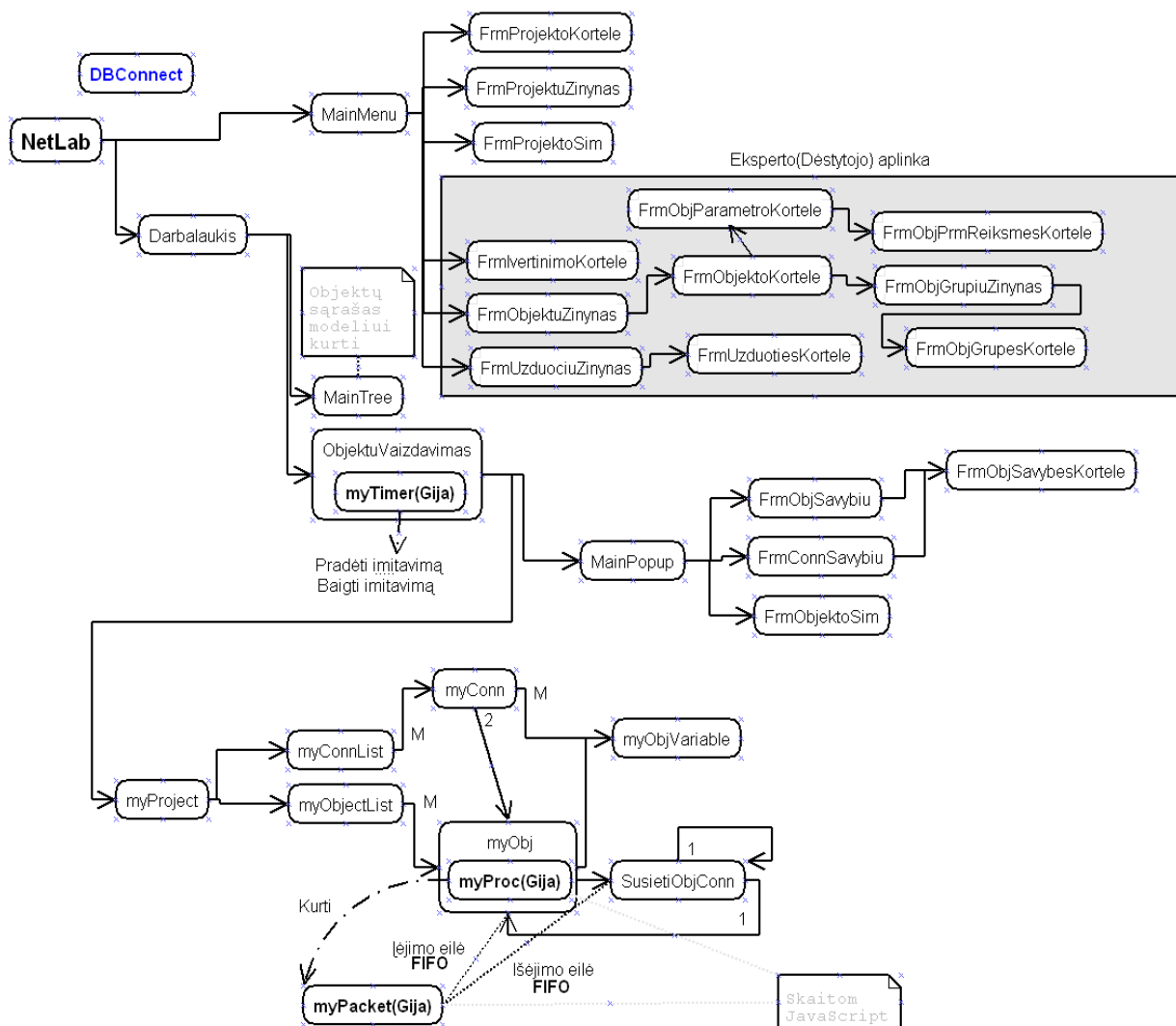
PRIEDAI

Duomenų bazės NetLab struktūra



Šaltinis: sudaryta darbo autoriaus remiantis sukurta NetLab programa

Apibendrinta NetLab programos struktūra



Šaltinis: sudaryta darbo autoriaus remiantis sukurta NetLab programa