

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PRAKTINĖS INFORMATIKOS KATEDRA

Lukas Židonis

ENERGETINIŲ SISTEMŲ NUOTOLINIS VALDYMAS

Magistro darbas

Darbo vadovas
doc. dr. A. Riškus

KAUNAS, 2004

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PRAKTINĖS INFORMATIKOS KATEDRA

TVIRTINU

(parašas) Katedros vedėjas
doc. dr. D. Rubliauskas
2004-05-10

**ENERGETINIŲ SISTEMŲ NUOTOLINIS
VALDYMAS**

Informacinių technologijų magistro baigiamasis darbas

(parašas)	Kalbos konsultantė Lietuvių kalbos katedros lektorė dr. J. Mikelionienė 2004 04 26	(parašas)	Vadovas doc. dr. A. Riškus 2004 04 26
(parašas)	Recenzentas doc. dr. J. Juozėnas 2004 05 26	(parašas)	Atliko IFM–8/3 gr. stud. L. Židonis 2004-04-26

KAUNAS, 2004

TURINYS

1. ĮVADAS	5
2. ANALITINĖ DALIS	6
2.1. PROJEKTO APLINKOS ANALIZĖ	6
2.2. KOMPIUTERIZUOTĄ VALDYMĄ REALIZUOJANTYS PROTOKOLAI	7
2.2.1. MODBUS PROTOKOLAS	7
2.2.2. EIB PROTOKOLAS	10
2.2.3. LONWORKS PROTOKOLAS	14
3. PROJEKTINĖ DALIS	19
3.1 REIKALAVIMŲ PROJEKTUOJAMAI SISTEMAI SPECIFIKACIJA	19
3.1.1. ĮVADAS	19
3.1.1.1. REIKALAVIMŲ DOKUMENTO TIKSLAI	19
3.1.1.2. LIKUSIŲ DOKUMENTO DALIŲ APŽVALGA	19
3.1.2. BENDRAS APRAŠYMAS	19
3.1.2.1. PRODUKTO PERSPEKTYVA	19
3.1.2.2. PRODUKTO FUNKCIJOS	19
3.1.2.3. VARTOTOJO CHARAKTERISTIKOS	20
3.1.2.4. BENDRI APRIBOJIMAI	20
3.1.2.5. IŠANKSTINĖS SĄLYGOS	20
3.1.3. SPECIFINIAI REIKALAVIMAI	20
3.1.3.1. FUNKCINIAI REIKALAVIMAI	20
3.1.3.2. NEFUNKCINIAI REIKALAVIMAI	21
3.1.3.3. SRITIES REIKALAVIMAI	21
3.2. DUOMENŲ STRUKTŪRA	21
3.3. PROJEKTUOJAMOS SISTEMOS ARCHITEKTŪRA	21
3.4. PROGRAMINIŲ MODULIŲ AR OBJEKTŲ SPECIFIKACIJOS	24
3.4.1. KLEINTO MODULIŲ SPECIFIKACIJOS	24
3.4.2. SAŠAJOS MODULIŲ SPECIFIKACIJOS	26
3.4.3. SERVERIO MODULIŲ SPECIFIKACIJOS	26
3.4.4. IMITATORIAUS MODULIŲ SPECIFIKACIJOS	27
3.5. TESTAVIMO MEDŽIAGA	29

4. VARTOTOJO DOKUMENTACIJA	31
4.1. SISTEMOS FUNKCINIS APRAŠYMAS	31
4.2. SISTEMOS VADOVAS	31
4.3. SISTEMOS INSTALIAVIMO DOKUMENTAS	33
4.4. SISTEMOS ADMINISTRATORIAUS VADOVAS	33
5. IŠVADOS	34
6. LITERATŪRA	35
7. TERMINŲ IR SANTRUPMŲ ŽODYNAS	35
8. PRIEDAI	37
8.1. PANAUDOJIMO ATVEJŲ UML DIAGRAMA	37
8.2. RTS MONITORIAUS KLASIŲ DIAGRAMA	37
8.3. VARTOTOJO PRISIJUNGIMO SEKŲ DIAGRAMA	39
8.4. PARAMETRO REIKŠMĖS NUSKAITYMO SEKŲ DIAGRAMA	39
8.5. PARAMETRO REIKŠMĖS KEITIMO SEKŲ DIAGRAMA	39
8.6. RTS MONITORIAUS EKRAŅŲ KOPIJOS	40
8.7. PROGRAMINĖS ĮRANGOS DISKELIS	40

1. ĮVADAS

Nuolat brangstant visų rūšių energetiniam kurui ir plėtojantis pramonei, energetinių ūkių modernizavimas tampa vis labiau aktualus. Kuo didesnė energetinė sistema, tuo jos modernizavimas sąlygoja didesnę ekonomiją.

Kompiuterizuotų energetinių sistemų pagrindinis privalumas lyginant su įprastinėmis energetinėmis sistemomis yra ekonominio pobūdžio. Šios sistemos suteikia galimybę gerokai sumažinti aptarnaujančio personalo skaičių, operatyviau reguliuoti suvartojamos energijos kiekį, tiksliau numatyti energijos poreikius, taupyti kurą, mažinti aplinkos taršą. Kaip pavyzdžiu galima būtų pateikti kelias miesto katilines, tiekiančias šilumos energiją skirtingiems mikrorajonams. Įdiegus nuotolinį valdymą, katilinės būtų valdomos iš bendros dispečerinės, o katilinėse personalas apsilankytų profilaktiškai arba avarinio atveju (tokiu būdu personalo skaičių galima būtų sumažinti kelis kartus). Kadangi aptarnaujančio personalo skaičius nedidelis, sistemos priežiūrą galima patikėti specialistams, operatyviau fiksuoti gedimus. Visa informacija matoma ekrane, todėl galima greitai nustatyti gedimo priežastį ir sumažinti dėl gedimo atsiradusius nuostolius. Kadangi informacija perduodama internetu, valdomo objekto parametrus galima stebėti iš bet kurio kompiuterio, turinčio interneto ryšį.

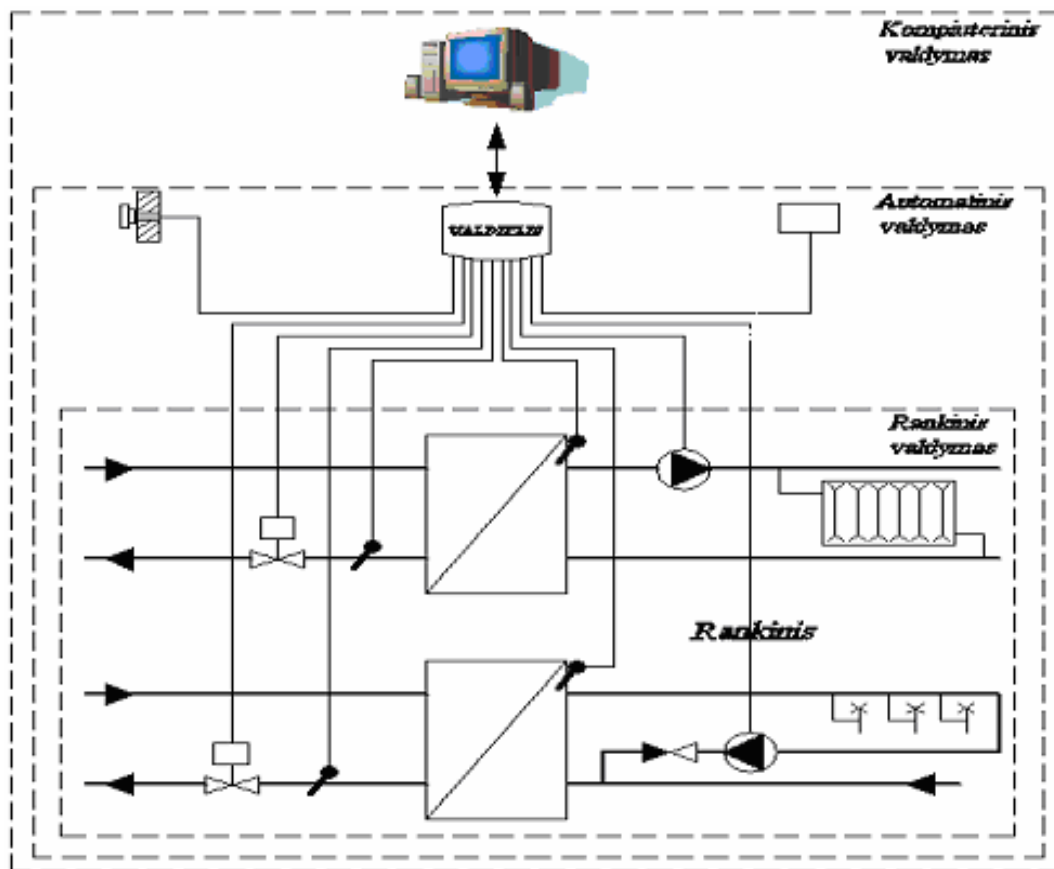
Dėl paminėtų priežasčių ir pasirinkta magistro baigiamojo darbo tema „Energetinių sistemų nuotolinis valdymas“. Kuriama realaus laiko sistema, leidžianti valdyti energetinę sistemą nuotoliniu būdu. Konkrečiu atveju energetinė sistema – katilinė. Kuriamos programinės įrangos pavadinimas „RTS“ (*Real Time System*).

2. ANALITINĖ DALIS

2.1. PROJEKTO APLINKOS ANALIZĖ

Energetines sistemas pagal valdymo technologinį lygį galima suskirstyti į (1 pav.):

1. Rankiniu būdu valdomos energetinės sistemos. Tai dažniausia techniškai pasenusios arba labai mažos galios, buitinės energetinės sistemos.
2. Automatinio būdu valdomos energetinės sistemos. Energetinę sistemą valdo specialus valdiklis (valdikliai) pagal iš anksto suderintą darbo režimą. Nors valdymas automatizuotas, katilinėje privalo būti personalas, kuris stebėtų šilumines reikšmes ir keistų darbo parametrus.
3. Kompiuterizuotos energetinės sistemos. Energetinių sistemų veikimas kontroliuojamas kompiuterinių programų nebūtinai iš to paties pastato ar net miesto. Duomenys perduodami dviem kryptim – iš energetinės sistemos (informacija apie sistemos būseną) ir į energetinę sistemą (darbo režimo keitimas, specialios užklauskos).



1 pav. Energetinės sistemos su skirtingų tipų valdymu pavyzdys

2.2. KOMPIUTERIZUOTĄ VALDYMĄ REALIZUOJANTYS PROTOKOLAI

Tam, kad skaitmeniniai duomenys būtų perduodami iš bet kokio taško A į tašką B, reikalingas taisyklių rinkinys, aprašantis kaip šie duomenys koduojami, dekoduojami ir t.t. Šis taisyklių rinkinys vadinamas protokolu. Kiekviena kompiuterizuota energetinė (ar bet kuri kita) sistema paremta duomenų perdavimo protokolu, nuo kurio tiesiogiai priklauso sistemos savybės, tokios kaip patikimumas, greitaveika, maksimalus aptarnaujamų įrenginių skaičius, sistemos kaina. Šiuo metu pasaulyje naudojama dešimtys protokolų, turbūt plačiausiai iš jų paplitęs TCP/IP (jis daugiau skirtas duomenims perduoti ir apdoroti, valdymo sistemoms jis nelabai tinkamas). Toliau panagrinėsime šiuos plačiau žinomus sistemų komunikacijų protokolus:

1. Modbus protokolas.
2. EIB protokolas.
3. LonWorks protokolas

2.2.1. MODBUS PROTOKOLAS

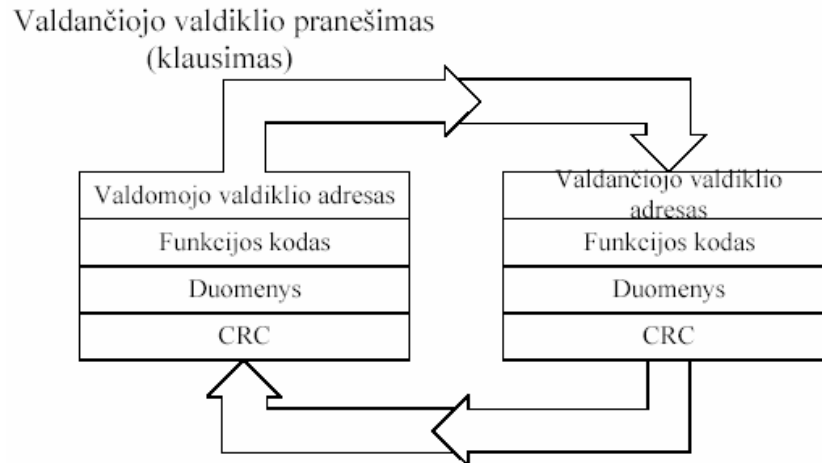
Modbus protokolas dažniausiai naudojamas energetinėse sistemose su Modicon tipo programuojamais valdikliais duomenų perdavimui organizuoti. Šis protokolas turi būtent tokią struktūrą, kurią paminėti valdikliai naudoja nepriklausomai nuo tinklo per kurį jie komunikuoja. Protokolas aprašo duomenų mainų žodžio turinio ir išdėstymo standartinį formatą.

Protokolas nustato, kaip kiekvienas sistemoje funkcionuojantis valdiklis identifikuoja savo adresą, atpažįsta jam siunčiamą pranešimą, nustato reikalingą atlikti veiksmą, išpakuoja jam skirtus duomenis. Jeigu reikia, valdiklis suformuoja gražinamą pranešimą.

Kitų tipų tinkluose (ne Modbus) šis protokolas yra užkoduojamas tinklo naudojamuose duomenų perdavimo paketuose.

Duomenims perduoti Modbus protokolu naudojamos RS232 arba RS485 sąsajos. Sistemoje dirba vienas vedantysis (*host* arba *master*) valdiklis ir vedamieji (*slave*) valdikliai. Tik vedantysis valdiklis gali inicijuoti mainus. Kiti valdikliai tik formuoja atsakymus į paklausimus. Vedantysis valdiklis gali individualiai komunikuoti su vedamaisiais valdikliais arba jiems visiems siųsti bendrą pranešimą. Vedamieji valdikliai gražina atsakymą vedančiajam po to, kai į juos kreipiamasi individualiai.

Klausimo – atsakymo ciklas. Klausimo – atsakymo ciklas, funkcionuojantis MODBUS protokole, pateiktas 2 pav.



2 pav. Klausimo – atsakymo ciklas Modbus protokole

Vedantysis valdiklis, formuodamas klausimą funkcijos kodo lauke, praneša adresuojamam vedamajam valdikliui, ką jis turi daryti. Duomenų lauke yra pateikiama papildoma informacija, reikalinga tai funkcijai įvykdyti. CRC laukas reikalingas pranešimo tikslumui nustatyti. Pavyzdžiui, vedantysis liepia vedamajam nuskaityti registrų turinius ir juos gražinti. Vedamasis valdiklis formuoja atsakymą, kuris yra reakcija į paklausimą (funkcijos kodas sutampa). Duomenų lauke yra talpinamos atitinkamos registrų reikšmės ar valdiklio būseną. Jeigu atsiranda klaidų, atitinkamai yra pakeičiamas funkcijos kodas. Jis praneša, kad atsakyme yra pranešimas apie klaidą. CRC laukas taip pat reikalingas pranešimo tikslumui nustatyti.

Modbus protokole gali būti naudojami du mainų režimai – ASCII arba RTU. Derindamas ryšio sistemą vartotojas turi pasirinkti vieną iš jų, kartu nustatydamas vienodus asinchroninių mainų režimus (mainų sparta, lyginumas ir kt.).

Pasirinkus **ASCII** (*American Standard Code for Information Interchange*) režimą, kiekvienas baitas yra perduodamas dviem ASCII simboliais. Šio darbo režimo pagrindinė ypatybė yra tai, kad tarp simbolių gali būti laiko tarpas (iki 1s). Šiame darbo režime skaičiai koduojami šešioliktainiais simboliais (0 .. 9, A ... F), pridedant po vieną START ir STOP bitus, be lyginimo kontrolės. Paketo gale yra perduodamas LRC (*Longitudinal Redundancy Check*).

Kai valdikliai komunikuoja **RTU** (*Remote Terminal Unit*) režime, kiekvienas baitas yra koduojamas dviejų – keturių bitu šešioliktainiais simboliais. Taigi šis darbo režimas užtikrina greitesnius duomenų mainus. Paketo gale yra perduodamas CRC (*Cyclical Redundancy Check*).

Abiejuose darbo režimuose pranešimai turi pradžios ir galo taškus. Tai leidžia imtuvui susirasti pranešimo pradžią, nuskaityti adresą ir identifikuoti jį. ASCII režime pranešimas prasideda nuo (:) – dvitaškio simbolio (ASCII 3A hex), pabaigos simboliai yra CR ir LF. Pranešimo struktūra pateikta 1 lentelėje:

1 lentelė. Pranešimo struktūra ASCII darbo režime

START	ADRESAS	FUNKCIJA	DUOMENYS	LRC	STOP
0X3A – 2 baitai	2 baitai	2 baitai	N baitų	2 baitai	0x0D ir 0x0A 4 baitai

Tarp simbolių gali būti tarpai, kurių trukmė gali siekti 1 sekundę. Jei per sekundę simbolis neperduodamas – formuojamas klaidos požymis.

RTU režime pranešimas prasideda „ramybės“ intervalu, kurio trukmė – 3,5 simbolio. Po jo seka įrenginio adresas. Pranešimo pabaiga yra 3,5 ilgio „ramybės“ intervalas, po kurio gali sekti kitas pranešimas. Visi perduodami simboliai yra šešioliktainiai 0 .. 9, A .. F. Šiame režime perduodamų simbolių seka negali nutrūkti. Po duomenų perduodamas 16-os bitų CRC kodas. Pranešimo struktūra pateikta 2 lentelėje:

2 lentelė. Pranešimo struktūra RTU darbo režime

START	ADRESAS	FUNKCIJA	DUOMENYS	LRC	STOP
T1 – T4	Baitas	Baitas	N baitų	2 baitai	T1 – T4

Vedamųjų įrenginių adresavimas. ASCII režime vedamojo įrenginio adresas yra perduodamas dviem baitais, o RTU režime – vienu. Vedamųjų įrenginių adresai yra iš diapazono 0 .. 247. Vedantysis įrenginys, kreipdamasis į vedantįjį, pranešimo adreso lauke formuoja adresą. Vedamasis įrenginys formuodamas atsakymą adreso lauke įrašo savo adresą. Taip vedantysis įrenginys identifikuoja vedamąjį. 0 adresas naudojamas kaip bendras visiems vedamiesiems įrenginiams. Funkcijos kodas perduodamas dviem baitais ASCII režime, vienu baitu – RTU režime. Funkcijų kodai gali būti iš srities 1 .. 255. Vedantysis įrenginys, formuodamas funkcijos kodą vedamajam, nustato pastarojo veiksmą (gražinti valdymo registrų turinius, atlikti matavimą, gražinti būseną ir pan.).

Vedamasis, formuodamas atsakymą, talpina tą patį funkcijos kodą (echoes). Jei vedamasis įrenginys funkcijos kodo neatpažįsta, jis atsakymo funkcijos kodo vyriausioje skiltyje formuoja loginį vienetą.

Duomenų lauko struktūra. Duomenų lauką gali sudaryti ASCII simbolių (ASCII režime) poros arba pavienių simbolių (RTU režime) seka. Duomenų lauke, kurį vedantysis

įrenginys siunčia vedamajam, yra papildoma informacija, kurią vedamasis įrenginys panaudoja tam tikrai funkcijai atlikti. Tai gali būti registrų adresai, duomenų lauke esantis baitų skaičius ir kt. Pavyzdžiui, jeigu vedantysis įrenginys paprašo gražinti registrų turinius, tai duomenų lauke pranešamas tų registrų kiekis. Jeigu vedantysis įrenginys perduoda registrų turinius į vedamąjį įrenginį, tai duomenų lauke nurodomas registrų kiekis ir jų turiniai. Jeigu nėra klaidų, tai vedamojo įrenginio atsakyme bus talpinami duomenys. Jeigu jų yra, duomenų lauke bus talpinamas klaidos kodas. Kai kurios funkcijos gali būti be duomenų lauko.

Duomenų formatas. Dirbant su MODBUS naudojamas UART arba ACE generuojamas duomenų formatas. ASCII režime naudojamas 7,N,1 arba 7,Y,1 formatas (Start, 7 duomenų bitai, lyginumo kontrolės skiltis (ar be jos) ir vienas stop bitas). RTU režime naudojamas 8,N,1 arba 8,Y,1 formatas (Start, 7 duomenų bitai, lyginumo kontrolės skiltis (ar be jos) ir vienas stop bitas). Baitai perduodami jaunesniu bitu į priekį (*LSB first*). Klaidų kontrolei yra naudojami du metodai:

- Lyginumo kontrolė (kiekvienam simboliui atskirai),
- Viso pranešimo kontrolė su LRC (ASCII režime) ir su CRC (RTU režime).

Pirmuoju atveju kiekvienam perduodamam pranešimo simboliui yra formuojama lyginumo kontrolės skiltis. Antruoju atveju skaičiuojama LRC arba CRC. Priėmimo vietoje taip pat atliekamas jos skaičiavimas ir vykdomas palyginimas su priimtąja.

Pagrindiniai Modbus protokolo privalumai:

- Palyginti nedidelė įrangos kaina;
- Standartinė sąsaja – paprasta perduoti duomenis į kompiuterį;
- Palyginti paprastas sistemos instaliavimas ir aptarnavimas.

Pagrindiniai Modbus protokolo trūkumai:

- Palyginti nedidelių galimybių protokolas;
- Riboto pritaikymo protokoliai.

2.2.2. EIB PROTOKOLAS

„Protokolas EIB (*European Installation Bus*) – tai tarptautinės asociacijos EIBA, jungiančios dešimtis žinomų elektrotechnikos gamintojų, bendroji europinė sistema, Šiuo metu EIBA, kurios būstinė yra Briuselyje, vienija apie 120 kompanijų, užimančių apie 80 proc. elektros instaliacijos gaminių rinkos Europoje. Be standartų ir normų kūrimo, EIBA kontroliuoja kokybę ir įvairių gamintojų produkcijos suderinamumą.

EIB yra skaitmeninės technikos raidos rezultatas. Tai sistema, jungianti į bendrą magistralę įvairius pastato instaliacijos įrenginius. Ji vykdo paprastas ir kompleksines valdymo ir kontrolės funkcijas. Tokia sistema pakeičia klasikinę elektros instaliaciją, kuri negali patenkinti didėjančių poreikių. Tradiciniai instaliaciniai jungikliai pakeisti skaitmeniniais įrenginiais, kurie informacijos mainus vykdo magistraliniu kabeliu, jungiančiu visus sistemos elementus. Magistralinis 4 gyslų kabelis (2 gyslos atsarginės) klojamas šalia 230 V kabelių, todėl instaliacijai prireikia iki 60 proc. mažiau kabelių, supaprastėja sistemos įdiegimas, projektavimas ir montavimas, o galimybių gerokai padaugėja. Šia sistema galima valdyti, reguliuoti ir kontroliuoti elektros įrenginius, o per juos – apšvietimą, šildymą, oro kondicionavimą ir vėdinimą, žaliuzes, praėjimo kontrolės sistemas ir kt. Sistema yra decentralizuota, neturinti pagrindinio valdymo centro. Kiekvienas jutiklis, daviklis ar vykdymo elementas turi savo valdiklius, kurie kompiuteriu užprogramuojami sistemos instaliavimo galutiniame etape. EIB sistemai kompiuteris nėra būtinas, tačiau gali būti naudojamas kaip centralizuotas valdymo ir pavaizdavimo modulis. Specialiais EIB prievadais per telefono tinklą, internetą ar kitų sistemų centrinius pultus galima automatizuoti ir dispečerizuoti pastatus.“ [Karpovičius: 2003: 19].

Pagrindiniai sistemos privalumai:

- vienas bendras informacinis laidas (sistema skaidri, mažiau laidininkų, mažesnė gaisro rizika, pigus ir lengvas aptarnavimas);
- lengvai realizuojami sudėtingi vartotojo reikalavimai;
- didelis lankstumas (tolesnė sistemos plėtra arba konfigūravimas nereikalauja keisti kabelių sistemos);
- pastato eksploatacinės elektros energijos taupymas;
- labai patikimas veikimas, atsparumas avaringumui.

Instaliuotų įrenginių valdymo sistema turi daug pritaikymo galimybių. Pateiksiu keletą pavyzdžių:

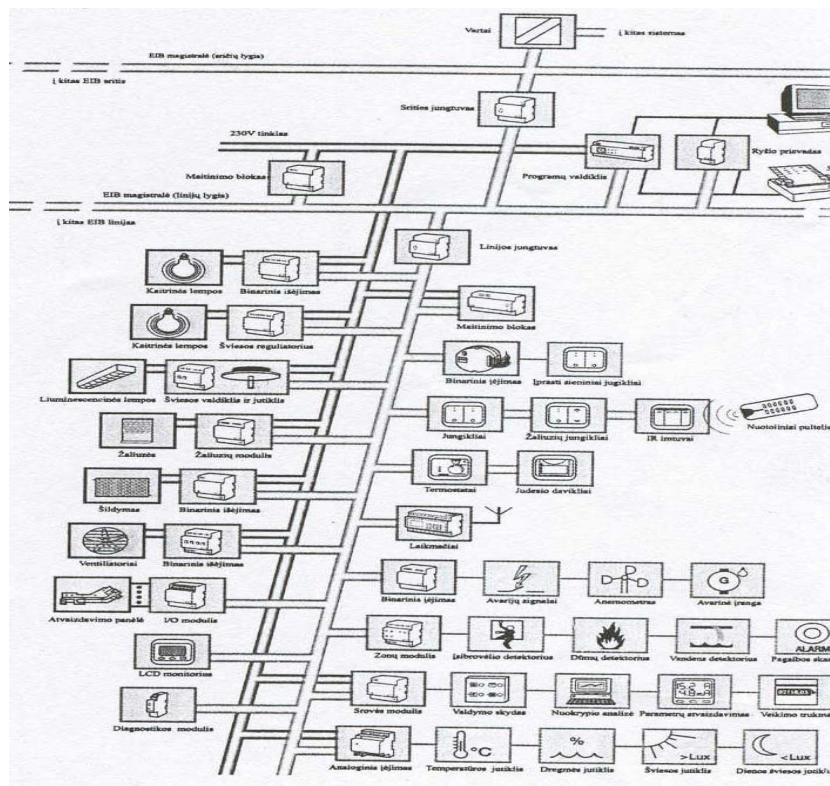
- bendras šildymo ir apšvietimo sistemos valdymas leidžia sumažinti temperatūrą tose patalpose, kuriose judesio daviklis nefiksuoja žmonių judėjimo;
- laikmatis, valdantis bet kurios pastato dalies apšvietimą, gali nustatinėti ir patalpų temperatūrą;
- judesio davikliai ir langų atidarymo jutikliai, veikiantys kaip apsaugos sistemos elementai, gali būti panaudoti apšvietimui ir šildymui valdyti (lango atidarymas

sumažina nustatytos temperatūros lygį, judesio daviklio signalai junginėja apšvietimą);

- vasarą žaliuzių automatinis uždarymas gali sumažinti patalpos išilimą – taip sumažinama kondicionavimo įrangos vartojama galia.

Pagrindinis sistemos trūkumas – didelė kaina.

Neseniai sistema EIB tapo standartu Europos Sąjungos valstybėse ir jau dominuoja intelektualią elektros instaliacijų sferoje. Šios sistemos naudotojai turi nuolatinės tobulinimo galimybes, nepriklausančias nuo gamintojo, aptarnavimą ir patikimumą, pagrįstą tūkstančiais realizuotu objektų.



3 pav. EIB sistemos schema

Techninis aprašymas.

EIB sistema savo topologija yra panaši į elektros tinklus (3 pav.). Visi sistemos mazgai (vartotojai) yra sujungiami į vieną tinklą. Kadangi vartotojus jungiantys laidai yra išvedžiojami kartu su elektros tiekimo laidais, instaliacijos kaina santykinai nedidelė. EIB sistemos įrenginiai sujungiami standartinių MSR PYCYM arba J-Y (St)Y 2 x 2 x 0.8 telekomunikacinių kabelių poromis. Kabeliai gali būti teisiams sienose, vamzdžiuose ir kt. – ten, kur jį įmanoma nutiesti

kartu su elektros (230 V / 400 V) tiekimo linijoms. Linija skirta ne tik informacijai tarp jutiklių ir pavarų perduoti, bet ir kaip saugus žemos įtampos šaltinis šiems įrenginiams. Tipiniai jutikliai – jungikliai, mygtukai, termometrai, šviesumo ir judesio jutikliai yra komandų iniciatoriai. Pavaros – kontrolės įrenginiai, srovės/įtampos reguliatoriai, jungikliai, ekranai dirba kaip komandų imtuvai .

EIB struktūra – hierarchinė. Tai reikalinga tam, kad jau įrengtą sistemą būtų galima plėtoti. Pačios paprasčiausios komandos vykdomos tiesiogiai fizinių įrenginių. Kiekvienoje linijoje gali būti prijungiami iki 64 įrenginių. Iki 12 tokių linijų gali būti sujungtos per linijų kartotuvą į vieną sritį. Iki 15 sričių gali būti sujungiamos per sričių kartotuvą. Be to, visa ši sistema gali keistis duomenimis su kitomis sistemomis per tam tikras sąsajas. Galimybėms padidinti gali būti instaliuojamas valdiklis, atliekantis šias funkcijas:

- laikrodis,
- įvykių aparato apdorojimas,
- prisijungimas,
- diagnostikos ir programavimo įrenginių prijungimas,
- greitesnė kontrolė.

Kiekviena sistemos linija turi nepriklausomą maitinimą. Tokiu būdu visa sistema išlieka funkcionali vienos iš linijų maitinimo dingimo atveju – ši savybė užtikrina aukštą patikimumą. Sistemos skirstymas į linijas ir sritis taip pat turi privalumą – lokalus vienos linijos ar srities duomenų srautas padidina visos sistemos pralaidumą. Tai pasiekama naudojant linijos kartotuvą – telegramos skirtos tos pačios linijos įrenginiams neišsiunčiamos į kitas linijas ar sritis. Tuo pačiu galimas duomenų keitimasis ir tarp dviejų skirtingų linijų. Tos pačios taisyklės galioja ir sričių kartotuvams. Atskirų įrenginių suskirstymas į linijas ir sritis neapriboja įrenginių nepriklausomybės.

Kadangi sistemos struktūra hierarchinė, atsiranda galimybė lengviau atlikti tikrinimus ir diagnostiką. Sistema paprasta: pradedama diegti nuo vienos linijos, vėliau, didėjant reikalavimams, sistema plečiama. Sistema gali būti išplėsta iki 12480 įrenginių.

EIB sistema sudaro sąlygas įgyvendinti nuo gamintojo nepriklausančius sprendimus. Dažnai gyvenamuose ar pramoniniuose pastatuose sistemos būseną turi būti ne tik registruojama, bet ir pranešama operatoriui ir atvaizduojama. Gali būti atliekamos šios operacijos:

- operacijos signalizavimas
- techniniai aliarmai
- pastato būsenos sekimas

- personalo monitoringas ir judesių ataskaitos

EIB sistemos informacija gali būti naudojama dvejopai – visi duomenys gali būti gaunami kaip informacija per sistemos liniją ir visi duomenys gali būti siunčiami linijos įrenginiams. Taigi jutikliai gali būti naudojami dvejopai. Dėl šios priežasties sistema tampa paprasčiau naudojama, o eksploatacija pigesnė.

Pavyzdžiui, šviesa gali būti kontroliuojama ta pačia sistemos linija, kuria iš namo vidaus ir išorės stebimi šie objektai:

- atviros durys ir langai,
- pastato viduje ir išorėje esančių objektų judėjimas,
- šildymo ir vėdinimo įrenginiai,
- liftų veiklos sutrikimai,
- įvairių talpų lygis.

Sistemos įrenginiai gali būti instaliuojami įrenginių skyduose, paviršiuose ar tiesiogiai mechanizme, gali būti paslėpti kabelių sistemose ir kt. Instaliuojami įrenginiai esant reikalui gali būti specialaus dizaino. Visi šie įrenginiai sujungiami tarpusavyje EIB linija. Sistemos kartotuvus ir linijos terminatoriai įrengiami dėžutėje ant sienos. Galutinis įrenginys (pvz., mygtukas) sujungiamas su sistemos kartotuvu.

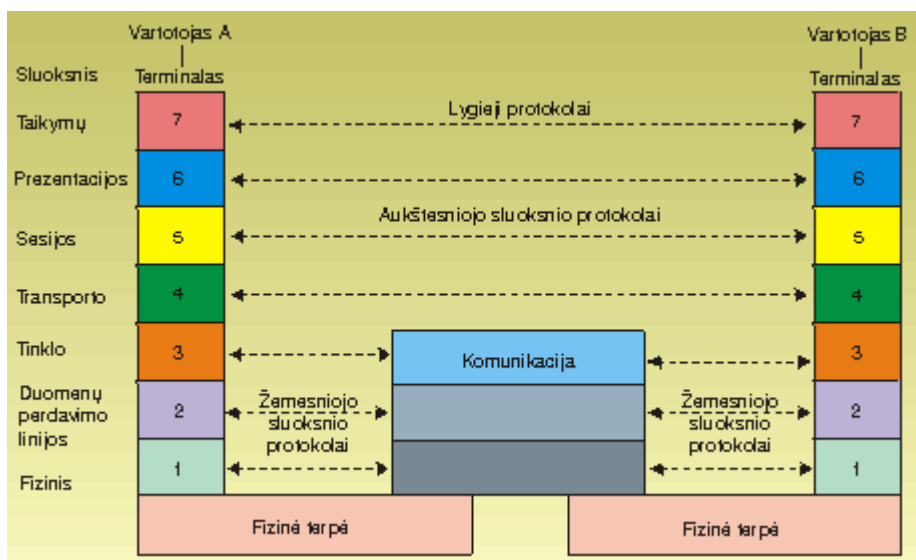
Sistemos įrenginių vieta parenkama pagal instaliacijos reikalavimus užtikrinant optimalų visų įrenginių išdėstymą. Įrenginių skyduose montuojami įrenginiai sujungiami duomenų rinktuvu, kuris užmaunamas ant DIN bėgelio.

2.2.3. LONWORKS PROTOKOLAS

LonWorks protokolas (arba *ANSI/EIA 709.1 Control Networking Standart*) – Echelon firmos sukurtos ir palaikomos valdymo ir kontrolės sistemų pagrindas. Protokolas suteikia galimybę įrenginiuose instaliuotoms programoms siųsti ir gauti žinutes tinklu nežinant tinklo tipologijos, kitų įrenginių funkcijų, adresų ir t. t. Protokolas suteikia galimybę nustatyti apribotą transakcijos periodą, prioritetinį ar neprioritetinį duomenų perdavimą, žinutės gavimo patvirtinimo reikalavimą, per nuotolį konfigūruoti tinklo adresus, parametrus, atsiųsti įrenginių programas, pranešti apie tinklo problemas, paleisti, stabdyti ar nutraukti įrenginių programų veikimą.

LonWorks protokolo struktūra – sluoksninė, duomenų mainai organizuojami paketais, komunikuojantys įrenginiai – lygiateisiai. Lygiai kaip ir Internet ar Ethernet protokolai,

LonWorks sukurtas pagal Tarptautiniu Standartų Organizacijos (ISO) Atvirųjų Sistemų jungties modelio (*Open System Interconnect – OSI*) reikalavimus, tačiau skirtingai nuo Internet ar Ethernet, LonWorks labiau skirtas ne duomenų apdorojimui bet kontrolės sistemoms. Kad užtikrinti šių sistemų greitą ir užtikrintą veikimą, LonWorks sudarytas iš septynių OSI sluoksnių (pav. 4).



4 pav. Sluoksniuotas OSI modelis

Sluoksninė OSI modelio sandara nustato kiekvienam sluoksniui tam tikrą fiksuotą tikslų skaičių. Visuomet egzistuoja sąsajos su viršuje ir apačioje esančiais sluoksniais ir su jais susijęs antraščių apdorojimas. Beveik kiekvieno sluoksnio paslaugų gali pageidauti keletas viršuje esančio sluoksnio vartotojų. Toks bendras žemesniojo sluoksnio paslaugų vartojimas yra vadinamas sutankinimu (*multiplexing*). Pageidautina (visų pirma klaidų aptikimo požiūriu) sudalinti informaciją į riboto ilgio blokus arba segmentus. Šis procesas yra vadinamas segmentavimu. Kita vertus, dėl to atsiranda surinkimo (*reassembling*) poreikis. Modelyje nėra atmetama galimybė, jog kiekviename sluoksnyje segmentų ilgis bus skirtingas. Taigi norint juos visus surinkti ir pristatyti į vietą tinkama tvarka, verta blokus sunumeruoti ir priėmimo taške tuos numerius patikrinti. Panašus procesas yra vadinamas nuoseklinimu (*sequencing*). Kiekviename sluoksnyje gali būti atliekamas ir srauto valdymas (*flow control*).

Visa LonWorks protokolu perduodama informacija sudaryta iš vieno ar daugiau paketų. Kiekvienas paketas sudarytas iš kintamo skaičiaus baitų ir kompaktiškai aprašo kiekvienam iš 7 sluoksnių reikalingus duomenis. Kompaktiškas aprašas leidžia LonWorks paketams būti labai trumpiems, dėl to sumažinama kiekvieno LonWorks įrenginio instaliavimo kaina.

Kiekvienas įrenginys ar kanalas nagrinėja paketus ir stebi ar žinutės skirtos jam. Jei taip, įrenginys apdoroja paketą ir nustato, ar atsiųsti duomenys įrenginio programai, ar tai – tinklo aptarnavimo paketas. Duomenys, skirti įrenginio programai, jai perduodami ir, jei reikalinga, išsiunčiamas patvirtinimas ar atsakymas paketo siuntėjui.

Kanalų tipai.

LonWorks protokolas nėra priklausomas nuo sąsajos tipo – tai LonWorks įrenginiams suteikia galimybę komunikuoti per bet kokią fizinę perdavimo liniją. Dėl šios priežasties, duomenų mainams organizuoti gali būti panaudojama visa eilė kanalų tipų.

Kanalas – tai fizinė komunikavimo terpė (tokia kaip vyta pora ar elektros linijos), prie kurios per siųstuvus – imtuvus (priklausančius nuo kanalo tipo) prijungta grupė LonWorks įrenginių. Kiekvienas kanalo tipas turi specifines charakteristikas, tokias kaip maksimalus prijungiamų įrenginių kiekis, signalo dažnis, atstumas (3 lentelė).

3 lentelė. Plačiai naudojami LonWorks kanalų tipai

Kanalo tipas	Linijos tipas	Signalų dažnis	Siųstuvai - imtuvai	Maksimalus įrenginių skaičius	Maksimalus atstumas
TP/FT-10	Vytos poros, laisva, magistralės topologija	78kbps	FTT-10, FTT-10A, LPT-10	64-128	500m (laisva topologija) 2200m (magistralės topologija)
TP/XF-1250	Vytos poros, magistralės topologija	1.25Mbps	TPT/XF-1250	64	125 m
PL-20	Elektros linija	5.4kbps	PLT-20, PLT-21, PLT-22	Priklauso nuo aplinkos	Priklauso nuo aplinkos
IP-10	LonWorks per IP	Nustatoma IP tinklo	Nustatoma IP tinklo	Nustatoma IP tinklo	Nustatoma IP tinklo

Ypatingai reikšmingas laisvos topologijos, vytytos poros kanalas TP/FT-10. Šis kanalas leidžia įrenginius sujungti vienu vytytos poros kabeliu bet kokių išdėstymų – nėra apribojimų segmentų ilgiams, leistina šakota topologija – vienintelis apribojimas – maksimalus kabelio ilgis tinklo segmentui.

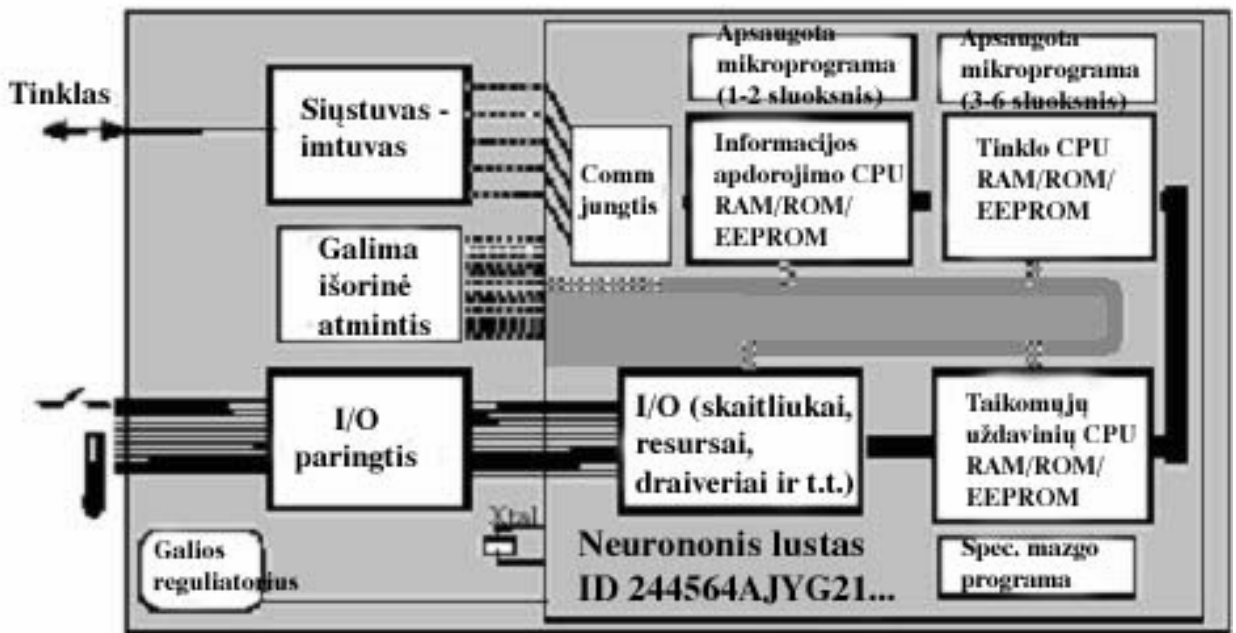
Neuroninis lustas.

Tam, kad standartinį įrenginį padaryti bendraujančiu LonWorks sistemoje, reikalingas neuroninis lustas – puslaidininkiais elementais paremta integrinė schema, suteikianti įrenginiui galimybę keistis duomenimis su kitais įrenginiais LonWorks protokolu. Neuroninį lustą sudaro trys procesoriai, apdorojantys komunikavimo ir programinius duomenis. Kiekvieno įrenginio gamintojas sukuria specialią programą, dirbančią luste, ir I/O įrenginius, kad keistis informacija su lustu. Echelon kompanija sukūrė pirmąjį originalų neuroninį lustą, o šiuo metu daugybė

Echelon partnerių, tokių kaip Motorola, Toshiba, Cypress Semiconductor masiškai gamina lustus įvairiems įrenginiams.

Neuroninis lustas sudarytas iš daugybės procesorių, skaitymo – rašymo ir tik skaitymo atminčių (RAM ir ROM), I/O sistemų. Tik skaitymo atmintis (ROM) savyje turi operacinę sistemą, LonWorks protokolą, I/O funkcijų biblioteką. Lusto neištrinamoje atmintyje esantys derinimo duomenys ir programa atsiunčiama pirmą kartą paleidus įrenginį per LonWorks tinklą. Lusto pavyzdys pateiktas 5 paveiksle.

Lusto gamybos metu jam suteikiamas nuolatinis pasauliniu lygiu unikalus 48 bitų ilgio adresas, vadinamas Neuronu ID.



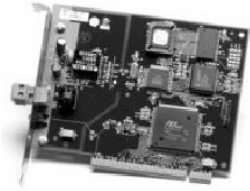
5 pav. LonWorks lusto schemos pavyzdys.

LonWorks sąsajos ir programinė įranga.

Aptarėme, kaip funkcionuoja LonWorks protokolas, kaip informacija perduodama iš standartinių įrenginių į LonWorks tinklą.

Tam kad duomenis perduoti į kompiuterį naudojami dviejų tipų įrenginiai:

1. Tinklo sąsajos adapteris (*network interfaces*) (6 pav);
2. Maršrutizatoriai (*routers*), (7 pav);



6 pav. Tinklo sąsajos adapteris



7 pav. Maršrutizatorius.

Maršrutizatoriai turi daug funkcijų, viena iš jų – transformuoti LonWorks protokolą į TCP/IP protokolą. Maršrutizatorius yra nepriklausomas įrenginys, dažniausiai montuojamas objektuose, kuriuose nereikalinga atlikti stebėjimo ir kontrolės, vadinasi, nereikia kompiuterio.

LonWorks tinklo sąsajos adapteris montuojamas į kompiuterį (kaip ir, pavyzdžiui, Ethernet tinklo adapteris). Senesnio modelio sąsajos adapteriai būdavo jungiami per ISA magistralę, šiuolaikiniai – per PCI magistralę. Skirtingai nei maršrutizatorius, sąsajos adapteris negali veikti autonomiškai ir skirtas tiesiogiai perduoti duomenis iš LonWorks tinklo į kompiuterį.

LonWorks programinė įranga skirta duomenims pateikti į standartinės programavimo aplinkas (Visual Basic, Si) arba galutinėms su vartotoju betarpiškai dirbančioms grafinėms programoms (paprastai sukurtoms specializuotomis tam skirtomis programomis). Duomenims į standartinės programavimo aplinkas perduoti (realiame laike) naudojami DDE, Activex, OPC serveriai.

LonWorks sistemos privalumai:

- Didelės protokolo galimybės
- Sistemos patikimumas.
- Nesudėtingas sistemos išplečiamumas.
- Didelis linijos tarp dviejų labiausiai nutolusių įrenginių ilgis.
- Suderinama su šiuolaikinėmis kompiuterinėmis sistemomis

Pagrindinis LonWorks sistemų trūkumas – didelė techninės ir programinės įrangos kaina. Pagal šį rodiklį LonWorks sistemos atsilieka ir nuo Modbus ir nuo EIB sistemų.

3. PROJEK TINĖ DALIS

3.1. REIKALAVIMŲ PROJEKTUOJAMAI SISTEMAI SPECIFIKACIJA

3.1.1. ĮVADAS

3.1.1.1. REIKALAVIMŲ DOKUMENTO TIKSLAI

Aiškiai ir nedviprasmiškai apibrėžti kuriamam programinės įrangos produktui keliamus reikalavimus. Tikslus reikalavimų specifikavimas padeda išvengti nesusipratimų su užsakovu, o taip pat garantuoja, kad kūrimo proceso eigoje užsakovo reikalavimai bus suprasti teisingai ir kuriamas produktas maksimaliai atitiks jo poreikius.

3.1.1.2. LIKUSIŲ DOKUMENTO DALIŲ APŽVALGA

Sekančiame šio dokumento skyriuje apžvelgiama bendros produkto savybės. 3.1.3. skyriuje išvardinami funkciniai, nefunkciniai ir srities reikalavimai.

3.1.2. BENDRAS APRAŠYMAS

3.1.2.1. PRODUKTO PERSPEKTYVA

Kuriamo produkto – RTS monitoriaus – klientinė dalis gali būti panaudota kaip sudėtinė bet kurios WWW svetainės dalis.

Taip pat, RTS monitoriaus serverinė dalis nėra priklausoma nuo pateikiamos klientinės dalies – serverio eksportuojamus metodus gali kviesti ir kitos programos, t.y. galimybė naudotis šia sistema gali būti integruota į bet kurią Java kalba rašomą programą.

Sistemos virtualių procesų imitatorius gali būti pakeistas realiai veikiančiu valdikliu ir su nežymiais modifikavimais sistema gali būti pritaikyta veikti realiai.

3.1.2.2. PRODUKTO FUNKCIJOS

Pagrindinės produkto funkcijos yra:

1. Serveryje imituojamų reikšmių perdavimas nustatytu laiko intervalu klientinei daliai;

2. Nustatymo reikšmių perdavimas iš klientinės – serverinei programos daliai.
3. Informacijos apie realiai veikiančius procesus perteikimas vartotojui
4. Vartotojo pageidaujimų pakeitimų perdavimas klientinei sistemos daliai.

3.1.2.3. VARTOTOJO CHARAKTERISTIKOS

Vartotojas turi turėti priėjimą prie kompiuterio su ryšiu į internetą. Taip pat tame kompiuteryje turi būti kokia nors WWW naršyklė (pvz., Internet Explorer, Netscape Navigator, Mozilla ar kt.) ir Javos virtuali mašina (*Java Virtual Machine*) - JVM. Jei JVM nėra suinstaliuotas, bus automatiškai pasiūlyta jį parsisiųsti ir suinstaliuoti. Vartotojas privalo gerai žinoti katilinės veiklos specifiką, būti apmokytas dirbti su programa bei registruotas serveryje.

3.1.2.4. BENDRI APRIBOJIMAI

Prie sistemos negali būti prisijungę keli vartotojai vienu metu. Prie sistemos gali jungtis tik vartotojas, administratoriaus užregistruotas į *vartotojai.txt* failą.

3.1.2.5. IŠANKSTINĖS SĄLYGOS

1. Vartotojo kompiuteryje turi būti WWW naršyklė su instaliuotu JVM.
2. Kompiuteryje, kuriame veiks serveris, turi būti suinstaliuota JVM mažiausiai 1.2 versijos.
3. Inerneto „kelyje“, kuriuo perduodama informacija iš kliento kompiuterio į serverį ir atgal, neturi būti jokių „ugniasienių“, blokuojančių Java RMI mechanizmo veikimą.

3.1.3. SPECIFINIAI REIKALAVIMAI

3.1.3.1. FUNKCINIAI REIKALAVIMAI

Prisijungimo tvarka – prie sistemos prisijungia tik registruoti vartotojai. Kiekvienas registruotas vartotojas identifikuojamas unikaliu vartotojo vardu. Prisijungiančio vartotojo atpažinimas vyksta pagal vartotojo įvestą vardą ir slaptažodį.

Proceso stebėjimo galimybė – pateikiamoje energetinės sistemos schemoje nuolat atnaujinamos stebimų parametrų vertės.

Nustatymo parametrų reikšmių keitimo galimybė – RTS monitoriaus klientinė dalis turi leisti keisti energetinės sistemos reikšmes.

Prieduose pateikiama panaudojimo atvejų diagrama (*USE CASE*) (13 pav.).

3.1.3.2. NEFUNKCINIAI REIKALAVIMAI

Vėlinimas – kadangi programa veikia realaus laiko režimu, pageidautina, kad vėlinimas būtų kuo mažesnis. Specifikuojamas vėlinimas – ne didesnis kaip 4 sekundės tarp RTS monitoriaus serverinės ir klientinės dalies,

Web programa – kuriamas produktas turi būti prieinamas interneto vartotojui, naudojančiam standartinę WWW naršyklę.

RMI panaudojimas – RTS monitoriaus mechanizmas turi būti realizuotas panaudojant RMI. Vienas iš pagrindinių projekto tikslų ir yra išsiaiškinti Java RMI technologijos veikimą ir galimybę pritaikyti realaus laiko sistemų realizavimui.

UML panaudojimas – sistemos projektas ir architektūra turi būti specifikuojami panaudojant UML priemones.

3.1.3.3. SRITIES REIKALAVIMAI

Reikalavimų dokumentas sudarytas pagal IEEE/ANSI 830-1993 standartą.

3.2. DUOMENŲ STRUKTŪRA

RTS monitorius vartotojų identifikavimui naudojami vartotojų registravimo failu vartotojai.txt. Realiam laike besikeičiantys parametrai imituojami specialaus programos priedėlio – imitatoriaus, atstojančio duomenų šaltinį. Jokia duomenų bazė programoje nenaudojama. Imitatoriaus klasių diagrama pateikiama priede (14 pav.)

3.3. PROJETUOJAMOS SISTEMOS ARCHITEKTŪRA

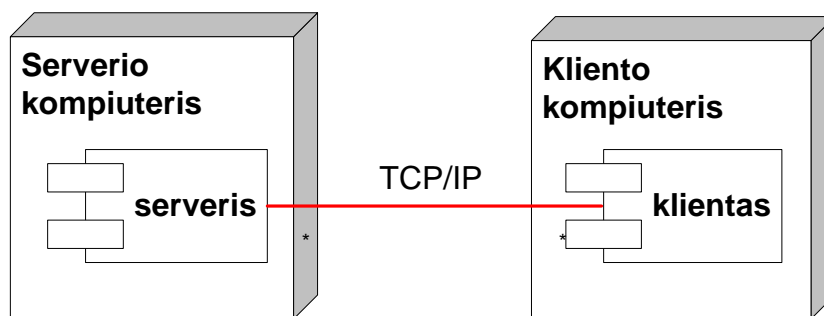
RTS monitoriaus veikimo principas pagrįstas kliento – serverio architektūra. Tokią sistemą sudaro serveris, atsakingas už informacijos nuolatinį perdavimą vartotojui ir priėmimą iš vartotojo bei klientas, priimančias serverio jiems persiunčiamą informaciją, pateikiantis ją vartotojui, taip pat persiunčiantis vartotojo pateikiamą informaciją serveriui.

Mūsų kuriamoje sistemoje šis kliento / serverio principas transformuojamas į paskirstytų objektų architektūros modelį. Šiuo atveju serveris sukuria ir eksportuoja tam tikrus objektus (eksportuojamus objektus), o klientai kviečia tam tikrus tų objektų servisus (eksportuojamus

metodus). Taip informacija perduodama iš serverio į klientą. Šias veiksmų sekas vaizdžiau iliustruoja paveikslai 15, 16 ir 17.

Iš principo, prisilaikant paskirstytų objektų architektūros kūrimo principų, informacijos perdavimui iš kliento į serverį, pastarasis turėtų kviesti kokius nors kliento eksportuojamus metodus. Tačiau tai nėra įmanoma realizuoti dėl reikalavimo, kad klientas galėtų veikti su bet kuria WWW naršykle, kurioje instaliuotas JVM. Norint, kad klientas galėtų eksportuoti savo objektus, reikia keisti sisteminės naršyklės nuostatas, kas gana ryškiai sumažintų galimybę prisijungti prie sistemos iš bet kurio kompiuterio.

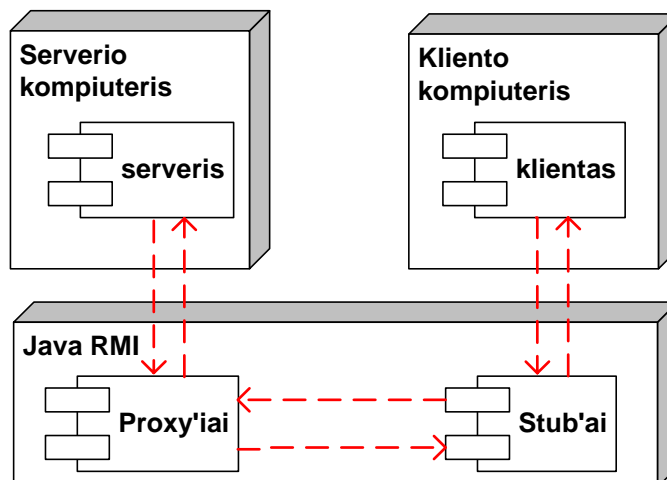
Todėl sistemos kūrimui buvo pasirinkta mišri architektūra, savyje apjungianti tiek paskirstytų objektų, tiek kliento – serverio architektūros principus.



8 pav. Bendra sistemos architektūra.

Programinės įrangos kūrimui buvo pasirinkta Java kalba. Paskirstytų objektų architektūros realizavimo priemonė buvo pasirinkta Java RMI.

Java RMI veikimo principas parodytas sekančioje diagramoje:



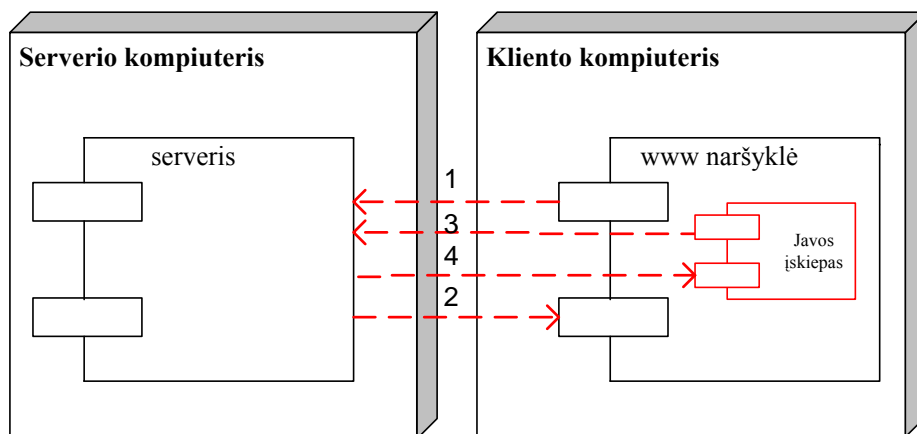
9 pav. Java RMI veikimo principo schema.

Kiekvienai serverio eksportuojamai klasei sugeneruojamos dvi papildomos klasės: *proxy*'is ir *stub*'as. Tuomet klientas bendrauja tik su *stub*'u, o serveris tik su *proxy*'iu. Šios dvi klasės pilnai paslepia duomenų perdavimą tarp atskirų kompiuterių – tik klientui, tiek serveriui atrodo, kad jie yra tame pačiame kompiuteryje.

Mano atveju serveris bus savarankiška programa, kuri nuolat veiks serverio kompiuteryje ir lauks prisijungimo iš kliento. Serverio kompiuteryje taip pat bus saugomas registruotų vartotojų sąrašas, už kurio papildymą ar pakeitimą bus atsakingas serveris.

Kliento programa – Java įskiepas. Įskiepas pasirinktas todėl, jog buvo iškeltas reikalavimas, kad programa būtų galima naudotis per WWW naršyklę. Taip pat įskiepą galima lengvai įterpti į bet kurį HTML puslapį.

RTS monitoriaus veikimo modelio diagrama:



10 pav. RTS monitoriaus veikimo modelio diagrama

Bendravimo tarp serverio ir kliento eiga:

1. WWW naršyklė kreipiasi į serverį.
2. Serveris nusiunčia WWW naršyklei HTML puslapį su kliento įskiepu.
- 3, 4. Kliento įskiepas ir serveris bendrauja tarpusavyje per RMI.

Kuriamoje programinėje įrangoje aiškiai išskiriamos dvi posistemos: serverio programinė įranga ir kliento programinė įranga.

Tačiau apibrėžiant programinės įrangos architektūrą, reikia numatyti šių dviejų posistemių komunikavimo metodą. Be to parametrus imituoti naudojamas imitatorius. Todėl apibrėžiant klasių paketų struktūrą išskirtini ne du, o keturi paketai:

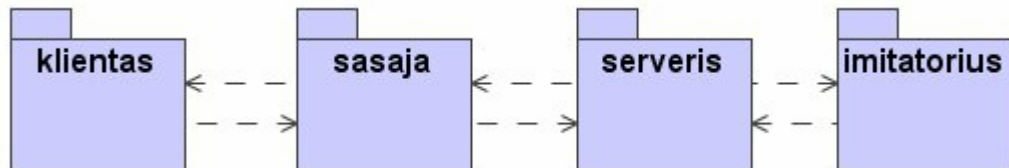
klientas – klientą realizuojančios klasės;

serveris – serverį realizuojančios klasės;

sasaja – klasės ir interfeisai, bendri tiek klientui, tiek serveriui;

imitatorius – klasių imituojančių parametrų kitimą realiame laike paketas.

Klasių paketų struktūrą atvaizduoja ši diagrama:



11 pav. Klasių paketų struktūros diagrama

Klasių diagrama pateikiama prieduose (pav. 14).

3.4. PROGRAMINIŲ MODULIŲ SPECIFIKACIJOS

RTS monitoriaus struktūra sudaryta iš kliento, sąsajos, serverio ir imitatoriaus programinių modulių.

3.4.1. KLIENTO MODULIŲ SPECIFIKACIJOS

Šį paketą sudaro klasės, reikalingos vartotojui pateikti ir persiųsti atgal serveriui visą informaciją.

4 lentelė. Kliento modulių specifikacija

Modulio vardas	Funkcinis aprašymas	Modulio sąsaja, dialogo struktūra
<i>index.htm</i>	Sukuria galimybę naudotis programa internetu, talpina savyje Javos įskiepą.	Suaktyvinamas interneto naršykle tiesiogiai vartotojo.
<i>PaleiskServeri.bat</i>	Paleidžia serverinę sistemos dalį suaktyvuodamas modulį <i>serveris.class</i> .	Suaktyvinamas tiesiogiai vartotojo.
<i>PaleiskVartotojuRegistracija.bat</i>	Paleidžia vartotojų registracijos	Suaktyvinamas tiesiogiai

	programą suaktyvuodamas modulį <i>VartotojuRegistracija.class</i> .	virtuotojo.
<i>Brez.jpg</i>	Šiluminė schema patalpinama Javos išskiepe.	Modulį pasiekama <i>KlientoApletas.class</i> modulis.
<i>virtotojai.txt</i>	Failas, kuriame saugoma informacija apie registruotus virtotojus – Prisijungimo vardas ir slaptažodis.	Modulis redaguojamas programa <i>VartotojuRegistracija.class</i> Modulyje esama informacija naudojama vykdant failą <i>serveris.class</i> .
<i>KlientoApletas.java</i>	Tai Javos išskiepas su integruotu šiluminės schemos paveikslėliu vizualiai atvaizduojantis besikeičiančius parametrus ir jų vietą.	Modulis sukuriamas <i>index.htm</i> puslapio, suaktyvinant metoda „init“.
<i>PrisijungimoPanele.java</i>	Tai virtotojų prisijungimo panelė į kurią įvedamas virtotojo vardas ir slaptažodis. Jei virtotojas registruotas – panele sunaikinama, virtotojui perleidžiamas valdymas, jei neregistruotas – pasiūloma jungtis iš naujo.	Modulį sukuria <i>KlientoApletas.class</i> modulis.
<i>ValdymoPanele.java</i>	Tai virtualios energetinės sistemos valdymo panelė. Šios panelės mygtukų pagalba kviečiamos kitos panelės, susijusios su virtualiai dirbančiais įrenginiais jų būsenoms modifikuoti.	Modulį sukuria <i>PrisijungimoPanele.class</i> modulis.
<i>Nuskaitymas.java</i>	Šio modulio paskirtis – tam tikru intervalu nuskaityti iš serverio sistemos parametrų reikšmes. Parinktas intervalas – 2.75 sekundės.	Modulį sukuria <i>KlientoApletas.class</i> modulis.
<i>Papildoma.java</i>	Šioje klasėje realizuoti statiniai metodai panelių centravimui ekrano atžvilgiu.	Į modulio statinius metodus kreipiamasi iš kitų modulių.
<i>Siurblys.java</i>	Panelės paskirtis – nustatyti siurblio darbo greitį.	Modulį sukuria <i>ValdymoPanele.class</i> modulis.
<i>Slėgis.java</i>	Panelės paskirtis – išjungti arba įjungti šildymo sistemos papildymą. Jei papildymas įjungtas – nustatyti šildymo sistemos slėgį.	Modulį sukuria <i>ValdymoPanele.class</i> modulis.
<i>ŠildymoSistema.java</i>	Panelės paskirtis – išjungti arba įjungti šildymo sistemą. Šildymo sistemą	Modulį sukuria <i>ValdymoPanele.class</i> modulis.

	išjungus – išjungiamas ir siurblys, įjungus – įjungiamas ir siurblys dirbti minimaliu greičiu.	
<i>Tustinti.java</i>	Panelės paskirtis – šildymo sistemos tuštinimas. Leidžiama tuštinti tik uždarius šildymo sistemą ir išjungus jos papildymą.	Modulį <i>ValdymoPanele.class</i> sukuria
<i>Saltinis.java</i>	Panelės paskirtis – šilumos šaltinio įjungimas / išjungimas.	Modulį <i>ValdymoPanele.class</i> sukuria
<i>Kreive.java</i>	Panelės paskirtis – keisti šildymo sistemos veikimo intensyvumą.	Modulį <i>ValdymoPanele.class</i> sukuria

3.4.2. SAŠAJOS MODULIŲ SPECIFIKACIJOS

Šį paketą sudaro klasės – sąsajos, reikalingos ryšiui tarp kliento ir serverio užtikrinti. Taip pat šiame pakete yra galimos klaidos klasė – *BlogoPrisijungimoKlaida.java*.

5 lentelė. Sąsajos modulių specifikacija

Modulio vardas	Funkcinis aprašymas	Modulio sąsaja, dialogo struktūra
<i>KintamujuSasaja.java</i>	Tai modulis – sąsaja. Joje išvardinti visi metodai, naudojami parametrus iš serverio klientui ir iš kliento serveriui perduoti.	Jos metodus realizuoja <i>Serveris.class</i> modulis.
<i>VatotojuSasaja.java</i>	Tai modulis – sąsaja. Joje suprogramuotas metodas naudojamas patikrinti ar vartotojas registruotas.	Šio modulio tipo objektas eksportuojamas per <i>KintamujuSasaja.class</i> modulį.
<i>BlogoPrisijungimoKlaida.java</i>	Tai modulis – klaida. Šis modulis perduodamas klientui įvedus blogą vartotojo vardo ir slaptažodžio kombinaciją.	Modulį <i>Vartotojai.class</i> sukuria

3.4.3. SERVERIO MODULIŲ SPECIFIKACIJOS

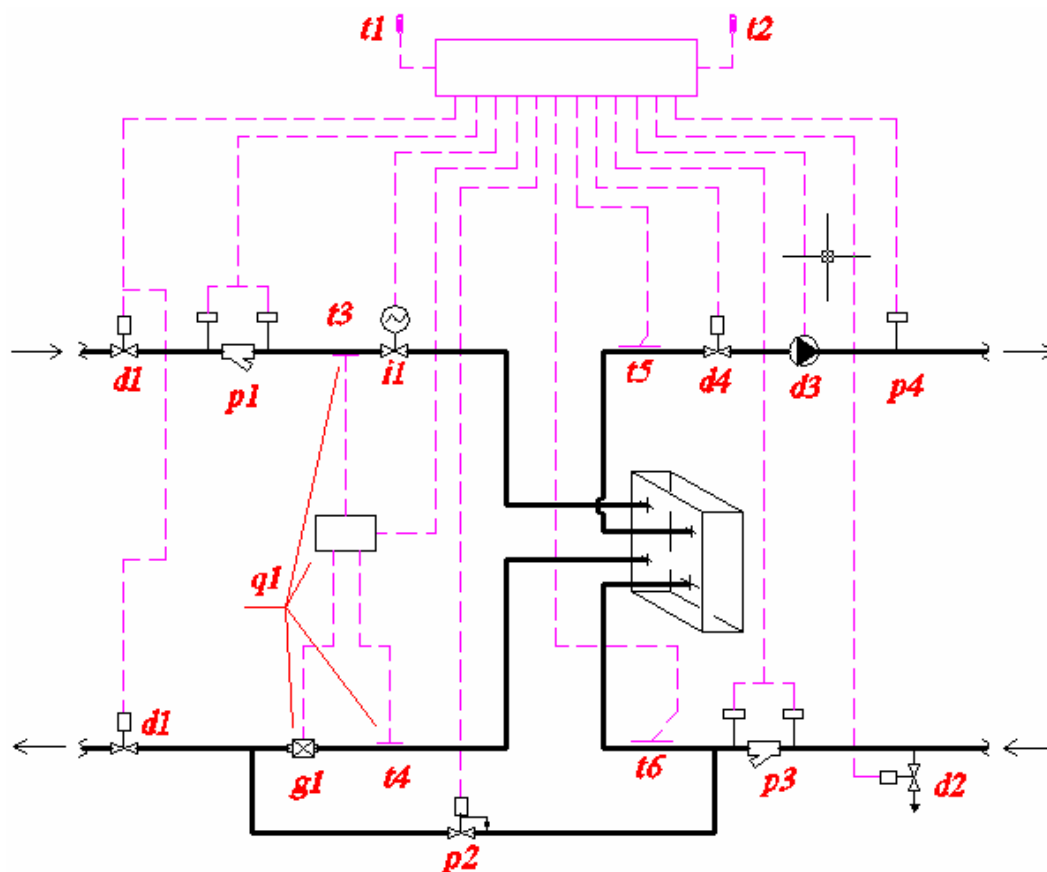
Šį paketą sudaro klasės, skirtos nuolatiniam procesui serverio kompiuteryje užtikrinti ir programinės įrangos klientinei daliai aptarnauti.

6 lentelė. Serverio modulių specifikacija

Modulio vardas	Funkcinis aprašymas	Modulio sąsaja, dialogo struktūra
<i>Serveris.java</i>	Tai pagrindinis programinės įrangos serverinės dalies modulis. Jis sukuria serveriniame kompiuteryje portą, per kurį komunikuoja serveris ir klientas. Taip pat šis modulis realizuoja visus KintamųjųSąsaja.java metodus.	Modulis sukuriama paleidžiant failą <i>PaleiskServeri.bat</i> .
<i>Vartotojai.java</i>	Šio modulio tikslas – apdoroti vartotojus. Šis modulis nuskaityti vartotojų failą, tikrina ar vartotojas priregistruotas.	Modulis sukuriama <i>Serveris.class</i> modulio.
<i>PrisijungimoTikrintojas.java</i>	Šio modulio paskirtis – neleisti prisijungti vartotojui prie sistemos, jei jau vienas vartotojas prisijungęs.	Modulis sukuriama <i>Serveris.class</i> modulio.
<i>Serveris_Stub.java</i>	Šis modulis generuojamas automatiškai Javos. Tai tinklinio komunikavimo priemonė	
<i>Vartotojai_Stub.java</i>	Šis modulis generuojamas automatiškai Javos. Tai tinklinio komunikavimo priemonė	
<i>VartotojuRegistracija.java</i>	Tai atskiras modulis, nesusijęs programiniais ryšiais su kitais moduliais. Šio modulio tikslas – papildyti, redaguoti ar įterpti naują vartotoją į registruotų vartotojų sąrašą	Modulis sukuriama paleidžiant failą <i>PaleiskVartotojuRegistracija.bat</i> .

3.4.4. IMITATORIAUS MODULIŲ SPECIFIKACIJOS

Šį paketą sudaro klasės, skirtos energetinės sistemos procesų imitavimui. Klasių aprašomų parametrų vietos energetinėje sistemoje pažymėtos 12 paveiksle.



12 pav. Parametrų vietų energetinėje sistemoje schema

7 lentelė. Imitatoriaus modulių specifikacija

Modulio vardas	Funkcinis aprašymas	Modulio sąsaja, dialogo struktūra
<i>Imitatorius.java</i>	Pagrindinis imitatoriaus paketo modulis sukuriantis kitus imitatoriaus modulius ir koordinuojantis jų veiklą	Modulis sukuriamas <i>Serveris.class</i> modulio
<i>D1.java</i>	Modulis – gija imituojanti parametro d1 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>D2.java</i>	Modulis – gija imituojanti parametro d2 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>D3.java</i>	Modulis – gija imituojanti parametro d3 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>D4.java</i>	Modulis – gija imituojanti parametro d4 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>G1.java</i>	Modulis – gija imituojanti parametro g1 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>I1.java</i>	Modulis – gija imituojanti parametro i1 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.

<i>P1.java</i>	Modulis – gija imituojanti parametro p1 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>P2.java</i>	Modulis – gija imituojanti parametro p2 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>P3.java</i>	Modulis – gija imituojanti parametro p3 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>P4.java</i>	Modulis – gija imituojanti parametro p4 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>Q1.java</i>	Modulis – gija imituojanti parametro q1 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>T1.java</i>	Modulis – gija imituojanti parametro t1 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>T2.java</i>	Modulis – gija imituojanti parametro t2 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>T3.java</i>	Modulis – gija imituojanti parametro t3 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>T4.java</i>	Modulis – gija imituojanti parametro t4 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>T5.java</i>	Modulis – gija imituojanti parametro t5 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.
<i>T6.java</i>	Modulis – gija imituojanti parametro t6 procesą	Modulis sukuriamas <i>Imitatorius.class</i> modulio.

3.5. TESTAVIMO MEDŽIAGA

Testavimas pradedamas redaguojant sąrašą vartotojų, kuriems leidžiama dirbti su sistema. *rts* kataloge paleidžiamas failas *PaleiskVartotojuRegistracija.bat*. Parodomas vartotojų registracijos langas (18 pav). Vadovaujantis taisyklėmis pateiktomis sistemos administratoriaus vadove sukuriamas sąrašas vartotojų galinčių dirbti su sistema. Sąrašas fiksuojamas faile „vartotojai.txt“ ir po redagavimo atrodo kaip parodyta paveiksle 19.

Paleidžiamas serveris. Tai atliekama suaktyvinant faila *PaleiskServeri.bat*. Parodomas pranešimas (20 pav).

Iš interneto naršyklės suaktyvinamas failas „index.htm“. Parodomas prisijungimo langas (22 pav). Įvedame neteisingą kombinaciją vardas – „vienas“, slaptažodis – „du“. Parodomas pranešimas (23 pav). Įvedame vieną iš priregistruotų kombinacijų – Lukas/pass. Parodomas pagrindinis darbo langas (25 pav).

Toliau programa testuojama spaudžiant įvairius valdymo skydo mygtukus ir stebint pokyčius ekrane. Programa buvo testuota atliekant visus 4.2 punkte aprašytus veiksmus, pastebėti trūkumai ištaisyti.

4. VARTOTOJO DOKUMENTACIJA

4.1. SISTEMOS FUNKCINIS APRAŠYMAS

RTS – programa leidžianti stebėti ir valdyti energetinės sistemos – katilinės – būklę realiaame laike iš bet kurio kompiuterio, turinčio interneto ryšį. Kadangi vietoje katilinės naudojama programa imituojanti realius procesus, norint RTS programą taikyti realiai, ją reikia šiek tiek modifikuoti vietoje imitatoriaus naudojant energetinės sistemos valdiklį.

Su programa turintys dirbti leidimą vartotojai registruojami sistemos administratoriaus. Prisijungiant reikalaujama įvesti vardą ir slaptažodį.

RTS programos procesų stebėjimas – tai energetinės sistemos parametrų, tokių kaip slėgio, temperatūros, įjungta/išjungta reikšmių kitimo stebėjimas. Reikšmių užklausimo dažnis gali būti parenkamas pagal poreikį programuojant.

Proceso valdymas atliekamas keičiant proceso nustatymus. Nustatymų keitimo įtaka kitiems proceso parametrams taip pat pavaizduojama ekrane. Proceso valdymo pavyzdžiu galėtų būti sistemos sustabdymas.

4.2. SISTEMOS VADOVAS

Programa pradedama interneto naršykle suaktyvinus failą index.htm. Jei gaunamas pranešimas (21 pav.) tai reiškia, kad serveris yra nepaleistas, arba netvarkinga serverio programinė įranga. Šiuo atveju reikia kreiptis į sistemos administratorių. Jei prisijungiama sėkmingai, parodomas prisijungimo langas.

Įvedamas vartotojo vardas ir slaptažodis. Galimi trys sistemos tolesnio darbo atvejai:

1. Klientinė sistemos dalis prisijungs prie serverio. Šiuo atveju parodomas langas (25 pav.).
2. Sistema praneš, kad blogas prisijungimo vardas ar slaptažodis ir jungimasis bus nutrauktas (23 pav.).
3. Sistema praneš, kad šiuo metu prie sistemos jau yra prisijungta (24 pav.). Šis pranešimas reiškia, kad kitas vartotojas dirba su sistema ir ja naudotis laikinai negalima.

Prisijungus prie serverio matyti energetinės sistemos šiluminė schema ir valdymo pultas. Schemoje langeliuose prie tam tikrų elementų nuolat keičiasi reikšmės. Šių reikšmių fizikinė reikšmė paaiškinta žemiau (parametrų pavadinimai parinkti pagal 14 pav.):

8 lentelė. Parametrų paaiškinimas

Parametro pavadinimas	Parametro paaiškinimas
d1	Parametras indikuoja šilumos šaltinio ventilio atvirą / uždara būseną (<i>ON / OFF</i>).
d2	Parametras indikuoja šildymo sistemos išleidiklio atvirą / uždara būseną (<i>ON / OFF</i>).
d3	Parametras indikuoja siurblio sukimosi greitį (<i>v-1 / v-2 / v-3 / OFF</i>).
d4	Parametras indikuoja šildymo sistemos ventilio atvirą / uždara būseną (<i>ON / OFF</i>).
g1	Parametras indikuoja šilumos šaltinio kontūru pratekantį debitą, m ³ /h.
i1	Parametras indikuoja šildymo intensyvumą, ($0 \leq x \leq 3.2$).
p1	Parametras indikuoja šilumos šaltinio kontūro filtro užterštumą, bar.
p2	Parametras indikuoja šildymo sistemos papildymo slėgį, bar.
p3	Parametras indikuoja šildymo sistemos kontūro filtro užterštumą, bar.
p4	Parametras indikuoja šildymo sistemos slėgį, bar.
q1	Parametras indikuoja suvartojamą energijos kiekį, kW.
t1	Parametras indikuoja vidaus temperatūrą, °C.
t2	Parametras indikuoja lauko temperatūrą, °C.
t3	Parametras indikuoja tiekiamo vandens iš šilumos šaltinio temperatūrą, °C.
t4	Parametras indikuoja grįžtamo vandens iš šilumos šaltinio temperatūrą, °C.
t5	Parametras indikuoja tiekiamo vandens į šildymo sistemą temperatūrą, °C.
t6	Parametras indikuoja grįžtamo vandens iš šildymo sistemos temperatūrą, °C.

Procesai valdomi valdymo pultu (26 pav). Valdymo pultą sudaro šie mygtukai:

9 lentelė. Valdymo pulto mygtukų paskirtis

Mygtuko pavadinimas	Mygtuko funkcija
Siurblio greitis	Leidžia pakeisti siurblio greitį. Siurblys gali dirbti trimis režimais – taigi leidžia parinkti vieną iš jų; Išjungus šildymo sistemą ir bandant įjungti siurblių, gaunamas pranešimas (27 pav) ir siurblio įjungti neleidžiama.
Šildymo sistemos slėgis	Leidžia įjungti / išjungti šildymo sistemos papildymą, taip pat leidžia keisti šildymo sistemos slėgį (29 pav).
Šildymo sistemos uždarymas / atidarymas	Leidžia įjungti / išjungti šildymo sistemą (30 pav). Išjungus šildymo sistemą automatiškai išjungiamas ir siurblys. Vėl įjungus šildymo sistemą siurblys įjungiamas dirbti mažiausiu greičiu.

Šildymo sistemos tuštinimas	Leidžia atidaryti / uždaryti šildymo sistemos išleidiklį (31 pav). Jei šildymo sistemos papildymas atidarytas, išleidiklis negali būti atidaromas, todėl gaunamas pranešimas (32 pav).
Šilumos šaltinio įjungimas / išjungimas	Leidžia įjungti / išjungti šilumos šaltinį (33 pav).
Šildymo intensyvumas	Leidžia nustatyti šildymo intensyvumą (34 pav). Intensyvumas nustatomas pagal koeficientą, kuris gali būti nuo 0.1 (minimalus šildymas) iki 3.4 (maksimalus šildymas)
Baigti darbą	Darbo su sistema pabaiga

Darbas su programa baigiamas paspaudus mygtuką „Baigti darbą“.

4.3. SISTEMOS INSTALIAVIMO DOKUMENTAS

Programa dirbs, jei ji bus perkelta į kompiuterį, atliekantį WEB serverio funkcijas. Prieš naudojantis programa reikia atlikti šiuos veiksmus:

1. Serveryje reikia suinstaliuoti JVM, jei tai nebuvo atlikta anksčiau;
2. Sistemoje privalo būti nurodytas kelias iki katalogo, kuriame yra java.exe, jei tai nebuvo atlikta anksčiau;
3. Katalogas RTS perkeliamas į WEB serverį;

Tam, kad klientinė programos dalis galėtų bendrauti su serverine programos dalimi reikia serverio kompiuteryje suaktyvinti failą *PaleiskServeri.bat*.

Norint paleisti klientinę programos dalį reikia WEB naršyklėje surinkti WEB serverio adresą, gale užrašyti „/rts“.

4.4. SISTEMOS ADMINISTRATORIAUS VADOVAS

Kadangi sistema suprogramuota Java, ji gali dirbti bet kokios platformos pagrindu. Serverio paleidimo failas *PaleiskServeri.bat* turėtų būti pakoreguotas atsižvelgiant į platformos .bat failų rašymo sintaksę.

Vartotojų prisijungimo duomenų keitimas. Vartotojų prisijungimo duomenų keitimas atliekamas suaktyvinant failą *PaleiskVartotojuRegistracija.bat*. Ekrane parodomas langas (18 pav.). Slankiojamos juostos pagalbą peržiūrimas registruotų vartotojų sąrašas. Mygtukų „Naujas vartotojas“, „Įvesti vartotoją“, „Trinti vartotoją“ pagalba redaguojamas vartotojų sąrašas. Darbas su programa baigiamas paspaudus mygtuką „Baigti darbą“.

5. IŠVADOS

1. Šiame darbe apžvelgti plačiausiai naudojami stebėjimo ir valdymo informacijos perdavimo protokoliai MODBUS, EIB ir LONWORKS. Pradiniame magistro tiriamojo darbo etape informacijai perduoti tarp įrenginio ir kompiuterio buvo pasirinktas LONWORKS protokolas. Neradus rėmėjų, galinčių paskolinti techninę įrangą, nutarta vietoj jos sukurti ir naudoti programinės įrangos paketą – *imitatorių* – imituojantį realius procesus.
2. Sukurta programinė įranga RTS monitorius leidžia stebėti energetinėje sistemoje vykstančius procesus per nuotolį, šiuo atveju – internetą. RTS monitorių lengva pritaikyti realiam darbui. Pakanka modulį *imitatorius* pakeisti valdikliu ir sukurti naują sąsają tarp valdiklio ir modulio *serveris*. Taip pat reikia pritaikyti vartotojo sąsają naudojamam energetinės sistemos tipui.
3. Visa programinė įranga parašyta Java kalba – tai leidžia programą RTS monitorius panaudoti bet kurioje operacinėje sistemoje.
4. RTS monitoriuje įdiegta apsaugos nuo neautorizuoto programos naudojimo priemonė – vartotojo identifikavimas – saugo nuo nepageidautinų asmenų prisijungimo. Tačiau atsižvelgiant į RTS monitoriaus taikymo sritį, būtina skirti daugiau dėmesio saugumo priemonių plėtotei. Kadangi saugumo problematika – labai plati tema, todėl šiame darbe plačiau nenagrinėta – tai dirva ateities tyrinėjimams.
5. Projektuojant sistemą nemažai sužinota apie naujas programavimo technologijas, priemones, palengvinančias projektavimo ir programavimo procesus bei įvairias su tuo susijusias problemas (kliento – serverio architektūra, Java RMI technologija ir kt.). Įgyta gera projektavimo, programavimo, dokumentavimo patirtis.

6. LITERATŪRA

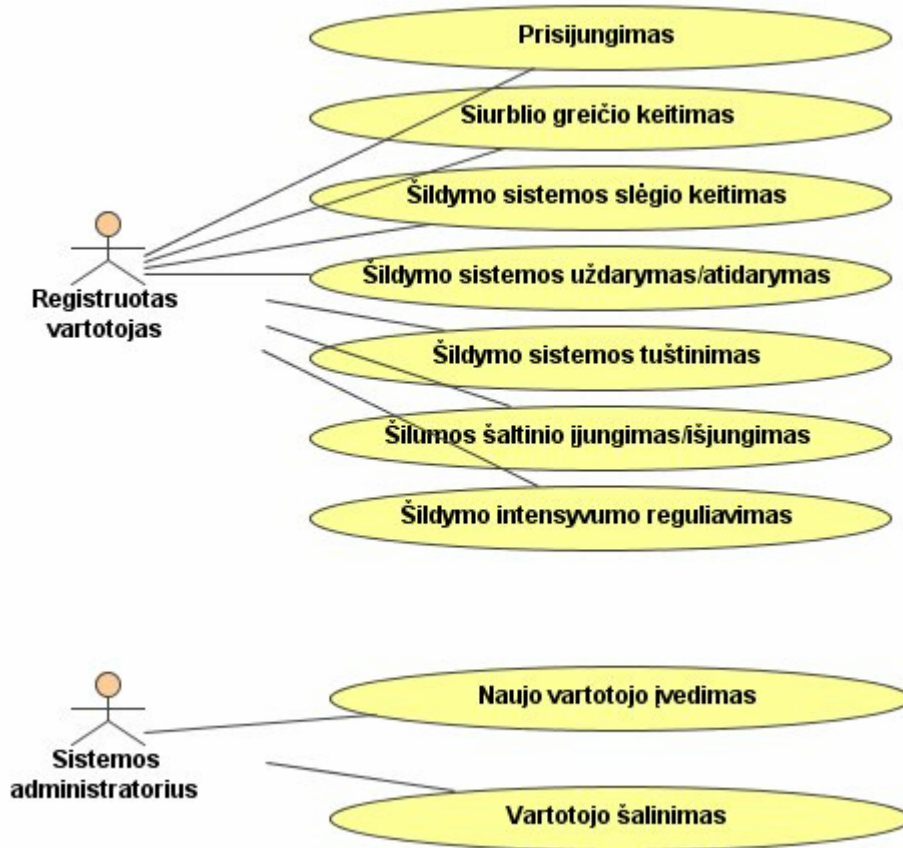
1. Introduction to the LonWorks® Platform [interaktyvus]. Echelon kompanija. – [žiūrėta 2003-03-27]. Prieiga per internetą: <<http://www.echelon.com/productdb/search.asp?systemID=35&sysLabel=LonWorks+Training>>.
2. Edvardas Karpovičius. Protingo namo projektavimas – intelektualios EIB instaliacijos sistemos//Elektros erdvės – 2003, Nr. 1, p. 19.
3. Kompiuterių komunikacijos paskaitos [interaktyvus]. KTU Elektroninių ir matavimo sistemų katedra [žiūrėta 2003-05-25]. Prieiga per internetą: <<http://www.rpk.ktu.lt/444/courses/KompKomunikacijos/KompKomunikacijuPaskaitos.pdf>>.
4. IEEE/ANSI 830-1993 reikalavimų dokumento standartas [žiūrėta 2003-11-27]. Prieiga per internetą: <http://standards.ieee.org/reading/ieee/std_public/description/se/830-1993_desc.html>.
5. Informacija apie Java RMI [žiūrėta 2004-01-10]. Prieiga per internetą: <<http://java.sun.com/products/jdk/rmi/index.html>>.
6. Informacija apie UML [žiūrėta 2003-06-10]. Prieiga per internetą: <http://www.uml.org>

7. TERMINŲ IR SANTRUPMŲ ŽODYNAS

- *JVM* – *Java Virtual Machine santrumpa*. Java virtuali mašina.
- *RMI(Remote Method Invocation)* – "Nutolusio metodo iškvietimas". Java programavimo kalbos priemonės paskirstytų objektų architektūrai realizuoti.
- *Serveris* – sistemos serverinė dalis, aptarnaujanti prisijungiančius klientus – atliekanti registruotų vartotojų atpažinimą, informacijos su klientu keitimąsi – ir užtikrinanti nepriklausomų procesų vykdymą realiame laike.
- *Klientas* – sistemos dalis, atsakinga už bendravimą su vartotoju: informacijos įvedimą ir perdavimą į serverį, taip pat už gautos iš serverio informacijos atvaizdavimą ekrane.
- *Vartotojas* – registruotas sistemos vartotojas, galintis naudotis jos teikiamomis paslaugomis.
- *Serverio objektas* – toks serverio objektas, kurio eksportuojamais metodais gali naudotis klientai.
- *Eksportuojamas metodas* – serverio objekto metodas, kurį gali iškviešti išorinės programos.

8. PRIEDAI

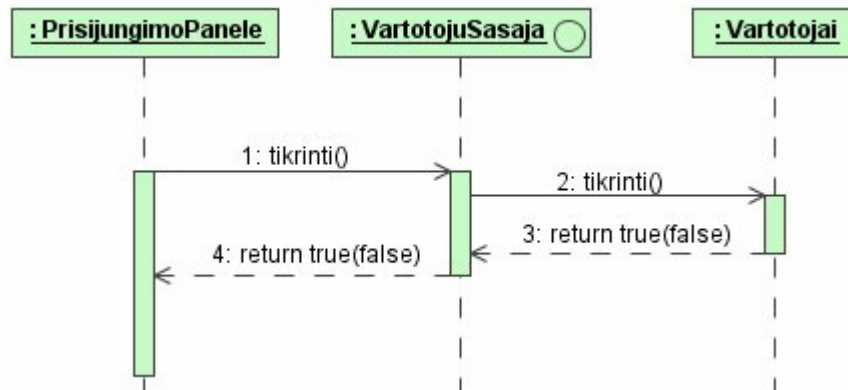
8.1. PANAUDOJIMO ATVEJŲ UML DIAGRAMA



13 pav. Panaudojimo atvejų diagrama.

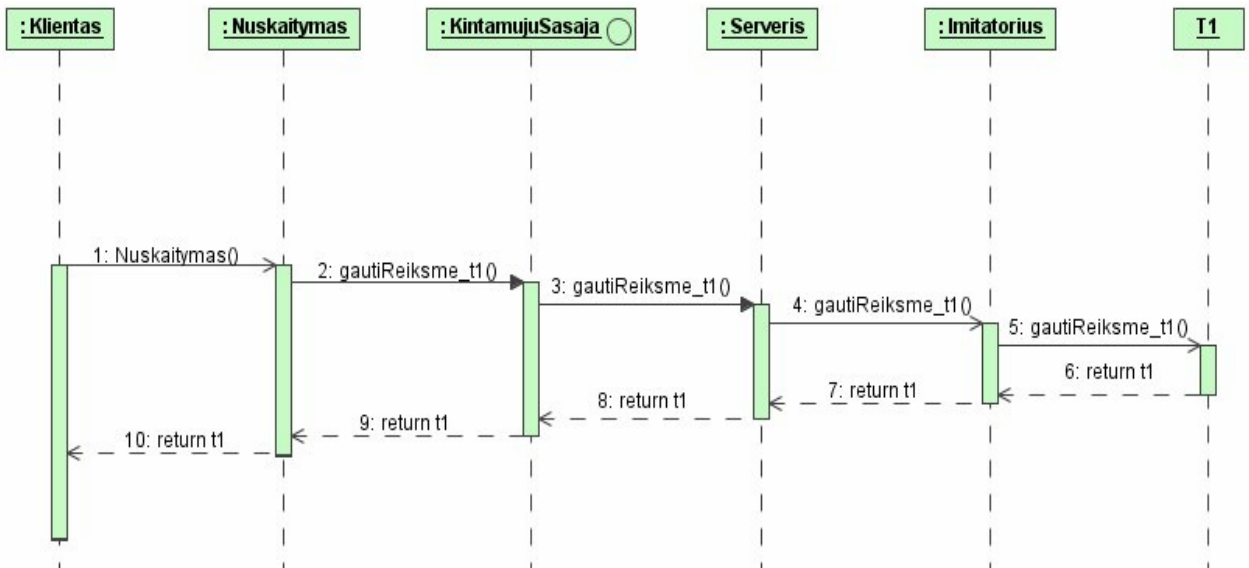
8.2. RTS MONITORIAUS KLASIŲ DIAGRAMA

8.3. VARTOTOJO PRISIJUNGIMO SEKŲ DIAGRAMA



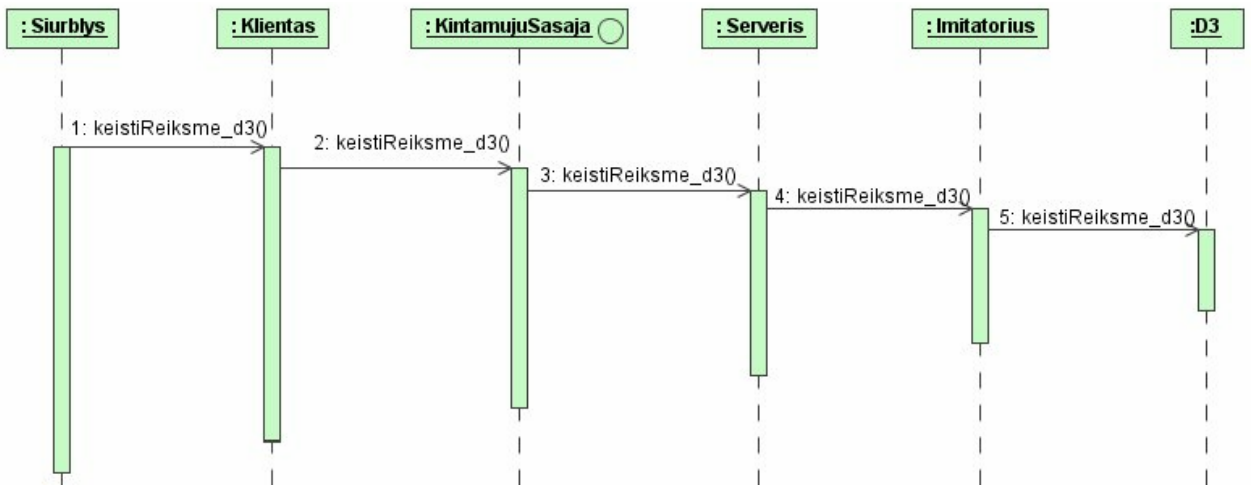
15 pav. Parametro reikšmės kelio sekų diagrama.

8.4. PARAMETRO REIŠMĖS NUSKAITYMO SEKŲ DIAGRAMA



16 pav. Parametro reikšmės nuskaitymo sekų diagrama.

8.5. PARAMETRO REIŠMĖS KEITIMO SEKŲ DIAGRAMA

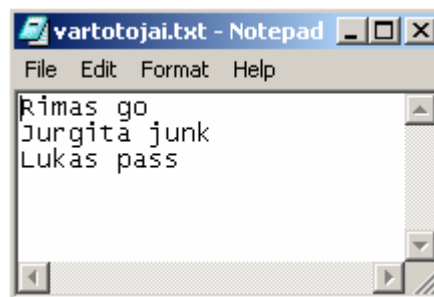


17 pav. Parametro reikšmės keitimo sekų diagrama.

8.6. RTS MONITORIAUS EKRAŪ KOPIJOS



18 pav. Vartotojų registravimo langas.



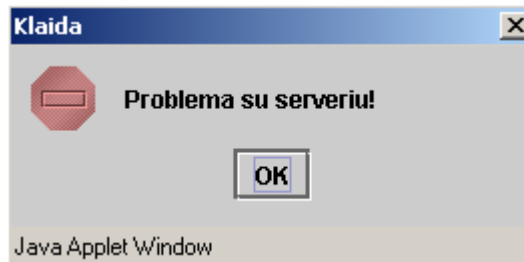
19 pav. Vartotojų registravimo failas.


```
Command Prompt
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

E:\>E:\Documents and Settings\Lukas\Desktop\MAGISTRINIS\rts PaleiskServeri.bat
E:\Documents and Settings\Lukas\Desktop\MAGISTRINIS\rts>java serveris/Serveris v
artotojai.txt

Serveris paleistas
```

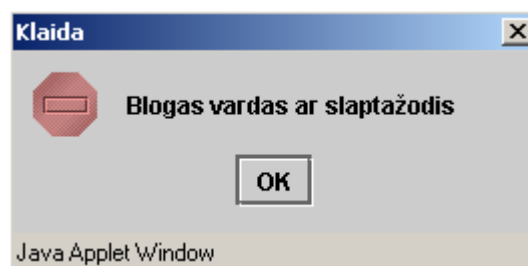
20 pav. Serverio paleidimo patvirtinimo langas.



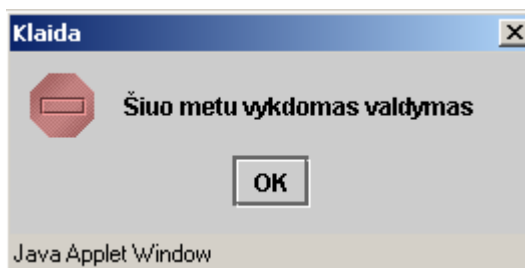
21 pav. Klaida „Problema su serveriu“.

Vardas
Slaptažodis

22 pav. Prisijungimo panelė.



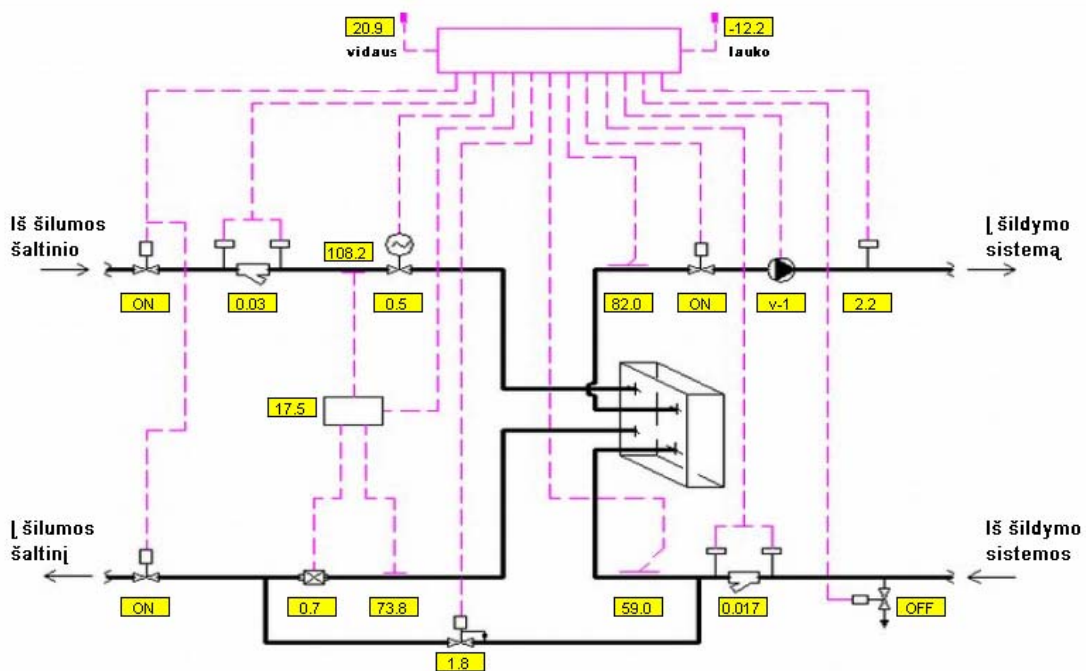
23 pav. Neteisingo prisijungimo pranešimas.



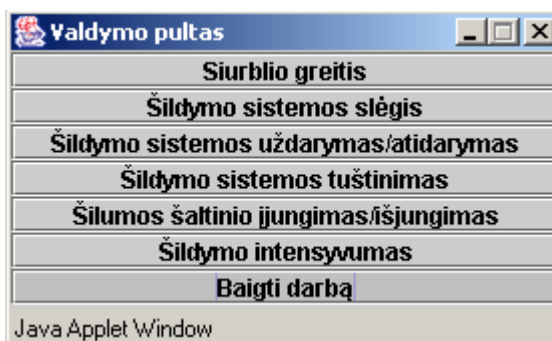
24 pav. Pakartotinio jungimosi pranešimas.

REAL TIME SYSTEM

Energetiniu sistemų nuotolinis valdymas



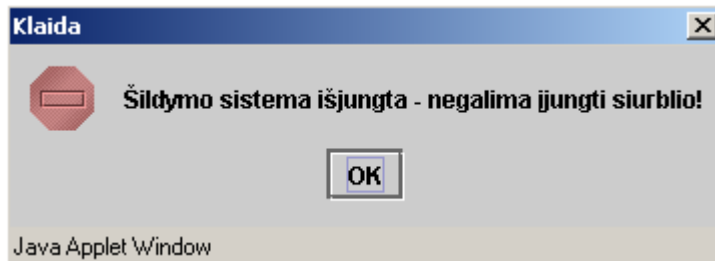
25 pav. Procesų stebėjimo langas.



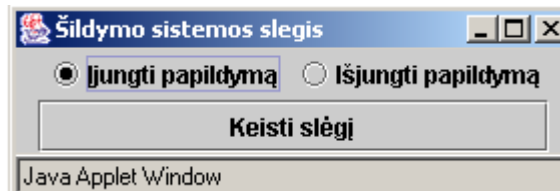
26 pav. Valdymo pulto langas.



27 pav. Siurblio greičio nustatymo langas.



28 pav. Klaidos „Šildymo sistema išjungta – negalima įjungti siurblio!“ langas.



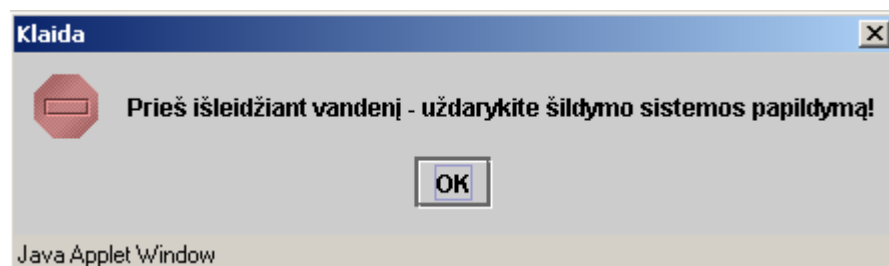
29 pav. Šildymo sistemos slėgio keitimo langas.



30 pav. Šildymo sistemos įjungimo / išjungimo langas.



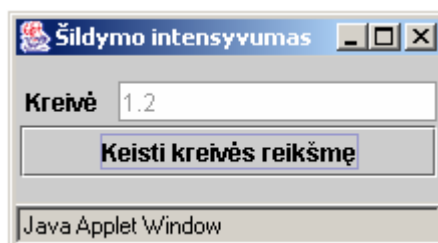
31 pav. Šildymo sistemos tuštinimo langas.



32 pav. Klaidos „Prieš išleidžiant vandenį – uždarykite šildymo sistemos papildymą!“ langas.



33 pav. Šilumos šaltinio įjungimo / išjungimo langas.



Pav 34. Šildymo intensyvumo keitimo langas.

8.7. PROGRAMINĖS ĮRANGOS DISKELIS