

VILNIAUS UNIVERSITETAS

VLADIMIRAS DOLGOPOLOVAS

PROGRAMAVIMO MOKYMO SI OBJEKTAI MOKSLINĖS  
KOMPIUTERIJOS MOKYMU I: MOKSLINIO TYRIMO STUDIJS  
NAUDOJANT STOCHASTINIUS REKURENTINIUS MODELIUS

Daktaro disertacijos santrauka  
Technologijos mokslai, informatikos inžinerija (07 T)

Vilnius, 2018

Disertacija rengta 2013–2017 metais Vilniaus universitete.

**Mokslinė vadovė**

prof. dr. Valentina Dagienė (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T).

**Disertacija ginama viešame disertacijos Gynimo tarybos posėdyje:**

**Pirmininkas**

prof. dr. Julius Žilinskas (Vilniaus universitetas, technologiniai mokslai, informatikos inžinerija – 07 T).

**Nariai:**

prof. dr. Eduardas Bareiša (Kauno Technologijos Universitetas, technologijos mokslai, informatikos inžinerija – 07 T),

prof. dr. Vitalij Denisov (Klaipėdos universitetas, technologijos mokslai, informatikos inžinerija – 07 T),

prof. dr. Saulius Gudas (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T),

prof. dr. Erkki Sutinen (Turku universitetas, Suomija, technologijos mokslai, informatikos inžinerija – 07 T).

Disertacija bus ginama viešame disertacijos Gynimo tarybos posėdyje 2018 m. liepos 2 d. 13:00 val. Vilniaus universiteto Duomenų mokslo ir skaitmeninių technologijų instituto 203 auditorijoje.

Adresas: Akademijos g. 4, LT-04812, Vilnius, Lietuva.

Disertacijos santrauka išsiuntinėta 2018 m. birželio 1 d.

Su disertaciją galima susipažinti Vilniaus universiteto bibliotekoje ir VU interneto svetainėje adresu: <https://www.vu.lt/lt/naujienos/ivykiu-kalendorius>

VILNIUS UNIVERSITY

VLADIMIRAS DOLGOPOLOVAS

SOFTWARE LEARNING OBJECTS FOR SCIENTIFIC COMPUTING  
EDUCATION: TEACHING SCIENTIFIC INQUIRY WITH  
RECURRENCE BASED STOCHASTIC MODELS

Summary of Doctoral Dissertation  
Technological Sciences, Informatics Engineering (07 T)

Vilnius, 2018

The doctoral dissertation was written in 2013–2017 at Vilnius University.

**Scientific Supervisor**

Prof. Dr. Valentina Dagienė (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

**The dissertation is defended at the Dissertation Defence Council:**

**Chairman**

Prof. Dr. Julius Žilinskas (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

**Members:**

Prof. Dr. Eduardas Bareiša (Kaunas University of Technology, Technological Sciences, Informatics Engineering – 07 T),

Prof. Dr. Vitalj Denisov (Klaipėda University, Technological Sciences, Informatics Engineering – 07 T),

Prof. Dr. Saulius Gudas (Vilnius University, Technological Sciences, Informatics Engineering – 07 T),

Prof. Dr. Erkki Sutinen (University of Turku, Finland, Technological Sciences, Informatics Engineering – 07 T).

The dissertation will be defended at the public meeting of the defence Council in the auditorium number 203 at the Institute of Data Science and Digital Technologies of Vilnius University, at 1 p.m. on 2nd of July 2018. Address: Akademijos st. 4, LT-04812 Vilnius, Lithuania.

The summary of the doctoral dissertation was distributed on the 1th of June 2018.

A copy of the doctoral dissertation is available for review at the Library of Vilnius University.

# **Turinys**

<b>Turinys</b>	<b>v</b>
<b>1 Įvadas: sąvokos ir apibrėžimai</b>	<b>1</b>
<b>2 Tyrimo kontekstas</b>	<b>2</b>
<b>3 TPACK modelio įgyvendinimo problema mokslinės kompiuterijos mokymo srityje</b>	<b>12</b>
<b>4 Mokslinės kompiuterijos TPACK modelio konteksto specifikacija</b>	<b>14</b>
<b>5 Mokslinės kompiuterijos ugdymo srities savybių metaanalizė</b>	<b>14</b>
<b>6 Lygiagrečio mokymo srities analizė</b>	<b>15</b>
<b>7 Eksperimentinis tyrimas: edukaciniai sprendimai</b>	<b>21</b>
<b>8 Sukurtų edukacinių sprendimų vertinimas</b>	<b>26</b>
<b>9 Išvados</b>	<b>28</b>
<b>Literatūra</b>	<b>29</b>
<b>Summary of Doctoral dissertation</b>	<b>31</b>

# 1 Įvadas: sąvokos ir apibrėžimai

## Projektavimo moksliniai tyrimai

- Projektavimo moksliniai tyrimai (angl. *design science research*) yra mokslinių tyrimų metodologija, orientuota į projektuojamų artefaktų kūrimą ir įgyvendinimą, siekiant sistemingai išryškinti artefakto funkcines savybes.
- Projektavimo moksliniai tyrimai paprastai taikomi kognityvinių artefaktų, pavyzdžiui, algoritmų, žmogaus ir kompiuterio sąsajos, projektavimo metodikų (įskaitant proceso modelius) ir kalbų, kategorijoms.

## Edukacinės technologijos, studijų proceso organizavimas, didaktinis metodas

- Edukacinės technologijos suprantamos kaip mokymosi metodų teorija ir praktika.
- Mokymų projektavimas, arba mokymo sistemų projektavimas, yra praktika, kurios reikia mokomajai patirčiai įgyti ir kuri skatina efektyvesnę ir patrauklesnę žinių ir įgūdžių įgijimą.
- Didaktika yra mokymo teorija, o platesne prasme – teorija ir praktinis mokymo bei mokymosi pritaikymas.

## Mokslinė kompiuterija

- Mokslinė kompiuterija (angl. *scientific computing*) yra tarpdisciplininė sritis, kurios pagrindas – pažangūs skaičiavimai ir pajėgumas juos atlikti, sprendžiant sudėtingus uždavinius. Mokslinės kompiuterijos pagrindas yra modelių ir imitacinio modeliavimo, skirtų natūraliosioms sistemoms suprasti, kūrimas.

## Programa kaip mokymosi objektas

- Didelė dalis informatikos mokymo turinio yra programavimo kalba rašomos programos (algoritmai). Programos, kaip mokymosi objektai, yra abstrakčios, todėl studentams, ypač pirmųjų kursų, sunkiai suprantamos.
- Programavimo kalba parašyta programa yra gerai struktūriškai sutvarkyta vykdomoji specifikacija. Programa yra lankstus dalykas, todėl galimybės ją keisti, modifikuoti, prisitaikyti ar netgi vizualizuoti yra beveik neribotos.
- Mokant informatikos mokymosi objektas gali būti įrankis (priemonė) neribotų galimybių eksperimentiniams tyrimams atlikti [1].

## Modeliais grindžiamas požiūris į mokslinės kompiuterijos mokymo problematiką

Modeliais grindžiamas metodas yra šio tyrimo objektas. Jis pasirinktas dėl keleto priežasčių:

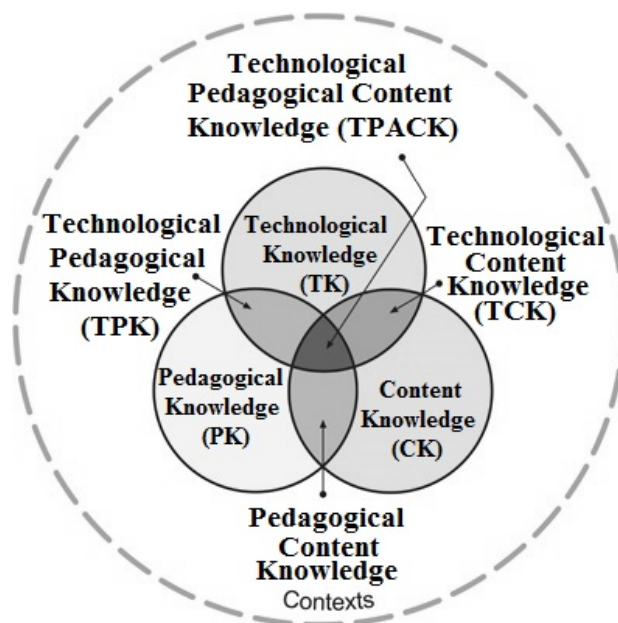
- Pirma, modeliai, jei yra laikomi kognityviniais modeliais, įskaitant intelektinius modelius, suteikia teorinius pagrindus modeliais grindžiamam požiūriui į mokymo technologijas;
- Kompiuteriniai modeliai yra pagrindiniai modeliais grindžiamo mokymo elementai. Tokių mokymo metodų pagrindas – imitacinis modeliavimas, padedantis

besimokantiejiems tobulėti;

- Be to, modeliai yra moksliniais tyrimais pagrįstų edukacinių technologijų esmė. Jų pobūdis skatina universalų požiūrį integruoto ugdymo aplinką, taikant projektavimo mokslinio tyrimo metodologiją (kompiuteriniai modeliai ir modeliavimas laikomi pažintiniais artefaktais).

## TPACK modelis

- Tiriamajam turiniui susisteminti naudojamas technologinių, pedagoginių ir turinio žinių (angl. *technological, pedagogical, content knowledge* – TPACK) modelis (žr. 1 pav.).



1 pav.: TPACK modelis. Remiantis [2, p. 9]

- Modelio paskirtis – pateikti konkrečių turinio aspektų aprašymą ir specifikacijas, apibrėžiant specifinius aspektus, pagal kuriuos galima projektuoti programavimo mokymosi išteklius, reikalingus mokant mokslinės kompiuterijos.

## 2 Tyrimo kontekstas

Šiame darbe siūlomas mokslinių tyrimų modelis – holistinis universitetinio tarpdisciplininio ugdymo sprendimas. Mokslinių tyrimų modelis – integruotas, žmogaus ir kompiuterio sąveika grindžiamas procesas, naudojamas projektuojant, kuriant ir įgyvendinant tam tikrą modeliavimą, pasitelkiant kompiuterius. Šis modelis yra glaudžiai susijęs su tokiais pažintiniais procesais kaip besimokančiojo supratimo ugdymas, taikant imitacinius kognityvinius samprotavimus. Modeliavimas gali būti laikomas artefaktu, kuris leidžia projektuojant ir vertinant sukurtus sprendimus taikyti projektavimo mokslinių tyrimų analizės metodologiją. Projektavimo mokslinis tyrimas ir modeliavimo įgūdžiai gali sudaryti reikiamą į mokslinius tyrimus orientuotos

pedagogikospagrindą ir tapti universalus universitetinio tarpdisciplininio ugdymo sprendimo pagrindu.

### **Problemos formulavimas**

Mokslinė kompiuterija (pertvarkant studijų programos sudarymo principus) galėtų tapti universitetinės gamtos mokslų, technologijų, inžinerijos mokslų ir matematikos (angl. *Science, technology, engineering, and mathematics – STEM*) ugdymo sistemos bendrąja disciplina. Mokslinės kompiuterijos mokymo metodologija galėtų būti grindžiama gerai žinomais ir išplėtotais metodais – mokslinių tyrimų metodu, imitaciniu modeliavimu ir projektavimo mokslinių tyrimų metodologija.

Mokslinių tyrimų mokymas galėtų būti apibrėžiamas šiuo aspektu: kaip kurti mokslinius imitacinius modelius (artefaktus, kurie gali būti suprojektuoti ir vertinami pagal projektavimo mokslinio tyrimo metodologiją) siekiant išspręsti specifines mokslines problemas.

Galima suformuluoti universitetines tarpdisciplinines studijas vienijančią ugdymo paradigmą: įvairiose srityse reikėtų mokyti mokslinių tyrimų pagal projektavimo mokslinių tyrimų metodologiją (kaip mokymo metodą modeliams ir imitaciniam modeliavimui projektuoti), t. y. pasitelkiant kompiuterinį imitacinį projektavimo (programavimo) modeliavimą, taip pat grindžiant holistiniu požiūriu į teorines prielaidas. Mokslinių tyrimų prasmė (imitacinio modeliavimo aspektu) yra universali visoms STEM disciplinoms. Galima apibendrinti ir suformuluoti tokį apibrėžimą: mokslinis tyrimas yra moksliniais modeliais grindžiamo imitacinių modelių kūrimas ir naudojimas.

Kyla svarbus klausimas – kodėl taikomi modeliai? Pateiktini argumentai:

- (1) Modeliais grindžiamas mokymų projektavimas (angl. *instructional design*) ir ugdymo teorijos yra gerai išplėtoti [3];
- (2) Modelis, kaip artefaktas, leidžia taikyti projektavimo mokslinių tyrimų praktinius ir analitinius metodus ir modeliais grindžiamą sistemos analizę, naudojamą tyrimams ir ugdymui;
- (3) Esama stipraus ryšio tarp studentų kognityvinės veiklos (mentalinių modelių) ir kompiuterinių modelių kūrimo (kūrimo, programavimo ir netgi iš anksto sukurtų modelių taikymo) [4];
- (4) Modelis yra imitacinio modeliavimo pagrindas.

Imitacinis modeliavimas gali būti laikomas pagrindiniu siūlomos metodologijos įrankiu. Tokį požiūrį lemia keletas priežasčių:

- (1) Tarp mentalinio modeliavimo ir kompiuterinio modeliavimo yra aiškus ryšys [5];
- (2) Imitacinis modeliavimas (vienokių ar kitokių imitacinių modelių kūrimas) yra visų modelių kūrimo dalis;
- (3) Imitacinis modeliavimas yra tam tikras modelių apibendrinimas;
- (4) Imitacinis modeliavimas suteikia galimybę kurti konstruktyvistinę aplinką ir į besimokantįjį orientuotą ugdymą.

Tyrimo klausimai:

- (1) Kokia yra mokslinės kompiuterijos mokymo ir mokymosi metodų kūrimo ir įgyvendinimo metodologinė sistema, apimanti tokias pagrindines ugdymo sistemos dalis kaip edukacinės technologijos, mokymo projektavimas, didaktiniai įrankiai?
- (2) Atsižvelgiant į bendruosius edukacinių technologijų aspektus, kokia yra universi-



tetinių STEM studijų programos metodologija, pagrindinį dėmesį skiriant tarpdisciplininiam ir moksliniais tyrimais grindžiamam ugdymui? Kaip mokslinės kompiuterijos disciplina galėtų būti integruojama į studijų programą?

- (3) Atsižvelgiant į studijų projektavimo aspektus, kaip turėtų būti organizuojamas mokslinės kompiuterijos mokymosi procesas, užtikrinantis moksliniais tyrimais grindžiamą ir konstruktyvistinį ugdymą?
- (4) Atsižvelgiant į didaktinius aspektus, kaip turėtų būti projektuojami mokymosi ištekliai mokslinei kompiuterijai mokyti?
- (5) Kokie yra tinkami stochastiniai rekurentiniai skaičiavimo modeliai, kuriais galima kurti mokymosi išteklių ir programinių mokymosi objektų, skirtų mokyti stochastikos pagrindų, bazinių tikimybinių pasiskirstymų, ribinių teoremų, eilių serijų sistemų, ir aparatinės ir programinės įrangos specifinių lygiagretinimo metodologijas?
- (6) Kaip projektuojama mokomoji priemonė, reikalinga imitaciniam modeliavimui mokyti ir į mokslinius tyrimus orientuotam mokymosi procesui ir grindžiama praktinėmis žiniomis apie lygiagretumo metodus, įskaitant žinias apie hibridines skaičiavimo platformas ir didelių duomenų apdorojimo temas? Kokie yra praktiniai mokymo priemonių įgyvendinimo pavyzdžiai?

### **Tyrimo tikslas ir uždaviniai**

- (G) Tyrimo tikslas – parengti mokymosi išteklių, įskaitant mokomuosius programavimo objektus, planavimo ir projektavimo principus, reikalingus mokant mokslinių tyrimų, remiantis mokslinės kompiuterijos edukaciniais sprendimais. Mokslinės kompiuterijos edukaciniai sprendimai ir mokymo metodai skiriami STEM tarpdisciplininėms studijoms. Sukurti mokymosi ištekliai (mokomieji programavimo objektai) yra skirti stochastikos pagrindų ir pažangiojo lygiagretinimo dalykams mokyti. Moksliniais modeliais grindžiamas imitacinis modeliavimas sudaro sąlygas konstruktyvistiniam mokymuisi. Ši universali metodologija grindžiama mokymo metodais, kuriais siekiama stiprinti tarpdisciplininį universitetinių studijų turinį, remiantis modeliavimu, problemų sprendimu ir moksliniais tyrimais.
- (U) Siekdami šio tikslo, spręsimė šiuos uždavinius:
  - (1) Sukurti mokymosi išteklių naudojimo ir integracijos metodologiją;
  - (2) Sukurti teorinį stochastinių rekurentinių reiškinų modelį Monte Karlo eksperimentams lygiagretinti. Šis modelis pagrindžia skaičiavimo modelių kūrimą;
  - (3) Sukurti algoritmus ir programinius sprendimus, skirtus eksperimentiniam eilių serijų sistemų dvigubo logaritmo dėsniai patvirtinti. Šie algoritmai grindžiami anksčiau sukurtais skaičiavimų modeliais stochastiniams rekurentiniams reiškiniams lygiagretinti.
- (T) Praktinis taikymas: pateikti pavyzdinių edukacinių sprendimų, pavyzdžiui, praktinių mokymosi išteklių rinkinį, įskaitant mokomuosius programavimo objektus, mokslinės kompiuterijos dalykui mokyti. Šių mokomųjų išteklių tikslas – pristatyti praktinius mokomųjų programavimo objektų, skirtų mokslinei kompiuterijai mokyti, pavyzdžius, reikalingus mokant stochastikos pagrindų, lygiagretinimo.

## Tyrimo metodai

Atliekant disertacijos tyrimą, buvo naudojama projektavimo mokslinio tyrimo metodologija mokymosi ištekliams kurti ir įgyvendinti.

Sprendžiamos problemos:

(A) Mokslinės kompiuterijos studijų kursai:

- (1) yra per daug techniniai;
- (2) paprastai reikalauja stiprių matematinių ir algoritminių pagrindų;
- (3) reikalauja daug prielaidinių žinių;
- (4) sunku motyvuoti besimokančiuosius.

(B) Apibrėžti:

- (1) nėra tinkamos ugdymo technologijos vizijos (be to, trūksta sprendimų, kaip kurti turinį ir mokyti mokslinės kompiuterijos);
- (2) nėra vizijos, kaip integruoti mokslinę kompiuteriją į platesnę studijų programą.

Dabartinio požiūrio į mokslinės kompiuterijos mokymą problemos:

- (A) tinkamų didaktinių metodų mokslinei kompiuterijai mokyti trūkumas;
- (B) šiuolaikinių ir tarpdisciplininių moksliniais tyrimais grindžiamų ugdymo metodų, susijusių su moksliniais skaičiavimais, trūkumas.

Pirminiai sprendimai:

- (1) Mokslinių skaičiavimų ugdymas turėtų būti grindžiamas mokslinio tyrimo mokymu;
- (2) Mokslinis tyrimas suprantamas (plačiąja prasme) kaip mokslinių tyrimų veikla, grindžiama moksliniu imitaciniu modeliavimu (įskaitant ir kompiuterinį imitacinį modeliavimą);
- (3) Mokslinis imitacinis modeliavimas grindžiamas kompiuteriniais modeliais (artefaktais);
- (4) Mokslinė kompiuterija grindžiama mokslinių modelių plėtojimo mokymu;
- (5) Imitaciniu modeliavimu grindžiamas mokymas remiasi su šiuo modeliavimu susijusiais kognityviniais procesais;
- (6) Taikomos į besimokantįjį orientuotos ugdymo technologijos;
- (7) Dėmesys sutelkiamas į projektavimo mokslinį tyrimą kaip į mokymo metodiką.

Artefakto tikslas: pateikti sprendimus (projektavimo principus), kaip kurti ir suvienodinti imitaciniu modeliavimu grindžiamus mokslinės kompiuterijos mokymui tinkamus mokymosi išteklius, žinių ir mokymosi objektus. Bendrieji kuriamos metodologijos reikalavimai:

- (1) turi pateikti moksliniu tyrimu ir imitaciniu modeliavimu grindžiamą mokslinės kompiuterijos sprendimą;
- (2) turi būti tinkama naudoti į besimokančiuosius orientuotoje ugdymo aplinkoje (remtis bendrąja mokymosi technologija, palaikyti konstruktyvistinius mokymosi metodus).

Vertinimas: grindžiamas formaliais projektavimo mokslinio tyrimo vertinimo reikalavimais, įskaitant autoriaus publikacijas [6–10] recenzuojamose žurnaluose (cituojami „Clarivate Analytics“ duomenų bazėje) ir teigiamais kolegų atsiliepimais šiuose leidiniuose.

## **Įgyvendinta tyrimų metodika**

Atliktas tyrimas remiasi gerai apibūdinta metodologija – projektavimo moksliniais tyrimais [11–23]. Nors projektavimo mokslinio tyrimo metodologija daugiausia taikoma projektuojant informacines sistemas, vis labiau ji reikalinga ir kitose srityse, pavyzdžiui, projektuojant ugdymo ar verslo valdymo technologijas. Bendras šios metodologijos taikymo įvairiose srityse bruožas – socialiniai ir techniniai aspektai. Atsižvelgtina ne tik į šiuos aspektus, bet ir į jų sąveiką. Siūlomos metodologijos tinkamumas informatikos mokymo ir mokslinės kompiuterijos moksliniams tyrimams gali būti pagrindžiamas šiais teiginiais:

- (P1) Mokslinė kompiuterija yra socialinio ir techninio tipo sistema;
- (P2) Mokslinė kompiuterija apima dalyvius (mokytojus, studentus, ugdymo institucijas, bendruomenes, kitas suinteresuotas šalis), edukacines technologijas, mokymo projektavimo metodus, mokymo priemones;
- (P3) Ugdymo priemonės yra artefaktai, įskaitant pažintinius artefaktus, pasireiškiančius kaip mokomieji programavimo objektai (programinė įranga kaip mokymosi objektas, ugdymo programinė įranga).

Tyrimo metodika – tai bendras požiūris tyrimo procedūras, nes konkrečius metodus nusako tyrimo tema ir mastas. Projektavimo mokslinis tyrimas taip pat žinomas kaip informacinių sistemų ir susijusių temų doktorantūros tyrimų metodologija [18, 20].

## **Projektavimo mokslinio tyrimo formalizavimas**

Projektavimo mokslinio tyrimo metodologinės formuluotės yra grindžiamos tam tikrais heuristiniais metodais [12]. Šis heuristinių metodų rinkinys yra įtrauktas į projektavimo struktūrą [11]. Tai sukuria sistemingą tyrimo metodologiją:

- (1) Koks yra tyrimo klausimas (projektiniai reikalavimai)?
- (2) Kas yra artefaktas? Kaip parodomas artefaktas?
- (3) Kokie projektavimo procesai naudojami kuriant artefaktą?
- (4) Kaip artefaktas ir projektavimo procesai atvaizduojami žinių bazėje?
- (4a) Kokios, jei yra, teorijos remia artefakto projektavimą?
- (5) Kurie vertinimai atliekami vidinių projektavimo ciklų metu? Kokie projektavimo tobulinimai atliekami kiekvieno projekto ciklo metu?
- (6) Kaip artefaktas integruojamas į taikomąją aplinką ir kaip jis išbandomas? Kokios metrikos naudojamos siekiant parodyti, kad, palyginti su ankstesniais artefaktais, šie artefaktai yra naudingesni ir tobulesni?
- (7) Kokių naujų žinių ir kaip pridedama prie žinių bazės (pavyzdžiui, recenzuojama literatūra, metaartefaktai, nauja teorija ir naujas metodas)?
- (8) Ar tyrimo klausimas tinkamai adresuotas?

Išvados: projektavimo mokslinis tyrimas, kaip tyrimo metodologija, numato visą mokslinio tyrimo užduočiai (suprojektuoti principus, kaip kurti mokymosi objektus, reikalingus mokant mokslinės kompiuterijos) būtiną formalizaciją.

## **Metodologiniai reikalavimai ir jų įgyvendinimas**

Metodologinių charakteristikų santrauka, formalūs projektavimo mokslinio tyrimo reikalavimai ir jų įgyvendinimas pateikiami 1 lentelėje.

**1 lentelė:** Formalūs mokslinio tyrimo reikalavimai ir įgyvendinimas

Charakteristikos	Projektavimo mokslinio tyrimo reikalavimai	Įgyvendinimas
Tikslai	Sukurti artefaktus, kurie leidžia pateikti praktinių problemų sprendimus	Sukurti mokslinės kompiuterijos mokymosi išteklių projektavimo principus ir palaikymo taikymo ir integravimo metodologiją
Pagrindinės veiklos	Apibrėžti problemą; siūlyti; kurti; įvertinti; apibendrinti	<p><b>Mokslinės kompiuterijos mokymo problemos:</b></p> <ul style="list-style-type: none"> <li>Nėra dokumentais patvirtintų mokslinės kompiuterijos mokymosi išteklių projektavimo principų ir palaikymo taikymo ir integravimo metodologijos</li> </ul> <p><b>Pasiūlymas (tiriamoji užduotis):</b></p> <ul style="list-style-type: none"> <li>Sukurti mokslinės kompiuterijos mokymosi išteklių projektavimo principus kuriė įgyvendina konstruktyvistine paradigma</li> <li>Pagal numatytą metodiką sukurti praktinių mokslinės kompiuterijos mokymosi išteklių pavyzdžių</li> </ul> <p><b>Vertinimas:</b></p> <ul style="list-style-type: none"> <li><i>Ex ante</i> vertinimas: išsami literatūros apžvalga, loginės išvados, patvirtinimas</li> <li><i>Ex post</i> vertinimas: prototipų demonstracijos, atvejų analizė</li> </ul> <p><b>Išvados:</b></p> <ul style="list-style-type: none"> <li>Sukurti mokymosi išteklių projektavimo principai ir palaikymo taikymo ir integravimo metodologija. Sukurti principai ir metodologija įgyvendina konstruktyvistinę paradigmą</li> <li><i>Ex ante</i> ir <i>Ex post</i> vertinimai yra įvykdyti</li> <li>Pateiktas mokymosi išteklių pavyzdys ir atvejų tyrimai</li> </ul>
Rezultatai	Artefaktai (konstrukčiai, modeliai, metodai, projektavimo principai, praktiniai atvejai) ir teorijų tobulinimas	Mokslinės kompiuterijos mokymosi išteklių projektavimo principai; mokslinės kompiuterijos mokymosi išteklių pavyzdžiai ir atvejų tyrimai
Žinių rūšis	Priskirtinas	Kaip kurti mokymosi išteklius tarpdisciplininiam ir vientisam požiūriui į mokslinės kompiuterijos mokymą stiprinti

Tyrėjo vaidmuo	Artefakto kūrėjas ir (arba) vertintojas	Tyrimo apimtis: mokslinės kompiuterijos ugdymo dabartinės būklės analizė; loginė analizė, teoriniai teiginiai ir psichologinio, technologinio ir mokomojo aktualumo patvirtinimas, taip pat įgyvendintų mokymosi išteklių teoriniai pagrindai; metodikos kūrimas; pavyzdinių mokymosi išteklių kūrimas; demonstracijų atvejų analizė
Empirinis pagrindas, įgyvendinimas	Nėra būtina	Nėra įgyvendinta
Rezultatų įvertinimas	Taikymas; modeliavimas; eksperimentai	Remiantis <i>Ex ante</i> ir <i>Ex post</i> vertinimų metodika, įgyvendinant projektavimo mokslinio tyrimo reikalavimus
Metodas	Kokybinis ir (arba) kiekybinis	Remiamasi kokybiniu požiūriu, įskaitant teorinę analizę ir literatūros apžvalgą
Specifiškumas	Apibendrinamas tam tikros klasės problemoms	Apibendrinamas tarpdisciplininio ugdymo turinio, reikalingo teikiant tyrimais grindžiamą universitetinį išsilavinimą, kūrimo problemoms

### Taikymo sritis, tyrimo klausimas

Universitetinių studijų sistemos bendrosios srities (bendrojo domeno) taikomųjų sričių (domenų) hierarchiją sudaro susijusių posričių (subdomenų) visuma:

- universitetinių studijų sistema;
- e. mokymasis;
- tarpdisciplininė universitetinė studijų programa;
- mokslinės kompiuterijos mokymas (disciplina);
- kompiuterinis imitacinis modeliavimas.

Tyrimo klausimas (projektavimo reikalavimai) yra mokslinės kompiuterijos mokymosi išteklių projektavimo principai ir mokomųjų programavimo objektų palaikymo taikymo ir integravimo metodologija. Projektuojami mokymosi ištekliai turi atitikti konstruktyvistinius ugdymo principus. Mokslinės kompiuterijos mokymosi išteklių projektavimo principai ir palaikymo taikymo ir integravimo metodologija turėtų skatinti konstruktyvistinį požiūrį į tarpdisciplininį universitetinį išsilavinimą ir formalizuoti edukacinius mokslinės kompiuterijos sprendimus:

- mokymo technologijų aspektu;
- mokymų metodų projektavimo aspektu;
- dalyko turinio kūrimo aspektu: pateikti mokslinės kompiuterijos mokymosi išteklių planavimo metodiką.

### Mokslinio tyrimo rezultatai

Aprašysime mokslinės kompiuterijos mokymosi išteklių projektavimo, taikymo ir integravimo metodologijos projektavimą, kuris apima įgyvendinimo gaires, rekomendacijas, projektavimo kriterijus, heuristiką, atvejų tyrimus. Pateikiame mokslinės

kompiuterijos mokymo sričių pavyzdžių:

- edukacinės technologijos sritis;
- mokymų projektavimo metodai;
- mokymosi išteklių ir žinių objektų plėtra;
- mokslinės kompiuterijos mokymosi objektai.

Iš ugdymo technologijų perspektyvos siūloma tarpdisciplininės STEM studijų programų kūrimo metodologija. Ja remiantis galima sukurti tarpdisciplininę ir inovatyvią mokymosi aplinką.

Iš mokymų projektavimo perspektyvos:

- (1) išplėtotą mokslinės kompiuterijos studijų apibrėžtis, kuri sudaro novatoriškos studijų metodikos įgyvendinimo pagrindą;
- (2) pateikiama mokymo turinio struktūrizavimo metodika;
- (3) sukurta mokomoji priemonė, grindžiama dedukciniais ir logines išvadas apibūdinančiais metodais.

Iš turinio projektavimo perspektyvos:

- (1) pateikiami studijų turinio kūrimo principai; projektavimo principai grindžiami metodais ir reikalavimais, leidžiančiais kurti mokymosi turinį, lemiantį konstruktyvią studijų aplinką;
- (2) pateikiama praktinių mokymosi išteklių, įskaitant mokomuosius programavimo objektus, pavyzdžių.

### **Pagrindimas, remiantis žinių baze. Teorinis pagrindimas.**

Darbe siūlomi edukaciniai sprendimai grindžiami sistetine literatūros apžvalga. Taip pat darbe pateikiamas sisteminis teorinis pagrindimas.

### **Tyrimo rezultatų vertinimas**

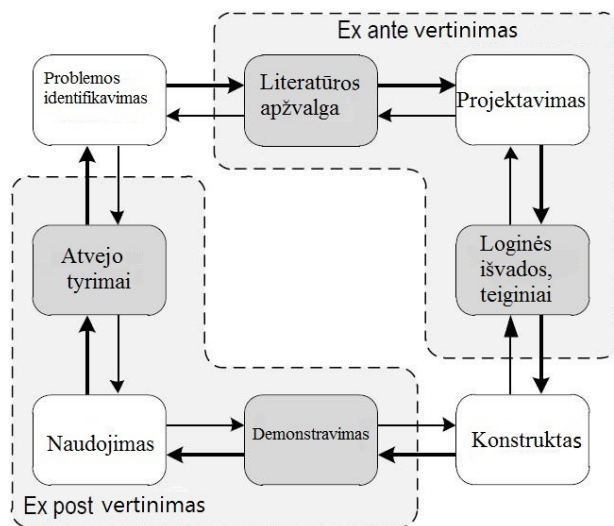
Tyrimo rezultatų vertinimas grindžiamas formaliomis vertinimo procedūromis, kurios atitinka formalius projektavimo mokslinio tyrimo metodologijos reikalavimus [18, 24, 25]. Praktinį vertinimą sudaro dvi pagrindinės dalys: *Ex ante* (vertinimas (remiamasi prognozėmis, o ne faktiniais rezultatais) ir *Ex post* vertinimas (remiamasi ankstesnių rezultatų analize). *Ex ante* vertinimas grindžiamas sistetine literatūros apžvalga, loginėmis išvadomis ir šį vertinimą patvirtinančiais modeliais. *Ex post* vertinimas apima galutinių rezultatų pristatymą ir atvejo tyrimus (žr. 2 pav.).

### **Įgyvendinimo pavyzdys**

Darbe pateikti mokomųjų programavimo objektų metodikos įgyvendinimo pavyzdžiai.

### **Mokslinis naujumas**

- (1) Sukurti mokslinės kompiuterijos studijų kurso, grindžiamo moksliniu tyrimu, mokymo išteklių projektavimo principai; sudaryta jų taikymo ir integravimo metodologija. Taikant konstruktyvistinį požiūrį į universitetines STEM dalykų studijas, pateikiama išsami mokslinės kompiuterijos mokymo srities analizė;
- (2) Įdiegti mokslinių tyrimų ir mokslinės kompiuterijos studijų projektavimo metodai, grindžiami modeliais;



2 pav.: Tyrimo rezultatų vertinimas. Adaptuota iš [25, p. 14]

(3) Pateiktas didaktinis stochastikos pagrindų ir pažangiojo lygiagretinimo modelis.

### Inžinerinis naujumas

- (1) Sukurtas stochastinių rekurentinių išraiškų išreikštinio lygiagretinimo skaičiavimo modelis;
- (2) Sukurtas skaičiavimo modelis, kuriuo eksperimentiškai patvirtinamos eilių serijų sistemų ribinės teoremos sunkiojo transporto eismo salygomis;
- (3) Sukurta modeliais grindžiama mokslinės kompiuterijos mokymosi išteklių kūrimo sistema;
- (4) Sukurtas mokymosi išteklių rinkinys (mokomieji programavimo objektai), pagal kurį gali būti mokoma stochastikos pagrindų;
- (5) Sukurtas mokymosi išteklių rinkinys (mokomieji programavimo objektai), pagal kurį gali būti mokoma pažangiojo lygiagretinimo.

### Praktinė pasiektų rezultatų reikšmė

- (1) Pateikti moksliniai rezultatai leidžia kurti inovatyvią mokslinės kompiuterijos studijų programą. Tokia studijų programa sustiprina tarpdiscipliniškumą ir yra orientuota į moksliniais tyrimais grindžiamus ir konstruktyvistinius edukacinius metodus. Tyrime pateikiama universali sistema, kuri apima ugdymo technologijas, mokymų projektavimą ir mokslinės kompiuterijos dalyko didaktiką ir pagal kurią mokslinės kompiuterijos dalykas gali būti integruojamas į universitetinių STEM studijų programas;
- (2) Pateikti inžineriniai rezultatai leidžia kurti praktiškai taikomus mokslinės kompiuterijos mokymosi išteklius kaip mokomuosius programavimo objektus pagal konstruktyvistinę paradigmą.

### Ginamieji teiginiai

- (1) Sukurtais mokymo išteklių projektavimo principais galima projektuoti ir tobulinti mokslinės kompiuterijos dalyko mokymosi išteklius, įskaitant atitinkamus programinės įrangos mokymosi objektus. Projektavimo principai atitinka pro-

jektavimo reikalavimus – modeliais grindžiamus ugdymo metodus ir konstruktyvistines paradigmas. Sukurta palaikomoji metodologija yra tinkama taikant mokymosi išteklius tarpdisciplininėje universitetinių studijų programoje ir juos integruojant į šią programą;

- (2) Sukurti stochastinių rekurentinių modelių Monte Karlo eksperimento lygiagretinimo algoritmai yra tinkami mokymosi ištekliams (mokymo programoms), tokiems kaip stochastikos pagrindų ir lygiagretinimo mokymosi ištekliai, projektuoti.

## **Aprobavimas ir publikacijos**

Disertacijos rezultatai buvo pristatyti ir aptarti šiose nacionalinėse ir tarptautinėse konferencijose:

- 4th Doctoral Consorciium on Informatics Engineering Education Research, Druskininkai, Lithuania, 2013.12.03–07
- 10th International Seminar on Informatics Contests, Druskininkai, Lithuania, 2014.06.03–06
- 5th Doctoral Consorciium on Informatics Engineering Education Research, Druskininkai, Lithuania, 2014.11.26–30
- ITiCSE 2014 19th Annual Conference on Innovation and Technology in Computer Science Education, June 23–25, 2014, Uppsala, Sweden
- 8th International conference on E-learning, The University of La Laguna, Tenerife, Spain, 2014.09.06–14
- 6th Doctoral Consorciium on Informatics Engineering Education Research, Druskininkai, Lithuania, 2015.12.08–12
- ISSEP 2017, The 10th International Conference on Informatics in Schools, November 13–15, 2017, University of Helsinki, Helsinki, Finland

Pagrindiniai disertacijos rezultatai paskelbti šiuose straipsniuose:

- Dolgopolovas V, Dagienė V, Minkevičius S, Sakalauskas L. (2015) Teaching scientific computing: a model-centered approach to pipeline and parallel programming with C. *Scientific Programming*. 2015 Jan 1;2015:11.
- Dolgopolovas V, Dagienė V, Minkevičius S, Sakalauskas L. (2014). Python for Scientific computing Education: Modeling of Queueing Systems. *Scientific Programming*, 2014 Jan 1;22(1):37-51.
- Minkevičius S, Dolgopolovas V, Sakalauskas L (2014) A law of the iterated logarithm for the sojourn time process in queues in series. *Methodology and Computing in Applied Probability*. 2016 Mar 1;18(1):37–57.
- Dolgopolovas V, Jevsikova T, Dagienė V, Savulionienė L. Exploration of Computational Thinking of Software Engineering Novice Students Based on Solving Computer Science Tasks. *International Journal of Engineering Education*. 2016 Jan 1;32(3):1-10.
- Dolgopolovas V, Jevsikova T, Dagienė V. From Android games to coding in C – An approach to motivate novice engineering students to learn programming: A case study. *Computer Applications in Engineering Education*. 2018 Jan 1;26(1):75–90.



## Tyrimo publikacijos tūrinis

- Tyrimo kontekstas
  - Motyvacija, tikslas, užduotys, tyrimo metodai.
  - Naujumas, ginamieji teiginiai, aprobavimas ir publikacijos, turinys.
- TPACK modelio įgyvendinimo problema mokant mokslinės kompiuterijos
- Teorinis tyrimas
  - Mokslinės kompiuterijos edukacinės srities teorinis modelis.
  - Stochastinių rekurentinių sąryšių skaičiavimo modeliai.
- Eksperimentiniai tyrimai: edukaciniai sprendimai
  - Eilių serijų sistemų parametrų skaičiavimo modelis perkrovos sąlygomis.
  - Atvejo analizė: stochastikos pagrindų mokomieji programavimo objektai.
  - Atvejo analizė: pažangiojo lygiagretinimo metodų mokomieji programavimo objektai.
- Tyrimo rezultatų vertinimas
  - Parengtų mokymosi išteklių pavyzdžių specifikacija.
  - Vertinimo metodikos aprašas.
  - Autoriaus atliktas tyrimo rezultatų vertinimas.
  - Ekspertų vertinimo santrauka.
- Priedai
  - Mokomųjų programavimo objektų pavyzdžiai.
- Šaltiniai

## 3 TPACK modelio įgyvendinimo problema mokslinės kompiuterijos mokymo srityje

TPACK modelis skirtas mokslinio tyrimo kontekstui sisteminti ir struktūrizuoti. Tai gali būti mokslinio tyrimo konteksto metastruktūra, sutelkianti dėmesį į tris pagrindines sritis: ugdymo technologijas, mokymų projektavimą ir didaktinius mokslinės kompiuterijos aspektus. Tai leidžia numatyti pagrindines tyrimo kryptis. Modelis nagrinėja šias sritis (šiuos domenus) ir jų ypatybes:

### (K) Kontekstas

- (1) tarpdisciplininės universitetinės studijos apskritai ir šių studijų sistemos aspektai tarpdisciplininiu inovacijų ir tyrimų požiūriu;
- (2) mokslinės kompiuterijos edukacinių aspektų, įskaitant ir esamus požiūrius į mokslinės kompiuterijos mokymą, analizė;
- (3) bendrieji ugdymo technologijų aspektai, orientuoti į konstruktyvistinius mokymo metodus, įskaitant informatikos edukacinius aspektus;
- (4) modeliais grindžiami metodai ir modeliavimas ugdant;
- (5) informatikos ir mokslinės kompiuterijos ugdymo sričių sankirta;
- (6) mokslinių tyrimų aspektai ir jų vieta STEM studijose;
- (7) didaktiniai mokomųjų programavimo objektų naudojimo aspektai, įskaitant galimą projektavimo mokslinių tyrimų metodologijos formalizavimą.

### (C) Technologinė sritis

- (1) aparatinės įrangos architektūra;

- (2) mokslinė kompiuterija ir konkrečios aparatinės įrangos architektūros lygiagretinimo aspektai;
  - (3) programinės įrangos inžinerijos technologijos apskritai ir modeliu grindžiami programinės įrangos kūrimo metodai.
- (T) Turinio sritis
- (1) Turinys grindžiamas stochastiniu rekurentiniu modeliu. Bendras modelis – tai daugianaris, nelinijinis stochastinis rekurentinis modelis.
- (P) Pedagoginė sritis grindžiama ugdymo technologijomis, mokomojo projektavimo ir mokslinės kompiuterijos didaktikos aspektais

Apibendrinant atliktą analizę galima padaryti tokias išvadas:

- Apibūdinant ugdymo turinio sritį (domeną), pagrindinis dėmesys turi būti skiriamas šioms funkcijoms:
  - (1) ugdymo technologijos turi būti ištirtos tarpdisciplininio universitetinio išsilavinimo požiūriu kaip vienijančios paradigmos, susijusios su universitetinių studijų programa;
  - (2) mokomoji medžiaga turi būti orientuota į studijas, kuriose studentams suteikiama galimybė kurti imitacinius modelius; modeliavimas galėtų būti įtrauktas į ugdymo sistemą, numatančią atitinkamus mokslinius tyrimus;
  - (3) didaktinis metodas gali būti orientuotas į formalizavimą, kuris leidžia tinkamai sutvarkyti anksčiau aprašytą mokymo (mokymosi) procesą; pagrindinis dėmesys galėtų būti skiriamas pažintiniams ugdymo aspektams, įskaitant tokius svarbius elementus kaip parama (angl. *scaffolding*) ir pagrindimas (angl. *grounding*).
- Nurodant technologinio turinio sritį, pagrindinis dėmesys galėtų būti skiriamas šiems dalykams:
  - (1) tinkamam lygiagretinimui, leidžiančiam įvairių aparatinės įrangos architektūrą;
  - (2) programinės įrangos inžinerijos metodams, skatinantiems į modelius orientuotą programinės įrangos kūrimo požiūrį;
  - (3) lygiagretinimą užtikrinantiems programinės įrangos kūrimo ar programavimo įrankiams;
  - (4) didelių duomenų apdorojimo ir (ar) programavimo ir pateikimo įrankiams.
- Turinio srities apibrėžimas. Pagrindinis dėmesys gali būti skiriamas mokymo (mokymosi) aspektams:
  - (1) įvadinės stochastikos ir tikimybių teorijos teoriniams aspektams;
  - (2) įvadinių ir stochastinių rekurentinių sąryšių teoriniams ir skaičiavimo aspektams;
  - (3) pasiskirstymui, ribinėms teorems ir stochastiniams procesams tikimybių teorijoje;
  - (4) Monte Karlo modeliavimo metodams ir Monte Karlo modeliavimo skaičiavimo aspektams;
  - (5) eilių teorijos pagrindams, pagrindiniams eilių teorijos parametrams ir charakteristikoms, eilių serijų sistemoms, aptarnavimo tinklams ir eilių teorijos modelių skaičiavimo aspektams.

## 4 Mokslinės kompiuterijos TPACK modelio konteksto specifikacija

Kontekstas suteikia aiškų ypatybių rinkinį, pagal kurį įgyvendinami tam tikrų sričių modeliai. Šios savybės gali būti nurodomos kaip:

- (1) tarpdiscipliniškumą užtikrinančios funkcijos; tai iš esmės svarbu apibrėžiant ugdymo politiką;
- (2) konstruktyvistiniai ugdymo metodai ir savybės; tai leidžia tinkamai orientuoti ugdymo technologijų plėtojimo kryptį ir suprojektuoti bei aprašyti mokymo projektavimo metodus;
- (3) modeliu grindžiami metodai; į imitacinių modelių taikymą orientuoti metodai; šios savybės tikslina konstruktyvistinių ugdymo metodų kūrimo kryptis;
- (4) programavimo mokymosi objektų aprašymo ir specifikacijų tikslinimas atsižvelgiant į mokslinės kompiuterijos studijų programas;
- (5) su moksliniu tyrimu susijusio konteksto specifikacija pagrindžia į modelius ir imitacinį modeliavimą orientuotų metodų vystymąsi;
- (6) projektavimo mokslinio tyrimo specifikacija ir jos ypatybės tyrimo kontekste; tai leidžia toliau kurti ir įgyvendinti projektavimo mokslinio tyrimo metodologiją ir ją taikyti šio tyrimo kontekste ir srityse.

## 5 Mokslinės kompiuterijos ugdymo srities savybių metaanalizė

### Ugdymo turinio srities analizė

Aprašytas metodas pagrindžia praktinę mokymo metodiką, skirtą konstruktyvistiniam, projektais grindžiamam mokymuisi. Pateiktos metodikos tikslas – remti universalų požiūrį į universitetinį STEM ugdymą, sudarant sąlygas bendram tarpdiscipliniškumo ir inovacijų pagrindui. Remdamiesi skaitmeninės pedagogikos praktika, į ugdymo procesą įtraukiame abdukcijos ir loginių ribojimų (angl. *circumscription*) sąvokas. Tai svarbu šiuolaikinei mokslo aplinkai, nes skatina mokslines inovacijas. Tuo pačiu metu tokie gerai parengti metodai, kaip antai modeliais grindžiamas požiūris į mokymą ir projektavimo mokslinis tyrimas, yra įgyvendinami kaip moksliniais tyrimais grindžiamos pedagogikos universitetinės STEM studijos.

Modeliu grindžiamas metodas suteikia pagrindą studijoms, kuriose stiprinami moksliniai tyrimai ir projektiniai konstruktyvistiniai mokymo metodai. Projektavimo mokslinio tyrimo formalizavimas remiasi universaliomis priemonėmis, todėl studentai gali studijuoti tarpdisciplininėmis grupėmis kaip studentai-mokslininkai, kelti hipotezes, projektuoti imitacinius modelius, vertinti rezultatus ir naudoti projektavimo mokslinio tyrimo technologijas. Dėstytojai turėtų suteikti tyrimui tinkamą pedagoginę aplinką kaip iš anksto suplanuotą daugialypį modelį, mokymo instrukcijas ir vientisas teorines žinias. Nagrinėjamos pedagoginio turinio sritys:

- (1) Edukacinė mokslinės kompiuterijos technologija;

- (2) Mokslinės kompiuterijos mokymo projektavimas;
- (3) Didaktiniai mokslinės kompiuterijos studijų aspektai.

### **Mokomojo turinio srities analizė**

Modeliavimo mokymosi aplinkai reikalingas sistemingas požiūris į turinio kūrimą. Mokymo turinio kūrėjas turėtų užtikrinti sklandų kelią nuo mokymosi aplinkos koncepcijos iki praktinių problemos sprendimų, kuriuos studentai galėtų imituoti kompiuteriais. Šiuo atveju svarbiausias veiksnys yra suprasti, kaip imitacinis modeliavimas gali paskatinti mokymąsi skirtinguose kontekstuose [4]. Tinkamas mokymosi būdas yra grindžiamas imitacinių modelių kūrimo mokymosi būdu. F. Landriscina [4, p. 99] teigia, kad studentai, kurdami imitacinį modelį, turi vartoti programavimo kalbą arba tam tikrą modeliavimo ir imitacinę programinę įrangą.

Nagrinėjamos mokomojo turinio srities dalys:

- (1) Pagrindiniai mokslinės kompiuterijos ugdymo turinio projektavimo principai;
- (2) Pagrindiniai mokslinės kompiuterijos įvadinio turinio, grindžiamo modeliais, projektavimo principai.

### **Technologinio turinio srities modelis**

Technologinio turinio srities modelis svarbus dėl dviejų priežasčių.

- (1) Pirma, technologija atlieka svarbų vaidmenį, suteikdama platformą kompiuteriniams eksperimentams, įskaitant modeliavimo, imitacinių modelių ir lygiagretinimo temas;
- (2) Antra, pati technologija yra tema, kuri turi būti įtraukta į studijų programą, ypač tiriant lygiagretinimo algoritmus.

Technologijos sritis susideda iš kelių pagrindinių dalių:

- (1) aparatinės platformos skaičiavimams;
- (2) programinės įrangos priemonės algoritmams įgyvendinti;
- (3) programinės įrangos inžinerijos temos ir technologijos.

## **6 Lygiagretinimo mokymo srities analizė**

Šiame skyriuje pateikti metamodeliai ir algoritmai suteikia išsamią informaciją apie taikomas programas, kurios bus aprašytos kituose skyriuose.

- (1) Pirma, tokios skaičiavimo platformos – didelio našumo skaičiavimo klasteriai – leidžia įgyvendinti skirtingus algoritmus ir programavimo metodus taikant lygiagretinimą. Tai apima bendrintosios, paskirstytosios ir hibridinės atminties sprendimus. Šio tyrimo tikslais didelio našumo skaičiavimo platforma yra naudojama kaip universali platforma, leidžianti išbandyti skirtingas lygiagretinimo technologijas. Tokiu būdu atliekama skirtingų programavimo sprendimų lyginamoji analizė, didinama studentų motyvacija ir skatinamas konstruktyvistinis požiūris į mokymąsi.
- (2) Antra, bendrojo pobūdžio stochastiniai rekurentiniai reiškiniai yra tinkamas skaičiavimų modelis. Jis gali būti taikomas tam tikroms sritims modeliuoti.

Pritaikius stochastini modelį, galima išbandyti daugialypius lygiagretinimo metodus, remiantis Monte Karlo modeliavimu.

- (3) Trečia, modeliu grindžiamu požiūriu galėtų būti remiamasi projektuojant ir tobulinant programavimo mokymosi objektus, skirtus lygiagretinimui mokyti. Šis požiūris tinkamas ne tik neišreikštiniais, bet ir išreikštiniais lygiagretinimo metodams.

### Programavimo mokymosi objektų modeliai lygiagretinimui mokyti

Šiame skyriuje aptarsime teorinius mokomuosius programavimo objektus, kurie yra grindžiami stochastiniais rekurentiniais modeliais ir kurių paskirtis yra lygiagretinimo mokymas. Tokio požiūrio privalumas yra tas, kad stochastiniai modeliai leidžia įgyvendinti daugialypio modeliavimo procesą projektuojant modelius ir algoritmus. Tuo pat metu rekurentiniai sąryšiai leidžia įtraukti papildomus matmenis projektuojant skaičiuojamuosius modelius. Tyrime ištirti dviejų tipų modeliai:

- (I) neišreikštiniai modeliai, paprastai įgyvendinami remiantis kategorinių duomenų tipų funkcinio programavimo paradigma. Tokių modelių atveju lygiagretinimas laikomas automatiniu procesu, grindžiamu programavimo skeletonais.  
(E) išreikštiniai modeliai, grindžiami lygiagretinimo metodais.

### Neišreikštinis lygiagretinimo mokymo modelis

Linijiniai stochastiniai rekurentiniai reiškiniai gali būti užrašomi taip:

$$\begin{aligned}x_0 &= B_0 \\x_i &= (x_{i-1} \otimes A_i) \oplus B_i, 1 \leq i \leq n\end{aligned}$$

Čia  $A_i, 1 \leq i \leq n$  ir  $B_i, 0 \leq i \leq n$  yra tarpusavyje nepriklausomi ir vienodai pasiskirstę (poromis) atsitiktiniai kintamieji, t.y.:

$A_i, 1 \leq i \leq n$  ir  $B_i, i = 0 \leq i \leq n$  yra nepriklausomi atsitiktiniai kintamieji, kurių pasiskirstymo funkcijos atitinkami lygios kintamųjų  $A$ , t.y.  $A(x) = P(A < x)$ , ir  $B$ , t.y.  $B(x) = P(B < x)$ , pasiskirstymo funkcijoms, o momentai (jei egzistuoja) yra atitinkamai lygus  $A_k = M(A^k)$  ir  $B_k = M(B^k)$ .

### Monte Karlo atrankos algoritmas

Atsitiktinių kintamųjų reikšmės  $x_i, 1 \leq i \leq n$  priklauso nuo atsitiktinių kintamųjų  $A$  ir  $B$ . Jų  $A$  ir  $B$  pagrindinis tipas žymimas  $R_A$  ir  $R_B$ , o pagrindinis kintamojo  $x$  tipas žymimas  $A$ . Tada:

- (1) apskaičiuojame koeficientų  $A$  ir  $B$  rinkinius:  $mc_A :: \tilde{R}_A \rightarrow R_A^*$ , čia  $R_A^*$  žymi pagrindinio tipo  $R_A$  sąrašą, o  $\tilde{R}_A$  žymi apibendrintą atsitiktinę reikšmę.
- (2) rekurentinio reiškinio reikšmėms apskaičiuoti taikome sąrašo „map“ funkcionalą:  $\tau_R :: R_A \times R_A \rightarrow A$ .
- (3) galutiniam rezultatui gauti galime taikyti sąrašo „reduce“ funkcionalą:  $\setminus_{\ominus}$ .

### Neišreikštinis stochastinių rekurentinių reiškinų lygiagretinimo modelis

Pritaikysime  $\tau_R$  operaciją rekurentiniams reiškiniams skaičiuoti [26, p. 104],  $x_n$  kin-

tamojo vertinimą galima apibrėžti šitaip:  $\tau_R = x \otimes /_{b_0} \oplus y$ :

$$x \otimes /_{b_0} \oplus y = \begin{cases} b_0, & \text{if } \#x(= \#y) = 0; \\ b_0 \otimes \pi_1 A \oplus \pi_2 A, & \text{if } \#x(= \#y) \neq 0; \end{cases}$$

čia

$$\begin{aligned} A &= \otimes / (x \curlywedge_{\otimes} y), \\ a \otimes b &= (a, b), \\ (a, b) \otimes (c, d) &= (a \otimes c, b \otimes c \oplus d), \\ \pi_1(a, b) &= a, \text{ and } \pi_2(a, b) = b. \end{aligned}$$

Gauname skaičiavimo modelį:  $\setminus_{\ominus} \circ \tau_R * \circ (mc_A \nabla mc_A) :: \tilde{R}_A \times \tilde{R}_A \rightarrow A$ .

Toliau aptarsime rekurentinio reiškinių vertinimo strategiją, kai naudojami kompiuterių klasteriai:

- (1) Naudojame lygiagretųjį „map“ funkcionalą *parMap* tam, kad paskirstytume koeficientų  $A$  ir  $B$  rinkinius tarp vertinimo mazgų (angl. *computational nodes*):

$$\begin{aligned} (Trans \tilde{R}_A, Trans R_A) &\Rightarrow (\tilde{R}_A \rightarrow R_{A*}) \rightarrow (\tilde{R}_A \rightarrow R_{A*}) \\ &\rightarrow (\tilde{R}_{A*} \rightarrow R_{A**}) \nabla (\tilde{R}_{A*} \rightarrow R_{A**}) \end{aligned}$$

- (2) toliau funkcionalą „MapRedr“ *parMapRedr* ( $\setminus_{\ominus}, e, \otimes /_{b_0} \oplus$ ) (čia funkcionalo „Redr“ vienetas žymimas  $e$ ) naudotume tam, kad įvertintume rekurentinį reiškinių kiekviename mazge:

$$\begin{aligned} (Trans R_{A*}, Trans A) &\Rightarrow (R_A \rightarrow R_A \rightarrow R_A) \\ &\rightarrow (R_A \rightarrow R_A \rightarrow A) \rightarrow R_{A*} \rightarrow R_{A*} \rightarrow A \end{aligned}$$

## Išreikštinis stochastinių rekurentinių reiškinių lygiagretinimo modelis

Netiesinis stochastinių rekurentinių reiškinių modelis:

Apibrėžiame atsitiktinį dydį  $\mathbf{a}_n = (a_n^1, a_n^2, \dots, a_n^p)$  ir  $\mathbf{b}_n = (b_n^1, b_n^2, \dots, b_n^p)$ ,  $n = 1, 2, \dots$ , čia  $a_n^i$  ir  $b_n^i$ ,  $i = 1, 2, \dots, p$  yra vienodai pasiskirstę atsitiktiniai dydžiai (kiekvienoje sekoje). Dydžiai  $a_n^i$ ,  $i = 1, 2, \dots, p$ ;  $n = 1, 2, \dots$  pasiskirstę kaip atsitiktinis dydis  $a$  su pasiskirstymo funkcija  $A(x) = \mathbf{P}(a < x)$  ir dydžiai  $b_n^i$ ,  $i = 1, 2, \dots, p$ ;  $n = 1, 2, \dots$ , pasiskirstę kaip atsitiktinis dydis  $b$  su pasiskirstymo funkcija  $B(x) = \mathbf{P}(b < x)$ . Atsitiktinio vektoriaus  $\mathbf{x}_n = (x_n^1, x_n^2, \dots, x_n^p)$ ,  $n = 1, 2, \dots$  pirmosios eilės nelineinį stochastinį rekurentinį reiškinių apibrėžiame šitaip:

$$x_{j+1}^{i+1} = f(x_{j+1}^i, x_j^{i+1}, a_{j+1}^{i+1}, b_{j+1}^{i+1}), \quad (1)$$

$x_1^i = f'(a_1^i, b_1^i)$ ,  $x_j^1 = f''(a_j^1, b_j^1)$ ,  $i = 1, 2, \dots, p$ ;  $j = 1, 2, \dots$ , čia  $f(x)$ ,  $f'(x)$ ,  $f''(x)$  apibrėžia nelineines atsitiktinių dydžių funkcijas.

## Išreikštinio modelio Monte Karlo eksperimentas

Monte Karlo eksperimentas [27] remiasi šiomis prielaidomis. Imdami atsitiktinį dydį  $a$ , kurio turi pasiskirstymo funkcija  $A(x) = \mathbf{P}(a < x)$ , apibrėžiame  $k$ -tąjį momentą (jei egzistuoja):  $a_k = \mathbf{M}a^k = \int_0^\infty a^k dA(x)$ .  $U_{[0,1]}$  apibrėžia tolygaus pasiskirstymo funkciją, o  $A^-$  apibrėžia funkcijos  $A$  apibendrintą atvirkštinę funkciją

$$A^-(u) = \inf\{x; A(x) \geq u\}, u \in [0, 1].$$

Naudojant lema [28, p. 36], sugeneruojamos atsitiktinio dydžio  $a \sim A$  reikšmes. Generuojame  $U$  pagal  $U_{[0,1]}$  ir atliekame transformaciją  $x = A^-(u)$ . Su šia transformacija sugeneruojamas atsitiktinio dydžio  $a$  neatsitiktinių reikšmių rinkinys:  $(a', a'', \dots)$ .

## Skaičiavimo modelis

Apibrėžkime atsitiktinius dydžius  $a$  ir  $b$  ir jų pasiskirstymo funkcijas  $A(x)$  ir  $B(x)$ , neatsitiktines Monte Karlo reikšmių kintamųjų  $a_j^i$  ir  $b_j^i$  matricas  $\mathbf{a}_n = (\mathbf{a}_n^1, \mathbf{a}_n^2, \dots, \mathbf{a}_n^p)$  ir  $\mathbf{b}_n = (\mathbf{b}_n^1, \mathbf{b}_n^2, \dots, \mathbf{b}_n^p)$ , čia  $\mathbf{a}_j^i$  ir  $\mathbf{b}_j^i$  yra neatsitiktiniai reikšmių vektoriai  $\mathbf{a}_j^i = ((a_j^i)^1, (a_j^i)^2, \dots, (a_j^i)^m)$  ir  $\mathbf{b}_j^i = ((b_j^i)^1, (b_j^i)^2, \dots, (b_j^i)^m)$ . Apibrėžiame reikšmių matricą  $\mathbf{x}_n = (\mathbf{x}_n^1, \mathbf{x}_n^2, \dots, \mathbf{x}_n^p)$ ,  $n = 1, 2, \dots$ , čia  $\mathbf{x}_j^i = ((x_j^i)^1, (x_j^i)^2, \dots, (x_j^i)^m)$  - reikšmių vektorius ir

$$(x_{j+1}^{i+1})^k = f((x_{j+1}^i)^k, (x_j^{i+1})^k, (a_{j+1}^{i+1})^k, (b_{j+1}^{i+1})^k), \forall k \leq m \quad (2)$$

čia

$$(x_1^i)^k = f'((a_1^i)^k, (b_1^i)^k); (x_j^1)^k = f''((a_j^1)^k, (b_j^1)^k), \quad i = 1, 2, \dots, p; \quad j = 1, 2, \dots, \\ \forall k \leq m, \text{ kai} \\ f(x), f'(x), f''(x) \text{ apibrėžia neatsitiktinių reikšmių nelineines komponentų funkcijas.}$$

## Išreikštinio modelio skaičiavimas naudojant didelio našumo skaičiavimo platformą

Išreikštinio lygiagretinimo modeliai yra grindžiami bendrintosios ir paskirstytosios atminties lygiagretinimo įrankiais, pavyzdžiui, MPI ir OpenMP. Toliau pateikiami modelių algoritmai (6.1–6.4).

---

**Algorithm 6.1** Nuoseklusis programavimo modelis

---

```
global  $p, n, m$   
  
procedure CALCULATESAMPLE( $seed, \lambda$ )  
   $s \leftarrow randomSample \leftarrow (seed, \lambda)$   
  return ( $s$ )  
  
procedure CALCULATERECURRENCE( $i, j, k$ )  
  comment: Inicializuojamas atsitiktinių dydžių generatorius: Seed, DistributionParameter  
   $a \leftarrow CalculateSample(Seed, DistributionParameter)$ ;  
   $b \leftarrow CalculateSample(Seed, DistributionParameter)$ ;  
   $(x_j^i)^k \leftarrow f((x_j^i)^k, a, b)$   
  
main  
  for  $k \leftarrow 1$  to  $m$   
     $(x_j^i)^k \leftarrow initialization$   
    do  $\left\{ \begin{array}{l} \mathbf{for } j \leftarrow 1 \mathbf{ to } n \\ \mathbf{do for } i \leftarrow 1 \mathbf{ to } p \\ \{ CALCULATERECURRENCE(i, j, k) \} \end{array} \right.$   
  output  $((x_j^i)^k)$ 
```

---

---

**Algorithm 6.2** Paskirstytosios atminties modelis

---

```
global  $p, n, m$   
  
procedure PROCESSRECURRENCE( $np, pid, seed, \lambda$ )  
  for  $k \leftarrow 1$  to  $m/np$   
     $(x_j^i)^k \leftarrow initialization$   
    do  $\left\{ \begin{array}{l} \mathbf{for } j \leftarrow 1 \mathbf{ to } n \\ \mathbf{for } i \leftarrow 1 \mathbf{ to } p \\ \mathbf{do} \left\{ \begin{array}{l} a \leftarrow randomSample \leftarrow (\lambda, seed) \\ b \leftarrow randomSample \leftarrow (\lambda, seed) \\ (x_j^i)^k \leftarrow f((x_j^i)^k, a, b) \end{array} \right. \end{array} \right.$   
  comment: gaunamas kiekvieno MPI proceso rezultatas  
   $overallResultsArray \leftarrow (x_j^i)^k$   
  if  $pid == InitialProcessNr$   
    then output ( $overallResultsArray$ )  
  
main  
   $MPI \leftarrow initMPI$   
   $Random \leftarrow initRandomGenerator$   
   $a, b, x \leftarrow initVariables$   
   $\left\{ \begin{array}{l} \mathbf{do } PROCESSRECURRENCE(NumberOfProcesses, ProcessId, Seed, StochasticsParameter) \end{array} \right.$   
   $MPI \leftarrow finalizeMPI$ 
```

---



---

**Algorithm 6.3** Bendrintosios atminties modelis

---

**global**  $p, n, m$

**main**

$a, b, x \leftarrow \text{initVariables}$

$\text{OpenMP} \leftarrow \text{initOpenMP}(\text{numOfThreads}) \leftarrow \text{privateVariables}(\text{thid}, i, j, k)$

$\text{OpenMP} \leftarrow \text{sharedVariables}(\text{overallResultsArray})$

$\text{Random} \leftarrow \text{initRandomGenerator}$

**comment:** lygiagretinimo pradžia; gijų išsišakojimas.

$$\left\{ \begin{array}{l} \mathbf{do\ for\ } k \leftarrow 1 \mathbf{ to\ } m/\text{numOfThreads} \\ \quad \left\{ \begin{array}{l} (x_j^i)^k \leftarrow \text{initialization} \\ \mathbf{for\ } j \leftarrow 1 \mathbf{ to\ } n \\ \mathbf{for\ } i \leftarrow 1 \mathbf{ to\ } p \\ \mathbf{do\ } \left\{ \begin{array}{l} a \leftarrow \text{random.Sample} \leftarrow (\lambda, \text{seed}) \\ b \leftarrow \text{random.Sample} \leftarrow (\lambda, \text{seed}) \\ (x_j^i)^k \leftarrow f((x_j^i)^k, a, b) \end{array} \right. \end{array} \right. \end{array} \right.$$

**comment:** gaunami kiekvienos gijos rezultatai

$\text{overallResultsArray} \leftarrow (x_j^i)^k$

**comment:** lygiagretinimo pabaiga

**output** ( $\text{overallResultsArray}$ )

---

---

**Algorithm 6.4** Hibridinio programavimo modelis

---

```
global  $p, n, m$ 
procedure PROCESSRECURRENCE( $np, pid, seed, \lambda$ )
   $(x_j^i)^k, a, b \leftarrow \text{initValues}$ 
  repeat
    {
      do if  $pid! = \text{startPidID}$ 
      then  $MPI \leftarrow \text{dataFromPreviousMPINode}$ 
       $\text{OpenMP} \leftarrow \text{initOpenMP}(\text{numOfThreads}) \leftarrow \text{privateVariables}(\text{thid}, i, j, k)$ 
       $\text{OpenMP} \leftarrow \text{sharedVariables}(\text{overallResultsArray})$ 
      comment: lygiagretinimo pradžia; gijų išsišakojimas.
      {
        do for  $k \leftarrow 1$  to  $m/(\text{numOfThreads} * np)$ 
        {
           $(x_j^i)^k \leftarrow \text{initialization}$ 
           $\text{Random} \leftarrow \text{initRandomGenerator}$ 
          for  $j \leftarrow 1$  to  $n$ 
          do {
            for  $i \leftarrow 1$  to  $p$ 
            {
              do {
                 $a \leftarrow \text{randomSample} \leftarrow (\lambda, \text{seed})$ 
                 $b \leftarrow \text{randomSmple} \leftarrow (\lambda, \text{seed})$ 
                 $(x_j^i)^k \leftarrow f((x_j^i)^k, a, b)$ 
              }
            }
          }
          comment: gaunami kiekvienos gijos rezultatai
           $\text{overallResultsArray} \leftarrow (x_j^i)^k$ 
          comment: lygiagretinimo pabaiga
        }
      }
    }
  until  $pid! = \text{lastPidID}$ 
  comment: MPI rezultatų paskirstymas
  if  $pid! = \text{lastPidID}$ 
  then  $\text{distributrData} \leftarrow MPI$ 
  if  $pid == \text{InitialProcessNr}$ 
  then {
    do  $\text{overallResultsArray} \leftarrow (x_j^i)^k$ 
    output ( $\text{overallResultsArray}$ )
  }
main
 $MPI \leftarrow \text{initMPI}$ 
 $\text{Random} \leftarrow \text{initRandomGenerator}$ 
 $a, b, x \leftarrow \text{initVariables}$ 
{
  do PROCESSRECURRENCE( $\text{NumberOfProcesses}, \text{ProcessId}, \text{Seed}, \text{StochasticsParameter}$ )
}
 $MPI \leftarrow \text{finalizeMPI}$ 
```

---

## 7 Eksperimentinis tyrimas: edukaciniai sprendimai

Šiame skyriuje, remiantis autoriaus išspręstomis mokslinėmis problemomis, pristatomas programinis sprendimas. Pirmiausia pateikiamas mokslinės užduoties apibūdinimas ir modeliavimo metodas. Po to pateikiami du atvejų tyrimai (praktiniai taikymai), pagrįsti aprašytais teoriniais rezultatais. Abiem atvejais reikalaujama tų pačių teorinių prielaidų: tikimybinių ir operacijų tyrimų metodų. Pirmas pavyzdys sukurtas Python programavimo kalba. Tai mokymo priemonė, kuria mokoma mokslinio tyrimo pagrindų statistinių metodų ar stochastikos pagrindų studijose. Antrajame pavyzdyje naudojami pažangesni programavimo metodai ir C programavimo kalba modeliams programuoti. Šis pavyzdys tinka moksliniams skaičiavimams mokyti mokslinės kompiuterijos studijose. Modelius teigiamai įvertino nepriklausomi recenzentai, modeliai buvo paskelbti „Scientific Programming Journal“ žurnale (cituojamas „Clarivate Analytics“ pagrindinių žurnalų sąrašė).

## Eksperimento planavimas

Skaičiavimo eksperimentas:

- SISTEMA: Eilių serijų sistema;
- PROBLEMOS FORMULAVIMAS: Eilių serijų sistemos ribinė teorema esant perkrovimo sąlygomis;
- MATEMATINIS MODELIS: Monte Karlo modeliavimas; stochastinės rekurentinės lygties skaičiavimo algoritmas;
- SKAIČIAVIMO MODELIS: Aparatinė įranga, priklausomas programinės įrangos įrankių skaičiavimo modelis;
- IMITACINIS MODELIAVIMAS: Modelio skaičiavimo eksperimentas naudojant didelio našumo skaičiavimo platformą.

Skaičiavimo modelio projektavimas:

- SKAIČIAVIMO MODELIS: Stochastinis rekurentinis modelis eilių serijų sistemai
- Antrinių modelių rinkinio skaičiavimo modelis dalijamas į submodelius:  $\{\text{SCM1}, \text{SCM2}, \text{SCM3}, \dots\}$
- MOKYMOŠI IŠTEKLIAI: Submodelių rinkinys, sudarytas iš programavimo mokymosi išteklių.

## Eilių serijų sistemos ribinės teoremos

### Apibrėžimai

Eilių serijų sistemų skaičiavimo modeliai leidžia tyrinėti eilių serijų sistemų elgesį įvairiomis (įskaitant perkrovą) sąlygomis. Pirmasis rezultatas (7.2 teorema) pateikia bendrą supratimą ir pavyzdį, kaip taikyti matematinį metodą stochastinės eilių sistemos analizei. Kitas rezultatas (7.3 teorema) yra sprendimas, kaip sukurti efektyvesnį (linijinį) skaičiavimo modelį (perkrovos sąlygomis).

Nagrinėjame  $k$ -fazių eilių serijų sistemą. Apibūdiname  $n$ -tojo kliento atvykimo laiką  $r_n^j$  į mazgą  $j$ ;  $\tau_n^j = r_{n+1}^j - r_n^j$  - tarpinis atvykimo laikas į mazgą  $j$ ,  $j \geq 1$  (jei  $j = 1$ ,  $\tau_n^1 = \tau_n$  - tarpinis atvykimo laikas į sistemą);  $s_n^j$  -  $n$ -tojo kliento fazės  $j$  aptarnavimo laikas; Tegul tarpiniai atvykimo laikai  $\{\tau_n\}$  į eilių sistemą ir kiekvienos sistemos fazės aptarnavimo laikai  $\{s_n^j\}$   $j = 1, 2, \dots, k$  yra tarpusavyje vienodai pasiskirstę atsitiktiniai dydžiai. Apibrėžiame  $E$  kaip pirmą momentą (jei egzistuoja).

Toliau apibrėžiame  $n$ -tojo kliento laukimo laiką  $j$ -toje sistemos fazėje  $v_n^j$ ; o  $w_n^j = \sum_{i=1}^j (v_n^i + s_n^i)$ ,  $j = 1, 2, \dots, k$  reiškia  $n$ -tojo kliento buvimo sistemoje laiką t.y. laiką, kuri  $n$ -tasis klientas praleidžia sistemoje iki  $j$ -osios fazės;  $t_n^j$  apibrėžia kliento  $n$  buvimo laiką mazge  $j$ , t. y.  $t_n^j = v_n^j + s_n^j$ . Vektoriai  $\hat{t}_n = \{t_n^j\}$ ,  $\hat{s}_n = \{s_n^j\}$ , ir  $\hat{v}_n = \{v_n^j\}$  apibrėžia buvimo, aptarnavimo ir laukimo laikus atitinkamai kiekviename mazge.

Nagrinėjame tokią modifikuotą eilių serijų sistemą, kai  $s_n^j = 0$ ,  $j = 1, 2, \dots, k$ ,  $n \geq k$ :

$$\delta_{j,n} = \begin{cases} s_{n-(j-1)}^j - \tau_n, & \text{if } n \geq k \\ 0, & \text{if } n < k. \end{cases}$$

Apibrėžiame:  $\alpha_j = E\delta_{j,n}$ ,  $\alpha_0 \equiv 0$ ,  $D\tau_n = \sigma_0^2$ ,  $Ds_n^j = \sigma_j^2$ ,  $\tilde{\sigma}_j^2 = \sigma_0^2 + \sigma_j^2$ ,  $s_n^0 =$

$\tau_n, j = 1, 2, \dots, k, \hat{\delta}_n = \max_{1 \leq j \leq k} \max_{0 \leq l \leq 2n} |\delta_{j,l}|, [x]$  – sveikoji skaičiaus  $x$  dalis. Laikome, kad egzistuoja konstanta  $\gamma > 0$  tokia, kad

$$\sup_{n \geq 1} E|s_n^j|^{4+\gamma} < \infty, j = 0, 1, 2, \dots, k \quad (3)$$

ir

$$\alpha_k > \alpha_{k-1} > \dots > \alpha_1 > 0. \quad (4)$$

### Eilių serijų sistemų ribinė teorema

**Teorema 7.1** (Iteruoto logaritmo kliento buvimo sistemoje laiko dėsnis [10])

*Jei sąlygos (3) ir (4) yra išpildytos, tada*

$$\mathbf{P} \left( \lim_{n \rightarrow \infty} \frac{w_n^j - \alpha_j \cdot n}{\tilde{\sigma}_j \cdot a(n)} = 1 \right) = \mathbf{P} \left( \lim_{n \rightarrow \infty} \frac{w_n^j - \alpha_j \cdot n}{\tilde{\sigma}_j \cdot a(n)} = -1 \right) = 1,$$

$j = 1, 2, \dots, k$  and  $a(n) = \sqrt{2n \ln \ln n}$ . □

### GI/G/1 eilių serijų sistemų pasiskirstymo teoremos

**Teorema 7.2** (Apie kliento buvimo laiko GI/G/1 eilių serijų sistemoje pasiskirstymo)

*Jei eilutė  $s_n^j, j \geq 0$  atitinka anksčiau pateiktas sąlygas ir sąlygą*

$$\alpha_k < \alpha_{k-1} < \dots < \alpha_1 < 0. \quad (5)$$

*tenkinama, tai GI/G/1 eilių serijų sistemai egzistuoja ribinis pasiskirstymo funkcijų vektorius ir*

$$\lim_{n \rightarrow \infty} \hat{t}_n(x) = \hat{t}(x).$$

□

**Teorema 7.3** (Kliento buvimo laiko GI/G/1 eilių serijų sistemoje pasiskirstymas esant perkrovos sąlygoms)

*Jei eilutė  $s_n^j, j \geq 0$  atitinka anksčiau pateiktas sąlygas ir sąlygą*

$$\alpha_k \geq \alpha_{k-1} \geq \dots \geq \alpha_1 \geq 0, \quad (6)$$

*tai GI/G/1 eilių serijų sistemoje ribinis pasiskirstymo funkcijų vektorius neegzistuoja ir*

$$\lim_{n \rightarrow \infty} \hat{t}_n(x) = \hat{t}(x) \equiv 0, \forall x.$$

□

### Parametrinis rekurentinis modelis

Apibūdiname  $n$ -tojo kliento atvykimo ir aptarnavimo laiką  $j$  mazge:  $t_n^j$  ir  $s_n^j$ ;  $\tau_n^j = t_n^j - t_{n-1}^j$ ;  $j = 1, 2, \dots, k$ ;  $n = 1, 2, \dots, N$ ;  $\tau_n = \tau_n^1$ ;  $\hat{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_k\}$ . Įrodyta [29] bendrosios formos rekurentinė lygtis:

$$\begin{aligned}
w_n^j &= w_n^{j-1} + s_n^j + \max(w_{n-1}^j - w_n^{j-1} - \tau_n, 0); \\
j &= 1, 2, \dots, k; n = 1, 2, \dots, N; \\
w_0^j &= 0, \forall j; w_n^0 = 0, \forall n.
\end{aligned} \tag{7}$$

□

**Tvirtinimas 7.1** (Parametrinė rekurentinio modelio išraiška skaičiuojant kliento buvimo laiką perkrovos sąlygomis)

Apibrėžiame parametrų vektorių  $\alpha$ :  $\hat{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_k\}$ . Nurodytomis sąlygomis rekurentinė lygtis (7) pateikiama šitaip:

$$w_n^j = f' * (1 - \mathbf{P}) + f'' * \mathbf{P}' \tag{8}$$

čia

$f'$  nelinijinė funkcija formos (7);

$f''$  linijinė funkcija tokia, kad:  $f''(\hat{w}, \hat{s}, \hat{\tau}) = w_n^{j-1} + s_n^j + \tau_n$ ,  
 $j = 1, 2, \dots, k; n = 1, 2, \dots, N; w_0^j = 0, \forall j; w_n^0 = 0, \forall n, 0$

$\mathbf{P}' : \hat{\alpha} \rightarrow \{0, 1\}$  yra vektoriaus  $\alpha$  predikatas (6). □

## Modelio įgyvendinimas

Norėdami įgyvendinti lygiagretinimo programavimo metodus naudojame MPI sąsają Open MPI (<http://www.open-mpi.org/faq/?category=slurm>). Kiti naudojami išteklių: gcc-4.4 4.4.5-8 GNU C programavimo kalbos kompiliatorius; gsl-bin 1.14+dfsg-1 GNU mokslinės bibliotekos (GSL) binarinis paketas; python 2.6.6-3+squeeze7 interaktyvi aukšto lygio objektinė kalba (numatytoji versija); python-numpy 1:1.4.1-5 – Numerical Python; python-matplotlib 0.99.3-1.

## VU HPC klasterio platforma ir įrankiai

Pateikiamas hibridinės platformos pavyzdys – Vilniaus universiteto HPC klasteris (<https://mif.vu.lt/cluster/>). Klasteryje veikia Debian GNU/Linux 6.0 operacinę sistemą (OS). Grupės išteklių valdymas įgyvendinamas naudojant SLURM – *The Simple Linux Utility for Resource Management* (<https://computing.llnl.gov/linux/slurm/>).

## Stochastikos pagrindų mokymas

Sistemoje pateikiami trys eilių serijų sistemų kompiuteriniai modeliai. Tikslas – suteikti studentams skaičiavimų lygiagretinimo pagrindus, supažindinti su daugiaprocesiu statistiniu modeliavimu ir imitaciniu modeliavimu.

- (1) Pirmasis – realiojo laiko modelis, tai – imitacinis modelis, kuriame naudojamas daugiaprocesis modulis „Python“.
- (2) Antrasis modelis skirtas kliento buvimo sistemoje laikui skaičiuoti ir grindžiamas stochastiniu imitaciniu modeliavimu [10]. Modelis tiesiogiai nenaudoja kelių

procesų, jis imituoja daugialypį procesą naudodamas „Python“ išvestis.

- (3) Trečiasis modelis naudoja „Python MPI mpi4py“ modulį, MPI ir Monte Karlo statistinio modeliavimo metodus [6].

Sukurtoji sistema apima pagrindinius tikimybinius ir pasiskirstymo funkcijų konceptus, eilių teorijos pagrindus ir kitus sudėtingesnius teorinius rezultatus ir metodus.

(R) Pagrindai:

- (1) atsitiktiniai skaičiai;
- (2) atsitiktinių skaičių pasiskirstymas;
- (3) atsitiktinių skaičių generatoriai;
- (4) centrinė ribinė teorema.

(P) programavimo kalbos „Python“ konceptai:

- (1) projektavimo šablonai;
- (2) koprogramos;
- (3) išvestys.

(T) Sudėtingesni rezultatai yra teoriniai faktai, tokie kaip:

- (1) kliento buvimo sistemoje laikas;
- (2) rekurentinė lygtis buvimo sistemoje laikui apskaičiuoti;
- (3) stochastiniai modeliavimo ir daugiaprocesiai metodai.

Visos šios teorinės ir programavimo struktūros leidžia studentui atlikti eksperimentus su skirtingais eilių serijų sistemų modeliais. Tokių eksperimentų tikslas yra dvejopas.

(E) Studentas turi suprasti šių žingsnių seką:

- (1) teoriniai faktai, kuriuos reikia ištirti;
- (2) konceptualusis modelis;
- (3) matematinis modelis;
- (4) algoritmai ir programavimo konstrukcijos;
- (5) skaičiavimo modelis;
- (6) stochastinis modeliavimas ir modeliavimo rezultatų stebėjimas.

Tai studentui suteiktų visą vaizdą apie bendrojo mokslinio tyrimo mastą.

(T) Studentas geriau supranta stochastinį modeliavimą ir pagrindines programavimo konstrukcijas (daugiaprocesį ir lygiagretųjį programavimą). Šios kompetencijos yra svarbios mokslinės kompiuterijos srityje.

### **Lygiagretinimo mokymas**

- Programavimo modelių rinkinys grindžiamas teorinių eilių serijų sistemos stochastinio imitacinio modeliavimo analize. Eilių serijų sistema parenkama dėl apibrėžčių paprastumo ir didelių lygiagretinimo galimybių.
- Modeliams programuoti naudojami skirtingi lygiagretinimo metodai. Tokiu būdu galima atlikti eksperimentų su skirtingais programavimo modeliais seriją, lyginti ir tirti lygiagretinimo rezultatus ir skirtingų lygiagretinimo metodų efektyvumą.
- Tokie lygiagretinimo metodai apima bendrintąją atmintį, paskirstytąją atmintį ir hibridinį lygiagretinimą. Šie metodai įdiegti naudojant MPI ir OpenMP API.

### **Mokomųjų programavimo objektų rinkinys mokant lygiagretinimo**

Mokomųjų programavimo objektų rinkinys mokant pažangiojo lygiagretinimo:

- (1) nuoseklusis programavimo modelis;
- (2) paskirstytosios atminties programavimo modelis;
- (3) paskirstytosios atminties (angl. *pipeline*) modelis;
- (4) bendrintosios atminties programavimo modelis;
- (5) bendrintosios atminties statinis (angl. *pipeline*) programavimo modelis;
- (6) bendrintosios atminties dinaminis programavimo modelis;
- (7) hibridinis programavimo modelis.

## 8 Sukurtų edukacinių sprendimų vertinimas

### Sukurtų mokymo išteklių pavyzdžių specifikacija

Išplėtotus ugdymo sprendimus – projektavimo principus ir palaikomąją taikomąją ir integravimo metodologiją – siūlome vertinti netiesioginiu metodu, t. y. įvertinti pateiktus mokymosi išteklių pavyzdžius:

- (1) stochastikos įvado mokymosi ištekliai (SLOFTIS) ir
- (2) pažangiojo lygiagretinimo mokymosi ištekliai bei mokomieji programavimo objektai (SLOFTAP).

Pateikti mokymosi ištekliai gali būti laikomi mokymo (mokymosi) medžiagos rinkiniu, įskaitant programinius mokymosi objektus ir pagalbinę metodiką, taip pat atitinkamus konstruktus, modelius ir sistemas. Toliau pateikiama išsamesnė sukurtų imties išteklių specifikacija (žr. 2 lentelę ir 3 lentelę).

**2 lentelė:** SLOFTIS parametrų aprašymas

No	Aprašymas	Įgyvendinimas
1	Konstruktai	<p>(R) Atsitiktinumai: atsitiktiniai skaičiai; atsitiktinių skaičių pasiskirstymas; atsitiktinių skaičių generatoriai; centrinė ribinė teorema.</p> <p>(P) Programavimo kalbos „Python“ konceptai: projektavimo šablonai; koprogramos; išvestys.</p> <p>(T) Sudėtingesni rezultatai, kurie apima teorinius faktus, tokius kaip: eilių sistemų specifikacijas ir parametrus, pavyzdžiui, kliento gyvenimo laiką; rekurentinę lygtį, skirtą kliento buvimo trukmei skaičiuoti; stochastinius modeliavimo metodus ir daugiaprocesius metodus.</p> <p>(I) Mokslinio tyrimo specifinės sąvokos: teoriniai faktai, kuriuos reikia iširti; konceptualusis modelis; matematinis modelis; algoritmai ir programavimo konstrukcijos; skaičiavimo modelis; stochastinis imitavimas ir modeliavimo rezultatų stebėjimas.</p>

**2 lentelė:** SLOFTIS parametrų aprašymas

No	Aprašymas	Įgyvendinimas
2	Modeliai ir mokomieji programavimo objektai:	Modeliai ir atitinkami programavimo mokymosi objektai: Stochastikos pagrindų: „Python“ vieno kauliuko modelis; „Python“ daugelio kauliukų modelis; eksponentinio pasiskirstymo Python modelis; Python daugiaprocesis modelis: Python modelis, skirtas normaliajam paskirstymui su MPI; Stochastinių pasikartojimų modelis (eilių serijų sistemoje): imitacinis modelis, grįstas daugiaprocesiniais servisais; Stochastinių pasikartojimų modelis (eilių serijų sistemoje): imitacinis modelis, grindžiamas daugiaprocesėmis paslaugomis; imitacinis modelis su „Python“ išvestimis; imitacinis modelis su „Python“ MPI.

**3 lentelė:** SLOFTAP parametrų aprašymas

No	Aprašymas	Įgyvendinimas
1	Konstruktai	(Q) Eilių serijų sistemos: eilių sistemų principai; eilių parametrai; Monte Karlo eksperimentas; (C) Lygiagretieji skaičiavimai: aparatinės įrangos platformos; programinės įrangos įrankiai; (S) Stochastika: stochastinis pasikartojimas, eilių serijų sistemų algoritminiai sprendimai; (M) Modeliavimo metodai: išilginis sistemos suskaidymas; skersinis sistemos suskaidymas;
2	Modeliai ir programavimo mokymosi objektai	Nuoseklusis programavimo modelis; paskirstytosios atminties programavimo modelis; paskirstytosios atminties komandų grandinių (angl. <i>pipeline</i> ) modelis; bendrintosios atminties programavimo modelis; bendrintosios atminties statinis komandų grandinių (angl. <i>pipeline</i> ) programavimo modelis; Bendrintos atminties dinaminio programavimo modelis; hibridinis programavimo modelis.

### Vertinimo schema

Siūlomi mokymosi išteklių susideda iš interaktyviosios mokomųjų programavimo objektų ir pagalbinių mokymo išteklių dalies. Vertinimo schema gali būti sukurta remiantis interaktyviosios dalies – programinės įrangos mokymosi objektų vertinimu ir gali būti grindžiama pateiktais bendraisiais interaktyviųjų mokymosi objektų vertinimo metodais [30], susijusiais su mokomaisiais programavimo objektais. Siūloma 4 lentelėje parodyta vertinimo schema.



**4 lentelė:** Projektavimo principų vertinimo schema

No	Pavadinimas	Aprašymai	Šaltiniai
	Įvertinti sukurtus mokymosi išteklius iš:		
1	Suderinamumas su mokymosi išteklių projektavimo ir pakartotinio naudojimo tikslais;	Sąveika; patvarumas; prieinamumas	[31]
2	Mokymosi išteklių taksonomijos	Redeker; Finlay; Churchill	[32], [33], [34]
3	Mokymosi išteklių standartai	IEEE Standartas „Mokymosi objektų metaduomenys“ (IEEE Std 1484.12.1 - 2002)	[35]
4	Kokybės vertinimo metrikos taikymas	Defude & Farhat	[36]
5	Galimi trūkumai ar ribojimai		
6	Tolesnės plėtros galimybė		

### Vertinimo rezultatai. Ekspertinis vertinimas

Vertinimas atliekamas ir pateikiamas šiame tyrimo skirsnyje: vertinimas pagal suderinamumą su mokymosi išteklių projektavimo tikslais; vertinimas, atsižvelgiant į mokymosi išteklių taksonomiją; vertinimas, atsižvelgiant į mokymosi išteklių standartus; vertinimas, atsižvelgiant į kokybės vertinimo metriką; vertinimas, atsižvelgiant į galimus trūkumus ar apribojimus; vertinimas, atsižvelgiant į tolesnės plėtros galimybę.

Ekspertų vertinimas grindžiamas aukšto lygio kompetencija ugdymo išteklių kūrimo, įgyvendinimo, taikymo ir integravimo srityje.

## 9 Išvados

Atlikus disertacijos tyrimus, padarytos šios išvados, t. y. siūlomi šie sprendimai ir jų įgyvendinimo pavyzdžiai:

- (1) Išsamiai išnagrinėtas konstruktyvistinis požiūris į mokslinės kompiuterijos ugdymą, jis pritaikytas universitetinių STEM studijų programoms, pagrindinį dėmesį skiriant tarpdisciplininiam ir mokslinių tyrimų mokymui.
- (2) Išnagrinėtas ir išplėstas mokslinės kompiuterijos TPACK modelis pritaikytas mokslinės kompiuterijos studijoms universitete. Taikant ypatybių modelius, atlikta įvairaus lygio, srities (domeno) savybių metaanalizė. Remiantis šio tyrimo schema, galima spręsti mokslinių tyrimų uždavinius.
- (3) Sukurta projektavimo, taikomosios ir integravimo metodologijos principais grindžiama mokslinės kompiuterijos mokymo projektavimo metodika, apimanti pagrindinius ugdymo sistemos komponentus: edukacines technologijas, mokomuosius projektus, didaktinius įrankius ir edukacinius metodus. Remiantis šia metodika, ypač problemų sprendimų ir mokslinių tyrimų metodais, galima pagerinti universitetinių tarpdisciplininių studijų turinį.

- (4) Remiantis praktiniais mokomaisiais ištekliais, išnagrinėtas ir įgyvendintas didaktinis mokslinės kompiuterijos metodas. Pateikiami ištekliai apima programavimo modelių rinkinį ir mokymo priemones, t. y. sukurti mokomųjų programavimo objektų pavyzdžiai. Projektavimo metodas ir atitinkami mokomieji programavimo objektai grindžiami modeliavimo paradigma, pagal kurią pritaikomi į mokslinius tyrimus orientuoti ugdymo metodai.
- (5) Pateiktas programavimo mokymo modelis, skirtas skaičiuojamiems stochastiniams reiškiniams tirti, gali būti laikomas teoriniu pagrindu kuriant atitinkamus mokymosi išteklius. Tokie mokymosi ištekliai (pateikiami kaip kompiuterio programų modeliai ir mokomieji programavimo objektai) integruojant teorinius dalykus, pavyzdžiui, stochastikos ir tikimybių pasiskirstymų pagrindus, ribines teoremas, eilių sistemas, teikia ir praktinių aparatinės ir programinės įrangos žinių.
- (6) Praktinė pasiūlytų ugdymo priemonių (konstruktyvistinio požiūrio, mokslinės kompiuterijos mokymo metodikos, tarpdisciplininio programavimo mokymo modelio) reikšmė yra imitacinių modelių kūrimo ir į mokslinius tyrimus orientuotų studijų įgyvendinimas. Šiomis priemonėmis suteikiama lygiagretinimo metodų, įskaitant didelio našumo skaičiavimo platformas ir didelių duomenų apdorojimą, praktinių žinių, taip pat daug dėmesio skiriama modeliavimui ir moksliniais tyrimais grindžiamai studentų veiklai.
- (7) Sukurtas skaičiavimo modelis skirtas eksperimentiniam kartotinio logaritmo eilių serijų sistemos perkrovimo sąlygoms dėsnio tyrimui. Parodyta, kad tam tikromis sąlygomis tiekiamas parametrinis rekurentinis sprendimas yra bent 2,6 karto veiksmingesnis, jei jis yra diegiamas procesoriams su RISC architektūra.
- (8) Sukurtas skaičiavimo modelis leidžia atlikti eilių serijų sistemos išsamų tyrimą perkrovimo sąlygomis. Šis tyrimas yra grindžiamas kompiuterio sistemos modeliavimu. Modeliavimo rezultatai yra gaunami įvertinant turimų skaičiavimų išteklių apribojimus šios sistemos ir modeliavimo eksperimento konfigūracijai: (1) M/M/1 eilių serijų sistema, su  $2^{10}$  aptarnavimo fazėmis,  $1E+7$  klientų skaičiumi,  $1E+2$  Monte Carlo bandymais; (2) M/M/1 eilių serijų sistema, su  $2^8$  aptarnavimo fazėmis,  $1E+8$  klientų skaičiumi,  $1E+2$  Monte Carlo bandymais; (3) M/M/1 eilių serijų sistema, su  $2^3$  aptarnavimo fazėmis,  $1E+10$  klientų skaičiumi,  $1E+2$  Monte Carlo bandymais; (4) M/ $\chi^2$ /1 eilių serijų sistema, su  $2^2$  aptarnavimo fazėmis,  $1E+9$  klientų skaičiumi,  $1E+2$  Monte Carlo bandymais. Simuliacijos rezultatai patvirtina teorines prielaidas 0,01 reikšmingumo lygiu.
- (9) Parengti ugdymo sprendimai ir mokymosi išteklių pavyzdžiai yra įvertinti aukšto lygio informatikos mokymo ir informatikos inžinerijos specialistų, kurių vertinimo rezultatai ir išvados buvo labai teigiami. Pasitikėjimo intervalo lygis yra 90%.

## Literatūra

- [1] V. Štuikys, *Smart Learning Objects for Smart Education in Computer Science: Theory, Methodology and Robot-Based Implementation*. Springer, 2015.

- [2] R. Burbaitė, *Advanced generative learning objects in informatics education: the concept, models, and implementation. Summary of doctoral dissertation, physical sciences, informatics (09P)*. PhD thesis, Kaunas University of Technology, 2014.
- [3] C. M. Reigeluth, *Instructional design theories and models: An overview of their current status*. Routledge, 2013.
- [4] F. Landriscina, *Simulation and learning: the role of mental models*, pp. 3072–3075 Springer, 2012.
- [5] F. Landriscina, *Simulation and Learning*. Springer, 2013.
- [6] V. Dolgopolovas, V. Dagienė, S. Minkevičius, and L. Sakalauskas, “Python for scientific computing education: Modeling of queueing systems,” *Scientific Programming*, vol. 22, no. 1, pp. 37–51, 2014. IOS Press.
- [7] V. Dolgopolovas, V. Dagienė, S. Minkevičius, and L. Sakalauskas, “Teaching scientific computing: a model-centered approach to pipeline and parallel programming with c,” *Scientific Programming*, vol. 2015, p. 18, 2015. Hindawi Publishing Corp.
- [8] V. Dolgopolovas, T. Jevsikova, and V. Dagienė, “From android games to coding in c - an approach to motivate novice engineering students to learn programming: A case study,” *Computer Applications in Engineering Education*, vol. 26, no. 1, pp. 75–90, 2018.
- [9] V. Dolgopolovas, T. Jevsikova, V. Dagienė, and L. Savulionienė, “Exploration of computational thinking of software engineering novice students based on solving computer science tasks,” *International Journal of Engineering Education*, vol. 32, no. 3, pp. 1–10, 2016.
- [10] S. Minkevičius, V. Dolgopolovas, and L. L. Sakalauskas, “A law of the iterated logarithm for the sojourn time process in queues in series,” *Methodology and Computing in Applied Probability*, vol. 18, no. 1, pp. 37–57, 2016. Springer.
- [11] A. R. Hevner, “A three cycle view of design science research,” *Scandinavian journal of information systems*, vol. 19, no. 2, p. 4, 2007.
- [12] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004. Springer.
- [13] V. Vaishnavi and W. Kuechler, “Design research in information systems,” *DSR journal*, 2004.
- [14] M. A. Helfert and B. A. Donnell An, “Practical aspects of design science,” 2012.
- [15] R. Baskerville, J. Pries-Heje, and J. Venable, “Soft design science methodology,” in *Proceedings of the 4th international conference on design science research in information systems and technology*, p. 9, ACM, 2009.
- [16] R. Botes and R. Goede, “Bridging the gap for it students: Action research and design science research as research approaches for life-long learners,” in *ISTE International Conference On Mathematics, Science Aand Technology Education*, p. 330, 2014.
- [17] T. D.-B. R. Collective, “Design-based research: An emerging paradigm for educational inquiry,” *Educational Researcher*, pp. 5–8, 2003.
- [18] S. Gregor and A. R. Hevner, “Positioning and presenting design science research for maximum impact,” *MIS quarterly*, vol. 37, no. 2, 2013.
- [19] D. Hovorka, “Design science research: A call for a pragmatic perspective,” in *Proceedings of SIGPrag Workshop, Sprouts Working Papers on Information Systems*, 2009.
- [20] P. Kotzé, A. van der Merwe, and A. Gerber, “Design science research as research approach in doctoral studies,” *AMCIS 2015 Proceedings: Design Science Research*, pp. 1–14, 2015.
- [21] T. Mettler, M. Eurich, and R. Winter, “On the use of experiments in design science research: A proposition of an evaluation framework,” *CAIS*, vol. 34, p. 10, 2014.
- [22] A. Dresch, D. P. Lacerda, and J. A. V. Antunes Jr, *Design Science Research. A Method for Science and Technology Advancement*. Design Science Research, Springer, 2015.
- [23] S. Weber, “Design science research: Paradigm or approach?,” in *AMCIS*, p. 214, 2010.
- [24] C. Sonnenberg and J. vom Brocke, “Evaluation patterns for design science research artefacts,” in *European Design Science Symposium*, pp. 71–83, Springer, 2011.
- [25] C. Sonnenberg and J. vom Brocke, “Evaluations in the science of the artificial-reconsidering

- the build-evaluate pattern in design science research,” *Design Science Research in Information Systems. Advances in Theory and Practice*, pp. 381–397, 2012.
- [26] D. B. Skillicorn, *Foundations of parallel programming*. Cambridge University Press, 2005.
- [27] A. Doucet, N. De Freitas, and N. Gordon, “An introduction to sequential monte carlo methods,” in *Sequential Monte Carlo methods in practice*, pp. 3–14, Springer, 2001.
- [28] C. P. Robert, *Monte carlo methods*. Wiley Online Library, 2004.
- [29] G. I. Ivchenko, V. Kastanov, and I. Kovalenko, *Queuing system theory*. Moscow: Vishaja Skola, 1982.
- [30] R. Burbaitė, *Išplėstiniai generatyviniai mokymosi objektai informatikos mokymuisi: koncepcija, modeliai ir realizacija*. PhD thesis, Kaunas University of Technology, 2014.
- [31] T. Kilby, “Learning objects,” *Web Based Training Information Center*, 2002.
- [32] G. H. Redeker, “An educational taxonomy for learning objects,” in *Advanced Learning Technologies, 2003. Proceedings. The 3rd IEEE International Conference on*, pp. 250–251, IEEE, 2003.
- [33] J. Finlay, “Context-neutral e-learning objects: a tale of two projects,” *The 7th HCI Educators Workshop: Effective Teaching and Training in HCI, 1st April 2004*, 2004.
- [34] D. Churchill, “Towards a useful classification of learning objects,” *Educational Technology Research and Development*, vol. 55, no. 5, pp. 479–497
- [35] L. T. S. Committee *et al.*, “Ieee standard for learning object metadata,” *IEEE Standard*, vol. 1484, no. 1, pp. 2007–04, 2002.
- [36] B. Defude and R. Farhat, “A framework to design quality-based learning objects,” in *Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference on*, pp. 23–27, IEEE, 2005.

### **Trumpos žinios apie autorių**

V. Dolgopolovas 1986 m. Vilniaus inžineriniame statybos institute (dab. Vilniaus Gedimino technikos universitetas) įgijo inžinieriaus-statybininko kvalifikacinį laipsnį, 2012 m. Vilniaus universitete – matematikos mokytojo kvalifikacija bei informatikos mokytojo profesines kompetencijas. Nuo 2013 iki 2017 metų Vilniaus universiteto Matematikos ir informatikos instituto informatikos inžinerijos doktorantas. V. Dolgopolovas yra Vilniaus universiteto Matematikos ir informatikos fakulteto Duomenų mokslo ir skaitmeninių technologijų instituto Edukacinių sistemų mokslo grupės narys.

## **Summary of Doctoral dissertation**

### **SOFTWARE LEARNING OBJECTS FOR SCIENTIFIC COMPUTING EDUCATION: TEACHING SCIENTIFIC INQUIRY WITH RECURRENCE BASED STOCHASTIC MODELS**

#### **Problem Statement**

Scientific computing (after revision of the curriculum) could become a unifying discipline for the whole university (engineering, STEM) educational system. The possible methodology could be based on such well-known or well-developed approaches like SI, model-based simulations, Design Science Research. SI (in a broader sense) should be taught: how to design scientific model-based simulations (artifacts, which could be designed and evaluated using DSR methodology) to solve specific scientific problems. The unifying teaching paradigm for interdisciplinary university education could be formulated: SI (research) should be taught by means of making model-based computer simulations in various fields by implementing Design Science Research

methodology (as a teaching method for designing models and simulations) and using seamless approach for theoretical prerequisites. The meaning of SI is universal for all disciplines. At the same time, a general definition (a broader sense) could be provided: SI is an activity of conducting (making and using) scientific model-based simulations. The next important question to answer is – why models? The following argumentation could be provided:

- (1) Theories of model-centered instruction and model-based education are well developed; [3];
- (2) Model as an artefact allows practical methods and analytic techniques of DSR and model-based system analysis to be used in research and education;
- (3) There is a strong connection between cognitive activities (mental models) of students and activities of making (developing, programming and even using predeveloped) computer models [4];
- (4) Model serve as a basis for model-based simulations.

Model-based simulations could be positioned as a basic tool within the provided methodology. There are several reasons for such an approach:

- (1) There is a connection between mental simulations and computer simulations activities [5];
- (2) In any case, simulations (of one or another type) are involved in the activity of any model development;
- (3) Simulations is a kind of generalization of models;
- (4) Simulations intersect with serious games and enable the constructionist environment and learner-centered education.

The research questions are formulated as follows:

- (1) What is the generalized methodological framework for design and implementation of the relevant teaching and learning methods for SCE, which covers such major parts of the educational system like educational technology, instructional design, didactic tools?
- (2) Focusing on general aspects of educational technology, what is the methodology for design of the university curriculum in general and Science, Technology, Engineering and Mathematics (STEM) university curriculum in particular, focusing on interdisciplinary and research-based education? How could SC the discipline be integrated into the curriculum?
- (3) Focusing on aspects of instructional design, how the teaching learning process for SCE should be designed enabling SI and constructionist based education?
- (4) Focusing on didactic aspects, how should the learning resources for SCE be designed?
- (5) What are the appropriate computational models which are based on modelling of stochastic recurrences, which enable the relevant implementation of the learning resources in the form of Software Learning Objects (SLOs) for teaching introductory stochastics, basic probability distributions, limit theorems, queuing systems, hardware and software specific parallelization methodologies?
- (6) How could educational tools which enable simulation-making and SI centered approach to a teaching and learning process, and promote practical knowledge of the relevant parallelization techniques, including hybrid computational platforms and big-data-related topics be designed? What are practical examples of implementations?

### **Research Goal, Objectives and Tasks**

- (G) The goal of the research is to develop design principles for the design of the learning resources, including a set of Software Learning Objects for the effective teaching of Scientific Inquiry based Scientific Computing educational solutions. Achieving this goal, Scientific Computing Education and the relevant Scientific Computing teaching methods are positioned to be focused on teaching SI within a Science, Technology, Engineering and Mathematics interdisciplinary curriculum. The Learning Resources (Software Learning Objects) are developed on the subjects of Introductory Stochastics and Parallelization. Making scien-

tific model-based simulations is the key to model-centered and constructionist approaches to learning. This methodology should serve as a universal platform for designing of teaching methods aiming to enhance university interdisciplinary curricula by simulation-centred and problem-solving research-based educational activities.

- (O) In achieving this goal, the research will address these objectives:
- (1) To develop the supportive methodology for practical application and integration of the relevant learning resources;
  - (2) To develop the theoretical model for parallelization of Monte Carlo experiments for stochastic recurrences. Such theoretical model provides foundations for the development of appropriate computational models;
  - (3) To develop the relevant algorithms and software solutions for experimental validation of the law of the iterated logarithm for queues in series, which are based on the previously developed computational models for parallelization of stochastic recurrence.

For the research objectives:

- (1) The supportive methodology should include a set of heuristics and feature models for teaching SI based Scientific Computing educational solutions. The process of the methodology development should be based on the Technological Pedagogical and Content Knowledge model including analysis the educational context, technological, content and pedagogical domains of the Scientific Computing Education domain and should be developed using a generalized Meta Analysis of Domain Features method, enabling to identify critical domains' features, which are relevant to the goal of the research;
  - (2) Theoretical model for parallelization provides foundations for the development of appropriate computational models. Based on this, to develop the set of computational models for explicit parallelization of Monte Carlo experiment for stochastic recurrences. Such computational models provide foundations for computational experiments to be implemented during the experimental phase of the research;
  - (3) The developed algorithms and software should provide a set of technological solutions for designing learning resources and for the implementation of sample learning resources.
- (T) Tasks of the research are to provide sample educational solutions of implementation of the developed design principles and supportive methodology in the form of a set of practical learning resources including Software Learning Objects within the scope of Scientific Computing Education. The aim of these sample Learning Resources is to present the practical examples of the Software Learning Objects for teaching SI based Scientific Computing:
- (1) teaching basics of stochastics;
  - (2) teaching parallelization.

## Research Methods

Design Science Research is used as the thesis research methodology for design and implementation of the learning resources.

‡ The problems to be solved

- (A) there is a need to specify Scientific Computing (SC) courses
  - (1) Scientific Computing (SC) courses are too technical;
  - (2) are mainly oriented in to mathematical and algorithmic foundations;
  - (3) require many prerequisites;
  - (4) hard to motivate students.
- (B) to specify SC educational technology
  - (1) there is no vision of suitable educational technology (including the lack of solutions of how to develop the content and teach SC);

(2) there is no vision on how to integrate SC to a broader curriculum.

Motivation: is based on various opinions and literature review (presented in section Existing approaches to scientific computing education). The origin of problems with the present approaches to Scientific Computing Education (SCE):

- (A) improper didactic approaches and teaching techniques to SCE;
- (B) the lack of modern and focused on inter-disciplinary and research based education approaches to SCE.

Tentative solutions

- (1) SCE should be based on teaching SI;
- (2) SI is understood (in a broader sense) as a research activity based on scientific simulations (including and computer simulations);
- (3) scientific simulations are based on computer models (artifacts);
- (4) SCE is based on teaching of how to develop scientific simulations;
- (5) simulation-based education is based on simulation-based cognitive reasoning processes;
- (6) learner-centered educational technologies like co-mediated learning should be used;
- (7) focus on DSR as a teaching technique.

Artifact: the aim of this study is to provide solutions (the design principles) of how to design and unify proper Learning Resources (LR), Knowledge Objects (KO), and Learning Objects (LO) for simulation-based SCE. The general requirements for the methodology are:

- (1) should provide a solution for SI and simulation-based SCE (as it was previously described);
- (2) should be suitable for use within a learner-centered educational environment (support co-mediated learning technology, support constructionist approach to learning).

Evaluation: is based on formal DSR evaluation requirements including author related publications [6–10] in peer-review journals which are cited in Clarivate Analytics database and positive reviews related to these publications.

### **Implemented research methodology**

The research by itself is based on well-known and well-described methodology – Design Science Research [11–23]. Although the primary application of DSR is information system design, the methodology has migrated into various fields like education technology and business management. The common feature of the described application is these are socio-technical domains. Not only technical aspects of the system but also social aspect and interactions should be considered. The relevance of the proposed methodology for the computer science educational domain is based on the next propositions:

- (P1) Has a type of a socio-technical system;
- (P2) Includes as participants (teachers, students, educational authorities, community, other stakeholders) as well as educational technology, instructional design methods, educational tools;
- (P3) Educational tools are (mainly) artifacts including cognitive artifacts in the form of software-based LO (software as a learning object, educational software);

The research methodology specifies the general approach to the research procedures, as specific methods are defined by the topic and the scope of the research. DSR is also known as methodology for doctoral research for IS and related topics [18, 20].

### **DSR methodological formalizations**

DSR methodological formalizations are based on a number of heuristics [12], which are presented below. This set of heuristics is incorporated into the design framework [11]. This approach provides us a systematic methodology for conducting a research.

1. What is the research question (design requirements)?
2. What is the artifact? How is the artifact represented?

3. What design processes (search heuristics) will be used to build the artifact?
4. How are the artifact and the design processes grounded in the knowledge base?
- 4a. What, if any, theories support the artifact design and the design process?
5. What evaluations are performed during the internal design cycles? What design improvements are identified during each design cycle?
6. How is the artifact introduced into the application environment and how is it tested? What metrics are used to demonstrate artifact utility and improvement over previous artifacts?
7. What new knowledge is added to the knowledge base and in what form (e.g., peer-reviewed literature, meta-artifacts, new theory, and new method)?
8. Has the research question been satisfactorily addressed?

Conclusions: Conclusions: Design Science Research as a methodology provides all the necessary formalizations for the implementation of the research task-design principles for development of LO for SCE. The presented design cycle provides the relevant formalizations for utilization of inductive-abductive-deductive reasoning approach for conducting the research.

### Methodological requirements and implementation

The summary of characteristics, DSR formal requirements and the relevant implementation is presented in Table 1.

**Table 1:** Formal requirements and implementations in the research

<b>Characteristics</b>	<b>DSR Requirements</b>	<b>Implementation</b>
Objectives	Develop artifacts that enable satisfactory solutions to practical problems	Develop Design Principles and the Supportive Application and Integration Methodology (DPSAIM) methodology for design of LR for SCE
Main activities	Define the problem; Suggest; Develop; Evaluate; Conclude	<p><b>Problems in SCE:</b></p> <ul style="list-style-type: none"> <li>– There is no documented LR DPSAIM for SCE</li> </ul> <p><b>Suggestion (research task):</b></p> <ul style="list-style-type: none"> <li>– To develop LR DPSAIM for SCE based on constructionist paradigm</li> <li>– To develop practical examples of LR for SCE using provided methodology</li> </ul> <p><b>Evaluation:</b></p> <ul style="list-style-type: none"> <li>– Ex-ante evaluation: comprehensive literature review, logical reasoning, assertion</li> <li>– Ex-post evaluation: demonstrations with prototypes, case studies</li> </ul> <p><b>Conclusions:</b></p> <ul style="list-style-type: none"> <li>– The LR DPSAIM for SCE is developed. The developed DPSAIM implements constructionist paradigm</li> <li>– Ex-ante and ex-post evaluations are fulfilled</li> <li>– Example of LR and case studies are presented</li> </ul>



**Table 1:** Formal requirements and implementations in the research

<b>Characteristics</b>	<b>DSR Requirements</b>	<b>Implementation</b>
Results	Artifacts (constructs, models, methods, design principles, instantiations) and improvement of theories	Constructionist LR DPSAIM for SCE; Samples of LR for SCE and case studies.
Type of knowledge	Prescriptive	How to design LR enhancing interdisciplinarily and seamless approach to SCE
Researcher's role	Builder and/or evaluator of the artefact	The scope of the research: analyse of the current state in the field of SCE; logical analyse, theoretical assertions, and confirmations of psychological, technological and instructional relevance and theoretical basics for the implemented LR DPSAIM; developing of the methodology; developing of sample LR; demonstrations via case studies
Empirical basis, Implementation	Not mandatory	Not implemented
Evaluation of results	Applications; Simulations; Experiments	Is based on Ex-ante and ex-post evaluations methodology implementing DSR requirements
Approach	Qualitative and/or quantitative	Qualitative approach is used, including theoretical analyse and literature review
Specificity	Generalizable to a certain class of problems	Is generalizable for problems of developing of interdisciplinary curriculum for inquiry-based university education

**Application domain, research question**

Application domain hierarchy consists of a set of sub related domains within the general domain of university educational system:

- (1) University educational system;
- (2) E-learning;
- (3) STEM university curriculum;
- (4) Interdisciplinary university curriculum;
- (5) Scientific computing education (discipline);
- (6) Simulation and modelling with computers;

The research question (design requirements) is DPSAIM for SLOs design, which support the relevant constructionist methodology for SCE. The DPSAIM should provide possibility of constructionist approach to interdisciplinary university education in general, and in particular, provide formalizations for SCE:

- (1) from the perspectives of educational technology;
- (2) from the perspectives of instructional design;
- (3) from the perspectives of educational content design: provide a methodology for design of LR for SCE.

**Description of the research results**

A set of formalizations for the developed DPSAIM is provided. These formalizations include implementation guidelines, recommendations, designing criteria, a set of heuristics, case studies and practical examples for SCE related domains:

- (1) Educational technology domain;
- (2) Instructional design techniques;
- (3) Development of LR (Learning Resources) and KO (Knowledge Objects);
- (4) Implementation of LOs for SCE.

From the perspectives of educational technology: the innovative methodology of designing of interdisciplinary STEM curriculum is proposed. The methodology is based on the proposition of the central place of scientific computing within university STEM curricula. This provides the basis for the design of interdisciplinary and innovations oriented learning environment.

From the perspectives of instructional design:

- (1) the revised definition of SCE is developed. The developed definition provide a basis for implementation of innovative instructional design techniques;
- (2) a methodology for structuring of the educational content is provided;
- (3) an instructional approach and innovative epistemic educational tool based on deductive and circumscriptive reasoning techniques are implemented.

From the perspectives of the content design:

- (1) the Design Principles (DP) of educational content design are provided; the DP are based on the presented approaches and requirements enabling development of the learning content providing constructionist educational environment and co-mediated learning technology;
- (2) practical examples of learning resources including software learning objects are designed.

### **Grounding by the knowledge base. Grounding by the supporting theories.**

The proposed educational solutions are grounded in a systematic literature review. Systematic grounding with supporting theories is provided as well.

### **Evaluations**

Evaluation of the research results is based on formal requirements to evaluation procedures under DSR methodology formal requirements [18, 24, 25]. The practical evaluation consists of two major parts: ex-ante evaluation and ex-post evaluation. Ex-ante evaluation is based on a systematic literature review, logical reasoning, and assertion evaluation patterns; ex-post evaluation are based on demonstrations and case studies.

### **Example of implementations**

Examples of implementation the developed methodology in the form of SLO are presented in the research.

### **Scientific Contribution of the Research**

The research provides the Design Principles and the Supportive Application and Integration Methodology (DPSAIM) for application and integration of the educational resources in the form of software learning objects for teaching scientific computing in general and teaching introductory stochastics and parallelization in particular.

- (1) The educational solutions for teaching Scientific Inquiry based Scientific computing education using Design Science Research (DSR) analytical techniques are introduced. The presented educational solutions allow developing learner-oriented educational contents, which focuses on the constructionist approach to learning, thus improving the efficiency of educational process enhancing students' Scientific Inquiry, professional knowledge, and skills.

DSR provides a relevant set of analytical techniques that enable the creation of a unifying approach to such strongly interdisciplinary in its nature field like scientific computing education. Such unifying approach, in the form of the developed Design Principles and the Supportive Application and Integration Methodology, provides a base for modernization of existing and creation of innovative university educational programs, enhancing diversity and interdisciplinarity in research and modern university education. The presented educational solutions are aimed at university STEM education that is focused on enhancing interdisciplinary and innovations in the modern university curricula.

- (2) The developed educational solutions use a Scientific Inquiry centered approach and provide a set of practical educational techniques and unifying teaching methods. Relying on Peirce pragmatism and principles of embodied cognition this study suggests a bridge from the cognitive theoretical constructions to practical educational techniques and methods, which are pragmatically useful and applicable for educational practitioners. The presented approach is based on such well theoretically grounded and practically effective solutions as the Design Science Research methodology, a model-centered approach to instructional design, model-based teaching methods, problem-solving and constructionist didactic approaches. For the purpose of this study, Scientific Inquiry centered approach is understood as an educational process of designing (developing, testing, evaluating, and improving) model-based scientific computer simulations. Computer simulations, underlying software, computational and conceptual models as cognitive artifacts allow the Design Science Research methodology to be implemented in the form of a practical teaching tool. Accordingly, an appropriate educational environment that is based on pre-designed multifaceted models and a seamless approach to theoretical prerequisites is introduced.
- (3) Several practical examples of the implementation of the developed educational solutions are provided – the Design Principles for designing learning resources for scientific computing education and supportive application and integration methodology – in the form of a set of practical learning resources. The first one covers the topic of introductory statistics – Teaching introductory stochastics and queueing models with Python. The second one covers queueing in series systems and probability topics and could be positioned within the scientific computing or programming curricula – Teaching parallelization methods with C.

### **Scientific Novelty**

- (1) Educational solutions – the Design Principles for the development of learning resources for teaching Scientific Inquiry based scientific computing education including the Supportive Application and Integration Methodology – are developed. A comprehensive analysis of the Scientific Computing educational domain from perspectives of constructionist approaches to education within STEM university curricula is provided;
- (2) Model-centered instructional design methods for teaching Scientific Inquiry and Scientific Computing are introduced;
- (3) A didactic model of introductory stochastics and parallelization using stochastic recurrence models is introduced.

### **Engineering Novelty**

- (1) An innovative computational model for explicit parallelization of stochastic recurrences is developed;
- (2) An innovative computational model for experimental confirmation of the limit theorem for the system of queues in series under heavy-traffic conditions is developed;
- (3) An innovative model-centred framework for designing of learning resources for SCE is developed;

- (4) An innovative set of learning resources in the form of SLOs for teaching introductory stochastics is developed;
- (5) An innovative set of learning resources in the form of SLOs for teaching parallelization is developed.

### **Practical Significance of the achieved Results**

- (1) The presented scientific results allow developing an innovative curriculum for SCE. Such a curriculum enhance interdisciplinarity and is focused on research-based and constructivists educational methods. The research provides a universal framework of the integral view at SCE with STEM university curricula covering educational technology, instructional design, and didactics of SCE.
- (2) The presented engineering results allow the development of practically applicable learning resources for SCE in the form of SLOs under the constructionist paradigm.

### **Defending Claims**

- (1) The developed Design Principles for design of educational resources allow design and development of the learning resources for teaching Scientific Inquiry based Scientific Computing Education including the relevant Software Learning Objects which corresponds to design requirements – model-centred education methods and the constructionist education paradigm. The developed supportive methodology is appropriate for the application and integration of the relevant learning resources within interdisciplinary university curricula.
- (2) Developed parallelization algorithms for Monte Carlo experiments for stochastic recurrence models are relevant for designing learning resources in the form of SLOs for teaching Scientific Inquiry based introductory stochastics and parallelization.

**Approbation and Publications** The results of the dissertation were presented and discussed in national and international conferences. The main results of the dissertation were published in a number of papers cited by Clarivate Analytics Master Journal List database.

### **Outline**

The dissertation consists of several major parts. “Research Context” section presents introductory topics including an overview of the research methods, research findings, and results. The research methodology is covered in details including formal requirements and implementation in the research.

“Problematic of implementation” section describes main topics covered in the research focusing on requirements of the TPACK model. This section specifies the main parts of the model including technological, content, and pedagogical domains.

Section “Specification of Context for SC TPACK model” describes context topic of TPACK model in details. This section covers SC, SCE and related topics from the point of educational technologies, teaching methods, requirements for SLO and didactic formalization combined with systematic literature review. This part covers such important topics as motivation for the research; discussion on SCE scope and definitions; discussion on educational technologies; approaches to instructional design; formalization techniques for design of SLO; discussion on didactic approaches for SCE.

Section “Meta analysis of domain features of SCE domain based on TPACK model” provides detail study of SCE domain. The focus is on PC, EC, and TC knowledge domains. The section provides a brief of practical implementation, which is based on the provided analysis.

Section “Models for teaching of parallelization” covers theoretical aspects of construction of SLO for teaching parallelization based on stochastic recurrence models. The advantage of such an

approach is that stochastic models allow the implementation of the process of multidimensional modeling during the construction of computational models and algorithms. At the same time, recurrences allow the implementation of additional dimensions for computational models under design. Two types of models are considered:

- (I) implicit models, generally implemented by categorical data types within the functional programming paradigm. For such models parallelization is considered to be an automatic process, based on software skeletons. In the research such models are not studied in detail, and this study is positioned as a topic for future research;
- (E) explicit models, based on explicit parallelization techniques. These models are studied in detail.

Section “Experimental research: educational implementations” covers the experimental part of the research. First, it studies and develops a big data computational model and algorithmic solution for experimental research of the limit behaviour of queues in series. Then the detailed results of the computational experiment are provided. Later, it describes two case studies of the implementation of learning resources for SCE and teaching parallelization, which are based on the developed computational model of queues in series. The first case study focuses on the topics of design and implementation of SLOs in an introductory SC course. The next covers the topics of design and implementation of SLOs for teaching parallelization. Practical examples of SLOs are presented in the Appendices section.

Section “Evaluation” provides Author and Expert evaluation and include: specification of the developed sample educational resources, outline of the evaluation methodology, Author evaluation of the Research results, Outline of the Expert evaluation.

## **Conclusions**

Concluding, as a result of the comprehensive study the next solutions and implementations are proposed:

- (1) the constructionist approach for Scientific Computing Education (SCE) is studied in detail and adapted for the needs of the university curriculum in general and STEM university curriculum in particular, focusing on interdisciplinary and research-based education;
- (2) the TPACK model, as related to the SCE within the university curriculum, is studied in detail and the relevant meta analysis of domain features is implemented in the form of features models of various levels. Such study provides an appropriate background for implementation of the research tasks;
- (3) the innovative Design Principles and the Supportive Application and Integration Methodology (DPSAIM) for teaching and learning SC, which covers such major parts of the educational system like educational technology, instructional design, didactic tools, and educational approaches is studied in detail and implemented. The proposed methodology allows enhancing university interdisciplinary curricula through problem-solving and research-based educational methods.
- (4) an appropriate didactic approach for SC education is studied in detail and implemented in the form of practical LR. Such LR include a set of programming models and educational tools in the form of SLOs. The design approach and the structure of the relevant SLO is based on a model-based paradigm, enabling the implementation of the SI centered educational methods;
- (5) the implemented comprehensive programming model for the computational study of stochastic recurrences provides a theoretical background for the relevant implementation of the LR. These LR (in the form of programming models and SLOs), besides their focus on such theoretical topics like introductory stochastics, basic probability distributions, limit theorems, queuing systems also provide practical knowledge of using and implementing the relevant hardware and software specific parallelization methodologies;

- (6) the practical value of using proposed educational tools in the educational practice is enabling simulation-making and SI centered approach to teaching and learning processes. These tools promote practical knowledge of the relevant parallelization techniques, including HPC computational platforms and big-data-related topics, focusing on simulation making and SI enabling practical students' activities;
- (7) the innovative computational model for experimental study of the law of the iterated logarithm for the system of queues in series under overloading conditions is developed. It is shown that under certain conditions it is possible to use at least a 2.6 times more efficient parametric recurrent solution if it is implemented on processors with RISC architecture;
- (8) the developed computational model enable to conduct a comprehensive study of the system of queues in series under overloading conditions. This study is based on the computer simulation of the system. The simulation results are obtained within the constraint on available computational resources for the following configurations of the system and modeling experiment: (1) M/M/1 system of queues in series with  $2^{10}$  servicing phases,  $1E+7$  number of customers,  $1E+2$  Monte Carlo trials; (2) M/M/1 system with  $2^8$  servicing phases,  $1E+8$  number of customers,  $1E+2$  Monte Carlo trials; (3) M/M/1 system with  $2^3$  servicing phases,  $1E+10$  number of customers,  $1E+2$  Monte Carlo trials; (4)  $M/\chi^2/1$  system with  $2^2$  servicing phases,  $1E+9$  number of customers,  $1E+2$  Monte Carlo trials. The results of the simulation confirm the theoretical assumptions with the significance level equal to 0.01;
- (9) the developed educational solutions and sample learning resources were evaluated by the high level experts in the field of informatics education and informatics engineering with a very positive outcome of the evaluation results. The level of the confidence interval is 90%.

Vladimiras Dolgopolovas

PROGRAMAVIMO MOKYMO SI OBJEKTAI MOKSLINĖS KOMPIUTERIJOS  
MOKYMU I: MOKSLINIO TYRIMO STUDIJS NAUDOJANT STOCHASTINIUS  
REKURENTINIUS MODELIUS

Daktaro disertacijos santrauka

Technologijos mokslai  
Informatikos Inžinerija (07 T)

Redaktorė Audra Ivanauskienė

Vladimiras Dolgopolovas

SOFTWARE LEARNING OBJECTS FOR SCIENTIFIC COMPUTING EDUCATION:  
TEACHING SCIENTIFIC INQUIRY WITH RECURRENCE BASED STOCHASTIC  
MODELS

Summary of Doctoral Dissertation

Technological Sciences  
Informatics Engineering (07 T)

Editor Zuzana Šiušaitė