

საქართველოს ტექნიკური უნივერსიტეტი  
GEORGIAN TECHNICAL UNIVERSITY

ISSN 1512-3979 (print)  
EISSN 1512-2174 (online)  
DOI.org/10.36073/1512-3979

**შ რ ო მ ე ბ ი**  
მართვის ავტომატიზებული სისტემები

**TRANSACTIONS**  
**AUTOMATED CONTROL SYSTEMS**

**N 2(34)**

ემდგენება საქართველოს  
ტექნიკური  
უნივერსიტეტის  
დაარსების 100  
წლისთავს  
(1922-2022)



DEDICATED  
TO THE 100th FOUNDATION  
ANNIVERSARY OF THE  
GEORGIAN TECHNICAL  
UNIVERSITY  
(1922-2022)

გამოიცემს 2006 წლიდან

პერიოდულობა:  
2 ნომერი წელიწადში

თბილისი-TBILISI-TBILJICH  
2022

საქართველოს ტექნიკური უნივერსიტეტი  
GEORGIAN TECHNICAL UNIVERSITY  
ГРУЗИНСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

---

ISSN 1512-3979 (print)  
EISSN 1512-2174 (online)  
DOI.org/10.36073/1512-3979

უ რ ო მ ე ბ ო

მართვის ავტომატიზებული სისტემები

TRANSACTIONS

AUTOMATED CONTROL SYSTEMS

Т Р У Д Ы

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

№ 2(34)

ედგნება  
საქართველოს  
ტექნიკური  
უნივერსიტეტის  
დაარსების 100  
წლისთავს  
(1922-2022)



DEDICATED  
TO THE 100th  
FOUNDATION  
ANNIVERSARY OF THE  
GEORGIAN TECHNICAL  
UNIVERSITY  
(1922-2022)

*გამოიცემა 2006 წლიდან*

*პერიოდულობა:  
2 ნომერი წელიწადში*

თბილისი-TBILISI-ТБИЛИСИ  
2022

**სარედაქციო კოლეგია:**

- აზმაიფარაშვილი ზ., ახობაძე მ., ბუაჩიძე ზ., გასიტაშვილი ზ., გიგინეიშვილი ა., გიორგაშვილი ლ., გოგიჩაიშვილი გ., გოცირიძე ი., თევდორაძე მ., იმნაიშვილი ლ., კაიშაური თ., კაპანაძე დ., კოტრიკაძე ქ., ლომინაძე ნ., ლომინაძე თ., ნატროშვილი დ., ობგაძე თ., ოთხოზორია ნ., პეტრიაშვილი ლ., სურგულაძე გ., ფრანგიშვილი ა. (თავმჯდომარე), ქართველიშვილი ი., ღურუკაია ზ., ჩხაიძე მ., ცაბაძე თ.
  - ბოსიკაშვილი ზ., თურქია ე., კაკუბავა რ., კვარაცხელია ვ., კიკნაძე მ., მელაძე ჰ., ჟვანია თ., სამხარაძე რ., სესაძე ვ., ხუციშვილი ს., მანშიაშვილი ბ., შერმაზანაშვილი ლ., შონია ო., ციციგური გ.
  - გერმანია: ბოტჰე კ., ვედეკინდი ჰ., მაიერ-ვეგენერი კ., რეისიგი ვ.
  - აშშ: ტრივედი კ. (დუკის უნივერსიტეტი), ჩიხრაძე ბ. (კორპორაცია Apple)
  - კანადა: ქაჩიბაია ვ. (IT Industry)
  - უნგრეთი: სვტრიქ ი. დებრეცენის უნივერსიტეტი
  - რუსეთი: ბაბაიანი რ. (მპი), ვასინი ა.(მსუ), შჩუკინი ბ.(მიფი), ფომინი ბ. (პეტერბურგის ტუ)
- pasuxismgebeli redactori: g. surgulaZe. statiebi: <http://www.gtu.ge/Journals/mas/>

**EDITORIAL BOARD:**

- Akhobadze M., Azmaiparashvili Z., Buachidze Z., Chkhaidze M., Gasitashvili Z., Gigineishvili A., Giorgaschvili L., Gogichaishvili G., Goziridze I., Ghurtskaia Z., Imnaishvili L., Kaishauri T., Kapanadze D., Kotrikadze K., Lominadze N., Lominadze T., Natroshvili D., Obgadze T., Prangishvili A. (Chairman), Surguladze G., Tevdoradze M., Tsabadze T.
  - Bosikashvili Z., Dzidziguri G., Kakubava R., Kiknadze M., Kvaratskhelia V., Meladze G., Samkharadze R., Shermazanashvili L., Sesadze V., Shanshiashvili B., Shonia O., Tsveraidze Z., Tsintsadze A., Zhvania T.
  - Germany: Bothe K.(Humboldt univ. Berlin), Meyer-Wegener K. (Erlangen univ.), Reisig W. (Humboldt univ.Berlin), Wedekind H.(Erlangen univ.)
  - USA: Trivedi K. (Duke University), Chikhradze B. (Apple Co.)
  - Canada: Kachibaia V. (IT Industry)
  - Hungary: Sztrik J. (University of Debrecen)
  - Russia: Babaian R.(IPU), Tshukin B.(Mephi), Vasin A.(MSU), Fomin B.(St-Petersburg, Techn.Univ.)
- Executive Editor: G. Surguladze. References: <http://www.gtu.ge/Journals/mas/>

**РЕДАКЦИОННАЯ КОЛЛЕГИЯ:**

- Азмаипарашвили З., Ахобадзе М., Буачидзе З., Гаситашвили З., Гигинеишвили А., Гиоргашвили Л., Гогичаишвили Г., Гоциридзе И., Гурцкая З., Имнаишвили Л., Каишаური Т., Картвелишвили И., Капанაძე დ., კიკნაძე მ., კოტრიკაძე კ., ლომინაძე ნ., ლომინაძე თ., ნატროშვილი დ., ობგაძე თ., პრანგიშვილი ა. (председатель), სურგულაძე გ., ტევდორაძე მ., ჩხაიძე მ.
  - Босикашвили З., Дзидзигури Г., Какубава Р., Кикнадзе М., Кварацхелия В., Меладзе Г., Самхарадзе Р., Сесадзе В., Туркия Е., Шаншиашвили Б., Шермазанашвили Л., Шония О., Цвераидзе З., Цинцадзе А.
  - Германия: Ботэ К., Рейсиг В. (Гумболдт унив. Берлин), Ведекинд Х., Меиер-Вегенер К. (Ерланген унив.)
  - США: Триведи К. (Университет Дюке), Чихрадзе Б. (Apple корпорация)
  - Канада: Качибая В. (IT Industry)
  - Россия: Бабаян Р. (ИПУ), Васин А. (МГУ), Щукин Б. (МИФИ), Фомин Б. (ЛЭТИ.С-т Петербург)
- Ответственный редактор: Г. Сургуладзе. Статьи: <http://www.gtu.ge/Journals/mas/>

ISSN 1512-3979, DOI.org/10.36073/1512-3979

© გამომცემლობა „ტექნიკური უნივერსიტეტი“, 2022  
Publishing House `Technical University`, 2022  
Издательство `Технический Университет`, 2022

**შინაარსი - CONTENTS – СОДЕРЖАНИЕ**

†	აკადემიკოსი გოჩა ჩოგოვაძე // Academician Gocha Chogovadze // Академик Гоча Чоговадзе (1941-2022)	5
➤	საქართველოს ტექნიკური უნივერსიტეტი 100 წლისაა – ზოგიერთი ისტორიული ინოვაციური სამეცნიერო შედეგები „მართვის ავტომატიზებული სისტემების (პროგრამული ინჟინერიის)“ დეპარტამენტში. გია სურგულაძე, ლილი პეტრიაშვილი, მაკა ცერცვაძე // Georgian Technical University Celebrates the 100th Anniversary – some historical innovative scientific results in the Department of "Automated Control Systems (Software Engineering)". Surguladze Gia, Petriashvili Lili, Tsertsvadze Maka // Грузинский Технический Университет отмечает 100-летие – некоторые исторические инновационные научные результаты кафедры «Автоматизированные системы управления (программная инженерия)». Сургуладзе Г., Петриашвили Л., Церцвадзе М.	7
	<b><u>გამოყენებითი ინფორმატიკა – APPLIED INFORMATICS – ПРИКЛАДНАЯ ИНФОРМАТИКА</u></b>	
➤	სტუ-100: „ხელოვნება-წარმოება-მეცნიერება“ (ისტორიული ფრაგმენტი) – პირველი ქართული ERP სისტემა და განვითარების ტენდენციები. გია სურგულაძე, გულბაათ ნარეშელაშვილი, კონსტანტინე თაგაური // GTU-100: "Art-Production-Science" (Historical Fragment) – the First Georgian ERP System and Development Trends. Surguladze Gia, Nareshelashvili Gulbaat, Tagauri Konstantine // ГТУ-100: „Искусство-Производство-Наука“ (Исторический фрагмент) - первая грузинская ERP система и тенденции развития. Сургуладзе Г., Нарешелашвили Г., Тагаური К.	23
➤	მართვის სისტემის მოდელირების თავისებურება დინამიკურად ცვალებადი საპრობლემო გარემოსათვის. თეიმურაზ სუხიაშვილი // Features of Modeling Control Systems for a Dynamicly Changing Problem Environment. Sukhiashvili Teimuraz // Особенности моделирования систем управления для динамически меняющей проблемной среды. Сухиашвили Т.	31
➤	ERP და SCADA სისტემების ინტეგრაცია საწარმოო ინდუსტრიაში და მათი ეფექტურობის ანალიზი. თამარ ხუჭუა // Integration of ERP and SCADA systems into industrial production and their analysis efficiency. Khuchua Tamar // Интеграция ERP и SCADA систем в производство и анализ их эффективности. Хучуа Т.	35
➤	RPA პროცესების რობოტიზაციის და ხელოვნური ინტელექტის ტექნოლოგიის ინსტრუმენტები. ირმა ბერძენიშვილი // RPA-Process Robotization and Artificial Intelligence Technology Tools. Berdzenishvili Irma // Роботизации RPA-процессов и инструменты технологии искусственного интеллекта. Бердзенишвили И..	41
➤	მანქანური სწავლების გამოყენება თანამედროვე ციფრულ ტექნოლოგიებში. ნინო თოფურია, თინათინ ხატიაშვილი // Application of Machine Learning in modern Digital Technologies. Topuria Nino, Khatiaшvili Tinatin //Применение машинного обучения в современных цифровых технологиях. Топурия Н., Хатиашвили Т..	46
➤	მიკროსერვისული არქიტექტურის აგება AZURE - ის პლატფორმაზე. თინათინ ხატიაშვილი // Building a microservice architecture on the AZURE platform. Khatiaшvili Tinatin // Построение микросервисной архитектуры на платформе AZURE. Хатиашвили Т..	51
➤	მართვის ინფორმაციული სისტემების აგება სუფთა არქიტექტურით ASP.NET Core გარემოში. გია სურგულაძე, ხატია ხატიაშვილი // Construction of management information systems with a clean architecture. Surguladze Gia, Khatiaшvili Khatia // Построение информационных систем управления с чистой архитектурой. Сургуладзе Г., Хатиашвили Х.	55
➤	Agile პროგრამირება და ტესტირება სუფთა არქიტექტურით ASP.NET Core გარემოში. ხატია ხატიაშვილი // Agile Programming and Testing with a Clean Architecture in an ASP.NET Core Environment. Khatiaшvili Khatia // Гибкое программирование и тестирование с чистой архитектурой в среде ASP.NET Core. Хатиашвили Х..	63

**ტექნიკური ინფორმატიკა – COMPUTER ENGINEERING – ТЕХНИЧЕСКАЯ ИНФОРМАТИКА**

- ფრინველების სათავსოს ტემპერატურისა და განათების ავტომატიზაცია. მარინა ქოზაშვილი, ზაალ აზმაიფარაშვილი, გურამ მურჯიკნელი, გ. მურჯიკნელი, იური მოდებაძე // Automation of the Temperature and Lighting of the Poultry House. Kozashvili M., Azmaiparashvili Z., Murjikneli Guram, Murjikneli G., Modbadze Iuri // Автоматизация температуры и освещения в помещени птичника. Козашвили М., Азмаипарашвили З., Мурджикнели Г., Мурджикнели Г. Модбадзе Ю. 70
- შინაური ფრინველების სათავსოს ავტომატური მომარაგება საკვებით. მარინა ქოზაშვილი // Automatic supply of food to the house of domestic birds. Marina Kozashvili // Автоматическое снабжение пищей помещения домашних птиц. Козашвили М. 74

**ფიზიკა – PHYSICS – ФИЗИКА**

- მორსატრევი თვითმტვირთავი აგრეგატის განივი დინამიკური მდგრადობის გამოკვლევა ვაკუუმში. ნინო ბჟალავა, მალხაზ ბიბილური, ნანა მამისაშვილი // Cross Section of a Self-Loading Tractor Dynamic Stability Study. Bzhalava Nino, Bibiluri Malkhaz, Mamisashvili Nana // Исследование динамической устойчивости самозагрузочного трактора поперечного сечения. Бжалава Н., Бибилури М., Мамисашвили Н. 77

**პრაქტიკული ინფორმატიკა – PRACTICAL INFORMATICS – ПРАКТИЧЕСКАЯ ИНФОРМАТИКА**

- იონმეტრული სენსორები და მათი გამოყენების თავისებურებანი. თამარ ქოზაშვილი, ზაალ აზმაიფარაშვილი, ვლადიმერ ფადიურაშვილი, ოლღა მელიქიძე // Ionometric Sensors and Features of their Use. Kozashvili Tamar, Azmaparashvili Zaal, Phadiurashvili Vladimer, Melikidze Olga 81
- ვებ-დიზაინის რამდენიმე თანამედროვე ტენდენცია. ცაცა ნამჩევაძე, თამარ აბულაძე // Some Modern Tendencies of Web-Design. Namchevadze Tsatsa, Abuladze Tamar // Некоторые современные тенденции Web-дизайна. Намчевадзе Ц., Абуладзе Т. 88
- API-ს ტიპები, Web-სერვისები და ინფორმაციის გაცვლის სისტემები. OpenAPI -ს პეციფიკაცია. ლელა პაპავა, სოფიკო გოგოლაძე // API Types, Web Services and Information Exchange Systems. OpenAPI Specification. Papava Lela, Gogoladze Sopiko // Типы API, Web-сервисы и системы обмена информацией. спецификация OpenAPI. ПапаваЛ. Гоголадзе С. 92
- საზომ მოწყობილობათა ცდომილებების გათვალისწინებით არასრული კუბური მოდელის სამკომპონენტო სიმპლექს-გისოსურ გეგმებში ექსპერიმენტის წერტილების გენერირების ალგორითმი. ნინო ბერაია, ლალი ტოკაძე // Algorithm for Generating Experimental Points for an Incomplete Cubic Model of Three-Component Simplex-lattice Designs, Subject to Errors of Measurement. Beraia Nino, Tokadze Lali // Алгоритм генерации экспериментальных точек в трехкомпонентных симплексно-решетчатых планах неполной кубической модели с учетом погрешностей измерительных устройств. Берая Н.О., Токадзе Л.Ш. 96
- ჩატბოტების ტექნოლოგიების გამოყენება ბიზნეს გარემოში. ნინო გრიგალაშვილი // Chatbots Usage in Business Sector. Grigalashvili Nino // Использование чат-ботов в бизнес-секторе. Григалашвили Н. 101
- ინფორმაციული სისტემების სწრაფქმედების ამაღლება Redis-Cache მეთოდის გამოყენებით .Net Core პლატფორმაზე. კონსტანტინე თაგაური // Raising the Information Systems Performance Using the Redis-Cache Method on the .Net Core Platform. Konstantine Tagauri // Повышение производительности информационных систем с использованием кеширования «redis-cache» на платформе .NET Core. Тагаури К. 104



**აკადემიკოსი გოჩა ჩოგოვაძე (1941 – 2022)**

საქართველოს ტექნიკური უნივერსიტეტის რეფორმატორი რექტორი (1988-1994), იუნესკოს (პარიზი) კულტურისა და განათლების დეპარტამენტის დირექტორი (1981-1988), დიდი საზოგადო და სახელმწიფო მოღვაწე, საქართველოს იუნესკოს საქმეთა ეროვნული კომისიის ვიცე-პრეზიდენტი (2004-2012), ევროპის და რუსეთის მრავალი უნივერსიტეტის საპატიო დოქტორი და საერთაშორისო აკადემიების წევრი, დიპლომატი, საქართველოს საგანგებო და სრულუფლებიანი ელჩი საფრანგეთსა და ესპანეთის სამეფოში (1994-2004), სტუ-ს „მართვის ავტომატიზებული სისტემების“ (1971) და სტუ-ს UNESCO-ს „ინფორმაციული საზოგადოების“ (2003) კათედრების დამაარსებელი, მათი პირველი გამგე. საქართველოს მეცნიერებათა ეროვნული აკადემიის ნამდვილი წევრი (1994-დან), ტექნიკის მეცნიერებათა დოქტორი (1975) - საპატიო პროფესორი - გოჩა გიორგის ძე ჩოგოვაძე - მრავალი სამეცნიერო წიგნის, პროექტისა და სტატიის ავტორი, მრავალი სტუდენტის აღმზრდელი და ახალგაზრდა მეცნიერის „სამეცნიერო ნათლია“, ქართული და უცხოური ლიტერატურის, პოეზიის, მუსიკისა და სიმღერის დიდი მოყვარული, საერთაშორისო მეგობრობის დიდოსტატი, სუფრის ორიგინალური თამადა, მოსიყვარულე და გულისხმიერი მეუღლე, მამა და ბაბუა, მუდამ სიკეთის მთესველი „ქართველი ფაუსტი“.

<https://gtu.ge/News/19748/>

[https://gtu.ge/Journals/mas/Referat/ASU-2022\(1\\_33\)/ASU-2022\(1\\_33\).pdf](https://gtu.ge/Journals/mas/Referat/ASU-2022(1_33)/ASU-2022(1_33).pdf) (გვ. 7-24)

### **Academician Gocha Chogovadze (1941 – 1922)**

Reformer Rector of the Georgian Technical University (1988-1994), Director of the Department of Culture and Education of UNESCO (Paris) (1981-1988), a great public figure and statesman, Vice-President of the National Commission for UNESCO Affairs of Georgia (2004-2012), Honorary Doctorate at multiple European and Russian Universities and member of international academies, diplomat, Ambassador Extraordinary and Plenipotentiary of Georgia to France and the Kingdom of Spain (1994-2004), GTU "Automated Control Systems" (1971) and GTU UNESCO "Information Society" (2003) the founder of the departments, their first governor. Full member of the National Academy of Sciences of Georgia (since 1994), Doctor of Technical Sciences (1975) - Honorary Professor - Gocha Chogovadze - author of many scientific books, projects and articles, educator of many students and "scientific godfather" of young scientists, A great lover of Georgian and foreign literature, poetry, music and song, a great master of international friendship, a loving and caring husband, father and grandfather, always a sower of kindness "*georgian Faust*".

### **Академик Гоча Чоговадзе (1941 – 1922)**

Ректор-реформатор Грузинского Технического Университета (1988-1994), директор департамента Культуры и Образования ЮНЕСКО (Париж 1981-1988), общественный и государственный деятель, вице-президент Национальной комиссии по делам ЮНЕСКО Грузии (2004-2012), почетный доктор и действительный член международных академий многих европейских и российских университетов, дипломат, Чрезвычайный и Полномочный Посол Грузии во Франции и Королевстве Испания (1994-2004), Основатель кафедр «Автоматизированные системы управления» (1971) и ЮНЕСКО «Информационное общество» (2003) в Грузинском Техническом Университете, Действительный член Национальной Академии Наук Грузии (с 1994), доктор технических наук (1975) и почетный профессор Информатики - *Гоча Георгиевич Чоговадзе* - автор многих научных книг, проектов и статей, воспитатель многих студентов и молодых ученых, их руководитель и „научный крестный отец“, большой любитель и знаток грузинской и зарубежной литературы, поэзии, музыки и песни, гроссмейстер интернациональных дружеских отношений, оригинальный руководитель („тамада“) стола, любящий и заботливый муж, отец и дедушка, всегда сеятель добра „*грузинский Фауст*“

<https://www.rah.ru/events/detail.php?ID=58471>

# სტუ 100 წლისაა – ზოგიერთი ისტორიული ინოვაციური სამეცნიერო შედეგი „მართვის ავტომატიზებული სისტემების“ კათედრაზე

გია სურგულაძე, ლილი პეტრიაშვილი, მაკა ცერცვაძე

საქართველოს ტექნიკური უნივერსიტეტი  
g.surguladze@gtu.ge, l.petriashvili@gtu.ge, maka@gtu.ge

## რეზიუმე

განხილულია „პროგრამული ინჟინერიის“ (მართვის ავტომატიზებული სისტემების) დეპარტამენტის ხელმძღვანელის, ტექნიკის მეცნიერებათა დოქტორის, პროფესორ გია სურგულაძის საქართველოს ტექნიკურ უნივერსიტეტში (პოლიტექნიკურ ინსტიტუტში) 50-წლიანი მოღვაწეობის (1972-2022) ფონზე ინფორმატიკის (კომპიუტინგის) სფეროში საგანმანათლებლო და სამეცნიერო მოღვაწეობის ზოგიერთი ინოვაციური შედეგი. წარმოდგენილია ზოგიერთი ის კონკრეტული სამეცნიერო შედეგი, რომელიც მისი ინიციატივით დამუშავდა, დაინერგა და განვითარდა 1975-1995 წლებში (საქართველოსა და მის ფარგლებს გარეთ - ისტორიული თვალსაზრისით) და დღესაც აქტიურად გამოიყენება. ესაა მონაცემთა რელაციური ბაზის ოპტიმალური სტრუქტურის ავტომატიზებული პროექტირების ალგორითმების კომპლექსის შექმნა ხელოვნური ინტელექტის მეთოდებით, მან გამოიყენება ჰპოვა ინტელექტუალურ მონაცემთა ბანკებში (ცოდნის ბაზების სახით). მნიშვნელოვანია ასევე 1983-1993 წლებში ავტომატიზებული სამუშაო ადგილების ქსელის შექმნა (პირველი ქართული ERP-ტიპის სისტემა). იგი რეალიზებულია ექსპერტული სისტემის სახით. მინსკის ფრეიმებისა და პეტრის ქსელების გრაფო-ანალიზური ინსტრუმენტით. შედეგები რეგისტრირებულია როგორც პოსტ-საკავშირო ალგორითმებისა და პროგრამების ფონდში, ასევე ევროპის მათემატიკური საზოგადოების European Mathematical Society - EMS) ორგანიზაციაში. სტუ-ში დაცულია 40-ზე მეტი დისერტაცია: მონაცემთა საცავების, ERP-სისტემების, რელაციური და NoSQL ბაზების, ლოგისტიკის მენეჯმენტის, შავი ზღვის ეკოლოგიის, ინტერდისციპლინური სწავლების ვირტუალიზაციის, Windows-, Web- და Mobile- ინტეგრირებული სისტემების დაპროექტების და სხვ. თემებზე.

**საკვანძო სიტყვები:** ინოვაცია. ინფორმაციული საზოგადოება. მართვის ავტომატიზებული სისტემები. ERP. რელაციური ბაზა. NoSQL ბაზა. ბაზის სტრუქტურა. ოპტიმიზაციის ალგორითმი. ხელოვნური ინტელექტი. ექსპერტული სისტემა. პეტრის ქსელი.

## 1. შესავალი:

### ისტორიული ინოვაციები

*ინოვაციები ისტორიულ ფონზე:* წინამდებარე სტატიაში წარმოდგენილია პროგრამული ინჟინერიის (მართვის ავტომატიზებული სისტემების) დეპარტამენტის არსებობის 50 წლის მანძილზე (1972-2022) გ. სურგულაძის (სტუდენტი, უმც.მ.თ., ასპირანტი, დოცენტი, პროფესორი) მიერ სამეცნიერო პროექტების, საზღვარგარეთის საგრანტო სტიპენდიების, საბიუჯეტო მეცნიერული თემების, ალგორითმების და პროგრამების საერთაშორისო და საქართველოს ფონდების საავტორო უფლებების, სტუ-ს და სხვა უნივერსიტეტების სასწავლო პროცესების და დოქტორანტების ხელმძღვანელობის მიმართულებებით შესრულებული სამუშაოების მოკლე



მიმოხილვა – სტუ-ის პროფესორის აკადემიური სტატუსით. მათი შინაარსი და ინოვაციური შედეგები ასახულია გ. სურგულაძის ავტორობით და თანაავტორობით გამოქვეყნებულ 400-ზე მეტ პუბლიკაციაში [1]. სტატიების, მონოგრაფიების და სახელმძღვანელოების ნახვა შესაძლებელია სტუ-ის და პარლამენტის (საჯარო) ბიბლიოთეკებში, 80 ელ-წიგნი მოთავსებულია სტუ-ს ვებგვერდზე: [https://gtu.ge/Learning/ElBooks/ims\\_books.php](https://gtu.ge/Learning/ElBooks/ims_books.php), სტატიები კი ვებ საიტებზე <https://gtu.ge/Journals/mas/> (სტუ), საქართველოს მეცნიერებათა აკადემიის, WASET-ის, კემბრიჯის და ევროპის სხვა უნივერსიტეტების გამოცემებში [2-5]. ინოვაციების მოკლე ჩამონათვალი:

1) **პირველად:** საქართველოში და პოსტსაბჭოური ქვეყნების უნივერსიტეტებში 1976 წ. საკანდიდატო დისერტაციის სახით შემუშავდა მონაცემთა რელაციური ბაზების სტრუქტურების ავტომატიზებული აგების ალგორითმული სქემები და პროგრამები (მაგდებურგი-გერმანია, მოსკოვი, პეტერბურგი). აკადემიური კურსი (საგანი) მონაცემთა ბაზების აგების საფუძვლები დაინერგა სპი-ის აგტ ფაკულტეტის სასწავლო პროცესში. 1988 წ. გამოიცა პირველი მონოგრაფია ამ სფეროში (გ. ჩოგოვაძე, გ. სურგულაძე, ვ. ქაჩიბაია);

2) საქართველოს მეცნიერების კომიტეტის სახალხო მეურნეობის მართვის ინსტიტუტში ხელმძღვანელ მუშაკთა კვალიფიკაციის ამაღლების მიზნით შევიმუშავეთ სისტემური ანალიზის მეთოდების გამოყენების ახალი მეთოდი გადაწყვეტილების მიღების პროცესში – (**პირველად**) ექსპერტულ შეფასებათა მეთოდების ავტომატიზაციის (პროგრამული პაკეტის შექმნის) საფუძველზე, პროდუქციის მწარმოებელი ფირმების მეცნიერული კონსულტირების მიზნით;

3) 1985-1990 წლებში „წარმოებისა და მეცნიერების“ დაახლოების მიზნით შემუშავდა ორგანიზაციული მართვის (მენეჯმენტის) პროცესების ავტომატიზებული სისტემა (რეალურ ობიექტებზე: მსუბუქი მრეწველობის ფაბრიკები, სოფლის მეურნეობის სამინისტროს ობიექტები, სავაჭრო-კომერციული სამმართველოები, ცენტრები და ა.შ.). პრაქტიკული შედეგების თეორიული განზოგადების საფუძველზე, **პირველად** შეიქმნა ავტომატიზებული სამუშაო ადგილების კომპიუტერული ქსელის დაპროექტების და აგების ტექნოლოგია, მეცნიერებათა დოქტორის ხარისხის დისერტაცია (სტუ, 04.1993, გ. სურგულაძე). ეს იყო პირველი ქართული ERP სისტემის ანალოგი;

4) **პირველად:** 1992 წ. სპი-ს მართვის ავტომატიზებული სისტემების კათედრის სასწავლო პროცესში დავიწყეთ დაპროგრამების სტრუქტურული C-ენის, ხოლო 1995 წლიდან ობიექტორიენტირებული C++ ენის ლექციების კითხვა. 1997 წ. გამოიცა პირველი ქართული წიგნი „დაპროგრამება C/C++-ენებზე“ (სტრუქტურული და ობიექტ-ორიენტირებული მეთოდები);

5) **პირველად:** 1996 წ. შემუშავდა და გამოიცა პირველი ქართული სახელმძღვანელო „მონაცემთა და ცოდნის ბაზების აგების საფუძვლები“. მასში აისახა მონაცემთა განაწილებული რელაციური ბაზების დაპროექტების ტექნოლოგია სისტემათა სტრუქტურების ოპტიმიზაციისა და პროცესების იმიტაციური მოდელირებით პეტრის ქსელების ბაზაზე (დაინერგა სტუ-ის სასწავლო პროცესში);

6) **პირველად:** 2001 წ. გერმანიის DAAD-ის გრანტის საფუძველზე მას-კათედრაზე შემუშავდა კონცეფცია „პროგრამული ინჟინერია უნიფიცირებული მოდელირების ენის (UML, Paradigm+) საფუძველზე, Linux (SUSE) და Windows ოს-ების პლატფორმაზე. 2003 წ. გამოიცა პირველი ქართული სახელმძღვანელო (გერმანულ პროფესორთან ერთად). ჩვენს მიერ სტუ-ში პირველად ჩატარდა ლექციები და „Linux/g++“-ლაბორატორიები *მას* კათედრის ჯგუფებთან (ფრანგულ-ქართული უნივერსიტეტის კომპიუტერულ ცენტრში - 40 PC სამუშაო ადგილით);

7) **მას კათედრაზე** შემუშავდა ბიზნეს-პროცესების (სამუშაო ნაკადების) მოდელირების და მართვის სისტემა (Workflowmanagement System). 2003 წ. სასწავლო პროცესში დაინერგა CASE და BI ტექნოლოგიების სწავლება (ე. თურქია). მონაცემთა საცავების (Data warehouse, OLAP, DataMining) აგების საფუძვლები (ლ. პეტრიაშვილი), 2003-2008 წწ. გამოიცა პირველი ქართულენოვანი ორი მონოგრაფია;

8) 2010-2015 წწ. **შემუშავდა** მართვის საინფორმაციო სისტემების პროგრამული აპლიკაციების ინტეგრაციის, ტესტირების (ვალიდაცია-ვერიფიკაციის), პროგრამულ-სერვერული ვირტუალიზაციის ეფექტური მეთოდები, MsBizTalk, Hadoop, MongoDB, BigData, Machine Learning: Linux და Pyhon საფუძველზე. დაცულია ხუთი დისერტაცია, გამოქვეყნებულია 4 მონოგრაფია;

9) **პირველად:** მას კათედრაზე 2011-2013 წწ. შემუშავდა ელექტრონული საარჩევნო სისტემის პროექტი მონაცემთა ობიექტ-რელაციური, მულტიმედიური ბაზების საფუძველზე. 2014 წ. გამოიცა პირველი ქართული მონოგრაფია (გერმანელ პროფესორთან, კლაუს მეიერ-ვეგენერთამ ერთად);

10) **2008-2017 წლებში** მას (პროგრამული ინჟინერიის) დეპარტამენტში, ჩვენ მიერ შეიქმნა ახალგაზრდა მეცნიერთა (დოქტორანტების) ჯგუფი: *ბიზნესის, მარკეტინგის და ლოგისტიკის მენეჯმენტის* პროცესების ავტომატიზაციის მიზნით. გამოიცა 4 მონოგრაფია ამ მიმართულებით;

11) **პირველად:** 2015 წ. **მას** (პროგრამული ინჟინერიის) დეპარტამენტში შემუშავდა მართვის საინფორმაციო სისტემების *დაპროგრამების კიბრიდული ტექნოლოგიები*, დაინერგა სასწავლო პროცესში-2015 წ. გამოიცა პირველი 1001 გვერდიანი ქართული მონოგრაფია (გ.ჩოგოვაძე, ა.ფრანგიშვილი, გ.სურგულაძე) 2017 წ.;

12) **პირველად:** 2015-2018 წწ. **მას** (პროგრამული ინჟინერიის) დეპარტამენტში შემუშავდა *შავი ზღვის ეკოლოგიური მონიტორინგის სისტემის* აგების კონცეფცია და პროგრამული სისტემა (UML, ORM, SharePoint, SQLServer). გამოიცა ქართული მონოგრაფია;

13) **პირველად:** 2020 წ. მას (პროგრამული ინჟინერიის) დეპარტამენტში ჩვენ მიერ შემუშავდა *მობილური აპლიკაციების დეველოპმენტის საფუძვლები (Java, Android)*. გამოიცა ქართული მონოგრაფია (სურგულაძე, კაკაშვილი, მარტიაშვილი - დოქტორანტები)

14) **პირველად:** 2020-2021 წ. დეპარტამენტში შემუშავდა სწავლების სამივე საფეხურისთვის „პროგრამული ინჟინერიის“ კონცენტრაციის დამამთავრებელი ნაშრომების (პროექტების) მეთოდური სახელმძღვანელოები. გამოიცა შესაბამისი დამხმარე სახელმძღვანელოები. სტუ-ს ვებ-გვერდზე ([https://gtu.ge/Learning/ElBooks/Doctoral\\_Library.php](https://gtu.ge/Learning/ElBooks/Doctoral_Library.php)) შეიქმნა „დოქტორანტის ბიბლიოთეკა“, სადაც განთავსდა ელ-მეთოდური მასალა და დაცული დისერტაციების საფუძველზე შექმნილი მონოგრაფიები;

15) **პირველად:** 2020-2021 წ. სტუ-ს იუნესკოს კათედრასა და პროგრამული ინჟინერიის დეპარტამენტში შემუშავდა „კოვიდ-19“ პანდემიის გათვალისწინებით „ინფორმაციის დიდაქტიკის“ ახალი კონცეფცია, რომელიც ორიენტირებულია *ინფორმაციული საზოგადოების ფორმირების* პრობლემის გადაწყვეტის მხარდასაჭერად. შეიქმნა ბაკალავრებისთვის სპეც კურსი (პროფ.ვ. დიდმანიძე), დამუშავდა სამეცნიერო პროექტი *„ინფორმაციული საზოგადოება და ინტერდისციპლინური სწავლება ციფრული ტექნოლოგიების ბაზაზე“*, გამოიცა მონოგრაფია (ჩოგოვაძე გ., სურგულაძე გ., თოფურია ნ., ხარიტონაშვილი მ.);



ნახ.1-ა. პროგ. გია სურგულაძის და მისი გუნდის 100 წიგნი (1985-2022)



ნახ.1-ბ. პროფ. გია სურგულაძის და მისი გუნდის 100 წიგნი (დასასრულა)

16) 2018 და 2021 წლებში სტუ-ში ჩვენი ფაკულტეტის მიერ ჩატარდა 2 საერთაშორისო კონფერენცია „საინფორმაციო საზოგადოება და განათლების ინტენსიფიკაციის ტექნოლოგიები“. 2023 წელს იგეგმება მე-3 კონფერენციის ჩატარება, რომელიც მიემდგინება სტუ-ს „იუნესკოს“ კათედრის დაარსების მე-20 წლისთავს და აკადემიკოს გოჩა ჩოგოვადის როლს „ინფორმატიკის დიდაქტიკისა და ინფორმაციული საზოგადოების ფორმირების“ სფეროში:

17) 2021/22 წწ. დეპარტამენტში შემუშავდა „პროგრამული ინჟინერიის“ სამეცნიერო, პედაგოგიური და მეთოდური მიმართულებით სამუშაო გეგმა ინოვაციური საკითხების კვლევისა და განვითარებისათვის. 1-ელ ნახაზზე წარმოდგენილია „მას – (პროგრამული ინჟინერიის) დეპარტამენტში პროფ. გ. სურგულაძის ინიციატივით და ხელმძღვანელობით (1972-2022 წლებში) შექმნილი სტუ-ს 100 წლისთავისადმი მიძღვნილი 100 წიგნი (კომპიუტინგის სფეროში).

## 2. ძირითადი ნაწილი

წინამდებარე სტატიის მიზანია დღევანდელი გადასახედიდან ისტორიული კუთხით შეაფასოს ის ორი მნიშვნელოვანი ინოვაციური შედეგი (წრმოდგენილი 16-დან), რომლებიც 1976 და 1990 წლებში იქნა შემოთავაზებული სტუს ასპირანტის და შემდეგ დოქტორანტის (გ. სურგულაძის) მიერ, კერძოდ მონაცემთა რელაციური ბაზის ოპტიმალური სტრუქტურის განსაზღვრისა და ინფორმაციული ინტელექტუალური ბანკის ასაგებად. აგრეთვე მენეჯმენტის ინფორმაციული სისტემების (დღევანდელი ტერმინოლოგიით – ERP სისტემის ტიპის) დასაპროექტებლად და ასაგებად (კომპლექსური ავტომატიზაცია პროდუქციის მწარმოებელი ფირმებისთვის).

ორივე მათგანში გამოყენებული იყო ხელოვნური ინტელექტის მეთოდოლოგია. კერძოდ, დასმული ამოცანების გადაწყვეტა: 1) რელაციური სტრუქტურების განსაზღვრის მიზნით ჯგუფთა თეორიის საფუძველზე (მონაცემთა ბაზებისათვის) [6]; 2) სემანტიკური ქსელების და სიმულაციური მოდელების აგებისა და მათი ანალიზის ალგორითმული სტრუქტურების შესამუშავებლად ექსპერტული სისტემისათვის მინსკის ფრეიმებისა და პეტრის სტოქასტური ქსელების საფუძველზე (ცოდნის ბაზებისათვის) [7].

### 2.1. მონაცემთა რელაციური ბაზის ოპტიმალური სტრუქტურის დაპროექტება

მათემატიკოს ედგარ კოდის მიერ 1970 წ. პირველად გაჟღერდა ტერმინი მონაცემთა რელაციური მოდელისა და ბაზების შესახებ. 72-74 წლებში გამოქვეყნდა მისი რამდენიმე სტატია, რომლებმაც დასაბამი მისცა ამ მიმართულების სწრაფად და ფართოდ განვითარებას მსოფლიოში. ე. კოდს 1981 წ. ამერიკის კომპიუტინგის ასოციაციის მიერ მიენიჭა ალან ტიურინგის პრემია. მან წარმოადგინა მონაცემთა რელაციური ბაზის სტრუქტურის დაპროექტების მეთოდი დამოკიდებულებათა ნორმალური ფორმებით (კოდის რელაციური ალგებრა) [8]. 1975 წლიდან დაიწყო რელაციური ბაზების თეორიული და პრაქტიკული საკითხების განვითარება სხვადასხვა ქვეყნების უნივერსიტეტებში. კარძოდ, ფრანგი კ. დელობელის (გრენობლი), გერმანელი ჰ. ვედეკინდის (დარმშტადტი) და ა.შ. ამერიკელების - კ. ვონგისა და ფ. ბერნშტაინის მიერ დამუშავდა მონაცემთა ბაზების ფუნქციონალურ დამოკიდებულებათა სინთეზის მეთოდი, რომელთა საფუძველს ბულის ალგებრისა და დამოკიდებულებათა თეორიების იზომორფიზმი შეადგენდა. მსგავსი ამოცანები გამოკვლეულ იქნა აგრეთვე ვ. ბრუნდოს, ლ. კალინიჩენკოს, მ. ცალენკოს, ჩოგოვადის, გ. სურგულაძის და ვ. ქაჩიბაიას (სტუ) ნაშრომებში [9-11].

ჩვენი, ქართველ ავტორთა შედეგების ორიგინალურობა მდგომარეობდა იმაში, რომ სხვა მეცნიერებისაგან განსხვავებით მონაცემთა რელაციური ბაზების ლოგიკური სტრუქტურების დაპროექტების ამოცანის გადასაწყვეტად შემუშავებულ იქნა ფუნქციონალურ დამოკიდებულებათა (ფდ) სიმრავლის კომპლექსური ანალიზის მეთოდი და შესაბამისი პროცედურების ავტომატიზებულ რეჟიმში განხორციელების ინსტრუმენტული საშუალებანი.

თუ ხელოვნური ინტელექტის ენაზე ვიტყვით, აგებულ იქნა მონაცემთა რელაციური ბაზის ოპტიმალური ლოგიკური სტრუქტურის განსაზღვრის „ჭკვიანი“ ალგორითმული სქემები. მათი რეალიზაცია განხორციელდა *Prolog-ის* (ლოგიკური პროგრამირების) და *Fortran-ის* (მათემატიკური პროგრამირების) ენებზე. ქვემოთ მოცემული გვაქვს აღნიშნული ამოცანის მათემატიკური მოდელი.

მონაცემთა რელაციური ბაზის ლოგიკური სტრუქტურის, როგორც კვლევის ობიექტის მოდელი წარმოდგენილია ოთხეულით [12]:

$$M_{rel} = \langle F^0(\bar{A}), F^k(\bar{A}), \tilde{L}, \tilde{Q} \rangle \quad (1)$$

სადაც  $\bar{A}$  საპრობლემო სფეროს ატრიბუტების დასახელებათა სიმრავლეა;  $(\{A_i\}, i = a, r)$ ;

- $F^0(\bar{A})$  - ფუნქციონალურ დამოკიდებულებათა საწყისი სიმრავლე;
- $F^k(\bar{A})$  - მონაცემთა ბაზის ოპტიმალური ლოგიკური სტრუქტურა;
- $\tilde{L}$  - ლოგიკურ-ალგებრული გარდაქმნის ოპერაციები  $F^0(\bar{A})$ -სი  $F^k(\bar{A})$  -ში;
- $\tilde{Q}$  - გარდაქმნის შეზღუდვები, რომლებიც დგება ფუნქციონალურ დამოკიდებულებათა საწყისი სიმრავლის აღწერისა და ლოგიკური გამოყვანის წესების, ნორმალურ ფორმათა თეორიის კანონებისა და გარდაქმნის დროის მინიმუმის გათვალისწინებით.

ფუნქციონალურ დამოკიდებულებათა საწყისი სიმრავლე გამოირჩევა შემდეგი ფორმით:

$$\langle A_i \rangle \rightarrow \{A_j\}, \text{ სადაც } i, j \in \gamma; \gamma = \overline{1, r}; i \neq j. \quad (2)$$

საწყისი  $F^0(\bar{A})$  მოდელის მიზნობრივ  $F^k(\bar{A})$  მოდელში გადასაყვანად საჭიროა შემდეგი შეზღუდვების ( $\tilde{Q}$ ) გათვალისწინება:

- $F^0(\bar{A})$  საწყისი სიმრავლის ფუნქციონალურ დამოკიდებულებათა აღწერის წესები;
- $F^k(\bar{A})$  მიზნობრივი სიმრავლისათვის დასაშვები ფუნქციონალური დამოკიდებულებების მისაღებად გამოყვანის წესების გამოყენების შესაძლებლობა (ტრანზიტულობის, პროექციულობის, ადიტიურობის, ფსევდოტრანზიტულობის, შთანთქმის და ა.შ.);
- $F^0(\bar{A})$  სიმრავლის გადაყვანის დროს  $F^k(\bar{A})$  სიმრავლეში უნდა იყოს შეძლებისდაგვარად მინიმალური  $t_{min} = f(\tilde{L}, m, n)$ .

ამგვარად, ისმება ამოცანა ზოგადი ლოგიკურ-ალგებრული ოპერაციების ( $\tilde{L}$ ) და მათი ეფექტური შესრულების მიმდევრობის ერთობლიობის დასადგენად, რომელიც უზრუნველყოფს სასრული ბიჯების რაოდენობით მიზნობრივი  $F^k(\bar{A})$  ფუნქციონალურ დამოკიდებულებათა სიმრავლის ფორმირებას ზემოაღნიშნული თვისებებით.

$F^0(\bar{A})$  სიმრავლის  $F^k(\bar{A})$ -ში გარდაქმნის ყოველ ბიჯზე გარანტირებული უნდა იყოს მონაცემთა სისრულე და არაწინააღმდეგობრიობა, უნდა იქნას დაცული მათი მთლიანობის ასპექტი.

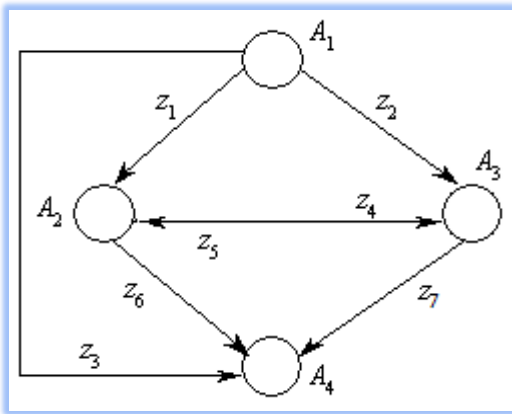
მონაცემთა ბინარული სტრუქტურებისათვის განიხილება ფუნქციონალურ დამოკიდებულებათა სიმრავლის ოპტიმალური სიმძლავრის (m) განსაზღვრის ამოცანა. ვუწოდოთ მას ვერტიკალური ოპტიმიზაცია.

n-არული სტრუქტურებისათვის ვერტიკალურთან ერთად განიხილება ჰორიზონტალური ოპტიმიზაციაც, რომელიც მდგომარეობს მონაცემთა ბაზის ფუნქციონალურ დამოკიდებულებათა (თითოეული სქემისათვის) n-ის ოპტიმალური მნიშვნელობის განსაზღვრაში. აქვე უნდა აღინიშნოს, რომ ოპტიმიზაციები ვერტიკალურად და ჰორიზონტალურად n-არულ სტრუქტურებში ხორციელდება კომპლექსურად. ყოველი ფუნქციონალური დამოკიდებულება  $F^0(\bar{A})$  სიმრავლის  $F^k(\bar{A})$ -ში გარდაქმნის პროცესში მოდიფიცირდება ან ამოიშლება სიმრავლიდან მხოლოდ მისი სხვა დამოკიდებულებებთან განსაზღვრული ალგორითმით ანალიზის ჩატარების შემდეგ. ეს უკანასნელი კი გარანტიაა მონაცემთა მთლიანობისა და არა-წინააღმდეგობრიობის დაცვისათვის.

ბინარული ფდ-ები შეიძლება წარმოდგენილ იქნას მატრიცების, გრაფების ან ერთეულოვანი რადიუსის მქონე ფაქტორ-სიმრავლეებით, ხოლო n-არული ფდ-ები ყოველთვის შეიძლება დაყვანილ იქნას ბინარულამდე და პირიქით - ციპფვის კანონი (Zipf's law) [6, 13].

მე-2 ნახაზზე მოცემულია ფდ-სიმრავლის წარმოდგენის საილუსტრაციო მაგალითი ორიენტირებული გრაფის საშუალებით, სადაც  $M = \{A_1, A_2, A_3, A_4\}$  - ატრიბუტთა სიმრავლეა, ხოლო

$Z_i, i = 1..7$  - ატრიბუტებს შორის ფდ-ებია.



$$F^0(\bar{A}) = \{ A_1 \rightarrow A_2, \\ A_1 \rightarrow A_3, \\ A_1 \rightarrow A_4, \\ A_2 \rightarrow A_3, \\ A_2 \rightarrow A_4, \\ A_3 \rightarrow A_2, \\ A_3 \rightarrow A_4, \\ A_4 \rightarrow A_3 \}$$

$$F^k(\bar{A}) = \{ A_1 \rightarrow A_2, \\ A_1 \rightarrow A_3, \\ A_2 \rightarrow A_3, \\ A_3 \rightarrow A_2, \\ A_2 \rightarrow A_4, \\ \}$$

შედეგი: ვარიანტი-1

ნახ. 2. ატრიბუტთა ფდ-ები გრაფის სახით

მარტივად რომ ვთქვათ, წარმოდგენილი მაგალითის საფუძველზე, ამოცანა მდგომარეობს საწყისი „ჭარბი“ ფდ-ებთან სიმრავლიდან ფდ-ებთან ისეთი მინიმალური ქვესიმრავლის (პროექციის) განსაზღვრაში, საიდანაც შესაძლებელი იქნება უდანაკარგოდ საწყისი სიმრავლის აღდგენა.

ასეთ ამოცანას წინა ეტაპზე უსწრებს ერთი პრობლემა, როგორცაა თვით ფდ-ებთან საწყისი სიმრავლის დადგენა კონკრეტული საპრობლემო სფეროსათვის. ანუ როგორ განისაზღვროს ატრიბუტთა შორის ფდ-ებთან სახეები, როგორცაა: ფუნქციონალური, სრული-ფუნქციონალური, ტრანზიტული და ა.შ. [12].

საპრობლემო სფეროს კონცეპტუალური სქემის (სტრუქტურის) საფუძველზე სემანტიკური (ინფორმაციული) კავშირების ასახვა ხდება რელაციებში (ცხრილებში - Tables). ცხრილი შეიძლება იყოს არანორმალიზებული და მისი დაყვანა ნორმალიზებულ ფორმაზე (მაგ., 1-ნფ) არაა რთული. ჩვენი ინოვაციური მიდგომით ასეთი პრობლემის მოგვარება დაევალა „პროგრამა რობოტს“, რომელსაც ჯგუფთა თეორიის პრინციპის საფუძველზე (მონაცემთა ბაზის სტრუქტურისათვის) ცხრილის ავტომატური სინტაქსური ანალიზით (ადამიანის გარეშე) უნდა უნდა წარმოედგინა საპრობლემო სფეროს საწყის ფდ-ებათა სიმრავლის სემანტიკური შედეგები, ანუ განესაზღვრა ცხრილში გასაღებური ატრიბუტის (ან შედეგნილი გასაღების) მნიშვნელობა. განვიხილოთ მოკლედ ასეთი რობოტი პროგრამის აგების ძირითადი მომენტები და შედეგები.

დავუშვათ, მოცემულია ცხრილი (რელაცია) ატრიბუტთა განსაზღვრულ სიმრავლეზე

$$U^R = \{A_1, A_2, \dots, A_n\} \subseteq U.$$

შემდგომი კვლევის პროცესების განსახორციელებლად რელაციაში შემოვიტანოთ ფიქტიური ატრიბუტი  $G$ , რომლის მნიშვნელობათა დომენიც შეესაბამება ნატურალურ რიცხვთა მწკრივს. იგი უნიკალურად განსაზღვრავს (გასაღების მსგავსად) ამ ცხრილის კორტეჟს (სტრიქონს). ობიექტ-ორიენტირებული პროგრამირების თვალსაზრისით რელაცია - კლასია, ხოლო კორტეჟი - ობიექტი.

თითოეულ კორტეჟს შეესაბამება ინდექსის მნიშვნელობა, ამ უკანასკნელის ადგილ-მდებარეობის მიხედვით ნორმალიზებულ ცხრილში. ამგვარად, გვაქვს  $R(U^R, G)$  (ნახ.3).

ცხრილში ყოველი  $A_i$  სვეტი შეიძლება განხილულ იქნას ისე, როგორც  $R[A, G]$  პროექცია, ანუ

$$R[A, G] = \{(a_i, g)\}, g \in G$$

სადაც  $(a_i, g)$  - წყვილთა სიმრავლეა.

იგი ბინარული რელაციაა  $F_i \subseteq G \times A_i$

$A_1$	$A_2$	...	$A_n$	$G$

$A_i$	$G$
$a_{i1}$	1
$a_{i2}$	2
...	...
$a_{im}$	m
...	

**ნახ.3. ფიქტიური ატრიბუტი - G**

მე-4 ნახაზზე განხილულია აღნიშნული მსჯელობის კონკრეტული მაგალითი.

$A_i$	$G$
დაპროგრამების საფუძვლები	1
მონაცემთა ბაზების დაპროექტება	2
დაპროგრამების საფუძვლები	3
მონაცემთა ბაზების დაპროექტება	4
პროგრამული ინჟინერიის საფუძვლები	5
დაპროგრამების საფუძვლები	6
ვებ-პროგრამირების ტექნოლოგია	7
მობილური პროგრამირების საფუძვლები	8
პროგრამული ინჟინერიის საფუძვლები	9
პარალელური დაპროგრამების თეორია	10

$A_i$	$R_G [A]$
დაპროგრამების საფუძვლები	{1,3,6}
მონაცემთა ბაზების დაპროექტება	{2,4}
პროგრამული ინჟინერიის საფუძვლები	{5,9}
ვებ-პროგრამირების ტექნოლოგია	7
მობილური პროგრამირების საფუძვლები	8
პარალელური დაპროგრამების თეორია	10

**ნახ.4. საილუსტრაციო მაგალითი**



ბინარული რელაციის (მარცხენა) ფრაგმენტი წარმოდგენილია ფაქტორ-სიმრავლის (მარჯვენა) ცხრილის სახით, რომელშიც  $A_i$  ატრიბუტის ყოველ მნიშვნელობას შეესაბამება ფიქტიური  $G$  ატრიბუტის მნიშვნელობათა სიმრავლე (ჯგუფი). ნაწილობრივ მოწესრიგებული სისტემების თეორიაში იგი ცნობილია როგორც „ერთეულოვანი რადიუსის არე“ [13].

ჩვენი მიზანია მანქანურ პროგრამას ვასწავლოთ ასეთი ფუნქციის შესრულება რელაციაში ახალი კორტეჟების დამატების გათვალისწინებით. ვინაიდან პროგრამა ვერ „ხედავს“ 1-ელ ნახაზზე გამოსახულ გრაფს, მისთვის საჭირო იქნება ამ გრაფის შესაბამისი ინციდენციის მატრიცის მიწოდება ლოგიკური პროგრამის შესასვლელზე, შემდეგ კი კორტეჟების ანალიზის განხორციელება. Prolog-ის ენაზე ესაა აქსიომების, ჰიპოტეზების და ხორნის დიზიუნქტების ერთობლიობა [ლიტ].

ალგორითმმა რომ აღმოაჩინოს ატრიბუტთა შორის ფუნქციონალური დამოკიდებულების არსებობა, აუცილებელია შემდეგი ლოგიკური პირობის შემოწმება:

$$A_i \perp A_{i+\lambda}; \perp = \begin{cases} " \rightarrow " , & \text{if } (k_{A_i} > k_{A_{i+\lambda}}) \ \& \ (F_i^G \supset F_{i+\lambda}^G) \\ " \leftarrow " , & \text{if } (k_{A_i} < k_{A_{i+\lambda}}) \ \& \ (F_i^G \subset F_{i+\lambda}^G) \\ " \leftrightarrow " , & \text{if } (k_{A_i} = k_{A_{i+\lambda}}) \ \& \ (F_i^G = F_{i+\lambda}^G) \end{cases} \quad (3)$$

$k_{A_i}$  და  $k_{A_{i+\lambda}}$  არის, შესაბამისად,  $A_i$  და  $A_{i+\lambda}$  ატრიბუტების ფარდობითი განაწილების კოეფიციენტები. ისინი ახასიათებს რელაციაში ატრიბუტის განსხვავებულ მნიშვნელობათა რაოდენობის ფარდობას ამ ატრიბუტის მნიშვნელობათა საერთო რიცხვთან. მაგალითად, მე-3 ნახაზის ცხრილებიდან ესაა  $6/10 = 0,6$ .

დავუშვათ, მოცემულია ატრიბუტთა სიმრავლე  $\{A_1, A_2, \dots, A_n\}$ . იგი უნდა შეესაბამებოდეს ფაქტორ-სიმრავლეთა ერთობლიობას:

$$\bigcup_{j=1}^m F_{i_j}^G = F_{i_1}^G \cup F_{i_2}^G \cup \dots \cup F_{i_m}^G$$

მაშინ, გამოსახულება  $(A_1, A_2, \dots, A_n) \rightarrow A_s$  მართებულია თუ

$$\left\{ \forall k \exists p : \left( \bigcup_{j=1}^m F_{i_j}^G \right)_k \subseteq (F_s^G)_p \right\}$$

ამგვარად, ორ ატრიბუტს შორის ფუნქციონალური დამოკიდებულების არსებობის დასადგენად, აუცილებელია მე-(3) პირობის შესრულების სამართლიანობა.

საპრობლემო სფეროს ბიზნეს-პროცესების კანონზომიერებათა ანალიზის საფუძველზე მიღებული რელაციათა (ფუნქციონალურ დამოკიდებულებათა ჩათვლით) ასეთი საწყისი სიმრავლე შემდეგ უნდა გადამუშავდეს სპეციალური ალგორითმების დახმარებით, მაგალითად, „წარმოსახვით ვექტორთა“ მეთოდი, ან „შეკუმშული მატრიცის“ მეთოდი და ა.შ. [12, 14].

ასეთი ალგორითმები აქტუალური გახდა დღესაც, რადგან NoSQL - იერერქიულ ტიპის ბაზებში ნორმალიზაცია – დენორმალიზაციის პრობლემა გაჩნდა. ანუ, რელაციათა დენორმალიზაცია NoSQL ბაზებში იძლევა უკეთეს (ოპტიმალურ) შედეგებს „მოთხოვნა – პასუხის“ დროის შემცირების თვალსაზრისით [14].

უნდა აღინიშნოს, რომ რელაციური მოდელები (და ბაზები) მიეკუთვნება სტატიკური ტიპის რელაციურ სქემებს, მათში ნაკლები სემანტიკაა. დამოკიდებულებათა ანალიზი ხდება ატრიბუტთა მნიშვნელობების სინტაქსური დამუშავების საფუძველზე. თუმცა, რელაციურ მონაცემთა ბაზებში გასაღებური ატრიბუტების, ფუნქციონალური და სხვა ტიპის დამოკიდებულებათა შემოტანამ, მონაცემთა ლოგიკური მთლიანობისა და აზრობრივად არასაწინააღმდეგო ინფორმაციის არსებობის შეზღუდვების კონტროლის შესაძლებლობით, გარკვეული ნაბიჯი გადაიდგა ბაზების ინტელექტუალიზაციისკენ. პარალელურად განვითარება დაიწყო სემანტიკური მონაცემთა ბაზების თეორიამ, კერძოდ, „სემანტიკური ნორმალური ფორმების“ დამუშავების მეთოდებმა [15,16].

## 2.2. ცოდნის ბაზის დაპროექტება ექსპერტული სისტემისთვის

ავტომატიზებული სამუშაო ადგილები განაწილებულია ორგანიზაციის (კორპორაციის) სტრუქტურის შესაბამისად, ფუნქციური განყოფილებების სპეციალისტებისთვის. ისინი გამოიყენებენ ლოგიკურად ერთიან მონაცემთა ბაზას, რომელიც ფიზიკურად შეიძლება დანაწევრებული იყოს ლოკალური პერსონალური კომპიუტერების ჰომოგენურ ან ჰეტეროგენულ ქსელში (დასაშვებია მონაცემთა გარკვეული დუბლირება სისტემის სწრაფქმედებისა და საიმედოობის ასამაღლებლად).

ქსელის მრავალმომხმარებლური რეჟიმი ასეთ დროს გამოიყენებს საერთო რესურსებს, როგორცაა სერვერების მეხსიერება და პროგრამული უზრუნველყოფა, მონაცემთა და ცოდნის ბაზები, ტექნიკური უზრუნველყოფა და ა.შ. ამგვარად, აუცილებელი ხდება სპეციალური პროგრამული საშუალებების შექმნა როგორც მართვის ორგანიზაციის, ასევე განაწილებული მონაცემთა და ცოდნის ლოკალური ბაზების მთლიანობის ასპექტების დასაცავად [12].

პირველი მათგანი წყდება ქსელის სისტემური პროგრამული უზრუნველყოფის საშუალებით სხვადასხვა დონის შეთანხმების ოქმების (Protocols) გამოყენებით [17]. მეორის გადაწყვეტა კი სცილდება მის კომპეტენციას. ეს განპირობებულია იმით, რომ მონაცემთა და ცოდნის ბაზების აგებისა და მათი მთლიანობის დაცვის საკითხები დაკავშირებულია ინფორმაციის სემანტიკურ (ძირითადად), სტრუქტურულ და პრაგმატულ ასპექტებთან.

საკვლევი სისტემის მოდელი, როგორც საპრობლემო სფეროს ადექვატურად ასახვის საშუალება გამოთვლითი სისტემის მეხსიერებაში, შეიძლება წარმოვადგინოთ ექვსეულით

$$S_m = \langle \bar{R}, \bar{F}, \bar{Y}, \bar{L}, \bar{Z}, \bar{Q} \rangle \quad (4)$$

სადაც

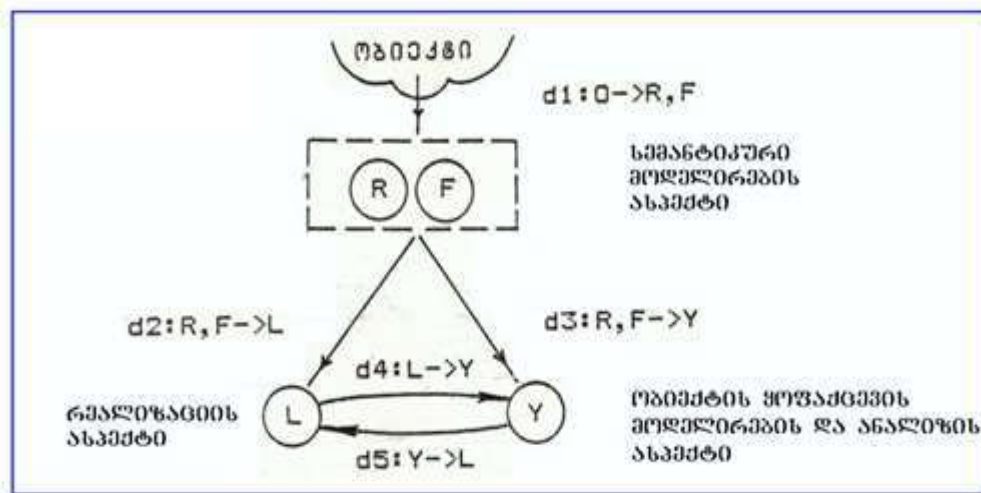
- $\bar{R}$  დეკლარაციული სახის დამოკიდებულებათა სიმრავლე (  $\{R_i\}, i = \overline{1, r}$  ), რომელიც ფორმირდება რელაციურ დამოკიდებულებათა თეორიის საფუძველზე;
- $\bar{F}$  არის პროცედურული სახის რელაციატა სიმრავლე (  $\{F_j\}, j = \overline{1, f}$  ), რომელთა სტრუქტურები აიგება ფრეიმების, გამოყვანის წესებისა და რელაციური ალგენრის თეორიების საფუძველზე;

–  $\bar{Y}$  არის მოდელის მმართველ სტრუქტურათა სიმრავლე ( $\{Y_\gamma\}, \gamma = \overline{1, y}$ ), რომელთა წარმოდგენა ხორციელდება პეტრის ქსელების გამოყენებითი თეორიის საფუძველზე.  $\bar{Y}$  მართავს დეკლარაციული ( $\bar{R}$ ) და პროცედურული ( $\bar{F}$ ) რელაციების დამუშავების თანამიმდევრობას მრავალმომხმარებლური რეჟიმის დროს გარკვეული მიზნების ( $\bar{Z}$ ) მისაღწევად. მიზნებად განიხილება მომხმარებელთა მოთხოვნები (ინფორმაციის მიღება განაწილებული ბაზებიდან, მონაცემებისა და გამოყენებითი პროგრამების [როგორც ცოდნის] განახლების შესრულება და ა.შ.), ხოლო მათი მიღწევის ძირითად კრიტერიუმებად ( $\bar{Q}$ ) კი - პასუხების (კორექციების) სისრულე, სისწორე, დასაშვები დრო და გამოყენებული რესურსების შესაძლოდ მცირე დანახარჯები. სისტემის მიზნები და კროტერიუმები კონკრეტდება პროექტირებისა და ექსპლუატაციის ეტაპების მიხედვით.

ამგვარად, (4) მოდელის სამეული  $\langle \bar{R}, \bar{F}, \bar{Y} \rangle$  შინაარსობრივად უახლოვდება ცოდნის ბაზების კონცეფციას ექსპერტულ სისტემებში [დის-40]. ამ თვალსაზრისით, განაწილებული სისტემების დაპროექტების შემოთავაზებული მეთოდოლოგია შეიძლება განხილულ იქნას როგორც „ცოდნის კონსტრუირების“ ინჟინრის (ექსპერტ-დამპროექტების) ინსტრუმენტი.

–  $\bar{L}$  არის მოდელის რეალიზაციის საშუალება (დაპროგრამების ენა). ჩვენი სისტემა რეალიზებულ იყო Nantucket ფირმის კლასიკური რელაციური ტიპის მონაცემთა ბაზის მართვის სისტემაზე Clipper. იგი კომპილატორია და ქმნის .exe ფაილებს, რაც მნიშვნელოვნად სწრაფი და მოქნილია მოდულების შექმნისა და შესრულების ეტაპებზე, ვიდრე ამ დროისთვის ცნობილი dBase, FoxPro, Paradox და სხვა ბაზები;

მე-5 ნახაზზე მოცემულია განხილული კომპონენტების ურთიერთკავშირის სქემა საკვლევი სისტემის მოდელის სახით, სადაც კარგად ჩანს როგორც მათი ასახვის დონეები, ასევე სისტემის აგების ეტაპების ასპექტები (პროგრამული სისტემის სასიცოცხლო ციკლის შესაბამისად).



ნახ.5. საკვლევი სისტემის მოდელი

$d_1$  – შეესაბამება საგნობრივი სფეროს არაფორმალიზებული ცოდნის (შინაარსის) ასახვის ინსტრუმენტს ფორმალიზებულში (სემანტიკური მოდელირების ასპექტი);

$d_2$  – დეკლარაციულ და პროცედურულ დამოკიდებულებათა ასახვას დაპროგრამების ენის სისტემაში (პროგრამული რეალიზაციის ასპექტი);

$d_3$  – დეკლარაციულ და პროცედურულ დამოკიდებულებათა ასახვას პეტრის ქსელების საშუალებით (ობიექტის ქცევის მოდელირებისა და ანალიზის ასპექტი);

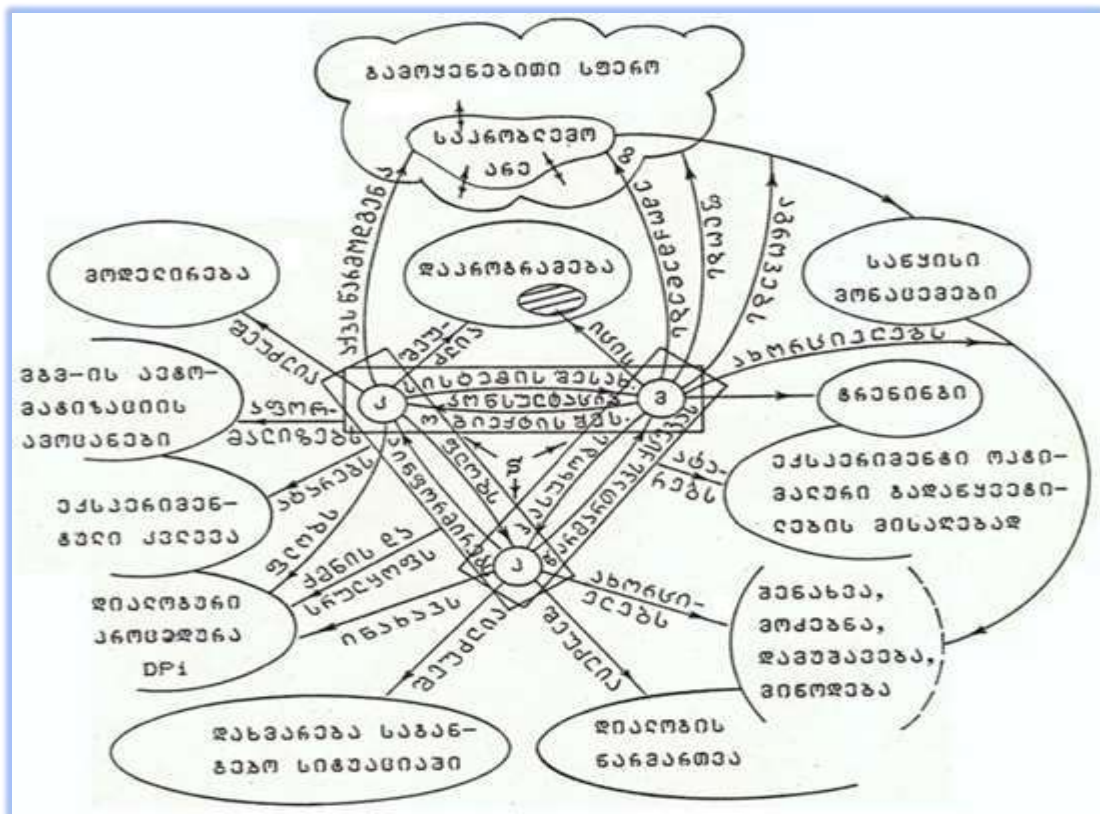
$d_4$  და  $d_5$  – ურთიერთასახვის ფუნქციებია პროგრამული პროდუქტისა პეტრის ქსელით და პირიქით.

ასახვის მექანიზმების პროგრამული რეალიზაცია განხორციელებულია ინტერფეისების სისტემის სახით, რომელიც ექსპერტ-დამპროექტებლის სამუშაო გარემოს რესურსია. იგი აგებულ იქნა პროტოფრეიმებისა და ელექტრონული ცხრილების იდეოლოგიით.

სისტემის დაპროექტების ტექნოლოგიის პერიოდული გამოყენების ციკლი დამოკიდებულია როგორც გამოყენებითი (საპრობლემო) სფეროს განვითარების დინამიკაზე (ობიექტური ფაქტორი), ასევე მისი ფუნქციური მომხმარებლის ცოდნის დონის ცვლილებაზე ობიექტის და სისტემის შესახებ (სუბიექტური ფაქტორი). მოცემული დროის მომენტისათვის არსებობს მომხმარებლის კონკრეტული ცოდნა, რომელიც მოდელირების მეთოდებით და ინსტრუმენტული საშუალებებით აისახება მონაცემთა და ცოდნის ბაზებზე და გამოყენებით პროგრამებში.

სისტემის ორ-ანალიზის, მოდელირების, პროექტირებისა და ექსპლუატაციის პროცესებში მონაწილეობს ფუნქციური (არაპროგრამიწისტი) მომხმარებელი (მ), ცოდნის კონსტრუირების ინჟინერ-კონსულტანტი (კ) და თვით პერსონალურ კომპიუტერზე რეალიზებული პროგრამა (პ).

მე-6 ნახაზზე ნაჩვენებია „მ-კ-პ“ სამეულის ურთიერთმიმართების ფორმალიზებული წარმოდგენა, მათი ძირითადი ფუნქციები, კავშირი საკვლევ ობიექტთან.



ნახ.6. საკვლევ სისტემის მოდელი

### 2.3. თანამედროვე განვითარების ტენდენცია: მონაცემთა საცავები, NoSQL ბაზები და დიდი მონაცემები

მესამე ინოვაციური გამოწვევა, რომელიც სტუ-ს მართვის ავტომატიზებული სისტემების კათედრაზე განხორციელდა XXI საუკუნის დასაწყისში (და დღესაც მნიშვნელოვანი და აქტუალურია) გახლდათ მონაცემთა საცავების (Data-warehouse, -repository, -storage) სამეცნიერო მიმართულების განვითარება, რომელიც მოიცავდა ხელოვნური ინტელექტის BI-ტექნოლოგიას, მონაცემთა მოპოვების (Data Mining) და მონაცემთა ონლაინ დამუშავების (OLAP) სისტემების ჩათვლით [18]. რელაციური მოდელებისა და ბაზების „მამამ“ - ედგარ კოდმა ჩამოაყალიბა მონაცემთა საცავების დაპროექტების და ფუნქციონირების პრინციპები, მირითადი 12 წესი, რომელსაც უნდა აკმაყოფილებდეს ნებისმიერი განაწილებული სისტემა მონაცემთა საცავით, რათა ჩატარდეს საინფორმაციო ბლოკების სრულფასოვანი ოპერატიული ანალიზური სამუშაოები [19].

მონაცემთა საცავების პროექტირების თეორიული, პრაქტიკული და მეთოდოლოგიური საფუძვლები, გახდა ერთ-ერთი მნიშვნელოვანი წინაპირობა დიდ მონაცემთა (Big Data) ეკოსისტემებისა და მონაცემთა მეცნიერების (Data Science) განვითარებისათვის, პროცესების მოპოვების (Process Mining) ჩათვლით [18].

*მას*-კათედრა გადაკეთდა ჯერ *მას*-მიმართულებად (2007), შემდეგ – *მას*-დეპარტამენტად (2013), ხოლო მოგვიანებით (2021) დაიშალა სამ დეპარტამენტად: „პროგრამული ინჟინერიის“, „ინფორმაციული სისტემების“ და „ინფორმაციული ტექნოლოგიების“, შესაბამისი ორგანიზაციული ცვლილებებით.

აღნიშნულ რეფორმებს ხელი არ შეუშლია *მას* (*პროგრამული ინჟინერიის*) დეპარტამენტის სამეცნიერო ინოვაციური გამოწვევების რეალიზაციისათვის. ეს მოვლენები კარგადაა ასახული დეპარტამენტის პროფესორის მონაგრაფიებსა (ნახ.1) და დოქტორანტების დისერტაციებში, რომლებიც 2004 – 2022 წლებში შეიქმნა (ლ. პეტრიაშვილი - „მონაცემთა საცავების ბიზნესისათვის“, ე. თურქია - „მონაცემთა მოპოვება და BI-ტექნოლოგიები“, დ. გულუა - „ქსელური ტექნოლოგიები ბიზნესისათვის“, ნ. თოფურია - „კონტენტ-მენეჯმენტი და ღრუბლოვანი ტექნოლოგიები“, ხ. ქრისტესიაშვილი - „ERP სისტემების დაპროექტების BPMN პროგრამები“, მ. ბიტარაშვილი - „Agile/UML მეთოდოლოგიები“, გ. ბასილაძე - „მულტიმედიური მონაცემთა ბაზები და ელექტრონული არჩევნები“, გ. კვიციანიძე - „NoSQL-ის MongoDB ბაზა განაწილებული ეკოსისტემისთვის“, ი. ბულია - „ვებ-სისტემების ინტეგრაცია“, მ. ხარიტონაშვილი - „ინტერდისციპლინური განათლება და ვირტუალური რეალობა“ და სხვ.).

### 3. დასკვნა

სტუ-ის ინფორმატიკისა და მართვის სისტემების ფაკულტეტის პროგრამული ინჟინერიის დეპარტამენტში (ისტორიულად - *მართვის ავტომატიზებული სისტემების კათედრა*) 1972-2022 წლებში პროფ. გ. სურგულაძის და მისი დოქტორანტების მიერ განხორციელებულია არა-ერთი ინოვაციური სამეცნიერო შედეგის დანერგვა როგორც საგანმანათლებლო, ასევე სხვადასხვა დარგის ობიექტებზე მენეჯმენტის პროცესების ინფორმაციული სისტემების სახით. შექმნილია ინფორმატიკის დიდაქტიკის მეცნიერული მიმართულების უნიკალური მეთოდური უზრუნველყოფის ლიტერატურული ბაზა (100 წიგნი: 25 მონოგრაფია, 20 სახელმძღვანელო, 55 დამხმარე და მეთოდური სახელმძღვანელო სტუდენტებისა და სპეციალისტებისათვის თეორიული,

პრაქტიკული და გამოყენებითი ინფორმატიკის სფეროში), დაცულია 40-ზე მეტი დისერტაცია კომპიუტინგის თანამედროვე მიმართულებებით.

### ლიტერატურა – References – Литература:

1. Website of Georgian Technical University. Prof. Gia Surguladze. Internet resource: [https://gtu.ge/katedrebi/kat94/g\\_sur\\_g.htm](https://gtu.ge/katedrebi/kat94/g_sur_g.htm) (10.12.2022)
2. zbMATH Open the first resource for mathematics. <https://zbmath.org/?q=surguladze%2C+G.+ai%3Asurguladze.g-g> (10.12.22)
3. Surguladze G., Topuria N., Gavardashvili A., Namchevadze Ts. (2018). Automation of Web-portal and Database Construction Processes for the Black Sea Ecosystem Monitoring. International Journal of Environmental and Ecological Engineering. World Academy of Scientific (WASET). v.12, N1, ISSN 1307-6892. Amsterdam, pp. 169-174.
4. Surguladze G., Topuria N., Petriashvili L. (2022). Information Society and Interdisciplinary Didactics of Informatics. International Scientific and Practical Conference “*EDUCATION AND SCIENCE OF TODAY: INTERS*”. Cambridge, UK: IV. DOI <https://doi.org/10.36074/logos-09.12.2022.30>. <https://archive.logos-science.com/index.php/conference-proceedings/issue/view/7/7>
5. Surguladze G., Topuria N., Gavardashvili A. (2017). Construction of the Multimedia Databases and Users Interfaces for Ecological System of Black Sea with Orm/Erm Information and Computer Technology, Modeling and Control”. Chapt.45. Nova Science Publishers. © Copyright, 3rd Quarter. 2017, USA ISBN 978-1-53612-094-3
6. Chogovadze G., Kachibaya V., Surguladze G. (1988). Theory of Relational Dependencies and Design of the Logical Schema of Databases. ISBN 5-511-00072-8; Monogr., Tbilisi State University. -230 p., (in Russian)
7. Chogovadze G., Surguladze G., Kachibaya V. (1991). Data and Knowledge Base Building Methods and Tools. Georgian technical Univ., Tbilisi, -145 p. (in Georgian)
8. Codd E.F. Further normalization of the database relational model. In Data Base Systems, Courant Computer Science Symposia 6. Prentice-Hall, Englewood Cliffs, N.J., 1972, pp. 65-98.
9. Wang C., Wedekind H. (1975). Segment Synthesis in Logocal Data Base Design. IBM J. RSD 19, N1, January, pp.71-77.
10. Wedekind H., Surguladze G. (1996). Technology of Designing of Distributed Systems on the Basis of Objectoriented Programming. ISSN 021-7164, GTU, Tbilissi. pp.96-100, (in Georgian)
11. Chogovadze G., Surguladze G., Chachanidze G. (1977). Designing of Relational Databases and Issues of Communication in Them. Abstracts of the 1-st All-Union School-Seminar "Intellectual Information Banks". – Moscow. MEPhI-GPI. Sukhumi, 23.09-2.10.1977, pp.55-58 (in Russian)
12. Chogovadze G., Gogichishvili G., Surguladze G., Sherozia T., Shonia O. Design and Building of the Management Information Systems. ISBN 99928-882-7-X. GTU, Tbilisi, 2001, 742 p. (in Georgian)
13. Gorbato V.A. Theory of partial ordering systems. Moscow, "Soviet radio", 1975, 336 p., (in Russian)
14. Tsertsvadze M., Kaishauri T., Surguladze G. (2022). Designing and Analyzing Relationships between Collections in NoSql Database MongoDB (History and Reality). Transactions of Georgian Technical University. Automated Control Systems. No 1(33), vol.2, pp. 154-163, (in Georgian)
15. Stamper R., Backhouse J., Marche S., Althaus K. (1987). Semantic normal form, Proc. of Informatics – a conference jointly sponsored by Aslib and BCS, Kings College
16. Stamper R., Ades Y. (2004). Semantic Normal Form and System Quality”. In Proc. IEE Conference on Requirements Engineering, Kyoto,
17. Ries L. (1990). Experiments with local networks of microcomputers. Translation from English, -Moscow, "Mir", 268 p., (in Russian)
18. Surguladze G., Petriashvili L. (2022). Design and Build of Data Warehouse for Internet Business. ISBN 978-9941-8-0623-0. GTU, "IT-Consulting Scientific Center", Tbilisi, -223 p. (in Georgian)
19. Codd's 12 Rules for Relational Database Management. Internet resource: <https://olap.com/learn-bi-olap/codds-paper/> (14.02.22)

**GEORGIAN TECHNICAL UNIVERSITY IS 100 YEARS OLD - SOME HISTORICAL  
INNOVATIVE SCIENTIFIC RESULTS OF THE DEPARTMENT OF  
"AUTOMATED CONTROL SYSTEMS"**

Surguladze Gia, Petriashvili Lili, Tsertsvadze Maka  
Georgian Technical University  
g.surguladze@gtu.ge, l.petriashvili@gtu.ge, maka@gtu.ge

**Summary**

The article discusses some innovative results of educational and scientific work in the field of informatics (computing) against the background of 50-year work (1972-2022) of the head of the Department of "Software Engineering" (Automated Control Systems), Doctor of Technical Sciences, Professor Gia Surguladze at the Technical University of Georgia (Polytechnic Institute) are discussed. The article presents some of the specific scientific results that were processed, introduced and developed by his initiative during the period 1975-1995 (in Georgia and beyond - from a historical point of view), and that are still actively used today. This is the creation of a complex of automated design algorithms of the optimal structure of the relational database using artificial intelligence methods, it has found application in intelligent data banks (in the form of knowledge bases). It is also important to create a network of automated workplaces in 1983-1993 (the first Georgian ERP-type system). It is implemented in the form of an expert system, with a graph-analytical tool of Minsk frames and Petri nets. The results are registered both in the Post-USSR Algorithms and Programs Foundation and in the European Mathematical Society (EMS) organization. More than 40 Ph.D theses were defended in this scientific direction at the Technical University of Georgia about: Data storage, ERP systems, relational and NoSQL databases, logistics management, Black Sea ecology, virtualization of interdisciplinary education, building Windows-, Web- and Mobile-integrated systems and others.

**ГРУЗИНСКОМУ ТЕХНИЧЕСКОМУ УНИВЕРСИТЕТУ 100 ЛЕТ - НЕКОТОРЫЕ  
ИСТОРИЧЕСКИЕ ИННОВАЦИОННЫЕ НАУЧНЫЕ РЕЗУЛЬТАТЫ КАФЕДРЫ «АСУ»**

Сургуладзе Г., Петриашвили Л., Церцвадзе М.  
Грузинский Технический Университет  
g.surguladze@gtu.ge, l.petriashvili@gtu.ge, maka@gtu.ge

**Резюме**

Некоторые инновационные результаты учебно-научной работы в области информатики (вычислительной техники) на фоне 50-летней работы (1972-2022 гг.) заведующего кафедрой «Программная инженерия» (автоматизированные системы управления), д.т.н. технических наук, профессор Гии Сургуладзе в Техническом университете Грузии (Политехнический институт). Представлены некоторые конкретные научные результаты, которые были обработаны, внедрены и развиты по его инициативе в 1975-1995 гг. (в Грузии и за ее пределами - с исторической точки зрения) и активно используются до сих пор. Это создание комплекса алгоритмов автоматизированного проектирования оптимальной структуры реляционной базы данных с использованием методов искусственного интеллекта, он нашел применение в интеллектуальных банках данных (в виде баз знаний). Важно также создание сети автоматизированных рабочих мест в 1983-1993 годах (первая грузинская система типа ERP). Она реализована в виде экспертной системы, с графоаналитическим аппаратом Минских фреймов и сетей Петри. Результаты зарегистрированы как в Постсоюзном фонде алгоритмов и программ, так и в организации European Mathematical Society (EMS). По данному научному направлению в Техническом Университете Грузии защищено более 40 диссертаций, охватывающие следующую тематику : хранилища данных, ERP-системы, реляционные и NoSQL базы данных, управление логистикой, экология Черного моря, виртуализация междисциплинарного образования, построение Windows-, Web- и Мобильных интегрированных систем и др.

# სტუ-100: „ხელოვნება-წარმოება-მეცნიერება“ (ისტორიული ფრაგმენტი) - პირველი ქართული ERP სისტემა და განვითარების ტენდენციები

გია სურგულაძე, გულბაათ ნარეშელაშვილი,  
კონსტანტინე თაგაური  
საქართველოს ტექნიკური უნივერსიტეტი  
g.surguladze@gtu.ge; g.nareshelashvili@gtu.ge; kokotagauri@gmail.com

## რეზიუმე

განხილულია საქართველოს 85-იან წლებში მეცნიერებისა და წარმოების ერთობლივი თანამშრომლობის საფუძველზე განხორციელებული ინოვაციური გამოწვევა. კერძოდ კომპლექსური ავტომატიზაციის პროექტის შესრულება და დანერგვა მსუბუქი მრეწველობის ობიექტებზე პირველი ქართული ERP სისტემის სახით. მეცნიერებისა და ტექნიკის კომიტეტის სახალხო მეურნეობის მართვის ინსტიტუტის, სტუ-ს მართვის ავტომატიზებული სისტემების კათედრისა და თბილისის N1 ტრიკოტაჟის ფაბრიკის („ოქსინო“) ბაზაზე. სისტემაში რეალიზებულია მარკეტინგის, საწარმოო და ტექნოლოგიური პროცესების მენეჯმენტის, დაპროექტების ავტომატიზაციის და ოპერატიული მართვის ამოცანები ხელოვნური ინტელექტის მეთოდების გამოყენებით. წარმოდგენილია მიღებული ინოვაციური სამეცნიერო-ტექნიკური შედეგების როგორც ისტორიული მნიშვნელობა, ასევე თანამედროვე განვითარების ტენდენციები.

**საკვანძო სიტყვები:** პროდუქცია. მარკეტინგი. წარმოება. მენეჯმენტი. ბიზნესპროცესი. ავტომატიზაცია. ERP სისტემა. ხელოვნური ინტელექტი. სიტუაციური ანალიზი. ცოდნის ბაზა. ექსპერტული სისტემა. სისტემის არქიტექტურა. DDD.

## 1. შესავალი

ყველაფერი დაიწყო იმით, რომ ყოფილი საბჭოთა კავშირის „უძრაობის“ პერიოდი შეიცვალა მ. გორბაჩოვის „პერესტროიკით“ (1985 წ.). ე. შევარდნაძის გადასვლით სსრკ საგარეო საქმეთა მინისტრად. მისი ვიზიტი აშშ-ის პრეზიდენტ რონალდ რეიგანთან დადებითი პოლიტიკური შედეგით დასრულდა. გაიხსნა დასავლეთის და აღმოსავლეთის ჩაკეტილი კარიბჭეები და კომპიუტერული ტექნიკა და ტექნოლოგიები დიდი სისწრაფით და მოცულობით შემოვიდა სსრკ-ში. დაიწყო ახალი ეტაპი ინფორმაციული სისტემებისა და ტექნოლოგიების სფეროში. უნივერსიტეტებს მიეცათ უფლება სასწავლო პროცესის პარალელურად განეხორციელებინათ სამეცნიერო-ტექნიკური პროექტები კომერციული ხელშეკრულებების ბაზაზე.

ერთ-ერთი აქტუალური პროექტი ამ დროისათვის იყო თბილისის N1 ტრიკოტაჟის ფაბრიკის („ოქსინოს“) დირექტორის, ბატონი თამაზ ცინცაძის ინიციატივა – შექმნილიყო მისი წარმოების ავტომატიზაციის სისტემა ახალი გამოთვლითი ტექნიკის საფუძველზე.

თ. ცინცაძე, ეს დიდებული ისტორიული პიროვნება გახლდათ გპი-ს სამთო-გეოლოგიის და მსუბუქი მრეწველობის ფაკულტეტის კურსდამთავრებული და 50-იანი წლების „გეპეის ჯაზის“ კვარტეტის აქტიური წევრი (ნახ.1).

(მაგ., სიმღერა-„ექსპრესი წყალტუბო – ქუთაისი“).



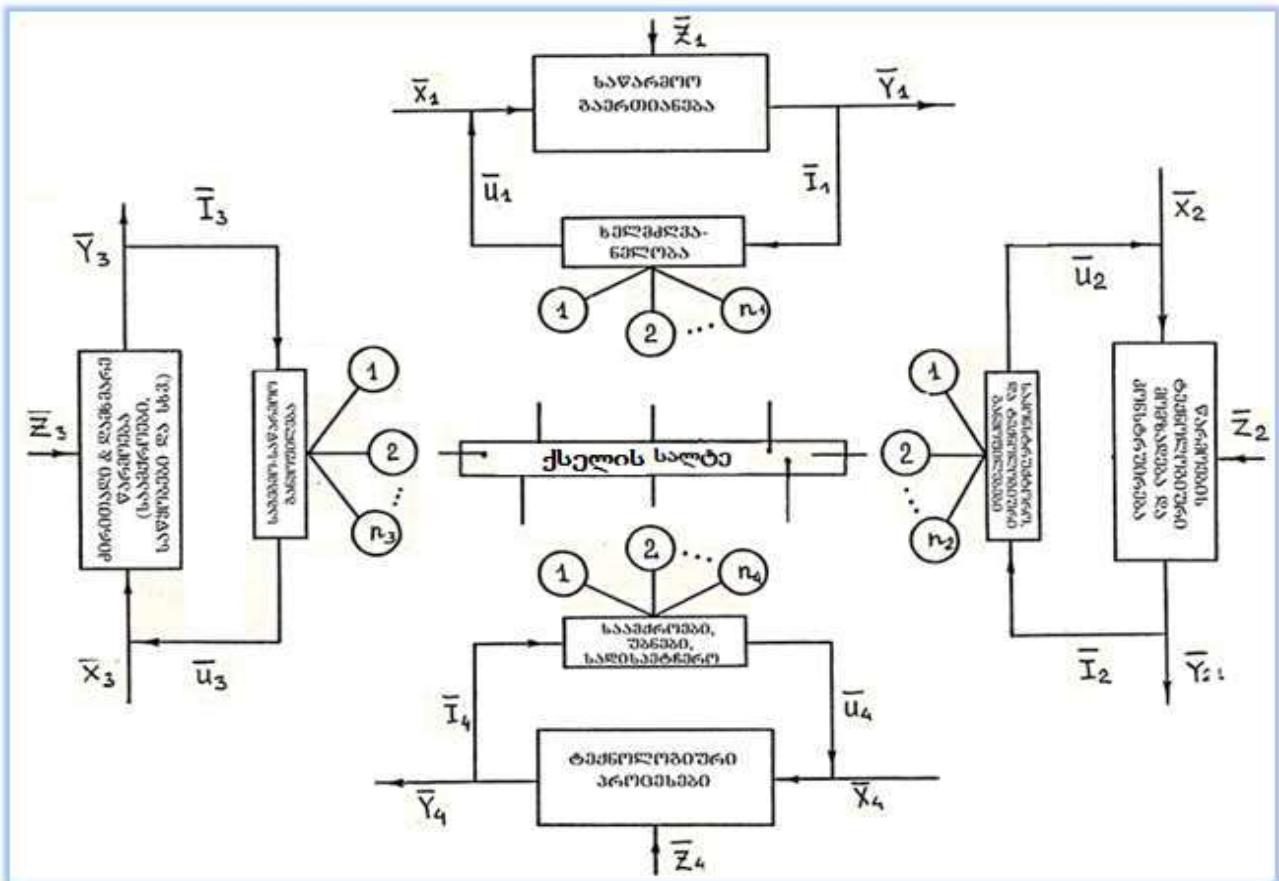
ნახ.1. „გეპეის კვარტეტი“ (50-ანი წ.): გურამ ბზვანელი, გურამ ბაკურაძე, თამაზ ცინცაძე და შოთა ხარაბაძე



საქართველოს მეცნიერებისა და ტექნიკის კომიტეტის ხელმძღვანელ მუშაკთა კვალი-ფიკაციის ამალგების ცენტრში (სადაც დღეს საგარეო საქმეთა სამინისტროა) „მას-ის და გამოთვლითი ტექნიკის“ ლექციებსა და სემინარებზე მოხდა ჩვენი გაცნობა და დაახლოება. მისი სადიპლომო ნაშრომი ეხებოდა ტრიკოტაჟის ფაბრიკის შემდგომი სრულყოფის საკითხებს ახალი (იმ დროისთვის) გამოთვლითი ტექნიკისა და ინფორმაციული ტექნოლოგიების ბაზაზე [1-3]. კურსის დასრულების შემდეგ ბატონმა თ. ცინცაძემ მოისურვა თავისი ინოვაციური ტექნოლოგიური წინადადებების განზოგადება და პრაქტიკულად რეალიზაცია პროგრამული სისტემის სახით. გაფორმდა „ისტორიული ხელშეკრულება“ და დაიწყო ინტესიური გუნდური მუშაობა ფაბრიკაში – ბიზნეს- და ტექნოლოგიური პროცესების შესასწავლად.

## 2. ძირითადი ნაწილი

სისტემის მოთხოვნილებებისა და არქიტექტურის (ინფრასტრუქტურის) დასადგენად. (დღევანდელი ხედვით, ჩვენი და ფაბრიკის სპეციალისტების ასეთი გუნდური მუშაობა მსგავსი იყო Agile-Scrum მიდგომისა (თუმცა ამ დროს ასეთი ცნება და მეთოდოლოგია არ არსებობდა !) დასაპროექტებელი მართვის სისტემის პირველი ზოგადი სქემა ასე გამოიყურებოდა (ნახ.2).



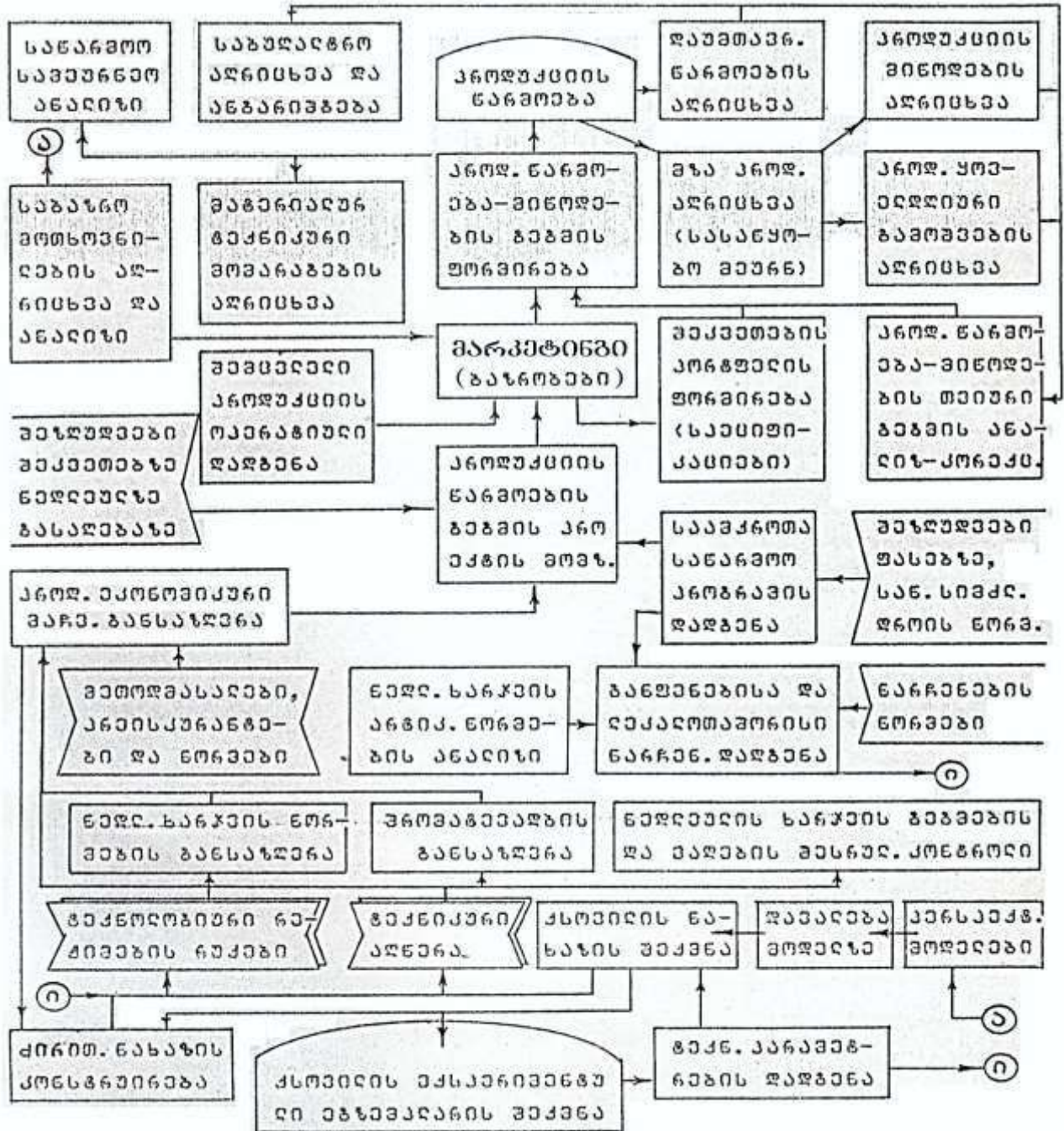
ნახ. 2. დასაპროექტებელი კომპლექსური სისტემის ზოგადი სტრუქტურა

ტერმინი Enterprise Resource Planning (ERP) პირველად გამოიყენა 1990-იან წლებში Gartner ჯგუფმა [1]. დღევანდელი გადასახედიდან, ჩვენი საპროექტო და დეველოპმენტის სამუშაოების შედეგი, რომელიც ფაბრიკის ფუნქციური განყოფილებებისა და საამქროების მენეჯერ-სპეციალისტების ავტომატიზებული სამუშაო ადგილების ქსელს წარმოადგენდა, კომპლექსურობით, სტრუქტურითა და შინაარსით ძალზე ახლოა სწორედ ERP სისტემის ტიპთან [4,5].

როგორც ამ ნახაზიდან ჩანს, მართვის პროცესების ინტეგრირებული სისტემა აერთიანებს მარკეტინგის, პროდუქციისა და რესურსების დაგეგმვის, წარმოებისა და მისი აღრიცხვის,

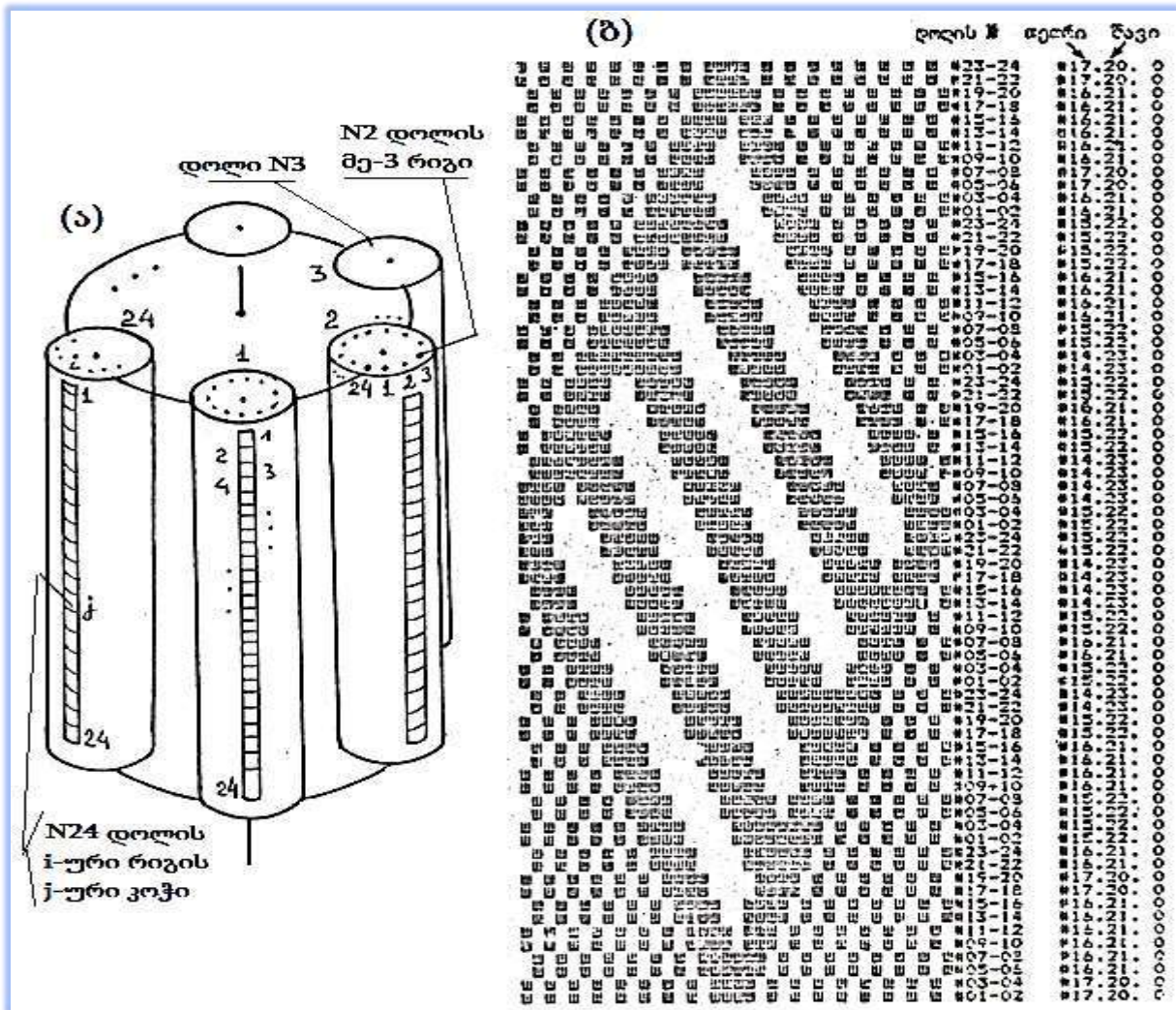
(მოიცავს დაუმთავრებელ წარმოებასაც), ტექნოლოგიური პროცესების, სასაწყობო მეურნეობის, კადრების მართვის, ფინანსების, ბუღალტერიისა და საბანკო ანგარიშების მენეჯმენტის და სხვ. ამოცანების ავტომატიზაციას ქსელური და ინფორმაციული ტექნოლოგიებით.

მე-3 ნახსაზე მოცემულია ჩვენი ინტეგრირებული სისტემის ინფორმაციულ-ტექნოლოგიური სქემის ძირითადი ვერსია, რომელიც ასახავს თითქმის სრულ სურათს მსუბუქი მრეწველობის ობიექტების მენეჯმენტის პროცესებს, ფუნქციურ ამოცანათა ბლოკებით.

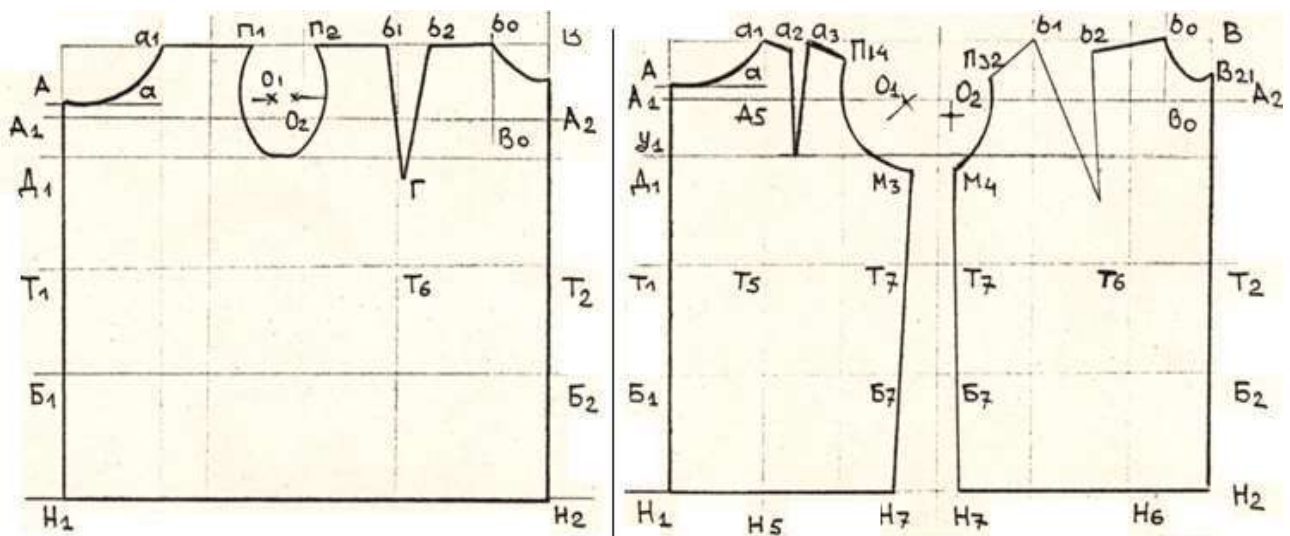


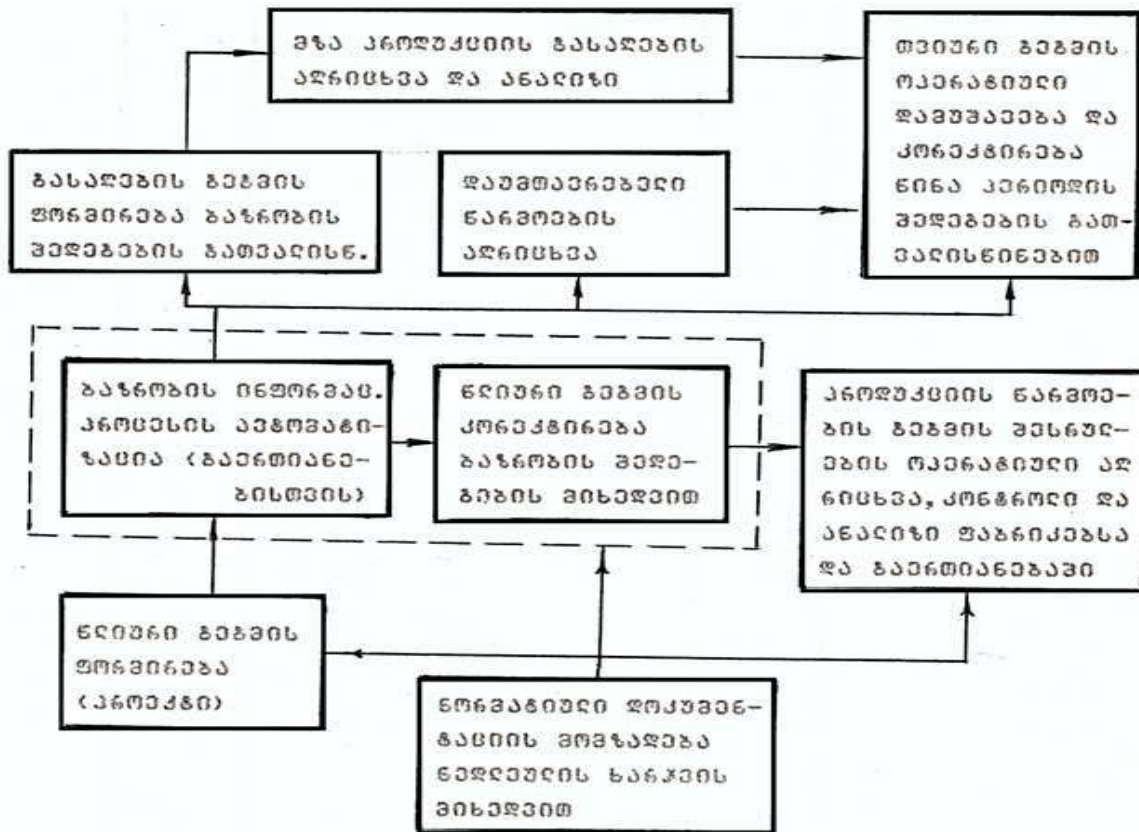
ნახ.3. ინფორმაციულ-ტექნოლოგიური სქემა

მე-4 და მე-5-ა,ბ ნახაზეებზე ნაჩვენებია, შესაბამისად, ტექნოლოგიური და საკონსტრუქტორო პროცესების ფრაგმენტები. აქ პროგრამულად ხორციელდება ამ პროცესების მართვა და საჭირო პარამეტრების გაანგარიშება, აგრეთვე თარგების ავტომატიზებული გამოხაზვა.



ნახ.4. მრგვალსაქსოვი სისტემის ფრაგმენტი (ა) და ქსოვილის ესკიზის მაკეტი (ბ), აგებული Robotron 1715-ზე (1988)

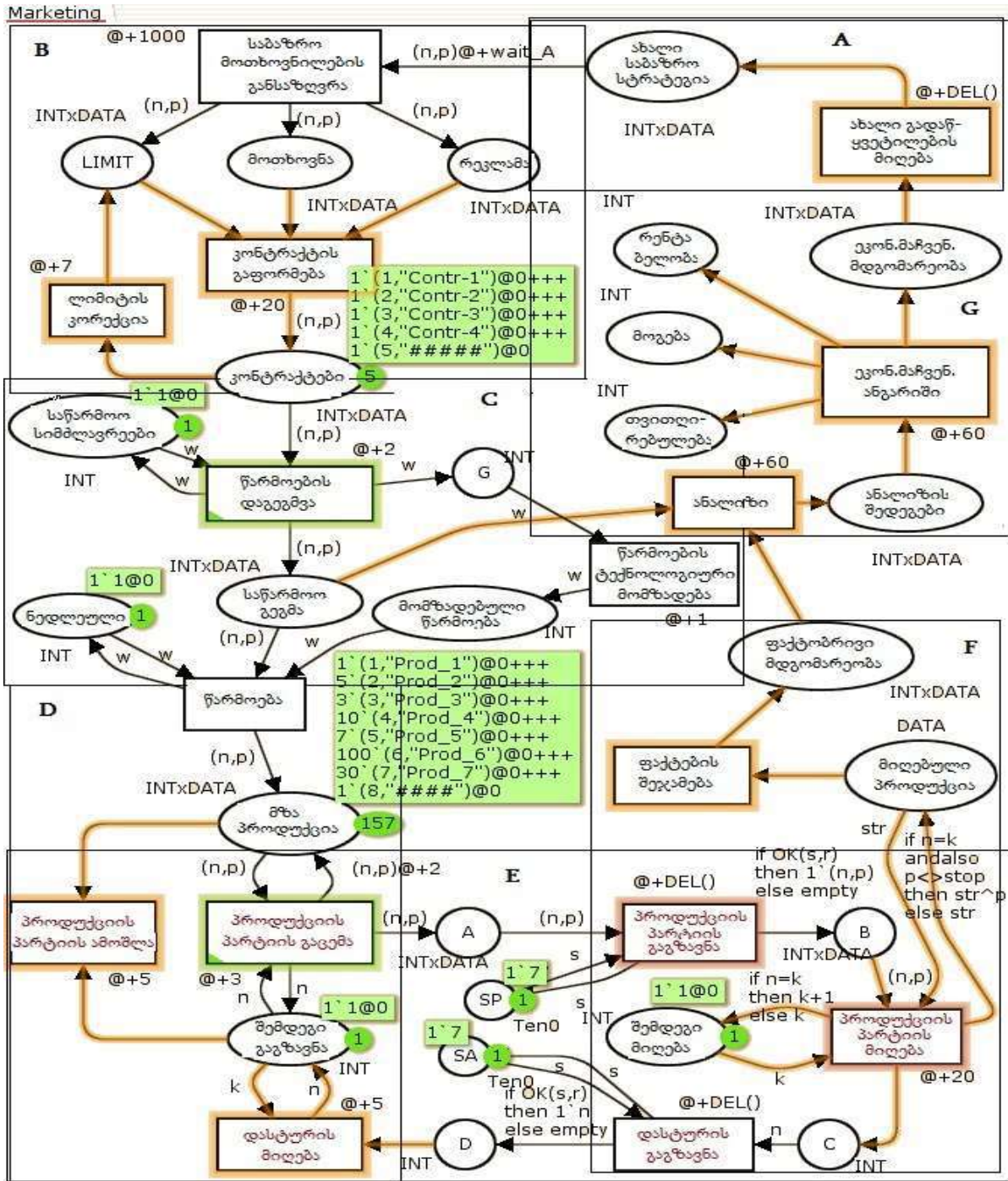




ნახ. 6. ფუნქციური ავტომატიზებული სამუშაო ადგილების ძირითად ამოცანათა ურთიერთკავშირის სქემა (ERP-სისტემის თვალსაზრისით)

ერთ-ერთი მნიშვნელოვანი მიდგომა ჩვენი სისტემის რეალიზაციის დროს იყო ხელოვნური ინტელექტის მეთოდების გამოყენება გადაწყვეტილების მიღების სისტემაში. დამუშავდა ექსპერტული სისტემა, რომლის „გულს“ მინსკის ფრეიმების საფუძველზე აგებული ცოდნის ბაზა შეადგენდა, ხოლო „თუ... მაშინ...“ პროცესის ლოგიკის სარეალიზაციოდ პეტრის სტოქასტური ქსელის დაპროექტა (ნახ.7) [6]. აქ წარმოდგენილია მარკეტინგული პროცესის იმიტაციური მოდელი საწარმოო ფორმის (ფაბრიკის) მაგალითზე. ნახაზზე გვაქვს შემდეგი ძირითადი იერარქიული მოდულები. ახალი საბაზრო სტრატეგიის ფორმირება (A), საბაზრო მოთხოვნების განსაზღვრა (B); პროდუქციის წარმოების დაგეგმვა (C); წარმოების ტექნიკური მომზადება და პროდუქციის წარმოება (D); პროდუქციის გაცემა (სასაწყობო მეურნეობა) და პროდუქციის გადაგზავნა (ტრანსპორტირება) (E), პროდუქციის მიღება და დამკვეთის ინფორმირება (F); ფაქტობრივი მდგომარეობის აღრიცხვა, საწარმოო და სარეალიზაციო გეგმების შესრულების ანალიზი, ეკონომიკური მაჩვენებლების ანგარიში და ანალიზი (G); გადაწყვეტილების მიღება ახალი საბაზრო სტრატეგიისთვის (A) და ა.შ. ციკლურად [4,5].

ხელოვნური ინტელექტის ერთ-ერთი მაგალითი ჩვენ სისტემაში იყო ოპერატიული მართვის ამოცანებისთვის გადაწყვეტილების მიღების სემიოტური მეთოდის გამოყენება. შემუშავდა რესურსების განაწილების მართვის პროცესის სემიოტური მოდელი [7]. მისი დეკლარაციული კომპონენტებით ხდება ტოპოლოგიური და საწარმოო სტრუქტურების ურთიერთკავშირთა ასპექტების ასახვა. აღნიშნული მართვის სემიოტური მოდელის პროგრამული კომპლექსი, ექსპერტიზის გავლის შემდეგ, მიღებულ იქნა „ალგორითმებისა და პროგრამების“ საკავშირო ფონდში (NIIOO 6379).



ნახ.7. საწარმოო ფირმის მარკეტინგული პროცესების პეტრის ქსელი

მარკეტინგული პროცესების ავტომატიზებული სისტემის პრაქტიკული გამოცდა ჩატარდა საქართველოს მსუბუქი მრეწველობის საწარმოო ობიექტების დარგობრივ ბაზრობაზე (საგამოფენო პავილიონი Expo Georgia). დარგობრივი სამინისტროს ხელმძღვანელობით ჩატარდა საქართველოს 20-ზე მეტი საქსოვი, სამკერვალო და სხვა ფაბრიკების საკუთარი პროდუქციის გამოფენა-გაყიდვის პროცესი (წლიური კონტრაქტების გაფორმება სავაჭრო ორგანიზაციებთან).

ჩენი პროგრამული ქვესისტემის („კონტრაქტების ავტომატიზაცია“) N1 ტრიკოტაჟის ფაბრიკის მაგალითზე იმდენად ეფექტური აღმოჩნდა, რომ ერთ კვირაში 7 ახალი დამკვეთი დაფიქსირდა (მაგ., თბილისის საქსოვი ფაბრიკები „ბახტრიონი“ და „1 მაისის“, ქუთაისის, ბათუმის, თელავის და სხვ.). მათი ინტერესი გამოიწვია სისტემის მოქნილობამ და კონტრაქტების

სწრაფად აგებამ. აგრეთვე მომავალი, საკავშირო ბაზრობის მოახლოებამ (მოსკოვში). ჩვენი 2 პერსონალური კომპიუტერით (Robotron 1715), მოსკოვის „ცსკ-ს მანეჟის“ პავილონში (1987) მოვემსახურეთ საქართველოს „საფეიქრო-საქსოვი“ ფირმების წარმომადგენლებს (სტუ-ს პროფ. თეიმურაზ დოლიძესთან ერთად). ამგვარად, ბატონი თამაზ ცინცაძის იდეა რეალიზებულ იქნა !

დღესაც, სტუ-ს „პროგრამული ინჟინერიის“ დეპარტამენტში (მართვის ავტომატიზებული სისტემების კათედრა) გრძელდება ტრადიციები. სადოქტორო დისერტაციები, სამაგისტრო და საბაკალავრო პროექტები სრულდება ინფორმაციული სისტემების დეველოპმენტის თემებზე. მათში ასახულია მენეჯმენტის საინფორმაციო სისტემების (MIS, ERP), გადაწყვეტილების მიღების მხარდამჭერი სისტემების (DSS), მონაცემთა საცავების (DWH, Big Data), ხელოვნური ინტელექტის სისტემების (AIS) და სხვ. პროგრამული დეველოპმენტის და ტესტირების პრობლემების გადაწყვეტა Agile-ის, Clear Architecture-ს, DDD-ს, DevOps და სხვა ტექნოლოგიების ბაზაზე [8-10].

### ლიტერატურა - References :

1. Enterprise Resource Planning. Internet resource: [https://en.wikipedia.org/wiki/Enterprise\\_resource\\_planning](https://en.wikipedia.org/wiki/Enterprise_resource_planning). (14.12.22)
2. Bukia G., Surguladze. (1985). CAD for the development of a targeted program to improve the Organizational and Technical Level of Enterprises. Abstr.of the Intern. Scientific-Technical Conf., Institute of National Economy Management. Science and Technology Committee of Georgia (in Russian)
3. Bukia G., Surguladze. (1986). Automated Ssystem for Building End User Applications. Abstr. of Republ. Scient.-Techn. Conf. of Georgia. Inst. of NEM. Science and Technology Committee of Georgia (in Russian)
4. Surguladze G., Turkia E., Gulua D. (2008). Perfection of Object-Oriented Projecting with a Process-Oriented Approach. Intern.Conferenc. “Educat, science and economics at univ.Integrat.to intern.educ.area”. Plock, Poland.
5. Surguladze G., Kristesiashvili Kh., Surguladze Gio. (2015), Modeling and Analysis of Enterprise Resource Management Busniness Process. ISBN 978-9941-20-557-6. Monogr., GTU, Tbilsi, 167 p. (in Georgian). [http://gtu.ge/book/gia\\_sueguladze/GiaSurgul\\_BPMN\\_ERP\\_2014.pdf](http://gtu.ge/book/gia_sueguladze/GiaSurgul_BPMN_ERP_2014.pdf)
6. Jensen K., Kristensen M.L., Wells L. (2007). Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. University of Aarhus. Denmark
7. Nareshelashvili G. (1984). Semiotic Model of Managing the Distribution of Resources in the Industry. PhD thesis. Technical University Petersburg.
8. Evans E. (2004). Domain Driven Design. Addison-Wesley Professional. 17th printing, Massachusetts, 2011
9. Starreveld A. (2021). Implementing Clean Architecture, DDD-style, with .NET Core. Published in VX Company. <https://medium.com/vx-company/implementing-clean-architecture-ddd-style-in-net-core-3bc3899f5978>
10. Chogovadze G., Surguladze G., Gulitashvili M., Dolidze S. (2020). Software Quality Management: Testing and Optimization. ISBN 978-9941-8-0629-2. GTU, „IT-Consulting Research Center“, Tbilisi, 364 p., (in Georgian).

(სტატია მიღებულია 11.11.2022)

## **GTU-100: "ART-PRODUCTION-SCIENCE" (HISTORICAL FRAGMENT) - THE FIRST GEORGIAN ERP SYSTEM AND DEVELOPMENT TRENDS**

Gia Surguladze, Gulbaat Nareshelashvili, Constantine Tagauri

Georgian Technical University

g.surguladze@gtu.ge; g.nareshelashvili@gtu.ge; kokotagauri@gmail.com

### **Summary**

The article discusses innovative challenges that were implemented as a result of joint cooperation between science and production domains in Georgia during 1985. In particular, execution and implementation of a complex automation project at light industry facilities in the form of the first Georgian ERP system. The latter has been carried out on the basis of the Institute of Public Economy Management of the Committee of Science and Technology, the Department of Automated Management Systems of GTU, and Tbilisi N1 Knitting Factory ("Oxino"). The system realized tasks related to marketing, production and technological process management, project automation and operational management using artificial intelligence methods. The historical significance of the obtained innovative scientific-technical results as well as the trends of modern development are presented.

*(Received 11.11.2022)*

## **ГТУ-100: „ИСКУССТВО-ПРОИЗВОДСТВО-НАУКА” (ИСТОРИЧЕСКИЙ ФРАГМЕНТ) – ПЕРВАЯ ГРУЗИНСКАЯ ERP СИСТЕМА И ТЕНДЕНЦИИ РАЗВИТИЯ**

Сургуладзе Г., Нарешелашвили Г., Тагаури К.

Грузинский Технический Университет

g.surguladze@gtu.ge; g.nareshelashvili@gtu.ge; kokotagauri@gmail.com

### **Резюме**

Обсуждается инновационный вызов, реализованный на основе совместной кооперации науки и производства в Грузии в 1985-х годах. В частности, внедрение и реализация проекта комплексной автоматизации на объектах легкой промышленности в виде первой грузинской ERP-системы. На базе Института управления народным хозяйством Комитета науки и техники, кафедры автоматизированных систем управления ГТУ и Тбилисской трикотажной фабрики N1 ("Оксино"). Система реализует задачи маркетинга, управления производственными и технологическими процессами, автоматизации проектирования и оперативного управления с использованием методов искусственного интеллекта. Представлено историческое значение полученных инновационных научно-технических результатов, а также тенденции современного развития.

*(Поступила в редакцию 11.11.22)*

# მართვის სისტემის მოდელირების თავისებურება დინამიკურად ცვალებადი საპრობლემო გარემოსათვის

თეიმურაზ სუხიაშვილი  
საქართველოს ტექნიკური უნივერსიტეტი  
sukhiashviliteimuraz08@gtu.ge

## რეზიუმე

განხილულია ობიექტ-ორიენტირებულ სისტემებში ნიმუშების ფორმირებისა და მათი ორგანიზაციის საშუალებები, რათა მარტივი იყოს მათი გამოყენება ახალი სისტემის აგების ან არსებულის განვითარებისას. წინა გამოცდილება და მიღებული შეთანხმებები იძლევა საფუძველს, რომ დამპროექტებელმა გამოიყენოს ტიპური საშუალებები (ნიმუშები) ტიპური პრობლემის გადასაწყვეტად. ნებისმიერი კარგად სტრუქტურირებული ობიექტ-ორიენტირებული სისტემა მოიცავს ნიმუშებს (რეალიზების მექანიზმებს) აბსტრაქციის ნებისმიერ დონეზე. მკაფიოდ გამოვყოფთ რა ნიმუშებს სისტემაში, იგი გახდება უფრო გასაგები და მარტივი განვითარებისათვის. ამიტომ, სისტემის აგებისას სასურველი იქნება აღიწეროს მისთვის დამახასიათებელი ნიმუშები, რათა დავეხმაროთ მათ, ვინც შემდგომში მოინდომებს მის ხელმეორედ გამოყენებას ან მოახდინოს მისი მოდიფიცირება.

**საკვანძო სიტყვები:** მართვის სისტემა. UML მოდელირება. ობ-დაპროექტება. Pattern. პროექტირების ნიმუში. არქიტექტურული ნიმუში. ნიმუში-მექანიზმი. ნიმუში-კარკასი.

## 1. შესავალი

პრაქტიკაში ინტერეს წარმოადგენს ორი სახის ნიმუში (pattern): პროექტირების ნიმუშები – მექანიზმები, რომლებიც აღწერს კლასების ნაკრების სტრუქტურას და ქცევას და არქიტექტურული ნიმუშები – კარკასები კი – სტრუქტურას და ქცევას მთლიანად სისტემისათვის [1, 2].

*პროექტირების ნიმუში* ან მექანიზმი, პირველ რიგში, უბრალოდ ასახელებს აბსტრაქციების სიმრავლეს, რომლებიც ერთობლივად მუშაობს ტიპური ქცევის უზრუნველსაყოფად, რომელსაც აქვს გარკვეული ინტერესი. ასეთი მექანიზმები მოდელირდება როგორც უბრალო კოოპერაციები, რამდენადაც ისინი კლასების ნაკრებისათვის ასახავს მხოლოდ კლასებს. ასეთი კოოპერაციის გახსნისას შესაძლებელია მისი სტრუქტურული ასპექტების (ჩვეულებრივ გამოსახული კლასების დიგრამის სახით) და ქცევითი (გამოსახული ურთიერთქმედების დიაგრამით). მეორეს მხრივ, მექანიზმი ასახელებს შაბლონს ერთობლივად მომუშავე აბსტრაქციებისათვის. ასეთი მექანიზმები მოდელირდება პარამეტრიზებული კოოპერაციების სახით, რომლებიც UML-ში გამოისახება კლასების შაბლონის სახით.

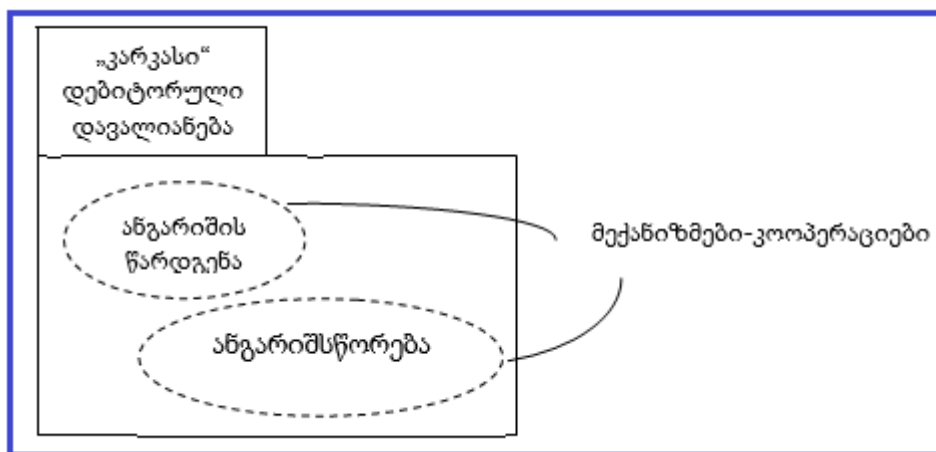
ასეთი კოოპერაციის გახსნისას ჩანს სტრუქტურული და ქცევითი ასპექტები. შეკვრისას კი გამოჩნდება თუ როგორ გამოიყენება ნიმუში საკვლევ სისტემასთან კოოპერაციის ნაწილების დაკავშირებით, სისტემაში არსებულ აბსტრაქციებთან. მექანიზმის მოდელირებისას პარამეტრიზებული კოოპერაციის სახით აღიწერება მართვის და დაკავშირების ის ელემენტები, რომელთა მეშვეობით შესაძლებელია შაბლონის ადაპტირება, მისი პარამეტრების ცვლილებით. ამდაგვარი კოოპერაციები შესაძლებელია გამოიყენებოდეს სისტემის სხვადასხვა ნაწილში და დაუკავშირდეს სხვადასხვა აბსტრაქციას.



არქიტექტურული ნიმუში ან კარკასი სთავაზობს გაფართოებად შაბლონს აპლიკაციებისათვის რომელიმე კონკრეტულ სფეროში. კარკასი – ეს მეტია ვიდრე მექანიზმი. ფაქტიურად შესაძლებელია ითქვას, რომ კარკასი – ეს მიკროარქიტექტურის სახეა, რომელშიც ჩართულია მექანიზმების სიმრავლე, რომლებიც მუშაობს ერთობლივად მოცემულ პრობლემურ გარემოში ტიპური პრობლემის გადასაწყვეტად. კარკასის სპეციფიციკების დროს ხდება არქიტექტურის ჩონჩხის აღწერა მთელი თავისი მმართველი ორგანოებით, რომლებიც იხსნება ამ კარკასის საჭირო კონტექსტში ადაპტირებით დაინტერესებული მომხმარებლების მიერ.

UML-ში ჩვეულებრივ მოდელირდება დაპროექტების ნიმუშები, რომლებსაც ასევე უწოდებენ მექანიზმებს [3,4]. ისინი გამოისახება კოოპერაციების სახით. ანალოგიურად არქიტექტურული ნიმუშები მოდელირდება კარკასების სახით, წარმოდგენილი სტერეო-ტიპული პაკეტების სახით.

ორივე ტიპის ნიმუშისათვის გათვალისწინებულია გრაფიკული გამოსახვა, რომელიც გამოსახულია 1-ელ ნახაზზე.



ნახ. 1

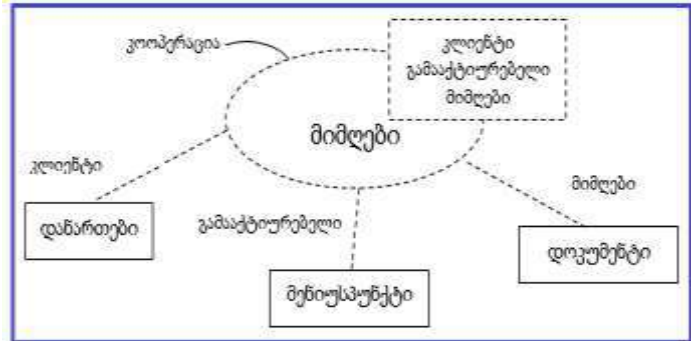
## 2. ძირითადი ნაწილი

პროექტირების ნიმუშები გამოიყენება ტიპური სიტუაციების მოდელირებისათვის, რომლებიც წარმოიშობა დაპროექტების პროცესში. იგი გამოისახება პარამეტრიზებული კოოპერაციის სახით. ნიმუში (კოოპერაცია) არის აბსტრაქციების ნაკრები, რომლის სტრუქტურა და ქცევა მოწოდებულია ერთობლივად მუშაობისას შეასრულოს გარკვეული სასარგებლო ფუნქცია. კოოპერაციის პარამეტრები ასახელებს იმ ელემენტებს, რომლებიც ნიმუშის გამომყენებელმა უნდა დააკავშიროს რამესთან. ამის გამო, პროექტირების ნიმუში გარდაიქმნება შაბლონად, რომელიც გამოიყენება კონკრეტულ კონტექსტში ელემენტების ჩასმით, შაბლონის პარამეტრების შესაბამისად.

დაპროექტების ნიმუშის მოდელირებისათვის პირველ რიგში უნდა მოვახდინოთ ტიპური პრობლემის ტიპური გადაწყვეტა და მისი მატერიალიზება მექანიზმის სახით. შემდეგ, წარმოვადგინოთ მექანიზმი კოოპერაციის სახით და აღწეროთ მისი სტრუქტურული და ქცევითი ასპექტები. ბოლოს, მოვახდინოთ დაპროექტების ნიმუშების იმ ელემენტთა იდენტიფიცირება, რომლებიც უნდა დავუკავშიროთ ელემენტებს კონკრეტულ კონტექსტში და გამოვსახოთ ისინი კოოპერაციის პარამეტრების სახით.

მე-2 ნახაზზე მოყვანილია დაპროექტების ნიმუშის გამოყენება, რომლის დოკუმენტაცია გვამცნობს, რომ ნიმუში ახდენს „მიმართვის ინკაფსულაციას ობიექტის სახით, რომელიც საშუალებას იძლევა მოვახდინოთ კლიენტების პარამეტრიზირება, რომლებიც გამოსცემენ სხვადასხვა მიმართვებს, დააყენონ მიმართვები რიგში და ასევე გამოიყენონ მოხსნის ოპერაციები [1]. როგორც მოდელიდან ჩანს, ამ ნიმუშს აქვს სამი პარამეტრი, რომლებიც უნდა დაუკავშირდეს ელემენტებს მოცემულ კონტექსტში.

მოდელი გვიჩვენებს ასეთი დაკავშირების მაგალითს, სადაც ნიმუშის პარამეტრები უკავშირდება კლასებს.



ნახ. 2.

ტიპური არქიტექტურული გადაწყვეტის მოდელირებისათვის გამოიყენება კარკასი (არქიტექტურული ნიმუში). კარკასით მოდელირდება ინფრასტრუქტურა, რომელიც შემდგომში ხელმეორედ უნდა იქნეს გამოყენებული, ან მოხდეს მისი ადაპტირება რომელიმე კონტექსტში.

კარკასები გამოისახება სტერეოტიპული პაკეტის სახით. იგი მოიცავს კლასებს, ინტერფეისებს, პრეცედენტებს, კომპონენტებს, კვანძებს, კოოპერაციებს და სხვა კარკასებს.

სხვა სიტყვებით, კარკასში ლაგდება ყველა აბსტრაქცია, რომლებიც მუშაობს რა ერთობლივად, ახდენს გაფართოებული შაბლონის ფორმირებას დანიარტებისათვის კონკრეტულ სფეროში. ზოგიერთი ელემენტი იქნება გახსნილი და გვთავაზობს რესურსებს, რომლებსაც იყენებს კლიენტი. ეს კარკასის შეკავშირების ელემენტებია, რომლებიც თქვენ შეგიძლიათ დაუკავშიროთ აბსტრაქციებს თავისი კონტექსტიდან. სხვა გახსნილი ელემენტები არის დაპროექტების ნიმუშები და შეუსაბამებენ რესურსებს, რომლებსაც უკავშირდება კლიენტი.

ეს არის კარკასის შესაერთებელი, რომლებიც ივსება პროექტირების ნიმუშთან დაკავშირებისას. ბოლოს, ზოგიერთი ელემენტი იქნება დახურული ან დაცული, ისინი შეესაბამება კარკასის ინკაფსულირებულ ელემენტებს, რომლებიც არ ჩანს გარედან.

არქიტექტურული ნიმუშის მოდელირებისას უნდა დავიმახსოვროთ, რომ კარკასი არქიტექტურის აღწერაა, შეიძლება არასრული და პარამეტრიზებული. შესაბამისად, ყველაფერი, რაც ცნობილია კარგად სტრუქტურირებული არქიტექტურის მოდელირების შესახებ, სრული სახით გამოიყენება კარგად სტრუქტურირებული კარკასების მიმართ.

არქიტექტურული ნიმუშის მოდელირებისას კარკასის საფუძვლად უნდა დაუდოთ უკვე არსებული, გასინჯული არქიტექტურა. დავამოდელიროთ კარკასი სტერეოტიპული პაკეტის სახით, რომელიც შეიცავს ყველა ელემენტს (განსაკუთრებით პროექტირების ნიმუშებს). ნიმუშის მომხმარებელმა უნდა იცოდეს, რომელი კლასები უნდა გაფართოვდეს, რომელი ოპერაციები იქნას რეალიზებული და რომელი სიგნალები დამუშავებული.

### 3. დასკვნა

UML-ში არსებული ნიმუშების ორგანიზაციის მოყვანილი საშუალებები საგრძნობლად გაუადვილებს დამპროექტებელს მართვის სისტემის მოდელირებას, როგორც ახლის, ისე არსებულის გაფართოებისას, გამოიყენებს რა ტიპურ საშუალებებს ტიპური პრობლემის გადასაწყვეტად არსებულ კონტექსტში.

**ლიტერატურა – References – Литература:**

1. Arlow D., Neustadt A. (2008). UML2 and the unified process. 2nd ed., Practical Object-Oriented Analysis and Design. St. Petersburg-Moscow (in Russian)
2. Sukhiashvili T. (2020). Unified Modeling Language (UML2) and Unified Software Development Process (UP). Georgian Techn. Univ., (in Georgian). Internet resource: <https://gtu.ge/book/uml2.pdf>
3. Sukhiashvili T. (2007). Processing of Modification Mechanisms in the Object-Oriented Design. Georgian Technical University. Automated Control Systems, No 1(2), pp. 115-118
4. Sukhiashvili T. (2016). Development means of adaptation at object-oriented design. Georgian Technical University. Automated Control Systems. No 2(31), pp. 93-95.

*(სტატია მიღებულია 15.11.2022)*

**FEATURES OF MODELING CONTROL SYSTEMS FOR  
A DYNAMICLY CHANGING PROBLEM ENVIRONMENT**

Sukhiashvili Teimuraz  
Georgian Technical University  
sukhiashviliteimuraz08@gtu.ge

**Summary**

When designing the architecture of a new system or developing an existing one, past experience and accepted conventions enable the designer to use typical tools to solve a typical problem. Any well-structured object-oriented system contains patterns (implementation mechanisms) at any level of abstraction. By explicitly identifying patterns (implementation mechanisms) in the system, it will become more understandable and easier to develop. Therefore, when building a system, it is desirable to describe patterns (implementation mechanisms) to help those who want to reuse it or modify it. The article discusses the means of formation and organization of patterns in object-oriented systems, in order to simplify their application when building a new system for the development of an existing one.

*(Received 15.11.2022)*

**ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ СИСТЕМ УПРАВЛЕНИЯ ДЛЯ  
ДИНАМИЧЕСКИ МЕНЯЮЩЕЙ ПРОБЛЕМНОЙ СРЕДЫ**

Сушиашвили Т.  
Грузинский Технический Университет  
sukhiashviliteimuraz08@gtu.ge

**Резюме**

Рассматриваются средства формирования и организации образцов в объектно ориентированных системах управления, чтобы упростить их применение с целью построения новой системы или дальнейшего развития существующей. Препятствием и принятыми соглашениями дает возможность проектировщику использовать типичные средства для решения типичной проблемы. Любая хорошо структурированная объектно-ориентированная система содержит образцы (механизмы реализации) на любом уровне абстракции. При построении системы желательно описать образцы (механизмы реализации), чтобы помочь тем, которые захотят повторно его использовать или произвести его модификацию.

*(Поступила в редакцию 15.11.22)*

## ERP და SCADA სისტემების ინტეგრაცია საწარმოო ინდუსტრიაში და მათი ეფექტურობის ანალიზი

თამარ ხუჭუა

საქართველოს ტექნიკური უნივერსიტეტი  
tamunaxuchua@yahoo.com

### რეზიუმე

განხილულია ERP და SCADA სისტემების ინტეგრაცია საწარმოო ინდუსტრიისათვის. თანამედროვე საწარმოო ორგანიზაციებისათვის მნიშვნელოვანია ავტომატიზების ისეთი სქემის არსებობა, რომლის მეშვეობით შესაძლებელი იქნება წარმოების მონაცემების რეალურ დროში მიღება და დამუშავება. დღეისათვის ბევრ საწარმოო ორგანიზაციაში ERP და SCADA (ასევე საწარმოო პროცესის ავტომატიზების სხვა პროგრამები) სისტემები ფუნქციონირებს ერთმანეთისაგან დამოუკიდებლად, SCADA სისტემიდან მონაცემების მიწოდების პროცესი, უმეტეს შემთხვევაში, ხორციელდება ხელით, რაც განაპირობებს გადაწყვეტილებისათვის საჭირო მონაცემების დაგვიანებით მიწოდებას და ამცირებს ინფორმაციის სანდოობას. ინტეგრაციის საშუალებით ERP სისტემას წვდომა აქვს SCADA-ს მონაცემთა ბაზაზე, წარმოების ოპერატიულ მონაცემებზე, მონაცემები ინტეგრაციის გზით გადაიცემა რეალურ დროში, მუშავდება და ინახება ორგანიზაციის ERP სისტემის ერთიან მონაცემთა ბაზაში. ოპერატიული მონაცემების რეალურ დროში მიღება-ანალიზის შესაძლებლობა ეხმარება როგორც წარმოების ოპერატორებს, ტექნიკური სფეროს პროფესიონალებსა და ორგანიზაციის ხელმძღვანელ პირებს საწარმოო პროცესის მიმდინარეობის უწყვეტ რეჟიმში მონიტორინგისა და შესაბამისად, სწრაფი გადაწყვეტილებების მიღებაში, ასევე მინიმუმამდე ამცირებს ERP სისტემისათვის მონაცემების ხელით მიწოდებაზე დახარჯულ დროს. ნაშრომი მიზნად ისახავს წარმოაჩინოს ის სარგებელი, რაც შეიძლება უზრუნველყოს ERP და SCADA სისტემების ინტეგრაციამ.

**საკვანძო სიტყვები:** წარმოების პროცესების ოპტიმიზება. ინტეგრაცია. ERP სისტემა. SCADA სისტემა. ავტომატიზება.

### 1. შესავალი

წარმოების სისტემის განვითარება და ეფექტური მეთოდების ძიება თანამედროვე საწარმოო ორგანიზაციის ყოველდღიურობას წარმოადგენს. ეფექტური ინტეგრირებული სისტემის შემუშავება, რომელიც უზრუნველყოფს ინფორმაციის რეალურ დროში გაცვლას აღჭურვილობის მუშაობისა და საწარმოო პროცესების შესრულებასთან დაკავშირებით, იძლევა მოწოდებული მონაცემების სიზუსტისა და სანდოობის გარანტიას. ინტეგრაცია გადაწყვეტილების მიმღებ პირებს აძლევს სწრაფი უკუკავშირის საშუალებას, რაც შესაძლებელს ხდის პროდუქტის ხარისხისა და პროცესის მუშაობის გაუმჯობესებას და უზრუნველყოფს ინფორმაციის უწყვეტობას, რაც მნიშვნელოვანია დროული გადაწყვეტილების მიღებაში. წარმოების ეფექტურობის გაზრდის თვალსაზრისით შესწავლილია რამდენიმე მეთოდი, რომელთაგან მნიშვნელოვნების მაღალი მაჩვენებლით გამოიყოფა მართვისა და კონტროლის ავტომატიზება, რომელიც მიზნად ისახავს საწარმოო პროცესსა და ორგანიზაციის მენეჯმენტს შორის კომუნიკაციის მანძილისა და ინფორმაციის მიწოდების შეფერხების აღმოფხვრას.

საწარმოო დანადგარებისა და აღჭურვილობების მუშაობის, ასევე წარმოების პროცესისა და შედეგების შესახებ მონაცემები გროვდება SCADA პროგრამული უზრუნველყოფის საშუალებით, შემდეგ, ინტეგრაციის გზით ხდება მათი მიწოდება ERP სისტემისათვის, სადაც შესაძლებელია მონაცემთა დამუშავება და წარმოების ეფექტურობის გაანალიზება.

კვლევები აჩვენებს წარმოების პროდუქტიულობის ზრდას 10%-15%-მდე საწარმოო ორგანიზაციაში ინტეგრირებული სისტემების გამოყენების შემდგომ. ნაშრომში განხილულია ინტეგრაციის დადებითი ასპექტები [1,4,5,7].

## 2. ინტეგრაციის მნიშვნელობა

ERP სისტემაში წარმოების ოპერატიული მონაცემების რეალურ დროში არსებობა და სანდოობა მოითხოვს უშუალო წვდომას საწარმოო პროცესში ჩართული სისტემის მიერ შეგროვებულ მონაცემებზე. რეალურ დროში მონაცემებზე წვდომის ნაკლებობა ართულებს სწორი გადაწყვეტილებების მიღებას ადმინისტრაციის ყველა დონეზე. ინტეგრირებული გარემოს არარსებობა მოითხოვს მონაცემების მიღებას დამოუკიდებელი სისტემებიდან. მონაცემები შეიძლება იყოს იმპორტირებული გარე წყაროებიდან ან უშუალოდ მომხმარებლის მიერ, მაგრამ ჩვენს მიერ დასმული ამოცანა მოითხოვს თანმიმდევრულ საინფორმაციო სისტემას და ხორციელდება ძირითადად ინტეგრირებული სისტემის სახით, რომელიც დაფუძნებულია საერთო, ცენტრალურ ან განაწილებულ მონაცემთა ბაზაზე [1,3,6].

სისტემებს შორის მონაცემთა გაცვლის დონეზე ინტეგრაცია შეიძლება განხორციელდეს სხვადასხვა გზით: საერთო მონაცემთა ბაზით; ცალკეული მონაცემთა ბაზების გამოყენებით და მონაცემთა გაცვლის საერთო ფორმატის შემუშავებით.

საწარმოო პროცესების მონაცემებზე რეალურ დროში წვდომის უზრუნველსაყოფად, აუცილებელია ორგანიზაციაში გამოყენებული პროგრამული უზრუნველყოფის ინტეგრირება.



ნახ.1. პროგრამულ უზრუნველყოფებს შორის ინტეგრაცია

როგორც ნახ.1-ში ჩანს, იმისთვის, რომ ERP სისტემამ შეძლოს მონაცემთა შეგროვება რეალურ დროში, უნდა არსებობდეს პირდაპირი კომუნიკაცია წარმოების პროცესთან, რადგან ERP-ს არ აქვს ასეთი ინსტრუმენტი, ამ მიზნით გამოიყენება SCADA პროგრამული უზრუნველყოფის ინსტრუმენტები, რომელიც ახორციელებს მონაცემთა შეგროვებას საწარმოო დანადგარებსა და პერიფერიულ მოწყობილობებთან ინტეგრაციის გზით. ინფორმაცია ინახება მონაცემთა ბაზაში, იმისათვის, რომ ინტეგრაცია ERP და SCADA სისტემებს შორის იყოს წარმატებული, აუცილებელია მონაცემთა ბაზების თავსებადობა [2,3,6,7].

საწარმოს რესურსების დაგეგმვისა (ERP) და პროცესის კონტროლის სისტემებს შორის ინტეგრაციის უზრუნველყოფით მიღებული სქემა საშუალებას აძლევს საწარმოო ორგანიზაციას საწარმოო პროცესი გახდეს ეფექტური და მოხდეს წარმოების ოპტიმიზაცია. ინტეგრირებული ERP სისტემა იღებს, ამუშავებს და ხელმისაწვდომს ხდის მონაცემებს როგორც რეალურ დროში, ასევე ნებისმიერი საჭირო პერიოდისათვის, ამარტივებს კოორდინაციას პროცესებს შორის, უზრუნველყოფს გამჭვირვალობასა და მონაცემთა ხელმისაწვდომობას.

### 3. ERP სისტემა

ERP (Enterprise Resource Planning) არის პროგრამული უზრუნველყოფა, რომელიც საშუალებას აძლევს ორგანიზაციას გამოიყენოს ბიზნეს პროცესების მართვის ინტეგრირებული აპლიკაციების სისტემა საწარმოო პროცესების მართვისა და ავტომატიზაციისთვის. საწარმოო ინდუსტრიას ERP სისტემის გამოყენება საშუალებას აძლევს მოახდინოს პროცესების სრული ავტომატიზება და მიიღოს ღირებული სარგებელი, როგორცაა ხარჯების შემცირება, პროცესების გამარტივება, ზრდის მართვა და კონკურენტული უპირატესობის მოპოვება. ERP სისტემა საწარმოო ორგანიზაციას საშუალებას აძლევს ერთი თვალის პრინციპით უზრუნველყოს მონიტორინგი და მართვა ისეთი პროცესების როგორცაა – მიწოდების ჯაჭვი, გაყიდვები და დისტრიბუცია, ინვენტარი, ფინანსები, ლოგისტიკა, ადამიანური რესურსების მართვა და სხვ. ERP სისტემა უზრუნველყოფს საწარმოო დანადგარებისა და მოწყობილობების ინტეგრირებას ერთ მონაცემთა ბაზაში, რაც უფრო ეფექტურია შიგა და გარე კომუნიკაციისა და მონაცემთა დამუშავებისათვის.

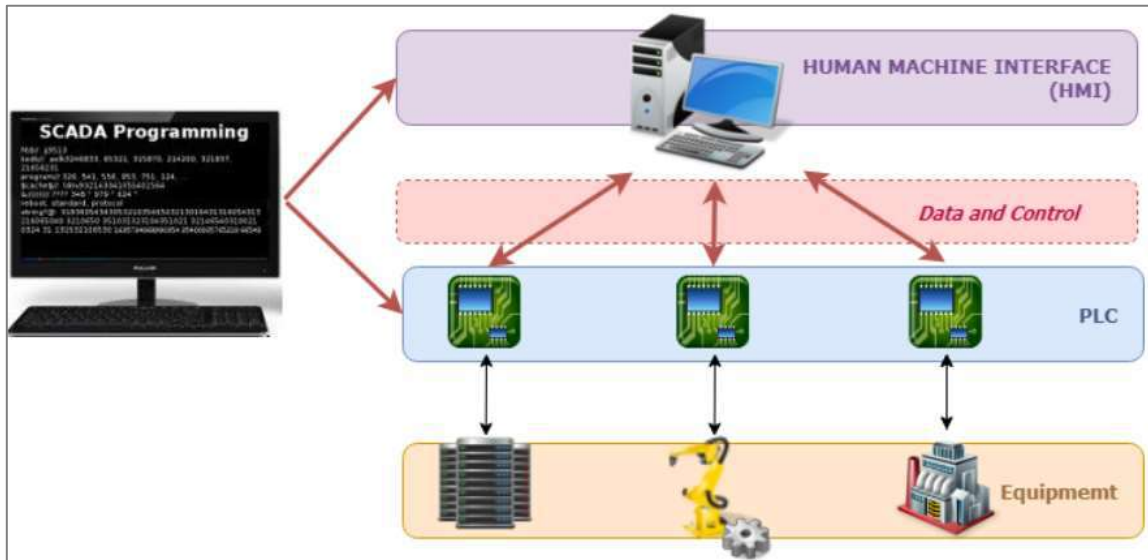
ERP სისტემა იძლევა ინფორმაციის სტანდარტიზაციას და ცენტრალიზაციას და გარდაქმნის მას სასარგებლო მონაცემებად, რაც ეხმარება ორგანიზაციის მენეჯმენტს მყისიერი გადაწყვეტილებების მიღებაში. ERP სისტემას შეუძლია გააუმჯობესოს ორგანიზაციის კონკურენტუნარიანობა ინფორმაციის დროულობისა და წვდომის სიმარტივის გზით [6,7].

### 4. SCADA სამეთვალყურეო სისტემები

SCADA (Supervisory Control And Data Acquisition) განისაზღვრება, როგორც სისტემა, რომელიც იძლევა საწარმოო პროცესების ზედამხედველობისა და კონტროლის საშუალებას. SCADA სისტემა შეიცავს პროგრამირებად ლოგიკურ კონტროლერებს (PLC) ან დისტანციურ ტერმინალურ ერთეულებს (RTU), რომელთა მეშვეობით მონაცემები მიიღება საწარმოო მოწყობილობებიდან. SCADA სისტემას დიდი გამოყენება აქვს გამოთვლით და საკომუნიკაციო ტექნოლოგიების, სამრეწველო პროცესების მონიტორინგისა და კონტროლის ავტომატიზებისათვის. SCADA სისტემა აგროვებს მონაცემებს, რომლებიც წარმოდგენილია წარმოების ოპერატორებთან წარმოების პროცესის მართვის, მონიტორინგისა და კონტროლისთვის - Human Machine Interfaces (HMI) [1,2,4].

ნახ.2-ზე ნაჩვენები მაგალითი ხსნის SCADA სისტემის არქიტექტურას - ყველა მოწყობილობა დაკავშირებულია ოპერატორთან, რომელიც ახორციელებს პროცესის მართვას უშუალოდ პროცესთან ურთიერთქმედებით HMI-ის მეშვეობით და აწვდის მენეჯერს შეგროვებულ ინფორმაციას. ამ მონაცემების დაკვირვებისთვის, HMI დაკავშირებულია მთავარ

ტერმინალთან, რომელიც შეიძლება იყოს PLC ან კონტროლერი, რომელიც იღებს საწარმოო დანადგარებიდან და მოწყობილობებიდან და ამუშავებს ინფორმაციას მონაცემთა გადაცემის ქსელის მეშვეობით. SCADA სისტემის საშუალებით შესაძლებელია წარმოების სრული პროცესის მონიტორინგი და კონტროლი რეალურ დროში, მოშორებულ ობიექტებზე [1,2,3].



ნახ.1. SCADA სისტემის ძირითადი კომპონენტები

### 5. MES სისტემა

MES (Manufacturing Execution System) არის ინტეგრირებული, კომპიუტერიზებული, ონლაინ სისტემა, რომელიც აერთიანებს ყველა მეთოდსა და ინსტრუმენტს, რომელიც საჭიროა წარმოების პროცესის განსახორციელებლად. MES, ზოგადად არის სპეციფიკური სისტემა თითოეული ტიპის საწარმოო ინდუსტრიისათვის.

MES სისტემები, როგორც წესი, ინტეგრირებულია უშუალოდ საწარმოო პროცესში მონაწილე მოწყობილობებთან და ხდება მონაცემების მომენტალურად მიწოდება. ERP სისტემები, არ არის ინტეგრირებული მოწყობილობებში, ის ინტეგრირებულია პროგრამულ უზრუნველყოფებთან, რომელიც უშუალოდ კავშირშია მოწყობილობებთან.

თანამედროვე SCADA სისტემები საშუალებას იძლევა უზრუნველყოს MES სისტემის ფუნქციის შესრულება - წარმოების ოპერაციებს მართვა და კონტროლი რეალურ დროში [4,5].

### 3. დასკვნა

ჩატარებული ანალიზის საფუძველზე ნათელია, რომ ERP და SCADA სისტემებს შორის ინტეგრაცია საწარმოო ორგანიზაციას საშუალებას აძლევს მიიღოს ინფორმაცია უშუალოდ საწარმოო პროცესიდან უწყვეტ რეჟიმში, რაც უზრუნველყოფს გადაწყვეტილების მიღების ოპტიმიზაციას და განაპირობებს საწარმოო პროცესებში ისეთი შემთხვევების პრევენციას როგორცაა - წარუმატებელი წარმოების ციკლი, პროცესის ცვლილებების დაგვიანება ან პროცესის დაგვიანებული ანალიზით გამოწვეული საწარმოო პროცესის შეფერხება-გაჩერების შემთხვევები. გამოვლენილი კიდევ ერთი გაუმჯობესება არის მონაცემების უსაფრთხოება და სანდოობა ოპერატორის მიერ თაღლითობის ან მონაცემთა ხელით შეყვანისას დაშვებულ

შეცდომებთან დაკავშირებით. კომუნიკაცია ეფექტური და საიმედოა, სისტემა ხდება ძირითადი ინსტრუმენტი ოპერატორებისა და მენეჯერების მიერ აღჭურვილობისა და პროცესის მართვისას გადაწყვეტილების მიღებისათვის. აღწერილი ინტეგრაციის გზით საწარმოო ინდუსტრიაში მიღებული შედეგები ძირითადად დადებითია, რადგან ორგანიზაციის საშუალებას ეძლევა ჰქონდეს მყისიერი უკუკავშირი მიმდინარე პროცესებზე, წარმოებული პროდუქტის ხარისხსა და ვადაზე. ERP და SCADA სისტემების ინტეგრაცია და მონიტორინგის სისტემის შემუშავება, რომელიც შეესაბამება წარმოების შესრულების კონტროლს, 10%-15%-მდე აუმჯობესებს წარმოების პროცესს და მნიშვნელოვან ღირებულებას მატებს ERP სისტემას როგორცაა - ინფორმაციული უსაფრთხოება, მონაცემების თანმიმდევრული მიწოდება, მონაცემების მიწოდება უწყვეტ რეჟიმში და გადაწყვეტილების რეალურ დროში მიღების შესაძლებლობა.

### ლიტერატურა – References – Література:

1. Inductive Automation, What is SCADA ? (2020). <https://www.inductiveautomation.com/resources/article/what-is-scada>; (15.11.22)
2. Phuyal S., Bista D., Izykowski J. (2020). Design and Implementation of Cost Efficient SCADA System for Industrial Automation. Internat. Jour. of Engineering and Manufacturing;
3. Church P., Mueller H., Ryan C., Gogouvitis S.V., Goscinski A., Tari Z. (2017). Migration of a SCADA system to IaaS clouds - a case study. Journal of Cloud Computing: Advances, Systems and Applications, 6-11
4. Sharma R., Navtake S., Chaudhari H.P. (2017). Automatic Material Storage and Retrieval System using PLC and SCADA. Internati. Jour. of Computational Intelligence Research. Vol.13, # 6, 1361-1365;
5. Aamir Shahzad, Young-Gab Kim. (2018). Secure SCADA-IoT Platform for Industrial Automation and Control: A Collaborative Communication Designed Model. Conf. presented at Symmetry 2017—The First Internat. Conf. on Symmetry, Barcelona, Spain;
6. Madanhire I., Mbohwa C. (2016). Enterprise resource planning (ERP) in improving operational efficiency: Case study. Procedia CIRP, 40 p;
7. Camarillo A., Ríos J., Althoff K.-D. (2018). Product lifecycle management as data repository for manufacturing problem solving, Materials. <http://doi.org/10.3390/ma11081469>. (14.11.22)

*(სტატია მიღებულია 18.11.2022)*

## INTEGRATION OF ERP AND SCADA SYSTEMS INTO INDUSTRIAL PRODUCTION AND THEIR ANALYSIS EFFICIENCY

Khuchua Tamar  
Georgian Technical University  
tamunaxuchua@yahoo.com

### Summary

The article reflects the results of integration of ERP (Enterprise Resource Planning) and SCADA (Supervisory Control and Data Acquisition) systems for the manufacturing industry. For modern



production organizations, it is important to have such an automation scheme, through which it will be possible to receive and process production data in real time. Today, in many production organizations, ERP and SCADA (as well as other production process automation programs) systems function independently of each other, the process of providing data from the SCADA system is, in most cases, carried out manually, which leads to late delivery of data needed for decision-making and reduces the reliability of information. Through the integration, the ERP system has access to the SCADA database, production operational data, the data is transferred through the integration in real time, processed and stored in the unified database of the organization's ERP system. The ability to receive and analyze operational data in real time helps both production operators, technical professionals, and organization managers to continuously monitor the production process and make quick decisions, as well as minimize the time spent on manual data input to the ERP system. The paper aims to demonstrate the benefits that can be provided by the integration of ERP and SCADA systems.

*(Received 18.11.2022)*

## **ИНТЕГРАЦИЯ ERP И SCADA СИСТЕМ В ПРОИЗВОДСТВО И АНАЛИЗ ИХ ЭФФЕКТИВНОСТИ**

Хучуа Т.

Грузинский Технический Университет  
tamunaxuchua@yahoo.com

### **Резюме**

Рассматриваются интеграция систем ERP (планирование ресурсов предприятия) и SCADA (диспетчерское управление и сбор данных) для производственной отрасли. Для современных производственных организаций важно иметь такую схему автоматизации, благодаря которой можно будет получать и обрабатывать производственные данные в режиме реального времени. На сегодняшний день во многих производственных организациях системы ERP и SCADA (а также другие программы автоматизации производственных процессов) функционируют независимо друг от друга, процесс предоставления данных из системы SCADA в большинстве случаев осуществляется вручную, что приводит к задержке предоставления данных, необходимых для принятия решений, и снижает достоверность информации. Через интеграцию ERP-система имеет доступ к базе данных SCADA, производственным оперативным данным, данные через интеграцию передаются в режиме реального времени, обрабатываются и хранятся в единой базе данных ERP-системы организации. Возможность получать и анализировать оперативные данные в режиме реального времени помогает операторам производства, техническим специалистам и руководителям организаций постоянно контролировать производственный процесс и принимать быстрые решения, а также минимизировать затраты времени на ручной ввод данных в ERP-систему. Целью статьи является демонстрация преимуществ, которые может дать интеграция систем ERP и SCADA.

*(Поступила в редакцию 18.11.22)*

# RPA პროცესების რობოტიზაციის და ხელოვნური ინტელექტის ტექნოლოგიის ინსტრუმენტები

ირმა ბერძენიშვილი  
საქართველოს ტექნიკური უნივერსიტეტი  
i.berdzenishvili@bdo.ge

## რეზიუმე

განხილულია RPA-პროცესების რობოტიზაციისა და AI (ხელოვნური ინტელექტის ტექნოლოგია) ინსტრუმენტები, რომლებიც გვეხმარება განვახორციელოთ მაღალი დონის ინტელექტუალური ავტომატიზაციები. RPA ტექნოლოგია თავისი ფუნქციონალით ძირითადად განკუთვნილი განმეორებადი და რობოტული ქმედებების დასაავტომატიზებლად, თუმცა ხელოვნური ინტელექტის ტექნოლოგიის საშუალებით როგორც არის მანქანური სწავლება, NLP (Natural Language Processing) – ბუნებრივი ენის დამუშავების ტექნოლოგიები, კოგნიტური ავტომატიზაცია შესაძლებელია არა მხოლოდ წესებზე დაფუძნებული ბიზნეს პროცესების, არამედ არასტრუქტურული შემომავალი ინფორმაციის წაკითხვა, გაანალიზება და შესაბამისი გადაწყვეტილების მიღებაც.

**საკვანძო სიტყვები:** RPA-პროცესი, რობოტიზაცია, ხელოვნური ინტელექტი. არასტრუქტურული ინფორმაცია. გადაწყვეტილების მიღება.

## 1. შესავალი:

### RPA-პროცესების რობოტიზაცია და მისი გამოყენების მაგალითები

რუტინული ამოცანები, როგორცაა სამუშაო პროცესის დამუშავება, ავტომატური ელფოსტის გაგზავნა, მიღება, მონაცემთა შეგროვება ონლაინ წყაროებიდან, ასევე ინვენტარის ავტომატური შევსება და კიდევ სხვა მრავალი პროცესი შეიძლება განხორციელდეს ციფრული რობოტების მიერ, რომელიც იგება RPA- ტექნოლოგიაში [1].

RPA-ს მთავარი იდეაა გააკეთოს ის, რაც დეველოპერმა დაავალა და დიდად არის დამოკიდებული ადამიანურ რესურსზე მაშინ როცა AI-ის დანიშნულებაა ქონდეს ფიქრის და სწავლის შესაძლებლობა – და არ ელოდოს ბრძანებას ადამიანისგან თუ რა უნდა მოიმოქმედოს. თუმცა, ისინი არ არიან ურთიერთგამომრიცხავი. RPA-ს და AI-ს შეუძლიათ იმუშაონ ერთობლივად ინტელექტუალური ბოტების საშუალებით, მაგალითად, RPA ინსტრუმენტი „Automation Anywhere“ ბოტებს საშუალებას აძლევს ისწავლოს თვითმმართველობის „IQ Bot“-ის მეშვეობით. ტრადიციული RPA ტექნიკის AI-სთან ინტეგრაციით, ბოტს შეუძლია შეასრულოს ისეთი ამოცანები, როგორცაა არასტრუქტურული მონაცემების ამოცნობა, ადამიანის მითითებებიდან სწავლა და მრავალი სხვ.

UiPath's Computer Vision ასევე აერთიანებს RPA-ს გამოყენებას AI-სთან, რადგან რობოტი ამოიცნობს და ურთიერთქმედებს ეკრანზე არსებულ ველებთან და კომპონენტებთან (მათ შორის PDF-ებსა და სურათებთან) და აქვს ტექსტის ანალიზის უნარი. UiPath ასევე საშუალებას აძლევს ავტომატიზაციას გაიგოს მნიშვნელობა ბუნებრივი ენის შეყვანიდან, რაც უზრუნველყოფს მომხმარებლისა და თანამშრომლების ურთიერთქმედების უკეთეს გამოცდილებას.

AI-სთან ასიმილაციის გზით, RPA-ს შეუძლია შეასრულოს უფრო მეტი კოგნიტური ამოცანები, ვიდრე მარტივი ნაბიჯ-ნაბიჯ პროცესები, რომლებიც მოითხოვს ძალიან მცირე ფიქრს ან გადაწყვეტილების მიღებას. ხელოვნური ინტელექტი საშუალებას აძლევს ავტომატიზაციას გადაიზარდოს რაღაც უფრო რთულში, და მოახდინოს უფრო მეტი „ადამიანური“-ს ინტეგრაცია სამუშაო პროცესში. ხელოვნური ინტელექტის დაახლოება RPA-სთან საშუალებას მისცემს ბიზნესებს ავტომატიზაცია მოახდინონ უფრო კომპლექსურ პროცესებზე, ამ პროცესებში იდეების ინტეგრირებით, რაც დაეხმარება ადამიანებს იმუშაონ უფრო მოქნილად და სწრაფად ვიდრე აქამდე.

ეკრანის სკრეფერები(screen scrappers) წარმოადგენდა რობოტული პროცესის ავტომატიზაციის ძირეულ ტექნოლოგიას და ეს არ იყო იდენტური ხელოვნური ინტელექტის შექმნისა. მიუხედავად იმისა, რომ რობოტული ავტომატიზაცია ეხება ადამიანის აქტივობის მიბაძვას მომხმარებლის ინტერფეისის საშუალებით, ხელოვნური ინტელექტი მიზნად ისახავს ადამიანის აზროვნების პროცესის მიბაძვას [2].

## 2. ძირითადი ნაწილი

### 2.1. AI ტექნოლოგია - CA (Cognitive Automation)

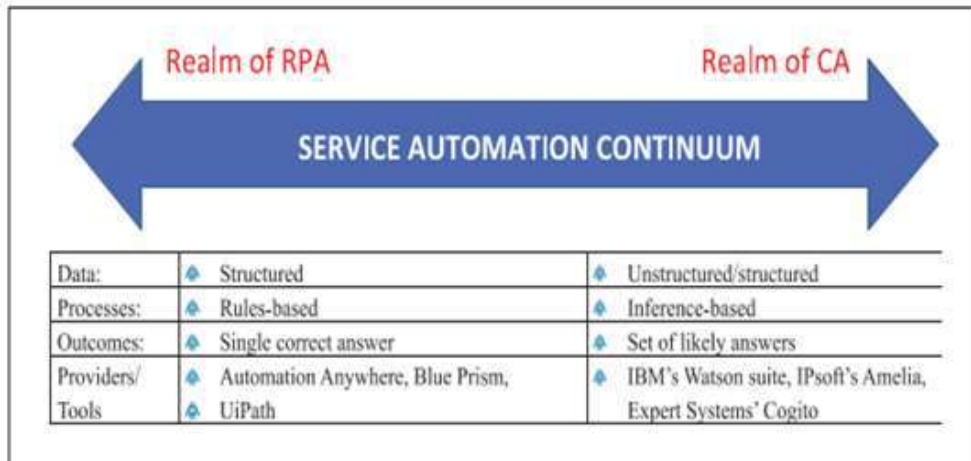
RPA არის ტექნოლოგია რომელიც ადამიანისგან ანთავისუფლებს რობოტულ საქმეს, ხოლო კოგნიტური ავტომატიზაცია ადამიანურ ინტელექტის რობოტში ჩადებას გულისხმობს. ინტელექტუალური ტექნოლოგიებისა (CA) ეპოქაში, ორგანიზაციები ცდილობენ თავიანთი ოპერაციებისა და პროცესების მთავრად კოგნიტური ავტომატიზაციების ტექნოლოგიებს. ეს არის გზა წარმატების გარანტორი, დღევანდელ კონკურენტუნარიანი და გლობალიზებული ბიზნეს გარემოში [3,6].

კოგნიტური ავტომატიზაციის დროს გამოიყენება AI ტექნოლოგიები: დოკუმენტების პროცესინგი, NLP, ხმოვანი ჩანაწერების გარდაქმნა, მანქანური სწავლება, რითაც ავტომატიზაციები გადადის შემდგომი განვითარების დონეზე

მაგალითად: კოგნიტური ავტომატიზაციას იყენებს AI შესაძლებლობებს, როგორც არის OCR (Optical Character Recognition), რომ ამოიღოს ტექსტი დოკუმენტიდან, ხოლო NLP-(natural Language Processing) ბუნებრივი ენის დამუშავება, რომ გაიგოს ისეთი პუნქტები, როგორც არის მომხმარებლების, ინვოისისთვის შესაბამისი ელემენტები და შემდგომ უკვე ორგანიზება და კლასიფიკაცია გაუკეთოს შესაბამის პროცესის ფლოუს ველებში. კოგნიტური ავტომატიზაციას შეუძლია AI-ხელოვნურ ინტელექტის გამოყენებით ასევე მხარდაჭერა გაუწიოს კიდევ უფრო მეტი ტიპის გადაწყვეტილებებს. მაგალითად: კოგნიტური ავტომატიზაციის აპლიკაციამ შეიძლება გამოიყენოს მანქანური სწავლების ალგორითმი, რათა დაადგინოს საპროცენტო განაკვეთი შესაბამისი სესხის მოთხოვნაზე.

ტრადიციულად RPA ახდენს ავტომატიზაციებს, რომლებიც დაფუძნებულია სტრუქტურულ მონაცემებზე. კოგნიტური RPA კიდევ უფრო წინ მიდის და შესაძლებლობას აძლევს კომპანიებს განახორციელონ ავტომატიზაციები, რომლებიც შეიცავენ არასტრუქტურულ მონაცემების წყაროებს, მათ შორის დასკანერებული დოკუმენტები, ელექტრონული მეილები, წერილები და ხმოვანი ჩანაწერები. მთავარი ღირებულება კოგნიტური ავტომატიზაციების არის შესაძლებლობა დაავტომატიზონ უფრო რთული, ნაკლებად წესებზე დაყრდნობილი პროცესები.

მთლიანობაში, რომ შევჯავშოთ, კოგნიტური RPA -ს მნიშვნელოვნება მაშინ წამოიწევს წინ, როდესაც საჭიროა არასტრუქტურული მონაცემების დამუშავება, ხოლო RPA-ი ტექნოლოგიის გამოყენება ხდება მაშინ, როდესაც სტრუქტურული მონაცემები წესებზე დაფუძნებული პროცესებია (ნახ.1) [1]. ისეთი დოკუმენტებისა კი, როგორცაა მეილი ან თავისუფალი ტექსტი, შემდგომ ხდება აღნიშნულ წაკითხული მონაცემების ავტომატიზაცია [5].



ნახ 1. ავტომატიზაცია (RPA და კოგნიტური ავტომატიზაცია) [1]

ქვემოთ მოცემული რამდენიმე მაგალითი არის ჩამონათვალი თუ რა შეუძლია აღნიშნულ ტექნოლოგიას [2]:

- Optical Character Recognition (OCR) და გამოსახულების იდენტიფიკაცია;
- შემავალი ინფორმაციის ამოქსტრაქტირება;
- ტექსტის ანალიტიკა;
- Sentiment analysis;
- კატეგორიზაცია;
- კლასიფიკაცია;
- ხმის ამოცნობა (Voice recognition).

## 2.2. კოგნიტური RPA-ის გამოყენების მაგალითები

ამ ტექნოლოგიების გამოყენება შესაძლებელია სხვადასხვა შემთხვევაში. პირველი მაგალითია RPA და კოგნიტური შესაძლებლობების შერწყმა ჩეთბოტებისთვის, რაც მომხმარებელს აგრძნობინებს, რომ იგი უგზავნის მესიჯებს ადამიანს. მეორე არის ხმოვანი ბოტების შექმნა სატელეფონო საუბრებისთვის. ბევრი მიმოწერა ელ. ფოსტის საშუალების ასევე შეიძლება იქნას ავტომატიზირებული. AI-ზე დაფუძნებულ ავტომატიზაციას შეუძლია თვალყური ადევნოს ტრიგერებს, და განსაზღვროს როდის არის დრო გაიგზავნოს ელ.წერილი, შემდეგ შეადგინოს და გააგზავნოს მთელი მიმოწერა.

დღეის მდგომარეობით მომხმარებლები ურთიერთობენ თქვენს ორგანიზაციასთან სხვადასხვა შეხების წერტილების და არხების გამოყენებით – ჩატი, ინტერაქტიული IVR, აპები, შეტყობინებები და სხვა. როდესაც თქვენ აღნიშნულ არხებს დაუმატებთ RPA-ს, შეგიძლიათ მისცეთ მომხმარებლებს მეტის გაკეთების საშუალება, ცოცხალი ადამიანის დახმარების გარეშე.

კოგნიტური RPA უნარები საშუალებას აძლევს ავტომატიზირებულ სისტემას გაიგოს მომხმარებლის განზრახვა, გააცნობიეროს მომხმარებელთან დაკავშირებული არასტრუქტურირებული მონაცემები, პროგნოზირების შემდეგ შეასრულოს მოთხოვნა ბექენდში. ხელოვნური ინტელექტისა და კოგნიტიურ ავტომატიზაციას ასევე შეუძლია თვალი ადევნოს მომხმარებლის მიერ სრულად განვლილ გზას და ჩაერთოს მასში მეტი ეფექტურობის მისაღებად.

განვიხილოთ საბანკო ჩეთბოტის მაგალითი, რომელიც ახდენს *ახალი საბანკო ანგარიშის გახსნის პროცესის* ავტომატიზაციას. თქვენს მომხმარებელს შეუძლია სთხოვოს ჩეთბოტს ონლაინ ფორმა, შეავსოს იგი და ატვირთოს დოკუმენტები. ფორმის პირველად დამუშავებაში შესაძლებელია ჩაერთოს რობოტი რათა მოახდინოს საკრედიტო ქულის შემოწმება და მომხმარებლის მართვის მოწმობიდან ან პირადობის მოწმობიდან მონაცემების ამოღება OCR-ის გამოყენებით. გამონაკლისების დამუშავება კოგნიტურ RPA-სთან, თუ სისტემა შეარჩევს გამონაკლისს – როგორცაა შეუსაბამობა მომხმარებლის სახელს შორის ფორმასა და პირადობის დამადასტურებელ დოკუმენტში, მას შეუძლია გადასცეს ის თანამშრომლებს შემდგომი დამუშავებისთვის. სისტემა იყენებს *მანქანურ დასწავლას*, რათა აკონტროლოს და გაიგოს, თუ როგორ ამოწმებს თანამშრომელი კლიენტის ვინაობას. შემდეგ ჯერზე, ის შეძლებს იგივე სცენარის დამუშავებას თავად ადამიანის ჩარევის გარეშე.

კიდევ ერთი მაგალითი შეიძლება იყოს *გადასახადების პროცესი*. თქვენს ავტომატიზაციას შეუძლია გამოიყენოს OCR ტექნოლოგია და *მანქანური დასწავლა* იმ ინვოისების დასამუშავებლად, რომელთაც დიდი დრო სჭირდებოდათ. მანქანური დასწავლა ეხმარება რობოტს გახდეს უფრო ზუსტი და ისწავლოს გამონაკლისები და შეცდომები მანამ, სანამ ადამიანის ჩარევა მცირე დოზით არის საჭირო.

ორგანიზაციებს სჭირდება როგორც ტრადიციული RPA, ასევე მოწინავე შემეცნებითი მეთოდები პროცესის ავტომატიზაციის დონის ასამაღლებლად, რადგან არსებობს სტრუქტურირებული და არასტრუქტურირებული მონაცემები, რომლებიც ხელს უწყობს მათ პროცესებს.

ხელოვნური ინტელექტის კომბინაციაში, ორგანიზაციებს შეუძლია იმუშაოს არა მხოლოდ უფრო სწრაფად, არამედ უფრო მოქნილად უკეთესი ეფექტურობის, ხარჯების დაზოგვისა და მომხმარებელთა კმაყოფილების მიზნების მისაღწევად.

კოგნიტურ ავტომატიზაციას შეუძლია გამოიყენოს ხელოვნური ინტელექტის ტექნიკა ისეთ გარემოებებში, სადაც საჭიროა დოკუმენტების დამუშავება, ბუნებრივი ენა და ხმა, რაც ავტომატიზაციას გადაიყვანს შემდეგ დონეზე.

### 3. დასკვნა

ინდუსტრიები დღევანდელ საჭიროებებიდან გამომდინარე ეძიებენ უფრო მეტად ინტელექტუალურ, ინოვაციურ და განახლებული შესაძლებლობების მქონე რობოტიზაციის გადაწყვეტებს, რათა შეძლონ გაუმკლავდნენ არასტრუქტურირებულ დოკუმენტებზე მუშაობას, გარკვეულ გადაწყვეტილების მიღებას ბიზნეს პროცესებში. არსებული სისტემების უფრო მეტად განვითარება საშუალებას მისცემთ განახორციელონ მაღალი დონის პროცესების ავტომატიზაცია და უფრო მეტი ღირებულება შეუქმნან დაინტერესებულ პირებს, მომხმარებლებს.

ხელოვნურის ინტელექტის განვითარებამ და ჩართვამ სხვა დონეზე გადაიყვანა RPA-ს ევოლუცია, რომელიც უნდა იყოს კოორდინირებული სხვა ტექნოლოგიებთან რეალურ სამყაროში .

### ლიტერატურა – References – Литература:

1. Kam K.H. Ng, Chun-Hsien Chen, C.K.M. Lee, Jianxin (Roger) Jiao, Zhi-Xin Yang. (2021). A systematic literature review on intelligent automation: Aligning concepts from theory, practice, and future perspectives, Advanced Engineering Informatics. Vol.47, Jan., 101246
2. Ribeiro J., Lima R., Eckhardt T., Paiva S. (2021). Robotic Process Automation and Artificial Intelligence in Industry 4.0 – A Literature review, ScienceDirect. Procedia Computer Science 181 (2021) 51–58
3. Kumar Vishnoi S., Bagga T., Tripathi A. (2019). Intelligent Automation, Planning & Implementation: A Review of Constraints. Intern.Jour.on Emerging Technologies 10(1a): 174-178.
4. Lacity M., Willcocks L., Gozman D. (2021). Influencing information systems practice: The action principles approach applied to robotic process and cognitive automation, Journal of Information Technology 1–25,
5. Lacity M., Willcocks L. (2021). Becoming Strategic with Intelligent Automation, University of Minnesota. USA
6. How Robotics and Cognitive Automation will transform the insurance Industry. (2020). Deloitte

*(სტატია მიღებულია 1.12.2022)*

## RPA – ROBOTIC PROCESS AUTOMATION AND AI TOOLS

Berdzenishvili Irma  
Georgian Technical University  
i.berdzenishvili@bdo.ge

### Summary

The article reviews RPA- (Robotic Process Automation) technology and AI (artificial intelligence technology) tools that help us implement high-level intelligent automations. RPA with its functionality is mainly intended for automating repetitive and robotic actions, however, through artificial intelligence components such as machine learning, NLP (Natural Language Processing) cognitive automation, we are able to read, analyze and make appropriate decisions not only for rule-based, structured business processes, but also for unstructured data.

(Received 1.12.2022)

## ИНСТРУМЕНТЫ АВТОМАТИЗАЦИИ РОБОТИЗИРОВАННЫХ ПРОЦЕССОВ (RPA) И ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Бердзенишвили И.  
Грузинский Технический Университет  
i.berdzenishvili@bdo.ge

### Резюме

Рассматриваются инструменты RPA (Robotic Process Automation) и AI (технология искусственного интеллекта), которые помогают нам внедрять интеллектуальную автоматизацию высокого уровня. RPA с его функциональностью в основном предназначен для автоматизации повторяющихся и роботизированных действий, однако с помощью компонентов искусственного интеллекта, таких как машинное обучение, когнитивная автоматизация NLP (обработка естественного языка), мы можем читать, анализировать и принимать соответствующие решения не только для правил. на основеструктурированных бизнес-процессов, но и для неструктурированных данных.

*(Поступила в редакцию 1.12.22)*

## მანქანური დასწავლის გამოყენება თანამედროვე ციფრულ ტექნოლოგიებში

ნინო თოფურია, თინათინ ხატიაშვილი  
საქართველოს ტექნიკური უნივერსიტეტი  
nino.topuria@gtu.ge, khatiashvili.t@gtu.ge

### რეზიუმე

განხილულია მანქანური დასწავლის, კერძოდ ღრმა დასწავლის გამოყენების პრაქტიკული მაგალითი. წარმოდგენილია Azure ღრუბლოვანი სერვისების ბაზაზე დაპროექტებული სისტემა, რომელიც იყენებს კომპიუტერულ ხედვას. კომპიუტერული ხედვა, კოგნიტური სერვისების შემადგენელი ერთ-ერთი სერვისია, რომელიც კომპიუტერს შესაძლებლობას აძლევს ადამიანის მსგავსად აღმოაჩინოს და ამოიცნოს ობიექტები.

**საკვანძო სიტყვები:** ხელოვნური ინტელექტი. ღრუბლოვანი გამოთვლები. მანქანური დასწავლა. ღრმა დასწავლა. კომპიუტერული ხედვა.

### 1. შესავალი

დღევანდელ ბიზნეს სამყაროში ტექნოლოგიები უმნიშვნელოვანეს როლს ასრულებს. ამასთან, იგი არა მარტო ბიზნესს, არამედ მომხმარებელსაც ძალიან ადგება. ტექნოლოგიების გამოყენებით, ბიზნესი თავის მომსახურებას უფრო მრავალფეროვანსა და მარტივს ხდის, რაც მომხმარებლისთვის ძალიან მნიშვნელოვანი და კომფორტულია.

ხელოვნური ინტელექტი (AI), ღრმა დასწავლა (DL) და მანქანური დასწავლა (ML) არის ტერმინები, რომლებიც განსაზღვრავს მოწინავე ტექნოლოგიურ სისტემებსა და კონცეფციებს. ეს კონცეფციები კი ახდენენ რევოლუციას მცირე, საშუალო და მსხვილ ბიზნესში. ინოვაციები მოიცავს როგორც საწარმოების ოპერაციებსა და სტრატეგიებს, ასევე სამუშაო პროცესებსა და პროექტებს. ხელოვნური ინტელექტი კომპიუტერული მეცნიერების ფართო სპექტრია, რომელიც ეხება ჭკვიანი მანქანების შექმნას. მათ შეუძლია შეასრულოს ამოცანები, რომლებიც, როგორც წესი, მოითხოვს ადამიანის ინტელექტს. ხელოვნური ინტელექტის მანქანებს შეუძლია მოდელირება და ადამიანის გონებრივ შესაძლებლობაზე გაუმჯობესება, თვითმართვადი მანქანების შემუშავებიდან ჭკვიანი ასისტენტების ჩათვლით.

ხელოვნური ინტელექტის განსაზღვრისას, როგორც უბრალოდ „ინტელექტუალური მანქანების მშენებლობა“ არის ის, რომ ის რეალურად არ ხსნის რა არის AI და რა ხდის მანქანას ინტელექტუალურს. AI არის ინტერდისციპლინარული მეცნიერება მრავალი მიდგომით, მაგრამ მანქანური დასწავლისა და ღრმა დასწავლის მიღწევები ახდენს პარადიგმის ცვლილებას ტექნიკური ინდუსტრიის პრაქტიკულად ყველა სექტორში.

ღრმა დასწავლა არის მანქანური დასწავლის ტიპი, რომელიც დაფუძნებულია ხელოვნურ ნეირონულ ქსელებზე. დასწავლის პროცესს უწოდებენ ღრმას, რადგან ხელოვნური ნეირონული ქსელების სტრუქტურა შედგება რამდენიმე ფენისაგან. ესენია შეტანის, გამოტანის და დაფარული. თითოეული ფენა შეიცავს ერთეულებს, რომლებიც შეტანის მონაცემებს გარდაქმნის ინფორმაციად. ეს ინფორმაცია მომდევნო ფენას შეუძლია გამოიყენოს პროგნოზირების კონკრეტული ამოცანისთვის. აღწერილი სტრუქტურის წყალობით კომპიუტერს შეუძლია ისწავლოს საკუთარი მონაცემების დამუშავების გზით.

მანქანური დასწავლა არის ხელოვნური ინტელექტის დახმარებით კომპიუტერული მოწყობილობებსათვის ისეთი შესაძლებლობების მინიჭება, რომლებიც მოწყობილობას აძლევს უნარს დაიწყოს ადამიანივით დასწავლა და მოქმედება, ხელახალი დაპროგრამების გარეშე.

მანქანური დასწავლა იყენებს სხვადასხვა ალგორითმს მათემატიკური მოდელების შესაქმნელად და პროგნოზების გასაკეთებლად ისტორიული მონაცემების ან ინფორმაციის გამოყენებით. ამჟამად, ის გამოიყენება სხვადასხვა ამოცანებისთვის, როგორცაა გამოსახულების ამოცნობა, მეტყველების ამოცნობა, ელექტრონული ფოსტის გაფილტვრა, Facebook-ის ავტომატური თეგირება, რეკომენდატორის სისტემის შექმნა და მრავალი სხვა.

## 2. ძირითადი ნაწილი

კომპიუტერული ხედვა ხელოვნური ინტელექტის ქვეჯგუფია, რომელიც კომპიუტერს აძლევს შესაძლებლობას დაინახოს ანუ აღმოაჩინოს, ამოიცნოს და მონიშნოს ობიექტები ისევე, როგორც ეს ადამიანს შეუძლია. როდესაც ადამიანი ხედავს სურათს, ის აცნობიერებს არა მხოლოდ სურათზე არსებულ მთავარ ობიექტს, არამედ სურათის ირგვლივ არსებულ გარემოს.

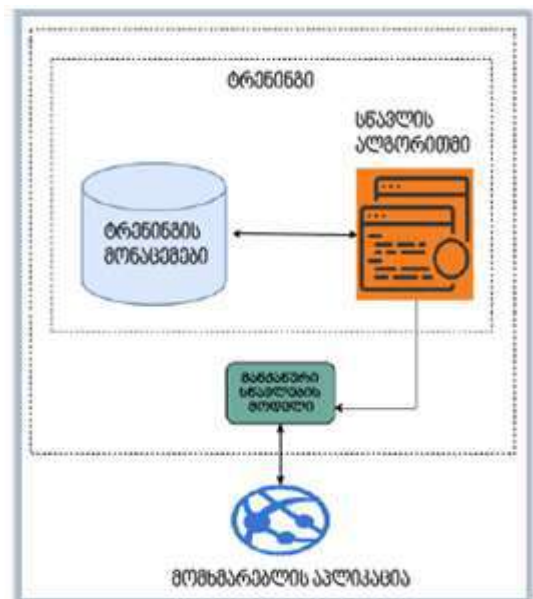
კომპიუტერული ხედვის დანერგვა აუცილებელია მრავალი პრობლემის გადასაჭრელად, ეს იქნება საზოგადოებრივი ტრანსპორტი, თუ სხვადასხვა ტიპის ადამინთა თავშეყრის ადგილი. სასურველია არსებობდეს ისეთი სერვისები ან აპლიკაციები, რომლებიც გამოდგება სხვადასხვა ტიპის ორგანიზაციების კონტროლის, სიტუაციის მართვისა და სივრცის მასშტაბირებისთვის.

მანქანურ დასწავლას აქვს შესაძლებლობა დაინერგოს Azure სივრცეში, მასში მოხდეს კოგნიტური სერვისების (Cognitive Services), კონკრეტულად კი კომპიუტერული ხედვის ჩაშენება და ზემოხსენებული პრობლემების გადაჭრა.

მაიკროსოფტის ღრუბლოვანი Azure პლატფორმაში ჩაშენებული სხვადასხვა სერვისის ცალსახად მოიცავს თვითმომსახურების მანქანური დასწავლის ობიექტებს (MLO). მათ შეუძლია ავტომატურად განათავსოს, დააკვირდეს და მოახდინოს მანქანური დასწავლის მოდელების კონფიგურირება ახალი სისტემის შესაქმნელად. იმისთვის რომ, სერვისს შეეძლოს მანქანური დასწავლის მხარდაჭერა, მას ცალსახად უნდა ჰქონდეს შემდეგი ძირითადი კომპონენტები (ნახ.1).

**ტრენინგის მონაცემები:** არის საწყისი მონაცემები, რომლებიც გამოიყენება მანქანური დასწავლის მოდელების მოსამზადებლად. შემდეგ მონაცემთა ნაკრები მიეწოდება მანქანური დასწავლის ალგორითმებს, რათა ასწავლოს პროგნოზის გაკეთება ან შეასრულოს სასურველი დავალება. მანქანური დასწავლის მოდელი განსაზღვრავს გამოსავალს, რომელიც მიიღება მისი ალგორითმის გაშვების შემდეგ შეგროვებულ მონაცემებზე დაყრდნობით.

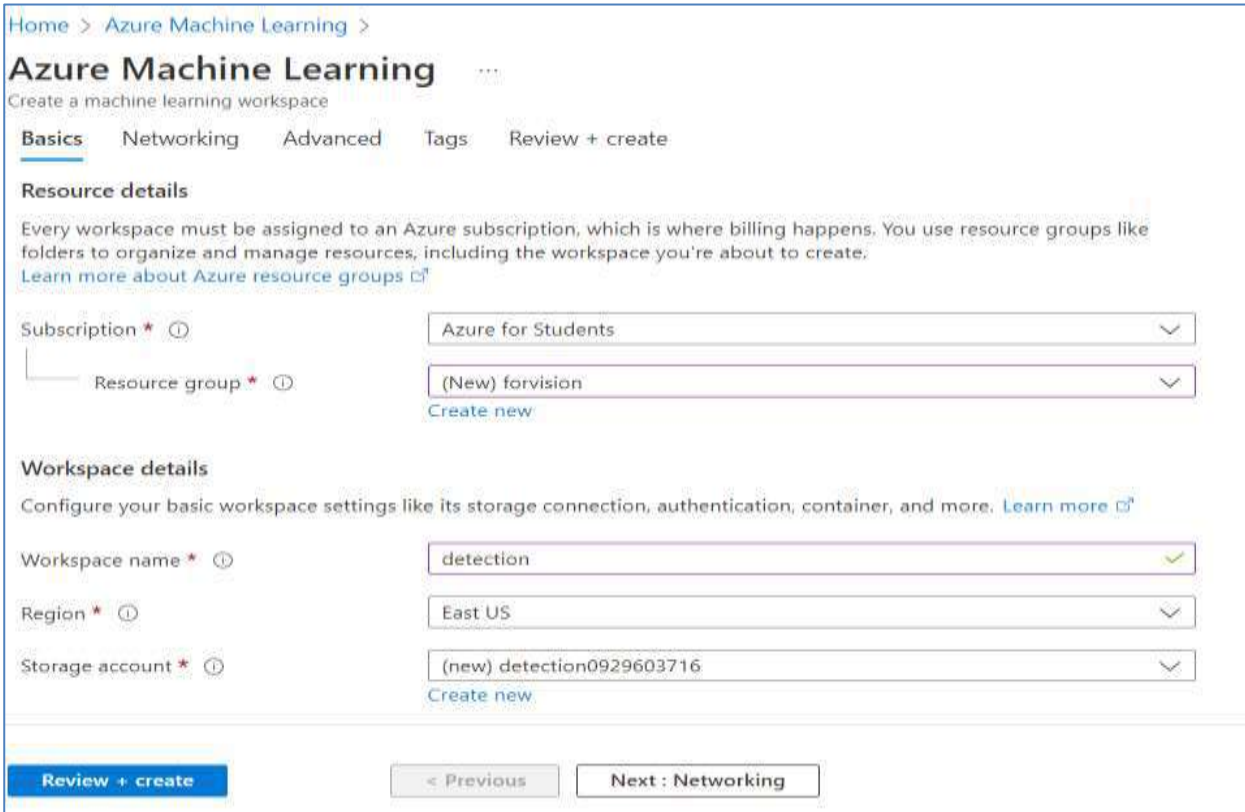
მნიშვნელოვანია, მოხდეს ისეთი მოდელის შერჩევა, რომელიც შეესაბამება დავალებას. თანამედროვე ტექნოლოგიების განვითარებასთან ერთად, მეცნიერ-ინჟინერთა გუნდმა შეიმუშავა სხვადასხვა მოდელი ისეთი ამოცანების ამოსახსნელად, როგორცაა მეტყველების და გამოსახულების ამოცნობა, პროგნოზირება და ა.შ.



ნახ. 1. მანქანური დასწავლის კომპონენტები

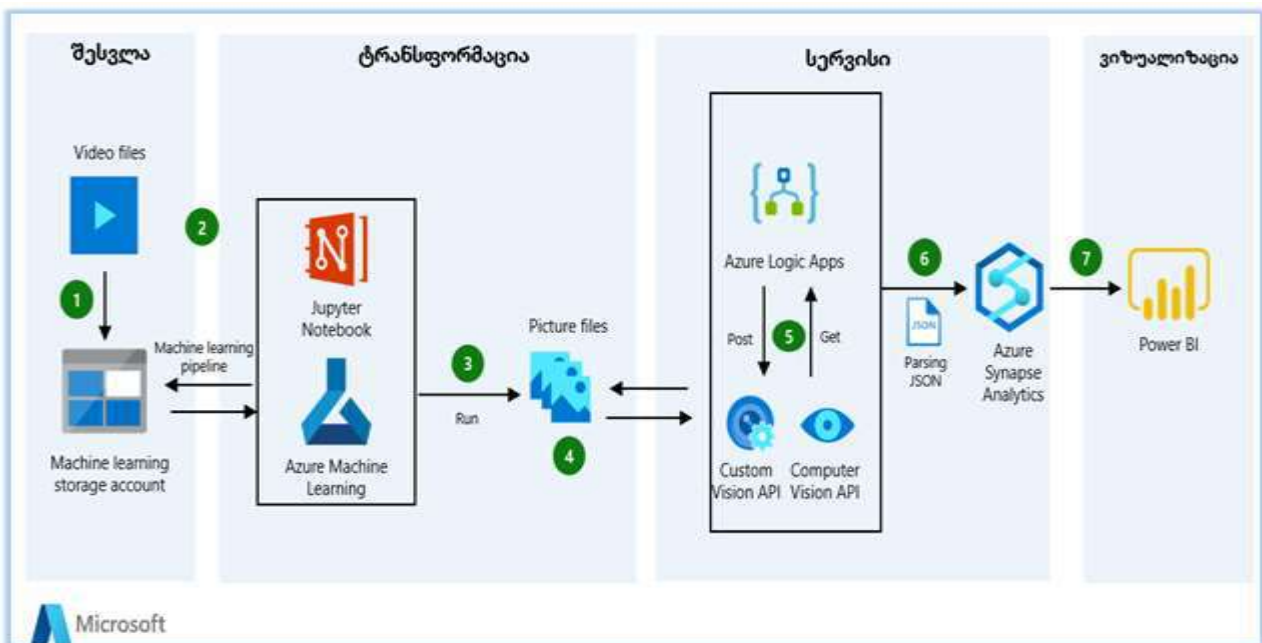


მაგალითის სახით განხილულია სისტემის დაპროექტების ეტაპები, რომელიც უზრუნველყოფს კონკრეტულ სურათზე (ფოტოზე) არსებული ობიექტების ამოცნობას და გაანალიზებას (ნახ. 2). ამისთვის, პირველ რიგში საჭიროა მანქანური დასწავლის სერვისის შექმნა Azure პორტალზე (Azure Machine Learning).



ნახ. 2. მანქანური დასწავლის სერვისი Azure-ის გარემოში

შემდეგ ეტაპზე საჭიროა კომპიუტერული ხედვის სერვისის შექმნა და Azure Machine Learning-ის ჩაშენება. არქიტექტურა შემდეგნაირად გამოიყურება (ნახ. 3).



ნახ. 3. მანქანური დასწავლის არქიტექტურა

ბოლო ეტაპზე ხდება აღნიშნული სერვისის გატესტვა. შედეგად ვიღებთ გამოსახულების სემენტაციას. ეს პროცესი განსაზღვრავს ობიექტის კლასის რომელი პიქსელი გვხვდება სურათში. გამოსახულების სემანტიკური სემენტაცია მონიშნავს ყველა პიქსელს, რომელიც ეკუთვნის ამ თეგს, მაგრამ არ განსაზღვრავს თითოეული ობიექტის საზღვრებს. სანაცვლოდ, ობიექტის ამოცნობა არ დაყოფს ობიექტს, მაგრამ ნათლად განსაზღვრავს თითოეული ცალკეული ობიექტის ადგილმდებარეობას „ყუთით“ (მართკუთხედი). სემანტიკური სემენტაციის შეთავსება ობიექტების გამოვლენასთან იწვევს ინსტანციის სემენტაციას, რომელიც ჯერ აღმოაჩენს ობიექტის მაგალითებს და შემდეგ თითოეულს ანაწილებს აღმოჩენილ უჯრებში ( ნახ. 4).



ნახ. 4

### 3. დასკვნა

დღევანდელ ბიზნეს სამყაროში ტექნოლოგიები ასრულებს უმნიშვნელოვანეს როლს. ამასთან, იგი არა მარტო ბიზნესს, არამედ მომხმარებელსაც ძალიან ადგება. ტექნოლოგიების გამოყენებით, ბიზნესი თავის მომსახურებას უფრო მრავალფეროვანს და მარტივს ხდის, რაც მომხმარებლისთვის ძალიან მნიშვნელოვანი და კომფორტულია.

#### ლიტერატურა – References – Литература:

1. Duc T.L., Leiva R.G., Casari P., Östberg P.O. (2019). Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey. ACM Comput. ACM Computing Surveys Volume 52 Issue 5 Article No.: 94 pp. 1–39. Internet resource: <https://dl.acm.org/doi/10.1145/3341145>
2. Bojchevski A., Shchur O., Zügner D., Günnemann S. (2018). NetGAN: Generating graphs via random walks. In Proceedings of the International Conference on Machine Learning. Internet resource: <https://arxiv.org/abs/1803.00816> (17.11.2022)
3. Wang H., Khoshgoftaar T.M., Napolitano A. (2010). in Proc. the 2010 Ninth International Conference on Machine Learning and Applications, A comparative study of ensemble feature selection techniques for software defect prediction (IEEE, Washington)
4. Raschka S. (2015). Python machine learning. (Packt Publishing Ltd, Birmingham)
5. Charleonnan A., Fufaung T., Niyomwong T., Chokchueypattanakit W., Suwannawach S., Ninchawee N. (2016). Predictive Analytics for Chronic Kidney Disease Using Machine Learning

Techniques. In Proceedings of the 2016 Management and Innovation Technology International Conference, Bang-San, Thailand, 12–14 October

6. Chen S. (2014). Cognitive Load Measurement from Eye Activity: Acquisition, Efficacy, and Real-time System Design; The University of New South Wales: Sydney, NSW, Australia

7. Liu C.C. (2017). Towards Practical Driver Cognitive Load Detection Based on Visual Attention Information; University of Toronto: Toronto, ON, Canada

8. Salvucci D.D., Anderson J.R. (1999). Mapping Eye Movements to Cognitive Processes; Carnegie Mellon University: Pittsburgh, PA, USA.

*(სტატია მიღებულია 10.11.2022)*

## APPLICATIONS OF MACHINE LEARNING IN MODERN DIGITAL TECHNOLOGIES

Nino Topuria, Tinatin Khatiashvili

Georgian Technical University  
nino.topuria@gtu.ge, khatiashvili.t@gtu.ge

### Summary

The article discusses a practical example of the use of machine learning, namely deep learning. A system designed on the basis of Azure cloud services, which uses computer vision, is presented. Computer vision is one of the cognitive services that allows computers to detect and recognize objects in the same way that humans do.

*(Received 10.11.2022)*

## ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ В СОВРЕМЕННЫХ ЦИФРОВЫХ ТЕХНОЛОГИЯХ

Топурия Н., Хатиашвили Т.

Грузинский Технический Университет  
nino.topuria@gtu.ge, khatiashvili.t@gtu.ge

### Резюме

В статье рассматривается практический пример использования машинного обучения, а именно глубокого обучения. Представлена система, спроектированная на базе облачных сервисов Azure, в которой используется компьютерное зрение. Компьютерное зрение - это одна из когнитивных служб, которая позволяет компьютерам обнаруживать и распознавать объекты так, как это делает человек.

*(Поступила в редакцию 10.11.22)*

# მიკროსერვისული არქიტექტურის აგება MsAzure-ს პლატფორმაზე

თინათინ ხატიაშვილი  
საქართველოს ტექნიკური უნივერსიტეტი  
khatiashvili.t@gtu.ge

## რეზიუმე

განხილულია აპლიკაციების დაპროექტების მიკროსერვისული და მონოლითური არქიტექტურა, მოცემულია მათ შორის განსხვავებება. წარმოდგენილია მიკროსერვისული არქიტექტურის დადებითი მხარეები და მისი გამოყენების უპირატესობები. აღწერილია მანქანური დასწავლის დახმარებით შექმნილი არქიტექტურა, რომელიც დაპროექტებულია მაიკროსოფტის Azure ღრუბლოვანი პლატფორმის ბაზაზე.

**საკვანძო სიტყვები:** მიკროსერვისული არქიტექტურა. მონოლითური არქიტექტურა. მანქანური დასწავლა. Azure. ღრუბლოვანი გამოთვლები.

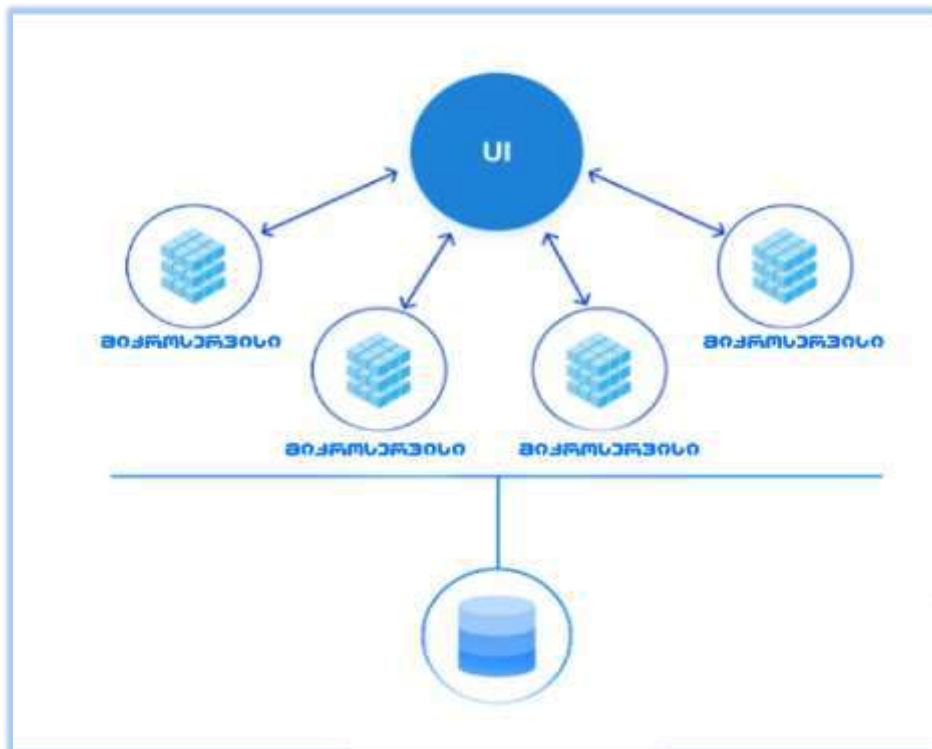
## 1. შესავალი

მიკროსერვისების არქიტექტურა, რომელიც ასევე უბრალოდ ცნობილია როგორც მიკროსერვისები, არის არქიტექტურული მეთოდი, რომელიც ეყრდნობა დამოუკიდებლად განლაგებულ სერვისებს. განახლება, ტესტირება, განლაგება და მასშტაბირება ხდება თითოეულ სერვისში. ასევე ამ სერვისებს აქვთ საკუთარი ბიზნეს ლოგიკა და მონაცემთა ბაზა კონკრეტული მიზნისთვის.

მიკროსერვისები აერთიანებს ძირითად ბიზნესს, დომენის სპეციფიკურ საკითხებს ცალკეულ, დამოუკიდებელ კოდურ ბაზებად. მიკროსერვისები არ ამცირებს სირთულეს, მაგრამ ისინი ნებისმიერ სირთულეს ხილულს და უფრო მართვადს ხდის. მიკროსერვისის გულისხმობს ამოცანების უფრო მცირე პროცესებად დაყოფას, რომლებიც ფუნქციონირებს ერთმანეთისგან დამოუკიდებლად, მაგრამ უზრუნველყოფენ ერთი მთლიანი აპლიკაციის ფუნქციონირებას.

მიკროსერვისის უპირატესობებია: სისწრაფე, მოქნილი მასშტაბირება, ტესტირება, დამოუკიდებლად განლაგების შესაძლებლობა, ტექნოლოგიის მოქნილობა და მაღალი საიმედოობა.

მიკროსერვისები ან (მიკროსერვისების არქიტექტურა) არის ღრუბლოვანი არქიტექტურული მიდგომა, რომელშიც ერთი აპლიკაცია შედგება მრავალი თავისუფლად დაკავშირებული და დამოუკიდებლად განლაგებული პატარა კომპონენტისგან ან სერვისისგან. მიუხედავად იმისა, რომ არ არსებობს ამ არქიტექტურული სტილის ზუსტი განმარტება, არსებობს გარკვეული საერთო მახასიათებლები. კერძოდ, ორგანიზაციის მიკრო-სერვისის არქიტექტურული სტილი არის მიდგომა ერთი აპლიკაციის შემუშავებისთვის, როგორც მცირე სერვისების კომპლექტი, თითოეული მუშაობს საკუთარ პროცესში და კომუნიკაციას უწევს მსუბუქ მექანიზმებს, ხშირად HTTP რესურსების API-ს. მიკროსერვისები დეცენტრალიზებულია და მუშაობს სხვადასხვა სერვერზე, მაგრამ ისინი საბოლოო პროდუქტის მისაღებად მაინც ერთიანდებიან. იდეალურ შემთხვევაში, თითოეული მიკროსერვისი ემსახურება ერთ ფუნქციას, რომელიც მარტივად მარშრუტირებად სერვისებს შორის API-სთან კომუნიკაციის საშუალებას იძლევა.



ნახ. 1. მიკროსერვისული არქიტექტურა

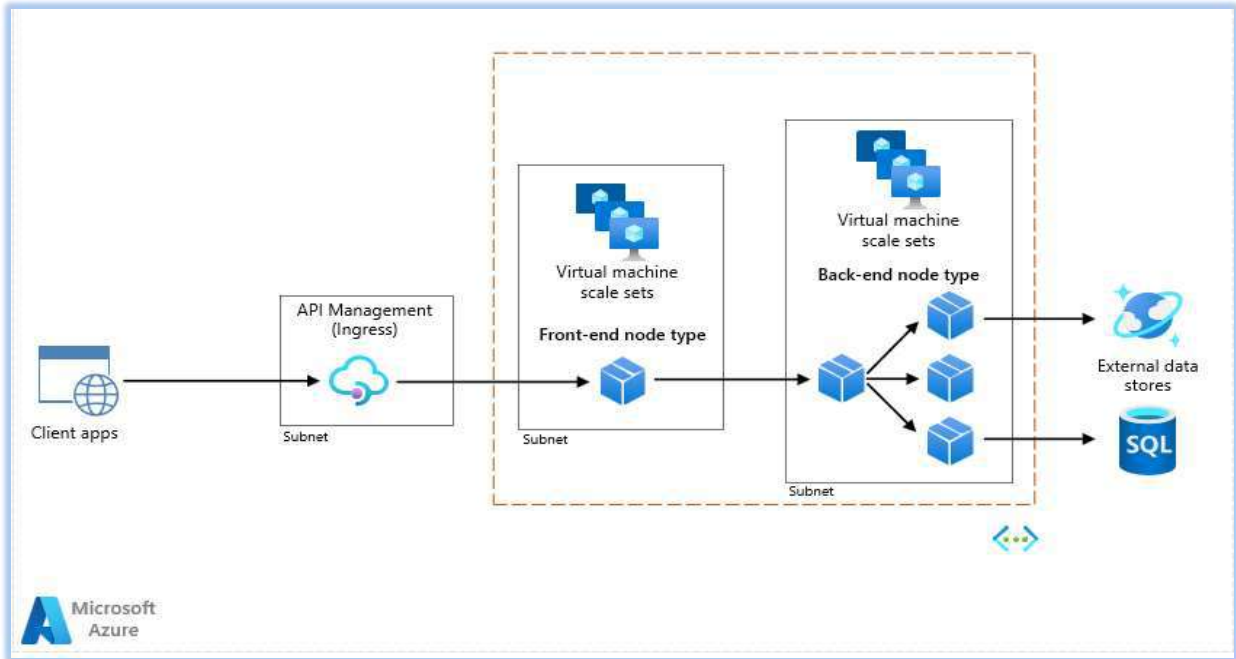
მონაცემთა ბაზები თანამედროვე კომპიუტერული ინფრასტრუქტურის ხერხემალია. მიკროსერვისულ აპლიკაციაში მონაცემთა მფლობელობა დეცენტრალიზებულია. ყველა სერვისი არის ავტონომიური და აქვს საკუთარი პირად მონაცემთა ბაზა, რომელიც შეესაბამება მის ფუნქციონირებას. ეს ნიშნავს, რომ ერთ სერვისს არ შეუძლია შეცვალოს სხვა სერვისის მონაცემთა ბაზაში შენახული მონაცემები, რაიმე ხარვეზის გარეშე. სწორედ ამიტომ საჭიარია სიფრთხილის გამოჩენა, იმისთვის რომ უნიკალური მონაცემების დუბლიერება გამოირიცხოს.

## 2. ძირითადი ნაწილი

წინამდებარე სტატიაში წარმოდგენილია მიკროსერვისული არქიტექტურის შედარება მონოლითურ არქიტექტურასთან. განსხვავება მიკროსერვისებსა და მონოლითურ არქიტექტურას შორის არის ის, რომ მონოლითური არქიტექტურა აგებულია, როგორც ერთი დიდი სისტემა და მას ხშირ შემთხვევაში აქვს კოდისთვის ერთადერთი ბაზა. აპლიკაციის ფუნქციონალების დამატებასთან ერთად, უფრო და უფრო რთულდება არქიტექტურა, რაც ართულებს სერვისების იზოლირებას ისეთი მიზნებისთვის, როგორცაა დამოუკიდებელი მასშტაბირება ან კოდის შენარჩუნება. ხოლო მიკროსერვისების არქიტექტურა აგებულია როგორც პატარა დამოუკიდებელი მოდული, რომელიც ეფუძნება ბიზნეს ფუნქციონირებას. მიკროსერვისების აპლიკაციაში, თითოეული პროექტი და სერვისი ერთმანეთისგან დამოუკიდებელია კოდის დონეზე. ამიტომ ადვილია მისი სრული კონფიგურაცია და განლაგება, ასევე მარტივია მასშტაბირება. მიკროსერვისები თითქმის თავიდანვე იყო გადაჯაჭვული

ღრუბლოვან არქიტექტურებთან, ამიტომ ისინი ერთმანეთისგან მრავალი თვალსაზრისით განსხვავდება.

მიკროსერვისული არქიტექტურის დახმარებით შესაძლებელია სხვადასხვა სერვისების აწყობა Azure მაიკროსოფტის ღრუბლოვან პლატფორმაზე. ნახაზი 2 ასახავს მიკროსერვისების არქიტექტურის ბაზაზე დაპროექტებულ სისტემას, რომელიც იყენებს Virtual Machine-ს, API მენეჯმენტსა და Sql მონაცემთა საცავს.



ნახ. 2. მიკროსერვისული არქიტექტურა Azure-ის პლატფორმაზე

- 1) Azure API მენეჯმენტი. ამ არქიტექტურაში API მენეჯმენტი მოქმედებს როგორც API კარიბჭე, რომელიც იღებს კლიენტების მოთხოვნებს და აგზავნის მათ სერვისებზე;
- 2) ვირტუალური მანქანების მასშტაბის კომპლექტები საშუალებას იძლევა შეიქმნას და მართვადი გახადოს იდენტური, დატვირთვით დაბალანსებული და მასშტაბირებადი ვირტუალური მანქანების ჯგუფი. ეს გამოთვლითი რესურსები ასევე თვალსაჩინოს ხდის ხარვეზების არსებობას და ხელს უწყობს დომენების განახლებას;
- 3) Node – კვანძები. არსებობის შემთხვევაში კვანძები არის ვირტუალური მანქანები, რომლებიც ეკუთვნის სერვისის კლასტერს;
- 4) Node type – კვანძის ტიპი წარმოადგენს ვირტუალური მანქანის მასშტაბის კომპლექტს, რომელიც განათავსებს კვანძების კრებულს. როგორც წესი, სერვერს აქვს მინიმუმ ერთი ტიპის კვანძი.

### 3. დასკვნა

მიკროსერვისები მეტად მოწესრიგებული არქიტექტურის ტიპია, რომელიც შესაძლებელს ხდის ციფრული პროდუქტის დანაწევრებასა და სამუშაოს გადანაწილებას.

მიკროსერვისები დაყოფილია კოლექციებად, რომლებიც გაერთიანებულია დიდ მაკროსერვისებში. ამ მიდგომით შესაძლებელია ფუნქციების კოდის სწრაფი განახლება

კონკრეტულ სერვისში ან აპლიკაციაში. მიკროსერვისი ცდილობს ისეთი პრობლემების გადაჭრას, როგორებიცაა: მონაცემთა ძებნა, ვებ სერვისების ფუნქციური უზუსტობების დადგენა და სხვა.

#### ლიტერატურა – References – Литература:

1. Fowler M. (2014). Microservices: a definition of this new architectural term. URL: <https://martinfowler.com/articles/microservices.html>. (5.11.22)
2. Zhang Y., Hua W., Zhou Z., Suh G.E., Delimitrou C. (2021). Sinan: ML-based and qos-aware resource management for cloud microservices. in ASPLOS
3. Kannan R.S., Subramanian L., Raju A., Ahn J., Mars J., Tang L.. (2019). Grand slam: Guaranteeing slas for jobs in microservices execution frameworks,” in EuroSys
4. The evolution of microservices. [Online]. Available: <http://www.slideshare.net/adriancockcroft/microservices-workshop-craft-conference>. (10.10.22)
5. Abeer Abdel Khaleq, Ilkyeun Ra. (2021). Intelligent autoscaling of microservices in the cloud for real-time applications. IEEE Access, 9:35464–35476.

## BUILDING A MICROSERVICE ARCHITECTURE ON THE AZURE PLATFORM

Tinatin Khatiashvili  
Georgian Technical University  
khatiashvili.t@gtu.ge

#### Summary

The article discusses microservice and monolithic application design architectures, and also provides a comparison between them. The advantages of microservice architecture and its use are presented. The architecture based on machine learning, developed on the basis of the Microsoft Azure cloud platform, is described.

## ПОСТРОЕНИЕ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ НА ПЛАТФОРМЕ AZURE

Хатиашвили Т.  
Грузинский Технический Университет  
khatiashvili.t@gtu.ge

#### Резюме

Рассматриваются микросервисная и монолитная архитектуры проектирования приложений, и показаны различия между ними. Представлены преимущества микросервисной архитектуры и ее использования. Описана архитектура на основе машинного обучения, разработанная на базе облачной платформы Microsoft Azure.

# სუფთა არქიტექტურაზე აგებული ორგანიზაციული მართვის საინფორმაციო სისტემის Agile ტესტირება

ხატია ხატიაშვილი, გია სურგულაძე

საქართველოს ტექნიკური უნივერსიტეტი

khatia.khatiashvili12@gmail.com, g.surguladze@gtu.ge

## რეზიუმე

განხილულია „სუფთა არქიტექტურაზე“ შესრულებული ორგანიზაციული მართვის საინფორმაციო სისტემების პროგრამული დეველოპმენტისა და ტესტირების საკითხები Agile მეთოდოლოგიის ბაზაზე. კვლევის ფარგლებში წარმოდგენილია სისტემატური დაკვირვების შედეგები პროგრამული აპლიკაციის დაპროექტებისა და ტესტირების ეტაპების პროცესებზე, დამკვეთისა და დეველოპერების გუნდური მუშაობისას ექაილ მეთოდით. წარმოდგენილია ექსპერიმენტის დროს გამოვლენილი ხარვეზები და მათი გადაჭრის გზები, განსაკუთრებით პროგრამული პროექტის Agile ტესტირების საფუძველზე. შემოთავაზებულია გუნდის მიერ ჩატარებული სამოქმედო სტრატეგიის ცვლილებები, აგრეთვე რეკომენდაციები Agile მეთოდოლოგიით „სუფთა არქიტექტურაზე“ შესრულებული პროექტების მართვის შემთხვევაში, ტესტირების პროცესის ეფექტიანად ჩატარებისთვის, რათა შესაძლებელი იყოს გამოვლენილი პრობლემების პრევენცია ადრეულ ეტაპებზე.

**საკვანძო სიტყვები:** ორგანიზაციული მართვის სისტემა. მენეჯმენტის პროცესი. პროგრამული აპლიკაცია. Clean Arcitecture. Agile Development. Agile Testing. Scrum მეთოდი.

## 1. შესავალი

ორგანიზაციული სისტემების მართვის პროცესების ავტომატიზაცია თანამედროვე ინფორმაციული ტექნოლოგიების ბაზაზე მოითხოვს დეველოპერებისაგან საბოლოო პროდუქტის ხარისხის სრულყოფას. აქ მნიშვნელოვანი წილი თვით პროგრამულ დეველოპმენტს და ტესტირებას ეკუთვნის [1,2]. პროგრამული აპლიკაციის დამუშავების იტერაციული ტესტირების პროცესი Agile მეთოდოლოგიის განვითარების პრინციპებს ეყრდნობა. იგი ძირითადად, სხვა მეთოდოლოგიებისგან განსხვავებით, „დამკვეთი – მომხმარებლების“ დინამიკურ მოთხოვნებს ითვალისწინებს. ტესტირების გუნდი კი შესაბამისად ამ მოთხოვნებისა, ახორციელებს დეველოპმენტის პროცედურებს. Agile დეველოპმენტის პროცესის თანმდევა რელიზების (ვერსიების) სიმრავლე, რამაც წარმოშვა DevOps მეთოდოლოგია. მისი მიზანია პროგრამული სისტემების შექმნის ციკლის დროის შემცირება და მაღალი ხარისხის პროგრამული უზრუნველყოფის უწყვეტი მიწოდება [3].

პროგრამული სისტემის დეველოპმენტის პროცესების სწორად გასაგებად, მნიშვნელოვანია, რომ პროექტის დაგეგმვის დროს გათვალისწინებული იქნას რამდენიმე მასთან დაკავშირებული კონცეფცია: ხარისხის უზრუნველყოფა, ხარისხის გეგმა და ხარისხის კონტროლი [3,4]:

- ხარისხის უზრუნველყოფა (QA) ეხება პროცესს, რომელიც გამოიყენება პროდუქტის შესაქმნელად და შეიძლება შესრულდეს მენეჯერის, კლიენტის ან თუნდაც მესამე მხარის მიმომხილველის მიერ. ხარისხის უზრუნველყოფა მოიცავს პროცესების სიებს, პროექტის აუდიტს და მეთოდოლოგიისა და სტანდარტების შემუშავებას;



- ხარისხის გეგმა (QP) ეხება პროცესს, რომელიც გამოიყენება თითოეული პროექტისთვის, რათა გამოცხადდეს პროექტის ვალდებულებები, მან დაიცვას მოქმედი სტანდარტების ნაკრები (კოდების სტანდარტები), რეგულაციები, პროცედურები (როგორ ხორციელდება QC და QA), გაიდლაინები (საუკეთესო პრაქტიკა) და ინსტრუმენტები. განვითარების სასიცოცხლო ციკლის განმავლობაში გარდა ამისა, QP უნდა შეიცავდეს მოსალოდნელ რისკებს და რისკების მართვას. ხარისხის დაგეგმვა უზრუნველყოფს ხარისხის გარანტიას (პროცედურები, ხარისხის უზრუნველყოფის ჩამონათვალი, ხარისხის ეტაპები).

- ხარისხის კონტროლი (QC) გამოიყენება იმის დასადასტურებლად, რომ პროდუქტები მისაღები ხარისხისაა. მაგალითად: გამართულად მუშაობს და მისაღები ვიზუალიც აქვს. ხარისხის კონტროლის აქტივობები მოიცავს ინსპექტირებას, მიწოდების შეუფერხებლობის მიმოხილვას და ტესტირების პროცესს.

სწორედ, Agile მეთოდოლოგიის Scrum მეთოდით პროგრამული უზრუნველყოფის განვითარება, Clean Architecture ტიპის პროექტებზე, მისი ტესტ-პროგრამების შემუშავება და მათი მოდულური ტესტირების პროცესის კვლევა არის ამ სტატიის ძირითადი მიზანი.

## 2. ძირითადი ნაწილი

დღეისათვის, პროგრამული უზრუნველყოფის სამყაროში ფართოდ გამოიყენება ე.წ. Clean Architectures. ის ეფუძნება პროგრამული უზრუნველყოფის დიზაინის პრინციპს: პასუხისმგებლობების გამიჯვნას. ერთ-ერთი ყველაზე ცნობილი არის ექსკუთხა არქიტექტურა, მაგრამ არსებობს მრავალი სხვა.

ჩვენი საპრობლემო სფერო ორგანიზაციული მართვის სისტემა (პირობითად, საჯარო სამსახური ან კერძო კორპორაცია). ექსპერიმენტის დაწყებამდე პროგრამული უზრუნველყოფის დეპარტამენტის გუნდი სპრინტ მეთოდოლოგიით მუშაობდა 24 თვის განმავლობაში. გუნდის წევრებმა მუშაობის დასაწყისში გაიარეს მაღალი დონის ტრენინგი. მათი მუშაობა პირველი 6 თვის განმავლობაში მიმდინარეობდა პროფესიონალი Scrum მასტერის ზედამხედველობით. ექსპერიმენტი 24 სამუშაო კვირის განმავლობაში მიმდინარეობდა, რაც დაახლოებით 180 შეხვედრას მოიცავდა. დაკვირვების ფარგლებში გამოვლინდა სპრინტ მეთოდოლოგიით ხელმძღვანელობის დროს ტესტირების მიერ განხორციელებული მიდგომების მნიშვნელოვანი ნაკლოვანებები: ტესტირების დროის მენეჯმენტი (2 კვირიანი სპრინტის განმავლობაში, პირველ კვირაში მათ ფაქტობრივად არ აქვთ მუშა პროდუქტი გასატესტად და სპრინტის ბოლოსკენ გროვდება საკმაოდ ბევრი საქმე, ტესტირების თვალსაზრისით).

ექსპერიმენტის შემდეგ გუნდმა შეცვალა მიდგომა, კერძოდ: ტესტირები მთლიან პროდუქტს ინტერფეისის მხრიდან აღარ ტესტავდნენ, სპრინტის დასაწყისშივე იწყებდა როგორც დეველოპერი პროგრამული უზრუნველყოფის განვითარებას, ასევე ტესტერი ე.წ. Unit-ტესტების წერას. გარკვეული სამუშაო დღეების შემდეგ პროგრამისტს მზად აქვს კოდი, ხოლო ტესტერს შესაბამისი პროგრამული უზრუნველყოფის გასატესტი ფუნქციები. შეცდომების აღმოფხვრის თვალსაზრისით მნიშვნელოვან როლს ასრულებს პროექტის არქიტექტურა, პროექტი იწერება Clean Architecture\_ზე რაც თავისთავად გულისხმობს პასუხისმგებლობების გამიჯვნას და შეცდომის აღმოჩენის შემთხვევაში დეველოპერს უმარტივდება მათი აღმოფხვრა.

### 3. სუფთა არქიტექტურა

სუფთა არქიტექტურის (Clean Architecture) მნიშვნელოვანი მიზანია დეველოპერებს შესთავაზოს კოდის ორგანიზება ისე, რომ იგი ასახავდეს ბიზნეს ლოგიკას, მაგრამ ინახავდეს მას მიწოდების მექანიზმისგან განცალკევებულად. სუფთა არქიტექტურის მთავარი წესი ისაა, რომ კოდის დამოკიდებულებები შეიძლება მხოლოდ გარე დონეზე გადავიდეს [5].

„დამოკიდებულების წესი“ არის "სუფთა არქიტექტურის" გული, სადაც ნათქვამია, რომ წყაროს კოდის დამოკიდებულება მხოლოდ აპლიკაციის ბირთვისკენ შეიძლება იყოს მიმართული. ამ არქიტექტურას მრავალი სახელი ეწოდა წლების განმავლობაში, პირველი სახელი იყო Hexagonal Architecture, რასაც მოჰყვა Ports-and-Adapters. ბოლო დრომდე მას მოიხსენიებდნენ, როგორც „ხახვის არქიტექტურას“ ან სუფთა არქიტექტურას. აპლიკაციები, რომლებიც მიჰყვება დამოკიდებულების ინვერსიის პრინციპს, ისევე როგორც დომენზე ორიენტირებული დიზაინის (DDD) პრინციპებს, როგორც წესი, მიდის მსგავს არქიტექტურამდე და ჩვეულებრივ შეიცავს: დომენის ფენას; განაცხადის ფენას; მონაცემთა ფენას; IoC ფენას; პრეზენტაციის ფენას და ტესტის ფენას.

სუფთა არქიტექტურა ათავსებს ბიზნეს ლოგიკას (Application) და აპლიკაციის მოდელს (Domain) აპლიკაციის ცენტრში ერთად, რომელსაც ეწოდება Core. Core უნდა იყოს სრულიად დამოუკიდებელი მონაცემთა წვდომისა და სხვა ინფრასტრუქტურული პრობლემებისგან. ეს მიიღწევა Core-ში ინტერფეისების ან აბსტრაქციების დამატებით, რომლებიც განხორციელებულია Core-ის გარეთ მდებარე ფენებით.

- დომენი: შეიცავს მოდულებს (ბიზნეს მოდელებს) და საცავის ინტერფეისებს;
- აპლიკაცია: შეიცავს სერვისის ინტერფეისებს და მათ სერვისებს, DTO-ებს (მონაცემთა გადაცემის ობიექტებს) და ხედის მოდელებს;
- მონაცემები: შეიცავს EF Core ტიპებს (DbContext, Migration), მონაცემთა წვდომის განხორციელების ტიპებს (საცავები), ინფრასტრუქტურის სპეციფიკურ სერვისებს (მაგალითად, FileLogger ან SmtptNotifier);
- Infrastructure.IoC: შეიცავს Dependency Container კლასს, რომელიც ეხმარება დამოკიდებულების ინვერსიის განხორციელებას;
- მომხმარებლის ინტერფეისის ტიპები: მოიცავს კონტროლერებს, ფილტრებს, ხედებს, გაშვებას;
- ტესტების ტიპები: მოიცავს მოდულის ტესტებს, ინტეგრაციის ტესტებს.

პროექტის განსახორციელებლად და, შესაბამისად, პროგრამული უზრუნველყოფის შესაქმნელად გუნდს დასჭირდა Visual Studio NET Core 2022, C# -ით, SQL Server მონაცემთა ბაზა და Entity Framework.

პროექტის არქიტექტურას რაც შეეხება ის დაყოფილია კონტროლერებად (Controllers), მოდელებად (Models), სერვისებად (Repository/Service) და ინტერფეისებად (Interfaces).

სატესტო პროექტის მოკლე აღწერილობა ასეთია: უნდა შეიქმნას (ვირტუალური) ორგანიზაციის თანამშრომლების მიღება გადინების და განსაკუთრებული პირობებით ხელფასის დარიცხვის სისტემა. პროგრამა უნდა გამოიყენონ კომპანიის თანამშრომლებმა, რომლებიც არიან სხვადასხვა დეპარტამენტში და აქვთ განსხვავებული პოზიციები, შესაბამისად სხვადასხვა უფლებები.

ამ პროგრამის გასატესტად დაიწერა Unit ტესტები. შესაბამისად რადგან ხარვეზი ტესტირების მიმართულებით იყო გუნდი შეთანხმდა Agile დეველოპმენტის ტესტირების გარკვეულ სტანდარტებზე.

იმისთვის რომ არსებული პროექტის Api გაიტესტოს, უნდა შეიქმნას ახალი პროექტი „xUnit Test“ ბიბლიოთეკით. ახლად შექმნილ პროექტში საჭიროა ჩაშენდეს „FakeItEasy“ და „FluentAssertions“ ბიბლიოთეკები.

ჯგუფს უნდა გაეკეთებინა არჩევანი „FakeItEasy“, „Nsubstitute“ ან „Moq“ ბიბლიოთეკებიდან ერთ-ერთზე, რადგან ისინი თავსებადია C# ენასთან (რომელზეც პროექტი იწერებოდა).

„Nsubstitute“ შედარებით ძველი ბიბლიოთეკაა, მას აქვს კარგი დოკუმენტაცია და ჩამოთვლილებიდან ყველაზე მარტივი სინტაქსი, მაგრამ არ აქვს გამოყოფილი კლასი. ფუნქციურად არ არის მნიშვნელოვანი განსხვავებები „Moq“-სა და „FakeItEasy“-ს შორის, მაგრამ „FakeItEasy“-ს არ აქვს „შემოწმებადი“ (Verifiable) ან „მოლოდინი“ (Expectations) ფუნქციონირება, მას აქვს მტკიცებულებები, რომლებიც ყოველთვის ცალსახადაა ნათქვამი ტესტის ბოლოს, რაც აადვილებს ტესტების წაკითხვას და გაგებას. სწორედ ამიტომ „FakeItEasy“ შეირჩა. უფრო დეტალურად შერჩეული ბიბლიოთეკის გამოყენების დროს:

- მკაფიოდ განცალკევებულია ტესტირების სამი ეტაპი: Arrange, Act, Assert;
- კლასების ტიპებისთვის არის უსაფრთხო, რადგან სანამ დაიწყება ტესტირება, იგი საჭიროებს კონსტრუქტორში შესაბამისი კლასის არგუმენტებს;
- საჭიროების/სურვილის შემთხვევაში შესაძლებელია კონსტრუქტორის არგუმენტების ავტომატური გენერირება;
- არასწორი ობიექტებიდან იღებს შესაბამის მოვლენებს (events);
- ყოველი შეცდომა „გაყალბებული ტიპის“ მაგალითს ჰგავს (ვიზუალურად ჩანს ეს წითელი ფერით);
- შეცდომების შესაბამისი შეტყობინებები ნათლად უჩვენებს ტესტერს/დეველოპერს თუ სად წარიმართა პროგრამა არასწორად. ტესტები იწერება როგორც კონტროლერებისთვის ასევე სერვისებისთვის. იმ შემთხვევაში თუ მეთოდმა ტესტირება წარმატებით გაიარა ჩნდება მწვანე ფერი და შესაბამისი შეტყობინება. ხოლო თუ მეთოდში რაღაც ხარვეზია წითელი ფერი და დეტალური ახსნა რომელ ნაწილშია პრობლემა.

მაგალითისთვის განვიხილოთ ორი მეთოდის ტესტირება კონტროლერში:

```
public class EmployeeControllerTests
{
    private EmployeeController _employeeController;
    private IEmployeeRepository _employeeRepository;
    private IHttpContextAccessor _httpContextAccessor;
    public EmployeeControllerTests()
    {
        //Dependencies
        _employeeRepository = A.Fake<IEmployeeRepository>();
        _httpContextAccessor = A.Fake<HttpContextAccessor>();
        //SUT
        _employeeController = new EmployeeController( _employeeRepository );
    }
}
```

```

[Fact]
public void EmployeeController_Index_ReturnsSuccess()
{
    //Arrange
    var employees = A.Fake<IEnumerable<Employee>>();
    A.CallTo(() => _employeeRepository.GetAll()).Returns(employees);
    //Act
    var result = _employeeController.Index();
    //Assert
    result.Should().BeOfType<Task<IActionResult>>();
}
[Fact]
public void EmployeeController_Detail_ReturnsSuccess()
{
    //Arrange
    var id = 1;
    var employee = A.Fake<Employee>();
    A.CallTo(() => _employeeRepository.GetByIdAsync(id)).Returns(employee);
    //Act
    var result = _employeeController.DetailEmployee(id, "EmployeeTeste1");
    //Assert
    result.Should().BeOfType<Task<IActionResult>>();
}
}

```

ბოლო სერვისების მიმართულებით მაგალითისთვის ავიღოთ 3 ტესტი:

```

public class EmployeeRepositoryTests
{
    private async Task<ApplicationDbContext> GetDbContext()
    {
        var options = new DbContextOptionsBuilder<ApplicationDbContext>()
            .UseInMemoryDatabase(databaseName: Guid.NewGuid().ToString())
            .Options;
        var dbContext = new ApplicationDbContext(options);
        dbContext.Database.EnsureCreated();
        if(await dbContext.Employees.CountAsync() < 0)
        {
            for (int i = 0; i < 10; i++)
            {
                dbContext.Employees.Add(
                    new Employee()
                    {
                        UserName = "UserName 1",
                        Email = "test1@gmail.com",
                        Salary = 2000,
                    }
                );
            }
        }
    }
}

```

```
        Category = EmployeeCategory.Temporary,
        AppUser = new AppUser()
        {
            FirstName = "firstname 1",
            LastName = "lastname 1",
            MobileNumber = "544444444",
            PrivateEmail = "0000000000",
            EmailConfirmed = "1",
            ProfileImageUrl = "/url",
            State = "stateTest"
        }
    });
    await databaseContext.SaveChangesAsync();
}
return databaseContext;
}
[Fact]
public async void EmployeeRepository_GetByIdAsync_ReturnsEmployee()
{
    //Arrange
    var id = 1;
    var dbContext = await GetDbContext();
    var employeeRepository = new EmployeeRepository(dbContext);

    //Act
    var result = employeeRepository.GetByIdAsync(id);

    //Assert
    result.Should().NotNull();
    result.Should().BeOfType<Task<Employee>>();
}

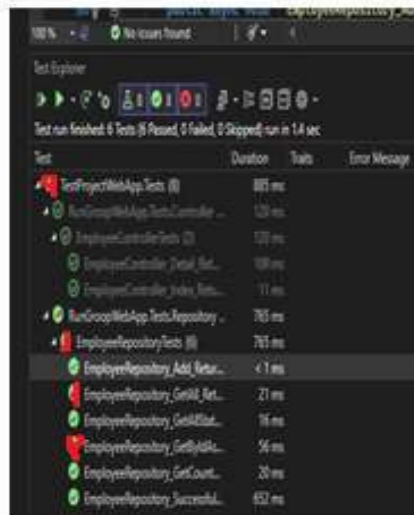
[Fact]
public async void EmployeeRepository_GetAll_ReturnsList()
{
    //Arrange
    var dbContext = await GetDbContext();
    var employeeRepository = new EmployeeRepository(dbContext);

    //Act
```

```
var result = await employeeRepository.GetAll();

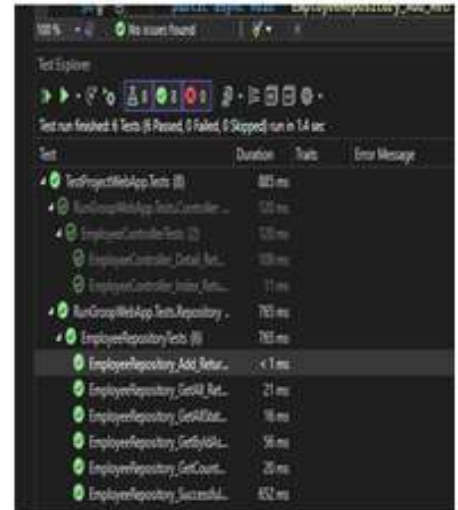
//Assert
result.Should().NotNull();
result.Should().BeOfType<List<Employee>>();
}
}
```

1) შედეგი ხარვეზით



1. შედეგი ხარვეზით

2) სწორი შედეგი



ნახ.2. სწორი შედეგი

### 3. დასკვნა

ექსპერიმენტით დადგინდა, რომ თუ კომპანია გადაწყვეტს Agile მეთოდოლოგიის Scrum მეთოდით მუშაობას იმისთვის, რომ ტესტირების ნაწილი განხორციელდეს წარმატებულად. მათ თავიდანვე უარი უნდა თქვან მთლიანი პროექტის ფუნქციონირების ინტერფეისით გატესტვაზე და API ტესტირება განახორციელონ unit ტესტების საშუალებით, ხოლო ამასთან თუ კომპანია არჩევანს გააკეთებს სუფთა არქიტექტურაზე, შეცდომების აღმოფხვრა ეფექტურ დროში განხორციელდება.

### ლიტერატურა – References – Литература:

1. Crispin L., Gregory J. (2009). Agile Testing: a Practical Guide for Testers and Agile Teams. Addison-Wesley. – 579 p.
2. Azarkerdar S. (2018). Agile Testing in Scrum: A Study of Major Factors for Successful Agile Testing Implementations in the View of Agile Testers. Uppsala University in Sweden. – 69 p.
3. Chogovadze G., Surguladze G., Gulitashvili M., Dolidze S. (2020). Software Quality Management: Testing and Optimization. ISBN 978-9941-8-0629-2. © „IT-Consulting Research Center“ Georgian Technical University, Tbilisi, - 363 p., (in Georgian)
4. Patel A. (2021). Clean Architecture with .NET and .NET Core — Overview. <https://medium.com/dotnet-hub/clean-architecture-with-dotnet-and-dotnet-core-aspnetcore-overview-introduction-getting-started-ec922e53bb97> (20.12.22)

(სტატია მიღებულია 10.12.2022)

## AGILE TESTING OF AN ORGANIZATIONAL MANAGEMENT INFORMATION SYSTEM BUILT ON A CLEAN ARCHITECTURE

Khatiashvili Khatia, Surguladze Gia  
Georgian Technical University  
khatiashvili.k@gtu.ge, g.surguladze@gtu.ge

### Summary

Issues of software development and testing of organizational management information systems performed on "clean architecture" based on Agile methodology are discussed. Within the framework of the research, the results of systematic observation of the processes of the software application design and testing stages, during the team work of the customer and developers using the Agile method, are presented. The flaws identified during the experiment and ways to solve them are presented, especially based on the Agile testing of the software project. Changes in the action strategy carried out by the team are proposed, as well as recommendations in the case of managing projects executed on the Agile methodology on "clean architecture", to conduct the testing process effectively, so that it is possible to prevent the identified problems in the early stages.

*(Received 10.12.2022)*

## ГИБКОЕ ТЕСТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ОРГАНИЗАЦИЕЙ, ПОСТРОЕННОЙ НА ЧИСТОЙ АРХИТЕКТУРЕ

Хатиашвили Х., Сургуладзе Г.  
Грузинский Технический Университет  
khatiashvili.k@gtu.ge, g.surguladze@gtu.ge

### Резюме

Рассматриваются вопросы разработки и тестирования программного обеспечения информационных систем управления организациями, построенных на «чистой архитектуре» с использованием Agile методологии. В рамках исследования представлены результаты систематического наблюдения за процессами этапов проектирования и тестирования программного приложения, при совместной работе заказчика и разработчиков по гибкому методу. Представлены недостатки, выявленные в ходе эксперимента, и пути их решения, особенно на основе Agile-тестирования программного проекта. Предложены соответствующие рекомендации с целью эффективного проведения процессов тестирования, чтобы можно было предотвратить выявленные проблемы на ранних стадиях программирования.

*(Поступила в редакцию 10.12.22)*

# Agile-ტესტირება მუტაციური ტესტებით მენეჯმენტის საინფორმაციო სისტემის აპლიკაციაში

ხატია ხატიაშვილი

საქართველოს ტექნიკური უნივერსიტეტი  
khatia.khatiashvili12@gmail.com

## რეზიუმე

განხილულია ორგანიზაციული მართვის საინფორმაციო სისტემების პროგრამული დეველოპმენტის და ტესტირების საკითხები Agile მეთოდოლოგიის ბაზაზე მუტაციური ტესტების დახმარებით. კვლევის ფარგლებში წარმოდგენილია სისტემატური დაკვირვების შედეგები პროგრამული აპლიკაციის დაპროექტებისა და ტესტირების ეტაპების პროცესებზე, დამკვეთისა და დეველოპერების გუნდური მუშაობისას Scrum მეთოდით. ყურადღება გამახვილებულია ტესტირების ეტაპის ვადებზე და ხარისხზე. შემოთავაზებულია სამოქმედო სტრატეგია და რეკომენდაციები Agile მეთოდოლოგიით „სუფთა არქიტექტურაზე“ შესრულებული პროექტების მართვის შემთხვევაში, ტესტირების პროცესის ეფექტიანად ჩატარებისთვის, რათა შესაძლებელი იყოს გამოვლენილი პრობლემების პრევენცია ადრეულ ეტაპებზე მუტანტი ტესტების დახმარებით.

**საკვანძო სიტყვები:** ორგანიზაციული სისტემა. მენეჯმენტის პროცესი. პროგრამული აპლიკაცია. Mutant Testing. Clean Architecture. Agile Development. Agile Testing.

## 1. შესავალი

### ➤ მოდულის ტესტირება

ტესტირების პირველი რაუნდის დროს პროგრამა ექვემდებარება შეფასებებს, რომლებიც ფოკუსირებულია პროგრამული უზრუნველყოფის კონკრეტულ მოდულზე ან კომპონენტებზე, რათა დადგინდეს, არის თუ არა თითოეული მათგანი სრულად ფუნქციონალური. ამ მცდელობის მთავარი მიზანია იმის დადგენა, ფუნქციონირებს თუ არა აპლიკაცია ისე, როგორც შემუშავებულია. ამ ფაზაში მოდულს შეუძლია მიმართოს ფუნქციას, ინდივიდუალურ პროგრამას ან თუნდაც პროცედურას და სამუშაოს შესასრულებლად, ჩვეულებრივ, გამოიყენება White-box ტესტირების მეთოდი. ამ ტესტირების ფაზის ერთ-ერთი ყველაზე დიდი უპირატესობა ის არის, რომ მისი გაშვება შესაძლებელია ყოველ ჯერზე, როცა კოდის ნაწილი შეიცვლება, რაც შესაძლებელს ხდის პრობლემების სწრაფად გადაჭრას. საკმაოდ ხშირია პროგრამული უზრუნველყოფის შემქმნელებისთვის, რომ შეასრულონ მოდულის ტესტები, სანამ პროგრამული უზრუნველყოფის მიწოდებას აწვდიან ტესტირებს ოფიციალური ტესტირებისთვის.

### ➤ ინტეგრაციის ტესტირება

ინტეგრაციის ტესტირება საშუალებას აძლევს ინდივიდებს გააერთიანონ პროგრამის ფარგლებში არსებული ყველა მოდული და გამოსცადონ ისინი ჯგუფურად. ტესტირების ეს დონე შექმნილია მოდულებს/ფუნქციებს შორის ინტერფეისის დეფექტების მოსაძებნად. ეს განსაკუთრებით მომგებიანია, რადგან ის განსაზღვრავს რამდენად ეფექტურად მუშაობს მოდულები ერთად. გასათვალისწინებელია, რომ რაც არ უნდა ეფექტურად მუშაობდეს



თითოეული მოდული, თუ ისინი სათანადოდ არ არის ინტეგრირებული, ეს გავლენას მოახდენს პროგრამული პროგრამის ფუნქციონირებაზე. ამ ტიპის ტესტების ჩასატარებლად, ინდივიდებს შეუძლიათ გამოიყენონ ტესტირების სხვადასხვა მეთოდი, მაგრამ კონკრეტული მეთოდი, რომელიც გამოყენებული იქნება სამუშაოს შესასრულებლად, დიდად იქნება დამოკიდებული ერთეულების განსაზღვრის გზაზე.

### ➤ სისტემის ტესტირება

სისტემის ტესტირება არის პირველი დონე, რომელშიც მთლიანი განაცხადის ტესტირება ხდება. ამ დონეზე მიზანია შეაფასოს, აკმაყოფილებს თუ არა სისტემა ყველა ჩამოთვლილ მოთხოვნას და დადგინდეს, რომ ის აკმაყოფილებს ხარისხის სტანდარტებს. სისტემის ტესტირებას ახორციელებენ დამოუკიდებელი ტესტირები, რომლებსაც არ მიუღიათ მონაწილეობა პროგრამის შემუშავებაში. ეს ტესტირება ხორციელდება გარემოში, რომელიც მჭიდროდ ასახავს წარმოებას. სისტემის ტესტირება ძალიან მნიშვნელოვანია, რადგან ის ადასტურებს, რომ აპლიკაცია აკმაყოფილებს ტექნიკურ, ფუნქციონალურ და ბიზნეს მოთხოვნებს, რომლებიც დაყენებულია მომხმარებლის მიერ. მისაღები ტესტირება საბოლოო დონე, Acceptance Testing (ან მომხმარებლის მიღების ტესტირება), ტარდება იმის დასადგენად, არის თუ არა სისტემა მზად გამოსაშვებად. პროგრამული უზრუნველყოფის განვითარების სასიცოცხლო ციკლის განმავლობაში, მოთხოვნების ცვლილებები ზოგჯერ შეიძლება არასწორად იქნას განმარტებული ისე, რომ არ დააკმაყოფილოს მომხმარებელთა მოთხოვნილებები. ამ ბოლო ფაზის განმავლობაში მომხმარებელი შეამოწმებს სისტემას, რათა გაარკვიოს, აკმაყოფილებს თუ არა აპლიკაცია მათი ბიზნესის საჭიროებებს. ამ პროცესის დასრულების და პროგრამული უზრუნველყოფის გავლის შემდეგ, პროგრამა გადაეცემა წარმოებას.

### ➤ მუტაციური ტესტირება

მუტაციის ტესტირება არის პროგრამული უზრუნველყოფის ავტომატური ტესტების გაშვების პროცესი კოდის წინააღმდეგ, რომელიც შეცვლილია მუტაციის შემოღებით მოდულის ტესტირების კომპლექტში პოტენციური ხარვეზების დასადგენად. თუ მუტაციის შემოღების შემდეგ ყველა ტესტი კვლავ წარმატებით დასრულდა ეს ნიშნავს რომ, სავარაუდოდ აკლია მოდულის ტესტს გარკვეული შემთხვევების განხილვა.

მიუხედავად იმისა, რომ ნაკლებად ცნობილია და ნაკლებად გამოიყენება, მუტაციის ტესტირებას აქვს შესაძლებლობა გააუმჯობესოს მოდულის ტესტირება. პროგრამული უზრუნველყოფის შემუშავებაში, კოდის ყველა სტრიქონი მნიშვნელოვანია. ამიტომ, რომ შეცდომების შეუმჩნეველად დაშვება ასე მარტივია. შეცდომების ადრეული გამოვლენა და აღმოფხვრა, სანამ ისინი რაიმე ზიანს მიაყენებს პროგრამას, არის პროგრამული უზრუნველყოფის განვითარების გადამწყვეტი ნაწილი. პროგრამული უზრუნველყოფის შემქმნელები იყენებენ სხვადასხვა ტიპის ტესტირებას, რათა მაქსიმალურად შეეცადონ თავიდან აიცილონ შეცდომები, სანამ პროგრამული აპლიკაცია წარმოებაში გამოვა.

ერთი ტიპის ტესტირება, რომელსაც პროგრამული უზრუნველყოფის დეველოპერები ხშირად ივიწყებენ, არის მუტაციის ტესტირება. მუტაციური ტესტირება მოდულის ტესტის მსგავსია. მაგრამ იმის ნაცვლად, რომ კოდის ტესტები დაწეროს, ცვლის აპლიკაციის ერთი ან მეტი მცირე ნაწილს ერთდროულად, იქამდე სანამ სრულად არ მოხდება ხელახლა ტესტირება რაიმე ახალი შეცდომის გარეშე.

მუტაციის ტესტირება არის პროგრამული უზრუნველყოფის ტესტირება, რომლის დროსაც დეველოპერი ახორციელებს ცვლილებებს ანუ „მუტანტებს“ საწყისი კოდის გარკვეული ნაწილისთვის და ამოწმებს, თქვენს მიერ დაწერილ ტესტებს შეუძლია თუ არა შეცდომების იდენტიფიცირება.

მუტანტის დანერგვის შემდეგ, თავდაპირველი მოდული ტესტით უნდა შემოწმდეს მუტანტის კოდი. თუ ერთი ან მეტი ტესტმა აჩვენა შეცდომა, ეს ნიშნავს რომ ტესტმა შეძლო „მოკლა მუტანტი“, რაც მიუთითებს იმაზე, რომ ტესტის შემთხვევა ეფექტურია და შეუძლია ამ ხარვეზის იდენტიფიცირება. თუ ყველა ტესტში, მუტანტმა შეძლო „გადარჩენა“, ეს მიუთითებს იმაზე, რომ არ არსებობს მოდულის ტესტი, რომელსაც შეუძლია ამ შეცდომის გამოვლენა და ეს პრობლემაა. ამის გადასაჭრელად, თქვენ უნდა დაამატოთ ახალი ტესტები ან განაახლოთ არსებული ტესტები წარუმატებლობის სცენარის დასაფარად.

რაც უფრო მეტი „მუტანტების“ რაოდენობა შეუძლია მოკლას ტესტმა, მით უფრო ძლიერი და ეფექტური იქნება მოდულის ტესტის ნაკრები. ამიტომ, მუტაციის ტესტირება მიზნად ისახავს ხარისხის და გამძლეობის დადგენას და გაზრდას. თუმცა, აუცილებლად გასათვალისწინებელი საკითხი ასეთი ტიპის ტესტირების დროს არის უზრუნველყოფა, რომ შეტანილი ცვლილებები გავლენას არ მოახდენს მთელ კოდზე. ამდენად, მუტანტები რაც შეიძლება პატარა უნდა იყოს. არსებობს მუტანტების რამდენიმე განსხვავებული ტიპი, რომლებიც შესაძლოა დაინერგოს მუტაციის ტესტირებისთვის. მუტანტების ყველაზე გავრცელებული ტიპებია:

1) მუტაციური მნიშვნელობები – ცვლადის მნიშვნელობის შეცვლა პროგრამაში შეცდომების გამოსავლენად. ყველაზე გავრცელებული სტრატეგია ცვლადის მნიშვნელობის გაზრდა ან შემცირებაა;

2) ლოგიკური ოპერატორების მუტაცია – ამ ტიპის მუტაციაში შეგიძლიათ შეცვალოთ „პირობები“ პროგრამული კოდის გადაწყვეტილების შესამოწმებლად ლოგიკური ოპერატორების ინვერსიების გამოყენებით. მაგალითად, თუ პირობა ამოწმებს მნიშვნელობას „==“, შეგიძლიათ შეცვალოთ იგი მნიშვნელობით „!=“;

3) განცხადებების შესრულების წესის შეცვლა;

4) მუტაციური განცხადებები – კოდის განცხადებების წაშლა ან დუბლიკატი, რომლებიც სიმულაციას უწევს შეცდომებს კოდის სხვაგან კოპირების ჩასმისას.

მაგალითად, თუ კოდში არის გამოხატულება:  $a + b$ , მუტაციის ტესტი დროებით შეიცვლება შემდეგით:  $a - b$ .

### ➤ მუტაციური ტესტირების ავტომატიზაცია

მუტაციის ტესტირება ძალზე შრომატევადი და რთულია ხელით შესასრულებლად. პროცესის დასაჩქარებლად მიზანშეწონილია მიმართოთ ავტომატიზაციის ხელსაწყოებს.

მუტაციის ტესტირების სახეები პროგრამული უზრუნველყოფის ინჟინერიაში, მუტაციის ტესტირება შეიძლება ფუნდამენტურად დაიყოს 3 ტიპად: განცხადების მუტაცია, გადაწყვეტილების მუტაცია და მნიშვნელობის მუტაცია.

1. პირობის მუტაცია - დეველოპერმა ამოჭრა და ჩასვა კოდის ნაწილი
2. მნიშვნელობის მუტაცია - პირველადი პარამეტრების მნიშვნელობები შეცვლილია
3. გადაწყვეტილების მუტაცია - საკონტროლო განცხადებები უნდა შეიცვალოს

### ➤ მუტაციის ქულა

მუტაციის ქულა განისაზღვრება, როგორც „მოკლული“ მუტანტების (აღმოჩენილი შეცდომები) პროცენტი მუტანტების საერთო რაოდენობასთან.

$$\text{მუტაციის ქულა} = (\text{მოკლული მუტანტები} / \text{მუტანტების საერთო რაოდენობა}) * 100$$

ტესტის შემთხვევები მუტაციის ადეკვატურია, თუ ქულა არის 100%. ექსპერიმენტულმა შედეგებმა აჩვენა, რომ მუტაციური ტესტირება ეფექტური მიდგომაა ტესტის შემთხვევების ადეკვატურობის გასაზომად. მაგრამ მთავარი ნაკლი ის არის, რომ მუტანტების გენერირებისა და ამ მუტანტის პროგრამის წინააღმდეგ თითოეული ტესტის შესრულების მაღალი ღირებულებაა.

### ➤ მუტაციური ტესტირების უპირატესობები

მუტაციის ტესტირების უპირატესობები შემდეგია:

- ეს არის ძლიერი მიდგომა პროგრამის მაღალი ხარისხით შესასწავლად.
- ამ ტესტირებას შეუძლია პროგრამის ყოვლისმომცველი ტესტირება.
- მუტაციის ტესტირებას მოაქვს შეცდომების გამოვლენის კარგი დონე პროგრამული უზრუნველყოფის შემქმნელისთვის
- ამ მეთოდს აქვს შესაძლებლობა აღმოაჩინოს პროგრამის ყველა ხარვეზი.
- მომხმარებლები რომლებიც სარგებლობენ ამ ტესტირებით ყველაზე საიმედო და სტაბილური სისტემის მიღების მაღალი ალბათობა აქვთ.

### ➤ მუტაციური ტესტირების უარყოფითი მხარეები

მეორეს მხრივ, მუტანტის ტესტირების უარყოფითი მხარეებია:

- მუტაციის ტესტირება ძალიან ძვირი და შრომატევადია, რადგან არსებობს მრავალი მუტანტური პროგრამა, რომელიც უნდა შეიქმნას.
- ვინაიდან ეს შრომატევადია, სამართლიანია იმის თქმა, რომ ეს ტესტირება არ შეიძლება გაკეთდეს ავტომატიზაციის ხელსაწყოს გარეშე.
- თითოეულ მუტაციას ექნება იგივე რაოდენობის ტესტის შემთხვევები, რაც ორიგინალ პროგრამას. ამრიგად, მუტანტის პროგრამების დიდი რაოდენობა შეიძლება გახდეს საჭირო.
- ვინაიდან ეს მეთოდი მოიცავს კოდის ცვლილებებს, ის საერთოდ არ გამოიყენება

თუ ორგანიზაციას სურს პროგრამის ამომწურავი ტესტირება, ტესტირების საუკეთესო გადაწყვეტილება არის მუტანტი ტესტირება. ეს არის ყველაზე სრულყოფილი ტექნიკა პროგრამის შესამოწმებლად. ეს არის მეთოდი, რომელიც ამოწმებს პროგრამის ეფექტურობასა და სიზუსტეს სისტემაში არსებული ხარვეზების ან შეცდომების გამოსავლენად.

[Stryker და PIT Testing](#) - ფრეიმვორქებია ოპტიმიზაციისათვის.

### ➤ Stryker ბიბლიოთეკა

კოდისთვის მუტაციების დანერგვა რთული და შრომატევადია, განსაკუთრებით თუ მუტანტების რაოდენობა ძალიან დიდია. ოპტიმიზაციის თვალსაზრისით C# საშუალებას აძლევს დეველოპერებს გამოიყენონ Stryker ბიბლიოთეკა, რათა ავტომატურად შეიტანონ ცვლილებები კოდში. Stryker-ის ბიბლიოთეკა იყენებს ზემოხსენებულ ტექნიკას კოდის მუტაციისთვის და ერთეულის ტესტების ეფექტურობის შესამოწმებლად. შემდეგი ცხრილი აჩვენებს კოდში არსებულ ზოგიერთ მუტაციას (ნახ.1).

<i>Original operation</i>	<i>Mutated operation</i>
+	-
-	+
*	/
<	>
!=	==
&&	
true	false

ნახ.1

Stryker არის ღია კოდის ინსტრუმენტი, რომელიც შესაძლებელია გამოყენებულ იქნას მუტანტების დასანერგად Javascript, Scala და C#-ში დაწერილი კოდისთვის. Stryker.NET არის ვერსია, რომელიც შესაძლებელია დეველოპერმა გამოიყენოს .NET Core და .NET-ში დაწერილი პროექტებისთვის. ის მხარს უჭერს 30-ზე მეტ მუტაციის ტიპს და აწარმოებს შედეგებს წასაკითხად HTML ფორმატში.

განვიხილოთ მაგალითი:

```
public static int Division(int a, int b)
{
    if (b != 0)
        { return a / b; }
    else
        { return 0; }
}
```

შესაბამისი მოდულის ტესტი:

```
[Fact]
public void Simple_division_should_return_correct_result()
{
    var actualValue = MathOperations.Division(100, 20);
    Assert.Equal(5, actualValue);
}
```

- მწვანე წერტილები მოკლულ მუტანტებზე მიუთითებს.
- გადარჩენილი მუტანტები წითელ წერტილებში იქნებიან.
- კოდი დაფარვის გარეშე მითითებული იქნება ნარინჯისფერი წერტილებით.

მუტანტი კოდი ასეთია (ნახ.2).

```

public static int Division(int a, int b)
{
    if (b != 0)
    if (b == 0)
    {
        return a / b;
    }
    else
    {
        return 0;
    }
}

public static int Division(int a, int b)
{
    if (b != 0)
    {
        return a / b;
    }
    else
    {
        return 0;
    }
}
    
```

ნახ.2

```

[Fact] public void Simple_division_returns_zero_when_divisor_is_zero()
{
    var actualValue = MathOperations.Division(100, 0);
    Assert.Equal(0, actualValue);
}
    
```

ამ შემთხვევაში ისევ იგივე შედეგს გამოიტანს პროგრამა რაც ნიშნავს რომ ამ შემთხვევაში არ გვაქვს საგანგაშო მდგომარეობა, მაგრამ კოდი შეიძლება გაუმჯობესდეს (ნახ.3).

```

public static int Division(int a, int b)
{
    if (b != 0)
    {
        return a / b;
    }
    else
    {
        return 0;
    }
}

public static int Division(int a, int b){
    if (b != 0)
        { return a / b; }
    return 0; }
}zzzzzzzzzz
    
```

ნახ.3. გაუმჯობესებული კოდი

მივიღებთ ასეთ შედეგს რაც ნიშნავს რომ ყველაფერი წესრიგშია (ნახ.4).

```

public static int Division(int a, int b)
{
    if (b != 0)
    if (b == 0)
    {
        return a / b;
    }
    else
    {
        return 0;
    }
}
    
```

ნახ.4

**ლიტერატურა – References – Литература:**

1. <https://reqtest.com/testing-blog/agile-testing-principles-methods-advantages/>
2. <https://www.agilealliance.org/agile101/the-agile-manifesto/>
3. <https://www.guru99.com/agile-testing-a-beginner-s-guide.html>

*(სტატია მიღებულია 10.12.2022)*

**Agile -TESTING WITH MUTATION TESTS IN MANAGEMENT  
INFORMATION SYSTEM EVALUATION**

Khatiashvili Khatia  
Georgian Technical University  
khatiashvili.k@gtu.ge

**Summary**

Issues of software development and testing of organizational management information systems based on Agile methodology with the help of mutational tests are discussed. Within the framework of the research, the results of systematic observation of the processes of the software application design and testing stages, during the team work of the customer and developers using the Scrum method, are presented. The focus is on the timing and quality of the testing phase. An action strategy and recommendations are offered in the case of managing projects executed on "clean architecture" with the Agile methodology, to effectively conduct the testing process, so that it is possible to prevent the identified problems in the early stages with the help of mutant tests.

*(Received 10.12.2022)*

**AGILE-ТЕСТИРОВАНИЕ МУТАЦИОННЫМИ ТЕСТАМИ АПЛИКАЦИИ  
ИНФОРМАЦИОННОЙ СИСТЕМЫ МЕНЕДЖМЕНТА**

Хатиашвили Х.  
Грузинский Технический Университет  
khatiashvili.k@gtu.ge

**Резюме**

Рассматриваются вопросы разработки программного обеспечения и тестирования информационных систем управления организациями на основе методологии Agile с помощью мутационных тестов. В рамках исследования представлены результаты систематического наблюдения за процессами этапов проектирования и тестирования программного приложения, при совместной работе заказчика и разработчиков по методу Scrum. Основное внимание уделяется срокам и качеству этапа тестирования. Предлагается стратегия действий и рекомендации в случае управления проектами, выполненными на «чистой архитектуре» по методологии Agile, для эффективного проведения процесса тестирования, чтобы можно было предотвратить выявленные проблемы на ранних стадиях с помощью мутантных тестов.

*(Поступила в редакцию 10.12.22)*

## ფრინველების სათავსოს ტემპერატურისა და განათების ავტომატიზაცია

მარინე ქოზაშვილი, ზაალ აზმაიფარაშვილი, გურამ მურჯიკნელი,  
გივი მურჯიკნელი, იური მოდებაძე  
საქართველოს ტექნიკური უნივერსიტეტი

kozashvilimarine@gmail.com, z.azmaiparashvili@gtu.ge, gurami.murjikneli@gtu.ge,  
g.murjikneli@gtu.ge, iura.modebadze@gtu.ge

### რეზიუმე

განხილულია მეფრინველეობის სათავსოში ზამთრობითა და ზაფხულობით ჰაერის (აგრეთვე იატაკის) ტემპერატურისა და განათების მუდმივ დონეზე შენარჩუნების საკითხები. ამისათვის გამოყენებულია სხვადასხვა ტიპის თერმორეგულატორები და ავტომატური მოწყობილობები.

**საკვანძო სიტყვები:** მეფრინველეობის ფერმა. ტემპერატურის რეგულირება. განათების რეგულირება. თერმორეგულატორი. ავტომატიზაცია.

### 1. შესავალი

დახურულ სათავსოში შინაური ფრინველების შენახვის მიზნით საჭიროა ოპტიმალური მიკროკლიმატის შექმნა. მისი ტემპერატურა უნდა იყოს  $10 - 26^{\circ} C$ , ხოლო განათება კი უნდა იცვლებოდეს შემოდგომა-ზამთრის დღის შუქის ხანგრძლივობიდან გამომდინარე. ასეთი რეჟიმის შესანარჩუნებლად ხდება ავტონომიური სათავსოს აღჭურვა გათბობისა და განათების სისტემით [1,2].

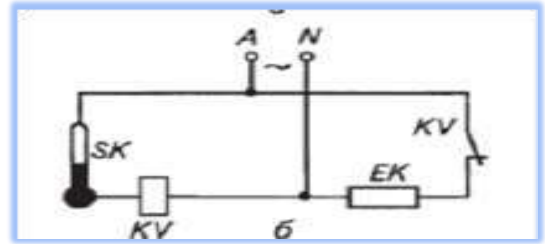
### 2. ძირითადი ნაწილი

შინაური ფრინველების სათავსოს გათბობა შეიძლება მოვახდინოთ როგორც მასში არსებული ჰაერით, ისე მისი იატაკის გათბობის შედეგად.

პირველ შემთხვევაში სათავსოს გათბობა ხდება უშუალოდ მისი ჰაერის გათბობით, რისთვისაც გამოიყენება ელექტრული მოწყობილობები (თერმორეგულატორები). ისინი გამოიყენება ტემპერატურის ავტომატური რეგულირებისათვის. მათი ზემოქმედების შედეგად ჰაერი სათავსოში, წყალი, ხელსაწყოთა ზედაპირები და ა.შ. იმყოფება სტაბილური ტემპერატურის ქვეშ.

1-ელ ნახაზზე მოცემულ სქემას აქვს დენის წყაროსთან მიერთებული ორი წრედი, მაგრამ მართვა ხდება EK გამაცხელებელი ხელსაწყოთი. როცა ტემპერატურა სათავსოში მოცემულზე ნაკლებია, წრედი თერმომეტრში გათიშულია და აჩვენებს შემცირებულ ტემპერატურას. ამასთან, დენი KV რელეს კოჭაში არ გადის, ხოლო რელეს გამთიშავი კონტაქტები ჩაკეტილია და მათი გავლით გამაცხელებელი ხელსაწყოსაკენ დენი გადის. მისი მუშაობის შედეგად სათავსოში ჰაერის ტემპერატურა თანდათან იზრდება, რომლის გარკვეული მნიშვნელობის დროს ხდება კონტაქტების ჩამოკლება თერმომეტრში. ამის შედეგად რელეს გრაფილში დენის გავლა აამუშავებს მას და გამოირთვება გამაცხელებელი ხელსაწყო. თუ ტემპერატურა რაიმე მიზეზით სათავსოში კვლავ დაიწევს, ამაზე რეაგირებას მოახდენს თერმომეტრი. ვერცხლისწყლის სვეტი

გათიშავს რელეს გრაგნილის კვების წრედს, რომლის კონტაქტები შეერთდებიან და ჩართვენ გამაცხელებელ ხელსაწყოს. შემდეგში ჩართვისა და გამორთვის პროცესები მეორდება [1].



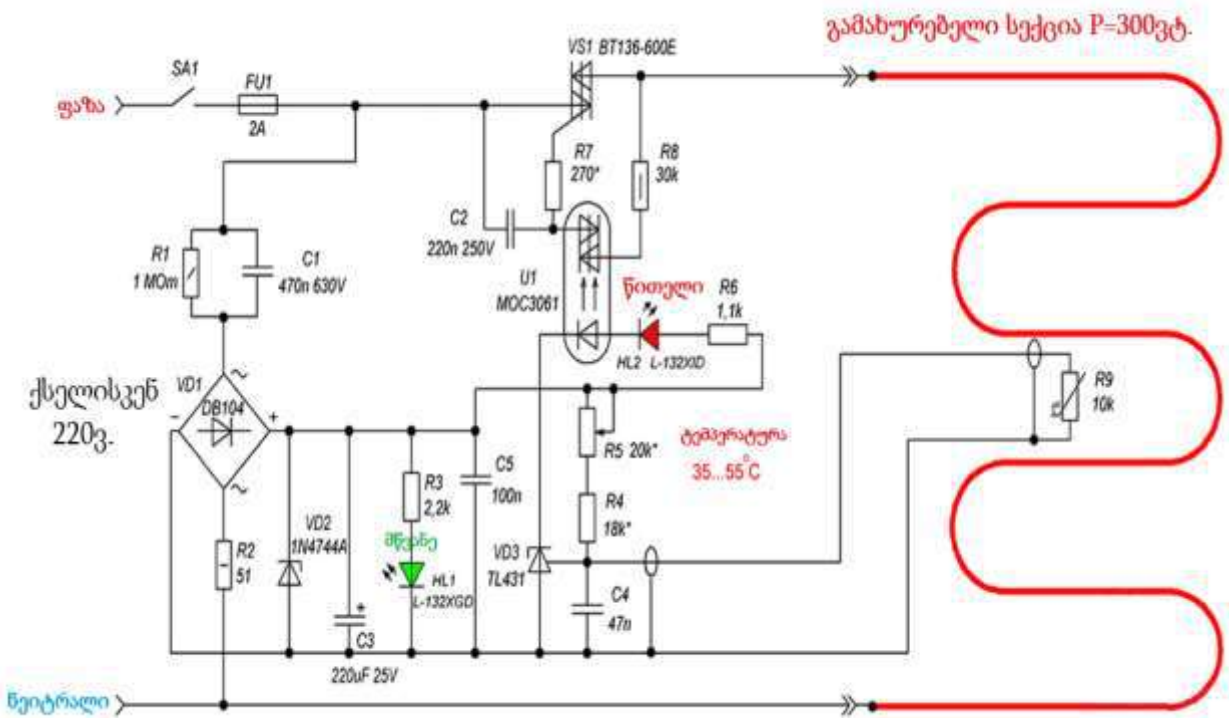
ნახ. 1

განხილული სქემიდან ვხედავთ, თუ როგორ ხდება ჰაერის ტემპერატურის ავტომატური მართვა. მასში კონტაქტური თერმომეტრი ატარებს მცირე დენს, ამიტომ ასეთ სქემებში სასურველია გამოვიყენოთ მაღალმგრძობიარე პოლარიზებული რელეები. იატაკის გამათბობელი თერმორეგულატორი (ნახ.2) უზრუნველყოფს გათბობის სისტემას და გარემოს ტემპერატურას. თავდან ხდება სათავსოს იატაკის გათბობა, საიდანაც სითბო გადაეცემა სათავსოს.

ნახ.2

ამ თვალსაზრისით შეიძლება გამოვიყენოთ ეს მოწყობილობები არ წვავს ჟანგბადს, და შესაბამისად, არ არის სახიფათო ფრინველებისათვის.

თბილი იატაკის თერმორეგულატორი არის მოწყობილობა, რომელიც აკონტროლებს თვით

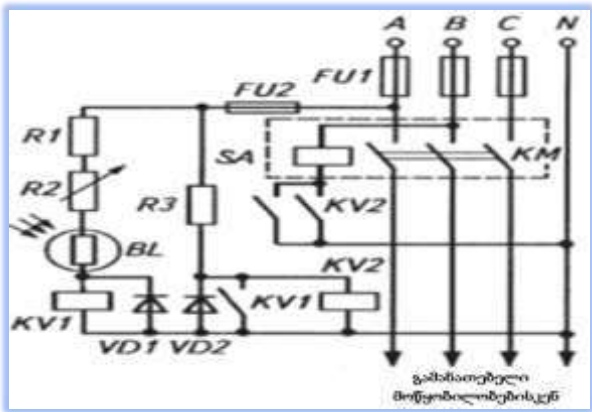


გათბობის სისტემასა და გარემოს ტემპერატურას. მოწყობილობის ჩართვისას ქსელური ცვლადი ძაბვა კვების ბლოკით (R1, R2, C1, C3, C5, VD1, VD2) იმართება და სტაბილიზდება 15 ვ-მდე, ხოლო მწვანე შუქდიოდი ინდიცირებას უკეთებს ძაბვის არსებობას. R4, R5 და R9 ელემენტებისაგან შედგენილი გამყოფი იძლევა ჩართვა - გამორთვის ზღურბლს და იატაკი ცივდება. მაქსიმალური წინაღობა აქვს 10 კო-მდე, ამასთან TL431 სტაბილიტრონის მარეგულირებელ შესასვლელს R4, R5 გამყოფით მიეწოდება 2,5 ვ-ზე მეტი ძაბვა და სტაბილიტრონი იღება. დენი გადის წრედში VD3, R6, HL2, U1, ოპტოსიმისტორი ღიაა და წითელი დიოდი ინდიცირებას უკეთებს ამას. ღია ოპტოსიმისტორი U1 ქმნის გამყოფს R7, R8, C2, VS1 ირთვება და იატაკი ცხელდება. როდესაც იატაკის ტემპერატურა იზრდება, R9 სენსორის



(თერმისტორის) წინაღობა მცირდება და შედეგად მომენტში, როცა ძაბვა სტაბილიტრონის მარეგულირებელ შესასვლელზე ხდება 2,5 ვ-ზე ნაკლები, TL431 იკეტება, მის მერე იკეტება ოპტოსიმიტორი და სიმიტორი, წითელი დიოდი ქრება, ხოლო სექციის გაცხელება ითიშება. რამდენიმე გრადუსით იატაკის გაცივების მიხედვით პროცესი მეორდება და მოწყობილობა აკავებს მოცემულ ტემპერატურას [2, 3].

ფრინველების სათავსოში მუდმივი სიდიდის ტემპერატურის შენარჩუნების გარდა, ყურადღება უნდა მიექცეს მისი განათების სტაბილობას. ამ თვალსაზრისით ყველაზე უფრო დიდი პრობლემები იქმნება ზამთრის პერიოდში, როდესაც, განსხვავებით ზაფხულისაგან, საჭირო ხდება სათავსოში სპეციალური მიკროკლიმატის შექმნა.



მისი ავტომატიზაცია გარკვეულ სიმნელებთანაა დაკავშირებული, რომლის მართვის მიზნით გამოიყენება ფოტორელე. მისი საშუალებით შეიძლება მოვახდინოთ დღის შუქის გახანგრძლივება შემოდგომა-ზამთრის პერიოდში (ნახ. 3).

ნახ.3

ფოტორელეს აგრეთვე შეიძლება დაემატოს (ფოტორეზისტორი), რომელიც ახდენს შუქის ჩართვას. ასეთ რელეში ფოტორეზისტორი BL ჩართულია მიმდევრობით KV1 რელესთან. შემსრულებელი KV2 რელე მართავს KM კოჭას. R2 რეზისტორის წინააღობის ცვლილებით KV1 რელეს კოჭას წრედში ახდენენ რელეს აწყობას. R1 წარმოადგენს სქემის დამცველ რეზისტორს.

იმ შემთხვევაში, როდესაც განათება საკმარისია, BL ფოტორეზისტორის წინააღობა მცირეა, ხოლო KV2 რელეს კოჭას წრედში დენი მეტია ამუშავების დენზე და ის ჩართულია, რელე KV2 გამორთულია. დღის სინათლის მკვეთრად შემცირებისას ფოტორეზისტორის წინააღობა მცირდება რელეს გამორთვის დენის მნიშვნელობამდე. შემდეგ KV1 რელეს კონტაქტები ითიშება, ხოლო KV2 რელეს კონტაქტების ჩართვა, რომელიც მართავს KM მაგნიტურ გამშვებს. SA-თი ხდება განათების დანადგარების მართვა [3]. სათავსოს სწორი კლიმატური პირობები აუმაჯობებს ფრინველების მიერ მოტანილ სარგებლიანობას. ავტონომიურ რეჟიმში ავტომატიზებული სათავსო იძლევა დისტანციური მართვის შესაძლებლობას და შეიძლება გახდეს „ჭკვიანი სახლის“ დამატება.

### 3. დასკვნა

როგორც აღნიშნული იყო, ფრინველების შენახვის ერთ-ერთი საჭირო პირობაა მათ სათავსოში მიკროკლიმატის შექმნა, ანუ ტემპერატურის ავტომატიზაცია. დასაწყისში განხილულია ტემპერატურის რეგულირების ერთ-ერთი ასეთი სქემა. შემდეგ განიხილება სათავსოს გათბობა იატაკის საშუალებით, საიდანაც სითბო გადაეცემა სათავსოს. ამ მიზნით გამოიყენება თერმორეგულატორი, რომლის ავტომატიზაცია გარკვეულ სიმნელებთანაა დაკავშირებული.

ყურადღება უნდა მიექცეს აგრეთვე ფრინველების სათავსოში მისი განათების სტაბილობას. ამ თვალსაზრისით პრობლემები იქმნება ზამთრის პერიოდში, როდესაც, განსხვავებით ზაფხულისაგან, საჭირო სათავსოში სპეციალური მიკროკლიმატის შექმნა.

#### ლიტერატურა – References – Литература

1. Bychkova A.A. Tomsk Polytechnic Institute. Automatic temperature regulation system in the storage room with flow ventilation (in Russian)
2. Dyadic V.F. , Baidali S.A. , Krinitcini N.S. Theory of automatic control. Help Guide. Tomsk Polytechnic University of National Research -- Publishing House of Tomsk Polytechnic University. Page 2011 – 196. (in Russian) .
3. [https://minifermer.ru/page\\_418.html](https://minifermer.ru/page_418.html) (in Russian).

*(სტატია მიღებულია 10.11.2022)*

### AUTOMATION OF THE TEMPERATURE AND LIGHTING OF THE POULTRY HOUSE

Kozashvili Marine, Azmaiparashvili Zaal, Murjikneli Guram,  
Murjikneli Givi, Modebadze Iuri  
Georgian Technical University

kozashvilimarine@gmail.com, z.azmaiparashvili@gtu.ge, gurami.murjikneli@gtu.ge,  
g.murjikneli@gtu.ge, iura.modebadze@gtu.ge

#### Summary

In this work is discussed, keeping the air as well as the floor temperature and lighting, at a constant level in winter and summer, in the poultry house. Different types of thermoregulators and automatic devices are used for this purpose.

*(Received 10.11.2022)*

### АВТОМАТИЗАЦИЯ ТЕМПЕРАТУРЫ И ОСВЕЩЕНИЯ В ПОМЕЩЕНИ ПТИЧНИКА

Козашвили М. , Азмаипарашвили З. , Мурджикнели Гур. ,  
Мурджикнели Г. Модебадзе Ю.  
Грузинский Технический Университет

kozashvilimarine@gmail.com, z.azmaiparashvili@gtu.ge, gurami.murjikneli@gtu.ge,  
g.murjikneli@gtu.ge, iura.modebadze@gtu.ge

#### Резюме

Рассматриваются вопросы сохранения температуры воздуха (а также пола) летом и зимой, а также освещения в помещении птичника на постоянном уровне. Для этого используются терморегуляторы и автоматические устройства разного типа.

*(Поступила в редакцию 10.11.22)*

# შინაური ფრინველების სათავსოს ავტომატური მომარაგება საკვებით

მარინე კოზაშვილი

საქართველოს ტექნიკური უნივერსიტეტი  
kozashvilimarine@gmail.com

## რეზიუმე

შინაური ფრინველების კვების ავტომატური მოწყობილობის დანიშნულებაა მათთვის მშრალი საკვების მიწოდება. დღეისათვის ასეთმა პროცესმა მიიღო ფართო გავრცელება. ისინი მოსახერხებელია იმით, რომ მათ არ სჭირდებათ ფერმერის უშუალო მონაწილეობა ფრინველების სათავსოს დასუფთავებასა და კვებაში. აღნიშნული მოწყობილობით სათავსოში ხდება დროის გარკვეულ შუალედებში საკვების განსაზღვრული რაოდენობის მიწოდება. ეს იძლევა იმის საშუალებას, რომ ფერმერმა მოახერხოს რამდენიმე დღით (კვირით) არ მივიდეს სათავსოში, რაც არ გამოიწვევს ფრინველების კვების რეჟიმის დარღვევას.

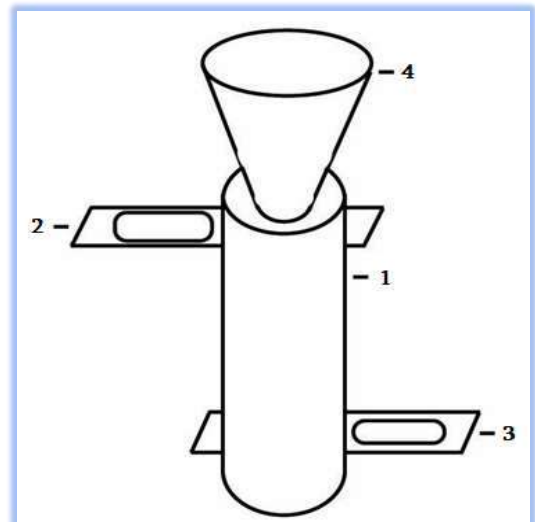
**საკვანძო სიტყვები:** ბუნკერი. ტაიმერი. დოზადორი. ფრინველი. ავტომატიზაცია.

## 1. ძირითადი ნაწილი

დღეისათვის ხშირია მოწყობილობები ბუნკერთა და ტაიმერთ. ისინი მოსახერხებელია იმით, რომ უზრუნველყოფენ ფრინველების ბიოლოგიურ მოთხოვნილებას ფერმერების უშუალო მონაწილეობის გარეშე. აღნიშნული მოწყობილობები შეიძლება იყოს შესრულებული სხვადასხვა მასალისაგან. მაგრამ საკვების არევა არ არის რეკომენდებული და ამ თვალსაზრისით უმჯობესია რამდენიმე კვების მოწყობილობის მოთავსება ერთ კონსტრუქციაში.

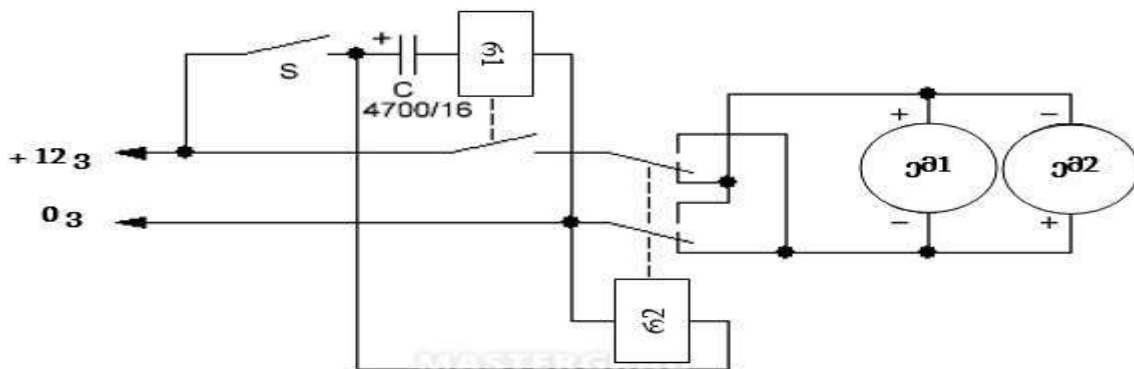
აღნიშნულ მოწყობილობებს შორის ყველაზე მეტად გავრცელებულია: ბუნკერული, პლასტმასის და მეტალის ჭურჭლის, ხის კონსტრუქციის, პედალური და არაპედალური მოწყობილობები ტაიმერთ, დოზატორით და სხვ [1].

ავტომატური მოწყობილობები ტაიმერებით და დოზატორებით ახალი სიტყვაა თანამედროვე ფრინველთა სათავსოს ტექნიკური აღჭურვილობის თვალსაზრისით. ამ მოწყობილობებით ხდება მარცვლოვანი ნარევისა და მშრალი საკვების მიწოდება ფრინველებისათვის. ტაიმერი ახორციელებს საკვების გადაადგილებისა და მიწოდების დროის რეგულირებას. განვიხილოთ საკვების დოზირებული მიწოდების მოწყობილობა, დამყარებული 3-ელექტრომაგნიტების საშუალებით ჩაკეტვა-გახსნის სარქველების მართვაზე და საკვების გადაადგილების ფაზების ხანგრძლივობის რეგულირებაზე ტაიმერის საშუალებით. მის სქემატურ კონსტრუქციას აქვს 1-ელ ნახაზზე მოცემული სახე.



ნახ.1

იგი შედგება პლასტმასის ვერტიკალური მილისაგან - 1, პლასტმასის ორი ზოლოვანი ფირფიტისაგან - 2 და 3, რომლებშიც გაკეთებულია ხვრელები საკვების გავლისათვის. ამ ფირფიტების განთავსება ხდება მილის ქვედა და ზედა ნაწილში გაკეთებულ ჭრილებში ურთიერთსაპირისპიროდ. ფირფიტების გადაადგილებით მარჯვნივ და მარცხნივ ტაიმერით მართვადი ელექტრომაგნიტების საშუალებით შესაძლებელია მილის ზემოთ განთავსებულ ჭურჭელში ჩაყრილი საკვების მოძრაობა მილის გამოსასვლელისაკენ დოზირებული პორციების სახით. პორციაში შემავალი საკვების რაოდენობის ცვლილება შესაძლებელია განხორციელდეს ტაიმერით განსაზღვრული რეჟიმის შესაბამისად. მოწყობილობის გამართვებული ელექტრული სქემა მოცემულია მე-2 ნახაზზე.



ნახ.2

სქემაზე ემ1-ით და ემ2-ით აღნიშნულია ელექტრომაგნიტები, ამასთან ისინი ჩართულია ურთიერთსაპირისპირო პოლარობით, რაც უზრუნველყოფს სარქველების ფირფიტების გადაადგილებას (ჩაკეტვა - გაღებას) ურთიერთსაპირისპირო მიმართულებით. ტაიმერის საშუალებით იმართება რ1 და რ2 რელეები, რომელთა კონტაქტების გადართვა-გადმორთვით მიიღება მუშაობის საჭირო რეჟიმი [1.2].

საკვების დოზირების ასეთი პრინციპი საგრძნობლად ამარტივებს მოწყობილობას. ის შეიძლება იყოს მოთავსებული სათავსოს ძირზე ან სიმაღლეზე ჩამოკიდებული კონსტრუქციის სახით და გააჩნდეს საკვები ღარის შევსების სხვადასხვა სიჩქარე. იგი მზადდება მაღალხარისხოვანი მსუბუქი პლასტმასის მასალისაგან. ტაიმერით მართვად საკვების ავტომატური მიწოდების მოწყობილობას აქვს სათადარიგო ბატარეა, რაც უზრუნველყოფს მის მაღალ საიმედოობას.

ავტომატიკა დოზატორისა და ტაიმერის გამოყენებით მოსახერხებელია იმით, რომ ღარში საკვების მიწოდება ხდება საჭირო დროს. ტაიმერით ხდება ასევე საკვების მიწოდება ფრინველების სათავსოში სასურველი დროითი რეჟიმის შესაბამისად [3].

შინაური ფრინველების კვების ერთ-ერთ რეჟიმს ტაიმერის საშუალებით აქვს შემდეგი სახე (ნახ. 1), რომლის ძირითადი ფუნქციებია:

1. 6 დამოუკიდებელი კვების დღის დაყენების საშუალება (1-დან 6 დღემდე);
2. მინიმალური შუალედი კვებათა შორის - 1 საათი, ხოლო მაქსიმალური - 24 საათი;
3. ბგერითი შეტყობინების ჩაწერის დრო - 6 წმ;
4. ბგერითი შეტყობინება 3-ჯერ ზედიზედ ყოველი კვების წინ.

ამგვარად, ჩვენ შეგვიძლია დავიცვათ შინაური ფრინველების კვების რეჟიმი ტაიმერით განხილული თანმიმდევრობით.

### 3. დასკვნა

შინაური ფრინველების კვებისათვის შეიძლება გამოვიყენოთ ბუნებრივი ავტომატური კვების მოწყობილობები. ამ თვალსაზრისით დღეისათვის ფართოდაა გავრცელებული ავტომატური მოწყობილობები ტაიმერებითა და დოზატორებით, რაც სიახლეა ფრინველთა სათავსოს ტექნიკური აღჭურვილობის თვალსაზრისით. დოზატორის მქონე მოწყობილობა ამარტივებს ამოცანას. ისინი შეიძლება იყოს მოთავსებული სათავსოს ძირზე ან სიმალლეზე ჩამოკიდებული კონსტრუქციის სახით და ჰქონდეს ღარის შევსების სხვადასხვა სიჩქარე. ავტომატიკა დოზატორითა და ტაიმერით მოსახერხებელია იმით, რომ საკვების მიწოდება ღარში ხდება საჭირო დროს. კვების მოწყობილობები დოზატორით მზადდება მაღალხარისხოვანი მსუბუქი პლასტმასისაგან.

#### ლიტერატურა - References - Литература:

1. Diadiki F., Baidal S., Krinitcini N. Theory of Automatic Control. Help Guide. National Research Tomsk Polytechnic University - Tomsk Polytechnic University Publishing House. 2011 – 196 p.
2. Maltsev V., Fromber Z., Iampolski V. Radio and Communications. 1987.
3. Razevich V. Digital Equipment Design System Orcad - M.: Colon - p, 2000.

*(სტატია მიღებულია 10.11.2022)*

### AUTOMATIC FOOD SUPPLY FOR DOMESTIC BIRDHOUSE

Kozashvili Marina

kozashvilimarine@gmail.com

Georgian Technical University

#### Summary

The purpose of the automatic feeding device for domestic birds is to supply them with dry food. To date, such a process has become widespread. They are convenient in that they do not require the direct participation of the farmer in cleaning and feeding the birdhouse. The purpose of the mentioned device is to deliver a certain amount of food at certain time intervals. This allows the farmer to leave the barn for several days (weeks) without disturbing the birds feeding regime.

*(Received 10.11.2022)*

### АВТОМАТИЧЕСКОЕ СНАБЖЕНИЕ ПИЩЕЙ ПОМЕЩЕНИЯ ДОМАШНИХ ПТИЦ

Козашвили М.

Грузинский Технический Университет

kozashvilimarine@gmail.com

#### Резюме

Целью автоматического снабжения помещения домашних птиц является подача им сухой пищи. На сегодняшний день такой процесс получил большое распространение. Он удобен тем, что нет необходимости непосредственно вмешаться в уборку помещения и питания птиц. Указанными устройствами в помещении в разные интервалы времени происходит подача определенного количества пищи. Это дает возможность фермеру не появляться в помещении несколько дней, что не вызовет нарушения режима питания.

*(Поступила в редакцию 10.11.22)*

## მორსათრევი თვითმტვირთავი აგრეგატის განივი დინამიკური მდგრადობის გამოკვლევა ვაკეზე

ნინო ბჟალავა, მალხაზ ბიბილური, ნანა მამისაშვილი  
საქართველოს ტექნიკური უნივერსიტეტი  
nino.bzhalava@gtu.ge; bibiluri@gtu.ge; n.mamisashvili@gtu.ge

### რეზიუმე

განხილულია თეორიული და ექსპერიმენტული კვლევების მეთოდები, რომლებიც მოიცავს: მოდერნიზებული მორსათრევი თვითმტვირთავი აგრეგატი ძარა-ისრის ჰიდრო-მექანიკური სისტემით აღჭურვილი მექანიზმების აგების და სტრუქტურული, აგრეთვე მორსათრევი აგრეგატების განივი მდგრადობის კვლევების საკითხებს. გამოთვლილია მორსათრევი თვითმტვირთავი აგრეგატის განივი დინამიკური მდგრადობის დიფერენციალური განტოლება ვაკეზე კრიტიკული სიჩქარით. ფიზიკური ფორმულების საშუალებით გამოთვლილია ინერციის მომენტები.

**საკვანძო სიტყვები:** მორსათრევი. სიჩქარე. აგრეგატი. ძარა. დინამიკური მდგრადობა. დიფერენციალურ განტოლება. ინერციის მომენტი.

### 1. შესავალი

მორსათრევი თვითმტვირთავი აგრეგატის (მთა) განივი დინამიკური მდგრადობის გამოსაკვლევად ვადგენთ მთ აგრეგატზე მოქმედი ყველა ძალების მომენტების ჯამურ განტოლებას, მთ აგრეგატის ტრაქტორის მუხლუხების საყრდენი ზედაპირის წიბოზე გამავალი  $X_0X_0$ ღერძის მიმართ, რომელიც მთ აგრეგატის გადაბრუნების დაწყებიდან გადაბრუნების გარდაუვალ მდგომარეობამდე არის მთ აგრეგატის განივი დინამიკური მდგრადობის დიფერენციალური განტოლება.

### 2. ძირითადი ნაწილი

აგრეგატის განივი დინამიკური მდგრადობის დიფერენციალურ განტოლება ვაკეზე კრიტიკული სიჩქარით  $V_3 = 4,07$  მ/წმ, მინიმალური რადიუსით  $R=2,00$  მ მობრუნების დროს, როცა კრიტიკული ცენტრიდანული აჩქარება  $a_3=8,3$  მ/წმ<sup>2</sup>-ია.

ანგარიშის შედეგად მივიღეთ, რომ მორსათრევი თვითმტვირთავი აგრეგატის განივი დინამიკური მდგრადობა, როცა აგრეგატის კრიტიკული სიჩქარე  $V_3 = 4,07$  მ/წმ, ცენტრიდანული აჩქარება  $a_3=8,3$  მ/წმ<sup>2</sup>, აგრეგატის მდგრადობის პირობის მიხედვით გვექნება:

$$191364=146974+44237;$$

$$191364 < 191212.$$

ამრიგად, დავადგინეთ, რომ აგრეგატის მდგრადობის მეორე პირობის თანახმად აგრეგატის კრიტიკული სიჩქარით მრუდწირული დროს იწყებს გადაბრუნებას.

განვსაზღვროთ  $t_5$  დრო, რომელიც შეესაბამება აგრეგატის ბრუნვის კუთხეს  $X_0X_0$ ღერძის გარშემო გადაბრუნების დაწყებიდან გადაბრუნების გარდაუვალ მდგომარეობამდე. ამისათვის საჭიროა ვიპოვოთ  $\Delta\phi_5$  კუთხის მაქსიმალური მნიშვნელობა, რომელიც აგრეთვე შეესაბამება

აგრეგატის  $X_0X_0$  ღერძის გარშემო ბრუნვის კუთხეს გადაბრუნების გადაბრუნების დაწყებიდან გადაბრუნების გარდაუვალ მდგომარეობამდე:

აგრეგატის სტატიკური მდგრადობის პირობიდან განვსაზღვროთ  $\Delta\varphi_5$  კუთხის მაქსიმალური მნიშვნელობა, როცა ვაკეზე მოძრავი მთა-ის წონებით გამოწვეული გამაწონასწორებელი ძალების მომენტების ჯამი ნულის ტოლია.

მივიღეთ, რომ ვაკეზე მთა-ის კრიტიკული სიჩქარით  $V_5 = 4,07$  მ/წმ მინიმალური რადიუსით მობრუნების დროს, როცა კრიტიკული ცენტრიდანული აჩქარება  $a_3 = 8,3$  მ/წმ<sup>2</sup>, ხოლო აგრეგატის  $X_0X_0$  ღერძის გარშემო ბრუნვის კუთხე გადაბრუნების დაწყებიდან გადაბრუნების გარდაუვალ მდგომარეობამდე  $\Delta\varphi_{5max} = 31^{\circ}31'$ , აგრეგატი არამდგრადია. აგრეგატის გადაბრუნების მთელი პროცესი დაწყებიდან დამთავრებამდე მიმდინარეობს პირველი რეჟიმის ერთ პერიოდში, რაც დაიწყო და დასრულდა ბორტთან შეკრულას შეჯახების გარეშე 0,57 წმ/ში.

განვსაზღვროთ ძარაზე შეკრულას გადაადგილების დროის პერიოდი მეორე რეჟიმის დროს საწყის მდგომარეობიდან ბორტთან შეკრულას შეჯახებამდე. ვიცით რა, შეკრულადან ბორტამდე მანძილი  $l_t = 0,5$  მ, მაშინ

$$t_t = \sqrt{\frac{2l_t}{a_3}} = \sqrt{\frac{2 \cdot 0,5}{8,3}} = 0,35 \text{ წმ.}$$

მივიღეთ, რომ აგრეგატის კრიტიკული სიჩქარით მინიმალური რადიუსით  $R = 2,0$  მ მობრუნების დროს, როცა ცენტრიდანული აჩქარება  $a_3 = 8,3$  მ/წმ<sup>2</sup>,  $t_5 = 0,57$  წმ აგრეგატის გადაბრუნების დაწყებიდან გადაბრუნების გარდაუვალ მდგომარეობამდე კუთხე  $\Delta\varphi_5 = 31^{\circ}31'$ . მივიღეთ, აგრეთვე, რომ  $\Delta\varphi_5 = 31^{\circ}31'$ , რაც შეესაბამება აგრეგატის გადაბრუნების დაწყებიდან აგრეგატის გადაბრუნების გარდაუვალ მდგომარეობამდე კუთხის მნიშვნელობას; რაც სრულდება 0,57 წმ-ში; ვიცით, რომ აგრეგატის გადაბრუნების დაწყებიდან ბორტთან შეკრულას შეჯახებამდე დროის მნიშვნელობა  $t_t = 0,35$  წმ, ვპოულობთ კუთხეს გადაბრუნების დაწყებიდან ბორტთან შეკრულას შეჯახებამდე:

$$\Delta\varphi'_5 = \frac{t_t \cdot \Delta\varphi_5}{t_5} = \frac{0,35 \cdot 31^{\circ}31'}{0,57} = 19^{\circ}43',$$

ხოლო მეორე რეჟიმის მეორე პერიოდში აგრეგატის  $X_0X_0$  ღერძის გარშემო ბრუნვის კუთხე.

$$\Delta\varphi_6 = \Delta\varphi_5 - \Delta\varphi'_5 = 31^{\circ}31' - 19^{\circ}43' = 11^{\circ}48'.$$

მაშასადამე, კრიტიკული სიჩქარით მოძრაობის დროს, როცა ცენტრიდანული აჩქარება  $a_3 = 8,3$  მ/წმ<sup>2</sup>, ვაკეზე მინიმალური რადიუსით მობრუნების დროს აგრეგატი ცენტრიდანული აჩქარების ნაზრდის გაუთვალისწინებლადაც დაკარგავს მდგრადობას და მობრუნების საწინააღმდეგო მიმართულებიდან გადაბრუნდება.

შემდეგ მთა-ის განივი დინამიკური მდგრადობის გამოკვლევა ჩავატაროთ ბორტთან შეკრულას შეჯახების გათვალისწინებით, როცა აგრეგატის გადაბრუნების პროცესი მიმდინარეობს მეორე რეჟიმის პირველ და მეორე პერიოდებში:

- აგრეგატის გადაბრუნების დაწყებიდან ბორტთან შეკრულას შეჯახებამდე;
- აგრეგატის ბორტთან შეკრულას შეჯახების შემდეგ გადაბრუნების გარდაუვალ მდგომარეობამდე.

ცნობილია, რომ აგრეგატი გადაბრუნების დაწყებიდან გადაბრუნების გარდაუვალ მდგომარეობამდე ანდომებს  $t_5=0,57$  წმ, ხოლო გადაბრუნების დაწყებიდან ბორტთან შეკრულას შეჯახებამდე დრო  $t_\ell=0,35$  წმ, მაშინ დროის პერიოდი ბორტთან შეკრულას შეჯახებიდან აგრეგატის გადაბრუნების გარდაუვალ მდგომარეობამდე

$$t_6 = t_5 - t_\ell = 0,57 - 0,35 = 0,22 \text{ წმ.}$$

მივიღეთ, რომ მთა-ის განივი დინამიკური მდგრადობის მეორე რეჟიმის მეორე პერიოდში დრო, რაც შეესაბამება ბორტთან შეკრულას შეჯახების შემდეგ პერიოდს გადაბრუნების გარდაუვალ მდგომარეობამდე  $t_6=0,22$  წმ-ს, როცა აგრეგატის ბრუნვის ცენტრიდანული აჩქარება  $a_j=8,3$  მ/წმ<sup>2</sup>, აგრეგატის ბრუნვის კუთხე  $\Delta\phi_6=11^\circ48'$ , მაშინ გადაბრუნების პროცესის მთლიანი დროის პერიოდი გადაბრუნების დაწყებიდან გადაბრუნების გარდაუვალ მდგომარეობამდე  $t_5=0,57$  წმ-ს. ამასთან, დრო  $t_\ell=0,35$  წმ-ს შეესაბამება გადაბრუნების დაწყებიდან ბორტთან შეკრულას შეჯახებამდე დროის პერიოდს, რაც მიმდინარეობს გადაბრუნების პროცესის მეორე რეჟიმის პირველ პერიოდში, როცა აგრეგატის  $X_0X_0$  ღერძის გარშემო ბრუნვის ცენტრიდანული აჩქარება  $a_j=8,3$  მ/წმ<sup>2</sup>, ბრუნვის კუთხე  $\Delta\phi'_6=19^\circ43'$ .

მაშასადამე, ორივე პერიოდის გავლას რასაც შეესაბამება  $X_0X_0$  ღერძის გარშემო ბრუნვის კუთხე  $\Delta\phi_7=31^\circ31'$  აგრეგატი ანდომებს  $0,57$  წმ-ს, რაც შეესაბამება დროის პერიოდს აგრეგატის გადაბრუნების დაწყებიდან გადაბრუნების გარდაუვალ მდგომარეობამდე, ბორტთან შეკრულას შეჯახებით გამოწვეული ცენტრიდანული აჩქარების ნაზრდის  $\Delta a'$ -ის გაუთვალისწინებლად.

გადავდივართ მთა-ის განივი დინამიკური მდგრადობის გამოკვლევაზე მეორე რეჟიმის მეორე პერიოდისათვის ჯამური ცენტრიდანული აჩქარებით ბორტთან შეკრულას შეჯახებით გამოწვეული ცენტრიდანული აჩქარების ნაზრდის გათვალისწინებით.

ფორმულების გამოანგარიშების შედეგად მივიღეთ, რომ მთ აგრეგატის კრიტიკული სიჩქარით  $V_{3*}=4,07$  მ/წმ ვაკეზე მრუდწირული მოძრაობის დროს, ძარაზე შეკრულას შუაში დაუფიქსირებელ მდგომარეობაში, როცა აგრეგატის ჯამური ცენტრიდანული აჩქარება  $a'_{3*}=8,55$  მ/წმ<sup>2</sup> ცენტრიდანულის აჩქარების ნაზრდის  $\Delta a'=0,25$  მ/წმ<sup>2</sup> გათვალისწინებით მეორე რეჟიმის მეორე პერიოდში, რომელსაც შეესაბამება აგრეგატის  $X_0X_0$  ღერძის გარშემო ბრუნვის კუთხე  $\Delta\phi_6=11^\circ48'$ , გაივლის  $t_7=0,2$  წმ. მაშასადამე, მთ აგრეგატის  $X_0X_0$  ღერძის გარშემო ბრუნვის კუთხე გადაბრუნების დაწყებიდან გადაბრუნების გარდაუვალ მდგომარეობამდე შეადგენს  $\Delta\phi_7=31^\circ31'$ , რაც მიმდინარეობს  $t_5=0,57$  წამის განმავლობაში და მთლიანად ხდება ბორტთან შეკრულას შეჯახების გარეშე; ამასთან, როდესაც ბორტთან შეკრულას შეჯახება ხდება, ე.ი. მეორე რეჟიმში გადაბრუნების პროცესის მიმდინარეობის დროს, კუთხე გადაბრუნების დაწყებიდან ბორტთან შეკრულას შეჯახებამდე შეადგენს  $\Delta\phi'_7=19^\circ43'$  რა სრულდება  $t_\ell=0,35$  წმ განმავლობაში და ხდება მეორე რეჟიმის დროს, ხოლო დარჩენილი ბრუნვის კუთხე  $\Delta\phi_6=11^\circ48'$  მიმდინარეობს მეორე რეჟიმის მეორე პერიოდში  $t_6=0,22$  წამში ცენტრიდანული აჩქარების ნაზრდის  $\Delta a'=0,25$  მ/წმ<sup>2</sup> გაუთვალისწინებლად და  $t_7=0,2$  წმ. ცენტრიდანული აჩქარების ნაზრდის გათვალისწინებით; ასეთ მცირე სხვაობა დროში  $t_7-t_6=0,02$  წმ გამოწვეულია აგრეგატისა და შეკრულას მასებს შორის დიდი სხვაობით.



### 3. დასკვნა

მთ აგრეგატის განივ დინამიკურ მდგრადობაზე ჩატარებულმა კვლევებმა მიგვიყვანა შემდეგ საერთო ბოლო დაკვნებამდე:

– მთ აგრეგატის ვაკეზე მაქსიმალური სიჩქარით  $V_{\max} = 2,84$  მ/წმ, მინიმალური რადიუსით  $R = 2,0$  მ მობრუნების დროს, როცა ძარაზე შუაში შეკრულა დაუფიქსირებელ მდგომარეობაშია, და ცენტრიდანული აჩქარება  $a = 4,03$  მ/წმ<sup>2</sup>, აგრეგატი მყარად მდგრადია, როგორც ცენტრიდანული აჩქარების ნაზრდის  $\Delta a = 0,13$  მ/წმ<sup>2</sup> გაუთვალისწინებლად, ასევე ცენტრიდანული აჩქარების ნაზრდის – გაუთვალისწინებით;

– მთ აგრეგატის ვაკეზე კრიტიკული სიჩქარით  $V_j = 4,07$  მ/წმ და მიმდინარეობს პირველ და მეორე რეჟიმებში, ვაკეზე მინიმალური რადიუსით მობრუნების დროს ორივე რეჟიმის შემთხვევაში აგრეგატი არამდგრადია და მობრუნების საწინააღმდეგო მიმართულებით გადაბრუნდება, როგორც ცენტრიდანული აჩქარების ნაზრდის გაუთვალისწინებით, ასევე – გაუთვალისწინებლად.

#### ლიტერატურა – References :

1. Balamtsharashvili Z., Chitidze Z., Dundua P., Kokaia G. (2005). Woodworking machinery and equipment. Tbilisi. Georgian Technical University. -268 p. (in Georgian)
2. Balamtsrashvilo Z., Chitidze Z., Mosulishvili D., Tkemaladze R. (2015). Search and research of self-loading logging unit with a hydro-mjechanical drive. Monograph Tbilisi. Technical University. 276 p. (in Georgian)
3. Chelidze-Tkeshelashvili N., Balamtashvili Z., Tkemaladze R., Mosulishvili D., Darakhvelidze G. (2016). Investigation of the transverse stability of the movement of an unloaded tractor unit during curvilinear movement on the plain. Transport and machine building, a collection of educational-methodical and scientific-research works. Tbilisi, No. 1(35) p. 37-44 (in Georgian)
4. Chelidze-Tkeshelashvili N., Balamtashvili Z., Tkemaladze R., Mosulishvili D., Darakhvelidze G. (2016). A study of the stability of the loading self-loading self-loading aggregate during the curvilinear movement on the plain. Transport, machinery and construction, teaching-methodical and scientific-research works collection. Tbilisi, No. 2(36), p. 56-64 (in Georgian)

*(სტატია მიღებულია 15.11.2022)*

## CROSS SECTION OF A SELF-LOADING TRACTOR DYNAMIC STABILITY STUDY

Bzhalava Nino, Bibiluri Malkhaz, Mamisashvili Nana  
Georgian Technical University

#### Summary

The article describes the methods of theoretical and experimental research, which include: issues of construction of mechanisms equipped with the hydromechanical system of the body-arrow of the modernized mountain and the issues of structural and transverse stability of mowing units. Moments of inertia are calculated using physical formulas.

*(Received 15.11.2022)*

## იონმეტრული სენსორები და მათი გამოყენების თავისებურებანი

თამარ ქოზაშვილი, ზაალ აზმაიპარაშვილი, ვლადიმერ ფადიურაშვილი,  
ოლღა მელიქიძე

საქართველოს ტექნიკური უნივერსიტეტი

kozashvili.t@gtu.ge, z.azmaiparashvili@gtu.ge, v.padiurashvili@gmail.com,  
olikomelkidze@mail.ru

### რეზიუმე

განხილულია ელექტროდული სისტემები, მათი სტრუქტურები, მუშაობის პრინციპები. განსაკუთრებული ადგილი აქვს დათმობილი ელექტროდების გამოყენებას პრაქტიკაში. ნაჩვენებია გამოყენების მაგალითები. მიმოხილულია ელექტროქიმიური მეთოდები შესაბამისი საზომი საშუალებებისათვის.

**საკვანძო სიტყვები:** ელექტროდი. იონები. პოტენციომეტრია. კონცენტრაცია.

### 1. შესავალი

ელექტროქიმიურ პროცესებში მიმდინარე რეაქციების შესწავლა, კვლევა და საერთოდ, ელექტროქიმიური პროცესების ტექნოლოგიური სქემები წარმოუდგენელია ელექტროდული სისტემების გარეშე. ელექტროდი ის პირველადი უჯრედი, ელემენტი თუ კვანძია, რომლებიც საშუალებას იძლევა გაიზომოს (განისაზღვროს) ელექტროქიმიური პროცესების მახასიათებლები. ელექტროდულ სისტემებს განსაკუთრებით მნიშვნელოვანი ადგილი უკავია ქიმიურ პროცესებისას წარმოქმნილი იონების გაზომვა-განსაზღვრაში.

ელექტროდების განვითარების ისტორია დაკავშირებულია ფიზიოლოგიური პროცესების შესწავლასა და კვლევასთან, ცალკეულ ნაწილებს შორის ელექტრული პოტენციალების სხვაობის გათვალისწინებით [1,2].

ცნობილია, რომ ანალიზურ ხელსაწყოთმშენებლობაში გაზომვის მეთოდები და საშუალებები დაფუძნებულია ნივთიერებათა ქიმიური შედგენილობის დადგენის ცნობილ ხერხებზე, რომელთა ცოდნა აუცილებელია გაზომვის თანამედროვე საზომი საშუალებების მუშაობის პრინციპის გასაგებად სხვადასხვა მეთოდების კომბინირების გზით.

### 2. ძირითადი ნაწილი

*ელექტროდული სისტემა* შედგება მინის (საზომი), დამხმარე ელექტროდისა და ელექტროლიტური გასაღებისაგან. მინის ელექტროდების ძირითადი დამახასიათებელი ელემენტია ელექტროდული მინა და მინის ელექტროდის შიგა კონტაქტური სისტემა:

- 1) ხსნარის ტემპერატურის მერყეობისას კონკრეტულ ელექტროდს უნდა ჰქონდეს მაღალი სტაბილობა და პოტენციალის დაბალი მგრძობელობა;
- 2) ხსნარი, რომლითაც შევსებულია მინის ელექტროდის ღრუ, უნდა უზრუნველყოფდეს ელექტროდის ვარგისიანობის ხანგრძლივ ვადას და pH-ის მინიმალურ ცვლილებას, ელექტროდის კორპუსის მინის გამოტუტვის ხარჯზე;
- 3) ელექტროდული სისტემის იზოპოტენციალური წერტილები პრაქტიკულად უნდა მოიცავდეს pH -ის განსაზღვრის მთელ ინტერვალს.

გამოკვლევულ კონტაქტურ ელექტროდებში, ლითონური კონტაქტური ელექტროდის სახით გამოყენებულია ვერცხლის, სპილენძის, კადმიუმის, ნიკელის, თუთია ტალიუმის, პლატინის ამაღლგამა და სხვა ლითონები. საუკეთესო შედეგები მიღებულია შემდეგი სისტემებისთვის:

- 1) Ag/AgCl; 0,1 ნ HCl;
- 2) Ag/AgCl; 0,01 – 1,0 ნ NaCl; 1,0 ნ აცეტატური ბუფერი ფარდობით მჟავა/Na-ის აცეტატი = 1:30;
- 3) Ag/0,01 H [AgBr<sub>3</sub>]<sup>-2</sup>.

პოტენციალის მაღალი სტაბილურობა და პირველი ორი სისტემის პოტენციალის დაბალი მგრძობელობა აისახება პოტენციალების დაბალი ტემპერატურული კოეფიციენტებით და ძნელად ხსნადი მარილების ხსნადობის (ვერცხლის ქლორიდი) მნიშვნელოვანი ზემოქმედებით, ხსნარში ქლორის იონების არსებობის გამო.

ქლორის იონების (ქლოროვანი ნატრიუმის) კონცენტრაციის, pH და აცეტატის ბუფერის შედგენილობის არჩევს შესაძლებლობა უზრუნველყოფს pH-ის და ელექტროდული სისტემების ფართო სპექტრს.

ექსპერიმენტული კვლევები ტემპერატურული დამოკიდებულების პოტენციალის ნიშნების ფაზების გაყოფის საზღვრამდე, რომლებიც აღმოცენდება მინის ელექტროდების შიგნით, ხსნარების კომპონენტების სხვადასხვა თანაფარდობისას (ქლორიდ-იონი, ძმარმჟავა, ნატრიუმის აცეტატი) ამ კომპონენტების კონცენტრაციის დიაპაზონში, საშუალებას იძლევა შეირჩეს ნახტომის ტემპერატურული დამოკიდებულების ურთიერთკომპენსაცია [4].

პოტენციალის მგრძობელობა ელექტროდის ტემპერატურის ცვლილებისას ასევე ზრდის დასაშვებ სიდიდეებს; კალომელის ელექტროდების ცნობილი კონსტრუქციები არ უზრუნველყოფს გაზომვის საჭირო სტაბილურობას, ხსნარების ვიზრაციის და ცვლადი ტემპერატურის არსებობისას. აქედან გამომდინარეობს სამი ძირითადი ამოცანა:

- 1) პოტენციალის მცირე მგრძობელობის მქონე კონტაქტური სისტემების შესწავლა 100°C (და მეტი - 150°C ) ტემპერატურამდე;
- 2) ისეთი ელექტროლიტური გასაღების შექმნა, რომელიც უზრუნველყოფს საკონტროლო ხსნართან კონტაქტს და საშუალებას იძლევა დამხმარე ელექტროდი მოაცილოს აპარატს და დაიცვას მაღალი ტემპერატურის მოქმედებისაგან;
- 3) შევსებული დამხმარე ელექტროდის კონსტრუქციის შემუშავება, რომელიც უზრუნველყოფს გაზომვის საჭირო საიმედოობას საწარმოო არეში 100°C ტემპერატურაზე.

ბოლო დროს მნიშვნელოვად ვითარდება იონომეტრია იონსელექციური ელექტროდების გამოყენებით, როგორც ფიზიკურ-ქიმიური კვლევის ახალი დარგი. დღეისათვის შემუშავებულია იონსელექციური ელექტროდების დიდი რაოდენობა, რომლებიც ემყარება ელექტროდული თვისების სხვადასხვა მასალების გამოყენებას.

განსაკუთრებით მნიშვნელობას იძენს იონომეტრია ეკოლოგიის, მედიცინისა და მრეწველობის სხვადასხვა დარგში. იგი შესაძლებლობას იძლევა თვალყური ვადევნოთ გარემოს იონურ ცვლილებას დინამიკაში სწრაფად, ზუსტად და საიმედოდ [3].

იონსელექციური ელექტროდი არის pH-მეტრებისა და იონმეტრების ყველაზე საპასუხისმგებლო ელემენტი, რითაც ლიმიტირებულია მათი საიმედოობა და მიეკუთვნება არააღდგენითი ნაკეთობების კატეგორიას.

დღეისათვის, შედარებით ხშირად გამოიყენება სამი სახის მინის ელექტროდი:

1) წყალბადური ფუნქციით (pH-ელექტროდი). არსებობს ამ სახის 50-მდე სხვადასხვა ტიპის ელექტროდი;

2) ლითონური ფუნქციით (pX-ელექტროდები, სადაც X – Na<sup>+</sup>, K<sup>+</sup>, Ag<sup>+</sup> ან სხვა იონებია). ამ სახის დაახლოებით 30-მდე სხვადასხვა ტიპის ელექტროდი გამოიყენება;

3) გამტარობის ელექტრონული ხასიათით (Eh-ელექტროდები).

ინდიკატორული ელექტროდის ძირითადი ელემენტია იონმგრძობიარე (განსაზღვრული იონებისადმი) მემბრანა, რომელიც შეიძლება სხვადასხვა მასალისაგან დამზადდეს. გამოყენებული მასალის მიხედვით, მემბრანული ელექტროდები იყოფა ოთხ ჯგუფად: მინის ელექტროდები; ელექტროდები მყარი მემბრანით; ელექტროდები თხევადი მემბრანით; ელექტროდები აირმგრძობიერე მემბრანით.

პირველი მემბრანული ელექტროდი იყო მინის ელექტროდი, რომელიც გამოყენებულ იქნა წყალბადიონების კონცენტრაციის გასაზომად pH-ის ფართო ზღვრებში 0-13.

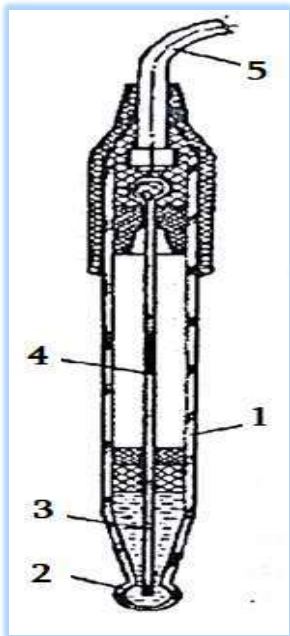
მინის ელექტროდს აქვს ელექტროდული ფუნქცია; მინის ელექტროდულ ფუნქციაში იგულისხმება მისგან დამზადებული ელექტროდის უნარი – პოტენციალის განსაზღვრული ცვლილებით უპასუხოს იმ ხსნარის შედგენილობის ცვლილებას, რომელშიც ჩაშვებულია ელექტროდი. მინის ელექტროდების ელექტროდული თვისებები გამოკვლეულ იქნა იმ საწარმოო მინის გამოყენებით, რომელსაც ჰქონდა მნიშვნელოვანი უარყოფითი მხარეები (ელექტროდული ფუნქციის მოკლე ინტერვალი, არასტაბილური პოტენციალი და სხვ.), რითაც მნიშვნელოვნად იზღუდება pH-მეტრიის შესაძლებლობები. ამ გამოკვლევებით დადასტურდა, რომ ელექტროდული თვისებები მჭიდრო კავშირშია მინის ქიმიურ შედგენილობასთან.

შესაფერისი მინის გამოყენებით ამ ელექტროდით სარგებლობისას, სწორი შედეგების მიღება შესაძლებელია 0 – 10 მდე pH-ის ინტერვალში. pH-ის ამ ფარგლებში მინის ელექტროდის პოტენციალი გამოითვლება ნერნსტის განტოლებით:

$$E = E^0 + \frac{RT}{F} \lg(H^+) \quad (1)$$

მინის ელექტროდი (ნახ.1) წარმოადგენს სპეციალური ელექტროდული მინისაგან 1 (ეს მინა უნდა შეიცავდეს უმეტესად ტუტე ლითონებს Li და Na) დამზადებულ მილს, ბოლოში სპეციალური თხელკედლიანი 15-20 მმ დიამეტრის ბურთულათი 2. მილსა და ბურთულაში მოთავსებულია ეტალონური ხსნარი 3. ის უნდა შეიცავდეს იონებს, რომლებზეც რეაგირებს თვით ელექტროდი (ჩვენს შემთხვევაში, წყალბად იონები) და იონებს, რომლებზეც რეაგირებს შესადარებელი (დამხმარე) ელექტროდი (ჩვეულებრივ იყენებენ ქლორის იონებს). ეტალონური ხსნარის სახით იყენებენ HCl-ის ნორმალურ ან დეცინორმალურ (ე.ი. წყალბად- იონების ცნობილი კონცენტრაციის) ხსნარს. ბურთულის შიგნით მოთავსებულია ქლორ-ვერცხლის ელექტროდი 4. მინის ელექტროდი ჩაშვებული საკვლევ სითხეში. ბურთულის მინის ზედაპირული შრიდან ხდება Na ან Li-ის ერთვალენტიანი იონების გადასვლა როგორც საკვლევ, ისე საკონტროლო ხსნარებში, ხოლო მათი ადგილების ჩანაცვლება - ხსნარიდან წყალბად- იონებით. ამის შედეგად, ბურთულას ზედაპირული შრე ორივე მხრიდან გაჯერებული აღმოჩნდება წყალბად-იონებით და მინის ელექტროდი შეიძენს წყალბადის ელექტროდის თვისებებს. მინის ელექტროდს ათავსებენ საკვლევ ხსნარში და ელექტროლიტური გამტარის საშუალებით უერთებენ სტანტარდულ კალომელის ელექტროდს, მიიღება გალვანური ელემენტი Ag/AgCl/0,1 N HCl/მინა/Hg<sub>2</sub>Cl<sub>2</sub>/Hg.

მინის ელექტროდების საიმედოობა განიხილება მათი მექანიკური სიმტკიცის შესწავლის შემდეგ. მექანიკური სიმტკიცე განისაზღვრება მინის ბურთულისა და მილის სიმტკიცით. თავის მხრივ, მინის სიმტკიცე დამოკიდებულია მრავალ ფაქტორზე: შედგენილობა, ტემპერატურა, ზედაპირის მდგომარეობა, ზომები, გამოცდის პირობები, დეფექტების არსებობა, თერმომედეგობა და სხვა. მინის ელექტროდის დადებითი მხარეა ის, რომ იგი შეიძლება

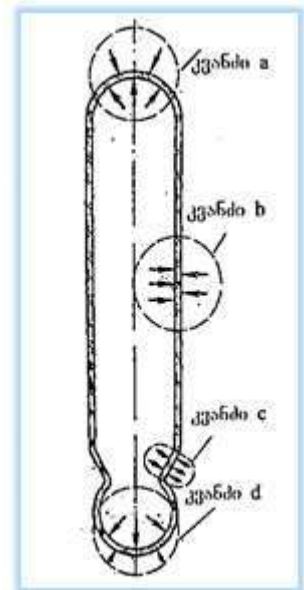


ნახ.1. მინის ელექტროდის საერთო ხედი

გამოყენებულ იქნეს ძლიერი მუანგველის ან აღმდგენის ხსნარებში და წონასწორობა მყარდება სწრაფად; უარყოფითი მხარეა, მინის ბურთულის დაზიანების შესაძლებლობა [4].

მინის ელექტროდის a, b, c და d კვანძებზე ჩატარებულმა წინასწარმა გათვლებმა აჩვენა მინის ელექტროდის ჰერმეტიკულად დახურული კონსტრუქციის გამოყენების პრინციპული შესაძლებლობა (ნახ.2).

დადგენილია, რომ ელექტროდები უძლებს წნევას 300-400 კგ/სმ<sup>2</sup>-მდე შეკუმშვისას, ხოლო გარღვევისას 20-25 კგ/სმ<sup>2</sup>-მდე; ე.ი. მათი რეალური სიმტკიცე 4-5 ჯერ ნაკლებია, ვიდრე გაანგარიშებით, რაც შეიძლება აიხსნას მინაზე ზედაპირული მიკროზარების და ნაფხაჭნების არსებობით, მინის კედლების ფაქტობრივი სისქის შეუსაბამობით და სხვ. ცხადია, რომ მინის ელექტროდებს შეუძლია გაუძლოს მნიშვნელოვან მექანიკურ დატვირთვას.



ნახ.2 მინის ელექტროდის ძირითადი კვანძები, რომლებზეც მოდის დატვირთვა

pH-ის გასაზომად ლაბორატორიულ პრაქტიკაში გამოიყენებოდა მინის ელექტროდები ЭСЛ-43-07, 0-დან 40°C -მდე და ЭСЛ-63-07, 25-დან 100°C ტემპერატურის ინტერვალში მუშაობისთვის.

განვიხილოთ იონსელექციური ელექტროდი. იონსელექციური ელექტროდის ძირითადი ელემენტია ელექტროდული მემბრანა. მემბრანით გამოყოფილ ელექტროლიტების ხსნარებს შორის აღმოცენებულ პოტენციალთა (მემბრანული პოტენციალი  $\varphi$ ) სხვაობა, რომელიც მარტივად ასე გამოისახება:

$$\varphi = \varphi_1 - \varphi_2 + \varphi_{დიფ}, \quad (2)$$

სადაც  $\varphi_1$  და  $\varphi_2$  მემბრანის შიგა და გარე ზედაპირების პოტენციალებია;  $\varphi_{დიფ}$  - დიფუზიური პოტენციალი.

მემბრანული პოტენციალის განტოლება ცნობილია, როგორც ბ. ნიკოლსკის იონური მიმოცვლის განტოლება, ის გამოისახება ასე (თუ ჩავთვლით, რომ  $\varphi_{დიფ} = 0$ ):

$$\varphi = \varphi^0 + \vartheta \lg(\alpha_A + K_{A/B} \frac{U'}{U''} \alpha_B) \quad (3)$$

სადაც  $\theta = 2,3 \frac{RT}{nF}$  (R აირთა უნივერსალური მუდმივა; T - აბსოლუტური ტემპერატურა; F - ფარადის რიცხვი (F=96500); n - იონების მუხტი);  $\varphi_1$  და  $\varphi^0$  - განსაზღვრული და ნორმალური პოტენციალი;  $\alpha_A$ ,  $\alpha_B$  - განსაზღვრავი და შემშლელი იონების აქტივობა;  $K_{A/B}$  - განტოლების კონსტანტა;  $U'$ ,  $U''$  - განსაზღვრავი იონების ძვრადობა ხსნარსა და მემბრანაში. თუ  $pH = \lg \alpha_H$ , მაშინ  $pX = -\lg \alpha_X$  ( $X = Na^+, K^+, Cl^-, Ca^{2+}, NH_4^+, NO_3^-$ , და სხვა იონები).

A და B იონების ელექტროდულ მემბრანას (M) და ხსნარს (p) შორის წონასწორული განაწილების პირობები, მაგალითად,  $H^+$  და  $Na^+$ , pH-ის ან pNa განსაზღვრის შემთხვევაში, გამოისახება ტოლობით:

$$\mu_{Na}^M = \mu_{Na}^p; \mu_H^M = \mu_H^p \quad (4)$$

(3) განტოლების შედეგები  $H^+$  და  $Na^+$  იონებისათვის, თუ:

$$\alpha_H \gg \alpha_{Na} \text{ მაშინ } \varphi = f(pH),$$

$$\alpha_H \ll \alpha_{Na} \text{ მაშინ } \varphi = f(pNa),$$

$$\alpha_H = \alpha_{Na} \text{ მაშინ } \varphi = f(pH \text{ და } pNa),$$

განტოლება (3) აპროგნოზებს წყალბადური (H) ფუნქციიდან ლითონურ ( $Na^+$ ) ფუნქციაზე გადასვლას. მინის ელექტროდის მნიშვნელოვანი პარამეტრია (3) განტოლების კონსტანტა K, რომელიც დამოკიდებულია მინაში და ხსნარში იონების ენერგეტიკულ მდგომარეობაზე. განტოლება (3) ძირითადად ასახავს იონსელექციური ელექტროდების ქცევას სხვადასხვა პირობებში, მაგრამ რიგ შემთხვევაში შეიმჩნევა ნორმალური პოტენციალის  $E_0$  გადანაცვლება და ელექტროდული მემბრანის წყალბადური ფუნქციის დაკარგვა, სხვადასხვა ფიზიკურ-ქიმიური ზემოქმედების შედეგად.

ელექტროდის კონსტრუქცია (ინდიკატორული და დამხმარე) უნდა უზრუნველყოფდეს ექსპლოატაციის სხვადასხვა პირობებს; დღეისათვის მრავალი ელექტროდი არსებობს. ელექტროდის (გადამწოდის) კონსტრუქცია უნდა იყოს მოსახერხებელი გამოსაყენებლად, ჰქონდეს უნიფიკაციის მაღალი დონე.

მინის იონსელექციური ელექტროდები მგრძობიარეა ტუტე ლითონების მიმართ:  $Na^+, K^+, Li^+, Rb^+, Cs^+, Ag^+, Na^+, Tl^+, NH_4^+$ .

pH-ის განსაზღვრისათვის ტუტე სილიკატური მინების ბაზაზე დამუშავებულია იონსელექციური მემბრანები. განსაკუთრებით განსხვავებულია მინის შედგენილობა, რომლისაგანაც მზადდება მემბრანა. დადგენილია, რომ  $Al_2O_3$  -ის შეყვანა მინაში დადებით გავლენას ახდენს მემბრანის სელექციურობაზე.

სპილენძის, ტყვიის კადმიუმის, ნიკელის, რკინის, კობალტის, ვერცხლისწყლის და ზოგიერთი ანიონის განსაზღვრისათვის ბოლო დროს მნიშვნელოვანი წარმატებებია მიღწეული ჰალკოგენიდური მინების დამუშავებისას.

ფუნქციური დანიშნულების მიხედვით, იონსელექციური ელექტროდები იყოფა სამ ჯგუფად: ა) **საზომი ელექტროდები** – დამოუკიდებელი ელექტროდები, რომლებიც განკუთვნილია საინდიკატორო და გამზომ რეჟიმებში მოშაობისათვის; ბ) **დამხმარე ელექტროდები** – წარმოადგენენ შემადარებელ ელექტროდებს, რომელთა საშუალებითაც ხორციელდება ეტალონური ხსნარის საყრდენი პოტენციალის დადგენა და საზომ ელექტროდთან ერთობლივი მუშაობისას გასაზომი სიგნალის (დიფერენციალური

პოტენციალის) გამოიყენება; გ) *კომბინირებული ელექტროდები* – რომლებიც ერთდროულად ასრულებს საზომი და დამხმარე ელექტროდების ფუნქციებს.

ლაბორატორიული იონომეტრიული ხელსაწყოების უმეტესობისათვის კომბინირებული ელექტროდი ყველაზე მოსახერხებელია [5,6]. ის უფრო კომპაქტურია, ვიდრე ელექტროდების წყვილი. ორი სენსორის ნაცვლად ერთი სენსორის გამოყენება ამცირებს ნიმუშში დამაბინძურებლების შეყვანის შესაძლებლობას. კომბინირებული ელექტროდების მოხერხებულობა, უპირველეს ყოვლისა, ლაბორატორიული ანალიზისთვის, დიდი ხანია დაფასებულია მთელ მსოფლიოში. ამ დროისთვის, პოტენციომეტრიული დანადგარების ევროპული მწარმოებლები აწარმოებენ ათობით კომბინირებულ ელექტროდს (ნახ.3) და ცალკეული ელექტროდების მხოლოდ ერთ მოდიფიკაციას (half-cell). ეს გამოწვეულია იმით, რომ მხოლოდ ძალიან იშვიათ შემთხვევებში, ელექტროდების წყვილი არ შეიძლება შეიცვალოს კომბინირებული ელექტროდით [7,8].



ნახ.3. სხვადასხვა მწარმოებლის მიერ დამზადებული ელექტროდები

ამგვარად, იონომეტრიული სენსორები ელექტროქიმიური ელექტროდებია, რომელთა გამოსასვლელი (პოტენციალთა სხვაობა) სიგნალი პირდაპირპროპორციულია ხსნარში არსებული იონების რაოდენობისა. ამ თვისების გამო, იონსელექციური ელექტროდები აქტიურად გამოიყენება ხსნარებში შემავალი სხვადასხვა იონების კონცენტრაციისა და აქტივობის დასადგენად.

### 3. დასკვნა

ელექტროდების პოტენციალის სტაბილურობა და მათი ხანგრძლივობა განსხვავებულ პირობებში დამოკიდებულია იმ გამოყენებული მასალის ქიმიურ მდგრადობაზე, რისგანაც მზადდება ელექტროდები, თუმცა ყოველთვის მასზე არაა დამოკიდებული; მაგალითად, მინის ელექტროდის ქიმიური მდგრადობის შედარებითი შეფასება აუცილებელია ელექტროდული მინის ხარისხის გაუმჯობესების გზების დასადგენად ელექტროდის კონკრეტული შედგენილობის არჩევისას.

პოტენციომეტრიაში ელექტროდული პოტენციალის მნიშვნელობას განსაზღვრავენ ელექტროდების მეშვეობით, რის შედეგადაც განსაზღვრავენ იონების კონცენტრაციას.

იონსელექციური ელექტროდები ფართოდ გამოიყენება საანალიზო ხსნარებში შემავალი სხვადასხვა იონების კონცენტრაციისა და აქტივობის დასადგენად და იმ პროცესების ანალიზისა და კონტროლისთვის სადაც ხდება გამოსაკვლევო ხსნარის იონური შემადგენლობის ცვლილებები.

#### ლიტერატურა - References - Литература:

1. Brouchek F., Ghudushauri T. Photometric methods (help manual). Tbilisi Technical University 2000, 52 p.(in Georgian)
2. Bokuchava L., analytical chemistry course. Tbilisi: Technical University, 2005 204 p. .(in Georgian)
3. Verkhovsky V.N., Smirnov A.D. Techniques of chemical experiments. Volume 2. Tbilisi: Education, 1987, 424 p. .(in Georgian)
4. Fadiurashvili V., Kolomikov S. Methodological instructions for conducting physical-chemical examination in expert laboratories. - Tbilisi: Technical University, 2011. .(in Georgian)
5. T. Dzagania, e. Butskhrikidze, vl. Fadiurashvili, O. Melikidze, T. Kozashvili Some points of view on the development of tools for the determination of nitrates in food products. Automated management systems. Proceedings # 1(32), Vol. 1.1 2021.(in Georgian)
6. Grekovich A.L.,Alexeeva M.V.,Shultze K.-D., Mikhelson K.N.Electroanalysis,2006,v.18,№ 16.(in Russian )
7. <http://www.ngpedia.ru/id612064pl.html>
8. <http://www.anchem.ru/literature/books/02.asp#01>

## IONOMETRIC SENSORS AND FEATURES OF THEIR USE

Tamar Kozashvili, Zaal Azmaparashvili, Vladimer Phadiurashvili, Olga Melikidze  
kozashvili.t@gtu.ge, z.azmaiparashvili@gtu.ge, v.padiurashvili@gmail.com,  
olikomelikidze@mail.ru

Georgian Technical University

### Summary

The article considers electrode systems, their designs, and principles of operation. The use of special electrodes occupies a special place in practice. Examples of use are shown. Electrochemical methods for the corresponding measuring instruments are considered

## ИОНОМЕТРИЧЕСКИЕ ДАТЧИКИ И ОСОБЕННОСТИ ИХ ПРИМЕНЕНИЯ

Козашвили Т., Азмаипарашвили З., Фадиурашвили В., Меликидзе О.

Грузинский технический университет

### Резюме

Рассмотрены электродные системы, их конструкции и принципы работы. Особое место в практике занимает использование специальных электродов. Показаны примеры использования. Рассмотрены электрохимические методы для соответствующих средств измерений.



# ვებ-დიზაინის რამდენიმე თანამედროვე ტენდენცია

ცაცა ნამჩევამე, თამარ აბულაძე  
აკაკი წერეთლის სახელმწიფო უნივერსიტეტი  
tsatsanamg@gmail.com, tamar.abuladze@atsu.edu.ge

## რეზიუმე

განხილულია ვებ-დიზაინის განვითარების თანამედროვე ტენდენციები: გაყოფილი ეკრანის დიზაინი; მრავალფენიანი დიზაინი; Scrollytelling; ანიმაციური ლოგო; ვიდეო; 3D ვიზუალი და VFX. აღნიშნულია, რომ ვებ-დიზაინის ტენდენციები მუდმივად იცვლება. ახალი ტენდენციების წარმოქმნა განაპირობებს იმას, რომ ვებ-დიზაინერებმა ყოველთვის უნდა გაიღრმავონ თავიანთი ცოდნა და შემოქმედება წარმატებული, მომგებიანი პროექტების შექმნისათვის.

**საკვანძო სიტყვები:** გაყოფილი ეკრანი. მრავალფენიანი დიზაინი. ანიმაციური ლოგო. ვიდეო. 3D. VFX.

## 1. შესავალი

ვებ-დიზაინი არის ვებ-გვერდის გაფორმების სტრუქტურა. მისი საშუალებით შესაძლებელია ვებ-გვერდის არა მარტო ესთეტიკური მიმზიდველობის გაზრდა, არამედ ფუნქციონალური შესაძლებლობების გაუმჯობესებაც. ვებ-გვერდის ხარისხი დამოკიდებულია ვებ-დიზაინერის კრეატიულობაზე და პროფესიონალიზმზე.

ვებ-დიზაინის ტენდენციები მუდმივად ცვალებადია. ამიტომ ვებ-დიზაინერებმა ყოველთვის უნდა გაიღრმავონ თავიანთი ცოდნა ვებ-დიზაინის ახალი ფორმების შესახებ.

## 2. ძირითადი ნაწილი

სტატიაში განხილულია ვებ-დიზაინის რამდენიმე თანამედროვე ტენდენცია: 1) გაყოფილი ეკრანის დიზაინი; 2) მრავალფენიანი დიზაინი; 3) Scrollytelling; 4) ანიმაციური ლოგო; 5) ვიდეო; 6) 3D ვიზუალი და VFX.

➤ *გაყოფილი ეკრანის დიზაინი.* იგი არის შედარებით ახალი ტექნიკა ვებ-დიზაინის ინდუსტრიაში. თანამედროვე პირობებში სულ უფრო და უფრო იზრდება გაყოფილი ეკრანის დიზაინის მქონე საიტების რაოდენობა. მათ პოპულარობას აქვს რამდენიმე მიზეზი: 1) კარგი ესთეტიკური ხარისხი; 2) სხვადასხვა ზომის ეკრანზე მორგების შესაძლებლობა; 3) მოსახერხებელი ნავიგაცია; 4) ეკრანის კონკრეტულ ნაწილზე მომხმარებლების ყურადღების მიპყრობა მარტივი დიზაინის ტექნიკის გამოყენებით და სხვ.

გაყოფილი ეკრანის დიზაინის გამოყენება განსაკუთრებით კარგია მაშინ, როდესაც საჭიროა 2 პროდუქტის თუ მომსახურების პოპულარიზაცია. ეს მიდგომა მომხმარებლებს საშუალებას აძლევთ ყურადღება მიაქციონ ორივე პროდუქტს (ან მომსახურებას) და სწრაფად გააკეთონ არჩევანი. გაყოფილი ეკრანის დიზაინის გამოყენება ასევე მოსახერხებელია მაშინ, როდესაც შესაძლებელია რამდენიმე პროდუქტის (ან მომსახურების) ერთ მხარეზე განთავსება, ხოლო მეორე მხარეზე - მათი ძირითადი მახასიათებლებისა.

გაყოფილი ეკრანის დიზაინი არ უნდა გამოვიყენოთ იმ შემთხვევაში, როდესაც კონტენტს აქვს რთული განლაგება. ამიტომ ასეთი დიზაინი იდეალურად შეეფერება მინიმალისტურ ვებ-საიტებს. გაყოფილი ეკრანის დიზაინის მაგალითია საიტი-<https://webflow.com/made-in-webflow/website/varbut> (ნახ.1) [1].

ნახ. 1



➤ *მრავალფენიანი დიზაინი*. ბოლო წლებში ვებ-საიტების პროექტირების პროცესში დიდი ყურადღება ეთმობოდა მინიმალისტურ დიზაინს. ბევრი დიზაინერი ერიდებოდა ვიზუალურად უფრო რთულ დიზაინს, რაც არასწორი მიდგომაა ვებ-საიტების შექმნისათვის. რიგ შემთხვევაში სასურველია მინიმალისტურ დიზაინზე უფრო რთული დიზაინის გამოყენება. მაგალითად, მრავალფენიანი დიზაინით სარგებლობა. ასეთ დიზაინს სჭირდება უფრო მეტი გრაფიკა და წინასწარ დაგეგმვა, ვიდრე მინიმალისტურს.

მრავალფენიან დიზაინში ჩვეულებრივი კომპონენტები, როგორცაა ფოტო-, ტიპოგრაფიული ელემენტები, ერთმანეთზეა ზედდებული. ისინი უზრუნველყოფს ვებ-საიტის ვიზუალურ მთლიანობას. მრავალფენიანი დიზაინის უპირატესობა არის ის, რომ აადვილებს დიდი რაოდენობის კონტენტის ერთ სექციაში ან შეზღუდულ სივრცეში, მაგალითად მობილური მოწყობილობების ეკრანზე მოთავსებას. მრავალფენიანი დიზაინი ქმნის 3D გამოსახულების



ილუზიას და საშუალებას აძლევს მომხმარებელს შეიგრძნოს სივრცე და მიიღოს დადებითი ემოციები. რა თქმა უნდა აღნიშნული დიზაინის შექმნა უფრო მარტივია კლასიკურ 3D-სთან შედარებით. ამის მაგალითია საიტი-<https://www.theshipwrecksurvey.com/> (ნახ. 2) [2].

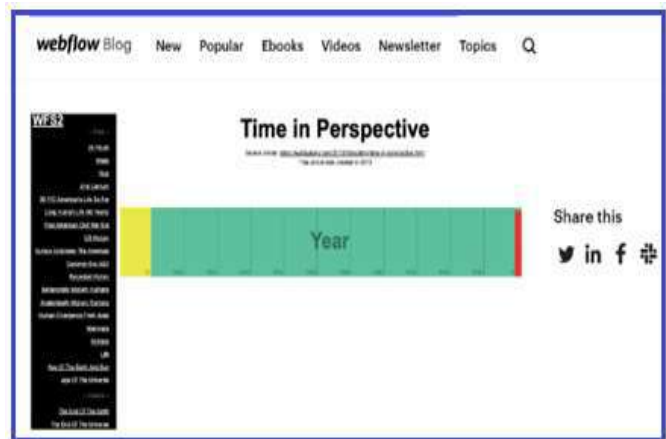
ნახ. 2

➤ *Scrollytelling* – სულ უფრო პოპულარული მიმართულება ხდება ციფრული ინტერფეისის გამოყენებისათვის და ვებ-გვერდზე რთული ამბის გადმოსაცემად. Scrollytelling შეიძლება ჩაითვალოს ვიზუალურ თხრობად. იგი ასევე მოიხსენიება როგორც „ნარატიული ვიზუალიზაცია“ – ელემენტების სერია, რომლებიც გამოჩნდება გარკვეული თანმიმდევრობით.

Scrollytelling დიზაინით ვებ-გვერდი შეიძლება შეიქმნას სხვადასხვა საშუალების გამოყენებით. ესენია: 1) ჰორიზონტალური გადახვევა; 2) გადახვევისას პროგრესის ზოლის გამოყენება; 3) ვიზუალური ელემენტების თანდათანობით გამოჩენა-გაქრობა; 4) პარალაქსის ეფექტი და სხვ.

პარალაქსის ეფექტი ერთ-ერთი პირველი ტრენდია თანამედროვე ვებ-დიზაინში. მასში ანიმაციური ობიექტები აზროვებრივადაა ერთმანეთთან დაკავშირებული და ქმნის საერთო სურათს. პარალაქსის ეფექტი არის ერთგვარი ილუზია, რომელიც იწვევს 3D-ს მსგავს ეფექტს. მასიური ფონის გამოყენებით, ობიექტებზე სიღრმეების დამატებით და მათი მოძრაობაში სწორად მოყვანით ეკრანი გარდაიქმნება ერთგვარ სცენად.

პარალაქსის ეფექტის გამოყენებისათვის საჭიროა გავითვალისწინოთ შემდეგი: 1) პარალაქსის ეფექტების მინიმუმამდე დაყვანა; 2) მოძრაობის შეზღუდვა ეკრანის მცირე ფართობზე; 3) მომხმარებლისთვის პარამეტრების ჩართვა პარალაქსის ეფექტების გამოსართავად. პარალაქსის ეფექტის დიზაინის მაგალითია საიტი-<https://wfs2.webflow.io/time-in-perspective#week> (ნახ.3) [3,4].



ნახ. 3

➤ *ანიმაციური ლოგო* – კომპანიის ბრენდის იდენტობის მაჩვენებელია. ვებ-დიზაინერი ინტერნეტ სივრცეში ლოგოს საშუალებით აყალიბებს ბრენდის იმიჯს. აღნიშნული კი არის წარმატებული მარკეტინგული სტრატეგიის საფუძველი.

თანამედროვე პირობებში ხშირად იყენებენ ანიმაციურ ლოგოს. ქვემოთ ჩამოთვლილია ის უპირატესობანი, რომლებიც მიიღწევა ანიმაციურ ლოგოს გამოყენებით:

- ზოგიერთ ბრენდს აქვს მსგავსი ლოგოები და უფრო მეტიც, ზოგჯერ ისინი შეიძლება იყოს ერთმანეთის კონკურენტები. ასეთ შემთხვევაში ლოგოსთვის უნიკალურობის მისაცემად დიზაინერს შეუძლია გარკვეული მოძრაობის დამატება;
- ექსპერტების აზრით დინამიკური სურათები და ვიდეო კონტენტები უკეთესად აღიქმება, ვიდრე სტატიკური. ისინი ადვილად დასამახსოვრებელია. განსაკუთრებით – ორიგინალური ანიმაციური ლოგოები, რაც ხელს უწყობს მომხმარებლის ინტერესების გაზრდას;
- ადამიანებს მოსწონთ ახალი იდეები. ეფექტური ანიმაციური ლოგო შეიძლება გახდეს გაოცების და სიხარულის გამომწვევი;
- ანიმაციურ ლოგოს შეუძლია ბიზნესსაქმიანობის უფრო დეტალური ახსნა, ვიდრე სტატიკურ ლოგოს. ის მოკლე ვიდეოა, რომელიც მოგვითხრობს პროდუქციის თუ კომპანიის შესახებ;
- დღესდღეობით სოციალური ქსელები პრდუქციის პოპულარიზაციის ერთ-ერთი მთავარი პლატფორმაა. მათში შესაძლებელია ვიდეოს გაზიარება, რაც ნიშნავს იმას, რომ ადამიანებს შეეძლება ლოგოს ნახვა;
- ანიმაციურ ლოგოს შეუძლია გაზარდოს SEO (Search Engine Optimization).

ანიმაციური ლოგოს მაგალითია <https://justcreative.com/wp-content/uploads/2017/10/Burger-King-Logo-Animation.gif> საიტი (ნახ.4) [5].



ნახ.4

➤ *ვიდეო*. ვებ-საიტის მთავარ გვერდზე ვიდეოს გამოყენება ვებ-დიზაინის ერთ-ერთი საუკეთესო ტენდენციაა. ის შეიძლება იყოს როგორც მცირე ზომის, ასევე სრული ეკრანის ვერსია. ვიდეო ფაილების გამოყენება ყველაზე სასარგებლო იქნება იმ შემთხვევებში, როდესაც საჭიროა:

- 1) ძვირადღირებული პროდუქციის გამოფენა, როგორც ამას აკეთებს კომპანია Apple;
- 2) პრეზენტაციების, ნახატების ან ხელოვნების სხვა სახეების გამოფენების ორგანიზება;
- 3) კონცერტების რეკლამირება;
- 4) სპორტული, პოლიტიკური და სხვა აქტივობების პრეზენტაციები;
- 5) სილამაზის სალონებში კოსმეტიკური პროცედურების ვიდეო ჩანაწერების ჩვენება;
- 6) პროდუქციის სტრუქტურის, ტექნიკის მუშაობის პრინციპების შესწავლა;
- 7) თამაშების რეკლამა ან კინო-ინდუსტრიის სიახლეების პრეზენტაცია და სხვ.

ტექნოლოგიების განვითარებასთან ერთად გამოჩნდა ვიდეოს ახალი შესაძლებლობანი. ესენია: რეალურ დროში შოპინგი, 360°-იანი გადაღება, კონტენტის გაქრობა და სხვა ინოვაციები.

საიტზე ვიდეოს / ვიდეოების გამოყენება საშუალებას მისცემს მომხმარებლებს ნათლად და დეტალურად შეისწავლონ ესა თუ ის პროდუქცია ან მომსახურება. რაც გამოიწვევს მათი მახასიათებლებისა და დადებითი ასპექტების უკეთ გააზრებას [6].

➤ *3D ვიზუალი და VFX*. 3D ვიზუალმა ბოლო ორი წლის განმავლობაში მნიშვნელოვანი პროგრესი განიცადა. სავარაუდოდ, პოპულარობა მომდევნო წლებში კიდევ უფრო გაიზრდება ტექნოლოგიისა და პროგრამული უზრუნველყოფის შესაძლებლობების განვითარების წყალობით. ნებისმიერ დიზაინერს შეუძლია ისწავლოს ლამაზი 3D სცენარების შექმნა მარტივად გამოსაყენებელი პროგრამული უზრუნველყოფით, როგორცაა Figma, Dimensions და სხვ. წარმოუდგენლად ცოცხალი, 3D-რეალური ვიზუალები ახდენს განსაცვიფრებელ ეფექტებსა და შთაბეჭდილებებს.

VFX ანუ Visual Effects აერთიანებს არსებულ კადრებს კომპიუტერში გენერირებულ გამოსახულებებთან და ქმნის რეალისტურ სცენებს. იგი არის კამერით გადაღებულ რეალურ სცენებში ჩასმული ანიმაციები. მრავალი მოსახერხებელი პროგრამული უზრუნველყოფის გამოჩენამ VFX გახადა ხელმისაწვდომი და გავრცელებული ვიდეორგოლების, მოკლემეტრაჟიანი ფილმების შექმნისათვის [7].

### 3. დასკვნა

ვებ-დიზაინის ტენდენციები მუდმივად იცვლება. ამიტომ ვებ-დიზაინერებმა ყოველთვის უნდა აიმაღლონ ცოდნის დონე. მასში ჩამოყალიბებულია ვებ-დიზაინის რამდენიმე თანამედროვე ტენდენცია. განხილული საკითხების გაცნობა სასარგებლო იქნება ვებ-დიზაინით დაიტერესებულ პირთათვის.

#### ლიტერატურა – References – Литература:

1. <https://fireart.studio/blog/4-ways-to-design-a-perfect-split-screen-homepage/>
2. <https://designshack.net/articles/trends/almost-too-many-layers/>
3. <https://webflow.com/blog/scrollytelling-guide>
4. <https://webflow.com/blog/parallax-scrolling>
5. <https://design4users.com/animated-logos-why-where-how-to-use/>
6. <https://merehead.com/ru/blog/10-unique-web-design-trends-follow-in-2022/>
7. <https://elementor.com/blog/3d-web-design/>

## SOME MODERN TENDENCIES OF WEB-DESIGN

Namchevadze Tsatsa, Abuladze Tamar

Kutaisi Tsereteli State University

tsatsanamg@gmail.com, tamar.abuladze@atsu.edu.ge

#### Summary

The article discusses some modern tendencies in web-design: 1).split-screen design; 2). multi-layered design; 3). scrollytelling; 4). animated logo; 5). video; 6) 3D visual and VFX. The article mentions that web-design tendencies are constantly changing. The formation of new tendencies stipulates the conditions, that in order to create profitable projects, web designers must always deepen their creativity.

## НЕКОТОРЫЕ СОВРЕМЕННЫЕ ТЕНДЕНЦИИ ВЕБ-ДИЗАЙНА

Намчевадзе Ц., Абуладзе Т.

Кутаисский Государственный Университет им. Ак.Церетели

tsatsanamg@gmail.com, tamar.abuladze@atsu.edu.ge

#### Резюме

Рассматриваются некоторые современные тенденции веб-дизайна: 1) Дизайн с разделенным экраном; 2) Многослойный дизайн; 3) Scrollytelling; 4) Анимированный логотип; 5) Видео; 6) 3D и VFX. Тенденции веб-дизайна постоянно меняются. Появление новых тенденций означает, что для создания прибыльных проектов веб-дизайнерам всегда необходимо углублять свой творческий потенциал.

# API-ს ტიპები, Web-სერვისები და ინფორმაციის გაცვლის სისტემები. OpenAPI-ს პეციფიკაცია

ლელა პაპავა, სოფიკო გოგოლაძე

საქართველოს ტექნიკური უნივერსიტეტი

lelapava64@gmail.com, gogoladzesopiko08@gtu.ge

## რეზიუმე

განხილულია ვებ სერვისების ზოგადი სტრუქტურა და მათი დანიშნულება თანამედროვე ციფრულ ტექნოლოგიებში. მოყვანილია OpenAPI-ს სპეციფიკაციები და მისი გამოყენება ჩვენი კონკრეტული ამოცანის გადასაჭრელად. კერძოდ, თუ როგორ უნდა განახორციელოს აღნიშნულმა სერვისმა ინფორმაციის გადატანა აპლიკაციებს შორის, ან ერთი კონკრეტული სერვერიდან ინფორმაციის წაკითხვა. ჩვენ მიერ დამუშავებულ მაგალითში, SQL Server-დან ხორციელდება ინფორმაციის წაკითხვა, შუალედური რგოლის View-ს გამოყენებით. ბაზიდან ამოღებულ ინფორმაციას სერვისის საშუალებით იღებს აპლიკაცია და საჭიროებისამებრ ამუშავებს მას.

**საკვანძო სიტყვები:** API. აპლიკაცია. პროგრამირება. ინტერფეისი. Swagger. SmartBear. OpenAPI. სპეციფიკაცია.

## 1. Web - სერვისების ზოგადი აღწერა

Web-სერვისები – ესაა ინფორმაციის გაცვლის სისტემები, სადაც გაცვლის ფორმატი შეზღუდული არაა. ყველაზე ხშირად კი ინფორმაციის გასაცვლელად გამოიყენება JSON და XML, რომლებსაც ეფექტურად იყენებს ინტერნეტი, რათა მოახდინოს აპლიკაციებს შორის პირდაპირი ურთიერთკავშირი. ეს სისტემები შეიძლება შეიცავდეს პროგრამებს, ობიექტებს, შეტყობინებებს ან დოკუმენტებს. სრული ვებ-სერვისი - ესაა ნებისმიერი სერვისი, რომელიც :

- ხელმისაწვდომია ინტერნეტით ან ინტრანეტის საშუალებით;
- იყენებს XML, JSON ...შეტყობინებების გაცვლის სტანდარტიზებულ სისტემას;
- არ არის მიზნული ერთ რომელიმე ოპერაციულ სისტემაზე ან პროგრამირების ენაზე;
- თვითაღწერადია XML, JSON ისინტაქსის საშუალებით;
- შესაძლებელია მისი მოძებნა მარტივი მექანიზმის საშუალებით;

ვებ-სერვისების საბაზისო პლატფორმად გვევლინება XML და HTTP. ყველა სტანდარტული ვებ-სერვისი მუშაობს შემდეგი კომპონენტების გამოყენებით:

- SOAP (ობიექტებზე წვდომის მარტივი პროტოკოლი);
- UDDI (უნივერსალური აღწერა, პოვნა და ინტეგრაცია);
- WSDL (ვებ-სერვისის აღწერის ენა) [1].

## 2. როგორ მუშაობს Web-სერვისები

Web-სერვისები იგება სხვადასხვა აპლიკაციის ინტეგრაციისთვის, ღია სტანდარტების და პროტოკოლების გამოყენებით. პროტოკოლები, რომლებიც იყენებს ვებ-სერვისებს შეიცავს:

- XML – გამოიყენება ტეგირებისთვის, მონაცემების კოდირებისა და დეკოდირებისათვის;
- SOAP – გამოიყენება მონაცემების გადაცემისთვის. SOAP პროტოკოლი შემუშავებული იქნა იმგვარად, რომ სხვადასხვა პროგრამირების ენამ შეძლოს მონაცემების გაცვლა სწრაფად და მარტივად, მინიმალური ძალისხმევის საშუალებით;

- WSDL– ატყობინებს კლიენტ აპლიკაციას ჩართულია თუ არა ვებ სერვისში და თუ არა, როგორ ჩაერთოს;
- UDDI– გამოიყენება იმ სერვისების გადასათვლელად, რომლებიც გამოიყენება ერთ აპლიკაციაში. იგი ასევე საშუალებას აძლევს ვებ-სერვისებს, იყოს ხელმისაწვდომი სხვა სერვისებისთვისაც;
- REST – მართალია ყველა ვებ-სერვისი არ იყენებს REST პროტოკოლს, მაგრამ აპლიკაციები, შექმნილი RESTful API საშუალებით, არის უფრო მსუბუქი, ადვილად სამართავი და მასშტაბირებადი [2].

### 3. Swagger რედაქტორი

API-ს პროექტირებისა და მოდელირებისას მაღალია შეცდომების ალბათობა. Swagger იყო პირველი რედაქტორი, რომელიც შეიქმნა API-ინტერფეისებისათვის, OpenAPI(OAS) და აგრძელებს შემმუშავებლების მოთხოვნების დაკმაყოფილებას. რედაქტორი ამოწმებს დიზაინს რეალური დროის რეჟიმში, ასევე ამოწმებს OAS-ის შესაბამისობას და უზრუნველყოფს ვიზუალურ უკუკავშირს შესასვლელზე [3].

Swagger ეხმარება მომხმარებლებს შექმნან, მოახდინონ დოკუმენტირება, გატესტონ და გამოიყენონ RESTful-ის ვებ-სერვისები. მისი გამოყენება შესაძლებელია API-ს შესაქმნელად მანამ, სანამ დაიწერება რაიმე სახის კოდი.

Swagger იძლევა უფლებას იმგვარად აღიწეროს API, რომ მანქანებმა მოახერხოს მისი წაკითხვა. API ინტერფეისების შესაძლებლობა –აღწეროს თავიანთი საკუთარი სტრუქტურა, არის Swagger-ის საფუძველი. შექმნილი API-ს სტრუქტურის წაკითხვისას შესაძლებელია ავტომატურად შეიქმნას ლამაზი და ინტერაქტიული დოკუმენტაცია API-ს შესახებ. ასევე შესაძლებელია ავტომატურად შეიქმნას კლიენტის ბიბლიოთეკა სხვადასხვა ენებზე და შესწავლილ იქნას ისეთი შესაძლებლობები, როგორცაა, ვთქვათ, ავტომატური ტესტირება. ყველაფერ ამას, Swagger აკეთებს დაბრუნებული YALM ან JSON-ის საშუალებით, რომელიც შეიცავს API-ს ზუსტ აღწერას. ფაილი, რეალურად შეიცავს API-ს რესურსების ჩამონათვალს, რომლებიც შეესაბამება OpenAPI-ს სპეციფიკაციას. იგი მოითხოვს ასეთ ინფორმაციას:

- რომელი ოპერაციები უჭერს მხარს API-ს ;
- რომლებია API-ს პარამეტრები და რას აბრუნებს ;
- ჭირდება თუ არა რაიმე სახის ავტორიზაცია;
- პირობები, საკონტაქტო ინფორმაცია და ლიცენზია API-ს გამოყენების შესახებ.

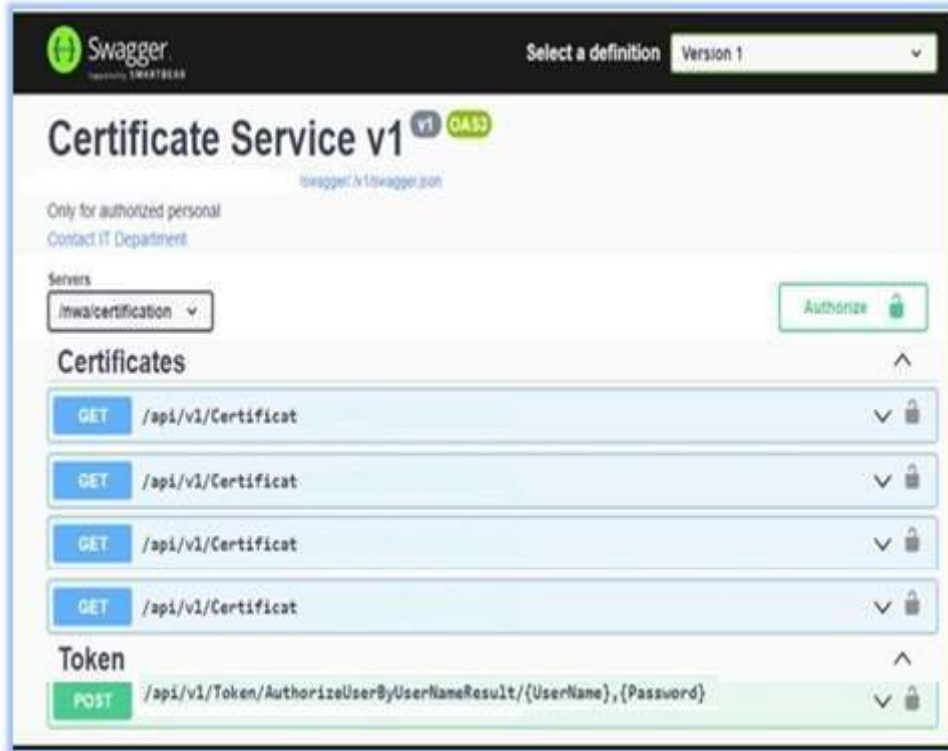
ასევე შესაძლებელია დაიწეროს Swagger-ის სპეციფიკაცია საკუთარი API-სთვის ხელით, ან შეიქმნას იგი ავტომატურად ანოტაციის საშუალებით.

არსებობს რამდენიმე საშუალება, რომლითაც Swagger-ს შეუძლია API-ს შემდგომი განვითარება [4].

### 4. ძირითადი ნაწილი

ჩვენ მიერ შექმნილია კონკრეტული API, რომელიც იყენებს Open API-ს სპეციფიკაციას, ვისარგებლეთ რა Swagger-ის ინტერფეისით (ნახ.1), აღნიშნულ სერვისში ჩავდეთ ვერსიულობა, რაც შესაძლებლობას გვაძლევს მარტივად მოვახდინოთ ლავირება შემდგომში დამატებულ და უკვე არსებულ ვერსიებს შორის.

ამავდროულად მარტივი შესამჩნევია, რომ სერვისი განკუთვნილია მხოლოდ ავტორიზებული მომხმარებლებისთვის.



ნახ.1

ამისათვის ვიყენებთ Bearer Authentication. რომელსაც მეორენაირად Token-ით ავტორიზაციასაც უწოდებენ. Token ჩანართში, HTTP Post-ის საშუალებით, შევავსებთ ავტორიზაციის გასავლელად საჭირო ყველა არსებულ ველს, ჩვენ შემთხვევაში ნათლად ჩანს, რომ საკმარისია User-ის სახელისა და პაროლის მითითება, ამ ოპერაციის შემდეგ დაგენერირდება უნიკალური Token, რომელსაც ჩავსვამთ ავტორიზაციის ღილაკზე დაწკაპუნების შემდეგ ეკრანზე გამოსულ ფანჯარაში და თუ მომხმარებელი დაშვებულია აღნიშნულ სისტემასთან სამუშაოდ, გაივლის ავტორიზაციას.

ჩვენ მაგალითში ჩანს, რომ გვჭირდება მხოლოდ HTTP GET ის გამოყენება, თუმცა არ ვართ შეზღუდულები არც POST, DELETE, PUT ა.შ. გამოყენებაში. აღნიშნული სერვისის გამოყენებით ჩვენ ვახორციელებთ ინფორმაციის გატარებას ერთი აპლიკაციიდან მეორე აპლიკაციისთვის.

## 5. დასკვნა

ყოველივე ზემოაღნიშნულიდან გამომდინარე ვასკვნით, რომ სპეციფიკაცია Open API არის თანამედროვეობის ერთ-ერთი ყველაზე მოქნილი საშუალება სერვისების შესაქმნელად. აქვს საინტერესო ინსტრუმენტები და მოხერხებული, ადვილად აღქმადი ინტერფეისი, რომელიც არანაირ დამატებით ძალისხმევას არ საჭიროებს.

## ლიტერატურა – References – Литература:

1. What are Web Services? Internet resource:[https://www.tutorialspoint.com/web-services/what\\_are\\_web\\_services.htm](https://www.tutorialspoint.com/web-services/what_are_web_services.htm)(2.12.2022)
2. Joshua S. Ponelat, Lukas L.Rosenstock -Designing APIs with Swagger and OpenAPI
3. LewisS.Webservices.Internetresource:<https://www.techtarget.com/searchapparchitecture/definition/Web-services>(2.12.2022)

4. API Design. Faster, Standardized API Design. Internet resource: <https://swagger.io/solutions/api-design/> (20.11.2022)
5. Definition Swagger. TechTarget Contributor. Internet resource: <https://www.techtarget.com/searchapparchitecture/definition/Swagger> (12.11.2022)
6. Swagger. API Development for Everyone. Internet resource: <https://swagger.io/> (13.11.2022)

*(სტატია მიღებულია 12.12.2022)*

## API TYPES, WEB SERVICES AND INFORMATION EXCHANGE SYSTEMS. OPENAPI SPECIFICATION

Papava Lela, Gogoladze Sopiko

Georgian Technical University

lelapapava64@gmail.com, gogoladzesopiko08@gtu.ge;

### Summary

The general structure of web services and their purpose in modern digital technologies are discussed. OpenAPI specifications and how to use it to solve our particular problem are given. In particular, how the said server should transfer information between applications, or read information from one specific server. In the example we worked on, information is read from SQL Server using the intermediate link View. The application receives the information extracted from the database through the service and processes it as needed.

*(Received 12.12.2022)*

## ТИПЫ API, WEB-СЕРВИСЫ И СИСТЕМЫ ОБМЕНА ИНФОРМАЦИЕЙ. СПЕЦИФИКАЦИЯ OPENAPI

Папава Л., Гоголадзе С.

Грузинский Технический Университет

lelapapava64@gmail.com; gogoladzesopiko08@gtu.ge;

### Резюме

Обсуждается общая структура веб-сервисов и их назначение в современных цифровых технологиях. Приведены спецификации OpenAPI и способы его использования для решения нашей конкретной задачи. В частности, как указанный сервер должен передавать информацию между приложениями или считывать информацию с одного конкретного сервера. В примере, над которым мы работали, информация считывается из SQL Server с помощью промежуточного звена View. Приложение получает информацию, извлеченную из базы данных через сервис, и обрабатывает ее по мере необходимости.

*(Поступила в редакцию 12.12.22)*



# საზომ მოწყობილობათა ცდომილებების გათვალისწინებით არასრული კუბური მოდელის სამკომპონენტთან სიმპლექს-გისოსურ გეგმებში ექსპერიმენტის წერტილების გენერირების ალგორითმი

ნინო ბერაია, ლალი ტოკაძე  
საქართველოს ტექნიკური უნივერსიტეტი  
n.beraia@gtu.ge, l.tokadze@gtu.ge

## რეზიუმე

განხილულია სიმპლექს-გისოსური (ნარევთა თვისებების შესასწავლად) სტანდარტული გეგმები, რომელთაც ახასიათებთ ექსპერიმენტატორისთვის მისაღები თვისებები. ეს უზრუნველყოფს ნარევთა თვისებების მარტივ აღწერას პირველი, მეორე, მესამე რიგის პოლინომებით, მაგრამ რეალურ ცხოვრებაში სტანდარტული გეგმის წერტილების შესაბამისი კოორდინატების მიღება შეუძლებელია, ვინაიდან ცალკეული კომპონენტების რაოდენობის განსაზღვრა ხდება საზომი მოწყობილობების საშუალებით, რომელთაც ყოველთვის გააჩნიათ ცდომილებები. ამის გამო სტანდარტული გეგმები მახინჯდება და იცვლება ამ დეფორმირებული გეგმების ძირითადი მახასიათებლებიც. ნაშრომში განხილულია სიმპლექსურ კოორდინატთა სისტემაში საზომ მოწყობილობათა ცდომილებების გათვალისწინებით ექსპერიმენტის წერტილების გენერირების ალგორითმი არასრული კუბური მოდელისთვის.

**საკვანძო სიტყვები:** ექსპერიმენტის დაგეგმვა. სიმპლექსური გეგმა.

## 1. შესავალი

ბუნებაში არსებული და ტექნიკაში გამოყენებული ნივთიერებების მეტი წილი წარმოადგენს სხვადასხვაგვარი კომპონენტების ნარევს. ასეთია მაგალითად სამთო ქანები, მინერალები, სამშენებლო მასალები, ტექნიკური შენადნობები, მინა და სხვ. ასეთი ნარევების (სისტემების) თვისებები პირველ რიგში დამოკიდებულია ძირითადი კომპონენტების ბუნებაზე და მათ ფარდობით შემცველობაზე. ცალკეული კომპონენტების პროპორციების (ფარდობითი შემცველობის) ცვლილებით, ესე იგი სისტემის შემადგენლობის ცვლილებით ნარევს შეიძლება მივანიჭოთ ჩვენთვის სასურველი თვისებები.

ასეთი სისტემების კვლევის მიზანს წარმოადგენს რაოდენობრივი დამოკიდებულებების დადგენა ცალკეული კომპონენტების პროპორციებსა და ნარევის თვისებებს შორის (“შემადგენლობა-თვისებები”), ფუნქციური დამოკიდებულებების ძიება, აგრეთვე იმ წერტილების კოორდინატების დადგენა, რომლებიც უზრუნველყოფენ ამა თუ იმ თვისებების ოპტიმიზაციას (მაქსიმიზაციას ან მინიმიზაციას) დასმული ამოცანის მიხედვით.

## 2. ძირითადი ნაწილი

ფუნქციური დამოკიდებულებების მისაღებად უნდა ჩატარდეს ცდები საკოორდინატო სივრცეში გარკვეული გეგმის მიხედვით. ნარევთა ძირითადი თვისებების გამო, სადაც კომპონენტების  $x_1, x_2, \dots, x_k$  წილობრივი შემცველობების ჯამი ერთის ტოლია ( $x_1 + x_2 + \dots + x_k = 1$ ), საკოორდინატო სივრცე წარმოადგენს სიმპლექსს, გეგმებს კი სიმპლექს-გისოსური გეგმები ეწოდება.

ამ სტატიაში შესწავლილი არასრული კუბური მოდელი სამი კომპონენტის ნარევისთვის აღიწერება შემდეგი ფორმულით:

$$\hat{y} = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3 + \beta_{123} x_1 x_2 x_3 \quad (1)$$

შესაბამისი გეგმა წარმოდგენილია ცხრილში 1.

სამკომპონენტიანი სიმპლექს-გისოსური გეგმა არასრული კუბური

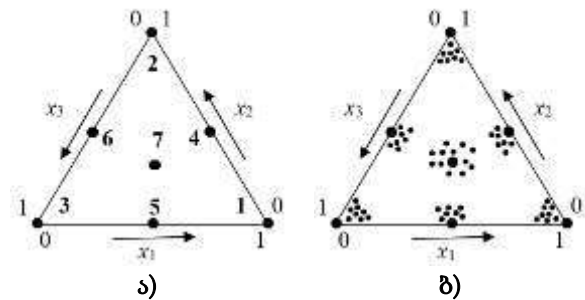
მოდელის შემთხვევაში

ცხრ.1

ცდის №	$x_1$	$x_2$	$x_3$	ამომახილის მნიშვნელობა y
1	1	0	0	$y_1^0$
2	0	1	0	$y_2^0$
3	0	0	1	$y_3^0$
4	1/2	1/2	0	$y_{12}^0$
5	1/2	0	1/2	$y_{13}^0$
6	0	1/2	1/2	$y_{23}^0$
7	1/3	1/3	1/3	$y_{123}^0$

რეგრესიული განტოლების (1) შვიდი კოეფიციენტის შესაფასებლად საკმარისია შვიდი ექსპერიმენტის განხორციელება: სამი ექსპერიმენტი სიმპლექსის წვეროებზე, სამი ექსპერიმენტი თითოეული კიდის შუა წერტილებში და ერთი ექსპერიმენტი სიმპლექსის შიგნით – სიმპლექსის მედიანების კვეთაზე (ნახ. 1, ა).

ეს იდეალური გეგმებია. ასეთი გეგმების წერტილების კოორდინატები გამოითვლება სიმპლექს კოორდინატთა სისტემაში მიღებული მეთოდოლოგიის შესაბამისად [1], მაგრამ ვინაიდან ნარევის ცალკეული კომპონენტების რაოდენობა განისაზღვრება გაზომვის საშუალებებით, რომლებიც ხასიათდება გარკვეული ცდომილებებით, იდეალური გეგმების ზუსტი პროპორციების დადგენა შეუძლებელი ხდება, რაც იწვევს გეგმების დამახინჯებას და რეგრესიული განტოლების დამახინჯებასაც [2].



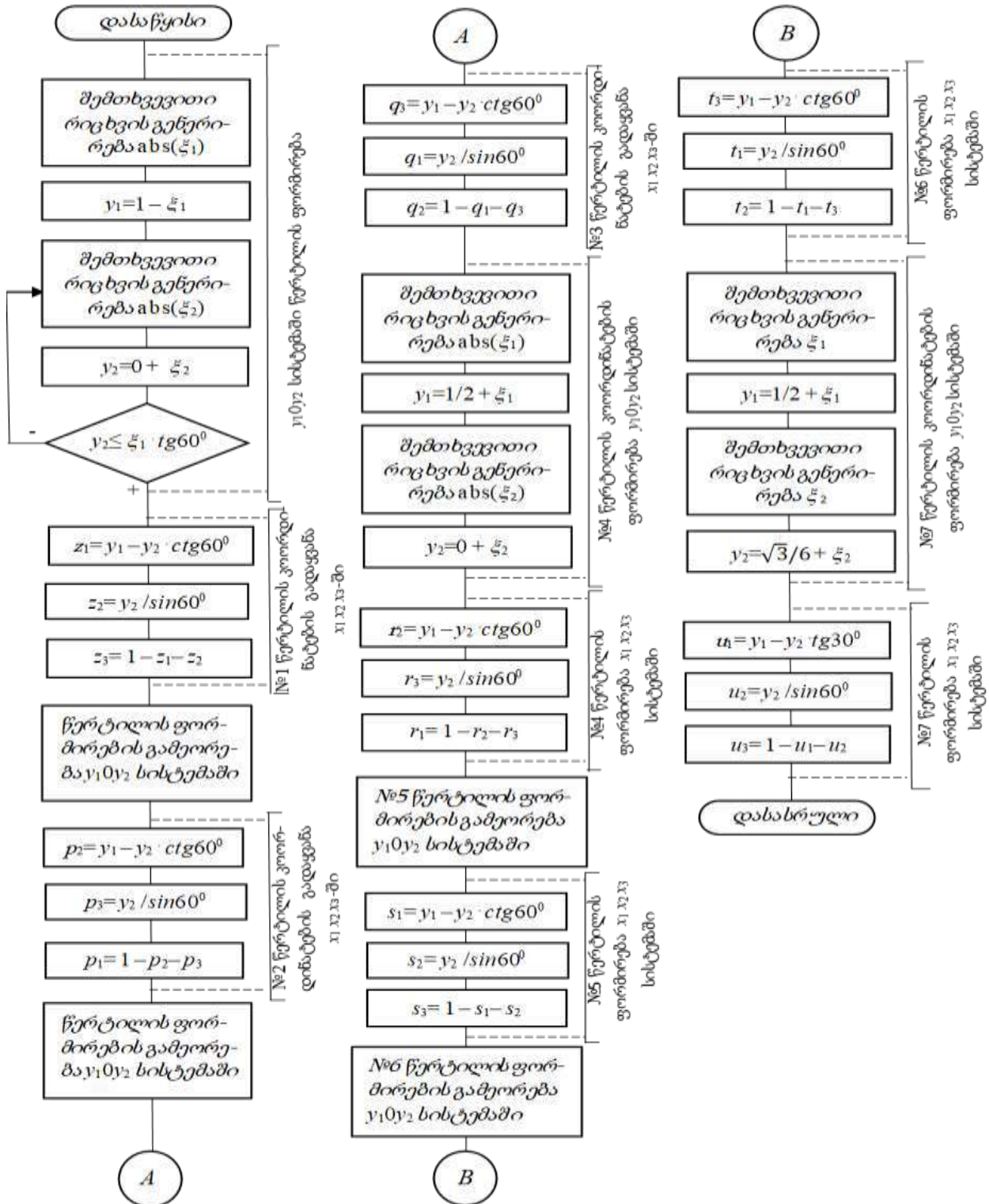
ნახ.1. არასრული მესამე რიგის სამკომპონენტიანი სიმპლექს-გისოსური გეგმები: ა) იდეალური; ბ) დეფორმირებული

გაზომვის ცდომილებათა ზეგავლენის შესწავლისათვის განვახორციელებთ ცდომილებათა იმიტაცია. იდეალური გეგმის წერტილების ირგვლივ დაგეგმვის დასაშვებ არეში ვიღებდით წერტილების ერთობლიობას, რომელთაგანაც შეიძლებოდა დეფორმირებული გეგმის ნებისმიერი კონფიგურაციის შექმნა (ნახ.1, ბ). ასეთნაირად იდეალური გეგმების წერტილების ირგვლივ ვიღებდით იმიტირებული წერტილების ღრუბლებს, იმიტაციას ვახორციელებდით მრავალჯერ. ამისათვის გეგმის იდეალური წერტილების მისაღებად გამოიყენებოდა ნორმალურად განაწილებული შემთხვევითი რიცხვები.

პრობლემა კი მდგომარეობს იმაში, რომ შემთხვევითი რიცხვების ზედდებისას წერტილების კოორდინატებზე დაცული უნდა იყოს ძირითადი პრინციპი – ეს წერტილები არ უნდა გამოვიდნენ სიმპლექსის გარეთ, თანაც სხვადასხვა წერტილის ირგვლივ ღრუბლების მისაღებად გვიწვევს განსხვავებული ალგორითმების გამოყენება. ყველაზე მარტივად ამოცანის გადაწყვეტა

ხდება ორი კომპონენტის შემთხვევაში, გაცილებით უფრო რთულია იმიტირებული წერტილების მიღება სამი კომპონენტის შემთხვევაში [3].

სიმპლექსის კოორდინატა სისტემაში მუშაობის ყველა თავისებურების გათვალისწინებით, ჩვენს მიერ შემოთავაზებულია არასრული მესამე რიგის სამკომპონენტური სიმპლექს-გოსოსური გეგმის შემთხვევითი წერტილების წარმოქმნის ალგორითმი (ნახ. 2).



ნახ.2. ექსპერიმენტული შეცდომის პირობებში სამკომპონენტური სიმპლექს-გოსოსური გეგმის არასრული კუბური მოდელის კოორდინატების გენერირების ალგორითმი

შედეგად, იდეალური სიმპლექს-გისოსური გეგმა, რომელიც წარმოდგენილია ცხრილში 1, მახინჯდება. მისი ვექტორ-სვეტები ექსპერიმენტული შეცდომის პირობებში მიიღებენ შემდეგ ფორმას:

$$x_1 = [z_1 \ p_1 \ q_1 \ r_1 \ s_1 \ t_1 \ u_1]; \quad (2)$$

$$x_2 = [z_2 \ p_2 \ q_2 \ r_2 \ s_2 \ t_2 \ u_2]; \quad (3)$$

$$x_3 = [z_3 \ p_3 \ q_3 \ r_3 \ s_3 \ t_3 \ u_3]; \quad (4)$$

შესაბამისად, შეიცვლება როგორც დაგეგმვის მატრიცა, ასევე კოვარიაციული მატრიცა, რომლებთანაც დაკავშირებულია გეგმების ოპტიმალურობის კრიტერიუმების უდიდესი რაოდენობა, რაც აუცილებლად გასათვალისწინებელია რეალურ პირობებში გეგმების განხორციელებისას.

### 3. დასკვნა

ასეთი მიდგომა ექსპერიმენტატორს მისცემს საშუალებას შეაფასოს ნარევის მახასიათებლების გაუარესების ხარისხი, რაც დამოკიდებულია არჩეულ საზომ ხელსაწყოებში/საშუალებებში არსებულ ცდომილებებზე. ალგორითმის შესაბამისად მიღებულია საზომი საშუალებების დამახასიათებელი ცდომილებები სხვადასხვა დიაპაზონისთვის გეგმების მრავალჯერადი (100-ჯერ და მეტი) გენერირებით. შედეგად, ექსპერიმენტატორს შეუძლია, რეგრესიის განტოლების კოეფიციენტების შეფასების საჭირო სიზუსტეზე დაყრდნობით, სწორად შეარჩიოს საზომი საშუალებების ცდომილება, რომელიც გამოიყენება ნარევის კომპონენტების გასაზომად.

### ლიტერატურა – References – Литература:

1. Zedginidze I.G. (1976). Planning an Experiment for Researching Mmany-component Systems. - M.: "Nauka", (in Russian)
2. Beliya N. (2005). The Research of Second Order Plans in the Presence of Experimental Errors. Tbilisi: Cargo. Technical University. 2005, 338 p. (in Russian)
3. Beraia N., Tokadze L. Algorithm for generating experimental points for a quadratic model of three-component simplex-lattice plans under experimental errors. Deutsche internationale Zeitschrift für zeitgenössische Wissenschaft / German International Journal of Modern Science, №27 2022, pp. 65-69.

*(სტატია მიღებულია 18.12.2022)*

## ALGORITHM FOR GENERATING EXPERIMENTAL POINTS FOR AN INCOMPLETE CUBIC MODEL OF THREE-COMPONENT SIMPLEX-LATTICE DESIGNS, SUBJECT TO ERRORS OF MEASUREMENT

Beraia Nino, Tokadze Lali

Georgian Technical University

n.beraia@gtu.ge, l.tokadze@gtu.ge

### Summary

Known for the perfect design on simplex, providing a relatively simple definition of regression coefficients varying degrees and with certain properties. But just to sustain the proportion of each component due to errors of the means of measurement is not possible. Ideal design are deformed, causing deterioration of characteristics. Therefore, we study the influence of errors of the means of measurement. To simulate a cloud of points around each point of the perfect design to develop appropriate algorithms and make cost-effective programs in the environment of MATLAB. For this purpose, an algorithm was developed for generation of coordinates of a three-component simplex-lattice design of the incomplete third order under experimental errors. By changing the values of errors possible to study the main characteristics of the designs and choose the most suitable for the experimenter.

**Keywords:** Design of experiment, designs on a simplex.

*(Received 18.12.2022)*

## АЛГОРИТМ ГЕНЕРАЦИИ ЭКСПЕРИМЕНТАЛЬНЫХ ТОЧЕК В ТРЕХКОМПОНЕНТНЫХ СИМПЛЕКСНО-РЕШЕТЧАТЫХ ПЛАНАХ НЕПОЛНОЙ КУБИЧЕСКОЙ МОДЕЛИ С УЧЕТОМ ПОГРЕШНОСТЕЙ ИЗМЕРИТЕЛЬНЫХ УСТРОЙСТВ

Берая Н.О., Токадзе Л.Ш.

Грузинский Технический Университет

n.beraia@gtu.ge, l.tokadze@gtu.ge

### Резюме

Известны идеальные планы на симплексах, обеспечивающие относительно простое определение коэффициентов регрессионных зависимостей различной степени и обладающие определенными свойствами. Но точно выдержать пропорции каждого из компонентов из-за погрешностей применяемых средств измерения не удастся. Идеальные планы деформируются, что вызывает ухудшение характеристик. Поэтому исследуется влияние погрешностей применяемых средств измерения. Для имитации облаков точек вокруг каждой точки идеального плана необходимо разработать соответствующие алгоритмы и составить экономичные программы в среде MATLAB. С этой целью в настоящей работе разработан алгоритм для генерации координат трехкомпонентного симплекс-решетчатого плана неполного третьего порядка в условиях ошибок эксперимента. Изменяя значения погрешностей возможно изучить основные характеристики планов и выбрать наиболее подходящие для экспериментатора.

**Ключевые слова:** Планирование эксперимента, планы на симплексе.

*(Поступила в редакцию 18.12.22)*

# ჩატბოტების ტექნოლოგიების გამოყენება ბიზნეს გარემოში

ნინო გრიგალაშვილი  
საქართველოს ტექნიკური უნივერსიტეტი  
grigalashvili.nino@gtu.ge

## რეზიუმე

განხილულია ჩატბოტების, როგორც ინოვაციური კომუნიკაციის ფორმის, გამოყენება ბიზნესის სხვადასხვა სფეროში. აღწერილია ჩატბოტების დანიშნულება და მისი უპირატესობები. მოყვანილია ჩატბოტების მაგალითები სხვადასხვა ინდუსტრიაში, როგორცაა მაგალითად, მომსახურების სფერო (customer service), E-commerce, განათლება და სხვ. წარმოდგენილია ChatGPT, როგორც ბოლოდროინდელი ინოვაციური ჩატბოტი და მოყვანილია მისი დადებითი და უარყოფითი მხარეები.

**საკვანძო სიტყვები:** ჩატბოტი, ChatGPT, AI, NLP

## 1. შესავალი:

### ჩატბოტის არსი და მისი გამოყენების სფეროები

ჩატბოტი ინოვაციური კომუნიკაციის ფორმაა, რომლის მიზანაც მომხმარებლისთვის ადამიანის საუბრის სიმულაციის შეთავაზებაა. არსებობს ჩატბოტები, რომლებიც ხელოვნურ ინტელექტზეა დაფუძნებული და არის ისეთებიც, რომლებიც ნაკლებად „ჭკვიანებია“, თუმცა მაინც აკეთებს გარკვეულ დონეზე მოთხოვნის ანალიზს და შემდეგ მზა პასუხების ბაზაში ტექტში ნაპოვნი ქივორდების საფუძველზე, ეძებს შესაბამის შესატყვის პასუხს [1].

ჩატბოტებმა „ფეხი“ მყარად ბიზნეს სექტორში მოიკიდეს, რისი გამომწვევი ერთ-ერთი მთავარი მიზეზი სწრაფი და ეფექტური კომუნიკაციაა. მომხმარებლებს, ძირითადად მსგავსი და მარტივი მოთხოვნები (კითხვები) აინტერესებთ: „როდის იხსნება ფილიალი“, „რა ღირს სერვისი“, „სამუშაო საათები“ და ა.შ. ამ შეკითხვებზე პასუხის მისაღებად ქოლ-ცენტრის ოპერატორებს მარტივად ანაცვლებს ჩატბოტები, რომლებიც მარტივად და მყისიერად სცემენ მომხმარებელს პასუხს.

მიუხედავად იმისა, რომ ჩატბოტები მუდმივად ვითარდება, მათი პასუხი შეიძლება არ იყოს ყოველთვის უტყუარი, მათი განვითარება ეხმარება როგორც მომხმარებლებს, ისე ბიზნესებს თავიანთი ინტერესების დასაკმაყოფილებლად.

ჩატბოტების დადებით მხარეებში შედის ისიც რომ, ჩატბოტს ქოლ-ცენტრ ოპერატორისგან განსხვავებით შეუძლია 24/7-ზე მყისიერი პასუხი, მულტიტასკინგი - რაც უფრო ბევრ მომხმარებელს ეკონტაქტება მით უფრო „ჭკვიანდება“. ასევე, უწყობს ხელს ბიზნესებს რომ შეამცირონ ხარჯები, ჩატბოტში ერთჯერადი ინვესტიცია ამცირებს ყოველთვიურ ხარჯებს გრძელი ვადებში [1].

ChatGPT არის GPT-ს (Generative Pretrained Transformer) ენის მოდელის ნაირსახეობა, რომელიც შეიქმნა კომპანია OpenAI-ს მიერ, 2022 წლის 30 ნოემბერს [2]. ChatGPT ჩატბოტისგან განსხვავებით არ არის წინასწარ დაპროგრამებული დეტერმინისტული პასუხების სიმრავლით, არამედ აგენერირებს ახალ პასუხებს შემომავალ მონაცემებზე დაყრდნობით. როგორც სხვა ენის

მოდელები, იგი დატრენინგებულია დიდი ზომის, მრავალფეროვან მონაცემებზე, როგორცაა სტატიები, წიგნები, ვებ-გვერდები, სოციალური მედიის სტატუსები, ენის სპეციფიკისა და სტრუქტურის დასასწავლად.

## 2. ძირითადი ნაწილი

ChatGPT -ის მოდელები დატრენინგებულია unsupervised learning-ის გამოყენებით, რაც გულისხმობს დიდი ზომის მონაცემების შეყვანას მოდელში ისე, რომ მოდელმა თვითონ იპოვოს კავშირები და დააგენერიროს პასუხი, სწორი პასუხის წინასწარი ცოდნის გარეშე [2].

პასუხების დასაგენერირებლად, ChatGPT იყენებს მულტი შრის მქონე ტრანსფორმერის ქსელს, რომელიც ღრმა სწავლების (deep learning) არქიტექტურის ერთ-ერთი ტიპია. ერთ-ერთი მნიშვნელოვანი ChatGPT-ის თვისება არის პასუხების გენერირება წარსულ ინტერაქციაზე დაყრდნობით. რაც ნიშნავს იმას, რომ მოდელს ესმის საუბარი რა მიმართულებით მიედინება და შეუძლია ყოველგვარი კონკრეტიკისა და კონტექსტის გარეშე, წინა კითხვებზე დაყრდნობით მიხედეს კონტექსტს და სწორ თემაზე გასცეს პასუხი, მიუხედავად იმისა, რომ ეს თემა ამ კითხვაში არ იყო ნაგულისხმევი.

ChatGPT-ს ასევე შეუძლია სხვადასხვანაირი NLP ტასკების (დავალებების) შესრულება, როგორცაა ერთი ენიდან მეორეზე გადათარგმნა, ტექსტის შეჯამება და ანალიზი. ეს ყოველივე გვაძლევს საშუალებას chatGPT სხვადასხვა ტიპის აპლიკაციებში გამოვიყენოთ.

ერთ-ერთი შეზღუდვა რაც აქვს chatGPT-ს არის ის, რომ მისი მასშტაბურობის და კომპლექსურობის გამო, იგი ძალიან მძიმე ხელსაწყოა. ამის შედეგად მისი გამოყენება რეალურ აპლიკაციებში, როგორცაა ჩატბოტი, შეიძლება გაჭირდეს, რადგან ამ უკანასკნელში სწრაფი პასუხებია მნიშვნელოვანი.

ასევე ერთ-ერთი ნაკლი, რაც აქვს chatGPT-ს ისაა, რომ მისი დაგენერირების მოდელიდან გამომდინარე, მას შეუძლია არაზუსტი პასუხი დაგვიბრუნოს კონკრეტულ კითხვებზე. ზოგიერთ შემთხვევაში, დაგენერირებული პასხები შეიძლება იყოს არარელევანტური ან/და აბსურდული, რაც არასასურველია.

დამატებით, chatGPT, როგორც ყველა NLP მოდელი, ლიმიტირებულია იმ მონაცემების რაოდენობითა და ხარისხით, რომელზეცაა დატრენინგებული. თუ მოდელში არ შევუშვით საკმარისად მრავალფეროვანი და განსხვავებული მონაცემები, მოდელი ზუსტ და სწორ პასუხებს ვეღარ დაგვიბრუნებს იმ შემთხვევაში, რომლებიც დატრენინგებული მონაცემების გარეთ ხვდება.

## 3. დასკვნა

ამგვარად, chatGPT მძლავრი NLP მოდელია, რომელიც იყენებს ნეირონული ქსელის არქიტექტურას ადამიანის მსგავსი პასუხების დასაგენერირებლად. იგი იმახსოვრებს წარსული ინტერაქციების თემას და შეუძლია შესაბამისად ადეკვატური, თემაში ჩამჯდარი პასუხების დაგენერირება. ეს მახასიათებელი სძენს მას განსაკუთრებულობას და ფასს. ზოგადად, ჩატბოტების გამოყენება სხვადასხვა ინდუსტრიაში მეტად პოპულარული ხდება და სხვადასხვა ინდუსტრიაში შესამჩნევია მათი დანერგვა, მიუხედავად იმისა რომ ეს სფერო ვითარდება, და შეცდომები შესაძლებელია მოხდეს, მათი ხშირი გამოყენება მათი განვითარებისთვის აუცილებელია.

**ლიტერატურა – References – Литература:**

1. Creative Chatbot ideas to boost your online business in 2023? Internet resource: <https://www.tidio.com/blog/best-chatbot-ideas-to-increase-online-sales/> (10.12.2022)
2. Introducing ChtaGPT <https://openai.com/blog/chatgpt> (10.12.2022)

*(სტატია მიღებულია 20.12.2022)*

## **CHATBOTS USAGE IN BUSINESS SECTOR**

Grigalashvili Nino

Georgian Technical University

[grigalashvili.nino@gtu.ge](mailto:grigalashvili.nino@gtu.ge)

### **Summary**

A chatbot is a computer program designed to simulate conversation between users. Chatbots which are often powered by Artificial Intelligence (AI) and natural language processing (NLP) technologies, enable them to understand and respond to inputs in a conversational manner. They are developed for providing information, giving recommendations and other. Chatbots are used in various fields including customer service, marketing and personal assistance, education, healthcare. Its usage has its positive and negative sides, but regardless everything chatbots help us to receive efficient and quick responses. ChatGPT which is AI powered chatbot is a trained language model which has vast knowledge base and generates human like responses making it an effective tool for customer service, personal assistance and others. Although every chatbot does make mistakes, it does become more advanced with usage.

*(Received 20.12.2022)*

## **ИСПОЛЬЗОВАНИЕ ЧАТ-БОТОВ В БИЗНЕС-СЕКТОРЕ**

Григалашвили Н.

Грузинский Технический Университет

[grigalashvili.nino@gtu.ge](mailto:grigalashvili.nino@gtu.ge)

### **Резюме**

Рассмотрено применение чатботов в различных сферах бизнеса, как новой инновационной формы коммуникации. Описано назначение чатботов и их преимущества. Приводятся примеры чатботов в различных областях, таких как: сфера обслуживания (customer service), E-commerce, образование и др. Рассмотрен ChatGPT, как инновационный чатбот последнего времени и приведены его положительные и отрицательные стороны.

*(Поступила в редакцию 20.12.22)*



# ინფორმაციული სისტემების სწრაფქმედების ამაღლება Redis-Cache მეთოდის გამოყენებით .Net Core პლატფორმაზე

კონსტანტინე თაგაური

საქართველოს ტექნიკური უნივერსიტეტი  
kokotagauri@gmail.com

## რეზიუმე

განხილულია ინფორმაციული სისტემების სრულყოფის (ოპტიმიზაციის) საკითხები მეხსიერების ქეშირების გამოყენების საფუძველზე. იგი დღეს ძალზე ეფექტური და პოპულარულია პროგრამული უზრუნველყოფის ინდუსტრიაში. უმჯობესდება აპლიკაციის მუშაობის (სწრაფქმედები) და მასშტაბურობის მაჩვენებლები. სტატიაში წარმოდგენილია Redis მონაცემთა ბაზის და ქეშირების (Redis Cache) მუშაობის სცენარის კონკრეტული მაგალითი .NET Core პლატფორმაზე. ნაჩვენებია თუ როგორ ხდება მონაცემთა მოპოვების პროცესი და როგორ მცირდება მოთხოვნების ნაკადისთვის მონაცემებზე წვდომის საჭირო დრო.

**საკვანძო სიტყვები:** ინფორმაციული სისტემა. მონაცემთა საცავი. კომპიუტერის მეხსიერება. ქეშირება. მონაცემთა ბაზა. Redis. აპლიკაცია. .NET Core პლატფორმა.

## 1. შესავალი

ორგანიზაციული მართვის (მენეჯმენტის) სისტემებში, რომლებიც რეალიზებულია კლიენტ-სერვერული ან სერვის-ორიენტირებული არქიტექტურით, ბოლო ათწლეულში მნიშვნელოვნად მოიმატა პროგრამული Agile-მეთოდოლოგიისა და NoSQL ტიპის მონაცემთა ბაზების გამოყენების ტენდენციამ, რადგან Big Data და ეკოსისტემების შექმნა დღევანდელი ინოვაციური პერსპექტივაა [1-4]. განსაკუთრებით საყურადღებოა აქ სისტემების სუფთა არქიტექტურის (Clean Architecture) პროექტირების კონცეფცია DDD (Domain Driven Design) მეთოდის ბაზაზე, საბანკო და მიკროსაფინანსო ორგანიზაციებში [5,6].

ასეთ პროგრამული აპლიკაციების ხარისხის სრულყოფა მოითხოვს თავის მხრივ მომხმარებლებისთვის მოთხოვნების დამუშავებისა და პასუხების მიწოდების დროის შემცირებას. საინფორმაციო სისტემების შემდგომი სრულყოფა ასეთი ოპტიმიზაციის ამოცანის გადაწყვეტით ახალი ციფრული ტექნოლოგიების ბაზაზე, მეტად მნიშვნელოვანია .

წინამდებარე სტატიაში ჩვენ განვიხილავთ .NET Core პლატფორმაზე აგებული პროგრამული აპლიკაციის შესაძლებლობებს Redis მონაცემთა ბაზის გამოყენებით [7]. კერძოდ კი, შევხებით “Redis-Cache” ქეშ-მექანიზმით ასეთი ოპტიმიზაციის საკითხის გადაწყვეტას [8].

ქეში მეხსიერების საცავია, რომელიც გამოიყენება ხშირი წვდომის მონაცემების დროებით შესანახად, ის მკვეთრად სრულყოფს წვდომის პროცესს და ამცირებს მონაცემთა ბაზასთან ზედმეტ მიმართვებს, ხშირად გამოყენებად მონაცემთა დროებით ბუფერში მოთავსებით (ნახ.1) [8].



ნახ.1. ქეშირების გამოყენება

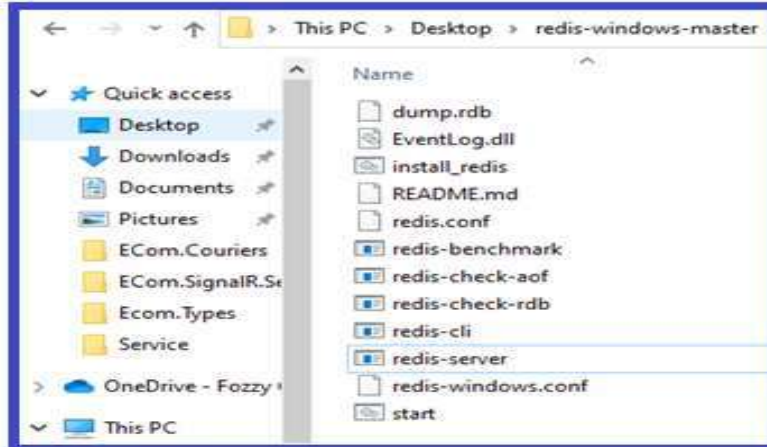
არსებობს ორი სცენარი, ერთი ქეშის გამოყენების გარეშე და მეორე ქეშით. ასე რომ, როდესაც ჩვენ არ ვიყენებთ ქეშს, ამ შემთხვევაში, დავუშვათ, რომ მომხმარებლებს სურთ მონაცემები, ისინი ყოველ ჯერზე მოხვდებიან მონაცემთა ბაზაში და ეს გაზრდის დროის სირთულეს და შეამცირებს შესრულებას იმ შემთხვევაში, თუ არსებობს გარკვეული სტატიკური მონაცემები, რაც მომხმარებლებს სურთ და ეს იგივეა. ყველა მომხმარებელი. იმ შემთხვევაში, როდესაც ჩვენ არ ვიყენებთ ქეშს, მაშინ თითოეული მიმართავს არასაჭირო მონაცემთა ბაზას მონაცემების მისაღებად. მეორე მხარეს, როგორც ხედავთ, ჩვენ ვიყენებთ ქეშს, და ამ შემთხვევაში, თუ ყველა მომხმარებლისთვის არის იგივე სტატიკური და იგივე მონაცემები, მაშინ მხოლოდ პირველი მომხმარებელი მოხვდება მონაცემთა ბაზაში და მიიღებს მონაცემებს და შეინახავს მას ქეშ მეხსიერებაში და შემდეგ სხვა ორმა მომხმარებელმა გამოიყენა ეს ქეშიდან მონაცემთა ბაზის ზედმეტად მიმართვის გარეშე.

## 2. ძირითადი ნაწილი

ძირითადად, არსებობს ქეშირების ორი ტიპი, რომლის მხარდაჭერაც აქვს .NET Core-ს: 1) In-Memory Caching და 2) Distributed Caching.

როდესაც ჩვენ ვიყენებთ In-Memory Cache-ს, ამ შემთხვევაში მონაცემები ინახება აპლიკაციის სერვერის მეხსიერებაში და საჭიროების შემთხვევაში, ვიღებთ მისგან ამ მონაცემებს შემდგომი გამყენების მიზნით. განაწილებულ ქეშირებაში არის მრავალი მესამე მხარის მექანიზმი, როგორცაა, მაგალითად, მონაცემთა არარელაციური ბაზა Redis და სხვ. [1,7]. აქ ჩვენ დეტალურად განვიხილავთ Redis Cache-ს და მისი მუშაობის სცენარს .NET Core პლატფორმაზე.

- **ბიჯი 1.** დავაყენოთ Redis ბაზა სისტემაში (სერვერზე) (ნახ.2).



ნახ.2. Redis-ბაზის ინსტალაციის ფრაგმენტი

• **ბიჯი 2.** დავაინსტალიროთ შემდეგი NuGet პაკეტები, რომლებიც საჭიროა ეტაპობრივად ჩვენს აპლიკაციაში:

```
<PackageReference Include="StackExchange.Redis" Version="2.2.50" />
<PackageReference Include="Microsoft.EntityFrameworkCore.Design" Version="3.1.10">
<PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="3.1.10" />
<PackageReference Include="Microsoft.EntityFrameworkCore.Tools" Version="3.1.10">
```

• **ბიჯი 3.** შექმნილი გვაქვს კლასი TransportSendingConfiguration, რომლის დანიშნულებაც არის განსაზღვროს, რომელი ლოგისტიკური სერვისისთვის რომელ State-ში მოხდეს სატრანსპორტო სისტემაში შეკვეთის გადაგზავნა (ნახ.3).

```

9 references | Константин Тараури, 125 days ago | 1 author, 1 change
public class TransportSendingConfigurationDTO
{
    1 reference | Константин Тараури, 125 days ago | 1 author, 1 change
    public LogisticService LogisticService { get; set; }

    3 references | Константин Тараури, 125 days ago | 1 author, 1 change
    public BasketStateRegistry? SendingBasketState { get; set; }
}
    
```

ნახ.3. TransportSendingConfiguration - კლასის შექმნა

- ბიჯი 4. დავამატოთ SQL Server connection string და Redis URL appsetting.json-ში (ნახ.4).

```

"RedisCacheSettings": {
  "UseRedis": true,
  "Host": "localhost"
},
"DistributedLockOptions": {
  "RedisConnectionString": "localhost"
}
    
```

ნახ.4. connection string -ს დამატება

- ბიჯი 5. შევქმნათ მეთოდი Redis-თან კავშირისთვის (მოცემულია არასრული ფორმით, გამომდინარე მისი სიდიდიდან) (ნახ.5).

```

private IDatabase Connection
{
    get
    {
        var useRedis = Environment.GetEnvironmentVariable("USE_REDIS")?.Equals("true") ?? false;

        if (!useRedis)
        {
            _logger.LogError("{msg}", $"{_missingConnectionEnvVars} {useRedis}");
            return null;
        }

        var db = _lazyConnection.Value.GetDatabase();
        if (db.Multiplexer.IsConnected)
        {
            return db;
        }

        _logger.LogDebug("{msg}", $"{_logTag}: Multiplexer not connected");
        return null;
    }
}
    
```

ნახ.5. ConnectionHelper კლასის შექმნა Redis-ისთვის

- **ბიჯი 6.** დავამატოთ ქემის სერვისი კონკრეტულ (TransportSendingConfiguration) ობიექტთან სამუშაოდ, რომელიც იმუშავებს შემდეგი ლოგიკით: თუ არსებობს რაიმე კონკრეტული გასაღებით ქემში მონაცემი, წამოიღებს არსებულ ჩანაწერს, თუ არა - შექმნის ქემში ჩანაწერს მოცემულ გასაღებზე და დაგვინრუნებს შექმნილ ობიექტს. მოცემული სერვისის მუშაობს მთლიანად ასინქრონულად. (ნახ.6).

```

public class TransportSendingConfigurationCacheService : ITransportSendingConfigurationCacheService
{
    private readonly IRedisCacheService _redisCacheService;

    0 references | Konstantine Tarapli, 125 days ago | 1 author, 1 change
    public TransportSendingConfigurationCacheService(IRedisCacheService redisCacheService)
    {
        _redisCacheService = redisCacheService;
    }

    1 reference | Konstantine Tarapli, 125 days ago | 1 author, 1 change
    public async Task<bool> ExistsTransportSendingConfigurationCacheKey(LogisticService logisticService)
    {
        var key = GetCacheKey(logisticService);
        return await _redisCacheService.KeyExists(key);
    }

    2 references | Konstantine Tarapli, 125 days ago | 1 author, 1 change
    public async Task<TransportSendingConfigurationCacheItem> GetTransportSendingState(LogisticService logisticService)
    {
        var key = GetCacheKey(logisticService);
        var cachedValue = await _redisCacheService.GetCachedValue<TransportSendingConfigurationCacheItem>(key);

        return cachedValue;
    }

    2 references | Konstantine Tarapli, 125 days ago | 1 author, 1 change
    public async Task SetTransportSendingStateKeyValue(LogisticService logisticService,
        TransportSendingConfigurationCacheItem value, TimeSpan? expiry = null)
    {
        var key = GetCacheKey(logisticService);
        await _redisCacheService.SetKeyValue(key, JsonConvert.SerializeObject(value), expiry);
    }

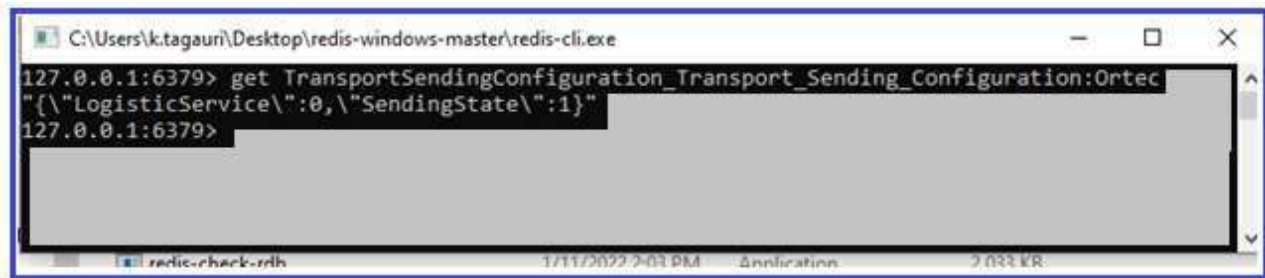
    3 references | Konstantine Tarapli, 125 days ago | 1 author, 1 change
    private string GetCacheKey(LogisticService logisticService)
    {
        return $"{CacheKeys.TransportSendingConfiguration}_Transport_Sending_Configuration:{logisticService}";
    }
}

```

ნახ.6. ქემირების სერვისის დამატება

**შენიშვნა:** რამდენიმე ტექნიკური ბიჯი გამოტოვებულია სტატიის ნაკლებად გადატვირთვის მიზნით, ისეთები, როგორებიცაა სერვისის შექმნა, რომელიც მოემსახურება ქემირების ლოგიკებს, ამ სერვისის ჩაშენება (Inject) Core-ის Pipeline-ში, კონტროლერის დაწერა პროდუქტებისთვის, ბაზისა და ბაზის კონტექსტის შექმნა და ა.შ.

პროგრამის გაშვების შემდეგ, შეგვიძლია შევამოწმოთ Redis-ის cli. შედეგად, შეგვიძლია ვივლით, რომ ჩვენმა .Net Core-ის პროგრამამ წარმატებით ჩაწერა Redis-ის ქეში ობიექტი და შესაბამისი გასაღებით მიკითხვისას დროის ძალიან მცირე მონაკვეთში დაგვიბრუნა ეს ობიექტი (ნახ.7).



```

C:\Users\k.tagauri\Desktop\redis-windows-master\redis-cli.exe
127.0.0.1:6379> get TransportSendingConfiguration_Transport_Sending_Configuration:Ortec
"{\"LogisticService\":0,\"SendingState\":1}"
127.0.0.1:6379>
  
```

ნახ.7. Redis-ის ქეში ობიექტის ჩაწერის შედეგი

### 3. დასკვნა

ქეში არსებული მონაცემები ჩვეულებრივ ინახება სწრაფი წვდომის აპარატურაში, როგორცაა RAM (Random-access memory) და ასევე შეიძლება გამოყენებულ იქნას პროგრამულ კომპონენტთან კორელაციაში. ქეშის ძირითადი დანიშნულებაა გაზარდოს მონაცემთა მოპოვების შესრულება, შეამციროს მათზე წვდომის საჭირო დრო. ქეშირების არარსებობის შემთხვევაში, სისტემა შეიძლება იმდენად გადაიტვირთოს, რომ მისი გამართულობა დადგეს საფრთხის ქვეშ. ამგავრად, დიდ სისტემებთან მუშაობისას, რეკომენდებულია ქეშირების გამოყენება.

### ლიტერატურა – References

1. Chogovadze G., Prangishvili A., Surguladze G. (2017). Hybrid Software Technologies and Data Engineering for Management Information Systems. Monograph, ISBN 978-9941-20-790-7. GTU, „Techn.Univ.“, Tb., -1001 p., (in Georgian)
2. Beck K. et al. (2001). Manifesto for Agile Software Development. Internet resource; <https://agilemanifesto.org/> (11.05.22)
3. Surguladze G, (2019). Computer Programming Methods and Methodologies (SP, OOP, VP, Agile, UML). ISBN 978-9941-1900-1. GTU, “IT-Consulting science center”, Tb., 200 p., (in Georgian)
4. Surguladze G., Mgebrishvili G. (2019). The Management of the Financial Organization by the use of Microservice Architecture. Transacti. of Georgian Technical University. Automated Control Systems - No 2(29), pp.117-123 (in Georgian)
5. Tagauri K. (2022). Solution of infrastructural (architectural) issues of projects and optimization of processes. GTU. II student scientific-practical conference “*Digital Transformation - Challenges and Progress*”, dedicated to the 100th anniversary of the Georgia Technical University and the 65th anniversary of the IMS faculty. 20-21.05. “Technical University”. Tb., (in Georgian)
6. Kleppmann M. (2017). Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. First Ed., Pub. by O'Reilly Media. CA, USA

7. Carlson J. (2013). Redis in Action. ©Publications by Manning Shelter Island. ISBN 9781935182054. Printed in USA

8. Jaydeep Patil. (20221). Implementation Of The Redis Cache In The .NET Core API. <https://www.c-sharpcorner.com/article/implementation-of-the-redis-cache-in-the-net-core-api/>

*(სტატია მიღებულია 21.12.2022)*

## RAISING THE INFORMATION SYSTEMS PERFORMANCE USING THE REDIS-CACHE METHOD ON THE .NET CORE PLATFORM

Konstantine Tagauri  
Georgian Technical University  
kokotagauri@gmail.com

### Summary

Issues of improvement (optimization) of information systems are discussed on the basis of the use of memory caching. It is very effective and popular in the software industry today. Improves application performance and scaling. The article presents a specific example of the Redis Database and Cache (Redis Cache) work scenario on the .net Core platform. It shows how the data extraction process takes place and how the time required to access the data for the flow of requirements is reduced.

*(Received 21.12.2022)*

## ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ КЕШИРОВАНИЯ «REDIS-CACHE» НА ПЛАТФОРМЕ .NET CORE

Тагаური К.  
Грузинский технический университет  
kokotagauri@gmail.com

### Резюме

Рассматриваются вопросы улучшения (оптимизации) информационных систем управления на основе использования кеширования памяти. Сегодня он очень эффективен и популярен в индустрии программного обеспечения. Улучшает производительность приложений и масштаб. В статье представлен конкретный пример рабочего сценария базы данных Redis и кэша (Redis Cache) на платформе .NET Core. Он показывает, как происходит процесс извлечения данных и как сокращается время, необходимое для доступа к данным для потока требований.

*(Поступила в редакцию 21.12.22)*

იბეჭდება ავტორთა მიერ წარმოდგენილი სახით

კომპიუტერული უზრუნველყოფა: გ. სურგულაძე, ე. თურქია, გ. ნარეშელაშვილი,  
მ. ცერცვაძე, გ. მაისურაძე, გ. დალაქიშვილი

გადაეცა წარმოებას 26.12.2022 წ. ქალაქის ზომა 60X84 1/8. პირობითი ნაბეჭდი  
თაბახი 7.15, სააღრიცხვო-საგამომცემლო თაბახი 7. ტირაჟი 50 ეგზ.  
იბეჭდება ავტორთა ხარჯით