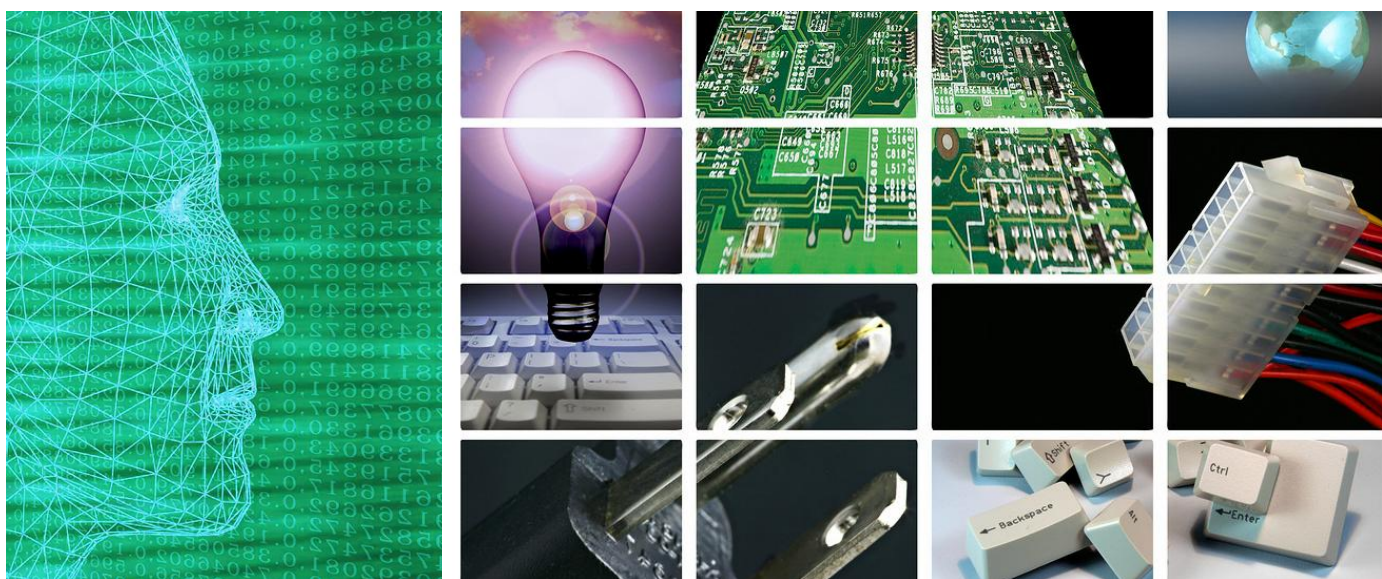


КОМПЮТЪРНИ НАУКИ И ТЕХНОЛОГИИ



Bulgaria Communications Chapter

Faculty of Computing & Automation

COMPUTER SCIENCE AND TECHNOLOGIES

Компютърни науки и технологии

Издание

на Факултета по изчислителна техника и
автоматизация
Технически университет - Варна

Гл. редактор: доц. д-р Ю. Петкова

Редакционна колегия:

проф. д.н. Л. Личев (Острава)
проф. д.н. М. Илиев (Русе)
проф. д-р М. Лазарова (София)
проф. д-р Г. Спасов (Пловдив)
проф. д-р Т. Ганчев (Варна)
доц. д-р П. Антонов (Варна)
доц. д-р Р. Вробел (Вроцлав)
доц. д-р Н. Атанасов (Варна)
доц. д-р М. Маринов, (Русе)

Печат: ТУ-Варна

За контакти:

Технически университет - Варна
ФИТА
ул. „Студентска” 1, 9010 Варна,
България
тел./факс: (052) 383 320
E-mail: yulka.petkova@tu-varna.bg

ISSN 1312-3335

Computer Science and Technologies

Publication

of Computing and Automation Faculty
Technical University of Varna

Chief Editor: Assoc. Prof. Y. Petkova, PhD

Editorial Board:

Prof. L. Lichev, DSc (Ostrava)
Prof. M. Iliev, DSc (Ruse)
Prof. M. Lazarova, PhD (Sofia)
Prof. G. Spasov, PhD (Plovdiv)
Prof. T. Ganchev, PhD (Varna)
Assoc. Prof. P. Antonov, PhD (Varna)
Assoc. Prof. R. Wrobel (Wroclaw)
Assoc. Prof. N. Atanasov, PhD (Varna)
Assoc. Prof. M. Marinov, PhD (Ruse)

Printing: ТУ-Varna

For contacts:

Technical University of Varna
Faculty of Computing and Automation
1, Studentska Str., 9010 Varna,
Bulgaria
Tel/Fax: (+359) 52 383 320
E-mail: yulka.petkova@tu-varna.bg

ISSN 1312-3335

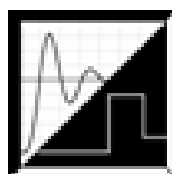
СЪДЪРЖАНИЕ

CONTENTS

1	SIMULATION RESEARCH OF A SYSTEM FOR CO-ORDINATED CONTROL OF TWO-MOTOR ELECTRIC DRIVE <i>Zhivko S. Zhekov</i>	6	СИМУЛАЦИОННО ИЗСЛЕДВАНЕ НА СИСТЕМА ЗА КООРДИНИРАНО УПРАВЛЕНИЕ НА ДВУДВИГАТЕЛНО ЕЛЕКТРОЗАДВИЖВАНЕ <i>Живко С. Жеков</i>	1
2	MODERN COMPUTATIONAL MODELS FOR INTERNET OF THINGS TECHNOLOGIES <i>Aydan M. Haka</i>	13	СЪВРЕМЕННИ ИЗЧИСЛИТЕЛНИ МОДЕЛИ ЗА ТЕХНОЛОГИИ ЗА ИНТЕРНЕТ НА ОБЕКТИТЕ <i>Айдън М. Хъкъ</i>	2
3	DATA ACQUISITION WITH OPENSOURCE COMEDI LIBRARIES AND MODULES WITH JAVA USING JAVA NATIVE INTERFACE UNDER LINUX OPERATING SYSTEM <i>Ivo Rakitin, Ventsislav Nikolov</i>	25	СБОР НА ДАННИ ПОСРЕДСТВОМ БИБЛИОТЕКИ И МОДУЛИ COMEDI С ОТВОРЕН КОД СВЪРЗАНИ С JAVA ЧРЕЗ JAVA NATIVE INTERFACE ПОД ОПЕРАЦИОННАТА СИСТЕМА LINUX <i>Иво Ракитин, Венцислав Николов</i>	3
4	AUTOMATION OF THE BANKING PAYMENT PROCESS IN COMPANIES <i>Ivan L. Ignev</i>	29	АВТОМАТИЗАЦИЯ НА ПРОЦЕСА НА ПЛАЩАНЕ ПО БАНКОВ ПЪТ В ПРЕДПРИЯТИЯТА <i>Иван Л. Игнев</i>	4
5	Z-WAVE – A SOLUTION TO THE CHALLENGES OF THE HOME AUTOMATION <i>Diyan Zh. Dinev, Bilyana Ilieva, Aydan Haka</i>	43	Z-WAVE – РЕШЕНИЕ ЗА ПРЕДИЗВИКАТЕЛСТВАТА НА ДОМАШНАТА АВТОМАТИЗАЦИЯ <i>Диян Ж. Динев, Биляна Илиева, Айдън Хъкъ</i>	5
6	STUDENT - PROGRAMMING LANGUAGE FOR EDUCATION IN COMPILERS <i>Ivaylo P. Penev, Dimitar S. Dimitrov</i>	49	STUDENT – ПРОГРАМЕН ЕЗИК ЗА ОБУЧЕНИЕ ПО КОМПИЛАТОРИ <i>Ивайло П. Пенев, Димитър Ст. Димитров</i>	6
7	BLOCKCHAIN CRYPTOCURRENCY TRADING SYSTEM FOR EDUCATIONAL PURPOSES <i>Diyan Zh. Dinev, Stanislav L. Lazarov</i>	57	БЛОКЧЕЙН СИСТЕМА ЗА ТЪРГОВИЯ НА КРИПТОВАЛУТИ ЗА ЦЕЛИТЕ НА ОБУЧЕНИЕТО <i>Диян Ж. Динев, Станислав Л. Лазаров</i>	7

8	DEVELOPMENT OF A TWO-JOINT ROBOT FOR EDUCATIONAL PURPOSES <i>Zhivko S. Zhekov</i>	65	РАЗРАБОТВАНЕ НА ДВУСТАВЕН РОБОТ ЗА УЧЕБНИ ЦЕЛИ <i>Живко С. Жеков</i>	8
9	HEALTHCARE SYSTEMS BASED ON BLOCKCHAIN <i>Magdalena Boycheva</i>	71	СИСТЕМИ В ЗДРАВЕОПАЗВАНЕТО, БАЗИРАНИ НА БЛОКЧЕЙН <i>Магдалена Бойчева</i>	9
10	DEVELOPMENT OF ARTIFICIAL INTELLIGENCE TECHNOLOGIES IN AUTONOMOUS VEHICLES <i>Marieta Yordanova</i>	79	РАЗВИТИЕ НА ТЕХНОЛОГИИТЕ С ИЗКУСТВЕН ИНТЕЛЕКТ В АВТОНОМНИТЕ ПРЕВОЗНИ СРЕДСТВА <i>Марията Йорданова</i>	10
11	COMPARISON OF DATA CLUSTERING ALGORITHMS <i>Ginka K. Marinova</i>	84	СРАВНЕНИЕ НА АЛГОРИТМИ ЗА КЛЪСТЕРИЗАЦИЯ НА ДАННИ <i>Гинка К. Маринова</i>	11
12	ADVANTAGES AND DISADVANTAGES OF MODERN SOFTWARE FRAMEWORKS <i>Ginka K. Marinova</i>	91	ПРЕДИМСТВА И НЕДОСТАТЪЦИ НА СЪВРЕМЕННИ СОФТУЕРНИ РАМКИ <i>Гинка К. Маринова</i>	12
13	ХАКАТОН M-CODELAB НА ТЕМА "УМНИ ТЕХНОЛОГИИ ЗА ПОЧИСТВАНЕ И ОПАЗВАНЕ НА МОРСКАТА ЕКОСИСТЕМА" СЕ ПРОВЕДЕ НА 19 И 20 НОЕМВРИ 2022 Г. В ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА <i>Цветослава Р. Къосева, Силвия И. Лешина, Бетина Р. Петкова</i>	100	HACKATHON M-CODELAB ON THE TOPIC "SMART TECHNOLOGIES FOR CLEANING AND PROTECTION OF THE MARINE ECOSYSTEM" WAS HELD ON NOVEMBER 19 AND 20, 2022 IN TECHNICAL UNIVERSITY OF VARNA <i>Tsvetoslava R. Kyoseva, Silvia I. Leshina, Betina R. Petkova</i>	13





СИМУЛАЦИОННО ИЗСЛЕДВАНЕ НА СИСТЕМА ЗА КООРДИНИРАНО УПРАВЛЕНИЕ НА ДВУДВИГАТЕЛНО ЕЛЕКТРОЗАДВИЖВАНЕ

Живко С. Жеков

Резюме: В статията се разглежда синтезирането на система за координирано управление на двудвигателно електрозадвижване, при което се извършва управление на два двигателя по синхронизиран начин, като второто електрозадвижване коригира движението си въз основа на състоянието на първото електрозадвижване. Извършено е моделиране на системата в конкретна програмна среда и са направени симулационни изследвания при различни настройки на регулаторите, потвърждаващи работоспособността ѝ.

Ключови думи: координирано управление, двудвигателно електрозадвижване, модул на оптимум

SIMULATION RESEARCH OF A SYSTEM FOR CO-ORDINATED CONTROL OF TWO-MOTOR ELECTRIC DRIVE

Zhivko S. Zhekov

Abstract: The report examines the synthesis of a system for coordinated control of a two-motor electric drive, in which two motors are controlled in a synchronized manner, with the second electric drive adjusting its movement based on the state of the first electric drive. The system has been modeled in a specific programming environment and simulation studies have been done at different settings of the regulators, confirming its operability.

Keywords: co-ordinated control, two-motor electric drive, modulus optimum

1. Въведение

Управлението на движението (motion control) е подклон на автоматизацията и обхваща системите или подсистемите, включени в движещи се части на машини, управлявани по определен начин. Системите за управление на движението се използват широко в различни сфери, в които процесите трябва да се автоматизират. Тези системи включват един или повече двигатели (електрически, пневматични, хидравлични), задвижващи един или повече изпълнителни механизми, (под)системи за управление на задвижванията (двигателите), датчици и контролер на движението при по-сложните системи, който да координира работата на всички задвижвания. Управлението може да се извършва в отворен или в затворен контур. Основните цели на една такава система е управление на положението, скоростта, ускорението и тласъка на изпълнителните механизми; управление на силата и/или въртящия момент. Други цели са: потискане на вибрациите на изпълнителните механизми; откриване на колизии (сблъсъци) и др. [1,2,3]

Системите за управление на движението с множество оси не използват непременно координирано управление на движението. При координираното управление на движението се извършва управление на няколко оси по синхронизиран начин, като всяка ос трябва да коригира движението си въз основа на състоянието (положение, скорост, ускорение) на останалите, така че работният орган да се придвижва с висока точност и голямо бързодействие от точка до точка или по зададена траектория. Поради тази причина системите за координирано управление на движението трябва да разполагат с

бързодействащи, затворени управляващи контури и обратни връзки по положение, скорост и ускорение с висока разделителна способност.

Съществуват следните основни схеми за координирано управление [4]:

- Схема с главно задвижване и подчинено задвижване (master/slave), при която главното задвижване изпълнява основното задание по скорост или положение, а неговият датчик на скорост или положение се явява източник на задание на подчиненото задвижване, което следва движението на главната ос. Двата двигателя може задвижват един общ или два различни изпълнителни механизма. Подчиненото задвижване може и да е повече от едно.
- При втората схема два или повече двигателя са куплирани към една ос. Основното задание по скорост или положение се подава едновременно и на двете задвижвания, а те имат за задача да го изпълняват колкото може по-точно и бързо. Проблемът тук е, че може да има разлика между характеристиките на двата двигателя и натоварването им, което води до разлика в изпълнението на заданието.
- При последната схема задвижванията на отделните двигатели получават различни основни задания по положение или скорост от контролер на по-високо йерархично ниво, който следи движението на работния орган и генерира тези задания, така че отработени правилно от задвижваните изпълнителни механизми, работният орган да изпълни предварително определената траектория на движение.

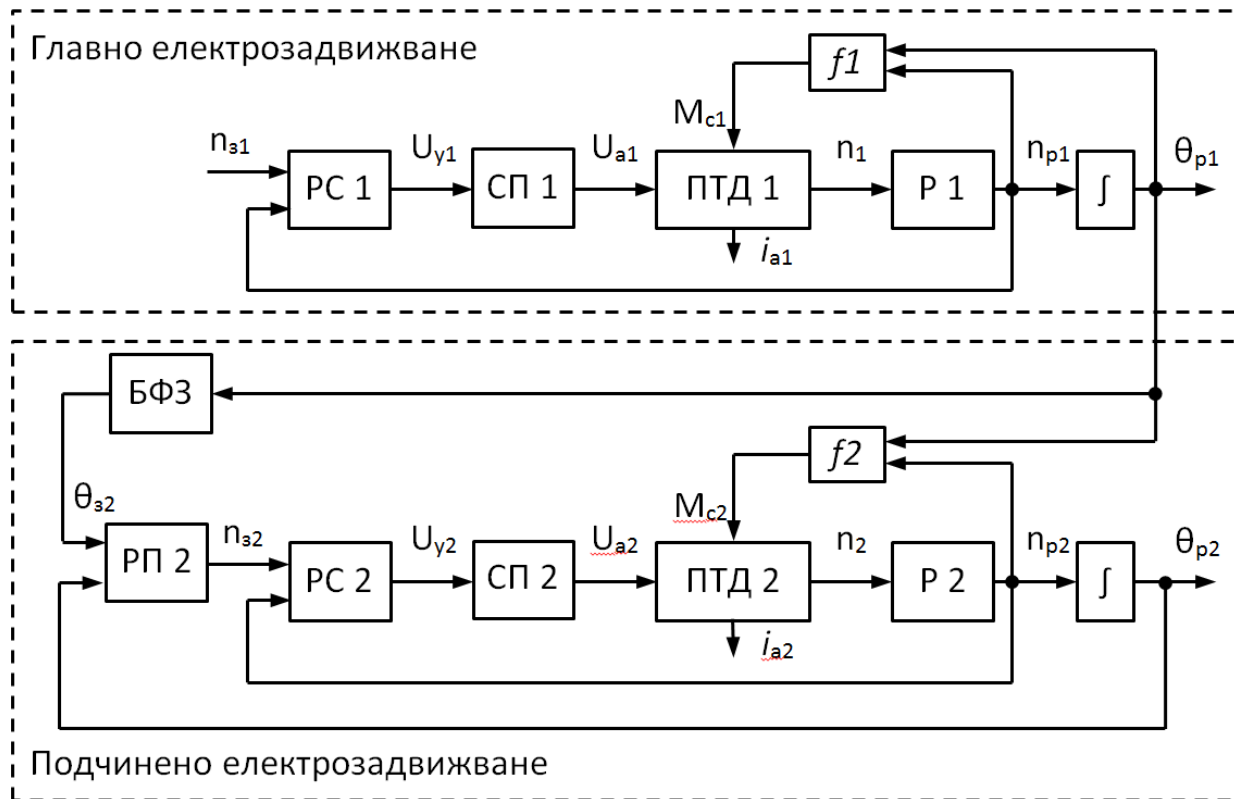
В настоящата публикация е представено моделиране на система от първия вид, при който трябва да се синхронизира работата на две оси – главна и подчинена (master/slave), които се задвижват от електродвигатели. Синтезирана е системата за управление, направени са настройки на регулаторите и са проведени симулационни изследвания при различни режими на работа.

2. Конфигуриране на системата за управление

Блоквата схема на системата е показана на фигура 1, където с индекс 1 и 2 са показани блокове, променливи и параметри, съответно на главното и подчиненото електрозадвижване. РС - регулатор на скорост, СП – силов електронен преобразувател, ПТД – постоянен ток двигател, Р – редуктор, f – нелинейни функции, формиращи съпротивителните моменти, БФЗ – блок за формиране на заданието по положение на подчинения канал, РП – регулатор на положение на подчинения канал.

Системата е двуканална, като първият канал (главен) е едноконтурен със затворен контур по скорост, а вторият канал (подчинен) е двуконтурен със затворени контури по скорост и положение. Целта е първият канал да поддържа постоянна скорост на ПТД1 и респективно на съответния изпълнителен механизъм*. След надвишаване на определено ъглово положение към втория канал се подава задание по положение, което е функция на положението на първия изпълнителен механизъм. При правилно отработване на заданието от втори канал вторият изпълнителен механизъм настига първия и определен ъглов път се движат заедно (синхронно), след което към втори канал се подава нулево задание по положение и вторият изпълнителен механизъм се връща в изходно положение. След това цикълът се повтаря. Пример за такъв тип механизъм е т.нар. „Летяща ножица“.

(* - Разглеждат се ъгловите положения на валовете на двигателите, а изпълнителните механизми се отчитат единствено чрез инерционните си моменти и оказването на съответните съпротивителни моменти към двата двигателя.)



Фиг. 1. Блокова схема на системата

n_{31} , n_1 , n_{p1} – съответно ъглови скорости в първи канал: задание по скорост; скорост на вала на ПТД1; скорост на вала на P1. n_{32} , n_2 , n_{p2} – съответно ъглови скорости във втори канал: задание по скорост; скорост на вала на ПТД2; скорост на вала на P2. θ_{p1} – ъглово положение на вала на P1, θ_{p2} – ъглово положение на вала на P2, θ_{32} – задание по положение на подчинения канал. U_{y1} и U_{y2} – управляващи напрежения на двата силови преобразувателя, U_{a1} и U_{a2} – напрежения подавани към котвените намотки на двата двигателя, i_{a1} и i_{a2} – котвени токове, M_{c1} и M_{c2} – съпротивителни моменти.

3. Настройка на регулаторите

Настройката на регулаторите е извършена по модул (технически) оптимум (мо) [5].

- Настройка на PC1

Некомпенсируемото звено е силовият преобразувател, който се представя със следната предавателна функция (1), където: $k_{сп}$ – коефициент на усилване, $T_{сп}=1/(2 \cdot f_{шим})$ – еквивалентна времеконстанта, $f_{шим}$ – носеща честота на широчино-импулсната модулация.

$$W_{cn1}(p) = \frac{k_{cn1}}{T_{cn1}p + 1} \quad (1)$$

Предавателната функция на ПТД с независимо възбуждане ($\Phi = \text{const}$), даваща връзка между скоростта и напрежението, е (2), където: $k_{д1}=1/c\Phi_1$, $T_{a1}=L_{a1}/R_{a1}$, $T_{m1}=(J_{\Sigma 1}R_{a1})/(c^2\Phi_1^2)$.

$$W_{\omega 1, u1}(p) = \frac{k_{\omega 1}}{T_{m1}T_{a1}p^2 + T_{m1}p + 1} \quad (2)$$

При $T_m > 10T_a$ електромагнитната времеконстанта T_a може да се пренебрегне в сравнение с електромеханичната T_m , при което в динамично отношение двигателят може да се разглежда като апериодично звено – (3).

$$W_{\omega 1, u1}(p) = \frac{k_{\partial 1}}{T_{m1}p + 1} \quad (3)$$

Предавателната функция на първото (то е и единствено, тъй като първи канал е едноконтурен) компенсируемо звено на обекта е (4), където: $k_{n,\omega} = 1/(2\pi)$ – преобразува дименсия [рад/с] в [об/с], k_{p1} - коефициент на редуктора.

$$W_{o11}(p) = \frac{k_{\partial 1} \cdot k_{n,\omega} \cdot k_{p1}}{T_{m1}p + 1} \quad (4)$$

Предавателната функция на PC1 при настройка по модулес оптимум е (5), където k_{ov11} - коефициент на обратната връзка по скорост.

$$W_{pc1}^{mo}(p) = \frac{1}{T_{cn1}p} \cdot \frac{1}{W_{o11}(p)} \cdot \frac{1}{k_{cn1}k_{ov11}} = \frac{T_{m1}}{T_{cn1}k_{\partial 1}k_{n,\omega}k_{p1}k_{cn1}k_{ov11}} + \frac{1}{T_{cn1}k_{\partial 1}k_{n,\omega}k_{p1}k_{cn1}k_{ov11}p} \quad (5)$$

При единична обратна връзка (както се вижда от фиг.1)

$$W_{pc1}^{mo}(p) = \frac{T_{m1}}{T_{cn1}k_{\partial 1}k_{n,\omega}k_{p1}k_{cn1}} + \frac{1}{T_{cn1}k_{\partial 1}k_{n,\omega}k_{p1}k_{cn1}p} = k_{n11} + \frac{k_{u11}}{p} \quad (6)$$

- Настройка на PC2 и РП2.

Предавателна функция на първото компенсируемо звено на обекта във втори канал е (7), k_{ov21} - коефициент на обратната връзка по скорост, а останалите коефициенти са аналогични.

$$W_{o21}(p) = \frac{k_{\partial 2} \cdot k_{n,\omega} \cdot k_{p2}}{T_{m2}p + 1} \quad (7)$$

Настройката на PC2 става аналогично както при PC1 и предавателната му функция е (8).

$$W_{pc2}^{mo}(p) = \frac{1}{T_{cn2}p} \cdot \frac{1}{W_{o21}(p)} \cdot \frac{1}{k_{cn2}k_{ov21}} = \frac{T_{m2}}{T_{cn2}k_{\partial 2}k_{n,\omega}k_{p2}k_{cn2}k_{ov21}} + \frac{1}{T_{cn2}k_{\partial 2}k_{n,\omega}k_{p2}k_{cn2}k_{ov21}p} \quad (8)$$

При единична обратна връзка

$$W_{pc2}^{mo}(p) = \frac{T_{m2}}{T_{cn2}k_{\partial 2}k_{n,\omega}k_{p2}k_{cn2}} + \frac{1}{T_{cn2}k_{\partial 2}k_{n,\omega}k_{p2}k_{cn2}p} = k_{n21} + \frac{k_{u21}}{p} \quad (9)$$

Предавателна функция на второто компенсируемо звено на обекта във втори канал е (10).

$$W_{o22}(p) = \frac{1}{p} \quad (10)$$

Предавателната функция на РП2 при настройка по модулес оптимум е (11), където k_{ov22} - коефициент на обратната връзка по положение.

$$W_{pn2}^{mo}(p) = \frac{1}{4T_{cn2}p} \cdot \frac{1}{W_{o22}(p)} \cdot \frac{k_{oe21}}{k_{oe22}} = \frac{k_{oe21}}{4T_{cn2}k_{oe22}p} \quad (11)$$

При единични обратни връзки

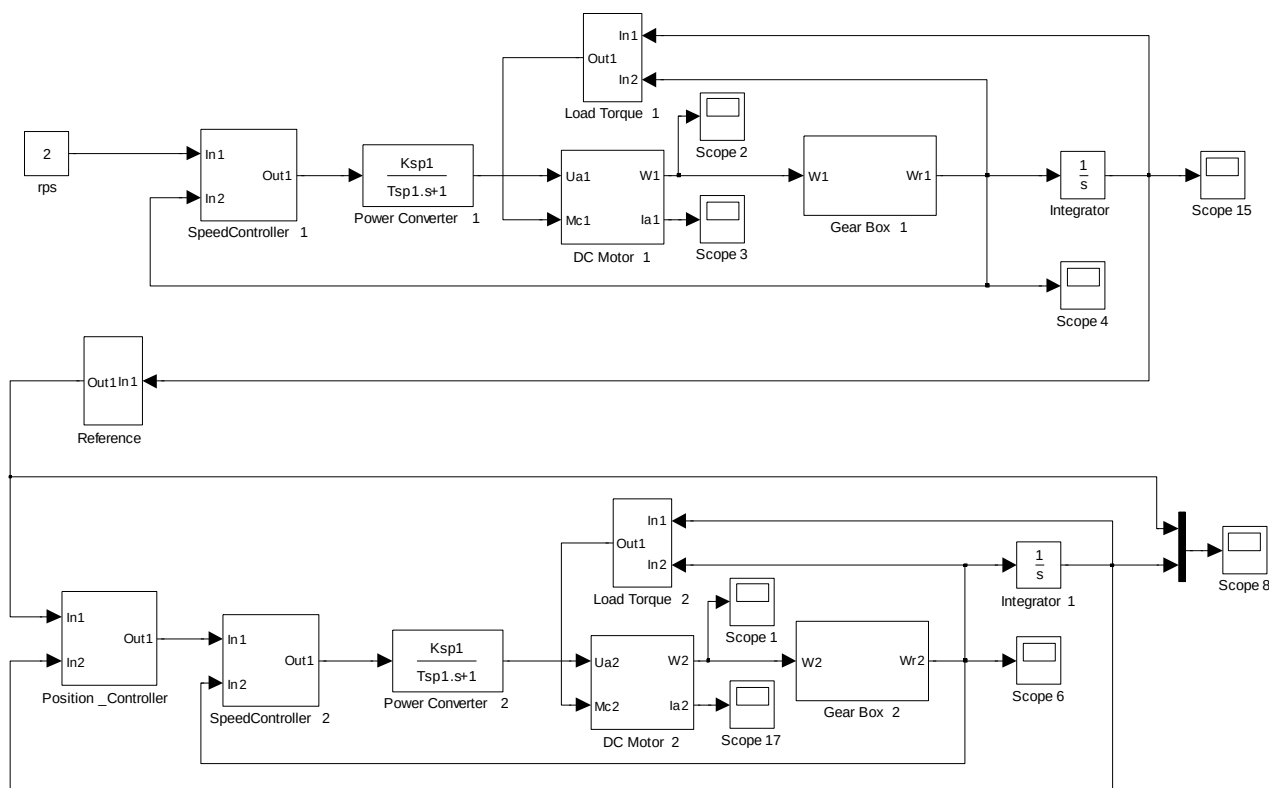
$$W_{pn2}^{mo}(p) = \frac{1}{4T_{cn2}p} = k_{n22} \quad (12)$$

Регулаторите на скорост и за двата канала са ПИ-регулатори, а регулаторът на положение за втори канал е пропорционален. Въпреки че регулаторът на положение е пропорционален, понеже съпротивителният момент, като основното смущение в електрозавиждаването, се отработва от вътрешния контур по скорост, няма да има статична грешка, но при стъпаловидно задание.

4. Симулационни изследвания и резултати

Моделирането на системата е извършено в програмна среда Matlab/Simulink. Двата двигателя (модел ПИВТ 6-25/3А) и редукторите са еднакви, тъй като се предполага, че в следствие на тяхна база ще бъде разработен лабораторен макет и имат следните параметри: $U_{a1H}=U_{a2H}=30V$; $i_{a1H}=i_{a2H}=1.4A$; $M_{1H}=M_{2H}=0.1Nm$; $k_{d1}=k_{d2}=13.9rad/c/B$; $T_{a1}=T_{a2}=0.002c$; $T_{M1}=T_{M2}=0.051c$ (отчетени са и инерционните моменти на механизмите); $k_{p1}=k_{p2}=1/10$. Поради еднаквите двигатели силовите преобразуватели и те са еднакви и имат следните параметри: $k_{cn1}=k_{cn2}=30$; $T_{cn1}=T_{cn2}=0.0005c$. $k_{no}=1/(2\pi)$.

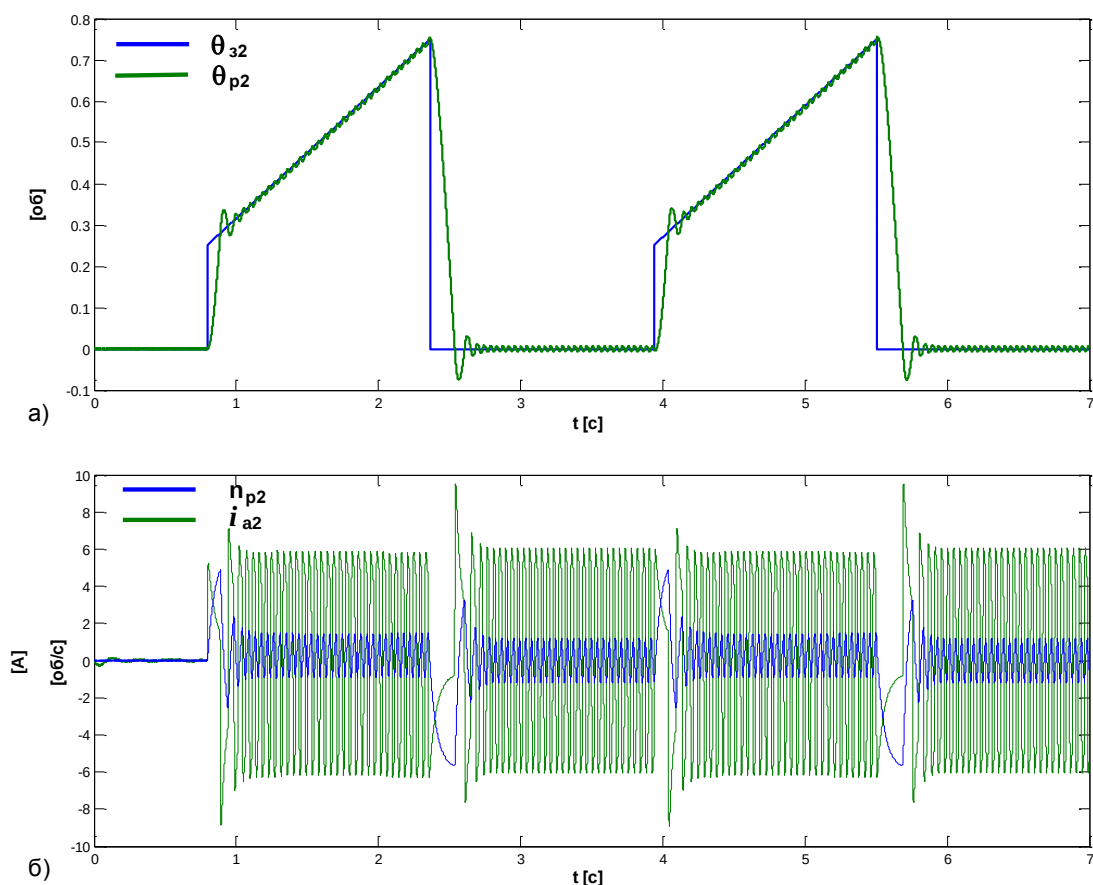
При така посочените параметри за коефициентите на двата регулатора на скорост се получава: $k_{n11}=k_{n21}=1.5$; $k_{n11}=k_{n21}=300$. Изходите на тези регулатори и интегралните им компоненти са ограничени в диапазона $[-1V \div 1V]$. За коефициента на регулатора на положение се получава: $k_{n22}=500$. Изходът му е ограничени в диапазона $[-6.7об/с \div 6.7об/с]$.



Фиг. 2. Модел на системата в Simulink

На фиг. 2 е показан моделът на системата в програмна среда Matlab/Simulink. На фигури 3, 4, 5 са представени по два цикъла от работата на втори канал (подчинен), като на тези с индекс **а)** е показано заданието по положение и отработването му, а на тези с индекс **б)** са показани скоростта след редуктора и тока на ПТД2. Заданието по положение θ_{32} се подава към втория канал на системата, след като положението на вала на редуктор Р1 достигне до 0.25об, а след достигане до 0.75об заданието се нулира. ПТД2 и ПТД1 са натоварени до $0.5M_H$ с реактивен съпротивителен момент от тип „сухо триене“, като при синхронизиране на работата на двата канала натоварването се повишава до номиналния момент на двигателите M_H .

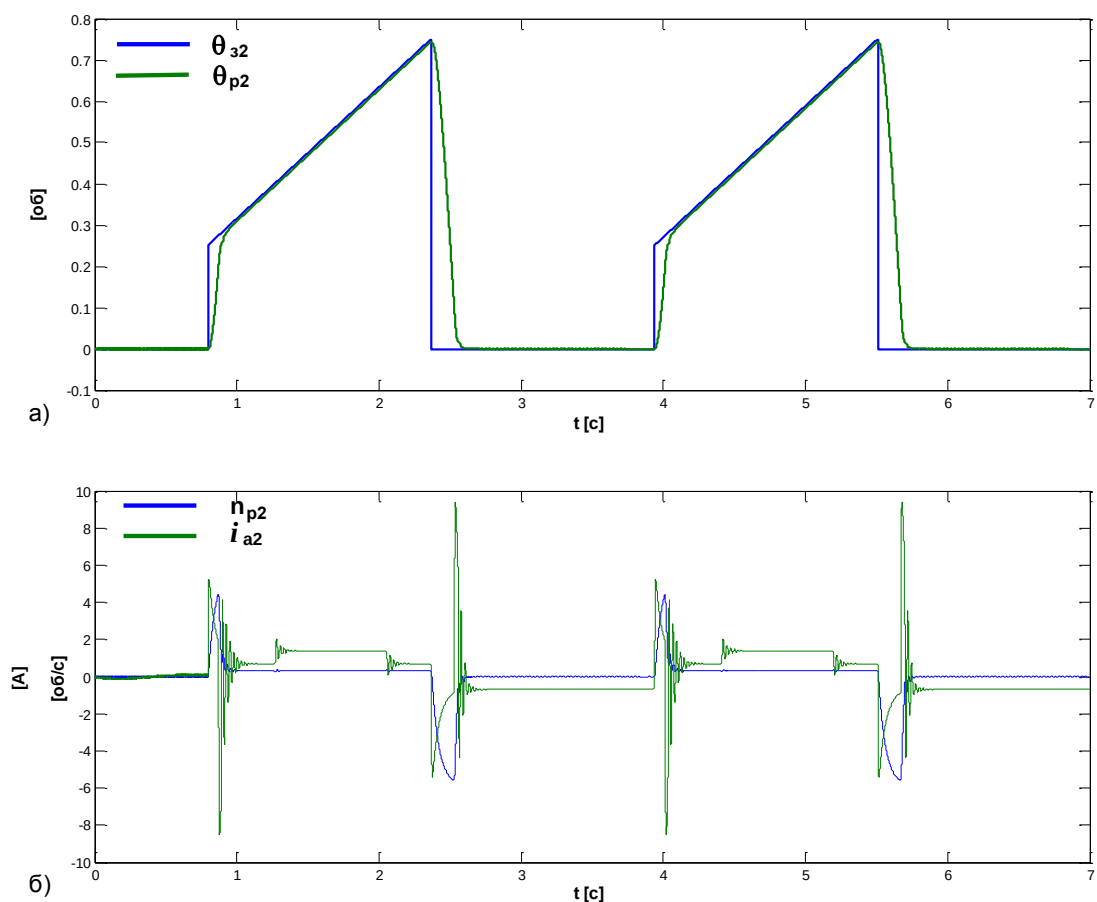
Работата на първи канал (главен) не е представена, тъй като заданието по скорост е едно и също $n_{31}=2\text{об/с}$ и след протичането на първоначалния преходен процес скоростта се стабилизира на 2об/с с изключение на моментите, в които ПТД1 се натоварва и разтоварва.



Фиг. 3. Работа на първи канал на системата при $k_{п22}=500$

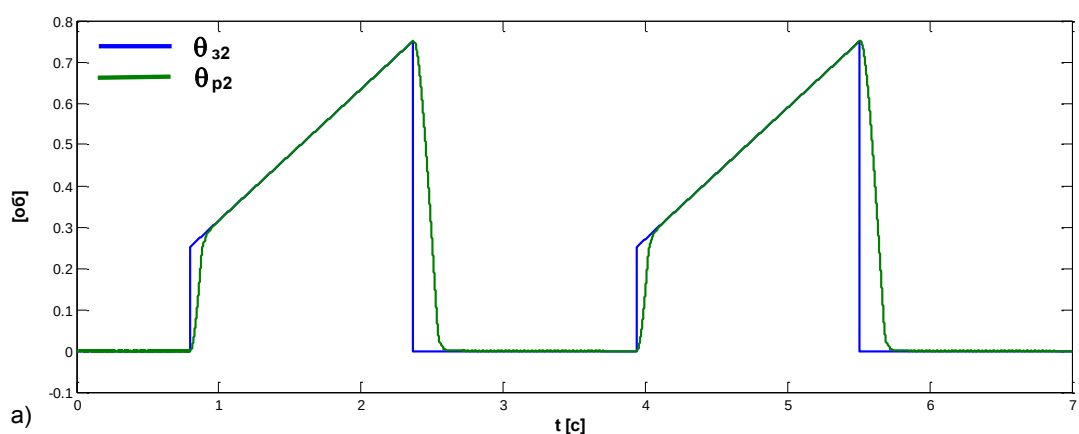
Както се вижда от фигура 3, при $k_{п22}=500$ (коефициента на РП2), преходните процеси на тока i_{a2} , скоростта n_{p2} и положението θ_{p2} са с незатихващи колебания. Освен това средната ефективна стойност на тока i_{a2} на ПТД2 е от порядъка на 4.5А, което е недопустимо. Поради тези причини коефициентът на РП2 е намален 10 пъти.

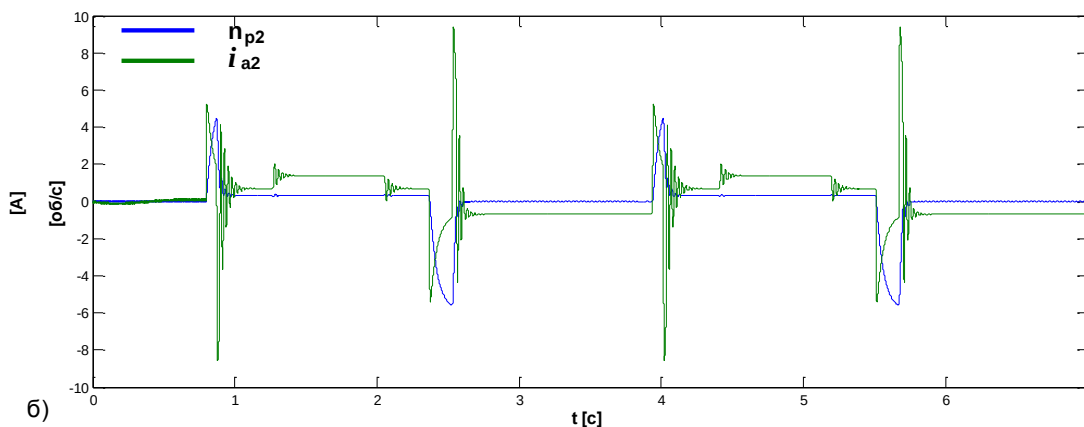
На фигура 4 а) се наблюдават апериодични преходни процеси на θ_{p2} , но поради това, че заданието θ_{32} е линейно нарастващо, а РП2 е пропорционален, има грешка в установен режим - $\varepsilon_\theta=6.5 \cdot 10^{-5}$ об. Преходните процеси на тока са колебателни, като за кратко токът достига до 9А, тъй като няма контур по ток, който да го ограничи, но средната му ефективната стойност е 1.14А, което е по-малко от номиналния ток на двигателя.



Фиг. 4. Работа на първи канал на системата при $k_{n22}=50$

За повишаване на бързодействието на системата към изхода на РП2 е добавена съставка, съдържаща производната на заданието θ_{32} , т.е. се реализира комбинирано управление по отклонение и по задание – фигура 5. Освен това грешката в установен режим при линейно нарастване на заданието е практически равна на нула. Токът и скоростта са почти идентични с тези от предходния експеримент. Същото важи и за средната ефективна стойност на тока, която отново е около 1.14A.





Фиг.5. Работа на първи канал на системата при $k_{n22}=50$

5. Заключение

Синтезирана е система за координирано управление на двудвигателно електрозадвижване, при което се извършва управление на два двигателя по синхронизиран начин, като второто електрозадвижване коригира движението си въз основа на състоянието на първото електрозадвижване. Използвани са два еднакви електродвигателя, тъй като се предполага, че в следствие на тяхна база ще бъде разработен лабораторен макет. Установено е, че настройката по модул на оптимум на регулатора на положение дава незатихващи колебателни преходни процеси. За да се отстрани този недостатък, е намален коефициентът на пропорционалност, а за да се подобри бързодействието на системата, е добавена съставка, съдържаща производната на заданието по положение и е реализирано комбинирано управление. Това води до задоволителни показатели на качеството, като същевременно средната ефективна стойност на тока на двигателите не надхвърля техния номинален ток. По време на преходните процеси токът кратковременно надвишава неколкостранно номиналния ток, тъй като не е предвиден контур по ток, в който да има токоограничение.

Литература

- [1]. Jun M., L. Xiaocong, T. Kiong, Advanced Optimization for Motion Control Systems, CRC Press, Taylor & Francis Group, 2020.
- [2]. Yamaguchi T., M. Hirata, Ch. Pang, High-Speed Precision Motion Control, CRC Press, Taylor & Francis Group, 2012.
- [3]. Gurocak H., Industrial Motion Control, John Wiley & Sons, 2016
- [4]. Kiong T., L. Heng, H. Sunan, Precision Motion Control, Springer-Verlag, 2008.
- [5]. Маринов Е., Ж. Жеков, Ц. Тодоров, Управление на електрозадвижвания (наръчник), ТУ-Варна, 2014.

За контакти:

д-р инж. Живко С. Жеков
 катедра „Автоматизация на производството”
 Технически университет - Варна
 E-mail: fitter@abv.bg

СЪВРЕМЕННИ ИЗЧИСЛИТЕЛНИ МОДЕЛИ ЗА ТЕХНОЛОГИИ ЗА ИНТЕРНЕТ НА ОБЕКТИТЕ

Айдън М. Хъкъ

Резюме: С развитието на съвременните комуникационни технологии продължава развитието и навлизането на технологиите за Интернет на обектите в различни области от живота. Наред с разрастването на тези технологии възниква нуждата от извършване на бързи и надеждни изчислителни процеси. За извършването на изчислителните процеси при технологиите за Интернет на обектите се използват основно три модела – Cloud Computing, Edge Computing и Fog Computing. В тази статия се разглеждат тези три модела, техните архитектури, предимства, недостатъци и перспективи.

Ключови думи: Cloud Computing, Edge Computing, Fog Computing, IoT.

Modern computational models for Internet of Things technologies

Aydan M. Haka

Abstract: With the development of modern communication technologies, the development and incursion of Internet of Things technologies continues in diverse areas of life. Along with the growth of these technologies comes the need to perform fast and reliable computing processes. Three models are mainly used for computing processes in Internet of Things technologies - Cloud Computing, Edge Computing and Fog Computing. This article examines these three models, their architectures, advantages, disadvantages, and perspectives.

Keywords: Cloud Computing, Edge Computing, Fog Computing, IoT

1. Въведение

Съвременните информационни и комуникационни технологии се развиват с изключително бързи темпове. Една от активно развиващите се области на тези технологии са комуникационните инфраструктури, базирани на 4G и 5G, както и тези от ново поколение 6G. Наред с тях се разраства използването и на технологиите за Интернет на обектите (Internet of Things - IoT). Тези технологии заемат място в много области от съвременния живот, а целта им е подобряване на функциите и обслужването в тези области. За подобряване работата, отказоустойчивостта, сигурността и бързодействието на технологиите в тази област се използват различни изчислителни модели, целящи подобряване на качеството на обслужване и потребителското изживяване [1]. Три са основните изчислителни модели, които се използват в работата с технологии за Интернет на обектите – Cloud Computing, Edge Computing и Fog Computing.

В тази статия се разглеждат основните три изчислителни модела, използвани при работата с технологии за Интернет на обектите. Представени са техните особености, архитектури и ползи. Представени са разликите между трите модела с цел определяне на най-подходящия за работа с технологии за Интернет на обектите.

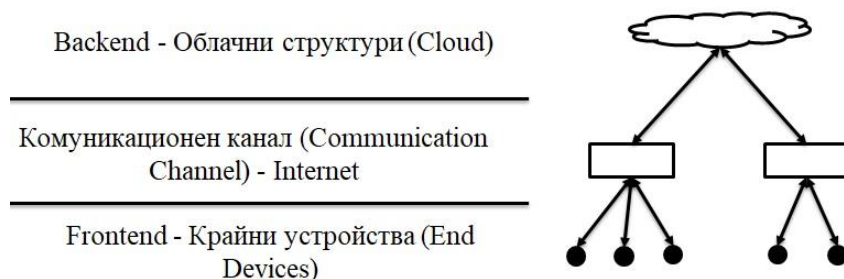
2. Моделът Cloud Computing

Моделът Cloud Computing представлява технология, която позволява на потребителите да имат достъп или да получават различни услуги от различни части на света през Интернет. Облачните изчисления не са само специфична технология, а голяма форма от няколко

специализирани технологии, известни като облачни бизнес модели [2]. Облачните изчисления са процес на предоставяне на всякакъв вид онлайн услуга на потребителя чрез Интернет като достъп до сървъри, уеб приложения, бази данни, софтуер, работа в мрежа, интелигентно обслужване, анализ и др. Облачните изчисления са пространство, където може да се съхраняват и използват данни, а също така представлява пространството, от което е с малка вероятност данните да бъдат загубени, изтрити или повредени. Националният институт за стандарти и тестване (National Institute of Standards and Testing - NIST) дефинира облачните изчисления като модел за позволяване на повсеместен, удобен мрежов достъп до споделен набор от конфигурируеми изчислителни ресурси, които могат бързо да бъдат осигурени и освободени с минимални усилия за управление или взаимодействие с доставчика на услуги. Има четири типа модели за внедряване на облачни изчисления: публичен, частен, хибриден и за общност [3]:

- Публичен (Public Cloud) – свободен и безплатен за всички. Тъй като потребителите съхраняват много от чувствителните си данни в облака, това повдига въпроса за сигурността на данните, понеже услугата е достъпна за всеки. Този тип услуга се предоставя от компании като Amazon, Microsoft, IBM, Google или Alibaba. Някои от предимствата на публичната облачна услуга са рентабилност, гъвкавост, потребителска съвместимост, независимост на местоположението;
- Частен (Private Cloud) – използва се за съхранение на данни на частни компании или организации. Тези видове частни облаци се управляват вътрешно от компанията или от трета страна. Тъй като частният облак обслужва само конкретни потребители, той поддържа високо ниво на сигурност и поверителност;
- Хибриден (Hybrid Cloud) – представлява комбинация от публичен и частен облак. Обикновено институции като банки, финансови дружества, заведения за здравеопазване, университети използват този модел. В тези институции не всички типове данни са еднакво важни. Част от тях са много чувствителни, така че трябва да се съхраняват в частен облак, а друга част не са и няма възможност за загуба на данни, така че може да бъдат съхранявани в публичен;
- За общност (Community Cloud) – работи като публичния облак. Разликата е, че този модел за облачни изчисления предоставя услуга в рамките на конкретна общност. Различни общности в различни държави, като военни, полицейски служби и много други организации, използват този тип облак. Има и много други видове облачни структури като distributed-cloud, multi-cloud или inter-cloud.

Архитектурата на облачните изчисления се състои от два основни компонента – Frontend и Backend [4]. Частта Frontend работи като клиент и комуникира с Backend чрез интернет (фигура 1). При облачните изчисления клиентската страна или интерфейсът е видим за крайния потребител и може да изпраща заявки към Backend частта чрез комуникационната среда. Защитата на данните и отговорът на заявките, зададени от Frontend, се осигурява от Backend частта.

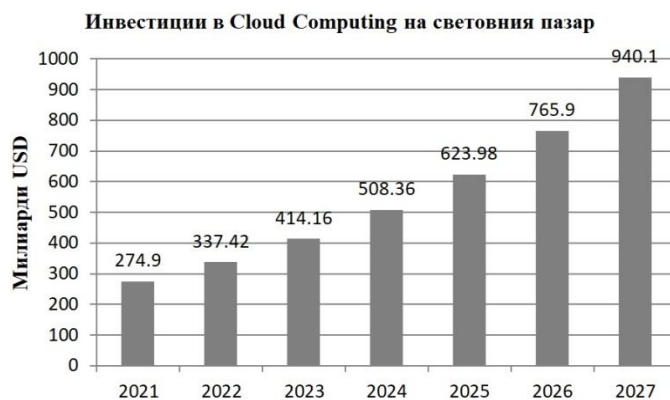


Фиг. 1. Абстрактен модел на Cloud Computing

Ползите от модела Cloud Computing се изразяват в [5]:

- Цена – елиминират разходите за закупуване на хардуер, софтуерни центрове за данни, сървъри, електричество за захранване и охлаждане. Освен това премахват разходите за администриране и управление на инфраструктурата;
- Производителност – поддържа и осигурява висока производителност на големи групи от различни потребители, както и оперативната съвместимост и мащабируемост между разнородни и разпределени обекти, свързани с Интернет;
- Скорост – високоскоростните услуги, предлагани от облачните изчисления, и огромните количества изчислителни ресурси могат да бъдат осигурени бързо за IoT приложения, което ги прави адаптивни за потребителите;
- Сигурност – сигурността и поверителността на потребителската информация са най-предизвикателната задача, идентифицирана в рамките на IoT. Облачните изчисления предоставят политики и технологии, които укрепват цялостната сигурност.

Статистиките и прогнозите за инвестициите в облачните изчисления показва, че в световен мащаб се инвестират и ще продължат да се инвестират финансови средства, като през 2027г. се очаква да достигнат над 900 милиарда USD със среден ръст от около 18.53% (милиарди USD) на година (фигура 2) [6].



Фиг. 2. Инвестиции в Cloud Computing

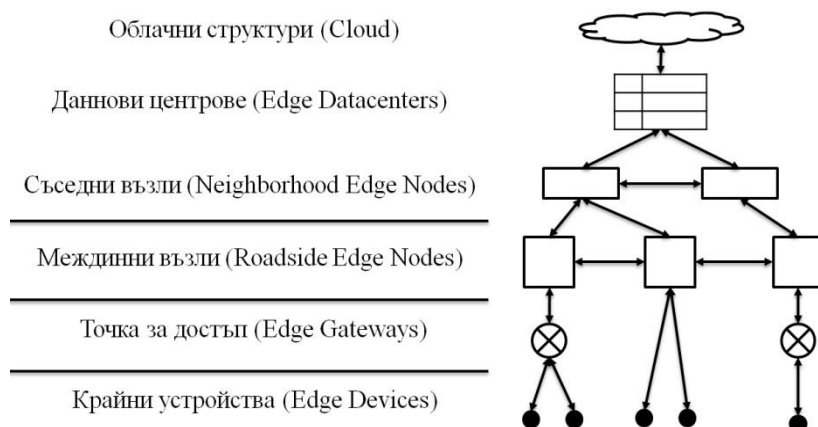
3. Моделът Edge Computing

Моделът Edge Computing измества изчислителните ресурси от изчислителните центрове (Data Centers) и облачните структури (Cloud) по-близо до крайните устройства. Целта на този модел е поддържане на по-ниски изисквания за латентност, с възможност за по-ефективно обработване на данните, за спестяване на мрежови разходи. Интернет на обектите е специфичен пример, при който може да се приложи този модел, заради огромния трафик, който се генерира от милиардите устройства, внедрявани всяка година [7]. Когато данните се обработват в област, по-близка до крайните устройства вместо в облачна структура, разходите за пренос се редуцират. Възможните печалби от модела Edge Computing включват приходи от всеки, който се възползва от по-висока скорост на данните и изчислителната мощност осигурена по-близо до крайния потребител. Разликата между Edge и Fog Computing е, че Fog Computing описва децентрализацията на изчислителната инфраструктура или извеждането на облачната структура по-близо до крайния потребител. Моделът Fog Computing позволява повтарящи се структури в концепцията на Edge Computing, така че предприятията да могат лесно да изтласкат изчислителната мощност от своите централизирани системи или облачни структури, за да подобрят скалируемостта и производителността [8].

Съществуват различни типове Edge Computing технологии като [9]:

- Fog Computing – отнася се до децентрализация на изчислителна инфраструктура чрез разширяване на облачна инфраструктура със стратегическо разполагане на възли между облака и крайните устройства. Така данните, изчисленията, съхранението и приложенията се разполагат по-близо до потребителя или устройствата за Интернет на обектите, където данните се нуждаят от обработка, като по този начин създава „мъгла“ извън централизираната облачна структура и намалява времето за предаване на данни, което може да е необходимо за обработката им;
- Edge Computing с множествен достъп (Multi-access Edge Computing – MEC) – според дефиницията на Европейския институт за телекомуникационни стандарти MEC предлага на разработчиците на приложения и доставчиците на съдържание възможности за облачни изчисления и среда за информационни услуги близо до мрежа от крайни устройства. Тази среда се характеризира с изключително ниска латентност и широка честотна лента, както и достъп в реално време до информация за радиомрежата, която може да се използва от приложения;
- Микроцентрове за данни – те предоставят същите компоненти като традиционните центрове за данни, но могат да бъдат разположени близо до източника на данни. Характеризират се с голяма гъвкавост, която позволява да бъдат бързо разгръщани в райони с недостатъчно обслужване или центрове за бедствия;
- Малки облачни структури – представляват малки центрове за данни с подобрена мобилност, поставени в непосредствена близост до крайни устройства. Те са специално проектирани да подобрят интерактивни мобилни приложения използващи множество ресурси чрез допълнителната наличност на изчислителни ресурси с ниска латентност;
- Звена за спешно реагиране – тези самостоятелни, мобилни звена установяват оперативно съвместими комуникации за лицата, реагиращи първи при извънредни ситуации. Те могат да бъдат разгърнати бързо във всеки кризисен обект, заедно с висококвалифициран екип за тактически операции, за да възстановят комуникациите за засегнатите райони.

Диаграмата на фигура 3 представя идеята зад архитектурата на Edge Computing на различни нива [10]. Най-горното ниво е това на облачните технологии, където данните се съхраняват под формата на големи масиви от данни без ограничение за съхранение. Това е централна част за съхранение и поддръжка на данните, но при Edge Computing това ниво се прилага като система за дългосрочно съхранение, но не се прилага за всяка междинна обработка. В последното ниво са разположени физическите устройства, които са с най-малки изчислителни възможности и много ограничена наличност на ресурси. Тези устройства извършват едно определено действие като светофари и инструменти за измерване на температура. Има две междинни нива между първото и последното. Първото междинно ниво се разглежда като съвкупност от възли, разположени преди йерархичните облачни мрежи. Тези възли имат възможността да извършват голям брой изчисления и формират маршрут и могат да генерират карта за предаване на информация като базови станции, даннови центрове с малък размер, работни станции и маршрутизатори. Второто междинно ниво е известно като точка за достъп или Gateway, с по-малки изчислителни възможности от възлите в първото междинно ниво. Това е нивото над това на крайните възли и включва сензори в коли, „умни“ телефони и детектори за движение.



Фиг. 3. Абстрактен модел на Edge Computing

Ползите от модела Edge Computing се изразяват в [11]:

- Намалена латентност и увеличена скорост – изнасянето на изчислителните действия по-близо до крайните устройства, влияе положително на латентността и скоростта;
- Сигурност – данните се анализират локално и се защитават от правилата в локалната мрежа или от затворена система на доставчик на услуги. При Edge Computing данните се препращат според изискването и заявката, направени от устройството, така че има много по-малка възможност от заплахата за сигурността, тъй като съхраняваните и споделени информационни данни са много малки по размер в сравнение с оригиналните данни, налични в центъра за данни. Ако данните се разпределят между различни крайни изчислителни възли, има по-малко заплахи за сигурността;
- Намаляване на разходите – позволява оптимизиране на потока от данни, което довежда до намалена честотната лента и разходи. Разходите за резервиране могат да бъдат сведени до минимум, ако информационните данни се съхраняват на крайни изчислителни устройства под формата на „бисквитки“, ако има често изискване за предаване на същите информационни данни. Следователно няма да има нужда от голямо хранилище в центъра за данни в облака. Въпреки че данните не са подобни, при използване на част от тях периферните изчислителни възли ги извличат от облачната структура с една операция. По този начин оптимизира потока от данни, за намаляване на оперативните разходи в облака;
- Ефективност и надеждност – Edge устройствата локално съхраняват и обработват данни и работят с Edge даннови центрове с цел преодоляване на всички периодични проблеми със свързаността. Крайните изчислителни възли и устройства се сглобяват според изискванията, за да работят в съответната среда. Това не води до загуба на данни, ако има прекъсване на честотната лента и предаването на информация през центъра за данни. Ефективността на бизнеса се постига чрез ниска латентност и минимално съхранение на данни в облака.
- Скалируемост – установяването на Edge даннови центрове и устройства за Интернет на обектите осигурява на бизнеса бързо мащабиране на своите операции. Крайният изчислителен възел е свързан с устройствата на крайния потребител, като сензори и ad hoc безжични мрежи, и няма ограничение за капацитета за съхранение на устройствата. Инструментът за обработка на данни и управлението им се намира в крайния възел, така че да може да изпълнява заявката, инициирана от устройствата, вместо да забавя времето си за реакция в координацията на центъра за данни, разположен на отдалечено разстояние;
- Подобряване производителността на приложенията – за целта информационните данни трябва да присъстват много близо до крайния потребител, което води до най-

добро време за реакция без никакво забавяне при предаването. За даден конкретен отрязък от време, ако предаването на данни е много бавно, то ще се счита за неуместно.

Прогнозите и статистиките за инвестициите в Edge Computing показват, че финансовите инвестиции в областта са значителни, като се очаква през годините да се увеличават със средно 12.46% (милиарди USD) (фигура 4) [12].



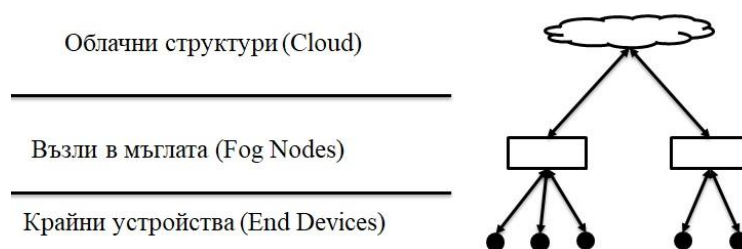
Фиг. 4. Инвестиции в Edge Computing

4. Моделът Fog Computing

Терминът „Fog Computing“ е въведен от Cisco Systems като нов модел за улесняване на безжичния трансфер на данни към разпределени устройства в мрежата за Интернет на обектите [13]. Устройствата в тези мрежи генерират огромно количество данни, които трябва да бъдат обработени и анализирани. Такава обработка изисква много ресурси (процесорна мощ, памет, захранване и др.), а те при крайните устройства са често ограничени. Това изисква прехвърляне на изчислителните задачи от крайните устройства към други, по-мощни устройства.

Изчисленията може да се прехвърлят върху облачни структури, но при тях има възможност да възникнат проблеми свързани със забавяне в предаването при критични по отношение на времето приложения, латентност и тесни места по комуникационната линия. С цел преодоляване слабостите на облачните структури се въвежда междинен слой между крайните устройства и облачните структури, известен като Fog Computing. В този среден слой се разполагат устройства с по-голяма изчислителна мощ, известни като „възли в мъглата“ (фигура 5). Тези възли представляват изчислителни ресурси, разположени по географски разпределен начин, близо до крайните устройства. Крайните устройства от своя страна прехвърлят изчислителните задачи към възлите в мъглата, вместо към облачната структура. Въпреки това използването на децентрализирани възли в мъглата за разтоварване води до значително забавяне, но също така и смекчава проблема с тесните места при комуникация с облачна структура. Съществуващо мрежово оборудване може да се използва от Fog computing (рутери или базови станции), като им предостави допълнителен изчислителен капацитет, така че да могат да действат като възли в мъглата. Възлите в мъглата може да се разглеждат като малки центрове за данни близо до потребителската мрежата, които обслужват крайни устройства в тях. Подобно на облачните структури, възлите в мъглата могат да използват виртуализация (контейнеризация), за да изпълняват приложения, принадлежащи на различни наематели по изолиран начин, като също така насърчават лесно управление на приложения, включително стартирането, автоматичното мащабиране и изключването им [14].

Абстрактният модел на Fog computing [15] показва, че някои стъпки за обработка на данни все още могат да бъдат прехвърлени в облака, особено тези за обработка на данни, които изискват много ресурси и не са критични по отношение на времето. Този модел включва три слоя: крайни устройства, възли в мъглата и облачни структури. Крайните устройства имат малък капацитет, но са огромен брой. Възлите в мъглата предлагат по-висок, но все пак ограничен капацитет и са по-малко на брой. Облачните структури предлагат голям капацитет, но техният брой е малък. Възлите в мъглата са на малко разстояние от крайните устройства и по този начин могат да бъдат достигнати по-бързо, което намалява латентността. Чрез комбиниране на тези три слоя, Fog Computing може ефективно да осигури критични за латентността приложения с необходимите ресурси за обработка.



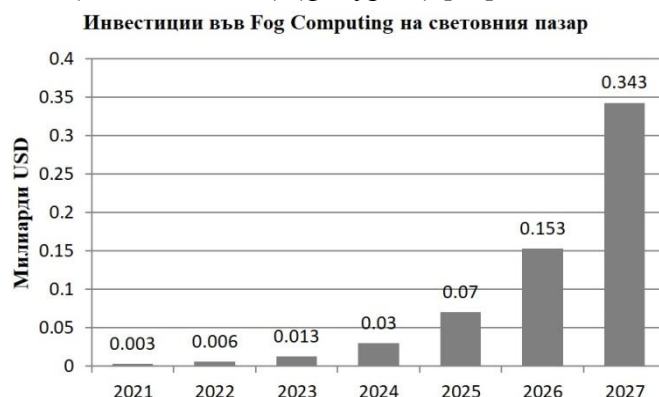
Фиг. 5. Абстрактен модел на Fog Computing

Ползите от модела Fog Computing се изразяват в [16]:

- Намаляване количеството на така наречените „големи данни“ до по-малко количество информация, като преобразува необработените данни в компактна форма на информация на ниво крайни устройства;
- Намаляване на данните, като филтрира ненужните. Проявява се чрез отчитане и събиране само на полезни данни и изхвърляне на допълнителните разточителни данни дори на ниво сензори;
- Осигуряване на по-малко забавяне поради разполагане на изчислителни възли в мъглата, което прави обработката по-лесна и по-бърза, и води до високоскоростна и по-добра реакция, което подобрява качеството на услугата (Quality of Service – QoS), предоставена на клиента;
- Увеличаване на достъпността, което се дължи на факта, че изчисленията в мъглата намаляват прекомерното разчитане на други изчислителни системи. По този начин се минимизират различни проблеми свързани със свързаността, тъй като възлите в мъглата осигуряват изпълнение на малки функции, като имат и изчислителна мощ;
- Предоставяне на по-висока сигурност, която може да бъде осигурена чрез криптиране на данните в самия източник с помощта на интелигентни крайни устройства, монтирани в сензорите, които предотвратяват изтичане или кражба на данни по време на предаването към облака. Освен това необработените данни се обработват на ниво периферни устройства, което създава по-малко проблеми със сигурността поради компактността на данните в информационните системи. Също така има по-малко заплахи за сигурността въз основа на комуникацията поради по-малкия брой препредавания;
- Повишаване качеството на услугата, тъй като крайните устройства са разположени близо до крайния потребител и могат да съберат повече информация за локалната настройка и изисквания. Тази функция спомага за разпознаване контекста на изискванията на потребителя и съответно генериране на заявката, като се има предвид контекста на задачата;
- Концепцията за мобилност може да се приложи за управление на голям брой устройства, свързани в Интернет с помощта само на няколко Gateway устройства.

Това би помогнало за покриване на голяма географска област. При тази техника Gateway устройствата в мъглата трябва да са предварително програмирани да се движат по предварително определен път, така че да покрят всички разпределени крайни устройства.

Статистиките и прогнозите за инвестициите в световния пазар на Fog Computing показват, че той заема своето място на пазара и се очаква през годините инвестициите да нараснат със средно 55.6% (милиони USD) (фигура 6) [17].



Фиг. 6. Инвестиции във Fog Computing

5. Разлики между Cloud, Edge и Fog Computing за Интернет на обектите

Централизираната облачна архитектура е изправена пред различни предизвикателства, които се отнасят до приложения за Интернет на обектите. Изчисленията в облачната структура не могат да отговорят на изискванията на чувствителни към времето приложения за Интернет на обектите, като предаване на видео поток, игри и добавена реалност. Тези предизвикателства може да се преодолеят с помощта на Edge и Fog Computing. Разликите между трите модела са обобщени в Таблица 1 [18]. Връзката между устройствата за Интернет на обектите се реализира от Edge и Fog Computing. Те представляват част от Cloud Computing като приближават облачната услуга по-близо до крайните устройства в мрежата [16].

Имайки предвид характеристиките на разглежданите изчислителни технологии се вижда, че Fog Computing е най-обещаващата за технологии за Интернет на обектите. При нея изчислителната мощ на възлите в мъглата е по-голяма, а това осигурява подобряване на ефективността. Това се гарантира от възможността, генерираните данни от устройствата за Интернет на обектите да се обработват и анализират в реално време. Работата на Fog Computing доближава възможностите на облачните мрежи, изчисления и възможност за съхранение до крайната мрежа, което решава проблема с обслужването в реално време на устройствата за Интернет на обектите и осигурява сигурни и ефективни приложения.

Различията между разглежданите изчислителни модели, които имат най-широко приложение при технологиите за Интернет на обектите влияят върху финансовите инвестиции при тях на световния пазар. От представената статистика и прогноза за инвестициите за разглежданите изчислителни модели на фигура 7 се вижда, че Cloud и Edge Computing имат постоянен и сигурен темп на инвестиционен растеж. Това показва, че тези модели имат широко приложение както сега, така и в бъдеще. От разглежданата статистика се вижда, че в сравнение с Cloud и Edge моделите, инвестициите във Fog Computing са много по-малко, но предпоставка за това е, че този модел е в началото на своето развитие. Освен това прогнозата показва, че с годините се очаква инвестициите в този модел да нараснат. Това показва, че се очаква ръст на използване и развитие на този изчислителен модел.



Фиг. 7. Инвестиции в разглежданите изчислителни модели

Таблица 1. Разлики между Cloud, Edge и Fog Computing

Характеристики	Cloud Computing	Edge Computing	Fog Computing
Изчислителен модел	Централизиран	Разпределен	Разпределен, както и централизиран
Капацитет за съхранение на данни	Много голям	Малък	Среден
Капацитет за кеширане	Много голям	Малък	Голям
Латентност	Висока	Много ниска	Ниска
Мобилност	Много трудна	Много лесна	Лесна
Изчислителен капацитет	Много голям	Малък	Голям
Размер на пространството за съхранение на данни	Много голям	Малък	Голям
Достъп	Глобален	Ограничен	Глобален, както и ограничен в зависимост от вида на приложението
Обработка на данните	Данните се прехвърлят и съхраняват в отдалечена система за съхранение, където се поддържат, управляват, архивират и предоставят на потребителите през мрежата	Данните се обработват на устройствата или самите сензори, без да се прехвърлят никъде	Данните се обработват във възел в мъглата или IoT gateway, който се намира в локалната мрежа
Информация за местоположението на данните	Без информация	Притежава такава информация	Притежава такава информация
Местоположение на сървърите	Сървърите са в интернет пространството	Сървърите са в крайната мрежа	Сървърите са в локалната мрежа, особено когато трябва да се обслужват услуги в реално време

6. Заключение

Тази статия представя трите основни изчислителни модела Cloud Computing, Edge Computing и Fog Computing, използвани при работата с технологии за Интернет на обектите. Представени са особеностите, архитектурите и ползите им. Представени са разликите между трите модела. Въз основа на описаните разлики се вижда, че най-подходящият модел за работа с технологии за Интернет на обектите е Fog Computing. Той осигурява голям изчислителен капацитет, голям размер на пространството за съхранение, ниска латентност, лесна мобилност и не натоварва крайните устройства с допълнителни изчисления. Тези характеристики на Fog Computing осигуряват среда с достатъчен изчислителен ресурс и мощност, които да обслужат трафика генериран от големия брой устройства за Интернет на обектите.

Благодарности

Научните изследвания, резултатите от които са представени в настоящата публикация, са извършени по проект НПЗ „Изследване на модели за управление на процеси с цел ефективен преход към икономика, базирана на 6G мрежи“ в рамките на присъщата на ТУ-Варна научноизследователска дейност, финансирана целево от държавния бюджет.

Литература

- [1]. Ericsson Mobility Report, June 2022, <https://www.ericsson.com/en/reports-and-papers/mobility-report>, последно посетен на 07.11.2022г.
- [2]. Kaushik, Shweta, and Charu Gandhi, Cloud Computing Technologies. //Cloud-Based Big Data Analytics in Vehicular Ad-Hoc Networks, pp: 233-253. Hershey, PA: IGI Global, 2021. <https://doi.org/10.4018/978-1-7998-2764-1.ch011>
- [3]. Malik, M., Wani, S., Rashid, A., CLOUD COMPUTING-TECHNOLOGIES. //International Journal of Advanced Research in Computer Science, Volume 9, No. 2, March-April 2018, pp: 379-384, ISSN: 0976-5697, DOI: <http://dx.doi.org/10.26483/ijarcs.v9i2.5760>
- [4]. VMWARE, What is Cloud Architecture?, <https://www.vmware.com/topics/glossary/content/cloud-architecture.html>, последно посетен на 07.11.2022г.
- [5]. Himani Tyagi and Rajendra Kumar, Cloud Computing for IoT. // Internet of Things (IoT): Concepts and Applications, 2020, pp: 25-41, ISBN: 978-3-030-37468-6, DOI: 10.1007/978-3-030-37468-6_2
- [6]. Cloud Computing Market Research Report, February 2020, <https://www.marketresearchfuture.com/reports/cloud-computing-market-1013>, последно посетен на 07.11.2022г.
- [7]. Jorge Pérez, Jessica Díaz, Javier Berrocal, Ramón López-Viana & Ángel González-Prieto Edge computing, Computing (2022), <https://doi.org/10.1007/s00607-022-01104-2>
- [8]. What Is Edge Computing?, <https://www.cisco.com/c/en/us/solutions/computing/what-is-edge-computing.html>, последно посетен на 07.11.2022г.
- [9]. X. Kong, Y. Wu, H. Wang and F. Xia, "Edge Computing for Internet of Everything: A Survey," in IEEE Internet of Things Journal, 2022, doi: 10.1109/JIOT.2022.3200431.
- [10]. Khaja Mannanuddin, M. Ranjith Kumar, Srinivas Aluvala, Y. Nagender, S. Vishali, Fundamental Perception of EDGE Computing. //IOP Conference Series: Materials Science and Engineering, Volume 981, International Conference on Recent Advancements in Engineering and Management (ICRAEM-2020) 9-10 October 2020, Warangal, India

- [11]. Swaroop S. Sonone, Kavita Saini, Swapnali Jadhav, Mahipal Singh Sankhla, Varad Nagar, Chapter Twelve - The future of edge computing, *Advances in Computers*. //Elsevier, Volume 127, 2022, pp: 333-358, ISSN 0065-2458, ISBN 9780128245064, <https://doi.org/10.1016/bs.adcom.2022.02.009>.
- [12]. Edge Computing Market Size, Trends, Share, Growth, Report 2030, <https://www.globenewswire.com/en/news-release/2022/07/07/2475942/0/en/Edge-Computing-Market-Size-to-Hit-at-USD-116-5-Billion-by-2030.html>, последно посетен на 07.11.2022г.
- [13]. IoT, from Cloud to Fog Computing, 2015, <https://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing>, последно посетен на 07.11.2022г.
- [14]. Bansal, S., Aggarwal, M., Aggarwal, H. Chapter 14 Advancements and Applications in Fog Computing. //Wiley Online Library, First published: 04 October 2019, <https://doi.org/10.1002/9781119593171.ch14>
- [15]. Mann, Z.Á. Notions of architecture in fog computing. *Computing* 103, 51–73 (2021). <https://doi.org/10.1007/s00607-020-00848-z>
- [16]. Atlam, Hany F., Robert J. Walters, and Gary B. Wills. 2018. "Fog Computing and the Internet of Things: A Review" *Big Data and Cognitive Computing* 2, no. 2: 10. <https://doi.org/10.3390/bdcc2020010>
- [17]. Fog Computing Market, November 2022, <https://www.marketresearchfuture.com/reports/fog-computing-market-2578>, последно посетен на 07.11.2022г.
- [18]. Mansoor Ahmad Rasheed, Jabar Saleem, Hudabia Murtaza, Hafiz Abdullah Tanweer, Mannan Ahmad Rasheed, Mishaal Ahmed, A Survey on Fog computing in IoT. // VFAST Transactions on Software Engineering <http://vfast.org/journals/index.php/VTSE@> 2021, Volume 9, Number 4, October-December 2021, pp: 68-81, ISSN(e): 2309-3978, ISSN(p): 2411-6246

За контакти:

гл. ас. д-р инж. Айдын Мехмед Хъкъ
кафедра „Компютърни науки и технологии”
Технически университет – Варна
E-mail: aydin.mehmed@tu-varna.bg



DATA ACQUISITION WITH OPENSOURCE COMEDI LIBRARIES AND MODULES WITH JAVA USING JAVA NATIVE INTERFACE UNDER LINUX OPERATING SYSTEM

Ivo Rakitin, Ventsislav Nikolov

Abstract: Data acquisition with COMEDI (**C**ontrol and **M**Easurement **D**evice **I**nterface) gives full control over a Data Acquisition (DAQ) device. The acquired data is read by C++ library with Java Native Interface (JNI) and displayed in a Java application. The article covers the topic of sending single instructions to the DAQ device using C/C++ code from Java program. This enables the possibility of linking high-level programming language like Java with a low-level DAQ device. This will give the Java programmer options to control DAQ devices from Java code. The acquired data can be further processed and displayed as needed via all means in Java.

Keywords: Data Acquisition, DAQ, JNI, COMEDI, Java, Linux

1. Introduction

Data acquisition is the process of sampling signals from the real life world representing physical conditions like temperature, speed, pressure, etc. and converting them into digital values. These values can be processed and manipulated on a computer or other digital device. Digital acquisition needs specific hardware and software for connecting and acquiring data. This article covers the linking of a high-level programming language like Java with low-level DAQ device using JNI calls [1] and COMEDI Lib [6]. COMEDI modules and COMEDI Lib (a set of libraries) are used in order to communicate with a DAQ device and acquire data from it. This approach gives flexibility over the drag and drop programming mainly used in OEM software accompanying DAQ boards.

2. COMEDI, COMEDI Lib and JNI

COMEDI consists of two parts: Kernel modules also called drivers and COMEDI Lib. Kernel modules control the device on low-level and libraries are on high-level. Many brands and devices are supported – National Instruments [7], Keithley Metrabyte [2], ADLink [3], Advantech [4], etc. The device hierarchy is as follows:

- **Channel:** The lowest level hardware component. It represents a single data channel – for example analog input or digital output.
- **Subdevice:** a set of functionally identical channels. For example – a set of 8 identical analog inputs.
- **Device:** a set of subdevices that are physically implemented on one interface card.

Some cards contain other specific components that are out of the aforementioned. For example EEPROM for storing configuration or calibration data. These special components are also classified as “subdevice”.

The high-level libraries (COMEDI Lib) contain interfaces to the devices, programming functions, structures, etc. The supported programming languages are C and C++. It is necessary to do some initializing before acquiring samples from DAQ device. For example:

```
comedi_range *ad_range;  
int i = 0, range_index = 0;  
it = comedi_open(DeviceName); //device name of the board in Linux - /dev/comedi0
```

```

comedi_set_read_subdevice(it, 0);
comedi_set_write_subdevice(it, 1);
maxdata = comedi_get_maxdata(it, subdev, 0); //Max value of a sample – 65535 is for 16 bit
maxdataint = maxdata;
int ranges_ai = comedi_get_n_ranges(it, subdev, i);
printf("Ranges %i ", ranges_ai);
for (range_index = 0; range_index < ranges_ai; range_index++)
{
    ad_range = comedi_get_range(it, subdev, i, range_index);
    printf(": range %i, min = %.4f, max = %.4f ", range_index, ad_range->min, ad_range->max);
}
comedi_set_global_oor_behavior(COMEDI_OOR_NUMBER);

```

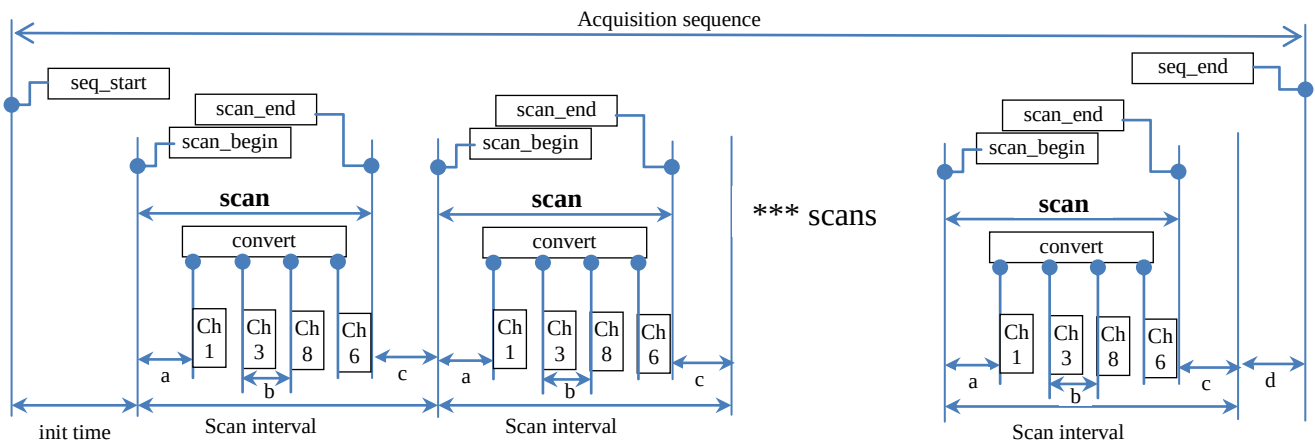


Fig. 1. – Asynchronous Acquisition Sequence
a. setup time; b. conversion interval; c. scan delay; d. termination time [8]

Functions used above are used for opening the device, getting maximum possible value for a sample and default ranges in volts. Reading from the device is done as follows:

```

comedi_range *ad_range;
comedi_insn daq_insn;
// configure the AI channel for read
daq_insn.subdev = subdev;
daq_insn.chanspec = CR_PACK(chan, range, AREF_GROUND);
daq_insn.insn = INSN_READ;
daq_insn.n = 1;
daq_insn.data = test_data;

retval = comedi_do_insn(it, &daq_insn); // send one instruction to the driver
if (retval < 0)
{
    comedi_perror("comedi_do_insn in getData()");
    return 0.0;
}
ad_range = comedi_get_range(it, subdev, chan, range);
return comedi_to_phys(test_data[0], ad_range, maxdata);

```

In order to do one single acquisition it is needed to send an instruction by defining `daq_insn` from data type `comedi_insn` (built-in the COMEDI Lib). Configuring the instruction for reading is necessary by assigning `INSN_READ` to the `insn` field. After sending instruction with `comedi_do_insn`, the data is available in `daq_insn.data`. Converting the acquired sample to volts is done by `comedi_to_phys` function.

All these functions can be linked with a Java program using Java Native Interface. In order to make this possible it is mandatory to `#include <jni.h>` and make `JNIEXPORT` for each `JNICALL` function.

```
JNIEXPORT jdouble JNICALL Java_ComediTest_getData(JNIEnv *env, jobject obj)
{
    double result = getData(0, 0);
    jdouble testdouble = result;
    return testdouble;
}

JNIEXPORT jdouble JNICALL Java_ComediTest_init(JNIEnv *env, jobject obj, jint debug)
{
    int d = debug;
    init(d);
    return 0;
}
```

JNI is a function interface framework that makes possible for the Java code to call foreign native libraries and applications which are written in languages like C++, C and others. These foreign programs are hardware and operating system specific. There are certain reasons to do this. For example:

- Communicating with a hardware via certain library or libraries;
- Performance benefits. Native functions can execute faster;
- External libraries that are not available under Java.
- Components that are needed in order to use JNI properly:
- Java code. This is our Java code. There must be at least one native method declared somewhere.
- Native code – This is the code written in C or C++. It must contain the logic which will be called.
- JNI .h file or header file – this includes the definitions of JNI that can be used in our native C or C++ code.
- GCC compiler or similar.

In the primary Java program there must be `System.loadLibrary(String path)` before calling already declared native methods. `System.loadLibrary` method tries to load the specified library into memory. All native declared methods must be available for call in the already loaded library. C/C++ code contains specific elements defined with `jni.h`. Some of which are:

- `JNIEXPORT` – functions that start with this definition are marked as exportable in the shared library and make them visible for JNI to call them.
- `JNICALL` – this is used with the combination of `JNIEXPORT`.
- `JNIEnv` – a structure which contains methods which helps our native code access Java elements.
- `JavaVM` – also a structure referencing the current running JVM. This structure gives the possibility of adding threads, start new JVM, destroy it, etc.

JNI can also communicate both ways – between the primary Java program and the target C++ library. JNI provides other ways to call functions from other programming languages.

When compiled as a shared object this C++ target program controlling the DAQ device can be loaded in Java and all functions are now available for execution (Fig. 2). Calling the external functions must be separated in different thread in order to make it possible for the graphic to refresh properly.

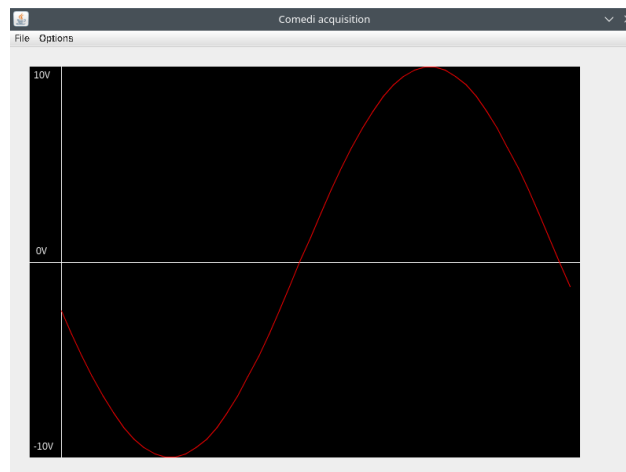


Fig. 2. – Java program displaying 1s of a signal in real time

3. Conclusions

This methodology gives the programmer flexibility over the control of the device and the acquired data. The different programming environments provided by the manufacturers of DAQ devices are overall low in performance and resource demanding, nevertheless providing an easy way to create DAQ software. The programming environments in most cases provide drag and drop operations so the created software is in most cases standalone which lacks the options to incorporate with other software or programming environments. Using COMEDI under Linux [5] on the other hand makes it quite versatile for the programmer however being more complex, to incorporate the control library written on C or C++ under other software packages.

References

- [1]. Liang, S. The Java native Interface. Programmer's Guide and Specification. ISBN: 0-201-32577-2 Addison-Wesley, 1999
- [2]. [https://download.tek.com/manual/24888A\(DAS4\).pdf](https://download.tek.com/manual/24888A(DAS4).pdf) (last accessed on 24.10.2022)
- [3]. <https://www.adlinktech.com/> (last accessed on 24.10.2022)
- [4]. <https://www.advantech.com> (last accessed on 24.10.2022)
- [5]. <https://www.gentoo.org/> (last accessed on 24.10.2022)
- [6]. <https://www.comedi.org/> (last accessed on 24.10.2022)
- [7]. <https://www.ni.com> (last accessed on 24.10.2022)
- [8]. <https://www.comedi.org/doc/index.html> (last accessed on 24.10.2022)

For contacts:

Assist. prof. Ivo Rakitin
Software and Internet Technologies
Technical University of Varna
E-mail: ivo.rakitin@tu-varna.bg

Assoc. prof. Ventsislav Nikolov
Computer Science and Engineering
Technical University of Varna
E-mail: v.nikolov@tu-varna.bg

АВТОМАТИЗАЦИЯ НА ПРОЦЕСА НА ПЛАЩАНЕ ПО БАНКОВ ПЪТ В ПРЕДПРИЯТИЯТА

Иван Л. Игнев

Резюме: В статията се представя метод за автоматизиране на плащанията по банков път и практическата му реализация в среда MS Excel. Описва се работата на софтуер, който минимизира използването на човешки ресурси при извършване на банкови плащания на доставчици и последващото им отразяване в счетоводните системи на предприятията.

Ключови думи: автоматизация, плащания, софтуер

AUTOMATION OF THE BANKING PAYMENT PROCESS IN COMPANIES

Ivan L. Igeev

Abstract: The article presents a method for automating bank payments and its practical implementation in an MS Excel environment. It describes the operation of software that minimizes the use of human resources when making bank payments to suppliers and their subsequent reflection in the accounting systems of the companies.

Keywords: automation, payments, software

1. Увод

Всяко предприятие извършва ежедневно плащания на доставчиците на необходимите за дейността му стоки и услуги. Процесът на плащане по банков път е трудоемък, защото от една страна се изисква попълването на платежни документи, а от друга въвеждане на тези документи в счетоводната база данни на предприятието. В повечето случаи с плащанията и осчетоводяването им се занимават различни служители.

В статията е представен анализ на процеса на плащане и проблемите при автоматизирането му.

Анализирани са предлаганите на пазара софтуерни решения за автоматизиране на плащанията.

Направен е обзор на начините за електронно подписване на документи и услугите за плащания предоставяни от банките.

Статията представя практическата реализация на метод за автоматизиране на плащанията по банков път и последващото им автоматизирано въвеждане в счетоводна база от данни с минимално човешко участие.

В частта си за автоматизиране на плащанията методът е универсален и е приложим към всеки счетоводен софтуер, отговарящ на изискванията на счетоводното законодателство на Република България.

В частта си за автоматизиране на счетоводните записвания методът е приложим за счетоводен софтуер, който има функционалност за импортиране на файл с данни, формирани по определен стандарт.

2. Изложение

2.1. Анализ на процеса на плащане по банков път

Плащанията на доставчици са ежедневие и необходимост за всяко предприятие. Осъществяването на банковите плащания отнема време и човешки ресурси заради следните трудоемки процеси:

Установяване на задължението за плащане

За установяване дължимата сума на конкретен доставчик се събират сумите по всички неплатени към момента фактури издадени от този доставчик, като трябва да се вземе предвид, че задължението по всяка фактура може да е с различен срок за плащане. От гледна точка на рационално използване на финансовите ресурси е необходимо да се плащат само тези фактури, по които е дошъл падежът (крайната дата за плащане).

Плащане

Най трудоемко е да се пишат хартиени платежни нареждания и да се носят в банката. Банките предоставят електронни услуги и възможности за създаване на шаблони за плащане, но попълването на тези електронни формуляри за всеки отделен доставчик отнема много време. Когато се плаща едновременно по няколко фактури от един доставчик е желателно да се изпишат номерата на тези фактури в платежния документ, за да може доставчикът, когато получи плащането, да разбере за кои фактури е платено. Доставчиците, които са големи фирми в бранша си, обикновено използват софтуер, който изисква съгласуване на всяко получено плащане на ниво конкретна фактура и изискват тази информация от контрагентите си.

Осчетоводяване на плащанията

В счетоводството на предприятията ежедневно се получават извлечения от банковите им сметки. Всяка сума от извлеченията се въвежда от счетоводител в счетоводната база от данни. В зависимост от софтуера, който се използва, може да се изисква и съгласуване на плащането с конкретна фактура.

Извод

Анализът на гореизброените процеси налага извода, че съществува необходимост от автоматизация на процеса на плащане по банков път с цел неговото ускоряване и минимизирането на използването на човешки ресурси.

2.2. Анализ на съществуващите решения на проблема

Големите предприятия използват интегрирани софтуерни решения, които обхващат всички процеси на тяхната дейност, включително и процеса на плащане. При тях проблемът с автоматизиране на плащанията в голяма степен е решен. Тези предприятия имат голям обем на дейността си, но са много малка част от фирмите в България.

Пример за такава фирма е Метро кеш енд кери България ЕООД. Освен че използва интегрирана бизнес система, Метро получава данни от доставчиците си по стандарта за електронен обмен на данни между бизнес системи (Electronic Data Interchange-EDI). Така получените данни автоматично се записват в базата от данни без човешка намеса.

Безкасовите плащания с банкови карти чрез POS (Point of Sales) терминали са много популярни и съответно обемът им е огромен. При тези плащания се генерират файлове с транзакции, които са подходящи за последваща обработка и импорт в счетоводна база данни [1].

В строителството автоматизацията на плащанията предполага цялостна система за контрол на съответния етап и плащанията, свързани с него. Технологията блокчейн (blockchain) е предпочитана за изграждане на такива системи [2, 3].

Подобни блокчейн системи се създават и за автоматизиране на процеса авансово плащане-доставка-доплащане [4].

Процесът на плащане на доставчиците в големите компании най-често е автоматизиран чрез софтуер за цялостно управление на бизнеса SAP (Systems Application and Products) [5].

Днес банките и операторите на различни платежни системи предлагат на своите клиенти автоматизирано плащане на битовите им сметки (ел. енергия, парно, вода, комуникационни услуги). Този процес е свързан с автоматизиране на разплащанията с доставчиците на тези услуги [6].

Друга област за автоматизация на плащанията е плащането на пътните такси за магистрала и пътища [7, 8].

В по-голямата си част фирмите от малкия и среден бизнес не могат да си позволят да използват такива интегрирани софтуерни решения, заради тяхната висока цена и скъпа поддръжка. Обикновено те закупуват софтуер според конкретните си нужди и финансови възможности.

Функционалности на счетоводния софтуер за малки и средни фирми

Доставчиците на счетоводния софтуер, наличен на пазара, предлагат различни функционалности по отношение на плащанията.

За целите на настоящата статия е направено проучване по отношение на предлаганите функционалности за плащания, като са систематизирани данните за едни от най-популярните софтуерни продукти на пазара в България.

В проучването не са разглеждани интегрирани бизнес системи, използвани от големите фирми, поради следните причини: 1) цената на внедряването е не по-малка от 10000 лева; 2) цената на годишната поддръжка е не по-малка от 5000 лева.

За целите на проучването са определени следните функционалности, които в тяхната съвкупност напълно могат да автоматизират процесите на плащания, а именно:

- Попълване на платежни документи;
- Заявки за плащане;
- Обобщаване в един документ на няколко фактури за плащания към доставчик;
- Генериране на файлове за импорт към системи за интернет банкиране;
- Осчетоводяване на група потвърдени преводни нареждания;
- Обвързване на фактура с плащане;
- Импорт на банкови извлечения;
- Самообучение.

Финансовите показатели, използвани по-долу в сравнителната таблица, означават:

- Минимална цена за базов софтуер – сума, дължима еднократно при покупка на софтуера за една фирма и едно работно място;
- Минимална годишна поддръжка – минималната сума дължима за една година за поддържане функционалността и актуализация на закупения софтуер;
- Цена на функционалност за плащания – цена на съответната функционалност, ако има изрично посочена такава.

Таблица 1. Сравнителна таблица на функционалностите за автоматизиране плащанията, предлагани от популярни софтуерни продукти

Софтуер	Описание	Функционалност за автоматизация на плащанията								Стойност на инвестицията в евро		
		Попълване на платежни документи	Заявки за плащане	Обобщаване в един документ на няколко фактури за плащания към доставчик	Генериране на файлове за импорт към системи за интернет банкиране	Осчетоводяване на група потвърдени преводни нареждания	Обвързване на фактура с плащане	Импорт на банкови извлечения	Самобучение	Минимална цена за базов софтуер	Минимална годишна поддръжка	Цена на функционалност за плащания
Ажур [8]	Система за управление на бизнеса с модули, използваща единна база данни	ДА	ДА	ДА	ДА	ДА	НЕ	НЕ	НЕ	249	79	79
Controlisy [10]	Софтуер за автоматизиране на счетоводната дейност	НЕ	НЕ	НЕ	НЕ	НЕ	ДА	ДА	ДА		468	
WorkFlow [11]	ERP система за малки и средни предприятия с предмет на дейност търговия, услуги или не сложно производство.	НЕ	НЕ	НЕ	НЕ	НЕ	НЕ	НЕ	НЕ	249		
Микроинвест Делта Про [12]	Счетоводен софтуер	ДА	НЕ	НЕ	НЕ	НЕ	НЕ	НЕ	НЕ	199		
McMaster [13]	Счетоводен софтуер	ДА	ДА	НЕ	ДА	ДА	ДА	ДА	НЕ	300	100	
Бизнес Навигатор [14]	Интегрирана система за фирмено управление	ДА	НЕ	НЕ	НЕ	НЕ	НЕ	НЕ	НЕ	549	169	
Булмар Офис [15]	Счетоводен софтуер	ДА	НЕ	НЕ	НЕ	ДА	НЕ	ДА	НЕ		230	
Плюс минус [16]	ERP система	НЕ	НЕ	НЕ	ДА	НЕ	ДА	ДА	НЕ		120	
kik-info.com [17]	Платформа за данъци и счетоводство	ДА	НЕ	НЕ	НЕ	НЕ	НЕ	НЕ	НЕ		75	
documents-online.net [18]	Сайт за попълване на платежни документи	ДА	НЕ	НЕ	НЕ	НЕ	НЕ	НЕ	НЕ			
e-docs.bg [19]	Сайт за попълване на платежни документи	ДА	НЕ	НЕ	НЕ	НЕ	НЕ	НЕ	НЕ			

Най-масовата функционалност е автоматичното попълване и последващо отпечатване на платежни документи. Спестява се времето за писане, но не и времето за подписване и представяне на разпечатаните документи в банката. Предлага се от всички популярни счетоводни програми, а също така е реализирана и в най-посещаваните сайтове в сферата на финансите и счетоводството.

Друга функционалност е генерирането на файл за масови плащания по утвърден от съответната банка стандарт. Тази функционалност изисква актуализация на софтуера и допълнително заплащане. Предлага се от доставчиците на интегрирани софтуерни решения, които създават различни модули на счетоводния си софтуер.

При автоматизация на счетоводните операции за плащания по банков път, източник на информация са банковите извлечения от конкретна банкова сметка.

Задължителни предпоставки за автоматизацията на счетоводните записвания, свързани с плащания по банков път на база информацията от банкови извлечения, са: 1) наличието на възможност информацията от всяко банково извлечение да бъде записана във файл в някой от популярните формати (txt, csv, xls); 2) банковата сметка на всеки доставчик да е обвързана със съответния му номер от счетоводната база от данни.

Докато генерирането на файлове с банкови извлечения е вече стандартна функционалност на сайтовете за интернет услуги на банките, то при появата на всяка нова банкова сметка в банково извлечение е необходима намесата на оператор за обвързването ѝ с номера на конкретния доставчик.

Решенията за автоматизацията на счетоводните записвания, основани на файлове с банкови извлечения, не са масови и се предлагат индивидуално според конкретните нужди на клиента. Обикновено се реализират като функционалност на конкретен счетоводен софтуер и задължително изискват намесата на квалифициран служител за съпоставяне на банковите сметки с номерата на контрагентите от счетоводната номенклатура.

От анализа на данните от проучването могат да се направят следните изводи:

Добавянето на функционалност за пълно или частично автоматизиране на плащанията към вече използвания от предприятието софтуер е свързано или със смяна на софтуера с такъв, който има такава функционалност, или използването на отделен софтуер за автоматизация на плащанията, подобен на CONTROLISY. В първия случай е необходимо време за преход към нов софтуер и финансов ресурс за покупката на базов софтуер и годишна поддръжка. Във втория случай - финансов ресурс за годишна поддръжка на функционалността и обучение на персонала за работа със софтуера за автоматизация.

Услуги за плащания предоставяни от банките

Към настоящия момент електронните услуги, предоставяни от банките, свързани с плащания и контрол на банковите сметки, са ежедневие.

На ниво фирми банките предлагат следните стандартни услуги за плащания:

- Извършване на плащания през сайта на банката;
- Създаване на шаблони за плащане;
- Предоставяне на софтуер за генериране на файл с информация за плащания за последващ импорт и плащане чрез сайта на банката;
- Предоставяне на възможност за импорт и плащане чрез сайта на банката, на файл с информация за плащания, създаден от софтуер на клиента по стандарт, определен от банката,

Удостоверителни услуги за електронно подписване

От момента на публикуването на Закона за електронния документ и електронния подпис до настоящия момент удостоверителните услуги станаха изключително популярни и се използват масово и от юридически, и от физически лица. Електронното подписване и електронното удостоверяване на подписа е важен елемент от процеса на плащане по електронен път и ускорява значително процеса на плащане. Към настоящия момент в България съществуват голям брой доставчици на удостоверителни услуги с различна степен на приложимост. С най-универсално приложение е квалифицирания електронен подпис (КЕП), издаван от специализирани доставчици на удостоверителни услуги.

Всяка банка предлага своя услуга за електронно подписване на плащанията, извършвани чрез нея. Затова при автоматизиране на процеса на плащане трябва да се използват електронните услуги на всяка отделна банка, чрез която ще се извършват плащания.

Възможности за последваща обработка на плащанията

След осъществяване на плащанията възниква необходимостта от тяхното отразяване в счетоводната база от данни. Масово това се извършва от счетоводител, който на база разпечатано извлечение от банкова сметка въвежда счетоводни операции за плащанията.

За да е възможно автоматизацията на този процес, са необходими следните предпоставки:

- Възможност за получаване на банковото извлечение в стандартен файлов формат, годен за обработка (txt, csv, xls).
- Налична функционалност в счетоводния софтуер за импорт на файл със счетоводни операции.
- Софтуер, който да преобразува електронното банково извлечение във файл за импорт на счетоводни операции.

Процесът на последваща обработка на плащанията е много труден за реализация, поради необходимостта от идентификация на доставчика по неговата банкова сметка (IBAN). Един и същ доставчик може да има няколко банкови сметки, по които да се извършват плащания, но е записан под един номер в номенклатурата на счетоводната база от данни. Логическото обвързване чрез софтуер на банковите сметки със счетоводния номер може да се извърши само от квалифициран оператор.

Изводи

От извършените анализи на процеса на плащане и възможностите за неговото автоматизиране може да се направи извода, че към настоящия момент не се предлага универсален софтуер, който да автоматизира процеса на плащанията, който да е приложим за всеки съществуващ счетоводен софтуер, да се настройва според нуждите на потребителя и да е на ниска цена.

Такъв софтуер е необходим на фирмите от малкия и средния бизнес, защото с автоматизиране на процеса на плащане се спестява много време и човешки ресурси.

2.3.Метод за автоматизиране на плащанията

Основавайки се на анализите и изводите посочени в т.2.2 се стигна до разбирането, че разработката на софтуер за автоматизиране на процеса на плащане трябва да се основава на следните принципи:

1. Създава се и се актуализира база от данни за фирмите платци, която да съдържа следната информация за всяка фирма: счетоводен номер на фирма, име на фирма, Булстат, пет банкови сметки, пет счетоводни сметки, съответстващи на банковите сметки.

2. Като основни данни за бъдещите плащания да се използва файл във формат txt, csv или xls, генериран от счетоводния софтуер, който да съдържа списък с неплатените фактури към определена дата и за всяка фактура да е представена минимум следната информация: номер на фирма платец, име на фирма платец, номер на търговски обект на фирмата платец, номер на фактура, дата на фактура, номер на доставчик, име на доставчик, обща стойност на фактурата.

3. След генерирането на всеки файл по т. 2 да се създава/актуализира база от данни за доставчиците, като за всеки отделен доставчик, определен от номера и името му, да се въведе Булстат, срок за плащане на фактурите в дни, до пет банкови сметки, поредния номер на банковата сметка, по която ще се плаща.

4. Падежът на всяка фактура да се изчислява като към датата на фактурата се добавя срокът за плащане на доставчика. За всяка фактура да се изчисляват броя дни до или след падежа, като разлика между датата на падежа и текущата дата (положително число в падеж, отрицателно число в просрочие).

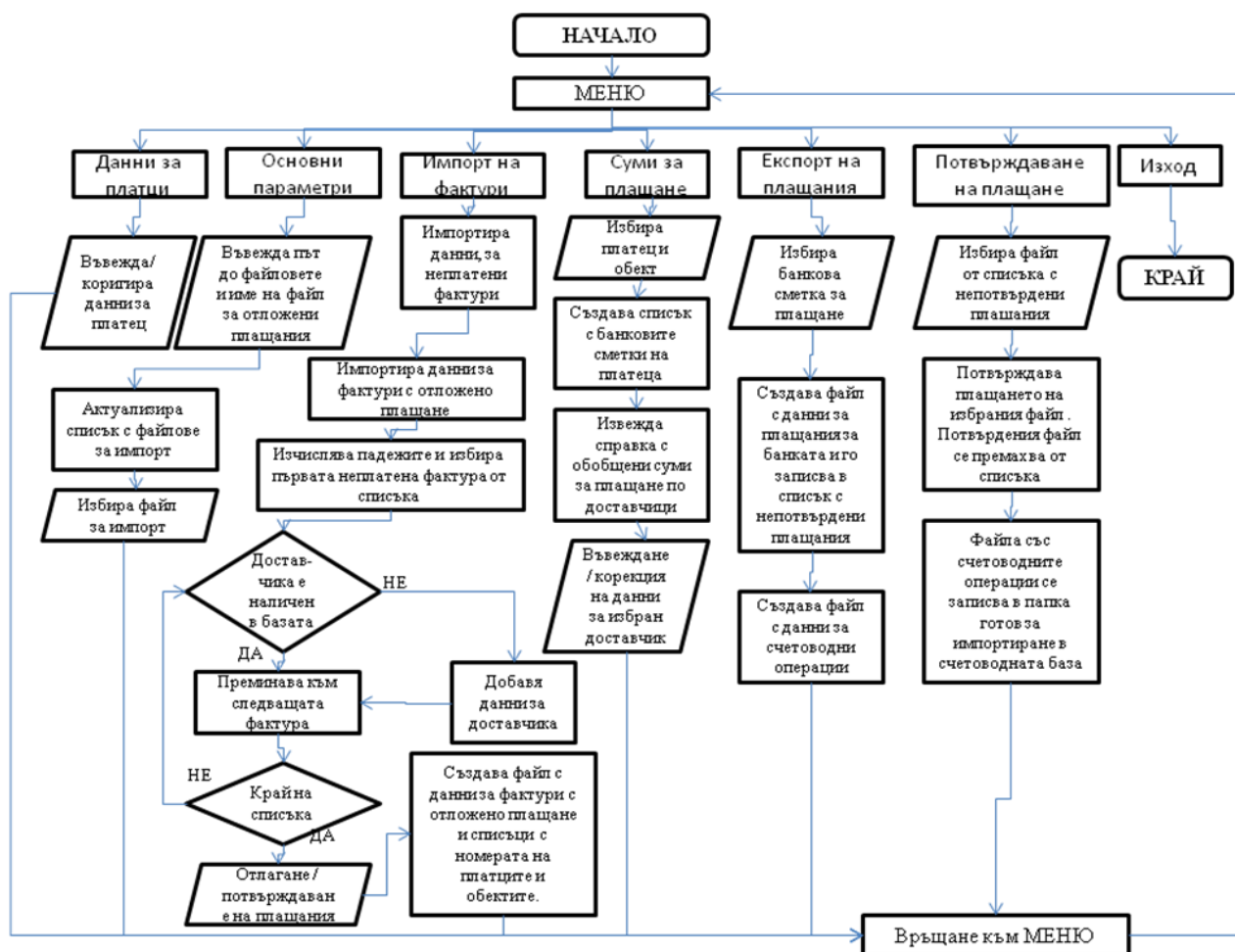
5. Всяка фактура трябва да може да бъде изключена или включена в списъка за плащания по т. 2, като изключените фактури да се съхраняват в отделен файл и да се добавят към всеки генериран файл по т. 2.

6. За избрани фирма и търговски обект, списъкът с неплатените фактури да се използва за създаване на файл, който да съдържа данни за общата сума, дължима на всеки отделен доставчик, основание за плащане, съдържащо номерата на неплатените фактури, счетоводен номер и банковата сметка на доставчика.

7. За избрана банкова сметка на фирма платец да се създава файл с данни за плащания, като се използват данните от файла по т. 6 и стандарта за файл предоставен от съответната банка.

8. Данните от файла по т. 6 и номерът на счетоводната сметка, съответстващ на избраната банкова сметка по т.7, да се използват за генериране файл с данни за счетоводни операции за импорт в счетоводната база от данни, изготвен по стандарт на доставчика на счетоводния софтуер, ако софтуерът предоставя такава функционалност.

Въз основа на тези принципи е дефинирана следната архитектура (фигура 1), реализираща предложения метод за автоматизиране на плащанията.



Фиг. 1. Алгоритъм за автоматизиране на плащанията

Изложеният метод има следните предимства:

- Използва данни за плащания от счетоводната база от данни;
- Използва номерата на доставчиците от счетоводната номенклатура;
- Обединяване на плащанията от няколко фактури от един и същ доставчик;
- Посочване номерата на фактурите в текста за плащане;
- Генериране на файл за импорт на счетоводни записи;
- Обвързване на всяка фактура с конкретно плащане;
- Възможност за настройка към всяка счетоводна система.

2.4. Реализация и начин на работа със софтуера за автоматизиране на банковите плащания на доставчици

Използвайки описаните основни принципи в т. 2.3, е създаден софтуер за автоматизиране на плащанията по банков път на доставчици.

За среда на разработка е избран MS Excel. Изборът се основава на:

- огромната популярност на електронната таблица на Microsoft;
- простотата и удобството за работа;
- липсата на необходимост от обучение;
- богатите възможности за манипулиране на данни, предоставяни от макро езика вграден в MS Excel.

За тестване на функционалностите на софтуера е избран счетоводен софтуер Масопому, който отговаря на изискването за наличие на възможност за импорт на файл със счетоводни операции. Тестовите са реализирани в счетоводството на ресторантите от веригата „Хепи“.

Софтуерът реализира пълен цикъл на автоматизация на плащанията, а именно:

- Определя задълженията към всеки доставчик на база неплатените фактури и падежа им.
- Генерира файлове за масови плащания на доставчици, използвайки стандартите на три банки, като без проблем може да се адаптира към стандарта на всяка банка.
- Генерира файл със счетоводни операции за импорт в счетоводната база от данни, като съгласува всяка отделна фактура с конкретно плащане.

База от данни за фирми платци

Комп. No.	Компания Име	Булстат	IBAN1	IBAN2	IBAN3	IBAN4	IBAN5	Сч.см.1	Сч.см.2	Сч.см.3	Сч.см.4
12	ХЕПИ ЛЕЙДИ ЕООД	103279225	BG41BNPA94401050269810	BG05UBBS80021087866420	BG50UNCR70001524289277			503100	503700	503400	
13	ХЕПИ 2000 ЕООД	130125231	BG15BNPA94401050363010	BG85BNPA94401050363011	BG14UNCR70001519877395	BG43UBBS80021008158120		503100	503102	503400	503700
14	ХЕПИ 2 ЕООД	103254297	BG16BNPA94401050507910	BG72UNCR70001519883238				503100	503401		
15	ХЕПИНС ЕООД	103279218	BG43UNCR70001522351925	BG47UBBS80021017169930				503400	503700		
16	ХЕНД ЕООД	103597021	BG95BNPA94401050499310	BG41BNPA94401050499312	BG68BNPA94401050499311	BG36UBBS80021001011110		503100	503102		503702
17	МАЙ ЕООД	103597014	BG39UNCR75271046682212	BG55UBBS80021050591330				503402	503700		

Фиг. 2. Изглед на база от данни за фирми платци

Данните за всяка отделна фирма, която извършва плащания, се съхраняват в отделна база от данни, където се записва номерът, името, Булстат на фирмата, до 4 банкови сметки и съответстващите им счетоводни сметки.

Достъпът за корекция на данни е с клавишна комбинация Ctrl+K.

Файл с данни за плащания

Файлът с данни за плащанията е във формат xls и генерира се от счетоводния софтуер и изглежда по следния начин (фигура 3):

Компани ния No.	Име на Компания	Име на Доставчик	Доставчик No.	Отдел	Фактура No.	Дата на Фактурата	Описание на Запис	Салдо в Базова Валута
13	ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ ЕТ	12933	124	483959	14.09.2022	Към поръчка 13501468	236.4
13	ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ ЕТ	12933	124	484040	15.09.2022	Към поръчка 13501479	2 677.97
22	ИНДАСТРИЪЛ КЕЙТЫ	ВАКЛИНОВ ЕООД	12184	123	3000098132	15.09.2022	Към поръчка 22445111	1 213.08
22	ИНДАСТРИЪЛ КЕЙТЫ	ВАКЛИНОВ ЕООД	12184	123	3000098171	16.09.2022	Към поръчка 22445120	191.52
25	АРЕА ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ ЕТ	12933	253	484689	19.09.2022	Към поръчка 25450258	5 683.72
25	АРЕА ЕООД	ВАКЛИНОВ ЕООД	12184	253	4000137125	20.09.2022	Към поръчка 25450265	836.22
25	АРЕА ЕООД	ВАКЛИНОВ ЕООД	12184	253	4000137129	21.09.2022	Към поръчка 25450264	2 970.98

Фиг. 3. Изглед на файла с данни за плащания

Изглед на основното меню

Път до файловете: S:\Common\dostavki_pl\

Име на входния файл: zavedenia_2022_09_29.xls

Файл за отложени плащания: otlogeni.xls

Файл за добавяне:

Име на Компания: ХЕПИ 2000 ЕООД

Булстат: 130125231

Импортирай фактури

Суми за плащане

Файл за УникрБул

Файл за УникрЕД2

Файл за BNP Pariba

Файл за ОББ

Други номера

ИЗХОД

Компания

Отдел

IBAN

13

124

BG15BNPA94401050363010

22

BG85BNPA94401050363011

23

BG14LNCR70001519877395

25

BG43UBBS80021008158120

39

Няма IBAN

45

57

58

59

Опресни

zavedenia_2022_06_15.xls

zavedenia_2022_06_22.xls

zavedenia_2022_06_29.xls

zavedenia_2022_07_06.xls

zavedenia_2022_07_13.xls

zavedenia_2022_07_20.xls

zavedenia_2022_07_28.xls

zavedenia_2022_08_04.xls

zavedenia_2022_08_10.xls

zavedenia_2022_08_17.xls

zavedenia_2022_08_24.xls

zavedenia_2022_09_01.xls

zavedenia_2022_09_07.xls

zavedenia_2022_09_14.xls

zavedenia_2022_09_21.xls

zavedenia_2022_09_29.xls

zita_2016_05_11.xls

zita_maket-14.01.2020.xls

zita_maket-20200219.xls

zita_maket-20200226.xls

zita_maket-20200310.xls

Потвърди

Отвори

Име на файл за банка:

Край на списъка

Избери файл за добавяне

Добави файл

ИЗТРИЙ

Видео

Видео в просрочие

Отложени

Отложени в просрочие

Справочник на клавишните комбинации

Ctrl+M-Параметри

Ctrl+S-Суми за плащане

Ctrl+B-Филтър на фактури по комп.и отдел

Ctrl+F-Разбивка по фактури

Ctrl+Q-Корекции данни за доставчици

Ctrl+N-Корекции данни за компании

Иван Игнев и Ваня Желязкова

Ви пожелават приятна работа!

Фиг.4. Изглед на основното меню

Задаване на основни параметри

С основните параметри (в жълто) се задава работната папка, името на файла с данни за плащания, името на файла с данни за отложени плащания.

Импорт на данни за плащания

След натискане на бутона „Опресни“, се актуализира списъкът с файлове с данни за плащания и от списъка се избира файл за импорт, чието име автоматично се изписва в полето име на входния файл. След това се натиска бутонът „Импортирай фактури“ за импортиране на данните от файла.

Процесът на импортиране протича в следната последователност:

- импортират се данните от избрания файл,
- извършва се проверка дали номерата и имената на доставчиците са налични в съществуващата база данни на доставчиците и ако има нови, се добавят.

Към вече импортираните данни се добавят данните от специален файл за отложени плащания. Изчислява се падежът на всяка фактура спрямо текущата дата и заложения срок за плащане за доставчик, като се отбелязва броят дни до датата и след датата на падежа.

Извеждат се посочените по-долу данни за плащания за оглед и корекция от оператор.

Комп. Име на Компания	Име на Доставчик	Доставчи	Отдел	Фактура No.	Дата на Факт	Описание на Запис	Салдо в Базе	Отложено	Срок за плащане (дни)	Падеж	Просрочие (дни)	29.09.2022
13 ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	483959	14.09.2022	Към поръчка 13501468	236.40		20	04.10.2022	(5)	
13 ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	484040	15.09.2022	Към поръчка 13501479	2 677.97		20	05.10.2022	(6)	
13 ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	484266	16.09.2022	Към поръчка 13501497	1 764.50		20	06.10.2022	(7)	
13 ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	484321	16.09.2022	КЛЕЧКИ ЗА СУШИ	252.00		20	06.10.2022	(7)	
13 ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	484441	17.09.2022	Към поръчка 13501519	3 572.82		20	07.10.2022	(8)	
13 ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	484535	19.09.2022	Към поръчка 13501523	548.10		20	09.10.2022	(10)	
13 ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	484643	19.09.2022	Към поръчка 13501532	879.40		20	09.10.2022	(10)	
13 ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	484535	19.09.2022	Към поръчка 13501523	548.10	1	20	09.10.2022	(10)	
13 ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	484643	19.09.2022	Към поръчка 13501532	879.40	1	20	09.10.2022	(10)	
13 ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	484839	20.09.2022	Към поръчка 13501547	488.10		20	10.10.2022	(11)	
13 ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	484839	20.09.2022	Към поръчка 13501547	488.10	1	20	10.10.2022	(11)	
13 ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	485020	21.09.2022	Към поръчка 13501568	349.02		20	11.10.2022	(12)	
13 ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	485020	21.09.2022	Към поръчка 13501568	349.02	1	20	11.10.2022	(12)	
13 ХЕПИ 2000 ЕООД	БАЛДАРАН СПРИНГ АД	117665	124	1000151592	14.09.2022	Към поръчка 13501470	254.16		14	28.09.2022	1	

Фиг.5. Импортирани фактури за плащане с добавени отложени плащания и изчислен падеж

Операторът оглежда плащанията и ако иска да отложи някое, го маркира с единица (1), съответно премахва единицата, ако иска да плати отложеното плащане.

С клавишна комбинация Ctrl+M операторът се връща към основното меню, като преди това автоматично се създава нов файл с отложени плащания съобразно корекциите на оператора.

Файлът с отложените плащания съдържа само маркираните с единица фактури и изглежда по следния начин (фигура 6):

Комп. No.	Име на Компания	Име на Доставчик	Доставчик No.	Отдел	Фактура No.	Дата на Фактурата	Описание на Запис	Салдо в Базова Валута	Отложено
13	ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	484535	19.09.2022	Към поръчка 13501523	548.10	1
13	ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	484643	19.09.2022	Към поръчка 13501532	879.40	1
13	ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	484839	20.09.2022	Към поръчка 13501547	488.10	1
13	ХЕПИ 2000 ЕООД	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ	12933	124	485020	21.09.2022	Към поръчка 13501568	349.02	1
13	ХЕПИ 2000 ЕООД	БАЛДАРАН СПРИНГ АД	117665	124	1000151761	19.09.2022	Към поръчка 13501545	215.64	1
13	ХЕПИ 2000 ЕООД	ВАКЛИНОВ ЕООД	12184	124	3000098241	20.09.2022	Към поръчка 13501548	1 958.88	1

Фиг. 6. Файл с данни за отложени плащания

База от данни доставчици

Доставчик No.	Доставчик Име	Булстат	Активен IBAN	IBAN1	IBAN2	IBAN3	IBAN4	IBAN5	Период за плащане в календар ни дни	Брой цифри от номера на фактурата
115756	ТИС ГРУПС ЕООД	204188612	1	BG06UNCR70001522689666					14	
115763	АГРО ТЪРГОВСКА КОМПАНИЯ ЕООД	103173510	1	BG15UNCR76301010BGTKLV					14	
115764	ЦДС ГРУП - ВАРНА ООД	204144893	1	BG92RZBB91551009503121					14	
115771	ПРОФАРМ ГРУП ООД	118556710	1	BG76CECB97901049393800					14	4
115781	ВЕЛАНОР ГРУП АД	202370406	1	BG84UNCR70001521047798					14	
115783	КУУЛ СПИРИТ ООД	201036994	1	BG89UNCR70001522564788					14	
115791	ФРОННЕРИ БЪЛГАРИЯ ЕООД	204185210	1	BG82INGB91451002559812					14	4

Фиг. 7. База от данни доставчици

Съхранява данни за всеки отделен доставчик, Булстата му, номерата на 5 банкови сметки с възможност за посочване на тази, която ще се използва за плащане, срок за плащане по фактурите. Последната колона показва колко цифри от номера на фактурата ще се изписват в описанието на плащането. Използва се, когато се плащат няколко фактури, а броят символи е ограничен.

Обобщаване на плащанията

Процесът на обобщаване на плащанията минава през задължителния избор на номер на фирма и номер на обект.

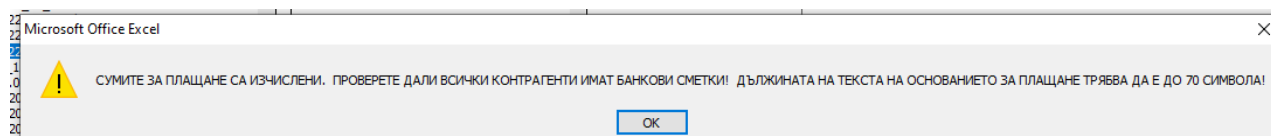
Компания	Отдел	IBAN
13	124	BG15BNPA94401050363010
22		BG85BNPA94401050363011
23		BG14UNCR70001519877395
25		BG43UBBS80021008158120
39		Няма IBAN
45		
57		
58		
59		

Име на Компания: ХЕПИ 2000 ЕООД
Булстат: 130125231

Фиг. 8. Избор на фирма и обект от менюто

Изборът се извършва, като се посочат номерата от първите два най-леви списъка, генерирани автоматично при импорт на данните. При посочване с мишката на номер на компания името и Булстат се изписват в полетата със син фон. Съответно в третия списък се появяват номерата на банковите сметки на фирмата.

Следващата стъпка е натискане на бутон „Суми за плащане“, който стартира процедура по обединяване на плащанията за избрана фирма, обект и доставчик, като извършва проверки за наличие на банкова сметка у доставчика и дали текстът на описанието на плащането съответства на максималния брой символи 70.



Появява се съобщение за завършено изчисление и предупреждение за максималния брой символи на текста за всяко плащане. Резултатът от изчислението изглежда по следния начин (фигура 9):

Компания	13 Обект	124 Дата	29.09.2022	14:00:40			
No.	Булстат	Доставчик	Сума	Банкова сметка	Банков код	Основание за плащане	Бр.символи (max 70)
12933	831314284	АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ ЕТ	10 768.31	BG62STSA93000021669647	STABGSGF	0959-040-266-321-441-535-643-839-020-C124	41
117665	200044397	БАЛДАРАН СПРИНГ АД	734.04	BG09UNCR70001520992112	UNCRBGSF	051592-51685-51761-C124	23
12184	122047217	ВАКЛИНОВ ЕООД	3 275.75	BG78UNCR70001515039241	UNCRBGSF	098134-98163-98184-98241-C124	29
117367	130992460	ГЕНЧЕВ 2002 ООД	486.00	BG31FINV91501003738011	FINVBGSF	0200000692-2000000701-2000000705-2000000716-2000000726-2000000738-2000000747-C124	82
110941	201741755	ГРИЙН МАРК ЛОГИСТИКА ООД	30 348.14	BG34BUIN95611000448876	BUINBGSF	04706-4707-4738-4758-4783-4796-4805-4831-4834-4845-4858-4876-4895-4898-4912-4928-4962-4970-4977-4995-4999-5002-C124	115
119766	202089010	ЕЛЕЯ ЕООД	1 084.37	BG72UNCR70001521095932	UNCRBGSF	022641-22666-22694-C124	23
16224	130996470	ИНГРЕДИА ООД	1 406.88	BG64PRCB92301013439819	PRCBGSGF	01000026692-1000026738-C124	27
10719	131032463	КОКА-КОЛА ХБК АД	914.08	BG57CTI92501000108677	CTIBGSGF	008092-14885-C124	17
18502	200597800	КЮ ЕНД И - БРЕД КЪМПАНИ ООД	521.40	BG42STSA93000027084159	STABGSGF	01101039148-1101039345-1101039399-1101039544-1101039659-C124	60
110474	200583626	МОРСКИ ДАР 09 ЕООД	385.00	BG02STSA93000025301922	STABGSGF	010645-10907-11035-C124	23
113166	121680279	ПЕТЛИТЕ ЕООД	307.40	BG56UNCR96601014010027	UNCRBGSF	02533-2566-2642-2824-C124	25
10514	125041015	ПИЛКО ЕООД	2 052.02	BG52RZBB91551088811914	RZBBGSGF	09090-9388-9583-9902-0108-0380-C124	35
110413	148127720	САЛЛА ЕСТЕЙ ЕООД (стар ЖВ СТРОЙ ЕООД)	120.00	BG20UBBS80021085874740	UBBSBGSF	017205-C124	11
14435	040479640	СОФСТОК ЛТД ООД	455.04	BG20UNCR76301077692578	UNCRBGSF	094292-C124	11
11102	121740930	ТОП ДРИНКС ООД	3 260.81	BG58PRCB92301050643501	PRCBGSGF	0133-002-404-783-168-477-847-191-C124	37
115791	204185210	ФОНЕРИ БЪЛГАРИЯ ЕООД	204.12	BG82INGB91451002559812	INGBBGSGF	00570-0630-C124	15
116138	204349916	ХЕПИ ФРЕШ ФУУД ООД	8 868.81	BG78UNCR70001522791190	UNCRBGSF	0320-324-347-379-389-401-418-429-436-446-467-479-496-508-538-541-559-C124	73
ОБЩО:			65 192.17				

Фиг. 9. Резултат от обобщаване на плащанията

Всяко описание на плащане има наставка с номера на обекта (С124), като наставката се избира от клиента.

При позициониране на курсора върху ред с данни и натискане на клавишната комбинация Ctrl+F, може да се видят кои фактури участват в съответното плащане.

Име на Компания	Отдел	Име на Доставчик	Фактура No.	Дата на Фактурата	Описание на Запис	Салдо в Базова Валута
ХЕПИ 2000 ЕООД	124	БАЛДАРАН СПРИНГ АД	1000151592	14.09.2022	Към поръчка 13501470	254.16
ХЕПИ 2000 ЕООД	124	БАЛДАРАН СПРИНГ АД	1000151685	16.09.2022	Към поръчка 13501509	264.24
ХЕПИ 2000 ЕООД	124	БАЛДАРАН СПРИНГ АД	1000151761	19.09.2022	Към поръчка 13501545	215.64
ОБЩО:						734.04

Фиг. 10. Детайлна справка за фактурите включени в обобщено плащане

Резултатът от обобщаването на плащанията служи като база за генериране на файл с плащания за съответната банка.

Създаване на файл за банка

Създаването на файл за банка задължително минава през избор на банкова сметка и стартиране на процедура за съответната банка.



Фиг. 11. Избор на банкова сметка и генериране на файл за банка

След натискане на бутон за генериране на файл за банка се проверява дали избраната сметка съответства на избраната банка. При коректен избор (в случая BNP Pariba) се генерира файл за банката и файл със счетоводни операции за плащанията. В списъка с файлове с данни за плащания се появява нова позиция с име съставено от номерата на компанията, обекта и текущата дата и час и място откъдето е генериран.

За краткост по-долу в текста са представени файлове с данни за плащания само към един доставчик.

Файлът за банка BNP Pariba представлява текстови файл със следното съдържание (фигура 12):

```
{1F01BNPAFRPPAXXX0000000000}{21101BNPABGSXAXXXN0000}{4:
:20:
:50H:/BG15BNPA94401050363010
ХЕПИ 2000 ЕООД
//BG
:52A:BNPABGSX
:30:220929
:21:
:23E:OTHR/DMST
:32B:BGN10768.31
:57D://STSA9300
STSA
//BG
:59:BG62STSA93000021669647
АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ ЕТ
//BG
:70:Ф959-040-266-321-441-535-643-839-02
0-C124
:71A:$HA
-}{5}
```

Фиг. 12. Файл с данни плащания по стандарт на BNP Pariba

Файлът се импортира чрез сайта на банката, плащането се удостоверява с електронно подписване и едва след потвърждение от банката за осъществяване на плащането файлът със счетоводните операции може да се импортира от счетоводител. За целта се избира файлът от списъка, натиска се бутонът „Потвърди”, след което файлът се изтрива от списъка и автоматично се записва на място известно на счетоводителя за последващ импорт в счетоводния софтуер. Файлът за импорт на счетоводни операции изглежда по следния начин (фигура 13):

JOURNALFORMAT	JournalNumber	TransactionNumberSeries	CompanyNumber							
GENERALJOURNALFORMAT	JournalNumber	EntryDate	EntryText	TypeOfEntry	AccountNumber	DebitBase	CreditBase	LocationName	SpecificationName	ReconcileVendorInvoiceNumber
JOURNAL	#ЗАПАЗИ Банкови операции	13								
GENERALJOURNAL	#ЗАПАЗИ 29.09.2022		Ф959-040-266-321-441-535-643-839-020-C124	Сметка	503100	0		10768.31 124	12933	
GENERALJOURNAL	#ЗАПАЗИ 29.09.2022	Ф483959	Доставчик	12933	236.4	0	124	12933	483959	
GENERALJOURNAL	#ЗАПАЗИ 29.09.2022	Ф484040	Доставчик	12933	2677.97	0	124	12933	484040	
GENERALJOURNAL	#ЗАПАЗИ 29.09.2022	Ф484266	Доставчик	12933	1764.5	0	124	12933	484266	
GENERALJOURNAL	#ЗАПАЗИ 29.09.2022	Ф484321	Доставчик	12933	252	0	124	12933	484321	
GENERALJOURNAL	#ЗАПАЗИ 29.09.2022	Ф484441	Доставчик	12933	3572.82	0	124	12933	484441	
GENERALJOURNAL	#ЗАПАЗИ 29.09.2022	Ф484535	Доставчик	12933	548.1	0	124	12933	484535	
GENERALJOURNAL	#ЗАПАЗИ 29.09.2022	Ф484643	Доставчик	12933	879.4	0	124	12933	484643	
GENERALJOURNAL	#ЗАПАЗИ 29.09.2022	Ф484839	Доставчик	12933	488.1	0	124	12933	484839	
GENERALJOURNAL	#ЗАПАЗИ 29.09.2022	Ф485020	Доставчик	12933	349.02	0	124	12933	485020	
JOURNALPOST	#ЗАПАЗИ Банкови операции	13								

Фиг. 13. Файл за импорт на счетоводни операции в ПП Масопому

Особеното тук е, че всяка фактура е съгласувана с плащането, т.е. постигнато е най-високо ниво на интеграция и автоматизация на процеса на плащане. От счетоводителя се изисква да импортира гореописания файл в счетоводния софтуер и счетоводните операции за плащанията са записани.

Файлът за плащане по стандарта на Обединена Българска Банка изглежда по следния начин (фигура 14):

IBAN311|АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ ЕТ|BG62S TSA93000021669647|STABGSF|BGN|10768.31|Ф959-040-266-321-441-535-643-839-02|0-C124|

Фиг. 14. Файл с данни за плащания по стандарт на ОББ

Файлът за плащане по стандарта на Уникредит Булбанк изглежда по следния начин (фигура 15):

FIX WIN BULBANK 29/09/2022 2209291419 29/09/2022 04/10/2022 00017 65192.17 BG14UNCR70001519877395 BGN
831314284 АЛЕКС-АЛЕКСАНДЪР ХРИСТОВ СПАСОВ ЕТ 10768.31 BG62STSA93000021669647 STABGSF Ф959-040-266-321-441-535-643-839-020-C124

Фиг. 15. Файл с данни за плащания по стандарт на Уникредит Булбанк

3. Заключение

В тази статия е представен метод за автоматизиране на банковите плащания към доставчици, чиято софтуерна реализация е внедрена и валидирана в счетоводството на реално предприятие. Към настоящия момент прогнозираният ефект от икономия на работно време и ускоряване на процеса на плащане се потвърждава. Съгласно наблюдения на експертите, извършващи дейностите по плащанията и осчетоводяванията, в предприятието се постига икономия до шест работни часа седмично при осъществяване на плащанията и до шестнадесет работни часа седмично при осчетоводяване на плащанията.

Софтуерът е в процес на доработка и актуализация на функционалностите съгласно обратната връзка от практическото му използване и желанията на клиентите.

Благодарности

Специални благодарности на Ваня Желязкова, която помогна при анализа на процеса на плащане и даде ценни съвети за практическата реализация на софтуера.

Литература

- [1]. Sanjaya, P., Dwi Hastuti, T., Koeswoyo, G.F., Accounting-based Digital Payment Systems for SMEs, 2021 International Conference on Computer & Information Sciences (ICCOINS), 13-15 July 2021 Universiti Teknologi PETRONAS (UTP)

- [2]. H. Luo, M. Das, J. Wang, J.C.P. Cheng, Construction Payment Automation through Smart Contract-based Blockchain Framework, 36th International Symposium on Automation and Robotics in Construction (ISARC 2019)
- [3]. Sigalov, K., Ye, X., König, M., Hagedorn, Ph., Blum, F., Severin, S., Hettmer, M., Hückinghaus, Ph., Wölkerling, J., Groß, D., Automated Payment and Contract Management in the Construction Industry by Integrating Building Information Modeling and Blockchain-Based Smart Contracts, Applied Sciences, 2021
- [4]. Raja, P. V. R. P., Jauharb, S. K., Ramkumarc, M., Pratapd, S., Accounting-based Digital Payment Systems for SMEs, Computers & Industrial Engineering, Volume 167, May 2022, 108038
- [5]. Shahriar, F. , Vendor payment automation & the role of SAP, BRAC Business School, BRAC University, Mohakkhali, Dhaka, April 2016
- [6]. Islam, K., Ferdaus, H., Rahman, S., Automation of Utility Bill Payment Using Telecom, Next-Generation Wireless Systems, ICNEWS 2006
- [7]. Pellegrini, M, The automation of tolls and the use of methods of deferred payment in Italy, 1982
- [8]. Kanojia, R., Pathak, S., Secured Vehicle Toll Payment System Using NFC, 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 18617732
- [9]. <https://www.bsoft.bg/upravlenie-na-plashtaniyata/>
- [10]. https://accounting.controlisy.com/index.php?page=news_details&news_id=23
- [11]. <http://www.businessoft.bg/bg/products/workflow>
- [12]. <https://microinvest.net/Счетоводен-софтуер>
- [13]. <http://www.cfinance.bg/index.html>
- [14]. <http://www.cfinance.bg/index.html>
- [15]. <http://www.bulmaroffice.com/bg/продукти/преглед/15/счетоводство-+/категория/1/описание/>
- [16]. <https://plusminus.com/bg/schetovoden-softuer>
- [17]. <https://kik-info.com/schovodstvo/platejno-narejdane.php>
- [18]. <https://www.documents-online.net/>
- [19]. <https://www.e-docs.bg/pages/help/bankovi-platezhni-dokumenti>

За контакти:

Докторант Иван Л. Игнев
катедра „КНТ”

Технически университет-Варна
E-mail: ignev.ivan@gmail.com



Z-WAVE – РЕШЕНИЕ ЗА ПРЕДИЗВИКАТЕЛСТВОТА НА ДОМАШНАТА АВТОМАТИЗАЦИЯ

Диян Ж. Динев, Биляна Илиева, Айдын Хъкъ

Резюме: В днешно време е невъзможно да се живее без компютри и компютърни мрежи поради бързото развитие на технологиите, които са проникнали в почти всеки аспект от ежедневието. Домашната автоматизация, която включва дистанционно управление и автоматизация на електрически и електронни машини, е едно от най-значимите и жизненоважни приложения на технологията в домовете. За да се свържат различни устройства от дома едно с друго и с интернет, се използват безжични мрежи за домашна автоматизация (WHAN). Има множество протоколи, като Home Plug и ZigBee, които могат да се използват за дистанционно управление на домашни уреди. В тази статия е разгледано развитието и приложенията на една сравнително нова технология Z-Wave за изграждане на интелигентни домове.

Ключови думи: IoT, Smart Homes, Wireless Sensor Networks, Z-Wave

Z-Wave – a solution to the challenges of the home automation

Diyan Zh. Dinev, Bilyana Ilieva, Aydan Haka

Abstract: Nowadays, it is impossible to live without computers and computer networks due to the rapid development of technology that has permeated almost every aspect of our daily life. Home automation, which includes remote control and automation of electrical and electronic machines, is one of the most significant and vital applications of technology in homes. Wireless home automation networks (WHANs) are used to connect various devices in homes to each other and to the Internet. There are multiple protocols, such as Home Plug and ZigBee that can be used to remotely control home appliances. This article reviews the development and applications of a relatively new Z-Wave technology for building smart homes.

Keywords: IoT, Smart Homes, Wireless Sensor Networks, Z-Wave

1. Въведение

Z-Wave е международен стандарт за безжична домашна автоматизация. Домашната автоматизация позволява да се свържат всички функции, свързани с електричеството, като светлина, отопление, готвене, охлаждане, сигурност и т.н., и да се приложи автоматизация на тези функции. Това води до повече сигурност и повече удобство в домовете и офисите. Домашната автоматизация също помага за пестене на енергия и други ресурси [1, 2].

Взаимното свързване на всички тези функции може да се осъществи с помощта на кабели или безжична технология. Особено за кабелна домашна автоматизация така наречената “Европейска инсталационна шина” или KNX е много популярна [3]. Кабелните решения са много надеждни, но изискват правилно планиране на кабелите и устройствата по време на изграждането на дома и инсталирането на всички комунални услуги.

За преоборудване или частични решения кабелна система за домашна автоматизация не е приложима. Тук безжичните решения влизат в действие. За съжаление на пазара все още няма ясен стандарт за протокол за безжична домашна автоматизация.

2. Изисквания към безжичните системи за управление на дома

За да се определи добра безжична технология за домашна автоматизация, трябва да се вземе предвид списък с изисквания [4]:

1. Надеждност на комуникацията: Важни функции като щори на прозорци или дори охранителни инсталации трябва да се управляват чрез безжичен сигнал. Следователно е от съществено значение всички съобщения да достигнат местоназначението си и да бъдат потвърдени от полученото устройство обратно към предавателя. Не всички безжични протоколи отговарят на това изискване.

2. Сигурност на комуникацията: Трябва да се гарантира, че неоторизирана трета страна не може – нарочно или случайно – да прихване или да се намеси в комуникацията на безжичната система. Обикновено технологиите за кодиране и механизмите за аутентикация гарантират това.

3. Ниско радио излъчване: Безжична технология за домашна автоматизация се използва в хола, следователно трябва да се вземат предвид въпроси като електромагнитно излъчване.

4. Лесна употреба: Домашната автоматизация ще направи живота на потребителя по-лесен, а не по-сложен.

5. Защита на инвестицията: Решенията за домашна автоматизация обикновено се инсталират по време на строителството на нови сгради или реновиране и трябва да отговарят на типичните жизнени цикли на продукта на оборудването за домашно инсталиране. Важно е да се уверите, че потребителят може да замени устройства или да разшири своите системи дори след години и да не се натъква на проблеми със съвместимостта.

6. Оперативна съвместимост: Функциите за домашна автоматизация като отопление, осветление или управление на прозорци се изпълняват с продукти на различни доставчици с опит в съответната област. Не е приемливо ограничаването само към един доставчик и да се купува, например отоплителна технология от доставчик с основна компетентност в осветлението, само за да се осигури оперативна съвместимост. Всяка инсталирана безжична технология трябва да се използва независимо от няколко производителя.

3. Z-Wave

Технологията Z-Wave е ключът към пълен контрол за домашните системи за сигурност и енергия, с минимален шум [5]. Със Z-Wave система за домашна автоматизация могат да се програмират всички основни електрически елементи на дома, като светлина, отопление, готвене, охлаждане и дори сигурността на дома [2].

Предимствата не свършват дотук. Въпреки това, че е сложна система, тя е лесна за използване и работи като енергийно ефективен и евтин вариант.

Системата работи чрез дистанционно управление и използва радио вълни с ниска мощност. Неговата мрежа покрива всички зони на дома, като радио вълните преминават лесно през стени, подове и мебели, правейки свързаността 100% надеждна.

Тази свобода на свързване означава, че може лесно да се започне с основен пакет, който да се надгражда с течение на времето с допълнителни компоненти, персонализиране на домашните системи за енергия и сигурност. Всеки Z-Wave модул може да действа като RF повторител и командите могат да маршрутизират през максимум четири устройства. Това дава на системата максимален обхват от 400 фута и маршрутизацията се управлява автоматично.

4. Z-Wave Типове устройства

Има два основни типа устройства, дефинирани в протокола Z-Wave: контролери и подчинени устройства. Контролерите могат да инициират предаване, както и да държат всички интелигентни устройства, свързани с мрежовите маршрути. От друга страна, подчинените устройства са само крайни устройства с функционалност входно – изходен тип (GPIO) с общо предназначение, които сякаш изпълняват заявките на контролера. Това е така

и за препредаването на съобщения: в приетия пакет контролерът инструктира конкретно устройство дали съобщението трябва да бъде препредадено или не. Контролерите се диференцират допълнително в зависимост от тяхната функционалност в мрежата. Основните видове са преносими и статични контролери.

Преносимите контролери - тези устройства може свободно да променят местоположението си в мрежата. Те има способността да откриват и преоткриват своята позиция в мрежата чрез пингване на околните възли. Обикновено преносимите контролери са устройства, управлявани от батерии, които се използват от потребителя за изпращане на команди към мрежата. Вторичната, но важна функция на преносимия контролер е да включва или изключва устройства в мрежата. Всяка Z-Wave мрежа трябва да има един основен контролер, който управлява процеса на включване / изключване и поддържа най-новата мрежова конфигурация. Други контролери копират тази информация от основния контролер. Операцията по включване на възел включва обмен на данни между избрания възел и първичния контролер, използвайки предаване с ниска енергия. Това значително опростява мрежовата конфигурация от гледна точка на потребителя, но също така един от преносимите контролери поема ролята на основния контролер.

Статичните контролери поемат фиксирана позиция в мрежата и се захранват от главната линия. Те винаги са в режим на слушане, следователно други устройства могат да комуникират с тях по всяко време. В по-сложна мрежа статичен контролер може да разшири своята функционалност и да стане устройство за съхранение на най-новата мрежова конфигурация, наречена Static Update Controller (SUC). Статичният контролер може дори да стане основен контролер в мрежата и да използва преносими контролери като прокси, за да включи и изключи други възли. В този случай статичният контролер се нарича SUC ID Server (SIS ID Server). Естествено, статичните контролери могат да работят като мостове към други елементи на домашната система, като TCP/IP мрежа или тривиални X10 устройства. Интересно е, че Z-Wave мрежата може да поддържа до 125 виртуални възли, използвайки специален контролер/и за мост. Контролерът мост, където е реализирана функционалността на виртуалния възел, позволява безпроблемно мигриране към Z-Wave мрежа в местата, където вече са използвани други технологии за домашна автоматизация. С други думи, може да се добавят Z-Wave устройства (като X10 устройства) към съществуваща мрежа и да се използва мостови контролер, за да осигури интерфейс между различните мрежи. На Z-Wave мрежата X10 модулите изглеждат като редовни възли, и обратно.

Подчинените устройства имат много по-проста функционалност от контролерите. Те не могат да инициират предаване, освен ако не отговорят на искането на контролера. Следователно, за да се получи информация от редовно подчинено устройство, контролерът трябва да проверява състоянието на устройството през периодични интервали. Подчинените устройства могат да бъдат използвани за повторно предаване на съобщението, като в този случай такива устройства се наричат маршрутизиращи подчинени, но те трябва да бъдат в режим на слушане през цялото време. Тъй като маршрутизираните подчинени устройства могат да предават нежелани съобщения, такива устройства могат да се използват, когато периодичното избиране не е приемливо, като например в аларми за пожар или кражба. Ако подчиненото устройство е предназначено да се използва като сензор, работещ с батерия, който периодично се събужда, за да комуникира със статичния контролер (например в продуктите от тип сензор за двигатели), отговорността на разработчика е да приложи

логиката в приложението на контролера, което ще постави подчинения в режим на дълбоко заспиване за определен период от време и ще създаде логика за съвпадение в приложението на подчинения, за да уведоми статичния контролер всеки път, когато подчиненото устройство се събуди.

5. Приложения

Z-Wave устройствата са интелигентни устройства за интелигентни домове. Благодарение на голямото разнообразие от поддържани продукти за интелигентен дом, Z-Wave предлага много възможности за собствениците на жилища. По този начин може да персонализира интелигентния дом, за да отговори на точните нужди на потребителя. Има следните приложения:

- Интелигентното осветление - позволява да се контролират светлините в мрежата. Интелигентното осветление с Z-Wave позволява да се намали интензитета им с дистанционно, да се включват или изключват, или да се променят цветовете им. Със Smart Home Hub или Z-Wave контролер може дори да се стигне до създаването на графици за включване на светлините в определени часове. Или, например, включване на светлините, когато се отворят вратите. Ако всичките светлини са свързани към Z-Wave контролер, може да се достъпят всички дистанционно. Това означава да се включват или изключват от дивана или от всяка точка на света. Има няколко различни начина да се направи осветлението по-интелигентно с различни продукти на Z-Wave. Може да се контролират „обикновените“ лампи чрез интелигентни щепсели или вградени ключове. Освен това има Z-Wave интелигентни крушки, които могат директно да се управляват дистанционно и могат да се контролират индивидуално. Повечето интелигентни крушки са LED и се предлагат в стандартен бял цвят или с възможност за промяна на цветовете. За да се контролират, превключвателят на светлината обикновено трябва да е включен през цялото време, за да контролира светлината.

- Умни щепсели - те са малки Z-Wave устройства, които могат да се поставят между устройствата и електрическия контакт. По този начин може да се контролира устройството, което е включено. Например, да се затъмни или да се включи или изключи устройство с телефон или чрез автоматизиран сценарий. Вградените ключове са инсталирани зад неинтелигентен превключвател. Те обикновено могат да замъгляват светлините или да ги включват и изключват. Тези вградени ключове позволяват да се контролира осветлението на ключа дистанционно чрез Homey Smartphone App или чрез Homey Flows.

- Умни сензори - могат да откриват промени в дома като добавят ново ниво на безопасност и сигурност към дома. Те играят съществена роля, когато става въпрос за автоматизирани сценарии. Ако даден сензор открие нещо необичайно, той изпраща сигнал на смартфон. Най-често срещаните сензори са сензори за наводнение, датчици за дим, сензори за движение и сензори за врати/прозорци. По същество основните им функционалности са доста ясни. Сензорите за вода/наводнения са сензори, използвани за откриване на вода и предупреждаващи за наводняване, така че могат да се предприемат подходящи действия, за да се избегнат повреди в къщата. Детекторите за пушек информират, ако в дома има пожар или проблем с въглеродния оксид. Сензорите за движение откриват движение. Някои от тях обаче следят светлината, температурата и влажността. И накрая, сензорите за врати/прозорци следят състоянието на вратите и прозорците. Всички тези сензори могат да докладват директно до смартфон, ако нещо се случи. Освен това те могат да помогнат да се създаде автоматизация на дома. Например включване на светлините, музиката или термостатите въз основа на откриването на движение и ако няма открито движение, сензорите могат да изключат всички устройства, което спестява много енергия. Тези сензори се свързват много добре с всеки интелигентен дом.

- Смарт брави - те позволяват да се знае кой влиза или излиза от дома. И щом веднъж интелигентното заключване е свързано към интелигентния хъб, то може да има дистанционен достъп до вратата от всяка точка на света. Това предлага много възможности. В случай че вратата не е заключена, може да се заключи с приложение за смартфони, което прави ключовете излишни. С интелигентното заключване винаги има достъп до вратата, за да бъде отворена. Някои умни брави дори предлагат до 100 уникални ключа. По този начин може да се предостави достъп до всички членове на семейството.

6. Предимства и недостатъци

По-долу са посочени предимствата на този протокол за безжична комуникация [6, 7]:

- Лесно внедряване - Подобно на други протоколи и стандарти за безжична комуникация, той е лесен за внедряване и не изисква повторно окабеляване и сложно окабеляване, тъй като активираните устройства комуникират безжично и повечето от тях работят с батерии.

- Ниски смущения - Друго предимство на Z-Wave е, че не страда от големи проблеми с радиосмущенията и физически препятствия, тъй като работи около честотната лента от 900 MHz на електромагнитния спектър в сравнение с честотната лента от 2,4 GHz, използвана от Wi-Fi 2,4 GHz, Bluetooth и Zigbee.

- Мрежово покритие - Фактът, че използва електромагнитно излъчване с по-ниски честоти от честотната лента от 2,4 GHz, използвана от конкурентни протоколи за безжична комуникация, му дава допълнително предимство по отношение на обхвата, който може да се простира до 100 метра или повече и може да премине през най-солидните обекти като стени и подове.

- Решетъчна мрежа (Mesh network) - Топология, базирана на Mesh мрежа. Тя може да разшири допълнително покритието на конкретна мрежа, защото способните устройства стават повторители на сигнала и колкото повече устройства са свързани в мрежата, толкова по-силна става тя;

- Енергийна ефективност - Той е идеален за малки интелигентни устройства, работещи с батерии, тъй като по същество е алтернатива с много по-ниска мощност на Wi-Fi технологията, но с по-широк обхват от Bluetooth и Zigbee технологиите.

- Оперативна съвместимост - Протоколът се притежава и поддържа от частна организация, за да се гарантира спазването на стандартите, като по този начин се гарантира, че всички активирани устройства са съвместими и напълно оперативно съвместими едно с друго.

Като всяко едно нещо и този протокол за безжична комуникация има своите недостатъци [6, 7]:

- Предаване на данни - Този протокол не е подходящ за стрийминг и големи трансфери на данни, подобно на ZigBee и за разлика от Wi-Fi и Bluetooth технологиите, тъй като е предназначен основно за ниска латентност на малки пакети данни със скорост до 100 килобита в секунда.

- Ограничен брой възли - за разлика от Zigbee, който може да поддържа до 65 000 възли в една мрежа, друг значителен недостатък на Z-Wave е, че поддържа само до 232 възли, като същевременно ограничава броя на допустимите хъбове до четири.

- Дървовидна топология - протоколът поддържа само дървовидна топологична структура, която има специфични недостатъци, включващи трудност при конфигуриране, проблеми с производителността, съответстващи на увеличен брой възли, податливост на гъст мрежов трафик в зони с тесни места и ограничения в дължината на сегмента.

- Разходи за внедряване - система, базирана на Z-Wave, е сравнително евтина за внедряване, тъй като има широк избор от достъпни устройства на пазара, но разходите за внедряване могат да се повишат при разширяване на мрежовото покритие.
- Проблеми със сигурността - протоколът също е присъщо защитен, но все още се нуждае от специфични мерки за сигурност, приложени от опитен потребител, за да бъде по-малко уязвим за хакване, което може да компрометира поверителността и сигурността на мрежата.
- Затворена система - актуализациите на протокола и софтуера или фърмуера на активираните устройства зависят от Z-Wave Alliance и няма ресурси с отворен код, които биха позволили на потребителите да прилагат свои собствени поправки и модификации.

7. Заключение

Целта на настоящата статия е да предостави информация за технологията Z-Wave, като включва базова информация за технологията, Z-Wave комуникационния протокол, Z-Wave видовете устройства и техните спецификации, приложенията на Z-Wave технологията, както и нейните предимствата и недостатъци. На база на тази информация са направени изводи за подходящи сфери на приложение на технологията

Литература

- [1]. Phan minh L. An and Taehong. K., "A Study of the Z-Wave Protocol: Implementing Your Own Smart Home Gateway," *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, 2018, pp. 411-415, doi: 10.1109/CCOMS.2018.8463281.
- [2]. Yassein M., Khalil W. and Khalil A., "Smart homes automation using Z-wave protocol," *2016 International Conference on Engineering & MIS (ICEMIS)*, 2016, pp. 1-6, doi: 10.1109/ICEMIS.2016.7745306.
- [3]. Boucif, N., Golchert, F., Siemer, A., Felke, P., and Gosewehr, F., "Crushing the Wave -- new Z-Wave vulnerabilities exposed", arXiv e-prints, 2020.
- [4]. Varol A. , "Compilation of Data Link Protocols: Bluetooth Low Energy (BLE), ZigBee and Z-Wave," *2019 4th International Conference on Computer Science and Engineering (UBMK)*, 2019, pp. 85-90, doi: 10.1109/UBMK.2019.8907068.
- [5]. Samuel S., "A review of connectivity challenges in IoT-smart home," *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*, 2016, pp. 1-4, doi: 10.1109/ICBDSC.2016.7460395.
- [6]. Babun, L., Aksu, H., Ryan, L., Akkaya, K., Bentley, E. S., and Uluagac, A. S. 2020. Z-IoT: Passive Device-class Fingerprinting of ZigBee and Z-Wave IoT Devices. ICC 2020 – 2020 IEEE International Conference on Communications. ICC 2020 – 2020 IEEE International Conference on Communications. DOI: 1109/icc40277.2020.9149285
- [7]. Badenhop, C. W., Graham, S. R., Ramsey, B. W., Mullins, B. E., and Mailloux, L. O. 2017. "The Z-Wave Routing Protocol and its Security Implications." *Computers & Security*. 68: 112-129. DOI: 1016/j.cose.2017.04.004

За контакти:

Гл. ас. д-р инж. Диян Желев Динев
 катедра „Софтуерни и Интернет Технологии”
 Технически университет – Варна
 E-mail: diyandinev@tu-varna.bg

STUDENT – програмен език за обучение по компилатори

Ивайло П. Пенев, Димитър Ст. Димитров

Резюме: Статията представя учебния програмен език STUDENT, който е създаден в Технически университет – Варна за целите на учебния процес по дисциплини, свързани с езикови процесори, компилатори и интерпретатори, формални езици и граматика и др. Представени са основните елементи и конструкции в езика. Показани са примерни програми на STUDENT за решаване на базови задачи.

Ключови думи: Учебен програмен език, учебен компилатор, езиков процесор, азбука, лексика

STUDENT - programming language for education in compilers

Ivaylo P. Penev, Dimitar S. Dimitrov

Abstract: The paper presents the training programming language STUDENT, which is created at Technical University of Varna for education in subjects like language processors, compilers and interpreters, formal languages and grammars. The elements and constructs of the language are described. Programs in STUDENT for solving basic problems are presented.

Keywords: Training programming language, training compiler, language processor, alphabet, lexis

1. Въведение

Учебните дисциплини „Езикови процесори“ и „Компилатори и интерпретатори“ имат важно значение в обучението на компютърни и софтуерни инженери, защото демонстрират ключовото взаимодействие между теория и практика в компютърните науки и информатиката. Същевременно обучението по тези дисциплини предоставя редица предизвикателства, особено по отношение на привличане и задържане на интереса към дисциплините от страна на студентите.

Един от най-важните фактори за успешното провеждане на тези дисциплини е изборът на програмен език, за който ще бъде разработвана транслираща система – компилатор и/или интерпретатор. От една страна езикът не трябва да е твърде сложен, за да бъде постижимо разработване на транслятор за този език в рамките на един учебен семестър. От друга страна транслираният език трябва да притежава базовите елементи и конструкции, характерни за масово използваните програмни езици, за да могат да бъдат изучени основните принципи, по които работят съвременните компилатори и интерпретатори.

Известни са множество учебни програмни езици и компилатори, използвани в различни университетски курсове [5, 8, 9]. Например в Технически университет – Виена е разработен програмен език MiniJava, който е подмножество на езиците Java и C [1, 14]. Този език е твърде опростен - входната програма се състои само от една функция без възможност за деклариране на променливи. Това лишава преподавателите от възможността да представят пълноценно най-важните концепции при транслирането на входни програми.

Друг популярен учебен език е COOL (Classroom Object-Oriented Language), разработен и използван за обучение в Станфордския университет [2, 11]. COOL е обектно-ориентиран език и има функционалности на практически използван език за програмиране. Богатите функционалности на езика правят разработването на компилатор за езика сериозно предизвикателство, поради което в курса се използват генератори на лексически и синтактичен анализатор. Автоматизираното генериране на код скрива от студентите методите и алгоритмите за анализ на входни програми и синтез на изходни програми.

ChocoPy е език за програмиране, предназначен за студенти по езикови процесори и компилатори в Калифорнийския университет, Бъркли [6, 13]. ChocoPy е ограничено подмножество на Python 3, което лесно може да бъде компилирано до целева архитектура като RISC-V [3]. Програмите на ChocoPy могат да се изпълняват директно с официалния компилатор CPython за езика Python.

В Университета в Нови Сад, Сърбия, е разработен компилатор за език miniC, подмножество на C [7]. Този език също е твърде опростен (напр. поддържа извикване на функции само с един аргумент).

LISS (Language of Integers, Sequences and Sets) е език за програмиране, който е разработен в Университета Минхо, Брага, Португалия за курса по компилатори и интерпретатори [4]. LISS поддържа следните типове данни: integer, boolean, масиви (от цели числа). Езикът също така поддържа оператори if, switch-case, for и while. LISS има синтаксис, близък до Pascal, който се използва малко в последните години и това го прави труден за разбиране от студентите.

В Технически университет – Варна са разработени учебен програмен език, наречен STUDENT, и учебен компилатор за този език. Компилаторът генерира код за виртуална машина. Този код се интерпретира и изпълнява от допълнително разработен интерпретатор за виртуалната машина. Езикът е проектиран така, че да съдържа всички основни елементи на практически използваните езици.

2. Програмен език STUDENT

2.1. Общо описание на STUDENT

Програмният език STUDENT е процедурно ориентиран език от високо ниво и е подмножество на езиците C и Java, които се изучават от студентите по компютърни науки и информатика във всички университети. STUDENT предоставя всички необходими функционалности за решаване на базови практически задачи.

2.2. Азбука на езика STUDENT

Азбуката на езика включва следните символи:

- малки и главни букви ('a'..'z', 'A'..'Z')
- цифри ('0'..'9')
- специални символи (например '+', '-', '%', '!', '*', '/', ';', '(', ')', '[', ']', '{', '}', '@', '&', '|', '!', '>', '<', '=', ' ', '"', "'")

2.3. Основни елементи на STUDENT

В езика са реализирани следните конструкции:

- *Оператори*
 - аритметични оператори (+, -, *, /, %)
 - логически оператори (&&, ||, !)
 - оператори за сравнение – релационни оператори (<, >, <=, >=, !=, ==)
 - оператор за присвояване (=)
 - унарни оператори (-)
 - условен оператор (if else)
 - оператор за цикъл (while)
 - съставен оператор ({ ... })
 - входно-изходни оператори (read, print)

- *Константи*
 - целочислени
 - символни
 - низови
 - булеви
- *Променливи* - имената на променливите трябва задължително да започват с буква, след която може да следва последователност от букви и/или цифри
 - локални – имат област на видимост в блока, в който са декларирани и във вложените блокове; нямат област на видимост извън блока, в който са декларирани
 - глобални – имат област на видимост във всички блокове на програмата
- *Изрази*
 - аритметични
 - логически
- *Подпрограми функции*
- *Управляващи символи* ('\\n', 't').

2.3. Ключови думи

Езикът STUDENT има 14 ключови думи:

- int
- char
- void
- boolean
- while
- return
- if
- else
- true
- false
- print
- read
- length
- program

2.4. Типове данни

В STUDENT са реализирани следните типове данни:

- *Прости* типове данни
 - целочислен тип
 - булев тип
 - символен тип
- *Структурирани* типове данни
 - едномерен масив

2.5. Лексика на STUDENT

Лексемите в STUDENT (най-малките значещи структурни единици в програма на езика) са ключови думи, оператори, разделители, константи и идентификатори. Те са представени в таблици 1 ÷ 7.

В лявата колона на всяка таблица е показано вътрешното име, с което се представя и обработва съответната лексема в компилатора, а в дясната колона е символният низ на съответната лексема. Например лексемата със символен низ “if” се представя и обработва с име **IF** в компилатора.

Таблица 1. Ключови думи

Име на лексемата в компилатора	Низ на лексемата
IF	'if'
INT	'int'
CHAR	'char'
ELSE	'else'
READ	'read'
TRUE	'true'
VOID	'void'
FALSE	'false'
WHILE	'while'
PRINT	'print'
LENGTH	'length'
RETURN	'return'
BOOLEAN	'boolean'
PROGRAM	'program'

Таблица 2. Оператори

Име на лексемата в компилатора	Низ на лексемата
PLUS	'+'
MINUS	'-'
MUL	'*'
DIV	'/'
MOD	'%'
BECOMES	'='

Таблица 3. Релационни оператори

Име на лексемата в компилатора	Низ на лексемата
NOT	'!'
EQUALS	'=='
NOTEQUALS	'!='
GREATER	'>'
LESS	'<'
GREATER_EQ	'>='
LESS_EQ	'<='

Таблица 4. Логически оператори

Име на лексемата в компилатора	Низ на лексемата
AND	'&&'
OR	' '

Таблица 5. Разделители

Име на лексемата в компилатора	Низ на лексемата
LSQUARE	'['
RSQUARE	']'
LBRACKET	'{'
RBRACKET	'}'
LPAREN	'('
RPAREN	')'
SEMICOLON	','
COMMA	','
SINGLE_QUOTE	'\"'
DOUBLE_QUOTES	'\"'
ARROW	'->'
AT	'@'

Таблица 6. Целочислени, булеви и символни константи

Име на лексемата в компилатора	Низ на лексемата
INT_LITERAL	'0' [1-9]{1}[0-9]*
CHAR_LITERAL	всеки ASCII символ
BOOLEAN_LITERAL	'true' 'false'

Таблица 7. Идентификатори

Име на лексемата в компилатора	Низ на лексемата
IDENTIFIER	[A-Za-z]{1}[A-Za-z0-9]*

Валиден идентификатор (всяко потребителско дефинирано име) в езика STUDENT започва с една буква (символ в диапазона ‘a-z’ или ‘A-Z’), следвана от произволен брой букви и/или цифри.

3. Примерни програми на STUDENT

Показани са три примерни програми на STUDENT за решаване на съответните базови задачи.

3.1. Програма на STUDENT за извеждане на символен низ “Hello, world!”

```
program {  
  main() -> void {  
    print("Hello, world!");  
  }  
}
```

Програмата на студент задължително трябва да започва с ключова дума “program”. Тялото на програмата се състои от списък от функции, който задължително трябва да включва главна функция с идентификатор “main”. Главната функция не може да връща резултат, различен от void. Това се дължи на факта, че на този етап компилираните програми на STUDENT не могат да се стартират от команден ред. Компилирането и стартирането става от използваната интегрирана среда за разработване.

3.2. Програма на STUDENT с функция за намиране на по-голямото от две цели числа, подадени като параметри

```
program {  
  greater(int num1, int num2) -> int {  
    if (num1 > num2) {  
      return num1;  
    }  
    else {  
      return num2;  
    }  
  }  
  main() -> void {  
    print(@greater(10, 5));  
  }  
}
```

Извикването на функция в STUDENT започва със специален символ “@”, предшестваш идентификатора на функцията и списъка с актуални параметри.

3.3. Програма на STUDENT за изчисляване на числата на Фибоначи

```
program {
    int num = 0;

    fib(int num, int[] memo) -> int {
        if (memo[num] > 0) {
            return memo[num];
        }
        if (num == 0) { return 0; }
        if (num == 1) { return 1; }

        int res = @fib(num - 1, memo) + @fib(num - 2, memo);
        memo[num] = res;
        return res;
    }

    main() -> void {
        print("Please enter a non-negative number:");
        read(num);
        if (num < 0) {
            print(num, "is a negative number, program will now terminate!");
            return;
        }
        int[] memo = int[num + 1];
        int i = 0;
        memo[0] = 0;
        if (num > 0) { memo[1] = 1; }
        @fib(num, memo);
        char[] fib = "Fibonacci number";
        while (i <= num) {
            print(fib, i, ':', memo[i]);
            i = i + 1;
        }
    }
}
```

Това е входната програма, с която студентите тестват разработените модули на компилатора, защото тя съдържа всички елементи и конструкции на езика.

4. Заключение

Програмният език STUDENT е базиран на програмни езици, които се изучават по няколко дисциплини и са познати за студентите. Езикът се дефинира чрез граматика от тип LL(1), което позволява да бъде разработен LL(1) синтактичен анализатор за програми на STUDENT. Този тип синтактични анализатори са особено подходящи за обучение, защото могат да бъдат реализирани в рамките на един учебен семестър, при което студентите придобиват знания за основните принципи на работа на съвременните компилатори.

Генерираните инструкции са подобни на инструкциите в абстрактния модел на JVM (Java virtual machine), което дава възможност на студентите да се запознаят с работата на реална виртуална машина.

Съществуват множество възможности за бъдещо развитие на езика STUDENT и съответно на учебния компилатор, напр. добавяне на многомерни масиви, добавяне на тип данни структура, допълнителни оператори (for, do ... while и др.).

Литература

- [1]. Roberts E., An overview of MiniJava, ACM SIGCSE Bulletin Volume 33, Issue 1, March 2001 pp 1–5
- [2]. Aiken, A.; Cool: a portable project for teaching compiler construction. ACM SIGPLAN Notices, Vol. 31, Iss. 7, ACM, New York, 1996, pp. 19-24.
- [3]. Waterman A., Lee Y., Patterson A., Asanović K.; 2014. The RISC-V Instruction Set Manual, Volume I: Base User-Level ISA Version 2.0. Technical Report UCB/EECS-2014-54. EECS Department, University of California, Berkeley.
- [4]. Da Cruz, D.; Henriques, P. R.; LISS – The Language and the Compiler, Departamento de Informática, Universidade do Minho, Compilers, Related Technologies and Applications, CoRTA'2007 Covilhã, Portugal, July 6, 2007, Braga, Portugal, pp.86-103.
- [5]. Na, Wang; ShiMing, Zhang; Construction of compiler technology course in application-based university, International Conference on Education Technology and Information System (ICETIS 2013), 2013
- [6]. Padhye, Rohan; Sen, Koushik; Hilfinger, Paul; ChocoPy: A Programming Language for Compilers Courses. SPLASH-E 2019: Proceedings of the 2019 ACM SIGPLAN Symposium on SPLASH-E Athens, Greece, October 2019 pp. 41–45
- [7]. Rakic, Zorica; Rakic, Predrag; Petric, Tara; miniC Project for Teaching Compilers Course, ICIST 2014 4th International Conference on Information Society and Technology, Belgrade, Serbia, 2014, pp. 360- 362
- [8]. Sekerinski, E; Zingaro, D.; Pascal0 Compiler. Course Notes for Compiler Construction, 2011
- [9]. Zatarain-Cabada, R.; Barron-Estrada, M.; Zepeda-Sanchez, L.; Quintero-Meza, R. ; Reyes Garcia, C. ; A Hybrid Learning Compiler Course, Hybrid Learning. ICHL 2010. Lecture Notes in Computer Science, ISBN: 978-3-642-14656-5, Vol 6248. Springer, Berlin, Heidelberg, 2010.
- [10]. Zhang, Haiyi; The Considerations of Teaching Computer Science Courses AI and Compilers, Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS); Athens, 2018
- [11]. <http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=Compilers>
- [12]. <https://eli.thegreenplace.net/2012/07/12/computed-goto-for-efficient-dispatch-tables>
- [13]. <https://chocopy.org/>
- [14]. <https://www.complang.tuwien.ac.at/ublu>

За контакти:

доц. д-р инж. Ивайло Пламенов Пенев
катедра „Компютърни науки и технологии”
Технически университет - Варна
E-mail: ivailo.penev@tu-varna.bg

инж. Димитър Стилиянов Димитров
катедра „Компютърни науки и технологии”
Технически университет - Варна
E-mail: d.dimitrov@tu-varna.bg

БЛОКЧЕЙН СИСТЕМА ЗА ТЪРГОВИЯ НА КРИПТОВАЛУТИ ЗА ЦЕЛИТЕ НА ОБУЧЕНИЕТО

Диян Ж. Динев, Станислав Л. Лазаров

Резюме: Криптовалутите са в центъра на обществения интерес, но най-големите иновации идват от блокчейн технологията, която се крие под повърхността, спестявайки на най-големите компании в света милиарди долари всяка година. Бързо променящата се среда налага прилагането на иновативни решения в областта на информационните технологии и уеб разработката. Последните изследвания показват, че се задава революция, която е започнала в близкото минало и се очаква да бъде за по-добро и далечно бъдеще. От една страна, финансовата криза и растящата инфлация от печатането на пари, а от друга, разумните очаквания за промяна на традиционните начини за обмен на информация между хора и устройства, предполагат, че концепцията за развитие на мрежа от трето поколение (Web 3.0) е напълно възможна. Вероятността тази сложна, сигурна, бърза, устойчива, децентрализирана и сигурна глобална система, да бъде базирана на блокчейн технология, е огромна. Целта на тази статия е да предостави система за търговия с криптовалута Blockchain чрез внедряване на нови криптовалути, интелигентни договори и NF токени за целите на обучението.

Ключови думи: Cryptocurrency, IoT, NF Token, Wireless Sensor Network

Blockchain cryptocurrency trading system for educational purposes

Diyan Zh. Dinev, Stanislav L. Lazarov

Abstract: Cryptocurrencies are at the center of public interest, but the biggest innovations come from blockchain technology, which lie beneath the surface, saving the world's largest companies billions of dollars each year. The rapidly changing environment requires the application of innovative solutions in the field of information technology and web development. Recent research indicates that a revolution is looming that started in the recent past and is expected to be for a better and distant future. On the one hand, the financial crisis and rising inflation from printing money, and on the other, the reasonable expectation to change the traditional ways of exchanging information between people and devices, suggest that the concept of the development of a third-generation network (Web 3.0) is completely possible. The probability of this complex, secure, fast, sustainable, decentralized, and secure global system to be based on Blockchain technology is huge. The goal of this paper is to provide a Blockchain cryptocurrency trading system by implementing new cryptocurrencies, smart contracts, and NF tokens for educational purposes.

Keywords: Cryptocurrency, IoT, NF Token, Wireless Sensor Network

1. Въведение

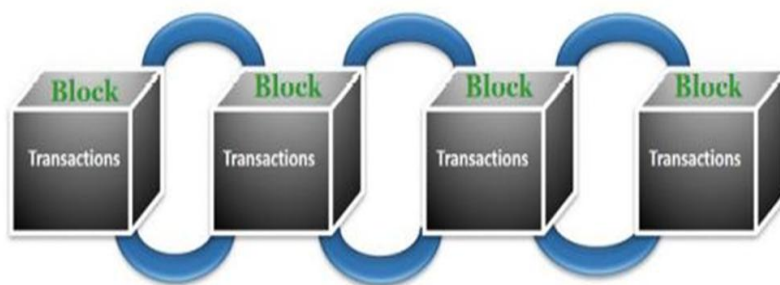
Бързо променящата се среда налага прилагането на иновативни решения в сферата на информационните технологии и уеб разработките. Последните изследвания сочат, че се задава революция, която е започнала в близкото минало и се очаква да е за по-добро и далечно бъдеще. От една страна финансовата криза и нарастващата инфлация от печатането на парични средства, а от друга обоснованото очакване за промяна на традиционните начини за обмен на информация между хора и устройства, подсказват, че концепцията за разработката на мрежа от трето поколение е напълно възможна. Вероятността тази сложна, сигурна, бърза, устойчива, децентрализирана и защитена глобална система да бъде на базата на блокчейн технологията, е огромна [3, 4].

Възможността за фалшификация на документи, патенти, авторски права и дори кражбата на самоличност предполагат научно решение на проблема чрез сложни алгоритми от сферата на криптографията и компютърните науки, за да решат много от проблемите, които стоят пред днешното общество. Едни от най-сериозните нарушения са кражбата на интелектуална собственост, лични или публични парични средства, та дори и кражбата на „дигитална самоличност“. Може би решението на този проблем може да се намери чрез множеството мерки за сигурност на транзакциите и обмен на данни, които предлага блокчейн технологията. Анонимността на транзакциите, премахването на нуждата от посредник между потребителите и ниските регулации са още аргументи в полза на тези технологии [1, 2]. Идеята, заложена в първоначалния замисъл, и разработката са от ключова важност. Макар и да крие доста рискове и възможности за злоупотреба, тази система си струва да бъде изследвана в дълбочина и да бъде разгърнат огромният ѝ потенциал, за да се прецени накъде ще се развива науката в тази сфера. Възможно е да се дадат иновативни решения на до момента нерешими проблеми. Основната цел на статията е да се изследва потенциала на блокчейн технологията като се разработи експериментална блокчейн система за тестване на транзакции на блокове, създаването на фиктивна криптовалута и така наречените първоначално предлагани монети ICO.

2. Същност на Blockchain технологията

Блокчейн, понякога наричана технология за разпределена книга (DLT), прави историята на всеки цифров актив непроменим и прозрачен чрез използването на децентрализирана мрежа и криптографско хеширане.

В технологията на разпределената книга записите и транзакциите се поддържат по разпределен децентрализиран начин с помощта на мрежови технологии Peer-2-Peer (P2P). Управлението и организацията на данните в DLT може да се извърши по различни начини. Например, данните могат да се управляват под формата на линеен свързан списък от блокове или могат да се управляват с помощта на Directed Acyclic Graph (DAG) или дървовидни структури от данни. Ако данните се управляват под формата на линейно свързан списък от блокове, тогава това е известно като "Blockchain". Една уникална характеристика на блокчейн е, че той напълно елиминира ролята на участието на доверени трети страни в поддръжката на блокчейн мрежата (фигура 1).



Фиг. 1. Множество блокове, съдържащи по няколко транзакции, свързани помежду си, образувачи „блокчейн мрежа“

„Блокчейн“ може да се дефинира като структура от данни, която е само за четене и данните не могат да бъдат модифицирани, след като бъдат въведени в блокчейна, а нови данни могат да се добавят само в края на блокчейна, което го прави силно неизменен.

Блокчейн технологията работи в P2P мрежа. P2P е различна мрежова парадигма от модела на комуникация клиент – сървър. Това е едно от предимствата на блокчейна, че той разчита на P2P, което означава, че не се изисква централен субект за управление на мрежата.

Blockchain по същество не е верига от блокове, но може да бъде и DAG. Блокчейн е некорумпиран цифровизиран, децентрализиран и публичен регистър на транзакции. Той може да бъде програмиран да записва не само финансови транзакции на криптовалути, но и практически всичко, което има стойност. Всички транзакции са пакетирани в блокове. Всеки блок съдържа криптографски хеш от предишния блок, времеви отпечатък и данни за транзакцията. По своя дизайн блокчейн е устойчива на модификация на данните. Веднъж записани, данните във всеки един блок не могат да се променят със задна дата, без да се променят всички следващи блокове. Това изисква консенсус на мнозинството в мрежата. Ако един блок данни се промени, всички следващи блокове ще бъдат анулирани. Когато една нова транзакция или редактиране на съществуваща транзакция влезе в блокчейн, обикновено по-голямата част от възлите в рамките на реализацията на блокчейн трябва да изпълняват алгоритми за оценка и проверка на историята на отделния блокчейн блок, който се предлага [4, 5]. Ако мнозинството от възлите достигнат до консенсус, че историята и подписът са валидни, новият блок от транзакции се приема в регистъра и се добавя нов блок към веригата от транзакции. Ако мнозинството не се съгласи с добавянето или промяната на записа в регистъра, то се отказва и не се добавя към веригата. Този модел на разпределен консенсус е това, което позволява на блокчейн да работи като разпределен регистър, без да е необходим някакъв централен, обединяващ орган, който решава кои транзакции са валидни. Данните, свързани с обмена, се записват в криптографски блокове, свързани йерархично и споделяни между много страни, наречени възли. Всяка транзакция е цифрово подписана, за да се гарантира нейната автентичност и да се попречи на някой да я фалшифицира. Това означава, че самият регистър и съществуващите транзакции в него могат да се приемат като цялостни.

3. Умни договори

Интелигентният договор е самоизпълняващ се договор, като условията на споразумението между купувача и продавача са директно записани в редове код. Кодът и съдържащите се в него споразумения съществуват в разпределена, децентрализирана блокчейн мрежа. Кодът контролира изпълнението, а транзакциите са проследими и необратими. Интелигентните договори позволяват надеждни транзакции и споразумения да се извършват между различни, анонимни страни без необходимост от централен орган, правна система или външен механизъм за прилагане. Въпреки че блокчейн технологията е започнала да се смята основно за основата на биткойн, тя еволюира далеч отвъд основата на виртуалната валута [9].

Интелигентните договори са самоизпълняващи се редове код с условията на споразумение между купувач и продавач, автоматично проверени и изпълнени чрез компютърна мрежа.

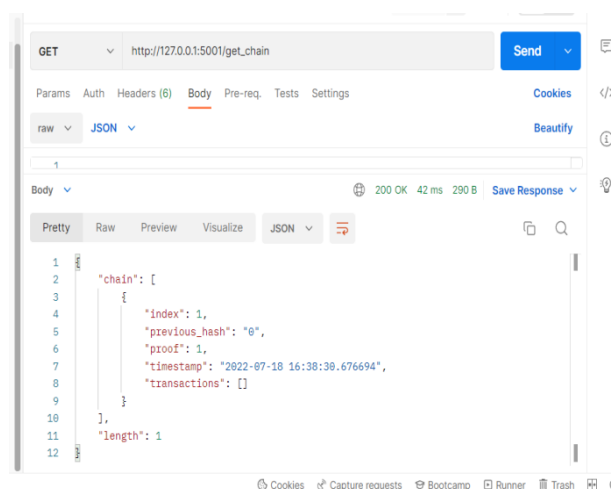
4. Локална блокчейн мрежа без запазване на състоянията при рестартиране на системата

За разработката се използва Anaconda, като се създава уеб приложение с помощта на Flask и различни библиотеки, които улесняват създаването на хеширащи функции и т.н. Нужните библиотеки и импорти са описани като коментари в началото на кода.

За проверка на състоянието на веригата и визуализация се използва софтуерът „Postman” за извършаване на POST & GET заявки за визуализация на кодовите фрагменти. [7].

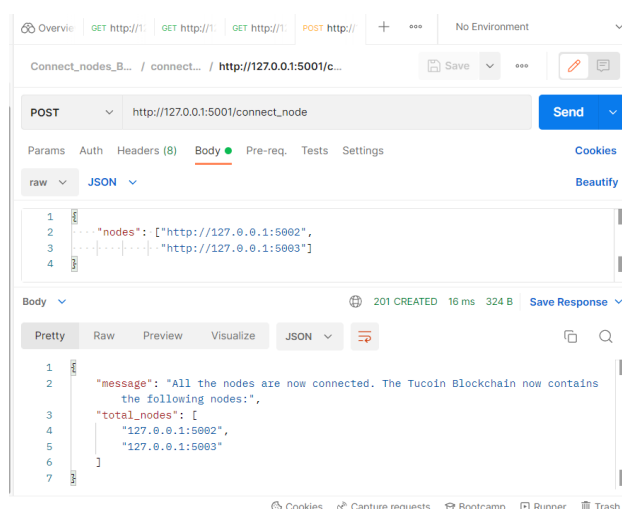
За тестване на функциите се стартират три различни нишки (конзоли) с портове 5001, 5002, 5003 в Spyder или PyCharm с идентичен код (tucoin_node_5001.py, tucoin_node_5002.py,

tucoin_node_5003.py), които се различават само по съобщенията към потребителя и съответните портове за стартиране на приложенията едновременно (фигура 2). За осъществяването на някои от функциите също се попълва нужната информация за POST заявките – в body се добавя съответната информация.



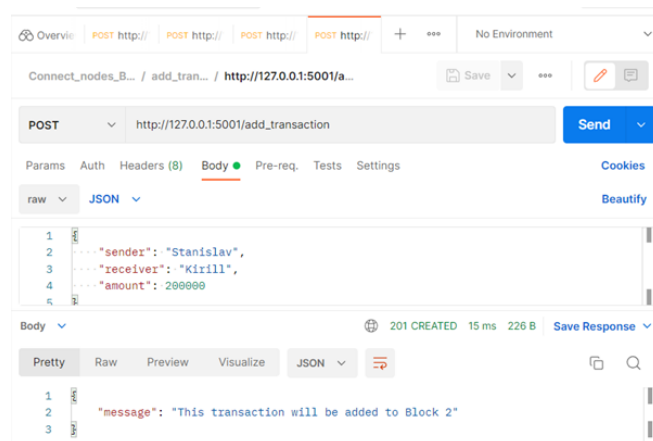
Фиг. 2. Инициализация на локалната Blockchain мрежа

Изпълнява се функцията за инициализирането на базовия блок, т.нар „генезисен“ блок - първоизточник на веригата. Всеки блок от веригата има индекс, който го идентифицира уникално, време за „изкопаване“, предходен блок, ако такъв съществува, и доказателство за работоспособност, транзакции и дължина на веригата (фигура 3).



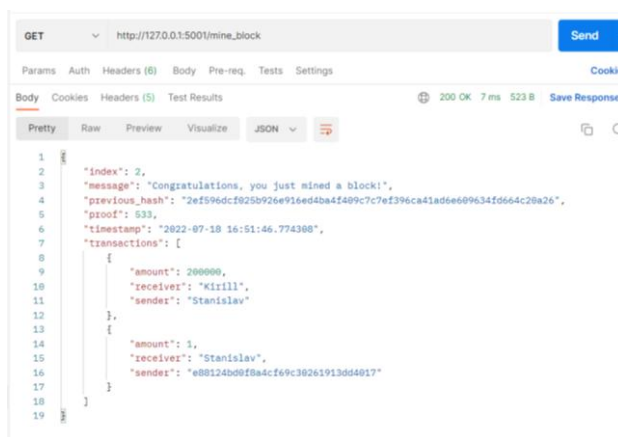
Фиг. 3. Свързване на различните нишки участници в споделената мрежа

Извършването на транзакция и запазването ѝ в даден блок на веригата е показано на фигура 4. Тук блокът още не е „изкопан“, затова се получава съобщение, че ще се добави в следващия блок, когато той бъде изкопан. Целта на използването на тази технология е да не позволява манипулация на веригата – веднъж добавен блок към нея не може да бъде променян, без да се промени всеки следващ блок във веригата, което теоретично е невъзможно, тъй като всеки участник в блокчейн мрежата пази локално копие на своето компютърно устройство и следи за промени, тоест може да се коригират нелегитимно извършени действия, хакване, променяне на информацията с цел печалба или нелегални действия.



Фиг.4. Извършаване на транзакция и запазването ѝ в даден блок на веригата

Определя се сложността, с която се копаят дадени блокове и се извършват транзакции с валути и се извършва така нареченото доказателство за работа или добиване, т.е. използва се компютърна мощ за решаване на математически проблеми. За да се извършва това копаене бързо, в експериментални условия този „проблем“ е сведен до максимална простота и бързина на намирането на решение, за да се „изкопае“ даден блок. В реалния свят се използват множество видео карти, системи върху чипове и ASIC и други устройства с огромна изчислителна мощ, за да е възможно да се изкопае дори един блок. Най-често потребители, които разполагат с персонален компютър с доста ниска ефективност, се обединяват с други в общи мрежи наречени „Mining Pools“. Това се прави за да се разпределят силите и да създаде възможност за изкопаване и получаване на „награда“ за този потребител под формата на частични такси - биткойн или друга криптовалута. Тоест търси се равновесие за отделните потребители, за да не се стига до там, че само големи и мощни корпорации да имат шанс да изкопават тези блокове (фигура 5).



Фиг. 5. Изкопаване на блок

При стартиране на програмата се зарежда прозорецът за конфигуриране на всички терминални устройства при създаване на нова топология или при редактиране на съществуваща вече такава. Трябва да се избере „Channel“ за работа, да се зададе обхват на терминала (в km) и максимален брой на свързаните крайни устройства.

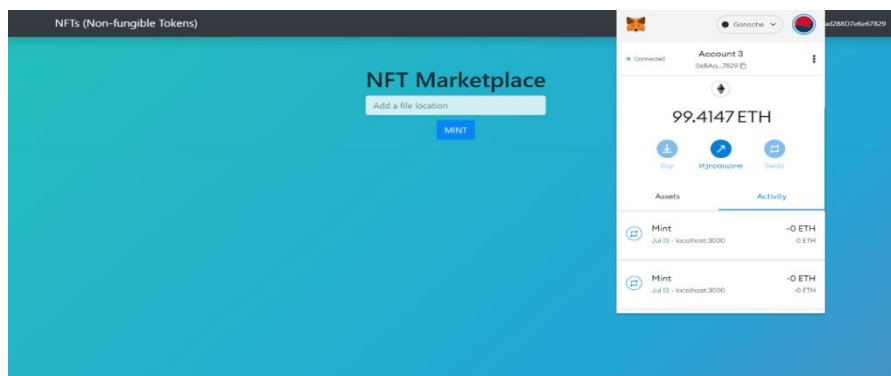
5. Графична среда за търгуване

Приложението е изработено със стандартни Node модули и притежава първоначален скелет на React, с впоследствие разработени умни договори с помощта на Solidity, добавени

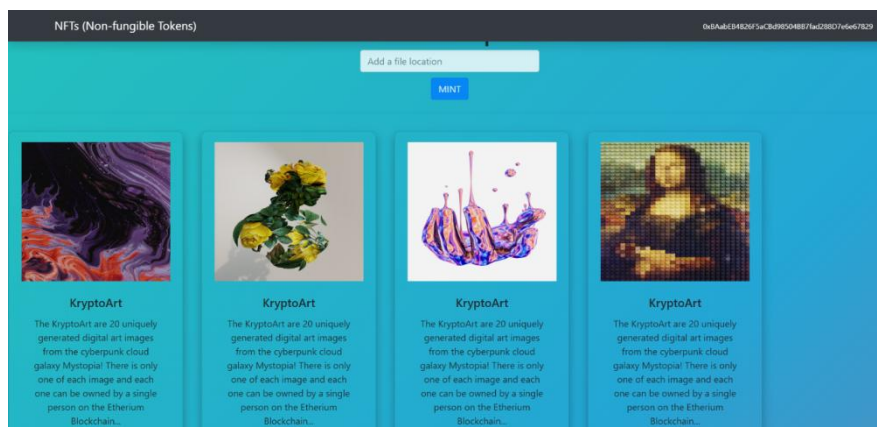
графични подобрения със CSS, javascript и т.н. Главната задача на приложението е да разработи подобрени договори със Solidity и да даде възможност на потребителите да търгуват с дигитално изкуство с помощта на крипто–портфейли анонимно и сигурно онлайн.

Опростена платформа за добиване на NFT (Non-fungible Tokens) единици, е показана на фигура 6. Те са уникални и принадлежат само на един единствен собственик и уникално идентифициране, като имат доказано един единствен притежател, който може да реши да търгува с тях с цел печалба или да създава/продава такива NF токени (фигура 7).

Използва се metamask, вързан към браузъра: крипто–портфейл за осъществяване на плащания/транзакции на GAS – валутата, използвана от Ethereum, за прикачване на умните договори. Този портфейл се свързва с Ganache, което представлява локална блокчейн мрежа за тестване.



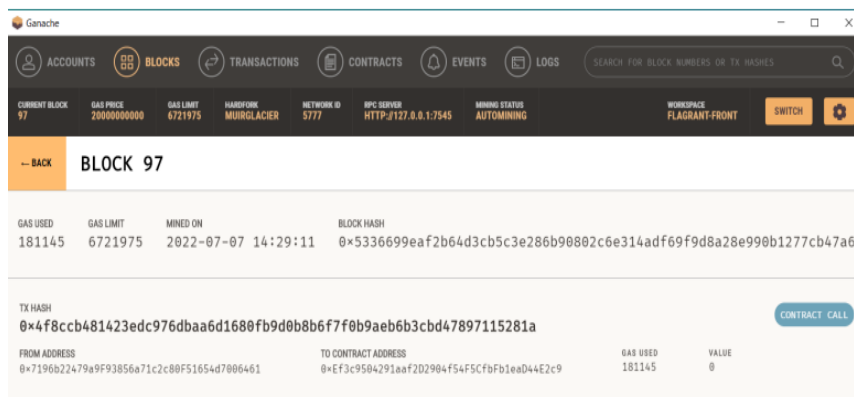
Фиг. 6. Платформа за добиване на NFT единици



Фиг. 7. Придобити NFT единици

За да се добие изображение, се поставя даден линк към изображение, след което се натиска бутона „Mint”. Произволен блок от мрежата е показан на фигура 8.

Може да се проследи номера на блока, кога е добит, колко Ethereum Gas е използван както и хеш–функцията на този блок. Там се намира адресът на договора както и адресът на потребителя, който е направил заявката. Има лимит за Ethereum Gas, за да се ограничат мащабни транзакции, които биха натоварили мрежата и екосистемата на Ethereum.



Фиг. 8. Подробен изглед на произволен блок от мрежата

Една възможна идея за допълнителна реализация в системата е да се създаде „ТУ- коин“, който да търси финансиране, за разработката на множество университетски проекти. А защо не и създаване на дигитално изкуство с логото на университета или сходна тематика с цел събиране на дарения, разплащане на университетски такси, гласуване на студентски съвет или за провеждане на сигурни онлайн референдуми с помощта на блокчейн технологията. Множество учени сочат блокчейн като бъдещето на електронното гласуване и развитието на демокрацията, тъй като всичко е с отворен код достъпно и видимо за всеки потребител, участник. Това са само част от идеите, които могат да развият един подобен по-машабен проект.

Този фиктивен „ТУ-коин“ е създаден в първото приложение, но няма никаква стойност, тъй като не е обществено търгуван и не е общодостъпен на пазара. В подобни случаи се организират обществени продажби с минимална цена на даден стартов коин, който често предполага възвръщане на актуални парични средства в световни валути, което връща себестойността на притежателите на тази криптовалута. Става дума за тези, които са инвестирали в тази валута. Това може да се приеме и като дарение за бъдещето на дадена валута. Притежателите могат да решат да търгуват с тази валута, когато нейната цена на пазара се увеличи или да я използват за разплащане или закупуване на стоки, които приемат тази валута.

7. Заключение

В тази статия е предложен софтуер за симулация, който позволява създаването на локална блокчейн система и търговия с криптовалута, служеща за целите на обучението.

Проведените тестове със системата еднозначно показват, че реализацията е успешна и че тя може да се използва по предназначение за целите на обучението.

Литература

- [1]. Gupta, S., & Sadoghi, M. (2021). Blockchain Transaction Processing. arXiv. https://doi.org/10.1007/978-3-319-77525-8_333
- [2]. Mirkute, A., Gunjalkar, A., Sawale, P. A., “Blockchain Transaction Processing and Modifying”, International Journal of Innovative Research in Science Engineering and Technology. 11. 4555-4559. 10.15680/IJIRSET.2022.1104087.
- [3]. Fan, F., Ventre, C., Basios, M., Kanhan, L., Martinez, D., Wu, F., “Cryptocurrency trading: a comprehensive survey. Financial Innovation”. 8. 10.1186/s40854-021-00321-6.

- [4]. Johnson , B., Co , S., Sun , T., Lim. C., Stjepanović , D., Saunders, J., Chan , G., "Cryptocurrency trading and its associations with gambling and mental health: A scoping review. Addictive Behaviors". 136. 107504. 10.1016/j.addbeh.2022.107504.
- [5]. Hewa , T., Hu, Y., Liyanage , M., Kanhare ,S. and Ylianttila , M., "Survey on Blockchain-Based Smart Contracts: Technical Aspects and Future Research," in IEEE Access, vol. 9, pp. 87643-87662, 2021, doi: 10.1109/ACCESS.2021.3068178.
- [6]. Alharby, M., Aldweesh, A. and Moorsel, A., "Blockchain-based Smart Contracts: A Systematic Mapping Study of Academic Research (2018)," 2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCB), 2018, pp. 1-6, doi: 10.1109/ICCB.2018.8756390.
- [7]. Frauenthaler, P., Sigwart, M., Spanring, C., Sober, M. and Schulte, S., "ETH Relay: A Cost-efficient Relay for Ethereum-based Blockchains," 2020 IEEE International Conference on Blockchain (Blockchain), 2020, pp. 204-213, doi: 10.1109/Blockchain50366.2020.00032.
- [8]. Dabboussi, D., Victor, F. and Prinz, W., "BCDM - A decision and operation model for blockchains," 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 2021, pp. 1-3, doi: 10.1109/ICBC51069.2021.9461146.
- [9]. Montes, J., Ramirez. C., Gutierrez, M. and Larios, V., "Smart Contracts for supply chain applicable to Smart Cities daily operations," 2019 IEEE International Smart Cities Conference (ISC2), 2019, pp. 565-570, doi: 10.1109/ISC246665.2019.9071650.
- [10]. Varsier, N., J. Schwoerer, "Capacity limits of LoRaWAN technology for smart metering applications", Proc. IEEE Int. Conf. Commun. (ICC, May 2017)

За контакти:

Гл. ас. д-р инж. Диян Желев Динев
 катедра „Софтуерни и Интернет технологии”
 Технически университет – Варна
 E-mail: diyandinev@tu-varna.bg



**СОФТУЕРНИ И ИНТЕРНЕТ
ТЕХНОЛОГИИ**



РАЗРАБОТВАНЕ НА ДВУСТАВЕН РОБОТ ЗА УЧЕБНИ ЦЕЛИ

Живко С. Жеков

Резюме: В статията е представен двуставен робот-манипулатор и неговата система за управление. Представени са сервозадвижванията на двете стави. Дадена е кинематичната схема на робота и решението на обратната кинематична задача. Показани са резултати от експериментални изследвания, потвърждаващи работоспособността на робота и системата за управление. Дадени са препоръки за подобряване на работата на разработения лабораторен робот.

Ключови думи: робот, манипулатор, управление, динамика, кинематика

DEVELOPMENT OF A TWO-JOINT ROBOT FOR EDUCATIONAL PURPOSES

Zhivko S. Zhekov

Abstract: The paper discusses the development of a two-joint robot manipulator and its control system. The servo drives of the two joints are presented. The kinematic diagram of the robot and the solution of the inverse kinematic problem are given. Results of experimental studies confirming the operability of the robot and the control system are shown. Recommendations are given for improving the performance of the developed laboratory robot.

Keywords: robot, manipulator, control, dynamics, kinematics

1. Въведение

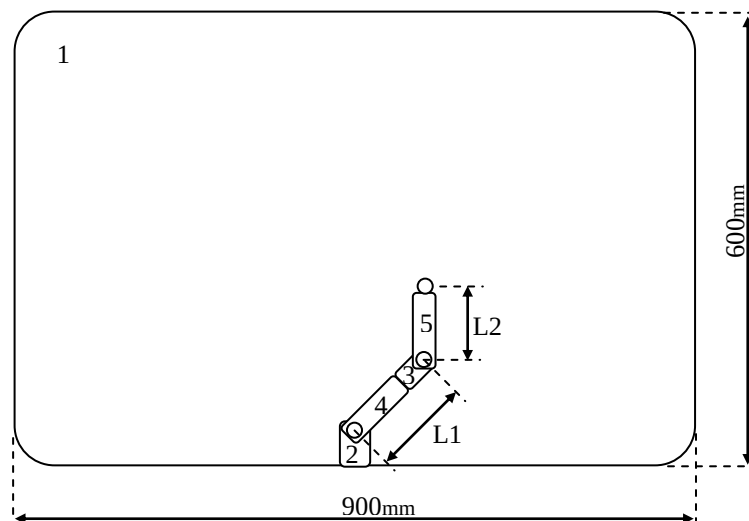
Статията представя разработен робот-манипулатор тип SCARA и неговата система за управление, за да може студентите чрез провеждане на лабораторни упражнения по съответните дисциплини да се запознаят със:

- кинематичната схема на работи от подобен тип;
- сервозадвижванията и тяхната настройка;
- разработването на управлението на кинематиката и динамиката на робота.

2. Описание на разработения робот и на системата за управление

Лабораторният робот-манипулатор се състои от две стави, завъртани от постоянноходкови сервозадвижвания Dynamixel MX-64T и Dynamixel AX-12A на фирма Robotis. Схемата на манипулатора е показана на фигура 1, където:

- 1. Работно поле;
- 2. Сервозадвижване 1 - модел MX-64T;
- 3. Сервозадвижване 2 - модел AX-12A;
- 4. Рамо 1 – $L_1=152\text{mm}$ (дължина на рамото и сервозадвижване 2);
- 5. Рамо 2 – $L_2=115\text{mm}$ (дължина на рамото и работния орган).

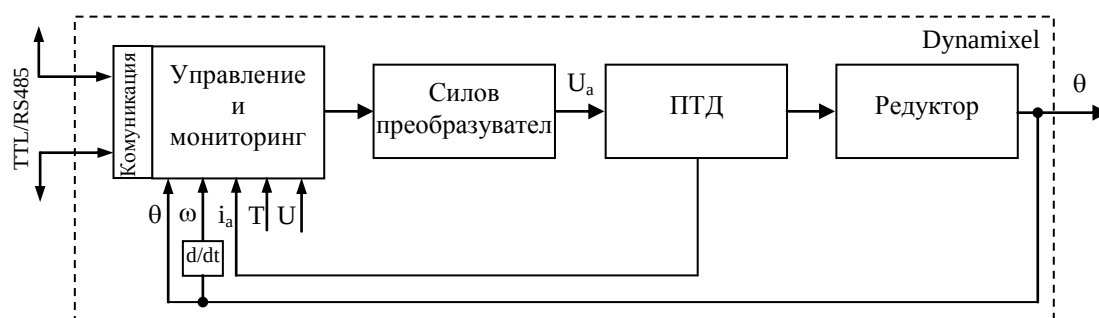


Фиг. 1. Схема на двуставния робот

Сервозадвиженията Dynamixel имат вграден микроконтролер, осъществяващ:

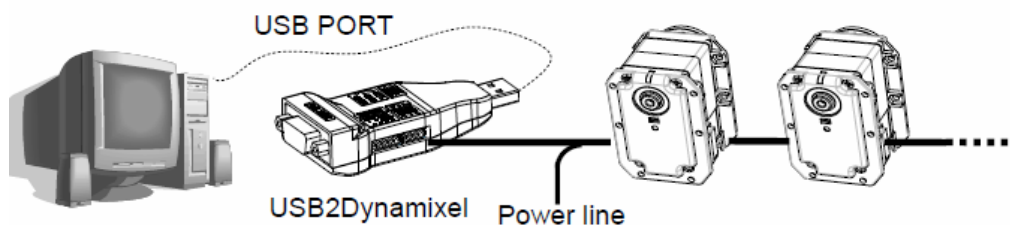
- управление по положение, скорост или момент;
- мониторинг на температурата, натоварването и захранващото напрежение;
- серийна комуникация с контролер или персонален компютър.

Параметрите на сервозадвиженията Dynamixel са представени в таблица 1, а блоковата им схема е показана на фигура 2.



Фиг. 2. Блокова схема на сервозадвижане Dynamixel

Схемата на системата за управление на манипулатора е показана на фигура 3. Заданията по положение за двете сервозадвижения се генерират посредством програмна среда Matlab, а връзката им с персоналния компютър се осъществява посредством преобразувател USB2Dynamixel. Сервозадвиженията използват мрежова топология, позволяваща свързването им едно след друго.

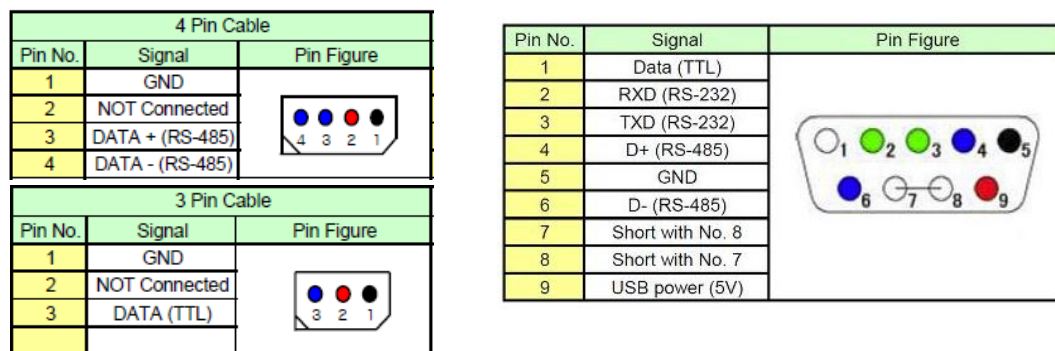


Фиг. 3. Схема на системата за управление на двуставния робот [1,2,3]

Табл.1. Основни характеристики на сервозадвижвания MX-64T и AX-12A [1,2]

модел	MX-64T	AX-12A
Общ вид		
Максимален момент	6.0Nm	1.5Nm
Ъглова скорост	63rpm	59rpm
Захранващо напрежение	10÷14.8V(номинално12V)	9÷12V(номинално11V)
Максимален ток	4.1A	1.5A
Размери	40.2x61.1x41mm	32x50x40mm
Маса	126g	54.6g
Резолуция	0.088°	0.29°
Диапазон на позициониране	0÷360°, [0÷4095]	0÷300°, [0÷1023]
Предавателно отношение	200:1	254:1
Вграден микроконтролер	STM32F103C8 72MHz, 32bit	ATMega8-16AU 6MHz, 8bit
Датчик за положение	абсолютен магнитен енкодер	потенциометър
Комуникационен протокол	TTL	TTL
Скорост на комуникацията	3mbps	1mbps
Регулатор по положение	ПИД регулатор	П регулатор с корекции

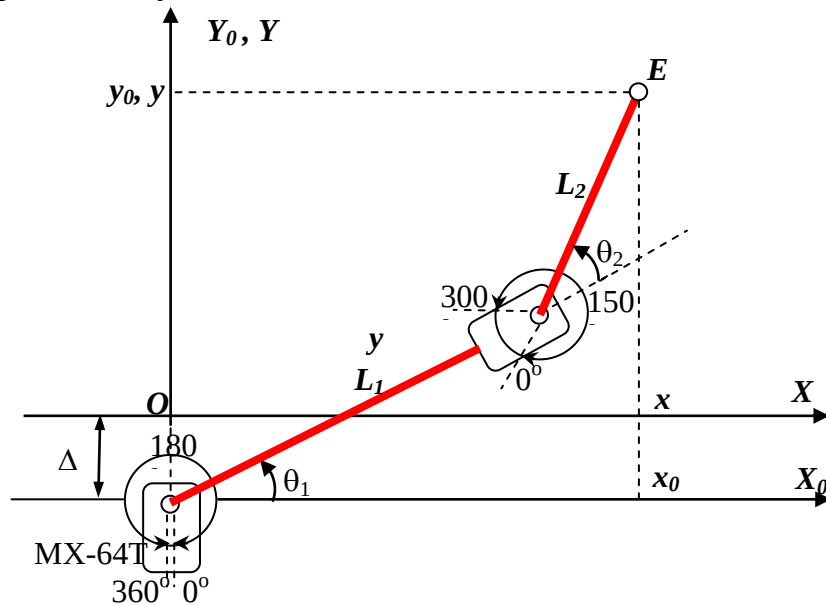
Преобразувателят USB2Dynamixel е устройство, осъществяващо комуникация между сервозадвижвания DYNAMIXEL и персонален компютър. Има възможност за преобразуване между USB и TTL/RS485/RS232 комуникация [3]. Изводите на преобразувателя са показани на фигура 4. В случая използваните сервозадвижвания са с TTL комуникация.



Фиг. 4. Изводи на преобразувател USB2Dynamixel [3]

Кинематичната схема на манипулатора е показана на фигура 5. X_0, Y_0 е кординатната система на манипулатора, а X, Y е кординатната система на работното поле, която е транслирана по ос Y в положителна посока спрямо първата кординатна система. Зададените кординати x и y на работния орган (т.Е) се преобразуват в кординати x_0, y_0 по следните зависимости: $x_0=x$, $y_0=y+\Delta y$. Посредством решаването на обратната кинематична задача се

получават ъглите на завъртане на двете стави θ_2 и θ_1 , съгласно зависимости (1) и (2), във функция от координати x_0, y_0 .



Фиг. 5. Кинематична схема на двуставния робот

$$\theta_2 = \pm 2 \arctan \sqrt{\frac{(L_1 + L_2)^2 - (x_0^2 + y_0^2)}{(x_0^2 + y_0^2) - (L_1 - L_2)^2}} \quad (1)$$

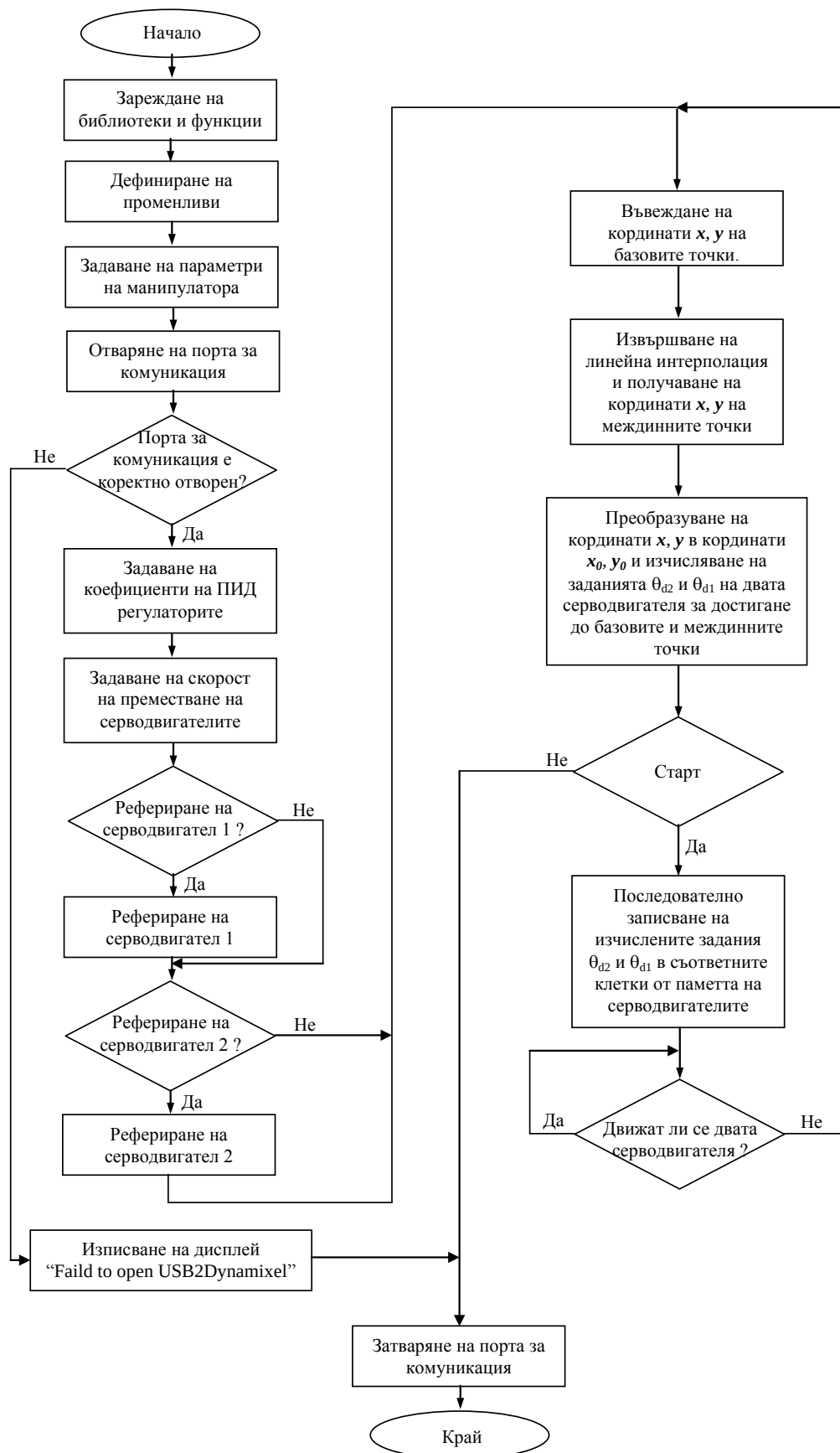
$$\theta_1 = \arctan \frac{y_0}{x_0} - \arctan \frac{L_2 \sin \theta_2}{L_1 + L_2 \sin \theta_2} \quad (2)$$

Заданията по положение на двете сервозадвижвания θ_{d2} и θ_{d1} се намират по зависимости (3) и (4) като получените по-горе ъгли се преобразуват от радиани в градуси, допълват с определен ъгъл понеже сервозадвижване MX-64T е завъртяно спрямо ос X_0 на -90° , а сервозадвижване AX-12A е завъртяно спрямо първата става на -150° и се премасхабират в необходимия за сервозадвижванията диапазон, съответно $[0 \div 4095]$ за MX-64T и $[0 \div 1023]$ за AX-12A.

$$\theta_{d2} = \left(\theta_2 \frac{180}{\pi} + 150 \right) \frac{1023}{300} \quad (3)$$

$$\theta_{d1} = \left(\theta_1 \frac{180}{\pi} + 90 \right) \frac{4095}{360} \quad (4)$$

Опростена блокова схема на алгоритъма за управление на двуставния робот, реализиран посредством програмна среда Matlab, е представен на фигура 6. В разработената програма се въвеждат координати на базови точки, извършва се линейна интерполация между тези точки и така генерираните координати се преобразуват посредством зависимости (1-4) в задания за двете сервозадвижвания.



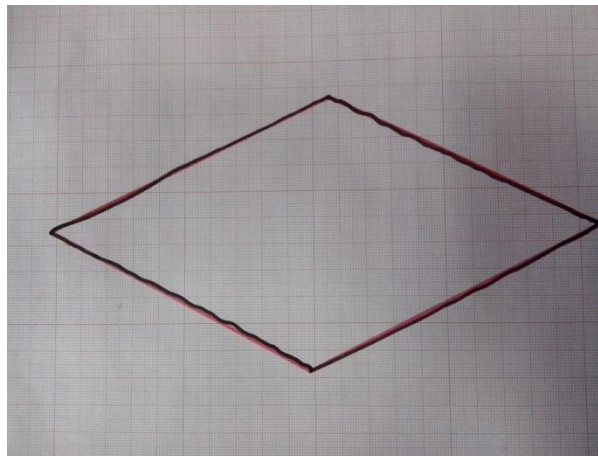
Фиг. 6. Опростена блокова схема на алгоритъма за управление на двуставния робот

3. Експериментални резултати

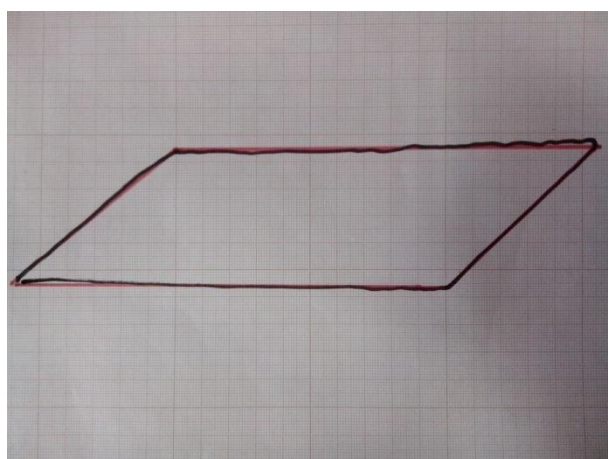
Общият вид на разработения двуставен робот е показан на фигура 7. На фигури 8, 9, 10 са представени резултати от експерименталните изследвания. Роботът е снабден с писец. Зададени са координатите на върховете на определени четириъгълни геометрични фигури, а страните им са изчертани с червени линии. С черните линии е показано изчертаването от робота на зададената фигура.



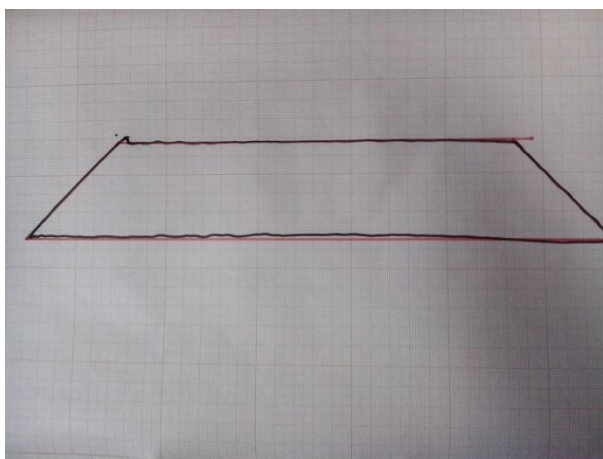
Фиг. 7. Общ вид на двуставния робот



Фиг. 8. Изчертаване на ромб



Фиг. 9. Изчертаване на успоредник



Фиг. 10. Изчертаване на трапец

4. Заключение

Наблюдава се сравнително добро съвпадение между заданието и изпълнението, като отклоненията се дължат основно на:

- недостатъчно прецизна изработка на механичната част на двуставния робот;
- налични луфтове в редукторите на сервозадвижванията;
- недостатъчно точно закрепване на работния орган – писец;
- недостатъчно точно задаване на параметрите на манипулатора в системата за управление;
- недостатъчно точно ориентиране на координатната система на манипулатора спрямо координатната система на работното поле.

Подобряване на работата на двуставния робот без преработка на механичната част и замяна на сервозадвижванията може да се осъществи като:

- по-точно се определят параметрите на механичната част на робота и координатните системи и се заложат в системата за управление;
- бъде извършена донастройка на ПИД регулаторите на сервозадвижванията.

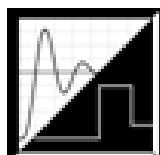
Разработен е двуставен робот-манипулатор и неговата система за управление. Представени са резултати от експериментални изследвания, потвърждаващи работоспособността на робота и системата за управление. Дадени са препоръки за подобряване на работата на разработения лабораторен робот, който може да се използва по дисциплини „Задвижвания на роботи“ и „Промислени роботи“ от учебните планове на специалности „АРУКС“ и „РМ“ на катедра Автоматизация на производството.

Литература

- [1]. AX-12 Manual v1.10.00, ROBOTIS.
- [2]. MX-64 Manual, ROBOTIS.
- [3]. USB2Dynamixel Manual, ROBOTIS

За контакти:

д-р инж. Живко С. Жеков
катедра „Автоматизация на производството“
Технически университет - Варна
E-mail: fitter@abv.bg



HEALTHCARE SYSTEMS BASED ON BLOCKCHAIN

Magdalena Boycheva

Abstract: The healthcare sector can potentially benefit from blockchain technology by making information systems patient-centric and facilitating the sharing of health data securely and efficiently. Blockchain can be applied to the medical industry in a variety of ways, leading to cost reductions and new ways for patients to access healthcare. Technology has the potential to transform healthcare, putting the patient at the center of the healthcare ecosystem and increasing the security, privacy and interoperability of healthcare data. This paper presents the applications in the healthcare sector, the interested parties and their features. This will lead to significant improvements to identify existing and emerging challenges and potential research opportunities.

Keywords: Blockchain, Healthcare, Features, Interoperability

1. Introduction

The technology that has had the biggest impact on our way of life in the last decade is Blockchain. Blockchain is a pervasive technology that integrates across platforms and hardware around the world. One area where blockchain technology has huge potential is healthcare. Data management in healthcare has attracted enormous attention in recent years due to its high potential to provide more accurate and cost-effective patient care. A problem facing the healthcare industry is interoperability, inaccessible medical records and the lack of comprehensive and reliable population health data. Problems ranging from inefficient pharmaceutical supply chains and counterfeiting, mishandling of patient data to poor communication between different stakeholders, create difficulties on a daily basis.

Today, healthcare needs quality healthcare facilities supported by advanced and newer technologies. Various interested parties (practitioners, medical professionals, hospitals, patients, insurers, etc.) need to organize, access and share health records without any modifications in a more secure and interoperable way. The origin of the data is also essential to prove the authenticity of the records. To deliver effective treatment with accurate diagnoses and provide cost-effective care, it is necessary to improve the security of data sharing between healthcare providers. Blockchain technology in healthcare can improve the productivity, security and transparency of medical data sharing in the healthcare system [1].

This paper examines the concept of blockchain technology, various types of applications and technical features.

2. Essence of Blockchain

The concept of blockchain first appeared in 2008, in the publication of Satoshi Nakamoto, as a means of realizing virtual currency. It is based on already known technologies (distributed databases, public-private key, Merkel tree hashing, consensus mechanism, encryption), the combination of which serves to solve the double-spending problem that existed in previous virtual currency proposals [2]. A blockchain is a distributed database maintained by nodes in a Peer-to-Peer (P2P) network. Blockchain features, including decentralization, immutability, transparency, and auditing, make transactions more secure and protected from wrongful acts. Blockchain can be defined as a chain of blocks. Blocks of data are linked together by a hash (a digital signature of random letters and numbers) to form a "chain" of data that contains the complete history of the transaction and makes it tamper-resistant [3]. The data on the blockchain is also protected by

cryptography (advanced encryption) so that participants can trust that blocks of data are authenticated and verifiable.

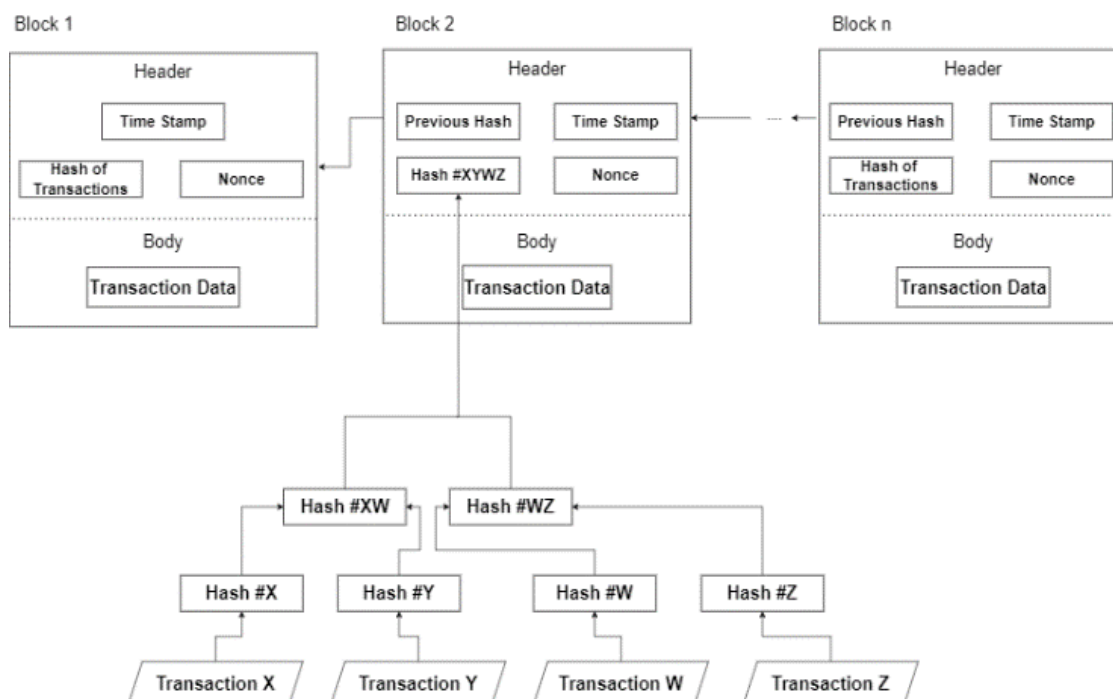


Fig. 1. General structure of a blockchain

The block consists of a header and a body (Fig. 1). The block header contains the hash of the previous block, the time and date the block was created, a Nonce, and a Merkle tree, which reduces the effort required to verify transactions. Nonce is the number that serves as proof of work done and as a mechanism to regulate the time to create a new block [2]. In a blockchain network, transactions are validated by a community of nodes and then recorded in a block. The block body consists of a transaction counter and a list. The maximum number of transactions a block can contain depends on the size of the block and the size of each transaction. Transactions in a block are hashed to generate a fixed-length hash output that is added to the block header. After the first block is created, each subsequent valid block must contain the hash output of the previous block's header. The header hash of the previous block, which is contained in each block, serves as a chain that connects each valid block to those before it. Thus, by linking each block to the previous ones, a chain of blocks called a blockchain is created [4]. If any of the transactions in a block change, the corresponding hash output will also change, breaking the chain to subsequent blocks in the blockchain. Any change in the content of a block is easily detected on the network. For this reason, once a transaction is added to a block and linked to the blockchain, it cannot be changed or reversed. Immutability is an important property that ensures that records, once created, cannot be retrieved or modified.

3. Blockchain-based healthcare applications

To implement blockchain in healthcare, many approaches and a range of different technologies can be used to achieve the desired result. In [5], the workflow of blockchain-based healthcare applications is reviewed. The workflow consists of four main layers, including raw healthcare data, blockchain technology, healthcare application and stakeholders.

Based on the reviewed publications [6,7,8,9], the existing systems in health care can be divided according to the following features: Application areas of health care, interested parties and technical features.

Healthcare applications include data management, supply chain management, health insurance claims, and the Internet of Things (IoT) in healthcare and medical devices. Data management includes global data sharing for research, management and storage (eg. cloud applications) and electronic health record (EHR). The need to use blockchain for data management arises due to a number of its features, including transparency, traceability, immutability, privacy and security from tampering. Supply chain management applications serve to identify, track and verify pharmaceutical supplies, as well as track the entire life cycle of healthcare devices. This is a top priority for the industry, especially in emerging markets where counterfeit prescription drugs cause tens of thousands of deaths annually [10]. Applications that improve insurance claims processes through the use of "smart contracts", the proper processing and management of payments. The use of blockchain technology will prevent the creation of false claims by patients and/or providers by more securely storing and tracking all transactions. IoT applications in healthcare and medical services that accelerate clinical or biomedical research. IoT and blockchain technology find applications in electronic medical record management, remote real-time patient monitoring, drug traceability, and combating infectious diseases such as the COVID-19 pandemic.

Blockchain as a decentralized technology allows multiple stakeholders to benefit from healthcare applications, such as researchers, business users, and patients. There are several issues that the healthcare sector faces every day. Even if some solutions have proven feasible for practical implementation, they require reduction of data size and operational costs, as well as better protection of personal information to maintain privacy and security. Based on the analysis, it is concluded that healthcare systems based on blockchain should have the following features:

- Medical data security;
- Data confidentiality;
- Access Control;
- Interoperability;
- Scalability;
- Lack of standardization;
- Key management and security.

A) Medical data security

Medical data is a fundamental element in healthcare and proper management and safe retrieval are crucial to its security. At present, a large part is stored and controlled by different service providers. This type of centralized storage leads to uncertainty in their management. With the development of electronic information technology, electronic medical records (EMRs) are a common way to store patient data, and it is extremely important to ensure their security and confidentiality [11]. The challenge facing the healthcare industry is the ability to record and store information easily and share it securely across different applications and systems. The application of blockchain technology in the healthcare industry can improve information security management. Blockchain is verified through a consensus system and stored across many nodes, making DDoS cyberattacks and record tampering extremely difficult. Blockchain is able to ensure the protection of medical data by using cryptographic algorithms acting as firewalls against unauthorized attacks.

In [12], the authors propose a blockchain-based solution for medical record processing using the Hyperledger platform. For this purpose, a framework called "Medichain" has been developed, which contains all the basic functionalities of a blockchain to protect patient data.

In 2020, the Veridat [13] application was created in the US, which uses the Bitcoin SV blockchain. The cloud-based service supports research integrity and data authenticity for the pharmaceutical industry. The application records key events within a workflow.

B) Data confidentiality

The protection of confidential information is one of the priorities of specialized hospitals and clinics. Currently, medical data use the cloud for storage, but their security and privacy are not guaranteed. A major disadvantage of the centralized model is the difficulty in preserving user privacy. Privacy threats include unauthorized access to important information, such as patient identity and disease information. Blockchain-based cloud technology can solve a number of problems related to the healthcare industry and data vulnerability [14]. Useful in this system are smart contracts that ensure transparency, precision and trust of the transactions performed. The record stored in the system is accessible only by trusted parties. Although there are many appropriate policies, guidelines, and compliance requirements to protect health information, breach of privacy and security remain key issues for eHealth systems [11]. Storing large volumes of data without adversely affecting the performance of the blockchain network also remains a challenge.

Chen et al. [15] propose a blockchain-based privacy-preserving scheme for sharing medical data between medical institutions and users. The system uses the K-anonymity technique for preprocessing medical data. To ensure security and privacy, medical data is shared using keyword searchable encryption.

In another study, Sapra et al. [16] propose a new model that integrates IoT networks with blockchain to address potential threats. Smart contracts play an important role in the integration process and are used for processing, authorization and access control in data management.

C) Access Control

Access control is a critical part of system security, limiting the actions that users can perform on resources. Current access control mechanisms face a number of problems due to third-party interference, privacy, and data security. Medical institutions are lax about data management and access rights are not clear. Blockchain technology uses smart contract functions to facilitate user authentication and authorization.

Cruz et al. [17] present the RBAC-SC framework for role-based access control using a smart contract. In RBAC, users are associated with roles and roles are associated with services.

The authors [18] present a blockchain-based access control architecture. The role-based MCPS model and the task-based access control model combine to classify users according to their roles.

D) Interoperability

Healthcare interoperability focuses on the exchange of data between different hospital systems. Exchange between different institutions can be challenging and requires considerable cooperation between the entities involved. This problem creates interference in the effective sharing of information. Lack of interoperability can lead to incomplete understanding of population health needs, leading to poorer outcomes and higher costs [19]. One of the key challenges is that healthcare organizations are at different levels reflecting quality, management mechanisms and use of standards. Interoperability is essential, an example of this being the need for a more robust infrastructure during the Covid pandemic that could help streamline patient-provider communication and facilitate the flow of information. Some organizations tend to use FHIR (Fast Healthcare Interoperability Resources), while others use the CDA (Clinical Document Architecture) standard for data exchange. These different standards reduce interoperability.

If a unified blockchain-based solution is used among healthcare centers, this problem can be avoided. However, if healthcare centers decide to use different blockchain-based solutions with different platforms, it will be very difficult to make them interoperable. The added security and the ability to establish trust between entities are two reasons why this technology can help solve the problem better than existing technologies today. The goal of healthcare interoperability is to

facilitate the exchange of health-related information, such as the electronic health record, between healthcare providers and patients so that data can be easily shared between different hospital systems.

In [20], one decentralized application appXchain is presented, whose architecture uses the interface layer to delegate requests from one blockchain network to another. The concept of verifier nodes is adopted, which are internal computing nodes for each network with the ability to communicate with other nodes on external networks to ensure the validity of responses and shared data.

E) Scalability

Scalability is a key issue in integrating blockchain and IoT. Due to the large volume of clinical data, scalability issues arise as blockchain-based solutions have data size limitations. The block size limit in blockchain aims to reduce performance costs and prevent DDoS attacks. This compromises the scalability needed to store the massive amount of patient health data, the size of which is difficult to manipulate. On the other hand, if the blockchain is designed to handle large volumes of data, it will incur additional operational costs due to performance. Blockchain requires individual nodes to process each transaction across the entire network, which provides security and verification of the system, but limits scalability [21]. Existing solutions require the blockchain to generate a large amount of transactions to be processed and connected to the network, resulting in a degradation of overall performance. Another case to consider is the possibility that the speed of data synchronization in the system will be lower than the speed of transactions required to be recorded in the blockchain system. An example of this is the validation mechanism in the current setup of the Ethereum blockchain platform, which requires all nodes on the network to participate. This results in significant processing delays if the data load is heavy.

F) Lack of standardization

The adoption of common standards for data collection, exchange and storage is necessary in any healthcare infrastructure. Health-related data is stored in isolation and in different standards, making it difficult to share. Different organizations do not work with a single standard, but maintain their own versions. Implementing blockchain technology in the healthcare sector would take time and effort due to the need for international certified standardization. Deciding the size, format and type of data that can be stored on the blockchain will benefit the standard license [22]. The right standards, used at the right time, can ensure interoperability, generate trust and help ease the use of technology.

G) Key management and security

The data stored in the blockchain ledger must be secured through various cryptographic processes that require the use of private and public keys. Due to the large amount of confidential data in healthcare systems, certain security mechanisms must be built in. Most cryptographic processes require keys to enable these processes because blockchain data is publicly stored and shared by all participants. Blockchain is an emerging technology, but little research has been done on key management and security. The technology allows healthcare providers to create an integrated health record system in which patients own a private key through which they control access to their data [23].

In [24], a BSN key backup and recovery scheme is proposed. In the scheme, each block of the blockchain can be encrypted with a different key, with lower storage costs and high performance, which greatly improves data security and privacy.

4. Conclusion

A key objective of the paper is to identify blockchain technology use cases in healthcare, challenges and limitations. Various cases of the application of technology in solving some of the most critical and challenging problems in the healthcare industry are examined. Blockchain technology in healthcare is in its infancy, but the number of proposed solutions is growing exponentially. The healthcare system as an organization faces a number of issues corresponding to data security and privacy, protection from unauthorized access, supply chain management and secure monitoring of eHealth. Health data is inaccessible and non-standardized across systems. They are pulled from various sources and stored in centralized IT systems, making management and sharing a challenging task. Another issue facing the healthcare industry is interoperability, inaccessible medical records, and the lack of comprehensive and reliable population health data. Blockchain's potential in healthcare is to overcome challenges related to data security, privacy, sharing and storage. One of the requirements for the healthcare industry is interoperability. In conclusion, it can be summarized that the management of medical records is receiving more and more scientific attention. Lack of standards, privacy, cost and trust between stakeholders remain a major problem in interoperability in blockchain technology. Scalability and interoperability issues are the main focus for future research in the area. Although blockchain technology offers promising features, further research is needed to better understand and efficiently and securely develop applications in the healthcare sector.

References

- [1]. Haleem, A., Javaid, M., Singh, R., Suman, R., Rab, S. Blockchain technology applications in healthcare: An overview // *International Journal of Intelligent Networks*, vol. 2, 2021, pp. 130-139, <https://doi.org/10.1016/j.ijin.2021.09.005>
- [2]. Huliyan, A. General taxonomy of blockchain technologies // *Computer Science and Technologies*, vol. 1, 2020, Technical University of Varna
- [3]. Liang, YC. Blockchain for Dynamic Spectrum Management.// *Dynamic Spectrum Management, Signals and Communication Technology*, Springer, Singapore, https://doi.org/10.1007/978-981-15-0776-2_5
- [4]. Agbo, C.C., Mahmoud, Q.H., Eklund, J.M. Blockchain Technology in Healthcare: A Systematic Review. // *Healthcare* 2019. <https://doi.org/10.3390/healthcare7020056>
- [5]. Khezzr, S., Moniruzzaman, M., Yassine, A., Benlamri, R. Blockchain Technology in Healthcare: A Comprehensive Review and Directions for Future Research. // *Appl. Sci.* 2019, 9, 1736. <https://doi.org/10.3390/app9091736>
- [6]. Hussien, H.M., Yasin, S.M., Udzir, S.N.I. et al. A Systematic Review for Enabling of Develop a Blockchain Technology in Healthcare Application: Taxonomy, Substantially Analysis, Motivations, Challenges, Recommendations and Future Direction. // *J Med Syst* 43, 320, 2019. <https://doi.org/10.1007/s10916-019-1445-8>
- [7]. Saeed, H., Malik, H., Bashir, U., Ahmad, A., Riaz, S., Ilyas, M., et al. Blockchain technology in healthcare: A systematic review. // *PLoS ONE*, 2022. <https://doi.org/10.1371/journal.pone.0266462>
- [8]. Azbeg, K., Ouchetto, O., Andaloussi, S., Fetjah, L. A Taxonomic Review of the Use of IoT and Blockchain in Healthcare Applications. // *IRBM*, 2021. <https://doi.org/10.1016/j.irbm.2021.05.003>
- [9]. Justinia, T., Blockchain Technologies: Opportunities for Solving Real-World Problems in Healthcare and Biomedical Sciences. // *Acta Inform Med*, 2019, doi: 10.5455/aim.2019.27.284-291

- [10]. 5 blockchain healthcare use cases in digital health, <https://stlpartners.com/articles/digital-health/5-blockchain-healthcare-use-cases/> (last visited 28.10.2022)
- [11]. Shi, S., He, D., Li, L., Kumar, N., Khan, M., Choo, K-K. Applications of blockchain in ensuring the security and privacy of electronic health record systems: A survey. // *Computers & Security*, vol. 97, 2020, doi: 10.1016/j.cose.2020.101966
- [12]. Johari, R., Kumar, V., Gupta, K., Vidyarthi, K. BLOSUM: BLOckchain technology for Security Of Medical records. // *ICT Express*, vol. 8, 2022, <https://doi.org/10.1016/j.icte.2021.06.002>
- [13]. Veridat, <https://www.veridat.io> (last visited 28.10.2022)
- [14]. Fatima, N., Agarwal, P., Sohail, S.S. Security and Privacy Issues of Blockchain Technology in Health Care—A Review.// *ICT Analysis and Applications. Lecture Notes in Networks and Systems*, vol 314, Springer, Singapore. https://doi.org/10.1007/978-981-16-5655-2_18
- [15]. Chen, Y., Meng, L., Zhou, H., Xue, G. A Blockchain-Based Medical Data Sharing Mechanism with Attribute-Based Access Control and Privacy Protection. // *Wireless Communications and Mobile Computing*, vol. 2021, pp. 12, <https://doi.org/10.1155/2021/6685762>
- [16]. Sapra, R., Dhaliwal, P. Chapter 19 - A blockchain solution for the privacy of patients' medical data, *Machine Learning, Big Data, and IoT for Medical Informatics*. // Academic Press, 2021, pp. 327-347.
- [17]. Cruz, J., Kaji, Y., Yanai N. RBAC-SC: Role-Based Access Control Using Smart Contract. // *IEEE Access*, vol. 6, pp. 12240-12251, 2018, doi: 10.1109/ACCESS.2018.2812844
- [18]. Chen, F., Huang, J., Wang, C., Tang, Y., Huang, C., Xie, D., Wang, T., Zhao, C. Data Access Control Based on Blockchain in Medical Cyber Physical Systems. // *Security and Communication Networks*, vol. 2021, 2021, pp.14, <https://doi.org/10.1155/2021/3395537>
- [19]. Why is interoperability in healthcare important?. <https://www.ibm.com/id-en/topics/interoperability-in-healthcare> (last visited 28.10.2022)
- [20]. Madine, M., Salah, K., Jayaraman, R., Al-Hammadi, Y., Arshad, J., Yaqoob, I. appXchain: Application-Level Interoperability for Blockchain Networks. // *IEEE Access*, vol. 9, 2021, pp. 87777-87791, doi: 10.1109/ACCESS.2021.3089603.
- [21]. Musamih, A. , et al. A Blockchain-Based Approach for Drug Traceability in Healthcare Supply Chain. // *IEEE Access*, vol. 9, 2021, pp. 9728-9743, doi: 10.1109/ACCESS.2021.3049920
- [22]. Hussien, H., Yasin, S., Udzir, N., Ninggal, M., Salman, S. Blockchain technology in the healthcare industry: Trends and opportunities. // *Journal of Industrial Information Integration*, vol. 22, 2021, <https://doi.org/10.1016/j.jii.2021.100217>.
- [23]. Attaran, M. Blockchain technology in healthcare: Challenges and opportunities. // *International Journal of Healthcare Management*, vol. 15, 2022, pp. 70-83, <https://doi.org/10.1080/20479700.2020.1843887>
- [24]. Zhao, H., Bai, P., Peng, Y., Xu, R. Efficient key management scheme for health blockchain. // *Special Issue: Internet of Things and Intelligent Devices and Services*, vol. 3, 2018, pp. 114-118, <https://doi.org/10.1049/trit.2018.0014>

For contacts:

Eng. Magdalena Boycheva
Computer Science and Engineering Department
Technical University of Varna
E-mail: m.boycheva@tu-varna.bg

РАЗВИТИЕ НА ТЕХНОЛОГИИТЕ С ИЗКУСТВЕН ИНТЕЛЕКТ В АВТОНОМНИТЕ ПРЕВОЗНИ СРЕДСТВА

Мариета Йорданова

Резюме: Транспортът е важна част за човешкото общество и е отговорен за развитието на цивилизациите. Той трябва да отговаря на изискванията за пътуване на хора и транспорт на стоки като се отличава с ефективност, безопасност и скорост. Автономното превозно средство може да се управлява без намеса на човешки водач. То събира информация за пътната обстановка и комуникира с участниците на движението чрез комуникация, която може да бъде постигната чрез усъвършенствани системи за изкуствен интелект и машинно обучение. Тъй като автономните превозни средства са без шофьор, те трябва да вземат решения сами. Изкуственият интелект помага на превозните средства да вземат свои собствени решения, като откриване на препятствия и извънредни ситуации. В настоящата статия са представени тенденциите в приложението на технологиите с изкуствен интелект в автономните превозни средства.

Ключови думи: автономни превозни средства, изкуствен интелект, машинно обучение

Development of artificial intelligence technologies in autonomous vehicles

Marieta Yordanova

Abstract: Transportation is an important part of human society and is responsible for the development of civilizations. It must meet the requirements for the travel of people and the transport of goods, being distinguished by efficiency, safety and speed. An autonomous vehicle can be driven without the intervention of a human driver. It collects information about the traffic situation and communicates with road users through communication that can be achieved through advanced artificial intelligence and machine learning systems. Since autonomous vehicles are driverless, they must make decisions on their own. Artificial intelligence helps vehicles make their own decisions, such as detecting obstacles and emergency situations. This article presents trends in the application of artificial intelligence technologies in autonomous vehicles.

Keywords: Autonomous driving, artificial intelligence, machine learning

1. Увод

Интелигентните транспортни системи (ИТС) трансформират големи транспортни мрежи чрез интегрирането на съвременни комуникационни и компютърни технологии в транспортната инфраструктура. Технологиите за ИТС подобряват безопасността и мобилността на транспортната мрежа чрез усъвършенствани приложения като електронно събиране на пътни такси, системи за навигация в автомобила и за избягване на сблъсъци, усъвършенствани системи за управление на трафика и за информация за пътниците [1].

Терминът ИТС обхваща набор от приложения, които използват съвременни комуникационни и компютърни технологии в транспортната инфраструктура. Основната функция на ИТС е потребителите да получават цялостно удовлетворение от по-чиста, по-безопасна и по-ефективна транспортна система. Системите, от които се състоят ИТС, в реално време събират, обработват и интегрират информация, която може да се използва за планиране на пътуването, динамично управление на трафика в определен район, докладване на данни от логистичен оператор до клиента и показване на трафик събития в географски информационни системи (Geographic Information Systems – GIS). Тъй като ИТС допринася за координирана система, политически и законодателни инициативи в Европа все повече обръщат внимание на тези системи. Европейската комисия установи правната рамка за

ускоряване внедряването на иновативните транспортни технологии в цяла Европа и поиска от европейските организации по стандартизация да разработят и приемат европейски стандарти в подкрепа на тази правна рамка. В тази област има значителна активност от организациите за стандартизация CEN, CENELEC, ETSI [5].

Технологиите, използвани в интелигентната транспортна система, са навигация на автомобили, сигнали за управление на трафика, автоматично разпознаване на номера или камери за скорост, приложения за видеонаблюдение, системи за откриване на спряло превозно средство или автоматично откриване на инциденти. Самоуправляващите се автомобили, известни също като автономни автомобили, са интелигентни транспортни системи, които могат да се използват за безопасно управление без човешка намеса. Те се управляват чрез методи, базирани на изкуствен интелект. Могат да се използват за научаване на цялата сензорна информация, за да се контролира превозното средство и да се поддържат различни задачи за автономно шофиране [5].

2. Класификация на автономните превозни средства

В класификацията въз основа на степента на автономност на превозните средства съществуват различни нива (фигура 1). През 2013 г. Националната администрация за безопасност на движението по пътищата (NHTSA) прави първата класификация, а през 2016 г. Общество на автомобилните инженери (SAE) представят класификация от 6 нива на автоматизация, която се определя като международен стандарт за всички превозни средства [2].

- **Ниво 0** – При това ниво всички контролни операции на превозните средства като пилотирането, спирането, ускорението, забавянето и аварийното спиране се изпълняват от водача. Превозните средства са с основни системи за предупреждение като температура на охлаждащата течност, налягане на маслото и т.н.
- **Ниво 1** – Това ниво определя добавянето на някои специфични контролни функции, които да помогнат на водача при шофиране, като използва автоматизирани системи за поддържане на лентата (Longitudinal Knowledge Assessment – LKA), адаптивният контрол на круиза (Adaptive Cruise Control – ACC) и за регулиране на скоростта. Всички автоматизирани системи работят независимо и все пак изискват някаква информация от водача.
- **Ниво 2** – Превозните средства при ниво 2 са частично автоматизирани, при което водачът има контрол над превозното средство по отношение на управление, ускоряване/забавяне и спиране, но водачът трябва да наблюдава шофирането и трябва да е готов да поеме контрола по всяко време, в случай че автоматизираните системи не могат да изпълнят правилно. Тези системи включват управление и ускоряване/забавяне едновременно. Автоматизирани автомобили от ниво 2 се предлагат на пазара и включват две или повече комбинирани автоматизирани функции.
- **Ниво 3** – При това ниво автоматизираната система за шофиране (Automated Driving System – ADS) управлява всички функции на шофиране в рамките на ограничена област на работа. При ограничаване на самоуправлението автомобилът е в пълен контрол и предупреждава водача да поеме отново контрола в ситуации, които изискват помощ от водача.
- **Ниво 4** – При това ниво се управляват всички функции на шофиране, но водачът може да поеме контрола при необичайни условия на околната среда. Превозните средства на това ниво са в състояние да се справят с незабавна реакция като аварийно спиране и да маневрират безопасно „състояние на минимален риск“ в случай на неизправност.

- **Ниво 5** – При това ниво превозните средства са напълно автоматизирани. ADS се справят с всички задачи за динамично шофиране, които могат да се управляват от човешки водач на превозно средство от ниво 0.

3. Развитието на автономните автомобили

Автономните автомобили на Waymo – новото име на проекта за самоуправляващи се автомобили на Google е Waymo LLC. Това е американска компания за технологии за автономно шофиране със седалище в Маунтин Вю, Калифорния и е дъщерно дружество на Alphabet Inc, компанията „майка“ на Google.

Първото разработване на технологията за самоуправление на Google е през 2009 г. в Google X (полусекретна организация за изследване и развитие), ръководена от Сергей Брин. През 2015 г. се провежда първото проучване на автономните автомобили със Firefly и се осъществява първото автономно шофиране по обществени места в Остин, Тексас като до края на 2013 г. са приети закони, позволяващи използване на автономните автомобили в Невада, Флорида, Калифорния, Мичиган и Тексас. Компанията Waymo става независима и е част от Alphabet като компания за технологии за автономно шофиране от 2016 г. През 2017 г. Waymo представя модифицирана версия на хибриден миниван Chrysler Pacifica 2017 г. В същата година се стартира програма за ранни водачи на Waymo, при която публично се изпитват автономните превозни средства от жителите на Финикс, Аризона. Първата в света комерсиална услуга за автономно пътуване е с Waymo One във Финикс през 2018 г. В района на Metro Phoenix се предлагат напълно автономни пътувания за избрани водачи на Waymo One през 2019 г. Година по-късно стартира Waymo Via за преместване на стоки и разширяване на тестовите в Ню Мексико и Тексас. Проведени са пилотни програми за превоз с камиони и доставка и изследване на по-безопасно и ефикасно пренасяне на стоки с Waymo Driver. През 2021 г. се стартира изследователска програма на Waymo SF Trusted Tester [6, 7].

Технологиите, които се използват от Waymo за автономните автомобили, са сензори, чипове, камери, лазерно базирана лидарна система, самоуправляващ се хардуер и радар. Изработени са бутони за мотоциклетисти за управление на определени функции: „помощ“, „заклучване“, „спиране“ и „започване на каране“ [3].

Автономните автомобили на Tesla – Tesla е компания за електрически превозни средства в САЩ, посветена на изследване, разработване и продажби на електрически превозни средства и батерии. Основана е на 1 юли 2003 г. от Мартин Еберхард и Марк Таленинг в Пало Алто Калифорния. Тяхната цел е да докажат, че електрическите превозни средства са по-добри, по-бързи и по-забавни за шофиране от автомобилите с гориво. През 2004 г. Илон Мъск се присъединява към компанията с първоначални инвестиции от 30 милиона долара, след 4 години става главен изпълнителен директор на Tesla. Първият проект за автономна кола е стартиран през 2013 г. и през 2014 г. е произведен първият модел Tesla S, който е подходящ само за ограничен достъп до магистрала за градско шофиране. Моделът е оборудван със система за разпознаване на лентата с автономно управление, спиране и регулиране на ограничението на скоростта въз основа на разпознаване на изображения на сигнали. Компанията все още не е пуснала своя напълно автономен автомобил, тъй като чака одобрение от Национална администрация за безопасност на движение по пътищата (NHTSA) [8, 9].

Технологиите, които се използват от Tesla за автономните автомобили, са камери, които осигуряват 360 градусова видимост до 250 метра обхват. Използват се също и напълно самоуправляващи се компютри (Full Self-Driving computer –FSD), които извършват бърза обработка на невронните мрежи. FSD е в състояние да осигури интелигентна производителност и контрол, за да позволи ново ниво на безопасност и автономност, без да

оказва влияние върху цената или обхвата. Въпреки това автомобилите на Tesla все още изискват активното наблюдение на водача и не са напълно автономни. Други технологии, които се използват, са за безжични комуникации и аудио системи [9, 10].

4. Кибератаки срещу автономните автомобили Tesla и Waymo

ИТС се развиват под формата на кооперативни интелигентни транспортни системи или свързани превозни средства. И двете форми използват комуникацията на данни между превозните средства (V2V), превозно средство към инфраструктура (V2I/I2V) и други обекти на пътя.

Развитието на кибер-физически системи, съдържащи усъвършенствани сензори, подсистеми и интелигентни приложения за помощ при шофиране през последното десетилетие оборудва безпилотни летателни и пътни превозни средства с автономни възможности за вземане на решения. Нивото на автономност зависи от състава и степента на усъвършенстване на сензора и оперативните приложения на автомобила. В резултат на това самоуправляващите се автомобили са компрометирани, възприемани като сериозна заплаха. Следователно е необходимо анализиране на заплахите и атаките срещу самоуправляващите се автомобили и ИТС и съответните им контрамерки за намаляване на тези заплахи и атаки.

От началото на тестовите за автономно шофиране на превозни средства е имало различни видове атаки срещу различни модули на самоуправляваща се кола, като вътрешното измервателно устройство, Lidar, GPS, камера, устройство за наблюдение на тласкащото устройство, AU и предупредителни съобщения. До момента са докладвани приблизително 126 инцидента. При такива атаки превозното средство не подготвя сигурна последователност от движения за маневриране в тясно пространство [4].

- **Tesla**

През 2016 г. се атакуват автономните автомобили на Tesla с атаки срещу Uconnect и GPS системите, които водят до неоторизиран достъп за поемане на контрол на UBU от водача или автомобила дистанционно. През 2017 г. се правят атаки за чужд контрол на скоростта на Tesla, което разкрива чувствителна информация. През 2018 г. се атакува антивирусната програма и Lidar, което причинява изтичането на чувствителна информация и предотвратяване на превозните средства на Tesla да получават чувствителна информация и да използват мрежова услуга.

- **Waymo**

През 2016 г. се атакуват автономните автомобили на Waymo, което довежда до приемане на грешна информация за пътната обстановка по мрежата. През 2018 г. се атакува GPS системите, което дава неправилно местоположение на автомобилите и се изтичат данни в задния кабелен канал.

5. Заключение

Масовото навлизане на автономните автомобили се очаква да донесе значителни промени в ежедневието на хората. В статията са посочени нивата на автономност на превозните средства и тяхното развитие. Различните нива на автономност характеризират самостоятелното управление на превозното средство без човешки фактор. Посочени са и направените хакерски атаки срещу автомобилите на Tesla и Waymo. Публикуваните данни сочат, че от 2016 г. до 2018 г. автомобилите на Tesla са по-често атакувани от автомобилите на Waymo за същия период. Двете компании използват различни подходи за решаване на такива проблеми.

В бъдеще ще бъдат изследвани ефективността и безопасността при пътуване с автономните автомобили при извънредни ситуации и причините за все още немасовото използване на такива превозни средства.

Концепцията за самоуправляваща се кола се обсъжда от 20-те години на 20-ти век, но първата полуавтоматична кола е разработена от Япония през 1977 г. През 80-те години на 20-ти век е разработена основна забележителност на самоуправляващата се кола с ограничена скорост до 31 км/ч. Военните проекти Demo I, Demo II, Demo III, финансирани от правителството на САЩ през 2000 г., са безпилотно превозно средство за навигация по дълги трудни терени извън обособено пътно платно. След няколко години в САЩ се предлага система за частична автоматизация (Ниво 2) от няколко автомобилни компании, а условна автоматизация (Ниво 3) е в процес на разработка. Първият пълен градски тест за самоуправляващ се автомобил е направен през 2007 г. Въпреки множеството прототипи, представени от компании като Google и Nissan, революцията е направена от TESLA през 2014 г. [3].

Литература

- [1]. Barouch Matzliach, Irad Ben-Gal, Evgeny Kagan, Detection of Static and Mobile Targets by an Autonomous Agent with Deep Q-Learning Abilities, Entropy, 2022
- [2]. Dr.T. Manikandan, J.S. Jayashwanth, S. Harish and K.N. Harshith Sivatej, Self-Driving Car, 2020
- [3]. Mohsin Raza, Autonomous Vehicles: Levels, Technologies, Impacts and Concerns, 2018
- [4]. Abdullahi Chowdhury, Gour Karmakar, Joarder Kamruzzaman, Alireza Jolfael, Rajkumar Das, Attacks on Self-Driving Cars and Their Countermeasures: A Survey, IEEE Access, 2020
- [5]. Intelligent Transportation Systems, <https://www.intechopen.com/books/845>, последно посетен на 10.11.2022 г.
- [6]. Waymo Story, <https://waymo.com/company/#story>, последно посетен на 10.11.2022 г.
- [7]. A first look at our Waymo fully self-driving Chrysler Pacifica Hybrid minivans, <https://blog.waymo.com/2019/09/a-first-look-at-our-waymo-fully-self.html>, последно посетен на 10.11.2022 г.
- [8]. Tesla cars will not be approved as fully self driving this year, Musk says, <https://www.reuters.com/business/autos-transportation/tesla-flags-its-cars-not-ready-be-approved-fully-self-driving-this-year-2022-10-20/>, последно посетен на 10.11.2022 г.
- [9]. Full Self-Driving Computer Installations, <https://www.tesla.com/support/full-self-driving-computer>, последно посетен на 10.11.2022 г.
- [10]. Model S, <https://www.tesla.com/models#avaya-chat-modal>, последно посетен на 10.11.2022 г.

За контакти:

Ас. инж. Мариета Методиева Йорданова
катедра „Компютърни науки и технологии”
Технически университет – Варна
E-mail: marieta.yordanova@tu-varna.bg

СРАВНЕНИЕ НА АЛГОРИТМИ ЗА КЛЪСТЕРИЗАЦИЯ НА ДАННИ

Гинка К. Маринова

Резюме: В тази статия е разгледано естеството на клъстеризация. Представен е сравнителен анализ и описание на методи за клъстерен анализ, използвани в техники за извличане на данни с различни принципи. Този процес е актуална област на непрекъснато развитие за големи масиви от данни за неуправляемо обучение.

Ключови думи: алгоритми, клъстерен анализ, метрика, клъстеризация, неуправляемо обучение

COMPARISON OF DATA CLUSTERING ALGORITHMS

Ginka K. Marinova

Abstract: In this article we discuss the nature of clustering. A comparative analysis and description of cluster analysis methods used in data mining techniques with different principles are presented. This process is a topical area of continuous development for large datasets for unsupervised learning.

Keywords: algorithms, cluster analysis, metrics, clustering, unsupervised learning

1. Увод

Клъстерният анализ е многовариантен статистически метод, при който се събират в многомерен масив данни, съдържащи информация за извадка от обекти, като обектите се подреждат в хомогенни групи, наречени клъстери [1, 2]. Клъстер е група от елементи, характеризиращи се с общо свойство, като основната задача на клъстерния анализ е да се намерят групи от подобни обекти в извадката [1, 2]. Предназначението на клъстеризацията е търсене на съществуващи структури. Обхватът на клъстерния анализ е инструмент за решаване на редица задачи в медицината, психоонкологията, психологията, компютърната лингвистика, маркетинг и много други области на човешката дейност.

Решаваните задачи чрез клъстерния анализ са:

- ✧ Представяне на данните чрез идентифициране на структурата на клъстера. Разделяне на извадката на групи от подобни обекти като се предоставя възможност за опростяване на по-нататъшната обработка на данните и вземане на решения чрез прилагане на собствен метод за анализ към всеки клъстер.
- ✧ Компресиране на данните. При прекалено голяма първоначална извадка тя може да бъде намалена, оставяйки един от най-типичните представители от всеки клъстер.
- ✧ Изследване на концептуални схеми за групиране на обекти.
- ✧ Генериране на хипотези на база на изследване на данни.
- ✧ Тестване на хипотези или изследване, като целта е да се определят идентифицираните групи действително ли присъстват в наличните данни.

2. Изложение

Процесът клъстерен анализ се състои от множество изчислителни процедури, използвани при създаването на класификации на обекти, състоящ се от следните фази:

- ✧ От целия набор от данни се прави извадка или се изключват част от данните;
- ✧ Избор на метрика за типизация на входните данни;
- ✧ Подбор на алгоритъм за клъстеризация;
- ✧ Дефиниране на количеството клъстери;
- ✧ Визуализация на получените резултати.

На фигура 1 е визуализирана блоковата схема на процеса клъстеризация на данни.



Фиг. 1. Блокова схема за решаване на задача за клъстеризация

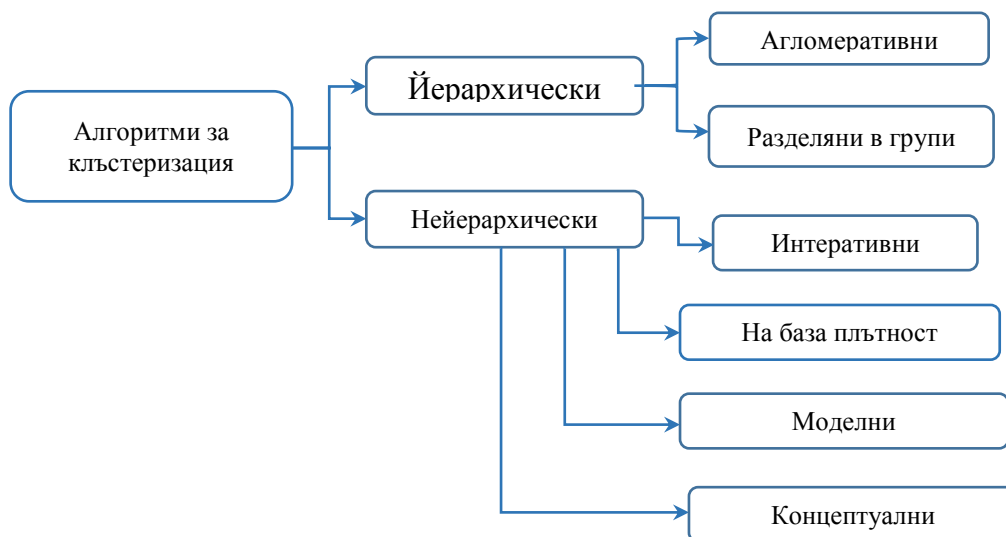
В процеса по подготовка на данните те се описват под формата на таблици, където колоните са атрибутите, а редовете са обекти на данните. Втората фаза се състои в избор на метрика, с помощта на която се определя сходството на обектите. Използваните метрики са различни според типа на данните, които се сегментират. Характерната метрика за двумерното пространство за разстояние между интервални данни е евклидовото разстояние между два вектора [26] $x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{ip})$ и $x_j = (x_{j1}, x_{j2}, x_{j3}, \dots, x_{jp})$:

$$d(x_i, x_j) = \sqrt{\sum_{m=1}^p (x_{im} - x_{jm})^2}, j = 1, \dots, n \quad (1)$$

За всеки определен вид данни се избира метрика. При решаването на конкретни задачи се срещат обстоятелства, когато не може да се намери показател за определен тип данни и е необходимо да се разработи собствена система за оценка на разстоянието между обектите. Резултатите от групирането могат да варират значително чрез използване на различни метрики за близост [5]. Следващата фаза е избор на алгоритъм, по който се групират обектите.

Изборът на подходящ алгоритъм е от съществено значение, тъй като полученият резултат до голяма степен зависи от алгоритъма. При решението на поставена задача е необходимо да се използват няколко алгоритъма, с цел комбинация, за да се получи по-точен резултат. Четвъртата фаза е изпълнение на избрания алгоритъм/и. Резултатът е групиране на обектите в клъстери. Петата фаза включва представяне на полученото групиране в удобна за описание форма. Визуализирането на резултатите от групирането е с идея да помогне за съответното тълкуване на резултатите от изпълнението на алгоритъма [3]. От проучените литературни източници и научни статии в областта на клъстерния анализ се установява, че са разработени и се прилагат много голям брой алгоритми. Алгоритмите за клъстеризация се разделят на две категории: йерархични и нейерархични което е показано на фигура 2.

Под формата на дендрограма се изобразяват йерархичните алгоритми, при която са обозначени последователностите, в които са разделени обектите на клъстери. Йерархичните алгоритми се разделят на две групи като агломеративни и разделяни в групи в зависимост от начина за формиране на йерархична декомпозиция.



Фиг. 2. Схема на алгоритми за клъстеризация

Представител на агломеративните алгоритми е CURE (Клъстеризиране с помощта на представители) [6, 7]. Концепцията му се състои в осъществяване на йерархично групиране, като се използва набор от определящи точки, за да се дефинира обект в клъстер [7, 8, 9]. Следващият агломеративен алгоритъм е на най-близкия съсед [9, 10, 11]. При този алгоритъм първият етап е да се сформира матрица на разстоянията между обектите и да се присвои клъстер на обектите. Вторият етап се състои в намиране на минималното разстояние между обектите. Третият етап е групирането на клъстери.

При *разделящите* алгоритми първоначално всички обекти са един клъстер, който в последствие се подразделя на по-малки клъстери. Алгоритъм BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) - Балансирано итеративно намаляване и клъстеризиране с помощта на йерархии се осъществява за огромно количество числови данни [12, 13, 14]. Този алгоритъм реализира двуетапен процес на групиране. Друг представител е MST алгоритъм (Минимално обхващащо дърво). Основната идея при този алгоритъм е групиране на големи набори от произволни данни [15, 16].

При *нейерархичните* итеративни алгоритми обектите се разпределят на итерации по клъстери. Установява се, че са с по-висока устойчивост на шум. Широко приложение се забелязва на алгоритъма на k-средните стойности [17, 18]. Основната концепция на този алгоритъм се състои в минимизиране на разстоянията в клъстери между обектите. Когато не може повече да се минимизира разстоянието, то алгоритъмът се прекратява.

Нейерархичен алгоритъм за клъстеризация на база плътност е DBSCAN (Пространствено клъстеризиране на базата на плътност на приложения с шум) [19, 20]. Идеята е, че за всички обекти, които са тясно свързани помежду си, се маркират се излишните обекти, които са разположени сами в области с ниска плътност [19, 20, 21].

Клъстеризиране въз основа на модел е алгоритъм EM (Expectation-maximization) [22, 23]. Идеята на моделния алгоритъм е, че има някакъв клъстерен модел и е необходимо да се намери максималното сходство между този модел и данните. Апаратът на математическата статистика се използва за представяне на такива модели. Този алгоритъм използва максимална вероятност. Същността на алгоритъма се състои в това, че към известните данни има неизвестни, които се отнасят към разпределението на клъстерите.

При концептуалните алгоритми се създава концепция за описание на всеки клъстер като се генерира йерархия на класове, но те не са свързани с йерархично групиране. Основната концепция при COWEB алгоритъм е създаването на последователно изграждане

на клъстерно дърво, като се има предвид полезността на присвояването на обект към клъстер [24, 25].

Таблица 1. Сравнителен анализ на алгоритми		
Алгоритъм	Предимства	Недостатъци
Йерархични - Агломеративни алгоритми		
CURE (Клъстеризиране с помощта на представители)	✧ осъществяване на групиране на високо ниво; ✧ определяне на клъстери с различни размери	✧ необходимост от задаване на прагови стойности; ✧ потребност от задаване на броя на клъстерите
Най-близкия съсед	✧ лесно разбираем за потребителите; ✧ скорост на използване; ✧ яснота и прозрачност на алгоритъма.	✧ изчислена сложност; ✧ клъстерите не се припокриват; ✧ изчислително приближение.
Йерархични - Разделящи по групи алгоритми		
Алгоритъм BIRCH (Итеративно намаляване и клъстеризиране с помощта на йерархии)	✧ използва се за голям обем от данни; ✧ двустепенна клъстеризация; ✧ работи с ограничен обем памет.	✧ работи само с числови данни; ✧ добре избира само клъстери от сферични форми; ✧ нужда от задаване на прагови стойности.
MST алгоритъм (Минимално обхващащо дърво)	✧ избира клъстери с произволна форма, вкл. изпъкнали и изпъкнали клъстери; ✧ избира най-доброто решение от няколко оптимални решения.	✧ ограничена приложимост; ✧ висока интензивност на работа.
Нейерархични - Итеративен алгоритъм		
k- средните стойности	✧ лекота и бързина на използване; ✧ яснота алгоритъма; ✧ прозрачност на алгоритъма.	✧ висока изчислителна сложност; ✧ бавно работи с големи бази от данни; ✧ необходимо е да се посочи броят на клъстерите.

Нейерархични - На база плътност алгоритми		
Алгоритм DBSCAN (Базирано на плътност пространствено групиране на приложения с шум)	✧ кратко време за изчисление; ✧ добър за модификация; ✧ не е необходимо предварително определяне на броя на клъстерите; ✧ устойчив на свръхнатоварване; ✧ не е чувствителен към подреждане на данните.	✧ висока сложност; ✧ не е цялостно детерминиращ; ✧ не възможно да групира голям набор от данни с големи разлики в плътността; ✧ изборът на праг на разстояние може да бъде затруднен при не добре описани данни и мащаб.
Нейерархични - Клъстериране въз основа на модел		
Алгоритм EM	✧ бързо приближаване при успешна инициализация; ✧ възможност за изграждане на желан брой клъстери; ✧ линейно нарастване на фалшивостта с нарастване на количеството данни.	✧ алгоритъмът може да спре на локален минимум; ✧ сходимостта на алгоритъма може да бъде ниска в случай на неуспешна инициализация.
Нейерархични - Концептуален алгоритъм		
Алгоритм DBSCAN (Базирано на плътност пространствено групиране на приложения с шум)	✧ кратко време за изчисление; ✧ добър за модификация.	✧ висока сложност; ✧ не е цялостно детерминиращ.

Заклучение

В статията са описани едни от най-разпространените алгоритми за клъстеризация. Клъстерният анализ намира широко приложение в психологията, психоонкологията, в статистиката, в извличане на информация, машинно обучение и събиране на данни и др. Процесът на изследване посредством методите и алгоритмите за клъстерен анализ е неразделна част от съвременната обработка на големи количества информация. Изборът на алгоритъм или комбинация на алгоритми зависи от конкретния случай и не е еднозначен, като резултатът е в пряка зависимост от използваната метрика за разстояние. При изучаването и тестването на съществуващи алгоритми върху реални данни е необходимо да се знаят техните предимства и недостатъци.

Литература

- [1]. Балабанова, И., Георгиев, Г., Моделиране и изследване на процеси в бизнеса и индустрията с изкуствен интелект и машинно обучение, методическо ръководство за лабораторни упражнения, ТУ-Габрово, pp. 155, 2022.
- [2]. Brauksa, I., Use of Cluster Analysis in Exploring Economic Indicator Differences among Regions: The Case of Latvia". *Journal of Economics, Business and Management*, vol. 1, № 1, pp. 42-45, 2013.
- [3]. Rodriguez M., et al., Clustering algorithms: A comparative approach, *PLoS ONE*, no. 14(1). available at: <https://doi.org/10.1371/journal.pone.0210236>, 2019.
- [4]. Kinnunen, T., et al., Comparison of Clustering Methods: a Case Study of Text-Independent Speaker Modeling. *Pattern Recognition Letters*, no. 32, pp. 1604-1617, 2011, DOI: 10.1016/j.patrec.2011.06.023.
- [5]. Rezankova H., Cluster Analysis of Economic Data, *STATISTIKA*, vol. 94, issue 1, pp. 73-86, 2014.
- [6]. Sudipto, G., et al., CURE: An Efficient Clustering Algorithm for Large Databases, *Journal Information systems*, vol. 1, pp.35-38, 2001.
- [7]. Kang, W., Ye, D., Study of CURE based clustering algorithm. In: 18th China Conference on Computer Technology and Applications (CACIS), vol. 1. Computer Technology and Application Progress, pp. 132–135. 2007.
- [8]. Zhao, Y., Research on user clustering algorithm based on CURE. *Comput. Eng. Appl.* 11(1), pp.457–465, 2012.
- [9]. Han, J., Kamber, M.: *Data Mining: Concepts and Techniques*, 3rd edn. China Machine Press, Beijing, 2012.
- [10]. Introduction to Nearest Neighbors Algorithm, electronic resource, Nearest Neighbors Algorithm | Advantages and Disadvantages (educba.com)
- [11]. Qing-sheng Z., A clustering algorithm based on natural nearest neighbour, *Journal of Computational Information Systems* 10(13), pp.5473-5480, 2014, DOI:10.12733/jcis10521
- [12]. Zhang, T., et al., BIRCH: An Efficient Data Clustering Method for Very Large Databases, electronic resource, <https://dl.acm.org/doi/10.1145/235968.233324>
- [13]. Yangyang, L., et al., A Large-Scale Data Clustering Algorithm Based on BIRCH and Artificial Immune Network, pp. 327-337, 2018, doi: 10.1007/978-3-319-93815-8_32
- [14]. Advantages And Disadvantages Of Birch, electronic resource, Advantages And Disadvantages Of Birch - 1734 Words | [Bartleby](http://Bartleby.com)
- [15]. Patel, N., Patel, K., A Survey on: Enhancement of Minimum Spanning Tree, Nimesh Patel *Int. Journal of Engineering Research and Applications*, ISSN: 2248-9622, Vol. 5, Issue 1(Part 3), pp.06-10, 2015.
- [16]. Andersen, C., An optimal minimum spanning tree algorithm, Master's Thesis, Department of Computer Science Aarhus University, pp.11 2008.
- [17]. Morissiette L., Chartier S., "The k-Means Clustering Technique: General Considerations and Implementation in Mathematica". *Tutorials in Quantitative Methods for Psychology*, vol. 9, issue 1, pp. 15-24, 2013.
- [18]. [Ali H., Kadhum K., "K- Means Clustering Algorithm Applications in Data Mining and Pattern Recognition". *International Journal of Science and Research*, vol. 2015, pp. 1577-1854, 2015.
- [19]. Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, Xiaowei Xu. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN // *ACM Trans. Database Syst.* ISSN 0362-5915. 2017, doi:10.1145/3068335.

- [20]. Kriegel, H., et al., The (black) art of runtime evaluation: Are we comparing algorithms or implementations? // Knowledge and Information Systems, pp. 341, 2016, ISSN 0219-1377. — doi:10.1007/s10115-016-1004-2.
- [21]. Hans-Peter Kriegel, Peer Kröger, Jörg Sander, Arthur Zimek. Density-based Clustering // WIREs Data Mining and Knowledge Discovery, pp. 231–240, 2011, doi:10.1002/widm.30.
- [22]. Trevor; H., et al. 8.5 The EM algorithm // The Elements of Statistical Learning (неопр.). — New York: Springer, ISBN 0-387-95284-5, pp. 236—243, 2001.
- [23]. Expectation–maximization algorithm, electronic resource, <https://resources.saylor.org/wwwresources/archived/site/wp-content/uploads/2011/11/CS405-6.2.3.2.2.2-WIKIPEDIA.pdf>
- [24]. Долгодорова, Е., КЛАСТЕРНЫЙ АНАЛИЗ: БАЗОВЫЕ КОНЦЕПЦИИ И АЛГОРИТМЫ, ВОПРОСЫ НАУКИ И ОБРАЗОВАНИЯ, Номер: 7 (19), стр.73-76, 2018.
- [25]. Wayne Iba, Langley, P., Cobweb models of categorization and probabilistic concept formation". In Emmanuel M. Pothos and Andy J. Wills (ed.). Formal approaches in categorization. Cambridge: Cambridge University Press. pp. 253–273. ISBN 9780521190480.
- [26]. Гочева-Илиева, С., Лекция 10. Многомерен кластерен анализ (КА), електронен ресурс, Пловдив, 2018, http://www.fmi-plovdiv.org/evlm/DBbg/database/courses_BIT/10%20lekcia%20PM.pdf

За контакти:

За контакти: ас. д-р инж. Гинка К. Маринова
кафедра „Компютърни науки и технологии“
Технически университет – Варна
e-mail: gmarinova@tu-varna.bg



ПРЕДИМСТВА И НЕДОСТАТЪЦИ НА СЪВРЕМЕННИ СОФТУЕРНИ РАМКИ

Гинка К. Маринова

Резюме: В тази статия са анализирани различни съвременни софтуерни рамки, използвани в много области на софтуерното инженерство за решаване на конкретни задачи. За всеки програмен език са създадени програмни рамки, предназначени за уеб разработка, проектирани с цел улеснение при създаването на уеб приложения, услуги и уеб ресурси. В статията е представено описание на предимствата и недостатъците на подбрани технологични рамки в сравнение с традиционното програмиране.

Ключови думи: софтуерна рамка, библиотека, React, Angular, Vue.js, Bootstrap, Node JS, Django, Ruby on Rails, Flutter, React Native, PyTorch, TensorFlow

Advantages and disadvantages of modern software frameworks

Ginka K. Marinova

Abstract: This article analyzes various modern software frameworks used in many areas of software engineering to solve specific tasks. For each programming language, there are web development frameworks designed to facilitate the creation of web applications, services, and web resources. The article describes the advantages and disadvantages of selected technological frameworks compared to traditional programming.

Keywords: software framework, library, React, Angular, Vue.js, Bootstrap, Node JS, Django, Ruby on Rails, Flutter, React Native, PyTorch, TensorFlow

1. Увод

Развитието и използването на достиженията и възможностите на информационните и комуникационни технологии е предпоставка за популяризиране и използване на софтуерни рамки. Разработката на съвременни и модерни приложения на избран програмен език включва използването на множество библиотеки или софтуерни рамки. В днешно време много рядко можем да срещнем създадени приложения без да са използвани софтуерни рамки и библиотеки. Тези абстрактни инструменти имат за цел да подпомогнат програмирането да е бързо и лесно. Библиотеките са съвкупност от функции или обекти за решаване на поставени задачи, като отделните функции и обекти са необходими за реализацията на конкретен проект. Софтуерните рамки са структури, които се използват за изграждане на софтуер за разработка на определен тип приложения. Тези инструменти са подобни на шаблони, които могат да се модифицират и добавят определени функции и функционалности с цел създаване на сложен и голям проект, който много хора да използват.

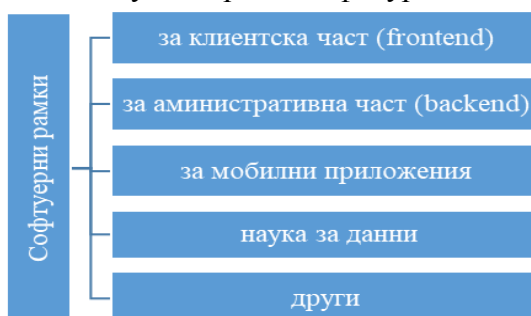
Причините за да се използват софтуерните рамки са:

- Способстват да се избегне дублиране и излишен код;
- Улеснение за разработчиците, за тестване и отстраняване на грешки;
- Рамките се поддържат от група хора, които ги тестват, за да може да се използват целенасочено;
- Тези инструменти са в помощ да се пише чист и сигурен код;
- Значително се намалява времето за разработване на приложение, като разработчиците се съсредоточават върху писането на специфичен за проекта код.

От прочетените и анализирани доклади и книги се установява, че са реализирани много софтуерни рамки, всяка с уникални характеристики, с които се отличава. Необходимото

условие е да се проучи коя рамка отговаря на нуждите за създаване на конкретен проект, преди да се започне изграждането му или писането на код.

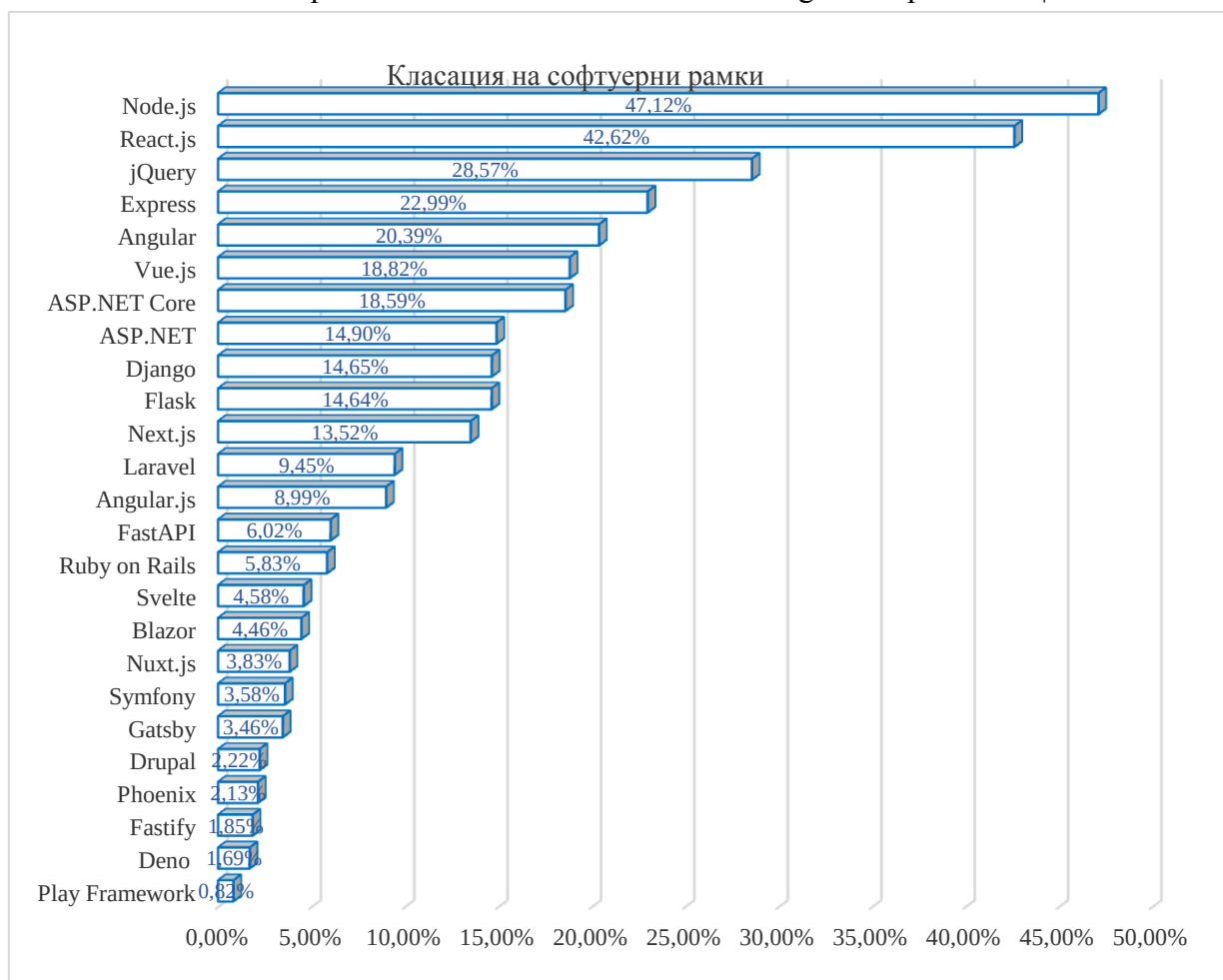
Видовете софтуерни рамки са визуализирани на фигура 1.



Фиг. 1. Видове софтуерни рамки

Софтуерните рамки се прилагат във всички области на софтуерното инженерство, уеб разработка на клиентска част (frontend) и административна част (backend), за мобилни приложения, наука за данни и много други. Изборът на подходящ инструмент за разработката и изпълнението на дадена задача има свой собствен ефект върху успеха и ефективността на проекта.

На фигура 2 е представено проучване - класация за последната година на софтуерни рамки и библиотеки според Most used web frameworks among developers 2022 | Statista.



Фиг. 2. Класация на уеб рамки за разработка на софтуер според Most used web frameworks among developers 2022 | Statista; Източник: <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>; (11.05.2022 г. до 1.06.2022 г., 58 743 участници в проучването на разработчици на софтуер от цял свят)

2. Предимства и недостатъци на уеб рамки

2.1 Рамки за клиентска част (frontend)

React е разработена от Meta като фронтенд JavaScript рамка [1, 2, 3, 4]. Тя е популярна и много употребявана рамка, използвана от милиони разработчици по света. Тази софтуерен инструмент е с декларативна, ефективна и гъвкава библиотека за изграждане на потребителски интерфейси, позволяващ да се създават сложни потребителски интерфейси от малки и изолирани части от код, наречени компоненти [1]. Логиката на компонентите е написана на JavaScript вместо в шаблони, като по този начин лесно се предават данните през приложението с цел дистанциране от DOM манипулации. При промяна на данните резултатно се актуализират и визуализират правилните компоненти. Декларативният интерфейс организира кода като по-предвидим и по-лесен за поддръжка.

Предимства [1, 2, 3, 4]:

- Предоставя на разработчиците да разделят функционалностите на малки преизползваеми компоненти с цел по-лесно управление;
- Въведени са новите JavaScript стандарти (ES6+), което прави изграждането на едно приложение по-приятно;
- Добра документация, която е необходимост за придобиване на нужните умения;
- Модулен дизайн;
- Лесен за създаване интерфейс;
- Гъвкавост;
- Ефективност и висока производителност;
- Използва се концепция за оптимизация, която предоставя възможността на изградените UI да се пази в паметта и да се синхронизира с DOM дървото.

Недостатъци [1, 2, 3, 4]:

- Объркващ JSX синтаксис;
- Проблеми с отстраняване на грешки и съвместимост;
- Непълно MVC съответствие;
- Сложност на оптимизацията за търсачки.

Angular е базирана TypeScript рамка с отворен код, разработена от Google [4]. Софтуерният инструмент Angular предоставя функционалност като двупосочно свързване, което позволява динамично да се променят данни на едно място на интерфейса, когато данните на модела се променят на друго, шаблони, маршрутизиране и други. Тази клиентска рамка използва различни шаблони за изглед за изграждане на динамични уеб приложения за мобилно устройство или настолен компютър.

Предимства [2, 5, 6, 7]:

- Лесен за управление и използване на изгледа;
- Данни, които синхронизират модела с изгледа;
- Промените в модела се отразяват пряко в изгледа;
- Съдействие при обработка на заявки;
- Взаимозависимост на функциите с оглед връзката им с компоненти и модули;
- Използват се препоръки, които позволяват на разработчиците да използват функции и да генерират инструкции за извършване на промени;

Недостатъци [2, 5, 6, 7]:

- Необходимо е разработчикът да е наясно с MVC;
- Сложен за програмиране, въпреки че Angular използва TypeScript;

- Грешки при интегриране, които могат да възникнат при преминаване от стара версия към нова;
- Трудности с шаблоните от страната на сървъра;
- Angular използва директно DOM, което намалява неговата скорост и ефективност в сравнение с React.

Vue.js е прогресивна JavaScript рамка за създаване на адаптивни потребителски интерфейси и приложения [8, 9]. Този персонализиран инструмент е лесен за научаване и приятен за писане. Прогресивната рамка Vue.js се използва за решаване на задачи на презентационния слой, лесно се интегрира с други библиотеки и проекти. Поддръжката на Vue.js може да бъде добавена към съществуващ проект, което значително разширява неговата функционалност и именно това го отличава от другите рамки.

Предимства [8, 9, 10, 11]:

- Написана е подробна документация;
- Адаптивност - осигурява бърз преход от други рамки;
- Добра интеграция, изразяваща се в това, че малките интерактивни части могат лесно да бъдат интегрирани в съществуващата инфраструктура, без това да повлияе неблагоприятно на цялата система;
- Мащабиране - разработват се сравнително големи шаблони за многократна употреба, които могат да бъдат разработени, без да се отделя огромно количество време с оглед на простата структура;
- Скорост и гъвкавост, позволяващи много по-добра производителност от други рамки.

Недостатъци [8, 9, 10, 11]:

- Малък пазарен дял в сравнение с React или Angular;
- Риск от прекомерна гъвкавост, изразяваща се в това, че може да има проблеми с интегрирането в огромни проекти и няма опит за възможни решения;
- Липса на пълна документация на английски език, понеже по-голямата част от кода е написан на китайски, което затруднява неговорещите този език;
- Тъй като е в процес на разработка, липсва поддръжка за важни добавки.

Bootstrap [12, 13] е уеб рамка, която улеснява разработчиците да създадат напълно адаптивен уебсайт или уеб приложение, разработен и поддържан от Twitter. Този инструмент е популярен, удобен и бърз във всички базови компоненти, но и лимитиран. Предоставя удобна за ползване решетка от колони и редове, която улеснява много подреждането на повечето елементи по страницата на дадено уеб приложение.

Предимства [12, 13]:

- Лесен за работа интерфейс;
- Голям набор от готови решения, осигуряващи основната функционалност;
- Висока скорост на създаване на висококачествено адаптивно оформление;
- Правилно визуализиране и работа на сайта във всички браузъри и операционни системи, поддържани от тази рамка.

Недостатъци [12, 13]:

- По-голям размер на крайните файлове при реализация на проект;
- Ниска скоростта на зареждане на уеб приложението;
- Липса на гъвкавост.

2.2 Рамки за административна част (backend)

Node JS [14, 15, 16, 17] е технология с отворен код, базирана на V8 JavaScript за изграждане на мащабируеми мрежови приложения. Тази платформа е със собствен интерфейс за взаимодействие с входно-изходните устройства на компютъра. Основният подход на Node.js е, че се управлява от събития. Популярността на технологичната рамка се дължи на леките и ефективни създадени базови приложения.

Предимства [14, 15, 16, 17]:

- Лесен за усвояване и разбиране за разработчици на JavaScript;
- Голямо разнообразие от функции;
- Широка поддръжка на общността;
- Поддържа стрийминг на файлове с голям размер;
- Управление на едновременни заявки за събития.

Недостатъци [14, 15, 16, 17]:

- Не е мащабируема;
- Слаба ефективност при операции с интензивно използване на процесора;
- Не е достатъчно тествана.

Django [18, 19, 20] е MVT уеб рамка, използвана за изграждане на уеб приложения, която може да работи с всяка предна рамка. Отличава се с простота, която помага на уеб разработчиците да се съсредоточат върху писането и изграждането на значим проект, с написан ефективен и мощен код. Това е една от известните уеб рамки в света и също така е една от най-използваните рамки.

Предимства [18, 19, 20]:

- Използван е всеобхватен подход за реализация на множество компоненти;
- Универсалност на този инструмент, тъй като е подходящ за разработване на уебсайтове и приложения като системи за управление на съдържание, новини или информационни ресурси, видео хостинг, социални мрежи и др.;
- Актуализация и реализация на подобрения;
- Мащабируемост - всеки компонент може да бъде заменен или модифициран, без да се засягат други компоненти;
- Вградени са инструменти за защита срещу често срещани хакерски атаки;
- Обширност - създадени са голям брой плъгини, различни инструкции и др.;
- Предоставена е обширна документация за подпомагане на разработчиците;
- Гъвкавост.

Недостатъци [19, 20, 21]:

- Липса на многозадачност, не позволява отделни процеси да работят с множество заявки едновременно;
- Настройката му създава трудности за начинаещи с шаблона за маршрутизиране на URL;
- Монолитна архитектура - модули, като ORM или формуляри, са трудни за замяна;
- Липсва поддръжка на WebSocket по подразбиране.

Ruby on Rails [21, 22, 23] е бекенд рамка с отворен код, реализирана на езика Ruby. Това средство предоставя структури и функции по подразбиране за бази от данни, уеб услуги и уеб страници, като приоритет на бързото разработване на приложения с иновативни функции. Изградена е върху архитектура Model-View-Controller (MVC). Проектирана е да включва по-малко код и повторения и се поддържа от голяма общност.

Предимства [21, 22, 23]:

- Висококачествена, мащабируема, сигурна и популярна рамка за уеб разработка;
- Реализация на много налични функции, по този начин разработчиците спестяват време и не пишат сложен код;
- Безопасност - сигурен жизнен цикъл на разработка и са вградени мерки за сигурност при този инструмент;
- Осигурява отлична производителност и оптимизира кода много ефективно;
- Използва се от огромна общност по целия свят;
- Автоматизирано тестване.

Недостатъци [21, 22, 23]:

- Характеризира се с бавна скорост на изпълнение, което затруднява мащабирането на уеб приложенията;
- Трудности при създаване на интерфейс за програмиране на приложения (API);
- Не е написана добра документация.

2.3 Рамки за мобилни приложения

Flutter [24, 25] е рамка за създаване на уеб приложения, реализирана от Google. Това е платформа с отворен код за изграждане на висококачествени и производителни мобилни приложения за мобилните операционни системи Android и iOS , като не е необходимо да се поддържат две отделни кодови бази. Flutter е базиран на езика Dart.

Предимства [24, 25, 26]:

- Малък риск от грешки в бизнес логиката, понеже версиите за iOS и Android имат една и съща основа;
- Спестяване на време за разработка;
- Лесен достъп до вградени функции за геолокация и камера;
- Добро решение за създаване на MVP;
- Набор от инструменти за създаване на красиви интерфейси и използване на SDK;
- Добра производителност;
- Реализация на мощен механизъм за изобразяване.

Недостатъци [24, 25, 26]:

- Необходимост от отделни тестове за всяка платформа;
- Сравнително голям размер на създаденото приложение;
- Малък брой разработчици работещи на Flutter.

React Native [24, 27, 28, 29] е популярен инструмент за създаване на уебсайтове и уеб приложения, базиран на JavaScript, за iOS и Android приложения, като се използва една и съща кодова база. Тази софтуерна рамка се състои от голямо количество вградени компоненти и API.

Предимства [24, 27, 28, 29]:

- Възможност за повторно използване на кода – крос платформа;
- Голяма общност на разработчици;
- Актуализация - обновяване на старите версии;
- Висока производителност;
- Безплатно ползване;
- Лесен потребителски интерфейс, отзивчив и по-бърз, с намалено време за зареждане, което подобрява цялостното потребителско изживяване;

- Реализирани са интегрируеми собствени компоненти;
- Бързи темпове, с които платформата завладя пазара, и прост подход за решаване на проблеми при разработката.

Недостатъци [24, 27, 28, 29]:

- Проблеми със съвместимостта и отстраняването на грешки;
- Файловете на реализираните приложения са големи;
- Недостатъчна гъвкавост;
- Малък избор от персонализирани модули;
- Възможни проблеми със съвместимостта - възможен е конфликт на пакети или инструменти за отстраняване на грешки в рамката.

2.4 Рамки - наука за данни

PyTorch [30, 31, 32] е софтуерна рамка за машинно обучение с отворен код, базирана на езика Python и библиотеката Torch. Този платформа се използва за реализацията на множество изследвания на областта на дълбокото обучение, за създаване на модели с ниска производителност и големи набори от данни, като задачата за композиция е да ускори процеса между създаването на прототипи за изследване и внедряването им в производството.

Предимства [30, 31, 32]:

- Добра производителност;
- Гъвкавост при реализация на решения вследствие на възможността да се пишат собствени функции и достъп до математическия апарат;
- Удобен за потребителя дизайн и структура;
- Интерактивно отстраняване на грешки;
- Наличие на добра документация с подробни описания на всяка функция;
- Обширна общност от разработчици;
- Лесно адаптиране на проекти;
- Поддръжка за паралелизъм на данните.

Недостатъци [30, 31, 32]:

- При крос-платформени решения е необходимо да се интегрират допълнителни библиотеки;
- Не са налични интерфейси за наблюдение и визуализация като TensorFlow;
- Нужен е API сървър при внедряване.

TensorFlow [33, 34, 35] е безплатна платформа, разработена на Python, с отворен код за машинно обучение и изкуствен интелект, създадена от Google. Това софтуерно средство разполага с цялостна, гъвкава екосистема от инструменти, библиотеки и ресурси, позволяващо на изследователи и ползватели да се запознаят и използват машинното обучение за решаване на конкретни задачи. Този инструмент се използва за модели с висока производителност, както и за големи набори от данни. Разработчиците го използват за бързо изграждане и внедряване на базирани на ML приложения.

Предимства [33, 34, 35]:

- Лесен за научаване;
- Добър инструмент за визуализация;
- Позволява да се създават модели за дълбоко обучение в реално време в браузъра с помощта на JavaScript.

Недостатъци [33, 34, 35]:

- Бавен в сравнение с рамки като MXNet и CNTK;
- Трудности при отстраняването на грешки;
- Не поддържа OpenCL.

Заключение

Използването на подходящи софтуерни рамки за реализация на конкретни проекти променя подхода към създаването на софтуерни продукти, позволявайки на разработчиците да изучат всички характеристики и да отделят повече време за внедряването на отделни функции към даден проект, за да реализират своите идеи в рамките на установена структура.

Всяка технологична рамка се характеризира със своите предимства и недостатъци. Необходимо е да се разбере, коя софтуерна рамка предлага най-добрите функции въз основа на спецификациите на даден проект и да отговаря на всички поставени потребности. При прилагането на избрана софтуерна рамка се постига компактен и разбираем код, който в бъдеще е по-лесен за коригиране на грешки, добавяне на нова функционалност и е по-лесен за поддръжка.

Процесът програмиране е сложен процес със или без технологична рамка, изискващ старание и постоянство от страна на разработчиците. При реализация на нестандартен и сложен проект е необходимо разработчиците да го програмират на база спецификациите, без да се използват софтуерни рамки.

Литература

- [1]. Prateek P., Mahajan, A., ReactJS: A Modern Web Development Framework. International Journal of Innovative Science and Research Technology, Volume 5, Issue 11, ISSN No:- 2456-2165, 2020 pp. 698-702.
- [2]. Kumar A., Singh R., Comparative analysis of AngularJS and ReactJS, International Journal of Latest Trends in Engineering and Technology, Vol.(7), Issue(4), 2016, pp.225-227, DOI: <http://dx.doi.org/10.21172/1.74.030e-ISSN:2278-621X>
- [3]. Khuat, T., Developing a frontend application using ReactJS and Redux, Degree Programme in Business Information Technology Bachelor's Thesis, Leppävaara, 2018, pp.61.
- [4]. Robbestad, S., ReactJS blueprints, Published by Packt Publishing Ltd., ISBN 978-1-78588-654-6, 2016, pp. 395
- [5]. Delcev, S., Draskovic, D., Modern JavaScript frameworks: A Survey Study, Zooming Innovation in Consumer Technologies Conference (ZINC), 2018, pp.106-109, doi: 10.1109/ZINC.2018.8448444.
- [6]. Ramos, M., et al. AngularJS in the wild: A survey with 460 developers. In: Proceedings of the 7th International Workshop on Evaluation and Usability of Programming Languages and Tools. 2016. pp. 9-16
- [7]. Freeman, A., Pro AngularJS. Apress, 2014, pp. 688
- [8]. Kyriakidis, A., Maniatis, K., The majesty of Vue. js. Packt Publishing, 2016, pp. 240
- [9]. Hanchett, E., Listwon, B., Vue. js in Action. Simon and Schuster, 2018. pp. 303
- [10]. Macrae, C., Vue. js: up and running: building accessible and performant web apps, O'Reilly Media, Inc., 2018, pp.151
- [11]. Bin Uzayr, S., Cloud, N., Ambler, T., Vue. js. In JavaScript Frameworks for Modern Web Development, Apress, Berkeley, CA, 2019, pp. 523-539
- [12]. Shahu, G., Adkar. P., A review paper on bootstrap framework, IRE Journals 2.10, 2019, pp.349-351

- [13]. Nakkiran, P., Behnam, N., Sedghi, H., The deep bootstrap framework: Good online learners are good offline generalizers, arXiv preprint arXiv, 2020, pp.1-28
- [14]. Sun, H., et al. Efficient dynamic analysis for Node. js., Proceedings of the 27th International Conference on Compiler Construction, 2018. pp. 196-206
- [15]. Mardan, A., Practical Node. js. Apress, 1st edition, 2014, pp.300
- [16]. Hahn, E., Express in Action: Writing, building, and testing Node. js applications. Simon and Schuster, 2016, pp. 256
- [17]. Teixeira, P., Professional Node. js: Building Javascript based scalable software. John Wiley & Sons, 2012, pp. 351
- [18]. Liawatimena, S., Warnars, H. et al., Django web framework software metrics measurement using radon and pylint, Indonesian Association for Pattern Recognition International Conference (INAPR), 2018, pp. 218-222
- [19]. Rubio, D., Beginning Django. Apress., 2017, pp. 624
- [20]. Dauzon, S., et al., Django: web development with Python. Packt Publishing Ltd, 2016. pp.730
- [21]. Borup, R., An Introduction to Ruby on Rails. In Proc. Southwest Fox Conf, 2010, pp. 50-55
- [22]. Hartl, M., Ruby on rails tutorial: learn Web development with rails, Addison-Wesley Professional, 2015, pp. 566
- [23]. Klochkov, D., Mulawka, J. Improving Ruby on Rails-Based Web Application Performance. Information 2021, 12, 319. <https://doi.org/10.3390/info12080319>
- [24]. Wu, W., React Native vs Flutter, Cross-platforms mobile application frameworks, Metropolia University of Applied Sciences, Thesis, 2018, pp.34
- [25]. Dagne, L., Flutter for cross-platform App and SDK development, Metropolia University of Applied Sciences, Bachelor's Thesis, 2019, pp. 32
- [26]. Tashildar, et al., Application development using flutter. International Research Journal of Modernization in Engineering Technology and Science, 2(8), 2020, pp.1262-1266
- [27]. Danielsson, W., React Native application development, Linköpings universitet, Swedia, 2016, pp.63
- [28]. Jagiello, J. (2019). Performance comparison between react native and flutter.
- [29]. Gill, O., Using React Native for mobile software development, Metropolia University of Applied Sciences, Bachelor's Thesis, 2018, pp. 34
- [30]. Paszke, A., et al., Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems32 (NeurIPS), 2019, pp. 1-12.
- [31]. Paszke, A., et al., Automatic differentiation in pytorch, 1st Conference on Neural Information Processing Systems (NIPS 2017), USA, 2017, pp.1-4
- [32]. Ketkar, N., Moolayil, J. Introduction to pytorch. In Deep learning with python, Apress, Berkeley, 2021, pp. 27-91
- [33]. Broughton, M., et al., Tensorflow quantum: A software framework for quantum machine learning." arXiv preprint arXiv:2003.02989, 2020, pp.3-56, <https://doi.org/10.48550/arXiv.2003.02989>
- [34]. Nelli, F., Deep learning with TensorFlow. In: Python Data Analytics. Apress, Berkeley, 2018, pp. 349-407.
- [35]. Chung, Y., et al., Comparison of Deep Learning Frameworks: About Theano, Tensorflow, and Cognitive Toolkit, Journal of Intelligence and Information Systems 23, no. 2, 2017, pp.1–17. doi:10.13088/JIIS.2017.23.2.001.

За контакти:

ас. д-р инж. Гинка К. Маринова
 катедра „Компютърни науки и технологии“
 Технически университет –Варна
 E-mail: gmarinova@tu-varna.bg

ХАКАТОН M-CODELAB НА ТЕМА “УМНИ ТЕХНОЛОГИИ ЗА ПОЧИСТВАНЕ И ОПАЗВАНЕ НА МОРСКАТА ЕКОСИСТЕМА” СЕ ПРОВЕДЕ НА 19 И 20 НОЕМВРИ 2022 Г. В ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - ВАРНА

Цветослава Р. Кьосева, Силвия И. Лешина, Бетина Р. Петкова

Резюме: На 19 и 20 ноември 2022 г. в гр. Варна софтуерната компания Methodia организира технологично състезание под шапката на образователната си инициатива Exceedia. Събитието бе организирано съвместно с Технически университет – Варна и проведено на територията на университета. Участниците разработваха своите проекти по тема “Умни технологии за почистване и опазване на морската екосистема”.

Ключови думи: хакатон, състезание по програмиране, екология, морска екосистема, технологии

Hackathon M-CODELAB on the topic "Smart technologies for cleaning and protection of the marine ecosystem" was held on November 19 and 20, 2022 in Technical University of Varna

Tsvetoslava R. Kyoseva, Silvia I. Leshina, Betina R. Petkova

Abstract: On November 19 and 20, 2022, in the city of Varna, the software company “Methodia” organized a technological competition under the umbrella of its “Exceedia” educational initiative. The event was organized jointly with the Technical University of Varna and held on the university's territory. The participants developed their projects on the topic "Smart technologies for cleaning and protecting the marine ecosystem".

Keywords: hackathon, programming competition, ecology, marine ecosystem, technology

1. Въведение

На 19 и 20 ноември 2022 г. в гр. Варна софтуерната компания Methodia организира технологично състезание под шапката на образователната си инициатива Exceedia. Събитието бе организирано съвместно с Технически университет – Варна и проведено на територията на университета. Участниците разработваха своите проекти по тема с голяма значимост за обществото, а именно “Умни технологии за почистване и опазване на морската екосистема”.

Включиха се студенти и ученици от 11 и 12 клас, които се разпределиха в 5 отбора:

Отбор 1 – Хакатонци

Отбор 2 – CyberTech

Отбор 3 – The Code of Duty

Отбор 4 – GreenBit

Отбор 5 – Save the Sea

2. Изложение

Темата на хакатона и четирите предизвикателства, около които отборите формулираха идеите на проектите си, бяха само основата, върху която младите таланти да изразят активно своето мнение по отношение на околната среда и нейното опазване. Вярваме, че първоначалният ентузиазъм на участниците преди събитието, които споделиха, че всеки от тях е „силно мотивиран да предизвика себе си и да стане част от намирането на дългосрочни решения за опазването на заобикалящите ни екосистеми“, се засили в края на събитието.

Подтеми, върху които участниците се фокусираха:

1. Изграждане на „еко град“ – свързано с тенденциите за намаляване на въглеродните емисии, умни решения при използването на природните ресурси, опазване на климата, разширяване на зелените площи и други.

2. Природни ресурси и енергийна ефективност – използване на природните ресурси за създаване на устойчиви алтернативи за производство на енергия като слънчева енергия, геотермална и вятърна енергия.

3. Добри практики за съхранение и опазване на черноморската екосистема като предотвратяване на производството на отпадъци и провеждане на разяснителни кампании за обществена информираност и ангажираност.

4. Влиянието на климатичните промени и предприемането на адекватни действия за подобряването на екологичните условия.

По време на хакатона участниците имаха възможността да реализират идеите си и технологични знания по един забавен и креативен начин, като имаха пълна свобода при избора за технологии и езици за програмиране. Също така през цялото време екип от вдъхновяващи ментори съдействаха със своите знания и опит.

А ето и идеите, които отборите представиха:

Отбор „Хакатонци“

Идеята им е насочена към създаване на приложение Trashn't по модел на социална мрежа, чиято цел е да се използва от всички като хъб за координация между хората, бизнеса и институциите в процеса от регистриране на локален замърсен участък до неговото изчистване и оставяне на „зелен отпечатък“. Първоначално включените функционалности са регистрация на потребител, локализиране на местоположение, създаване на събитие за информиране на потребителите.

При разработката на проекта са използвани Java & XML за мобилното приложение, C# (ASP.NET) за връзката с API, MySQL за базата от данни и C за хардуерната реализация на прототипа (фигура 1).

При API – Entity Framework за комуникация и работа с DB. Също така е използван и Swagger за по-лесна работа, дебъгване и тестване на API.

За мобилното приложение:

- OsmDroid (пакет на Open Street Map) за изчертаване на картата и различните компоненти по нея;
- Retrofit за имплементация на интерфейси за заявките към API;
- Picasso – за извличане на снимки от интернет, обработването им и изобразяването по екрана.

За кошчето:

Използва модул NodeMCU с процесор ESP32, който дава възможност за комуникация през интернет (WiFi) и сензор за движение. Използван е серийният интерфейс на модула за тестване и дебъгване. Там е имплементирана логика за четене и обработване на сигналите от сензора за движение и изпращането на данните към API (фигура 2).

```

1 using System.ComponentModel.DataAnnotations.Schema;
2 using System.ComponentModel.DataAnnotations;
3
4 namespace trashnt_web.Models
5 {
6     11 references | Tsvetelin Petrov, 11 days ago | 1 author, 3 changes
7     public class Trash
8     {
9         [Key]
10         3 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
11         public int Id { get; set; }
12         0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
13         public string Label { get; set; }
14         0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
15         public float Latitude { get; set; }
16         0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
17         public float Longitude { get; set; }
18         0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
19         public string GeoLabel { get; set; }
20         0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
21         public string Photo { get; set; }
22         0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
23         public int UserId { get; set; }
24         0 references | 0 changes | 0 authors, 0 changes
25         public User user { get; set; }
26         0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
27         public DateTime CreateDate { get; set; }
28         0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
29         public int Range { get; set; }
30         0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
31         public int Amount { get; set; }
32         1 reference | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
33         public int IsActive { get; set; }
34     }
35 }

```

Фиг. 1. Код от проекта на отбор „Хакатонци“

```

7 [ApiController]
8 [Route("api/[controller]")]
9 3 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
10 public class TrashesController : Controller
11 {
12     private readonly ITrashesService _trashes;
13     private readonly ILogger<TrashesController> _logger;
14
15     0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
16     public TrashesController(ILogger<TrashesController> logger, ITrashesService trashes)
17     {
18         _logger = logger;
19         _trashes = trashes;
20     }
21
22     [HttpGet("getAll")]
23     0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
24     public async Task<ActionResult<List<Trash>>> getAll()
25     {
26         return Ok(await _trashes.getAll());
27     }
28
29     [HttpGet("getActive")]
30     0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
31     public async Task<ActionResult<List<Trash>>> getActiveTrashes()
32     {
33         return Ok(await _trashes.getActive());
34     }
35
36     [HttpPost("getOneTrash")]
37     0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
38     public async Task<ActionResult<User>> checkUserLogin(Trash trash)
39     {
40         if(trash == null)
41             return BadRequest("Poluted area not found.");
42         var dbTrash = await _trashes.GetById(trash.Id);
43         if (dbTrash != null)
44             return Ok(dbTrash);
45         return BadRequest("Poluted area not found.");
46     }
47
48     [HttpPost("addImage")]
49     0 references | Tsvetelin Petrov, 11 days ago | 1 author, 1 change
50     public string UploadImage([FromForm] IFormFile file)
51     {
52     }
53 }

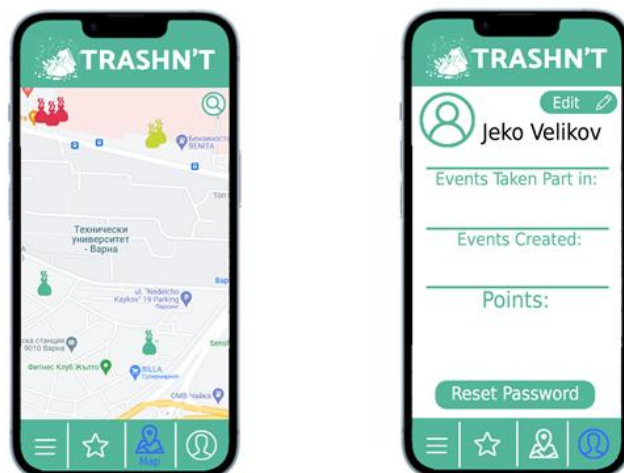
```

Фиг. 2. Код от проекта на отбор „Хакатонци“

При бъдещо развитие на приложението (фигура 3) може да се включат допълнителни стимули за постижения в почистването на локализиран проблем, да се разшири извън границите на Варна, добавяне на хардуерни компоненти за отчитане на ниво на боклук в

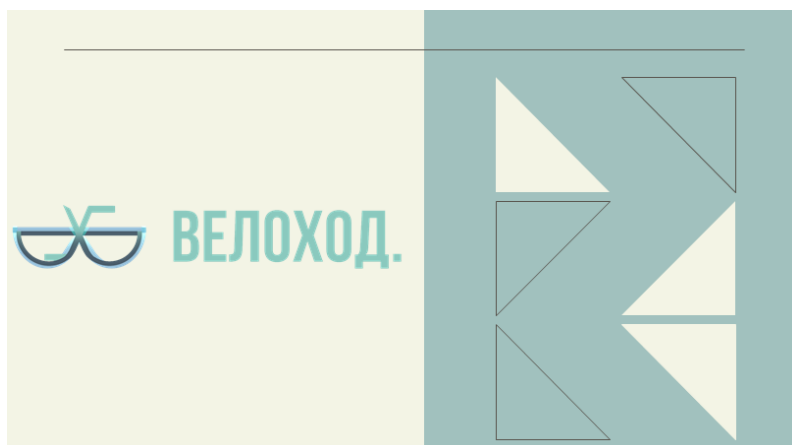
обществените контейнери а отпадъци и сигнализиране за тяхното изпразване чрез електромобили.

С поглед към бъдещето



Фиг. 3. Бъдещо развитие на отбор „Хакатонци“

Отбор „CyberTech“



Фиг. 4. Лого на проект „Велоход“

Отборът създаде сайт с името „Велоход“ (фигура 4). В него е включена статистика относно вредните емисии (фигура 5), които автомобилите отделят, за да могат да покажат и съществената си цел, а именно да стимулират повече хора да се придвижват с велосипед. Идеята е да се създаде приложение, в което да се визуализират свободни велосипеди в града и тяхната локация (фигура 6), за да бъде максимално улеснено и тяхното намиране и използване от хората. Самото приложение беше представено като прототип и дизайн, а потенциалът за развитие е със създаване на профил, функционалност за отчитане на време на ползване и изминато разстояние, калкулатор за изчисляване на брой събрана валута, която да може да се обменя за отстъпки в различни компании. Сайтът предвижда създаване на демо калкулатор, който изчислява в грамовете колко вредни емисии са спестени при използване на велосипед.

```

51     </div>
52     <div class="hr_border"></div>
53     <div class="col-md-2">
54         <div class="logo">
55             
56         </div>
57     </div>
58     <div class="col-md-10">
59         <div class="do_u_know">
60             <p>
61                 Знаете ли че, средно колата отделя 130 грама вредни емисии на 1 километър.
62             </p>
63         </div>
64     </div>
65 </div>
66 </section>
67
68 </header>
69 <section id="our_idea">
70     <div class="container">
71         <div class="row">
72             <div class="col-md-6">
73                 <div class="our_idea"></div>
74                 <h2>Идеята на нашия проект</h2>
75                 <div class="p1">
76                     <p>Целта на проекта е да намалим вредните емисии от колите чрез използването на велосипеди или други превозни средства, които не излъчват CO <span>
77                     <sub>2</sub></span></p>
78                     <!-- co2 na stepen + snimkata sushto -->
79                 </div>
80                 <div class="p2">
81                     <p>Чрез използване на нашите колелета имате шанс да получите ваучер за безплатно ползване или отстъпка до 50%. </p>
82                 </div>
83             </div>
84             <div class="col-md-6">
85                 
86             </div>
87         </div>
88     </div>
89 </section>
90
91 <section id="map">
92     <div class="bike">
93         </div>
94     <div class="container">
95         <div class="row">
96             <div class="col-md-8">
97                 <p>Тук може да намерите нашите колелета или другите превозни средства, които може да ползвате.</p>
98             </div>
99         </div>

```

Фиг. 5. Код от проекта на отбор „CyberTech”

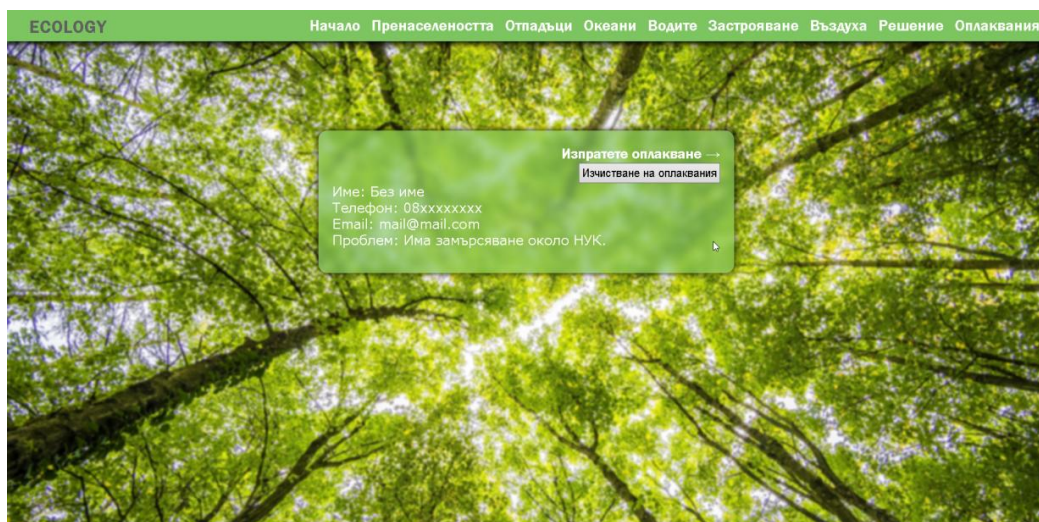
```

520 @media only screen and (max-width: 1200px) {
521     #we_love .we_love_title h2 {
522         font-size: 50px;
523     }
524     #we_love .we_love_title a {
525         margin-top: 0;
526         font-size: 20px;
527     }
528     #we_love .we_love_title a > span {
529         font-size: 14px;
530     }
531 }
532 @media only screen and (max-width: 992px) {
533     .main_nav ul li {
534         /*display: block !important;*/
535         margin-top: 20px;
536     }
537     .main_nav ul li a {
538         color: #fff !important;
539     }
540     #we_love .img img {
541         position: absolute;
542         left: 0px;
543         top: 0px;
544         z-index: -1;
545         width: 974px;
546         height: 429px;
547         opacity: 0.7;
548         background-repeat: no-repeat;
549         background-attachment: fixed;
550         background-size: cover;
551     }
552     #we_love .we_love_title h2 {
553         text-align: center;
554     }
555     footer,
556     footer .social_icon {
557         text-align: center;
558     }
559     footer p {
560         margin-top: 5px 0 10px;
561     }
562 }

```

Фиг. 6. Код от проекта на отбор „CyberTech”

Отбор „The Code of Duty“



Фиг. 7. Начална страница на уебсайта на отбор „The Code of Duty“

Отборът представи информативен сайт (фигура 7), който обхваща широка гама от проблеми и тяхното влияние върху екологията. Идеята е свързана с предоставяне на допълнителна информация относно причинителите на всеки от изброените проблеми и също така възможностите за тяхното решение. При разработването на проекта са използвани HTML, CSS, JavaScript като има възможност да се пренапише с React и да бъде добавена база от данни Firebase, където да се съхранява всеки потребител и неговата информация, а оплакването да отива при институция, която се грижи за екологията (фигура 8, фигура 9).

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title></title>
8     <link rel="stylesheet" href="./styles/style.css" />
9   </head>
10  <body>
11    <header>
12      <nav class="navigation">
13        <h1 class="navigation_title">Ecology</h1>
14        <ul class="navigation_list">
15          <li class="navigation_item">
16            <a href="#nachalo" onclick="onHashChange(event)">Начало</a>
17          </li>
18          <li class="navigation_item">
19            <a href="#problem" onclick="onHashChange(event)">Пренаселеността</a>
20          </li>
21          <li class="navigation_item">
22            <a href="#trash" onclick="onHashChange(event)">Отпадъци</a>
23          </li>
24          <li class="navigation_item">
25            <a href="#oceans" onclick="onHashChange(event)">Океани</a>
26          </li>
27          <li class="navigation_item">
28            <a href="#waters" onclick="onHashChange(event)">Водите</a>
29          </li>
30          <li class="navigation_item">
31            <a href="#towns" onclick="onHashChange(event)">Застрояване</a>
32          </li>
33          <li class="navigation_item">
34            <a href="#airpollution" onclick="onHashChange(event)">Въздуха</a>
35          </li>
36          <li class="navigation_item">
37            <a href="#reshenie" onclick="onHashChange(event)">Решение</a>
38          </li>
39          <li class="navigation_item">
40            <a href="#reports" onclick="onHashChange(event)">Оплаквания</a>
41          </li>
42        </ul>
43      </nav>
44    </header>
45    <main class="main">
46      <section class="nachalo">
47        
48        <h3 class="section_title">
49          Идеята на проекта е да засегнем проблемите с:
50        </h3>
```

Фиг. 8. Код от проекта на отбор „The Code of Duty“

```

1 function sendReport(event) {
2   event.preventDefault();
3   let name = document.getElementById('name').value;
4   let phone = document.getElementById('phone').value;
5   let email = document.getElementById('email').value;
6   let message = document.getElementById('message').value;
7   let dataArr = [];
8
9   dataArr.push(name);
10  dataArr.push(phone);
11  dataArr.push(email);
12  dataArr.push(message);
13  let uid = Math.floor(Math.random() * 10000000);
14  localStorage.setItem(uid, dataArr);
15  redirectToHome();
16 }
17
18 let reportList = document.getElementById('reports');
19 let data = Object.entries(localStorage);
20 data.forEach((element) => {
21   let reportArray = element[1].split(',');
22   let ulCard = document.createElement('ul');
23   let nameLi = document.createElement('li');
24   let phoneLi = document.createElement('li');
25   let emailLi = document.createElement('li');
26   let messageLi = document.createElement('li');
27   nameLi.textContent = 'Име: ' + reportArray[0];
28   phoneLi.textContent = 'Телефон: ' + reportArray[1];
29   emailLi.textContent = 'Email: ' + reportArray[2];
30   messageLi.textContent = 'Проблем: ' + reportArray[3];
31   ulCard.append(nameLi);
32   ulCard.append(phoneLi);
33   ulCard.append(emailLi);
34   ulCard.append(messageLi);
35   reportList.append(ulCard);
36 });
37
38 function clearReports() {
39   localStorage.clear();
40   location.reload();
41   window.location.href = 'index.html#nachalo';
42 }
43
44 if (localStorage.length > 0) {
45   document.getElementById('clearReports').style.display = 'block';
46 } else {
47   document.getElementById('clearReports').style.display = 'none';
48 }

```

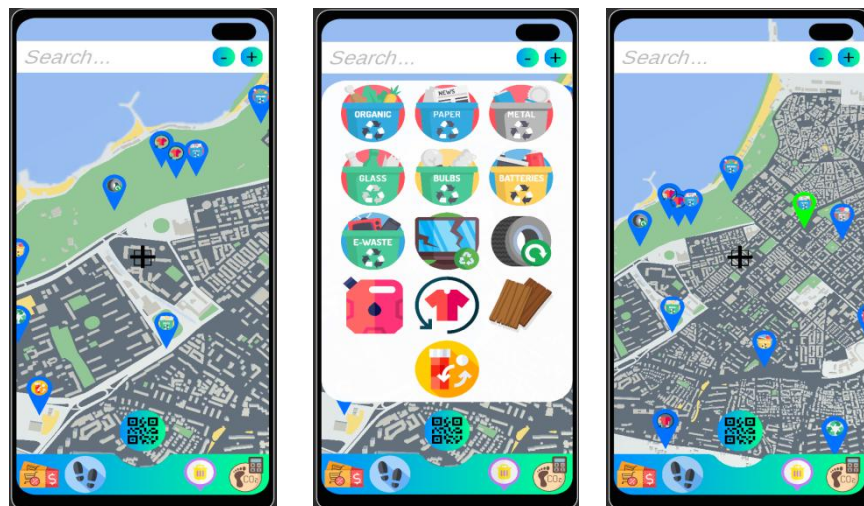
Фиг. 9. Код от проекта на отбор „The Code of Duty“

Отбор „GreenBit“

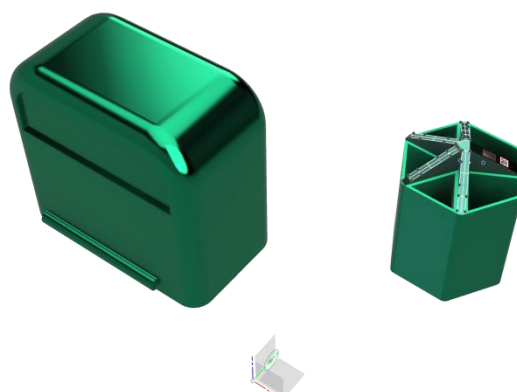
Цел на проекта е хората да бъдат мотивирани да опазват околната среда като събират отпадъците си разделно и ги рециклират правилно, за чиято цел е направен дизайн и прототип на умно (smart) кошче.

Идеята на умните кошчета е те лесно да могат да заменят обичайните и да бъдат интерактивни и с напълно самоподдържащо се хранване, осигурено от слънчеви панели. Тъй като те разчитат на разделното изхвърляне на отпадъци, кошчетата ще разполагат със модулни сензори, с които да засичат вида отпадък, поставен във тях. Чрез тях, нискоенергиен микроконтролер ще може да изведе информация за часа, локацията, запълването на контейнера, вида отпадък и наградата за потребителя във виртуални точки. Тази информация после ще се извежда под формата на QR код на електронномастилен дисплей, който ще се чете от мобилно приложение (фигура 10).

Логото е направено с CorelDRAW, а 3D моделите на кошчетата са изработени на AutoCAD и Fusion 360. UI дизайна е направен с Adobe XD (фигура 11).

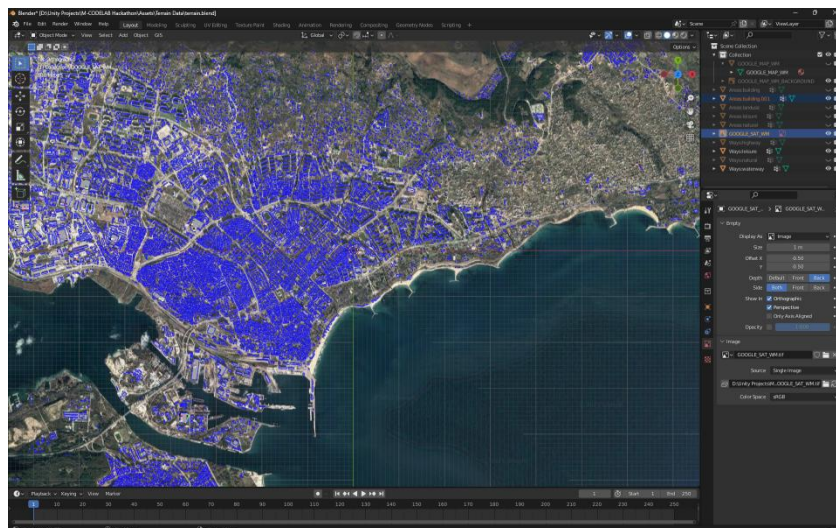


Фиг. 10. Скриншот от мобилното приложение на отбор „Greenbit“



Фиг. 11. Дизайн на „умно кошче“ на отбор „Greenbit“

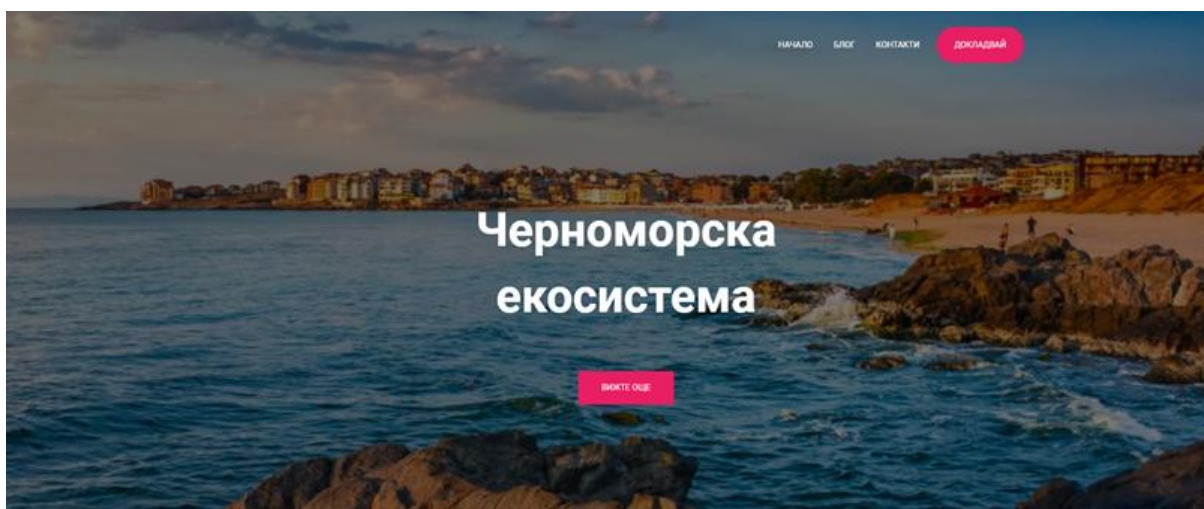
Използван е Unity 3D, за да се създаде среда, в която да се пресъздаде триизмерен модел на града. С помощта на add-on за blender, наречен BlenderGIS, се извлича сателитна карта на града с данни от Google Maps, разделена на няколко слоя, включително и триизмерни модели на сградите. Във Unity, вградените библиотеки предоставят услугата за местоположение от GPS апаратурата на мобилни устройства телефона, с които може да се свърже физическата локация на потребителя с пространството във виртуалната среда. Логото е направено с CorelDRAW, а 3D моделите на кошчетата са изработени на AutoCAD и Fusion 360. UI дизайна е направен с Adobe XD (фигура 12). Приложението също така се възползва от камерата на телефона, за да сканира QR кодовете от умните кошчета. Като допълнителен стимул за потребителя, програмата в по-нататъшна разработка, има възможността да следи активното движение, броя крачки и екологично спестеният въглероден диоксид.



Фиг. 12. Сателитна карта, използвана при разработката на проекта на отбор „Greenbit“

Потенциалът за разширяване и развитие на идеята е свързан с рециклирането на отпадъците и поддържане на градските зони чисти, което е стъпка към намаляване на въглеродния отпечатък. Проектът цели предоставяне на удобство на гражданите и подпомагане разделното изхвърляне чрез сензор, за разпознаване на материала на боклука (хартия, метал, пластмаса, стъкло, общ), разположен върху кошчето и захранващ се чрез слънчеви панели. При използване на кошчето се предвижда да бъде генериран QR код, чрез който хората могат да се възползват да получат пособия и материали от специално изградени диспенсъри (за хигиенни материали – салфетки, дезинфектант и др.), разположени в различни части на града.

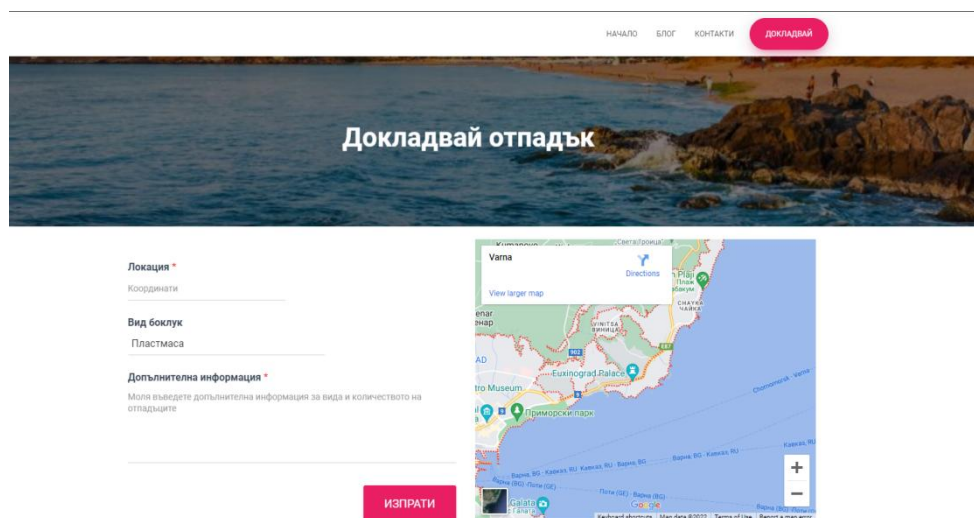
Отбор “Save the Sea“



Фиг. 13. Начална страница на уебсайта на отбор „Save the Sea“

Уебсайт (фигура 13), който предоставя информираност за проблемите на Черноморската екосистема. Сайтът е направен на WordPress, като е използван Elementor за изграждането на структурата и дизайна. Създадена е контактна форма, в която потребителят има възможност да локализира местоположението на отпадъка на картата и да го отбележи. За улеснение на докладващите отпадък, е направено drop-down меню, от което да се избере видът боклук.

Идеята на уебсайта е потребителите да се регистрират, след което да регистрират наличието на замърсяване (фигура 14). Потвърждаване създаването на профил до 24 ч. с цел да се регулират фалшиви профили. Може да се пренапише на JavaScript с React и да се добавят още функционалности.



Фиг. 14. Контактна форма от уебсайта на отбор „Save the Sea”

Заклучение

Тези два дни бяха изпълнени с много силни емоции както за участниците, така и за всички ментори и организатори. Класирането на отборите спрямо представените идеи и тяхната реализация беше изключително предизвикателство за журито. Отборите показаха висока мотивация, много знания и екипна работа, което доведе до едно много достойно представяне.

Затова и оценката на журито определи два отбора с еднакви резултати на първо място - „Хакатонци“ и „GreenBit“, отбор „CyberTech“ на второ, а „The Code of Duty“ и „Save the Sea“ заеха третото място с равен брой точки.

Източници

Показаните снимки и код, използвани като фигури за онагледяване на описаните проекти, са предоставени от самите автори на приложенията, участници в отборите от събитието, а именно:

1. Отбор „Хакатонци“
Цветелин Ангелов Петров
Кристиан Иванов Иванов
Найден Николаев Николов
Стефан Василев Филков
Иван Василев Филков

2. Отбор „CyberTech“
Преслав Даниелов Ковачев
Ямур Османлъ
Явор Руменов Нанков
Борис Руменов Василев

3. Отбор “The Code of Duty”
Стелиан Росенов Христов
Ангел Пламенов Ангелов
Любослав Пламенов Николов

4. Отбор „GreenBit“
Виктор Иванов Иванов
Саша Веселинова Ботева
Антон Евгениев Атанасов
Стефан Лучиян Повара

5. Отбор „Save The Sea“
Илиян Драгомиров Петров
Петър Станимиров Янков
Димитър Радославов Линов

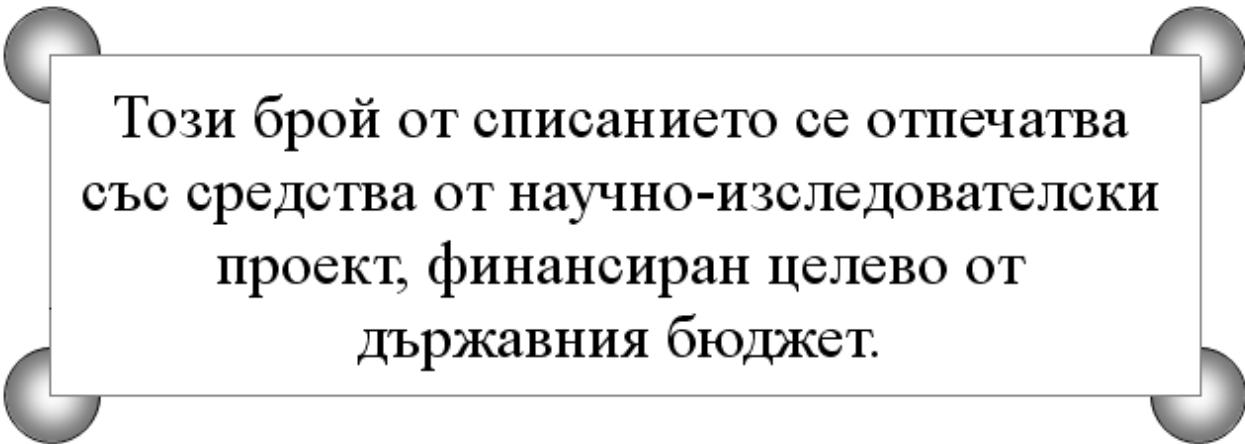
За контакти:

Силвия И. Лешина
Мениджър Развитие на Таланта
Методия АД
E-mail: silvia.leshina@methodia.com



ИЗИСКВАНИЯ ЗА ОФОРМЯНЕ НА СТАТИИТЕ ЗА СПИСАНИЕ "КОМПЮТЪРНИ НАУКИ И ТЕХНОЛОГИИ"

- I. Статиите се представят разпечатани в два екземпляра (оригинал и копие) в размер до 6 страници, формат А4 на адрес: Технически университет – Варна, ФИТА, ул. „Студентска” 1, 9010 Варна, както и в електронен вид на имейл адрес: yulka.petkova@tu-varna.bg.
 - II. Текстът на статията трябва да включва: УВОД (поставяне на задачата), ИЗЛОЖЕНИЕ (изпълнение на задачата), ЗАКЛЮЧЕНИЕ (получени резултати), БЛАГОДАРНОСТИ към сътрудниците, които не са съавтори на ръкописа (ако има такива), ЛИТЕРАТУРА и информация за контакти, включваща: научно звание и степен, име, организация, поделение (катедра), e-mail адрес.
 - III. Всички математически формули трябва да са написани ясно и четливо (препоръчва се използване на Microsoft Equation).
 - IV. Текстът трябва да бъде въведен във файл във формат WinWord 2000/2003 с шрифт Times New Roman. Форматирането трябва да бъде както следва:
 1. Размер на листа - А4, полета: ляво - 20мм, дясно - 20мм, горно - 15мм, долно - 35мм, Header 12.5мм, Footer 12.5мм (1.25см).
 2. Заглавие на български език - размер на шрифта 16, удебелен, главни букви.
 3. Един празен ред - размер на шрифта 14, нормален.
 4. Имена на авторите - име, инициали на презиме, фамилия, без звания и научни степени - размер на шрифта 14, нормален.
 5. Два празни реда - размер на шрифта 14, нормален.
 6. Резюме и ключови думи на български език, до 8 реда - размер на шрифта 11, нормален.
 7. Заглавие на английски език - размер на шрифта 12, удебелен.
 8. Един празен ред - размер на шрифта 11, нормален.
 9. Имена на авторите на английски език - размер на шрифта 11, нормален.
 10. Един празен ред - размер на шрифта 11, нормален.
 11. Резюме и ключови думи на английски език, до 8 реда - размер на шрифта 11, нормален.
 12. Основните раздели на статията (Увод, Изложение, Заключение, Благодарности, Литература) се формират в едноколонен текст както следва:
 - a. Наименование на раздел или на подраздел - размер на шрифта 12, удебелен, центриран, един празен ред преди наименованието и един празен ред след него - размер на шрифта 12, нормален;
 - b. Текст - размер на шрифта 12, нормален, отстъп на първи ред на параграф – 10 мм; разстояние от параграф до съседните (Before и After) за целия текст – 0.
 - c. Цитиране на литературен източник - номер на източника от списъка в квадратни скоби;
 - d. Текстът на формулите се позиционира в средата на реда. Номерация на формулите - дясно подравнена, в кръгли скоби.
 - e. Фигури - центрирани, разположение спрямо текста: “Layout: In line with text”. Номер и наименование на фигурата - размер на шрифта 11, нормален, центриран. Отстояние от съседните параграфи – 6 pt.
 - f. Литература – всеки литературен източник се представя с: номер в квадратни скоби и точка, списък на авторите (първият автор започва с фамилия, останалите – с име), заглавие, издателство, град, година на издаване, страници.
 - g. За контакти: научно звание и степен, име, презиме (инициали), фамилия, организация, поделение (катедра), e-mail адрес, с шрифт 11, дясно подравнено.
- Образец за форматиране можете да изтеглите от адрес <http://cs.tu-varna.bg/> - Списание КНТ, Spisanie_Obrazec.zip.



Този брой от списанието се отпечатва
със средства от научно-изследователски
проект, финансиран целево от
държавния бюджет.