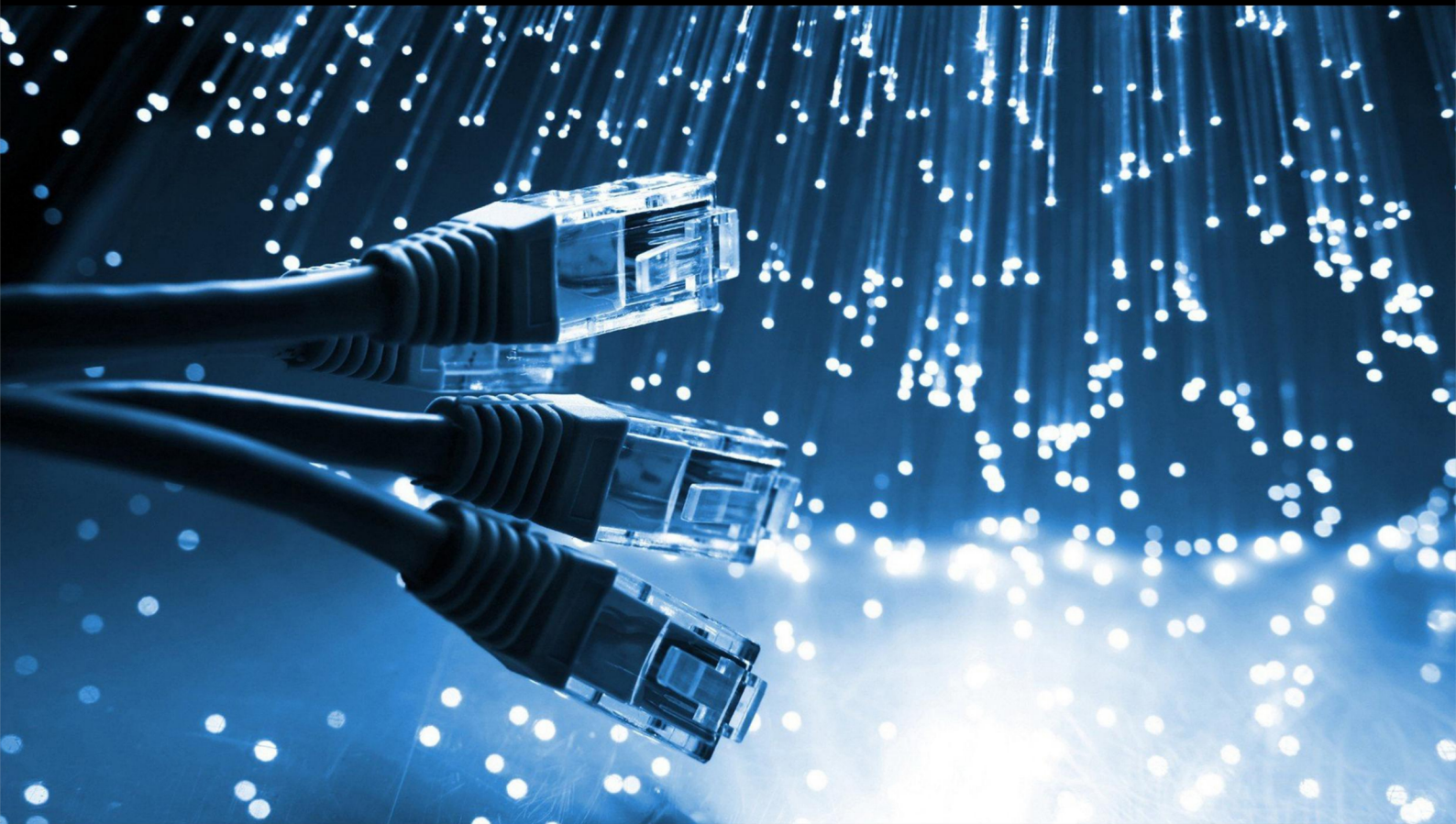


JCSI

Journal of Computer Sciences Institute

Volume 15/2020



Department of Computer Science
Lublin University of Technology

jcsi.pollub.pl

ISSN: 2544-0764

Redakcja JCSI

e-mail: jcsi@pollub.pl
www: jcsi.pollub.pl
Katedra Informatyki
Wydział Elektrotechniki i Informatyki

Politechnika Lubelska
ul. Nadbystrzycka 36 b
20-618 Lublin

Redaktor naczelny:

Tomasz Zientarski
e-mail: t.zientarski@pollub.pl

Redaktor techniczny:

Beata Pańczyk,
e-mail: b.panczyk@pollub.pl

Recenzenci numeru:

dr hab. Małgorzata Charytanowicz, prof. PL
dr inż. Marek Miłosz, prof. PL
dr inż. Sławomir Przyłucki
dr inż. Dariusz Gutek
dr inż. Tomasz Szymczyk
dr inż. Maria Skublewska-Paszkowska
dr Edyta Łukasik
dr inż. Elżbieta Miłosz
dr inż. Jacek Kęsik
dr Mariusz Dzieńkowski
dr inż. Marcin Badurowicz
dr inż. Jakub Smółka
dr inż. Krzysztof Dziedzic

Skład komputerowy:

Monika Kaczorowska
e-mail: m.kaczorowska@pollub.pl

Projekt okładki:

Marta Zbańska

JCSI Editorial

e-mail: jcsi@pollub.pl
www: jcsi.pollub.pl
Department of Computer Science
Faculty of Electrical Engineering and
Computer Science
Lublin University of Technology
ul. Nadbystrzycka 36 b
20-618 Lublin, Poland

Editor in Chief:

Tomasz Zientarski
e-mail: t.zientarski@pollub.pl

Assistant editor:

Beata Pańczyk,
e-mail: b.panczyk@pollub.pl

Reviewers:

Małgorzata Charytanowicz
Marek Miłosz
Sławomir Przyłucki
Dariusz Gutek
Tomasz Szymczyk
Maria Skublewska-Paszkowska
Edyta Łukasik
Elżbieta Miłosz
Jacek Kęsik
Mariusz Dzieńkowski
Marcin Badurowicz
Jakub Smółka
Krzysztof Dziedzic

Computer typesetting:

Monika Kaczorowska
e-mail: m.kaczorowska@pollub.pl

Cover design:

Marta Zbańska

Spis treści

1. SIECI NEURONOWE W ROZPOZNAWANIU PISMA ODRĘCZNEGO BERNADETTA MICHALIK, MAREK MIŁOSZ.....	109-113
2. WPŁYW JAKOŚCI INTERFEJSU SKLEPÓW INTERNETOWYCH NA ICH ZAPAMIĘTYWALNOŚĆ KATARZYNA PYTKA, MAREK MIŁOSZ.....	114-120
3. PORÓWNANIE METODY UWIERZYTELNIANIA NFC Z METODAMI DOSTĘPNYMI W ANDROID DMYTRO MALENKO.....	121-125
4. ANALIZA PORÓWNAWCZA SYSTEMÓW BAZ DANYCH DEDYKOWANYCH DLA SYSTEMU ANDROID KAMIL WAŁACHOWSKI, GRZEGORZ KOZIEL.....	126-132
5. ANALIZA PORÓWNAWCZA USŁUG CYFROWYCH MIĘDZY POLSKĄ, A DALEKIM WSCHODEM DOMINIK KLEJ.....	133-138
6. NOWOCZESNE TECHNOLOGIE TWORZENIA GRAFICZNEGO INTERFEJSU UŻYTKOWNIKA W APLIKACJACH INTERNETOWYCH MATEUSZ KAPROŃ, BEATA PAŃCZYK.....	139-142
7. WERYFIKACJA DOKŁADNOŚCI SYGNAŁÓW GSM I GPS DOSTĘPNYCH NA URZĄDZENIACH MOBILNYCH MATEUSZ WŁAŻ, GRZEGORZ KOZIEL.....	143-149
8. KORPORACYJNA PLATFORMA INTRANETOWA - PORÓWNANIE ROZWIĄZANIA OTWARTEGO I KOMERCYJNEGO KONRAD CZARNY.....	150-155
9. ANALIZA PORÓWNAWCZA ROZWIĄZAŃ WYKORZYSTYWANYCH W TESTOWANIU AUTOMATYCZNYM AGNIESZKA DOROTA WAC, TOMASZ KAMIL WATRAS, GRZEGORZ KOZIEL.....	156-163
10. ANALIZA PODATNOŚCI UŻYTKOWNIKÓW SYSTEMÓW INFORMATYCZNYCH NA ATAK PHISHINGOWY SZYMON GRZEBIELEC.....	164-167
11. INTERFEJSY BEZDOTYKOWE W WIRTUALNEJ RZECZYWISTOŚCI ANDRIY DRAGNI, TOMASZ SZYMCZYK.....	168-171
12. SYSTEM WYKORZYSTUJĄCY TECHNOLOGIE MOBILNE SŁUŻĄCY DO TESTOWANIA CZASU REAKCJI MICHAŁ WOŚ, ARTUR ZDZIOCH.....	172-177
13. ANALIZA PORÓWNAWCZA WYBRANYCH WERSJI SZKIELETU PROGRAMISTYCZNEGO SYMFONY MICHAŁ JUSIĘGA, MARIUSZ DZIENKOWSKI.....	178-183
14. METODY KONTENERYZACJI W ZASTOSOWANIACH USŁUG INTERNETOWYCH KRZYSZTOF FERENC.....	184-188
15. ANALIZA WYDAJNOŚCI JĘZYKÓW DZIAŁAJĄCYCH NA JAVA VIRTUAL MACHINE NA PRZYKŁADZIE JĘZYKÓW JAVA, SCALA I KOTLIN KATARZYNA BUSZEWICZ.....	189-195
16. PLATFORMA E-LEARNINGOWA „QUIZER” JAKO NARZĘDZIE TWORZENIA INTERAKTYWNYCH QUIZÓW Z FUNKCJONALNOŚCIĄ MULTIPLAYER KRZYSZTOF DZIEDZIC, MAREK GUDKÓW, TOMASZ WIŚNIEWSKI.....	196-201
17. ANALIZA TYPOWYCH BŁĘDÓW PROGRAMISTYCZNYCH POPEŁNIANYCH PRZEZ STUDEN-TÓW PIERWSZEGO I DRUGIEGO ROKU INFORMATYKI MONIKA KACZOROWSKA.....	202-205
18. KLASYFIKACJA OBIEKTÓW NA PODSTAWIE ICH ZDJĘĆ RENTGENOWSKICH PIOTR NOWOSAD, MAŁGORZATA CHARYTANOWICZ.....	206-213
19. CZĘŚCIOWO NADZOROWANE UCZENIE Z ZASTOSOWANIEM KLASTERYZACJI ORAZ KLASYFIKATORA DRZEW DECYZYJNYCH W PRZYPADKU BADANIA OBCIĄŻENIA POZNAWCZEGO MARTYNA WAWRZYK.....	214-218

Contents

1. NEURAL NETWORKS IN RECOGNITION OF HANDWRITING BERNADETTA MICHALIK, MAREK MIŁOSZ.....	109-113
2. INFLUENCE THE QUALITY OF THE INTERFACE OF ONLINE STORES ON THEIR MEMORABILITY KATARZYNA PYTKA, MAREK MIŁOSZ.....	114-120
3. COMPARISON OF THE NFC AUTHENTICATION METHOD WITH THE METHODS AVAILABLE IN ANDROID DMYTRO MALENKO.....	121-125
4. COMPARATIVE ANALYSIS OF DATABASE SYSTEMS DEDICATED FOR ANDROID KAMIL WAŁACHOWSKI, GRZEGORZ KOZIEL.....	126-132
5. DIGITAL SERVICES IN POLAND AND THE FAR EAST - COMPARATIVE ANALYSIS DOMINIK KLEJ.....	133-138
6. MODERN TECHNOLOGIES FOR CREATING GRAPHICAL USER INTERFACE IN WEB APPLICATIONS MATEUSZ KAPROŃ, BEATA PAŃCZYK.....	139-142
7. VERIFICATION OF THE PRECISION OF GSM AND GPS SIGNALS AVAILABLE ON MOBILE DEVICES MATEUSZ WŁAŻ, GRZEGORZ KOZIEL.....	143-149
8. CORPORATE INTRANET PLATFORM – OPEN AND COMMERCIAL SOLUTIONS COMPARISON KONRAD CZARNY.....	150-155
9. COMPARATIVE ANALYSIS OF SOLUTIONS USED IN AUTOMATED TESTING AGNIESZKA DOROTA WAC, TOMASZ KAMIL WATRAS, GRZEGORZ KOZIEL.....	156-163
10. ANALYSIS OF THE VULNERABILITY OF IT SYSTEM USERS TO A PHISHING ATTACK SZYMON GRZEBIELEC.....	164-167
11. CONTACTLESS INTERFACES IN VIRTUAL REALITY ANDRIY DRAGNI, TOMASZ SZYMCZYK.....	168-171
12. THE SYSTEM USING MOBILE TECHNOLOGIES TO TEST REACTION TIME MICHAŁ WOŚ, ARTUR ZDZIOCH.....	172-177
13. COMPARATIVE ANALYSIS OF A SELECTED VERSION OF THE SYMFONY FRAMEWORK MICHAŁ JUSIEGA, MARIUSZ DZIENKOWSKI.....	178-183
14. CONTAINERIZATION METHODS IN INTERNET SERVICE APPLICATIONS KRZYSZTOF FERENC.....	184-188
15. PERFORMANCE ANALYSIS OF LANGUAGES WORKING ON JAVA VIRTUAL MACHINE BASED ON JAVA, SCALA AND KOTLIN KATARZYNA BUSZEWICZ.....	189-195
16. THE "QUIZER" E-LEARNING PLATFORM AS A TOOL FOR CREATING INTERACTIVE QUIZZES WITH MULTIPLAYER FUNCTIONALITY KRZYSZTOF DZIEDZIC, MAREK GUDKÓW, TOMASZ WIŚNIEWSKI.....	196-201
17. ANALYSIS OF TYPICAL PROGRAMMING MISTAKES MADE BY FIRST AND SECOND YEAR IT STUDENTS MONIKA KACZOROWSKA.....	202-205
18. OBJECT CLASSIFICATION USING X-RAY IMAGES PIOTR NOWOSAD, MAŁGORZATA CHARYTANOWICZ.....	206-213
19. SEMI-SUPERVISED LEARNING WITH THE CLUSTERING AND DECISION TREES CLASSIFIER FOR THE TASK OF COGNITIVE WORKLOAD STUDY MARTYNA WAWRZYK.....	214-218

Sieci neuronowe w rozpoznawaniu pisma odręcznego

Neural networks in recognition of handwriting

Bernadetta Michalik*, Marek Miłośz

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

Artificial neural networks consist of many simple elements capable of processing data. These are tools inspired by the construction of the human brain, used in machine learning. The aim of the research was to analyze the accuracy of the created neural network in the process of handwriting recognition. The article presents the results obtained during the learning and testing of a convolution network with a different number of hidden layers. Each time learning and testing the network was carried out using the same set of images (taken from the publicly available IAM database) depicting handwritten words in English.

Keywords: handwriting; artificial neural network; word recognition

Streszczenie

Sztuczne sieci neuronowe składają się z wielu prostych elementów zdolnych do przetwarzania danych. To narzędzia inspirowane budową ludzkiego mózgu, stosowane w uczeniu maszynowym. Celem badań była analiza dokładności odpowiedzi stworzonej sieci neuronowej w procesie rozpoznawania pisma odręcznego. W artykule przedstawiono wyniki uzyskane podczas nauki i testowania sieci konwolucyjnej o różnej liczbie warstw ukrytych. Każdorazowo uczenie i testowanie sieci realizowane było za pomocą tego samego zbioru obrazów (zaczepniętych z ogólnodostępnej bazy IAM Handwriting Database) przedstawiających słowa pisane odręcznie w języku angielskim.

Słowa kluczowe: sztuczne sieci neuronowe; rozpoznawanie słów

*Corresponding author

Email address: bernadetta768@gmail.com (B. Michalik)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Sztuczne sieci neuronowe (SSN) są narzędziami do modelowania obliczeniowego, które pojawiły się w wielu dyscyplinach w zakresie kształtowania złożonych problemów w świecie rzeczywistym. SSN można zdefiniować jako struktury złożone z gęsto połączonych prostych elementów przetwarzających (zwanych sztucznymi neuronami lub węzłami), które są zdolne do wykonywania masowo równoległych obliczeń w celu przetwarzania danych i reprezentacji wiedzy [1]. Mimo że SSN są abstrakcjami odpowiedników biologicznych, ich ideą nie jest replikacja działania systemów biologicznych, ale wykorzystanie wiedzy o funkcjonowaniu sieci biologicznych do rozwiązywania złożonych problemów. Atrakcyjność SSN wynika z ich niezwykłych właściwości takich jak nieliniowość, wysoka równoległość, odporność na awarie i uszkodzenia, zdolność do obsługi nieprecyzyjnych i rozmytych informacji oraz ich zdolność do generalizowania [2].

Obecnie coraz więcej osób korzysta z obrazów do reprezentowania i przesyłania informacji. Popularne jest również wydobywanie ważnych informacji z obrazów. Rozpoznawanie obrazu jest ważnym obszarem badawczym ze względu na jego szerokie zastosowania. W stosunkowo młodej dziedzinie komputerowego rozpoznawania wzorców jednym z trudniejszych zadań jest dokładne rozpoznawanie ludzkiego pisma, ponieważ istnieje znaczna różnorodność pisma między każdym z osobna. Chociaż człowiek zazwyczaj nie ma z tym żadnych problemów, trudniej jest nauczyć kom-

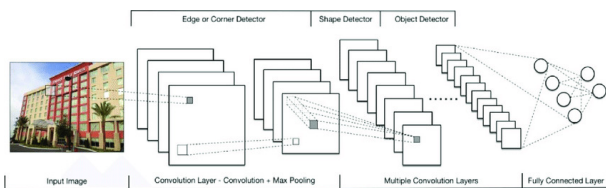
putery rozpoznawać odręczne pismo. Jednym z rozwiązań jest użycie optycznego rozpoznawania znaków (OCR) do konwersji zeskanowanych dokumentów papierowych na formę do odczytu maszynowego (ASCII). Technologia ta umożliwia rozróżnianie drukowanych lub odręcznych znaków tekstowych wewnątrz cyfrowych obrazów fizycznych dokumentów. Odbywa się to poprzez zrobienie najpierw zdjęcia dokumentu lub zeskanowanie go. Tworzy to obraz rastrowy złożony z danych, które komputer rozumie, a przez specjalnie zaprogramowane algorytmy, z których większość jest wykorzystywana w dziedzinie sztucznej inteligencji, komputer rozpoznaje wzorce na obrazie. Następnie program tworzy lub wyprowadza kody znaków, zwykle ASCII, które są równoważne rozpoznany znakom z obrazu wejściowego. Większość programów OCR musi zostać przeszkolonych, aby mogły lepiej rozpoznawać znaki [3].

2. Głębokie uczenie

Technologia uczenia maszynowego zasila wiele aspektów współczesnego społeczeństwa: od wyszukiwania w sieci przez filtrowanie treści w sieciach społecznościowych po rekomendacje na stronach e-commerce i jest coraz bardziej obecna w produktach konsumentskich, takich jak aparaty fotograficzne i smartfony. Systemy uczenia maszynowego służą do identyfikacji obiektów na obrazach, transkrypcji mowy na tekst, dopasowania wiadomości, postów lub produktów do zainteresowań użytkowników oraz wyboru odpowied-

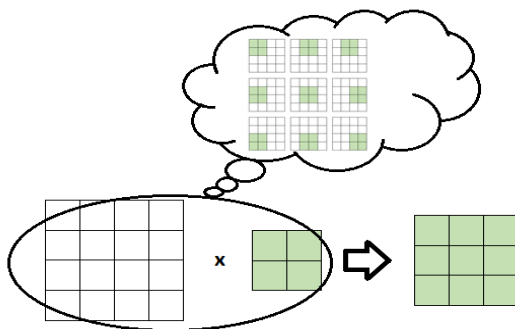
nich wyników wyszukiwania. Coraz częściej aplikacje te korzystają z klasy technik zwanych głębokim uczeniem się, które potrafią przetwarzać surowe dane do wektora cech. Uczenie się reprezentacji to zestaw metod, które pozwalają zasilać maszynę surowymi danymi i automatycznie odkrywać reprezentacje potrzebne do wykrywania lub klasyfikacji [4]. Metody dogłębnego uczenia się to metody uczenia się reprezentacji z wieloma poziomami reprezentacji, uzyskiwane przez złożenie prostych, ale nieliniowych modułów, z których każdy przekształca reprezentację na jednym poziomie w reprezentację na wyższym, nieco bardziej abstrakcyjnym poziomie. Dzięki wystarczającej liczbie takich transformacji można nauczyć się bardzo złożonych funkcji. W przypadku zadań klasyfikacyjnych wyższe warstwy reprezentacji wzmocniają aspekty danych wejściowych, które są ważne dla dyskryminacji i eliminują nieistotne zmiany [5].

W ostatnich latach coraz częściej wykorzystuje się konwolucyjne sieci neuronowe (CNN, ConvNets), narzędzia do głębokiego uczenia się. Są one szczególnie odpowiednie dla obrazów jako danych wejściowych. Na wejściu mogą się również pojawić dane w postaci tekstu, sygnałów i inne. CNN składają się z trzech rodzajów warstw. Są to warstwy spłotowe, warstwy pulujące i warstwy w pełni połączone [6]. Uproszczona architektura CNN została zilustrowana na rysunku 1.



Rysunek 1: Schemat konwolucyjnej sieci neuronowej [7]

Warstwa spłotowa jest miejscem, gdzie wykonuje się operacje matematyczne na macierzach, z których jedna z nich jest zbiorem danych wejściowych a druga jądrem.



Rysunek 2: Procedura mnożenia danych wejściowych i wag

Po lewej stronie diagramu (rysunek 2) znajduje się macierz obrazu (tablica 4x4 w kolorze białym), pośrodku znajduje się jądro (tablica 2x2 w kolorze zielonym), a po prawej wynik spłotu. Szczegóły przejścia jądra przez macierz obrazu przedstawiono w chmurze na diagramie. Oto, co się stało: filtr odczytuje kolejno, od lewej do prawej i od góry do dołu wszystkie piksele

w obszarze działania jądra i pomnożył wartość każdego z nich przez odpowiednią wartość jądra. Wynikiem działań warstwy spłotowej jest mapa aktywacyjna [8].

Głównym zadaniem warstwy łączącej jest próbowanie w dół w celu zmniejszenia złożoności kolejnych warstw, a tym samym dalsze zmniejszanie liczby parametrów i złożoności obliczeniowej modelu. W dziedzinie przetwarzania obrazu można to uznać za podobne do procesu zmniejszania rozdzielczości. Warstwy pulujące wprowadza się do modelu, przeciwdziałając przeuczeniu [9]. Ostatnim elementem jest w pełni połączona warstwa, która zawiera neurony bezpośrednio połączone z neuronami w dwóch sąsiadujących warstwach, bez połączenia z żadnymi warstwami w nich zawartymi. Jest to analogiczne do sposobu, w jaki neurony są ułożone w tradycyjnych formach sztucznych sieci neuronowych [10].

3. Realizacja badań

Badania zaprezentowanego poniżej modelu zostały przeprowadzone na urządzeniu o parametrach przedstawionych na rysunku 3. Dodatkowo, aby nie zakłócać przebiegu testów, wszystkie niepotrzebne procesy zostały zamknięte.

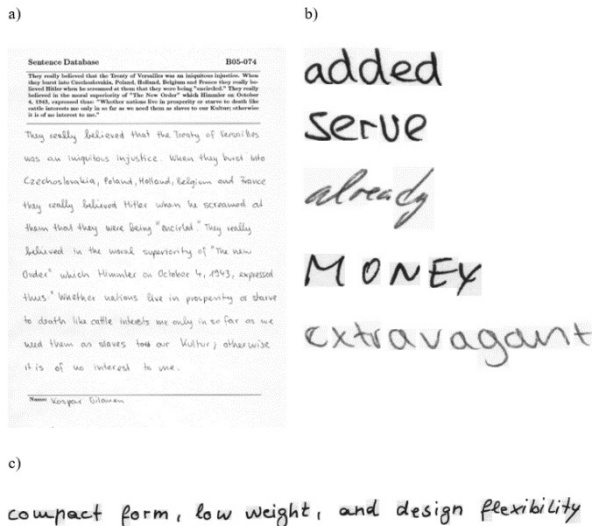
System	
Procesor:	Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz 2.40 GHz
Zainstalowana pamięć (RAM):	8,00 GB
Typ systemu:	64-bitowy system operacyjny, procesor x64

Rysunek 3: Parametry urządzenia.

3.1. Zestaw danych

W badaniach wykorzystano zasoby bazy IAM (rys. 4). Jest to zbiór zdjęć przedstawiających odręcznie napisane teksty w języku angielskim. Pierwsza publikacja bazy IAM ukazała się w 1999 roku. Zbiór wykorzystywany jest głównie do nauki rozpoznawania pisma odręcznego jak również do badań nad identyfikacją autora. Do utworzenia bazy przyczyniło się ponad 600 autorów. Składa się z 1539 stron odręcznie pisanego tekstu, 5685 wyodrębnionych zdań, 13353 linii z tekstem i 115320 słów. Zasoby bazy to zeskanowane dokumenty zapisane w formacie png, które posiadają 256 poziomów szarości i rozdzielczość 300 dpi [11].

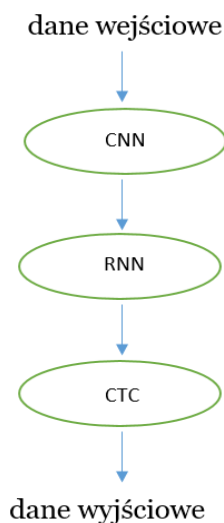
Na potrzeby pracy wykorzystano jedynie zbiór wyodrębnionych słów, które podzielono na trzy zestawy: treningowy, walidacyjny i testowy. Pierwszy z nich stanowi 70 % całego zbioru pojedynczych słów, drugi to 20 %, natomiast reszta (10 %) jest zbiorem testowym.



Rysunek 4: Przykład zasobów bazy IAM: strona (a), wiersz (c) i pojedyncze słowa (b) [11].

3.2. Model sieci neuronowej

Skonstruowano model sieci neuronowej, który stanowi połączenie trzech warstw: CNN (convolutional neural network), RNN (recurrent neural network) i CTC (connectionist temporal classification). Uproszczony schemat sieci przedstawiono na rysunku 5.



Rysunek 5: Schemat modelu

Na wejściu sieci konwolucyjnej znajduje się zbiór, który składa się z szarych obrazów. Ze względu na różną długość wyrazów przyjęto jedną (uniwersalną) wielkość, która wynosi 128 x 32. Obrazy o innych rozmiarach zostały przekształcone do rozmiarów uniwersalnych.

Listing 1: Definicja sieci konwolucyjnej z pięcioma warstwami ukrytymi

```
def conv_net(self):
    conv = tf.nn.conv2d(self.input,
    filter=tf.Variable(tf.random.truncated_normal([
    3, 3, 1, 16], stddev=0.5)), padding='SAME',
    strides=(1,1,1,1))
    relu = tf.nn.relu(conv)
```

```
pool = tf.nn.max_pool2d(relu, (1, 2, 2, 1), (1,
2, 2, 1), 'SAME')
conv2 = tf.nn.conv2d(pool,
filter=tf.Variable(tf.random.truncated_normal([
3, 3, 16, 32], stddev=0.5)), padding='SAME',
strides=(1,1,1,1))
relu2 = tf.nn.relu(conv2)
pool2 = tf.nn.max_pool2d(relu2, (1, 2, 2, 1),
(1, 2, 2, 1), 'SAME')
conv3 = tf.nn.conv2d(pool2,
filter=tf.Variable(tf.random.truncated_normal([
3, 3, 32, 64], stddev=0.5)), padding='SAME',
strides=(1,1,1,1))
relu3 = tf.nn.relu(conv3)
pool3 = tf.nn.max_pool2d(relu3, (1, 1, 2, 1),
(1, 1, 2, 1), 'SAME')
conv4 = tf.nn.conv2d(pool3,
filter=tf.Variable(tf.random.truncated_normal([
3, 3, 64, 128], stddev=0.5)), padding='SAME',
strides=(1,1,1,1))
relu4 = tf.nn.relu(conv4)
pool4 = tf.nn.max_pool2d(relu4, (1, 1, 2, 1),
(1, 1, 2, 1), 'SAME')
conv5 = tf.nn.conv2d(pool4,
filter=tf.Variable(tf.random.truncated_normal([
3, 3, 128, 256], stddev=0.5)), padding='SAME',
strides=(1,1,1,1))
relu5 = tf.nn.relu(conv5)
pool5 = tf.nn.max_pool2d(relu5, (1, 1, 2, 1),
(1, 1, 2, 1), 'SAME')
```

Listing 1 opisuje sieć konwolucyjną zawierającą pięć warstw ukrytych. Składa się z warstwy spłotowej, ReLU (rectified linear Units) oraz warstwy pulującej. Dane z wyjścia sieci spłotowej są jednocześnie danymi wejściowymi sieci RNN, która składa się z dwóch warstw (listing 2).

Listing 2: Definicja sieci rekurencyjnej

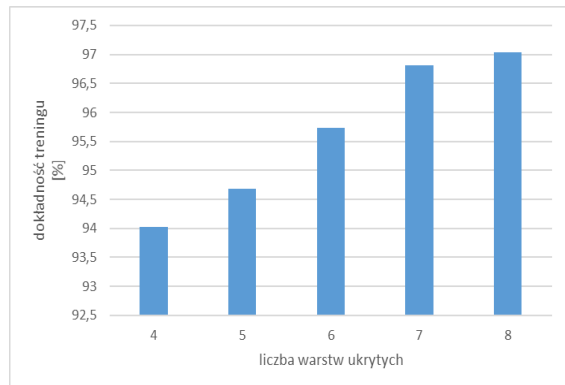
```
def rnn_net(self):
    rnn_size = 256
    cell1 = tf.contrib.rnn.LSTMCell(rnn_size)
    cell2 = tf.contrib.rnn.LSTMCell(rnn_size)
    cell = tf.contrib.rnn.MultiRNNCell([cell1,
    cell2], state_is_tuple=True)
    ([encoder_outputs, encoder_state], _) =
    tf.nn.bidirectional_dynamic_rnn(cell_fw=cell,
    cell_bw=cell, inputs=tf.squeeze(self.pool5,
    axis=[2]), dtype=tf.float32)
    concat =
    tf.expand_dims(tf.concat([encoder_outputs,
    encoder_state], 2), 2)
    self.rnnOutput =
    tf.squeeze(tf.nn.atrous_conv2d(value=concat,
    filters=tf.Variable(tf.random.truncated_normal([
    1, 1, rnn_size * 2, len(self.charList) + 1],
    stddev=0.5)), rate=1, padding='SAME'))
```

4. Wyniki

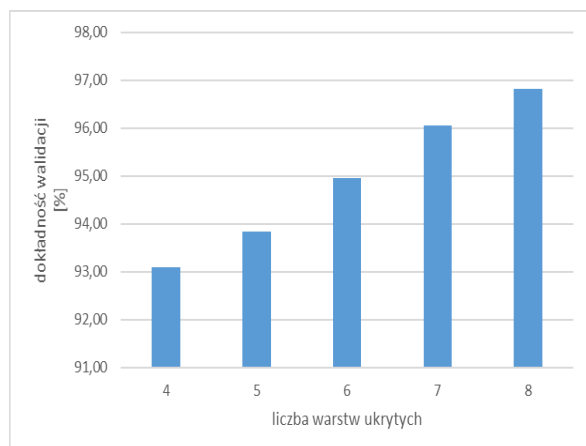
Na rysunku 5, 6 i 7 przedstawiono dokładność uzyskaną podczas poszczególnych etapów pracy sieci neuronowej przy różnej liczbie warstw ukrytych. W każdym przypadku zauważono, że wraz ze wzrostem liczby warstw ukrytych rośnie dokładność modelu. Najwyższe wyniki uzyskano podczas działań na zestawie treningowym, czyli na danych służących dopasowaniu wag. Nieco niższe wyniki, ale nadal zadowalające, uzyskano podczas walidacji. Rzeczywistą dokładność modelu przed-

stawia rysunek 10. Nauczona sieć jest sprawdzana z wykorzystaniem zbioru testującego. Wszystkie wyniki przekraczają próg 90 %. Najwyższe wyniki uzyskano dla modelu z siedmioma (95,92 %) i ośmioma (96,77 %) warstwami ukrytymi sieci konwolucyjnej.

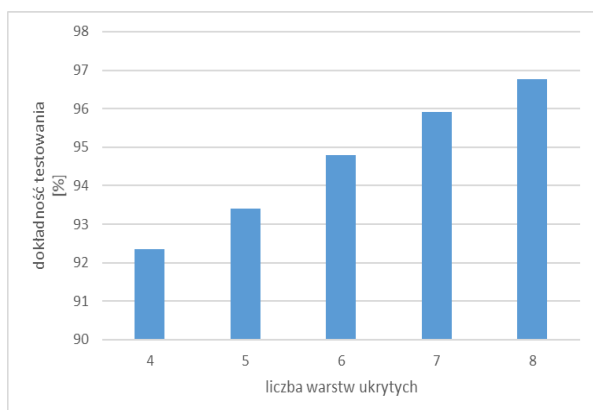
Szczegółowe wyniki przeprowadzonych badań przedstawiono w tabeli 1.



Rysunek 6: Wyniki uzyskane podczas treningu sieci w zależności od ilości warstw ukrytych



Rysunek 7: Wyniki uzyskane podczas walidacji w zależności od ilości warstw ukrytych



Rysunek 8: Wyniki uzyskane podczas testowania sieci w zależności od ilości warstw ukrytych

Tabela 1: Zbiorcze zestawienie wyników

Liczba warstw ukrytych	Dokładność trenowania [%]	Dokładność walidacji [%]	Dokładność testowania [%]
4	94,02	93,10	92,34
5	94,69	93,84	93,41
6	95,73	94,96	94,80
7	96,81	96,05	95,92
8	97,03	96,83	96,77

5. Wnioski

Sztuczne sieci neuronowe są coraz częściej wykorzystywane przez naukowców i programistów ze względu na ich bardzo dobre wyniki. W powyższych badaniach sprawdzano jak zmienia się dokładność odpowiedzi sieci wraz ze zmianą jej parametrów- w tym przypadku zmianie podlegała liczba warstw ukrytych. Wyniki wykazały wzrost dokładności modelu wraz ze wzrostem liczby warstw ukrytych sieci konwolucyjnej. Nie zanotowano znaczących różnic pomiędzy kolejnymi wynikami. Już przy czterech warstwach ukrytych wyniki modelu przekroczyły 90 %, a najlepsze rezultaty zaobserwowano przy siedmiu i ośmiu warstwach (wyniki przekroczyły 95 %).

Literatura

- [1] R. Tadeusiewicz, M. Szaleniec, Leksykon sieci neuronowych. Wydawnictwo Fundacji "Projekt Nauka" (2015).
- [2] K. Różanowski, Sztuczna inteligencja rozwój, szanse i zagrożenia. Zeszyty Naukowe Warszawskiej Wyższej Szkoły Informatyki (2007).
- [3] R. Mithe, S. Indalkar, N. Divekar, Optical character recognition. International journal of recent technology and engineering (IJRTE), 2 (2013) 72-75.
- [4] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, (2015) 5325-5334.
- [5] Y. LeCun, Y. Bengio, G. Hinton, Deep learning. Nature, 521 (2015) 436-444.
- [6] M. Liang, H. Hu, Recurrent convolutional neural network for object recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, (2015) 3367-3375.
- [7] Y. Ma, Z. Xiang, Q. Du, W. Fan, Effects of user-provided photos on hotel review helpfulness: An analytical approach with deep learning. International Journal of Hospitality Management, 71 (2018) 120-131.
- [8] Y. Hou, H. Zhao, Handwritten digit recognition based on depth neural network. International Conference on Intelligent Informatics and Biomedical Sciences, Okinawa, 2017.

- [9] S. Albawi, T. A. Mohammed, S. Al-Zawi, Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET), (2017) 1-6 .
- [10] T. N. Sainath, A. R. Mohamed, B. Kingsbury, B. Ramabhadran, Deep convolutional neural networks for LVCSR. In 2013 IEEE international conference on acoustics, speech and signal processing, (2013) 8614-8618.
- [11] <http://www.fki.inf.unibe.ch/databases/iam-handwriting-database> [05.02.2019]

Influence the quality of the interface of online stores on their memorability

Wpływ jakości interfejsu sklepów internetowych na ich zapamiętywalność

Katarzyna Pytka*, Marek Miłosz

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

The article describes influence the quality of the interface of online stores on their memorability on selected examples. Nielsen heuristics, LUT's list and mamorability experiment were used in interface quality assesment.

Keywords: interface quality; web interface; memorability; memorability experiment

Streszczenie

W artykule opisano wpływ jakości interfejsu na zapamiętywalność stron sklepów internetowych na wybranych przykładach. W badaniach jakości interfejsu wykorzystano heurystyki Nielsena, listę LUT oraz przeprowadzono eksperyment zapamiętywalności.

Słowa kluczowe: jakość interfejsu; interfejs strony; eksperyment zapamiętywalności; zapamiętywalność

*Corresponding author

Email address: katarzyna.pytka@pollub.edu.pl, m.milosz@pollub.pl (K. Pytkal)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Wielka internetowa rewolucja, która miała miejsce około 20 lat temu wpłynęła w znacznym stopniu na nawyki i przyzwyczajania społeczeństwa [6]. W dzisiejszych czasach system robienia zakupów wygląda zupełnie inaczej niż odbywało się to w przeszłości (przed szerokim rozpowszechnieniem Internetu w Polsce). Według badań przeprowadzanych przez Gemius Polska internauci w Polsce coraz chętniej wyszukują informacji w sieci na temat poszukiwanych produktów zanim dokonają zakupu. Dla porównania na przestrzeni kilku ostatnich lat: w 2015 takie zachowania deklaruje 76,5% [1] badanych, w 2017 – 79% [2], a w 2018 już 83,6% [3].

Według najnowszego raportu Gemius Polska [4] dotyczącego e-commerce w Polsce coraz więcej konsumentów kierując się m.in. całodobową dostępnością produktów, wygodą (brakiem konieczności jechania do sklepu), dogodnymi metodami płatności oraz dostawą do domu lub paczkomatu, decyduje się na zakup produktów w sklepach internetowych. Odsetek takich osób z roku na rok rośnie [4]. W 2016 roku dokonanie zakupu w e-commerce deklarowało 50% badanych, w 2017 było to 54% [2], w 2018 – 56% [3], a w 2019 już 62% [4].

Warto podkreślić, że zgodnie z najnowszymi badaniami zdecydowana większość robiących zakupów przez Internet w grupie badanych należy do grupy wiekowej od 35 do 49 roku życia [4]. Gdzie w 2018 roku były to młodsze roczniki z przedziału wiekowego od 25 do 34 lat. W związku z taką tendencją zmian rok do roku można przypuszczać, że za kilkanaście i kilkadziesiąt lat robienie zakupów online stanie się ich naturalnym nawykiem dla każdej grupy wiekowej.

Z roku na rok liczba dostępnych dla konsumentów rozwiązań e-commerce rośnie. Aktualnie przez Internet można kupić wszystko od produktów spożywczych

po podnośniki samochodowe do warsztatów i inne ciężkie sprzęty. Zwiększa się wybór dla konsumentów, a z perspektywy przedsiębiorców rośnie konkurencja na rynku. Istotnym elementem jest to, aby oprócz pozyskiwania nowych klientów, utrzymać tych, którzy już dokonali w przeszłości zakupu na stronie i chętnie wrócą na platformę.

Oprócz takich czynników jak obsługa klienta po dokonaniu zakupu (czas realizacji zamówienia, kontakt ze sprzedającym, estetyka zapakowania produktu) jednym z elementów wpływających na to, czy zwiększona zostanie sprzedaż i konsument wróci w przyszłości do sklepu jest jakość interfejsu sklepów internetowych [5]. Przyjazny interfejs to m.in. taki, który dostosowany do urządzeń mobilnych, umożliwiając szybkie i wygodne poruszanie się po stronie, łatwo się go nauczyć, jest bezpieczny i niezawodny.

2. Metody analizy jakości interfejsu omówione na wybranych przykładach

Interfejs strony WWW stanowi jeden z najistotniejszych elementów jej budowy. Stanowi on warstwę pośredniczącą pomiędzy człowiekiem, a komputerem. Na rynku jest wiele dostępnych rozwiązań. To, które z nich wybierze użytkownik w dużej mierze zależy od tego, czy interfejs będzie dla niego wygodny, łatwy i szybki w obsłudze. Analiza i ocena pomaga w jego ulepszeniu [7].

2.1. Heurystyki Nielsena

Jedną z metod badania jakości interfejsu jest analiza heurystyczna, która sprawdza głównie problemy z jego użytecznością [8]. Wykrywa ona te elementy, które generują najwięcej błędów [9]. Heurystyki Nielsena to 10 zasad użyteczności opracowanych przez Jakoba Nielsena, wybitnego specjalistę w dziedzinie użyteczno-

ści [10]. W tym podrozdziale przedstawiony zostanie opis heurystyk dla wybranych sklepów internetowych.

Sklep <http://alefarby.pl/> [11]. Rezultaty analizy tego sklepu z wykorzystaniem heurystyk Nielsena są następujące:

1. Dzięki widocznej ścieżce kategorii i podkategorii produktu oraz ścieżce z poszczególnymi krokami w koszyku, użytkownik na bieżąco widzi, gdzie się znajduje na stronie oraz czy dodał coś do koszyka. Widoczny jest również status zalogowany/niezalogowany użytkownik.
2. Strona jest zrozumiała dla użytkownika. Produkty są posegregowane na kategorie i podkategorie.
3. Minusem badanego sklepu internetowego dotyczącym tej heurystyki jest brak możliwości zmiany adresu dostawy podczas ciągłego korzystania z jednej przeglądarki dla użytkownika, który nie jest zarejestrowany, ale podał już adres dostawy.
4. Wszystkie symbole, elementy i szablony są zachowane w obrębie strony. Produkty w sklepie są odpowiednio pogrupowane. Brakuje zdjęć przy niektórych produktach, co powoduje brak spójności z ogólnym szablonem na stronie produktu.
5. System spełnia tę zasadę w połowie. Przy wypełnianiu formularzy, takich jak rejestracja, logowanie wszystko działa poprawnie tzn. system waliduje dane i sprawdza, czy użytkownik podał je w odpowiedniej formie. Natomiast minusem jest to, że sklep nie pyta użytkownika ponownie o wpisane dane podczas procesu zakupu.
6. Ta zasada również nie jest zachowana w procesie zakupowym. Przy podsumowaniu transakcji brak informacji o produkcie, sposobie dostawy i pozostałych parametrach koszyka. Użytkownik musi wrócić do poprzednich kroków, aby sprawdzić, czy na pewno wszystko zostało podliczone poprawnie. Brak filtrów.
7. Brak sprzedaży komplementarnej i brak wyświetlenia produktów ostatnio oglądanych czy kupowanych.
8. Brak elementów zbędnych w danym momencie.
9. Komunikaty o błędach są sformułowane w jasny i zwięzły sposób. Każdy konsument powinien zrozumieć, co należy zrobić.
10. Brak zakładki pomoc. Brak informacji na stronie głównej, z kim lub w jaki sposób można się kontaktować w sprawie pytań czy potrzeby pomocy. Po skierowaniu kursora na zadaną ikonkę brak wyjaśnienia, co po kliknięciu w nią może się wydarzyć.

Sklep <http://megamajster.pl/> [12]. Rezultaty analizy tego sklepu z wykorzystaniem heurystyk Nielsena są następujące:

1. Interfejs zawiera ścieżki do produktu, kategorii i podkategorii oraz wyraźnie rozpisane kroki w koszyku zakupowym. Dzięki temu w wygodny sposób może sprawdzić gdzie aktualnie znajduje się na stronie i wrócić do poprzedniego kroku. Minusem jest brak widocznie oznaczonego koszyka z dotychcza-

sowymi zakupami podczas przeglądania strony oraz statusu zalogowania użytkownika na stronie.

2. Strona jest zrozumiała dla użytkownika. Produkty są posegregowane na kategorie i podkategorie.
3. Ta heurystyka jest spełniona w 100%. Oprócz obecności ścieżek przy produktach znajdują się również przyciski: Powrót, Dalej.
4. Wszystkie symbole, elementy i szablony są zachowane w obrębie strony. Produkty pogrupowane są w poprawny sposób. Minus stanowi brak spójnej kolorystyki. Przykład: Napis na zielonym tle to raz „Darmowa Dostawa;”, innym razem „Oferty Specjalne”
5. System nie jest zgodny z tą heurystyką. Podczas testów natrafiono na jeden formularz, który prawdopodobnie wysłano pusty i nie pojawiały się żadne komunikaty. Drugim przykładem jest błędny komunikat informujący, że można dokonać zakupu bez rejestracji wpisując dowolny login zamiast adresu e-mail. Jednak system w kolejnym kroku nie chce przepuścić dowolnego loginu niebędącego adresem e-mail.
6. Zasada nie jest zachowana na stronie. Elementy graficzne użyte przy produktach w postaci ikon takie jak: koszyk, waga, symbol na zielonym tle opisane są, jako legenda w prawym górnym rogu. Przy długiej liście produktów użytkownik musi się cofać na górę strony, aby rozszyfrować symbole. Na niektórych podstronach, na których znajdują się wspomniane wyżej elementy, brak legendy opisującej, co oznaczają ikony. Brak widocznych filtrów dla produktów.
7. Jest sekcja z produktami ostatnio oglądanymi oraz sprzedaż komplementarna. Po dodaniu do koszyka wiertarki system proponuje akcesoria.
8. Interfejs przytłacza dużą ilością informacji dookoła.
9. Komunikaty sformułowane są w sposób zrozumiały dla każdego użytkownika witryny. Podczas testów natrafiono na brak komunikatu o wysłaniu wiadomości przy jednym z formularzy – możliwość wysłania pustego formularza. Błędny komunikat informujący, że można dokonać zakupu bez rejestracji, natomiast wpisując dowolny login zamiast adresu email, system nie chce przepuścić dalej i nie udziela bezpośrednio informacji, co zrobić.
10. Klient ma dostęp do zakładki pomoc, gdzie może dowiedzieć się jak poruszać się po sklepie. Brak objaśnień po skierowaniu kursora na grafiki przy produktach.

Sklep <http://slet24.pl/> [13]. Rezultaty analizy tego sklepu z wykorzystaniem heurystyk Nielsena są następujące:

1. W systemie widoczna jest ścieżka kategorii i podkategorii do wszystkich produktów w sklepie. Podczas całego procesu zakupowego, przeglądania różnych produktów widoczna jest ilość produktów dodanych do koszyka, łączna kwota produktów w koszyku, liczba produktów obserwowanych i liczba produktów dodanych do porównania. Widoczny jest rów-

nież wyraźnie status czy użytkownik jest już zalogowany do systemu czy nie.

2. Strona jest zrozumiała dla użytkownika. Produkty są posegregowane na kategorie i podkategorie.
3. Użytkownik ma w pełni możliwość powrotu do poprzedniej strony i kategorii, dzięki ścieżce produktów. Wszystkie kroki w koszyku zakupowym są również w pełni widoczne w obrębie jednej strony dla wszystkich użytkowników.
4. Wszystkie symbole, elementy i szablony są zachowane w obrębie strony. Produkty pogrupowane są w poprawny sposób.
5. Strona wspiera walidację danych we wszystkich formularzach dostępnych na stronie.
6. Strona wyświetla aktywne filtry podczas przeglądania produktów oraz na koniec procesu zakupowego prezentuje pełne podsumowanie transakcji.
7. Brak zakładki z ostatnio oglądanymi produktami.
8. Interfejs strony jest bardzo estetyczny, nie atakuje użytkownika zbędnymi informacjami podczas przeglądania.
9. Komunikaty o błędach są sformułowane w jasny i zwięzły sposób. Natrafiono na dwa przypadki komunikatów w języku angielskim.
10. Każdy użytkownik ma dostęp do zakładki przewodnik kupującego po stronie slet24.pl. Wszystkie najważniejsze informacje są w niej bardzo spójnie przedstawione. Brak w systemie ikon i grafik, które byłyby niejasne lub niezrozumiałe dla użytkownika.

Sklep <http://swiatnarzedzi.pl/> [14]. Rezultaty analizy tego sklepu z wykorzystaniem heurystyk Nielsena są następujące:

1. Na stronie wyraźnie oznaczono ścieżkę kategorii i podkategorii prowadzących do produktów. Zawiera również wyraźnie rozpisane kroki w koszyku zakupowym. Dostęp do koszyka, konta wyszukiwarki i menu głównego jest stale dostępny dla użytkownika. Widoczny status zalogowania użytkownika.
2. Strona jest zrozumiała dla użytkownika. Produkty są posegregowane na kategorie i podkategorie.
3. Użytkownik ma w pełni możliwość powrotu do poprzedniej strony i kategorii, dzięki ścieżce produktów. Minusem jest to, że nie wszystkie kroki w koszyku zakupowym są w pełni widoczne dla wszystkich użytkowników.
4. Wszystkie symbole, elementy i szablony są zachowane w obrębie strony. Produkty pogrupowane są w poprawny sposób.
5. Strona wspiera walidację danych we wszystkich formularzach dostępnych na stronie.
6. Strona wyświetla aktywne filtry podczas przeglądania produktów oraz na koniec procesu zakupowego prezentuje pełne podsumowanie transakcji.

7. Na stronie znajduje się zakładka z ostatnio oglądanymi produktami w sklepie. Również w koszyku widoczne są produkty w sprzedaży komplementarnej.
8. Interfejs strony jest bardzo estetyczny, nie atakuje użytkownika zbędnymi informacjami podczas przeglądania.
9. Komunikaty o błędach są sformułowane w jasny i zwięzły sposób. Każdy konsument powinien zrozumieć, co należy zrobić.
10. Strona ma wbudowaną zakładkę informacje FAQ, która pokazuje podstawowe procesy w sklepie. Jednak nie mówi bezpośrednio o budowie strony lub gdzie można znaleźć podstawowe informacje. Brak w systemie ikon i grafik, które byłyby niejasne lub nieadekwatne dla danej funkcjonalności.

2.2. Lista LUT

W celu uzyskania bardziej obiektywnych wyników w ocenie użyteczności interfejsu stron internetowych warto wykorzystać listy kontrolne. Są to kwestionariusze ocen zawierające kilka sekcji z pytaniami. Na pytania odpowiada się zazwyczaj Tak/Nie lub w skali ocen od 1 do 5.

Jednym z rodzajów listy kontrolnej jest [11] lista „LUT” opracowana na podstawie Heurystyk Nielsena (tabela 1). W tym przypadku do odpowiedzi na pytania używa się skali liczbowej.

2.3. Punkty WUP

Po wykonanej kontroli przy pomocy listy można obliczyć metrykę jakości interfejsu, czyli tzw. Punkty WUP (ang. *Web Usability Points*) [15]. Im wyższa wartość wskaźnika WUP tym lepsza jakość interfejsu. Punkty WUP wyznaczone dla badanych stron przedstawiono w tabeli 2.

2.4. Punkty WUP

Po wykonanej kontroli przy pomocy listy można obliczyć metrykę jakości interfejsu, czyli tzw. Punkty WUP (ang. *Web Usability Points*) [15]. Im wyższa wartość wskaźnika WUP tym lepsza jakość interfejsu. Punkty WUP wyznaczone dla badanych stron przedstawiono w tabeli 2.

Tabela 1: Rezultaty oceny autora artykułu z wykorzystaniem listy LUT

Obszar	Podobszar	Pytanie	http://alefar by.pl/	https://www.megamajster.pl/	https://www.slet24.pl/	https://swiatnarzedzi.pl/
Nawigacja i struktura	Łatwość nawigowania	Czy dostęp do wszystkich sekcji aplikacji jest łatwy i intuicyjny?	3	4	5	5
		Czy dostęp do wszystkich funkcji aplikacji jest łatwy i intuicyjny?	3	3	5	5
	Hierarchia informacji	Czy hierarchia informacji nie jest zbyt głęboka?	5	4	5	5
	Struktura informacji	Czy struktura informacji jest przemyślana?	5	4	5	5
		Czy struktura informacji jest spójna?	5	5	5	5
		Czy struktura informacji jest zrozumiała dla użytkownika?	5	4	5	5
	Elementy ekranu	Czy wspierają proces nawigacji?	4	5	4	5
Komunikaty, feedback, pomoc dla użytkownika	Komunikaty (ogólne)	Czy dostarczają wystarczająco dużo informacji zwrotnych dotyczących stanu operacji wykonywanych przez użytkownika?	3	5	5	5
	Komunikaty o błędach	Czy zawierają podpowiedzi dotyczące rozwiązania problemu?	3	4	3	3
	Informacje zwrotne i pomoc	Czy pojawiają się w miejscach, gdzie mogą być potrzebne?	2	5	4	5
		Czy treść pomocy jest dostępna dla przeciętnego użytkownika?	4	5	5	4
		Czy treść pomocy jest zrozumiała dla przeciętnego użytkownika?	5	4	5	4
		Czy prezentowane podpowiedzi bądź rozwiązania problemów są możliwe do wykonania przez zwykłego użytkownika?	5	5	4	5
	Interfejs aplikacji	Layout	Czy layout jest czytelny?	2	3	5
Czy layout jest dostosowany do różnych rozdzielczości?			1	1	5	5
Czy layout jest dostosowany do urządzeń mobilnych?			1	1	5	4
Czy układ graficzny jest spój-			4	3	5	5

		ny?				
		Czy layout wspiera realizację zadań?	5	5	5	5
	Dobór barw	Czy kontrast pomiędzy tekstem a tłem jest odpowiedni?	3	5	3	3
		Czy dobór barw umożliwia skorzystanie z aplikacji przez osoby z zaburzeniami widzenia barw?	4	4	5	5
		Czy dobór barw umożliwia skorzystanie z aplikacji przy wykorzystaniu różnego rodzaju wyświetlaczy?	1	4	5	5
Treści podstron	Teksty	Czy są zrozumiałe dla użytkownika?	4	5	5	5
	Nazewnictwo	Czy używane w aplikacji nazewnictwo jest spójne?	5	5	5	5
		Czy używane w aplikacji nazewnictwo jest zrozumiałe?	4	5	5	5
	Etykiety	Czy używane w interfejsie etykiety dostarczają wystarczająco dużo informacji?	5	4	5	5
		Czy elementy interfejsu posiadają niezbędne etykiety?	5	4	5	5
Wprowadzenie danych	Formularze	Czy posiadają czytelny projekt?	5	3	5	5
		Czy umożliwiają wprowadzenie niezbędnych danych?	5	5	5	5
		Czy są dostosowane do urządzeń mobilnych?	1	1	5	5
	Dane	Czy zwykły użytkownik nie ma trudności z wprowadzeniem danych do formularza?	3	5	5	5
		Czy formularze posiadają elementy podpowiedzi dotyczące wprowadzanych danych (m.in. formatu, zakresu)?	1	3	5	5
		Czy formularze posiadają elementy walidujące wprowadzone dane?	4	5	4	5

Tabela 2: Wartości punktów WUP dla analizowanych stron

Strona	Średnie dla obszarów					WUP
	Nawigacja i struktura	Komunikaty, feedback	Interfejs aplikacji	Treści podstron	Wprowadzanie danych	
http://alefarby.pl/	4,25	3,33	2,63	4,50	3,17	3,58
https://www.megamajster.pl/	4,21	4,58	3,47	4,67	3,67	4,12
https://www.slet24.pl/	4,75	4,17	4,67	5,00	4,83	4,63
https://swiatnarzedzi.pl/	5,00	4,17	4,57	5,00	5,00	4,75

3. Eksperyment zapamiętywalności i jego wyniki

Zapamiętywalność to zjawisko zwiększenia efektywności pracy użytkownika z aplikacją czy stroną internetową. Przeprowadzenie eksperymentu zapamiętywalności pozwala zmierzyć to zjawisko, czyli określić jak szybko użytkownik jest w stanie nauczyć się efektywnie korzystać z rozwiązania.

Zapamiętywalność można przedstawić przy pomocy wskaźnika poziomu zapamiętywalności ML (ang. *Memorability Level*) [15].

Zanim zostanie przeprowadzony eksperyment zapamiętywalności istotnym elementem jest zaplanowanie etapów badania i zadań dla grupy badawczej liczącej 9 osób. Każdy z uczestników badania miał za zadanie znaleźć na wybranej stronie odpowiedzi na 4 pytania. Informacje do znalezienia przez grupę badawczą na poszczególnych stronach:

<http://alefarby.pl/> [11]

1. Ile zapłacisz za dostawę w przypadku przesyłki za pobraniem powyżej 1 tony?
2. W jakich miastach są umiejscowione sklepy stacjonarne?
3. Dokończ zdanie: Najtańsze pędzle w sklepie to ...
4. W jakiej cenie możesz kupić farbę białą do betonu o pojemności 10l, firmy Aksil.

<https://www.megamajster.pl/> [12]

1. Z ilu kroków składa się zamówienie? Czy trzeba mieć założone konto
2. Porównaj parametry wkrętarek firmy Makita: AR410HR i AR411HR.
3. Ile rodzajów generatorów i agregatów prądotwórczych jest dostępnych w sklepie?
4. Co oznaczają litery T, P, M przy konkretnych produktach?

<https://www.slet24.pl/> [13]

1. Czy można odebrać zamówienie osobiście?
2. W jakiej cenie można kupić wiertło do wiercenia na sucho, diamentowe, firmy RUBI, 10mm?
3. Czy jest możliwość dokonania zakupu bez rejestracji?
4. Ile maksymalnie produktów możesz porównać przy użyciu narzędzia do porównywania produktów w tym sklepie?

<https://swiatnarzedzi.pl/> [14]

1. Ile produktów dostępnych jest w kategorii narzędzia dekarskie?
2. Czy sklep daje możliwość zakupu inwestycyjnego dla firm w ramach leasingu?
3. Jaki jest okres gwarancji dla wiertarki udarowej, firmy STANLEY 750W z walizką FMEH750k-QS?
4. Czy sklep posiada certyfikat e-bezpiecznego zakupu?

Przy każdym zadaniu prowadząca badanie mierzy czas wykonania zadania przy pomocy stopera i zapisuje go zaokrąglając do pełnych sekund.

Po przeprowadzonych badaniach został obliczony wskaźnik poziomu zapamiętywalności ML oraz średni czas odpowiedzi na zadane pytanie (tabela 3).

Tabela 3: Porównanie średniego czasu znalezienia odpowiedzi na zadania z wartością WUP

Strona	Średni czas znalezienia odpowiedzi na pytanie [s]	Punkty WUP
http://alefarby.pl/	37,7	3,58
https://www.megamajster.pl/	35,22	4,12
https://www.slet24.pl/	31,45	4,68
https://swiatnarzedzi.pl/	30,28	4,75

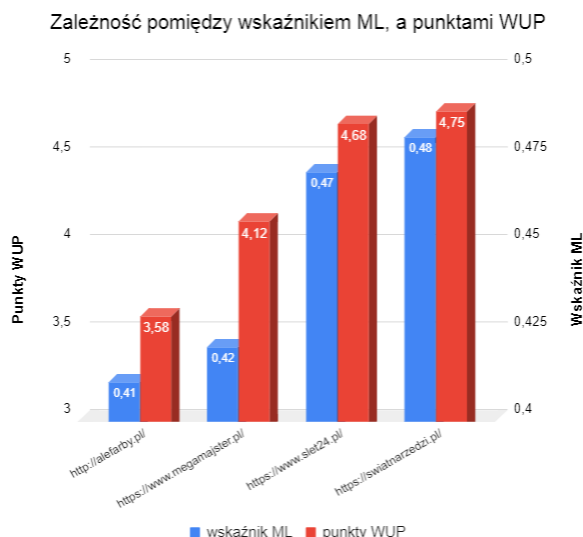
Tabela 4: Porównanie wskaźnika ML i punktów WUP

Strona	Wskaźnik ML	Punkty WUP
http://alefarby.pl/	0,41	3,58
https://www.megamajster.pl/	0,42	4,12
https://www.slet24.pl/	0,47	4,68
https://swiatnarzedzi.pl/	0,48	4,75

4. Wnioski

Porównując wskaźniki ML oraz punkty WUP (tabela 4) (rysunek 1) zauważalna jest zależność. Im wyższa jakość interfejsu tym wyższa wartość wskaźnika ML. Intuicyjny, dobrze zorganizowany i przejrzysty interfejs powoduje zwiększenie zapamiętywalności na stronach sklepów internetowych.

Opisowa analiza 10 heurystyk Nielsena również pokazuje, że większą liczbę błędów w użyteczności i strukturze strony internetowej wykazują sklepy z niższą wartością wskaźnika ML.



Rysunek 1: Porównanie wskaźnika ML i punktów WUP

Ponadto podczas analizowania wyników eksperymentu zapamiętywalności zauważalna jest również zależność pomiędzy czasem znalezienia odpowiedzi na zadane pytania, a jakością interfejsu (tabela 3). Im niższa jakość interfejsu tym użytkownikom więcej czasu zajmuje znalezienie odpowiedzi na nurtujące ich pytania.

Literatura

- [1] Raport Gemius Polska, E-commerce w Polsce 2015, <https://www.gemius.pl/files/reports/E-commerce-w-Polsce-2015.pdf> [25.09.2019]
- [2] Raport Gemius Polska, E-commerce w Polsce 2017, <https://www.gemius.pl/e-commerce-aktualnosci/najnowsze-dane-o-polskim-e-commerce-juz-dostepne.html> [25.09.2019]
- [3] Raport Gemius Polska, E-commerce w Polsce 2018 z dnia 28.06.2018, <https://www.gemius.pl/wszystkie-artykuly-aktualnosci/internauci-zaufali-e-sklepom-raport-e-commerce-w-polsce-2018-juz-dostepny.html> [25.09.2019]
- [4] Raport Gemius Polska, E-commerce w Polsce 2019 z dnia 28.06.2019, <https://www.gemius.pl/wszystkie-artykuly-aktualnosci/raport-e-commerce.html> [25.09.2019]
- [5] Strona pomocy home.pl <https://pomoc.home.pl/baza-wiedzy/jak-zaprojektowac-przejrzysty-interfejs-sklepu-ktory-bedzie-wspomagal-sprzedaz> [25.09.2019]
- [6] Raport Gemius Polska, Pokolenia Polaków w sieci, <https://www.gemius.pl/e-commerce-aktualnosci/id-20-40-60-co-wiemy-o-pokoleniach-polakow-w-sieci.html> [25.09.2019]
- [7] Projektowanie interfejsu człowiek-komputer, Politechnika Gdańska, <https://eti.pg.edu.pl/documents/176468/27264685/ocena.pdf> [29.09.2019]
- [8] M. Laskowski, Propozycje metodyk badania użyteczności interfejsu aplikacji https://www.researchgate.net/profile/Maciej_Laskowski/publication/236618668_Propozycje_metodyk_badiana_uzytecznosci_interfejsow_aplikacji/links/549ddb300cf2b803713a7d53/Propozycje-metodyk-badiana-uzytecznosci-interfejsow-aplikacji.pdf [30.09.2019]
- [9] M. De Jong, T. Van Der Geest, Characterizing Web Heuristics <https://pdfs.semanticscholar.org/1a3a/889448bc9ca93a75fc5e146bb954baca0953.pdf> [30.09.2019]
- [10] J. Nielsen, 10 Usability Heuristics for User Interface Design <https://www.nngroup.com/articles/ten-usability-heuristics/> [29.09.2019]
- [11] Sklep Alefarby, <http://alefarby.pl/> [29.09.2019]
- [12] Sklep Megamajster, <https://www.megamajster.pl/> [29.09.2019]
- [13] Sklep Slet24, <https://www.slet24.pl/> [29.09.2019]
- [14] Sklep Świat narzędzi, <https://swiatnarzedzi.pl/> [29.09.2019]
- [15] M. Miłosz, M. Borys, M. Laskowski, Memorability Experiment vs. Expert Method in Websites Usability Evaluation. ICEIS 2013 – Proceedings of the 15th International Conference on Enterprise Information Systems, Angers, France, vol. 3, July 2013, 4-7.

Comparison of the NFC authentication method with the methods available in Android

Porównanie metody uwierzytelniania NFC z metodami dostępnymi w Android

Dmytro Malenko*

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

Most multi-user applications need an access system so that users can log in from different devices. To identify a user, at least one authentication method must be used. The article presents a comparison of the authentication method using NFC technology with other methods that are available for Android devices and differ in both implementation and specific usage.

Keywords: android; nfc; safety

Streszczenie

Większość aplikacji przeznaczonych dla wielu użytkowników powinno korzystać z systemu autoryzacji, aby użytkownicy mogli logować się z różnych urządzeń. Aby zidentyfikować użytkownika, należy użyć co najmniej jednej metody uwierzytelniania. W artykule przedstawiono porównanie metody uwierzytelniania wykorzystującej technologię NFC z innymi metodami dostępnymi dla urządzeń z Androidem, różniącymi się zarówno implementacją, jak i określonym zastosowaniem.

Słowa kluczowe: android; nfc; bezpieczeństwo

*Corresponding author

Email address: dimon95me@gmail.com (D. Malenko)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Technologia NFC (ang. Near Field Communication) została opublikowana w 2004 r. [1]. Od tego czasu stopniowo zdobywała popularność. W tej chwili prawie każdy telefon jest wykonany z wykorzystaniem tej technologii. Jednak technologia wciąż zyskuje na popularności. Jest stosowana w systemach płatności jako środek rozpowszechniania informacji (tag NFC), jako klucz dostępu. Wykorzystanie tej technologii jest ograniczone tylko wyobraźnią. Można śmiało powiedzieć, że technologia ta nie osiągnęła jeszcze swojego szczytu technicznego i stale się rozwija. Urządzenia do noszenia z obsługą tej technologii są już produkowane.

Jedną z ważnych cech jest niewielki rozmiar części technologicznej i niskie zużycie energii. W przypadku z tagami NFC zapewnienie zasilania jest konieczne tylko po stronie odbiorczej. Obecnie rozmiary części technologicznej pozwalają osadzić tę technologię niemal wszędzie. Od elektronicznych zegarków i bransoletek fitness po tkanki miękkie w ludzkim ciele [2].

W tej chwili dla programistów nie ma natychmiastowych i wygodnych rozwiązań. Poszukiwanie bibliotek dla najwygodniejszego zarządzania NFC za pomocą kodu wykazało, że obecnie najlepszym rozwiązaniem jest standardowa biblioteka NfcHelper. Jednocześnie, szukając oficjalnej dokumentacji, można dowiedzieć się, jak odczytywać i zapisywać informacje na poziomie urządzenia [3].

Z czasem potrzebne są nowe metody uwierzytelniania. Zbadanie nowej metody autoryzacji użytkownika za

pomocą taga NFC pomoże dostrzec zalety i wady tego podejścia. Pomoże również zrozumieć, które warunki są najbardziej odpowiednie dla szybkiego, bezpiecznego i poprawnego odczytu danych z taga.

2. Analiza literatury

Peter Konrad Tysowski w artykule „Near-field communication (NFC) system providing NFC tag geographic position authentication and related methods” przedstawił alternatywną metodę autoryzacji za pomocą tagów zawierających geodane [4]. System sugeruje obecność telefonu, który może pobierać informacje o lokalizacji nie tylko z GPS. Podejście to dotyczy dużych i małych miast o niewystarczającym zasięgu sieci.

Artykuł „Wearable Authentication Device” Jamesa A Van Boscha i Pavla A. Shostaka przedstawia koncepcję noszonego urządzenia, które ma ulepszone połączenie z czytnikiem NFC [5]. Jego zwartość pozwala na dopasowanie projektu do wymiarów pierścienia. Zasada działania polega na lokalizacji dwóch anten zamiast jednej. Dzięki temu możliwe jest szybkie nawiązanie kontaktu zamiast wybierania domniemanej lokalizacji anteny.

Artykuł napisany przez Jurisa Klonovsa, Christoffera Kjeldgaarda Petersena, Henninga Olesena, Allana Hammershoj z tytułem „ID Proof on the Go: Development of a Mobile EEG-Based Biometric Authentication System” mówi o zastosowaniu elektroencefalografu i tagów NFC [6]. Główny nacisk położony jest na podpis elektroencefalograficzny. System sugeruje biome-

tryczne rozpoznanie osoby na podstawie aktywności jego mózgu. Zgodnie z artykułem aktywność mózgu każdej osoby jest wyjątkowa. Aby uniknąć przypadkowej autoryzacji klienta, postanowili dodać technologię NFC i technologię rozpoznawania twarzy do schematu uwierzytelniania. System jest w początkowej fazie rozwoju i został zaprojektowany z myślą o postępie technologicznym, kiedy części systemu można będzie wykonać z bardziej kompaktowych urządzeń.

Ciekawe podejście zostało zastosowane w artykule Muhammada Qasima Saeeda i Colina D. Waltera „Offline NFC Tag Authentication”. Najważniejsze jest użycie tagów NFC, gdy nie masz połączenia z Internetem. Integralność danych w tagach zapewniają podpisy cyfrowe. Konwencjonalne protokoły nie zapewniają ochrony przed użyciem kopii takich tagów. W tym artykule zaproponowano opcję zapobiegania stosowaniu kopii tagów. Rozwiązanie oparte jest na protokole protokół żądanie-odpowiedź z wykorzystaniem kryptografii klucza publicznego i PKI [7]. Proponowana opcja różni się bardzo od oryginalnego podejścia. Te podejście się odnosi do uwierzytelniania tagiem NDEF.

Artykuł „Mobile wireless communications device providing near field communication (NFC) unlock and tag data change features and related methods” mówi o odblokowaniu urządzenia za pomocą znacznika NFC. Ta metoda odblokowania urządzenia obejmuje dwie opcje odczytu. Pierwsza opcja to prosty odczyt i odblokowanie. Druga opcja polega na odblokowaniu urządzenia poprzez przepisanie klucza zgodnie z harmonogramem [8].

Artykuł „Elliptic curve-based RFID/NFC authentication with temperature sensor input for relay attacks” omawia sposoby uniknięcia przechwytywania informacji przez zewnętrznych słuchaczy. Rozważane są również wady tych metod. Główną ideą tego artykułu jest zmiana temperatury powierzchni tagu, co wpływa na sygnał i upośledza zdolność do przechwytywania sygnału. Z tego powodu czas transmisji danych na odległość zmienia się, a sygnał jest zniekształcony [9].

Artykuł „Two-factor user authentication using near field communication” autorstwa Philipa Hewinsona opisuje dwukierunkową weryfikację tożsamości. Unikalny klucz jest przechowywany w pamięci telefonu. Przy pierwszym podłączeniu taga do czytnika uwierzytelnienie zakończy się sukcesem, do tagu zapisywany jest oddzielny klucz, który identyfikuje tag. Przy następnym połączeniu z tagiem porównywany jest klucz identyfikujący telefon i klucz identyfikujący taga. Posiadanie dwóch kluczy do sprawdzania zwiększa bezpieczeństwo odblokowywania. Niezawodność staje się jeszcze większa dzięki dynamicznej zmianie klucza taga [11].

Standardowym protokołem przesyłania danych do znacznika NFC jest NDEF. Artykuł „Secure and Lightweight Authentication Protocol for NFC Tag Based Services” opisuje możliwości uszkodzenia danych na tagu z podanym protokołem i ich konsekwencje. Jako rozwiązanie zaproponowano dodatkowe protokoły, które są bardziej korzystne, gdy stosuje się niedrogie

tagi z mniejszą pamięcią. Dodatkowymi zaletami dostarczonych protokołów są bezpieczniejsze rejestrowanie i przechowywanie danych, które zapobiegają fałszowaniu, DoS, modyfikacji danych i atakowi typu phishing [12].

Interesującą implementację anteny NFC do telefonów zaproponowano w artykule „NFC Antenna Design for Low-Permeability Ferromagnetic Material”, napisanym przez Byungje Lee, Byeongkwan Kim, Frances J. Harackiewicz, Byeonggi Mun, Hyunwoo Lee. W artykule opisano nową konstrukcję anteny opartą na zmianie struktury wewnętrznej pętli i uzwojenia. Zastosowanym materiałem jest kompozyt ferrytowo-polimerowy, który jest lepszy i bardziej skuteczny niż drogie elementy ferrytowe. Proponowana antena ma większy o 23% zasięg odczytu, i o 65% lepszą modulację obciążenia w porównaniu z rozwiązaniami obecnymi odpowiednikami na rynku [13].

Artykuł „Automated antenna impedance adjustment for Near Field Communication (NFC)” proponuje nowe podejście do odczytu różnych typów urządzeń. W telefonie moduł NFC jest ułożony inaczej niż na karcie bankowej lub w tagu NFC. Aby zapewnić pełną zgodność, urządzenie, które odczytuje, musi zawierać kilka rodzajów anten. Jako rozwiązanie zaproponowano jedną antenę z automatyczną regulacją anteny w oparciu o cyfrowe strojenie kondensatorów. W takim przypadku antena dostosowuje się do urządzenia nadawczego, skutecznie go identyfikuje i wymienia informacje [14].

3. Technologie badawcze

W artykule zbadano kilka różnych metod autoryzacji. Każda metoda ma swoje zalety i wady. Każda metoda została wybrana z uwzględnieniem bezpieczeństwa, wygody, dostępności rozwoju, wszechstronności i popularności wykorzystania.

Początkowo proponowane metody autoryzacji:

1. Biometryczne rozpoznawanie twarzy
2. Rozpoznawanie odcisków palców
3. Klasyczne hasło
4. Maskowane hasło
5. Odblokowanie za pomocą kodu QR
6. Odblokowanie za pomocą technologii NFC
7. Graficzny wzór

Wdrożenie autoryzacji przy użyciu biometrycznego rozpoznawania twarzy nie jest obecnie dostępne przy użyciu gotowych bibliotek do implementacji w aplikacjach innych firm. Rozwiązaniem do wprowadzenia metody autoryzacji wykorzystującej biometryczne rozpoznawanie twarzy jest jedynie samodzielne opracowanie i dostosowanie tej metody.

Rozpoznawanie odcisków palców jest obecnie bardzo popularne i niedrogi pod względem rozwoju. Niestety urządzenie wykorzystane w próbach przedstawionych poniżej nie posiadało czytnika odcisków palców dlatego nie uwzględniono tej metody odblokowywania w testach praktycznych.

Klasyczne hasło jest bardzo popularne i jest używane na prawie wszystkich systemach. Ta metoda autory-

zacji zostanie wykorzystana w badaniu i porównana z innymi metodami autoryzacji.

Maskowane hasło składa się z kilku pól, w których wpisany jest jeden znak. Każde pole odpowiada za określony znak hasła. Niektóre pola są nieaktywne i odpowiednio ponumerowane. Numeracja służy jako wskazówka, aby nie pomylić się podczas wprowadzania poszczególnych znaków hasła. Ta metoda autoryzacji nie jest najpopularniejsza i najwygodniejsza, a jednocześnie zapewnia wysoki poziom bezpieczeństwa i jest stosowana głównie w aplikacjach bankowych. Ta metoda jest bardzo dobra do badań ze względu na łatwość rozwoju i wysoki poziom bezpieczeństwa.

Odblokowanie za pomocą kodu QR wykorzystuje moduł kamery. Kamera jest skierowana na zakodowany obraz, a informacje zakodowane na obrazie są konwertowane przez urządzenie na dane cyfrowe. Ta metoda charakteryzuje się niskim poziomem bezpieczeństwa, ponieważ zakodowany obraz można sfotografować, a następnie wydrukować lub zapisać w pamięci innego urządzenia. Ta metoda nie nadaje się do autoryzacji i jest lepsza do udostępniania informacji w otwartej formie. Ta metoda nie nadaje się do badań ze względu na niskie bezpieczeństwo.

Aby odblokować za pomocą technologii NFC, należy użyć urządzenia, które odczytuje informacje oraz urządzenia, które przechowuje informacje. Urządzeniem przechowującym informacje może być telefon lub znacznik NFC. Zaletą tagów NFC jest to, że nie trzeba zużywać energii elektrycznej do interakcji z urządzeniami odczytującymi informacje. Metoda autoryzacji z wykorzystaniem technologii NFC jest świetna do badań, ponieważ hasło jest przechowywane w pamięci przenośnej i jest obowiązkowe, aby obsługiwać technologię NFC za pomocą urządzenia odczytującego informacje, co zapewnia dodatkowy poziom bezpieczeństwa.

Odblokowanie za pomocą wzoru jest zbiorem macierzy punktów. Przesuwanie palcem od punktu do punktu tworzy wzór, który tłumaczy się na obiekt. Cechy metody odblokowania z wzorem to szybkie wprowadzanie hasła i łatwość użycia. Podczas korzystania ze złożonej kombinacji hasło jest trudne do podejrzenia za pierwszym razem. Ponieważ metoda odblokowywania za pomocą hasła graficznego ma średni poziom bezpieczeństwa, wysoki poziom wygody i jest popularna wśród nowoczesnych metod odblokowywania, będzie stosowana w badaniach porównaniu z innymi wybranymi metodami autoryzacji. W rezultacie wybrano następujące metody autoryzacji:

1. Klasyczne hasło
2. Maskowane hasło
3. Autoryzacja za pomocą technologii NFC
4. Logowanie za pomocą wzoru

4. Środowisko testowe

Do przeprowadzenia testów używany jest telefon Motorola Droid Turbo XT1254 z systemem operacyjnym Android 6.0.1. Ta wersja dobrze nadaje się do testowania, ponieważ obsługuje aktualizację 28 wersji biblioteki SupportLibrary. Biblioteka SupportLibrary zawiera

funkcje kolejnych systemów operacyjnych Android, częściowo zastępując potrzebę aktualizacji systemu operacyjnego urządzenia [15]. Zaletą tego modelu telefonu jest obsługa technologii NFC.

Jako przetestowane oprogramowanie stworzono aplikację mobilną dla systemu Android, począwszy od Android API 10. Aplikacja imituje działanie aplikacji bankowej, skupiając się na bezpieczeństwie i użyteczności oraz zawiera wszystkie zaimplementowane wybrane metody autoryzacji.

5. Procedura testowa

W testowaniu metod autoryzacji wzięło udział 9 osób. Podczas testowania dla każdej metody autoryzacji było zastosowano osobne hasło. Każdy użytkownik logował się co najmniej 10 razy w każdej z metod autoryzacji.

Każda próba autoryzacji obejmuje wejście do aplikacji i podanie odpowiedniego loginu. Po wprowadzeniu loginu i naciśnięciu przycisku potwierdzenia aplikacja komunikuje się z serwerem, skąd otrzymuje informacje o metodzie dołączonej do tego loginu. Jeśli w bazie danych nie ma takiego loginu, wyświetlane jest powiadomienie o braku użytkownika o tej nazwie. Następnie dane są wprowadzane zgodnie z wyświetlaną metodą, a login i hasło są sprawdzane na serwerze. Jeśli test się powiedzie, nastąpi przejście do nowego okna użytkownika. Jeśli test się nie powiedzie, zostanie wyświetlone powiadomienie z informacją, że weryfikacja nie została zakończona przy użyciu nazwy użytkownika i hasła.

Wygoda każdej metody autoryzacji została oceniona przez użytkowników na podstawie kryteriów szybkości autoryzacji, łatwości pracy z metodą autoryzacji i liczby udanych prób autoryzacji.

6. Analiza prezentowanych metod uwierzytelnienia

Analiza przedstawionych metod autoryzacji została przetestowana przez kilku użytkowników na tym samym urządzeniu. Przy każdej autoryzacji śledzono czas. Odwołanie do zapisu czasu rozpoczęło się natychmiast po przejściu do metody autoryzacji użytkownika docelowego. Koniec zliczania był wskazywany przez pomyślne zwrócenie odpowiedzi serwera do użytkownika. W wyniku praktycznych testów metod autoryzacji uzyskane wartości zostały wyświetlone w tabeli 1 i tabeli 2.

Tabela 1: Dane analityczne

	Minimalny czas auto- ryzacji [ms]	Średni czas auto- ryzacji [ms]	Maksymalny czas autory- zacji [ms]
Klasyczne hasło	5456	6804	8785
Autoryzacja przy użyciu technologii NFC	2017	3511	5575
Autoryzacja	1372	2746	3891

przy użyciu graficznego klucza			
Maskowane hasło	2746	10065	19715

Tabela 2: Dane statystyczne

	Liczba błędów podczas wprowadzania hasła [%]	Łączna liczba prób logowania	Ocena użytkowników
Klasyczne hasło	9	35	6/10
Autoryzacja przy użyciu technologii NFC	0	51	9/10
Autoryzacja przy użyciu graficznego klucza	7,6	64	8/10
Maskowane hasło	31,2	55	5/10

Warto zauważyć, że przy planowaniu autoryzacji nie zauważono skrócenia czasu autoryzacji. Wynika to z faktu, że użytkownik musi powtórzyć autoryzację setki razy, aby zmniejszyć szybkość korzystania z tej metody.

Klasyczne hasło w szybkości autoryzacji jest gorsze niż wszystkie metody autoryzacji, z wyjątkiem złożonego hasła. Czas potrzebny do autoryzacji wynosi od 5,5 do 8,5 sekundy. Użytkownicy logowali się średnio przez prawie 7 sekund. Jednocześnie trudno jest podejrzeć hasło, ponieważ nowowprowadzone znaki są wyświetlane przy wprowadzaniu tylko przez krótki czas. Ponadto znaki są natychmiast ukryte po wprowadzeniu nowego znaku. Z punktu widzenia wygody pozostaje zwykłą i najpopularniejszą metodą autoryzacji, ponieważ można ją łatwo wdrożyć w większości projektów. Wsparcie i rozwój nie wymagają dużego wysiłku ze strony programistów. Tej metody autoryzacji można łatwo wdrożyć w większości środowisk programistycznych. Błędy podczas wprowadzania hasła wynikały z natychmiastowego ukrycia poprzedniego znaku. Z drugiej strony takie podejście zwiększa bezpieczeństwo podczas korzystania z klasycznej metody wprowadzania hasła.

Korzystanie z metody autoryzacji z wykorzystaniem technologii NFC ma dodatkowe wymagania. Aby ta metoda działała, urządzenie musi obsługiwać technologię NFC. Przechowywanie klucza w pamięci zewnętrznej pozwala wydajniej i szybciej używać złożonych haseł, których nie trzeba zapamiętywać. Badanie wyka-

zało, że metoda autoryzacji wykorzystująca technologię NFC zajmuje drugie miejsce pod względem szybkości autoryzacji. Czas autoryzacji przy użyciu technologii NFC trwał średnio 3,5 sekundy. A maksymalna i minimalna autoryzacja zajęła odpowiednio 5,5 i 2 sekundy. Ponadto ta metoda okazała się najbardziej skuteczna w obliczaniu liczby nieważnych autoryzacji. Ponieważ hasło znajduje się na oddzielnym urządzeniu, użytkownik musi tylko podnieść telefon do urządzenia, na którym hasło jest przechowywane. Użytkownicy docenili wygodę autoryzacji przy użyciu technologii NFC.

Autoryzacja za pomocą wzoru wzbudziła największe zainteresowanie użytkowników, ponieważ jest ona najbardziej znana dzięki możliwości odblokowania ekranu w ten sam sposób. Wyróżniającymi pozytywnymi cechami metody autoryzacji przy użyciu klucza graficznego są łatwość i szybkość użytkowania. Szybkość odblokowania pokazała najlepsze wyniki ze wszystkich badanych metod autoryzacji. Średni czas autoryzacji nie przekraczał 3 sekund, a minimalny i maksymalny czas autoryzacji trwał odpowiednio 1,3 i 3,8 sekundy. Jednocześnie ta metoda ma największą szansę na podejrzenie klucza. Pomimo łatwości i szybkości metody autoryzacji za pomocą hasła graficznego, nie jest zalecane korzystanie z tej metody w celu ochrony ważnych informacji.

Metoda autoryzacji wykorzystująca maskowane hasło znajduje się obecnie tylko w bardzo bezpiecznych systemach, na przykład w systemach bankowych. Ta metoda pokazuje najlepsze wyniki bezpieczeństwa. Nawet jeśli wprowadzono litery hasła jeden raz, niektóre z nich nadal będą nieznane. Badanie metody autoryzacji za pomocą złożonego hasła wykazało, że stosowanie tej metody potrzebuje największej ilości czasu od wszystkich przedstawionych metod autoryzacji. Maksymalny czas autoryzacji wynosił prawie 20 sekund. Najszybsza autoryzacja zajęła 3 sekundy. A średni czas autoryzacji wynosił 10 sekund. Zaleca się stosowanie go w celu zapewnienia maksymalnej ochrony danych lub autoryzacji, co nie musi być wykonywane zbyt często. Użytkownicy ocenili tą metodę jako raczej niski wynik z powodu trudności z wprowadzeniem hasła i ilości czasu spędzonego na wprowadzaniu hasła. Chociaż ta metoda nie jest najbardziej rozpoznawalna wśród użytkowników, jest najbezpieczniejsza ze wszystkich badanych metod w tym artykule.

7. Wnioski

Autoryzacja za pomocą NFC ma wiele zalet i kilka wad. Atutami tej technologii są szybkość, dokładność i mnogość zastosowań. Ze słabości można zauważyć, że technologia nie jest nowa i dla programistów nie została jeszcze szybko opanowana. Technologia NFC może również utrudniać niektóre niuanse bezpieczeństwa. Jednocześnie technologia NFC aktywnie się rozwija i zyskuje na popularności. Technologia jest bardziej odpowiednia do autoryzacji w aplikacjach. Podczas praktycznej analizy podanych metod wykazano, że każda z prezentowanych metod ma swoje zalety i wady, które czynią te technologie wyjątkowymi.

Przy odpowiedzialnym podejściu można stworzyć doskonały klucz ze zwykłego tagu NFC. Dane tego klucza będą trudne do przechwycenia, zastąpienia lub uszkodzenia. Potwierdzają to artykuły i patenty omówione w podanym artykule.

Badanie wykazało zalety technologii NFC tylko w odniesieniu do technologii uwierzytelnienia. W rzeczywistości technologia NFC ma wiele sposobów użycia w innych obszarach, takich jak autoryzacja w systemach różnych aplikacji, płatności zbliżeniowe, transfer danych między urządzeniami i inne. Wszystkie powyższe sposoby użycia są również uzupełnione o dodatkowy poziom bezpieczeństwa.

Literatura

- [1] www.androidauthority.com/what-is-nfc-270730/ [30.06.2019]
- [2] www.arstechnica.com/features/2018/01/a-practical-guide-to-microchip-implants/oggle.it [01.03.2018]
- [3] www.medium.com/@ssaurel/create-a-nfc-reader-application-for-android-74cf24f38a6f74cf24f38a6f&usg=AOvVaw00JumUuBVTM1fhpfM6Rk0 [09.11.2017]
- [4] www.patents.google.com/patent/US8831514B2/en [09.09.2014]
- [5] www.patents.google.com/patent/US20140266624A1/en [18.09.2014]
- [6] www.ieeexplore.ieee.org/abstract/document/6449282 [04.02.2013]
- [7] www.ieeexplore.ieee.org/abstract/document/6470914 [11.03.2013]
- [8] www.patents.google.com/patent/US9276643B2/en [01.03.2016]
- [9] www.sciencedirect.com/science/article/pii/S0167923613002509 [01.03.2014]
- [10] www.patents.google.com/patent/US8478196B1/en [02.07.2013]
- [11] www.ieeexplore.ieee.org/abstract/document/7568941 [15.09.2016]
- [12] www.ieeexplore.ieee.org/abstract/document/7153937 [13.07.2015]
- [13] www.ieeexplore.ieee.org/abstract/document/6698315 [06.01.2014]
- [14] www.ieeexplore.ieee.org/abstract/document/6578295 [16.08.2013]
- [15] www.medium.com/google-developer-experts/exploring-the-v28-android-design-support-library-2c96c6031ae8 [19.03.2018]

Comparative analysis of database systems dedicated for Android

Analiza porównawcza systemów baz danych dedykowanych dla systemu Android

Kamil Wałachowski*, Grzegorz Kozieł

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

The article presents a comparative analysis of mobile databases dedicated for Android. The comparative analysis was carried out on the example of a relational SQLite database with selected nonrelational databases: Realm, ObjectBox and SnappyDB. Theoretical issues were discussed, such as stored data types. Research was carried out to check the performance of mobile databases in terms of: saving, editing, deleting, searching and sorting data. CPU and RAM usage were examined during saving data. The research also included a comparison the size of the database files on the internal disk.

Keywords: ObjectBox; Realm; SQLite; SnappyDB

Streszczenie

W artykule przedstawiona została analiza porównawcza mobilnych baz danych dedykowanych dla systemu Android. Analizę porównawczą przeprowadzono na przykładzie relacyjnej bazy danych SQLite z wybranymi nierelacyjnymi bazami danych: Realm, ObjectBox oraz SnappyDB. Omówione zostały zagadnienia teoretyczne takie jak przechowywane typy danych. Przeprowadzone zostały badania, mające na celu sprawdzenie wydajności mobilnych baz pod kątem: zapisu, edycji, usuwania, wyszukiwania oraz sortowania danych. Zbadane zostało wykorzystanie procesora oraz pamięci RAM podczas zapisu danych. Badania objęły także porównanie wielkości plików baz danych znajdujących się na dysku wewnętrznym.

Słowa kluczowe: ObjectBox; Realm; SQLite; SnappyDB

*Corresponding author

Email address: kamil.walachowski@gmail.com (K. Wałachowski)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Co roku obserwowany jest znaczący wzrost liczby urządzeń mobilnych znajdujących się w posiadaniu ludzi. Liczba aktywnych kart sim wydanych przez polskich operatorów telekomunikacyjnych jest większa od liczby mieszkańców Polski [1,2]. Świadczy to o dużym zapotrzebowaniu na urządzenia mobilne i dostępne za ich pomocą usługi. Ponad 92% Polaków korzysta z telefonów komórkowych [3]. Ich podstawowym zastosowaniem jest komunikacja [3]. Jednak możliwości urządzeń mobilnych, zbliżone do komputerów stacjonarnych, pozwalają na użycie ich na wiele innych sposobów, o czym świadczy bogactwo różnorodnych aplikacji dostępnych na tego typu urządzenia [4].

Wiele z tych aplikacji wymaga przechowywania danych na urządzeniu mobilnym. Realizacja tego zadania możliwa jest na wiele sposobów. Jednak istotne jest zapewnienie możliwości trwałego zapisu danych, oferującego możliwość łatwego i wydajnego dostępu do nich, wyszukiwania oraz modyfikowania treści, a także ich usuwania w razie potrzeby. Funkcje te oferują bazy danych, które stały się nieodzowne w każdej niemal aplikacji.

Początkowo wszystkie aplikacje mobilne korzystały z jedynej dostępnej na rynku relacyjnej bazy danych SQLite. Obecnie na rynku pojawiają się nowe bazy NoSql, mogące w przyszłości zastąpić popularną bazę SQLite. Bazy NoSql udostępniają mechanizm przechowywania danych w bazach, w których sposób modelowania jest inny niż tabelaryczny. W niniejszym artykule podjęto się analizy porównawczej czterech wybranych baz danych dostępnych dla urządzeń pracujących pod kontrolą systemu operacyjnego Android [5].

2. Mobilne bazy danych

SQLite to biblioteka implementująca niezależny, bezserwerowy, bezkonfiguracyjny silnik bazy danych. Kod SQLite jest własnością publiczną, a zatem jest darmowy do użytku w dowolnym celu, komercyjnym lub prywatnym. SQLite to wbudowany silnik bazy danych SQL (ang. Structured Query Language). W przeciwieństwie do większości innych baz danych SQL, SQLite nie ma osobnego procesu serwera. SQLite czyta i zapisuje dane bezpośrednio do zwykłych plików na dysku. Kompletna baza danych SQL z wieloma tabelami, indeksami, wyzwalaczami i widokami znajduje się w jednym pliku na dysku. Format pliku bazy danych jest wieloplatformowy, swobodnie można przenosić bazę danych między systemami 32-bitowymi i 64-bitowymi [6].

Realm jest to obiektowa baza danych NoSql zaprojektowana na platformę mobilną. Jej rdzeń składa się z niezależnej biblioteki C++. Główną zaletą bazy Realm jest prostota, dane są bezpośrednio ujawniane jako obiekty i mogą być wyszukiwane za pomocą kodu. Realm do przechowywania plików wykorzystuje własne pliki, które można bezpiecznie przenosić między syste-

mami Android oraz iOS. Realm umożliwia łatwą pracę z wieloma wątkami, umożliwia także szyfrowanie danych przy pomocy AES-256 [7].

ObjectBox jest to lokalna baza danych, zorientowana obiektowo, wspierająca obsługę relacji. Baza ta obsługuje wiele platform i języków opartych na JVM (ang. Java Virtual Machine), takich jak Java oraz Kotlin. ObjectBox, podobnie jak Realm, odznacza się prostotą pisania kodu, posiada wiele gotowych funkcji ułatwiających pracę. Ponad to ObjectBox wspiera wielowątkowość. Wszystkie dane zapisywane są w pojedynczym pliku bazy danych [8].

SnappyDB jest najmniej zaawansowaną bazą danych spośród prezentowanych. SnappyDB jest to darmowa baza typu klucz-wartość bazująca na bazie LevelDB, dedykowana dla systemu Android. Pozwala ona przechowywać i pobierać prymitywne typy, pozwala także na serializację obiektów i tabel w bezpieczny sposób. SnappyDB nie wspiera transakcji, relacji, wielowątkowości. Cała baza mieści się w jednym pliku znajdującym się w pamięci wewnętrznej urządzenia mobilnego [9].

Wersje testowanych baz:

- Realm 5.12.0;
- SnappyDB 0.4.0;
- SQLite 3.8;
- ObjectBox 2.3.4.

3. Przegląd literatury

Trevor Perrier oraz Fahad Pervaiz w artykule „SQL in a Mobile World Benchmarking Embedded Mobile Databases” dokonują oceny trzech typów modeli przechowywania danych na urządzeniach mobilnych. Testy zostały przeprowadzone na bazie relacyjnej SQLite, bazie typu klucz-wartość CouchLite oraz bazie obiektowej DbO4. Autorzy opracowali kompleksowy zestaw testów, aby przetestować każdą bazę danych w aplikacji na systemie Android. Testy porównawcze polegały na utworzeniu plików bazy danych, skonfigurowaniu dowolnego niezbędnego schematu, a następnie uruchomieniu każdego z sześciu zapytań typu CRUD dziesięć razy. Przeprowadzone przez autorów testy wyraźnie pokazują, że model relacyjny ma najlepsze wyniki przy dużym obciążeniu i najbardziej skomplikowanych zapytaniach. Baza obiektowa DbO4 była najszybsza w przypadku tworzenia, wstawiania oraz odczytu prostych zapytań [10].

W artykule „Mobile Data Store Platforms: Test Case based Performance Evaluation” K. Kussainova i B. Kumalakova dokonano porównania baz typu SQL oraz NoSql. Na potrzeby przeprowadzanych testów autorzy wybrali bazę relacyjną SQLite, bazę obiektową Realm oraz bazę typu klucz-wartość SnappyDB. Bazy zostały zbadane pod kątem zużycia pamięci, czasu wykonywania funkcji zapisu oraz odczytu. Zostały stworzone zapytania odczytujące nawet 12000 rekordów. Według przeprowadzonych testów Realm okazał się bazą najszybciej odczytującą proste zapytania, aż 12000 rekordów w 4ms. Badania wykazały także, że

SnappyDB ma najmniejsze zużycie pamięci RAM, średnio 12MB [11].

W artykule „A Comparison of Lightweight Databases in Mobile Systems” twórcy badają systemy baz danych specjalnie zaprojektowane i zoptymalizowane dla systemów mobilnych. Autorzy pracy zwracają uwagę na ograniczenia systemów mobilnych - pamięć oraz źródło energii (baterie). Praca opisuje konieczność zastosowania tzw. lekkich systemów bazodanowych oraz przedstawia i krótko charakteryzuje każdą z trzech baz danych - SQLite, DbO4 oraz H2. Do porównania wcześniej wymienionych baz wykorzystane zostało narzędzie PolePosition13. Narzędzie to posiada pięć standardowych testów porównawczych do oceny szybkości działania baz danych. Wykonano trzynaście różnych testów mających na celu sprawdzenie wydajności baz. Na podstawie przeprowadzonych badań autorzy stwierdzili, że baza relacyjna H2 jest najszybszą i najbezpieczniejszą bazą danych wykorzystywaną na urządzeniach mobilnych [12].

Artykuł „Porównanie możliwości wykorzystania oraz analiza wydajności baz danych na systemach mobilnych” przedstawia formy składowania danych na urządzeniach mobilnych oraz opisuje sposoby połączeń z zewnętrznymi systemami bazodanowymi takimi jak MySQL, PostgreSQL a także Microsoft SQL Server. Twórcy w swojej pracy starają się odpowiedzieć na pytanie, który sposób przechowywania danych ma największy sens w poszczególnych przypadkach. Do przeprowadzenia testów wykorzystane zostały dwa urządzenia mobilne: jeden z systemem Android, drugi z systemem Windows Phone. Badanie zapisu oraz odczytu zostało przeprowadzone dla:

- par klucz-wartość;
- rekordów z danymi tekstowymi;
- rekordów z danymi numerycznymi;
- rekordów z danymi binarnymi.

Zapisywanych oraz odczytywanych było kolejno 10, 100, 1000, 10000, 100000 oraz 1000000 rekordów. Na podstawie przeprowadzonych badań na obu telefonach, autorzy zauważyli, że najdłuższe czasy wykonywania operacji zapisu oraz odczytu notuje lokalna baza SQLite. Autorzy spostrzegli także, że wszystkie testowane bazy danych są efektywne do ok. 10000 rekordów, wówczas czasy wykonywania operacji odczytu i zapisu nie przekraczają kilkunastu sekund. Badania wykazały, że najlepszą bazą znajdującą się na zewnętrznym serwerze jest Microsoft SQL Server [13].

4. Metodyka badawcza

4.1. Scenariusze badawcze

Badaniom poddane zostały wybrane bazy danych dostępne dla systemu Android. Badania zostały przeprowadzone na czterech bazach: Realm, SQLite, ObjectBox oraz SnappyDB. Celem przeprowadzenia badań stworzony został zestaw danych. Dane zostały wygenerowane automatycznie przy pomocy darmowej biblioteki Java Faker [14]. Wszystkie niżej opisane badania zostały przeprowadzone pięciokrotnie na dwóch telefo-

nach, w celu otrzymania rzetelnych wyników. Testy przeprowadzane były na urządzeniach znajdujących się w stanie optymalnym. Oznacza to, że wszystkie aplikacje działające w tle zostały wyłączone, działały tylko niezbędne usługi. Dodatkowo przed wykonaniem każdej próby, wyczyszczone zostały dane aplikacji oraz pamięć podręczna.

Operacja zapisu danych została przeprowadzona z wykorzystaniem transakcji w technologiach, które to umożliwiają. Zbadany został czas dodawania kolejno 100, 1000, 10000 oraz 100000 rekordów. SnappyDB nie wspiera transakcji, dlatego dodawanie elementów odbywało się pojedynczo przy pomocy pętli for.

Operacje usuwania oraz edycji danych zostały wykonane na bazie zawierającej 100000 rekordów. Zmierzony został czas usuwania, a także edycji 100, 1000, 10000 oraz 100000 rekordów z wykorzystaniem transakcji. W przypadku operacji edycji aktualizowane było pole imię. Analogicznie jak w przypadku operacji dodawania, usuwanie i aktualizacje danych w bazie SnappyDB odbywała się przy pomocy pętli for.

Operacje wyszukiwania danych zostały przeprowadzone na bazie zawierającej 100000 rekordów. Pierwszy test obejmował wyszukiwanie danych po ciągu znaku, z tabeli wyszukane zostało słowo kluczowe 'monitor'. Kolejne badanie miało za zadanie sprawdzenie prędkości odczytu zapytań warunkowych. Eksperymenty wyszukiwania nie zostały wykonane na bazie SnappyDB, ponieważ nie wspiera on wykonywania zapytań złożonych. Poniżej przedstawione zostały zapytania wykorzystane do eksperymentu:

1. `SELECT * FROM users WHERE hardware = "monitor";`
2. `SELECT * FROM users WHERE age > 10 AND CONTAINS(country, "Ja") LIMIT 10;`
3. `SELECT DISTINCT country FROM users WHERE weight > 50 AND height < 180;`

Operacja sortowania wykonana została na bazie zawierającej 100000 rekordów. Badanie to zostało wykonane dla baz umożliwiających sortowanie – SQLite, Realm oraz ObjectBox. Posortowany został wiek użytkowników od najmniejszego do największego.

Ostatnie badanie miało na celu sprawdzenie wykorzystania zasobów urządzenia mobilnego. Testy obejmowały sprawdzenie wykorzystania pamięci RAM oraz procesora podczas zapisu 100000 rekordów do pustej bazy. Dodatkowo sprawdzone zostały rozmiary plików baz danych znajdujących się na dysku wewnętrznym telefonu.

4.2. Stanowisko badawcze

Badania przeprowadzone zostały na dwóch urządzeniach mobilnych o różnej specyfikacji. Pierwszym z nich był smartfon HUAWEI P9 Lite mini o specyfikacji:

- CPU: Qualcomm Snapdragon 425 8917, 4x1,4 GHz;
- Pamięć RAM: 2 GB;
- GPU: Adreno 308 @500 MHz;
- Pamięć wbudowana: 16GB;
- System: Android 7.0 Nougat.

Drugim urządzeniem, na którym przeprowadzono badania, był HUAWEI P10 Lite mini o specyfikacji:

- CPU: HiSilicon Kirin 658, 8x2,1 GHz;
- Pamięć RAM: 4 GB;
- GPU: ARM Mali-T830 MP2;
- Pamięć wbudowana: 16GB;
- System: Android 8.0 Oreo.

5. Eksperyment

5.1. Typy danych

Tabela 1 przedstawia zestawienie typów danych dostępnych w badanych mobilnych bazach danych.

Tabela 1: Dostępne typy danych w poszczególnych bazach

Grupa typów	SQLite	Realm	ObjectBox	SnappyDB
Typy znakowe	TEXT	STRING	STRING CHARACTER	STRING
Typy numeryczne	INTEGER REAL	INT LONG* FLOAT DOUBLE	INTEGER SHORT LONG FLOAT DOUBLE BYTE	INT SHORT LONG DOUBLE FLOAT
Typy daty	-	DATE	DATE	-
Typy logiczne	-	BOOLEAN	BOOLEAN	BOOLEAN
Typy binarne	BLOB	-	-	-
Inne	NULL	REALMMODEL REALMLIST	-	-

SQLite, w przeciwieństwie do baz obiektowych, nie posiada typu logicznego boolean. Zamiast tego wartości logiczne są przechowywane jako liczby całkowite 0 (fałsz) i 1 (prawda). Podobnie jak w przypadku boolean, SQLite nie ma zarezerwowanej klasy pamięci do przechowywania dat. Zamiast tego wbudowane funkcje SQLite mogą przechowywać daty i godziny jako wartości TEXT, REAL lub INTEGER [5,6].

Realm, poza powszechnie wykorzystywanymi typami, daje do dyspozycji także nowatorskie rozwiązania. Innowacyjnie przedstawia się tutaj możliwość przechowywania listy - RealmList <T>, gdzie T może być następującymi typami: String, Integer, Boolean, Float, Double, Short, Long, Byte, byte[] oraz Date [7].

Bazy SnappyDB oraz ObjectBox obsługują podstawowe typy danych, które nie wyróżniają się na tle innych baz.

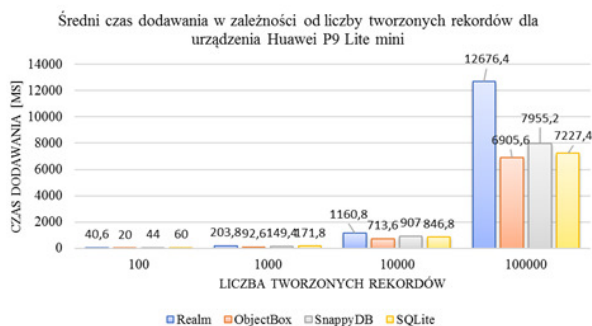
5.2. Zapis danych do bazy

Rysunek 1 przedstawia średnie czasy zapisu danych w zależności od liczby tworzonych rekordów dla urządzenia Huawei P9 Lite.

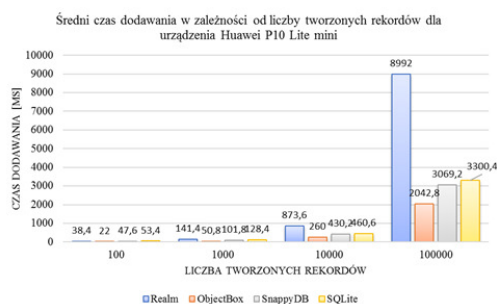
Rysunek 2 przedstawia średnie czasy zapisu danych w zależności od liczby tworzonych rekordów dla urządzenia Huawei P10 Lite mini.

Najlepsze czasy zapisywania danych na obu urządzeniach zanotowała baza ObjectBox. Najgorsze wyniki osiągnęła baza obiektowa Realm. Wszystkie badane bazy prezentują porównywalne wyniki, gdy zapisywane jest mniej niż 1000 rekordów, różnica czasu zapisu między najgorszą, a najlepszą bazą wynosi zaledwie 0,1

sekundy. W codziennym użytkowaniu aplikacji jest to nieodczuwalne.



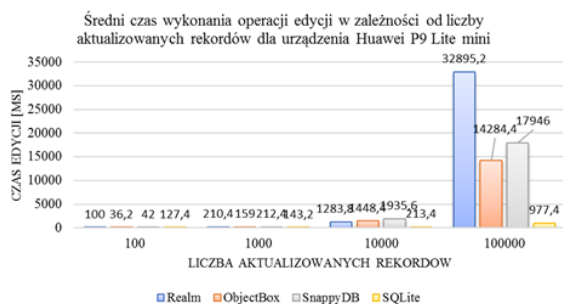
Rysunek 1: Średni czas dodawania w zależności od liczby tworzonych rekordów dla urządzenia Huawei P9 lite mini



Rysunek 2: Średni czas dodawania w zależności od liczby tworzonych rekordów dla urządzenia Huawei P10 lite mini

5.3. Edycja danych

Na rysunku 3 przedstawiony został średni czas wykonania operacji edycji w zależności od liczby aktualizowanych rekordów dla urządzenia Huawei P9 Lite mini.



Rysunek 3: Średni czas wykonania operacji edycji w zależności od liczby aktualizowanych rekordów dla urządzenia Huawei P9 Lite mini

Na rysunku 4 przedstawiony został średni czas wykonywania operacji edycji w zależności od liczby aktualizowanych rekordów dla urządzenia Huawei P10 Lite mini.

W drugim przeprowadzonym badaniu najlepsze czasy edycji na obu urządzeniach zanotowała baza SQLite. W przypadku aktualizacji 100000 rekordów na urządzeniu Huawei P10 Lite, czas edycji w bazie SQLite był 33 razy szybszy w porównaniu do najwolniejszej bazy Realm oraz 5,9 razy szybszy w porównaniu do drugiej co do prędkości edycji bazy ObjectBox. Podobnie jak w przypadku zapisu danych różnice czasów edycji do 1000 rekordów nie przekraczają 0,1 sekundy. Analo-

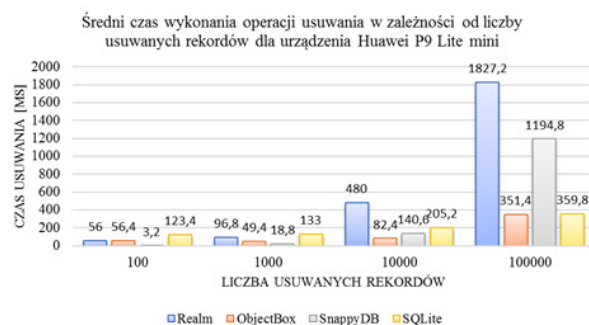
gicznie ma się sytuacja między urządzeniami, w przypadku edycji do 1000 rekordów, czasy są niemalże identycznie i różnice między prędkościami poszczególnych baz nie przekraczają 50 ms.



Rysunek 4: Średni czas wykonania operacji edycji w zależności od liczby aktualizowanych rekordów dla urządzenia Huawei P10 Lite mini

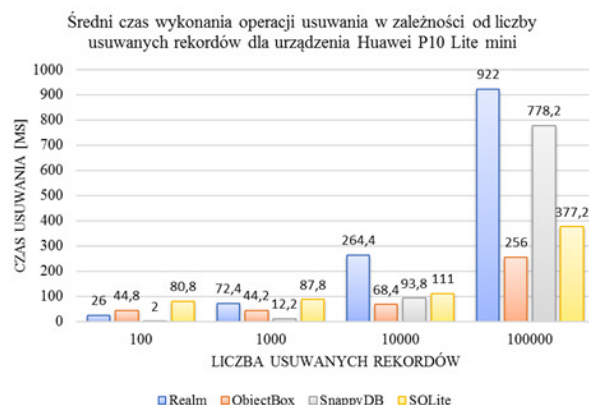
5.4. Usuwanie danych

Rysunek 5 przedstawia średnie czasy wykonania operacji usuwania w zależności od liczby rekordów dla urządzenia Huawei P9 Lite mini.



Rysunek 5: Średni czas wykonania operacji usuwania w zależności od liczby usuwanych rekordów dla urządzenia Huawei P9 Lite mini

Rysunek 6 przedstawia średnie czasy wykonania operacji usuwania w zależności od liczby rekordów dla urządzenia Huawei P10 Lite mini.



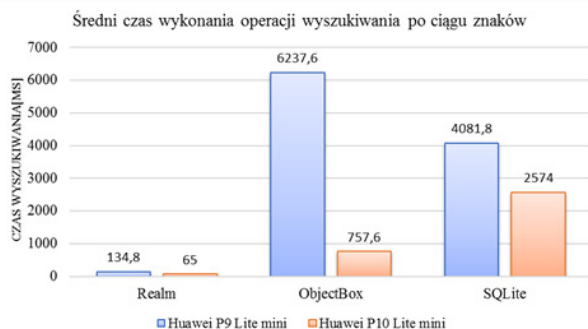
Rysunek 6: Średni czas wykonania operacji usuwania w zależności od liczby usuwanych rekordów dla urządzenia Huawei P10 Lite mini

Najlepsze czasy usuwania na obu urządzeniach osiągnęła baza obiektowa ObjectBox. Najgorsze wyniki ponownie zanotowała baza Realm. Poniżej 1000 usu-

wanych rekordów najdłuższe czasy odczytu uzyskuje SQLite. SnappyDB ma najmniejsze czasy usuwania, gdy usuwane jest mniej niż 1000 rekordów. Baza ta potrafi usunąć 100 rekordów w czasie 2 milisekund, niestety jej wydajność znacząco spada w przypadku usuwania liczby rekordów większej niż 1000. Gdy liczba rekordów jest mniejsza niż 1000, różnica w czasach usuwania badanych baz nie przekracza 0,1 sekundy.

5.5. Wyszukiwanie

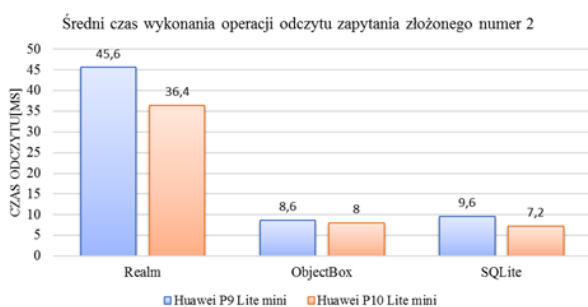
Rysunek 7 przedstawia średnie czasy wykonania operacji wyszukiwania po ciągu znaków.



Rysunek 7: Średni czas wykonania operacji wyszukiwania po ciągu znaków

Najlepsze czasy wyszukiwania danych po ciągu znaku na obu urządzeniach notuje obiektowa baza Realm. Na urządzeniu P10 Lite wyszukuje ona dane 11,6 razy szybciej niż baza ObjectBox oraz 39,6 razy szybciej niż SQLite.

Rysunek 8 przedstawia średni czas wykonania operacji odczytu poniższego zapytania złożonego. `SELECT * FROM users WHERE age > 10 AND CONTAINS(country, "Ja") LIMIT 10;`

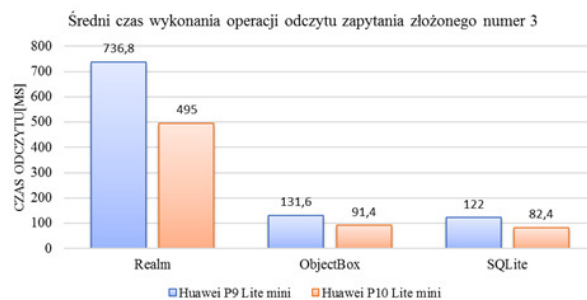


Rysunek 8: Średni czas wykonania operacji odczytu zapytania złożonego numer 2

Rysunek 9 przedstawia średni czas wykonania operacji odczytu poniższego zapytania złożonego: `SELECT DISTINCT country FROM users WHERE weight > 50 AND height < 180;`

Najlepszym wynikiem w przypadku wykonania zapytań złożonych może poszczycić się baza SQLite, niemalże na równi z bazą ObjectBox. W przypadku tych dwóch baz czas odczytu złożonego zapytania numer 2 nie przekraczał 10 milisekund na obu urządzeniach. Najgorsze wyniki zanotowała baza Realm, w przypadku zapytania drugiego notująca na urządzeniu Huawei P9

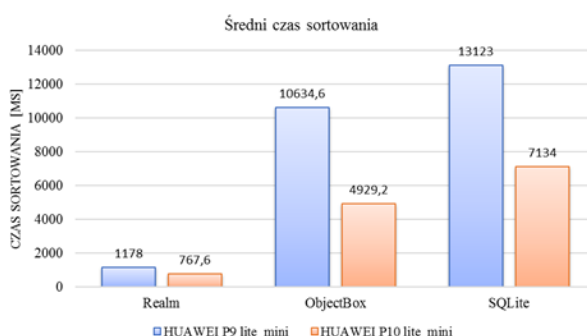
Lite mini czasy 5,3 razy gorsze od najszybszej w tym teście bazy ObjectBox.



Rysunek 9: Średni czas wykonania operacji odczytu zapytania złożonego numer 3

5.6. Sortowanie

Rysunek 10 przedstawia średni czas sortowania danych.

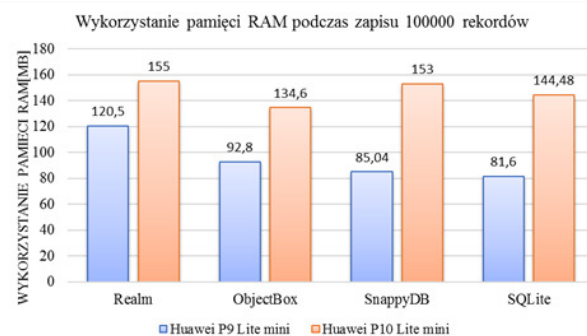


Rysunek 10: Średni czas sortowania wieku użytkowników od najmniejszego do największego.

Najlepsze czasy sortowania rekordów uzyskuje baza Realm. W przypadku operacji wykonanych na telefonie Huawei P10 Lite mini, czasy sortowania bazy Realm są 6,4 razy krótsze niż czasu konkurencyjnej, obiektowej bazy ObjectBox. Najgorzej w tym teście wypadła baza SQLite, notująca najdłuższe czasy wykonania operacji sortowania. Różnica między najlepszą, a najgorszą bazą testowaną na urządzeniu P9 Lite wynosi aż 11,94 sekundy. Różnica ta jest bardzo duża i byłaby wyczuwalna podczas normalnego użytkowania aplikacji.

5.7. Wykorzystywanie zasobów

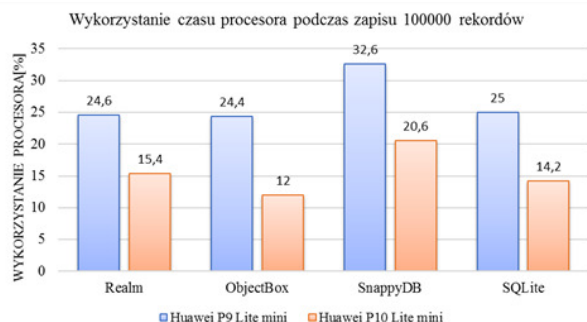
Rysunek 11 przedstawia wykorzystanie pamięci RAM podczas zapisu 100000 rekordów.



Rysunek 11: Średnie wykorzystanie pamięci RAM podczas zapisu 100000 rekordów

Największe zużycie pamięci RAM powoduje Realm. W przypadku badania wykonanego na telefonie Huawei P10 Lite, badane bazy prezentują niemalże identyczne zużycie pamięci RAM. Podczas operacji zapisu, różnica między największym, a najmniejszym zużyciem wynosi 20MB. Wykorzystanie przez bazę Realm 155MB pamięci RAM stanowi zaledwie 4% pamięci RAM posiadanej przez model P10 Lite.

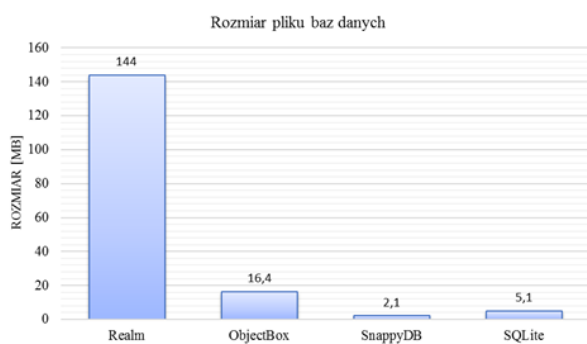
Rysunek 12 przedstawia wykorzystanie czasu procesora podczas zapisu 100000 rekordów.



Rysunek 12: Średnie wykorzystanie czasu procesora podczas zapisu 100000 rekordów

Podczas operacji zapisu największe obciążenie procesora na obu urządzeniach generuje baza SnappyDB. Pozostałe testowane bazy wykorzystują procesor w podobny sposób. W przypadku pracy na urządzeniu Huawei P9 Lite, zużycie procesora w czasie wykonania operacji zapisu jest niemalże identyczne dla baz SQLite, Realm oraz ObjectBox i wynosi około 25%. Dla porównania SnappyDB wykonując tę samą operację na tym samym telefonie obciąża procesor aż w 33%.

Rysunek 13 przedstawia rozmiary plików baz danych znajdujących się na dysku wewnętrznym.



Rysunek 13: Rozmiar pliku bazy danych

Na podstawie przeprowadzonego badania można zaobserwować, że rozmiar pliku bazy Realm jest kilkukrotnie większy od pozostałych. Najmniej miejsca na dysku zajmuje prosta baza SnappyDB, która spośród testowanych baz posiada najmniej funkcjonalności. W przypadku zapisanych 100000 rekordów baza SQLite zajmuje jedynie 5,1MB. Według dokumentacji, maksymalny rozmiar bazy danych może wynosić około 1,4E14 bajtów (140 terabajtów) [5].

6. Wnioski

Celem niniejszego artykułu było porównanie baz danych dedykowanych dla systemu Android. Na podstawie przeprowadzonych badań należy podkreślić, że nie można jednoznacznie wyznaczyć najlepszej bazy danych. Przy wyborze systemu bazodanowego, najważniejsze jest określenie indywidualnych potrzeb stawianych przez wymagania aplikacji, oraz wybór pod tym względem odpowiedniej bazy danych.

Podczas wykonywania operacji CRUD wszystkie z badanych baz wykazywały niemalże podobne czasy w przypadku pracy z liczbą rekordów mniejszą niż 1000. Podczas operacji zapisu, edycji oraz usuwania dużej liczby rekordów najgorsze czasy notowała baza Realm. Najlepiej z aktualizacją rekordów poradziła sobie relacyjna baza SQLite, wykonująca operacje edycji kilkukrotnie szybciej niż pozostałe badane rozwiązania bazodanowe. Badanie operacji wyszukiwania danych ciągu znaku wykazało, że baza Realm jest kilkunastokrotnie szybsza od bazy SQLite. W przypadku odczytu złożonych zapytań najlepsze czasy notowała baza SQLite oraz ObjectBox, mająca niewiele słabsze czasy. Operacja sortowania najszybciej wykonywana była na bazie Realm. Różnice czasu sortowania pomiędzy bazą Realm, a pozostałymi badanymi technologiami wynosiły nawet kilka sekund. Wykorzystanie zasobów takich jak procesor oraz RAM było niewielkie w stosunku do całkowitych zasobów badanych smartfonów. Największą rozbieżność można zauważyć w przypadku badania wielkości pliku bazy danych. Rozmiar pliku bazy Realm jest wielokrotnie większy od pozostałych, co w przypadku obecnie wykorzystywanych smartfonów, posiadających ponad 100GB pamięci wewnętrznej, jest mało znaczące.

Literatura

- [1] Ranking użytkowników sieci komórkowej <https://www.parkiet.com/Analizy/305159869-Liczba-kart-SIM-W-I-kwartale-2019-roku-na-rynku-telefonii-komorkowej-w-Polsce-widac-stabilizacje-baz-klientow.html> [03.11.2019]
- [2] Raport Digital 2019 <https://mobirank.pl/2019/02/01/raport-mobile-i-digital-na-swiecie-w-2019-roku/> [03.11.2019]
- [3] Sondaż CBOS na temat wykorzystania telefonów komórkowych <https://technologia.dziennik.pl/internet/artykuly/555355.sondaz-telefon-y-komrkowe-smartfony.html> [03.11.2019]
- [4] Strona sklepu Google Play <https://play.google.com/store/apps?hl=pl> [03.11.2019]
- [5] Kumar Karn Vikas, Kumar Aditya Sunny, Android SQLite Essentials, Packt Publishing, 2014
- [6] Dokumentacja SQLite <https://www.sqlite.org/docs.html> [03.11.2019]
- [7] Dokumentacja Realm <https://realm.io/docs/java/latest/> [03.11.2019]
- [8] Dokumentacja ObjectBox <https://docs.objectbox.io/getting-started> [03.11.2019]

- [9] Oficjalna strona projektu SnappyDB <https://github.com/nhachicha/SnappyDB> [03.11.2019]
- [10] T. Perrier, F. Pervaiz, NoSQL in a Mobile World Benchmarking Embedded Mobile Databases, 2013
- [11] K. Kussainov, B. Kumalakov, Mobile Data Store Platforms: Test Case based Performance Evaluation, 2016
- [12] Farzad Tavakkoli, Azam Andalib, Asadollah Shahbahrani and Reza Ebrahimi Atani, A Comparison of Lightweight Databases in Mobile Systems, 2011
- [13] M. Grudzień, K. Korgol, D. Gutek, Comparison of the possible uses and performance analysis of databases on mobile operating systems, 2016
- [14] Oficjalna strona projektu Java Faker <https://github.com/DiUS/java-faker> [03.11.2019]

Digital services in Poland and the Far East - comparative analysis

Analiza porównawcza usług cyfrowych między Polską, a Dalekim Wschodem

Dominik Klej*

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

The following article focuses on illustrating the content of the master thesis with the same title. The methodology of developed by author digital indicator will be presented, as well as the results and conclusions based on the comparative analysis. The research concerns the comparative analysis between digital services in Poland, Japan and China, based on the IDI, WDC and ITC digital indicators. The analysis is based on publicly available data of digitization.

Keywords: digital services; DAI; Poland; Japan

Streszczenie

Artykuł skupia się na zobrazowaniu treści pracy magisterskiej o tym samym tytule. Przedstawiona w nim zostanie metodologia opracowanego, autorskiego wskaźnika cyfryzacji, a także jego wyniki i wnioski na podstawie przeprowadzonych badań porównawczych. Badania dotyczą analizy porównawczej pomiędzy usługami cyfrowymi w Polsce, Japonii oraz Chinach, na podstawie wskaźników cyfryzacji IDI, WDC oraz ITC. Analiza odbywa się na podstawie publicznie dostępnych danych na temat cyfryzacji.

Słowa kluczowe: usługi cyfrowe; DAI; Polska; Japonia

*Corresponding author

Email address: dominik.klej@gmail.com (D. Klej)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Niniejszy artykuł przedstawia zestawienie wskaźnika cyfryzacji DAI (Digital Adoption Index), opracowanego przez Bank Światowy ze wskaźnikiem DDI (Digital Development Index) opracowanym przez autora artykułu. Celem artykułu jest dostarczenie rzetelnie opracowanych informacji statystycznych oraz wniosków wysnutych na podstawie ich analizy, w sposób transparentny, aby czytelnik mógł się zapoznać z przebiegiem prowadzonych w pracy magisterskiej działań, na każdym kroku postępowania pracy. Tak opracowane dane będą mogły w obrazowy sposób zaprezentować różnice pomiędzy wymienionymi krajami, na które osoby zainteresowane niniejszą tematyką, będą mogły się powołać w dowolnie zdefiniowanym przez siebie celu.

Powodem napisania artykułu jest niewielka ilość aktualnych źródeł informacji i opracowań nawiązujących do przedstawionej w tytule pracy tematyki.

2. Metodologia i pojęcia

2.1. Metodologia

Do przeprowadzenia analizy porównawczej zostały wykorzystane oficjalne, publicznie dostępne dane (w głównej mierze ze strony Banku Światowego). Zebrane dane poddane zostały obróbce statystycznej oraz interpretacji uzyskanych wyników, na skutek czego zostały opracowane wnioski dotyczące poziomu stopnia cyfryzacji wybranych krajów.

Uzyskane na skutek wyszukiwania dane, zostały odpowiednio podzielone i zgrupowane tematycznie. Na-

stępnie z poszczególnych klastrów tematycznych zostały wyliczone średnie ważone.

Wyniki uzyskane na skutek całego opisanego procesu, zostały zestawione i porównane z wynikami wskaźnika cyfryzacji DAI (Digital Adoption Index), opracowanego przez Bank Światowy.

2.2. Pojęcia

W niniejszym podrozdziale zostały przedstawione niezbędne informacje do zrozumienia zagadnień i metodologii działania podczas wykonywanych działań statystycznych i analiz.

Cyfryzacja – to pojęcie wyrażające rozpowszechnienie oraz skalę wpływu technologii informatycznych i techniki cyfrowej na życie codzienne i gospodarkę danego kraju. Cyfryzacja często nazywana jest również „dygitalizacją” lub „digitalizacją”, czyli spolszczoną wersją wyrazu „digitalization”. Jednakże nie można utożsamiać ze sobą tych dwóch pojęć jako, że dygitalizacja oznacza „nadawanie postaci cyfrowej danym pisanym i drukowanym” [1].

Treści cyfrowe – „to wszelkie informacje, do których użytkownik końcowy posiada dostęp za pośrednictwem sieci, lub innych kanałów pośredniczących (jak CD lub DVD).” [2]

2.3. Wykorzystywane działania statystyczne

Jako, że niniejsza praca naukowa opiera się na działaniach porównawczych wykorzystując działania statystyczne, należy wymienić i opisać wszelkie pojęcia wykorzystane i związane z niniejszą pracą magisterską.

Średnia arytmetyczna ważona – Jest to średnia arytmetyczna elementów, którym przypisywane są różne wagi. Waga poszczególnych elementów średniej służy temu, aby przypisać większe bądź mniejsze znaczenie wybranej wartości, w skali całej obliczanej średniej. Wówczas elementy wchodzące w skład średniej, o większej wadze, będą miały większy wpływ na wartość średniej od elementów posiadających mniejszą wagę. We wzorze 2.1. zmienna w oznacza wagę danej zmiennej wchodzącej w skład średniej.

$$\bar{x} = \frac{w_1x_1 + w_2x_2 + \dots + w_nx_n}{w_1 + w_2 + \dots + w_n} \quad (1)$$

Unitaryzacja – Unitaryzacja zmiennych, jest to działanie polegające na ujednoliceniu zakresu pozyskanych wartości statystycznych. Celem tego zabiegu jest uzyskanie zmiennych o ujednoliconym zakresie zmienności, definiowanym przez różnicę pomiędzy ich wartościami maksymalnymi i minimalnymi w ujęciu klasycznym lub maksimum z medianowych odchyłeń bezwzględnych w ujęciu pozycyjnym, równym stale 1. [3]

Unitaryzacja klasyczna – W unitaryzacji klasycznej parametry normalizacyjne przyjmują wartości z przedziału [0;1]. Wartości te można pozyskać za pomocą wzoru 2.2. W tym wzorze R jest wartością danego parametru wchodzącego w skład zakresu liczb poddawanych unitaryzacji.

$$\frac{R_1 - \min R}{\max R - \min R} \quad (2)$$

Normalizacja – „Jest to działalność zmierzająca do uzyskania optymalnego, w danych okolicznościach, stopnia uporządkowania w określonym zakresie, poprzez ustalenie postanowień przeznaczonych do powszechnego i wielokrotnego stosowania, dotyczących istniejących lub mogących wystąpić problemów” [3]

3. Wskaźnik DAI – opis i metodologia

3.1. Opis

Indeks adopcji cyfrowej, jest to złożony, ogólnosiłowy wskaźnik autorstwa Banku Światowego przy współpracy z firmą Microsoft. Wskaźnik ten mierzy poziom adopcji i rozprzestrzeniania się technologii cyfrowych na terenie poszczególnych krajów z całego świata. Miara tego wskaźnika wyznaczana jest z perspektywy trzech wymiarów ekonomicznych: ludzi, rządu oraz przedsiębiorstw. Index ten opisuje 171 krajów w skali od 0 do 1. Ostateczna miara jest wynikiem obliczenia średniej arytmetycznej z subindeksów. Każdy subindeks obejmuje technologie niezbędne do opisanego rozwoju cyfryzacji, takie jak: zwiększenie produktywności i przyspieszenie szeroko zakrojonego rozwoju dla biznesu, poszerzenie możliwości i poprawa dobrobytu dla ludzi oraz zwiększenie wydajności i odpowiedzialności za świadczenie usług dla rządu. Pierwotnie, indeks adopcji cyfrowej został skonstruowany jako część raportu „World Development Report 2016: Digital Divi-

dends” [4], a później został zaktualizowany w celu odzwierciedlenia nowych źródeł danych i ulepszonej metodologii. Poprzez pomiar względnej adopcji cyfrowej, wskaźnik DAI staje się użyteczny na potrzeby decydentów w kwestii opracowywania strategii cyfrowych z dostosowanymi politykami promującymi adopcję cyfrową w różnych grupach użytkowników.

3.2. Metodologia DAI

Metodologię wyznaczania tego wskaźnika stanowi złożony indeks, który mierzy głębokość i szerokość przyjęcia technologii cyfrowych w 171 krajach, obejmujących każdy region i grupę dochodową. Opiera się na trzech subindeksach sektorowych obejmujących przedsiębiorstwa, ludzi i rządy, przy czym każdy indeks posiada jednakową wagę. **DAI (Ekonomia) = DAI (Przedsiębiorstwa) + DAI (Ludzie) + DAI (Rządy)** **DAI (Przedsiębiorstwa)**: Klaster Przedsiębiorstwa to prosta średnia z czterech znormalizowanych wskaźników: procent firm ze stronami internetowymi, liczba bezpiecznych serwerów, prędkość pobierania i zasięg 3G w kraju. **DAI (Ludzie)**: klaster Ludzie to prosta średnia z dwóch znormalizowanych wskaźników z Gallup World Poll [5]: mobilny dostęp w domu i dostęp do Internetu w domu. **DAI (Rząd)**: Klaster rządowy to prosta średnia z trzech wskaźników częściowych: podstawowych systemów administracyjnych, usług publicznych online i identyfikacji cyfrowej. Dane dotyczące usług publicznych online są dostarczane przez Indeks usług online ONZ. Dane dotyczące podstawowych systemów administracyjnych identyfikacji cyfrowej zostały zebrane przez Bank Światowy. Wskaźniki złożone są znormalizowane w skali od 0 do 1, aby zapewnić, że mają one równy wpływ na zmienność wskaźnika ogólnego. W tabelach 1, 2 oraz 3 przedstawiono konstrukcje poszczególnych klastrów wchodzących w skład wskaźnika DAI.

Tabela 1: Przedstawienie klastra biznesowego z uwzględnieniem wag oraz źródeł dla poszczególnych komponentów [6]

Komponent	Waga	Źródło
Strony firmowe	1/4	WBG Enterprise Surveys
Bezpieczne serwery	1/4	Netcraft
Prędkość pobierania	1/4	Ookla NetIndex
Zasięg 3G	1/4	GSMA

Tabela 2: Przedstawienie klastra rządowego z uwzględnieniem wag oraz źródeł dla poszczególnych komponentów [6]

Komponent	Indykator	Waga	Źródło
System zarządzania informacjami finansowymi	Wykonanie budżetu	1/20	WBG
	Formuowanie budżetu	1/20	
	Pojedyncze konto skarbowe	1/20	
	Źródło danych finansowych	1/20	
System informacji o zasobach ludzkich	Funkcjonalność HRMIS	1/10	
	Funkcjonalność e-płatności	1/10	

E-podatki	Funkcjonalność zarządzania podatkami	1/10	
	Funkcjonalność e-zgłoszeń	1/10	
e-cła	Funkcjonalność e-celna	1/5	
e-zamówienia	Funkcjonalność e-zamówień	1/5	
Dostęp do usług	Zakres usług	1/2	
Podpis cyfrowy	Funkcjonalność podpisu	1/4	
Funkcje karty	Typ karty	1/8	
	Funkcje biometryczne		

Tabela 3: Przedstawienie klastra obywateli z uwzględnieniem wag oraz źródeł dla poszczególnych komponentów [6]

Komponent	Waga	Źródło
Dostęp do urządzeń mobilnych w domach	1/3	Gallup World Poll
Dostęp do Internetu w domach	1/3	Gallup World Poll
Koszt dostępu do Internetu	1/3	TBC

3.3. Wyniki i wnioski

Tabela 4 przedstawia dane z ubiegłych lat na temat wyliczonych wskaźników DAI oraz ich subwskaźników dla rozpatrywanych w niniejszym artykule krajów.

Tabela 4: Zestawienie wskaźników DAI dla wybranych krajów[4]

Kraj	Rok	Indeks Adopcji Cyfrowej	Klastr przedsiębiorstwa	Klastr Ludzie	Klastr Rząd
Chiny	2014	0,501	0,471	0,400	0,632
	2016	0,586	0,548	0,525	0,686
Japonia	2014	0,816	0,731	0,787	0,932
	2016	0,835	0,761	0,835	0,909
Polska	2014	0,651	0,731	0,643	0,578
	2016	0,690	0,756	0,684	0,631

Można zauważyć, iż najwyższą wartością wskaźnika DAI w 2014 i 2016 roku wyznacza się Japonia. Oznacza to, iż spośród wymienionych krajów, poziom adopcji cyfryzacji w Japonii jest najwyższy i pozostawia daleko w tyle Chiny i Polskę. Największy wpływ na wskaźnik ma wysokość wskaźnika DAI dla rządu, podczas gdy w Polsce wskaźnik ten jest najniższy w porównaniu do pozostałych. Ta różnica może wynikać z większej centralizacji Japońskiej gospodarki względem Polskiej, jednakże tą kwestię należało by poddać dodatkowej, szczegółowej analizie, aby jednoznacznie to rozstrzygnąć. Podobna dysproporcja co w Japonii, pojawia się

w Chinach, gdzie ponownie DAI rządowe jest wyraźnie wyższe od pozostałych. Podobnie, porównując subwskaźniki w Polsce i w Chinach, można zauważyć iż pomimo ogólnej przewagi Polski nad Chinami, DAI rządowy w Chinach pozostaje większy od Polskiego. Pokazuje to jak wielka jest różnica w sposobie myślenia i zarządzania gospodarką w sferze usług cyfrowych pomiędzy Polską, a Dalekim Wschodem. Nie da się również pominąć faktu, iż tak jak w porównaniu do Japonii, każdy niemal każdy subwskaźnik DAI w Polsce na przestrzeni lat jest niższy, tak w przypadku Chin sytuacja jest dokładnie odwrotna. Fakt, że dwa tak bliskie sobie kraje geograficznie jak i kulturowo posiadają tak wielką dysproporcję w dziedzinie cyfryzacji na niekorzyść Chin, jest zastanawiająca i z pewnością warta głębszej analizy.

Nie bez znaczenia pozostaje kwestia przyswajania cyfryzacji w przedsiębiorstwach, ponieważ pomimo różnic w pozostałych subwskaźnikach, w przypadku DAI przedsiębiorstw, wskaźnik Polski oraz Japoński jest niemal identyczny.

Ostatecznie, jeśli posługiwać się wyłącznie tym wskaźnikiem, krajem najlepiej adoptującym usługi cyfrowe w każdej płaszczyźnie (przedsiębiorstwa, obywatele, rząd) jest Japonia, a krajem wykazującym się najniższą adaptacją są Chiny.

Aby nie kierować się bezrefleksyjnie wyłącznie podanymi wartościami wskaźnika DAI, należało by przeprowadzić analizę wzbogaconą o rozpatrzenie możliwych scenariuszy dla innych komponentów wchodzących w skład wskaźnika, jako że sam Bank Światowy zaleca wprowadzenie zmian w swoim wskaźniku DAI. Z uwagi na fakt, iż dane zebrane na potrzeby poszczególnych komponentów wskaźnika nie są publicznie dostępne, nie było możliwości sprawdzenia jak zmieniłyby się wartości wskaźnika po zastąpieniu jednego, bądź kilku komponentów wchodzących w skład wskaźnika DAI innymi komponentami. Dlatego też w niniejszym opracowaniu postanowiono zamiast modyfikować opracowany już, i wciąż udoskonalany wskaźnik ekonomiczny – opracować własny, a następnie porównać wyniki.

Jak już wspomniano, niestety, nie wszystkie dane źródłowe są dostępne. W wielu przypadkach, Bank Światowy nie publikuje danych źródłowych, ani też sposobu przeliczania tych danych na potrzeby wskaźnika DAI. Tak też jest w przypadku komponentu „Strony firmowe”, czy „Zasięg 3G”. Podobnie, firmy zewnętrzne, które realizują zbieranie danych dla Banku Światowego nie zawsze są organizacjami non-profit, a zatem również nie upubliczniają tak wrażliwych danych.

4. Własny wskaźnik cyfryzacji (DDI)

Na podstawie własnej analizy oraz publicznie dostępnych danych, udało się zebrać następujące komponenty wchodzące w skład opracowywanego, autorskiego wskaźnika cyfryzacji DDI (Digital Development Index). Podzielono je na trzy grupy:

1. **Platności** – wskazujące jak bardzo dany kraj wspiera płatności elektroniczne, tym samym wypierając

płatności gotówkowe, a także w jakim stopniu wdrożone zostały rozwiązania mające na celu cyfryzację pieniądza,

2. **Eksport** – określa stopień cyfryzacji kraju w postaci eksportu dóbr i usług cyfrowych poza swoje granice. Przypuszczalnie, w krajach o większym stopniu cyfryzacji, poziom eksportu zaawansowanych technologii i usług z nimi związanych do innych krajów będzie wyższy,
3. **Dostęp do cyfryzacji** – uwzględnia rozpowszechnienie cyfryzacji i jej wpływ na życie przeciętnego obywatela, a także to, na ile prosty lub utrudniony dostęp do cyfryzacji i jej dobrodziejstw ma przeciętny człowiek żyjący w danym kraju

4.1. Dobór wag

Do opracowania wag dla poszczególnych elementów autorskiego wskaźnika DDI wykorzystano jedną z naukowych metod decyzyjnych i wielokryterialnego podejmowania decyzji – *Multiple Criteria Desision Making (MCDM)* – o nazwie **AHP** (Analytic Hierarchy Process), która uwzględnia przydzielanie wag poszczególnym komponentom poprzez ankietowanie ekspertów.

Zebrano opinię 10 ankietowanych osób na temat istotności stosunku poszczególnych komponentów, a także grup komponentów wchodzących w skład wskaźnika DDI. Próbką osób została grupa absolwentów studiów magisterskich o kierunku informatycznym do 28 roku życia.

Zadaniem ankietowanych było ocenić istotność stosunku pomiędzy poszczególnymi komponentami na różnych poziomach hierarchii struktury wskaźnika DDI w skali od 1 do 9.

Po zebraniu wyników i przedstawieniu ich w postaci macierzy, na skutek operacji matematycznych wyznaczane są wagi dla każdego z komponentów, a także współczynnik konsekwencji, zgodnie z procedurą stosowaną w metodologii AHP.

4.2. Wskaźnik DDI - wyniki

Tabele 5, 6 oraz 7 przedstawiają wyniki uzyskane dla poszczególnych klastrów autorskiego wskaźnika DDI.

Tabela 5: Klaster „Płatności” autorskiego wskaźnika

Komponent	Kraj	Rok	Wartość	Waga
Płatności elektroniczne wykorzystywane do dokonywania płatności	Polska	2014	0,343	0,397
		2016	-	
	Japonia	2014	1	
		2016	-	
	Chiny	2014	0	
		2016	-	
Telefony komórkowe wykorzystywane do płacenia rachunków	Polska	2014	1	0,237
		2016	-	
	Japonia	2014	0	
		2016	-	
	Chiny	2014	0,579	
		2016	-	
Liczba bankomatów	Polska	2014	0,119	0,157

	Japonia	2016	0,241	0,108
		2014	0,996	
		2016	1	
	Chiny	2014	0	
		2016	0,366	
		2016	0,366	
Własność konta w instytucji finansowej lub u dostawcy usług pieniądza mobilnego	Polska	2014	0	0,066
		2016	-	
	Japonia	2014	1	
		2016	-	
	Chiny	2014	0,565	
		2016	-	
Posiadanie karty kredytowej	Polska	2014	0,02	0,036
		2016	-	
	Japonia	2014	1	
		2016	-	
	Chiny	2014	0	
		2016	-	
Posiadanie karty debetowej	Polska	2014	0,025	0,036
		2016	-	
	Japonia	2014	1	
		2016	-	
	Chiny	2014	0	
		2016	-	

Tabela 6: Klaster „Eksport” autorskiego wskaźnika

Komponent	Kraj	Rok	Wartość	Waga
Eksport towarów ICT	Polska	2014	0,031	0,648
		2016	0	
	Japonia	2014	0,670	
		2016	0,619	
	Chiny	2014	0,969	
		2016	1	
Eksport usług ICT	Polska	2014	0,657	0,253
		2016	0,853	
	Japonia	2014	0	
		2016	0,020	
	Chiny	2014	0,706	
		2016	1	
Stosunek eksportu zaawansowanych technologii do produkowanego eksportu	Polska	2014	0,012	0,451
		2016	0	
	Japonia	2014	0,485	
		2016	0,456	
	Chiny	2014	1	
		2016	0,988	

Tabela 7: Klaster „Dostęp do cyfryzacji” autorskiego wskaźnika

Komponent	Kraj	Rok	Wartość	Waga
Gospodarstwa domowe z dostępem do internetu	Polska	2014	0,543	0,411
		2016	0,622	
	Japonia	2014	0,809	
		2016	1	
	Chiny	2014	0	
		2016	0,168	
Osoby korzystające z internetu	Polska	2014	0	0,234
		2016	0,252	
	Japonia	2014	0,846	
		2016	1	
	Chiny	2014	0,846	
		2016	1	
Stałe subskrypcje	Polska	2014	0,268	0,147

szerokopasmowe	Japonia	2016	0,286	
		2014	0,899	
	Chiny	2016	1	
		2014	0	
Liczba bezpiecznych serwerów	Polska	2016	0,270	0,115
		2014	1	
	Japonia	2016	0,551	
		2014	0,846	
	Chiny	2016	0	
		2014	0,153	
Subskrypcje telefonów komórkowych	Polska	2016	1	0,06
		2014	0,828	
	Japonia	2016	0,544	
		2014	0,673	
	Chiny	2016	0	
		2014	0,08	
Stałe abonamenty telefoniczne	Polska	2016	0,041	0,032
		2014	0	
	Japonia	2016	0,984	
		2014	1	
	Chiny	2016	0,117	
		2014	0,029	

5. Wnioski

Opierając się na wskaźniku DAI jako wzorcu do porównań, a także po obliczeniu wartości autorskiego wskaźnika DDI dla każdego z trzech wymienionych krajów za lata 2014-2016 otrzymano zestawienie przedstawione w tabelach 8 i 9.

Tabela 8: Zestawienie wskaźników DAI dla wybranych krajów [4]

Kraj	Rok	Indeks Adopcji Cyfrowej	Klaster przedsiębiorstwa	Klaster Ludzie	Klaster Rząd
Polska	2014	0,651	0,731	0,643	0,578
	2016	0,690	0,756	0,684	0,631
Japonia	2014	0,816	0,731	0,787	0,932
	2016	0,835	0,761	0,835	0,909
Chiny	2014	0,501	0,471	0,400	0,632
	2016	0,586	0,548	0,525	0,686

Tabela 9: Wyniki autorskiego wskaźnika DDI

Kraj	Rok	Wyniki całkowite	Platności	Eksport	Dostęp do cyfryzacji
Polska	2014	0,224	0,394	0,142	0,355
	2016	0,248	-	0,160	0,552
Japonia	2014	0,703	0,763	0,483	0,815
	2016	0,567	-	0,453	0,963
Chiny	2014	0,368	0,198	0,930	0,202
	2016	0,863	-	0,996	0,402

5.1. Analiza wyników

Porównując wyniki obu wskaźników, najbardziej rzuca się w oczy fakt, iż we wskaźniku DDI Polska wypada znacznie gorzej. Wynika to w największej mierze z najniższego poziomu eksportu zaawansowanych technologii, w których prym wiodą Chiny, zostawiając tak Polskę jak i Japonię daleko w tyle. Jednak wskaźnik eksportu nie jest jedynym niekorzystnym wskaźnikiem dla Polski, ponieważ pozostałe również nie znajdują się na wysokim poziomie. Nawet jeśli Polska przewyższa Chiny pod względem cyfryzacji w sferze płatności oraz dostępu do cyfryzacji dla obywateli swojego kraju, to z uwagi na tę niekorzystną dla Polski różnicę w eksporcie zaawansowanych technologii, wynik całkowity dla Polski jest blisko o połowę niższy.

Z kolei Japonia zdecydowanie zdominowała sferę płatności, jak i dostępu do cyfryzacji. Wynik końcowy Japonii jest zdecydowanie najlepszy spośród wszystkich krajów w rozpatrywanym autorskim wskaźniku DDI, co świadczy o wysokim poziomie cyfryzacji.

Warto również poruszyć kwestię różnicy wartości wskaźnika dla Japonii oraz Chin w roku 2016 względem roku 2014. Wynika to z faktu, iż nie opublikowano wyników dla komponentów znajdujących się w klastrze „płatności” za rok 2016. W związku z tym, iż dane te nie zostały uwzględnione, wyniki za 2016 rok nie uwzględniają wpływu tego klastra na ostateczny wynik. W przypadku Japonii, która w sferze płatności odnotowuje najwyższy poziom cyfryzacji, wiąże się to ze spadkiem o niemal 0,15. Ponadto, w roku 2016 Japonia odnotowuje niższy poziom eksportu. Różnica ta wynika prawdopodobnie z dominującej pozycji Chin, które wraz z upływem czasu coraz bardziej rozwijają swój eksport, przez co ucierpiał wynik dla Japonii. W przypadku Chin z kolei, z uwagi na to, iż w 2014 roku poziom cyfryzacji w sferze płatności był najniższy z pośród wszystkich rozpatrywanych krajów, z powodu jego utraty pozostałe dwa wskaźniki (w tym przede wszystkim eksport) uzyskały większy wpływ na ostateczny wynik – stąd tak ogromna różnica. Ponadto, warto zwrócić uwagę na fakt, iż Chiny odnotowują niemal dwa razy wyższą wartość wskaźnika „dostęp do cyfryzacji”, co również miało duży wpływ na ostateczny wynik za rok 2016.

5.2. Wnioski końcowe

Przedsięwzięcie zmierzenia poziomu cyfryzacji w różnych krajach na świecie o różnych gospodarkach, kulturach i sytuacjach geopolitycznych z pewnością jest zadaniem trudnym. Nie istnieje jeden poprawny sposób, w jaki można by dokonać pomiarów cyfryzacji, a każdy wskaźnik opracowany przez różne podmioty i składających się z innych komponentów, nie jest wolny od wad i możliwości do ulepszeń. Podobnie wskaźnik DDI, który z pewnością nie jest idealny i posiada wiele wad. Z pewnością jednak, badania w zakresie pomiarów poziomu cyfryzacji na świecie, a także nowe sposoby jego wyznaczania są potrzebne i powinny być kontynuowane, jako że istnieje zapotrzebowanie na tego rodzaju informacje, jak i z uwagi na to, że zagadnienie to

jest zbyt złożone aby jedna instytucja była w stanie objąć wszelkie dostępne zmienne mogące mieć wpływ na poziom cyfryzacji. W procesie opracowywania wskaźnika cyfryzacji ważne jest przede wszystkim uwzględnienie jak największej liczby zmiennych, przemyślana struktura wskaźnika, jak i nieskrępowany dostęp do danych. W sytuacji gdy duże koncerny i korporacje są w stanie wydać mnóstwo pieniędzy na pozyskanie danych poprzez przeprowadzenie ankiet bądź dane zebrane z innych podmiotów gospodarczych lub państwowych, przeciętny absolwent uczelni nie jest w stanie pozyskać dostępu do tak kluczowych na potrzeby analizy danych. Z pomocą przychodzi instytucja Banku Światowego, która udostępnia niektóre zebrane dane na temat cyfryzacji. Jednak zagłębiając się w temat analizy wskaźników cyfryzacji, po pewnym czasie można zauważyć, iż liczba tych danych, ich aktualność, a także zakres tematyki, jest wciąż niesatysfakcjonująca, jako że bardzo trudno jest opracować wskaźnik cyfryzacji, który byłby rzetelny i trafny co do faktycznej sytuacji w danych miejscach na świecie na podstawie tych dostępnych publicznie danych. Z pewnością w tej sytuacji pomogło by uchylenie rąbka tajemnicy odnośnie szczegółów konstrukcji wskaźnika cyfryzacji, a także danych jakie udało się zebrać dla poszczególnych elementów wchodzących w ich skład. Tym niemniej, warto pamiętać o tym, iż tematyka cyfryzacji, jej pomiarów

oraz sposobów wyznaczania wciąż się rozwija – podobnie jak dostęp do informacji, który z roku na rok staje się coraz bardziej powszechny.

Literatura

- [1] Słownik języka polskiego (sjp) – „Dygitalizacja”: <https://sjp.pwn.pl/sjp/dygitalizacja:2555621.html> [23.10.2019]
- [2] Wikipedia: <https://pl.wikipedia.org/wiki> [23.10.2019]
- [3] Transformacja zmiennych diagnostycznych: http://webcache.googleusercontent.com/search?q=cache:dGhpC70RX9gJ:kolegia.sgh.waw.pl/pl/KAE/struktura/ISiD/struktura/ZSS/zaklad/sklad/Strony/oferta_dydaktyczna_panek.aspx+&cd=1&hl=pl&ct=clnk&gl=pl [23.10.2019]
- [4] World Development Report 2016: Digital Dividends: <http://www.worldbank.org/en/publication/wdr2016> [23.10.2019]
- [5] Gallup Poll: <https://www.gallup.com/analytics/232838/world-poll.aspx> [23.10.2019]
- [6] Bank Światowy: Indeks Adopcji Cyfrowej metodologia (2016): <http://pubdocs.worldbank.org/en/587221475074960682/WDR16-BP-DAI-methodology.pdf> [3.10.2019]

Modern technologies for creating graphical user interface in web applications

Nowoczesne technologie tworzenia graficznego interfejsu użytkownika w aplikacjach internetowych

Mateusz Kaproń*, Beata Pańczyk

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

The goal of the article is to compare the three most commonly used tools for creating a graphical user interface in web applications. The analysis was carried out for the currently most popular tools: Angular, React and Vue. Test applications with identical user interface, implemented in three technologies, were used for the research. The article compares, above all, the performance related to page loading time and memory usage..

Keywords: Angular; React; Vue; web application

Streszczenie

Celem artykułu jest porównanie trzech najczęściej stosowanych narzędzi do tworzenia graficznego interfejsu użytkownika w aplikacjach internetowych. Analiza została przeprowadzona dla najbardziej popularnych obecnie narzędzi: Angular, React i Vue. Do badań wykorzystano aplikacje testowe z identycznym interfejsem użytkownika, zaimplementowane w trzech technologiach. W artykule porównano przede wszystkim wydajność związaną z czasem ładowania stron oraz zajętość pamięci.

Słowa kluczowe: Angular; React; Vue; aplikacje internetowe

*Corresponding author

Email address: mateusz.kapron@pollub.edu.pl (M. Kaproń)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Liczba narzędzi do tworzenia graficznego interfejsu użytkownika i bibliotek opartych na języku JavaScript wciąż rośnie. Nauka ich wszystkich nie jest praktycznie możliwa. W niniejszym artykule rozpatrywane będą trzy najpopularniejsze narzędzia, wskazane jako najczęściej stosowane w 2019 roku dla aplikacji webowych typu SPA (ang. Single Page Application) [1, 2]. Architektura typu SPA [3] pozwala na dynamiczne nadpisywanie treści, co przekłada się na szybkość działania i wydajność aplikacji.

Angular [4, 5, 6] to framework i platforma do tworzenia aplikacji typu SPA, napisany w języku TypeScript. Jest wspierany oraz rozwijany przez Google. Najnowszą i najczęściej używaną wersją jest Angular v8, wydany w maju 2019r. Wśród firm korzystających z tego narzędzia znajduje się: Google, Microsoft, JPMorgan, Time Warner Cable, McDonald's, UPS, Apple i wiele innych.

React [7, 8, 9] to biblioteka JavaScript stworzona przez deweloperów Facebooka. Biblioteka jest opisywana, jako „narzędzie JavaScript do budowania interfejsów użytkownika”. React wykorzystuje wirtualny DOM (ang. Document Object Model) i przechowuje cały model dokumentu w pamięci. Po zmianie stanu wyszukuje różnice między wirtualnym a rzeczywistym DOM, i w miarę potrzeb y dokonuje aktualizacji.

Vue.js [10, 11, 12] mimi iż jest najmłodsza, progresywną biblioteką języka JavaScript, staje się powoli

silnym konkurentem wobec dwóch poprzednich technologii.

2. Cel i tezy badawcze

Celem badań jest porównanie najbardziej popularnych technologii tworzenia interfejsu graficznego aplikacji internetowych na przykładzie Angular, React i Vue. Podstawowym kryterium porównania jest wydajność związana z czasem ładowania strony oraz obciążenie pamięci w przeglądarce. Analizowane są również inne cechy badanych narzędzi takie jak: wybrane metryki kodu, dostępność dokumentacji, trudność wytwarzania oprogramowania dla początkującego programisty aplikacji typu SPA (ang. Single Page Application).

Postawiono następujące tezy badawcze:

Wydajność interfejsu graficznego tworzonego w Vue jest porównywalna z analogicznym GUI dla React i Angular.

Angular jest najbardziej skomplikowanym narzędziem dla początkującego programisty aplikacji typu SPA w porównaniu do React i Vue.

Vue jest najbardziej przyjazną technologią dla początkującego programisty aplikacji typu SPA w porównaniu do React i Angular.

3. Metoda badań

Analizę wydajności czasowej i obciążenia pamięci wykonano za pomocą aplikacji testowej, dla której zaimplementowano trzy różne technologie generowania widoków. Wszystkie aplikacje korzystają z danych

udostępnianych przez trzy zewnętrzne API (ang. Application programming interface).

3.1. Aplikacja testowa

Aplikacja testowa generuje losowo pojedynczego użytkownika (lub wskazaną liczbę użytkowników w oparciu o zadane kryteria wyboru). W przypadku jednego użytkownika, na podstawie jego miejsca zamieszkania, aplikacja wyświetla dodatkowe dane o aktualnej pogodzie w tym miejscu. Z kolei data urodzenia losowego użytkownika jest daną wejściową do generowania dodatkowych ciekawostek, które wtedy miały miejsce. Rysunek 1 przedstawia przykładowy widok strony generującej dodatkowe informacje o losowym użytkowniku.



Rysunek 1: Interfejs graficzny aplikacji testowej

W przypadku wyboru większej liczby użytkowników (do 500), aplikacja umożliwi podanie kryterium wyboru płci i lokalizacji użytkowników. Taka funkcjonalność aplikacji pozwoliła zbadać wydajność interfejsu graficznego w oparciu o przygotowane scenariusze testowe.

3.2. Kryteria i scenariusze badawcze

Wybrane technologie zbadano według następujących kryteriów:

- czas ładowania strony w przeglądarce internetowej z czyszczeniem i bez czyszczenia pamięci przeglądarki;
- wykorzystywanie pamięci RAM;
- inne cechy technologii, takie jak dostępność dokumentacji i przykładów, wybrane metryki kodu.

W celu przeprowadzenia badań wydajności opracowano 6 scenariuszy testowych:

- S1 - generowanie pojedynczego użytkownika wraz z dodatkową informacją o pogodzie, walucie oraz losową historią wybraną na podstawie daty urodzenia;
 - S2 - generowanie jednego użytkownika,
 - S3 - generowanie 500 użytkowników;
 - S4 - generowanie 500 mężczyzn;
 - S5 - generowanie 500 mężczyzn z Australii;
 - S6 - generowanie 500 kobiet z Brazylii.
- Dla każdego scenariusza zostały wyznaczone:
- czasy ładowania strony,
 - obciążenie pamięci.

4. Wyniki badań

Badania wykonano na komputerze Lenovo y50 z systemem operacyjnym Windows 10, o następujących podzespołach:

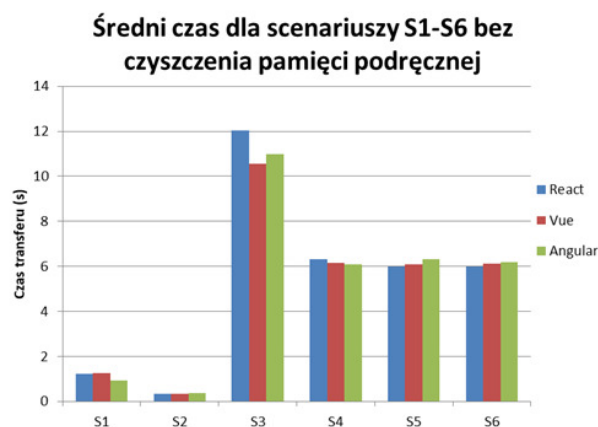
- Procesor Intel(R) Core(TM) i7-4720HQ CPU @ 2.60GHz,
- Dysk twardy SSD o pojemności 440GB,
- Pamięć RAM 16GB DDR3 1600MHz,
- Szybkość łącza internetowego dla pobierania wynosi 100Mb/s oraz wysyłania 12Mb/s.

Czas ładowania stron oraz obciążenie pamięci dla wszystkich aplikacji testowych określono za pomocą narzędzia deweloperskiego wbudowanego w przeglądarkę Firefox Developer Edition (wersja 71.0 beta). Wszystkie pomiary powtarzano po 5 razy.

4.1. Testy wydajnościowe

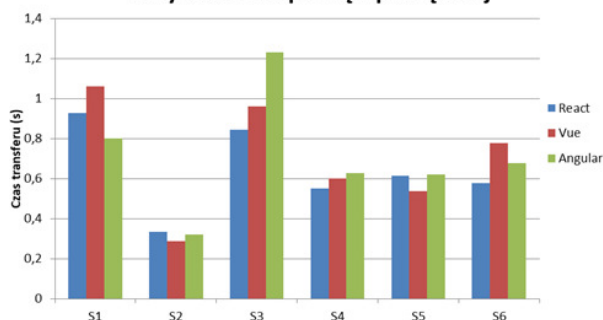
Porównanie średnich czasów ładowania strony dla wszystkich scenariuszy testowych przedstawiają rysunki 2 i 3.

Z kolei rysunki 4 i 5 prezentują średnie obciążenie pamięci dla wszystkich scenariuszy i technologii.



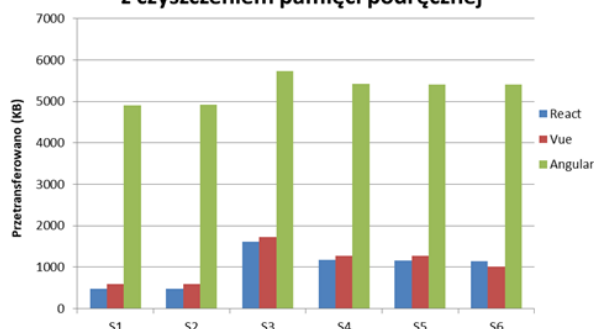
Rysunek 2: Średni czas dla scenariuszy S1-S6, bez czyszczenia pamięci podręcznej przeglądarki

Średni czas dla scenariuszy S1-S6 z czyszczeniem pamięci podręcznej



Rysunek 3: Czasy dla scenariuszy S1-S6 z czyszczeniem pamięci podręcznej przeglądarki

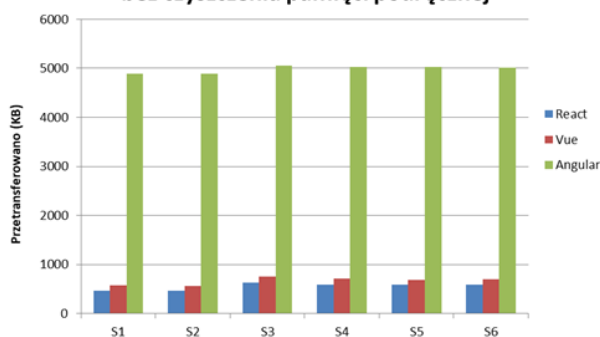
Przetrasferowane pliki dla scenariusza S1-S6 z czyszczeniem pamięci podręcznej



Rysunek 4: Przetrasferowane pliki dla scenariusza S1-S6 z czyszczeniem pamięci podręcznej

Analizując szybkość ładowania strony bez czyszczenia pamięci cache w przeglądarce (rys. 2), dla aplikacji typu SPA w przypadku scenariuszy S2, S5 i S6 najszybszy okazał się React, dla scenariuszy S1 i S4 jest to Angular, a dla S3 – Vue. W przypadku, gdy pamięć przeglądarki jest czyszczona (rys. 3) – najwydajniejszym dla S3, S4 i S6 jest React, dla S1 – Angular, a Vue dla S2 i S5. Podsumowując wyniki badania wydajności czasowej można stwierdzić, że żadna z technologii nie jest znacząco lepsza w stosunku do pozostałych.

Przetrasferowane pliki dla scenariusza S1-S6 bez czyszczenia pamięci podręcznej



Rysunek 5: Przetrasferowane pliki dla scenariusza S1-S6 bez czyszczenia pamięci podręcznej

Analizując liczbę transferowanych danych [KB] bez czyszczenia pamięci cache w przeglądarce (rys. 5), w przypadku wszystkich scenariuszy najbardziej efek-

tywny jest React, jednak Vue zużywa niewiele więcej pamięci. W przypadku, gdy pamięć przeglądarki jest czyszczona (rys. 4) – wyniki są bardzo zbliżone do tych, gdy cache przeglądarki nie był czyszczony. W przypadku tego kryterium – Angular wypada dużo gorzej od pozostałych dwóch technologii.

4.2. Dostępność dokumentacji

Każde z omawianych narzędzi posiada rozbudowaną dokumentację, która jest stale rozwijana i aktualizowana. Zarówno Angular [6], React [9], jak i Vue [12] posiadają bardzo obszerne dokumentacje, a ich twórcy starają się w łatwy sposób przedstawić sposoby ich wykorzystania. Dla każdej technologii, poza darmową dokumentacją, kursem, tutorialiem, istnieje również możliwość nauki poprzez inne płatne kursy online. Warto zwrócić uwagę, że większość kursów dostępnych w sieci jest w języku angielskim. W pracy do porównania dostępności kursów skorzystano z danych portalu [13]. Liczba kursów dla Angular wynosi 599, dla React - 570 a dla Vue tylko 216. Wynika to z faktu, że jest to technologia najnowsza, która dopiero zdobywa popularność wśród programistów front-end.

4.3. Wybrane metryki kodu

Korzystając z aplikacji testowych, w tabeli 1 przedstawiono porównanie metryk kodu i innych mierzalnych wartości istotnych dla omawianych narzędzi.

Tabela 1: Metryki kodu i inne wartości mierzalne dla aplikacji testowych

Narzędzie	React	Vue	Angular
Zajmowana pamięć [MB]	154	156	273
Liczba plików	29 325	18 128	31 011
Liczba folderów	4188	3023	3909
Liczba komponentów	6	5	3
Liczba wierszy kodu	397	337	263

4.4. Wybrane metryki kodu

Subiektywna ocena niemierzalnych cech technologii widoków została zestawiona w tabeli 2. Każde kryterium oceniono w skali od 1 do 10, gdzie większa liczba oznacza lepszą ocenę.

Tabela 2: Oceny różnych właściwości badanych technologii (zielonym kolorem są zaznaczone najwyższe oceny, czerwonym – najniższe)

Narzędzie	React	Vue	Angular
Metryki kodu	8	8	7
Dokumentacja	9	9	9
Wydajność	8	7	7
Próg wejścia	6	6	2
Czas implementacji	6	5	3
Sumaryczna ocena	37	35	26

React i Vue uzyskały zbliżone wyniki w ostatecznej ocenie punktowej. Angular uzyskał mniej punktów ze względu na wysoki próg wejścia oraz czas implementa-

cji aplikacji testowej. Porównując metryki kodu, aplikacja w Angular ma dużo więcej plików i zajmuje dużo więcej pamięci w porównaniu do React i Vue. Dokumentacja dostępna dla każdego narzędzia jest na podobnym poziomie. Z tabeli 2 wynika, że pod względem metryk kodu, wydajności, progu wejścia, czasu implementacji React i Vue są bardzo zbliżone w ocenie.

5. Wnioski

Rozważane w niniejszym artykule technologie są najczęściej stosowanymi narzędziami do tworzenia aplikacji webowych typu SPA. Podsumowując rezultaty badań przedstawione w rozdziale 4, oraz mając na uwadze doświadczenie własne autorów, można sformułować następujące wnioski:

- Angular jest dobrym narzędziem dla doświadczonych programistów. Nauka tego frameworka jest trudna i wymaga dużej wiedzy wstępnej. Struktura aplikacji w Angular nadaje się bardzo dobrze w przypadku rozbudowanych serwisów.
- Struktura aplikacji w React i Vue jest lepsza w przypadku prostych i mniej rozbudowanych serwisów.
- Vue jest technologią, której można się nauczyć najszybciej, dlatego też warto ją polecić szczególnie początkującym programistom front-end.
- czasy ładowania stron dla wszystkich technologii z czyszczeniem i bez czyszczenia pamięci podręcznej nie wykazują istotnych różnic (Rys. 2 i 3), ale stosunkowo najbardziej wydajny jest React;
- obciążenie pamięci jest największe w przypadku Angular. React i Vue – wykazują porównywalne wielkości (Rys. 4 i 5).

Powyższe wnioski dają podstawę stwierdzeniu, że postawione na wstępie tezy badań zostały potwierdzone.

Literatura

- [1] Najpopularniejsze technologie JavaScript, <https://2018.stateofjs.com>, [22.05.2019]
- [2] Serwis społecznościowy dotyczący tematyki wytwarzania oprogramowania <https://stackoverflow.com>, [22.10.2019].
- [3] S. Michael: Single Page Web Applications Programowanie aplikacji internetowych z JavaScript. HELION, Gliwice 2015.
- [4] F. Yakov.: Angular2 programowanie z użyciem języka TypeScript. HELION, Gliwice 2018.
- [5] G. Kunz., Angular2 tworzenie interaktywnych aplikacji internetowych. HELION, Gliwice 2017.
- [6] Dokumentacja Angular, <https://angular.io/docs>, [26.10.2019].
- [7] Freeman A., Pro React 16, Apress, 2019.
- [8] A. Cassio de Sousa, React dla zaawansowanych. Helion, Gliwice 2017.
- [9] Dokumentacja React, <https://reactjs.org/docs/getting-started.html>, [28.10.2019].
- [10] O. Fillipova, Vue.js 2 Tworzenie reaktywnych aplikacji WWW. Helion, Gliwice 2017.
- [11] A. Freeman, Pro Vue. js 2. Apress, 2018.
- [12] Dokumentacja Vue, <https://vuejs.org/v2/guide/>, [26.10.2019]
- [13] Kursy online, <https://www.udemy.com/>, [23.10.2019]

Verification of the precision of GSM and GPS signals available on mobile devices

Weryfikacja dokładności sygnałów GSM i GPS dostępnych na urządzeniach mobilnych

Mateusz Właż*, Grzegorz Kozieł*

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

The research work presents the study of the quality of telecommunications and satellite signals. For this purpose, a proprietary application for mobile devices has been designed, which allowed the collection data. The tests were performed on several models of cell phones and for different operators. The test checked the impact of such things as weather conditions, device architecture, device age, type of terrain. Based on the results, heat maps were created to visualize the research.

Keywords: quality of signal; mobile devices; telecommunications signal; satellite signal

Streszczenie

Praca badawcza prezentuje badanie jakości sygnału telekomunikacyjnego oraz satelitarnego. W tym celu zaprojektowano autorską aplikację na urządzenia mobilne, która umożliwiła gromadzenie danych. Badania wykonano na kilku modelach telefonów komórkowych oraz dla różnych operatorów. W badaniu sprawdzono wpływ na sygnał takich czynników jak: warunki atmosferyczne, architektura urządzenia, wiek urządzenia, rodzaj terenu. Na podstawie wyników stworzono mapy cieplne w celu wizualizacji badań.

Słowa kluczowe: jakość sygnału; urządzenia mobilne; sygnał telekomunikacyjny; sygnał satelitarny

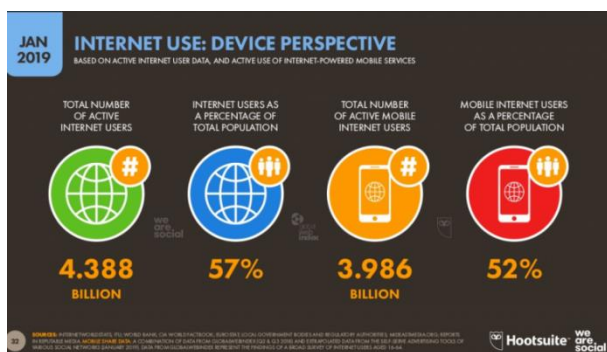
*Corresponding author

Email addresses: mateusz.wlaz@pollub.edu.pl (M. Właż), g.kozieł@pollub.pl (G. Kozieł)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Co roku obserwowany jest znaczący spadek liczby telefonów stacjonarnych [1,2,3]. Są one systematycznie wypierane przez telefony komórkowe. Obecnie liczba kart sim obsługiwanych przez polskich operatorów telekomunikacyjnych przekracza liczbę mieszkańców Polski [4, 5]. Świadczy to o dużej intensywności użytkowania urządzeń mobilnych oraz o znaczącym zapotrzebowaniu na tego typu usługi. Ponad 92% Polaków wykorzystuje telefony komórkowe [6]. Są one najczęściej używane do prowadzenia rozmów i wysyłania wiadomości [6].

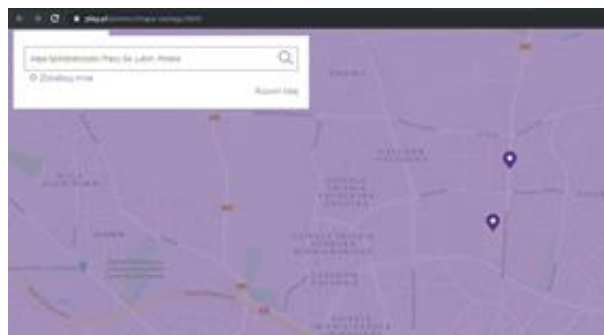


Rysunek 1: Dane o użytkownikach Internetu [7].

Nie można jednak pominąć innych zastosowań, wśród których dominują usługi dostępne w Internecie.

52% populacji świata uzyskuje dostęp do Internetu za pomocą smartfonów (Rys.1) [7]. 42% korzysta na smartfonach z mediów społecznościowych.

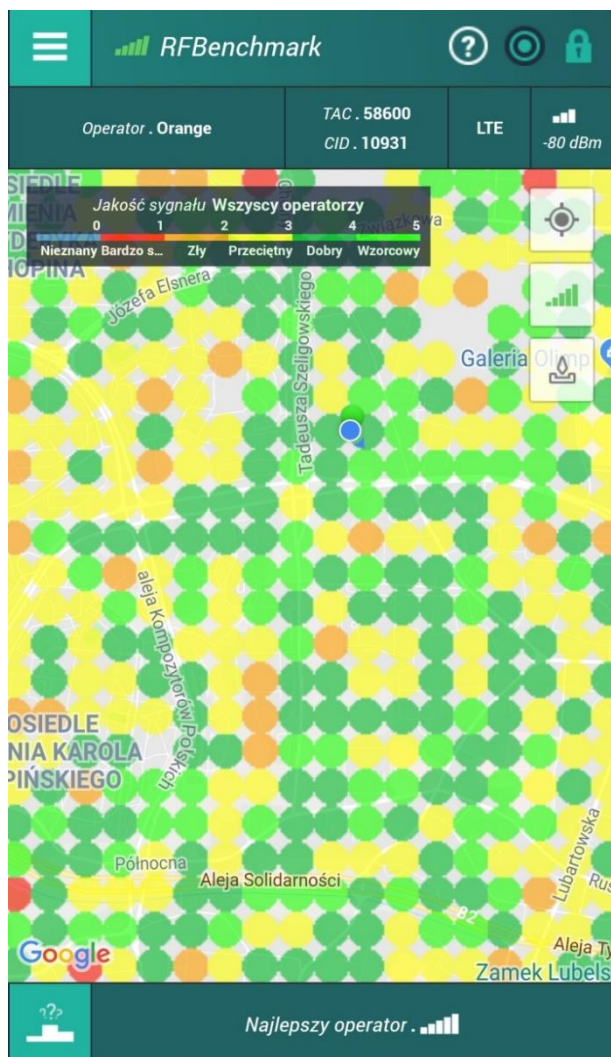
Oczywiście zastosowania urządzeń mobilnych są o wiele szersze, o czym świadczyć może chociażby ogromna liczba różnorodnych aplikacji dostępnych dla urządzeń mobilnych [8]. Jednak poprawne działanie wielu z nich uwarunkowane jest dobrą jakością połączenia z Internetem oraz precyzją lokalizacji urządzenia. Dane operatorów dostępne na ich stronach [9] prezentują poglądowo zasięg sygnału i dostępność poszczególnych usług. Nie wskazują jednak siły sygnału ani też jakości połączenia (Rys. 2).



Rysunek 2: Mapa zasięgu sygnału LTE operatora Play [9].

Te jednak mogą się znacząco zmieniać w zależności od obszaru czy typu terenu. Zależność tą można

z powodzeniem zaobserwować na mapach prezentowanych przez aplikację RFBenchmark (Rys. 3).



Rysunek 3.: Jakość sygnału GSM na wybranym terenie.

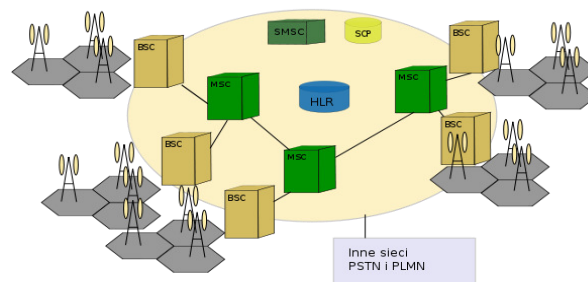
Ze względu na zaobserwowane problemy ze słabą jakością sygnału GSM i GPS autorzy postanowili przeprowadzić badania pozwalające na weryfikację jakości wymienionych sygnałów na wybranych obszarach. W tym celu stworzona została autorska aplikacja pracująca pod kontrolą systemu operacyjnego Android. Posłużyła ona do zmierzenia dostępnego poziomu badanych sygnałów i zapisania wyników badań w bazie danych. Uzyskane wyniki zostały następnie przeanalizowane i zaprezentowane w niniejszym artykule.

1.1. Infrastruktura sieci telefonii komórkowej

Systemy, które są obecnie użytkowane bazują na technologiach drugiej, trzeciej oraz czwartej generacji. Każda z podanych generacji różni się między sobą wynikami dotyczącymi ich specyfikacji wynikającej z budowy takiej technologii (Rys. 4).

Użytkownik sieci znajdujący się w elemencie zwanym komórką (szare sześciokąty) kontaktuje się z innym użytkownikiem poprzez BS (ang. Base Station), czyli maszt ujęte na komórkach. Stacje bazowe (BS) zarządzane są poprzez BSC (ang. Base Station Control-

ler), które przekazują dane dalej do sieci szkieletowej komunikując się z MSC (ang. Mobile Switching Centre). MSC są to cyfrowe centrale telefoniczne, które odpowiadają za zestawianie ze sobą połączeń. Dodatkowym elementem mieszczącym się w sieci szkieletowej jest HLR (ang. Home Location Register), czyli baza danych, która przechowuje dane o użytkownikach i zezwala lub odrzuca możliwość korzystania z danych zasobów radiowych.



Rysunek 4: Architektura sieci dla trzeciej generacji [10].

1.2. Sygnały GSM i GPS

Sygnał jest to zmienność wielkości fizycznej w wielkości zmieniającej się (czyli najczęściej czasie). Dla telekomunikacji sygnały są to nośniki informacji przekazywanych na dowolne odległości.

GPS (ang. Global Positioning System) jest to system nawigacyjny służący lokalizacji na lądzie, morzu czy w powietrzu. Pozwala on na otrzymanie takich danych jak położenie, aktualna prędkość z jaką obiekt się porusza. W skrócie system nazywany jest jako GPS, który został stworzony przez Stany Zjednoczone i po raz pierwszy w pełni użyto go w 1978 roku. Początkowo system ten miał służyć tylko dla celów militarnych, chociażby do naprowadzania pocisków na konkretny cel jednak szybko został on wprowadzony do codziennego użytku cywilnego.

System stworzono na podstawie trzech segmentów:

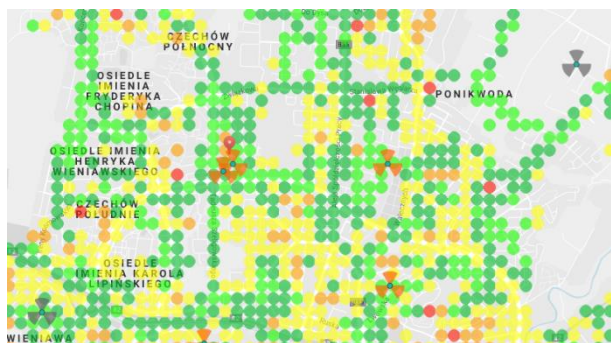
- **kosmiczny** (satelity krążące wokół Ziemi)
- **kontroli** (monitorowanie stacji kontrolnych znajdujących się na równiku)
- **użytkownika** (wykorzystanie technologii GPS w codziennym życiu).

GSM (ang. Global System for Mobile Communications) jest to standard telefonii komórkowej należącej do drugiej generacji sieci komórkowych (2G). Po raz pierwszy dla tej generacji użyto technologii cyfrowej (1G oparto na technologii analogowej). W Polsce 2G pojawiło się w 1996 roku, umożliwiając nie tylko wykonywanie połączeń ale również możliwość wysyłania SMS, a nawet na łączność z Internetem poprzez urządzenie mobilne.

1.3. Metodyka badawcza

Badanie sygnałów polegało na wykorzystaniu autorskiej aplikacji na urządzenia mobilne w celu zgromadzenia potrzebnych danych. Wykonano dwa rodzaje badań: dla sygnałów telekomunikacyjnych oraz satelitarnych. Badania zostały powtórzone kilkakrotnie na

różnych modelach urządzeń mobilnych (Huawei P20 Lite, Huawei P9 Lite, Huawei P10 Lite, Samsung Galaxy A40, Samsung Galaxy Tab 4).



Rysunek 5: Mapa zasięgu aplikacji RFBenchmark.

Porównując mapę zasięgu operatora oraz mapę zasięgu otrzymaną przy użyciu aplikacji RFBenchmark, można zauważyć różnicę w wynikach. Operator nie określa dokładności czy mocy sygnału, który jest oferowany w danej lokalizacji. Przedstawia on tylko fakt występowania zasięgu w danej lokalizacji (Rys. 2). Mapa zasięgu pozyskana z aplikacji RFBenchmark nie odzwierciedla jednak pokrycia zasięgu przedstawionego przez operatora. Wybierając opcję zasięgu tylko dla sygnału LTE, występują miejsca, w których nie ma zasięgu (Rys. 5). W celu weryfikacji poprawności otrzymanych danych stworzono autorską aplikację. Dla sygnału satelitarnego wartości, które pobrano są to:

- **długość geograficzna** – współrzędna opisująca kąt pomiędzy południkiem 0, a aktualnym południkiem
- **szerokość geograficzna** – tak jak długość geograficzna jest to współrzędna opisująca kąt pomiędzy płaszczyzną równika, a obecnym równoleżnikiem
- **dokładność** – parametr wyrażony w badaniu w metrach, określa przybliżoną dokładność urządzenia
- **prędkość** – prędkość poruszania się podczas badania, wyrażona w metrach na sekundę
- **wysokość** – wysokość nad poziom morza, mierzona w metrach
- Sygnał telekomunikacyjny określony został poprzez:
- **siła odbieranego sygnału** – wskaźnik siły odbieranego sygnału, mierzona w wartościach ujemnych dBm
- **Level** – abstrakcyjna jakość sygnału opisywana wartością od 0 do 4
- **ASU (ang. Arbitrary Strength Unit)** – jest to wskaźnik, który jest liczbą całkowitą, która jest proporcjonalna do siły odbieranego sygnału

Tabela 1. Tabela opisująca jakość sygnału dla 2G [11].

RSSI	Jakość
> -52 dBm	Doskonała
-53 dBm do -109 dBm	Dobra
-110 dBm do -112 dBm	Dostateczna
<-113 dBm	Słaba
<=-110 dBm	Brak sygnału

Jakość sygnału została określona w standardach dotyczących sygnału GSM (Tabela 1). Każda technologia posiada stworzoną specyfikację, która określa dopuszczalne wartości siły sygnału oraz zasady funkcjonowania tych technologii. Im bliżej 0 dBm, tym lepszy jest sygnał.

Dla 2G wartość ASU określana jest jako RSSI (ang. Received Signal Strength Indicator) i wyliczana na podstawie poniższego wzoru:

$$dBm = 2 \times ASU - 113$$

Badanie zostało podzielone na dodatkowe dwa rodzaje: mobilne (zostały przemierzone wybrane ulice Lublina wraz z kilkoma urządzeniami) oraz stacjonarne (pomiar wykonywany podczas znajdowania się urządzenia w tej samej lokalizacji).

1.4. Stanowisko badawcze

Badania wykonane zostały na urządzeniach mobilnych, czyli Huawei P20 Lite, Samsung Galaxy A40, Huawei P10 Lite, Huawei P9 Lite, Samsung Galaxy Tab 4. Pierwsze trzy urządzenia należą do grupy nowszych urządzeń (2017 i nowsze). Na wszystkich urządzeniach została zainstalowana autorska aplikacja, która służyła do gromadzenia danych dotyczących sygnałów.

Wykonano dwa rodzaje badań: stacjonarne, w którym urządzenia znajdowały się w jednej lokalizacji przy oknie i nie były użytkowane podczas badań. Drugie mobilne, czyli przejazd wyznaczoną trasą po Lublinie. Wszystkie badania zostały powtórzone kilkakrotnie.

2. Eksperyment

Celem przeprowadzenia badań naukowych było porównanie i zweryfikowanie faktycznego stanu jakości sygnałów telekomunikacyjnych oraz satelitarnych. W tym celu wykonano kilka badań w celu uzyskania wyników, które można przeanalizować oraz zaprezentować w formie wykresów lub map cieplnych. Badania zostały powtórzone kilkakrotnie, aby można było mówić o powtarzalności wyników.

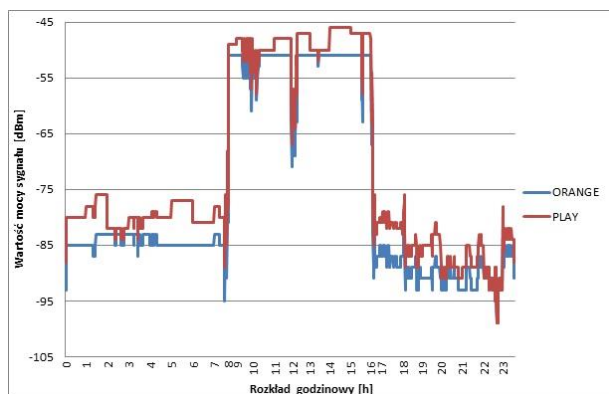
Poniższe badania przedstawiają częściowe wyniki prac nad sprawdzeniem jakości sygnałów GSM oraz GPS.

2.1. Siła sygnału GSM w stałej lokalizacji

Celem tego badania było sprawdzenia jakości sygnału dla dwóch operatorów telekomunikacyjnych: Play oraz Orange. Wyniki uwzględniają warunki pogodowe. Badanie wykonano dla sieci drugiej generacji.

Wyniki badania (Rys. 6) są to wyniki z jednej doby (wyniki zostały przeskalowane w taki sposób, że punkt 0 to godzina 00:00), badanie zostało przeprowadzone w bloku znajdującym się na osiedlu. Telefony (2 jednokomórkowe modele: Samsung Galaxy A40) podczas pomiarów znajdowały się w jednej lokalizacji. Przedstawione badanie zostało powtórzone kilkakrotnie dla jak najlepszych wyników.

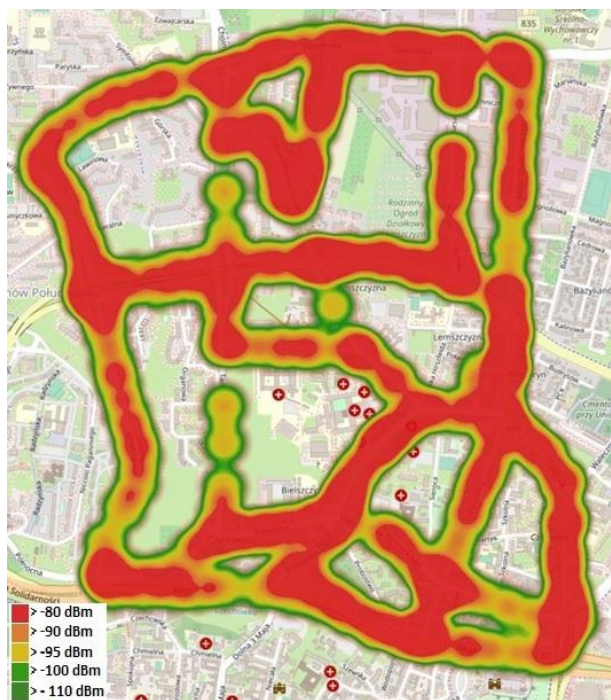
Na rysunku 6 można wyraźnie zauważyć wzrost jakości sygnału w godzinach 7-17. Badanie wykazuje, że jakość sygnału na terenie pomiaru prawie przez cały okres jego trwania była lepsza dla sieci Play.



Rysunek 6: Wykres mocy sygnału 2G dla operatorów.

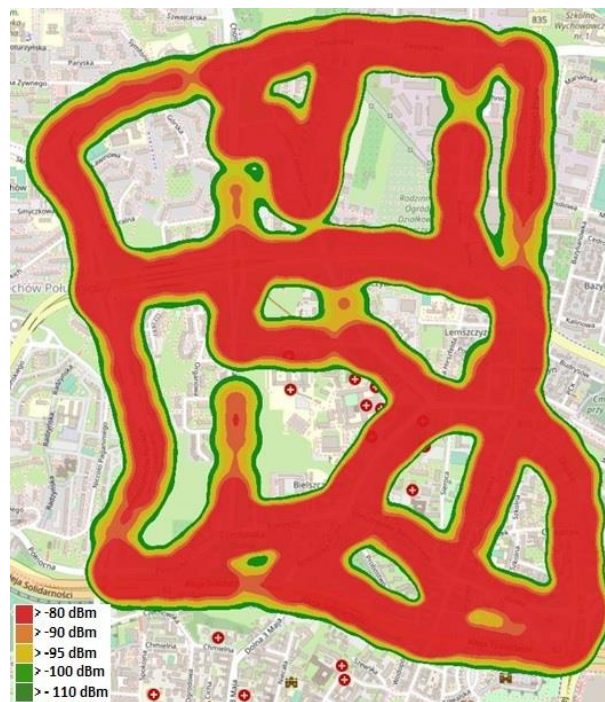
2.2. Rozkład siły sygnału GSM na wybranych drogach

Celem tego badania było przeprowadzanie badań podczas poruszania się pojazdem po określonych trasach miasta Lublin. W badaniu wykorzystano dwóch operatorów Play oraz Orange. W celu jak najdokładniejszego przeprowadzenia badań poniższe wykresy zostały przedstawione dla tego samego modelu telefonu (Samsung Galaxy A40) oraz prace badawcze wykonywano w tym samym czasie, aby wyeliminować możliwość wpływu czynników zewnętrznych na wyniki badania.



Rysunek 7: Mapa cieplna mocy sygnału GSM dla Orange.

Powyższa mapa cieplna (Rys. 7) przedstawia wynik badania dla poruszania się głównymi drogami badanej części Lublina. Pomiary zostały powtórzone pięciokrotnie, a wykres został stworzony na podstawie ich średnich wartości. Niniejsze badanie ukazuje wynik jakości sygnału sieci dla operatora Orange. Im bardziej czerwone zaznaczenie tym moc sygnału w danym miejscu była większa.



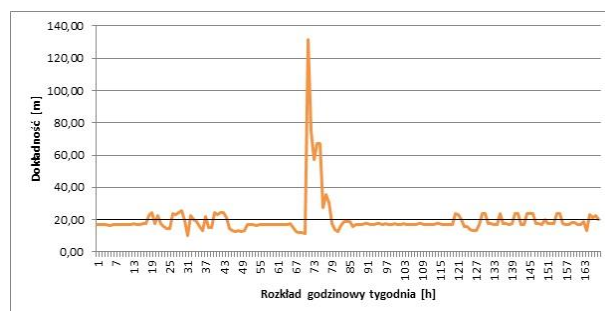
Rysunek 8: Mapa cieplna mocy sygnału GSM dla Play.

Mapa cieplna (Rys. 8) przedstawia jakość sygnału dla operatora komórkowego Play. Podobnie jak w powyższym przykładzie badanie przeprowadzono, przemierzając główne drogi Lublina. Czerwony kolor oznacza najlepszą wartość sygnału.

Na podstawie przedstawionych map cieplnych można określić, że operator sieci Play dla drugiej generacji posiada lepsze pokrycie terenu oraz lepszą jakość sygnału.

2.3. Jakość sygnału GPS w stałej lokalizacji

Celem tego badania było przeprowadzenie sprawdzenia jakości sygnału dla systemu nawigacji w warunkach stacjonarnych. Wyniki uwzględniają warunki pogodowe. Badanie wykonano dla systemu GPS. Powtórzone zostało na kilku modelach telefonów.

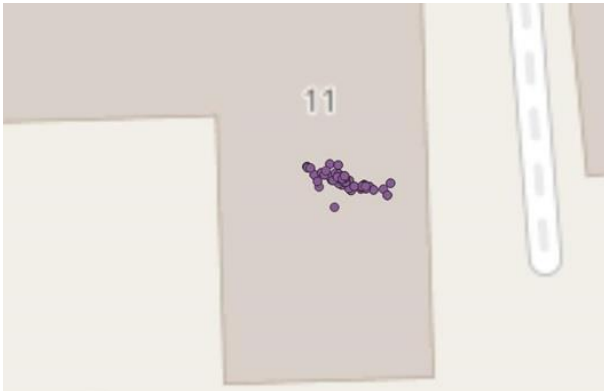


Rysunek 9: Wykres dokładności GPS względem rozkładu godzinowego dla tygodnia.

Prezentowany wynik badania (Rys. 9) prezentuje dokładność sygnału satelitarnego, podawaną w metrach w stosunku do rozkładu godzinowego w ciągu tygodnia. Wykres został przeskalowany w taki sposób, aby początkowa godzina pomiaru (punkt 0 na osi rozkładu godzinowego) był to dzień odpowiadający północy

z niedzieli na poniedziałek. Każde kolejne 24 godziny to jeden dzień tygodnia. Badanie zostało przeprowadzone w mieszkaniu i powtórzono na Samsung Galaxy A40, Huawei P20 Lite oraz Samsung Galaxy Tab 4.

Jak można zauważyć z powyższego wykresu dokładność sygnału w ciągu tygodnia oscyluje w granicach 20 metrów. Jednak od godziny 72 do 78 występuje bardzo duży skok dokładności, sięgający 130 metrów. W tym okresie w danej okolicy występowała bardzo silna burza, można więc na tej podstawie określić, że złe warunki atmosferyczne mają znaczący wpływ na jakość sygnału satelitarnego co za tym idzie również na jego dokładność. Podobny skok widoczny jest również na innych urządzeniach, na których w danym momencie były prowadzone pomiary.



Rysunek 10: Wizualizacja punktów pomiarowych dla Huawei P20 Lite.

Przedstawione pomiary (Rys. 10 oraz Rys. 11) przedstawiają sytuację dla pomiaru stacjonarnego i ukazują dokładność sygnału satelitarnego. Urządzenia podczas badania znajdowało się ciągle w mieszkaniu w okolicach okna. Badania zostały wykonywane w tym samym czasie, aby nie można było mówić o wpływie różnych czynników zewnętrznych. Wyniki badania powstał przez nałożenie punktów z długości oraz szerokości geograficznej na mapę.



Rysunek 11: Wizualizacja punktów pomiarowych dla Samsung Galaxy Tab 4.

Okres badania jest to tygodniowy czas pomiarów. Jak można zauważyć na powyższych mapach dla Huawei P20 Lite, które jest nowym urządzeniem, dokładność sygnału jest bardzo dobra. Sygnał niemal odbiera-

ny jest w jednym miejscu. Natomiast dla Samsung Galaxy Tab 4 punkty pomiarowe znacznie rozbiegają się po mapie. Można na tej podstawie sądzić, że na dokładność odbieranego sygnału ma znaczny wpływ urządzenie, z którego aktualnie korzysta użytkownik. Ciężko jednak jednoznacznie określić czy jest to wpływ architektury urządzenia czy możliwość rozstrojenia się anteny odbierającej sygnał satelitarny.

2.4. Wpływ rodzaju terenu na dokładność GPS

Badanie zostało wykonane w celu wykazania wpływu warunków terenu oraz otoczenia na dokładność sygnału satelitarnego. W tym celu przeprowadzone zostały następujące pomiary:

- pomiar pomiędzy wysokimi blokami, które są ustawione obustronnie wzdłuż jezdni
- pomiar pomiędzy domami jednorodzinnymi, ustawionymi obustronnie wzdłuż jezdni
- pomiar sygnału w lesie
- pomiar sygnału na pustym terenie, na którym prawie nie istnieją zabudowania oraz teren nie jest zalesiony (np. łąki)
- pomiar dokładności sygnału wraz ze wzrostem prędkości

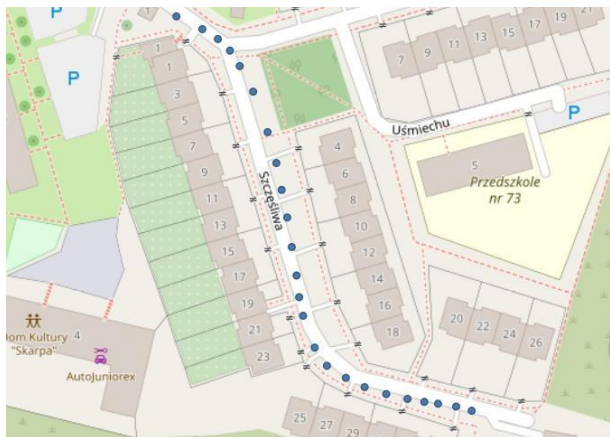


Rysunek 12: Mapa punktów pomiarowych pomiędzy blokami.

Praca badawcza przeprowadzona została na urządzeniu pomiarowym Huawei P10 Lite. Ruch odbywał się zawsze w jednym kierunku.

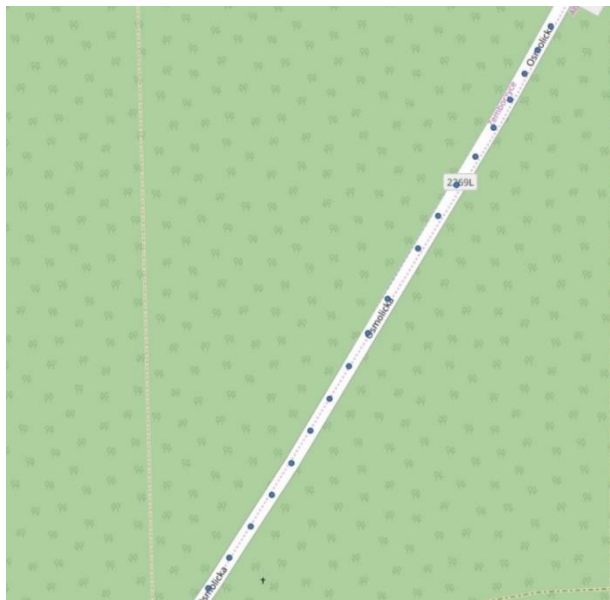
Mapa przedstawiona na rysunku 12 przedstawia rodzaj pomiaru wykonanego pomiędzy wysokimi blokami. Pomiar wykonano poruszając się w jednym kierunku ulicą Niccolo Paganiniego w Lublinie. Jak można zauważyć naniesione punkty pomiarowe (wyznaczone na podstawie długości oraz szerokości geograficznej) znacznie odbiegają od trasy pomiaru. Średnia dokładność pomiarów wyniosła 6,83 metra.

Kolejnym etapem badania było sprawdzenie warunków działania systemu pomiędzy niższymi budynkami. W tym wypadku została wybrana osiedlowa uliczka z domami jednorodzinnymi. Pomiar wykonano jadąc w jedną stronę ulicą Szczęśliwą w Lublinie.



Rysunek 13: Mapa punktów pomiarowych pomiędzy domami jednorodzinnymi.

Jak można zauważyć na rysunku 13 wyniki pomiarów ponownie odbiegają od faktycznej trasy, która została przebyta. Dla tej pracy badawczej średni wynik dokładności sygnału GPS wyniósł 9,19 metra.



Rysunek 14: Mapa punktów pomiarowych w lesie.

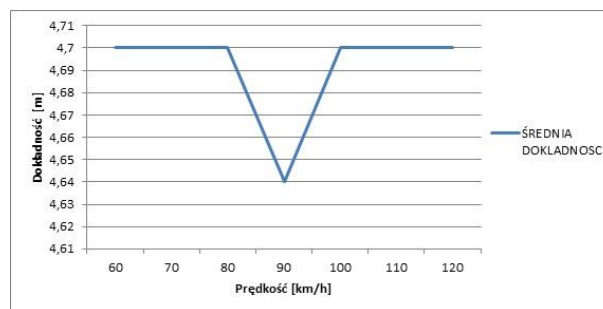
Następnym etapem było sprawdzenie jakości sygnału w wysokim lesie. Podczas wykonywania pomiarów las znajdował się po obu stronach drogi. Pomiar wykonywano poruszając się w jednym kierunku ulicą Osmoličką. Jak można zauważyć na rysunku 14, wyniki pomiaru wykazują całkiem dobre odwzorowanie trasy, którą przebyto podczas badania (nie odbiega ona znacznie od prawego pasa ruchu). Średni wynik dokładności sygnału dla tego pomiaru wyniósł 4,85 metra.

Przedostatnią pracą badawczą, która została wykonana był test jakości sygnału na terenie, na którym prawie nic się nie znajduje. Trasę przebyto w jednym kierunku, jej odwzorowanie na mapie zostało zaprezentowane na rysunku 15. Badanie wykazało, że średnia dokładność dla takiego rodzaju terenu wyniosła 4,71 metra.



Rysunek 15: Mapa punktów pomiarowych na pustym terenie

Ostatnią próbą badawczą, która została wykonana to sprawdzenie wpływu prędkości na jakość sygnału GPS. W tym celu wykonano pomiar na trasie szybkiego ruchu. Pomiary wykonane dla prędkości od 60 do 120 kilometrów na godzinę z interwałami co 10 kilometrów na godzinę. Dla każdego pomiaru wykonano 20 prób co 50 metrów na podstawie, których wyliczono średnią dokładność sygnału satelitarnego. Badanie wykazało niezmienną jakość sygnału GPS w zależności od prędkości poruszania się (Rys. 16), jedynie przy prędkości 90 kilometrów na godzinę wystąpił delikatny spadek średniej dokładności. Jest to jednak bardzo niewielka różnica (0,06 metra).



Rysunek 16: Wykres prędkości do dokładności sygnału GPS.

3. Wnioski

W ramach tej pracy wykonano badania dotyczące pomiaru jakości sygnałów telekomunikacyjnych oraz dodatkowo wykonano testy dokładności sygnału satelitarnego. W tym celu została stworzona aplikacja służąca do gromadzenia wyżej wymienionych danych.

Operatorzy komórkowi nie informują użytkowników o faktycznych i dokładnych stanach sieci jakie mogą uzyskać tylko podają im teoretyczną wiedzę czy na danym terenie będą posiadali zasięg. Takie podejście operatorów skutkuje często znanymi sytuacjami, podpisanie umowy z operatorem, a następnie rozwiązywanie jej z powodu braku dostępu do sieci.

Operatorzy podczas podawania takich danych powinni brać pod uwagę warunki atmosferyczne oraz natężenie ruchu sieci, ponieważ jakość sygnału w tym momencie ulega pogorszeniu. Na wyniki zaprezentowane w powyższej pracy badawczej mogą mieć wpływ chociażby rodzaje aparatów, z których użytkownik korzysta. W aparatach telefonicznych występuje efekt rozstrajania się anten, dlatego na urządzeniach, które przepracowały więcej godzin wyniki odróżniają się od badań przeprowadzonych na nowszych urządzeniach. Średnia różnica wartości między nowszym, a starszym modelem to około 10 dBm.

Dla sygnałów telekomunikacyjnych można zauważyć poprawienie jakości sygnału w godzinach 7-17. W godzinach 17-7 średnia wartość siły sygnału wynosi około -110dBm, natomiast w godzinach 7-17 plasuje się ona na poziomie około -80 dBm. Może to być powodem użycia techniki regulacji mocą w momencie kiedy liczba użytkowników wzrasta [12]. W pobliżu miejsca pomiaru stacjonarnego znajduje się wiele budynków użyteczności publicznej (policja, szpital) oraz biura firm informatycznych. Wraz ze wzrostem użytkowników operator zwiększa moc wysyłanego sygnału co powoduje wzrost jakości sygnału.

Dla sygnału satelitarnego można zauważyć problemy z jakością sygnału w gorszych warunkach atmosferycznych. Powszechnie panuje informacja, że warunki atmosferyczne nie mają wpływu na jakość sygnału satelitarnego. Na podstawie badania można stwierdzić, że złe warunki nie mają jednoznacznie wpływu na brak sygnału, ponieważ nawet przy bardzo złej pogodzie sygnał wciąż był dostępny. Jednak warunki atmosferyczne wpływają na pogorszenie sygnału co przekłada się na pogorszenie dokładności. W budynku w trakcie burzy dokładność spadała nawet do 130 metrów, gdzie przy normalnych warunkach atmosferycznych wynosiła około 15 metrów.

Dla wyników dotyczących dokładności sygnału satelitarnego można dojść do wniosku, że otoczenie, w którym znajduje się użytkownik ma bardzo duże znaczenie na dokładność sygnału GPS. W przypadku terenów znajdujących się blisko zabudowań dokładność jest gorsza niż dla terenu, na którym nic się nie znajduje. Dla pomiarów pomiędzy blokami średnia dokładność wynosiła 6,83 metra, na osiedlu domków jednorodzinnych osiągnięto dokładność rzędu 9,19 metra. Dokładność w lesie wyniosła 4,85 metra, a na pustym terenie 4,71 metra.

W badaniu wykazano również, że jakość użytych urządzeń ma wpływ na wyniki pomiarowe sygnałów telekomunikacyjnych. Na starszych urządzeniach wyni-

ki pomiarów są gorsze. W badaniu nie wykazano jednak jakichkolwiek znaczących różnic w porównaniu do wersji oprogramowania urządzenia. Badanie wykonano na dwóch jednakowych modelach z różnymi wersjami oprogramowania oraz dla wyeliminowania błędów wykonano badanie powtórnie na tym samym modelu aparatu z różnymi systemami.

Literatura

- [1] <https://wiadomosci.dziennik.pl/wydarzenia/artykuly/79674.polacy-masowo-oddaja-telefon.html> [25.10.2019]
- [2] <https://komorkomania.pl/21420.komorki-uparcie-wypieraja-telefony-stacjonarne> [25.10.2019]
- [3] <https://www.money.pl/gielda/uke-liczba-abonentow-telefonii-stacjonarnej-spadla-do-4-1-mln-w-2018-r-6396870592546433a.html> [25.10.2019]
- [4] <https://www.parkiet.com/Analizy/305159869-Liczba-kart-SIM-W-I-kwartale-2019-roku-na-rynku-telefonii-komorkowej-w-Polsce-widac-stabilizacje-baz-klientow.html> [25.10.2019]
- [5] <https://mobirank.pl/2019/02/01/raport-mobile-i-digitalna-swiecie-w-2019-roku/> [25.10.2019]
- [6] <https://technologia.dziennik.pl/internet/artykuly/555355.sondaz-telefony-komrkowe-smartfony.html> [25.10.2019]
- [7] <https://www.mobiletrends.pl/raport-cyfrowy-wzrost-w-2019-roku/> [25.10.2019]
- [8] <https://play.google.com/store/apps?hl=pl> [25.10.2019]
- [9] <https://www.play.pl/pomoc/mapa-zasiegu.html> [29.10.2019]
- [10] <https://bankowymokiem.pl/posts/cdma-co-to-za-technologia-i-do-czego-mozna-ja-wykorzystac> [29.10.2019]
- [11] https://www.etsi.org/deliver/etsi_gts/07/0707/05.00.00_60/gsm_ts_0707v050000p.pdf [17.11.2019]
- [12] http://malopolska.btsearch.pl/PM/2_4.php [29.10.2019]

Corporate intranet platform – open and commercial solutions comparison

Korporacyjna platforma intranetowa - porównanie rozwiązania otwartego i komercyjnego

Konrad Czarny*

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

The article presents a comparison of CMS systems that can be used when creating an intranet platform for a corporate network. The primary goal is to indicate which of the technologies is the most suitable for a large international corporation. Two open (WordPress, Drupal) and commercial (SharePoint, Confluence) tools were included in the study.

Keywords: SharePoint; Drupal; Confluence; WordPress

Streszczenie

Artykuł przedstawia porównanie gotowych rozwiązań możliwych do wykorzystania podczas tworzenia platformy intranetowej stosowanej przez sieć korporacyjną. Kluczowym celem jest wskazanie, która z przedstawionych technologii jest najbardziej odpowiednia dla dużej, międzynarodowej korporacji. W badaniach uwzględniono po dwa otwarte (WordPress, Drupal) oraz komercyjne (SharePoint, Confluence) narzędzia.

Słowa kluczowe: SharePoint; Drupal; Confluence; WordPress

*Corresponding author

Email address: czarny.konrad95@gmail.com (K. Czarny)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Artykuł porównuje rozwiązania otwarte oraz korporacyjne możliwe do wykorzystania przy tworzeniu korporacyjnej sieci intranetowej. Technologie do porównania wybrane zostały na podstawie analiz ich popularności przedstawianych corocznie przez stronę w3techs.com [1, 2] z uwzględnieniem ich wykorzystania do tworzenia sieci intranetowej. W przypadku rozwiązań otwartych wybrano narzędzie WordPress i Drupal, natomiast w przypadku rozwiązań komercyjnych SharePoint oraz Confluence [1, 2, 3, 4].

2. Cel badań oraz metody i narzędzia badawcze

Celem badań było porównanie wskazanych systemów CMS (ang. Content Management System), pozwalających na implementację sieci intranetowej oraz wybranie najbardziej odpowiedniego dla wymagań charakteryzujących międzynarodową korporację.

Do przeprowadzenia analizy porównawczej wybrano metodę analizy wielokryterialnej. W tym celu przeanalizowano wymagania, jakie musi spełniać aplikacja intranetowa. Analiza ta została przeprowadzona na przykładzie innych publikacji naukowych [3], oraz z wykorzystaniem przykładowej aplikacji, funkcjonującej w międzynarodowej korporacji.

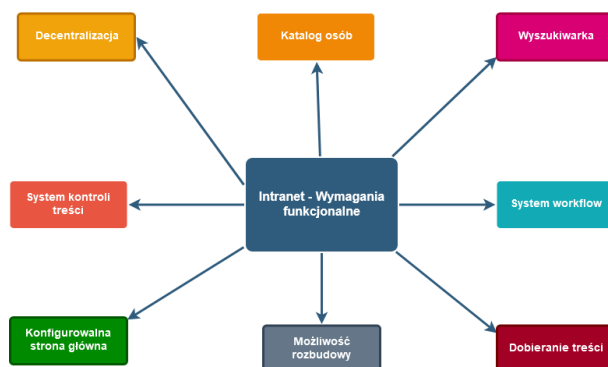
W celu wykonania analizy funkcjonalności wybranych technologii, wykorzystano publicznie dostępne informacje, takie jak dokumentacje [5-8] i artykuły [9]. Zebrane dane oceniono pod kątem spełnienia wymagań stawianych aplikacjom intranetowym.

3. Aplikacja intranetowa

Pierwszym krokiem potrzebnym do przeprowadzenia analizy porównawczej wybranych technologii jest określenie wymagań funkcjonalnych i нефункциональных, wymogów bezpieczeństwa oraz procesów korporacyjnych, które mają bezpośrednie przełożenie na sposób realizacji określonych wymagań [3].

3.1. Specyfikacja wymagań funkcjonalnych

Korporacyjna platforma intranetowa wymaga sprawnego podejścia do aktualizowania i uruchamiania nowych funkcji, które będą służyć jej użytkownikom. Istotne jest także zachowanie pewnej równowagi pomiędzy funkcjonalnością aplikacji, a jej bezpieczeństwem i wydajnością. Szczegółowe wymagania funkcjonalne charakteryzujące aplikację intranetową skierowaną do międzynarodowej korporacji przedstawia rysunek 1.

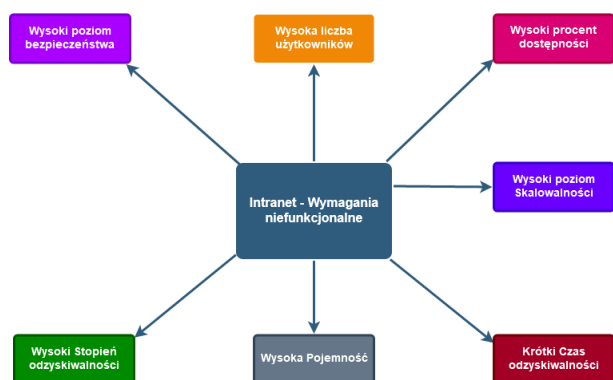


Rysunek 1: Schemat wymagań funkcjonalnych aplikacji intranetowej

Wymagania te, zostały wykorzystane jako kryteria służące analizie porównawczej, oraz szerzej opisane w rozdziale 5.

3.2. Specyfika wymagań niefunkcjonalnych

Mimo iż wymagania niefunkcjonalne nie posiadają bezpośredniego wpływu na funkcjonalność systemu, to nakładają na niego szereg zaleceń, opisujących w jaki sposób owe wymagania mają być realizowane. Wynika to bezpośrednio z wymagań oraz procesów obowiązujących w danej organizacji, oraz z ograniczeń występujących w zastosowanej technologii. Szczegółowe wymagania niefunkcjonalne charakteryzujące aplikację intranetową dla międzynarodowej korporacji zostały przedstawione na rysunku 2.

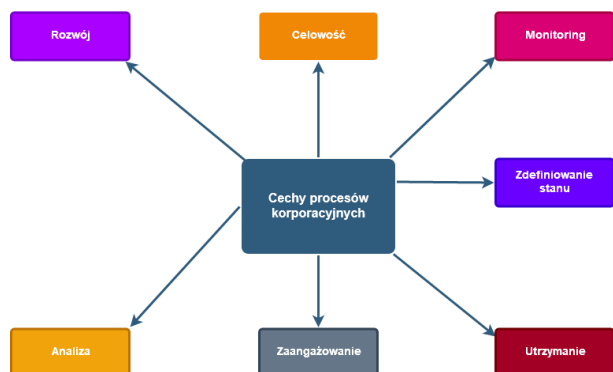


Rysunek 2: Schemat wymagań niefunkcjonalnych aplikacji intranetowej

Wymagania te, zostały wykorzystane jako kryteria służące analizie porównawczej, oraz szerzej opisane w rozdziale 5.

3.3. Charakterystyka procesów korporacyjnych

Proces biznesowy lub korporacyjny odnosi się do szerokiego zakresu ustrukturyzowanych, często powiązanych ze sobą działań lub zadań wykonywanych przez ludzi bądź sprzęt, w celu wytworzenia usługi czy produktu dla konkretnego konsumenta. Procesy biznesowe są wdrażane w celu osiągnięcia założonego celu organizacyjnego. Zachodzą one na wszystkich poziomach organizacyjnych. Niektóre z nich są widoczne dla klientów, inne natomiast nie.



Rysunek 3: Schemat procesów korporacyjnych aplikacji intranetowej

Międzynarodowa organizacja posiada wiele procesów ukierunkowanych na osiągnięcie różnych celów. Na rysunku 3 zostały przedstawione ich charakterystyki, które mają wpływ na projekt aplikacji [4].

3.4. Wymagania dotyczące bezpieczeństwa

System taki jak Intranet zawiera zazwyczaj poufne informacje o znaczeniu krytycznym dla firmy, takie jak: szczegółowe informacje o realizowanych projektach, dane osobowe pracowników, wiedzę instytucjonalną, prywatne rozmowy czy dokumenty zawierające własność intelektualną i inne informacje zastrzeżone. Jest to system, który wymaga solidnych zabezpieczeń. Istnieje wiele wszelkiego rodzaju zagrożeń wewnętrznych i zewnętrznych – wewnętrzne naruszenia bezpieczeństwa, nieautoryzowany dostęp, czy słabe hasła mogące zagrozić systemowi. Zarządzanie i ochrona danych internetowych są niezbędne dla każdej firmy. Każda firma jest zagrożona nie tylko złośliwymi atakami z zewnątrz, ale także wewnętrznymi naruszeniami ze strony pracowników (zamierzonymi lub nie). Każde oprogramowanie intranetowe musi mieć zintegrowane rozwiązanie bezpieczeństwa, aby zapewnić maksymalną możliwą ochronę. Może to wydawać się oczywiste, jednak protokoły logowania są pierwszą linią obrony intranetu. Funkcjonalność ta jest pierwszą barierą broniącą portal przed zagrożeniami. Rozwiązanie takie powinno wspierać integrację z „pojedynczym logowaniem” (ang. *single sign-on*, SSO), Active Directory (AD) lub Lightweight Directory Access Protocol (LDAP) w celu zapewnienia płynnego i centralnie zarządzanego procesu uwierzytelnienia. System ten powinien również zapewniać solidną kontrolę nad dostępem wewnętrznym oraz dostępem partnerów zewnętrznych. Pracownicy i partnerzy powinni móc uzyskać dostęp do intranetu za pośrednictwem platform mobilnych lub stacjonarnych **bezpiecznie**, bez względu na to gdzie się znajdują. Aplikacja, powinna spełniać takie wymogi bezpieczeństwa jak [10]:

- EU-USA & SWISS-US Privacy Shield;
- certyfikaty GLBA (ang. Gramm-Leach-Bliley Act);
- certyfikaty HIPAA (ang. Health Insurance Portability and Accountability Act);
- zgodność z RODO (ang. General Data Protection Regulation, GDPR).

W celu zapewnienia odpowiedniego poziomu bezpieczeństwa, organizacja korzystająca z intranetu będzie chciała wykorzystać wiele warstw obrony w celu ochrony kluczowych informacji i obsługi wszystkich krytycznych aspektów bezpieczeństwa sieci i aplikacji, w tym uwierzytelniania i autoryzacji. Mocny poziom bezpieczeństwa intranetu powinien obejmować takie kwestie jak przestrzeganie normy ISO 27001, regularne audyty stron trzecich oraz spójne aktualizacje w oparciu o trendy w branży. Zapewnia to, iż program bezpieczeństwa systemu dotrzyma kroku aktualnym standardom zabezpieczeń. Dobrą praktyką jest również zaimplementowanie systemu zarządzania bezpieczeństwem informacji (SZBI) [13]. Opisuje on ramy zasad i procedur,

które obejmują wszystkie prawne i fizyczne oraz techniczne kontrole zaangażowane w procesy zarządzania ryzykiem informacyjnym w organizacji, a także systematycznie ocenia i ustala priorytety zagrożeń bezpieczeństwa informacji, biorąc pod uwagę indywidualne podatności danej firmy. Należy również upewnić się, iż o bezpieczeństwo intranetu dbają certyfikowani specjaliści, którzy posiadają wiedzę specjalistyczną w zakresie bezpieczeństwa aplikacji, sieci oraz architektury. Specjaliści Ci mogą zdefiniować wiodące w branży zasady bezpieczeństwa i jego kontrole, co jest kluczowym zagadnieniem pozwalającym zabezpieczeniu danego systemu na obronę przed nowymi zagrożeniami.

4. Przegląd analizowanych technologii

W artykule przeanalizowano możliwości oferowane przez dwa systemy otwarto źródłowe (Word Press i Drupal) oraz dwa systemy komercyjne (SharePoint i Confluence).

4.1. WordPress

WordPress, jest to najpopularniejsze na świecie narzędzie do tworzenia stron internetowych, oparte na języku PHP oraz relacyjnej bazie danych MySQL na licencji GNU (General Public License) [9]. Zawiera wbudowaną architekturę rozszerzeń oraz motywów. Używany jest on przez ponad 60 milionów witryn internetowych, co stanowi 33.6% z 10 milionów najpopularniejszych stron sieciowych [3]. Za jego pomocą można przygotować prosty blog, sklep internetowy (z popularną wtyczką WooCommerce), a także w pełni funkcjonalną stronę biznesową. Opanowanie podstaw jego użytkowania nie wymaga prawie żadnych umiejętności programistycznych, czy projektowych. Dzięki możliwości wyboru tysięcy darmowych projektów witryn, zwanych motywami, oraz wtyczek zapewniających szeroką gamę funkcjonalności, do stworzenia własnej witryny sieciowej wystarczy podstawowa wiedza informatyczna. Dzięki stworzeniu natywnych aplikacji pod systemy takie jak Android czy iOS, zaprojektowanych przez firmę Automattic, możliwy jest wygodny i bezproblemowy dostęp do tworzonych witryn również za pośrednictwem platform mobilnych.

4.2. Drupal

Drupal to kolejny otwarto źródłowy system zarządzania treścią, oparty na języku PHP oraz licencji GNU General Public License [10]. Dostarcza on kod po stronie serwera dla minimum 2.3% wszystkich stron internetowych na świecie, takich jak blogi, strony korporacyjne, witryny partii politycznych czy rządowych [4]. Obecnie drupal posiada 114 tysięcy aktywnie rozwijających go użytkowników, co skutkuje stworzeniem ponad 42 tysięcy darmowych modułów rozszerzających jego wbudowane funkcjonalności, czy prawie 3 tysięcy darmowych układów stron [10]. Jego podstawowe wydanie, znane jako Drupal Core, zawiera szereg znanych z innych systemów CMS funkcjonalności, takich jak system kont, zarządzanie menu, kanały RSS, taksono-

nię, modyfikację układu strony czy system administracji.

4.3. SharePoint

SharePoint to wprowadzona na rynek w 2001 roku, multiplatformowa, internetowa, komercyjna platforma kolaboracyjna, integrująca się z pakietem Microsoft Office [7]. Sprzedawana jest ona przede wszystkim jako system zarządzania treścią i przechowywania dokumentów. Produkt ten jest wysoce konfigurowalny, a jego użycie może różnić się znacznie w zależności od używającej go organizacji. Według statystyk podawanych przez tworzącą go firmę – Microsoft, SharePoint ma 190 milionów użytkowników wśród 200 tysięcy organizacji klienckich [1].

SharePoint można podzielić na dwie główne edycje:

- SharePoint Serwer - udostępniany organizacjom wymagającym większej kontroli nad funkcjonalnościami i wyglądem aplikacji. Produkt ten instalowany jest w infrastrukturze klienckiej. Jest rzadziej aktualizowany niż jego bratnia wersja, posiada jednak dostęp do szeregu unikalnych dla siebie funkcji i możliwości dostosowywania;
- SharePoint Online – hostowana przez Microsoft, wersja systemu SharePoint, zawarta zazwyczaj w planie subskrypcji Microsoft Office 365, posiada jednak również możliwość kupna bezpośredniego. Zaletą tej wersji usługi jest to, iż nie wymaga ona utrzymywania własnej infrastruktury serwerowej. Rezultatem tego jednak jest mniej opcji dostosowania produktu. Ograniczony jest do podstawowego zestawu scenariuszy współpracy, hostingu plików oraz zarządzania dokumentami i treścią.

4.4. Confluence

Confluence to komercyjne oprogramowanie do współpracy między użytkownikami stworzone przez australijską firmę Atlassian w języku Java [8]. Pierwsza wersja została opublikowana w 2004 roku. Podstawowa wersja, Confluence Standalone, jest wyposażona we wbudowany serwer internetowy Tomcat, oraz bazę danych HSQL, obsługuje jednak również inne bazy danych. Confluence jest sprzedawany jako oprogramowanie dla przedsiębiorstw, oraz licencjonowany jako oprogramowanie lokalne lub usługa oferowana w chmurze.

5. Kryteria porównawcze

Z punktu widzenia wymagań związanych z użytkowaniem aplikacji intranetowej, system powinien udostępniać:

- **Katalog osób** – rozbudowany katalog osób, z opcją pobierania danych z wielu źródeł, oraz możliwość dodawania informacji, takich jak języki czy kwalifikacje, jest niezbędna w systemie;
- **System Workflow** – umożliwia organizacjom usprawnienie kluczowych procesów biznesowych. Pozwala to użytkownikom, którzy nie posiadają umiejętności technicznych, na proste i intuicyjne

tworzenie formularzy automatyzujących procesy biznesowe. Dzięki temu firma jest w stanie zachować wysoką wydajność, eliminując jednocześnie potrzebę zatrudniania osób z branży IT, lub administratorów systemu. Istnienie Workflow jest bardzo pożądane;

- System kontroli treści – Intranet musi zawierać także prosty w obsłudze oraz zorientowany na użytkownika system zarządzania treścią, z takimi funkcjonalnościami jak tagowanie, słowa kluczowe czy ocenianie stron w celu możliwości wyszukania ich pod kątem najlepszych ocen dla określonych haseł. Dostępne powinny być również dodatkowe funkcjonalności, takie jak daty interakcji czy kontrola wersji, które pomagają zachować aktualność całej zawartości witryny.
- Dobieranie treści – wraz z rozrostem organizacji, zwiększa się także liczba projektów, stanowisk czy grup pracowników. Kierowanie wszystkich informacji do każdego pracownika, niezależnie od biura w którym się znajduje czy używanego przez niego języka jest wysoce nieefektywne. Treści powinny być ukierunkowywane na unikalne potrzeby określonego pracownika.
- Wyszukiwarkę – wydajna i wszechstronna wyszukiwarka, powinna umożliwić dostęp do wyselekcjonowanych treści intranetu, oraz systemów pamięci masowej w chmurze.
- Konfigurowalną stronę główną – dzięki odpowiednim informacjom, widżetom i treściom dobranym specjalnie dla określonych grup, strona główna powinna zapewniać spersonalizowaną obsługę wielu modułów. Dzięki takiej funkcjonalności nowi użytkownicy intranetu, mogą szybko otrzymywać istotne dla nich informacje. To z kolei zapewni im łatwe wdrożenie się w specyfikę pracy w danej organizacji.
- Witryny zespołu (decentralizacja) – jedną z kluczowych funkcjonalności w nowoczesnych intranetach są „witryny zespołu” (ang. teams). Pozwalają one na unikalne dla danej organizacji projektowanie struktury oraz hierarchii witryny, czyniąc ją bardziej modularną. Idea witryn zespołu polega na możliwości tworzenia wielu podstron w obrębie intranetu w celu pracy nad poszczególnymi projektami w firmie. Każda z takich witryn posiada odrębny od reszty intranetu system uprawnień oraz strukturę, a także może być modyfikowalna wedle uznania w celu spełnienia wymagań danego projektu. Zmiany takie nie wpływają na resztę intranetu czy inne witryny.
- Możliwość rozbudowy – wraz z rozwojem firmy, intranet musi podlegać ciągłemu procesowi rozbudowy w celu dostosowania do zmieniających się potrzeb korporacji. Wymaga to wysokiej modułowości systemu oraz architektury łatwej do rozbudowy i długiego okresu wsparcia. Możliwość ta powinna być oficjalnie zapewniana przez producenta.
- Możliwość korzystania dla dużej liczby użytkowników – organizacja korzystająca z intranetu, składa się zazwyczaj z wielu pracowników i kontraktorów

z różnych kontynentów. Wymaga to więc, aby każdy z nich posiadał własne konto z przydzielonymi do niego informacjami i uprawnieniami. System taki musi być w stanie potencjalnie obsłużyć nawet setkę tysięcy użytkowników z różnych krajów oraz stref czasowych, co jest często nieosiągalne dla zwykłej witryny internetowej. System powinien być w stanie obsługiwać ponad 100 000 użytkowników.

- Dużą pojemność – biorąc pod uwagę wspomnianą wyżej liczbę użytkowników oraz funkcjonalności takie jak zarządzanie dokumentami i tworzenie witryn zespołu, system dodatkowo musi być w stanie przyjąć ogromną ilość danych oraz plików, rzędu dziesiątków terabajtów. Wymaga to nie tylko odpowiedniej infrastruktury sieciowej i fizycznej, ale także systemu zdolnego do symultanicznej analizy wielu informacji. System powinien być w stanie obsługiwać przynajmniej 1 terabajt danych.
- Skalowalność – wraz z rozwojem organizacji, oraz wzrostem liczby jej pracowników (a co za tym idzie, osób korzystających z intranetu), w parze z możliwością rozbudowy o nowe funkcjonalności, musi iść także skalowalność systemu pod względem liczby jego użytkowników oraz bazodanowych informacji trzymany na serwerach. Producent powinien podać jaki jest wpływ rozbudowy systemu na jego wydajność.
- Wysoki poziom bezpieczeństwa – aby chronić poufne informacje organizacji zarówno przed osobami z zewnątrz, jak i przed pracownikami, którzy nie powinni mieć do nich dostępu, aplikacja potrzebuje rozbudowanego systemu ról. Dzięki temu, można zagwarantować, że dostęp do danych wrażliwych mają tylko uprawnione osoby. Kryterium do wskazania poziomu bezpieczeństwa systemu jest indywidualna ocena pod kątem opisanych w podrozdziale 3.4 cech, oraz danych historycznych na temat wykrytych błędów.
- Wysoki procent dostępności – dostępność określa liczbę punktów procentowych czasu, w którym aplikacja jest dostępna dla użytkowników. Aplikacja, która ma służyć jako główny system korporacji, wymaga osiągnięcia dostępności na poziomie minimum 98% [5]. Oznacza to iż wszelkie przestoje związane z wdrożeniami nowej wersji aplikacji, naprawą błędów czy jej konfiguracją nie powinny przekraczać 2% ogólnego czasu dostępu.
- Wysoki stopień odzyskiwalności – termin ten określa liczbę punktów procentowych w jakich aplikacja ma zostać przywrócona do swojego poprzedniego stanu po ewentualnym błędzie krytycznym. Dla międzynarodowej firmy operującej na dziesiątkach terabajtów danych i informacji, stopień odtworzenia gwarantowany przez aplikację musi wynosić **minimum 95%** [5]. Oznacza to, iż praktycznie wszystkie dane powinny zostać przywrócone do stanu sprzed wystąpienia błędu.
- Krótki czas odtworzenia – to czas, w trakcie którego aplikacja powinna zostać przywrócona do stanu poprzedniego po ewentualnym błędzie krytycznym.

Dla dużej organizacji, do której dostęp dziennie mają uzyskiwać setki tysięcy osób, czas ten nie powinien przekraczać **16 godzin** [5]. Biorąc pod uwagę 8 godzinny dzień pracy obowiązujący w większości firm, czas taki gwarantuje przywrócenie aplikacji do stanu sprzed wystąpienia błędu z dnia na dzień, między okresami największej używalności. Przestój powinien być możliwie minimalną niedogodnością dla użytkownika końcowego.

6. Analiza porównawcza

W celu porównania omawianych technologii, wybrano 15 opisanych w poprzednim rozdziale kryteriów, które powinien spełniać szkielet aplikacyjny stosowany do tworzenia intranetu dla międzynarodowej korporacji. Kryteria te zostały wyselekcjonowane, w oparciu o opisane wymagania funkcjonalne i нефункционалne, charakterystykę procesów korporacyjnych oraz wymagania dotyczące bezpieczeństwa. Najwygodniejszym i najbardziej czytelnym sposobem zestawienia omawianych wymagań, jest prezentacja tabelaryczna (Tabela 1).

Tabela 1: Spełnianie wymagań przez określone technologie

	WordPress	Drupal	SharePoint	Confluence
Katalog osób	Jako plugin	Brak	Wbudowany	Brak
System workflow	Jako plugin	Wbudowany	Wbudowany	Wbudowany
System kontroli treści	Wbudowany	Niedoskonały	Wbudowany	Wbudowany
Dobieranie treści	Brak	Brak	Wbudowane	Brak
Wyszukiwarka	Wbudowana	Wbudowana	Wbudowana	Wbudowana
Konfigurowalna strona główna	Wbudowana	Wbudowana	Wbudowana	Wbudowana
Decentralizacja	Jako plugin	Wbudowana	Wbudowana	Wbudowana
Możliwość rozbudowy	Wbudowana	Wbudowana	Wbudowana	Możliwa, ale nie wspierana
>100 000 użytkowników	Tak	Tak	Tak	Zależy od licencji
Pojemność >1 TB	Tak	Tak	Tak	Tak
Skalowalność	Tak	Tak	Tak	Zależy od licencji
Poziom bezpieczeństwa	Niski	Niski	Wysoki	Wysoki
Procent dostępności	Zależy od implementacji	Zależy od implementacji	Min 95%	Min 95%
Stopień odżykiwalności	Zależy od implementacji	Zależy od implementacji	Min 98%	Min 98%
Czas odżykiwalności	Zależy od implementacji	Zależy od implementacji	<16 godzin	<16 godzin
Suma punktów	20	20	30	23

Każde z prezentowanych kryteriów oceniono w skali punktowej 0-2 z tą samą wagą, gdzie:

- 2 punkty oznaczają bezwarunkowe spełnienie wymagań (kolor zielony);
- 1 punkt oznacza, że dane kryterium dostępne jest np. w postaci dodatkowej wtyczki (kolor żółty);
- 0 punktów oznacza brak danej funkcji (kolor czerwony).

Dana technologia mogła uzyskać maksymalnie 30 punktów. Końcowe wyniki zestawione są w ostatnim wierszu tabeli 1.

Na ostateczne wyniki porównania duży wpływ ma fakt, że procent dostępności, stopień i czas odżykiwania danych dla SharePoint i Confluence, są gwarantowane przez producentów. W przypadku ich odpowied-

ników o otwartym dostępie, wartości te są zależne od konkretnej implementacji oraz umiejętności osób odpowiedzialnych za obsługę systemu.

7. Wnioski

Analizując wymagania funkcjonalne, нефункционалne oraz procesy wykorzystywane w międzynarodowych organizacjach, należy uznać, iż aplikacje o zamkniętej strukturze kodu są odpowiedniejsze do stworzenia korporacyjnego systemu intranetowego. Wysoki w porównaniu do otwartoźródłowych systemów, poziom bezpieczeństwa zapewniany przez nie jest jednym z najważniejszych warunków, jakie powinna spełniać korporacyjna platforma intranetowa. Trzymanie w tajemnicy ewentualnych luk w strukturach systemu bezpieczeństwa, wsparcie twórców tych szkieletów aplikacyjnych oraz szereg najwyższej klasy ekspertów stojących za zaimplementowanymi w nich zabezpieczeniami daje możliwie wysoką gwarancję bezpieczeństwa wrażliwych danych.

Porównując dwa wybrane z tej kategorii szkielety aplikacyjne (SharePoint oraz Confluence), lepszy jest ten pierwszy. Jego podstawowa implementacja nie tylko spełnia wymienione wymagania funkcjonalne, oraz pozytywnie odwzorowuje ich implementację poprzez wymagania нефункционалne, ale także oferuje wystarczająco elastyczny system rozbudowy. Dzięki temu, zespół programistyczny o odpowiednich umiejętnościach będzie w stanie wypracować wszelkie nieobecne w nim funkcjonalności a następnie zaimplementować je do już istniejącego systemu.

Przeprowadzona analiza skupiała się na wybraniu odpowiedniego szkieletu aplikacyjnego do określonego typu aplikacji, wymagającej dostępu dla minimum 100 000 użytkowników i przetwarzającej setki gigabajtów danych oraz wymagającej możliwości ciągłej rozbudowy, dostosowywania, ciągłego dostępu oraz jak największego poziomu bezpieczeństwa.

W przypadku intranetowej aplikacji dla klienta korporacyjnego, najodpowiedniejszą jest platforma SharePoint, która mimo swoich kosztów eksploatacji, związanej z zespołem programistycznym i licencjami, daje największe możliwości oraz spełnia najwięcej z ocenianych w niniejszym artykule wymagań.

W przypadku technologii o otwartym dostępie, oba porównywane rozwiązania są na podobnym poziomie.

Literatura

- [1] W3techs, <https://w3techs.com/technologies/details/cm-sharepoint/all/all> [01.09.2018]
- [2] W3techs, <https://w3techs.com/technologies/details/cm-confluence/all/all> [01.09.2018]
- [3] W3techs, <https://w3techs.com/technologies/details/cm-wordpress/> [01.09.2018]
- [4] W3techs, <https://w3techs.com/technologies/details/cm-drupal/> [01.09.2018]
- [5] J. Robertson, Designing Intranets: Creating Sites that Work. Step Two Designs, 2010

- [6] A. Rockley, P. Kostur, S. Mannig, Managing Enterprise Content: A Unified Content Strategy. New Riders, 2003.
- [7] Microsoft, <https://docs.microsoft.com/en-us/sharepoint/> [01.09.2018]
- [8] Atlassian, <https://confluence.atlassian.com/> [01.09.2018]
- [9] WordPress, <https://codex.wordpress.org/> [01.09.2018]
- [10] Drupal, <https://www.drupal.org/documentation/> [01.09.2018]
- [11] C-sharpcorner, <https://www.c-sharpcorner.com/UploadFile/40e97e/sharepoint-non-functional-requirements/> [01.09.2018]
- [12] Bezpieczna chmura, <https://zaufanatrzeciastrona.pl/post/jak-bezpiecznie-przenieść-się-do-chmury-wymagania-prawne/> [01.09.2018]
- [13] OpenSecurity, <https://opensecurity.pl/bezpieczenstwo-informacji-wg-iso-27001/> [01.09.2018]

Comparative analysis of solutions used in automated testing

Analiza porównawcza rozwiązań wykorzystywanych w testowaniu automatycznym

Agnieszka Dorota Wac*, Tomasz Kamil Watras, Grzegorz Koziel

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

The article presents a comparative analysis of tools supporting the production of automated tests for specific test scenarios. Within the framework of the research, the described tools were installed and then their functionalities were described. Using each of the tools, author's automatic tests were written and performed. The tool discussion was examined both in terms of the process of creating the tests and the speed of their execution. In addition, the selected tools were evaluated and subjective evaluations of the comfort of using the tool were taken into account, taking into account aspects such as the ease of installation and configuration of the tool, the need for programming skills, the simplicity of creating tests, additional functionalities and access to documentation. The summary includes the evaluation of the compared applications, and also discusses specific cases for which the tool will work better.

Keywords: Selenium; Katalon Studio; SahiPro; TestComplete

Streszczenie

W artykule przedstawiona została analiza porównawcza narzędzi wspomagających wytwarzanie zautomatyzowanych testów dla sprecyzowanych scenariuszy testowych. W ramach badań oprogramowanie zostało zainstalowane, a następnie opisano jego funkcjonalności. Przy pomocy każdego z narzędzi zostały napisane i wykonane autorskie testy automatyczne. Omawianie programy zbadano zarówno pod kątem procesu tworzenia testów, jak i prędkości ich wykonywania. Dodatkowo zostały ocenione pod względem subiektywnej oceny komfortu używania danego narzędzia, biorąc pod uwagę aspekty takie, jak łatwość instalacji i konfiguracji, potrzeba znajomości programowania, prostota tworzenia testów, dodatkowe funkcjonalności oraz dostęp do dokumentacji. W podsumowaniu zawarta została ocena porównywanych aplikacji, a także omówione zostały konkretne przypadki, dla których dane oprogramowanie sprawdzi się lepiej.

Słowa kluczowe: Selenium; Katalon Studio; SahiPro; TestComplete

*Corresponding author

Email address: agnieszka.wac@pollub.edu.pl (A. D. Wac)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Szybki rozwój technologii informatycznych, duża konkurencja na rynku oraz presja ze strony klientów firm informatycznych wymuszają zwiększanie tempa wytwarzania i dostarczania oprogramowania. Zmiany standardów oraz wymagań wpływają na stosowanie coraz to innych, nowszych technologii. Zmiana metodyk zarządzania projektami i przejście w kierunku metodyk zwinnych zmusza twórców do większej elastyczności, ale też wprowadza więcej zmian w tworzonym rozwiązaniu.

Wszystkie wymienione zjawiska skutkują zwiększeniem ryzyka popełnienia błędu podczas procesu wytwarzania oprogramowania. Skutecznym sposobem redukcji liczby defektów oraz zwiększania niezawodności dostarczanych rozwiązań informatycznych jest testowanie. Stosowane jest ono w większości dużych projektów. Naturalnym wydaje się być wykonywanie testów manualnych. Jednak wysokie koszty pracy ludzkiej oraz brak odpowiedniej liczby wykwalifikowanych testerów skutkują wprowadzaniem testów automatycznych. Testy te są szczególnie opłacalne w przypadku wielokrotnego powtarzania tych samych przypadków

testowych. Wówczas raz opracowany test automatyczny jest uruchamiany wielokrotnie bez konieczności angażowania zasobów ludzkich.

W niniejszym artykule przedstawiono narzędzia pozwalające na automatyzację testów funkcjonalnych aplikacji internetowych oraz takie, które wspomagają testowanie interfejsu programistycznego aplikacji opartych o architekturę REST (ang. representational state transfer).

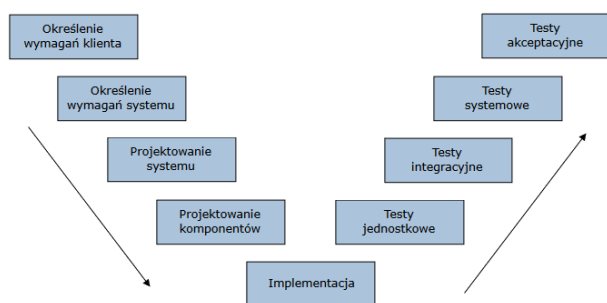
2. Modele wytwarzania oprogramowania

Zgodnie z podejściem ISTQB (International Software Testing Qualification Board - Stowarzyszenie Jakości Systemów Informatycznych) w różnych modelach wytwarzania oprogramowania stosowane są odmienne techniki planowania i realizacji testów [1].

Najbardziej rozpowszechnionym rozwiązaniem jest model wodospadu. Charakteryzuje się on tym, że każdy kolejny etap rozpoczyna się bezpośrednio po akceptacji ukończenia poprzedzającego [2]. Proces wytwórczy dzieli się na określone fazy: specyfikacja wymagań, projektowanie, implementacja, integracja, testy i weryfikacja, wdrożenie, utrzymanie. Podejście to sprawdza się dobrze w małych projektach, o ściśle okre-

ślonych wymaganiach, ponieważ pozwala na precyzyjne rozmieszczenie zasobów i weryfikację każdego etapu wytwarzania oprogramowania. Nie jest jednak polecane do dynamicznie zmieniających się oraz rozbudowywanych projektów. Dostarczenie gotowego produktu odbywa się dopiero w końcowych etapach.

Inne podejście prezentowane jest w modelu V. Jego nazwa pochodzi od specyficznego, równoległego sposobu wykonywania poszczególnych etapów, co prezentuje rysunek 1. W pierwszym etapie (konceptyjnym) określone zostają wymagania i wybierane są idee możliwe do zrealizowania przy określonych zasobach i możliwościach technicznych. Wymaga to zaangażowania zarówno zespołów architektów, jak i programistów oraz testerów. W kolejnych fazach - implementacyjnej oraz weryfikacyjnej, tworzony i sprawdzany jest gotowy produkt. Dzięki ścisłej współpracy wewnątrz projektu możliwe jest wypracowanie odpowiednich rozwiązań już na etapie powstawania idei, a w efekcie zapobieganie błędom zanim zostaną one zaimplementowane.



Rysunek 1: Rozkład etapów modelu V

Modele sekwencyjne stosuje się w przypadku, gdy założenia projektu są z góry znane, proces budowania aplikacji dokładnie opisany, a klient oczekuje gotowego produktu. Coraz częściej jednak tworzone są systemy o dużym stopniu skomplikowania, rozwijane przez długi okres czasu. W tej sytuacji popularność zyskują techniki zwinne [3]. Opierają się one na iteracyjnym modelu wytwarzania oprogramowania. Techniki te kładą nacisk na współpracę zespołu z klientem, wytwarzanie funkcjonujących części aplikacji oraz szybkiego reagowania na konieczność wprowadzania zmian. Klient nie otrzymuje gotowego produktu, ale uczestniczy w procesie jego tworzenia, otrzymując kolejne działające elementy systemu. Konieczność ciągłej weryfikacji poprawności działania i integralności systemów sprzyja automatyzacji testów.

Do wyboru odpowiedniego modelu wytwarzania oprogramowania konieczna jest dokładna analiza specyfiki danego projektu. Należy odpowiednio wcześniej określić sposób dostarczania działających aplikacji, możliwości współpracy z klientem, a także rozważyć korzyści płynące z rzetelnego opracowania i utrzymywania dokumentacji oraz postępowania ściśle według wytycznych.

3. Testy

Bez względu na wybór modelu, system należy sprawdzić pod wieloma względami. Testowana powinna być poprawność działania poszczególnych elementów, ich zgodność z wymaganiami, ale również aspekty нефunkcjonalne, takie jak stabilność, niezawodność, wydajność, użyteczność, bezpieczeństwo, kompatybilność, utrzymywalność, przenaszalność [4].

3.1. Rodzaje testów

Wczesne wykrycie możliwych błędów i podatności pozwala na sprawniejsze ich wyeliminowanie. Wprowadzanie zmian na wysokim stopniu zaawansowania projektu zawsze wiąże się ze zwiększeniem kosztów i wydłużeniem czasu wykonania produktu. Dlatego też często łączone jest wykonywanie testów białych i czarnoskrzynkowych. Pierwszy rodzaj skupia się wykryciu błędów metodami poprzez wprowadzanie gotowych zestawów danych i wymuszanie przejścia przez wszystkie ścieżki systemu, co wymaga znajomości programowania. Testy czarnoskrzynkowe pozwalają na weryfikację czy oprogramowanie spełnia swoje założenia. Zadaniem testera jest dobranie wartości i wykonanie działań sprawdzających czy uzyskany wynik działania danej funkcjonalności jest zgodny z przewidywanym w dokumentacji.

Aby mieć pewność, że każdy wykryty błąd został naprawiony konieczne jest wykonanie testów potwierdzających – retestów. Pozwalają one na weryfikację poprawności usunięcia defektu. Jednocześnie dodanie nowej funkcjonalności lub rozbudowa już istniejącej, powinna wiązać się z przeprowadzeniem testów regresyjnych, których celem jest wykrycie potencjalnych błędów spowodowanych wprowadzeniem zmian w innych częściach funkcjonalności.

W sylabusie ISTQB wyróżnia się testy [1] modułowe, integracyjne, systemowe oraz akceptacyjne. Każdy z nich wymaga skonfigurowania odpowiedniego środowiska testowego. Odnoszą się one również do innych aspektów i funkcjonalności oprogramowania.

Testy modułowe (jednostkowe) mają na celu sprawdzenie działania odseparowanych modułów aplikacji. Umożliwiają wykrycie błędów na poziomie funkcji. Symulowanie ich działania w środowisku docelowym jest uzyskiwane przy pomocy zaślepek, sterowników, atrapy systemu lub wirtualizacji usług [1].

W kolejnym etapie sprawdzana jest integracja modułów, czyli poprawność ich komunikacji. Wykonywane testy angażują nie tylko metody danego oprogramowania, ale również bazy danych, czy zewnętrzne serwisy. Sprawdzana jest prawidłowość w przesyłaniu komunikatów, ich kompletność, zabezpieczenia oraz synchronizacja.

Testy systemowe wymagają poprawnej i kompletnej konfiguracji całego systemu, odwzorującego środowisko produkcyjne lub docelowe. Pozwalają na sprawdzenie, czy stworzony system działa zgodnie z założeniami. Umożliwiają również wykrycie potencjalnych problemów współpracy ze środowiskiem docelowym, np.

w przypadku wykorzystywania innego systemu operacyjnego lub braku kompletnej konfiguracji.

Ostatnim etapem jest wykonanie testów akceptacyjnych. Ich celem nie musi być wyszukiwanie kolejnych defektów oprogramowania, ale sprawdzenie gotowości systemu do wdrożenia w docelowym środowisku. Wykonanie testów akceptacyjnych wiąże się z budowaniem zaufania do tworzonego oprogramowania. Umożliwia zapewnienie odbiorców końcowych, że wytworzony system działa w sposób przewidziany w specyfikacji i realizuje określone procesy biznesowe.

3.2. Testy manualne a automatyczne

Celem przeprowadzenia testów jest zapewnienie określonego poziomu jakości oprogramowania. Wiele czynności weryfikujących poprawność działania aplikacji może zostać zautomatyzowana, nadal jednak są aspekty, które wymagają ręcznej weryfikacji.

Testy manualne polegają na ręcznym uruchamianiu poszczególnych funkcji systemu przez testera. Pozwalają na wykrycie błędów działania aplikacji i interfejsów, a także ocenę takich aspektów systemu, jak łatwość obsługi, ergonomiczność i przydatność dla użytkownika. Zaletą testów manualnych jest to, że wykonywanie ich nie wymaga znajomości języków programowania, umożliwiają szybkie przetestowanie nowych lub zmieniających się funkcjonalności, a także wykonanie bardzo złożonych scenariuszy, nieopłacalnych do automatyzacji. Ich wadą jest konieczność bazowania na zasobach ludzkich, co powoduje wzrost kosztów i czasu potrzebnego do realizacji projektu.

Testy automatyczne opierają się na wykorzystaniu narzędzi i bibliotek do uruchamiania zaprogramowanych przypadków testowych. W początkowej fazie powstawania projektu wymagają większych kosztów niż testy manualne, jednak pozwalają na zapobieganie wielu błędom na późniejszych etapach, a także skrócenie czasu potrzebnego na przetestowanie niektórych aspektów oprogramowania dzięki możliwości ponownego użycia napisanych testów. Wymagają jednak ciągłej aktualizacji oraz weryfikacji działania, a także kontrolowania, czy ich wykonanie jest nadal zasadne i użyteczne. Automatyzacja jest najbardziej korzystna w przypadku konieczności wykonywania testów [5]:

- powtarzalnych,
- dla wielu różnych zestawów danych,
- na różnych platformach sprzętowych oraz konfiguracjach,
- czasochłonnych lub niemożliwych do ręcznej weryfikacji,
- wymagających wprowadzenia dużej ilości danych wejściowych,
- dla krytycznych funkcjonalności,
- wydajnościowych.

3.3. Narzędzia do testów funkcjonalnych aplikacji internetowych

SeleniumIDE jest to darmowy dodatek stworzony dla dwóch najpopularniejszych przeglądarek internetowych, czyli Chrome oraz Firefox. Jest on dostępny do

pobrania z oficjalnych sklepów dodatków. Tester nie musi się przejmować konfiguracją środowiska testowego ani instalowaniem skomplikowanych narzędzi. Po zainstalowaniu dodatku, narzędzie jest gotowe do pracy. Dzięki prostemu interfejsowi, użytkownik jest w stanie stworzyć zautomatyzowane testy bez korzystania z dokumentacji. W celu ułatwienia zarządzania tworzonymi testami, ma możliwość tworzenia projektów, a także definiowania przypadków użycia. Program oferuje szerokie możliwości nagrywania i edycji testów oraz definiowania ich przy pomocy dostępnych akcji.

Selenium WebDriver jest to udostępniony dla użytkownika zestaw funkcji dostępnych w WebDriver API, dołączonego do silnika Selenium, które pozwalają między innymi na wyszukiwanie elementów strony za pomocą selektorów CSS, xpath oraz atrybutów dostępnych dla znaczników HTML. Narzędzie jest przeznaczone dla bardziej zaawansowanych użytkowników posiadających znacznie większą wiedzę na temat wytwarzania oprogramowania w wysokopoziomowych językach, wspieranych przez Selenium WebDriver, takich jak C#, Java, Ruby, Python oraz JavaScript [6]. Użytkownik samodzielnie definiuje wykonywane kroki testowe, sposób obsługi błędów oraz generowanie informacji na temat przeprowadzonych testów.

TestComplete jest to komercyjne rozwiązanie do wytwarzania testów automatycznych, przeznaczone dla bardziej wymagających użytkowników. Narzędzie oferuje automatyzację testów dla aplikacji internetowych, desktopowych oraz mobilnych wykorzystujących systemy operacyjne Android oraz iOS. Użytkownik jest w stanie nagrywać testy, które mogą być zapisane jako skrypt do ponownego użycia. Program oferuje także integrację z innymi produktami takimi jak Jira czy też systemy do wersjonowania wersji [7].

SahiPro, oferujące automatyzację testów dla aplikacji internetowych, jest kolejnym komercyjnym narzędziem. Pozwala na nagrywanie testów, które są następnie dostępne do edytowania przez użytkownika. Wygenerowane testy mogą być zapisane jako funkcje w języku JavaScript celem ponownego użycia. Program wspiera głównie przeglądarki Firefox oraz Internet Explorer.

Katalon Studio jest samodzielnym narzędziem oferującym możliwość automatyzowania testów tworzonych przez użytkownika. Jest to rozwiązanie darmowe [8], oparte na silniku Selenium. Wszystko to zostało obudowane w interfejs, który zawiera najpotrzebniejsze elementy do tworzenia testów. Narzędzie to oferuje automatyzację testów aplikacji internetowych, mobilnych oraz serwisów internetowych. Program nie wymaga od użytkownika znajomości języków programistycznych. Korzystanie z narzędzi deweloperskich w przeglądarce pomoże użytkownikom w sprawniejszym tworzeniu testów.

Tabela 1: Dostępne opcje narzędzi do testowania aplikacji internetowych

	Selenium IDE	Selenium WebDriver	Test Complete	Sahi-Pro	Katalon Studio
Licencja	Darmowa	Darmowa	Płatna	Płatna	Darmowa
Testy UI	Tak	Tak	Tak	Tak	Tak
Testy aplikacji mobilnych	Nie	Nie	Nie	Tak	Tak
Testy aplikacji desktopowych	Nie	Nie	Tak	Tak	Tak
Testy Serwisów internetowych	Nie	Nie	Nie	Nie	Tak
Nagrywanie testów	Tak	Nie	Tak	Tak	Tak
Tworzenie testów od podstaw	Tak	Tak	Tak	Tak	Tak
Modyfikacja testów	Tak	Tak	Tak	Tak	Tak
Raportowanie wyników	Tak	Możliwe	Tak	Tak	Tak

3.4. Narzędzia do testów REST API

Do testów REST API wybrane zostały trzy narzędzia: Katalon Studio, Postman i SoapUI. Wszystkie te programy są dostępne za darmo. Umożliwiają testowanie poprawności działania aplikacji serwisów internetowych.

Postman jest najbardziej popularną z testowanych aplikacji. Początkowo dostępny jako rozszerzenie do przeglądarki Chrome, aktualnie udostępnia również oprogramowanie na systemy Windows. Wyróżnia się przejrzystym i czytelnym interfejsem. Wspomaga również pracę zespołów programistycznych i testerskich, umożliwiając współdzielenie tworzonych projektów [9] oraz szeroki zakres integracji z innymi aplikacjami. Jego zaletą jest dostarczenie przez twórców dokładnej dokumentacji oraz istnienie dużego wsparcia społeczności użytkowników.

Katalon Studio udostępnia funkcjonalność testowania serwisów internetowych, pomimo, że nie jest narzędziem dedykowanym do tego celu. Nie wymaga znajomości języków programowania do tworzenia przypadków testowych, ale umożliwia również generowanie skryptów w języku Groovy. Zaletą narzędzia Katalon Studio jest możliwość integracji z wieloma aplikacjami wspomagającymi tworzenie projektów, takimi jak: Jira, Jenkins, Bamboo, a nawet Postman. Dla mniej zaawansowanych użytkowników udostępniony został pakiet przykładowych projektów, ułatwiający szybkie zapoznanie się z narzędziem.

Dzięki prostemu interfejsowi graficznemu i funkcjonalnościom na poziomie klasy korporacyjnej, SoapUI pozwala na szybkie i proste tworzenie zautomatyzowanych testów funkcjonalnych, regresyjnych oraz obciążeniowych. Ze względu na ściśle określoną strukturę tworzonego projektu umożliwia parametryzację każdego poziomu testów. Dostarcza użytkownikowi narzędzi do budowania skomplikowanych scenariuszy

testowych, a jednocześnie nie wymaga od niego wiedzy z zakresu programowania, ponieważ definiowanie kolejnych kroków odbywa się przez dodawanie asercji. Najciekawszą z funkcjonalności udostępnianych przez to narzędzie jest możliwość generowania testów obciążeniowych i bezpieczeństwa. Ich wyniki przedstawiane są w czytelnej postaci wykresów i raportu [10].

4. Wyniki badania

Proces badania możliwości testowanego oprogramowania pozwolił na ocenę jego przydatności pod względem funkcjonalnym i niefunkcjonalnym.

4.1. Metodyka badawcza

Dla każdego narzędzia do testów aplikacji internetowych przygotowano następujące zasady podczas wykonywania scenariuszy testowych:

- za pomocą każdego narzędzia zostało stworzone 10 kont pocztowych,
- nazwa tworzonego konta pocztowego jest w formacie: MGR_NazwaNarzedzia_NrKonta,
- **NazwaNarzedzia** oznacza uproszczoną nazwę testowanego narzędzia, a **NrKonta** kolejno tworzone konto od 0 do 10. Konto posiadające „0” jako **NrKonta** oznacza, że dane konto powstało podczas nagrywania procesu testowego.
- podczas tworzenia nowego konta pocztowego na portalu Wirtualnej Polski, istnieje szansa na to, że system będzie wymagał od użytkownika uzupełnienia mechanizmu zabezpieczającego CAPTCHA. Takie próby zaraportowano i następnie zatrzymano, po czym proces ponawiano,
- po stworzeniu testu poprzez nagranie, odtwarzany jest proces w celu zweryfikowania poprawności pobierania elementów przez badany program.

Stworzono dwa przykładowe scenariusze testowe, które są przedstawione w tabelach 2 oraz 3. Testy były tworzone i wykonywane na komputerze stacjonarnym.

Tabela 2: Scenariusz tworzenia konta pocztowego

Nazwa	Tworzenie konta pocztowego	
Cel	Sprawdzenie poprawności tworzenia nowego konta pocztowego	
Warunki wstępne	Wylogowany użytkownik z domeny pocztowej	
Wymagania	Brak danego konta w systemie	
Tworzenie konta pocztowego		
LP	Opis wykonanych czynności	Oczekiwany Wynik
1.	Wejście na stronę www.wp.pl	Przeniesienie do witryny portalu WP
2.	Kliknięcie w odnośnik poczty email	Przeniesienie na witrynę poczty email portalu WP
3.	Kliknięcie w odnośnik zakładania nowego konta	Przeniesienie na witrynę zakładania konta pocztowego
4.	Uzupełnienie danych	Odpowiednie wypełnienie formularza
5.	Zaznaczenie elementu CAPTCHA	Zaznaczenie pola wyboru CAPTCHA
6.	Kliknięcie przycisku do utworzenia konta	

Tabela 3: Scenariusz logowania do poczty

Nazwa		Zalogowanie się do konta pocztowego
Cel		Sprawdzenie poprawności logowania do konta pocztowego
Warunki wstępne		Wylogowany użytkownik z domeny pocztowej
Wymagania		Powinno istnieć konto w systemie
Logowanie do konta pocztowego		
LP	Opis wykonanych czynności	Oczekiwany Wynik
1.	Wejście na stronę www.wp.pl	Przeniesienie do witryny portalu WP
2.	Kliknięcie w odnośnik poczty email	Przeniesienie na witrynę poczty email portalu WP
3.	Uzupełnienie danych logowania	Uzupełnione wartości
4.	Uzupełnienie danych	Odpowiednie wypełnienie formularza
5.	Kliknięcie przycisku do logowania	Przejdzie do witryny poczty użytkownika.
6.	Sprawdzenie czy istnieje przycisk Wyloguj	Istnieje przycisk do wylogowania
7.	Wylogowanie użytkownika poprzez naciśnięcie przycisku wyloguj	

Tworzenie testów dla zdefiniowanych scenariuszy testowych wykonano na zainstalowanych i skonfigurowanym środowisku. Rozpoczęto od tworzenia testów dla scenariusza służącego do tworzenia nowych kont pocztowych na portalu Wirtualna Polska.

Nagranie testu w SeleniumIDE wygenerowało odpowiednie kroki testowe. Kompletny scenariusz zdublikowano i zmodyfikowano wartości dotyczące tworzonego konta. Uruchomiono scenariusz testowy, który wykonał zdefiniowane wcześniej testy. Czasy ich wykonywania przedstawia tabela 4.

Tabela 4: Czasy tworzenia nowych kont za pomocą SeleniumIDE

Test	1	2	3	4	5	6	7	8	9	10	Ręczne nagrywanie
Czas [s]	9	10	9	10	9	9	9	10	10	10	29

Wyznaczone czasy w testach automatycznych oscylowały przy wynikach od 9 do 10 sekund, dając średni czas 9.5s. Przeprowadzając ręczne nagrywanie osiągnięto wynik 29s.

SeleniumWebDriver został wykorzystany dla języka C#, co wymagało użycia zintegrowanego środowiska deweloperskiego jakim jest Visual Studio 2017 w wersji Community. Po stworzeniu projektu, pobraniu odpowiedniej biblioteki oraz pobraniu sterownika przeglądarki Chrome w wersji 77, przystąpiono do zdefiniowania kroków testowych. Głównym ciałem klasy jest klasa testowa, która posiada w swojej definicji metadane pozwalające na parametryzację tego przypadku testowego.

Po poprawnym skonfigurowaniu kroków, przeprowadzono testy dla scenariusza tworzenia nowych kont, które się nie powiodły. System detekcji automatyzacji Captcha działa prawidłowo zapobiegając używaniu zautomatyzowanych testów, co niestety uniemożliwiło poprawne przeprowadzenie tego scenariusza testowego.

St...	No	Name	Start T...	End Time	Run Time	Details
1	CreatingAccount1	[KeywordTests - CreatingAccountRecording]	19:52:40	19:52:58	0:00:17	Details
2	CreatingAccount2	[KeywordTests - CreatingAccountRecording]	19:52:58	19:53:15	0:00:16	Details
3	CreatingAccount3	[KeywordTests - CreatingAccountRecording]	19:53:15	19:53:32	0:00:16	Details
4	CreatingAccount4	[KeywordTests - CreatingAccountRecording]	19:53:32	19:53:49	0:00:16	Details
5	CreatingAccount5	[KeywordTests - CreatingAccountRecording]	19:53:49	19:54:06	0:00:17	Details
6	CreatingAccount6	[KeywordTests - CreatingAccountRecording]	19:54:06	19:54:45	0:00:39	Details
7	CreatingAccount7	[KeywordTests - CreatingAccountRecording]	19:54:45	19:55:06	0:00:20	Details
8	CreatingAccount8	[KeywordTests - CreatingAccountRecording]	19:55:06	19:55:27	0:00:20	Details
9	CreatingAccount9	[KeywordTests - CreatingAccountRecording]	19:55:27	19:55:48	0:00:20	Details
10	CreatingAccount10	[KeywordTests - CreatingAccountRecording]	19:55:48	19:56:05	0:00:17	Details

Rysunek 1: Raport po wykonaniu scenariusza testowego

W aplikacji TestComplete stworzono projekt i przystąpiono do nagrania testu. Po wykonaniu scenariusza, otrzymano wymagane kroki testowe. Następnie stworzono scenariusz testowy wewnątrz programu i zdefiniowano testy dla poszczególnych nowych kont. Po uruchomieniu i pomyślnym zakończeniu został wygenerowany raport widoczny na rysunku 2.

Uzyskane czasy oscylowały w przedziale 16-20 sekund, ze średnią 17.77s. Test numer 6, którego czas wyniósł 39 sekund został świadomie odrzucony podczas analizy, ponieważ system portalu wykrył zautomatyzowany system tworzenia i wymagał uzupełnienia elementu Captcha.

W programie SahiPro, po uruchomieniu narzędzia wybrano kontroler, za pomocą którego nagrano test według scenariusza do tworzenia nowych kont pocztowych. Następnie wygenerowane kroki testowe zostały umieszczone w odpowiedniej funkcji. Po wykonaniu tego skryptu, żaden test się nie powiódł. Wynika to z tego powodu, że testowane narzędzie wymusza korzystanie z autorskiego proxy i powoduje wykrycie przez system Captcha, przez co testy się nie powiodły.

W Katalon Studio stworzono projekt i za pomocą dostępnego rozszerzenia do przeglądarki rozpoczęto nagrywanie testów. Nagrany test został zdublikowany, a następnie zmodyfikowany pod wymagane dane testowe, co przedstawia rysunek 3.

Test Suites	Command	Target	Value
CreateAccount	open	https://www.wp.pl/	
CreateAccount1	store	mgr_katalon_1	AccountName
CreateAccount2	click	link=Poczta	
CreateAccount3	click	link=załóż konto	
CreateAccount4	click	//div[@id=app]/div/div/div[2]/div/div[3]/form/div/div/input	
CreateAccount5	type	//div[@id=app]/div/div/div[2]/div/div[3]/form/div/div/input	Test
CreateAccount6	type	//div[@id=app]/div/div/div[2]/div/div[3]/form/div/div[2]/div/input	katalon
CreateAccount7			
CreateAccount8			
CreateAccount9			
CreateAccount10			

Rysunek 2: Stworzony scenariusz testowy w KatalonStudio

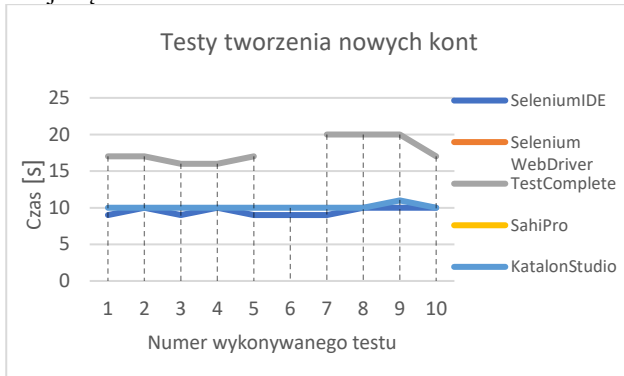
Uruchomiony scenariusz testowy pomyślnie utworzył konta pocztowe, a czas przeprowadzenia testów przedstawia tabela 5.

Tabela 5: Czasy tworzenia nowych kont w Katalon Studio

Test	1	2	3	4	5	6	7	8	9	10	Ręczne nagrywanie
Czas [s]	1	1	1	1	1	1	1	1	1	1	29
	0	0	0	0	0	0	0	0	1	0	

Uzyskane czasy kształtują się na poziomie 10s, w jednym przypadku wyniósł on 11s.

Porównano również czasy wykonania scenariusza w poszczególnych narzędziach, co przedstawia wykres na rysunku 4. Test numer 6 dla narzędzia TestComplete nie został uwzględniony ze względu na opisany wcześniej błąd.



Rysunek 3: Wykres czasów wykonywania scenariusza tworzenia nowych kont w poszczególnych narzędziach

W SeleniumIDE nagrano test kierując się scenariuszem testowym. Następnie wygenerowany test zduplikowano i zmodyfikowano pod wymagane dane. Po uruchomieniu scenariusza testowego, poszczególne przypadki wykonywały się w czasie przedstawionym w tabeli 6. Logowanie ręczne i wpisanie skomplikowanego hasła zajęło 14s. Dzięki automatyzacji czas ten spadł do średniej wartości 4.8s.

Tabela 6: Czasy logowania do kont za pomocą SeleniumIDE

Test	1	2	3	4	5	6	7	8	9	10	Ręczne logowanie
Czas [s]	5	5	5	5	5	5	4	4	4	6	14

Po ręcznym utworzeniu kont dla tego narzędzia, został opracowany automatyczny test wykonujący proces logowania.

Czasy wykonania poszczególnych testów są przedstawione poniżej (tabela 7). Średni czas wyniósł 11.03 sekundy.

Tabela 7: Czasy wykonywania testów logowania w Visual Studio 2017 przy użyciu Selenium WebDriver

Test	1	2	3	4	5	6	7	8	9	10
Czas [s]	12248	12575	12227	12852	8044	8009	8533	13856	9786	12250

W TestComplete do wcześniej utworzonego projektu, dodano kolejny projekt testowy pod nazwą *Logging*, którego celem jest wykonanie scenariusza służącego do zweryfikowania poprawności logowania w systemie.

St...	No	Name	Start Ti...	End Time	Run Time	Details
✓	1	Logging1 [KeywordTests - Logging]	21:00:26	21:00:38	0:00:11	Details
✓	2	Logging2 [KeywordTests - Logging]	21:00:38	21:00:48	0:00:10	Details
✓	3	Logging3 [KeywordTests - Logging]	21:00:48	21:00:59	0:00:10	Details
✓	4	Logging4 [KeywordTests - Logging]	21:00:59	21:01:10	0:00:11	Details
✓	5	Logging5 [KeywordTests - Logging]	21:01:10	21:01:21	0:00:11	Details
✓	6	Logging6 [KeywordTests - Logging]	21:01:21	21:01:32	0:00:10	Details
✓	7	Logging7 [KeywordTests - Logging]	21:01:32	21:01:43	0:00:10	Details
✓	8	Logging8 [KeywordTests - Logging]	21:01:43	21:01:54	0:00:11	Details
✓	9	Logging9 [KeywordTests - Logging]	21:01:54	21:02:09	0:00:14	Details
✓	10	Logging10 [KeywordTests - Logging]	21:02:09	21:02:20	0:00:11	Details

Rysunek 4: Raport dotyczący wykonanych testów logowania w TestComplete

Wykonanie testów wygenerowało raport, na podstawie którego możliwe jest sprawdzenie czy ukończyły się one prawidłowo, co ukazano na rysunku 5. Średni czas logowania wyniósł 11.3s.

W SahiPro przystąpiono do procesu nagrywania testu logowania użytkownika. Następnie zmodyfikowano wygenerowany skrypt w sposób umożliwiający parametryzację.

Proces wykonywania testów przeszedł częściowo pomyślnie. Ostatni krok testowy się nie powiódł z powodu błędu działania serwera wykorzystywanego przez firmę SahiPro. Wykonane testy zostały przy średnim czasie 15.59 sekundy, co jest przedstawione w tabeli 8.

Tabela 8: Czasy logowania przy użyciu SahiPro

Test	1	2	3	4	5	6	7	8
Czas [s]	11,45	15,12	17,36	17,15	16,49	15,67	16,89	14,6

W Katalon Studio nagrano proces logowania użytkownika. Stworzono scenariusz testowy w programie, dodano do niego zduplikowane i zmodyfikowane testy. Po uruchomieniu i pomyślnym ukończeniu scenariusza, zebrano czasy wykonywania poszczególnych testów do tabeli 9. Średni czas wyniósł 5.9s.

Tabela 9: Czasy logowania przy użyciu Katalon Studio

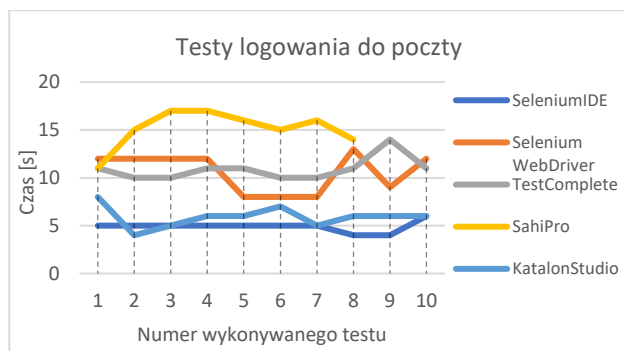
Test	1	2	3	4	5	6	7	8	9	10	Ręczne logowanie
Czas [s]	8	4	5	6	6	7	5	6	6	6	16

Porównano również czasy wykonania scenariusza w poszczególnych narzędziach, co przedstawia wykres na rysunku 6.

Wykonywanie testów REST API odbyło się przy pomocy darmowego zewnętrznego serwisu <http://petstore.swagger.io/>. Każde testowane narzędzie wykonało zdefiniowany w przypadkach testowych zestaw zapytań w celu oceny jego możliwości.

W ramach testów REST API zbadane zostały funkcjonalności takie, jak:

- tworzenie żądań GET z parametrami,
- tworzenie żądań POST z parametrami w ciele zapytania,
- parametryzacja testów,
- generowanie skryptów do testów automatycznych.



Rysunek 5: Wykres czasów wykonywania scenariusza logowania w poszczególnych narzędziach

4.2. Analiza porównawcza narzędzi do testów funkcjonalnych aplikacji internetowych

Narzędzia zostały porównane pod wieloma aspektami w skali od 1 do 5 kolejno oznaczając jako bardzo słaba ocena, słaba, średnia, dobra oraz bardzo dobra ocena. Porównanie zebrane w postaci tabeli.

Tabela 10: Porównanie narzędzi do automatyzacji testów

	SeleniumIDE	Selenium WebDriver	TestComplete	SahiPro	Katalon Studio
Początkowa instalacja	Wystarczy zainstalowanie rozszerzenia w przeglądarce 5	Potrzeba dodatkowego środowiska. Pobranie odpowiednich bibliotek 3	Wystarczy instalacja samodzielnego narzędzia 5	Instalacja narzędzia poprzez użycie odpowiednich komend 3	Pobranie spakowanego narzędzia i rozpakowanie go. Dodatkowa instalacja rozszerzenia w przeglądarce 4
Ilość dodatkowej konfiguracji	Po uruchomieniu wystarczy stworzyć projekt 5	Pobranie odpowiednich bibliotek, sterowników przeglądark 3	Wystarczy stworzenie projektu 5	Uruchomienie przeglądarki i kontrolera 5	Wystarczy stworzenie projektu 5
Czas wdrożenia	Intuicyjne GUI, pozwala na natychmiastowe tworzenie testów 5	Należy znać jeden ze wspieranych języków programowania 3	Nagrywanie testów nie wymaga większej wiedzy na temat używania narzędzia 4	Nagrywanie testów nie wymaga większej wiedzy na temat używania narzędzia 4	Nagrywanie testów nie wymaga większej wiedzy na temat używania narzędzia 4
Znajomość programowania	Wymaga podstawowej znajomości programowania przy edycji wygenerowanego skryptu 3	Wymaga zaawansowanej znajomości programowania 2	Wymaga podstawowej znajomości programowania przy edycji wygenerowanego skryptu 3	Do edytowania wygenerowanego skryptu wymaga podstawowej znajomości JavaScript 3	Wymaga podstawowej znajomości programowania przy edycji wygenerowanego skryptu 3
Generowanie raportu	Brak możliwości generowania raportu 1	Brak domyślnego mechanizmu, dostępne jedynie przez środowisko IDE 3	Możliwość wygenerowania raportu ze szczegółami 5	Możliwość wygenerowania raportu ze szczegółami 5	Możliwość wygenerowania raportu ze szczegółami 5
Nagrywanie scenariuszy	Jest to podstawowa funkcja narzędzia 5	Brak możliwości nagrywania 1	Jest to podstawowa funkcja narzędzia 4	Jest to podstawowa funkcja narzędzia 5	Jest to podstawowa funkcja narzędzia 5
Generowanie skryptów z nagranych scenariuszy	Po zapisaniu projektu można edytować plik, w którym znajduje się skrypt napisany w Selenese 2	Brak możliwości nagrywania 1	Możliwość wygenerowania skryptu z nagranych testów szczegółami 5	Możliwość wygenerowania skryptu z nagranych testów szczegółami 3	Możliwość wygenerowania skryptu z nagranych testów szczegółami 5
Możliwość ponownego	Można używać dany test	Jest możliwość wydzielenia	Jest możliwość wydzielenia	Jest możliwość wydzielenia	Jest możliwość wydzielenia

użycia skryptów	poprzez skopiowanie jego zawartości 4	odpowiednich metod 4	odpowiednich skryptów 4	odpowiednich metod 4	lenia odpowiednich metod 4
Integracja z innymi narzędziami	Brak możliwości integracji z innymi narzędziami 1	Możliwe integracje zależne od używanego środowiska 5	Domyślna integracja z Jira, CI/CD oraz repozytoriami GIT 4	Skomplikowane korzystanie z używanych bibliotek 2	Możliwość integracji z CI/CD, repozytoriami GIT i wieloma innymi 4
Średnia ocena	3,44	2,78	4,33	3,78	4,33

4.3. Analiza porównawcza narzędzi do testów REST API

W tabeli 11 zebrane zostało podsumowanie najważniejszych aspektów i funkcjonalności dla testowanych narzędzi REST API.

Tabela 11: Porównanie narzędzi do testów REST API

	Katalon Studio	Postman	SoapUI
Czas wytworzenia testu	Szybkie, nieskomplikowany interfejs.	Bardzo szybkie, tworzenie testów jest proste i intuicyjne.	Stworzenie testu wymaga dokładnego zapoznania się z interfejsem.
Znajomość programowania	Przydatne do weryfikacji poprawności odpowiedzi i pisania testów w Groovy, nie wymagane przez dostępność gotowych fragmentów kodów lub dodawania z poziomu odpowiedzi.	Przydatna, ale nie wymagana ze względu na możliwość generowania gotowych fragmentów kodu, skrypty tworzone w JavaScript.	Wymagane do tworzenia skryptów w Groovy.
Intuicyjność	Bardzo intuicyjny interfejs, duża ilość ułatwień i podpowiedzi.	Bardzo intuicyjny interfejs, pozwala na pracę bez konieczności wcześniejszego wdrożenia.	Wymaga poznania struktury aplikacji, mało intuicyjny interfejs.
Prostota / zaawansowanie	Łączy przyjazny interfejs z dużą liczbą funkcjonalności.	Bardzo proste narzędzie.	Narzędzie idealne dla bardziej zaawansowanych użytkowników.
Parametryzacja	Możliwość tworzenia zaawansowanych skryptów do zapytań i testów.	Konfiguracja środowisk uruchomieniowych z określonymi parametrami.	Parametryzacja na każdym poziomie testów.
Tworzenie testów automatycznych	Przy pomocy edytora lub skryptów.	Bardzo szybkie, przypisane do zapytania.	Testy grupowane i segregowane w określonej strukturze, tworzone przez asercje.
Jakość raportu z testów	Tylko podstawowe informacje.	Raport czytelny i szczegółowy.	Zawiera oczekiwane informacje, ale przedstawione w mniej przystępnej formie.
Integracja z innymi narzędziami	Największe możliwości integracji, już z poziomu aplikacji.	Szerokie możliwości integracji.	Integracja z użyciem SoapUI CLI.
Współdzielenie projektów	Wyłącznie przez GIT/SVN.	Możliwość współdzielenia przestrzeni roboczych z różnymi uprawnieniami, system komentarzy.	Brak.
Tworzenie serwisów mock	Brak.	Dostępne.	Dostępne.
Dodatkowe zalety	Narzędzie współpracuje z dużą ilością zewnętrznych aplikacji, nie jest dedykowane wyłącznie do testów serwisów REST.	Monitor – uruchamianie testów według określonego harmonogramu.	Testy bezpieczeństwa i obciążeniowe.

5. Wnioski

Celem niniejszego artykułu była analiza porównawcza rozwiązań wykorzystywanych w testowaniu automatycznym. Na podstawie przeprowadzonych badań należy podkreślić, że nie można wyznaczyć najlepszego rozwiązania do tworzenia zautomatyzowanych testów. Najważniejszym elementem wyboru odpowiedniego narzędzia, jest określenie indywidualnych wymagań, sposobu testowania produktu, parametryzacji testów, powtarzalności. Po ustaleniu tych elementów można przystąpić do wyboru.

Dla twórców własnego oprogramowania, którzy posiadają kontrolę nad używanymi funkcjonalnościami, lepszym rozwiązaniem może się okazać pisanie własnych testów przy wykorzystaniu Selenium WebDriver. Dostarcza ono możliwość integracji ciągłej integracji i wdrażania oprogramowania przez użycie używanego zintegrowanego środowiska deweloperskiego. Przez opcję definiowania danych, które mają zostać użyte podczas testów oraz stworzenia własnego systemu raportowania jest to narzędzie odpowiadające bardziej wymagającym zespołom.

Subiektywnie oceniając uniwersalnym narzędziem, które sprawdzi się w większości sytuacji i ma najmniej próg wejścia technicznego, będzie narzędzie oferowane przez firmę SmartBear. Kolejnym rozwiązaniem będzie zarówno dodatek do przeglądarki z narzędzia KatalonStudio oraz SeleniumIDE. Pisanie własnych rozwiązań przy użyciu Selenium WebDriver jest przeznaczone dla specyficznych rozwiązań, które przyciągną użytkowników, którzy chcą dowolnie kontrolować proces testowy. SahiPro jest to narzędzie oferujące najłabszą dokumentację, proces tworzenia testów oraz obsługę techniczną.

W przypadku testów serwisów internetowych nie jest możliwe jednoznaczne określenie rankingu testowanych aplikacji. Każda z nich posiadała zarówno funkcjonalności wyróżniające ją na tle pozostałych, jak i elementy, których brak zdecydowanie wpływał niekorzystnie na jej odbiór przez użytkownika. Wykonanie testów trzech aplikacji: Postman, SoapUI oraz Katalon Studio pozwoliło jednak na subiektywne stwierdzenie najważniejszych elementów przydatnych podczas tworzenia testów aplikacji REST:

- możliwość szybkiego tworzenia zapytań,
- dostępność nieskomplikowanej metody automatyzacji i parametryzacji testów,
- generowanie czytelnych raportów z przeprowadzonych weryfikacji,
- możliwość integracji z innymi narzędziami.

Główną zaletą aplikacji Postman była prostota i przejrzystość interfejsu. Narzędzie to sprawdzi się idealnie podczas pracy w średnich zespołach testerskich, ponieważ umożliwia udostępnianie przestrzeni roboczych, a także pozwala na definiowanie zestawów zmiennych i parametrów dla różnych środowisk uruchomieniowych. Nie jest jednak najlepszym wyborem dla zaawansowanych użytkowników, ponieważ nie pozwala na zaawansowaną edycję skryptów. Z tym aspektem radzi sobie bardzo dobrze Katalon Studio, które umożliwia edycję zarówno zapytań, jak i testów. Jest jednak bardzo wąsko wyspecjalizowane i wymaga integracji z wieloma innymi narzędziami (takimi jak systemy kontroli wersji i współdzielenia projektów, generowania raportów, itp.). Nie pozwala również na tworzenie sztucznych API dzięki serwerom typu mock. SoapUI jest zdecydowanie narzędziem przeznaczonym dla bardziej zaawansowanych użytkowników. Świadczy o tym poziom skomplikowania interfejsu oraz tworzenia testów automatycznych. Z kolei dużą zaletą tego narzędzia jest możliwość parametryzacji testów na różnych, niezależnych poziomach.

Literatura

- [1] Stowarzyszenie Jakości Systemów Informatycznych. Certyfikowany tester. Sylabus poziomu podstawowego certyfikatu ISTQB 2018. Wersja 1.0.1. 2019
- [2] R. Pawlak. Testowanie oprogramowania. Podręcznik dla początkujących. Helion, 2014.
- [3] <https://www.unity.pl/blog/dlaczego-innowacyjnego-projektu-nie-zrealizujesz-za-pomoca-modelu-waterfall/> [10.11.2019]
- [4] ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.
- [5] <https://smartbear.com/learn/automated-testing/best-practices-for-automation> [10.11.2019]
- [6] https://www.seleniumhq.org/docs/03_webdriver.jsp [10.11.2019]
- [7] <https://smartbear.com/product/testcomplete/features/> [10.11.2019]
- [8] <https://docs.katalon.com/katalon-studio/docs/overview.html> [10.11.2019]
- [9] <https://www.getpostman.com/product/workspaces> [10.11.2019]
- [10] <https://kordiankicza.pl/soapui-dla-zielonych-cz-1-tworzymy-pierwszy-projekt/> [10.11.2019]

Analysis of the vulnerability of IT system users to a phishing attack

Analiza podatności użytkowników systemów informatycznych na atak phishingowy

Szymon Grzebielec*

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

This article presents an analysis of users' vulnerability to phishing attacks. The study was carried out using a self-prepared attack. A phishing attack was carried out on a group of 100 people. The subjects were divided into two groups of 50 people. The first group was attacked from a private, trusted account. The second group was attacked from a foreign email address. The attacked people were asked to complete the survey, its results and conclusions are presented in this article.

Keywords: network security; network attacks; phishing

Streszczenie

W niniejszym artykule przedstawiono analizę podatności użytkowników na atak phishingowy. Badania dokonano za pomocą samodzielnie przygotowanego ataku. Atak phishingowy przeprowadzono na grupie 100 osób. Badanych podzielono na dwie grupy po 50 osób. Pierwszą grupę zaatakowano z prywatnego, zaufanego konta. Drugą grupę natomiast zaatakowano z obcego adresu e-mail. Zaatakowane osoby poproszono o wypełnienie ankiety, jej wyniki oraz wnioski przedstawiono w niniejszym artykule.

Słowa kluczowe: bezpieczeństwo sieci; ataki sieciowe; phishing

*Corresponding author

Email address: szymon.grzebielec@live.com (S. Grzebielec)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

W obecnych czasach można zauważyć szybki rozwój technologiczny oraz postępujący proces informatyzacji. Dotyczy to zarówno prywatnych przedsiębiorstw jak i administracji publicznej. Systemy teleinformatyczne są używane głównie do gromadzenia, przechowywania i przetwarzania danych. Ich szybkie i komfortowe użytkowanie jest możliwe między innymi za sprawą Internetu.

Rozwój Internetu oznacza, między innymi, że coraz większa liczba komputerów jest na stałe lub czasowo przyłączona do tejże sieci globalnej [1, 2]. Daje to określone korzyści, ale stwarza również i zagrożenia, których każdy użytkownik powinien być świadomy. Jednym z nieodłącznych ryzyk, które wiąże się z używaniem internetu jest możliwość naruszenia prywatności przez innych użytkowników sieci, którzy nie byli do tego uprawnieni. Zdobycie uprawnień do działania w określonym systemie komputerowym pracującym w sieci bez upoważnienia ze strony jego właściciela nosi nazwę „ataku”.

Istnieje wiele rodzajów ataków sieciowych. Mogą być to ataki pasywne oraz aktywne [3]. Do najpopularniejszych z nich należą SQL Injection, phishing, broken authentication and session management czy DDoS attack [4].

Według danych CERT Polska ok. 280 tys. komputerów dziennie jest atakowanych przez hakerów na różne sposoby [5]. W związku z tak częstym zagrożeniem kradzieży danych zasadne jest badanie najpopularniejszych ataków, ich możliwości oraz świadomości użytkowników i ich podatności na próby kradzieży danych.

1.1. Phishing

Słowo phishing często jest tłumaczone jako łowienie hasła, co pochodzi od angielskiego zwrotu password harvesting fishing. Phishing jest formą ataku opartą na oszukaniu ofiary. Do powodzenia ataku niezbędna jest znajomość inżynierii społecznej. Napastnik używa socjotechniki aby nakłonić atakowaną osobę do wykonania pewnych działań. Zazwyczaj chodzi o zalogowanie się z podanego linku lub pobranie złośliwego oprogramowania. W taki sposób wyłudzone są dane osobowe, loginy i hasła do kont bankowych, portali społecznościowych, komunikatorów internetowych czy informacje dotyczące kart kredytowych [3]. Zwykle atakujący udają kogoś innego. Mogą udawać znajomych lub kogoś z rodziny ale także przedstawicieli prywatnych firm lub instytucji państwowych. Ataki phishingowe można podzielić na trzy typy. Są to spear phishing, clone phishing oraz whaling.

Spear phishing jest to atak skierowany konkretnie pod wybraną osobę, grupę ludzi czy przedsiębiorstwo. Przed dokonaniem ataku napastnik zbiera jak najwięcej informacji dotyczących swojej ofiary [6, 7].

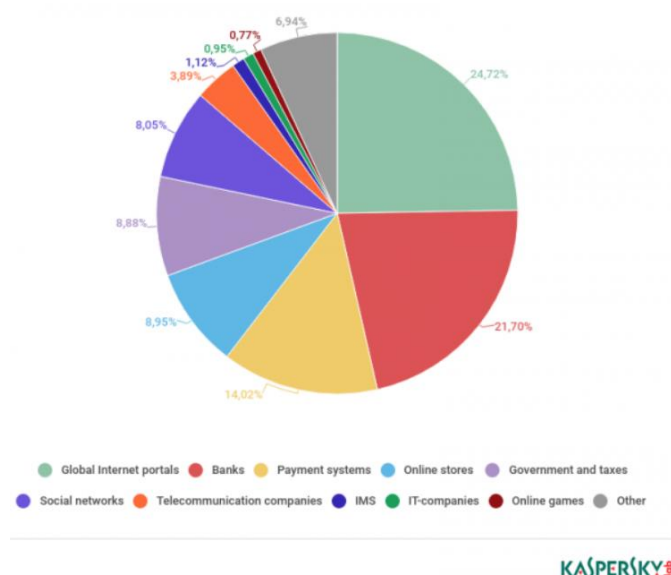
Clone Phishing jest takim rodzajem ataku phishingowego, w którym dostarczona, legalna wiadomość e-mail posiadająca załącznik lub link zostaje skopiowana. Skopiowana wiadomość jest używana do ataku a załącznik lub link zostaje zastąpiony złośliwym oprogramowaniem [8].

Whaling(ang. wielorybnictwo) jest to atak phishingowy spersonalizowany pod kadrę kierowniczą, managerów czy szefów firm lub korporacji. Zwykle jest to wiadomość e-mail

od osoby podszywającej się pod instytucję państwową czy kancelarię prawniczą [9].

Jedną z form phishingu jest przygotowanie strony internetowej imitującej inną stronę WWW na przykład banku lub portalu społecznościowego. Napastnik umieszcza na takiej stronie skrypt który przechwytuje dane logowania a następnie wysyła je na serwer lub adres e-mail atakującego.

Brak świadomości czym jest phishing, jak działa oraz brak umiejętności obrony przed tego typu atakiem może skutkować utratą wielu ważnych danych co z kolei może prowadzić do utraty pieniędzy. Według informacji udostępnionych na stronie www.prnews.pl tylko 36,7% badanych wie na pewno czym jest phishing [10]. Strona www.retruster.com udostępniła raport dotyczący phishingu na rok 2019 według, którego 90% naruszeń danych stanowi wyłudzenie informacji a liczba prób phishingu wzrosła w ciągu ostatniego roku o ok. 65% [11]. Jak podaje Kaspersky Lab najczęstszym obiektem ataków phishingowych w pierwszym kwartale 2019 roku były banki, stanowiły ponad 25% wszystkich ataków. W czołówce znajdują się również globalne portale internetowe oraz systemy płatnicze [12]. Na rys. 1. przedstawiono pełny wynik badania.



Rysunek 1: Rozkład organizacji narażonych na ataki phishingowe według kategorii w 1. kwartale 2019 r. [12].

Portal www.webroot.com podaje, że co miesiąc powstaje ok. 1,5 mln stron phishingowych [13]. Przytoczone badania pozwalają stwierdzić, że atak phishing jest niezwykle popularną oraz niebezpieczną formą ataku internetowego.

1.2. Metodyka badawcza

Badaniu został poddany atak phishingowy. Dokonano analizy tego ataku. W tym celu wykorzystano wcześniej przygotowaną stronę phishingową. Aby dokonać dokładniejszej analizy ataku phishingowego wykonano go na grupie stu osób. Badane osoby podzielono na dwie grupy po 50 osób. Pierwszą grupę zaatakowano z prywatnego, zaufanego konta. Było to konto na portalu społecznościowym Facebook użytkownika znanego badanym. Użyto socjotechniki wysyłając wiadomość o treści: "Cześć, biorę udział w konkursie i potrzebuję twojej pomocy. Wystarczy, że klikniesz w www.facebook-konkursy.eeu.pl i oddasz głos na mój komentarz. Z góry dziękuję za pomoc." Drugą grupę zaatakowano z obcego adresu e-mail, nieznanego badanym. Użyto socjotechniki

wysyłając wiadomość z adresu e-mail Konkursy.Facebook2019@gmail.com o treści: "Witaj, zostałeś wylosowany w naszym konkursie i masz szansę wygrać Iphone 10s. Wystarczy, że klikniesz w podany link i będziesz postępować według wskazówek. Nie czekaj klikaj w www.facebookkonkursy.eeu.pl i ciesz się nowym telefonem." Następnie badanych poinformowano o fakcie wykonania na nich ataku, poinformowano jak się przed nim zabezpieczyć oraz zapewniono ich, że ich dane nie zostaną nigdzie udostępnione a atak przeprowadzono tylko dla celów naukowych. Potem poproszono badanych o wypełnienie ankiety na temat przeprowadzonego ataku phishingowego. W ankiecie wzięły udział 93 osoby ze 100 zaatakowanych, 50 respondentów zaatakowano z zaufanego konta, natomiast 43 z obcego adresu e-mail.

1.3. Stanowisko badawcze

Do przeprowadzenia badań użyto komputera typu laptop o następujących parametrach:

- dostęp do Internetu,
- procesor Intel Core I5 2,3 GHz,
- pamięć RAM 8GB,
- karta graficzna NVIDIA GeForce GTX 1050, 4096 MB DDR5,
- system operacyjny Windows 10

2. Eksperyment

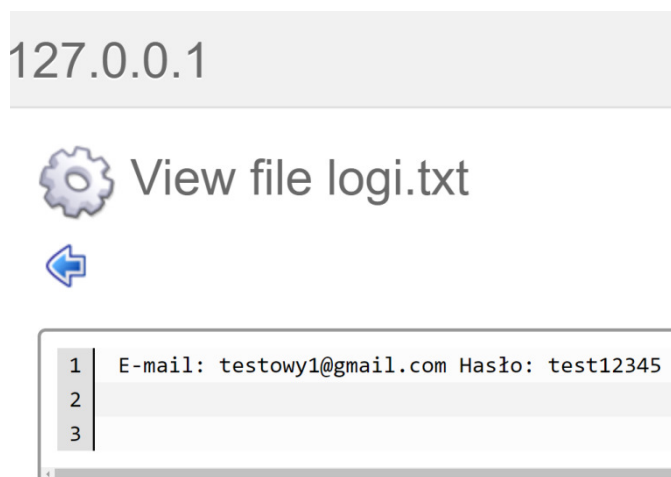
Celem przeprowadzenia badań naukowych było przeanalizowanie podatności użytkowników na atak phishingowy. Badanie przeprowadzono na podstawie samodzielnie wykonanego ataku. W przedstawionym eksperymencie dokonano ataku na portal społecznościowy Facebook. Według danych StatCounter Facebook we wrześniu 2019 roku miał udział w światowym runku mediów społecznościowych na poziomie 72,94% co czyni go najpopularniejszym medium społecznościowym na świecie. W Polsce ten odsetek wynosił 75,79% [14]. Zważając na ten fakt wybrano go do przeprowadzenia ataku.

W początkowej fazie eksperymentu przygotowano oraz przetestowano stronę internetową podszywającą się pod portal społecznościowy Facebook (rys. 2.). Strona została umieszczona na darmowym serwerze.



Rysunek 2: Strona phishingowa.

Po próbie logowania z utworzonej strony, dane logowania zostały wysyłane na serwer. Zapisywano je w pliku logi.txt. Na rysunku 3. pokazano wyniki testu logowania.



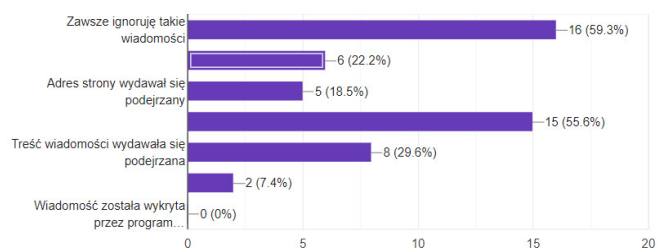
Rysunek 3: Plik logi.txt.

Po przeprowadzeniu testów wykonano atak na grupie 100 osób. Badanych podzielono na 2 grupy po 50 osób. Pierwszą grupę zaatakowano z prywatnego, zaufanego konta a drugą z obcego adresu e-mail. W wyniku przeprowadzonego ataku 8% badanych padło jego ofiarą. Następnie przeprowadzono ankietę wśród zaatakowanych osób. Respondentom zadano 12 pytań.

Według wyników ankiety 87,5% badanych spośród tych, którzy padli ofiarą ataku zostało zaatakowanych z prywatnego konta. Używając do ataku obcego adresu e-mail napastnik podszywał się pod organizatora konkursu. Badanie pokazało, że 59,3% respondentów ignoruje takie wiadomości a 22,2% badanych w ogóle ignoruje wiadomości od obcych (rys. 4.)

8. Jeśli na poprzednie pytanie odpowiedziałeś tak pomiń to pytanie. Dlaczego nie kliknąłeś w podany link?

27 responses

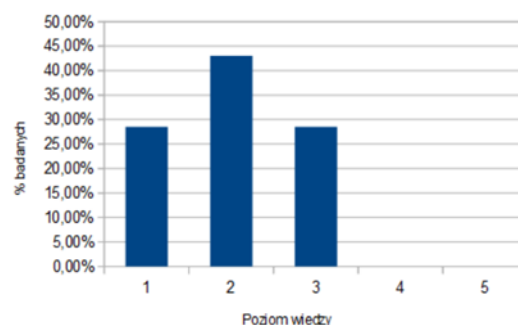


Rysunek 4: Wyniki ankiety na temat przeprowadzonego ataku phishingowego z obcego adresu e-mail.

Pozwala to stwierdzić, że większe prawdopodobieństwo ataku phishingowego istnieje przy użyciu zaufanego konta. Używanie obcego adresu e-mail może być nieskuteczne ze względu na ignorowanie wiadomości przez odbiorców.

Śród osób, które padły ofiarą ataku tylko 14% wiedziało czym jest phishing natomiast 86% osób nie widziało lub nie było pewne. Respondentów zapytano jak oceniają swoją wiedzę z zakresu bezpieczeństwa systemów informatycznych w skali od 1 do 5, gdzie 1 to brak wiedzy natomiast 5 to ekspert. Odpowiedzi osób, które padły ofiarą ataku pokazano na rysunku 5.

Jak oceniasz swoją wiedzę z zakresu bezpieczeństwa systemów informatycznych

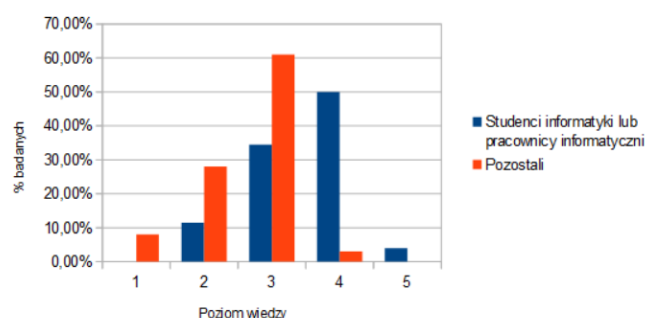


Rysunek 5: Poziom wiedzy osób, które padły ofiarą ataku.

Jak widać na rysunku 5. poziom wiedzy jest niski. Ponad 70% respondentów oceniło go jako poniżej przeciętnej. Żadna z badanych osób nie uważa, że poziom jej wiedzy na temat bezpieczeństwa systemów informatycznych jest wysoki. Wynik badania pozwala twierdzić, że brak wiedzy na temat phishingu oraz niski poziom ogólnej wiedzy z zakresu bezpieczeństwa systemów informatycznych ma wpływ na wyższą podatność użytkownika na atak phishingowy.

Respondentów zapytano czy studiują informatykę lub pracują na stanowisku informatycznym. Twierdząco odpowiedziało 28% z nich. Wśród osób, które padły ofiarą ataku nie znalazł się ani jeden student informatyki lub pracownik stanowiska informatycznego. 80,7% studentów informatyki i pracowników informatycznych deklaruje, że wie czym jest phishing. Wśród pozostałych natomiast taką wiedzę deklaruje jedynie 20,9%. Studenci informatyki i osoby pracujące na stanowiskach informatycznych znacznie lepiej oceniają swoją wiedzę z zakresu bezpieczeństwa systemów informatycznych w porównaniu do pozostałych. Zostało to zaprezentowane na rysunku 6.

Jak oceniasz swoją wiedzę z zakresu bezpieczeństwa systemów informatycznych?



Rysunek 6: Porównanie poziomu wiedzy studentów informatyki lub pracowników IT oraz pozostałych osób.

Wyniki uzyskane w kwestionariuszu pokazują, że ponad połowa studentów informatyki lub pracowników stanowisk informatycznych deklaruje swój poziom wiedzy powyżej przeciętnej, 4% nawet uważa się za eksperta w dziedzinie bezpieczeństwa a tylko 11,5% z nich ocenia swój poziom wiedzy poniżej średniej. Sytuacja wygląda inaczej wśród pozostałych badanych. Większość z nich czyli 61% uważa swoją wiedzę za przeciętną a tylko 3% z nich ocenia swoją wiedzę na 4 w skali do 5. Nikt z nich nie uważa się za eksperta w tej dziedzinie. 8% z pozostałych badanych deklaruje kompletny brak wiedzy z zakresu bezpieczeństwa systemów informatycznych. Przytoczone wyniki badania pozwalają

twierdzić, że studia informatyczne oraz praca na stanowiskach związanych z informatyką ma znaczący wpływ na poziom wiedzy na temat bezpieczeństwa systemów informatycznych. Osoby te cechują się większą wiedzą z tego zakresu co jak wcześniej udowodniono przekłada się na podatność użytkownika na atak phishingowy. Z przeprowadzonych badań wynika, że im wyższa jest wiedza użytkownika na temat phishingu oraz ogólnego bezpieczeństwa systemów informatycznych tym jego podatność na atak phishingowy jest niższa.

Badanych pytano czy po przeczytaniu wiadomości kliknęli w podany link. Wśród osób, które zadeklarowały wiedzę powyżej średniej (4 albo 5 w skali od 1 do 5) odsetek osób, które kliknęły w proponowany link był niższy od 1%. Na pytanie dlaczego nie kliknęli w podany link ankietowani odpowiadali, że adres strony wydawał się podejrzany- 76% oraz, że treść wiadomości wydawała się podejrzana- 24%. Wynik badania pokazuje, że osoby posiadające wysoką wiedzę na temat bezpieczeństwa systemów informatycznych zachowują większą czujność podczas korzystania z Internetu co przekłada się na ich mniejszą podatność na atak. Wynik tego badania potwierdza wcześniej wyciągnięte wnioski

3. Wnioski

Celem badania prezentowanego w niniejszym artykule była analiza podatności użytkowników na atak phishingowy. W wyniku przeprowadzonego na 100 osobach ataku 8% badanych padło jego ofiarą.

Korzystając z internetu jesteśmy narażeni na liczne ataki mające na celu kradzież naszych danych, loginów, haseł, wiadomości. Możemy stać się ofiarą jednego z nich na przykład ataku phishingowego. Osiągnięcie pełnego bezpieczeństwa systemu informacyjnego jest niemożliwe jednak korzystając z sieci Internet należy zachować szczególną uwagę. Nie należy logować się za pomocą linków wysyłanych przez nieznanych nadawców. Logując się do systemu należy zwracać szczególną uwagę czy link, z którego się logujemy jest poprawny (np. nie zawiera literówek, nie różni się od oryginalnego adresu strony). Atak phishingowy wykorzystuje podatność ofiary na socjotechnikę dla tego zalecane jest zwracać uwagę na treść oraz nadawcę otrzymywanych wiadomości. Większość badanych zwraca uwagę na adres linku, co może być powodem uniknięcia ataku phishingowego.

Większe prawdopodobieństwo powodzenia ataku phishingowego istnieje przy użyciu zaufanego konta niż w przypadku obcego adresu. Może mieć na to wpływ fakt, iż wiele osób ignoruje wiadomości otrzymane od nieznanych nadawców.

Badane osoby, które padły ofiarą ataku phishingowego nie widziały lub nie były pewne czym on jest oraz deklarowały niską wiedzę na temat bezpieczeństwa internetowego. Wiedzę większą niż przeciętna (powyżej 3 w skali 1-5) na temat bezpieczeństwa internetowego deklarowały jedynie osoby studiujące informatykę lub pracujące na stanowisku informatycznym.

Literatura

- [1] <https://www.wirtualnemedial.pl/artykul/22-miliardy-urzadz-podlaczonych-do-internetu> [28.10.2019]
- [2] <http://www.benchmark.pl/aktualnosci/ile-osob-ma-dostep-do-internetu-na-swiecie-juz-ponad-4-miliardy.html> [28.10.2019]
- [3] <http://edu.pjwstk.edu.pl/wyklady/bdk/scb/main06.html> [28.10.2019]
- [4] <https://devpark.pl/pl/najpopularniejsze-wedlug-owasp-rodzaje-atakow-sieciowych-i-jak-sie-przed-nimi-zabezpieczyc/> [28.10.2019]
- [5] <https://www.cert.pl/> [28.10.2019]
- [6] <https://www.firmex.com/thedealroom/spear-phishing-whos-getting-caught/> [28.10.2019]
- [7] <https://www.secureworks.com/research/threat-group-4127-targets-google-accounts> [28.10.2019]
- [8] https://www.theregister.co.uk/2008/04/16/whaling_expedition_continues/ [28.10.2019]
- [9] <https://www.lifewire.com/what-is-whaling-2483605> [28.10.2019]
- [10] <https://www.prnews.pl> [28.10.2019]
- [11] <https://retruster.com/blog/2019-phishing-and-email-fraud-statistics.html> [28.10.2019]
- [12] https://www.securelist.pl/analysis/7483.spam_i_phishing_w_2018_r.html [28.10.2019]
- [13] <https://www.webroot.com> [28.10.2019]
- [14] <https://www.gs.statcounter.com/> [28.10.2019]

Contactless interfaces in virtual reality

Interfejsy bezdotykowe w wirtualnej rzeczywistości

Andriy Dragni*, Tomasz Szymczyk

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

This work presents the results research on interfaces enabling navigation in virtual reality. An original program presenting a typical museum gallery has been written. It implemented a number of different moving scenarios. The paper analyzes which of these methods is best for the user. The optimal speed was tested and determined, using the user feels comfortable and does not lose the sense of control over the program.

Keywords: virtual reality; contactless interfaces; Cardboard glasses

Streszczenie

W niniejszym artykule przedstawiono wyniki badania interfejsów umożliwiających poruszanie się w wirtualnej rzeczywistości. Został napisany autorski program prezentujący typową muzealną galerię. Zaimplementowano w nim szereg różnych scenariuszy poruszania się. W pracy dokonano analizy, który z tych sposobów jest najlepszy dla użytkownika. Zbadano i ustalono optimum prędkości poruszania się, przy której użytkownik czuje się komfortowo i nie traci poczucia panowania nad programem.

Słowa kluczowe: wirtualna rzeczywistość; interfejsy bezdotykowe; okulary Cardboard

*Corresponding author

Email address: andrij.dragni@gmail.com (A. Dragni)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Wirtualna rzeczywistość (ang. Virtual Reality) rozwija się bardzo dynamicznie. Każdego roku są tworzone nowe narzędzia do prezentacji świata wirtualnego. Ponadto opracowywane i tworzone są zupełnie nowe urządzenia i interfejsy służące do przemieszczania się i interakcji z VR [1, 2]. Coraz silniej akcentuje się zjawisko immersji, czyli „zanurzenia się”, zatracenia w cyfrowym świecie. Chociaż pierwsze urządzenie techniczne z dziedziny VR było opracowane jeszcze w latach sześćdziesiątych przez Ivana Sutherlanda, a po latach dziewięćdziesiątych ubiegłego wieku rynek VR bardzo się załamał, to obecnie „święci triumfy”. Należy jednak stwierdzić, że jest to technologia bardzo kosztowna. Niezbędny jest zarówno bardzo wydajny (zwłaszcza graficznie) komputer a do tego kolejnym wydatkiem są okulary prezentujące grafikę w wysokiej rozdzielczości (ang. High Definition). Pomimo tego, dzięki szybko rosnącemu rynkowi odbiorców, ceny regularnie obniżają się a technologie są coraz to bardziej wymyślne i zaawansowane.

Nieustannie rozwija się także dziedzina wykorzystania wirtualnej rzeczywistości. Początkowo była to domena wojska - symulatorów, zwłaszcza lotniczych. Bardzo intensywnie można było zauważyć pierwiastki VR w filmach jako stosunkowo „proste i względnie tanie” narzędzie do tworzenia bardzo efektownych efektów specjalnych. Kolejnym miejscem dla tej technologii był szeroko pojęty marketing i reklama. VR pojawia się także w muzealnictwie, edukacji, a ostatnio także w medycynie.

Wszechobecność, ekspansywność i silnie nakreślone kręgi rozwoju zmuszają programistów do tworzenia

nowych interfejsów komunikacji człowiek - komputer właśnie w dziedzinie wirtualnych projekcji. W niniejszej pracy przedstawione zostały badania elementarnych, acz bardzo użytecznych, interfejsów służących do interakcji i umożliwiających poruszanie się w wirtualnej rzeczywistości. Skoncentrowano się na interfejsach bezdotykowych, czyli takich, które nie wymagają użycia fizycznych kontrolerów.

2. Rodzaje interfejsów

Na potrzeby pracy stworzono autorskie oprogramowanie umożliwiające badania ergonomii (naturalności) interfejsów. Zostało stworzonych pięć scenariuszy badawczych. Każdy scenariusz przedstawiał inny sposób poruszania się w wirtualnej rzeczywistości. Prędkość poruszania się i GUI (ang. Graphical User Interface) można zmieniać w czasie rzeczywistym.

Przetestowano następujące sposoby poruszania się w wirtualnej rzeczywistości:

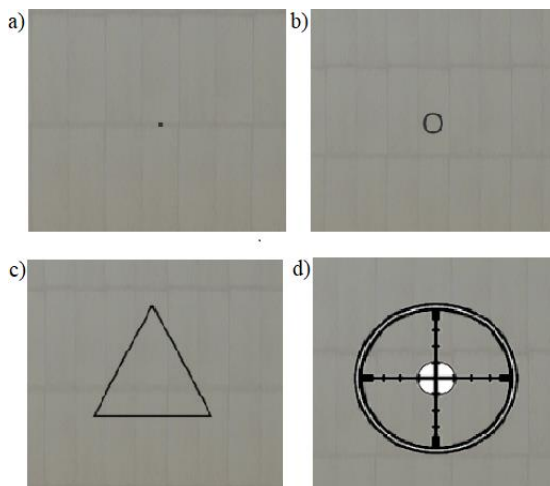
1. „Głowa do dołu – ruch do przodu”, żeby iść do przodu użytkownik musi pochylić głowę do dołu
2. „Głowa do dołu – ruch do przodu, głowa do góry – ruch do tyłu”, żeby iść do przodu użytkownik musi pochylić głowę do dołu, żeby iść do tyłu – podnieść w górę
3. „Idź tam gdzie patrzysz”, sterowana postać idzie do ostatniego punktu na podłodze gdzie patrzył użytkownik
4. „Teleportacja” (szybkie przemieszczanie się w punkt patrzenia), po minięciu określonego czasu (kiedy licznik się wyzeruje) postać sterowana przez użytkownika przemieszcza się w miejsce, w którym

skupił swój wzrok użytkownik (działa tylko kiedy użytkownik patrzy na podłogę)

5. „Idź po linii”, jedyny sposób poruszania się, gdzie użytkownik nie steruje postacią, ruch postaci jest wcześniej zaprogramowany, użytkownik może tylko zatrzymać się patrząc na obraz lub podłogę

W celu ograniczenia zjawiska choroby wirtualnej wprowadzono statyczny kursor [3]. Ma on także za zadanie ułatwianie orientacji i interakcji w VR.

Rodzaje kursorów które miały do wyboru osoby badane pokazane na rysunku 1.



Rysunek 1: Rodzaje kursorów.

3. Implementacja aplikacji

Postać, którą steruje użytkownik jest umieszczona w wirtualnej galerii sztuk i jest interaktywna. Obrót kamery postaci ma taką samą prędkość co obrót głowy użytkownika. Postać może się dowolnie obracać i patrzeć w każdym kierunku. Sposób poruszania się różni się w zależności od wybranego interfejsu.

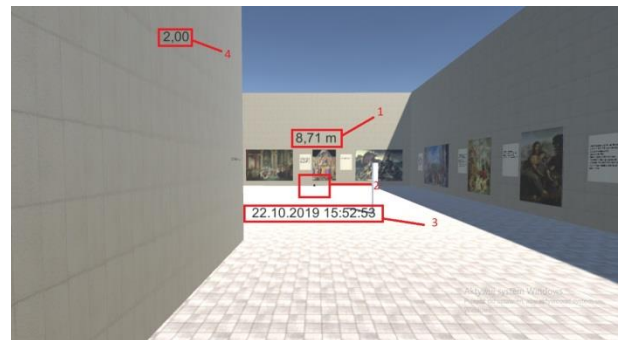
Na rysunkach 2 i 3 jest przedstawiony ten sam obraz w aplikacji z tą tylko różnicą, że rysunek 3 przedstawia co jest przesyłane do smartfonu, użytkownik widzi każdą półkę innym okiem, - obraz stereoskopowy. Administrator, nadzorujący badanie, widzi na swoim monitorze dokładnie to samo co badany ale jako obraz 2D. Pokazano to na rysunku 2. Rysunek 4 przedstawia opis interfejsu.



Rysunek 2: Wygląd aplikacji z poziomu komputera (administratora), przedstawiona galeria sztuki.



Rysunek 3: Wygląd aplikacji z poziomu smartfonu (użytkownika), przedstawiona galeria sztuki.



Rysunek 4: Tłumaczenie interfejsu gdzie 1 – odległość do obiektu wskazywanego przez kursor, 2 – miejsce położenia kursora, 3 – czas i data, 4 – prędkość.

Listing 1: Funkcja Update(), która odpowiada za przemieszczanie się i sterowanie kamerą postaci w scenariuszu „Głowa do dołu – ruch do przodu”.

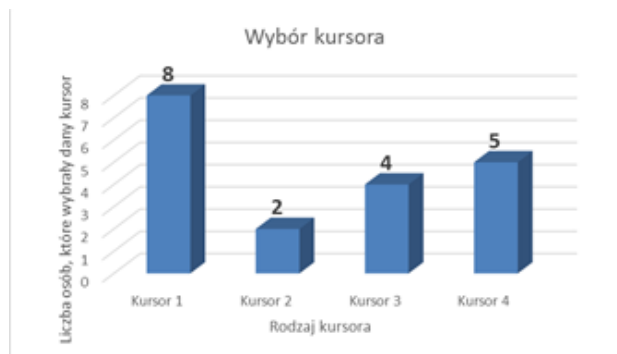
```
private void Update()
{
    speed = speed1 * 0.5f;
    if (Input.GetKeyDown(KeyCode.Equals))
        speed1++;
    if (Input.GetKeyDown(KeyCode.Minus))
        speed1--;
    if (Input.GetKeyDown(KeyCode.P))
        speed1 = 0;
    Speed.text = speed1.ToString("F");
    float horizontalRotation = Input.GetAxis("Mouse X")*2;
    transform.Rotate(0, horizontalRotation, 0);
    verticalRotation -= Input.GetAxis("Mouse Y")*5;
    verticalRotation = Mathf.Clamp(verticalRotation, -verticalRotationLimit, verticalRotationLimit);
    Camera.main.transform.localRotation = Quaternion.Euler(verticalRotation, 0, 0);
    WhatYouSee();
    if (verticalRotation > stopien)
    {
        transform.position += transform.forward * speed * Time.deltaTime;
    }
}
```

Na listingu 1 przedstawiona jest część skryptu zawierająca główną funkcję aplikacji która umożliwia przemieszczanie i poruszanie się postaci w każdym kierunku. Dla każdego sposobu przemieszczania się jest napisany oddzielny skrypt zawierający daną funkcję z różnicą w sposobie przemieszczania się [4-8].

4. Wyniki badań

Badania przeprowadzono na grupie składającej się z dwudziestu osób. Byli to studenci informatyki w wieku od 19 do 25 lat. Przeprowadzono serię badań. Pierwsze badanie polegało na tym, żeby osoba badana wska-

zała najbardziej ergonomiczny i użyteczny dla niej kursor. Badani mogli wybierać kursor spośród kursorów pokazanych na rysunku 1.



Rysunek 5: Wybór kursora przez użytkownika.

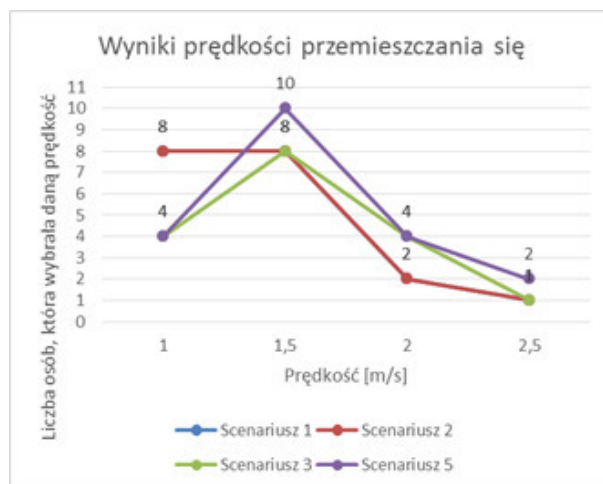
Jak pokazano na rysunku 5, większość osób wybrało jako kursor kropkę (Rys.1a), ponieważ nie zasłania obrazu, dobrze ją widać i łatwo trafić w punkt, który ciekawi użytkownika. Większość badanych miała uwagi do danego kursora, uważając, że kursor „kropka” powinien różnić się kolorem od otoczenia, ponieważ np. czarny kursor jest słabo widoczny, kiedy użytkownik patrzy na obiekt, który znajduje się w cieniu.



Rysunek 6: Wybór sposobu przemieszczania się.

Następne badanie dotyczy sposobów poruszania się i polega na wyborze sposobu poruszania się i doboru optymalnej prędkości, z którą użytkownikom jest komfortowo poruszać się po scenie. Ponad połowa badanych (55%) wybrała sposób przemieszczania się „Idź tam gdzie patrzysz”, jest to przedstawione na rysunku 6. Według badanych ten sposób przemieszczania się był wygodny i najbardziej komfortowy. Nie wymusza on nieustannego skupienia się na sposobie poruszania. Wadą danego sposobu przemieszczania się jest słaba precyzja wyboru miejsca dążenia oraz brak możliwości zatrzymania się w dowolnym momencie.

Następny scenariusz badawczy miał na celu empiryczne wskazanie optymalnej i ergonomicznej prędkości. Badano prędkość poruszania się w zakresie od 0,5 m/s do 3 m/s. Celem było wybranie, z jaką prędkością użytkownik będzie mógł się swobodnie poruszać, w wirtualnym świecie, nie czując presji ani dyskomfortu. Rysunek 7 pokazuje, że w zależności od sposobu przemieszczania się prędkość, z którą użytkownikom jest komfortnie się poruszać się różni.



Rysunek 7: Wybór prędkości (wykres dla scenariusza 1 i 2 pokrywają się).

Z danych uzyskanych podczas badań wynika, że najwięcej badanych wybrało prędkość 1,5 m/s. Wyliczając średnią z uzyskanych odpowiedzi należy wskazać, że średnia prędkość, z którą użytkownicy nie odczuwali dyskomfortu podczas przemieszczania się, dla scenariusza pierwszego i drugiego wynosi 1,4 m/s, a dla scenariusza trzeciego i piątego jest to 1,6 m/s. Ogromny wpływ na to ma widoczność dokąd przemieszcza się postać. W pierwszym i drugim scenariuszu użytkownik aby móc się poruszać patrzył do dołu lub w górę, przez co nie widział dokładnie dokąd podąża. W tych scenariuszach wyraźnie badani preferują mniejszą prędkość, żeby nie zgubić się i kontrolować ruch w świecie wirtualnym. Trzeci i piąty scenariusz pozwalają poruszać się z większą wygodą dzięki lepszej widoczności, lecz z mniejszą precyzyjnością, sposoby przemieszczania się w danych scenariuszach pozwalają na szybsze pokonanie większego dystansu, przez co użytkownicy wybierali większą prędkość przemieszczania się.

W scenariuszu czwartym był odliczany czas, jaki potrzebują użytkownicy dla tego, żeby skupić się na miejscu do którego chcą się przemieścić za pomocą „teleportacji”. Wyniki przedstawiono na rysunku 8.



Rysunek 8: Wybór czasu skupienia się przed „Teleportacją”.

5. Wnioski

Z przeprowadzonych badań można wywnioskować, że dla wirtualnej rzeczywistości dobrymi parametrami ustawień są takie, które są bardzo zbliżone do rzeczywistości realnej. Zbędne GUI użytkownika, takie jak odległość i data, w większości nie przeszkadzają badanym, a nawet przydatne i pomagają lepiej orientować się w VR [9]. Wyliczona średnia prędkość przemieszczania się w wirtualnej rzeczywistości, jest od 1,4 m/s do 1,6 m/s, w zależności od sposobu przemieszczania się. W realnej rzeczywistości człowiek przemieszcza się średnio z prędkością 1,45 m/s [10]. Większość badanych osób wybrało kropkę jako kursor, ponieważ nie zasłania obrazu i pozwala na łatwe celowanie. Dane badania pokazały że kiedy świat wirtualny jest zbliżony do świata rzeczywistego, to i parametry poruszania się muszą być zbliżone. W innym przypadku osoba badana, lub inny użytkownik, który przemieszcza się w podobnym świecie wirtualnym z nienaturalnymi parametrami, może poczuć się źle i odczuwać chorobę wirtualną. Żaden z badanych nie wybrał interfejsu poruszania się po linii jako wygodny, ponieważ brak kontroli nad poruszaniem się postaci powoduje, że tracą orientację w świecie, co też skutkuje złym samopoczuciem. Wynika z tego że najbardziej dostosowany i wygodny interfejs w wirtualnej rzeczywistości powinien być w jak największej ilości aspektów podobny do świata realnego.

Literatura

- [1] Ł. Pelka, Ł. Podstawka, T. Szymczyk, Analiza porównawcza gogli do VR, Journal of Computer Sciences Institute 10, 2019.
- [2] Ł. Rosiński, Raport dotyczący sytuacji na rynku VR, The Farm 51, 2018.
- [3] I. Plakhotniuk, M. Popko, T. Szymczyk, Health Aspects of Immersion in VR. Virtual Disease - Factor or Myth?. INTED 2019, 13th International Technology, Education and Development Conference.- 2019, s. 543-551
- [4] A. Kempa, T. Staś, Wstęp do programowania w C#: Łatwy podręcznik dla początkujących, Tomasz Staś, 2014.
- [5] P. Buchwald, Urządzenia mobilne w systemach rzeczywistości wirtualnej, Helion, 2018.
- [6] W. Malina, M. Szwoch, Podstawy projektowania interfejsów użytkownika, Helion, 2017
- [7] Dokumentacja Unity: <https://docs.unity3d.com> [04.2019]
- [8] Dokumentacja Unity dla VR: <https://developer.oculus.com> [dostęp: 04.2019]
- [9] Rzeczywistość wirtualna: https://pl.wikipedia.org/wiki/Rzeczywistość_wirtualna [10.2019]
- [10] Średnia prędkość chodu człowieka: https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D0%B4%D1%8C%D0%B1%D0%B0_%D1%87%D0%B5%D0%BB%D0%BE%D0%B2%D0%B5%D0%BA%D0%B0 [10.2019]

The system using mobile technologies to test reaction time

System wykorzystujący technologie mobilne służący do testowania czasu reakcji

Michał Woś*, Artur Zdziach

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

Reaction time to a light or sound stimulus is an important element of life. Constructing and testing a measuring device for collecting the response time stimulus focused on two study groups. The results which were obtained showed that people who specialize in sports have 0.15 seconds, better time than people who do not professionally deal with sport. Be tested for both groups with additional verification (stopwatch) confirmed the measuring apparatus along with the application.

Keywords: response time; mobile technologies; innovative solution

Streszczenie

Czas reakcji na bodziec świetlny lub dźwiękowy stanowi ważny element życia. Konstruując oraz testując urządzenie pomiarowe do zbierania czasu odpowiedzi na bodziec, skupiono się na dwóch grupach badanych. Wyniki jakie zostały uzyskane wykazały, że osoby, które specjalizują się w dyscyplinach sportowych mają 0,15 s lepszy czas od osób, które nie zajmują się zawodowo sportem. Wykonanie testu na obu grupach wraz z dodatkową weryfikacją (stoper), potwierdziło działania urządzenia pomiarowego wraz z aplikacją.

Słowa kluczowe: naprężenia; dynamika molekularna

*Corresponding author

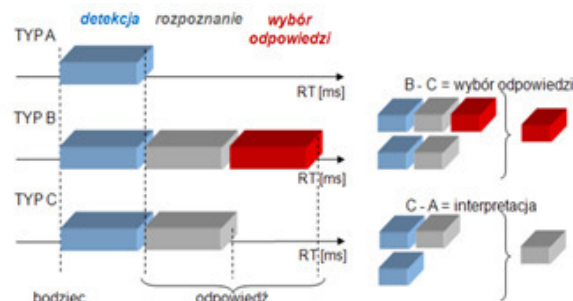
Email address: m.wos.official@gmail.com (M. Woś)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Od zarania dziejów człowiek był zmuszony walczyć o przetrwanie. Jednym z czynników które pomagały kiedyś polować, a dziś wykonywać swoją pracę, jest czas reakcji na odebrany bodziec. Obecnie refleks jest mierzony podczas badań u lekarza medycyny pracy. Podobne badania wykonuje się u sportowców. Współczynnik ten pomaga określić, czy ktoś jest zdolny do wykonywania pracy, gdzie niezbędna jest szybka odpowiedź układu bodźco – przewodzącego na reakcje na otoczenie np. wśród kierowców samochodów ciężarowych. Ważnym elementem treningu jest czas odpowiedzi na bodziec wśród sportowców uprawiających dyscypliny, w których ogromną rolę pełni szybkość reakcji. Czas reakcji na bodziec u zdrowego człowieka wynosi ok. od 180 ms do 200 ms. Udowodniono, że poprzez odpowiednie treningi można zmniejszyć czas reakcji człowieka. Badania dzięki zastosowaniu odpowiedniej metodyki i odpowiednich urządzeń pomiarowych pozwalają na zbadanie osób uprawiających dyscypliny sportowe. Potwierdza to artykuł pt: „Effect of yoga training on reaction time, respiratory endurance and muscle strength”, gdzie autor udowadnia, że ćwiczenia jogi wpływają na czas reakcji. Poprawa czasu reakcji jaką stwierdził Autor wynosiła od 30% do 50% u osób, które co trzeci dzień uczęszczały na lekcje jogi [1]. W czasie zbierania danych została wykorzystana metoda badawcza Dondersa typu A[2] (rys. 1). Reakcję typu A, określa się mianem reakcji prostej, gdzie podczas reakcji na bodziec badany reaguje jak najszybciej (np.

pojawienie się sygnału świetlnego i przyciśnięcie odpowiedniego przycisku) na jeden określony sposób.



Rysunek 1: Typy reakcji według Dondersa
[<http://www.wbc.poznan.pl/Content/185464/index.pdf>]

2. Materiały i metody

System pomiarowy do analizy czasu reakcji składa się z siedmiu elementów.

1. Urządzenie pomiarowe (rys. 2), w którego skład wchodzi odpowiednio komponenty:
 - komputer Raspberry Pi wraz z obudową;
 - płytką drukowaną wraz z zlutowanymi złączami jack 3,5 mm.
2. Pięć przycisków.
3. Zasilacz Raspberry Pi.

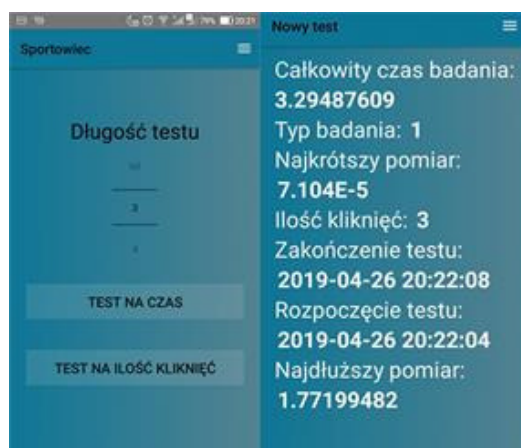
Przewaga rozwiązania przedstawionego w artykule nad rozwiązaniami obecnie dostępnymi na rynku polega na małym rozmiarze urządzenia, oraz modularności podłączenia przycisków pomiarowych. Taki rodzaj

podłączenia umożliwia uzyskanie różnych konfiguracji pomiarowych.



Rysunek 2: Z lewej urządzenie pomiarowe wraz z wyjściami jack 3,5 mm. Po prawej przycisk do zbierania pomiarów

Kolejna część systemu to aplikacja mobilna (rys. 3), która służy do wysyłania/odbierania żądania z urządzenia pomiarowego. Aplikacja pozwala na ustawienie urządzenia pomiarowego w dwóch trybach. Ponadto daje ona możliwość ustawienia długości pomiaru, użytkownik ma możliwość wybrania testu o długości od 3 do 60 kliknięć lub od 3 sec. do 60 sec. Pozwala to na dokładne sprawdzenie urządzenia pomiarowego, aplikacji mobilnej oraz umożliwia konfigurację rodzaju badania.



Rysunek 3: Przykładowy wygląd interfejsu aplikacji pomiarowej

Metoda badawcza jaka została zastosowana w czasie badania udowodniła poprawność działania urządzenia pomiarowego. Badanie polegało na wykonaniu 2 serii, każdorazowo po 10 pomiarów czasu reakcji na bodziec. Osoby w czasie badania były w pozycji siedzącej oddalonej od biurka na odległość ok. 50 cm. Przyciski do pomiaru były ustawione w linii prostej przed badanym,

linia miała długość 80 cm. Osoby badane praworęczne posługiwały się tylko prawą ręką w czasie badania, natomiast leworęczne lewą ręką. Grupa badawcza która została przebadana przez rozwiązanie liczyła 20 osób, w tym 10 osób, którzy nie zajmują się zawodowo sportem (6 zawodników siatkówki, 4 zawodników tenisa ziemnego). Kolejne 10 osób to osoby nie zajmujące się sportem. Zakres prac obejmował następujące zagadnienia:

- przegląd literatury na temat metod testowania czasu reakcji na bodziec u sportowców;
- przegląd literatury na temat sposobów uzyskiwania danych w czasie pomiarów czasu reakcji sportowców;
- stworzenie innowacyjnego projektu urządzenia pomiarowego;
- wybór składników systemu: mikrokontrolerów, przycisków, platformy oraz metod przechowywania wyników pomiaru;
- stworzenie urządzenia pomiarowego do rejestrowania danych z pomiarów czasu reakcji na bodziec;
- stworzenie aplikacji wraz z bazą danych umożliwiającej przetworzenie danych uzyskanych z pomiarów;
- opracowanie scenariuszy pomiarów czasu reakcji na bodziec w dowolnym miejscu za pomocą urządzenia jak i systemu.

3. Problem badawczy i cel badań

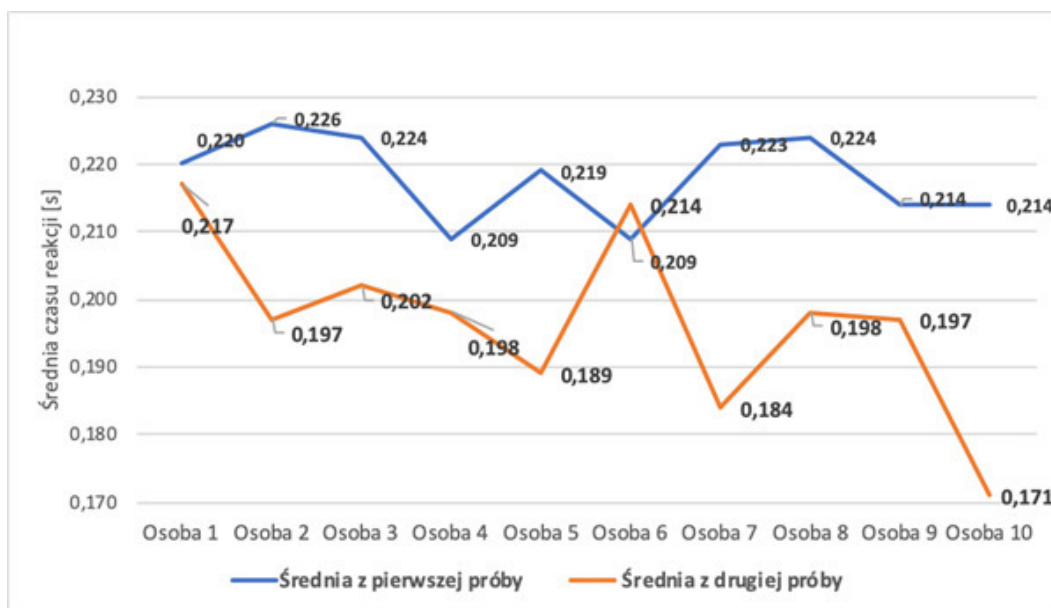
Głównym problemem badawczym było odpowiednie dostosowanie grupy badanej, tak by zweryfikować poprawność działania systemu. Osoby, które zostały wybrane do badania zostały podzielone na dwie grupy. Pierwszą grupę stanowili sportowcy uprawiający: tenis ziemny, siatkówkę oraz piłkę nożną. Do drugiej grupy należały osoby, które nie uprawiają żadnych dyscyplin sportowych. Głównym celem badań była weryfikacja poprawności działania urządzenia pomiarowego oraz wykazaniem różnic długości czasu reakcji między sportowcami a osobami, które nie uprawiają sportu.

4. Rezultaty

Po zebraniu wyników pomiarowych z obydwu prób zostały wykonane analizy zaprezentowane w tabelach 1-6 oraz na rysunkach 4-10.

Tabela 1: Średnie czasu reakcji dla grupy osób nie zajmujących się sportem

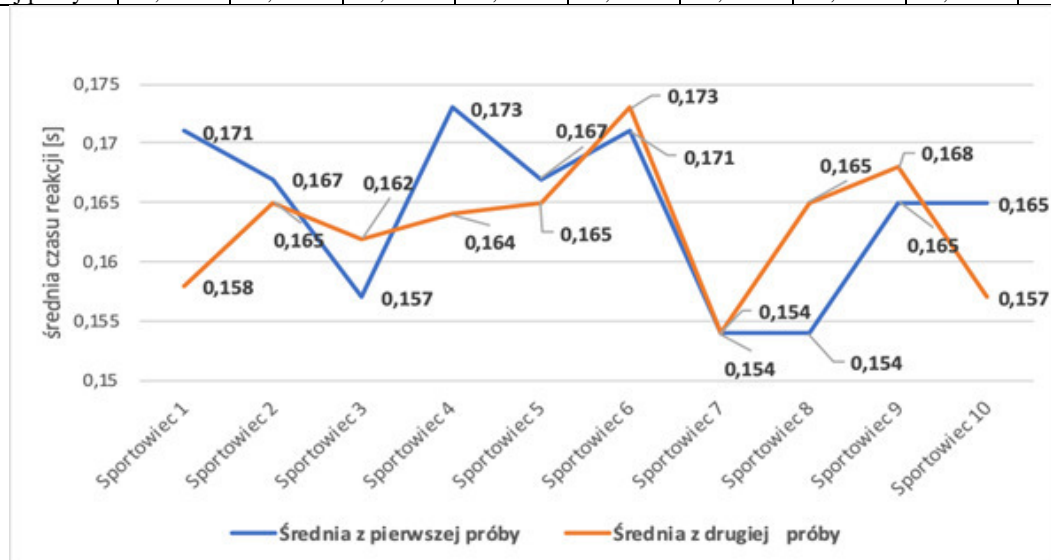
	Osoba 1	Osoba 2	Osoba 3	Osoba 4	Osoba 5	Osoba 6	Osoba 7	Osoba 8	Osoba 9	Osoba 10
Średnia z pierwszej próby	0,22s	0,226s	0,224s	0,209s	0,219s	0,209s	0,223s	0,224s	0,214s	0,214s
Średnia z drugiej próby	0,217s	0,197s	0,202s	0,198s	0,189s	0,214s	0,184s	0,198s	0,197s	0,171s



Rysunek 4: Wykres dla danych z tabeli 1

Tabela 2: Średnie czasu reakcji dla grupy sportowców

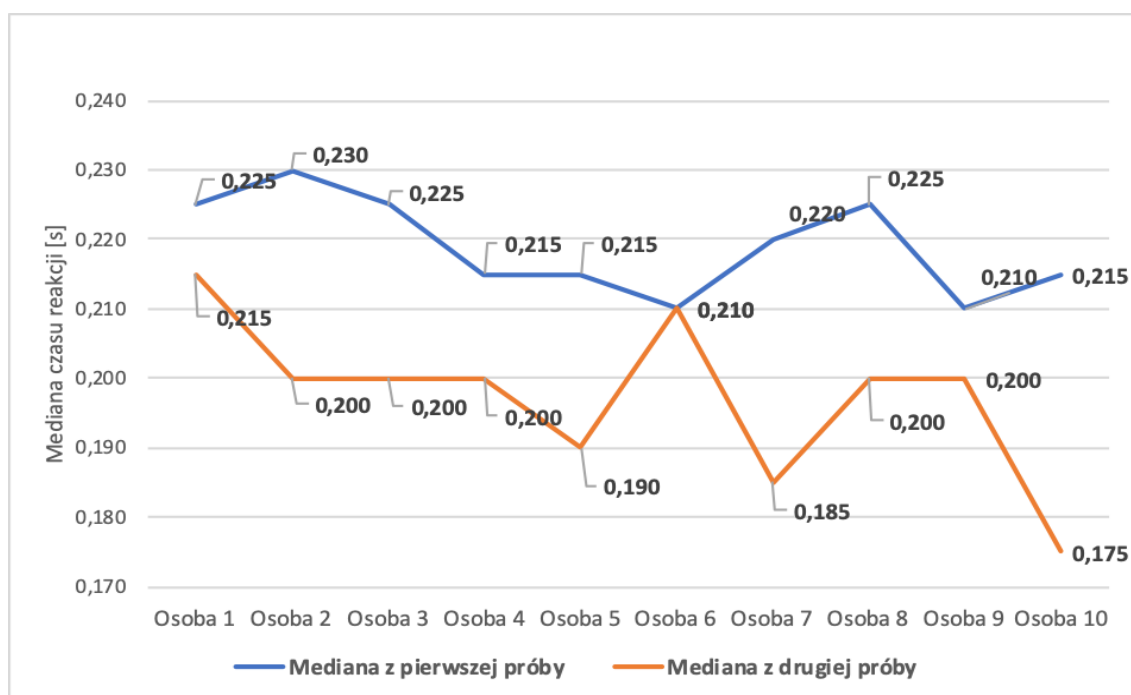
	Sportowiec 1	Sportowiec 2	Sportowiec 3	Sportowiec 4	Sportowiec 5	Sportowiec 6	Sportowiec 7	Sportowiec 8	Sportowiec 9	Sportowiec 10
Średnia z pierwszej próby	0,171s	0,167s	0,157s	0,173s	0,167s	0,171s	0,154s	0,154s	0,165s	0,165s
Średnia z drugiej próby	0,158s	0,165s	0,162s	0,164s	0,165s	0,173s	0,154s	0,165s	0,168s	0,157s



Rysunek 5: Wykres dla danych z tabeli 2

Tabela 3: Mediany czasu reakcji dla grupy osób nie zajmujące się sportem

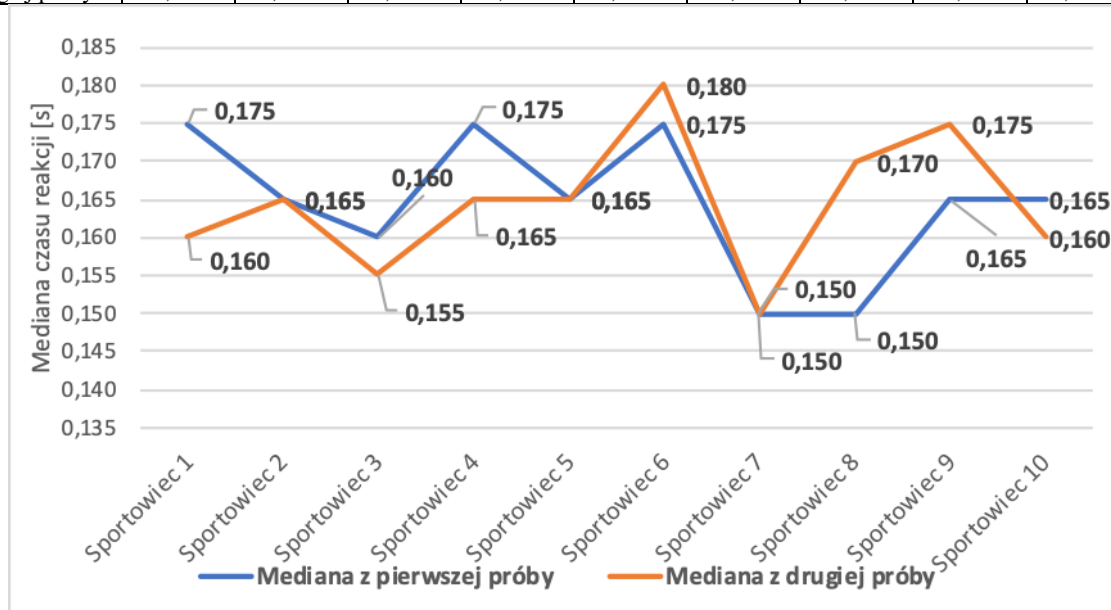
	Osoba 1	Osoba 2	Osoba 3	Osoba 4	Osoba 5	Osoba 6	Osoba 7	Osoba 8	Osoba 9	Osoba 10
Mediana z pierwszej próby	0,225s	0,23s	0,225s	0,215s	0,215s	0,21s	0,22s	0,225s	0,21s	0,215s
Mediana z drugiej próby	0,215s	0,2s	0,2s	0,2s	0,19s	0,21s	0,185s	0,2s	0,2s	0,175s



Rysunek 6: Wykres dla danych z tabeli 3

Tabela 4: Mediany czasu reakcji dla grupy sportowców

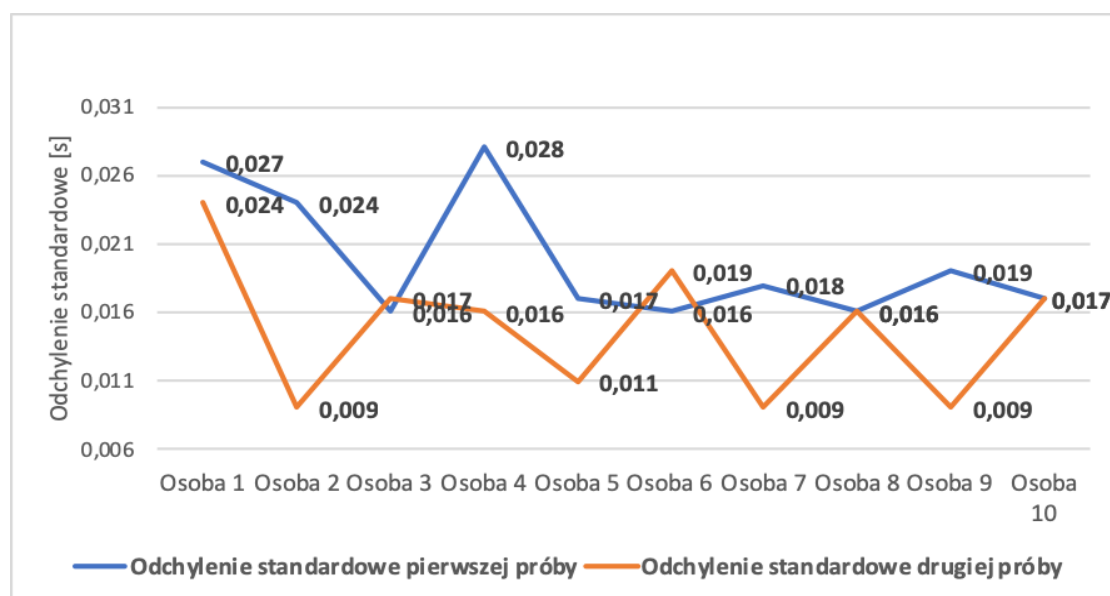
	Sportowiec 1	Sportowiec 2	Sportowiec 3	Sportowiec 4	Sportowiec 5	Sportowiec 6	Sportowiec 7	Sportowiec 8	Sportowiec 9	Sportowiec 10
Mediana z pierwszej próby	0,175s	0,165s	0,16s	0,175s	0,165s	0,175s	0,15s	0,15s	0,165s	0,165s
Mediana z drugiej próby	0,16s	0,165s	0,155s	0,165s	0,165s	0,18s	0,15s	0,17s	0,175s	0,16s



Rysunek 7: Wykres dla danych z tabeli 4

Tabela 5: Odchylenie standardowe czasu reakcji dla grupy osób nie zajmujące się sportem

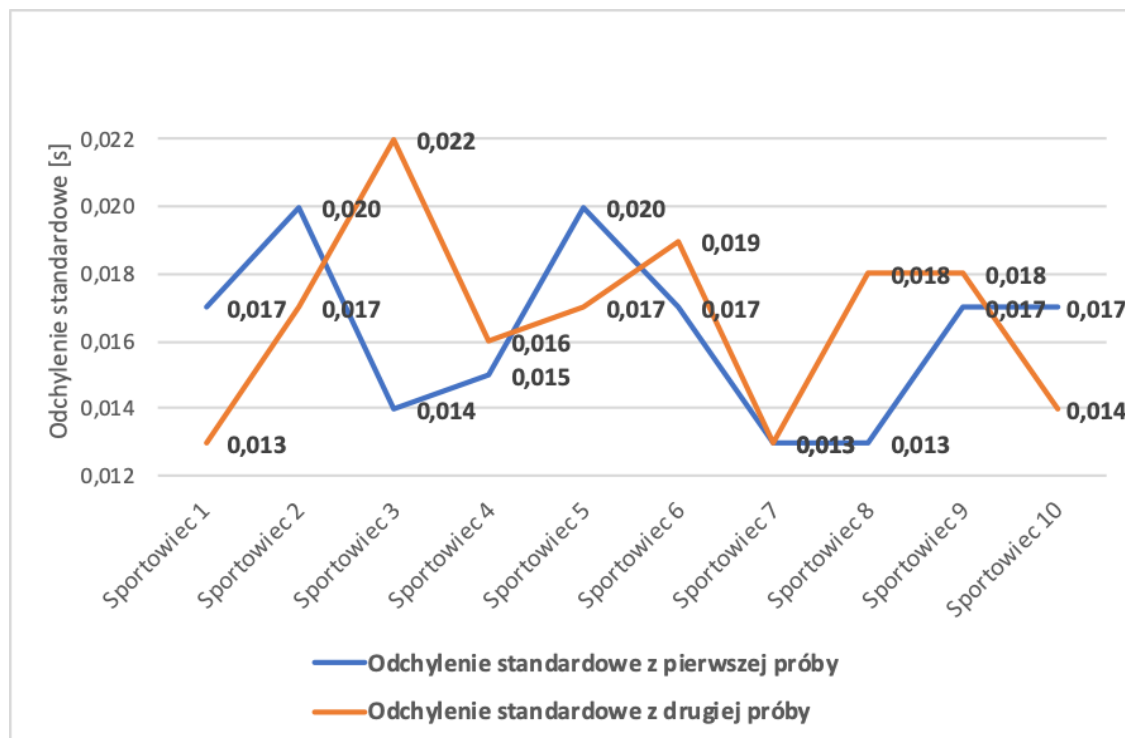
	Osoba 1	Osoba 2	Osoba 3	Osoba 4	Osoba 5	Osoba 6	Osoba 7	Osoba 8	Osoba 9	Osoba 10
Odchylenie standardowe pierwszej próby	0,027s	0,024s	0,016s	0,028s	0,018s	0,017s	0,019s	0,016s	0,02s	0,017s
Odchylenie standardowe drugiej próby	0,024s	0,009s	0,017s	0,016s	0,011s	0,019s	0,009s	0,016s	0,009s	0,017s



Rysunek 8: Wykres dla danych z tabeli 5

Tabela 6: Odchylenie standardowe czasu reakcji dla grupy sportowców

	Sporto- wiec 1	Sporto- wiec 2	Sporto- wiec 3	Sporto- wiec 4	Sporto- wiec 5	Sporto- wiec 6	Sporto- wiec 7	Sporto- wiec 8	Sporto- wiec 9	Sporto- wiec 10
Odchylenie standardowe z pierwszej próby	0,018s	0,021s	0,014s	0,016s	0,02s	0,018s	0,013s	0,013s	0,018s	0,018s
Odchylenie standardowe z drugiej próby	0,013s	0,018s	0,022s	0,016s	0,018s	0,019s	0,013s	0,018s	0,018s	0,014s



Rysunek 9: Wykres dla danych z tabeli 6

Analiza statystyczna wykazała istotne różnice w czasie uzyskiwanych pomiarów czasu reakcji na bodziec. Można zauważyć, że najkrótszy średni czas dla grupy osób nie zajmujących się sportem wyniósł w pierwszej próbie 0,209 s, natomiast w próbie drugiej 0,171 s. W grupie sportowców średni najkrótszy czas reakcji na bodziec wyniósł w pierwszej oraz drugiej próbie 0,154 s. Najwyższa mediana czasów jaka została zarejestrowana wśród badanych nie zajmujących się sportem wyniosła 0,21 s natomiast w drugiej próbie 0,185 s. Wśród osób specjalizujących się w dyscyplinach sportowych sytuacja wyglądała następująco: najkrótszy czas w pierwszej próbie wyniósł 0,15 s, natomiast w drugiej próbie było to 0,15 s. Odchylenie standardowe wykazało, że u osób wykonujących sport zawodowo, wartość odchylenia wokół średniej jest większa niż u sportowców, gdzie odchylenie jest mniejsze. Reasumując wyniki analizy statystycznej, można przyjąć, że osoby, które uprawiają sport zawodowo mają większe predyspozycje do szybszej odpowiedzi na bodziec.

5. Dyskusja

Urządzenia dostępne na rynku, które oferują pomiar czasu reakcji na bodziec, zazwyczaj zajmują dużą powierzchnię i nie są przeznaczone do pomiaru mobilnego. Dodatkowo takie urządzenia służą także do pomiaru czasu reakcji wśród zawodowych kierowców. Takiego rodzaju urządzenia badają odpowiednio czas reakcji według sprecyzowanych norm takich jak: reakcje poprawne, pominięte oraz spóźnione. Mierniki posiadają w praktyce dwa typy bodźców dźwiękowy oraz świetlny, lub są ze sobą połączone, dzięki czemu takie urządzenia ma możliwość weryfikacji oddziaływania na hybrydowe bodźce. Na rynku występują urządzenia firm Psychelab [3] oraz SIMFLEX [4]. Urządzenia te służą do pomiaru refleksu wśród sportowców. Posiadają jednak wadę, urządzenia te są duże i nieprzystosowane do łatwego demontażu i montażu. Urządzenie przedstawione w tym artykule jest przenośne z możliwością dowolnego rozmieszczenia przycisków. Rozwiązanie to zapewni dokładny pomiar czasu reakcji. Artykuł „Assessment of simple reaction time in badminton players” [5], potwierdza wyniki analizy, z danych pochodzących z testów stworzonego rozwiązania. Wskazano w czasie badania, że średni najszybszy czas wyniósł 0,15 s wśród grupy grających zawodowo w badmintona. Podobne czasy zostały uzyskane w czasie badań wraz z urządzeniem pomiarowym, gdzie w grupie sportowców średni najkrótszy czas wyniósł 0,154 s. Potwierdza to, że urządzenie pomiarowe wraz z aplikacją mobilną działa poprawnie.

6. Wnioski

Jak wykazano w artykule, istnieje możliwość skonstruowania urządzenia mierzącego czas reakcji w formie

mobilnej. Istotną kwestią jest jednak łatwość połączenia odpowiednich komponentów. Daje to możliwość łatwego sprawdzenia czasu reakcji wśród sportowców np. podczas zawodów na wyjeździe. Badania jakie zostały przeprowadzone potwierdziły, także poprawność zbierania danych pomiarowych przez system.

Literatura

- [1] B. Balakumar, D. P. Thrombe, T. K. Nambinarayanan, S. Thakur, A. Chandrase, N. Krishnamurthy, Effect of yoga training on reaction time, respiratory endurance and muscle strength, Pondicherry, Indian J Physiol Pharmacol, (1992), 229 - 233.
- [2] D. J. McKeefry, Neil R. A. Parry, Ian J. Murray, Simple reaction times in color space: the influence of chromaticity, contrast, and cone opponency. Investigative Ophthalmology and Visual Science, Investigative Ophthalmology and Visual Science, (2003), 2267–2276.
- [3] https://www.ru.nl/donders/vm-site/new-website/general/biography_fc/. [22 04 2019]
- [4] <http://www.psychelab.pl/aparatura/miernik-czasu-reakcji.html>. [2019 04 30]
- [5] Z. Bańkosz, H. Nawara, M. Ociepa, Assessment of simple reaction time in badminton players, Trends in Sports Sci. 1(20) (2013), 54–61.
- [6] <http://symulatorrefleksu.pl/symulator> [29 04 2019].
- [7] J. M. Wyma, E. W. Yund, T. J. Herron, B. Red, D. L. Woods, Factors influencing the latency of simple reaction time, frontiers in Human Neuroscience, (2015), 131-136.
- [8] B. Hommel, Inverting the Simon effect by intention: Determinants of direction and extent of effects of irrelevant spatial information, Psychological Research, (55) (1993), 270-279.
- [9] W. H. B. Prinz, Common mechanisms in perception and action: Attention and Performance, Oxford University, tom XIX, (2002).
- [10] J. R. Simon, J. D. Wolf, Choice reaction times as a function of angular stimulus-response correspondence and age, Ergonomics, (6) (1963), 99-105.
- [11] E. Hita, J. L. Gomez, L. Jiménez del Barco, J. Romero, Spatial and chromatic dependencies on visual reaction time, Journal of Optics, (1986), 197-202.
- [12] M. Green, “How long does it take to stop?” methodological analysis of driver perception-brake times, Transportation Human Factors, (2000), 195-216.
- [13] P. Milin, H. Baayen, Analyzing reaction times, International Journal of Psychological Research, (2010), 1-27.
- [14] K. Low, J. O. Miller, Motor processes in simple, go/no-go, and choice reaction time tasks: a psychophysiological analysis, Journal of Experimental Psychology: Human Perception and Performance, (2001) 266–289.

Comparative analysis of a selected version of the Symfony framework

Analiza porównawcza wybranych wersji szkieletu programistycznego Symfony

Michał Jusięga*, Mariusz Dzieńkowski

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

This article is about research during which selected versions of the Symfony programming framework were compared in terms of their performance. The following versions of the framework were analysed: 3.0, 3.1, 3.2, 3.3, 3.4 LTS, 4.0, 4.1, 4.2, 4.3 and 4.4 LTS. For the purpose of the research, a simple test application in PHP was developed in ten variants corresponding to selected versions of the framework and consisting of 17 fragments of code – methods in the class, each of which operates on one basic component of Symfony. The application prepared in this manner was subject to performance tests in a two-stage experiment. After the experiment, the quantitative analyses were conducted in which the following aspects were taken into consideration: the average values of execution times and the average amounts of memory usage for specific code fragments for individual versions of the Symfony framework components and the average time of execution and demand for memory for the entire tested application. The obtained results for each code fragment representing a given component were visualized in the forms of graphs. The performed analyses showed that the best version of the Symfony programming framework in terms of performance is version 4.1.

Keywords: Symfony; performance; memory usage

Streszczenie

Zrealizowano badania, podczas których porównywano pod kątem wydajności wybrane wersje szkieletu programistycznego Symfony. Analizie poddano następujące wersje tego szkieletu: 3.0, 3.1, 3.2, 3.3, 3.4 LTS, 4.0, 4.1, 4.2, 4.3 oraz 4.4 LTS. Na potrzeby badań, opracowano prostą aplikację testową w języku PHP, w 10-ciu wariantach odpowiadających wybranym wersjom frameworka, składającą się z 17-stu fragmentów kodu - metod w klasie, z których każda operuje na jednym komponencie Symfony. Przygotowaną w ten sposób aplikację poddano testom wydajnościowym, w dwuetapowym eksperymencie. Po zakończeniu eksperymentu przeprowadzono analizy ilościowe, w których wzięto pod uwagę uśrednione wartości czasów wykonywania poszczególnych fragmentów kodu dla określonych wersji komponentów szkieletu Symfony oraz średnie czasy wykonywania się i zapotrzebowania na pamięć fizyczną przez całą aplikację testową. Uzyskane wyniki, dla każdego fragmentu kodu reprezentującego dany komponent, zostały zwizualizowane w formie wykresów. Przeprowadzone analizy wykazały, że najlepszą pod względem wydajnościowym wersją szkieletu programistycznego Symfony jest wersja 4.1.

Słowa kluczowe: Symfony; wydajność; użycie pamięci

*Corresponding author

Email address: michal.jusiega@gmail.com (M. Jusięga)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Podczas tworzenia aplikacji internetowych, wybór oprogramowania i jego wersji ma ogromny wpływ na optymalne działanie tworzonego systemu oraz ewentualne możliwości jego rozwoju. Decydować o tym może szereg wspieranych standardów kodowania, dostępność aktualizacji, których kolejne wersje zawierają nowe funkcjonalności oraz poprawki związane przede wszystkim ze zwiększaniem bezpieczeństwa. Od początku powstania tego frameworka duży nacisk kładziono na kwestie związane z jego bezpieczeństwem, wydajnością i szybkością. Dzięki temu Symfony jest obecnie jednym z najpopularniejszych, najbezpieczniejszych i najszybszych szkieletów programistycznych PHP [1].

Symfony, oprócz tego, że jest frameworkiem, jest jednocześnie zbiorem komponentów, na bazie których w efektywny sposób można realizować zaawansowane aplikacje internetowe [2]. Komponenty są odpowie-

dzialne za realizację poszczególnych zadań. Pracując nad określoną aplikacją, programista dobiera i dostosowuje odpowiednie komponenty w taki sposób, aby spełniały jej wymagania.

Na tle innych szkieletów programistycznych PHP, Symfony wyróżnia się silnie prowadzonym i przestrzegany standardem procesu wprowadzania nowych wersji [3]. Standard ten definiuje wewnętrzne procedury ustalone wewnątrz zespołu doświadczonych programistów pracujących nad rozwojem frameworka, określające w jakim okresie czasowym poprawki lub nowe funkcjonalności mogą zostać dopuszczone do użytku przez społeczność korzystającą z tego szkieletu [4]. Opis całego procesu i plany wydawania oraz utrzymywania kolejnych wersji znajdują się na stronie frameworka [5].

W przypadku poprawek bezpieczeństwa, takie aktualizacje mogą występować średnio co miesiąc, a w zależności od złożoności błędu nawet codziennie. Wykonanie czynności aktualizacji jest zazwyczaj bez-

pieczne dla aplikacji i z reguły programista nie musi modyfikować własnego kodu źródłowego, aby dostosowywać się do wprowadzonych zmian. Takie aktualizacje są bardzo zalecane.

Kolejny przypadek aktualizacji określa standard wprowadzenia zazwyczaj nowych funkcjonalności do aplikacji. Zawierają one również poprawki błędów i wydawane są co pół roku: w maju oraz w listopadzie. Tego typu aktualizacje oprogramowania są dla programisty impulsem do podjęcia działań, polegających na dostosowywaniu swojego kodu do zachodzących zmian. Takie postępowanie uważane jest jako dbałość o bezpieczeństwo.

Ostatni możliwy przypadek aktualizacji komponentów frameworka Symfony, który zachodzi raz na dwa lata dotyczy aktualizacji całego frameworka. Zazwyczaj tego typu aktualizacja zawiera ogromne ilości zmian, które nie są wstecznie kompatybilne z poprzednimi wersjami. W tym przypadku dokonanie aktualizacji wymaga dużej uwagi i determinacji ze strony programisty oraz włączenia się w szczegóły wprowadzonych zmian kodu źródłowego aplikacji.

Istotną, a zarazem trudną do rozstrzygnięcia kwestią jest to, czy opłaca się aktualizować komponenty w celu zwiększenia szybkości działania aplikacji i jej optymalnego funkcjonowania, ponieważ to wiąże się często z dużymi kosztami związanymi z wysiłkiem i poświęconym czasem programisty. Z drugiej strony regularne aktualizowanie frameworka może sprawić, że istniejąca aplikacja będzie się skutecznie dostosowywać do ciągle zmieniających się potrzeb biznesu.

2. Cel i teza

Celem pracy jest zbadanie i ocena pod względem wydajności, a także wykorzystania pamięci RAM wybranych wersji frameworka Symfony począwszy od wersji 3.0 do 4.4 LTS włącznie [6].

Za tezę badawczą przyjęto stwierdzenie:

Aktualizacja komponentów szkieletu Symfony ma wpływ na wzrost ich wydajności, a tym samym przyczynia się do wzrostu wydajności całego frameworka.

3. Metoda badawcza

Eksperyment został zrealizowany na bazie dwóch scenariuszy. Pierwszy z nich dotyczył oceny wydajności poszczególnych komponentów wchodzących w skład badanych wersji Symfony. Jako miary do porównań w tym przypadku, użyto średniego czasu wykonania metody operującej na danym komponentcie. Drugi scenariusz dotyczył wyznaczenia średnich czasów wykonania całej aplikacji testowej składającej się z 17 metod, z których każda wykorzystuje inny komponent. Dodatkowo w tym scenariuszu dokonano pomiarów zapotrzebowania aplikacji na pamięć fizyczną. Testy odbywały się na opracowanej do tego celu, prostej aplikacji testowej. Narzędziem wykorzystanym do realizacji pomiarów: wyznaczania czasów i użycia pamięci był framework PHPBench.

3.1. Aplikacja testowa

Na potrzeby badań, opracowano prostą aplikację w języku PHP, w dziesięciu wariantach - odpowiadających wybranym wersjom frameworka, składającą się z siedemnastu fragmentów kodu - metod w klasie, z których każda operuje na jednym, podstawowym komponencie Symfony. W tabeli 1 znajduje się lista wszystkich, wykorzystanych w aplikacji, komponentów oraz syntetyczny opis metod wykonujących proste działania na poszczególnych komponentach.

Tabela 1: Opis metod operujących na wybranych komponentach

Komponent	Opis metody
Asset	Utworzenie obiektu klasy Package oraz wywołanie metody getUrl, generującej adres URL zasobu.
Console	Uruchomienie instancji aplikacji, na bazie której wyszukiwane są dwie podstawowe komendy: list i help.
CssSelector	Utworzenie obiektu konwertera, uruchamiającego parser przekształcający wyrażenie CSS do XPath.
Dependency- Injection	Utworzenie i kompilacja fabryki kontenera.
Crawler	Utworzenie obiektu klasy Crawler, na podstawie dokumentu HTML, podanego jako parametr konstruktora.
DotEnv	Inicjalizacja obiektu klasy DotEnv oraz parsowanie treści w schemacie klucz=>wartość.
Expression- Language	Inicjalizacja obiektu klasy ExpressionLanguage oraz wywołanie dwóch poleceń: wykonania i kompilacji podanego wyrażenia.
Finder	Utworzenie obiektu klasy Finder wyszukującego pliki w aktualnym katalogu.
Http- Foundation	Wyodrębnienie superglobalnych zmiennych, przekształcenie ich do postaci obiektowej i przekazanie do zmiennej \$request oraz przygotowanie obiektu klasy Response.
HttpKernel	Utworzenie kolekcji dostępnych kontrolerów, utworzenie żądania za pomocą elementów komponentu HttpFoundation, przygotowanie dyspozytora zdarzeń, analizatorów kontrolerów i argumentów i na ich podstawie utworzenie obiektu HttpKernela służącego do analizy żądania zawartego w obiekcie \$request.
Options- Resolver	Utworzenie obiektu klasy OptionResolver, przekazanie domyślnych parametrów, analiza danych za pomocą metody resolve.
Process	Utworzenie obiektu klasy Process i przekazanie składowych wiersza poleceń jako zmiennej typu tablicowego.
Routing	Utworzenie kolekcji 1000 unikalnych adresów URL oraz pobranie wylosowanego obiektu klasy Route.
Serializer	Utworzenie obiektu klasy stdClass, normalizacja obiektu do postaci JSON oraz denormalizacja zmiennej tablicowej do klasy stdClass.
Stopwatch	Utworzenie obiektu klasy Stopwatch, uruchomienie stopera, zatrzymanie stopera i wyznaczenie interwału czasowego.
Validator	Utworzenie obiektu klasy Validator, przekazanie wartości oraz tablicy ograniczeń, rzutowanie zmiennej do postaci znakowej.
Yaml	Użycie dwóch metod: do przekształcenia postaci tekstowej do postaci zmiennej oraz w drugą stronę.

3.2. Narzędzia badawcze

W podrozdziale 3.1 opisane zostały metody, z których każda operuje na jednym komponencie frameworka Symfony i wykorzystuje dostępne możliwości tych komponentów. Metody te zostały przygotowane do celów badawczych – do przeprowadzenia pomiarów

czasów ich wykonywania się. Do analizy wydajności wykorzystano framework PHPBench [7], który został również napisany w języku PHP. Umożliwia on konstruowanie scenariuszy wydajnościowych w celu testowania całych aplikacji lub wybranych fragmentów kodu. Do podstawowych właściwości narzędzia PHPBench należą:

- przebiegi – cykl (liczba) kolejnych wykonań wybranego fragmentu kodu,
- iteracje – liczba powtórzeń całego testu,
- izolacja procesu – uruchamianie testów w osobnych procesach,
- raportowanie – generowanie raportów w postaci kodu XML i przekształcenie go do czytelnej postaci,
- monitoring użycia pamięci fizycznej.

W przykładzie 1 (Listing 1) przedstawiono dwa polecenia uruchomienia testu wydajnościowego wykorzystującego PHPBench.

Listing 1: Polecenia uruchomienia testów wydajnościowych dla badanej wersji szkieletu Symfony

```
./vendor/bin/phpbench
run .\components\SymfonyXX\benchmarks\ --
bootstrap=. \components\SymfonyXX\vendor\autoload.p
hp --iterations=50 --revs=50 --report=aggregate --
dump-file=raporty/slow/symfonyXX.xml;
./vendor/bin/phpbench report --report=aggregate --
output=html --file=raporty/slow/symfonyXX.xml >
raporty/slow/symfonyXX.html;
```

Pierwsza komenda uruchamia test danej (XX) wersji Symfony, wykorzystującej opisaną w rozdziale 3 aplikację symulującą użycie wybranych komponentów. Każdą metodę podczas jednego testu wywołano 50 razy, co dało w sumie 2500 pomiarów czasów ich wykonania, ponieważ cały test powtarzano 50-krotnie. Dla tak ustawionej liczby powtórzeń, PHPBench wyznaczył średnie czasy trwania pojedynczego wykonania się każdej metody wchodzącej w skład aplikacji testowej. Za pomocą tego narzędzia obliczono również średni czas trwania całego testu dla poszczególnej wersji aplikacji testowej. Dodatkowo framework PHPBench umożliwił wykonanie pomiarów zapotrzebowania aplikacji na pamięć RAM.

Druga komenda, przedstawiona w przykładzie 1 uruchamia generator raportu, który przekształca otrzymane rezultaty z postaci XML do postaci HTML i prezentuje je w formie tabelarycznej.

3.3. Środowisko testowe

Stanowisko badawcze, wykorzystane do przeprowadzenia testów, przedstawia tabela 2.

Tabela 2: Opis stanowiska badawczego

Nazwa	Wartość
System operacyjny	Microsoft Windows 10 Home
Procesor	Intel(R) Core(TM) i7-4700HQ CPU @ 2.40GHz
Dysk	Crucial_CT250MX200SSD3 (250 GB, SATA-III)
Karta graficzna	Intel(R) HD Graphics 4600 (1 GB)
Pamięć RAM	2 x 8 GB DDR3-1600 DDR3 SDRAM
Karta sieciowa	Intel(R) Dual Band Wireless-AC 7260

4. Wyniki

Do badań wyselekcjonowano 17 komponentów dla dziesięciu wersji frameworka Symfony. Po wstępnej analizie otrzymanych wyników dokonano wyboru sześciu komponentów, w których od wersji 3.0 do 4.4 zaszły istotne zmiany, które w dużej mierze wpłynęły na ich wydajność. Wyniki dla tych komponentów są przedstawione w postaci wykresów, na których oś Y reprezentuje średni czas wykonania się jednej z siedemnastu metod (w mikrosekundach), natomiast na osi X znajdują się oznaczenia kolejnych wersji frameworka.

Rysunki 1-6 prezentują wyniki, dla wybranych wersji frameworka, średniego czasu wykonywania się kodu zawartego w pojedynczej metodzie operującej na danym komponencie. Z kolei rysunki 7 i 8 przedstawiają sumaryczne wyniki odnoszące się do:

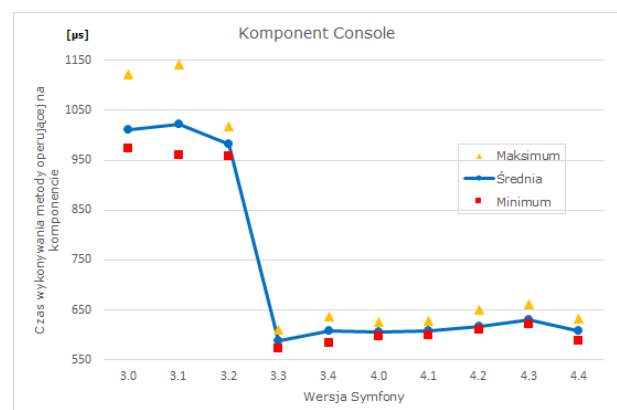
- średniego czasu wykonania się aplikacji testowej (wszystkich fragmentów kodu) dla poszczególnych wersji Symfony,
- średniego zapotrzebowania na pamięć RAM, podczas wykonywania aplikacji testowej zawierającej komponenty z poszczególnych wersji szkieletu programistycznego.

4.1. Wydajność wybranych komponentów

Komponent Console służy do tworzenia aplikacji, które są uruchamiane z poziomu wiersza poleceń. Aplikacje wykorzystujące ten komponent zachowują się tak, jakby były napisane w bash'u lub shell'u, dlatego jest on szczególnie przydatny do tworzenia synchronicznych lub asynchronicznych zadań uruchamianych bez użycia interfejsu graficznego.

Listing 2: Metoda testowa dla komponentu Console

```
final public function benchConsoleComponent(): void
{
    $application = new Application();
    $application->find('list');
    $application->find('help');
}
```



Rysunek 1: Wydajność komponentu Console

Na rysunku 1, od wersji 3.3 frameworka Symfony, widoczny jest znaczny wzrost wydajności komponentu Console. Trudno jest wytłumaczyć tę skokową poprawę, ze względu na brak oficjalnych informacji dotyczących

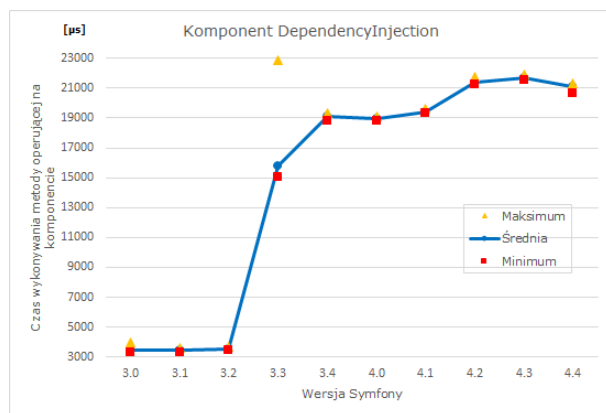
wprowadzanych ulepszeń odnoszących się do kwestii optymalizacyjnych.

Jednym z najważniejszych komponentów frameworka Symfony jest DependencyInjection. Można powiedzieć, że ten komponent jest mózgiem działania każdej aplikacji internetowej wykorzystującej mechanizm wstrzykiwania zależności. Wiąże on dostępne klasy oraz interfejsy (zarówno te napisane przez programistę jak i zewnętrzne biblioteki) i rejestruje je jako serwisy. Dzięki temu komponentowi zamiast wielokrotnego tworzenia obiektu tej samej klasy w różnych miejscach dla danego procesu następuje wstrzykiwanie zależności poprzez konstruktor lub za pośrednictwem dowolnej metody dostępowej, przekazującej argumenty do obiektu. Komponent ten implementuje interfejs PSR-11, który określa standard działania wielu aplikacji.

W związku z tym jest on bardzo ważny w podstawowym funkcjonowaniu każdej opracowanej aplikacji i dlatego jego wydajność ma kluczowe znaczenie.

Listing 3: Metoda testowa dla komponentu DependencyInjection

```
final public function
    benchDependencyInjectionComponent(): void
{
    $containerBuilder = new ContainerBuilder();
    $containerBuilder->register(__CLASS__,
        __CLASS__)->setPublic(true);
    $containerBuilder->setParameter('key', 'value');
    $containerBuilder->compile();
    $containerBuilder->get(__CLASS__);
}
```



Rysunek 2: Wydajność komponentu DependencyInjection

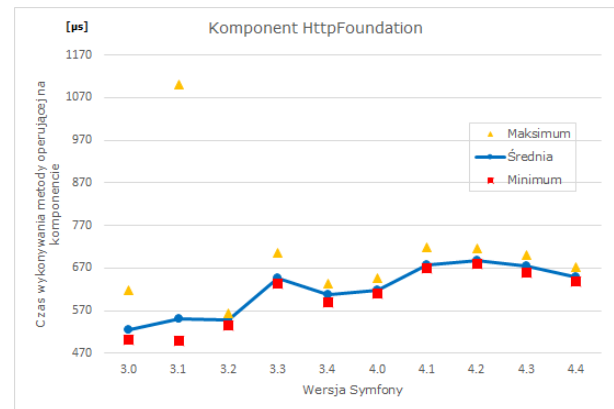
Widoczny na rysunku 2, od wersji 3.3 duży spadek wydajności tego komponentu, związany jest z trwającymi bez przerwy pracami związanymi z dodawaniem nowych funkcjonalności czy wprowadzaniem usprawnień, których celem jest eliminacja błędów i poprawa bezpieczeństwa.

Kolejnym, komponentem poddanym testom, jest HttpFoundation. Definiuje on w sposób zorientowany obiektowo żądania i odpowiedzi związane z protokołem HTTP. Komponent ten zastępuje obiektami w PHP zmienne superglobalne takie jak: \$_GET, \$_POST, \$_FILES, \$_SERVER, \$_COOKIE, \$_SESSION.

Listing 4: Metoda testowa dla komponentu HttpFoundation

```
final public function benchHFComponent(): void
```

```
{
    $request = Request::createFromGlobals();
    $response = new Response(
        'Content',
        Response::HTTP_OK,
        ['content-type' => 'text/html']
    );
    $response->prepare($request);
}
```



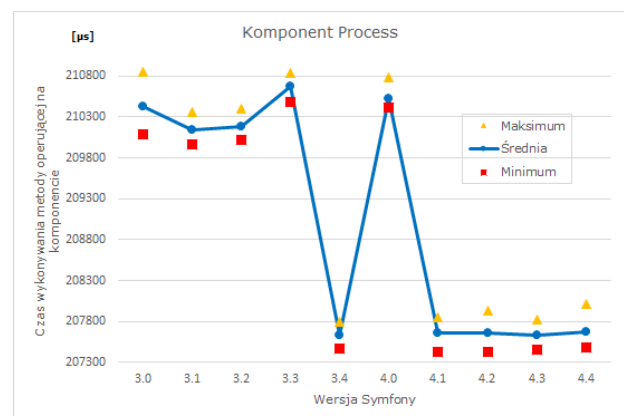
Rysunek 3: Wydajność komponentu HttpFoundation

HttpFoundation to bardzo ważny i w związku z tym stale poprawiany komponent. Usprawnianiu podlegają głównie aspekty bezpieczeństwa, co jednak negatywnie przekłada się na jego wydajność (rysunek 3).

Process to kolejny istotny komponent Symfony, który odpowiada za pośrednictwo pomiędzy językiem PHP, a systemem operacyjnym, na którym został uruchomiony. Zastępuje on typowe funkcje PHP ubogie w funkcjonalności dostępne dla programisty. Ważne jest to, że umożliwia on wykonywanie takich samych poleceń bez względu na system operacyjny, co tym samym daje programiście większą elastyczność pracy. Kolejną kluczową cechą tego komponentu jest możliwość uruchomienia procesu w trybie strumieniowania, dla którego w czasie rzeczywistym można odbierać informacje podczas wykonywania polecenia.

Listing 5: Metoda testowa dla komponentu Process

```
final public function benchProcessComponent(): void
{
    $process = new Process(['ls']);
    $process->run();
}
```



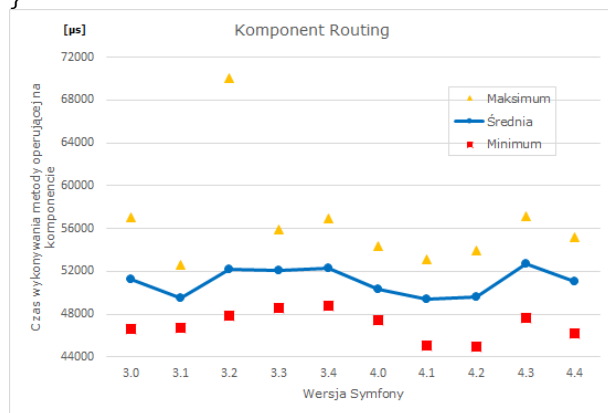
Rysunek 4: Wydajność komponentu Process

Process jest interesującym przykładem komponentu, w którym po skokowym wzroście wydajności w wersji 3.4, a następnie dużym spadku i powrotu do wyników sprzed wersji 3.4, nastąpiła znaczna poprawa i stabilizacja wydajności. Ten wyraźny wzrost wydajności był wynikiem wprowadzenia poważnych zmian, polegających na przykład na rezygnacji z używania jednej z funkcji PHP. Dzięki wsparciu społeczności, w wersji 4.1, zostały wprowadzone kolejne poprawki, które poprawiły działanie i wydajność tego komponentu.

Komponent Routing mapuje żądania HTTP do zestawu odpowiednich zmiennych opisanych przez programistę. Umożliwia on zbudowanie systemu routingu dla aplikacji internetowych, w których każdy unikalny adres URL zostanie powiązany z odpowiednim fragmentem metody kontrolera w architekturze MVC.

Listing 6: Metoda testowa dla komponentu Routing

```
final public function benchRoutingComponent(): void
{
    $routes = new RouteCollection();
    for($i=self::ROUTE_MIN;$i<=self::ROUTE_MAX;
    $i++)
    {
        $routes->add('route_'. $i, new Route('/route_'. $i));
    }
    $matcher=new UrlMatcher($routes,
        new RequestContext('/'));
    $matcher->match('/route_'.random_int(self::ROUTE_MIN,
        self::ROUTE_MAX));
}
```



Rys. 5. Wydajność komponentu Routing

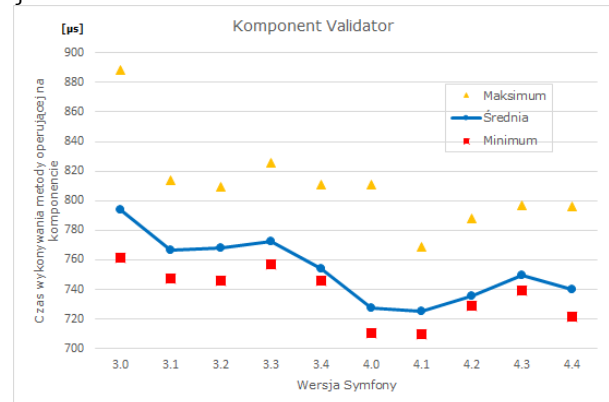
Rysunek 5 ukazuje zmiany zachodzące w wydajności komponentu Routing w kolejnych wersjach frameworka. Widoczny wzrost wydajności dla wersji 4.0, 4.1, 4.2 ma związek z zastosowaniem optymalizacji w jednej z bibliotek, która niewłaściwie implementuje część procesu routingu. Koncepcja polegająca na zrezygnowaniu z dopasowywania wyrażeń regularnych jedno po drugim, na skompilowanie ich w jedno duże wyrażenie regularne, wymaga ostatecznie tylko jednego dopasowania [8, 9].

Komponent Validator służy do sprawdzania poprawności właściwości klas, wartości zwracanych przez metody dostępne oraz całych obiektów, za pomocą zdefiniowanych reguł, które mogą być definiowane za pomocą kodu PHP, adnotacji, plików: YAML, XML lub

bezpośrednio przy użyciu prostych obiektów zawierających logikę walidacji.

Listing 7: Metoda testowa dla komponentu Validator

```
final public function benchValidatorComponent(): void
{
    $validator = Validation::createValidator();
    $violations = $validator->validate(
        'Michał',
        [new Length(['min' => 10]),
        new NotBlank(),
    ]);
    (string) $violations;
}
```

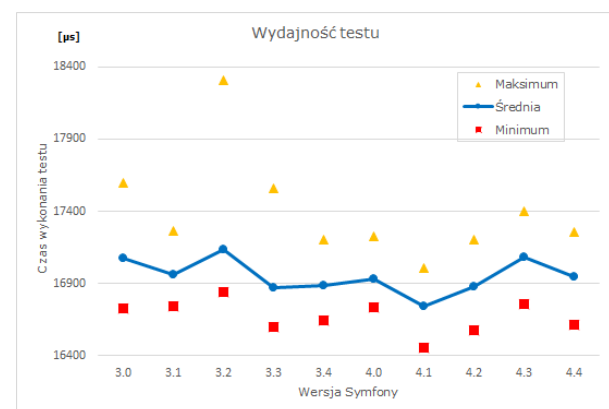


Rys. 6. Wydajność komponentu Validator

Rysunek 6 pokazuje, że wydajność komponentu Validator wzrosła w wersjach 4.0 i 4.1 frameworka, a następnie spadła w kolejnych wersjach. We wprowadzanych wydaniach Symfony, główne zmiany w obrębie tego komponentu koncentrowały się na wprowadzaniu coraz większej liczby nowych reguł, które dawały programistom szeroką bazę możliwości walidacji.

4.2. Wydajność szkieletu programistycznego

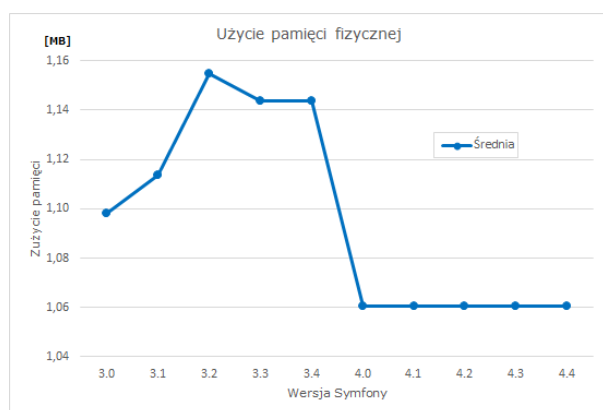
Ocena poszczególnych wersji szkieletu Symfony została dokonana na podstawie dziesięciu aplikacji, z których każda była zbudowana z komponentów danej wersji frameworka. Na tej podstawie wyznaczono sumaryczne średnie czasy wykonania, które zostały pokazane na rysunku 7.



Rys. 7. Średni czas trwania wykonywania się testów dla badanych wersji komponentów (wersji szkieletu programistycznego Symfony)

Dodatkowo wykonano pomiary zapotrzebowania na pamięć fizyczną poszczególnych wersji aplikacji (rysunek 8).

Z rysunku 7 wynika, że spośród dziesięciu wersji szkieletu Symfony najlepszą pod względem wydajności okazała się wersja 4.1. Potwierdza to osiągnięty najmniejszy średni czas wykonania aplikacji testowej. Uzyskany rezultat wynika z wysokiej wydajności komponentów Routing i Validator. Ponadto istotną informacją jest to, że zapotrzebowanie aplikacji na pamięć wyraźnie spadło od wersji 4.0 frameworka i w kolejnych wersjach utrzymywało się na takim samym poziomie (rysunek 8).



Rys. 8. Średnie użycie pamięci RAM przez aplikację testową

5. Wnioski

Przeprowadzone badania wykazały różnice wydajnościowe pomiędzy analizowanymi wersjami. Zmiany wprowadzane w kolejnych wersjach Symfony wiązały się z dodawaniem kolejnych tysięcy, a nawet dziesiątek tysięcy linii kodu. Zwiększanie się objętości frameworka, poprzez dodawanie nowych funkcjonalności, poprawek i zabezpieczeń skutkowało spadkiem jego wydajności. Z drugiej jednak strony wprowadzanie nowych rozwiązań i usprawnień miało pozytywny wpływ na kwestię szybkości działania aplikacji opartych na tym szkielecie. Z rysunku 7 wynika, że do wersji 4.1 wydajność Symfony rosła, a w następnych wersjach spadała. Biorąc pod uwagę zapotrzebowanie na pamięć fizyczną, należy przyznać, że nowe wersje frameworka (od 4.0)

są coraz bardziej wydajne (rysunek 8). W związku z tym nie można jednoznacznie potwierdzić, postawionej na początku pracy tezy, że aktualizacja komponentów przyczynia się do wzrostu ich wydajności oraz wzrostu wydajności całego frameworka.

Na podstawie przeprowadzonych badań, okazało się, że najlepszą wersją Symfony, rozpatrywaną w niniejszej pracy, jest ta oznaczona numerem 4.1. To właśnie w tej wersji autorzy wprowadzili nowy mechanizm obsługujący szybszą analizę adresów URL [8, 9]. Późniejsze wersje uzyskały gorsze wyniki, co można powiązać z wprowadzeniem do nich nowych możliwości i poprawek bezpieczeństwa.

Bezpieczeństwo ma fundamentalne znaczenie dla każdej aplikacji internetowej jednakże utrzymywanie jego wysokiego poziomu jest często realizowane kosztem mniejszej wydajności. Pomimo pogorszenia tej wydajności systematyczna aktualizacja komponentów jest konieczna ze względu na potrzebę zapewnienia coraz większego stopnia bezpieczeństwa, ale powinna ona uwzględniać potrzeby i specyfikę danej aplikacji.

Literatura

- [1] PHP Benchmarks, <http://www.phpbenchmarks.com/en/> [03.12.2019]
- [2] Symfony, <https://symfony.com/components> [04.12.2019]
- [3] Zaninotto F., Potencier F. The Definitive Guide to Symfony. Apress, 2007.
- [4] Coding Standards, <https://symfony.com/doc/current/contributing/code/standards.html> [04.12.2019]
- [5] Symfony Releases, <https://symfony.com/releases> [04.12.2019]
- [6] <https://symfony.com/blog/> [04.12.2019]
- [7] PHPBench's documentation, <https://phpbench.readthedocs.io/en/latest/> [04.12.2019]
- [8] Fast request routing using regular expressions, <https://nikic.github.io/2014/02/18/Fast-request-routing-using-regular-expressions.html> [15.12.2019]
- [9] Making Symfony's Router 77.7x faster, <https://medium.com/@nicolas.grekas/making-symfonys-router-77-7x-faster-1-2-958e3754f0e1> [15.12.2019]

Containerization methods in internet service applications

Metody konteneryzacji w zastosowaniach usług internetowych

Krzysztof Ferenc*

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

This work describes the most important methods of isolating running processes of selected software on one physical machine without using full virtualization of the operating system. It includes a comparison of the basic components of each of the methods described, including the features and interface, as well as the results of performance testing in a test environment prepared for this purpose. Research is based on running an identical application and comparing the use of basic system resources, as well as the time of launching containers. It may also be important to compare the methods studied to classical virtualization. At the end of the work, conclusions are described that may be relevant when choosing a method in future projects depending on requirements and needs.

Keywords: containers; virtualization; isolation; docker

Streszczenie

Niniejsza praca opisuje najważniejsze metody izolacji uruchomionych procesów wybranego oprogramowania na jednej maszynie fizycznej bez użycia do tego celu pełnej wirtualizacji systemu operacyjnego. Zawiera porównanie podstawowych składników każdej z opisanych metod, w tym cechy i interfejsu, a także wyniki badania wydajności w przygotowanym do tego celu środowisku testowym. Badania oparte są na uruchamianiu identycznej aplikacji i porównywaniu wykorzystania podstawowych zasobów systemowych, a także czasu uruchamiania kontenerów. Istotne jest tu również porównanie badanych metod do klasycznej wirtualizacji. Na końcu pracy opisane są wnioski, które mogą być istotne przy wyborze metody w przyszłych projektach w zależności od wymagań i potrzeb.

Słowa kluczowe: kontenery; wirtualizacja; izolacja; docker

*Corresponding author

Email address: krzysztof.ferenc1@pollub.edu.pl (K. Ferenc)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Przy tworzeniu nowoczesnych serwisów internetowych i aplikacji użytkowych dostępnych w sieci coraz większą rolę odgrywają mikroserwisy jako sposób na modularyzację aplikacji działających w środowisku produkcyjnym. Mikroserwisy jako wzorzec projektowy [1] używany przez architektów oprogramowania opiera się na koncepcie kontenerów, a te z kolei na wirtualizacji.

Jedną z najważniejszych zalet mikroserwisów jest ich uniwersalność, dzięki czemu idealnie nadają się do odtwarzania jednolitego środowiska niezależnie od fizycznej maszyny, na której zostają uruchomione. Taka konstrukcja sprawia, że nierzadko wykorzystywane są również przez programistów i testerów podczas wytwarzania oprogramowania.

Niniejsza praca skupia się na omówieniu wybranych sposobów konteneryzacji aplikacji i usług internetowych, w których mogą być one dzielone na mikroserwisy i separowane od siebie na poziomie systemu operacyjnego.

Poza samym omówieniem cech charakterystycznych dla konteneryzacji w odróżnieniu od innych podobnych rozwiązań, praca zawiera również porównanie wybranych dostępnych metod konteneryzacji, a także testy wydajnościowe dla przykładowego przypadku testowego.

Postawiono następującą tezę badawczą:

Kontenery w porównaniu z klasycznym rozwiązaniem wirtualizacji całego systemu operacyjnego wykazują się znacznie większą wydajnością przy uruchamianiu usług internetowych. Wykorzystanie procesora oraz pamięci operacyjnej przez maszynę wirtualną jest wyższe, a czas uruchamiania dłuższy.

2. Przegląd dostępnych rozwiązań

Według danych firmy Sysdig, Inc. [2] najpopularniejszym konteneryzatorem używanym w środowiskach produkcyjnych w 2018 roku był Docker z udziałem na poziomie 83% w porównaniu z konkurencyjnymi rozwiązaniami [3]. Ten wynik wskazuje na znaczny spadek z poziomu 99% w 2017 według danych tej samej firmy. Te same dane określają także procentowy udział w 2018 roku narzędzi takich jak: CoreOS rkt – 12%, Apache Mesos – 4% i LXC – 1%.

Do porównania spośród wymienionych najpopularniejszych narzędzi [3] wybrane zostały Docker i rkt z uwagi na najwyższą popularność, a także kontenery LXC/LXD, które są najstarszym i najdłużej używanym z wymienionych rozwiązań. Oprogramowanie Apache Mesos nie zostało opisane w niniejszej pracy z powodu relatywnie krótkiego istnienia (od lipca 2016 roku), a także głównego nastawienia na zastosowanie w dużych centrach danych i zarządzania klastrami maszyn [4], co stanowi kontrast do pozostałych systemów, które

uwzględniają przypadki użycia w ramach jednej maszyny fizycznej.

3. Metodyka badawcza

3.1. Stanowisko badawcze

Każde z porównywanych rozwiązań posiada odmienne wymagania w zakresie warstwy sprzętowej, jak i systemu operacyjnego, na którym narzędzie konteneryzacji jest wspierane. To co je łączy w zakresie wymagań to fakt, że działają natywnie na systemach bazujących na Linuxie, który to jest najpopularniejszy w środowiskach serwerowych [5]. Z uwagi na to, do celów badania obsługi i wydajności poszczególnych rozwiązań wybrany został system Arch Linux jako system operacyjny hosta.

Głównym powodem tej decyzji jest prostota przeniesienia oprogramowania z innych systemów bazujących na jądrze Linux, a także możliwość stworzenia lekkiej instalacji z minimalną liczbą dodatkowego oprogramowania, które poprzez nierównomierne zużycie zasobów mogłoby zaburzyć odczyty wydajności systemów konteneryzacji.

Do testów została wykorzystana fizyczna maszyna w formie komputera stacjonarnego o architekturze procesora x86_64, nazywanej często amd64. Jest to jedyny możliwy wybór, który jest wspierany przez wszystkie konteneryzatory.

3.2. Narzędzia badawcze

Pozyskiwanie wyników ma charakter eksperymentalny, poprzez uruchomienie przykładowej aplikacji internetowej w formie obrazu obsługiwanej przez systemy konteneryzacji jako działający kontener. Następnie przy użyciu programów pomiarowych zostanie zmierzony czas i zasoby używane przez każdy z kontenerów.

Do celów uruchamiania aplikacji internetowej w kontenerze posłuży prosty serwer działający na frameworku Node.js, który odpytany na odpowiednim adresie i porcie zwraca odpowiedź tekstową. Kod programu przedstawia Listing 1.

Listing 1: Przykład kodu uruchamiającego serwer aplikacji internetowej

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log('Server running at http://${hostname}:${port}/');
});
```

Wybór wspomnianego frameworka opiera się o wysoką dostępność i łatwość użycia w dowolnym środowisku. Większość dokumentacji i poradników wybiera również ten konkretny przykład do pokazania łatwości w użyciu danego rozwiązania wirtualizacji.

Do pomiaru czasu wykonywania polecenia uruchamiania kontenerów wykorzystany zostanie wbudowany w większość systemów bazujących na Linuxie program GNU time [6].

Pomiar wykorzystanych zasobów procesora i pamięci systemowej wykonany zostanie natomiast z użyciem wbudowanych narzędzi diagnostycznych w każdym z systemów konteneryzacji lub z użyciem wbudowanego w system operacyjny programu ps, którym można monitorować zarówno wykorzystanie przez proces procesora maszyny gospodarza, jak i użycie pamięci.

Właściwości badania statystycznego dobrane zostaną w taki sposób, aby jak najmniej wpłynąć na wykorzystanie zasobów przez narzędzia pomiarowe, a jednocześnie możliwe było wskazanie charakterystyki zmian badanych wartości. Na końcu uzyskane wyniki zostaną uśrednione dla wszystkich wykonanych uruchomień danego kontenera i czasu, następnie porównane razem z wynikami uzyskanymi dla klasycznej metody wirtualizacji jako próba porównawcza.

4. Porównanie funkcji kontenerów

Różnice między opisywanymi systemami konteneryzacji pojawiają się na warstwie obsługiwanych metod izolacji zasobów, a także samej architektury narzędzi. Tabela 1 zawiera zestawienie wspieranych funkcji w omawianych narzędziach.

Tabela 1: Porównanie funkcji systemów konteneryzacji

Funkcja	LXC/LXD	Docker	rkt
Centralna usługa zarządzająca kontenerami	Tak	Tak	Nie
Izolacja systemu plików	Tak	Tak	Tak
Ograniczanie przestrzeni dyskowej	Nie	Nie	Nie
Ograniczanie przepustowości przestrzeni dyskowej	Nie	Tak	Nie
Ograniczanie pamięci systemowej	Tak	Tak	Tak
Ograniczanie mocy obliczeniowej	Tak	Tak	Nie
Izolacja sieci	Tak	Tak	Tak
Wirtualizacja zagnieżdżona	Tak	Tak	Nie
Migracja na żywo	Tak	Nie	Nie
Izolacja uprawnień systemu plików	Tak	Tak	Tak

5. Interfejs

Istnieją dwa główne rodzaje interfejsów używanych do komunikacji z usługą zarządzania kontenerami – tekstowy i graficzny.

Interfejs tekstowy jest uniwersalny dla każdego z rozwiązań, natywnie wspierany, a także jest podstawą przy rozpoczynaniu nauki z mikroserwisami. Zaletą jego używania jest przejrzystość prezentacji danych i precyzja podczas wykonywania operacji. Początkujących administratorów lub też ludzi nieobeznanych z terminalem może odrzucić pozorna archaiczność tego sposobu interakcji, ponieważ nie jest ona najbardziej

intuicyjna i wymaga wcześniejszego przygotowania w postaci przeczytania fragmentu dokumentacji.

Z przedstawionych metod obsługi tworzenia obrazu wynika, że czas i ilość kroków znacznie różni się w zależności od wybranego systemu konteneryzacji. Dla Dockera wystarczy jeden plik konfiguracyjny i jedno polecenie do budowania obrazu. System rkt udostępnia narzędzie do tworzenia pliku konfiguracyjnego bez potrzeby tworzenia go ręcznie i nie wymaga dodatkowego polecenia budowania obrazu. LXD posiada najbardziej skomplikowaną procedurę wymagającą ręcznego pobrania, stworzenia systemu plików i spakowania go do archiwum, następnie stworzenia i spakowania pliku konfiguracyjnego, aby na końcu wywołać polecenie tworzenia obrazu.

Niezmienne prosta jest natomiast obsługa już utworzonego obrazu. Przetawione tutaj przykładowo polecenia uruchamiania kontenerów analogicznie funkcjonują przy wykonywaniu innych operacji, jak włączanie i wyłączanie kontenera, usuwanie kontenerów i obrazów, listowanie i uruchamianie dodatkowych procesów w działających kontenerach.

Graficzne zarządzanie jest zdecydowanie rzadziej stosowaną formą interakcji i często nie oferuje pełni możliwości w porównaniu do klasycznego terminala tekstowego. Z pośród badanych metod konteneryzacji jedynie Docker posiada szeroki wybór narzędzi, które pozwalają na większość podstawowych operacji zarządzania obrazami, sieciami, wolumenami i samymi kontenerami w tym ich tworzeniem, monitorowaniem i usuwaniem.

6. Wydajność

W efekcie dobrania odpowiednich właściwości badania statystycznego uzyskane dane opierają się na dwudziestu próbach uruchomienia każdego z kontenerów w nieobciążonym środowisku testowym na czas wystarczający do pełnej gotowości izolowanej usługi. Dane o zużywanych zasobach zbierane były co sześć sekund, dzięki czemu wpływ narzędzi pomiarowych na działanie kontenerów był marginalny.

Najdłuższy czas uruchomienia usługi do stanu pełnej gotowości wyniósł 125 sekund, co wystąpiło na maszynie wirtualnej stanowiącej próbę porównawczą. Z tego powodu czas wszystkich badań został ustalony na najbliższą wartość pomiarową przekraczającą ten czas, czyli 126 sekund, na co przypadają 22 pomiary zużywanych zasobów.

Po wyznaczonym czasie każdy z kontenerów został wyłączony, a następnie włączony kolejny.

6.1. Czas uruchamiania

Z wykresu na Tabeli 2 można odczytać bardzo podobne wyniki czasów uruchamiania kontenerów przy zastosowaniu do tego celu narzędzi Docker i LXD. Ponadto wahania między najkrótszym, a najdłuższym czasem uruchomienia przy próbie nie przekraczały odpowiednio 0,29 i 0,27 sekundy.

Nieco inaczej sytuacja wygląda w przypadku rkt i tutaj czasy uruchamiania są wyraźnie większe, co

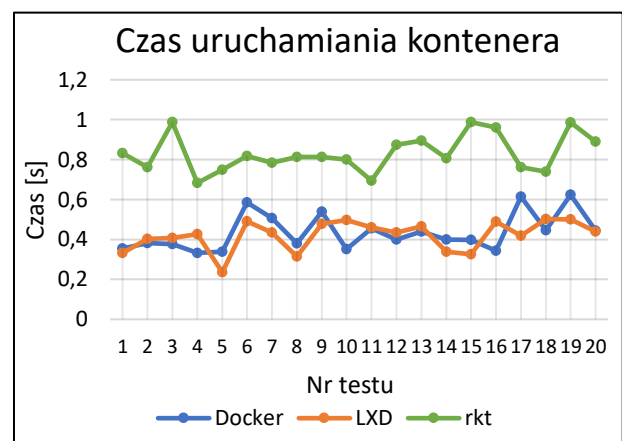
przekłada się na powolniejszy start usług. Nie jest to natomiast ogromna różnica. Ekstrema również są nieco bardziej oddalone – o dokładnie 0,3 sekundy.

Tabela 2: Wyniki badania czasu uruchamiania [s]

Nr. testu	Docker	LXD	rkt	KVM
1	0,35	0,33	0,83	74,47
2	0,38	0,40	0,76	77,89
3	0,38	0,41	0,99	70,12
4	0,33	0,43	0,68	74,42
5	0,34	0,24	0,75	74,63
6	0,59	0,49	0,82	79,36
7	0,51	0,43	0,78	81,20
8	0,38	0,31	0,81	80,29
9	0,54	0,48	0,81	77,38
10	0,35	0,50	0,80	79,61
11	0,46	0,46	0,69	76,55
12	0,40	0,44	0,87	77,15
13	0,44	0,46	0,89	75,73
14	0,40	0,34	0,81	78,27
15	0,40	0,33	0,99	70,13
16	0,34	0,49	0,96	77,49
17	0,61	0,42	0,76	75,93
18	0,45	0,50	0,74	79,59
19	0,62	0,50	0,99	77,19
20	0,44	0,44	0,89	72,21

Testy dla maszyny wirtualnej (KVM) wykazały ogromną różnicę między czasem uruchamiania. W porównaniu do najdłuższego czasu uruchamiania kontenera - 0,99 sekundy, najkrótszy czas uruchamiania maszyny wirtualnej z identyczną aplikacją wyniósł 70,12 sekundy, co stanowi niemal 70-krotną różnicę (dokładnie 6982,83%).

Wyniki dla systemów konteneryzacji przedstawione zostały na rysunku 1.



Rysunek 1: Czas uruchamiania pojedynczego kontenera - porównanie

6.2. Wykorzystanie mocy procesora

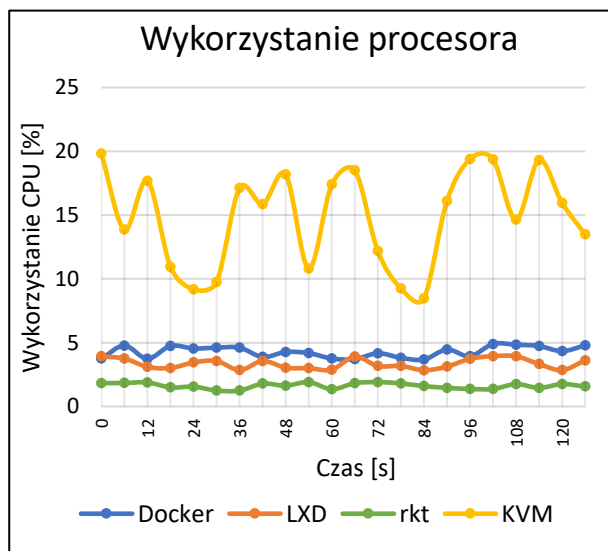
Wykorzystywana moc procesora (Tabela 3), reprezentowana w procentach, wskazuje na podobne, niewielkie wahania dla każdego z systemów konteneryzacji – od 0,67 % dla rkt, do 1,22% dla Dockera. Inaczej jest jednak w przypadku maszyny wirtualnej, gdzie zauważalne są różnice między najniższym i najwyższym wykorzystaniem procesora na poziomie 11,35%.

Faktem jest także to, że narzędzie rkt najmniej obciąża procesor, co może być wynikiem braku centralnej usługi zarządzania kontenerami.

Tabela 3: Wyniki badania wykorzystania mocy procesora [%]

Czas [s]	Docker	LXD	rkt	KVM
0	3,75	3,92	1,83	19,81
6	4,77	3,76	1,84	13,87
12	3,74	3,11	1,89	17,69
18	4,75	3,02	1,49	10,94
24	4,55	3,45	1,55	9,17
30	4,61	3,57	1,25	9,76
36	4,61	2,85	1,24	17,13
42	3,88	3,57	1,79	15,85
48	4,27	3,04	1,63	18,18
54	4,19	3,00	1,91	10,80
60	3,76	2,88	1,35	17,41
66	3,70	3,92	1,83	18,52
72	4,17	3,17	1,91	12,18
78	3,81	3,18	1,81	9,25
84	3,68	2,84	1,59	8,47
90	4,46	3,13	1,46	16,10
96	3,94	3,73	1,37	19,39
102	4,90	3,95	1,37	19,37
108	4,85	3,93	1,75	14,64
114	4,74	3,32	1,46	19,32
120	4,34	2,85	1,76	15,95
126	4,79	3,61	1,56	13,48

Obserwacje te można zauważyć również na wykresie z Rysunku 2. Widoczne są wyraźne wzrosty i spadki funkcji powiązanej z wynikami maszyny wirtualnej (KVM).



Rysunek 2: Wykorzystanie mocy procesora przez pojedynczy kontener - porównanie

6.3. Wykorzystanie pamięci operacyjnej

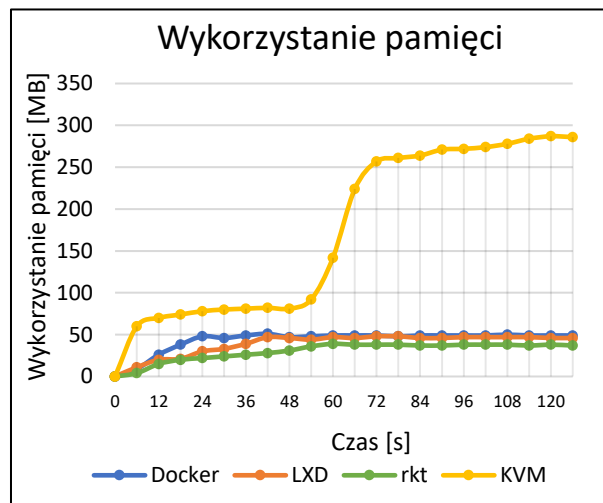
W tabeli 4 w każdym przypadku widać wzrost wykorzystania mocy procesora w czasie działania instancji kontenera. Dla systemów konteneryzacji wykorzystanie to stabilizuje się po około 54-ech sekundach od uruchomienia,

po którym to czasie nie zwiększa się. Maszyna wirtualna natomiast poza znacznie większym wykorzystaniem pamięci operacyjnej charakteryzuje się dwoma skokami w zapotrzebowaniu – w 6-tej i 60-tej sekundzie od uruchomienia.

Tabela 4: Wyniki badania wykorzystania pamięci [MB]

Czas [s]	Docker	LXD	rkt	KVM
0	0	0	0	0
6	9	11	4	60
12	26	20	15	70
18	38	21	20	74
24	48	30	22	78
30	46	33	24	80
36	49	39	26	81
42	51	47	28	82
48	47	46	31	81
54	48	44	36	92
60	49	47	39	142
66	49	46	38	224
72	49	48	38	257
78	48	48	38	261
84	49	46	37	264
90	49	46	37	271
96	49	47	38	272
102	49	47	38	274
108	50	47	38	278
114	49	47	37	284
120	49	46	38	287
126	49	46	37	286

Wizualna reprezentacja danych na Rysunku 3 dodatkowo uwidacznia skoki w wykorzystaniu zasobów pamięci dla maszyny wirtualnej, a także szybką stabilizację dla konteneryzatorów.



Rysunek 3: Wykorzystanie pamięci operacyjnej – porównanie

7. Wnioski

Przeprowadzone badania jasno wskazują, że istnieją różnice między badanymi rozwiązaniami z zakresu konteneryzacji aplikacji internetowych. Dla uruchomionej aplikacji w różnych systemach konteneryzacji wykorzystywane były różne ilości dostępnych zasobów systemowych maszyny gospodarza.

Wyniki wykorzystania mocy procesora i pamięci operacyjnej pokazują, że największe zapotrzebowanie z badanych rozwiązań ma konteneryzator Docker. Pomimo jednak tego, czas uruchamiania jest znacznie krótszy od rkt, czyli systemu o najmniejszych wymaganiach sprzętowych.

Najbardziej zbalansowanym pod względem używanych zasobów, jak i czasu uruchamiania jest system konteneryzacji LXD, którego zapotrzebowanie na moc obliczeniową procesora jest o 70% wyższe od rkt, jednak o 24% niższe od Dockera. Podobnie wygląda sytuacja przy zapotrzebowaniu na pamięć operacyjną, gdzie LXD jest o 26% mniej wydajny od rkt, jednak o 11% wydajniejszy od Dockera.

Cechą wyróżniającą kontenery rkt jest ich długi czas uruchamiania w porównaniu do innych rozwiązań. Oznacza to, że kontenery te nie nadają się najlepiej do krótko żyjących procesów. Z uwagi jednak, że rkt wykazuje najniższe zużycie zasobów – świetnie nada się w roli konteneryzatora do długożyjących procesów, co w kontekście aplikacji internetowych może być wykorzystane do uruchomienia API, które musi być stale dostępne do użycia.

Kwestią bardziej indywidualną może być interfejs, który udostępnia każdy z systemów izolacji aplikacji. Komunikacja z oprogramowaniem wygląda podobnie w każdym przypadku z uwagi na ustandaryzowany sposób wykorzystania wiersza poleceń. Udogodnieniem mogą być także polecenia podzielone na kategorie, które jednak w pełnym zakresie dostępne są jedynie w narzędziu Docker, jednak częściowo występują w każdym z badanych rozwiązań.

Istotną kwestią, jaką należy jednak brać pod uwagę jest popularność. W przypadku konteneryzacji zdecydowaną przewagę osiągnął Docker, który w ciągu ostatnich lat stał się liderem, a przez wielu również jedynym znanym sposobem izolacji aplikacji bez używania do tego celu zasobożernych maszyn wirtualnych.

Ponieważ popularność często niesie za sobą również wsparcie przez producentów oprogramowania w postaci gotowych do użycia obrazów, może okazać się, że Docker jest najprostszym sposobem uruchomienia istniejącej na rynku aplikacji internetowej. Mniejsze znaczenie

ma to w przypadku tworzenia własnego obrazu dla kontenera.

Bez względu na wybrane narzędzie konteneryzacji, wyniki badań udowadniają, że charakteryzują się one większą wydajnością w porównaniu do klasycznego rozwiązania izolacji aplikacji, jakim jest pełna wirtualizacja systemu operacyjnego, co potwierdza postawioną we wstępie tezę badawczą.

Istnieją podobne badania [7] wydajności systemów konteneryzacji w porównaniu z maszynami wirtualnymi. Z uwagi na różnice w zastosowanych scenariuszach nie jest możliwe jednoznaczne porównanie wyników, jednak wnioski są podobne i wskazują, że użycie kontenera do udostępnienia usługi jest bardziej wydajne niż wykorzystanie maszyny wirtualnej, co również udowadnia postawioną tezę.

Literatura

- [1] S. Newman, Building Microservices: Designing Fine-Grained Systems, O'Reilly Media, 2015.
- [2] E. Carter, 2018 Docker usage report, <https://sysdig.com/blog/2018-docker-usage-report/> [24.03.2020].
- [3] B. Doerrfeld, 5 Container Alternatives to Docker, <https://containerjournal.com/topics/container-ecosystems/5-container-alternatives-to-docker/> [24.03.2020].
- [4] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. Joseph, R. Katz, S. Shenker, I. Stoica, Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center, University of California, Berkeley, 2010, <https://people.eecs.berkeley.edu/~alig/papers/mesos.pdf> [24.03.2020].
- [5] W3Cook.com, OS Market Share and Usage Trends : <https://web.archive.org/web/20150806093859/http://www.w3cook.com/os/summary/> [24.03.2020].
- [6] GNU Time – GNU Project – Free Software Foundation, <https://www.gnu.org/software/time/> [29.01.2020].
- [7] B. Russell, KVM and docker LXC Benchmarking with OpenStack Trends, <https://www.slideshare.net/BodenRussell/kvm-and-docker-lxc-benchmarking-with-openstack?gid=5d451a56-0bbb-4c30-a411-8a20c83d4992> [24.03.2020].

Performance analysis of languages working on Java Virtual Machine based on Java, Scala and Kotlin

Analiza wydajności języków działających na Java Virtual Machine na przykładzie języków Java, Scala i Kotlin

Katarzyna Buszewicz*

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

This article presents the results of a literature study related to the construction and operation of Java Virtual Machine, as well as performance tests of selected languages using the aforementioned runtime environment on the example of Java, Scala and Kotlin. Performance testing was carried out using two applications built using the Apache Maven archetype with the built-in Java Microbenchmark Harness library.

Keywords: Java; Scala; Kotlin; Performance

Streszczenie

Niniejszy artykuł przedstawia wyniki badania literaturowego związanego z budową oraz działaniem Java Virtual Machine, jak również badania wydajności wybranych języków, wykorzystujących wspomniane środowisko uruchomieniowe na przykładzie Javy, Scali oraz Kotlin. Badanie wydajności przeprowadzono wykorzystując dwie aplikacje, zbudowane przy użyciu archetypu Apache Maven z wbudowaną biblioteką Java Microbenchmark Harness.

Słowa kluczowe: Java; Scala; Kotlin; wydajność

*Corresponding author

Email address: buszewicz.katarzyna@gmail.com (K. Buszewicz)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

W 2020 roku minie 25 lat od powstania języka Java. Przez wiele lat była najpopularniejszym językiem programowania [1], a podczas jej instalacji wyświetlany jest slogan „3 billion devices run Java”. Na tak wielki sukces wpłynęło wiele czynników, między innymi jego środowisko uruchomieniowe, jakim jest Java Virtual Machine. Umożliwia ono translację na kod maszynowy, zgodny z architekturą platformy, na której aplikacja jest uruchamiana, a więc pozwala na działanie raz napisanego kodu na dowolnym urządzeniu z dostępem do JVM.

Jednak nie tylko Java może działać na swojej maszynie wirtualnej. Na przestrzeni lat powstało wiele nowych języków, uruchamianych za pośrednictwem wspomnianego środowiska uruchomieniowego. Część z nich opartych jest na Javie i rozwija ją o nowe elementy, które eliminuje konieczność pisania powtarzającego się kodu, a także najczęściej występujące wyjątki. Przykładami takich języków są Scala, która swoje zastosowanie znalazła głównie w Big Data oraz Kotlin, ogłoszony 17 maja 2017 jako oficjalnie wspierany język do programowania na systemy Android [2].

2. Cel badań

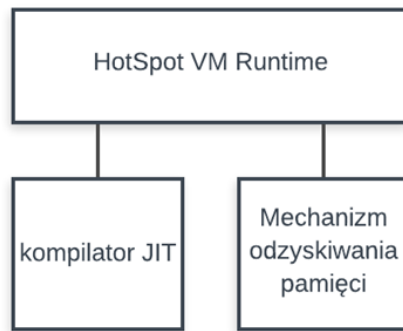
Celem badań zaprezentowanych w niniejszym artykule było porównanie wydajności Javy oraz wybranych języków, będących jej udoskonalonymi wersjami na przykładzie Scali oraz Kotlin. Badania zrealizowano zgodnie z wiedzą na temat działania środowiska uruchomieniowego, wspólnego dla wspomnianych języków, jakim jest Java Virtual Machine, co było możliwe

dzięki zastosowaniu biblioteki Java Microbenchmark Harness.

3. Java Virtual Machine

Najważniejszym założeniem twórców Javy było stworzenie języka zapewniającego przenaszalność, co promowano sloganem „napisz raz, uruchom gdziekolwiek” (ang. WORA - write once, run anywhere) [3]. Koniecznym do spełnienia tego wymogu był moduł pośredniczący pomiędzy wysokopoziomowym kodem Javy oraz kodem maszynowym, wykonywanym przez procesor. Moduł ten, z punktu widzenia każdej aplikacji Java, powinien stanowić abstrakcję procesora [4]. W tym celu opracowano specyfikację Java Virtual Machine, stanowiącą podstawę licznych implementacji, z których najpopularniejszą obecnie jest Oracle HotSpot VM [5].

Wysokopoziomowymi elementami składowymi wirtualnej maszyny Java, przedstawionymi na Rysunku 1, są: kompilator Just-In-Time, środowisko uruchomieniowe VM Runtime oraz mechanizm odzyskiwania pamięci [6]. Występuje ona w wersji 32 lub 64-bitowej, konfigurowalna jest bogatym zestawem flag (mogących różnić się w zależności od implementacji). Dzięki wprowadzeniu instrukcji `invokedynamic` w wersji siódmej pozwala na interpretację zmiennych dynamicznie typowanych, co umożliwia uruchamianie skryptów tworzonych w językach interpretowanych, jak na przykład JavaScript.



Rysunek 1: Podstawowe elementy budowy Java HotSpot VM

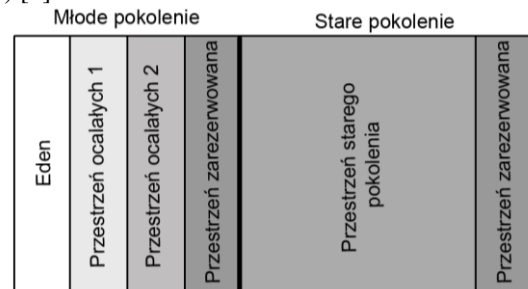
HotSpot VM Runtime, oprócz weryfikacji oraz interpretacji kodu bajtowego zawartego w plikach z rozszerzeniem `.class`, zapewnia ładowanie klas, analizę argumentów wiersza poleceń, obsługę wyjątków, wielowątkowość, jak również kontrolę cyklu życia maszyny wirtualnej oraz obsługę jej błędów krytycznych [6]. Zapewnia więc pełną kontrolę cyklu życia aplikacji od momentu uruchomienia, aż do momentu usunięcia powiązanych instancji maszyny wirtualnej i wątków oraz zwolnienia zajmowanych zasobów sprzętowych. Dostarczony do uruchomienia **kod bajtowy nie musi pochodzić z kompilacji kodu napisanego w Javie, o ile jest zgodny ze specyfikacją** – to właśnie ta właściwość otworzyła pole twórcom innych języków.

Za dodatkowe optymalizacje kodu oraz kompilację do kodu natywnego, zgodnego z architekturą platformy, na jakiej jest on uruchamiany, odpowiada **kompilator Just-In-Time**. Jego wykorzystanie jest czasochłonne i nie jest opłacalne dla fragmentów kodu, które są rzadko wywoływane. Z tego względu, w początkowej fazie działania aplikacji przeprowadzane jest tak zwane „**rozgrzewanie**” maszyny wirtualnej [7], polegające na zliczaniu wywołań metod oraz określanie, które z nich są „gorącymi punktami” (ang. hot spots). To stąd zaczerpnięto nazwę maszyny wirtualnej Oracle – HotSpot.

Trzecim elementem składowej maszyny wirtualnej Java jest mechanizm zarządzania pamięcią – **Garbage Collector** („odśmiecacz”). Obiektem jego zainteresowania jest sarta, na której przechowywane są obiekty powstałe podczas działania aplikacji. Jej struktura, zaprezentowana na Rysunku 2, budowana jest zgodnie z tak zwaną hipotezą śmiertelności niemowląt, zgodnie z którą obiekty najmłodsze mają znacznie mniejsze szanse na przetrwanie niż obiekty starsze [6], wobec tego sarta dzielona jest na generację młodego pokolenia oraz generację starego pokolenia. Obszary pamięci zarezerwowanej wykorzystywane są do dynamicznych zmian dostępnego rozmiaru sarty.

Uruchomienie Garbage Collector’a wiąże się z przerwą w działaniu aplikacji, z tego względu opracowano cztery różne algorytmy, przeznaczone zarówno dla aplikacji wymagających wysokiej przepustowości (algorytm równoległy, wszystkie wątki aplikacji muszą być dostępne), takich, które wymagają jak krótszego czasu odpowiedzi (algorytm szeregowy, jeden wątek zajęty jest odśmiecaniem, a pozostałe obsługują aplikację), a także wymagających

korzyści płynących z obu wspomnianych typów (algorytm CMS oraz algorytm G1). Podczas procesu odzyskiwania pamięci usuwane są nieużywane już obiekty, a te, które będą jeszcze potrzebne, przenoszone są do kolejnych sektorów sarty (przestrzeni starego pokolenia) [6].



Rysunek 2: Budowa sarty Java

4. Metoda badania wydajności

Charakterystyka pracy maszyny wirtualnej Java, opisana szerzej w Rozdziale 2, narzuca na sposób badania wydajności kodu pewne wymagania. Okazuje się, że różnica czasu pobranego przed wykonaniem operacji oraz po wykonaniu operacji jest miarą niewystarczającą ze względu na rozgrzewanie maszyny. Jednokrotne uruchomienie kodu nie odzwierciedla jego pracy w środowisku produkcyjnym, nie pozwala na osiągnięcie progu wywołań umożliwiającego dokonywanie optymalizacji przez kompilator Just-In-Time. Kolejnym problemem w przypadku badań okazuje się zdolność kompilatora do eliminowania martwego kodu – jeśli wynik działania metody do testów nie jest nigdzie użyty, taka metoda zostanie usunięta.

Aby rozwiązać wyżej opisane problemy, wykorzystano bibliotekę Java Microbenchmark Harness (w skrócie: JHM) autorstwa Alekseja Shipilëva, współtwórcy kompilatora JIT [8]. Biblioteka stworzona jest w języku Java, lecz zapewnia kompatybilność z innymi językami działającymi na JVM. Użycie jej wymaga odpowiedniej konfiguracji, między innymi wykorzystującej odrębną metodę `main()`, lecz dzięki wykorzystaniu archetypów dla narzędzia Apache Maven [9] możliwe jest utworzenie jednolicie skonfigurowanych projektów zarówno w Javie, w Scali, jak również w Kotlinie. Umożliwia ona między innymi symulację procesu rozgrzewania maszyny wirtualnej, oznaczenie metody jako hot spot, wsparcie do omijania eliminacji martwego kodu dzięki klasie `Blackhole`, imitującej konsumenta zwracanych przez metody testowe danych, jak również ustawienia wyników pomiaru, konfigurację ilości wątków czy zestawu parametrów. Wykorzystanie wspomnianej biblioteki do testowania wydajności rekomendowane jest także przez twórców Scali [10].

W niniejszej pracy zbadano wydajność aplikacji, rozumianą jako ilość operacji wykonanych w jednostce czasu. Wszystkie pomiary zostały wykonane na laptopie podłączonym do zasilacza, z trybem maksymalnej wydajności, wyposażonym w procesor Intel Core i5-8300H o taktowaniu 2,3 GHz, 16 GB pamięci operacyjnej SO-DIMM DDR4 2666MHz, z systemem Windows 10 64

b. Badane aplikacje umieszczone były na dysku HDD - 5400 obrotów na s. Do badań wykorzystano następujące wersje języków: Javę 1.8 [5], Scalę 2.13.1 [10] oraz Kotlin 1.3.61 [11].

Badanie wydajności przeprowadzono wykorzystując dwie różne aplikacje, każdą z nich zaimplementowano w każdym z badanych języków, każdą uruchomiono dla trzech różnych przypadków. Testy wydajności uruchamiane były jednowątkowo, po dwa cykle, składające się z pięciu iteracji rozgrzewkowych oraz dziesięciu iteracji pomiarowych - konfigurację tę można zaobserwować w użytych adnotacjach z biblioteki Java Microbenchmark Harness na każdym, niżej zamieszczonym listingu.

Pierwsza aplikacja służy do wyliczenia zadanego wyrazu ciągu Fibonacciego metodą rekurencyjną. Jest to rozwiązanie nieoptymalne, o złożoności $O(2n)$. Warto zaznaczyć, że Scala oraz Kotlin posiadają wsparcie rekursji ogonowej, dzięki której algorytm ten mógłby zostać znacznie zoptymalizowany. Aplikacje w Javie i Scali składają się z trzech klas: Fibonacci, posiadającej właściwą implementację algorytmu wyliczającego wartość wyrazu ciągu Fibonacciego, MyState zawierającej przypadki testowe (numery wyrazów ciągu do wyliczenia) oraz FibonacciBenchmark (w przypadku Kotliny singleton ten umieszczono w jednym pliku razem z metodą do pomiaru), zawierającej metodę do uruchamiania pomiaru wraz z odpowiednią konfiguracją. Przykład 1 prezentuje implementację w Javie, Przykład 2 prezentuje implementację w Scali, a Przykład 3 prezentuje implementację w Kotlinie. Pomiary przeprowadzono dla wyliczenia wyrazu piątego, trzydziestego oraz pięćdziesiątego.

Listing 1: Ciąg Fibonacciego - Java

```
public class Fibonacci {
    //Właściwa metoda wyliczająca wyraz ciągu
    //Fibonacciego
    public static long countFibonacciSequence(int
number) {
        if(number <= 1) return number;
        else return countFibonacciSequence(number - 2)
+ countFibonacciSequence(number - 1);
    }
}

import org.openjdk.jmh.annotations.Param;
import org.openjdk.jmh.annotations.Scope;
import org.openjdk.jmh.annotations.State;

//parametry do wywołań - numery wyrazów do wyliczenia
@State(Scope.Thread)
public class MyState {
    @Param({"5", "30", "50"})
    public Integer iterations;
}

//Klasa testująca z konsumentem danych - Blackhole
// (choć biblioteka umieści go automatycznie)
import org.openjdk.jmh.annotations.*;
import org.openjdk.jmh.infra.Blackhole;
import java.util.concurrent.TimeUnit;

public class FibonacciBenchmark extends MyState{
```

```
@Benchmark
@BenchmarkMode(Mode.Throughput)
@OutputTimeUnit(TimeUnit.SECONDS)
@Warmup(iterations = 5)
@Threads(1)
@Fork(2)
@Measurement(iterations = 10)
public void testFibonacci(Blackhole blackhole) {

blackhole.consume(Fibonacci.countFibonacciSequence(it
erations));
}
}
```

Listing 2: Ciąg Fibonacciego - Scala

```
object Fibonacci {
    //Właściwa metoda wyliczająca wyraz ciągu
    Fibonacci
    def countFibonacciSequence(number: Int) : Long = {
        if(number <= 1) return number
        countFibonacciSequence(number - 2)
+ countFibonacciSequence(number - 1)
    }
}

import org.openjdk.jmh.annotations.{Param, Scope}
@org.openjdk.jmh.annotations.State(Scope.Benchmark)
class MyState {
    //parametry do wywołań - numery wyrazów do wyliczenia
    @Param(Array("5", "30", "50"))
    var iterations: Int = 0
}

//Klasa testująca z konsumentem danych - Blackhole
import java.util.concurrent.TimeUnit
import org.openjdk.jmh.annotations._

class FibonacciBenchmark extends MyState {
    @Benchmark
    @BenchmarkMode(Array(Mode.Throughput))
    @OutputTimeUnit(TimeUnit.SECONDS)
    @Warmup(iterations = 5)
    @Threads(1)
    @Fork(2)
    @Measurement(iterations = 10)
    def testFibonacci(state : MyState): Any =
Fibonacci.countFibonacciSequence(state.iterations)
}
```

Listing 3: Ciąg Fibonacciego - Kotlin

```
object Fibonacci {
    //Właściwa metoda wyliczająca wyraz ciągu
    Fibonacci
    fun countFibonacciSequence(
numerWyrazu: Int): Long {
        return if (numerWyrazu <= 1)
            numerWyrazu.toLong()
        else
            countFibonacciSequence(numerWyrazu - 2)
+ countFibonacciSequence(numerWyrazu - 1)
    }
}

//Klasa testująca z konsumentem danych - Blackhole
import com.kbuszewicz.Fibonacci
import org.openjdk.jmh.annotations.*
import java.util.concurrent.TimeUnit
```



```
//parametry do wywołań - numery wyrazów do wyliczenia
@State(Scope.Thread)
open class MyState {
    @Param( "5", "30", "50") var size: Int = 0
}
open class FibonacciBenchmark : MyState() {
    @Benchmark
    @BenchmarkMode(Mode.Throughput)
    @OutputTimeUnit(TimeUnit.SECONDS)
    @Warmup(iterations = 5)
    @Threads(1)
    @Fork(2)
    @Measurement(iterations = 10)
    fun countFibonacciSequence(): Long {
        return Fibonacci.countFibonacciSequence(size)
    }
}
```

Druga aplikacja ma na celu sprawdzenie wydajności języków w pracy z wyrażeniami regularnymi. Wykorzystuje ona wyrażenie zbudowane zgodnie z normą RFC 5322 [12], służące do testowania poprawności adresów e-mail. Każda implementacja składa się z dwóch plików: TestData, zawierającego przypadki testowe oraz MyBenchmark, zawierającego metodę testującą wraz z konfiguracją. Badanie przeprowadzono na trzech różnych przypadkach:

- Przypadek 1:
adres.mail:"/><(){}[]].mail@domena;'.ABC.domena.pl
- Przypadek 2: poprawny.adres@domena.pl
- Przypadek 3: krotki@adres.pl.

Listing 4: Test wyrażen regularnych - Java

```
//Klasa zawierająca przypadki testowe
import org.openjdk.jmh.annotations.Param;
import org.openjdk.jmh.annotations.Scope;
import org.openjdk.jmh.annotations.State;
@State(Scope.Thread)
public class TestData {
    @Param({
        "adres.mail:"/><(){}[]].mail@domena;'.ABC.domena.pl",
        "poprawny.adres@domena.pl",
        "krotki@adres.pl"})
    public String data;
}

//Test wyrażenia regularnego
import org.openjdk.jmh.annotations.*;
import java.util.concurrent.TimeUnit;

public class MyBenchmark extends TestData{
    @Benchmark
    @BenchmarkMode(Mode.Throughput)
    @OutputTimeUnit(TimeUnit.SECONDS)
    @Warmup(iterations = 5)
    @Threads(1)
    @Fork(2)
    @Measurement(iterations = 10)
    public Boolean checkRegex() {
        String regex = "(?:[a-z0-9!#$%&'*/+=?^_`{|}~.-]+(?:\\.[a-z0-9!#$%&'*/+=?^_`{|}~.-]+)*\\(?:\\x01-\\x08\\x0b\\x0c\\x0e-\\x1f\\x21\\x23-\\x5b\\x5d-\\x7f)\\\\\\\\[\\x01-\\x09\\x0b\\x0c\\x0e-\\x7f])*)@";
    }
}
```

```
\\x7f])*)@"(?:[a-z0-9!#$%&'*/+=?^_`{|}~.-]+(?:\\.[a-z0-9!#$%&'*/+=?^_`{|}~.-]+)*\\(?:\\x01-\\x08\\x0b\\x0c\\x0e-\\x1f\\x21\\x23-\\x5b\\x5d-\\x7f)\\\\\\\\[\\x01-\\x09\\x0b\\x0c\\x0e-\\x7f])*)@";
return data.matches(regex);
}
```

Listing 5: Test wyrażen regularnych – Scala

```
//Klasa zawierająca przypadki testowe
import org.openjdk.jmh.annotations.{Param, Scope, State}

@State(Scope.Thread)
class TestData {
    @Param(Array(
        "adres.mail:"/><(){}[]].mail@domena;'.ABC.domena.pl",
        "poprawny.adres@domena.pl",
        "krotki@adres.pl"))
    var data : String = ""
}

//Test wyrażenia regularnego
import java.util.concurrent.TimeUnit
import org.openjdk.jmh.annotations._

class MyBenchmark extends TestData {
    @Benchmark
    @BenchmarkMode(Array(Mode.Throughput))
    @OutputTimeUnit(TimeUnit.SECONDS)
    @Warmup(iterations = 5)
    @Threads(1)
    @Fork(2)
    @Measurement(iterations = 10)
    def checkRegex() : Boolean = {
        val regex = "(?:[a-z0-9!#$%&'*/+=?^_`{|}~.-]+(?:\\.[a-z0-9!#$%&'*/+=?^_`{|}~.-]+)*\\(?:\\x01-\\x08\\x0b\\x0c\\x0e-\\x1f\\x21\\x23-\\x5b\\x5d-\\x7f)\\\\\\\\[\\x01-\\x09\\x0b\\x0c\\x0e-\\x7f])*)@";
        regex.pattern.matcher(data).matches()
    }
}
```

Listing 6: Test wyrażen regularnych – Kotlin

```
//Klasa zawierająca przypadki testowe
import org.openjdk.jmh.annotations.*

@State(Scope.Thread)
open class TestData {
    @Param(
        "adres.mail:"/><(){}[]].mail@domena;'.ABC.domena.pl",
        "poprawny.adres@domena.pl",
        "krotki@adres.pl")
    var data : String = ""
}

//Test wyrażenia regularnego
import org.openjdk.jmh.annotations.*
```

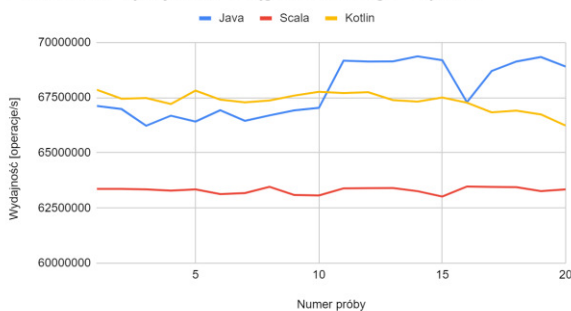
```
import java.util.concurrent.TimeUnit

open class MyBenchmark : TestData() {
    @Benchmark
    @BenchmarkMode(Mode.Throughput)
    @OutputTimeUnit(TimeUnit.SECONDS)
    @Warmup(iterations = 5)
    @Threads(1)
    @Fork(2)
    @Measurement(iterations = 10)
    fun checkRegex(): Boolean {
        val regex = "(?:[a-z0-9!#$%&'*/=?^`{}~-]+(?:\\.[a-z0-9!#$%&'*/=?^`{}~-]+)*|\"(?:\\\\x01-\\\\x08\\\\x0b\\\\x0c\\\\x0e-\\\\x1f\\\\x21\\\\x23-\\\\x5b\\\\x5d-\\\\x7f)\\\\\\\\\\\\\\\\\\\\x01-\\\\x09\\\\x0b\\\\x0c\\\\x0e-\\\\x7f)\"*)\"@{?:([a-z0-9]{?:[a-z0-9-]*[a-z0-9])?\\\\.)+[a-z0-9]{?:[a-z0-9-]*[a-z0-9])?|\\\\[?:(?:25[0-5]|2[0-4][0-9]|01?[0-9][0-9]?\\\\.){3}(?:25[0-5]|2[0-4][0-9]|01?[0-9][0-9]?|\\\\[a-z0-9-]*[a-z0-9]:(?:\\\\x01-\\\\x08\\\\x0b\\\\x0c\\\\x0e-\\\\x1f\\\\x21-\\\\x5a\\\\x53-\\\\x7f)\\\\\\\\\\\\\\\\\\\\x01-\\\\x09\\\\x0b\\\\x0c\\\\x0e-\\\\x7f)+)\\\\\\\\}"
        return data.matches(regex.toRegex())
    }
}
```

5. Wyniki badań

Zamieszczone poniżej rysunki 3-5 prezentują wyniki pomiarów wydajności aplikacji wyliczającej wartość zadanego wyraz ciągu Fibonacciego. Wynika z nich, że w większości przypadków Java okazała się mniej wydajna od pozostałych języków, które uzyskały zbliżone wartości. Porównanie wszystkich trzech wykresów ukazuje, jak bardzo nieoptymalny jest zastosowany algorytm rekurencyjny. Dla wyliczania wyrazu 50. wydajność wszystkich języków znacząco spadła, co przekłada się także na mniejszą dokładność otrzymanych wyników.

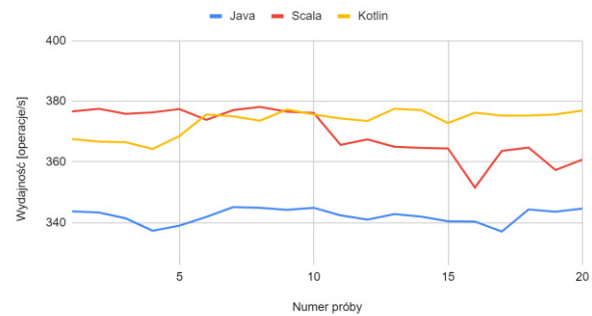
Porównanie wydajności - Ciąg Fibonacciego - wyraz 5



Rysunek 3: Porównanie wydajności – Ciąg Fibonacciego – wyraz 5

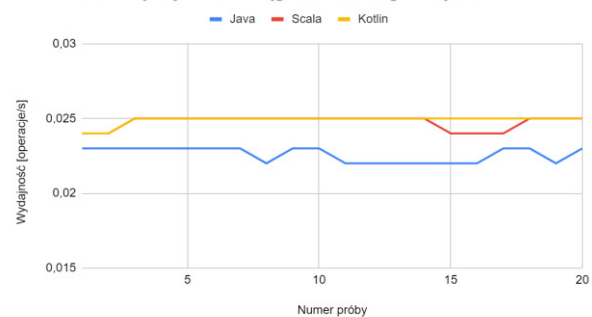
Tabela 1 przedstawia miary statystyczne dla uzyskanych pomiarów. Wynika z niej, że Kotlin w dwóch przypadkach okazał się najbardziej wydajnym językiem. W jednym przypadku była to Java. Scala w dwóch przypadkach plasuje się pomiędzy wynikami pozostałych języków, lecz jej rezultaty zbliżone są do wyników Kotlin. Pogrubieniem oznaczono najwyższe średnie wyniki.

Porównanie wydajności - Ciąg Fibonacciego - wyraz 30



Rysunek 4: Porównanie wydajności – Ciąg Fibonacciego – wyraz 30

Porównanie wydajności - Ciąg Fibonacciego - wyraz 50



Rysunek 5: Porównanie wydajności – Ciąg Fibonacciego – wyraz 50

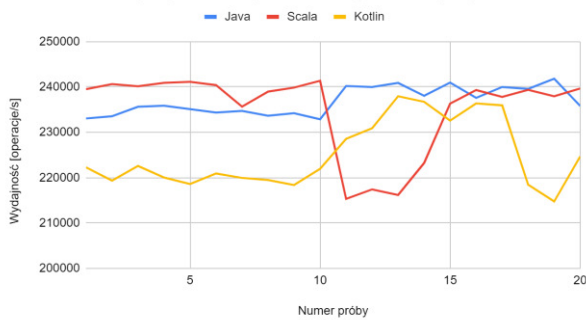
Tabela 1: Miary statystyczne – Ciąg Fibonacciego

Ciąg Fibonacciego – wyraz 5			
	Java [op/s]	Scala [op/s]	Kotlin [op/s]
Średnia	67841531	63302384	67342733
Odchylenie standardowe	1218689	138881	407298
Ciąg Fibonacciego – wyraz 30			
	Java [op/s]	Scala [op/s]	Kotlin [op/s]
Średnia	342,2261	369,5602	373,3022
Odchylenie standardowe	2,4362	7,9624	4,1432
Ciąg Fibonacciego – wyraz 50			
	Java [op/s]	Scala [op/s]	Kotlin [op/s]
Średnia	0,0226	0,0248	0,0249
Odchylenie standardowe	0,0005	0,0004	0,0003

Kolejne próbki w badaniu wydajności aplikacji testującej wyrażenia regularne, zaprezentowane na rysunkach 6-8 okazały się mieć znaczne wahania wartości, szczególnie w przypadku próbek 11-20. We wszystkich trzech przypadkach Java uzyskała najmniejsze wahania. Najbardziej pod tym względem prezentuje się Scala.

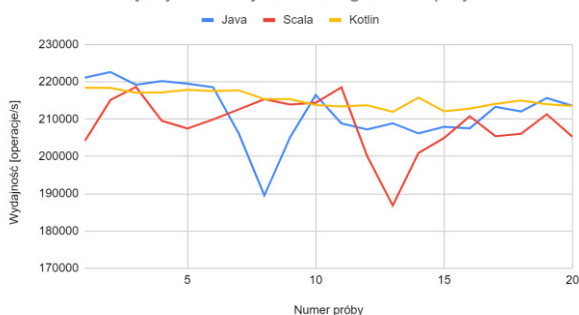
Tabela 2. prezentuje miary statystyczne dla uzyskanych pomiarów, zgodnie z którymi Java uzyskała w dwóch przypadkach największą wydajność przy jednocześnie najmniejszych wahanach wartości, jednak wartości średnie wydajności wszystkich języków są do siebie zbliżone. W jednym przypadku najbardziej wydajny okazał się Kotlin. Pogrubieniem oznaczono wyniki o najwyższej średniej wartości.

Porównanie wydajności - wyrażenia regularne - przykład 1



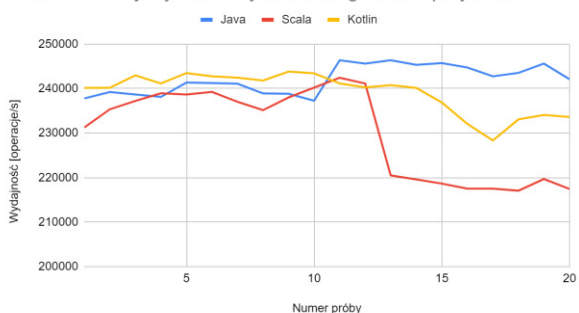
Rysunek 6: Porównanie wydajności wyrażenie regularne – przykład 1

Porównanie wydajności - wyrażenia regularne - przykład 2



Rysunek 7: Porównanie wydajności – wyrażenie regularne, przykład 2

Porównanie wydajności - wyrażenia regularne - przykład 3



Rysunek 8: Porównanie wydajności – wyrażenie regularne, przykład 3

Tabela 2: Miary statystyczne – wyrażenia regularne

Wyrażenie regularne – przykład 1			
	Java [op/s]	Scala [op/s]	Kotlin [op/s]
Średnia	236894	235056	225022
Odchylenie standardowe	3027,2212	8965,2080	7410,1065
Wyrażenie regularne – przykład 2			
	Java [op/s]	Scala [op/s]	Kotlin [op/s]
Średnia	211979	208555	215243
Odchylenie standardowe	7779,2440	7423,2723	2123,2199
Wyrażenie regularne – przykład 3			
	Java [op/s]	Scala [op/s]	Kotlin [op/s]
Średnia	242030	230130	239129
Odchylenie standardowe	3203,9172	10039,6288	4498,0414

6. Wnioski

Podsumowując, przeprowadzone badania nie wykazały znacznych różnic w wydajności wszystkich badanych języków, lecz Java oraz Kotlin uzyskały po trzy razy najlepsze średnie wyniki na sześć przeprowadzonych testów. Scala trzykrotnie uzyskała najgorszy średni wynik.

Podobne dane zaprezentowano w artykule na podstawie pracy magisterskiej [13], porównującym wybrane implementacje Java z serwisu [14] z implementacjami przekonwertowanymi z Kotlin oraz własnymi, stworzonymi w sposób zgodny ze stylem oraz możliwościami oferowanymi przez Kotlin. Dla sześciu przebadanych algorytmów, implementacja Javy okazała się szybsza niż obie implementacje w Kotlinie, a w kolejnych dwóch przypadkach implementacja Javy oraz implementacja własna autora wykazały zbliżony czas wykonania algorytmu, jednak dłuższy niż w przypadku implementacji będącej konwersją aplikacji Java do Kotlin. Przytoczone wyżej badania nie zostały przeprowadzone przy użyciu biblioteki Java Benchmark Harness.

Przeprowadzanie badań wydajności aplikacji działających na Java Virtual Machine wiąże się z koniecznością uwzględnienia charakterystyki jej działania. Obarczone jest to także błędami wynikającymi z pracy ekosystemu, w jakim badania są przeprowadzane, a więc innych procesów tworzących przerwanie i powodujących zmianę przetwarzanego przez procesor wątku. Wpływ na wyniki mają także dostępne zasoby sprzętowe.

Obecnie na rynku dostępne są narzędzia wspierające przeprowadzanie testów wydajności, takie jak biblioteka Java Benchmark Harness, pozwalające na uzyskanie możliwie najbardziej obiektywnych wyników, uwzględniających zasady działania środowiska uruchomieniowego JVM.

Literatura

- [1] Data is beautiful: Most popular programming languages 1965 - 2019
<https://www.youtube.com/watch?v=Og847HVwRSI> [18.10.2019]
- [2] TechCrunch.com: Google makes Kotlin a first class language for writing android apps
<https://techcrunch.com/2017/05/17/google-makes-kotlin-a-first-class-language-for-writing-android-apps/> [18.10.2019]
- [3] B. Evans, Java: The Legend, Wydawnictwo O'Reilly Media, Sebastopol 2015
- [4] J. Engel, Programming for the Java™ Virtual Machine, Addison Wesley, Boston 1999
- [5] T. Lindholm, The Java Virtual Machine Specification. Java SE 8 Edition, Oracle Parkway, Redwood City 2014
- [6] Hunt C., John B.: Wydajność Javy, Wydawnictwo Helion, Gliwice 2013
- [7] J. Kubryński, Co każdy programista Java powinien wiedzieć o JVM, Programista, Nr 3/2015, s. 24-27

- [8] Tutorials.Jenkov.com: JMH - Java Microbenchmark Harness <http://tutorials.jenkov.com/java-performance/jmh.html> [09.02.2020]
- [9] Dokumentacja Apache Maven <https://maven.apache.org/guides> [09.02.2020]
- [10] Dokumentacja języka Scala <https://docs.scala-lang.org/> [12.01.2020]
- [11] Dokumentacja języka Kotlin <https://kotlinlang.org/docs/reference/> [26.01.2020]
- [12] Dokumentacja normy RFC 5322 <https://tools.ietf.org/html/rfc5322> [29.02.2020]
- [13] Medium.com, J. Anioł - Java vs. Kotlin — Part 1: Performance: <https://medium.com/rsq-technologies/comparative-evaluation-of-selected-constructs-in-java-and-kotlin-part-1-dynamic-metrics-2592820ce80> [12.03.2020]
- [14] The Computer Language Benchmarks Game: Java versus Kotlin fastest programs <https://benchmarksgame-team.pages.debian.net/benchmarksgame/fastest/java-kotlin.html> [12.03.2020]

The "Quizer" e-learning platform as a tool for creating interactive quizzes with multiplayer functionality

Platforma e-learningowa „Quizer” jako narzędzie tworzenia interaktywnych quizów z funkcjonalnością multiplayer

Krzysztof Dziedzic^{a,*}, Marek Gudków^b, Tomasz Wiśniewski^c

^a Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

^b Euro-Forum, ul. Graniczna 4/7, 820-010 Lublin, Poland

^c Charger, ul. Kopernika 20/17, 21-040 Świdnik, Poland

Abstract

Modern IT and teaching solutions are increasingly used in the distance learning process. The introduction of game elements in the e-learning allows to increase the interest and motivation of the user when performing tasks. The "Quizer" e-learning platform has been designed to create interactive multimedia courses with elements of gameplay, including the multiplayer format. The paper discusses the technologies and tools used to build the platform. An example of a game designed in a multiplayer format is presented.

Keywords: e-learning; e-learning platform; quiz wizard; game

Streszczenie

W procesie kształcenia na odległość coraz częściej stosowane są nowoczesne rozwiązania informatyczne i dydaktyczne. Wprowadzenie do e-learningu elementów rozrywki pozwala na zwiększenie zainteresowania i motywacji użytkownika podczas wykonywania stawianych mu zadań. Platforma e-learningowa „Quizer” powstała z myślą tworzenia interaktywnych kursów multimedialnych z elementami rozrywki, w tym w formacie multiplayer. W pracy omówiono zastosowane do budowy platformy technologie i narzędzia. Przedstawiono przykład gry wykonanej w formacie multiplayer.

Słowa kluczowe: e-learning; platforma e-learningowa; kreator ćwiczeń; gra

*Corresponding author

Email address: k.dziedzic@pollub.pl (K. Dziedzic)

©Published under Creative Common License (CC BY-SA v4.0)

1. Introduction

Modern information technologies are increasingly used in education as well as trainings and courses. The use of a laptop, projector, interactive board, the Internet and specialised computer software has become a standard in the work of many teachers [1-3]. Constant progress in information technology and new technologies prompts search and, consequently, the emergence of new solutions that meet the needs of teachers. This is also the case in distance learning [4]. Information technologies supporting distance learning have been evolving since the 1980s. However, it was only the development of the Internet and its dissemination that caused the rapid development of e-learning. Currently, virtually every school, university or training centre uses e-learning to some extent in its operations [5]. In their work numerous scientists note the positive aspects of the use of e-learning in the process of educating pupils and students. They point to the improvement of their 'hard' skills such as the use of new technologies, as well as 'soft' skills such as communication and presentation skills [6]. Currently, Moodle, Udemy, Olat, LinkedIn Learning and Blackboard are the most commonly used e-learning platforms. They enable course management and are often referred to as LMS (Learning Management System) or LCMS (Learning Content Management System) platforms [7,8]. SCORM and newer xAPI are

the most commonly used formats for sharing didactic content. Among the tools for creating courses, the solutions that one may choose from include Articulate Storyline, Articulate Rise, Adobe Captivate, iSpiraling, Gomo or Active Presenter free for individual users. Extensions are also being developed that add new opportunities to commercial solutions present on the market. An example thereof would be the MoodleRec plugin for content search presented in De Medio's paper [9]. MoodleRec can sort a set of supported standard repositories of educational objects and suggest an ordered list of them, upon a simple query based on keywords. New possibilities for the development of technology and e-learning solutions are also sought and researched in many works. In his paper, Marciniak presented intelligent e-learning systems. This is a new approach to e-learning that allows to conduct learning in an adaptive manner, so that when giving content it is possible to take into account the specific needs of learners. The personalised approach is especially useful when working with large groups of learners. In the e-learning systems discussed by the author, the ability to make independent decisions concerning what teaching content is to be delivered to the learner in response to identified needs is implemented with the use of so-called teaching strategies and an IT module implemented in the LMS system, the so-called ITS agent. The basic function of

an ITS Agent is to confront the teacher's expectations with the learner's real progress and make decisions as to the content provided. There is a lot of research and development work being carried out that deals with the problem of personalisation of the curriculum, without focusing on a specific LMS platform [10]. Dorça et al. shows three different strategies for automatic detection and accurate adaptation of the learning content taking into account student results [11]. The need to monitor the e-learning process is vital in this situation, which is emphasised in many works. This facilitates the adaptation and personalisation of courses by the teacher, and in many cases also by the student. The e-learning platform used at Kajaani University of Applied Sciences, applying adaptive mechanisms for adjusting courses for students' needs, is an example here [12].

The constantly growing popularity of computer games, especially among young people, should be used in educational processes. Gamification is an approach applied in the educational process, which is currently gaining popularity and is being used more and more frequently. Introducing elements of gameplay allows to boost the user's interest and motivation in the content presented. Gamification is successfully applied to educational games: elements of strategy and gameplay are combined with the educational content [13]. Participants of such gameplay often have to show prompt reactions and knowledge connected with the questions asked. Their activity and correctness of responses are measured and recorded for further analysis [14]. Simulation games are increasingly used in higher education with students, in particular future engineers, doctors and pilots [15]. Modern solutions such as virtual and augmented reality as well as artificial intelligence combined with simulation games allow to enhance and increase the realism of the presented content [16]. This contributes to a better understanding of them.

In the distance learning process, elements of gameplay may also be successfully introduced into the process of both employee training and student education [17]. Designing gamification in e-learning requires a multidisciplinary approach and includes, among others, areas of game design, information technology, psychology or pedagogy. Its application should focus on the long-term activity to increase or maintain the level of commitment and motivation of pupils, students or employees. Positive effects of using a programming course with elements of gamification via an e-learning platform are presented in Pankiewicz's paper. He draws attention to the need for further research and development work on the use of elements of gamification in e-learning [18].

This article is devoted to a new solution in e-learning, namely the "Quizer" platform for creating interactive multimedia courses.

2. Construction of the Quizer e-learning platform

The "Quizer" e-learning platform was designed to create interactive multimedia courses. It is characterised by the functionality of a multimedia course creator that enables

the introduction of elements of gamification and a module that monitors user activity during quiz solving. The platform is a new solution targeted at companies and institutions as well as at schools and universities that would like to improve the qualifications of their employees, teachers and students, conduct online trainings as well as monitor the progress of the implementation of training plans. It allows the preparation of training materials, presentations as well as final tests with extensive statistics of final results. The ability to access courses from various types of mobile devices has reduced the barrier of having to conduct stationary trainings. The QUIZER platform is also an alternative for companies that professionally deal with the preparation and conduct of trainings. Thanks to the solutions applied in the QUIZER platform, one may effectively prepare stationary as well as online trainings, monitor and check the results of carried out activities. The diagram of the Quizer interactive e-learning platform is shown in Figure 1.

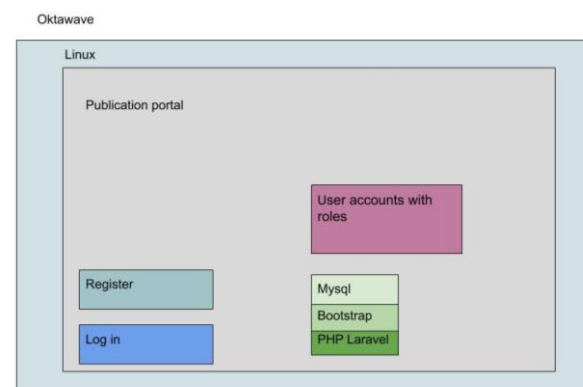


Figure1: The diagram of Quizer platform

Technologies such as Bootstrap, PHP Laravel, MySQL were used for its construction. The platform operates in the cloud. The cloud infrastructure used is Oktawave and it provides uniform support for the Quizer platform from various devices, including PCs, notebooks, tablets and smartphones. Its use increases reliability, data security and at the same time reduces the costs of the Quizer platform maintenance. The virtual application server was running on the Linux operating system. Comparison of average data recording speed for three different files is shown in Fig. 2.

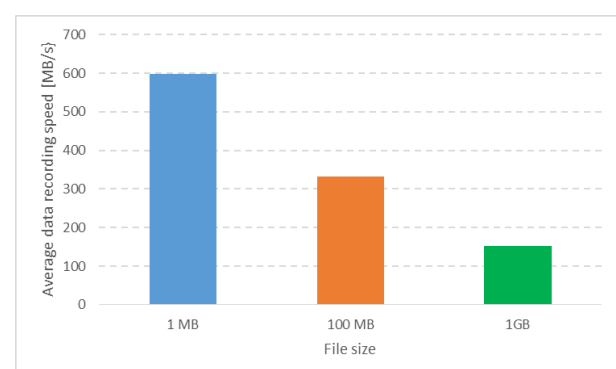


Figure2: Average data recording speed for Oktawave cloud

Cloud infrastructure was tested by attempting to record data in real time, taking into account different file sizes. During these attempts, 3 different file sizes were used, namely 1 MB, 100 MB and 1024 MB, corresponding to typical file sizes used and giving a comparison when attempting to record them in real time. An instance with the Debian system in version 4.9.65-3 was used for testing. The recording speed decreases as the file size increases. The recording time decreases as the file size decreases. The obtained results have confirmed the legitimacy of using cloud infrastructure to build the Quizer platform.

3. Interactive quiz wizard with the functionality of creating quizzes with elements of gamification in single-player and multiplayer formats

The Interactive Quiz Wizard is a tool that allows you to create quizzes. In the Quizer platform two solutions have been used: quizzes can be created in single-player (a single player) and multiplayer (many players) formats. The games may be played against a computer or other users. The quiz wizard screen when designing a quiz is shown in Fig. 3.



Figure 3: The quiz wizard screen when designing a quiz

In the first case, the single player wizard allows to create multimedia educational courses and presentations. Thanks to the built-in functions, it enables to create interactions such as pick one, pick many, drag and drop, type, crossword, word search, connect the lines and more. The wizard also enables to design conditional interactions, depending on previous user actions, using built-in or user-defined variables. By using variables, one may also monitor and examine user activity in a given course. The wizard also enables to assign, count and present points scored by individual users. It allows the use of text, graphics, audio, video as well as assigning various animations and transition effects of individual components on a single screen. With the option of random selection of screens, one can obtain the effect of several different quizzes in one package. The quizzes prepared in the wizard can be exported to SCORM or HTML formats and played on computers even without internet access. It is also possible to publish the quiz directly to the Presenter, thanks to which it is automatically located on the presentation platform, which constitutes a separate component of the QUIZER platform.

In the second case, the multiplayer wizard is a component of the QUIZER platform that enables to design multiplayer games for one or many users at the same time. The wizard makes it possible to define the type of games such as: pick one, pick many, type, drag and drop, who's first. When designing a game, we have influence on such game parameters as its name, description, type of interaction, number and type of questions, number of correct answers or answer variants for each question, random selection of questions from a defined set and the time for providing an answer. Questions may be in the form of text, graphics or audio. Thanks to the implemented logic of awarding points, one can receive additional points depending on the achievements of individual players in a group play. The diagram of the quiz wizard application with the technologies specified for its construction is shown in Fig. 4. The interactive quiz wizard consists of content management modules and an editor for creating courses that enable sending and downloading data from the publication portal. It operates in the Oktawave cloud infrastructure on Linux Debian.

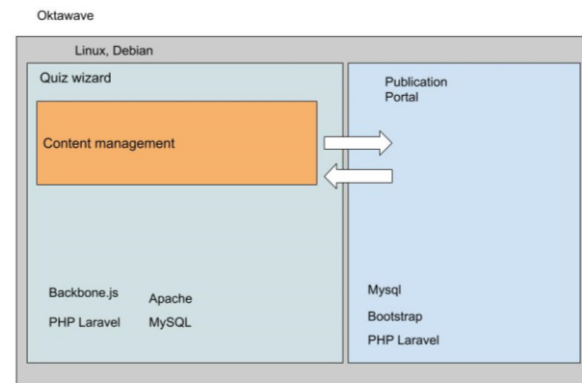


Figure 4: The diagram of a quiz wizard

4. Presenter function for single-player and multiplayer quizzes

A presenter is a very important component of the Quizer platform. As in the case of the wizard, one may distinguish a single-player and multiplayer presenter. The game edition in the presenter is shown in Fig. 5.

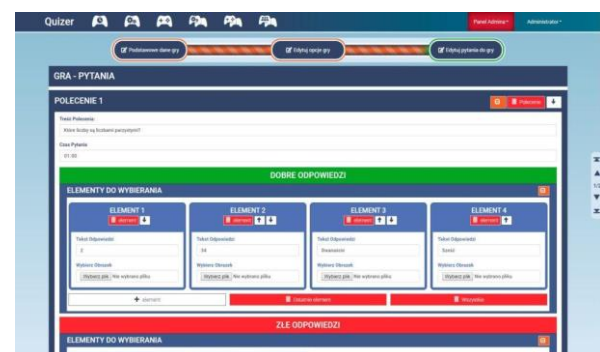


Figure 5: Game edition in the presenter

The single-player presenter is one of the components of the QUIZER platform that allows the presentation of

courses created in the wizard in a transparent and organised manner. By connecting the wizard with the presenter, it is possible to immediately publish courses directly to the presenter without the need for additional operations. Thanks to the presentation platform, one may create end-user and parent accounts, assign accounts to defined groups, and assign courses to specific groups or users. Thanks to the solutions introduced, courses on the presenter platform may be accessed from a personal computer, tablet and smartphone. Users receive points for completing courses, so that course progress and the degree of knowledge acquisition may be monitored.

The multiplayer presenter is a component of the QUIZER platform that enables to conduct a game among numerous players at the same time. The player has a possibility to create a game in the wizard component or choose one of those previously prepared. After choosing a game, the user can invite friends to the game or set its status to public. In this situation, all logged-in users will be able to join the game. The option of notifications about new available games allows users to stay informed about invitations to participate in the game. Games prepared in the creator can have a group (cooperative) or competitive character. Depending on the chosen variant, the game proceeds according to established rules.

The Vue.js library that is used to build Single Page applications has been used in multiplayer games. It can support the structure, logic and style of a component in one file, enabling to create flexible components very easily. The created components may also be reused in other projects. Vue.js allows the use of small reactive components in existing templates. The choice of library was dictated by the small file size (18 KB). This allows Vue.js users to separate the template for the virtual DOM (document object model) from the compiler, thus reducing execution time. Despite its size, Vue.js has an advantage over extensive frameworks such as AngularJS or Ember JS. Vue uses the most frequently used internet technologies. It uses the EventBus mechanism that enables to keep the status of events between active users on an ongoing basis. Listing 1 demonstrates a code snippet that shows what variables are passed between users who play multiplayer games.

Listing 1: Sample program code illustrating what variables are passed between users who play multiplayer games

```
export const EventBus = new Vue({
  data: function() {
    return {
      playerId: null,
      username: "",
      questions: {},
      options: {},
      currentQuestionNumber: 1,
      questionsAmount: 0,
      gameStatus: 'inHall', // ['inHall', 'waiting',
        'started', 'finished']
      playerSeqNumber: 0,
      playerData: {},
      questionTypesForShuffle: {},
      questionOrder: {},
```

```
      gameType: "",
      shuffledQuestionTypes: null,
      allElementClick: [],
      isPlayerRoomOwner: false,
      playerRoomId: false,
      // for fixing doubled rendering of
      ConnectedUsers
      hasRegisteredGameStatusListener: false,
      gameMode: null,
      maxGamePoints: 0
    }
  }
})
```

Vue.js uses standard HTML-based templates and single file components, which has allowed developers to write in a language used to describe the form of CSS presentation (with support for CSS modules and preprocessors) resulting in smaller complexity and compilation size. In the case of Vue.js, developing templates on a large scale is very simple, which was an important aspect when choosing this technology for the multiplayer system design process. In addition, the uncomplicated structure makes searching for code blocks containing errors fast.

When writing communication between users, the socket.io library was used, thanks to which data can be exchanged between the server and users. The initialisations of the socket.io library are shown in example 2.

Listing 2: Sample program code initialising the socket.io library

```
export const SocketInstance =
  socketIO(location.protocol + '//' +
    window.location.hostname + ':3011',{query:
      {playerName:
        document.getElementById('usernameValue').value,
        userId: userId}});
```

Bootstrap was used in the project as part of the multiplayer functionality because it belongs to the group of the most popular CSS libraries. It is a free open source tool created and developed by Twitter programmers. Bootstrap operates under MIT license, is based on ready-made CSS and HTML solutions and uses JavaScript. It has a number of solutions that significantly facilitate the work of creating a graphic interface for responsive websites and applications thanks to the option of styling components. Bootstrap is one of the solutions that allows the construction of responsive websites. A responsive website (RWD from Responsible Web Design) is a type of page that adjusts the layout of elements and its width to the screen resolution of the device on which it is displayed.

The advantages of responsive websites include:

- adjusting the displayed content depending on the device and its screen resolution,
- eliminating problems related to device detection, domain redirection and template switching,
- eliminating the need to update content in different places due to the fact that there is only one version of the site,
- eliminating the duplication of content (Duplicate content phenomenon) that occurs when identical text content appears in several URLs.

The Gulp tool was used to optimise the solution and make the multiplayer work faster in the project. Gulp is a task runner tool, whose aim is to automate tasks, and thus improve the programming workflow.

The game editors used the jQuery library, which optimally enabled the editing of HTML fields. jQuery is a JavaScript programming library that facilitates the manipulation of the DOM tree. jQuery includes efficient methods for the change and manipulation of attributes and HTML elements, dynamic addition of CSS styles to DOM tree elements, and manipulation of the DOM tree. DOM is a document object model that is a way of representing HTML and XML documents as an object model. It is independent of the programming language. DOM provides a structured representation of the document, which makes it possible to modify its content and virtual presentation. The use of jQuery library in the multiplayer game wizard has made it easier to manipulate the course structure.

The Laravel PHP framework is responsible for the Wizard and Presenter module on the backend side. Laravel has functionalities that allow easy implementation of authorisation. As a result, unauthorised persons do not gain access to protected content. Laravel uses MVC (Model-View-Controller), thanks to which developers have at their disposal a number of additional built-in functions and may develop applications at various levels of advancement. Predefined framework packages enable personalisation tailored to specific business requirements.

MySQL, which is a database with a number of additions created by various groups of programmers, is the database used in the project. These tools may be downloaded additionally, but the database also has extensions already built in, including tools that allow the use of graphical interfaces. MySQL also has functions to increase database security. Databases running on MySQL engine provide Immediate Consistency.

5. The "What animal is this?" game as an example of a multiplayer game

Educational games are a form of teaching that helps students acquire knowledge and skills. They should have a clearly defined teaching goal and be adapted to the age and abilities of students. The game rules applied should be comprehensible and respected by students. Educational games are especially attractive for teenagers and children. The interactive quiz wizard allows the creation of games in a single-player and multiplayer format adapted to virtually every age group and subject. It is possible to use several modes such as cooperative dynamic, competitive dynamic, cooperative turn-based, competitive turn-based. The game "What animal is this?" for the youngest children is an example of an educational game in the multiplayer format with a cooperative and dynamic mode. In the type of gameplay used, the game itself is the opponent of the group of players. For the game to be successful, a group of players must score a certain percentage result by working together (e.g. if players achieve 60% - the game is com-

pleted successfully, with a lower score - players lose the game). This is an example of semi-cooperation because all people pursue the same group goal, but they also earn points individually. The response time is limited. The game combines learning the names of animals and orientation. Students are asked to indicate which animal the person specified in the question has.

During the game players respond to:

- text questions that can be enriched with graphics, photos or audio files. Due to the size limitations of the screens of mobile devices, the number of response options to choose from amounts to a maximum of 4. These can be single or multiple choice questions,
- drag and drop questions consisting in assigning appropriate elements (words / graphics) to appropriate places (word order / group of items),
- type-in questions, enter the correct word / enter the appropriate number. These are input questions where the user must enter the predefined variable in the field to complete them.

An example of a game screen is shown in Fig. 6. The advantage of using games in this age group is the ability to create a situation where teaching is associated with fun. It enables to activate and motivate all students to learn as well as encourages the use of computer technologies to increase their knowledge.



Figure 6: Screen of the game in multiplayer format

Educational games are an alternative to other types of games that most frequently do not bring benefits, especially to the youngest children.

6. Conclusions

The Quizer e-learning platform is a new solution that enables the creation of interactive quizzes with the option of creating single-player and multiplayer games. It allows to make learning more attractive and increase user's interest and motivation by introducing elements of gamification into the educational process. The Quizer e-learning platform may be used by schools, universities, companies and institutions to create courses and trainings. The use of tools such as the Oktawave cloud infrastructure, Vue.js., Bootstrap, Gulp, Laravel, jQuery and MySQL database has enabled the development of an e-learning platform. Educational games in a single or

multiplayer format can be used to complement or enhance the traditional teaching process, but they should not completely replace it.

Acknowledgement: The article was financed as part of the project: "Creating an innovative interactive Quizer platform as a result of R&D works" co-financed by the European Union under the European Regional Development Fund, implemented in the Regional Operational Programme of the Lublin Voivodeship for 2014-2020, Priority Axis: 1 RESEARCH AND INNOVATION, Action 1.2 Purposeful research. Contract No. RPLU.01.02.00-06-0065/16

References

- [1] D. Wójcicka-Migasiuk, A. Urzędowski, Internet tools in education at different levels of teaching, *Advances in Science and Technology Research Journal*, 8(24), 2014, 66–71.
- [2] K. Dziedzic, M. Barszcz, M. Pańnikowska-Łukaszuk, A. Jankowska, The role of computer animation in teaching technical subjects, *Advances in Science and Technology Research Journal*, 9, 2015, 134–138.
- [3] K. Dziedzic, M. Włodarczyk, M. Pańnikowska-Łukaszuk, The usage of computer visualization in teaching technical subjects, *Advances in Science and Technology Research Journal* 8, 2014, 72–75.
- [4] A. Duda, S. Korga, S. Gnapowski, The role of e-learning in educational processes, *Advances in Science and Technology Research Journal*, 8(24), 2014, 61–65.
- [5] M. Zając, Nauczyciele wobec technologii informacyjnej i e-edukacji, *E-mentor*, 3(20), 2007.
- [6] A. Csikosova, A. Senova, K. Culkova, Improving of communication and presentation skills of the universities students trough e-learning, *Procedia - Social and Behavioral Sciences*, 46, 2012, 2847–2851.
- [7] V. Caputi, A. Garrido, Student-oriented planning of e-learning contents for Moodle, *Journal of Network and Computer Applications*, 53, 2015, 115–127.
- [8] C. De Medio, C. Limongelli, F. Sciarrone, M. Temperini MoodleREC: A recommendation system for creating courses using the moodle elearning platform, *Computers in Human Behavior*, 2019.
- [9] W. Prządka, Analiza porównawcza narzędzi e-learningu, *Journal of Computer Sciences Institute*, 2017, 64–69.
- [10] J. Marciniak, Inteligentne systemy e-learningowe jako przykład wykorzystania sztucznej inteligencji w kształceniu na odległość, *EduAkcja. Magazyn edukacji elektronicznej*, 2(12), 2016, 87–101.
- [11] F.A. Dorça, L.V. Lima, M.A. Fernandes, C.R. Lopes, Comparing strategies for modeling students learning styles through reinforcement learning in adaptive and intelligent educational systems: an experimental analysis, *Expert System Applications*, 40(6), 2013, 2092–2101.
- [12] K. C. Deepak, Evaluation of Moodle Features at Kajaani University of Applied Sciences – Case Study, *Procedia Computer Science*, 116, 2017, 121–128..
- [13] J. Montusiewicz, M. Barszcz, K. Dziedzic, J. Kęsik, M. Miłosz, M. Tokovarov, The concept of a 3D board game to recognize architectural monuments, *INTED 11th International Technology, Education and Development Conference. IATED Academy*, 2017, 8665–8674.
- [14] J. Montusiewicz, M. Miłosz, J. Kęsik, M. Barszcz, K. Dziedzic, M. Tokovarov, P. Kopniak, Developing an educational board game using information technology, *EDULEARN 17: 9th International Conference on Education and New Learning Technologies, Barcelona, IATED Academy*, 2017, 7396–7405.
- [15] J. B. Haugea, J. Riedel, Evaluation of simulation games for teaching engineering and manufacturing, *Procedia Computer Science*, 15, 2012, 210–220.
- [16] M. Dąbrowski, E-learning 2.0-przegląd technologii i praktycznych wdrożeń, *E-mentor*, 1(23), 2008.
- [17] K. Hejduk, Grywalizacja w e-learningu. Gra strategiczna on-line Akademii PARP „Gwiazdne szkolenia”, *Zarządzanie Zasobami Ludzkimi*, 3(103), 2015, 77–89.
- [18] M. Pankiewicz, Analiza wpływu wykorzystania elementów grywalizacyjnych w kursie e-learningowym, *EduAkcja. Magazyn edukacji elektronicznej*, 1(11), 2016, 36–42.

Analysis of typical programming mistakes made by first and second year IT students

Analiza typowych błędów programistycznych popełnianych przez studentów pierwszego i drugiego roku Informatyki

Monika Kaczorowska*

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

The research paper contains a review and analysis of common programming mistakes made by first and second year students of Computer Science. The data were collected during the courses entitled “Algorithmics and Data Structures” and “Numerical Methods”, where students have to write programs in the C++ language. The article includes examples of three selected mistake types. A comparison of mistakes made by first and second year students is presented. The analysis carried out shows that the percentage of mistakes made decreases when the students are in the second year, but three types of mistakes demonstrate the opposite trend. It can be concluded that those three types of mistakes are related to the course of Numerical Methods, where students have to deal with a significant amount of mathematical expressions. The results show that the students have the most significant problems with memory management.

Keywords: programming mistakes; C++; education

Streszczenie

Artykuł zawiera przegląd i analizę typowych błędów programistycznych popełnianych przez studentów pierwszego i drugiego roku Informatyki. Dane wykorzystane podczas analizy zostały zgromadzone w trakcie zajęć z przedmiotów: Algorytmy i Struktury Danych oraz Metody Numeryczne. Podczas zajęć studenci piszą programy w języku C++. Artykuł zawiera przykłady trzech wybranych typów błędów. W artykule przedstawione zostało porównanie błędów popełnianych przez studentów pierwszego oraz drugiego roku. Przeprowadzona analiza wykazała, że liczba popełnianych błędów jest mniejsza dla studentów drugiego roku ale przy trzech rodzajach błędów można było zaobserwować odwrotną tendencję. Błędy te powiązane są ze specyfiką przedmiotu Metody Numeryczne. Podczas tego przedmiotu studenci mają do czynienia w większym stopniu z wykonywaniem obliczeń matematycznych. Wyniki pokazują, że studenci mają największe problemy z zarządzaniem pamięcią.

Słowa kluczowe: błędy programistyczne; C++; edukacja

*Corresponding author

Email address: m.kaczorowska@pollub.pl (M. Kaczorowska)

©Published under Creative Common License (CC BY-SA v4.0)

1. Introduction

Programming is an essential part of any Computer Science course. IT students study several programming languages such as: C, C++, Python and Java. They write programs not only during classes dedicated to programming, but also use their programming skills during other classes, where they have to write some lines of code. It is said that the more programs are written, the fewer mistakes will appear, as it is obvious that every learner makes mistakes. It can be said that making mistakes in programming is a part of the learning process.

One of the first topics concerning programming mistakes appeared during a workshop on Empirical Studies of Programming [1]. In [2] the authors focused on the mistakes which students made in the Java language and how they managed them. There are papers where the authors analyse how long it takes students to solve a problem, for example [3]. In [4] the authors identified the most common errors made in Java. M. Hristova et al. made a list of common mistakes in Java and created the software which allowed to interpret the mistakes [5]. In [6] the authors presented the software which allowed

to interpret the mistakes and enabled students to learn Java programming as well. Neil C. C. Brown et al. in their paper [7] presented the BlackBox project which has been developed since 2013. The data have been collected from users of BuleJT IDE, which is dedicated for the Java programming language. Papers [8-9] present the mistakes based on the data collected from Blackbox. The authors analyse the frequency, time to fix and the spread of errors among users and their development during a year. In [10] and [11] the authors compared the ranking of Java programming mistakes made by educators and the ranking of programming mistakes made on the basis of data collected from BlackBox. S. Hubalovsky and J. Sediya wrote about mistakes in object oriented programming [12]. The authors mention that mistakes which do not cause syntax errors are not recognised by programmers, because programs containing them seem to work properly. In [13] the authors present common mistakes in OpenMP in C/C++. Interestingly, they also point out the good practices of using OpenMP. In paper [14] S. Júnior et al. analyse a class of mistakes which may prevent students

from obtaining a proper solution. The authors write that students do not often inform the teacher about their difficulties, and if they do it – they have a difficulty in defining their problem. The problem, it is argued, might be caused by the fact that students often do not understand the programming vocabulary. A. Stefik and S. Siebert found that the languages, that are closer to English may be more intuitive to beginner programmers [15]. In [16] the authors draw attention to the fact that the feedback focuses more on identifying mistakes and less on fixing the problems. To the best of the author's knowledge, research papers dedicated to mistakes in the C/C++ language are not numerous. Owing to this fact, the present analysis of the issue was conducted.

The purpose of the present paper is to review and analyse the common programming mistakes made by IT students. The analysis addressed programs of students of the Lublin University of Technology (LUT). Computer Science studies at the LUT include the following courses: "Programming in C" in the first semester, "Algorithms and Data Structures" in the second, and "Numerical Methods" in the third one. Students write programs in C/C++ during the courses "Algorithms and Data Structures" and "Numerical Methods". The methodology applied is discussed in section 2. The section includes a description of the data and a review of common programming mistakes. Section 3 contains the results and Section 4 concludes the paper.

2. Methodology

The programs of first and second year students were considered to analyse the programming mistakes in C++ language. Students wrote programs in C++ in a structural way. Students had the course of Algorithmics and Data Structures during the first year, and they had a course Numerical Methods during the second year. The students had to write the programs in C++ during both of courses. They had to send their programs via a dedicated platform or present their programs at the end of the classes. The mistakes which students made during the classes were also collected. The programs of the two courses mentioned of the same 43 students were analysed and their mistakes were collected.

2.1. Data

Eight programs were collected from each student during the Algorithms and Data Structures course and six programs during the Numerical Methods course. The total number of analysed files was 602.

The data which were collected during Algorithms and Data Structures concerned the following topics:

- finding array min/max,
- sorting,
- searching information in an array,
- data structures (implementing stack, queue, list or binary tree).

The data which were collected during Numerical Methods concerned the following topics:

- Newton and Lagrange interpolation,
- mean square approximation,

- searching for zero in nonlinear equations,
- numerical integration,
- solving systems of linear equations.

Students had to write a program where they had to create, for example, an array, implement an algorithm and test it.

2.2. Common programming mistakes

The classification of programming mistakes in structural C++ was defined and the programming mistakes were divided into three groups: syntax errors, memory management errors and logic errors. The first group includes errors which do not allow the program to compile. Memory management errors are related to the situations when the program wants to use a memory area which is not accessible. The third group contains logical errors which can sometimes be difficult to find, and some of them appear when somebody starts to learn programming and is not experienced enough. Each group is discussed in details below. The following list presents the most typical mistakes made in C/C++.

The first group, syntax errors:

1. No semicolon.
2. Invalid number of parameters passed to function.
3. No brackets or odd number of brackets, for example in expressions.
4. Variable names containing the space.
5. No library attached to file.
6. Visibility of variables in switch case section. Students tend to declare the variable in the case section and the program cannot be compiled.

The second group contains memory management mistakes:

7. Out of array range. Students use an array and sometimes they do not remember that the first index of it is 0 and the last one is the length of array minus one. They try to refer to the element, which is out of array.
8. Trying to access the memory which was not allocated. Students declare a pointer and, for example, try to read elements and write them into an array. If a student uses the pointer variable to access an array, he/she has to allocate the memory before using it.
9. Memory leak. Students do not delete the allocated memory for example when they use the pointers, define a queue, stack or list. This kind of errors can also appear when students dynamically allocate memory in functions and do not deallocate before end of function.
10. No passing variable by reference. Students create a function where they want to change the variable value and pass this variable into the function without reference.

The third group includes logical mistakes:

11. Shadowing variable. Students create a variable, for example, in the main function and they also create a variable with the same name and type in a loop, causing the first variable to be shadowed by the second one.

12. Incorrect condition in the conditional statement or loop. Students place only one '=' sign in a condition. The incorrect condition in the conditional statement and loop as well causes the wrong operation of a program. Sometimes students also do not know when the algorithm should be ended, and they do not define the correct condition which ends the algorithm.
13. Incorrect initial value of variable. Students do not remember that the neutral element of multiplication is 1 and the neutral element of addition is 0. When searching for a min value of an array they incorrectly assume that it is sufficient to put in the initial value of min equal to some big number. They do not understand that such an idea is not flexible and it is better to assume that the first element is the min and then start comparing it with other elements.
14. Uninitialised variable. For example, when students create an array, they forget that the variable defining the length of the array has first to be initialised and only then used.
15. Dividing two integers. The results of division of two integers is not necessarily an integer.
16. No brackets in condition statement or loop. Students do not remember that if they want to place more than one instruction in a condition statement or a loop, they need to use brackets, otherwise only the first instruction will be assigned to the condition statement or loop.
17. Use of magic numbers. Use of unnamed constants the sense of which is not clear.

Listings 1, 2 and 3 show examples of mistakes made by students. Listing 1 presents the mistake of using a dynamic array without memory allocation. Listing 2 shows an incorrect condition in a conditional statement or loop. Listing 3 illustrates a mistake in finding the min or max value in an array.

Listing 1: Using an array without memory allocation

```
int* tab;
for(int i=0;i<10;i++){
    cin>>tab[i];
}
```

Listing 2: Incorrect condition in a conditional statement or loop

```
int i=5;
if(i%2=1)
    cout<<"Odd number";
else
    cout<<"Even number";
```

Listing 3: Finding min value

```
int min=999;
for(int i=0; i<5;i++){
    if(tab[i]<min){
        min=tab[i];
    }
}
```

3. Results

The results of the research are presented below. If the same mistakes appeared several times during one class, it was calculated only one time. The numbers assigned to the specific mistake types in section 2.2 were used for the annotation of the figures presented below. Number 18 on the charts is related to other mistakes which are not specified in the classification. Figure 1 presents the number of mistakes made by first year students during the "Algorithms and Data Structures" course. The maximum possible number of mistakes amounts to 344, due to the fact that 43 programs were analysed during each of the 8 classes. The most common programming mistakes are related to memory management: out of the array range, trying to access the memory which was not allocated, memory leak or no passing variable by reference. The mistake related to initialising a variable with specific value appeared often as well.

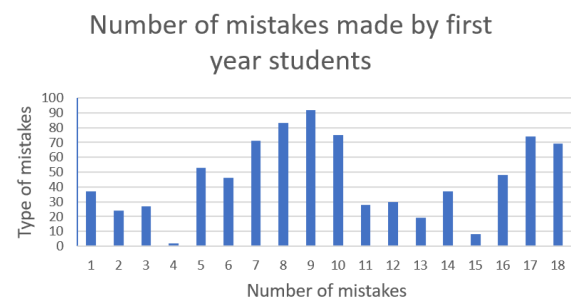


Figure 1: Number of mistakes made by first year students during the "Algorithm and Data Structures" course.

Figure 2. presents the number of mistakes made by second year students during the "Numerical Methods" course. The maximum possible number of mistakes equals 258, due to the fact that 43 programs were analyzed during each of the 6 classes. The most common programming mistakes are related to no passing variable by reference and performing calculations: no brackets or odd number of brackets, dividing two integers.

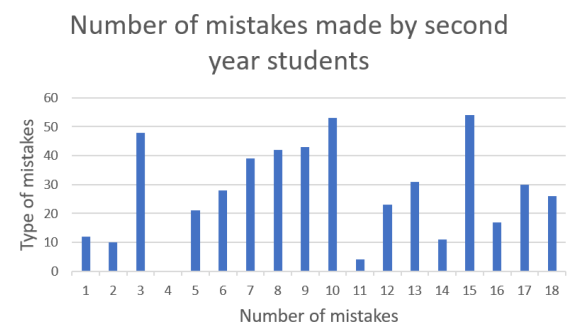


Figure 2: Number of mistakes made by second year students during the "Numerical methods" course.

Figure 3 shows a comparison between the mistakes made by first year students and the second year students during the "Algorithms and Data Structures" and "Numerical Methods" courses. It can be observed that the percent of mistakes decreased in most cases. The opposite situation was observed in the case of the following

mistakes: no brackets or odd number of brackets, incorrect initial value of variable and dividing two integers. A similar percentage is noticeable for mistakes related to an incorrect condition in a conditional statement or loop for first and second year students.

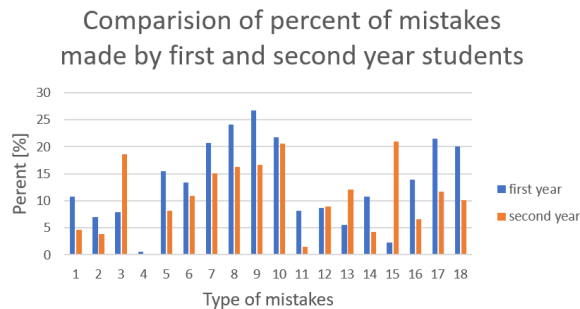


Figure 3: Comparison of percent of mistakes made by first and second year students during the “Algorithms and data structured” and “Numerical Methods” courses.

4. Conclusions

A classification of common programming mistakes in C++ and analysis of these mistakes were conducted. The obtained results show that a higher number of mistakes is made during the first year. Students have problems with syntactic errors, which are revealed during program compilation. The most problematic for students are mistakes related to memory management. This kind of mistakes could be difficult to detect for beginners. They may not be observed during each execution of a program and may require analysis of the code line by line. Second year students also have a problem with pointers. There are some mistakes which are made by second year students more often. This is caused by the situation that during the “Numerical Methods” course the students write more programs related to mathematical expressions and numbers. The mistakes related to “no brackets or odd number of brackets”, “incorrect initial value of variable” or “dividing two integers” were made mostly by students of the second year. It can be observed that only one mistake related to dividing two integers constitutes about 20% of all of students errors.

The obtained results may help to adjust the design of the course and textbooks. The results also show which topics the teacher should focus on more. The article seems to be useful not only for academic teachers and students, but also for school teachers and learners. Further research will focus on a bigger group of students and the object oriented approach to programming in C++.

Literature

- [1] E. Soloway, S. Iyengar, editors, *Empirical Studies of Programmers: Papers Presented at the First Workshop on Empirical Studies of Programmers*, Intellect Books (1986).
- [2] M. C. Jadud. Methods and tools for exploring novice compilation behaviour. In *Proceedings of the Second International Workshop on Computing Education Research*, ICER '06, New York, NY, USA (2006) 73-84.
- [3] P. Denny, A. Luxton-Reilly, E. Tempero, All syntax errors are not equal, In *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE'12, New York, NY, USA (2012) 75-80.
- [4] J. Jackson, M. Cobb, C. Carver, Identifying top Java errors for novice programmers, In *Frontiers in Education*, 2005. FIE '05. Proceedings 35th Annual Conference (2005).
- [5] M. Hristova, A. Misra, M. Rutter, R. Mercuri, Identifying and correcting Java programming errors for introductory computer science students, In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '03, New York, NY, USA (2003) 153-156.
- [6] T. Sirkiä, J. Sorva, Exploring programming misconceptions: an analysis of student mistakes in visual program simulation exercises, In *Proceedings of the 12th Koli Calling International Conference on Computing Education Research* (2012) 19-28.
- [7] N. C. C. Brown, M. Kölling, D. McCall, I. Utting. Blackbox, A large scale repository of novice programmers' activity, In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, SIGCSE '14, New York, NY, USA (2014) 223-228.
- [8] A. Altmirri, N. C. Brown, 37 million compilations: Investigating novice programming mistakes in large-scale student data, In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (2015) 522-527.
- [9] N. C. Brown, A. Altmirri, Investigating novice programming mistakes: educator beliefs vs. student data, In *Proceedings of the tenth annual conference on International computing education research* (2014) 43-50.
- [10] A. Ahadi, R. Lister, S. Lal, A. Hellas, Learning programming, syntax errors and institution-specific factors, In *Proceedings of the 20th Australasian computing education conference* (2018) 90-96.
- [11] N. C. Brown, A. Altmirri, Novice Java programming mistakes: Large-scale data vs. educator beliefs, *ACM Transactions on Computing Education (TOCE)*, 17(2) (2017) 1-21.
- [12] Š. Hubálovský, J. Šedivý, Mistakes in object oriented programming, *Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej, Technologie Informacyjne*, 18, (2010) 205-210.
- [13] M. Süß, C. Leopold, Common mistakes in OpenMP and how to avoid them, In *International Workshop on OpenMP*, Springer, Berlin, Heidelberg (2005) 312-323.
- [14] A. S. Júnior, J. C. A. de Figueiredo, D. Serey, Analyzing the Impact of Programming Mistakes on Students' Programming Abilities, In *Brazilian Symposium on Computers in Education*, vol. 30, No. 1, (2019) 369.
- [15] S. Andreas, S. Susanna, An Empirical Investigation into Programming Language Syntax. *Trans. Comput. Educ.* 13, 4, Article 19 (2013).
- [16] H. Keuning, J. Jeuring, B. Heeren, A systematic literature review of automated feedback generation for programming exercises. *ACM Transactions on Computing Education (TOCE)*, 19(1) (2018) 1-43.

Object classification using X-ray images

Klasyfikacja obiektów na podstawie ich zdjęć rentgenowskich

Piotr Nowosad* , Małgorzata Charytanowicz

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

The main aim of the presented research was to assess the possibility of utilizing geometric features in object classification. Studies were conducted using X-ray images of kernels belonging to three different wheat varieties: Kama, Canadian and Rosa. As a part of the work, image processing methods were used to determine the main geometric grain parameters, including the kernel area, kernel perimeter, kernel length and kernel width. The results indicate significant differences between wheat varieties, and demonstrates the importance of their size and shape parameters in the classification process. The percentage of correctness of classification was about 92% when the k-Means algorithm was used. A classification rate of 93% was obtained using the K-Nearest Neighbour and Support Vector Machines. Herein, the Rosa variety was better recognized, whilst the Canadian and Kama varieties were less successfully differentiated.

Keywords: object classification; geometric features; image processing; X-ray imaging

Streszczenie

Głównym celem artykułu było zbadanie możliwości wykorzystania cech geometrycznych obiektów w procesie ich klasyfikacji. Materiał badawczy stanowiły zdjęcia rentgenowskie ziaren trzech odmian pszenicy: kama, kanadyjskiej i rosa. W ramach pracy opracowano metody pozwalające na wyznaczenie cech geometrycznych obiektów znajdujących się na obrazach cyfrowych, takich jak długość, szerokość, średnica, pole i obwód. Otrzymane wyniki wykazały istotne różnice pomiędzy parametrami charakteryzującymi kształt i wielkości poszczególnych odmian pszenicy i możliwość ich zastosowania w procesie klasyfikacji. Procent poprawnie zaklasyfikowanych ziaren za pomocą algorytmu k-średnich wynosił 92%. Nieco lepsze wyniki, rzędu 93%, uzyskano za pomocą metod K-najbliższych sąsiadów i wektorów wspierających. Najlepiej rozróżnialną odmianą okazała się rosa w porównaniu do odmian kanadyjskiej i kama.

Słowa kluczowe: klasyfikacja obiektów; cechy geometryczne; przetwarzanie obrazów; obrazowanie rentgenowskie

*Corresponding author

Email address: piotr.nowosad@pollub.edu.pl (P. Nowosad)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Dynamiczny rozwój technik informacyjnych pozwolił na opracowanie nowych metod gromadzenia, przetwarzania i analizy informacji. Możliwe stała się praktyczne zastosowanie wielu algorytmów, które dotychczas ze względu na ograniczenia sprzętowe i czasowe nie mogły być stosowane. Nowoczesne technologie pozwoliły również na wykorzystanie obrazu jako źródła danych. Metody przetwarzania i analizy obrazów cyfrowych stanowią obecnie jedną z ważniejszych gałęzi informatyki [1, 2].

Procesem łączącym wstępne przetwarzanie obrazu z jego analizą jest segmentacja, polegająca na wyodrębnianiu z obrazu jednorodnych obszarów według przyjętego z góry kryterium. Uważana jest ona za jedną z najtrudniejszych i najważniejszych operacji, umożliwiającą identyfikację obiektów oraz ich opis ilościowy i jakościowy. Z drugiej strony połączenie statystycznych metod analizy danych z technikami komputerowymi pozwoliło na rozwój działu statystyki zwanego wielowymiarową analizą klasyfikacyjną i analizą skupień [3, 4]. Systemy uczące się są obecnie podstawą inteligentnej analizy danych [5]. Wyróżniamy systemy uczące się pod nadzorem, wśród których warto wspomnieć o różnego typu klasyfikatorach, oraz systemy uczące się bez nadzoru, których zadaniem jest wykry-

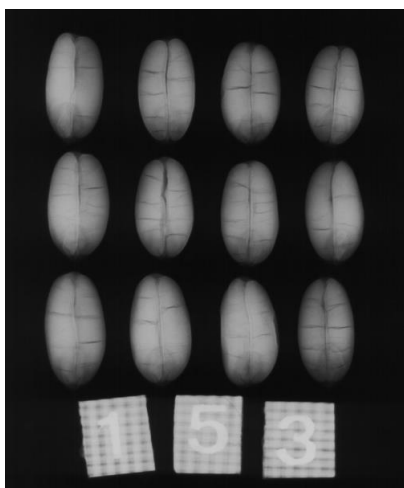
wanie wewnętrznej struktury zbioru danych. Przykładem jest tu analiza skupień [6, 7]. Rezonans magnetyczny, tomografia komputerowa, skanery i systemy wizyjne umożliwiły uzyskanie obrazów wysokiej jakości, stanowiących cenne źródło informacji w naukach technicznych, medycznych i przyrodniczych [8-10].

Celem niniejszej pracy jest zbadanie możliwości wykorzystania cech geometrycznych obiektów do ich klasyfikacji. Materiał badawczy stanowią zdjęcia rentgenowskie ziaren pszenicy. Zastosowanie metod przetwarzania obrazów cyfrowych do tego typu zagadnień jest często stosowane w praktyce. W pracy [11] przedstawiono metodę oceny zmian właściwości ziarna w zależności od warunków zewnętrznych [11]. Za pomocą progowania i filtracji z obrazów ziaren uzyskano obrazy binarne, które wykorzystano do pozyskania danych w celu przeprowadzenia dalszych analiz. Zagadnienie wyznaczenia cech geometrycznych odmian ziarna pszenicy, na podstawie których możliwa jest ich dyskryminacja, omówione zostało w artykułach [12-14]. Do identyfikacji parametrów geometrycznych wykorzystano obrazy cyfrowe ziaren. Następnie opracowano metodykę opartą na przekształcaniach filtracyjnych i metodach segmentacji, umożliwiających automatyczne wykonywanie pomiarów. Prezentowane rozwiązania wymagają zwykle posiadania specjalistycznego oprogramowania lub przygotowania własnego w oparciu

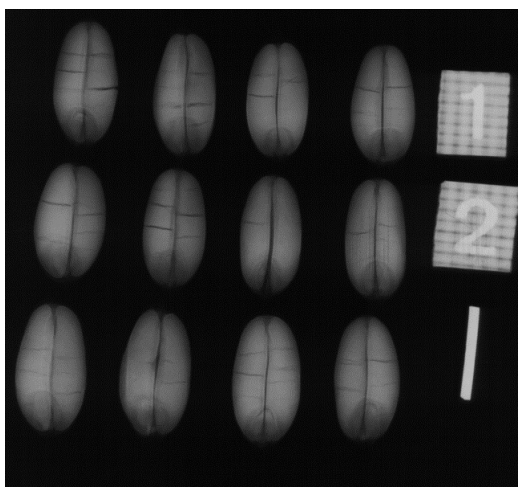
o istniejące biblioteki zawierające funkcje do wykonywania podstawowych operacji na obrazach cyfrowych [15-18]. Z tego względu w ramach badań opracowany został autorski program do wyznaczania cech geometrycznych obiektów znajdujących się na obrazach cyfrowych.

2. Materiał badawczy

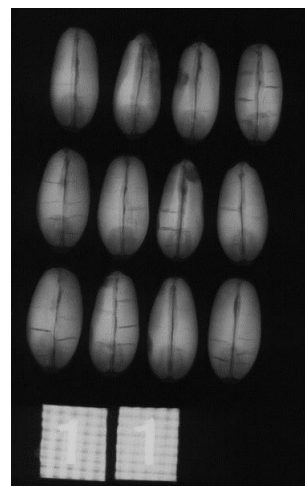
Materiał badań stanowiły obrazy rentgenowskie ziaren trzech odmian pszenicy: kama, kanadyjskiej i rosa, wybranych losowo ze zbioru kombajnowego. Do uzyskania zdjęć rentgenowskich został użyty aparat rentgenowski ELEKTRONIKA Model 25. Każdy obraz zawierał 12 ziaren. W celu wykonania pomiarów cech geometrycznych obrazy rentgenowskie zostały zeskanowane skanerem Epson Perfection V700 o rozdzielczości 600 dpi i zapisane w plikach graficznych o 24-bitowej głębi bitowej. Ich wielkość nie przekraczała 300 KB. Przykładowe obrazy ziaren poszczególnych odmian pszenicy, w skali 2:5 w stosunku do obrazów oryginalnych, zostały przedstawione na Rysunkach 1 – 3.



Rysunek 1: Materiał badawczy – odmiana kama



Rysunek 2: Materiał badawczy – odmiana kanadyjska

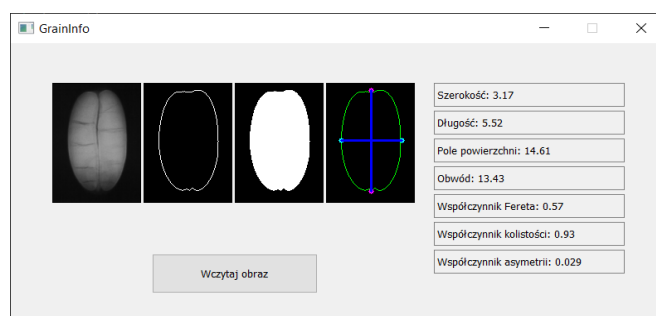


Rysunek 3: Materiał badawczy – odmiana rosa

W celu dokonania pomiarów każde ziarno zapisane zostało jako odrębny plik graficzny. Wyodrębniono 144 obrazy ziaren, po 48 dla każdej odmiany. Do wyznaczenia rzeczywistych wymiarów ziaren użyty został wzorek o długości 5 mm, odpowiadający 114 pikselom na obrazie cyfrowym. Na tej podstawie określono skalę, która wynosiła 0,04386.

3. Wyznaczania cech geometrycznych obiektów

Na potrzeby prowadzonych badań opracowany został autorski program GrainInfo, napisany w środowisku Microsoft Visual Studio. Analiza obrazu została wykonywana z wykorzystaniem biblioteki OpenCV, natomiast interfejs graficzny użytkownika i operacje matematyczne są wykonywane za pomocą biblioteki Qt [19-22]. Program umożliwia wizualizację struktury wewnętrznej ziaren i wyznaczanie podstawowych parametrów geometrycznych obiektów. Główne okno programu przedstawione zostało na Rysunku 4.



Rysunek 4: Główne okno programu GrainInfo

Program GrainInfo umożliwia automatyczny pomiar podstawowych parametrów charakteryzujących kształt i wielkość ziaren. Jednostką pomiaru są milimetry. Wykonane pomiary obejmują:

- pole ziarna, traktowane jako pole rzutu ziarna,
- obwód ziarna, traktowany jako obwód rzutu ziarna,
- szerokość ziarna,
- długość ziarna,

- współczynnik Fereta,
- współczynnik kistości,
- współczynnik asymetrii.

Współczynnik Fereta jest to iloraz maksymalnej średnicy obiektu w poziomie do maksymalnej średnicy w pionie. Współczynnik kistości obliczany jest według zależności:

$$V_c = \frac{4\pi A}{D^2} \quad (1)$$

gdzie A oznacza pole powierzchni, natomiast D obwód obiektu. Współczynnik asymetrii zdefiniowany został jako iloraz wartości bezwzględnej różnicy pola lewej i prawej części rzutu ziarna do pola rzutu całego ziarna.

4. Analiza danych

Podstawowe statystyki opisowe badanych parametrów przedstawia Tabela 1. Jednostką pomiaru są milimetry.

Tabela 1: Podstawowe statystyki opisowe badanych parametrów

Odmiana kama ($n = 48$)					
Zmienna	Średnia	Mediana	Min	Max	Odch.std
szerokość	3,145	3,180	2,807	3,465	0,168
długość	5,596	5,614	5,044	6,228	0,236
pole	14,189	14,339	11,321	16,784	1,251
obwód	13,271	13,355	11,754	14,518	0,574
wsp. Fereta	0,562	0,560	0,520	0,610	0,023
wsp. kolist.	0,928	0,930	0,870	0,970	0,024
wsp. asym.	0,027	0,022	0,002	0,080	0,022
Odmiana kanadyjska ($n = 48$)					
Zmienna	Średnia	Mediana	Min	Max	Odch.std
szerokość	2,727	2,719	2,500	3,026	0,118
długość	5,312	5,307	4,825	5,789	0,191
pole	11,434	11,358	9,649	13,560	0,738
obwód	12,368	12,281	11,228	13,684	0,513
wsp. Fereta	0,514	0,510	0,460	0,560	0,024
wsp. kolist.	0,851	0,855	0,680	0,920	0,042
wsp. asym.	0,036	0,035	0,001	0,105	0,024
Odmiana rosa ($n = 48$)					
Zmienna	Średnia	Mediana	Min	Max	Odch.std
szerokość	3,581	3,575	2,982	3,991	0,236
długość	6,310	6,294	5,658	6,974	0,300
pole	18,173	18,242	13,294	21,993	1,890
obwód	15,916	15,987	13,246	17,807	0,882
wsp. Fereta	0,568	0,560	0,510	0,630	0,032
wsp. kolist.	0,833	0,850	0,600	0,950	0,080
wsp. asym.	0,020	0,018	0,001	0,053	0,013

Największymi wymiarami charakteryzowała się odmiana rosa, natomiast dla odmiany kanadyjskiej otrzymano najmniejsze wartości szerokości, długości, pola i obwodu ziaren. Współczynniki Fereta, kistości i asymetrii były mniej zróżnicowane. Współczynnik Fereta był zbliżony dla odmian kama i rosa, dla odmiany kanadyjskiej uzyskał wartość najmniejszą. Największy współczynnik kistości uzyskała odmiana kama, dla pozostałych dwóch odmian współczynnik ten był zbliżony.

Wnioskowanie statystyczne przeprowadzono na poziomie istotności $\alpha = 0,05$. Do weryfikacji hipotezy

o zgodności rozkładu z rozkładem normalnym wykorzystany został test istotności Shapiro-Wilka. W każdej grupie rozkład badanych parametrów geometrycznych nie różnił się istotnie od rozkładu normalnego (otrzymano $p > 0,1$). Z tego względu do porównania wartości średnich przedstawionych parametrów zastosowano parametryczne testy istotności. Następnie zbadano możliwość wykorzystania analizy skupień do wydzielenia jednorodnych grup ziaren pod względem ich parametrów geometrycznych. Jakość klasyfikacji zbadano naiwną metodą Bayesa, metodą K -najbliższych sąsiadów oraz metodą wektorów wspierających.

4.1. Analiza wariancji

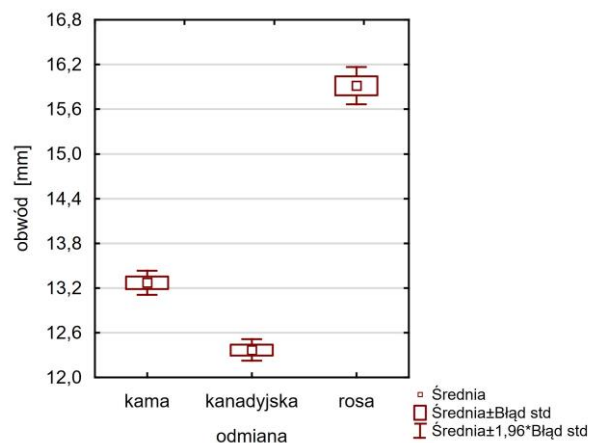
Hipotezę o równości wartości średnich badanych parametrów geometrycznych trzech odmian pszenicy zweryfikowano za pomocą metody analizy wariancji. Jednorodność wariancji sprawdzono testem Levene'a. Ponieważ wariancje badanych parametrów w badanych grupach różniły się istotnie ($p < 0,001$) zastosowano test F Welcha uwzględniający nierówne wariancje. W przypadku odrzucenia hipotezy o równości średnich danego parametru do stwierdzenia istotnych różnic pomiędzy grupami wykorzystano test Tukeya.

Wartość testu F Welcha analizy wariancji wraz z wartością istotności p przedstawia Tabela 2.

Tabela 2: Wyniki analizy wariancji

Zmienna	$F (2,141)$	p Welcha
szerokość	287,316	$p < 0,00001$
długość	187,926	$p < 0,00001$
pole	303,144	$p < 0,00001$
obwód	287,882	$p < 0,00001$
współczynnik Fereta	64,102	$p < 0,00001$
współczynnik kistości	79,998	$p < 0,00001$
współczynnik asymetrii	7,201	$p < 0,0006$

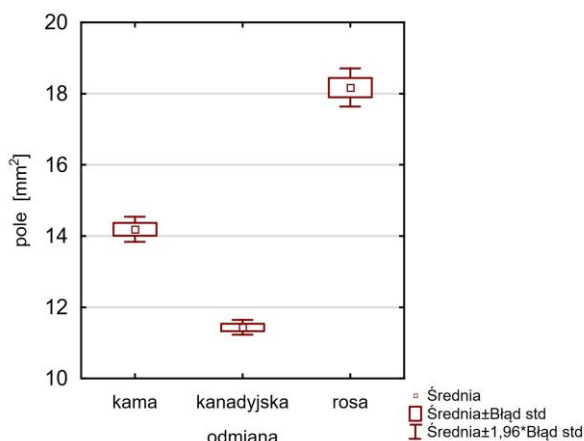
Na podstawie przeprowadzonego testu F Welcha dla każdego parametru stwierdzono statystycznie istotne różnice pomiędzy wartościami średnimi w badanych grupach odmian pszenicy: kama, kanadyjska i rosa. Wyniki testu Tukeya potwierdziły istotne statystycznie różnice pomiędzy poszczególnymi grupami.



Rysunek 5: Obwód ziaren dla trzech odmian pszenicy: istotne różnice pomiędzy odmianami kama – kanadyjska ($p < 0,0001$), kama – rosa ($p < 0,0001$), kanadyjska – rosa ($p < 0,0001$)

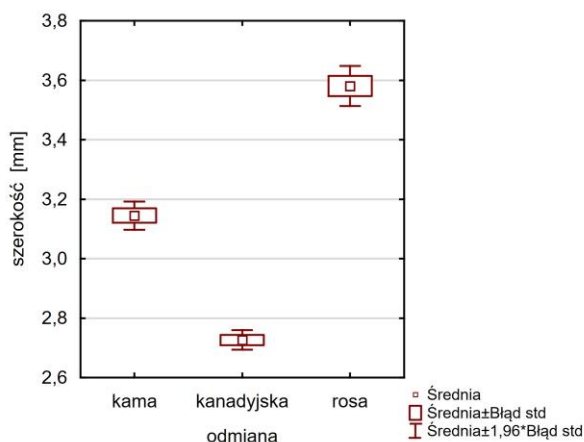
Skategoryzowane wykresy typu ramka-wąsy ilustrujące różnice pomiędzy wartościami średnimi badanych parametrów przedstawione są na Rysunkach 5 – 11.

Wszystkie trzy odmiany różnią się istotnie pod względem obwodu ziaren. Najmniejsze wartości charakteryzują odmianę kanadyjską, największe – odmianę rosa. Wartości średnie i odchylenia standardowe odmian kama i kanadyjskiej wynoszą odpowiednio $13,271 \pm 0,574$ mm i $12,368 \pm 0,513$ mm, natomiast dla odmiany rosa uzyskano wartości $15,916 \pm 0,882$ mm. Odmiana rosa bardziej różni się wymiarami od pozostałych dwóch odmian.



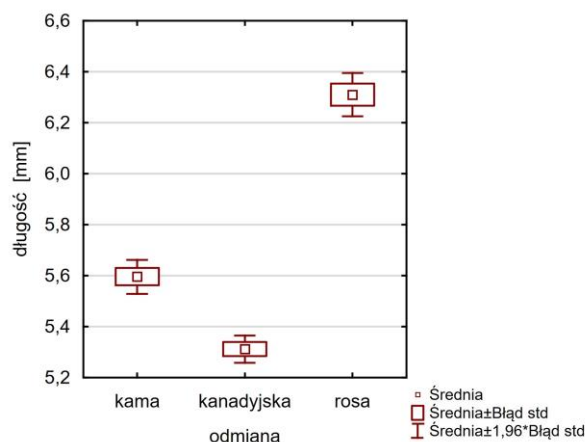
Rysunek 6: Pole ziaren dla trzech odmian pszenicy: istotne różnice pomiędzy odmianami kama – kanadyjska ($p < 0,0001$), kama – rosa ($p < 0,0001$), kanadyjska – rosa ($p < 0,0001$)

Wszystkie trzy odmiany odróżnia istotnie pole ziaren. Najmniejsze wartości charakteryzują odmianę kanadyjską, największe – odmianę rosa. Podobnie jak w przypadku obwodu, odmiana rosa różni się w większym stopniu. Wartości średnie i odchylenia standardowe dla odmian kama i kanadyjskiej wynoszą $14,189 \pm 1,251$ mm i $11,434 \pm 0,738$ mm, natomiast dla odmiany rosa uzyskano wynik $18,173 \pm 1,890$ mm.



Rysunek 7: Szerokość ziaren dla trzech odmian pszenicy: istotne różnice pomiędzy odmianami kama – kanadyjska ($p < 0,0001$), kama – rosa ($p < 0,0001$), kanadyjska – rosa ($p < 0,0001$)

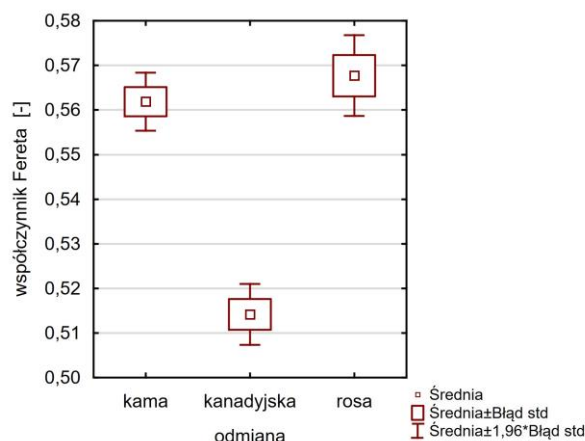
Wszystkie trzy odmiany różnią się istotnie pod względem szerokości ziaren. Najmniejsze wartości charakteryzują odmianę kanadyjską ($2,727 \pm 0,118$), nieco większe odmianę kama ($3,145 \pm 0,168$), a największe – odmianę rosa ($3,581 \pm 0,236$).



Rysunek 8: Długość ziaren dla trzech odmian pszenicy: istotne różnice pomiędzy odmianami kama – kanadyjska ($p < 0,0001$), kama – rosa ($p < 0,0001$), kanadyjska – rosa ($p < 0,0001$)

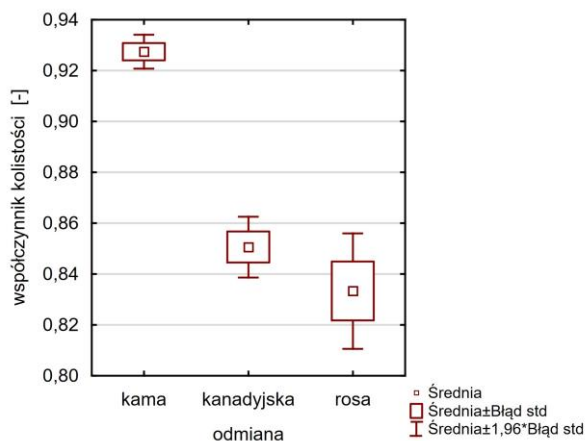
Wszystkie trzy odmiany różnią się istotnie pod względem długości ziaren. Najmniejsze wartości charakteryzują odmianę kanadyjską ($5,312 \pm 0,191$), nieco większe odmianę kama ($5,312 \pm 0,191$), a największe – odmianę rosa ($6,310 \pm 0,300$).

Przeprowadzone analizy wykazały, że ziarna badanych odmian pszenicy: kama, kanadyjskiej i rosa różnią się istotnie pod względem wielkości. Stwierdzono istotne różnice pomiędzy wartościami średnimi obwodu, pola, szerokości i długości ziaren. Największymi wymiarami charakteryzuje się odmiana rosa, najmniejszymi – odmiana kanadyjska.



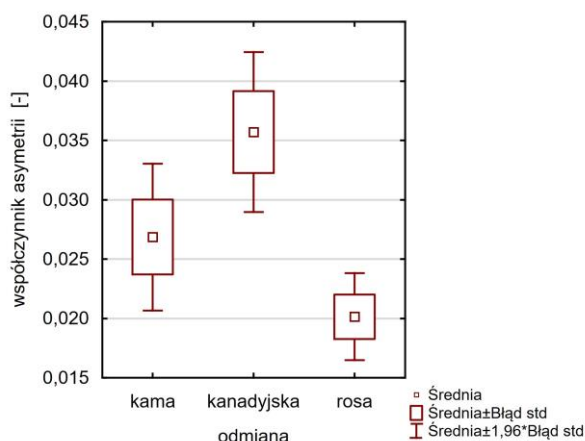
Rysunek 9: Współczynnik Fereta dla trzech odmian pszenicy: istotne różnice pomiędzy odmianami kama – kanadyjska ($p < 0,0001$), kanadyjska – rosa ($p < 0,0001$)

Współczynnik Fereta dla odmiany kanadyjskiej jest istotnie niższy od współczynników Fereta odmian kama i rosa.



Rysunek 10: Współczynnik kolistości dla trzech odmian pszenicy: istotne różnice pomiędzy odmianami kama – kanadyjska ($p < 0,0001$), kama – rosa ($p < 0,0001$)

Współczynnik kolistości dla odmiany kama jest istotnie wyższy od współczynników kolistości odmian kanadyjskiej i rosa.



Rysunek 11: Współczynnik asymetrii dla trzech odmian pszenicy: istotne różnice pomiędzy odmianami kanadyjska – rosa ($p < 0,0005$)

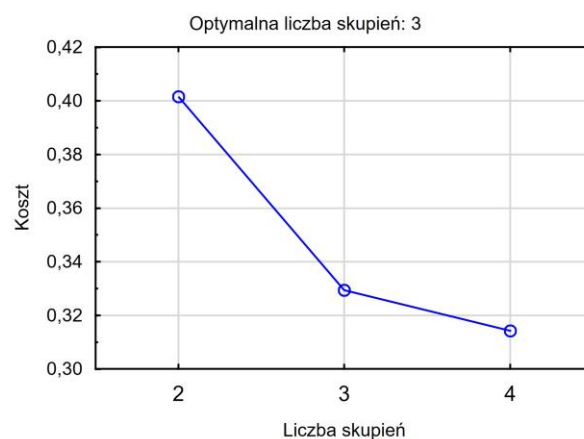
Współczynnik asymetrii dla odmiany kanadyjskiej jest istotnie wyższy od współczynnika asymetrii odmiany rosa.

Współczynnik Fereta różnicuje istotnie odmianę kanadyjską od odmian kama i rosa, współczynnik asymetrii odmianę rosa od kama i kanadyjskiej, a współczynnik kolistości odmianę kama od kanadyjskiej i rosa. Współczynnik kolistości osiąga największe wartości dla ziaren odmiany kama, co oznacza, że ich kształt w większym stopniu jest zbliżony do koła niż pozostałych dwóch odmian.

4.2. Analiza skupień

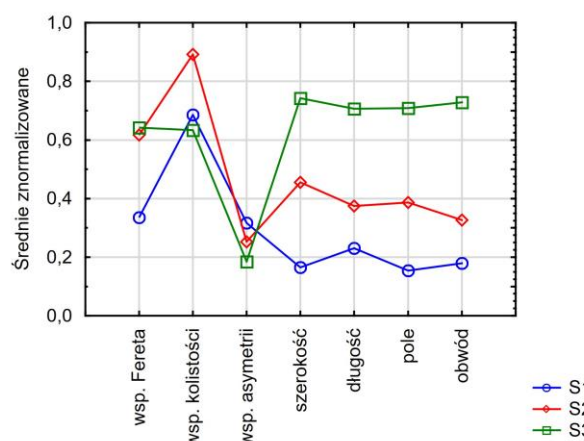
W celu zbadania, czy na podstawie podstawowych cech geometrycznych ziaren pszenicy możliwy jest ich podział na poszczególne odmiany, podjęto próbę wyodrębnienia jednorodnych grup za pomocą uogólnionej analizy skupień metodą k -średnich. Wykorzystano 144 elementy (ziarna) opisane za pomocą siedmiu zmien-

nych ilościowych (szerokość, długość, pole, obwód, współczynnik Fereta, współczynnik kolistości i współczynnik asymetrii). Jako miarę odległości wybrano odległość euklidesową, początkowe centra skupień zostały wybrane tak, by zmaksymalizować odległości skupień. Liczba skupień wyznaczana jest automatycznie, bez oddzielnej ingerencji w analizę. Do jej oceny wykorzystany został sprawdzian krzyżowy, w którym liczba grup została ustalana za pomocą doboru najlepszego podziału danych na segmenty na podstawie minimalizacji odległości elementów skupienia od jego środka. Automatycznie została wybrana poprawna liczba skupień równa trzy. Wyniki sprawdzianu krzyżowego ilustruje wykres osypiska przedstawiony na Rysunku 12.



Rysunek 12: Uogólniona analiza skupień: wykres osypiska

Analiza wykresu potwierdza poprawę precyzji podziału na segmenty w przypadku zwiększenia ich liczby z dwóch do trzech. Kolejne zwiększanie liczby segmentów nie przynosi już tak znacznej poprawy. Na Rysunku 13 przedstawiono znormalizowane średnie zmien-nych ilościowych dla skupień.



Rysunek 13: Wykres znormalizowanych wartości średnich parametrów geometrycznych ziaren dla trzech skupień wyznaczonych metodą k -średnich

Liczba ziaren każdego skupienia oraz ich przynależność do poszczególnych odmian pszenicy zawiera Tabela 3. Skupienie pierwsze zawiera 47 ziaren odmiany kanadyjskiej, 1 ziarno odmiany rosa i 7 ziaren odmiany kama. Skupienie drugie jest prawie jednorodne pod względem odmiany: zawiera 45 ziaren odmiany rosa i 1 ziarno odmiany kama. Skupienie trzecie zawiera 40 ziaren odmiany kama, 1 ziarno odmiany kanadyjskiej i 2 ziarna odmiany rosa. Wymiary ziaren różnicują wyznaczone skupienia. Skupienie 1 zawiera ziarna o najmniejszych wymiarach, skupienie 2 zawiera ziarna o największych wymiarach. Otrzymany podział na grupy potwierdzają również wyniki dla współczynników Fereta, asymetrii i kolistości.

Tabela 3: Liczba ziaren należących do poszczególnych skupień z podziałem na odmiany

Skupienie	Odmiana			Razem
	kanadyjska	rosa	kama	
Skupienie 1	47	1	7	55
Skupienie 2	0	45	1	46
Skupienie 3	1	2	40	43
Razem	48	48	48	144

Poprawnie zaklasyfikowano 132 ziarna, co stanowi niemal 92% ich całkowitej liczby. Liczba niepoprawnie zaklasyfikowanych ziaren wynosi 12. Odmiana rosa w większym stopniu odróżnia się od odmiany kanadyjskiej i kama.

Przeprowadzone analizy potwierdziły możliwość dokonania podziału zbioru ziaren na trzy grupy odpowiadające poszczególnym odmianom pszenicy bez odgórnej ingerencji w analizę.

4.3. Metody klasyfikacji

W celu zbadania możliwości identyfikacji ziaren na podstawie ich cech geometrycznych wykorzystano trzy klasyczne metody klasyfikacji:

- naiwną metodę Bayesa,
- metodę K -najbliższych sąsiadów
- metodę wektorów nośnych.

W pierwszym kroku badań dokonano losowego podziału zbioru danych na zbiór uczący i zbiór testowy w proporcjach:

- 90% : 10%,
- 80% : 20%,
- 70% : 30%.

W celu porównania metod klasyfikacji wykorzystano taki sam podział na zbiór uczący i testowy. Liczebności zbiorów: uczącego i testowego z podziałem na odmiany dla każdego podziału przedstawia Tabela 4.

Tabela 4: Liczebności zbiorów uczących i testowych z uwzględnieniem podziału na trzy odmiany

Podział	Liczba ziaren		Razem
	Zbiór uczący	Zbiór testowy	
90% : 10%	129 (3 x 43)	15 (3 x 5)	144 (3 x 48)
80% : 20%	114 (3 x 38)	30 (3 x 10)	144 (3 x 48)
70% : 30%	102 (3 x 34)	42 (3 x 14)	144 (3 x 48)

Każda odmiana zawierała 48 ziaren. Po dokonaniu podziału liczba ziaren każdej odmiany w zbiorze uczącym była taka sama. Odpowiednio zbiory testowe zawierały po 5, 10 i 14 ziaren każdej odmiany. W pierwszym przypadku liczebność zbioru testowego wynosiła 15, w drugim – 30, w trzecim 42.

Tabele 5-7 zawierają wyniki klasyfikacji uzyskane naiwną metodą Bayesa dla wyodrębnionych zbiorów testowych o liczebności.

Tabela 5: Wyniki klasyfikacji: naiwna metoda Bayesa, zbiór testowy: 10%

Odmiana	Liczba ziaren		Razem
	Trafne	Błędne	
kama	5 (71,4%)	2 (28,6%)	7
kanadyjska	3 (100%)	0 (0%)	3
rosa	5 (100%)	0 (0%)	5
Razem	13 (87%)	2 (13%)	15

W wyniku klasyfikacji naiwną metodą Bayesa wykonaną dla zbioru testowego stanowiącego 10% wszystkich ziaren 2 ziarna odmiany kanadyjskiej zostały błędnie zaklasyfikowane jako ziarna odmiany kama. Pozostałych 13 ziaren zaklasyfikowano poprawnie, co stanowi 87% liczby ziaren zbioru testowego.

Tabela 6: Wyniki klasyfikacji: naiwna metoda Bayesa, zbiór testowy: 20%

Odmiana	Liczba ziaren		Razem
	Trafne	Błędne	
kama	10 (83,3%)	2 (16,7%)	12
kanadyjska	8 (100%)	0 (0%)	8
rosa	10 (100%)	0 (0%)	10
Razem	28 (93%)	2 (7%)	30

Dla zbioru testowego stanowiącego 20% wszystkich ziaren 2 ziarna odmiany kanadyjskiej zostały błędnie zaklasyfikowane jako ziarna odmiany kama. Poprawnie zaklasyfikowano 28 ziaren, co stanowi 93% liczby ziaren zbioru testowego. Liczba niepoprawnie zaklasyfikowanych ziaren wynosi 2, jest to 7% liczby ziaren.

Tabela 7: Wyniki klasyfikacji: naiwna metoda Bayesa, zbiór testowy: 30%

Odmiana	Liczba ziaren		Razem
	Trafne	Błędne	
kama	11 (84,6%)	2 (15,4%)	13
kanadyjska	12 (100%)	0 (0%)	12
rosa	14 (82,4%)	3 (17,6%)	17
Razem	37 (88%)	5 (12%)	42

Dla zbioru testowego stanowiącego 30% wszystkich ziaren 2 ziarna odmiany kanadyjskiej zostały błędnie zaklasyfikowane jako ziarna odmiany kama oraz 3 ziarna odmiany kama zostały błędnie zaklasyfikowane jako ziarna odmiany rosa. Poprawnie zaklasyfikowano 37 ziaren, co stanowi 88% ziaren zbioru testowego zawierającego 42 ziarna. Liczba niepoprawnie zaklasyfikowanych ziaren wynosi 5.

Tabele 8-10 zawierają wyniki klasyfikacji uzyskane metodą K -najbliższych sąsiadów z parametrem $K = 1$

dla zbioru testowego o liczebności 15, 30 i 42 ziarna odpowiednio.

Tabela 8: Wyniki klasyfikacji: metoda *K*-najbliższych sąsiadów, zbiór testowy: 10%

Odmiana	Liczba ziaren		Razem
	Trafne	Błędne	
kama	5 (71,4%)	2 (28,6%)	7
kanadyjska	3 (100%)	0 (0%)	3
rosa	5 (100%)	0 (0%)	5
Razem	13 (87%)	2 (13%)	15

W wyniku klasyfikacji metodą *K*-najbliższych sąsiadów wykonaną dla zbioru testowego stanowiącego 10% wszystkich ziaren 2 ziarna odmiany kanadyjskiej zostały błędnie zaklasyfikowane jako ziarna odmiany kama. Poprawnie zaklasyfikowano 13 ziaren, co stanowi 87% całkowitej liczby ziaren zbioru testowego. Liczba niepoprawnie zaklasyfikowanych ziaren wynosi 2.

Tabela 9: Wyniki klasyfikacji metoda *K*-najbliższych sąsiadów, zbiór testowy: 20%

Odmiana	Liczba ziaren		Razem
	Trafne	Błędne	
kama	10 (90,9%)	1 (9,1%)	11
kanadyjska	9 (100%)	0 (0%)	9
rosa	10 (100%)	0 (0%)	10
Razem	29 (93%)	1 (7%)	30

Dla drugiego zbioru testowego stanowiącego 20% wszystkich ziaren 1 ziarno odmiany kanadyjskiej zostało błędnie zaklasyfikowane jako ziarno odmiany kama. Poprawnie zaklasyfikowano 29 ziaren, co stanowi 93% ich liczby dla zbioru testowego. Liczba niepoprawnie zaklasyfikowanych ziaren wynosi 1 co stanowi 7% liczby ziaren.

Tabela 10: Wyniki klasyfikacji: *K*-metoda najbliższych sąsiadów, zbiór testowy: 30%

Odmiana	Liczba ziaren		Razem
	Trafne	Błędne	
kama	13 (92,9%)	1 (7,1%)	14
kanadyjska	13 (100%)	0 (0%)	13
rosa	14 (82,4%)	1 (17,6%)	15
Razem	40 (95,2%)	2 (4,8%)	42

Dla trzeciego zbioru testowego stanowiącego 30% wszystkich ziaren 1 ziarno odmiany kanadyjskiej zostało błędnie zaklasyfikowane jako ziarno odmiany kama oraz 1 ziarno odmiany kama zostało błędnie zaklasyfikowane jako ziarno odmiany rosa. Poprawnie zaklasyfikowano 40 ziaren, co stanowi 95,2% liczby ziaren zbioru testowego. Liczba niepoprawnie zaklasyfikowanych ziaren wynosi 2 co stanowi 4,8% liczby ziaren.

Tabele 11-13 zawierają wyniki klasyfikacji uzyskane metodą wektorów wspierających dla rozpatrywanych zbiorów testowych o liczebności 15, 30 i 42 ziarna odpowiednio.

W wyniku klasyfikacji metoda wektorów wspierających, wykonaną dla zbioru testowego stanowiącego 10% wszystkich ziaren 2 ziarna odmiany kanadyjskiej

zostały błędnie zaklasyfikowane jako ziarna odmiany kama.

Tabela 11: Wyniki klasyfikacji: metoda wektorów wspierających, zbiór testowy: 10%

Odmiana	Liczba ziaren		Razem
	Trafne	Błędne	
kama	5 (71,4%)	2 (28,6%)	7
kanadyjska	3 (100%)	0 (0%)	3
rosa	5 (100%)	0 (0%)	5
Razem	13 (87%)	2 (13%)	15

Poprawnie zaklasyfikowano 13 ziaren, co stanowi 87% ich całkowitej liczby. Liczba niepoprawnie zaklasyfikowanych ziaren wynosi 2. Wyniki były identyczne jak w przypadku metody *K*-najbliższych sąsiadów.

Tabela 12: Wyniki klasyfikacji metoda wektorów wspierających, zbiór testowy: 20%

Odmiana	Liczba ziaren		Razem
	Trafne	Błędne	
kama	10 (83,3%)	2 (16,7%)	12
kanadyjska	8 (100%)	0 (0%)	8
rosa	10 (100%)	0 (0%)	10
Razem	28 (93%)	2 (7%)	30

Dla zbioru testowego stanowiącego 20% wszystkich ziaren 2 ziarna odmiany kanadyjskiej zostały błędnie zaklasyfikowane jako ziarna odmiany kama. Poprawnie zaklasyfikowano 28 ziaren. Liczba niepoprawnie zaklasyfikowanych ziaren wynosi 2. Wyniki były porównywalne z wynikami otrzymanymi naiwną metodą Bayesa i gorsze niż w przypadku metody *K*-najbliższych sąsiadów.

Tabela 13: Wyniki klasyfikacji: metoda wektorów wspierających, zbiór testowy: 30%

Odmiana	Liczba ziaren		Razem
	Trafne	Błędne	
kama	13 (86,7%)	2 (13,3%)	15
kanadyjska	12 (100%)	0 (0%)	12
rosa	14 (93,3%)	1 (6,7%)	15
Razem	39 (93%)	3 (7%)	42

Dla zbioru testowego stanowiącego 30% wszystkich ziaren 2 ziarna odmiany kanadyjskiej zostały błędnie zaklasyfikowane jako ziarna odmiany kama. Ponadto jedno ziarno odmiany kama zostało błędnie zaklasyfikowane jako ziarno odmiany rosa. Poprawnie zaklasyfikowano 39 ziaren, co stanowi 93% liczby ziaren zbioru testowego. Liczba niepoprawnie zaklasyfikowanych ziaren wynosi 2 co stanowi 7%. Wyniki były gorsze niż w przypadku metody *K*-najbliższych sąsiadów.

Na podstawie badań wykonanych naiwną metodą Bayesa oraz metodami *K*-najbliższych sąsiadów i wektorów wspierających można stwierdzić, iż wśród metod klasyfikacji najgorsze wyniki uzyskano dla naiwnej metody Bayesa. Metoda *K*-najbliższych sąsiadów metoda okazała się najlepszym klasyfikatorem. Dla zbioru testowego stanowiącego 10% całkowitej liczby obiektów wyniki były porównywalne, dla zbioru stanowiącego 20% ogólnej liczby ziaren najlepszą okazała się

metoda K -najbliższych sąsiadów. W trzecim teście, przeprowadzonym na zbiorze testowym o liczności 42 ziaren, stanowiącym 30% całkowitej liczby obiektów metodą Bayesa błędnie zaklasyfikowanych zostało 5 ziaren, metodą wektorów wspierających 3 ziarna, natomiast metodą K -najbliższych sąsiadów błędnie zaklasyfikowano tylko 2 ziarna.

5. Podsumowanie i wnioski

W niniejszej pracy przeprowadzone badania dotyczące wykorzystania podstawowych cech geometrycznych do klasyfikacji obiektów. Zastosowana w tym celu technika rentgenowska okazała się niedrogą i nieinwazyjną metodą obrazowania struktury wewnętrznej materiału badawczego. Badania przeprowadzone na podstawie obrazów ziaren pszenicy potwierdziły możliwość rozróżnienia odmian pszenicy na podstawie cech geometrycznych ziaren. W wyniku przeprowadzenia analizy skupień metodą k -średnich poprawnie zaklasyfikowanych zostało 132 ziarna, co stanowi 92% ich całkowitej liczby. W badaniach dotyczących podstawowych metod klasyfikacji procent poprawnie zaklasyfikowanych ziaren wynosił 93%. Najlepsze wyniki uzyskano metodą K -najbliższych sąsiadów.

Literatura

- [1] R. C. Gonzalez, R. E. Woods, Digital Image Processing, Prentice-Hall Inc., New Jersey, 2002.
- [2] R. Tadeusiewicz, Komputerowa analiza i przetwarzanie obrazów, Wydawnictwo Fundacji Postępu Telekomunikacji, Kraków 1997.
- [3] B. Mirkin, Clustering: A Data Recovery Approach, Chapman and Hall/CRC, 2012.
- [4] D. F. Morrison, Multivariate Statistical Methods, Brooks/Cole Thomson Learning, Belmont, California, 2005.
- [5] M. Romaniuk, O. Hryniewicz, Interval based, nonparametric approach for resampling of fuzzy numbers. 2019, Soft Computing, 23 (14), 5883–5903.
- [6] J. Koronacki, J. Ćwik, Statystyczne systemy uczące się, WNT 2008.
- [7] M. Krzyśko, W. Wołyński, T. Górecki, M. Skorzybut, Systemy uczące się. WNT, Warszawa, 2008.
- [8] H. Czachor, M. Charytanowicz, S. Gonet, J. Niewczas, G. Józefaciuk, L. Lichner, Impact of long term mineral and organic fertilization on water stability, wettability and porosity of aggregates of two silt loamy soils. 2015, European Journal of Soil Science, 66 (3), 577–588.
- [9] M. Klatka, E. Grywalska, M. Partyka, M. Charytanowicz, E. Kiszczak-Bochyńska, J. Roliński: Th17 and Treg cells in adolescents with Graves' disease. Impact of treatment with methimazole on these cell subsets, 2014, Autoimmunity, 47 (3), 201–211.
- [10] P. Kulczycki, P. A. Kowalski, Bayes Classification for Nonstationary Patterns, International Journal of Computational Methods, 2015, 12, ID 1550008.
- [11] T. Guz, Z. Kobus, E. Kusińska, R. Nadulski, Morphometric features of rye caryopses stored in a silo, Inżynieria Rolnicza Agricultural Engineering, 2012, 1 (4), 71–79.
- [12] P. Zapotoczny, Discrimination of wheat grain varieties using image analysis and neural networks. Part I. Single kernel texture. Journal of Cereal Science, 2012, 54, 60–68.
- [13] M. Charytanowicz, J. Niewczas, P. Kulczycki, P. A. Kowalski, S. Łukasik, Discrimination of Wheat Grain Varieties Using X-ray Images. Information Technologies in Biomedicine, E. Pietka, P. Badura, J. Kawa, W. Więclawek (red.), Advances in Intelligent Systems and Soft Computing, Springer, 2016, 471, 39–50.
- [14] M. Charytanowicz, P. Kulczycki, P. A. Kowalski, S. Łukasik, R. Czabak-Garbacz, An Evaluation of Utilizing Geometric Features for Wheat Grain Classification using X-ray Images. Computers and Electronics in Agriculture, 2018, 144, 260–268.
- [15] J. Niewczas, A. Strumiłło, Szczypiński, P. Makowski, W. Woźniak, Computer system for analysis of x-ray image of wheat grains, International Agrophysics 1999.
- [16] G. D. Jasmin, Shape based Object Classification Rusing Knowledge Vector Code International Journal of Innovative Research in Computer and Communication Engineering, 2017, 5 (7), 13440.
- [17] D. Zhang, G. Lu, Review of shape representation and description techniques. Pattern Recognition, 2004, 37, 1–19.
- [18] M. Zhu, T. J. Hastie, Feature Extraction for Nonparametric Discriminant Analysis. Journal of Computational and Graphical Statistics, 2003, 12(1), 101–120.
- [19] Visual Studio 2017 – Now Ready for Your Windows Application Development Needs, <https://blogs.windows.com/windowsdeveloper/2017/03/07/visual-studio-2017-now-ready-windows-application-development-needs/>, [22.04.2020].
- [20] Qt Software, <http://doc.qt.io/>, [01.04.2020].
- [21] C++ Programming Language, <https://www.techopedia.com/definition/26184/c-programming-language>, [22.04.2020].
- [22] About OpenCV, <https://opencv.org/about/>, [22.04.2020].

Semi-supervised learning with the clustering and Decision Trees classifier for the task of cognitive workload study

Częściowo nadzorowane uczenie z zastosowaniem klasteryzacji oraz klasyfikatora Drzew Decyzyjnych w przypadku badania obciążenia poznawczego

Martyna Wawrzyk*

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

The paper is focused on application of the clustering algorithm and Decision Trees classifier (DTs) as a semi-supervised method for the task of cognitive workload level classification. The analyzed data were collected during examination of Digit Symbol Substitution Test (DSST) with use of eye-tracker device. 26 participants took part in examination as volunteers. There were conducted three parts of DSST test with different levels of difficulty. As a results three versions were obtained of data: low, middle and high level of cognitive workload. The case study covered clustering of collected data by using k-means algorithm to detect three clusters or more. The obtained clusters were evaluated by three internal indices to measure the quality of clustering. The David-Boudin index detected the best results in case of four clusters. Based on this information it is possible to formulate the hypothesis of the existence of four clusters. The obtained clusters were adopted as classes in supervised learning and have been subjected to classification. The DTs was applied in classification. There were obtained the 0.85 mean accuracy for three-class classification and 0.73 mean accuracy for four-class classification.

Keywords: clustering; semi-supervised learning; eye-tracker

Streszczenie

Celem artykułu było zastosowanie klasteryzacji wraz z klasyfikatorem Drzew Decyzyjnych jako częściowo nadzorowanej metody klasyfikacji poziomu obciążenia poznawczego. Dane przeznaczone do analizy zostały zebrane podczas badania DSST (z ang. Digit Symbol Substitution Test) z użyciem urządzenia eye-tracker. 26 wolontariuszów wzięło udział w badaniu. Zostały przeprowadzone trzy części testu DSST o różnych poziomach trudności. W wyniku tego, otrzymano trzy wersje danych: z niskim, średnim i wysokim poziomem obciążenia poznawczego. Do analizy danych został użyty algorytm klasteryzacji *k-means* do wyznaczenia trzech lub większej liczby klastrów. Uzyskane klastry zostały poddane ocenie przy użyciu trzech wewnętrznych indeksów w celu zmierzenia jakości klasteryzacji. Indeks David-Boudin'a wykazał najlepsze rezultaty w przypadku istnienia czterech klastrów. Na podstawie tej informacji można sformułować hipotezę, iż dane są podzielone na 4 klastry, co oznaczałoby istnienie dodatkowego poziomu poznawczego. Uzyskane klastry zostały zaadoptowane jako klasy w uczeniu pod nadzorem. Do klasyfikacji danych został użyty klasyfikator Drzew Decyzyjnych. Otrzymano średnią dokładność równą 0.85 w przypadku 3-klasowej klasyfikacji oraz 0.73 średnią dokładność dla 4-klasowej klasyfikacji.

Słowa kluczowe: klasteryzacja; uczenie częściowo nadzorowane; eye-tracker

*Corresponding author

Email address: martyna.wawrzyk@pollub.edu.pl (M.Wawrzyk)

©Published under Creative Common License (CC BY-SA v4.0)

1. Introduction

Clustering of unlabeled data is a commonly used method in unsupervised learning. It allows to detect the clusters of data with no labels. There were used many types of clustering methods in analysis of oculography data. In research [1], [2], [3] there were conducted the bidimensional clustering to detect the fixations. *K-means* algorithm can be used to identify the microsaccades [4] and areas of visual interest [5]. The Gaussian Mixture Models were applied for clustering gaze locations in dynamic scenes [6]. The hierarchical clustering was used in analysis of eye movements [7].

There is possibility to combine two methods from unsupervised learning and supervised learning [8].

In effect there is obtained the semi-supervised learning method. In research [9] there were performed the spectral clustering and semi-supervised Gaussian process regression in order to analyze the tracking of gaze.

But the one of the main problems of clustering is an evaluation of a model. There are two types of indices, which helps to assess the quality of clustering: external indices and internal indices [10]. Internal indices are used to measure the quality of a clustering without external information [11]. External indices evaluates the clustering with external information [12].

In case study of this paper there were used the digitalized version of the Digit Symbol Substitution Test (DSST) [16]. The DSST is a commonly used test in clinical neuropsychology in order to measure cogni-

tive dysfunction. The test allows to measure the speed of processing data, memory and others cognitive functions of a patient. For this reason is widely used in neuropsychology [17].

The aim of this paper is to apply a semi-supervised learning with K-means algorithm clustering with Decision Trees classifier. The assumption is the existence of three clusters, taking into account the fact of using a three-level cognitive workload study. The application of this algorithm is designed to check out if there are more than 3 clusters. The obtained clusters were evaluated by using three indices. Finally the clusters were adopted as a classes in supervised learning in order to classify the features of clusters.

2. The research procedure

In the study there were used the computerized version of DSST test. The examined person has to match symbols to the numbers according to a template located on the bottom of the screen. In order to match the number to the symbol the user needs to click the proper symbol on the template. The currently active letter is marked by graphical frame (Fig.1). After clicking the frame is moving to the next letter. The time to match the symbols is specified and letters are generated randomly.

The interface of application is presented in Fig.1. The application was developed in Java and is operated using a computer mouse. The case study was divided into three stages. There were performed DSST test three times with the following settings:

- 4 different symbols to assign; the test lasted 90 s.
- 9 different symbols to assign; the test lasted 90 s.
- 9 different symbols to assign; the test lasted 180 s.

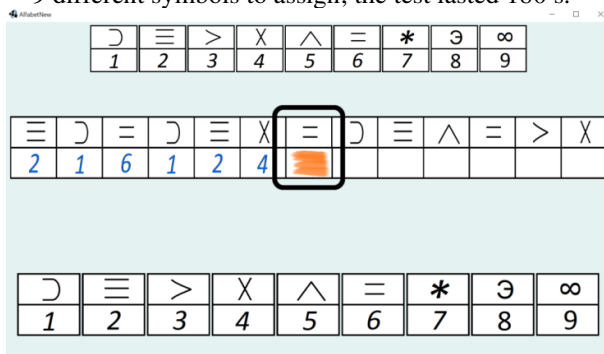


Figure 1: Procedure of data processing

2.1. Set up and equipment

An eye-tracker device and computer were used in the study. The experiment was performed in dedicated laboratory illuminated with standard fluorescent light. The data was recorded by Tobii Pro TX300 screen-based eye-tracker. The technology of Tobii Pro TX300 is based on video-oculography. It collects data using the dark pupil and corneal reflection method with the frequency of 300 Hz.

The Tobii Studio 3.2 was used to design the experiment. This software is dedicated for eye-tracker experiment and is compatible with eye-tracker device. The

monitor with the following parameters: 23'' TFT monitor at 60 Hz was applied to present the visual stimuli.

The experiment was conducted in sitting position with distance between participant and monitor in range from 50 to 80 cm. The same procedures was issued for each participant.

2.2. Experiment

The study was performed with 26 participants aged 20-24. The duration of the examination of single participant lasted 15 minutes. The study was divided into three parts. There were conducted the process of calibration in each stage by using the eye-tracker device. The calibration process was consisted of 9-point built-in procedure. After the calibration, the instruction was displayed on the monitor to inform about the procedure of assignment the symbols to the numbers. The study consisted of three parts of DSST test. Each part had a different number of symbols to assign and lasted for different period of time. Each part had a short initial trial to familiarize the participants with the task. Each participant had to finish three parts of DSST test.

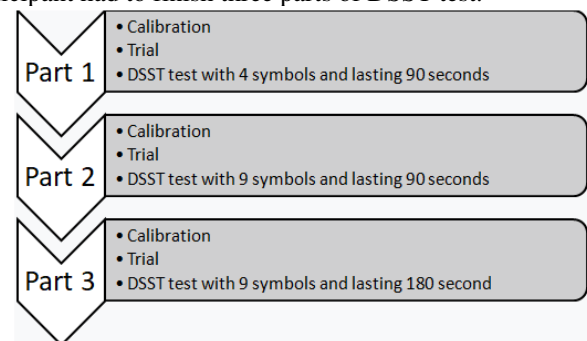


Figure 2: Procedure of the examination

2.3. Data set

There were obtained 156 files: 78 files generated from eye-tracker device and 78 files generated from the application. From each participant there were received 6 files: 3 from eye-tracker device and 3 from the application (the data from each part of test were saved in separate files).

3. Methods applied

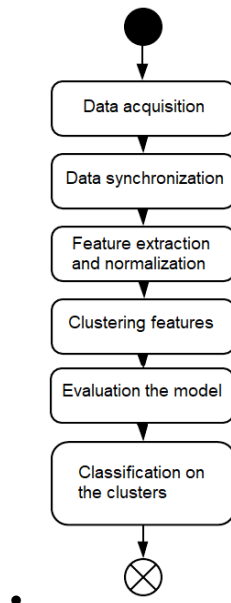
3.1. Data processing

The data processing was divided into six stages (Fig. 3). There were obtained two separate files from each part of experiment. Consequently after data acquisition there were performed data synchronization.

In sequence the features were extracted. The following metrics were selected as features for further data processing data:

- number of blinks, mean and duration of blinks;
- mean, standard deviation, maximal and minimal duration of saccades;
- mean, standard deviation, maximal and minimal duration of fixations;

- mean and standard deviation of left/right pupil diameter;
- number of responses;
- number of mistakes in responses;
- mean time of response;



• Figure 3: Procedure of data processing

3.2. Unsupervised learning

Clustering is the method performed in unsupervised learning, which is used to for extracted clusters of unlabeled data. The k-means algorithm were chosen for the clustering of extracted features. K-means clustering is based on defining number of centroids and assigning of each data point to the nearest cluster.

Also specific internal indices to evaluate the model were used. It is necessary to assess to quality of clustering. In this paper there were considered three type of internal indices. The Calinski-Harabasz index is defined as a ratio between the within-cluster dispersion and the between-cluster dispersion [13]. In the case of this index there is a need to make a line-plot (dependence of index value on the number of clusters). If the peak on the line-plot were observed, this point (number of clusters) should be chosen as the best clustering. The next index is Silhouette index, which is a coefficient between mean-intra cluster distance to mean nearest-cluster distance [14]. The value 1 gives the best results, the value -1 means incorrect clustering, the value close to 0 means clustering overlaps. The last index is a David-Boudin index, which is defined as the average similarity measure of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances [15]. The lowest values of index means the better clustering.

3.3. Supervised learning

The received clusters were adopted as classes in supervised learning. The Decision Trees (DTs) were conducted in classification of features. It was performed for k-

class classification. The model of DTs and clustering were implemented in Google Colab environment.

4. Results

4.1. Clustering

There were generated plots to visualize the distribution of clusters. The PCA algorithm were used to reduce data before the generated of plots. The Fig. 4 illustrates the arrangement of features assuming existence of three clusters, in turn Fig. 5 presents the arrangement in case of four clusters. The red points represent the centroids of the clusters. The Fig.5 presents that the points of cluster 4 are visible separated from other clusters.

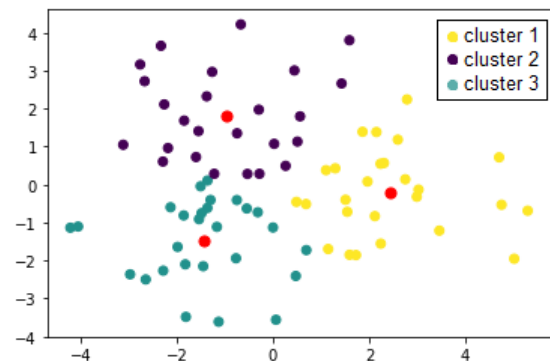


Figure 4: The data distribution into 3 clusters

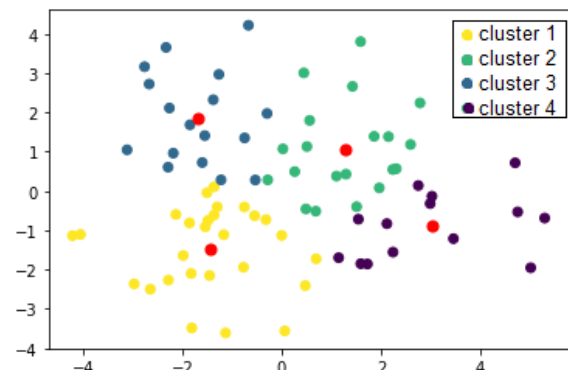


Figure 5: The data distribution into 4 clusters

The Table 1 contains the results of three chosen indices to evaluate the quality of clustering. The comparison was conducted between five clusters. Each metrics must be interpreted individually. In case of Calinski-Harabasz index it is worth to visualizing by plot and find the characteristic point. The Fig.6 presents the dependence of Calinski-Harabasz index value on the number of clusters. Base on the chart, it cannot be determined which number of clusters represents a real distribution of data. There were not observed the peak which means the best clustering. The silhouette index also does not specify which number of clusters gives the best results. The Davies-Boundin score with the lowest values indicating better clustering. The number of four clusters presents the lowest values in compared to others. Based on this information it is possible to formulate the hypothesis of the existence of four clusters.

Table 1. The values of internal indexes for k-clusters

Clustering metrics/ k-clusters	2	3	4	5
Calinski-Harabasz index	16.04	13.48	12.28	11.12
Davies-Boudin index	2.04	1.97	1.80	1.83
Silhouette index	0.16	0.14	0.14	0.13

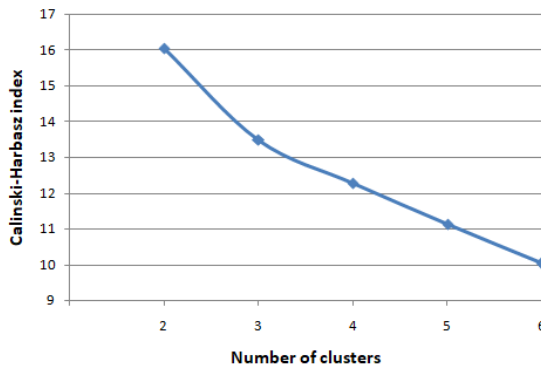


Figure 6: The line-plot of Calinski-Harabasz index

4.2. Classification results

The Table 2 presents the mean accuracy values of k-class classification. The best results were obtained for two-class and three-class classification. In case of four-class and five-class classification there were obtained good results.

Table 2. The values accuracy for k-class classification

K-class	2	3	4	5
Mean accuracy of classification	88.5%	85.75%	73.85%	75.75%

Table 3 and Table 4 present the confusion matrix of three-class and four-class classification. There were observed the correct classification the class 2 in case of three-class classification. The confusion matrix of four-class classification presents the correct classification of class 3 and class 4.

Table 3. Mean confusion matrix for three-class classification

	class 1	class 2	class 3
class 1	3	2	0
class 2	0	5	0
class 3	3	0	3

Table 4. Mean confusion matrix for four-class classification

	class 1	class 2	class 3	class 4
class 1	4	0	2	0
class 2	0	2	1	0
class 3	0	0	4	0
class 4	0	0	3	0

5. Discussion and conclusion

The aim of this paper was to applied a semi-supervised learning in case study of cognitive workload based on eye-tracker data. There were conducted study with the computerized version of Digit Symbol Substitution Test (DSST). The collected data contained eye-tracking features related to blinks, fixation, saccades and pupil diameter.

Firstly there were used a k-mean algorithm to detect clusters of analyzed data. The obtained data were evaluated by three clustering metric to assess the quality of clustering. The David-Boudin metric indicates better clustering in case of four clusters compared to others. Based on this it may be concluded that the data is divided into 4 clusters. Fig. 5 illustrates the distribution of data with four clusters. The four cluster is visible separately from others clusters. The cluster no 4 may represent the different level of cognitive workload.

The results of classification present very good mean accuracy in case of three-class classification and also four-class classification. Also there were extracted the results in case of two-class and five-class to make a comparison.

In summary, a hypothesis for the collected data being able to be divided into four has been suggested. But there cannot be made an assumption that the data has a distribution into four clusters. The clustering metrics cannot indicate which number of clusters gives the best results. These indices may inform which number of cluster is better than other.

References

- [1] T. Urruty, S. Lew, N. Ihadaddene and D. A. Simovici, Detecting eye fixations by projection clustering. *ACM Transaction on Multimedia Computing, Communications and Application*, 3 (4), 5:1–5:20, 2007
- [2] N. Flad, T. Fomina, H. H. Buelthoff and L. L. Chuang, Unsupervised Clustering of EOG as a Viable Substitute for Optical Eye Tracking. *Eye Tracking and Visualization*, Cham, 2017, 151–167
- [3] R. S. Hessels, D. C. Niehorster, C. Kemner and I. T. C. Hooge Noise-robust fixation detection in eye movement data: Identification by two-means clustering (I2MC). *Behaviour Research Methods*, 49 (5), 1802–1823, 2017
- [4] J. Otero-Millan, J. L. A. Castro, S. L. Macknik and S. Martinez-Conde Unsupervised clustering method to detect microsaccades. *Journal of Vision*, 14 (2), 18–18, 2014
- [5] A. Santella and D. DeCarlo Robust clustering of eye movement recordings for quantification of visual interest. *Proceedings of the 2004 symposium on Eye tracking research & applications*, San Antonio, Texas, 2004, 27–34
- [6] P. K. Mital, T. J. Smith, R. L. Hill and J. M. Henderson, Clustering of Gaze During Dynamic Scene Viewing is Predicted by Motion. *Cognitive Computation*, 3 (1), 5–24, 2011
- [7] Z. Kang and S. J. Landry An Eye Movement Analysis Algorithm for a Multielement Target Tracking Task:

- Maximum Transition-Based Agglomerative Hierarchical Clustering. *IEEE Transactions on Human-Machine Systems*, 45 (1), 13–24, 2015
- [8] M. Aamir and S. M. A. Zaidi Clustering based semi-supervised machine learning for DDoS attack classification. *Journal of King Saud University - Computer Information Sciences*, 2019
- [9] K. Liang, Y. Chahir, M. Molina, C. Tijus and F. Jouen Appearance-based gaze tracking with spectral clustering and semi-supervised Gaussian process regression. *Proceedings of the 2013 Conference on Eye Tracking* South Africa, Cape Town, South Africa, 2013, 17–23
- [10] K. Wang, B. Wang and L. Peng CVAP: Validation for Cluster Analyses. *Data Science Journal*, 8 (0), 88–93, 2009
- [11] A. Thalamuthu, I. Mukhopadhyay, X. Zheng and G. C. Tseng Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics* (Oxford, England), 22 (19), 2405–2412, 2006
- [12] S. Dudoit and J. Fridlyand A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3 (7), 2002
- [13] T. Caliński and J. Harabasz A dendrite method for cluster analysis. *Communications in Statistic*, 3 (1), 1–27, 1974
- [14] P. J. Rousseeuw Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53–65, 1987
- [15] D. L. Davies and D. W. Bouldin A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1 (2), 224–227, 1979
- [16] C. Boake From the Binet-Simon to the Wechsler-Bellevue: tracing the history of intelligence testing. *Journal of Clinical and Experimental Neuropsychology*, 24 (3), 383–405, 2002
- [17] V. Sicard, R. D. Moore, i D. Ellemberg Sensitivity of the Cogstate Test Battery for Detecting Prolonged Cognitive Alterations Stemming From Sport-Related Concussions. *Clinical Journal of Sport Medicine: Official Journal Canadian Academy Sport Medicine*, 29 (1), 62–68, 2019