



ZESZYTY NAUKOWE WYDZIAŁU

ELEKTRONIKI I INFORMATYKI

POLITECHNIKI KOSZALIŃSKIEJ NR **14**



0001011010101

POLITECHNIKA KOSZALIŃSKA

**Zeszyty Naukowe
Wydziału Elektroniki i Informatyki**

Nr 14

KOSZALIN 2019

Zeszyty Naukowe Wydziału Elektroniki i Informatyki Nr 14

ISSN 1897-7421
ISBN 978-83-7365-512-6

Przewodniczący Uczelnianej Rady Wydawniczej
Zbigniew Danielewicz

Przewodniczący Komitetu Redakcyjnego
Aleksy Patryn

Komitet Redakcyjny
Krzysztof Bzdrya
Walery Susłow
Wiesław Madej
Józef Drabarek
Adam Słowik

Strona internetowa
<https://weii.tu.koszalin.pl/nauka/zeszyty-naukowe>

Projekt okładki
Tadeusz Walczak

Skład, łamanie
Maciej Bączek

© Copyright by Wydawnictwo Uczelniane Politechniki Koszalińskiej
Koszalin 2019

Wydawnictwo Uczelniane Politechniki Koszalińskiej
75-620 Koszalin, ul. Raławicka 15-17

Koszalin 2019, wyd. I, ark. wyd. 5,25, format B-5, nakład 100 egz.
Druk: INTRO-DRUK, Koszalin

Spis treści

<i>Tomasz Królikowski, Walery Susłow</i>	5
Analityczny projekt informatyzacji przedsięwzięcia szkoleniowego	
<i>Patryk Widuliński, Krzysztof Wawryn</i>	23
Detekcja anomalii w plikach za pomocą wybranych algorytmów inspirowanych mechanizmami immunologicznymi	
<i>Anita Wąchała, Karol Hliwa, Antoni Wiliński</i>	43
Badania profili inteligencji wielorakich wśród studentów pierwszego roku informatyki	
<i>Paweł Poczekajło</i>	69
Interaktywna stacja pogodowa w oparciu o układ ESP z obsługą sieci WiFi	
<i>Paweł Poczekajło</i>	77
Pomiar oraz porównanie wrażliwości różnych realizacji filtrów FIR 3D	
<i>Rafał Wojszczyk, Piotr Stola</i>	83
Weryfikacja użycia metody oceny jakości implementacji wzorców projektowych w zastosowaniu produkcyjnym	
<i>Ewelina Hałat, Sławomir Kukła</i>	93
Wykorzystanie narzędzi do modelowania i symulacji w racjonalizacji procesów wytwarzania wyrobów cukierniczych	

Tomasz Królikowski
Wydział Technologii i Edukacji
Walery Susłow*
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
* walery.suslow@tu.koszalin.pl

Analityczny projekt informatyzacji przedsięwzięcia szkoleniowego

Słowa kluczowe: firma szkoleniowa, informatyczny system zarządzania, proces biznesowy, modelowanie konceptualne, architektura modułowa systemu

Dynamiczna sytuacja gospodarcza w kraju i na świecie utrzymuje wysoki popyt na usługi szkoleniowe. Świadczenie ich wymaga coraz bardziej zaawansowanego wsparcia informatycznego. Celem niniejszego artykułu jest usystematyzowanie założeń do projektowania systemu informatycznego, przeznaczonego do zarządzania procesami biznesowymi, występującymi w ramach przedsięwzięć realizujących szkolenia, w tym te prowadzone w trybie e-learningowym. Zakłada się, że wykonane przez autorów przedprojektowe, konceptualne modelowanie procesów biznesowych i czynności niezbędnych w zakresie organizacji, prowadzenia i rozliczania szkoleń, dostarczy projektantom zweryfikowanej wiedzy domenowej i poprzez to zwiększy jakość projektowanego systemu informatycznego, wspomagającego funkcjonowanie firmy świadczącej usługi szkoleniowe. Ponadto zwiększy ekonomiczną efektywność przedsięwzięć szkoleniowych zarządzanych za pośrednictwem takiego systemu.

Zaproponowany procesowy i architektoniczny model systemu informatycznego został opracowany na podstawie analizy biznesowej typowej firmy szkoleniowej z wykorzystaniem techniki BPMN (*Business Process Modeling Notation*). Model ten został zweryfikowany w ramach pilotażowych projektów wdrożeniowych, obejmujących realizację poszczególnych procesów, związanych z organizacją szkoleń na Politechnice Koszalińskiej, w tym z wykorzystaniem uczelnianej platformy e-learningowej. Wdrożenie i monitorowanie w warunkach rzeczywistych,

kluczowych modułów oprogramowania służącego wspomaganie procesów zarządzania szkoleniami, potwierdziło zgodność prezentowanego modelu z przyjętymi założeniami konceptualnymi.

Przeprowadzone badania składają się z:

- Modelowania procesowego w języku BPML (*Business Process Modeling Language*),
- mapowania konceptualnego (*Concept Mapping*),
- modelowania systemowego z wykorzystaniem języka UML (*Universal Modeling Language*).

Wiedza domenowa w projektach informatycznych

Terminem *wiedza domenowa* informatycy określają prawdziwą, weryfikowalną wiedzę odnoszącą się do zdefiniowanego obszaru ludzkiej działalności. Gdy używają wiedzy domenową konceptualną czy odwołują się do opinii eksperta domeny, to podkreślają w taki sposób wykorzystanie danej konkretnej domeny, która jest przedmiotem dyskursu, zainteresowania lub problemu [1]. W świetle przyjętej w Polsce klasyfikacji [2], wiedza niezbędna do rozpoczęcia prac projektowych nad systemem zarządzania przedsięwzięciem szkoleniowym należy generalnie do terminologicznego (definicje), metodycznego (pragmatyka) i normatywnego (potrzeby, oczekiwania, kryteria) obszarów domeny.

W inżynierii oprogramowania wiedza domenowa jest wiedzą o środowisku, w którym będzie działał docelowy system informatyczny [3]. Posiadanie doskonałej znajomości stuki programowania nie stanowi skarbu samo w sobie. Zespół projektowy może przystępować do pisania kodu tylko wtedy, gdy ma dobrą wiedzę domenową. Czyli, zespół potrzebuje ekspertów domenowych tak samo jak i ekspertów technicznych.

Twórcy oprogramowania zwykle pozyskują wiedzę domenową od użytkowników oprogramowania w danej domenie (są oni traktowani jako specjaliści domeny). Ta wiedza dotyczy zazwyczaj przepływów pracy użytkowników, danych biznesowych, polityki gospodarczej, charakterystycznych ograniczeń oraz terminów kluczowych do rozwoju oprogramowania. Pozyskana od użytkowników wiedza domenowa często bywa niekompletna, nieformalna i źle zorganizowana. Jednak powinna ona być przekształcana przez projektantów zespołu informatycznego w programy komputerowe i bazy danych.

Można nawet zdefiniować programowanie jako mapowanie wiedzy domenowej na kod, a stąd dobre opanowanie zarówno kodowania i domeny jest obowiązkowe. Nie można napisać kod systemu, nie wiedząc, co system powinien robić. W wielu przypadkach programista musi rozumieć domenę nawet bardziej szczegółowo niż

specjalista domenowy, ponieważ specjalista może rozwiązywać wszelkiego rodzaju wyjątki i przypadki na podstawie doświadczenia i zdrowego rozsądku, wykorzystując intuicję, a programista powinien jawnie obsłużyć wszystkie te sytuacje w kodzie, zanim one powstaną.

Komunikacja między użytkownikami oprogramowania i programistami jest często trudna. Muszą oni zbudować wspólny język porozumiewania się. Rozwijanie wspólnego słownictwa, niezbędnego do komunikacji, często może trwać za długo by zmieścić się w planowanym okresie trwania projektu.

W nietypowych projektach, jak ten omawiany w niniejszym artykule, bardzo trudne jest pozyskanie zawodowego eksperta dziedzinowego. Tak aby doradzać przy budowie systemu zarządzania przedsięwzięciem szkoleniowym, nie wystarczy być ekspertem w edukacji ani w zarządzaniu przedsiębiorstwem, obie specjalizacje są niezbędne, a poza tym należy posiadać zaawansowaną wiedzę w zakresie komunikacji z produkcyjnym zespołem informatycznym. Autorzy niniejszego artykułu, z racji swojego zawodu i posiadanego doświadczenia w projektowaniu oprogramowania dla edukacji, spodziewają się, że prezentowana przez nich wiedza okaże się przydatna dla twórców systemów szkoleniowych.

Podejście procesowe do zarządzania firmą szkoleniową

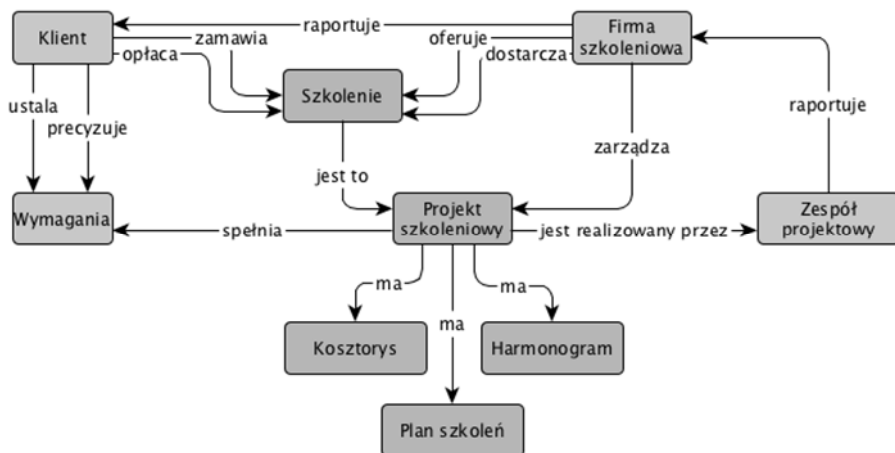
Założeniem procesowego podejścia do modelowania informatycznego systemu zarządzania jest optymalizacja firmy przede wszystkim w ujęciu procesowym (nie funkcjonalnym, do którego zazwyczaj są przyzwyczajeni informatycy). Celem podejścia procesowego jest uzyskanie wysokiego poziomu niezawodności w konkretnym działaniu biznesowym, w ramach niniejszego tematu – w organizacji i przeprowadzeniu szkoleń. Podejście to wymaga w pierwszej kolejności identyfikacji procesów biznesowych i opracowania ich mapy [4]. Znajomość mapy procesów pozwala na ich doskonalenie przy wsparciu ze strony osób zarządzających planowanym systemem informatycznym.

Traktując typową firmę szkoleniową jako organizację procesową [5], dokonuje się pewnego uproszczenia. W rzeczywistości firmy szkoleniowe w Polsce nie są zarządzane w oparciu o koncepcję organizacji procesowej. Niemniej jednak ze względu na ogólną tendencję popularyzacji podejścia procesowego nie tylko w środowisku menadżerskim, ale i wśród deweloperów informatycznych systemów zarządzania, próba zbadania potencjału takiego podejścia w konkretnym zastosowaniu wydaje się uzasadniona.

Przy wyborze techniki przeprowadzenia analizy procesowej autorzy zrezygnowali z wykorzystania potężnej, ale mniej popularnej w kraju, techniki IDEF [6] (*Integration DEFinition Methods*), na rzecz bardziej nowoczesnej BPMN [7] (*Business Process Modeling Notation*). Istotnym elementem przygotowania badań procesowych była pojęciowa (konceptualna [8]) analiza domeny problemu na

podstawie ontologicznej wiedzy z zakresu nauk o zarządzaniu [9]. Mapę głównych pojęć (terminów) domeny przedstawiono na rysunku 1.

Istotnym pojęciem domeny jest „szkolenie” – usługa świadczona przez firmę szkoleniową na rzecz klienta. Specyfiką omawianej mapy jest próba ukazania elementów dynamicznej (procesowej) ontologii domeny przy pomocy nazw relacji zachodzących pomiędzy pojęciami. Statyczne relacje zostały zachowane jedynie w opisie struktury szkolenia w ujęciu organizacyjnym, przyjęto bowiem koncepcję szkolenia jako autonomicznego projektu (przedsięwzięcia).



Rys. 1. Mapa kluczowych pojęć dotyczących przedsięwzięcia szkoleniowego [Źródło: opracowanie własne]

Kluczowe procesy, składające się na zarządzanie szkoleniami, to:

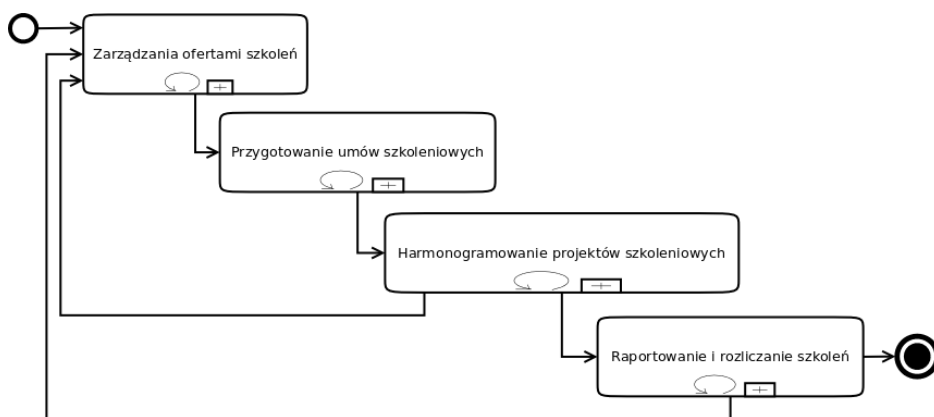
- proces zarządzania ofertami szkoleń (proces ofertowania),
- proces przygotowania umów szkoleniowych,
- proces harmonogramowania projektów szkoleniowych,
- proces raportowania i rozliczania szkoleń (patrz rys. 2).

Proces zarządzania ofertami szkoleń

Firma realizująca szkolenia otrzymuje wiele zapytań ofertowych od różnych podmiotów gospodarczych, w tym agencji marketingowych [10]. Z jednej strony, z uwagi na przetargowy charakter pracy agencji, tylko mała część z przygotowanych ofert dochodzi do realizacji. Z drugiej strony, mimo wydatków związanych z przygotowaniem ofert, wygrywane w przetargach zlecenia są atrakcyjne pod

względem merytorycznym i ekonomicznym. Problemem jest to, że agencje bardzo często formułują jedynie ideę szkolenia i oczekują od wykonawcy technicznego i merytorycznego doradztwa, mającego na celu doprecyzowanie planu realizacji szkoleń.

Również w przypadku innych klientów, firma realizująca szkolenia, musi poświęcać dużo czasu na doprecyzowanie ich wymagań [11] — zapytania ofertowe zazwyczaj są bardzo ogólne. Nawet gdy klient deklaruje szczegółowo przemyślany temat i zakres szkolenia, do wystawienia oferty zazwyczaj brakuje dokładnego opisu celów szkolenia, warunków wstępnych, wskaźników jakości i innych elementów składających się na treść oferty. Czasem, przed przygotowaniem wyceny, firma realizująca szkolenia musi poświęcić wiele godzin na spotkania z klientem w celu doprecyzowania oczekiwań.

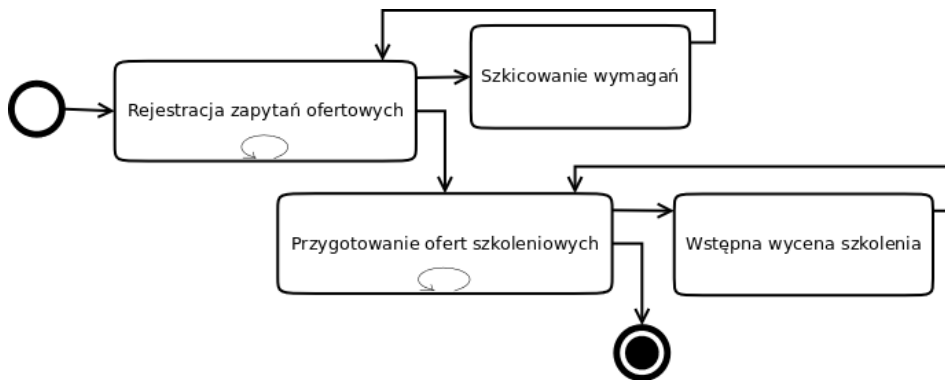


Rys. 2. Mapa kluczowych procesów biznesowych przedsięwzięcia szkoleniowego [Źródło: opracowanie własne]

Firma szkoleniowa może również pośredniczyć w realizacji szkoleń – zawierać umowy z zewnętrznymi dostawcami szkoleń, kupować od nich materiały szkoleniowe, korzystać z usług specjalistów z firm zewnętrznych, na podstawie podpisanych umów o współpracy. Te bardziej złożone konfiguracje biznesowe będą dodatkowo komplikować proces zarządzania szkoleniami oraz w przyszłości – procesy harmonogramowania, raportowania i rozliczania szkoleń.

Warto również zwrócić uwagę na fakt, że jedynie po otrzymaniu szczegółowej wyceny godzinowej szkolenia można oszacować dokładny koszt całego projektu szkoleniowego [12]. Wbrew pozorom, nie jest to łatwy etap procesu ofertowania. Pracownicy firmy szkoleniowej mają różne stawki godzinowe, odmienną wydajność, inwencję twórczą itp. W większości projektów pojawiają się również koszty dodatkowe, przykładowo zakup lub utworzenie pomocy multimedialnych czy specjalistycznego oprogramowania, wypożyczenie sprzętu i wynajem

pomieszczeń. Nawet po zebraniu wszystkich danych, niezbędnych do kalkulacji kosztów, koordynator szkoleń jest w stanie bez wsparcia komputerowego jedynie orientacyjnie zweryfikować historię współpracy z klientem i z dużym marginesem niepewności zdecydować o tym, jaką marżę naliczyć na projekt oraz jaki termin wyznaczyć do jego realizacji. Istotnym utrudnieniem może być także jednoczesna realizacja przez firmę wielu projektów szkoleniowych i brak systemu do dokładnego ich harmonogramowania.



Rys. 3. Struktura procesu zarządzania ofertami szkoleń [Źródło: opracowanie własne]

Przygotowanie pojedynczej oferty szkoleniowej zajmuje firmie od kilku do kilkunastu roboczogodzin. Działanie to często wymaga zaangażowania kluczowych pracowników firmy szkoleniowej, nie tylko osób odpowiedzialnych za zdobywanie zleceń i za kontakt z klientem. Wynika z tego wprost konieczność komputerowego wspomaganie procesu zarządzania ofertami szkoleniowymi.

Podproces rejestracji zapytań ofertowych (patrz rys. 3) powinien zbierać wszystkie niezbędne wymagania klienta w inteligentny sposób. Oznacza to, że nawet w przypadku, gdy klient nie będzie posiadał sprecyzowanej wizji potrzebnego szkolenia, interaktywny formularz rejestracyjny powinien pomóc mu w sprecyzowaniu zapotrzebowania. Zdaniem autorów, podstawą do prowadzenia dialogu z klientem podczas zbierania wymagań mogą posłużyć mechanizmy, stosowane przez specjalistów od sztucznej inteligencji przy konstruowaniu programów komputerowych przeznaczonych do prowadzenia konwersacji między człowiekiem a komputerem przy użyciu języka naturalnego (ang. chatterbot). Najbardziej znanym obecnie przykładem takiego oprogramowania jest ALICE Bot [13]. Dodatkowo przy konstruowaniu kodu, który będzie decydował o kompletności i spójności zapotrzebowania, zaleca się wykorzystanie sieci neuronowej. Po nauczaniu pozwoli ona budować zapytania ofertowe z uwzględnieniem tradycji i priorytetów firmy szkoleniowej.

Konstruowanie podprocesu rejestracji zapytań ofertowych polega na tym, że należy opracować algorytm szkicowania wymagań klienta, który w prosty sposób utrzymałby kluczowe parametry zgłaszanego szkolenia (ewentualnie poprzez umożliwienie dołączenia do formularza zgłoszeniowego plików tekstowo-graficznych: zdjęć, wykresów, schematów, notatek, szkiców). Podproces ten powinien wymuszać rejestrację przedstawiciela klienta jako użytkownika systemu. Dzięki temu każdy z klientów będzie miał dostęp do swoich zapytań ofertowych, przedstawionych w aktualnej, strukturyzowanej formie. Jednocześnie łatwo będzie zorganizować klientowi dostęp do historii jego poprzednich zapytań. W ten sposób czas potrzebny na zbieranie wstępnych wymagań od klienta może zostać skrócony do niezbędnego minimum.

System informatyczny powinien natychmiast informować odpowiedzialnego pracownika firmy szkoleniowej o zdarzeniu otrzymania nowego zapytania ofertowego. Kluczowym pomysłem dla budowania mechanizmu przetwarzania takiego zdarzenia jest to, że prowadzony przez system mechanizm kontekstowych komentarzy ma pozwalać odpowiedzialnemu pracownikowi na natychmiastową pisemną komunikację z klientem w sprawie danego zapytania, a przedstawicielowi klienta – na szybkie ustosunkowanie się do komunikatów pracownika. Algorytm szkicowania wymagań powinien być stosowany do wszystkich zapytań wpływających do firmy szkoleniowej. Pozwoli to oszczędzić czas poświęcany na precyzyjne formułowanie zapytań ofertowych tworzonych za pośrednictwem systemu informatycznego.

Po rejestracji, odpowiednio przygotowane zapytanie, może zostać w łatwy sposób podzielone na podzadania i rozesłane do odpowiednich pracowników bądź dostawców usług szkoleniowych. Dzięki udziałowi dostawców w generowaniu ofert, cały proces ofertowania można w pełni zautomatyzować. System informatyczny powinien być gwarantem tego, że informacje o potrzebie stworzenia wyceny otrzymają jedynie zainteresowane osoby. W ten sposób zostanie również zachowana poufność komercyjna prowadzonych negocjacji.

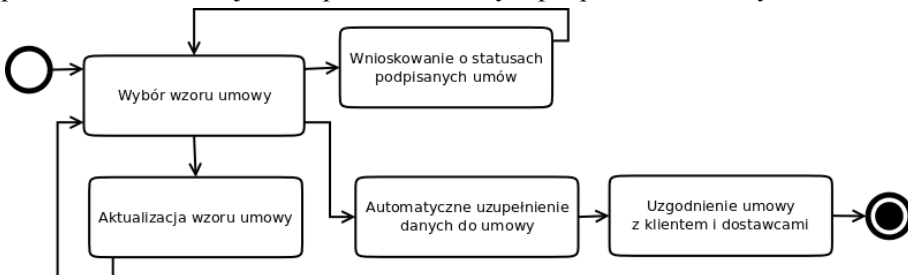
Wprowadzenie w firmie szkoleniowej ujednoczonego algorytmu wyceny szkoleń dodatkowo usystematyzuje przepływ dokumentów. Każdy z dostawców będzie dostawał fragmenty dokumentacji przedsięwzięcia szkoleniowego, odpowiadające jego kompetencjom. Dzięki zestawowi odpowiednich formularzy, wyceny będą zdecydowanie bardziej precyzyjne oraz zawierać niezbędne dane. Pozwolą również zredukować czas przeznaczony na wycenę. Oferty przygotowane przez pracowników firmy szkoleniowej i przez zewnętrznych dostawców będą mogły być poddane automatycznej analizie dzięki algorytmowi automatycznej analizy ofert, który zweryfikuje zmienne brzegowe, i wybierze najlepszą ofertę według przyjętego kryterium. Dany algorytm stanowi autorskie know-how budowy systemu informatycznego.

Proces przygotowania umów szkoleniowych

Każda firma szkoleniowa powinna posiadać przygotowane przez prawników wzory umów dla różnych wariantów współpracy z klientem (umowę o dzieło, umowę o dzieło z rozszerzonym zakresem gwarancji, umowę poufności, umowę o dzieło z klauzulą poufności, umowę współpracy, szablon zamówienia oraz inne tego typu dokumenty). Wybór rodzaju umowy dla danego zlecenia szkoleniowego zależy od wielu czynników. Przykładowo, jeżeli już została podpisana umowa współpracy z klientem, to do kolejnych szkoleń podpisywane powinny być jedynie zlecenia na wykonanie usługi. Stąd, wiedza o stanie faktycznym podpisanych dokumentów, dotyczących danego klienta, jest kluczowa przy przygotowaniu nowych umów szkoleniowych.

Każdorazowo, aby określić status podpisanych z klientem umów, niezbędne jest sprawdzenie: czy poprzednie umowy zostały już podpisane przez firmę szkoleniową, czy zostały wysłane do klienta, czy wróciły do firmy szkoleniowej podpisane kopie tych umów. Dopiero po ustaleniu stanu faktycznego dokumentacji związanej z danym klientem można wybrać odpowiedni wzór nowej umowy. Na jego bazie, wspólnie z klientem, zostaną doprecyzowane zapisy dotyczące konkretnego zlecenia szkoleniowego.

W procesie uzgadniania ostatecznej treści aktualnej umowy szkoleniowej (patrz rys. 4) bierze udział kilku decydentów z każdej strony. Tradycyjny tryb wymienia się między nimi wątpliwościami, zapisanymi w komentarzach w dokumencie tekstowym lub arkuszu kalkulacyjnym (MS Office, OpenOffice, Libre Office), powoduje często zamieszanie co do aktualnie uzgodnionego stanu przygotowywanej umowy. Proces ten, prowadzony tradycyjnie przez wymianę mailową załączników (dokumentów tekstowych i arkuszy kalkulacyjnych), często wydłuża się i powoduje opóźnienia. Dopiero ustalenie ostatecznej wersji zapisów pozwala formalnie podpisać umowę o współpracy przy danym zleceniu. W praktyce, szczególnie przy współpracy z agencjami, często zdarza się na tyle krótki termin realizacji umowy szkoleniowej, że firma szkoleniowa musi zaczynać prace nad zleceniem jeszcze przed ostatecznym podpisaniem umowy.



Rys. 4. Kluczowe czynności procesu przygotowania umowy szkoleniowej
[Źródło: opracowanie własne]

Podpisaną umowę należy przechowywać w dokumentacji firmy prowadzącej szkolenia. Po realizacji szkolenia umowa w komplecie z innymi dokumentami powinna być przekazana do archiwum. Praktyka prowadzenia dokumentacji nie zawsze sprzyja ułatwieniu dostępu do umów, szczególnie tych zarchiwizowanych. Jednocześnie logika biznesu szkoleniowego nakazuje wracać do otwartych i zrealizowanych umów przy przygotowaniu kolejnych umów z tym samym klientem lub na podobne szkolenia z innymi klientami. Stworzenie komputerowej bazy wiedzy o faktycznym stanie i zawartości wszystkich podpisanych umów szkoleniowych przyspieszyłoby proces uzgadniania warunków nowotworzonych umów i podniosłoby jakość dokumentów. Zawartość tej bazy stanowi tajemnicę firmy szkoleniowej.

Zautomatyzowanie procesu wnioskowania o statusach podpisanych umów i wyboru wzoru umowy, na podstawie którego należy przygotowywać kolejną z konkretnym klientem, powinno być jednym z istotnych zadań przy wdrożeniu systemu informatycznego zarządzania przedsięwzięciem szkoleniowym. Przechowywanie w hurtowni danych aktualnej, elektronicznej kopii każdej umowy wyeliminuje błędy związane z ustalaniem ostatecznej formy nowo tworzonej umowy. Korzystanie ze zgromadzonych danych powinno pozwolić również na automatyczne uzupełnianie odpowiednich pozycji nowej umowy, dotyczących atrybutów klienta oraz zakresu prac do wykonania. Dane te muszą być pobierane z hurtowni danych (centralnego repozytorium informacyjnego firmy szkoleniowej). Przy mniejszym rozmiarze firmy, mechanizmy hurtowni mogą być zastąpione mechanizmami bazy danych.

Kluczowe mechanizmy systemu informatycznego, realizującego czynność wnioskowania, powinny umożliwiać systematyczną aktualizację wzorów oraz dostarczać funkcję komentowania wszystkim interesariuszom przedsięwzięcia. Pozwoli to wdrożyć scentralizowane informowanie interesariuszy i ułatwi tworzenie ostatecznej wersji dokumentu umowy. Mechanizm zarządzania uprawnieniami użytkowników systemu informatycznego powinien zapewnić bezpieczeństwo dostępu do danych które mogłyby przy ich wycieku stanowić punkt przewagi dla konkurencji.

Proces harmonogramowania projektów szkoleniowych

Przy generowaniu oferty, sporządzanej na podstawie wyceny dokonanej przez merytorycznych pracowników, należy wstępnie oszacować liczbę roboczogodzin, potrzebnych do przygotowania i realizacji zleconego szkolenia. Konieczne jest również wygenerowanie wstępnego harmonogramu prac przygotowawczych i zajęć szkoleniowych, do którego powinien być dodany margines bezpieczeństwa (czasu, zasobów ludzkich oraz materialnych). Należy oszacować optymalne terminy rozpoczęcia i ukończenia szkoleń, które muszą uwzględniać zaangażowanie

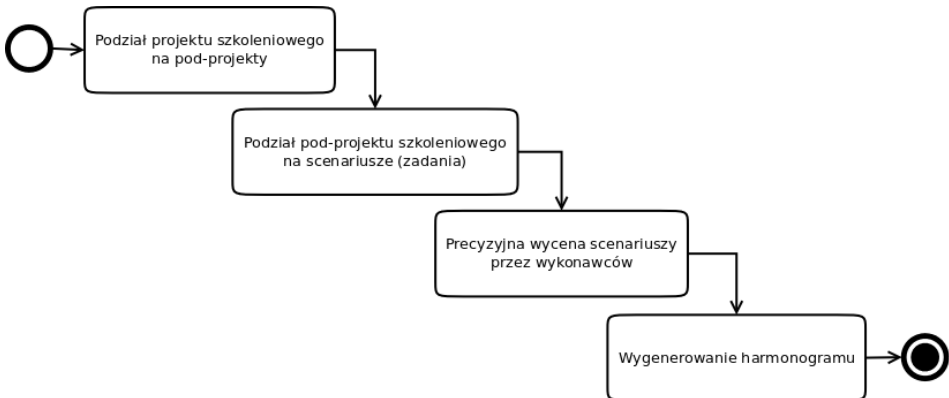
pracowników w innych projektach i czynnościach, prowadzonych przez firmę szkoleniową. Na podstawie wycen, większy projekt szkoleniowy może być podzielony na etapy (moduły), do których będą również potrzebne wstępne wyceny i harmonogramy.

Komputeryzacja procesu harmonogramowania projektów szkoleniowych (patrz rys. 5) wymaga po stronie firmy prowadzącej szkolenia dużych zmian, mających na celu:

- usystematyzowanie przydziału prac,
- optymalizację podziału zadań,
- doprecyzowanie algorytmu szacowania terminów,
- optymalizację wykorzystania zasobów ludzkich firmy.

Wprowadzenie zmian w procesie pozwoli na zwiększenie efektywności firmy szkoleniowej, a co się z tym wiąże— na znaczący wzrost jej dochodów.

Na podstawie podpisanych umów, system zarządzania firmą szkoleniową powinien umożliwić generowanie listy precyzyjnie sformułowanych zadań do wykonania. Cały projekt szkoleniowy należy podzielić na podprojekty, a następnie na tak zwane scenariusze. Scenariusze powinny stanowić zamkniętą całość z biznesowego punktu widzenia klienta. Przygotowane w ten sposób jednostki zadaniowe należy rankingować pod względem ich wartości biznesowych, tak, aby w przypadku ewentualnych opóźnień w projektach szkoleniowych móc dostarczyć klientowi wartości kluczowe dla jego biznesu. Wprowadzenie takiej klasyfikacji zadań pozwoli na uzasadniony podział projektu szkoleniowego na moduły. Dzięki temu będzie można aktywnie zarządzać realizacją projektu i na bieżąco zgłaszać ewentualne problemy i zmiany.



Rys. 5. Kluczowe czynności procesu harmonogramowania szkoleń [Źródło: opracowanie własne]

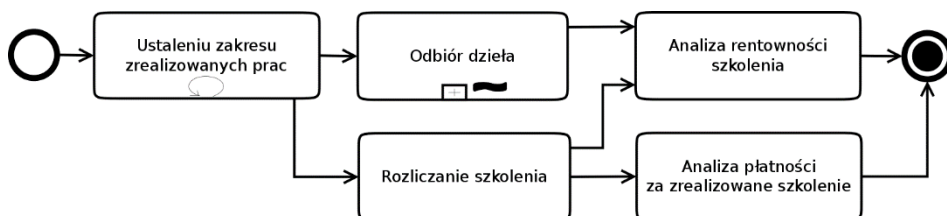
Zgodnie z proponowaną strukturą procesu harmonogramowania, wymagania dotyczące każdego scenariusza będą przed jego realizacją wyceniane przez wykonawców (pracowników firmy szkoleniowej) w jednostkach „trudności” zawartych w nim zadań. Zastosowanie takiego podejścia pozwoli oszacować, na podstawie średniej tygodniowej, jaki procent scenariusza zostanie wykonany w planowanym okresie. Używając archiwalnych statystyk dotyczących realizacji zadań wycenianych wcześniej przez tych samych wykonawców, system będzie w stanie zweryfikować najnowszą wycenę wykonawców i obliczyć kalendarzowy czas realizacji zadań przez dany konkretny zespół szkoleniowy. Zastosowanie autorskiej metody wyceny scenariuszy powiązanej z archiwalnymi statystykami pozwoli na bardziej realistyczne harmonogramowanie zadań, a co za tym idzie, na zautomatyzowane monitorowanie wydajności wykonawców i na zwiększenie zadowolenia klienta.

Istotną zmianą w podejściu do harmonogramowania powinny być również zmiany osobowo-zespołowe. W polskich firmach szkoleniowych nie istnieją zazwyczaj ewidentne podziały na zespoły – wszyscy pracownicy wykonują wszystkie zadania, co skutkuje bardzo wysokim narzutem na wprowadzenie nowych pracowników. Koncepcja autorskiego systemu informatycznego zarządzania przewiduje podział firmy szkoleniowej na interdyscyplinarne mini-zespoły, z wewnętrzną specjalizacją pracowników. Do takich zespołów będą przypisywane zadania do realizacji. Wtedy, w przypadku wyjątków, system będzie w stanie przekierować zadanie do alternatywnego zespołu, żeby zostało ono rozwiązane jak najmniejszym kosztem w wyznaczonym terminie.

Proces raportowania i rozliczania szkoleń

Nieuniknione w trakcie realizacji przedsięwzięcia zmiany kadrowe i organizacyjne będą powodować w firmie szkoleniowej sytuacje, w których proces rozliczania projektów szkoleniowych ze zleceniodawcami i dostawcami może sprawiać problemy. Przykładowo w praktyce zdarza się, że z powodu błędów w rejestrach przepracowanych godzin trudno jest ustalić, ile godzin każdy pracownik przepracował w danym projekcie szkoleniowym. Dodatkowo z niektórymi dostawcami szkoleń rozliczanie może odbywać się w oparciu o kwotę ustaloną za projekt, a z innymi na podstawie faktycznie przepracowanych godzin. W ustaleniu zakresu zrealizowanych prac może co prawda pomóc analiza kalendarzy pracowników realizujących szkolenia, ale jest to czasochłonny proces. Komputeryzacja podprocesu raportowania postępów prac nad projektem szkoleniowym zdecydowanie przyczyni się do zwiększenia dyscypliny pracy i zmniejszy liczbę błędów popełnianych przy rozliczeniu szkoleń. Raportowanie może być sprzężone z harmonogramowaniem szkoleń, co zaowocuje ułatwieniem monitoringu postępów i opóźnień w prowadzonych szkoleniach.

Po ustaleniu zakresu zrealizowanych prac (które należy rozliczyć) powinien nastąpić podproces odbioru dzieła (patrz rys. 6). Wymaga on obustronnej zgody: najpierw prace odbiera firma szkoleniowa od swoich pracowników lub dostawców szkoleń, a następnie – klient końcowy (zleceniodawca szkoleń). W praktyce ten dwuetapowy podproces biznesowy często generuje nieporozumienia co do stanu zleconej usługi: czy jest ona już ostatecznie odebrana, czy nie? W ramach odbioru dzieła powinno nastąpić również przygotowanie protokołów odbioru, na podstawie których zostaną wystawione faktury. Jednocześnie powinno być wykonane zadanie rozliczania szkoleń, po którym w systemie powinna zostać uruchomiona automatyczna analiza płatności za zrealizowane szkolenie.



Rys. 6. Proces raportowania i rozliczania szkoleń [Źródło: opracowanie własne]

Wygenerowane w ramach procesu raportowania i rozliczania szkoleń dokumenty powinny być jednoznacznie powiązane z konkretnymi projektami szkoleniowymi, ułatwi to analizę rentowności tych projektów. Przy spełnieniu danego wymogu pion zarządczy firmy szkoleniowej będzie miał natychmiastową informację o tym, że niektóre zrealizowane szkolenia przynoszą straty. Nie jest to jedyna zaleta wymuszonej asocjacji wszystkich dokumentów prowadzonych przez system zarządzania z konkretnymi szkoleniami. Umożliwi to również zwiększenie skuteczności zarządzania projektami dzięki ułatwieniu chronologicznej analizy ekonomicznych wskaźników współpracy z każdym konkretnym klientem. Taka funkcjonalność staje się możliwa do realizacji w ramach opisywanego systemu. Nie wymaga ona dużych nakładów pracy projektanta, bo chodzi jedynie o wygenerowanie stosownych raportów na podstawie danych już zawartych w bazie.

Problemy w zarządzaniu firmą szkoleniową może stwarzać utrudniona analiza płatności. Pion zarządczy firmy szkoleniowej powinien mieć aktualną informację o tym, czy wymagane od klientów kwoty wpłynęły już na konto firmy szkoleniowej oraz czy przelewy dla dostawców zostały już zrealizowane. Informacje te są związane ze stanem kont bankowych, zaznaczanych w umowach zlecenia i w wystawianych fakturach, przez co wiedza ta może być rozproszona. Jedynie w programie księgowym (u księgowego firmy szkoleniowej) wiedza taka jest skupiona w jednym miejscu, ale, ze względów bezpieczeństwa, nie powinien on być

podłączony do internetu i nie może udostępniać informacji o stanie kont do analizy osobom trzecim. Ponadto bardzo często rozliczenia nie są prowadzone przez księgowych na bieżąco (często księgowość jest obsługiwana na zasadzie outsourcingu – przez zewnętrzny podmiot, a nie własnych pracowników etatowych).

Proces rozliczania szkoleń można w znacznym stopniu zautomatyzować poprzez raportowanie wykonanych godzin przez wykonawców i dostawców, co pozwoli na automatyczne naliczanie kwot za wykonanie dzieła. Oznaczenie przez klienta w systemie informatycznym zadań jako „odebranych” pozwoli firmie szkoleniowej i dostawcom wnioskować o możliwości rozliczenia danego zlecenia.

Dzięki przechowywaniu informacji o wymaganiach dotyczących zlecenia danego klienta w systemie, wszyscy interesariusze będą mieli wspólny stan wiedzy na temat spełnienia wymagań projektu szkoleniowego. Pozwoli im to lepiej weryfikować kompletność wykonanych prac. Automatyczne zliczanie przepracowanych godzin i zrealizowanych zadań umożliwi określenie wynagrodzenia należnego wykonawcy (dostawcy). Automatyczne przypominanie o wystawieniu faktury oraz automatyczne monitorowanie zrealizowanych zobowiązań i należności pozwoli zaoszczędzić czas przeznaczony na ręczne zarządzanie tymi obszarami. Dzięki funkcji wystawiania faktur przez system oraz możliwości wgrania faktur kosztowych dla danego projektu będzie możliwe analizowanie rentowności każdego zlecenia – pozwoli to na lepsze szacowanie przyszłych ofert dla każdego klienta.

Proponowana struktura systemu do zarządzania szkoleniami

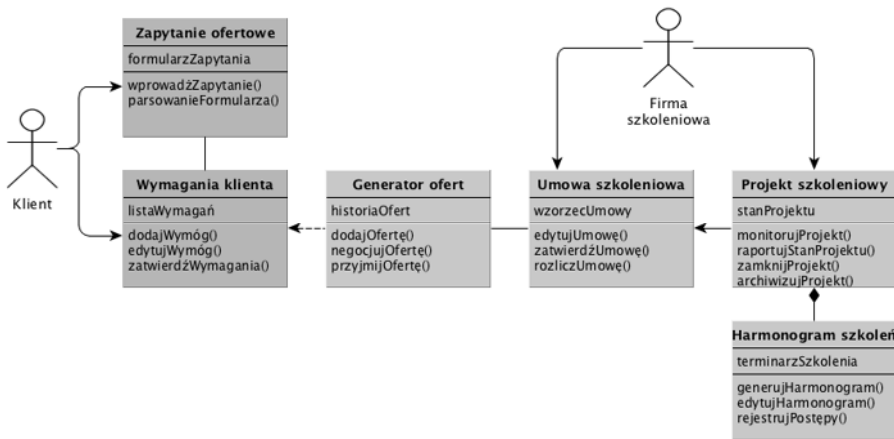
Obiekty biznesowe, wynikające z analizy dynamiki, zostały przedstawione na rysunku 7 w notacji UML. Można zauważyć, że aktywność biznesowa klienta powinna koncentrować się na precyzyjnej specyfikacji wymagań do zamawianych szkoleń, a aktywność firmy szkoleniowej – na ofertowaniu, przygotowaniu, raportowaniu i rozliczeniu umów oraz harmonogramowaniu projektu szkoleniowego.

Na podstawie przeprowadzonej analizy procesowej i obiektowej można zasugerować deweloperom strukturę modułów systemu do zarządzania szkoleniami. Przede wszystkim powinny być poddane pod rozważenie następujące moduły systemu:

- Moduł obsługi zapytań ofertowych;
- Moduł szkicowania i precyzowania wymagań;
- Moduł generowania ofert;
- Moduł wspomagający tworzenie umów szkoleniowych;
- Moduł harmonogramowania szkoleń;
- Moduł monitorowania postępów projektów szkoleniowych;
- Moduł raportowania i rozliczania szkoleń.

Lista wymienionych modułów nie zawiera propozycji dotyczących oczywistych składników każdego systemu informatycznego, np. modułów zarządzania użytkownikami systemu, archiwizacji dokumentów czy dostępu do repozytorium danych. Ze względu na wysoką dynamikę biznesu szkoleniowego wśród tych modułów powinien być moduł mobilnego dostępu. Umożliwienie uczestnikom procesów dostępu do dokumentacji z poziomu urządzeń mobilnych może stworzyć narzędzie do budowania przewagi konkurencyjnej na rynku usług szkoleniowych.

Oprócz wymienionych modułów, w zależności od modelu biznesowego firmy szkoleniowej, przydatny może być również moduł do wspomagania współpracy z zewnętrznymi dostawcami i partnerami.



Rys. 7. Biznesowe obiekty systemu zarządzania firmą szkoleniową [Źródło: opracowanie własne]

Na zakończenie warto podkreślić, że opisana w danym artykule koncepcja budowy systemu zarządzania szkoleniami została oparta na doświadczeniach autorów z wewnątrzuczelnianym projektem systemu do oceny modułowych efektów kształcenia na Politechnice Koszalińskiej. Zainteresowany czytelnik może zapoznać się z opisem tamtego projektu za pośrednictwem czterech artykułów naszego autorstwa [14-17]. Rozwiązania użyte w tamtym systemie stanowią prototypy niektórych rozwiązań dla systemu zarządzania opisanego w niniejszym artykule.

Podsumowanie

Artykuł prezentuje usystematyzowane przez autorów założenia do projektowania informatycznego systemu zarządzania firmą szkoleniową. Autorzy spodziewają się, że przedprojektowe konceptualne modelowanie procesów biznesowych w połączeniu z analizą pojęciową i obiektową domeny problemu,

pozwoili zaproponować spójny i logiczny model systemu informatycznego. Model ten został zweryfikowany w wybranych fragmentach podczas organizacji szkoleń na Politechnice Koszalińskiej, w tym z wykorzystaniem uczelnianej platformy e-learningowej. Wdrożenie koncepcji budowy systemu w postaci umowy na zasadzie licencji niewyłącznej, jest wskaźnikiem sukcesu przeprowadzonych badań. Przyniosło także wymierne korzyści finansowe zarówno firmie wdrażającej, jak i Politechnice Koszalińskiej.

Chciałoby się na końcu zwrócić uwagę czytelnika na to, że autorzy artykułu zastosowali trzyetapowy schemat rozważania nad wiedzą domenową: od modelu pojęciowego domeny, poprzez modele procesowe i do modelu konceptualnego struktury systemu informatycznego. Ten schemat może mieć znaczenie metodologiczne – stanowić wzór do naśladowania przy analizie innych domen, które należałoby rozpracować przed uruchomieniem dla nich dedykowanego projektu informatycznego.

Bibliografia

1. B. Hjørland, H. Albrechtsen, *Toward A New Horizon in Information Science: Domain Analysis*. Journal of the American Society for Information Science, 1995, 46(6), pp. 400-425.
2. M. Trocki, *Zarządzanie wiedzą w projektach*, SGH w Warszawie, Warszawa 2011, s. 87.
3. B. Hjørland, *Domain analysis in information science: Eleven approaches – traditional as well as innovative*, Journal of Documentation, 2002, 58(4), pp. 422 – 462.
4. A. Anders, *Zarządzanie procesowe i mapowanie procesów biznesowych*, Wydawnictwo PWE, Warszawa 2008.
5. J. Auksztol [i in.], *Modelowanie organizacji procesowej*, [red. nauk.] J. Auksztol, M. Chomuszek, Wydawnictwo Naukowe PWN, Warszawa 2012.
6. B. Guszczak, *Charakterystyka analizy procesowej w przedsiębiorstwie*, „Logistyka”, nr 3, Poznań 2015, s. 1730-1739.
7. S. Drejewicz, *Zrozumieć BPMN. Modelowanie procesów biznesowych*, Wyd. Helion, Gliwice 2012.
8. W. Susłow, *Analiza i modelowanie konceptualne w inżynierii systemów oprogramowania – ujęcie humanistyczne*, Wydawnictwo Uczelniane Politechniki Koszalińskiej, Koszalin 2013.
9. J.R. Sobczyk, *Filozofia procesu i podejście procesowe – implikacje ontologiczne w naukach o zarządzaniu*, Studia i Prace Uniwersytetu Ekonomicznego, nr 5, Kraków 2009, s. 19-29.

10. R. Szczepaniak [i in.], *Training Partners. Zarządzanie projektem szkoleniowym*, [red. nauk.] B. Gancarz, Wyd. Helion, Gliwice 2008.
11. H. Paluch, *Prawne podstawy świadczenia usług szkoleniowych na rynku niepublicznym*, „e-mentor”, nr 2 (19), Warszawa 2007, <http://www.e-mentor.edu.pl/artykul/index/numer/19/id/415>, [8.01.2019].
12. M. Dubel-Borycka, *Analiza potrzeb szkoleniowych i plan szkoleń dla Volvo Poland*, <https://www.parp.gov.pl/files/74/517/20719.pdf>, [8.01.2019].
13. A. L. I. C. E. *Artificial Intelligence Foundation*, <http://alice.pandorabots.com>, [8.01.2019].
14. W. Susłow, A. Słowik A., T. Królikowski, *Metodologiczne podejście do procedury oceny efektów kształcenia zawodowego informatyka*, Polskie Stowarzyszenie Zarządzania Wiedzą. Seria: Studia i Materiały, nr 56, Bydgoszcz 2011, s. 196-206.
15. W. Susłow, A. Słowik, T. Królikowski, *Procedura oceny efektów kształcenia, osiągniętych przez studentów specjalności informatycznych*, Polskie Stowarzyszenie Zarządzania Wiedzą. Seria: Studia i Materiały, nr 56, Bydgoszcz 2011, s. 184-195.
16. W. Susłow, A. Słowik, M. Statkiewicz, T. Królikowski, *Niekosztowna, otwarta implementacja systemu do weryfikacji modułowych efektów kształcenia*, Studia Informatica "Informatyka stosowana. Wybrane metody i problemy." Zeszyty Naukowe Wydziału Nauk Ekonomicznych i Zarządzania nr 31, zeszyt naukowy 740. Wydawnictwo Uniwersytetu Szczecińskiego, Szczecin 2012, s. 217-231.
17. W. Susłow, A. Słowik, M. Statkiewicz, T. Królikowski, *Specyfikacja wymagań systemu informacyjnego do pomiaru modułowych efektów kształcenia*, Studia Informatica "Informatyka stosowana. Wybrane metody i problemy." Zeszyty Naukowe Wydziału Nauk Ekonomicznych i Zarządzania nr 31, zeszyt naukowy 740. Wydawnictwo Uniwersytetu Szczecińskiego, Szczecin 2012, s. 233-248.

Streszczenie

Celem niniejszego artykułu jest usystematyzowanie założeń do projektowania systemu informatycznego, przeznaczonego do zarządzania procesami biznesowymi, występującymi w ramach przedsięwzięć realizujących szkolenia, w tym te prowadzone w trybie e-learningowym. Zakłada się, że wykonane przez autorów przedprojektowe, konceptualne modelowanie procesów biznesowych i czynności niezbędnych w zakresie organizacji, prowadzenia i rozliczania szkoleń, dostarczy projektantom zweryfikowanej wiedzy domenowej i poprzez to zwiększy jakość projektowanego systemu informatycznego, wspomagającego funkcjonowanie firmy świadczącej usługi szkoleniowe.

Abstract

The purpose of this article is standardisation of the assumptions for the design of an IT system intended for managing business processes present within the framework of training implementation undertakings, including those carried out in the e-learning mode. It is assumed that the pre-project, conceptual modelling of business processes and operations necessary for the organization, running and settlement of training, developed by the authors, shall provide the designers with the revised domain knowledge and by this it will increase the quality of the proposed IT system supporting the functioning of a company providing training services.

Keywords: training company, management information system, business process, concept modelling, modular system architecture

Patryk Widuliński*

Krzysztof Wawryn

Wydział Elektroniki i Informatyki

Politechnika Koszalińska

* patryk.widulinski@tu.koszalin.pl

Detekcja anomalii w plikach za pomocą wybranych algorytmów inspirowanych mechanizmami immunologicznymi

Słowa kluczowe: systemy wykrywania intruzów, sztuczne systemy immunologiczne, wirusy, szkodliwe oprogramowanie, algorytm negatywnej selekcji, generacja receptorów, anomalia, wykrywanie anomalii

1. Wstęp

Od ostatnich kilkudziesięciu lat ochrona przed atakami wirusowymi jest ważnym zadaniem projektantów oprogramowania zabezpieczającego. Wskutek tego na całym świecie postępuje rozwój systemów wykrywania intruzów (SWI). Systemy te używane są na całym świecie do zabezpieczania przed atakami sieciowymi na poziomie zarówno lokalnym, jak i globalnym. Problemami, które między innymi napotyka projektanci oprogramowania zabezpieczającego są wykrywanie wcześniej nieznanych ataków i rosnąca złożoność oprogramowania komputerowego. Nakłonieni do poszukiwania nowatorskich rozwiązań tych problemów, projektanci dostrzegli analogie między zadaniami realizowanymi przez systemy wykrywania intruzów i ludzki układ odpornościowy. Przystosowanie mechanizmów ludzkiego układu odpornościowego pozwala na rozwiązanie wymienionych wyzwań. W literaturze pojawiło się dużo podejść do tematu systemów wykrywania intruzów bazujących na sztucznych systemach immunologicznych. Większość z nich korzysta z systemów wykrywania anomalii i algorytmach selekcji negatywnej w celu wykrywania infekcji w oprogramowaniu komputerowym. Metoda negatywnej selekcji oparta jest na rozpoznawaniu wzorców własnych i obcych i prowadzi do wykrywania anomalii w oprogramowaniu. Podstawy mechanizmów

systemów wykrywania intruzów opartych o sztuczne systemy immunologiczne opisane są w [1-5]. Inicjują one bardziej skomplikowane systemy zaprezentowane w [6-13].

W artykule zaprezentowano system wykrywania intruzów w postaci nieprawidłowości (infekcji) w treści skompilowanych programów komputerowych. Mechanizm wykrywania oparty jest na algorytmie negatywnej selekcji. Składa się z dwóch zadań: generacji receptorów i wykrywania anomalii. Sekcje kodu w programach reprezentowane są przez ciągi binarne. Generacja receptorów realizowana jest na dwa sposoby: losowo i korzystając z algorytmu negatywnej selekcji do podziału zbioru szablonów na własne i obce. Szablony własne są odrzucane, tak, by infekcje wykrywały tylko szablony obce, które stają się receptorami. Anomalie w programie wykrywane są przy pomocy wygenerowanych receptorów. Program jest zainfekowany, jeżeli między jego fragmentem a jednym z receptorów występuje określona liczba bitów pasujących.

Artykuł zorganizowany jest w następujący sposób. Proponowany system wykrywania intruzów omówiono w rozdziale 2. Metodę losową omówiono w rozdziale 3. Metodę szablonów omówiono w rozdziale 4. Obie metody przebadano eksperymentalnie i przedstawiono wyniki w rozdziale 5. Następnie dokonano analizy porównawczej obu metod w rozdziale 6, a w rozdziale 7 przedstawiono wnioski.

2. System wykrywania intruzów

Proponowany przez nas system wykrywania intruzów (SWI) może być wykorzystywany do wykrywania nieregularności w programach komputerowych w ich skompilowanej formie.

SWI monitoruje obszar w systemie operacyjnym podatny na infekcje. W przypadku systemu Microsoft Windows mógłby to być przykładowo katalog C:\Windows. Monitorowany obszar zawiera programy, które muszą być chronione przez SWI. W celu umożliwienia poprawnego działania SWI, programy w tym obszarze muszą inicjalnie istnieć w ich niezmienionej (niezainfekowanej) formie. Programy w tym obszarze muszą również być prawidłowymi plikami wykonywalnymi w formacie Win32 PE (Portable Executable). Typowy, prawidłowy plik PE zawiera sekcje które mają określone właściwości informujące o ich przeznaczeniu (sekcje kodu, zainicjalizowanych danych, niezainicjalizowanych danych) i uprawnieniach odczytu, zapisu i wykonania względem procesora głównego (CPU).

Proponowany system zawiera główny blok sterujący, który nadzoruje działanie bloków generacji receptorów i wykrywania anomalii. System skanuje pliki PE w monitorowanym obszarze w poszukiwaniu sekcji kodu, a następnie generuje specjalne ciągi binarne nazwane receptorami dla każdego pliku. Wygenerowane

receptory przechowywane są w osobnym zbiorze i wykorzystywane są do wykrywania anomalii w plikach w obszarze monitorowanym.

W SWI zaimplementowano dwie różne metody do generacji receptorów: metodę losową i metodę szablonów. Użytkownik systemu decyduje z jakiej metody ma korzystać system.

3. Metoda losowa

3.1. Generacja receptorów

W celu wykrywania nieregularności (anomalii) w kodzie programu, SWI musi skonstruować zbiór receptorów.

Receptorami nazywamy ciągi binarne o długości l . Jeśli ciągi te zostaną użyte w określonych wzorach, zyskują one zdolność do rozpoznawania obcych struktur w kodzie. Schemat blokowy generacji losowej receptorów przedstawiono na Rys. 1. Liczba receptorów do wygenerowania przez system określona jest przez użytkownika i oznaczona jest jako R_{max} . Maksymalna liczba receptorów, które mogą być wygenerowane przez system dla pojedynczego pliku jest zdefiniowana następującym równaniem:

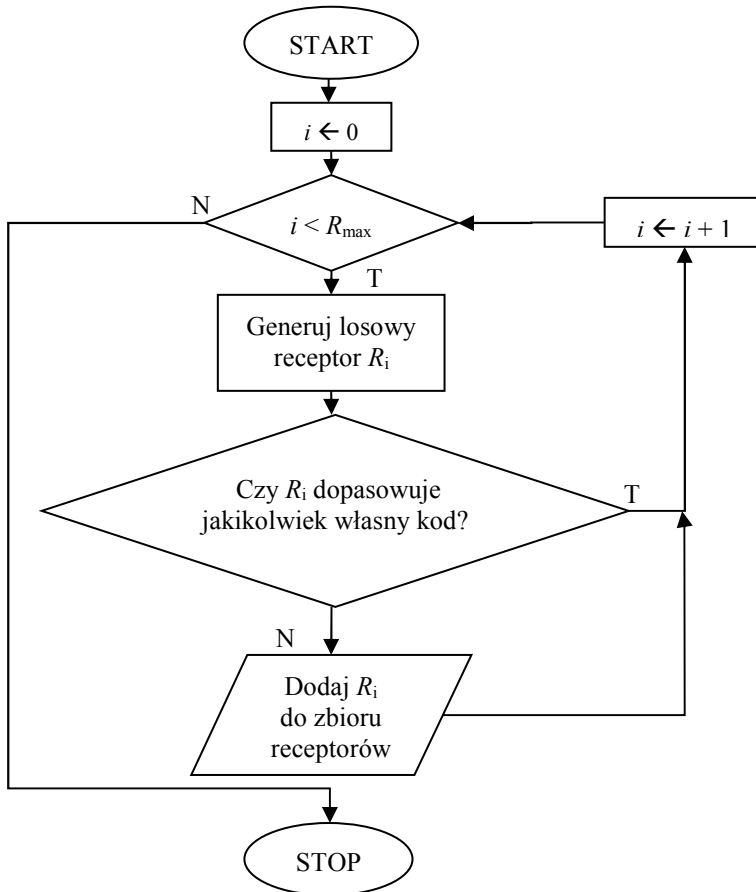
$$\max(R_{max}) = 2^l, \tag{1}$$

gdzie l oznacza długość receptora w bitach.

Metoda losowa zakłada losową generację receptorów. W algorytmie tym wykorzystano generator 8-bitowych liczb pseudolosowych do generacji receptorów o rozmiarze l . Generator wytwarza pseudolosowe liczby 8-bitowe z zakresu $[0; 255]$. Liczba 8-bitowych fragmentów receptora do losowej generacji oznaczona jest przez f i określona równaniem 2:

$$f = \text{ceil}(\text{div}(l, 8)), \tag{2}$$

gdzie l jest długością receptora w bitach, $\text{div}(a, b)$ jest wynikiem dzielenia a przez b i $\text{ceil}(a)$ jest funkcją zaokrąglającą liczbę a w górę. Po wygenerowaniu 8-bitowych fragmentów są one łączone ze sobą w jeden ciąg binarny na zasadzie konkatencji, tworząc losowy receptor.



Rys. 1. Schemat blokowy algorytmu losowej generacji receptorów

Jeśli l nie jest podzielne przez 8, w receptorze obecne są bity zbędne. Zbędne bity przechowywane są w pamięci, ale nie są dalej używane w algorytmach generacji i detekcji. Przykłady losowej generacji receptorów pokazane są na Rys. 2 i Rys. 3.

01001011	11011001	00000101	11101111
4B	D9	05	EF

Rys. 2. Przykład losowej generacji receptora dla $l = 28, f = 4$ (wyżej: reprezentacja binarna, niżej: reprezentacja szesnastkowa), bity niepogrubione są zbędne

11100001	10011010
E1	9A

Rys. 3. Przykład losowej generacji receptora dla $l = 16, f = 2$ (wyżej: reprezentacja binarna, niżej reprezentacja szesnastkowa), brak bitów zbędnych

Na powyższych rysunkach przedstawiono dwa przykłady losowej generacji receptorów. Na Rys. 2., dla $l = 28$ generator liczb pseudolosowych wygenerował 4 bajty: $(4B)_{16}$, $(D9)_{16}$, $(05)_{16}$ i $(EF)_{16}$. Bajty te zostały połączone ze sobą następnie w procesie konkatencji, co spowodowało utworzenie 32-bitowego receptora $(4BD905EF)_{16}$. Jednakże, ponieważ $l = 28$, cztery bity tego receptora nie będą dalej wykorzystane przez zaproponowane algorytmy. W przykładzie na Rys. 3. bity zbędne są nieobecne, ponieważ liczba $l = 16$ jest podzielna przez 8, toteż liczby 8-bitowe $(E1)_{16}$ i $(9A)_{16}$ utworzyły receptor $(E19A)_{16}$ i będzie on w całości dalej wykorzystany – jeżeli pomyślnie przejdzie testy dopasowania struktur własnych.

Cechą charakterystyczną podejścia inspirowanego sztucznymi systemami immunologicznymi, wykorzystującego receptory jest obecność progu aktywacji, oznaczonego m . Próg aktywacji jest minimalną liczbą kolejnych bitów, które muszą być dopasowane między receptorem a odczytanym fragmentem programu by receptor ten można było uznać za dopasowany (lub „aktywowany”). Jeżeli liczba dopasowanych kolejno bitów między receptorem a odczytanym fragmentem kodu jest mniejsza niż m , to receptor ten nie jest uznawany za dopasowany/aktywowany.

Ponieważ celem receptora jest wykrywanie struktur obcych, niezbędne jest sprawdzenie każdego losowo wygenerowanego receptora pod kątem wykrywalności kodu własnego. W tym celu algorytm otwiera plik PE przez interfejs systemu plików i lokalizuje informacje na temat sekcji kodu. Następnie system porównuje nowo wygenerowany receptor ze wszystkimi fragmentami kodu w sekcjach kodu w programie PE. By osiągnąć poprawne porównanie, algorytm odczytuje dane pliku fragmentami o długości l (takiej samej jak długość receptora), zaczynając od początku każdej sekcji kodu.

Odczytany fragment kodu	10101000	1101100100000101	01101100
Receptor	01001011	1101100100000101	11101111
Odczytany fragment kodu	A8	D9 05	6C
Receptor	4B	D9 05	EF

Rys. 4. Przykład receptora dopasowanego do odczytanego fragmentu programu dla $l = 32$, $m = 16$, receptor został dopasowany z przesunięciem okna $k = 8$

Kolejne m bitów nazywane jest oknem. Obecna pozycja okna względem receptora nazywana jest przesunięciem okna i oznaczona jest przez k . Jeżeli nowo wygenerowany receptor zostanie dopasowany w jakimkolwiek miejscu w programie (dopasowane zostanie co najmniej m bitów między receptorem a programem, z dowolnym przesunięciem okna k), to jest on odrzucany i nie może być wykorzystany do wykrywania anomalii, ponieważ wykrywa on również kod własny. W przypadku gdy receptor nigdy nie dopasowuje struktur własnych dodawany jest on do finalnego zbioru receptorów. Ostateczna liczba receptorów dodana do zbioru oznaczona jest przez R_n .

3.2. Wykrywanie anomalii

Proponowany system wykrywa anomalie w kodzie programów w obszarze monitorowanym przy użyciu receptorów wygenerowanych na etapie, gdy pliki były w ich zaufanym, inicjalnym stanie.

Zanim algorytm rozpocznie skanowanie programów PE w celu poszukiwania anomalii, zliczane są receptory obecne w zbiorze receptorów. Liczba ta oznaczana jest jako R_n . Dla każdego receptora obecnego w zbiorze, algorytm dokonuje odczytu fragmentów sekcji kodu o rozmiarze l i porównuje odczytane próbki z receptorem.

Jeżeli dopasowane zostanie m kolejnych bitów, receptor jest aktywowany i anomalia zostaje w ten sposób wykryta. Informacja o wykryciu anomalii jest przekazywana następnie użytkownikowi, wraz z informacją o lokalizacji wystąpienia anomalii.

4. Metoda szablonów

4.1. Generacja receptorów

Generacja receptorów metodą szablonów w proponowanym systemie opiera się na zmodyfikowanej metodzie opartej o szablony wprowadzonej w [14] i rozwiniętej w [15].

Szablonem nazywamy ciąg binarny zawierający bity istotne i nieistotne. Bity istotne w szablonach mają ustaloną wartość, a bity nieistotne oznaczane są przez gwiazdkę „*”.

W pierwszym kroku konstruowana jest tablica **T**, składająca się z szablonów o długości l bitów. Dla określonego progu aktywacji m w szablonach występuje dokładnie m ustalonych bitów. Pozostałe $l - m$ bitów w szablonie jest nieistotne.

Liczba wszystkich szablonów określona jest następującym wzorem:

$$L_s = (l - m + 1) \cdot 2^m. \tag{3}$$

Dla $l = 16$ i $m = 8$ oraz dla $l = 32$ i $m = 8$ liczby szablonów L_s wynoszą odpowiednio $9 \cdot 256$ i $25 \cdot 256$. Liczby te są zbyt wielkie do zilustrowania idei proponowanego algorytmu. W związku z tym w Tabeli 1 przedstawiono przykład tablicy szablonów **T** dla $l = 6$ i $m = 4$.

Tabela 1. Tablica szablonów **T** dla $l = 6$ i $m = 4$

i	m ustalonych bitów	$T[i,1]$	$T[i,2]$	$T[i,3]$
1	0000	0000**	*0000*	**0000
2	0001	0001**	*0001*	**0001
3	0010	0010**	*0010*	**0010
4	0011	0011**	*0011*	**0011
...				
14	1101	1101**	*1101*	**1101
15	1110	1110**	*1110*	**1110
16	1111	1111**	*1111*	**1111

Przyjęto, że przykładowy zbiór wzorców własnych (zaufanych fragmentów programu) **S** jest następujący:

$$S = \{101110, 101101, 101100, 101011, 101010\}. \tag{4}$$

Na podstawie tablicy \mathbf{T} i zbioru \mathbf{S} powstaje pochodna tablica \mathbf{T}_a . Tablica \mathbf{T}_a składa się z szablonów własnych i obcych. Jeżeli między szablonem w tablicy \mathbf{T} i co najmniej jednym wzorcem własnym ze zbioru \mathbf{S} co najmniej $m = 4$ kolejne bity są dopasowane, to szablon ten uznawany jest za szablon własny i jest oznaczany jako 0 w tablicy \mathbf{T}_a . W przeciwnym wypadku szablon jest uznawany jako obcy i oznaczany jako 1 w tablicy \mathbf{T}_a .

Przykładowo, szablon $\mathbf{T}[14,3] = **1101$ w tablicy \mathbf{T} jest dopasowany do szablonu $\mathbf{S}(2) = 101101$, więc $\mathbf{T}_a[14,3] = 0$, podczas gdy szablon $\mathbf{T}[2,3] = **0001$ w tablicy \mathbf{T} nie dopasowuje żadnego wzorca własnego ze zbioru \mathbf{S} , więc $\mathbf{T}_a[2,3] = 1$.

Tabela 2. Tablica \mathbf{T}_a dla $l = 6$ i $m = 4$

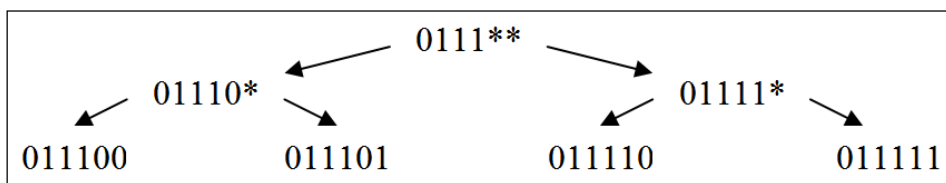
i	m ustalonych bitów	$T_a[i,1]$	$T_a[i,2]$	$T_a[i,3]$
1	0000	1	1	1
2	0001	1	1	1
3	0010	1	1	1
4	0011	1	1	1
5	0100	1	1	1
6	0101	1	0	1
7	0110	1	0	1
8	0111	1	0	1
9	1000	1	1	1
10	1001	1	1	1
11	1010	0	1	0
12	1011	0	1	0
13	1100	1	1	0
14	1101	1	1	0
15	1110	1	1	0
16	1111	1	1	1

Każdy szablon obcy reprezentuje w tym przypadku cztery możliwe receptory. Przykład generacji receptora z szablonu obcego $\mathbf{T}[8,1] = 0111**$ przedstawiono na Rys. 5. Przykład ten oparty jest na drzewie binarnym. Pierwsze dwa liście zawierają 4 bity ustalone szablonu, a ich pierwsze bity nieistotne zastąpione są zerem i jedyneką. W przykładzie otrzymano na tej podstawie liście $01110*$ i $01111*$.

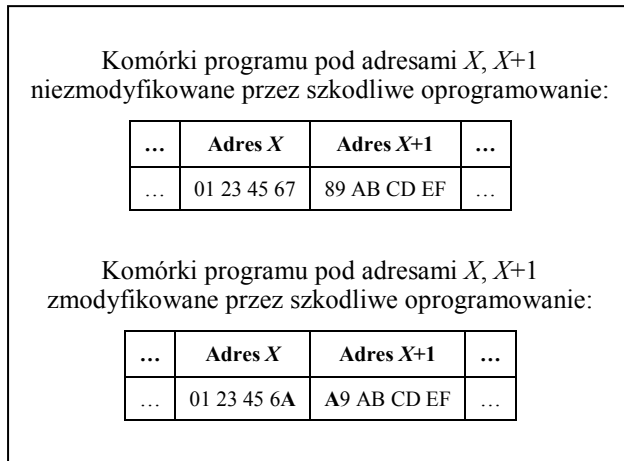
Ostatecznie, gdy pozostałe bity nieistotne „*” zastąpiono zerami i jedynkami, otrzymano cztery liście: 011100, 011101, 011110 i 011111. Liście te stają się receptorami, ponieważ na pewno nie dopasowują struktur własnych.

Na podstawie drzewa binarnego, cztery liście (receptory) szablonu obcego $T[11,2] = *1010*$ są następujące: 010100, 010101, 110100 i 110101. Dwa liście (receptory) 010100, 010101 są takie same jak liście szablonu obcego $T[6,1]$ i są zbędne. Całkowita liczba możliwych receptorów jest równa czterokrotności liczby szablonów obcych w T_a . W zbiorze receptorów otrzymanych na podstawie tablicy T_a występuje dużo receptorów identycznych, a więc zbędnych. Potrzebna jest zatem metoda eliminacji receptorów zbędnych. W tym celu drzewo binarne każdego szablonu obcego z kolumny $T_a[i,1]$ jest porównywane z drzewami binarnymi wszystkich szablonów obcych z kolumny $T_a[i,2]$ i ostatecznie z drzewami binarnymi wszystkich szablonów obcych z kolumny $T_a[i,3]$, eliminując zbędne gałęzie i liście. Po redukcji zbędnych gałęzi i liści zbiór liści staje się ostatecznym zbiorem receptorów R .

Dla zbioru wzorców własnych S określonego w (4), zbiór receptorów $R = \{000000, 110001, 000010, 110011, 000100, 110101, 000110, 110111, 001000, 111001, 010000, 010001, 010010, 010011, 010100, 010101, 010110, 010111, 011000, 011001, 111111\}$. Wszystkie 21 receptorów ze zbioru R wykorzystane zostanie do wykrywania anomalii.



Rys. 5. Generacja receptora na podstawie szablonu $T[8,1] = 0111**$



Rys. 6. Przykład jednobajtowej anomalii umieszczonej między 32-bitowymi komórkami

Wykorzystując metodę szablonów, system generuje receptory o długości $l = 32$ bitów z progiem aktywacji $m = 16$ bitów. Rozmiar odczytywanych komórek programu wynosi zatem 4 bajty. Metoda ta, wykorzystana w systemie z wymienionymi parametrami, nosi nazwę metody S-32 ze względu na liczbę bitów receptorów ze zbioru \mathbf{R} . Zbiór receptorów ten nadaje się do wykrywania anomalii występujących w obrębie pojedynczej odczytanej czterobajtowej komórki, ale nie nadaje się do wykrywania anomalii które wystąpiły między komórkami (rozpoczynającymi się na przykład w najmłodszej bajcie komórki o adresie X , a kończącymi w najstarszej bajcie komórki o adresie $X+1$ – ilustruje to Rys. 6). Proponowany system zawiera więc także zmodyfikowaną wersję metody szablonów, która generuje dodatkowy zbiór receptorów \mathbf{R}_{MK} . Zbiór \mathbf{R}_{MK} zawiera receptory o długości $l = 16$ bitów z progiem aktywacji $m = 8$ bitów wygenerowane na podstawie par kolejnych sąsiadujących ze sobą czterobajtowych komórek programu. Przy generacji tego zbioru pod uwagę brane są tylko najmłodszy bajt komórki i najstarszy bajt komórki po niej następującej. Modyfikacja metody tej nosi nazwę metody RMK (receptorów międzykomórkowych).

Należy zauważyć, że przykładowo liście drzew binarnych szablonów, które zostały odrzucone jako potencjalne receptory ze względu na dopasowanie struktury własnej na końcu programu, mogłyby nadal służyć jako receptory do wykrywania anomalii na początku programu. W związku z tym proponowany system zawiera również metodę sekcji (MS). Metoda sekcji dzieli sprawdzany program na cztery równe części (sekcje) i traktuje je jak osobne programy, co powoduje że receptory z sekcji pierwszej nie kolidują z kodem własnym z innych sekcji. W metodzie tej wykorzystano metodę szablonów 32-bitowych.

4.2. Wykrywanie anomalii

Proponowany system wykrywa anomalie w programach w obszarze chronionym korzystając z wygenerowanego wcześniej zbioru receptorów 32-bitowych \mathbf{R} i opcjonalnie również zbioru receptorów 16-bitowych \mathbf{R}_{MK} . W przeciwieństwie do etapu generacji receptorów, na etapie wykrywania anomalii weryfikowany program odczytywany jest w jego aktualnym (obecnym) stanie. Pierwsza 32-bitowa komórka programu odczytywana jest przez system i porównywana następnie z każdym 32-bitowym receptorem ze zbioru \mathbf{R} , korzystając z progu aktywacji $m = 16$. Za każdym razem, gdy między odczytaną komórką a receptorem dopasowane zostanie m kolejnych identycznych bitów z tym samym przesunięciem okna k , wykryta zostaje anomalia i zostaje ona zgłoszona użytkownikowi. Jeżeli żaden z receptorów nie jest aktywowany, odczytywana jest kolejna 4-bajtowa komórka pamięci i analogicznie porównywana jest ze wszystkimi receptorami. Proces powtarza się do momentu porównania wszystkich komórek programu ze wszystkimi receptorami.

5. Badania eksperymentalne

Proponowany system został zaimplementowany w języku C# w celach przeprowadzenia badań skuteczności opisanych metod.

Jako przykładowy plik do ochrony wykorzystano napisany w języku C++ program w formacie Win32 PE o nazwie „test.exe” i rozmiarze 6584 bajtów. Badanymi metodami były metoda losowa (L), metoda szablonów 32-bitowych (S-32), metoda szablonów 32-bitowych i 16-bitowych (RMK) i metoda sekcji (MS). Parametrami dla metod: losowej i S-32 były parametry $l = 32$, $m = 16$, dla metody RMK $l = 32$, $m = 16$ dla receptorów głównych w zbiorze \mathbf{R} oraz $l = 16$, $m = 8$ dla receptorów pomocniczych w zbiorze \mathbf{R}_{MK} , i $l = 32$, $m = 8$ dla receptorów w metodzie sekcji, przy czym liczba sekcji jest równa 4. Metody sprawdzono dla losowo wprowadzonych anomalii o rozmiarach z zakresu [1 B; 8 B], przy czym dla każdego rozmiaru anomalii przeprowadzono 100 prób wykrycia. Wyniki badań przedstawiono w poniższych tabelach. Oznaczenia w tabelach są następujące: NRA – największa liczba aktywowanych receptorów dla 100 prób, CW_{avg} – średni czas wykrycia anomalii w milisekundach, W – wykrywalność anomalii w procentach.

Tabela 3. Metoda losowa, $R_{max} = 1000$, $R_n = 927$

<i>Rozmiar anomalii [B]</i>	<i>NRA</i>	<i>CWavg</i>	<i>W</i>
1	1	1046	10,00%
2	1	1030	20,00%
3	1	1041	40,00%
4	1	1020	20,00%
5	1	1025	30,00%
6	3	1037	20,00%
7	2	1050	20,00%
8	2	1045	20,00%
Średnia	1,5	1036,75	22,50%

Tabela 4. Metoda losowa, $R_{max} = 5000$, $R_n = 4693$

<i>Rozmiar anomalii [B]</i>	<i>NRA</i>	<i>CWavg</i>	<i>W</i>
1	1	5396	30,00%
2	2	5321	50,00%
3	2	5313	40,00%
4	3	5311	70,00%
5	4	5298	90,00%
6	4	5264	60,00%
7	4	5348	80,00%
8	5	5211	60,00%
Średnia	3,125	5307,75	60,00%

Tabela 5. Metoda losowa, $R_{max} = 10000$, $R_n = 9427$

<i>Rozmiar anomalii [B]</i>	<i>NRA</i>	<i>CWavg</i>	<i>W</i>
1	2	10072	40,00%
2	3	10017	40,00%
3	2	10020	90,00%
4	3	10072	60,00%
5	4	10063	90,00%
6	3	9952	100,00%
7	3	10216	100,00%
8	5	9995	100,00%
Średnia	3,125	10050,875	77,50%

Tabela 6. Metoda S-32, $R_n = 32$

<i>Rozmiar anomalii [B]</i>	<i>NRA</i>	<i>CWavg</i>	<i>W</i>
1	4	12	30,00%
2	3	22	50,00%
3	8	8	20,00%
4	2	13	30,00%
5	14	21	50,00%
6	6	26	60,00%
7	6	26	60,00%
8	8	35	80,00%
Średnia	6,375	20,375	47,50%

Tabela 7. Metoda RMK, $R_n = 46$

Rozmiar anomalii [B]	NRA	CWavg	W
1	4	12	30,00%
2	3	22	50,00%
3	11	8	40,00%
4	11	13	70,00%
5	14	21	50,00%
6	6	26	60,00%
7	6	26	60,00%
8	11	8	100,00%
Średnia	8,25	17	57,50%

Tabela 8. Metoda sekcji, $R_n = 780$

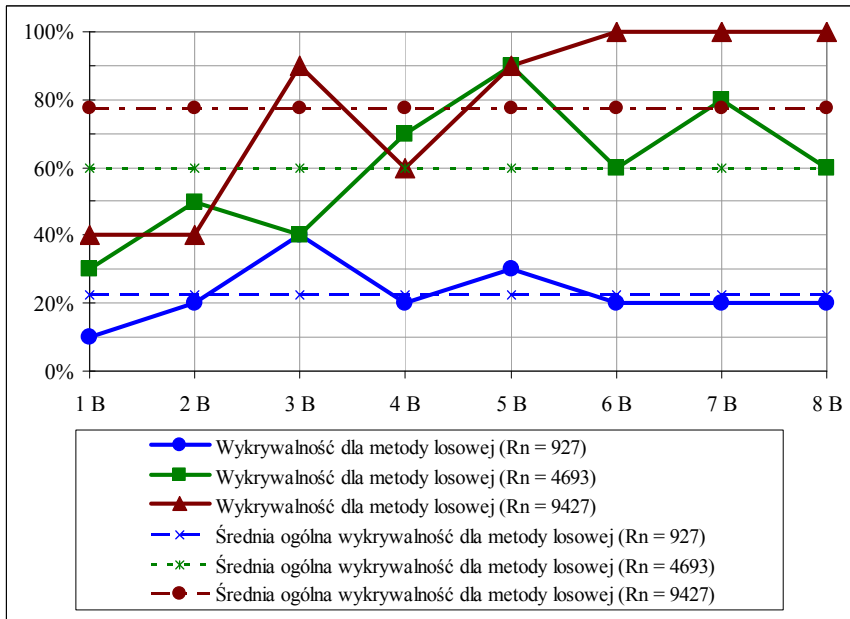
Rozmiar anomalii [B]	NRA	CWavg	W
1	5	10	60,00%
2	7	17	100,00%
3	9	16	90,00%
4	11	20	100,00%
5	13	20	100,00%
6	15	17	100,00%
7	24	16	100,00%
8	20	19	100,00%
Średnia	13	16,875	93,75%

Ponieważ generacja receptorów może odbyć się tylko dla niezmodyfikowanego programu, algorytm generacji musiał wykonać się tylko raz dla metod losowej, S-32 i RMK oraz 4 razy dla metody sekcji. Czas generacji receptorów dla metody losowej i $R_{max} = 1000$ wyniósł 1 s, dla $R_{max} = 5000$ wyniósł 5 s, a dla $R_{max} = 10000$ wyniósł 10 s. Czas generacji receptorów dla metody S-32 wyniósł 140 s, dla metody RMK 175 s, a dla metody sekcji 290 s.

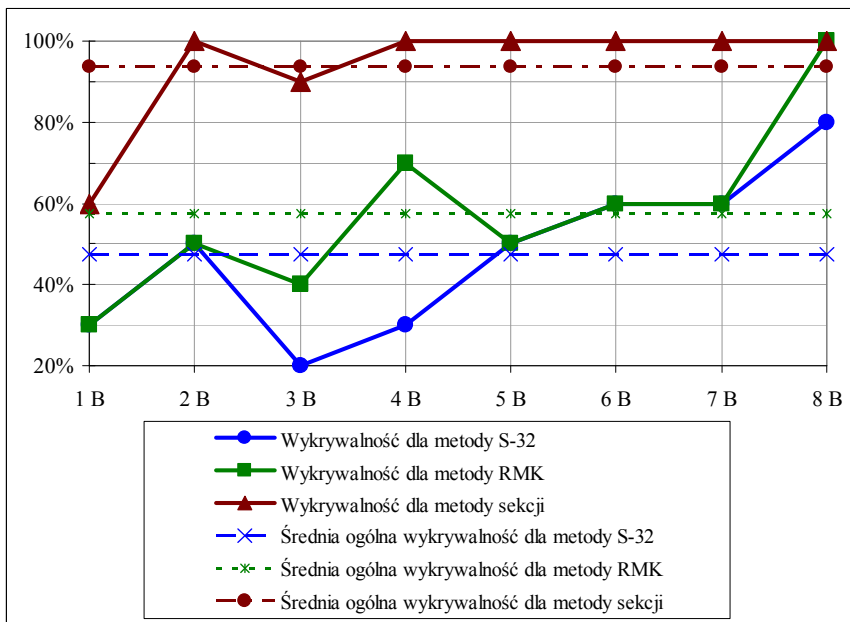
6. Analiza porównawcza

Dla sprawdzanego pliku „test.exe” system generował receptory metodą losową w tempie 1003,1 receptorów na sekundę, podczas gdy generacja metodą S-32 odbywała się w tempie 0,23 receptora na sekundę. Mimo większej liczby receptorów gotowych do wykrycia anomalii metoda losowa miała dla $R_{max} = 1000$ niższą wykrywalność na poziomie 22,5% w porównaniu do metody szablonów 32-bitowych, która uzyskiwała średnią wykrywalność na poziomie 47,5%. Zwiększenie liczby maksymalnej receptorów dla metody losowej do $R_{max} = 5000$ podniosło wykrywalność metody losowej do 60%, jednak spowodowało to również zwiększenie zajętości pamięci operacyjnej systemu ze względu na konieczność przechowania większej liczby receptorów. Implementacja 16-bitowych receptorów pomocniczych w proponowanej metodzie RMK podniosła średnią wykrywalność anomalii w programie o 10 punktów procentowych (uzyskując wynik 57,5%) i obniżyła średni czas wykrycia anomalii z 20,4 do 17 ms. Zajętość pamięci w tym przypadku zwiększyła się o dodatkowe 14 receptorów 16-bitowych w zbiorze **RMK**. Można zauważyć różnicę w zajętości pamięci między metodą losową a metodą RMK. Metoda losowa dla $R_{max} = 5000$ wykorzystwała 4693 czterobajtowe komórki pamięci operacyjnej, a metoda RMK wykorzystwała jedynie 46 takich komórek. Średni czas wykrycia anomalii dla metody losowej wyniósł 5307,75 ms, a dla metody RMK wyniósł 17 ms. Czasy generacji receptorów dla metody losowej i metody RMK wyniosły odpowiednio 5 s i 120 s. Średnia wykrywalność tych metod dla zadanych parametrów była porównywalna. Można zauważyć, że metoda losowa bardzo szybko generuje potrzebne receptory, jednak jest ich więcej i zajmują one więcej pamięci niż w innych metodach. Powoduje to, że średni czas wykrycia anomalii jest większy niż w innych metodach, gdyż receptorów do sprawdzenia jest więcej. Zwiększając liczbę maksymalną receptorów do $R_{max} = 10000$ uzyskano średnią wykrywalność na poziomie 77,5%. Dzieliąc plik na cztery fragmenty, uzyskano w metodzie sekcji wykrywalność na poziomie 93,75%. Liczba receptorów wzrosła wtedy jednak z 32 do 780, co zwiększyło zajętość pamięci i wydłużyło czas generacji receptorów o 150 s. Można zauważyć, że dla $R_{max} = 1000$ w metodzie losowej rozmiar anomalii przestaje mieć wpływ na wykrywalność już powyżej 3 B, gdyż oscyluje ona w okolicach 30% mimo coraz większej liczby bajtów anomalii. W porównaniu, wykrywalność metody S-32 rosła w miarę wzrostu rozmiaru anomalii – dla 4 B wykrywalność wyniosła 30%, a dla 8 B już 60%. Rozmiar anomalii ma więc mniejsze znaczenie w metodzie losowej niż w metodach wykorzystujących szablon.

Wykresy wykrywalności metod w funkcji rozmiaru anomalii przedstawiono na Rys. 7 i Rys. 8.



Rys. 7. Wykrywalność dla metody losowej w funkcji rozmiaru anomalii



Rys. 8. Wykrywalność metod opartych na szablonach w funkcji rozmiaru anomalii

7. Wnioski

Proponowany system wykorzystuje algorytmy sztucznych systemów immunologicznych w celu wykrycia anomalii w programie. Algorytm negatywnej selekcji zaimplementowano w dwóch różnych metodach: losowej i szablonów. Implementację algorytmów przetestowano poprzez serię badań eksperymentalnych, a następnie badania te porównano. Badania wskazują, że metoda losowa jest bardzo szybka w generowaniu receptorów (trwało to maksymalnie 5 s), w porównaniu z metodą szablonów w której czas generacji receptorów wyniósł do 290 s. Wyższy czas generacji receptorów w metodach szablonowych spowodował jednak wzrost jakości samych receptorów. Duża liczba receptorów w przypadku metody losowej powodowała również wysoką w porównaniu z metodami szablonowymi zajętość pamięci operacyjnej i dłuższy czas wykrycia anomalii.

W przypadku gdy minimalizacja zajętości pamięci nie ma priorytetu nad potrzebą szybkiego rozpoczęcia pracy systemu, metoda losowa może okazać się odpowiednia. Jeżeli istotny jest natomiast czas wykrywania anomalii, odpowiedniejsza w użyciu może być metoda szablonów. Najskuteczniejsza pod względem wykrywalności dla zadanych parametrów okazała się metoda selekcji, która osiągnęła poziom 93,75%. Dalsze prace nad badaniami mogą uwzględniać na przykład zachowanie współczynnika wykrywalności w zależności od liczby bitów receptora i progu aktywacji.

Literatura

1. Somayaji A., Forrest S., Hofmeyr S., Longstaff T., *A sense of self for unix processes*, w: *IEEE Symposium on Security and Privacy*, 1996, s. 120-128.
2. Somayaji A., Hofmeyr S., Forrest S., *Principles of a computer immune system*, w: *New Security Workshop*, Langdale, Cumbria 1997, s. 75-82.
3. Forrest S., Perelson A.S., Allen L., Cherukuri R., *Self-nonsel self discrimination in a computer*, w: *IEEE Symposium on Security and Privacy*, IEEE Computer Society, 1994, No 202.
4. Kephart J., *A biologically inspired immune system for computers*, w: *Fourth International Workshop on Synthesis and Simulation of Living Systems, Artificial Life IV*, 1994, s. 130-139.
5. Dasgupta D., *Immunity-based intrusion detection systems: a general framework*, w: *22nd National Information Systems Security Conference (NISSC)*, 1999.
6. Andrews P.S., Timmis J., *Tunable detectors for artificial immune systems: from model to algorithm*, w: *Bioinformatics for Immunomics*, Springer, New York, NY, USA 2010, vol. 3, s. 103-127.
7. Sobh T.S., Mostafa W.M., *A cooperative immunological approach for detecting network anomaly*, w: *J Applied Soft Computing*, 2011, vol. 11(1), s. 1275-1283.

8. Wang D., Zhang F., Xi L., *Evolving boundary detector for anomaly detection*, w: *Expert Systems with Applications*, 2011, vol. 38(3), s. 2412-2420.
9. Powers S.T., He J., *A hybrid artificial immune system and self organizing map for network intrusion detection*, w: *Information Sciences*, 2008, vol. 78(15), s. 3024-3042.
10. Li G.Y., Guo T., *Receptor editing-inspired real negative selection algorithm*, w: *Computer Science*, 2012, vol. 39, s. 246–251.
11. Laurentys C.A., Ronacher G., Palhares R.M., Caminhas W.M., *Design of an artificial immune system for fault detection: a negative selection approach*, w: *Expert Systems with Applications*, 2010, vol. 37(7), s. 5507–5513.
12. Fanelli R., *A hybrid model for immune inspired network intrusion detection*, Springer-Verlag, Phuket, Thailand 2008.
13. Mostardinha P., Faria B.F., Zúquete A., Vistulo de Abreu F., *A negative selection approach to intrusion detection*, w: Coello, C.A.C., Greensmith, J., Krasnogor, N., Liò, P., Nicosia, G., Pavone, M., *Artificial Immune Systems, Lecture Notes in Computer Science*, 2012, vol. 7597, s. 178-190.
14. Wierchoń S.T., *Generating optimal repertoire of antibody strings in an artificial immune system*, *Intelligent Information Systems*, 2000, s. 119-133.

Streszczenie

Ochrona systemu operacyjnego przed infekcjami wirusowymi jest zagadnieniem, nad którym od kilku dekad pracują projektanci oprogramowania antywirusowego. Rosnąca w ostatnich latach złożoność szkodliwego oprogramowania skłoniła naukowców do poszukiwania inspiracji w rozwiązaniach naturalnych, takich jak układ immunologiczny ssaków. W artykule przedstawiono system wykrywania intruzów w systemie operacyjnym wykorzystujący algorytm negatywnej selekcji. Algorytm ten wykorzystuje ciągi binarne zwane receptorami do wykrywania zmian w chronionych programach. W systemie zaimplementowano dwie metody generacji receptorów: metodę losową i metodę szablonów. Metody te zostały przetestowane eksperymentalnie. Wyniki działania metod przeanalizowano i porównano, a następnie wyciągnięto wnioski.

Abstract

Protection of the operating system against virus infections is an area of research which has been worked on by antivirus software designers since several decades. Increasing malware complexity led scientists to seek inspiration in natural solutions, such as the mammal immune system. In the article, an intrusion detection system has been proposed. The system's inner workings are based on the negative selection

algorithm. The algorithm uses binary strings called receptors to detect modifications in the protected programs. In the system, two receptor generation methods have been presented: the random generation method and the template generation method. The methods have been tested experimentally. The results of both methods have been analysed and compared, and conclusions have been drawn.

Keywords: intrusion detection systems, artificial immune systems, viruses, malware, negative selection algorithm, receptor generation, anomaly, anomaly detection.

Anita Wąchała

Karol Hliwa

Antoni Wiliński*

Wydział Elektroniki i Informatyki

Politechnika Koszalińska

* awilin@o2.pl

Badania profili inteligencji wielorakich wśród studentów pierwszego roku informatyki

Słowa kluczowe: Inteligencje wielorakie, edukacja inżynierska, osobowość studenta, profil inteligencji

Wstęp

Inteligencje wielorakie (ang. *multiply intelligences*) zajmowały prof. Howarda Gardnera już w latach 80. dwudziestego wieku. W największym skrócie, bowiem inteligencje te będą omawiane dalej, teoria ta zakłada występowanie u każdej istoty ludzkiej wielu równoprawnych inteligencji o zróżnicowanej sile (intensywności) tworząc specyficzny, niepowtarzalny profil, zmienny w czasie, stanowiący potencjał rozwojowy zwłaszcza dla dziecka. Gardner po przeprowadzeniu licznych badań m.in. [1] opublikował swoje przemyślenia w książce [2], także po polsku [3], a następnie, po dwudziestu latach ocenił ich rozwój i wpływ na praktykę wychowania [4]. Prace Howarda Gardnera wywołały duży oddźwięk w świecie nauki, w tym uznanie i krytykę [5, 6, 7]. Pojawiły się propozycje rozszerzenia liczby rozważanych inteligencji z początkowej siedmiu lub ośmiu u Gardnera do większej obejmującej inteligencje np. duchową, emocjonalną, seksualną czy cyfrową. Zwłaszcza ta ostatnia winna być przedmiotem zainteresowania w środowisku autorów i tytułowej populacji studentów.

Teoria Howarda Gardnera o inteligencjach wielorakich bazuje na kilku podstawowych tezach sformułowanych przez jej autora [8]:.

- 1) Wszyscy operujemy pełnym zestawem (minimalnie) 8 inteligencji – czyni to nas ludźmi,

- 2) Każdy człowiek ma inny profil inteligencji (u każdego człowieka inteligencje są rozwinięte w różny sposób) – nawet bliźniaki mają różne profile.

Odzwierciedleniem wizji Howarda Gardnera może być podana przez niego metafora porównująca mózg człowieka do zbioru komputerów. Można wyobrazić sobie mózg jako sieć komputerów stosunkowo niezależnych i wyspecjalizowanych w rozwiązywaniu specyficznych zadań.

Te „komputery” odpowiadają za zadania z różnych obszarów aktywności człowieka, takich jak (kolejność niżej wymienionych inteligencji nie jest narzucona, należy pamiętać, że są równoprawne):

- 1– przyrodnicza;
- 2– matematyczno-logiczna;
- 3– językowa;
- 4 – muzyczna;
- 5 – wizualna;
- 6 – ruchowa;
- 7 – interpersonalna;
- 8 – intrapersonalna;

To oczywiście gardnerowskie inteligencje, które bardziej szczegółowo można opisać jak poniżej.

Inteligencja przyrodnicza

Ten typ inteligencji opiera się na poznaniu, docenianiu i zrozumieniu natury. Osoba obdarzona inteligencją przyrodniczą „czuje” naturę, troszczy się o świat, dba o zwierzęta, pielęgnuje rośliny. Odznacza się zdolnością do rozumienia i czerpania wniosków i korzyści ze środowiska naturalnego. Typowe profesje charakterystyczne dla tego typu inteligencji to np. kynolog, botanik, weterynarz czy rolnik. Dzieci z dobrze rozwiniętą inteligencją przyrodniczą świetnie radzą sobie w klasyfikacji różnych przedmiotów w hierarchie (stopniują rzeczy ważniejsze i mniej ważne), dostrzeganiu wzorców funkcjonujących w naturze. Dzieci – przyrodnicy aktywnie spędzają czas na świeżym powietrzu, lubią wycieczki, zbieranie grzybów, kolekcjonowanie liści [9].

Inteligencja matematyczno-logiczna

Druga inteligencja tak bardzo obecna we współczesnych szkołach i tak ważna nie tylko w edukacji, ale i w ‘dorosłym życiu’. Jej znaczenie dla rozwoju społeczeństw jest nie do przecenienia. Do dziś za duży błąd uważa się okres rezygnacji z obowiązkowych matur z matematyki.

To rodzaj inteligencji oparty na myśleniu przyczynowo-skutkowym. Przejawia się w postrzeganiu świata poprzez ciągi zdarzeń, logiczne myślenie. Dotychczasowe badania miernika IQ dotyczyły właśnie tego rodzaju inteligencji [9].

Osoby o tym typie inteligencji korzystają z logiki, liczb, schematów, łatwo dostrzegają relacje i połączenia pomiędzy informacjami. Rzeczy, które nie mają sensu logicznego są dla nich bezwartościowe. Mają skłonność do myślenia abstrakcyjnego oraz koncepcyjnego. Są dociekliwi, systematyczni, dokładni. Łatwo myślą i zachowują się w sposób uporządkowany, algorytmiczny. W trakcie zajęć szkolnych lubią zadawać pytania. Są dobrze zorganizowani i posługują się argumentami logicznymi. Te umiejętności są niezwykle ważne dla matematyków, informatyków, bankowców, fizyków, chemików, lekarzy, inżynierów [10].

Inteligencja językowa

To inaczej umiejętność swobodnego korzystania ze słów oraz języka. Osoby o inteligencji werbalnej starannie dobierają słowa, gdyż potrafią wychwycić subtelne różnice znaczeniowe między słowami, mają poczucie rytmu oraz dźwięku słów. Mają bogate słownictwo, chętnie używają synonimów. Słowa wykorzystują w celach rozrywkowych, informacyjnych czy perswazyjnych. Lubią literaturę, zabawę słowami, chętnie uczestniczą w debatach, dobrze radzą sobie z pracami pisemnymi. Łatwiej niż inni uczą się kolejnego języka obcego. W związku z tymi umiejętnościami dobrze sprawdzają się w zawodzie: pisarza, dziennikarza, publicysty, prawnika, nauczyciela, tłumacza [10].

Inteligencja muzyczna

Ten rodzaj inteligencji pojawia się najwcześniej. Dzieci muzyczne chętnie śpiewają, nucą, grają na czym się da, po prostu otaczają się muzyką i dźwiękami. Inteligencję muzyczną łatwo rozwijać, łącząc z muzyką codzienne czynności: słuchać muzyki podczas wykonywania prac domowych, śpiewać, nucić, rymować, zabierać dziecko na koncerty i spacerować po parku, zachęcać do występów publicznych i komponowania. Można zapisać pocichę na zajęcia z rytmiki, śpiewu czy gry na instrumencie. W dorosłym życiu ta inteligencja ujawnia się umiłowaniem muzyki, poszukiwaniem muzyki dobrej, ulubionej, także umiejętnością gry na instrumentach muzycznych. Zawody związane z tą inteligencją są dość oczywiste [9].

Inteligencja wizualna

Ten rodzaj inteligencji umożliwia rozumienie otoczenia dzięki kształtom i wyobrażeniom pochodzącym ze świata zewnętrznego oraz wyobraźni. Osoba obdarzona tym typem inteligencji „myśli obrazami”, czyli używa wyobraźni. Dziecko zwraca uwagę na szczegóły, detale, wszystko jest dla niego ważne. Cechuje je duża wrażliwość na otaczające przedmioty, kolory i wzory. Lubi wykonywać

różnego rodzaju prace plastyczne, układać puzzle, czytać mapy, wie jak łączyć ze sobą harmonijnie kolory. Jest obserwatorem, bardzo często używa wyobraźni – wizualizacji, chętnie obrazuje zagadnienia w postaci diagramów, schematów i tabel. W proces zapamiętywania angażuje wszystkie zmysły. Dziecko z rozwiniętą inteligencją przestrzenną jest bardzo kreatywne i twórcze. Dzieci o rozwiniętej inteligencji wizualno-przestrzennej mają bogatą wyobraźnię, preferują książki z wieloma ilustracjami, świetnie radzą sobie z układankami, klockami i zabawkami konstrukcyjnymi [9]. W życiu dorosłym ten rodzaj inteligencji przydatny jest w wielu zawodach związanych z wyobraźnią i przestrzenią. To może być inżynier dróg i mostów, urbanista, grafik komputerowy, filmowiec ale także poeta, przyrodnik, kartograf.

Inteligencja ruchowa

Dzieci obdarzone inteligencją ruchową uwielbiają taniec i sport. Z przyjemnością wykonują własnoręcznie różnego typu prace, np. majsterkowanie, rzeźbienie. Na co dzień posługują się językiem ciała i dużo gestykują. Potrafią, bez wykorzystania logicznego i świadomego myślenia wykonywać trudne manewry, zarówno własnym ciałem jak i innymi przedmiotami. Mają doskonale wypracowane reakcje automatyczne. Potrafią planować działania, dzielić zadania, mają świetną organizację przestrzenną. Dzieci z rozwiniętym tym typem inteligencji przepadają za grami ruchowymi, najlepiej grupowymi. Dorośli mylnie nazywają je „nadpobudliwymi ruchowo”. Dzieci takie są bardzo wrażliwe na dotyk.

Dzieci w wieku szkolnym z rozwiniętą inteligencją ruchową często mają problemy w szkole, ponieważ oczekuje się od nich skupienia i ciszy, czyli po prostu siedzenia w ławce przez 45 minut. Należy zapewnić im maksymalnie dużo przerw. Zajęcia dla takich dzieci powinny być bardzo urozmaicone, ciekawe, z zastosowaniem różnego typu urządzeń (np. mikroskopy, rzutniki, tablice interaktywne). Najłatwiej zapamiętują to, co było wykonywane, trudniej to, co było wyłącznie omówione [9]. W życiu dorosłym ten rodzaj inteligencji cechuje przede wszystkim sportowców, u których w dzieciństwie zdiagnozowano te zalety. To także inteligencja rzemieślników uprawiających ruch i rzeźbiarzy.

Inteligencja interpersonalna

Osoby o tym typie inteligencji po prostu muszą przebywać wśród ludzi. Uczą się przez kontakty międzyludzkie. Potrafią być doskonałymi słuchaczami i doradcami. Mają szerokie zainteresowania i często uczęszczają na wiele dodatkowych zajęć. Opinie krytyczne tylko dodatkowo motywują je do działania.

Dzieci z dobrze rozwiniętą inteligencją interpersonalną są asertywne, komunikatywne, łatwo nawiązują i utrzymują kontakty społeczne, potrafią współpracować, mają zdolności przywódcze i umiejętności mediacyjne. Cechuje je

łatwość poznawania, rozumienia myśli, uczuć, poglądów i zachowań innych ludzi, są tolerancyjne. [9]. W życiu dorosłym to inteligencja dominująca dla polityków, wyższych urzędników, nauczycieli, przewodników turystycznych, prawników i księży.

Inteligencja intrapersonalna

Osoby z rozwiniętą inteligencją intrapersonalną posiadają tzw. „mądrość życiową”, intuicję, wewnętrzną motywację i silną wolę do działania. Takie osoby są nieco skryte, wolą pracę w samotności, bywają nieśmiałe.

Dzieci o mocnej inteligencji intrapersonalnej są indywidualistami, są odpowiedzialne, znają swoje mocne strony, budują wewnętrzną motywację. Nie boją się trudnych pytań i chętnie podejmują ryzyko. Lubią znać opinię innych, którą potem analizują, poszerzając swoje horyzonty myślenia. Nie porzucają swoich ambicji i nie rozpamiętują porażek i słabości.

Najlepszym sposobem rozwoju dziecka posiadającego dobrze rozwiniętą inteligencję intrapersonalną, jest stworzenie mu optymalnych warunków, w których samo decydowało o zakresie pracy, tempie jej wykonywania. Takie dziecko musi mieć możliwość doświadczenia samodzielnej nauki. Poszukuje „swojego” miejsca [9]. Dorosły o takiej inteligencji to często typ badacza, myśliciela, filozofa, pisarza ale także pracującego samotnie informatyka, poety, przyrodnika, podróżnika.

Dla zweryfikowania popularności idei prof. Howarda Gardnera w polskim środowisku edukacyjnym, także w Internecie autorzy zapoznali się z kilkunastoma witrynami pojawiającymi się po wpisaniu po polsku słów kluczowych "inteligencje wielorakie" spośród których za najważniejsze uznano:

- 1) Witrynę Oskara Rożewicza [8] dotyczącą inteligencji wielorakiej i jej wpływu na edukację; autor w swoim artykule odnosi się do teorii Howarda Gardnera o Inteligencjach Wielorakich, udowadnia, iż zdania po kilkudziesięciu latach na temat odkrycia Gardnera są podzielone – jedni w nią wierzą a drudzy uważają, że wnioski wysunięte przez Gardnera są zbyt dalekosiężne; Oskar Rożewicz [8] zalicza się zdecydowanie do grupy osób, którzy wierzą w teorię Howarda Gardnera i w swoim artykule przedstawia o czym jest ta teoria i jaki ma wpływ na metodykę nauczania wszystkich przedmiotów oraz wyjaśnia czym jest Inteligencja Wieloraka – według Gardnera Inteligencja Wieloraka to biopsychologiczny potencjał do przetwarzania informacji na określone sposoby w celu rozwiązywania problemów lub modelowania produktów cenionych w kulturze lub społeczeństwa. Oskar Rożewicz w swoim artykule odnosi się do tego o czym mówi teoria Inteligencji Wielorakiej. Autor tego artykułu bardzo dokładnie omawia teorię Gardnera i charakteryzuje typy inteligencji jakie są

obecne u każdego z nas – Gardner wymienił aż 8 typów inteligencji, do których zaliczył inteligencję przyrodniczą, matematyczno-logiczną, ruchową, językową, wizualno-przestrzenną, muzyczną oraz interpersonalną i interpersonalną. Rożewicz w swoim artykule bardzo szczegółowo przedstawił typy inteligencji wg Gardnera opierając je na przykładach. Po przeanalizowaniu artykułu Rożewicza dochodzimy do wniosku, że nauczyciele w swojej pracy powinni angażować w swoich uczniach różne typy inteligencji, gdyż wtedy efekty kształcenia są bardziej miarodajne i wiarygodne.

- 2) Test inteligencji wielorakiej autorstwa Damiana Chodorka [11], który był przeznaczony dla osób dorosłych powyżej 18 roku życia i zawierał 90 pytań, których zadaniem było wybadanie jakie są predyspozycje typów inteligencji u osób badanych. Dzięki temu testowi osoby badane miały się dowiedzieć, które z 8 inteligencji u nich dominują, a które wymagają pracy oraz rozwoju. Test miał także na celu uświadomienia osobom badanym jaki sposób życia, funkcjonowania oraz uczenia się będzie dla nich prawdopodobnie najlepszy i najbardziej zadawalający. Autor tego tekstu udowadnia, że test powinien zrobić każdy dorosły, by sprawdzić swój współczynnik IQ. Damian Chodorek swój test oparł na teorii Howarda Gardnera.
- 3) Na stronie internetowej Edyty Litwin [12] zostały przedstawione i opisane główne założenia teorii Howarda Gardnera, które mówią o tym, że każdy człowiek posiada wszystkie rodzaje inteligencji i są one rozwinięte w różnym stopniu, inteligencje tworzą profil niepowtarzalny dla innych, profile te są dynamiczne i zmieniają się w trakcie rozwoju człowieka, wszystkie inteligencje współpracują ze sobą w różnych konfiguracjach, inteligencję można rozwijać poprzez różnorodne ćwiczenia oraz inteligencje są potencjałami, które można aktywować i wszystkie inteligencje są równoprawne. Autor tego artykułu w bardzo ciekawy i wizualny sposób przedstawia teorię Howarda Gardnera – są one przedstawiane w formie rysunków oraz kolorów umieszczonych na wachlarzu (każdy kolor przedstawia inny typ inteligencji), jest to bardzo dobry sposób przedstawienia teorii Gardnera, gdyż wiele osób jest wrokojcami i bardzo szybko dzięki tym graficznym odnośnikom dotyczącym inteligencji, łatwiej sobie przyswoją podstawowe wiadomości dotyczące teorii Gardnera. Warto zauważyć, iż przy każdym opisie typów inteligencji znajduje się jej odpowiednik graficzny, który ma za zadanie przyciągnąć osoby zainteresowane typami inteligencji do zapoznania się z treścią oraz do przeanalizowania czym jest dany typ inteligencji; Autor do każdego typu inteligencji dopisuje jaka jest sylwetka dziecka, jak można dany typ rozwijać oraz przedstawia rady dla ucznia bądź rodzica jak

wykorzystać dany typ inteligencji w procesie uczenia. Poprzez charakterystykę typów inteligencji autor pokazuje u kogo najczęściej taki typ występuje. Artykuł ten jest napisany bardzo prostym i dostępnym językiem, tak by każdy kto go przeczyta mógł samodzielnie zinterpretować wyniki testu znajdującego się w linku podanym na końcu tekstu.

- 4) Na stronie Szkoły Podstawowej nr 22 im. Generała Stanisława Maczka w Kielcach [13] został umieszczony artykuł dotyczący teorii inteligencji wielorakiej według Howarda Gardnera. Praktycznie niczym się on nie różni od artykułów przedstawionych powyżej, jednakże nie ma w nim graficznego przedstawienia typów i dlatego nie jest tak ciekawie przedstawiony jak tekst na stronie internetowej Stronygratis.pl.
- 5) Na stronie internetowej www.focus.pl [14] znajduje się jedynie test inteligencji wielorakiej, który ma na celu sprawdzenie jakie są mocne strony osoby, która ten test będzie rozwiązywać.

W wielu miejscach znaleziono propozycje testów na inteligencje wielorakie – często zaskakująco różniące się pomiędzy sobą i noszące znamiona wybitnie autorskich opracowań.

Autorzy przetestowali na sobie wszystkie wyżej omawiane testy w celu sprawdzenia korelacji pomiędzy wynikami.

Badania własne przeprowadzono na pierwszym roku kierunku informatyka Politechniki Koszalińskiej w marcu 2019 roku z wykorzystaniem testu ze strony internetowej Stronygratis.pl, zachowując anonimowość studentów.

Warto zauważyć, że badania prowadzone są na dość licznej próbie 71 studentów. Howard Gardner w swoich początkowych badaniach bazował na rezultatach także mniejszych grup studenckich np. w pracy [1] autorzy wspominają grupy 20 osobowe.

W artykule autorzy zamiennie używają następujących terminów – inteligencje wielorakie, inteligencje cząstkowe i subinteligencje dla określenia tego samego pojęcia.

Metodologia badań

Badania przeprowadzono na wybranym przez autorów teście [13] opisanym na stronie [\[sp22.kielce.eu/zawartosc/inteligencje-wielorakie\]](http://sp22.kielce.eu/zawartosc/inteligencje-wielorakie). Studenci w grupach po 15-20 osób zapraszani byli do laboratorium gdzie przy stanowiskach komputerowych uzyskiwali dostęp do arkusza .xls zawierającego pytania. Wypełnianie ankiet trwało zazwyczaj od 5-12 minut. Po zakończeniu testu wektor odpowiedzi wraz ze swoim numerem identyfikacyjnym studenci zapisywali w dedykowanym pliku na pulpicie komputera udostępniając te dane organizatorom

badania. Ten tryb pozwalał na zachowanie względnej anonimowości a jednocześnie umożliwiało osiągnięcie przewidywanych celów eksperymentu – kojarzenia grup najbardziej podobnych profili wg rozmaitych kryteriów.

Użycie określenia „względna anonimowość” wynika z wiedzy o badanej grupie studentów – studenci częściowo znają się, choć nie jest to powszechne, mogą domyślać się swoich tożsamości, jednak nie znają wyników ankiet wykonywanych na sąsiednich stanowiskach lub w innych grupach respondentów. Autorzy uważają więc, że podjęta próba anonimizacji wyników jest całkowicie wystarczająca i skuteczna w aspekcie psychologicznym. Rozważane cechy studentów z pewnością nie są danymi wrażliwymi, są uzyskiwane w warunkach samooceny i dobrowolności i były przez niektórych spośród nich przekazywane niefrasobliwie, bez szczególnej uwagi dla celu badań. Na podstawie autorskich obserwacji można jednak ocenić, że zachowania takie były raczej rzadkie i nie przekraczały ok. 5% całkowitej liczby respondentów.

Ankiety udostępniano studentom w formie arkusza .xls zawierającego dokładnie te same pytania, które proponowano w źródle [sp22.kielce.eu/zawartosc/inteligencje-wielorakie].

Jest to zestaw 24 pytań (konstatacji do oceny ich adekwatności dla respondenta), które w zakamuflowany sposób (oczywiście powierzchownie) powiązane są z poszczególnymi intencjami wielorakimi prof. Gardnera. W ujęciu aksjologicznym ankieta ma być platformą spontanicznych, szybkich reakcji respondenta, nie zaś miejscem drobiazgowej analizy, o co autor ankiety zamierzał studenta zapytać. Tego punktu widzenia organizatorzy zachęcali studentów do reagowania w rozsądnie krótkim czasie.

Oto treść ankiety zgodna z [sp22.kielce.eu/zawartosc/inteligencje-wielorakie].

Które z poniższych stwierdzeń dotyczy Ciebie?	0- 5
Lubię śpiewać i śpiewam dobrze.	0
Uwielbiam krzyżówki i inne gry słowne.	0
Lubię spędzać czas w samotności.	0
Wykresy, mapy, tabele graficzne pomagają mi się uczyć.	0
Najlepiej uczę się, gdy mogę omówić nowe zagadnienia.	0
Lubię sztukę, plastykę, fotografię i prace ręczne.	0
W wolnym czasie często słucham muzyki.	0
Dobrze żyję z ludźmi o odmiennych charakterach i zainteresowaniach.	0
Często myślę o swoich celach i marzeniach związanych z przyszłością.	0
Lubię się uczyć o Ziemi i o przyrodzie.	0
Opieka nad domowymi i innymi zwierzętami sprawia mi przyjemność.	0

Lubię zadania związane z ruchem fizycznym i odgrywaniem scenek.	0
Prace pisemne są dla mnie zazwyczaj łatwe.	0
Łatwo przychodzi mi nauka nowego materiału z matematyki.	0
Gram lub chciałbym grać na jakimś instrumencie muzycznym.	0
Jestem dobry w zajęciach fizycznych takich jak sport lub taniec.	0
Lubię gry liczbowe i łamigłówki logiczne.	0
Najlepiej uczę się, gdy mogę wykonać ćwiczenia praktyczne.	0
Uwielbiam malować, rysować lub projektować coś na komputerze.	0
Często pomagam innym z własnej inicjatywy.	0
Bez względu na pogodę, lubię przebywać na świeżym powietrzu, na dworze.	0
Uwielbiam wyzwanie, gdy trzeba rozwiązać trudny problem matematyczny.	0
Cisza i spokój podczas nauki i rozmyślań są dla mnie ważne.	0
Czytam dla przyjemności codziennie.	0

Studenci w przytłaczającej większości wypełnili ankietę bez trudu, bez dodatkowych pytań i w większości zaciekawieni rezultatami przyszłych porównań i wyników statystycznych.

Studenci mogli oceniać pytania w skali 0-5, a więc w sumie uzyskać wynik od 0 do 120 pkt. Gdyby student wszystkie sugestie ocenił na zero, to uzyskałby końcowy wynik w postaci wektora ośmiu zer dla ośmiu jego inteligencji cząstkowych, gdyby odpowiedział na wszystkie pytania oceną 5 to uzyskałby wynik w postaci wektora ośmiu ocen po 15 pkt dla każdej subinteligencji. To skrajne teoretyczne wyniki. Jakie uzyskano w rzeczywistości?

Badania porównawcze przeprowadzono także w dwóch innych grupach tego samego wydziału. Pierwszą była niewielka grupa studentów elektroniki pierwszego roku a drugą liczniejsza już grupa studentów pierwszego roku studiów drugiego stopnia informatyki.

Wyniki badań

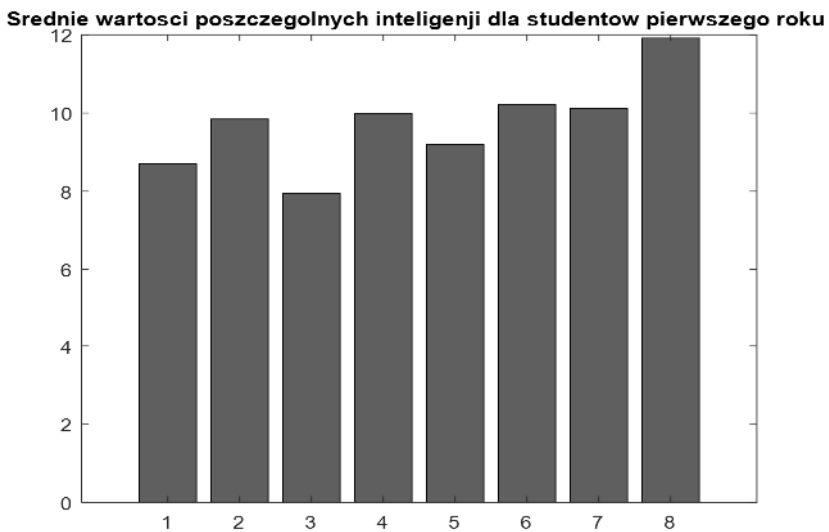
Wyniki badań całego pierwszego roku studentów informatyki do dalszych badań zgromadzono w formie macierzy C o wymiarach $m \times n$, gdzie $m = 1, 2, \dots, M$ to wiersze macierzy zawierające dane kolejnych studentów od pierwszego do ostatniego o numerze wiersza $M = 71$. W każdym wierszu umieszczono 8 cech

studenta (inteligencji cząstkowych). Tym inteligencjom nadano zgodnie z sugestiami Gardnera [3] kolejno numery:

- 1 – przyrodnicza;
- 2 – matematyczno-logiczna;
- 3 – językowa;
- 4 – muzyczna;
- 5 – wizualna;
- 6 – ruchowa;
- 7 – interpersonalna;
- 8 – intrapersonalna;

Wyniki te umieszczono w środowisku Matlab i dokonano kilku podstawowych i, zdaniem autorów, interesujących obliczeń statystycznych. Przede wszystkim obliczono średnią wartość poszczególnych inteligencji cząstkowych dla całego roku.

Wyniki w formie histogramu przedstawiono na rys. 1.



Rys. 1. Średnie wartości inteligencji cząstkowych, kolejno: 1 – przyrodnicza; 2 – matematyczno-logiczna; 3 – językowa; 4 – muzyczna; 5 – wizualna; 6 – ruchowa; 7 – interpersonalna; 8 – intrapersonalna

Histogram z rys. 1 wskazuje na raczej dość nieoczekiwaną dominację inteligencji intrapersonalnej do czego autorzy wrócą we wnioskach. Najbardziej pożądana dla studentów informatyki technicznej, zdaniem autorów, inteligencja

matematyczno-logiczna nie wypada tutaj okazała. To inteligencja numer 2 na rysunku, plasuje się ona dopiero na piątym miejscu w klasyfikacji Gardnera.

Wartości średnie i odchylenia standardowe dla kolejno wszystkich ośmiu subinteligencji dla całej badanej grupy studentów wyniosły:

$m_i = [8.7042 \quad 9.8451 \quad 7.9437 \quad 9.9859 \quad 9.1972 \quad 10.2113 \quad 10.1127 \quad 11.9155]$;

a ich odchylenie standardowe z próby s_i dla $i = 1, 2, \dots, 8$ wyniosło:

$s_i = [3.1550 \quad 3.1697 \quad 2.9611 \quad 3.3573 \quad 3.4749 \quad 2.9322 \quad 2.5554 \quad 1.9030]$.

To właśnie na podstawie wektora m_i widać pozycje inteligencji m_2 (logiczno-matematycznej) na tle pozostałych subinteligencji studentów.

Interesującym badacza pytaniem będzie oczywiście to, czy obserwowane inteligencje cząstkowe różnią się w sposób statystycznie uzasadniony oraz czy występują korelacje pomiędzy nimi.

Zastosowano test istotności dla dwóch średnich – przede wszystkim dla dwóch kluczowych inteligencji – drugiej, matematyczno-logicznej i ostatniej, ósmej – intrapersonalnej. Postawiono hipotezę zerową o równości tych średnich $H_0: m_2 = m_8$.

Zakładano, że odchylenia standardowe dla obu wektorów prób są obliczane na podstawie prób i są różne. Dla przeprowadzenia testu założono także normalny rozkład w obu próbach.

Test istotności dla dwóch średnich przeprowadzono przy założeniu, że odchylenia standardowe w porównywanych próbach są różne [15].

Wówczas test ma postać:

$$h = \frac{m_j - m_k}{2 \sqrt{\left(\frac{S_j^2}{N} + \frac{S_k^2}{N}\right)}}$$

gdzie:

m_j, m_k – średnie arytmetyczne porównywanych inteligencji cząstkowych;

S_j, S_k – odchylenie standardowe tych inteligencji;

N – liczba studentów biorących udział w badaniach (tu $N = 71$).

Test istotności przeprowadzono za pomocą matlabowej funkcji `ttest2` z parametrami `ttest2(x,y,'Vartype', 'unequal')` gdzie:

x – to wektor samoocen studenckich dla inteligencji matematyczno-logicznej x_j dla $j = 1, 2, \dots, N$ (gdzie $N = 71$ i liczba studentów uczestniczących w eksperymencie;

y – to wektor samoocen studenckich dla inteligencji intrapersonalnej y_j dla $j = 1, 2, \dots, N$ (gdzie $N = 71$ i liczba studentów uczestniczących w eksperymencie;

Parametry 'Vartype' i 'unequal' oznaczają odpowiednio zmienny, obliczany poziom istotności i nierówność wariancji (tu w wektorze s_i widać znaczną różnicę odchyleń standardowych).

Po uruchomieniu funkcji `ttest2` uzyskano wynik $h = 1$ oznaczający możliwość odrzucenia hipotezy zerowej o równości średnich z bardzo małym prawdopodobieństwem popełnienia błędu pierwszego rodzaju (odrzuć hipotezy, gdyby była prawdziwa). Tak więc statystycznie potwierdzono to, co intuicyjnie wydaje się oczywiste. Podobnie można byłoby sprawdzić hipotezy dotyczące różnic pomiędzy innymi średnimi, gdzie z pewnością α , nie są już te różnice tak oczywiste, z drugiej jednak strony, nie są one aż tak istotne dla oceny grupy studentów. Najważniejszą konkluzją tego fragmentu badań jest to, że studenci pierwszego roku informatyki mają najsilniej rozwiniętą inteligencję intrapersonalną i jest ona statystycznie większa od pożądanej, zdaniem autorów, inteligencji matematyczno-logicznej.

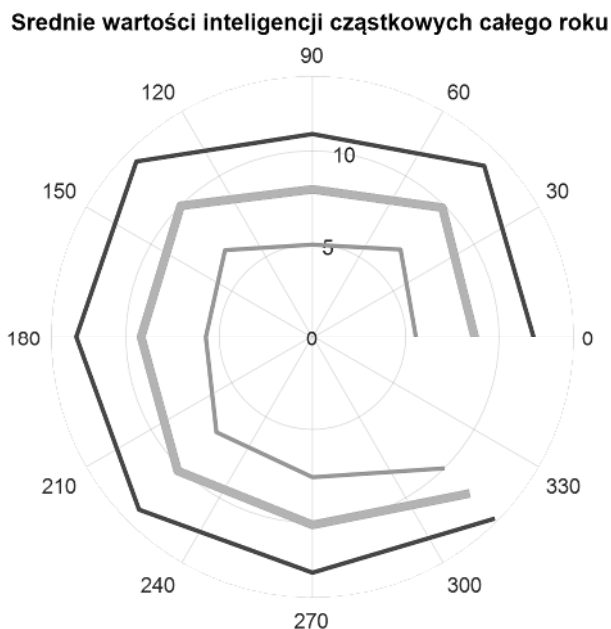
Studując prace Gardnera warto zauważyć jego wysiłki w kierunku poszukiwania osób o bogatej zróżnicowanej osobowości wyrażającej się dużymi odchyleniami inteligencji cząstkowych od średniej inteligencji w grupie. Gardner ze swoim doktorantem Hatchem [1] badali np. wskaźnik ile razy badana osoba przekracza w górę lub w dół barierę $b_i = m_i + k s_i$; lub $b_i = m_i - k s_i$; dla $i = 1, 2, \dots, 8$; gdzie k – współczynnik wzmocnienia oddalenia od średniej. Gardner rozważał $k = 1$ i $k = 2$. Podobnie postąpiono w przedstawianych tu badaniach. Gardner na 20 osób uczestniczących w badaniach znalazł jedną, która mieściła się w przedziale $k = \pm 1$ nie przekraczając z żadną ze swoich inteligencji cząstkowych przedziału położonego blisko średnich.

W podobnych badaniach przeprowadzonych na grupie studentów informatyki znaleziono:

5 na 71 osób pozostających ze swoimi inteligencjami cząstkowymi w przedziale $m_i \pm s_i$, dla $i = 1, 2, \dots, 8$, oraz

17 na 71 osób pozostających ze swoimi inteligencjami cząstkowymi na zewnątrz przedziału $m_i \pm 2s_i$, dla $i = 1, 2, \dots, 8$.

Na rys. 2 przedstawiono w formie wykresu biegunowego średnie wartości inteligencji cząstkowych dla całego roku studentów informatyki oraz wstęgę utworzoną po dodaniu i odjęciu odchylenia standardowego od tej średniej.



Rys. 2. Wykres biegunowy przedstawiający średnie wartości inteligencji cząstkowych (krzywa grubsza w środku) umieszczonej pomiędzy dwiema krzywymi symbolizującymi odchylenie standardowe dodane i odejęte od średniej. Na wykresie kolejno: przy 0 stopni – inteligencja przyrodnicza; 45 stopni – inteligencja matematyczno-logiczna; 90 stopni – językowa; 135 stopni – muzyczna; 180 stopni – wizualna; 215 stopni – ruchowa; 270 stopni – interpersonalna; 315 stopni – intrapersonalna

Biorąc pod uwagę pierwszą próbę stosunek $1/20$ jest zbliżony do $5/71$. Wyniki są więc podobne do uzyskanych przez klasyka.

Studenci interesowali się własnym profilem inteligencji nałożonym na średnie dla całej grupy. Taki wykres przedstawiono na rys. 3. Motywująca i wartościowa w aspekcie rozwoju osobowego wydaje się być informacja, czy jest się powyżej, czy poniżej średniej dla całego roku. Na rysunku przedstawiony jest profil studenta, który dla pięciu inteligencji ma wartość indywidualną powyżej średniej dla roku, dla dwóch – poniżej i raz jest na poziomie średniej.

Profil studenta na tle średnich inteligencji cząstkowych całego roku



Rys. 3. Profil studenta na tle średnich wartości inteligencji całego roku (krzywa pogrubiona). Na wykresie kolejno: przy 0 stopni – inteligencja przyrodnicza; 45 stopni – inteligencja matematyczno-logiczna; 90 stopni – językowa; 135 stopni – muzyczna; 180 stopni – wizualna; 215 stopni – ruchowa; 270 stopni – interpersonalna; 315 stopni – intrapersonalna

Kolejne badania poświęcono korelacji pomiędzy inteligencjami cząstkowymi. Korelacje były obliczane [15] wg wzoru na współczynnik korelacji Pearsona pomiędzy inteligencją X_j i X_k , $j, k \in \{1, 2, \dots, 8\}$

$$r(X_j, X_k) = \frac{\sum_{i=1}^N (X_{ji} - m_j)(X_{ki} - m_k)}{\sqrt{\sum_{i=1}^N (X_{ji} - m_j)^2} * \sqrt{\sum_{i=1}^N (X_{ki} - m_k)^2}}$$

gdzie:

i – numer obserwacji (studenta);

N – liczba studentów (tu $N = 71$);

X_{ji} – wartość j -tej inteligencji cząstkowej i -tego studenta;

X_{ki} – wartość k -tej inteligencji cząstkowej i -tego studenta;

m_j, m_k – średnie arytmetyczne w j -tej i k -tej kolumnie macierzy obserwacji.

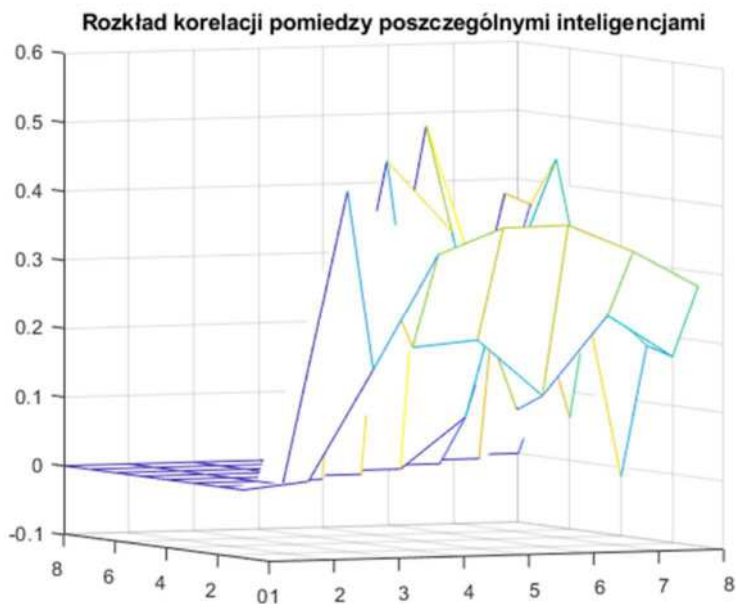
Do obliczenia współczynników korelacji wykorzystano matlabową funkcję *corrcoef* wstawiając jako jej parametr całą macierz obserwacji o 71 wierszach i 8 kolumnach.

Uzyskano macierz korelacji R jak poniżej. To oczywiście macierz symetryczna względem przekątnej. W celu graficznego przedstawienia tych współczynników połowę z nich przyrównano do zera i jedną połowę przestawiono na rys. 2.

Macierz współczynników korelacji pomiędzy poszczególnymi inteligencjami cząstkowymi:

R=

1.0000	0.0096	0.1695	0.3354	0.3716	0.3728	0.3320	0.2789
0.0096	1.0000	0.4252	0.1944	0.2026	0.1189	0.2332	0.1707
0.1695	0.4252	1.0000	0.4612	0.3565	0.0935	0.2746	0.1817
0.3354	0.1944	0.4612	1.0000	0.5048	0.2136	0.4515	-0.0136
0.3716	0.2026	0.3565	0.5048	1.0000	0.0730	0.3804	0.2985
0.3728	0.1189	0.0935	0.2136	0.0730	1.0000	0.3914	0.0621
0.3320	0.2332	0.2746	0.4515	0.3804	0.3914	1.0000	0.1136
0.2789	0.1707	0.1817	-0.0136	0.2985	0.0621	0.1136	1.0000



Rys. 4. Współczynniki korelacji pomiędzy inteligencjami cząstkowymi od x_1 do x_8

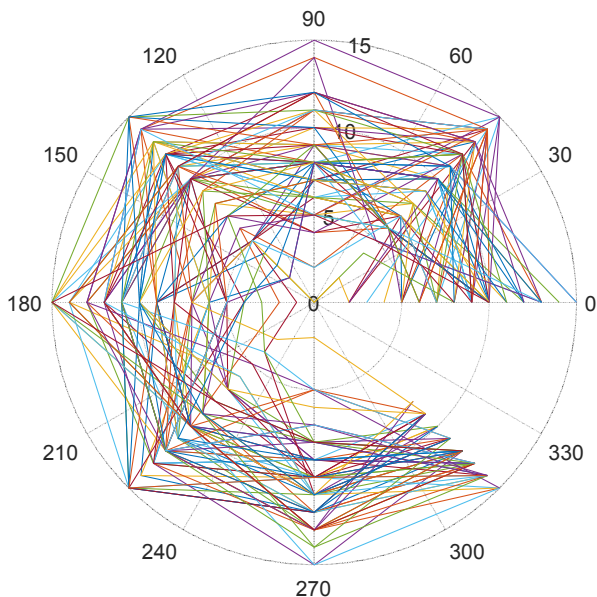
Rys. 4 przedstawia poglądowo zróżnicowanie współczynników korelacji Pearsona, których dokładne wartości podano w macierzy R. Przyjrzyjmy się niektórym z tych współczynników. Wg [16] korelacja powyżej 0.4 (dodatnia lub ujemna) to korelacja umiarkowana. W macierzy R nie widać korelacji bardzo silnych (np. powyżej 0.9) ale kilka tych umiarkowanych występuje.

Np. korelacja między inteligencją muzyczną i wizualną jest w tej grupie najsilniejsza i wynosi ok. 0.5. Raczej ten współczynnik nie dziwi – obie inteligencje wiążą się z wyobraźnią, kreatywnością, myśleniem prawą półkulą mózgową. Dość wysokie są także współczynniki korelacji pomiędzy inteligencją językową a muzyczną (0.46) oraz językową i matematyczno-logiczną (0.42). Tę ostatnią można wytłumaczyć zapewne edukacyjnym naciskiem na te dwa bardzo ważne przedmioty nauczania w ostatnich dziesięciu latach życia studenta. Relatywnie wysoki jest także współczynnik korelacji pomiędzy inteligencją interpersonalną a muzyczną (0.45), także daje się wytłumaczyć aktywnością grupową osób grających czy po prostu lubiących muzykę.

Z kolei najniższe współczynniki korelacji to 0.11 pomiędzy inteligencją interpersonalną i intrapersonalną. Obie te inteligencje cząstkowe są ważne w zawodzie informatyka. O ile inteligencja intrapersonalna jest w tej grupie wysoka (najwyższa), to niepokoi inteligencja interpersonalna ważna dla przyszłej pracy zespołowej w wielu firmach IT w kraju i na świecie. Badani studenci są jednak dopiero na pierwszym roku studiów, ale warto to zauważyć, by kształtować ich kompetencje społeczne. Bardzo niskie są korelacje pomiędzy inteligencją matematyczno-logiczną i przyrodniczą (0.01), co raczej nie dziwi, choć pamiętać należy, że wielkie osiągnięcia sztucznej inteligencji mają wiele korzeni w przyrodzie i medycynie.

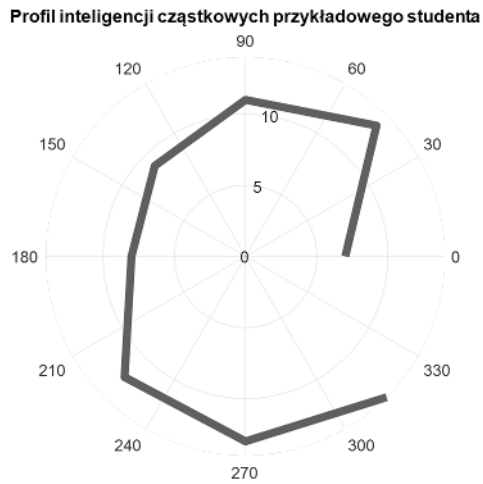
Interesującym wynikiem jest przegląd rozkładu wszystkich inteligencji studentów całego roku na jednym wykresie. By ten rozkład wizualizować zastosowano matlabową funkcję *polar* umożliwiającą przedstawienie przestrzeni wielowymiarowej w formie wykresu radialnego, na którym odległość od centrum wykresu wyznacza wartość zmiennej a poszczególne zmienne (tu – inteligencje cząstkowe) przedstawiane są promieniowo, co pewien ustalony kąt pełnego okręgu. W danym przypadku – ośmiu zmiennych – inteligencji cząstkowych, kąt pełny 360 stopni podzielony został na sektory po 45 stopni, co umożliwiło przedstawienie ośmiu zmiennych (inteligencji cząstkowych) bez nakładania ostatniej zmiennej na oś inicjującą wykres – przedstawiony na rys. 5.

Wykres biegunowy rozkładu inteligencji cząstkowych całego roku



Rys. 5. Wykres biegunowy profili inteligencji cząstkowych 71 studentów pierwszego roku informatyki. Na wykresie kolejno: przy 0 stopni – inteligencja przyrodnicza; 45 stopni – inteligencja matematyczno-logiczna; 90 stopni – językowa; 135 stopni – muzyczna; 180 stopni – wizualna; 215 stopni – ruchowa; 270 stopni – interpersonalna; 315 stopni – intrapersonalna

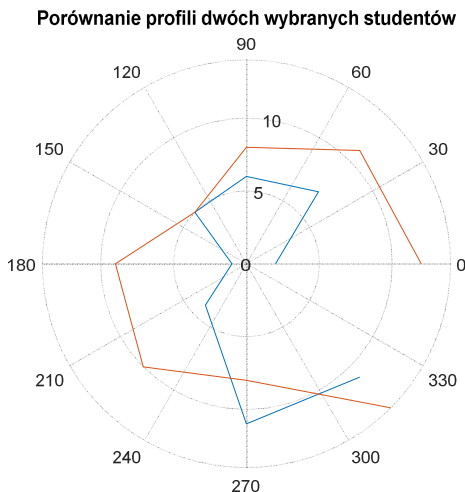
Wykres ten ilustruje wyjątkowo równomierny rozkład wszystkich gardnerowskich inteligencji wśród studentów pierwszego roku. Na wykresie nie widać żadnych szczególnych zagęszczeń, czy to pożądaných, czy też nie. Profil jednego, wybranego studenta przedstawiono na rys. 6.



Rys. 6. Rozkład (profil) inteligencji cząstkowych przykładowego studenta

Z profilu można odczytać, że student ma stosunkowo wysoką inteligencję matematyczno-logiczną rzędu 14 (wartość zmiennej na promieniu 45 stopni) przy niskiej inteligencji przyrodniczej (wartość ok. 7 przy kącie zero stopni) i podobnie niskiej inteligencji wizualnej rzędu 8 (przy kącie 180 stopni).

W trakcie zajęć podsumowujących badania studenci interesowali się porównaniem dwóch profili np. swojego z kolegą czy koleżanką – jak na rys. 7.



Rys. 7. Porównanie rozkładu inteligencji cząstkowych dwóch wybranych studentów

Największe zainteresowanie budziło jednak poszukiwanie studentów o profilach najbardziej zbliżonych do wskazanej osoby. Było to ciekawe doświadczenie bowiem studenci z różnych powodów nie znali się wzajemnie zbyt dobrze. Po pierwsze, dlatego, że to pierwszy rok, często uczestniczący w zajęciach z podziałem na mniejsze grupy, po drugie, to nowa cecha tego pokolenia mileniów zanurzonego w wirtualnej rzeczywistości smartfonów, gier, aplikacji.

Zastosowano tu następujące zasady. Każdy student, który nie życzył sobie, by jego nazwisko było upublicznione w audytorium mógł sobie to zastrzec i jego nazwisko byłoby zastąpione nickiem. To zapewnienie prywatności było konieczne ponieważ trudno było przewidzieć, które nazwiska będą pokazane na ekranie jako te, o profilach najbardziej zbliżonych do profilu rozpatrywanego studenta. Następnie zgłaszał się student zainteresowany poznaniem „najbliższych” sobie pod względem profili inteligencji.

Program w matlabie posługiwał się następującą formułą wyszukiwania:

Macierz „odległości” pomiędzy profilami studentów $D = d(i, j)_{N \times N}$

„Odległość” między dwoma studentami i-tym i j-tym:

$$d(i, j) = \sum_{k=1}^8 |x_{ik} - x_{jk}|$$

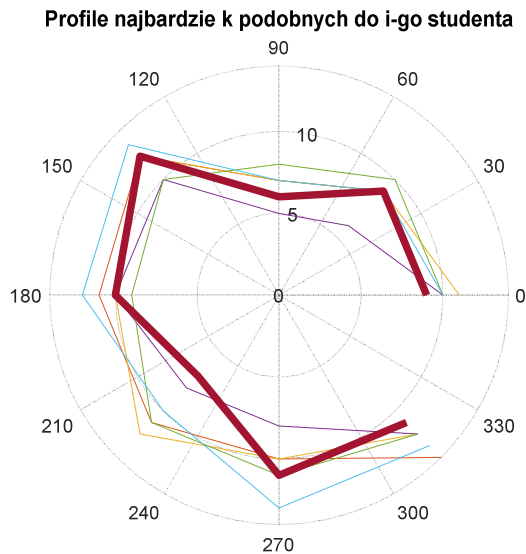
gdzie:

x_{ik} – wartość k-tej inteligencji cząstkowej i-tego studenta;

x_{jk} – wartość k-tej inteligencji cząstkowej j-tego studenta.

Po ustaleniu macierzy „odległości” pomiędzy studentami i wyborze studenta (zgłoszeniu się studenta zainteresowanego ustaleniem najbardziej podobnych do niego pod względem inteligencji wielorakich kolegów) następowało sortowanie studentów. Jeżeli zainteresowany student miał numer np. iD to sortowanie następowało w kolumnie iD macierzy D od najmniejszego $d(iD, j)$ do największego $d(iD, j)$, dla $j = 1, 2, \dots, N$. Z tak utworzonego wektora w kolejności rosnącej odległości wybierano k pierwszych (najbliższych) przypadków zgodnie z tytułową nazwą metody kNN (k NearestNeighbours).

Na ekranie ukazywał się wykres jak na rys. 8 oraz dodatkowo nazwiska studentów (o ile na to wyrazili zgodę).



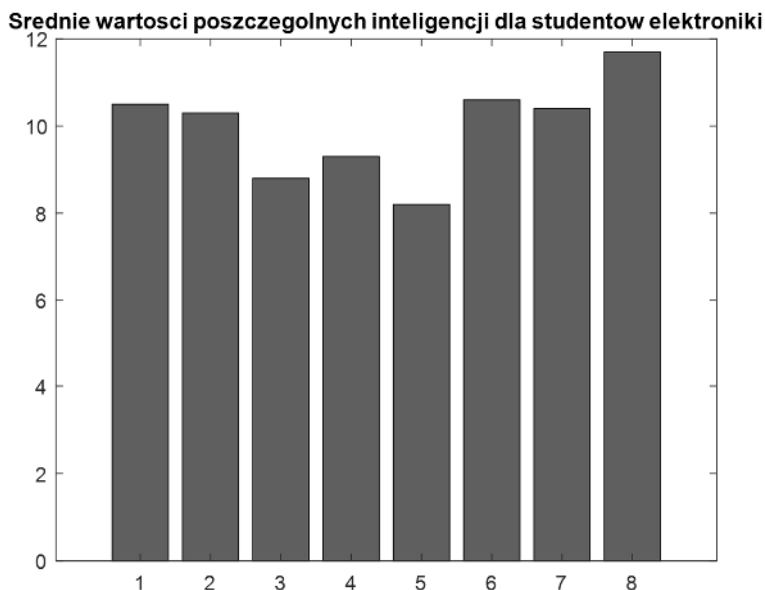
Rys. 8. Profile $k=5$ studentów najbardziej podobnych pod względem inteligencji wielorakich do studenta o profilu zaznaczonego pogrubionym wykresem

Ten fragment prezentacji wyników badań wzbudził największe zainteresowanie studentów. Ze swoim „sąsiedztwem” zapoznała się ponad połowa studentów uczestniczących w zajęciach.

Badania innych grup studenckich

Poza informatykami pierwszego roku studiów, którzy byli przedmiotem głównego zainteresowania badających, ankietowaniu poddano także dwie dostępne w warunkach dydaktycznych wydziału grupy respondentów. Pierwszą była stosunkowo nieliczna grupa studentów pierwszego roku elektroniki tego samego wydziału, druga zaś, to grupa studentów informatyki pierwszego roku, ale drugiego stopnia studiów, a więc trzy lata starsza od badanej grupy podstawowej. Badawczą hipotezą zerową było, że studenci ci w aspekcie badanych inteligencji wielorakich nie różnią się od studentów pierwszego roku informatyki. W grupach tych z uwagi na mniejsze liczby studentów zastosowano nieco inną metodykę zbierania danych. W obu badanych grupach pytania ankietowe były wyświetlane na ekranie a studenci zapisywali swoje odpowiedzi na kartkach, które następnie umieszczane były w arkuszu excelowskim a ostatecznie w matlabie.

Wyniki przeprowadzone na 10-osobowej grupie studentów elektroniki dały histogram inteligencji cząstkowych przedstawiony na rys. 9.



Rys. 9. Histogram średnich wartości inteligencji cząstkowych dla grupy studentów elektroniki pierwszego roku. Na wykresie jak poprzednio kolejno: 1 – przyrodnicza; 2 – matematyczno-logiczna; 3 – językowa; 4 – muzyczna; 5 – wizualna; 6 – ruchowa; 7 – interpersonalna; 8 – intrapersonalna.

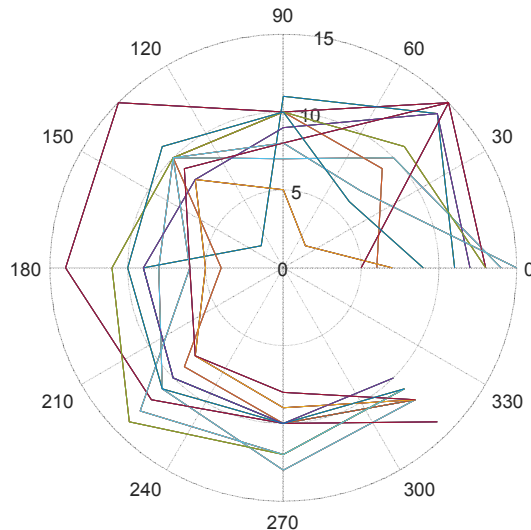
Dokładne wartości średnich i odchyłeń standardowych wyniosły:

$$m_i = [10.5000 \quad 10.3000 \quad 8.8000 \quad 9.3000 \quad 8.2000 \quad 10.6000 \quad 10.4000 \quad 11.7000];$$

$$s_i = [3.5355 \quad 4.3729 \quad 1.8135 \quad 3.2335 \quad 3.0478 \quad 2.0111 \quad 1.5055 \quad 1.0593];$$

W wynikach zwraca uwagę duże odchylenie standardowe dla inteligencji matematyczno-logicznej. To rozproszenie ocen w zakresie tej inteligencji dobrze widoczne jest na wykresie biegunowym sporządzonym dla grupy elektroników i przedstawionym na rys. 10.

Wykres biegunowy rozkładu inteligencji cząstkowych całego roku elektroniki



Rys. 10. Profile inteligencji wielorakich dla studentów elektroniki. Zwraca uwagę szczególnie duża wariancja w zakresie inteligencji matematyczno-logicznej (na promieniu 45 stopni). Na wykresie kolejno: przy 0 stopni – inteligencja przyrodnicza; 45 stopni – inteligencja matematyczno-logiczna; 90 stopni – językowa; 135 stopni – muzyczna; 180 stopni – wizualna; 215 stopni – ruchowa; 270 stopni – interpersonalna; 315 stopni – intrapersonalna

Różnica pomiędzy średnimi wartościami tej ważnej inteligencji cząstkowej w grupach informatyków i elektroników jest niewielka. Średnie wynoszą 9.84 dla informatyków i 10.30 dla elektroników. Zastosowany jak poprzednio test *ttest2* jednoznacznie wskazuje na brak uzasadnienia dla odrzucenia hipotezy zerowej co oznacza brak różnicy pomiędzy inteligencją matematyczno-logiczną pomiędzy grupami informatyków i elektroników.

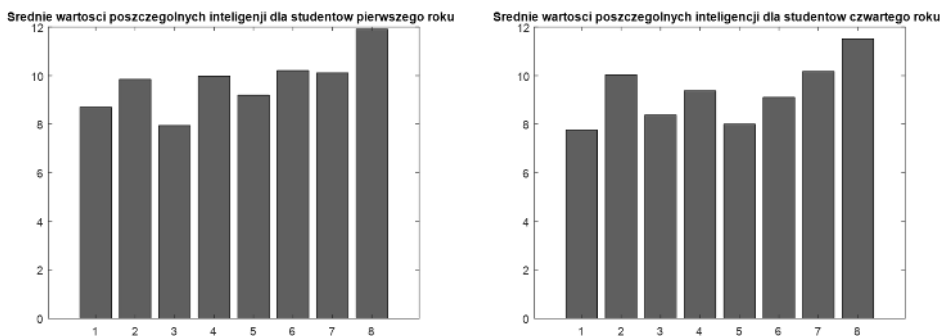
Kolejną rozpatrywaną grupą studencką był pierwszy rok informatyki na studiach drugiego stopnia. Test wypełniło 29 studentów tej grupy i rozkład inteligencji wielorakich uzyskano ze średnimi:

$$m_i = [7.7586 \ 10.0345 \ 8.3793 \ 9.3793 \ 8.0000 \ 9.1034 \ 10.1724 \ 11.5172];$$

i odchyleniami standardowymi:

$$s_i = [3.0898 \ 2.6791 \ 2.5969 \ 3.2339 \ 3.3274 \ 3.2221 \ 2.9889 \ 2.2932];$$

Dla łatwiejszego porównania tych wyników z wynikami uzyskanymi przez studentów pierwszego stopnia na rys. 11 przedstawiono oba histogramy obok siebie. Różnice są dostrzegalne, ale statystycznie nieistotne.



Rys. 11. Porównanie rozkładów inteligencji cząstkowych dla 71 studentów pierwszego roku informatyki studiów pierwszego stopnia (z lewej) i 29 studentów informatyki studiów drugiego stopnia (histogram z prawej). 1 – przyrodnicza; 2 – matematyczno-logiczna; 3 – językowa; 4 – muzyczna; 5 – wizualna; 6 – ruchowa; 7 – interpersonalna; 8 – intrapersonalna

Dla przedstawionych danych postawiono dwie hipotezy zerowe o braku różnicy pomiędzy średnimi wartościami inteligencji matematyczno-logicznej (drugi słupek) oraz o braku różnicy pomiędzy średnimi inteligencjami intrapersonalnymi (ostatni – najwyższy słupek na obu wykresach). Hipotezy te sprawdzone za pomocą przywołanej wcześniej matlabowej funkcji `ttest2` nie zostały odrzucone przy poziomie istotności 5% co oznacza, że nie możemy mówić o różnicach w inteligencji studentów pierwszego i drugiego stopnia studiów.

Wnioski

Podsumowując badania można wymienić kilka konkluzji. Po pierwsze zapoznano studentów z teorią prof. Gardnera nie przydając jej jakiegoś szczególnego znaczenia, zwracając jednak uwagę na pozytywne aspekty wiedzy o własnych możliwościach, o silnych i słabych stronach. Wykorzystanie teorii Gardnera w edukacji, ciągle dyskutowane, przez wielu nauczycieli wysoko cenione przez innych krytykowane, spełnia jednak oczekiwania naszego ustawodawcy zalecającego indywidualizację procesu nauczania na każdym szczeblu kształcenia. Autorzy podchodzili do badań z pozytywnym nastawieniem do tej teorii w nadziei na jej przyszłe wykorzystania i wdrożenie.

Z przebiegu zajęć wynikało duże zainteresowanie studentów, szczególnie w perspektywie przyszłego dostosowania się do wymagań rynku pracy. Zaskoczeniem był stosunkowo niski wskaźnik inteligencji matematyczno-logicznej, typowej dla przedstawicieli studiów w dziedzinie nauk inżynieryjno-technicznych. Wskaźnik ten powtórzył się także w dwóch pozostałych badanych grupach. Natomiast wysoka średnia inteligencja intrapersonalna raczej nie powinna dziwić.

Studenci informatyki są w większości introwertykami, pewnymi siebie, pewnymi łatwego dostępu do dowolnej informacji, skupionymi na własnych problemach. Dodatkowo cechy takiego zachowania pogłębia przynależność do tzw. pokolenia Y powszechnie i z łatwością korzystającego z technologii informatycznych i komunikacyjnych. Z kolei inteligencja interpersonalna jest dla nich niezwykle ważna w kontekście pracy w firmach IT, w których praca zespołowa jest podstawowym sposobem realizacji zadań. Co prawda bywa to często praca zdalna lub niewymagająca szczególnie hałaśliwych zachowań czy ekstrawertycznej dynamiki.

Z tego powodu eksperyment poszukiwania osób o podobnych inteligencjach może przyczynić się do utworzenia i pogłębienia relacji pomiędzy studentami, pobudzenia ich wspólnych zainteresowań.

Najciekawszym zamknięciem eksperymentu byłoby powtórne jego przeprowadzenie za kilka lat np. na drugim stopniu studiów a może nawet w warunkach zatrudnienia. Z oczywistych powodów jest to utrudnione, choć możliwe. Na tym jednak etapie badań autorzy postanowili podzielić się tym co zaobserwowali licząc na zainspirowanie do podobnych badań w innych środowiskach. Ważnym rozszerzeniem badań byłoby włączenie do nich także pracowników firm IT, najlepiej tego samego regionu, co uczelnia. Pozwoliłoby to na ewentualne stwierdzenie zmian w rozkładach inteligencji spowodowanym oddziaływaniem środowiska pracy. Kolejnym istotnym kierunkiem dalszych badań mogłaby być tzw. kalibracja testów pozwalająca na większą obiektywizację i powtarzalność wyników. Autorzy dziękują studentom dodatkowo zaangażowanym w organizację testów, wśród których do szczególnie aktywnych należeli – Paweł Wartel, Adam Kruger, Maciej Bąk, Patryk Madej.

Bibliografia

1. Gardner, H., & Hatch, T. (1989). Educational implications of the theory of multiple intelligences. *Educational researcher*, 18(8), 4-10.
2. Gardner, H. (1992). *Multiple intelligences* (Vol. 5, p. 56). Minnesota Center for Arts Education.
3. Gardner H., *Inteligencja wielorakie*, Wyd. Laurum, Warszawa 2002 r.
4. Gardner, H. (2003). Multiple intelligences after twenty years. *American Educational Research Association, Chicago, Illinois*, 21.
5. Smith, M. K. (2002). Howard Gardner and multiple intelligences. *The encyclopedia of informal education*, 15, 2012.
6. Battro, A. M. (2010). The teaching brain. *Mind, Brain, and Education*, 4(1), 28-33.

7. Battro, A. M., Fischer, K. W., & Léna, P. J. (Eds.). (2010). *The educated brain: Essays in neuroeducation*. Cambridge University Press.
8. Rożewicz O. *Inteligencje wielorakie*. Web. o-rozewicz.pl/2018/08/03/inteligencje-wielorakie. Dostęp 14 marca 2019.
9. <https://dziecisawazne.pl/8-inteligencji-wedlug-prof-h-gardnera/> Dostęp 5 maja 2019
10. <http://edulider.pl/edukacja/koncepcja-inteligencji-wielorakich-wedlug-howarda-gardnera> Dostęp 11 kwietnia 2019
11. Briefitip *Test inteligencji wielorakich*. Web. briefitip.pl/test-inteligencji-wielorakiej-rodzaje-inteligencji. Dostęp 2 maja 2019.
12. Litwin E. *Inteligencje wielorakie*. Web. edytalitin.pl/tl/Inteligencje-wielorakie. Dostęp 11 marca 2019.
13. SP 22 Kielce *Inteligencje wielorakie*. Web. sp22.kielce.eu/zawartosc/inteligencje-wielorakie. Dostęp 15 marca 2019.
14. <https://www.focus.pl/artykul/nie-tylko-iq> Dostęp 22 kwietnia 2019
15. Starzyńska W., (2006) *Statystyka praktyczna*, PWN, Warszawa
16. <https://www.statystyka-zadania.pl/wspolczynnik-korelacji-liniowej-pearsona/> Dostęp 21 kwietnia 2019

Abstrakt

Celem artykułu jest określenie rozkładu inteligencji wielorakich zdefiniowanych w latach 80. ub. wieku przez Howarda Gardnera. Badaną grupą byli przede wszystkim studenci pierwszego roku informatyki. Na ich tle rozważano także dwie inne grupy studenckie. Postawiono kilka hipotez badawczych m.in. po to, by stwierdzić różnicowanie wśród studentów całego roku, znaleźć uśredniony rozkład inteligencji, inteligencje dominujące, korelacje pomiędzy poszczególnymi inteligencjami oraz ewentualne różnice na tle innych grup studenckich. Badana grupa ok. 70 studentów informatyki w dziedzinie nauk technicznych była specyficznie wyselekcjonowana, po pierwszym trudnym semestrze i powinna zdradzać wyraźne preferencje związane z logiczno-matematycznym aspektem inteligencji. Poza tym interesujące były korelacje pomiędzy inteligencjami i ewentualne podobieństwa pomiędzy profilami inteligencji studentów mierzone wg kryterium kNN (k najbliższych sąsiadów). Badania przeprowadzone były w warunkach pełnej dobrowolności przystąpienia do testu i na zasadzie samooceny wg pytań sugerowanych w jednym z internetowych źródeł. Zdaniem autorów wyniki wydają się być interesujące, choć nieco zaskakujące.

Abstract

The aim of the article is to determine the distribution of multiple intelligences defined in the 80s of the previous century by Howard Gardner. The first year group of computer science students were the study group. Two other student groups were also considered against them. Several research hypotheses have been proposed, including in order to find a differentiation among students of the whole year, find the average distribution of intelligence, dominant intelligences, correlations between individual intelligences and possible differences in comparison with other student groups. The studied group of about 70 students of computer science in the field of technical sciences was specifically selected after the first difficult semester and should reveal clear preferences related to the logical-mathematical aspect of intelligence. In addition, the correlations between the intelligences and possible similarities between the student intelligence profiles were interesting as measured by the criterion kNN (k nearest neighbors). The research was carried out in conditions of full voluntary acceptance of the test and on the basis of self-assessment according to questions suggested in one of the Internet sources. According to the authors, the results seem to be interesting, although somewhat surprising.

Keywords: Multiple intelligences, engineering education, student's personality, profile of intelligence

Paweł Poczekajło
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
pawel.poczekajlo@tu.koszalin.pl

Interaktywna stacja pogodowa w oparciu o układ ESP z obsługą sieci WiFi

Słowa kluczowe: stacja pogodowa, sieć WiFi, ESP8266

1. Wprowadzenie

Urządzenia interaktywne, których główna funkcjonalność powiązana jest z dostępem do internetu są obecnie coraz powszechniej stosowane. Komunikacja, zdalne sterowanie czy nawet pobieranie i udostępnianie informacji to standardowe funkcjonalności nawet podstawowych urządzeń RTV i AGD. Również smartfony, które dziś ma prawie każdy, stanowią często element większego systemu ułatwiającego nam różne czynności (zazwyczaj poprzez dodatkowe aplikacje instalowane na urządzeniu mobilnym). Główną zaletą takich rozwiązań jest oszczędność czasu i wygoda wynikająca z dostosowania warstwy sprzętowej i programowej do różnych potrzeb. Jednymi z najczęściej wyszukiwanych na co dzień informacji są dane dotyczące warunków atmosferycznych. Istnieje wiele aplikacji pogodowych (również darmowych) na same smartfony, które dostępne są np. w serwisach Google Play [1] lub App Store [2]. Wadą tego rozwiązania są uciążliwe reklamy (zwłaszcza w przypadku bezpłatnych wersji) oraz udostępnianie różnych informacji np. lokalizacyjnych (z GPS), co obecnie dla wielu użytkowników, może dyskwalifikować wykorzystanie takiego oprogramowania. Alternatywą są urządzenia w postaci stacji pogodowych, które dzięki czujnikom umieszczanym na zewnątrz budynku dają możliwość podglądu na wyświetlaczu panujących warunków atmosferycznych. Wadą tego rozwiązania jest zazwyczaj słaba przewidywalność pogody, gdyż stacje bazują jedynie na bieżących pomiarach. Ponadto, często zdarza się że zastosowane czujniki są kiepskiej jakości, co wprowadza błędy pomiarowe. Lepsze urządzenia, które pobierają dane np. z internetu, są z kolei znacznie droższe. Warto też zaznaczyć, że dostępne stacje

pogodowe to praktycznie wyłącznie samodzielne urządzenia, które nie mają możliwości podłączenia dodatkowych modułów pozwalających na rozbudowę.

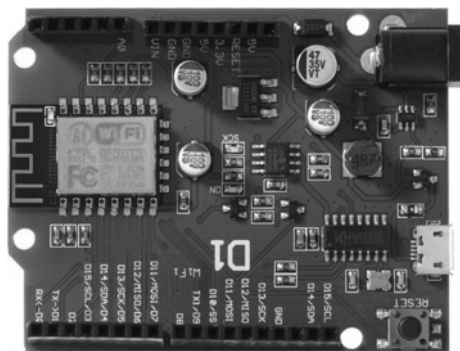
W niniejszym artykule przedstawiono prototyp prostej interaktywnej (dostęp do sieci www) stacji pogodowej, bazującej na uniwersalnych modułach mikroprocesorowych, co gwarantuje łatwą rozbudowę przy względnie niskim koszcie. Pobieranie informacji z sieci ma zapewnić stabilność i wiarygodność danych, które otrzymywane będą z serwerów bazujących na pomiarach ze stacji meteorologicznych. W takiej sytuacji możliwe jest również odczytanie i analiza pogody przewidywanej na najbliższe godziny, czy dni. Zastosowania takiego urządzenia to np. stanowiska informacyjne przy plażach, ostrzegające o nadchodzącym deszczu lub też burzy.

2. Projekt stacji pogodowej (urządzenia)

Zgodnie z założeniami, urządzenie musi zapewniać łączność z siecią www, dodatkowo przyjęto, że najlepiej jest to zrealizować przez zastosowanie bezprzewodowego połączenia WiFi. Na rynku dostępnych jest wiele rozwiązań, w tym najpopularniejsze bazujące na układzie ESP8266 [3]. W projekcie zastosowano gotowy moduł z układem ESP, który ułatwia podłączenie i uruchomienie całości. Za prezentację (wizualizację) danych odpowiada wyświetlacz. Na tym etapie nie przewidziano elementów pozwalających na sterowanie stacją (np. ustalenie lokalizacji dla której chcemy otrzymać prognozę pogody – wpisano na stałe kilka przykładowych miast). Zasilanie dostosowano do wykorzystanej płytki prototypowej.

2.1. Płytki prototypowa

Zastosowana płytki prototypowa (rysunek 1), to moduł kompatybilny wymiarowo z płytką Arduino Uno, na którym wlutowano podstawkę z układem ESP8266.



Rys. 1. Płytki z układem ESP8266, zgodna z Arduino Uno

Zgodność z Arduino jest również zapewniona na poziomie środowiska programistycznego (Arduino IDE), w którym wystarczy dograć odpowiednie biblioteki. Pozwala to na bardzo łatwe i szybkie zaprogramowanie całego urządzenia. Dużą zaletą zastosowanej płytki są też elementy modułu zasilającego, który udostępnia do dyspozycji użytkownika 5V oraz 3,3V prądu stałego. Zasilanie zewnętrzne doprowadzone do gniazda DC musi być na poziomie od 6V do 24V (lub 5V do gniazda microUSB). Do komunikacji z komputerem zastosowano konwerter USB-UART w postaci układu CH340. Jest to jedno z najprostszych rozwiązań, które pozwala na zaprogramowanie mikroprocesora bez stosowania specjalnych programatorów (wykorzystując tzw. bootloader).

2.2. Mikrokontroler

Główny układ sterujący to wspomniany ESP8266. Zazwyczaj występuje on w wersji wbudowanej na podstawce z odpowiednimi wyprowadzeniami oraz anteną WiFi w postaci obwodu drukowanego, a cały moduł ma oznaczenie ESP-12E i został wyposażony w 4MB pamięci typu flash. Mikrokontroler wraz z bezpośrednimi peryferiami jest dodatkowo ekranowany przed ewentualnymi zakłóceniami. Zasilanie zgodne jest ze standardem 3,3V, więc wykorzystując zewnętrzne moduły należy zwrócić uwagę, aby były kompatybilne z takim napięciem. Podłączenie na piny układu ESP napięcia wyższego niż wskazane może spowodować jego nieodwracalne uszkodzenie. ESP8266 bazuje na procesorze Tensilica L106. Jest to 32-bitowa jednostka typu RISC taktowana standardowo zegarem 80MHz (możliwe jest również taktowanie 160MHz). Moduł ma 11 wejść/wyjść cyfrowych (GPIO), jedno wejście przetwornika analogowo-cyfrowego (ADC), interfejs SPI, interfejs I2C oraz dwa niezależne interfejsy UART. Główne zalety ESP8266 to bardzo małe wymiary oraz niski pobór prądu (w stanie spoczynku nawet poniżej 10 μ A). Obsługiwany jest standard WiFi 802.11 b/g/n, ale moduł pracuje jedynie na częstotliwości 2,4GHz. Ważnym aspektem jest również fakt, że urządzenie może pracować zarówno jako klient sieciowy jak i punkt dostępowy (access point) i to w tym samym czasie.

2.3. Wizualizacja danych

Prezentacja danych dla użytkowników może odbywać się z wykorzystaniem najróżniejszych elementów, jak np. diody LED, czy wyświetlacze 7-seg. lub ciekłokrystaliczne. W prezentowanym projekcie, ze względu na ograniczanie kosztów, zastosowano wyświetlacz alfanumeryczny 2x16 (dwa wiersze po szesnastu znaków) bazujący na sterowniku HD44780. Standardowo wyświetlacz ten jest dostosowany do pracy z napięciem (i logiką) 5V i dla niższych napięć (np. 3,3V) działanie nie jest możliwe. Na rynku dostępne są specjalne przystawki i przejściówki, jednak prostszym (konstrukcyjnie) rozwiązaniem jest zakup samego wyświetlacza kompatybilnego z napięciem 3,3V lub jego odpowiednie

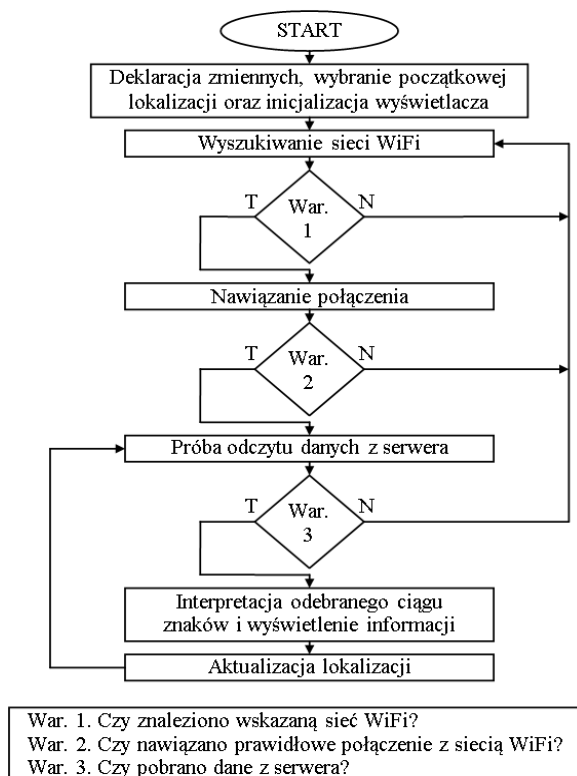
dostosowanie (jeśli konstruktor przewidział taką możliwość). Przykładem jest wyświetlacz o oznaczeniu CBC016002BS00, który dostosowany jest do pracy z napięciem 3,3V. Dokładnie taki wyświetlacz został zastosowany w prezentowanym projekcie. Złożone urządzenie umieszczone zostało w obudowie DT-065, która posiada odpowiednie wycięcie na wyświetlacz alfanumeryczny 2x16. Na rysunku 2 zaprezentowano działające urządzenie.



Rys. 2. Widok działającego urządzenia

3. Program mikrokontrolera

Program mikrokontrolera to aplikacja, której główna funkcjonalność sprowadza się do nawiązania połączenia z siecią WiFi, pobrania bloku danych z określonego serwera oraz interpretacji i wyświetlenia odpowiednich informacji. Ponieważ w artykule prezentowany jest prototyp urządzenia, poszczególne funkcje zostały maksymalnie uproszczone, m.in. określono kilka stałych lokalizacji, dla których podawane są warunki pogodowe. Również kwestia połączenia bezprzewodowego, została sprowadzona do wyszukania jednej (z góry określonej) sieci o wprowadzonym w programie identyfikatorze i hasle dostępu. Na rysunku 3 przedstawiono schemat blokowy programu.



Rys. 3. Schemat blokowy programu interaktywnej stacji pogodowej

3.1. Dane meteorologiczne

Dane dotyczące pogody mogą być pobierane z różnych serwerów meteorologicznych. Jako priorytet przyjęto łatwość pobierania i obsługi informacji oraz bezpłatny dostęp do jak największej liczby różnych danych. Jednym z najpopularniejszych serwisów jest [4], który bazuje na informacjach zbieranych ze stacji meteorologicznych z całego świata. Ze względu na automatyczną interpretację danych, bardzo dobrym rozwiązaniem jest prezentacja na stronie informacji dla konkretnej lokalizacji w postaci prostej tabelki [5] lub nieobrobionego ciągu znaków (ang. „raw data”) [6]. Przykładowy, odebrany przez stronę www ciąg danych meteorologicznych dla miasta Koszalin wygląda następująco:

```

„9:00 PM CEST October 19, 2018|46|N/A|N/A|91%|44|SSW at 2|
30.27|Mostly Cloudy|6.0|07:30 AM (CEST)|05:48 PM (CEST)|10
|1830|N/A|N/A|N/A|N/A|Koszalin|Poland|04:28 PM
(CEST)|12:58 AM (CEST)|”

```

Poszczególne informacje rozdzielone są symbolem „|”, więc w praktyce wystarczy odliczyć odpowiednią liczbę kolejnych wystąpień tego znaku, aby wyselekcjonować interesujące informacje. Kolejno można odczytać następujące dane (zależnie od konkretnej stacji meteorologicznej część danych może być niedostępna):

- godzina i data,
- temperatura,
- czynnik chłodzący,
- wskaźnik (odczucie) ciepła,
- wilgotność,
- punkt rosy,
- siła i kierunek wiatru,
- ciśnienie,
- warunki,
- widoczność,
- godzina wschodu słońca,
- godzina zachodu słońca,
- liczba dni ze wzrostem temperatury,
- wartość max. dnia poprzedniego,
- wartość min. dnia poprzedniego,
- (brak danych)
- (brak danych),
- (brak danych),
- miasto,
- państwo,
- godzina wschodu księżyca,
- godzina zachodu księżyca,
- kod stacji pogodowej,
- (brak danych).

4. Podsumowanie

W artykule zaprezentowano prototyp interaktywnej stacji pogodowej. Główną zaletą jest prosta budowa, bazująca na gotowych układach, dzięki czemu możliwe jest szybkie rozszerzenie urządzenia o dodatkowe moduły np. wykonawcze. Stacja działa prawidłowo i spełnia określone założenia, w związku z tym, może stanowić podstawę do dalszego rozwoju projektu (np. rozbudowa o zdalną konfigurację, GPS) i jego praktycznego zastosowania.

Bibliografia

1. GooglePlay [online], https://play.google.com/store/apps/category/WEATHER/collection/topselling_free?hl=pl (data dostępu 15.05.2019r.)
2. AppStore [online], <https://support.apple.com/pl-pl/HT207492> (data dostępu 15.05.2019r.)
3. Espressif [online], <https://www.espressif.com/en/products/hardware/esp8266ex/overview> (data dostępu 15.05.2019r.)
4. Weather Underground [online], <https://www.wunderground.com/> (data dostępu 15.05.2019r.)

5. Weather Underground [online], https://www.wunderground.com/auto/mobile_metric/global/stations/12105.html#forecast (data dostępu 15.05.2019r.)
6. Weather Underground [online], <http://www.wunderground.com/auto/raw/global/stations/12105.html> (data dostępu 15.05.2019r.)

Streszczenie

W artykule przedstawiono prototyp prostej interaktywnej stacji pogodowej, działającej w oparciu o sieć WiFi. Wykorzystany został moduł bazujący na układzie Espressif ESP8266. Dane meteorologiczne pobierane są z wybranego serwera pogodowego dla wskazanej lokalizacji. Prezentacja poszczególnych informacji (po odpowiednim zinterpretowaniu) realizowana jest z zastosowaniem wyświetlacza alfanumerycznego.

Abstract

The prototype of a simple interactive weather station is presented in the paper. It is based on communication with WiFi network. A module with microcontroller Espressif ESP8266 is used in the project. Meteorological data are received from a weather server for chosen localization. Information are interpretation by the microcontroller and presented on alphanumeric display.

Keywords: weather station, WiFi network, ESP8266

Paweł Poczekajło
Wydział Elektroniki i Informatyki
Politechnika Koszalińska
pawel.poczekajlo@tu.koszalin.pl

Pomiar oraz porównanie wrażliwości różnych realizacji filtrów FIR 3D

Słowa kluczowe: filtry FIR 3D, wrażliwość, kwantowanie współczynników, struktura bezpośrednia, ortogonalna struktura rotatorowa

1. Wprowadzenie

Systemy cyfrowego przetwarzania sygnałów (CPS) stanowią obecnie jedno z najintensywniej rozwijanych układów, co wynika głównie z powszechności ich stosowania. Różne przeznaczenie takich systemów skutkuje koniecznością implementacji w różnych jednostkach sprzętowych (np. mikroprocesory, procesory DSP, układy FPGA). Ponieważ zasadniczym zagadnieniem przy realizacji CPS jest wykonywanie operacji arytmetycznych, istotnym problemem staje się uwzględnienie skończonej precyzji obliczeń, która wynika wprost z dostępnej jednostki arytmetyczno-logicznej danego procesora. Podstawowym czynnikiem charakteryzującym dany system CPS jest zespolona charakterystyka częstotliwościowa, w tym głównie charakterystyka amplitudowa. Długość rejestrów i dokładność obliczeń są jednymi z kluczowych kwestii analizowanych pod kątem wpływu na otrzymywane charakterystyki. Powiązane jest to bezpośrednio ze skończoną precyzją zapisu współczynników (np. transmitancji) danego systemu. Zagadnienie to określane jest jako wrażliwość charakterystyki amplitudowej badanego systemu na kwantowanie współczynników [1, 2]. Zazwyczaj wrażliwość prezentowana jest jako charakterystyka wyznaczona zgodnie z równaniem:

$$S_i = \frac{\Delta T(\omega_h, \omega_v, \omega_d)}{\Delta b_{ije}}, \quad (1)$$

gdzie:

$\Delta T(\omega_h, \omega_v, \omega_d)$ – zmiana charakterystyki amplitudowej filtru 3D,

Δb_{ije} – zmiana wybranego współczynnika systemu opisanego transmitancją

$$T(z_h, z_v, z_d) = \sum_{e=0}^l \sum_{j=0}^m \sum_{i=0}^k b_{ije} z_h^{-i} z_v^{-j} z_d^{-e}$$

W różnych publikacjach wykazano, że określone realizacje systemów CPS, np. ortogonalne, mogą charakteryzować się niższą wrażliwością niż bezpośrednie realizacje bazujące na operacji splotu. W artykule podjęto kwestię pomiaru i porównania wrażliwości dla różnych realizacji separowalnych filtrów 3D o skończonej odpowiedzi impulsowej (ang. FIR) opisanych transmitancją [3]:

$$\begin{aligned} T(z_h, z_v, z_d) &= T_h(z_h) T_v(z_v) T_d(z_d) \\ &= \left(\sum_{i=0}^k b_{hi} z_h^{-i} \right) \left(\sum_{j=0}^m b_{vj} z_v^{-j} \right) \left(\sum_{e=0}^l b_{de} z_d^{-e} \right), \end{aligned} \quad (2)$$

gdzie:

k, m, l – stopnie transmitancji składowych,

b_{hi}, b_{vj}, b_{de} – współczynniki składowych transmitancji jednowymiarowych.

Finalnie planuje się uzyskać liczbowe dane pozwalające na określenie, jaki odsetek filtrów FIR w realizacji ortogonalnej z wykorzystaniem rotatorów jest lepszy (pod względem niższej wrażliwości) niż ich bezpośrednie implementacje z zastosowaniem splotu.

2. Analizowane filtry i metody pomiarowe

Do pomiarów wrażliwości wybrano dwa rodzaje struktur realizujących zadane filtry FIR 3D. Pierwsza to struktura bezpośrednia bazująca na splotu 3D, druga to odpowiadająca jej ortogonalna realizacja rotatorowa otrzymana zgodnie z [3, 4]. W pierwszym przypadku współczynnikami są stałe transmitancji (splotu), w drugim wartości funkcji trygonometrycznych opisujące poszczególne rotatory. Przyjęto, że analizowane są dolnoprzepustowe (LP), górnoprzepustowe (HP), środkowo-przepustowe (BP) i środkowozaporowe (SB) filtry 3D o stopniach transmitancji (2) $k=m=l=4$. Pomiarów dokonano przy kwantowaniu współczynników przez zaokrąglanie dla rejestrów 8-, 16- i 24-bitowych zapisanych w systemie stałopozycyjnym U2 (odpowiednio Q1.8, Q1.15 i Q1.23). Jednocześnie dla każdego typu filtrów (LP/HP/BP/SB) określono grupę po 1000 systemów o różnych częstotliwościach granicznych. Finalnie więc, pomiarom podlegało 12000 różnych filtrów w dwóch realizacjach każdy (bezpośredniej i rotatorowej-ortogonalnej). Dla tak dużej liczby filtrów, przeprowadzenie analizy wrażliwości osobno dla

każdego współczynnika (a może być ich nawet do 125 dla jednego filtru) jest bardzo czasochłonne. Również sama prezentacja w formie wykresów jest kłopotliwa i praktycznie niemożliwa do zamieszczenia w niniejszej publikacji. Zdecydowano więc, że wrażliwość wyznaczana będzie jako wartość skalarna, dla jednocześnie skwantowanych wszystkich współczynników w danej realizacji, zgodnie ze wzorem:

$$S_{MSE} = \frac{1}{n} \sum_{\omega_h, \omega_v, \omega_d} (M(\omega_h, \omega_v, \omega_d))^2, \quad (3)$$

gdzie:

n – liczba próbek charakterystyki wrażliwości,

$M(\omega_h, \omega_v, \omega_d) = T_Q(\omega_h, \omega_v, \omega_d) + T_C(\omega_h, \omega_v, \omega_d)$ – wrażliwość rzeczywista (bezwzględna) charakterystyki amplitudowej,

$T_Q(\omega_h, \omega_v, \omega_d)$ – charakterystyka amplitudowa badanego filtru 3D o założonej realizacji ze skwantowanymi wszystkimi współczynnikami,

$T_C(\omega_h, \omega_v, \omega_d)$ – charakterystyka amplitudowa filtru 3D o realizacji splotowej ze współczynnikami określonymi z pełną dostępną precyzją.

Zgodnie ze wzorem (3) obie realizacje o skwantowanych współczynnikach porównywane są do jednej wzorcowej charakterystyki $T_C(\omega_h, \omega_v, \omega_d)$. Takie podejście ma jedną wadę, mianowicie przy pomiarach realizacji rotatorowej uwzględnia się błędy wynikające z poszczególnych etapów syntezy (otrzymywania) systemów ortogonalnych. Teoretycznie, tego typu błędy nie powinny być brane pod uwagę przy wyznaczaniu wrażliwości. W związku z tym, dla struktur rotatorowych dodatkowo wyznaczono wrażliwość w następującej postaci:

$$Sr_{MSE} = \frac{1}{n} \sum_{\omega_h, \omega_v, \omega_d} (M(\omega_h, \omega_v, \omega_d))^2, \quad (4)$$

gdzie:

n – liczba próbek charakterystyki wrażliwości,

$M(\omega_h, \omega_v, \omega_d) = T_Q(\omega_h, \omega_v, \omega_d) + T_R(\omega_h, \omega_v, \omega_d)$ – wrażliwość rzeczywista (bezwzględna) charakterystyki amplitudowej,

$T_Q(\omega_h, \omega_v, \omega_d)$ – charakterystyka amplitudowa badanego filtru 3D o ortogonalnej realizacji rotatorowej ze skwantowanymi współczynnikami,

$T_R(\omega_h, \omega_v, \omega_d)$ – charakterystyka amplitudowa filtru 3D o ortogonalnej realizacji rotatorowej ze współczynnikami określonymi z pełną dostępną precyzją.

3. Wyniki pomiarów

Wszystkie realizowane pomiary oraz analizy i symulacje wykonane zostały z wykorzystaniem środowiska obliczeniowego Scilab. Prezentacja otrzymanych wyników, nawet w formie skalarnych wartości dla każdego filtru, zajęła by przynajmniej kilka stron. Dlatego też, w tabeli 1 zebrano wyniki porównań w postaci liczby filtrów w danej realizacji (np. rotatorowej) z mniejszym błędem, niż w drugiej realizacji (np. ze splotem 3D), co też było głównym celem badania.

Tabela 1. Wyniki porównań wrażliwości różnych realizacji filtrów.

<i>Grupa filtrów i długości rejestrów</i>	<i>Liczba filtrów o realizacji bezpośredniej z niższym S_{MSE} niż odpowiadająca jej ortogonalna realizacja rotatorowa</i>	<i>Liczba filtrów o ortogonalnej realizacji rotatorowej z niższym S_{MSE} niż odpowiadająca jej realizacja bezpośrednia</i>	<i>Liczba filtrów o ortogonalnej realizacji rotatorowej z niższym S_{RMSE} niż odpowiadająca jej realizacja bezpośrednia</i>
<i>Dolnoprzepustowe (rejstry 8-bitowe)</i>	0	1000	1000
<i>Dolnoprzepustowe (rejstry 16-bitowe)</i>	6	994	994
<i>Dolnoprzepustowe (rejstry 24-bitowe)</i>	182	818	996
<i>Górnoprzepustowe (rejstry 8-bitowe)</i>	0	1000	1000
<i>Górnoprzepustowe (rejstry 16-bitowe)</i>	7	993	994
<i>Górnoprzepustowe (rejstry 24-bitowe)</i>	870	130	993
<i>Środkowoprzepustowe (rejstry 8-bitowe)</i>	0	1000	1000
<i>Środkowoprzepustowe (rejstry 16-bitowe)</i>	1	999	999
<i>Środkowoprzepustowe (rejstry 24-bitowe)</i>	797	203	1000
<i>Środkowozaporowe (rejstry 8-bitowe)</i>	11	989	989
<i>Środkowozaporowe (rejstry 16-bitowe)</i>	1	999	1000
<i>Środkowozaporowe (rejstry 24-bitowe)</i>	500	500	1000
<i>W ujęciu procentowym względem liczby wszystkich analizowanych filtrów</i>	19,8%	80,2%	99,7%

4. Wnioski i podsumowanie

Z tabeli 1 można wywnioskować, że przy niższych poziomach kwantowania (rejstry 8- i 16-bitowe), filtry ortogonalne o realizacji rotatorowej mają prawie zawsze niższą wrażliwość niż odpowiadające im filtry w realizacji bezpośredniej ze splotem 3D. Pojedyncze przypadki, gdzie struktura bezpośrednia ma niższą wartość S_{MSE} , wynikają z losowego dopasowania współczynników przy ich kwantowaniu. Dla rejestrów 24-bitowych, przy pomiarze S_{MSE} , struktury rotatorowe w większości przypadków wypadają gorzej. Jednak już pomiary Sr_{MSE} , które nie uwzględniają błędów syntezy, jednoznacznie wskazują na niższą wrażliwość systemów ortogonalnych (99,7% przypadków), co potwierdza wstępne założenia. Z przeprowadzonych analiz, dodatkowo wynika więc, że w sytuacji gdy współczynniki kwantowane są z większą dokładnością, istotne stają się błędy metod i działań wykorzystywanych w procesie syntezy filtru o danej strukturze. Zgodnie z tym, dalsze prowadzone prace i badania skupią się na poprawieniu kluczowych aspektów syntezy ortogonalnych filtrów rotatorowych w celu minimalizacji powstających niedokładności obliczeniowych i tym samym poprawie wrażliwości.

Bibliografia

1. A. Antoniou, "Digital Signal Processing: Signals, Systems, and Filters", McGraw Hill, New York (2005). ISBN: 0071454241
2. S.K. Mitra, J.F. Kaiser, "Handbook for Digital Signal Processing", John Wiley & Sons, New York (1993). ISBN: 0471619957
3. P. Poczekajło, K. Wawryn, "Algorithm for realisation, parameter analysis, and measurement of pipelined separable 3D finite impulse response filters composed of Givens rotation structures", IET Signal Processing, vol. 12 (2018), iss. 7, pp. 857-867. DOI: 10.1049/iet-spr.2017.0450
4. P. Poczekajło, R. Wirski, "Synthesis and Realization of 3-D Orthogonal FIR Filters Using Pipeline Structures", Circuits Systems and Signal Processing, vol. 37 (2018), iss. 4, pp. 1669-1691. DOI: 10.1007/s00034-017-0618-2

Streszczenie

W artykule przedstawiono wyniki pomiaru wrażliwości charakterystyki amplitudowej na kwantowanie współczynników dla różnych realizacji filtrów cyfrowych. Przeanalizowano filtry separowalne FIR 3D w realizacji bezpośredniej (splot) i ortogonalnej z zastosowaniem rotatorów. Pomiar wrażliwości sprowadzono do wyznaczenia błędu średniokwadratowego charakterystyki amplitudowej. Wykorzystano trzy poziomy kwantowania współczynników (8-, 16- i 24-bity) oraz

cztery rodzaje filtrów (dolnoprzepustowe, górnoprzepustowe, środkowoprzepustowe i środkowozaporowe).

Abstract

Measurements for sensitivity of amplitude characteristic on coefficients quantization for different realisation of digital filters is presented in the paper. Separable 3D filters in direct realisation (convolution) and orthogonal realisation with use rotation are analysed. Sensitivity is measured like a mean squared error of amplitude characteristic. Three quantization levels of coefficients (8-, 16- and 24-bits) and four type of filters (low-pass, high-pass, band-pass and band-stop) are used.

Keywords: FIR 3D filters, sensitivity, coefficients quantization, direct structure, orthogonal rotation structure

Rafał Wojszczyk*

Wydział Elektroniki i Informatyki
Politechnika Koszalińska

Piotr Stola

Quick-Solution, Koszalin

*rafal.wojszczyk@tu.koszalin.pl

Weryfikacja użycia metody oceny jakości implementacji wzorców projektowych w zastosowaniu produkcyjnym

Słowa kluczowe: wzorce projektowe, jakość oprogramowania, zwinne wytwarzanie oprogramowania

1. Wprowadzenie

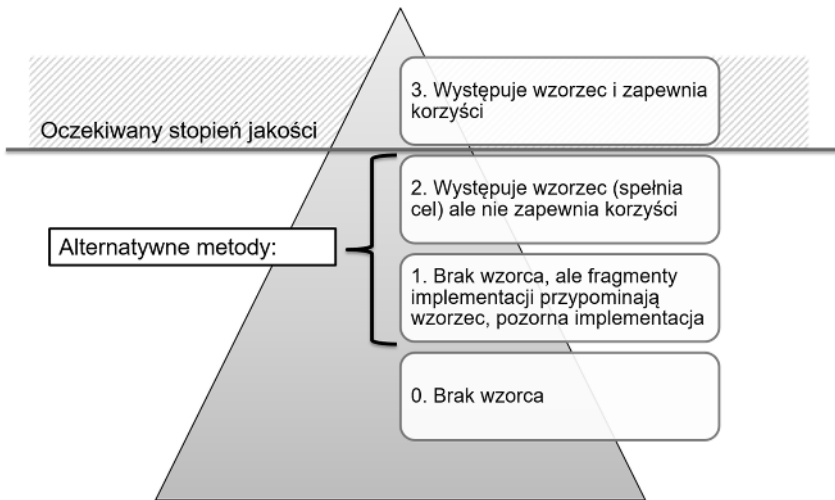
Programista implementując wzorce projektowe wykonuje to na podstawie przykładowych szablonów z [2] lub [5] oraz własnej wiedzy, w trakcie czego skupia się zazwyczaj na realizacji celu wzorca (rozwiązaniu problemu programistycznego). Realizacja celu wzorca zgodnie z szablonem z [2] nie oznacza jeszcze korzystnej implementacji, ponieważ każdy program komputerowy jest inny. Korzystna implementacja wzorca to taki fragment kodu źródłowego, który spełnia dodatkowe oczekiwania, inaczej zapewnia korzyści w wybranym kryterium. Przyjmując kryterium niskiego kosztu rozbudowy i integracji, oznacza to, że kod wzorców nie będzie wymagał dodatkowych modyfikacji przy rozbudowie i integracji z tym kodem. Zatem koszt rozbudowy będzie składał się z kosztu dodania nowych fragmentów kodu, które wykorzystują istniejącą implementację wzorca. W tym kontekście programista pracujący w zwinnym zespole po wykonaniu swojej pracy (napisaniu kodu źródłowego, zazwyczaj bez kompletnej dokumentacji) poszukuje odpowiedzi na pytanie: czy implementacja danego wzorca projektowego zapewni korzyści oczekiwane od tego wzorca? Odpowiedź na to pytanie umożliwia autorska metoda oceny jakości implementacji wzorców projektowych. Celem artykułu jest przedstawienie wyników z użycia metody w produkcyjnym środowisku.

W drugim rozdziale przedstawiono definicję jakości oprogramowania oraz alternatywne metody. Trzeci rozdział zawiera opis przeprowadzonego eksperymentu, co zostało poprzedzone krótkim opisem proponowanej metody. W czwartym rozdziale przedstawiono wyniki otrzymane w trakcie eksperymentu. Piąty rozdział stanowi podsumowanie pracy.

2. Jakość implementacji wzorców projektowych i alternatywne metody

2.1. Definicja jakości

Tekst Kryterium oceny jakości pod względem kosztu rozbudowy i integracji oprogramowania, jest jednym z najbardziej znaczących dla wytwórcy. Wytwórca, który nieprzerwanie utrzymuje i rozwija swój produkt programowy, nawet przez wiele lat, powinien zadbać o to, żeby koszty utrzymania i rozwoju były jak najniższe. W tym celu wykorzystywane są wzorce projektowe. Powszechnie przyjęło się, że programiści implementują wzorce na 2 stopniu jakości, tj. tak, aby implementacja spełniała wyłącznie postawiony cel wzorca, np. jedna instancja obiektu we wzorcu Singleton. Niepożądany jest 1 stopień jakości implementacji, taka implementacja nie spełnia celu i zawiera błędy, np. publiczny konstruktor klasy wzorca Singleton. Zarówno 1 i 2 stopień jakości implementacji nie zapewniają korzyści, które wyjaśniono we wstępie, zapewnia to dopiero implementacja na 3 stopniu jakości. Zestawienie wymienionych stopni jakości przedstawia rysunek 1. Pozostawienie przez wytwórcę implementacji na 1 lub 2 stopniu w produkcyjnym oprogramowaniu spowoduje dodatkowe koszty w przyszłości.



Rys. 1. Zestawienie stopni jakości implementacji wzorców projektowych

2.2. Alternatywne metody

Jakość kodu źródłowego powszechnie jest kojarzona z metrykami oprogramowania obiektowego. Niestety popularne metryki nie mają zastosowania do implementacji wzorców projektowych [3], pomimo kosztu użycia dopuszczalnego w zwinnych zespołach wytwórczych. Koszt użycia metody w takich zespołach jest bardzo ważny, ponieważ zwinne zespoły są często małe i nie dysponują dużym budżetem na utrzymanie jakości oprogramowania.

Pośród badań naukowych związanych z tym zagadnieniem dominuje problem wyszukiwania wystąpień wzorców projektowych [7][8]. Wynikiem działania metody wyszukującej wystąpienia jest liczba wystąpień wzorców w badanym fragmencie kodu programu lub odpowiedniku kodu. Jedno wystąpienie wzorca to w większości metod wyłącznie informacja o zgodności fragmentu kodu z szablonem opisującym referencyjny wzorzec, a następnie na podstawie tego fragment kodu jest klasyfikowany jako wystąpienie wzorca. Większość metod wyszukujących wystąpienia wzorców działa binarnie, tj. wskazuje wystąpienie wzorca lub brak wzorca, co odpowiada ocenie jako 2 lub 0 stopień jakości implementacji. Wybrane metody umożliwiają dodatkowo wskazanie niepełnego wystąpienia wzorca (np. zawiera błędy lub braki w implementacji), odpowiada to 1 stopniu jakości implementacji. Koszt użycia metod wyszukujących wystąpienia wzorców jest w większości przypadków akceptowalny przez zwinne zespoły wytwórcze.

Inne badania dotyczą metod weryfikacji implementacji wzorców, które polegają ponownie na wykazaniu zgodności badanego fragmentu kodu z szablonem wzorca projektowego [4][6]. Wynikiem działania metody weryfikującej implementację jest wskazanie fragmentu kodu, który jest zgodny z szablonem wzorca. Pełna zgodność z szablonem odpowiada 2 stopniu jakości implementacji, natomiast odstępstwa od tego odpowiadają 1 i 0 stopniu. Koszt użycia metod weryfikujących implementację wzorców jest większy niż możliwości zwinnego zespołu, ponieważ wymagana jest szczegółowa dokumentacja.

Reasumując, alternatywne metody nie pozwalają na odróżnienie implementacji zgodnej z 3 stopniem od 2 stopnia jakości, tj. nie można ocenić czy implementacja danego wzorca zapewnia oczekiwane korzyści, a w tym niższe koszty rozbudowy i integracji. Prowadzi to do sytuacji, że wytwórca nie wie jakie koszty poniesie w przyszłości.

3. Środowisko eksperymentu

3.1. Metoda oceny jakości implementacji wzorców projektowych

W eksperymencie wykorzystano autorską metodę, opisaną wcześniej w [9] [10]. Działanie metody rozpoczyna się od zbudowania modeli: kodu programu, wzorca i metryk. Następnie ekwiwalent kodu programu oceniany jest względem szablonu

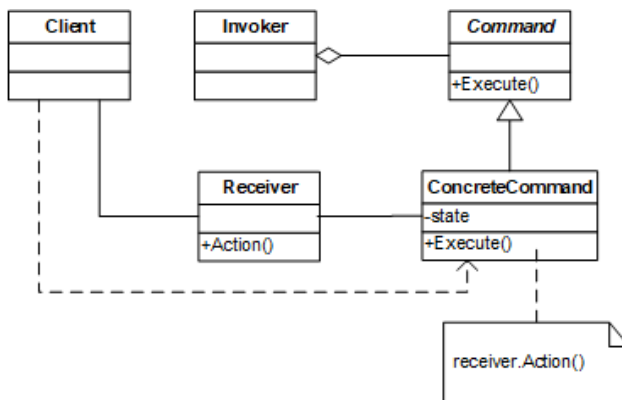
wzorca. Szablon wzorca opisany jest przez zbiór charakterystyk i uszczegóławiających je metryk, co stanowi analogię do znanych modeli jakości, np. ISO 9126. W innych słowach ocena polega na pomiarze odległości kodu programu od szablonu wzorca, następnie odległość jest interpretowana względem wybranego kryterium oceny. Wynik z interpretacji to odpowiedni stopień jakości implementacji. tekst.

3.2. Wykorzystane wzorce projektowe i oprogramowanie

Weryfikację metody przeprowadzono przy współpracy z firmą Quick-Solution z Koszalina, która udostępniła kod źródłowy. Kod składa się z 11 projektów (pakietów), 52 unikalnych klas, które zawierają łącznie 3374 wiersze kodu. Firma specjalizuje się w wytwarzaniu oprogramowania na zamówienie w technologii Microsoft .NET. Podstawowy obszar wytwarzanego oprogramowania to aplikacje webowe, mobilne i desktopowe. Zespół pracuje według metodyki Scrum, nad czym czuwa lider zespołu, który jest jednocześnie Scrum masterem.

Eksperyment przeprowadzono z wykorzystaniem wzorców Polecenie (ang.: Command) oraz Fabryka (ang.: Factory), które należą do jednych z popularniejszych wzorców [1].

Celem wzorca Polecenie jest [2]: *kapsulkowanie żądań w formie obiektu. Umożliwia to parametryzację klienta przy użyciu różnych żądań oraz umieszczanie żądań w kolejkach i dziennikach, a także zapewnia obsługę cofania operacji.* Implementacja wzorca jest przydatna, gdy na jednym obiekcie (np. konto bankowe) można wykonać wiele różnych operacji. Rysunek 2 przedstawia diagram klas z przykładową implementacją wzorca, na podstawie [2]. Diagram z rysunku 2 przedstawia wariant strukturalny, modyfikacją tego wariantu jest wariant z dynamicznym mapowaniem (połączenia w klasie Client są tworzone dynamicznie, np. przez mechanizm refleksji lub wstrzykiwania zależności).

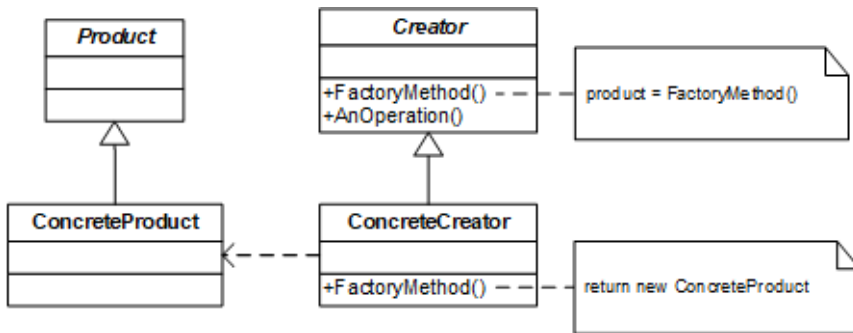


Rys. 2. Diagram klas wzorca polecenie

Elementy, z których składa się wzorec Polecenie to [2]:

- Abstrakcyjna klasa *Command* – deklaruje wspólny punkt do wykonywania operacji, inne nazwy: klasa nadrzędna, typ nadrzędny, ogólne polecenie,
- *ConcreteCommand* – obejmuje implementację operacji *Execute* o postaci wywołania odpowiednich operacji obiektu *Receiver*, inne nazwy: konkretne polecenie, klasa podrzędna,
- *Receiver* – wykonuje konkretne polecenie (algorytm), inne nazwy: odbiorca,
- *Client* – tworzy obiekty konkretnych poleceń i określa połączenia (mapuje) z odbiorcami, inne nazwy: mapa, mapowanie połączeń,
- *Invoker* – żąda obsłużenia polecenia, inne nazwy: nadawca.

Celem wzorca Fabryka jest [2]: *określenie interfejsu do tworzenia obiektów, przy czym wzorec umożliwia podklasom wyznaczenie klasy danego obiektu (proces tworzenia przekazywany do podklas)*. Implementacja wzorca jest przydatna, gdy z jednej operacji mogą powstawać różne obiekty niosące informacje. Rysunek 3 przedstawia diagram klas przykładowej implementacji wzorca, na podstawie [2] oraz [5]. W [2] wzorce Fabryk, tj. Fabryka Abstrakcyjna (ang.: Abstract Factory) oraz Metoda Wytwórcza (ang.: Factory Method) opisano oddzielnie, choć są zaliczane do jednej grupy. W praktyce jednak programiści unifikują te wzorce i określają jako dwa warianty wzorca Fabryka.



Rys. 3. Diagram klas wzorca fabryka

Elementy, z których składa się wzorec Fabryka to [2]:

- *Product* – deklaruje interfejs obiektów generowanych przez fabrykę, inne nazwy: produkt,
- *ConcreteProduct* – zawiera implementację interfejsu klasy *Product*, inne nazwy: produkt konkretny,

- *Creator* – zawiera deklarację metody wytwórczej, która zwraca obiekty typu *Product*, inne nazwy: wytwórca,
- *ConcreteCreator* – przesłania metodę z *Creator*, tak aby zwracać egzemplarz klasy *ConcreteProduct*, inne nazwy: wytwórca konkretny.

Wykorzystanie wzorca Fabryka zostało zaplanowane oraz zaprojektowane przez programistów rozwijających aplikację. Wzorzec został opisany w dokumentacji projektowej, która zawierała wyłącznie wybrane fragmenty aplikacji. Następnie wzorzec został zaimplementowany w jednej z pierwszych iteracji zgodnie z wiedzą i doświadczeniem zespołu. Natomiast wykorzystanie wzorca Polecenie nie było początkowo planowe, dlatego wzorzec został zaimplementowany w ramach refaktoryzacji. Zmiany wynikające z refaktoryzacji nie zostały naniesione na dokumentację projektową, co spowodowało rozbieżności pomiędzy kodem źródłowym a pierwotną dokumentacją. Eksperyment przeprowadzono w iteracji, w której wykonano refaktoryzację do wzorca Polecenie. W planie wydania (tzw. backlog) późniejszych iteracji (względem iteracji, w której wykonano eksperyment) zaplanowana jest rozbudowa oprogramowania, co wymaga wykorzystania implementacji wymienionych wzorców projektowych.

Eksperyment składał się z 3 etapów. W pierwszym etapie przeprowadzono tzw. „burzę mózgów” wspólnie z zespołem firmy Quick-Solution, której celem było opracowanie referencyjnych implementacji obu wykorzystanych wzorców. Opracowane implementacje spełniają 3 stopień jakości, tzn. realizują oczekiwany cel wzorca oraz zapewniają niski koszt rozbudowy. W przypadku wzorca Polecenie polega to na bez kosztowym dodawaniu nowego rodzaju polecenia. W przypadku wzorca Fabryka wynika to dodania nowych rodzajów obiektów do istniejących operacji oraz utworzenie nowych operacji, które mogą generować różne obiekty. Następnie w drugim etapie odwzorowano referencyjne implementacje do wymaganych modeli wykorzystywanych przez metodę. Natomiast 3 etap stanowi bezpośrednio proces oceny [11].

4. Wyniki eksperymentu

Wyniki oceny implementacji wzorców projektowych uzyskane w trakcie eksperymentu przedstawia tabela 1, są to wyłącznie fragmenty kodu poniżej ocenione 3 stopnia jakości implementacji (dla wzorca Polecenie od P1 do P4, wzorzec Fabryka – F1).

Tab. 1. Wyniki oceny implementacji uzyskane w trakcie eksperymentu

Nr usterki	Element w kodzie źr	Stopień jakości	Usterka i sugestia zmian	Konsekwencje
P1	Command	2	Rodzaj typu to klasa, klasę należy zastąpić interfejsem	Ograniczona możliwość dziedziczenia w poleceniach konkretnych (niektórego polecenia mogą wymagać dziedziczenia z wybranej klasy, np. powiązane z kontekstem ORM, ang. object-relational mapping)
P2	Concrete-Command-Execute (int pam)	2	Występuje metoda o podobnej sygnaturze, należy zmienić nazwę metody	Istnieje ryzyko użycia przez programistę innej metody Execute niż oczekiwana
P3	Client	2	Nie wszystkie polecenia konkretne są uwzględnione w mapie połączeń, należy dodać brakujące polecenia do mapy	Niewykorzystane polecenia konkretne mogą zostać usunięte, lub ponownie zaimplementowane
P4	Invoker1, Invoker2,	1	Występuje pojedyncze wywołanie operacji tzw. in-line (pominięto deklarację typu nadrzędnego), wywołanie należy poprzedzić deklaracją ICommand	Zaburza ideę użycia wzorca Polecenie, ogranicza elastyczność kodu, wykonanie wybranych poleceń jest poza kontrolą wzorca Polecenie
F1	Serialize Factory. Get Serializer	2	Modyfikator wewnętrzny (ograniczony zasięg dostępności metody), należy zmienić modyfikator dostępu na publiczny	Ograniczona dostępność fabryki, nie będzie dostępna poza pakietem (ryzyko ponownej implementacji)

W przypadku wzorca Polecenie wystąpiło kilka usterek. Przepuszczalnie wiele z nich nie zostałyby poprawionych w ramach dalszych prac, co w przypadku rozbudowy oprogramowania o nowe funkcje w przyszłości oznacza większy nakład

pracy. Dzięki wykorzystaniu metody, usterki mogą zostać poprawione jeszcze w trakcie trwania iteracji, w której wykonano eksperyment. Sumarycznie około 10% kodu wzorca jest poniżej 3 stopnia jakości implementacji.

W przypadku wzorca Fabryka (który został wcześniej zaprojektowany) wystąpiła zaledwie jedna usterka, związana z modyfikatorami dostępu. Tego typu usterki wynikają często z przeoczenia osób implementujących i przypuszczalnie zostałyby sukcesywnie naprawiane w ramach innych prac z kodem, jednakże byłoby to znacząco rozciągnięte w czasie. Dzięki ocenie jakości implementacji tego wzorca, wykryte usterki mogą być naprawione odpowiednio wcześniej. Sumarycznie zaledwie 2% kodu wzorca jest poniżej 3 stopnia jakości implementacji.

Koszt zmian do wykonania wynikających z wykrytych usterek jest niewielki w przypadku wzorca Fabryka. Programiści uznali, że usterkę są w stanie sami zauważyć i poprawić „przy okazji” w trakcie innych prac. Znacznie więcej zainteresowania wzbudziły wyniki oceny jakości wzorca Polecenie. W opinii zespołu Quick-Sotulion *„zlokalizowanie tych błędów na początku fazy implementacji jest niezwykle efektywne i zaoszczędza znacznie więcej czasu niż w przypadku, gdy kod zostałby wydany w wersji produkcyjnej”*. Koszt prac bez użycia proponowanej metody zespół Quick-Solution oszacował na 20 rbh, co zawierało czas poświęcony na szukanie usterek, testowanie, prace implementacyjne. Do szacowania wykorzystano mieszaną metodę, bazującą na ocenie eksperta oraz szacowania przez analogię. Po zapoznaniu się z wynikami oceny jakości zespół oszacował koszt prac na 6 rbh (prace implementacyjne), zatem szacunkowa oszczędność w rozbudowie oprogramowania wynosi 14 rbh. Przedstawiona czasochłonność nie uwzględnia czasu potrzebnego na opracowanie referencyjnych implementacji, jednakże jest to koszt jednorazowy. Raz opracowane modele będą wykorzystywane wielokrotnie w produkcyjnym zastosowaniu metody.

5. Podsumowanie

W pracy omówiono krótko koncepcję jakości implementacji wzorców projektowych oraz trzy stopnie jakości, do których można zaklasyfikować wystąpienia wzorców. Przedstawiono również problem oceny jakości implementacji wzorców z punktu widzenia programisty w zwinnym zespole wytwórczym. Problem ten wymagał opracowania autorskiej metody. W pracy przedstawiono wyniki z weryfikacji metody w produkcyjnym zastosowaniu. W trakcie weryfikacji wykazano, że użycie metody pozwala zaoszczędzić 14 rbh. Bez używania metody zespół wytwórczy musiałby liczyć się z 20 rbh ukrytych kosztów. Przyjmując, że budżet roboczogodzin jednego pracownika w trakcie dwutygodniowej iteracji wynosi 70 rbh (średnio 7 rbh dziennie * 10 dni), to dzięki metodzie zaoszczędzono 20% czasu pracy jednego pracownika. Uzyskany wynik potwierdza, że opracowana

metoda ma potencjał w praktycznym zastosowaniu, co pomoże w obniżeniu wybranych kosztów w wytwarzaniu oprogramowania.

Dalsze prace zakładają dopracowanie narzędzia wspierającego metodę oraz integrację z środowiskami IDE. Planowane jest również rozszerzenie metody o nowe rodzaje wzorców oraz dostosowanie do innych popularnych rozwiązań należących do tzw. dobrych praktyk.

Bibliografia

1. Czynzyn-Egird D., Wojszczyk R.: Determining the Popularity of Design Patterns Used by Programmers Based on the Analysis of Questions and Answers on Stackoverflow.com Social Network, Springer, 2016
2. Gamma E. i inni: WZORCE PROJEKTOWE Elementy oprogramowania wielokrotnego użytku. Helion, Gliwice, 2010.
3. Khaer Md. A. i inni: An Empirical Analysis of Software Systems for Measurement of Design Quality Level Based on Design Patterns, Computer and Information Technology
4. Mehlitz P.C., Penix J., Design for Verification Using Design Patterns to Build Reliable Systems, 2003
5. Metsker S. J.: C# Wzorce projektowe, Helion, Gliwice 2005
6. Nicholson J. i inni, Automated verification of design patterns: A case study, Elsevier 2013
7. Tsantalis N. i inni: Design Pattern Detection Using Similarity Scoring. IEEE Transactions on Software Engineering, Volume: 32, Issue: 11, s. 896-908, 2006, IEEE, 2007
8. Singh Rao R., Gupta M., Design Pattern Detection by Sub Graph Isomorphism Technique, International Journal Of Engineering And Computer Science, Volume 2 Issue 11 Page No. 3101-3105, 2013
9. Wojszczyk R., Wójcik R.: The Model of Quality Assessment of Implementation of Design Patterns, w: Omatu S. et al. (eds) Distributed Computing and Artificial Intelligence, 13th International Conference, Advances in Intelligent Systems and Computing, Vol. 474, strony od515 do 524, Springer, 2016
10. Wojszczyk R.: The model and function of quality assessment of implementation of design patterns, w: Applied Computer Science, Vol. 11, No. 3, strony od 45 do 56, Politechnika Lubelska, Lublin, 2015
11. Wojszczyk R.: The Experiment with Quality Assessment Method Based on Strategy Design Pattern Example, w: Świątek J., Borzemski L., Wilimowska Z. (eds) Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology –

ISAT 2017. Advances in Intelligent Systems and Computing, Vol. 656, strony od 103 do 112, Springer, 2018

Streszczenie

Programiści pracujący zgodnie z metodykami zwinnymi bardzo chętnie implementują wzorce projektowe w trakcie swojej pracy. Często robią to bazując wyłącznie na własnym doświadczeniu, co nie uwzględnia dodatkowych korzyści, których można oczekiwać po implementacji wzorców.

Celem artykułu jest przedstawienie przebiegu i wyników z weryfikacji metody oceny wzorców projektowych w zastosowaniu produkcyjnym. Eksperyment przeprowadzono wspólnie z zewnętrzną firmą, pracującą według zwinnej metodyki wytwórczej.

Abstract

Developers working according to agile methodologies are very happy to implement design patterns in their work. They often do so based solely on their own experience, which does not take into account the additional benefits that can be expected from implementing patterns.

The purpose of the article is to present the mileage and results of verification of the method of assesment design patterns in production. The experiment was conducted jointly with an external company, working according to a agile manufacturing methodology.

Keywords: design patterns, software quality, agile development

Ewelina Hałat*

Sławomir Kukła

Katedra Inżynierii Produkcji

Akademia Techniczno-Humanistyczna w Bielsku-Białej

*ewelinahalat130695@gmail.com

Wykorzystanie narzędzi do modelowania i symulacji w racjonalizacji procesów wytwarzania wyrobów cukierniczych

Słowa kluczowe: Modelowanie i symulacja, racjonalizacja procesów wytwarzania, przemysł spożywczy

1. Wprowadzenie

Coraz częściej obserwuje się potrzebę doskonalenia procesów produkcyjnych. Ciągłe wymagania rynkowe, konkurencyjność oraz rozwój technologii sprawiają, że przedsiębiorstwa stosują w swojej działalności narzędzia i metody odchudzonej produkcji, które pozwalają racjonalizować procesy wytwarzania, rozwiązywać problemy oraz efektywnie i skutecznie osiągać wyznaczone cele.

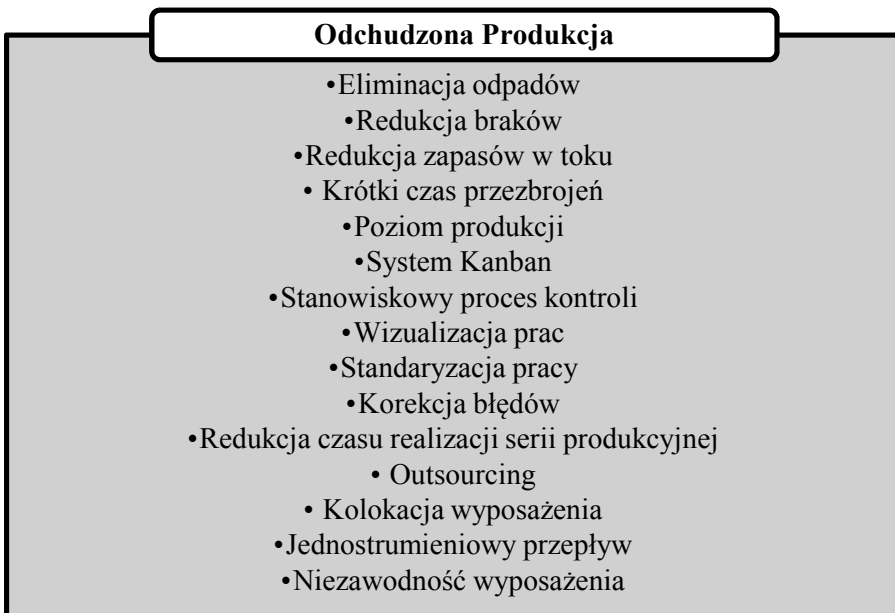
Przedsiębiorstwa muszą udoskonalać swoją działalność z powodu zmieniającego się otoczenia oraz z powodu silnej konkurencji. Ważnym czynnikiem jest szybki rozwój technologii informatycznych oraz wychodzenie na przeciw wobec oczekiwań klientów. Przedsiębiorstwa mają szansę doskonalić zarządzanie jak i funkcjonowanie procesów produkcyjnych poprzez zastosowanie odpowiednich metod i narzędzi wspomagających zarządzanie. Pojawiające się w firmie problemy mogą być rozwiązane za pomocą tych koncepcji. Pozytywne skutki z ich wdrożenia pokazują, iż wykorzystywanie nowoczesnych narzędzi i metod stało się niezbędne [9, 10].

Podstawą rozwoju innowacyjności produkcji staje się umiejętność wykorzystywania narzędzi wspomagania komputerowego na etapie projektowania systemów produkcyjnych. Coraz częściej do projektowania i analizy systemów wykorzystuje się modelowanie 3D oraz symulację komputerową. Zastosowanie

narzędzi informatycznych do symulacji i wizualizacji umożliwia analizowanie i zarządzanie systemami produkcyjnymi na podstawie modelu komputerowego zwanego cyfrową fabryką [7].

Komputerowy model systemu produkcyjnego musi wiernie odwzorowywać istotne elementy rzeczywistego systemu oraz ich wzajemne powiązania. Model pozwala na różne testy i symulacje, co wpływa na zmniejszenie kosztów funkcjonowania systemów wytwórczych. Model cyfrowej fabryki ułatwia i wspomaga planowanie i doskonalenie procesów na różnych poziomach zarządzania organizacją [4].

W przedsiębiorstwach produkcyjnych coraz częściej wykorzystuje się zasady szczupłej produkcji w doskonaleniu logistyki przepływów materiałowych. Szczupłe wytwarzanie ma przede wszystkim na uwadze klienta, wytwarzanie produktów oraz wykonywanie procesów o jak największej jakości i wartości. Ma na celu wyeliminowanie strat, wykrycia marnotrawstwa, które w żaden użyteczny sposób nie wnosi wartości do wytwarzanych wyrobów [8, 11]. Na rysunku numer 1 przedstawiono najważniejsze zasady odchudzonej produkcji.



Rys. 1. Najważniejsze zasady odchudzonej produkcji [11]

Przy produkcji żywności niezbędne jest utrzymanie wysokiej jakości. Konsumentom wymagają wiele i będą oczekiwać jeszcze więcej od producentów żywności. Jakość żywności to cechy i kryteria, które pozwalają opisać produkty pod względem ich wartości odżywczej, a przede wszystkim bezpieczeństwa dla zdrowia człowieka. Najważniejszą kwestią jest zapewnienie bezpieczeństwa zdrowotnego wyrobu, co ściśle jest związane z wdrażaniem systemów zarządzania jakością. Przedsiębiorstwa coraz częściej wdrażają systemy oparte na zasadach analizy zagrożeń i podstawowych punktów kontroli, aby wspomagać działania dobrej praktyki higienicznej oraz dobrej praktyki produkcyjnej. Dzięki temu przedsiębiorstwa mogą osiągnąć sukces. Konsument kupując żywność przede wszystkim bierze pod uwagę bezpieczeństwo zdrowotne. Przy produkcji żywności należy wykorzystywać działania pro jakościowe, które będą wspierać jakość wyrobów spożywczych oraz ich bezpieczeństwo. Szczupłe wytwarzanie wymaga wykorzystania wielu narzędzi i metod zarządzania. Poprzez ich zastosowanie można polepszyć produktywność, wykryć i usunąć marnotrawstwo, a także usprawnić przebieg zachodzących w przedsiębiorstwie procesów [5, 6].

Szczupłe wytwarzanie daje wiele korzyści, zapewnia wysoką jakość produktów oraz charakteryzuje się niskimi kosztami wytwarzania. Przyczynia się także do tego, iż klienci są zadowoleni z nabytych produktów i są lojalni. Zmniejszenie zapasów oraz produkcji w toku umożliwia krótki okres dostaw. Kształcenie świadomości pracowników, odnośnie doskonalenia produkcji i eliminowania strat, zwiększa ich zaangażowanie w racjonalizację procesów wytwarzania [1].

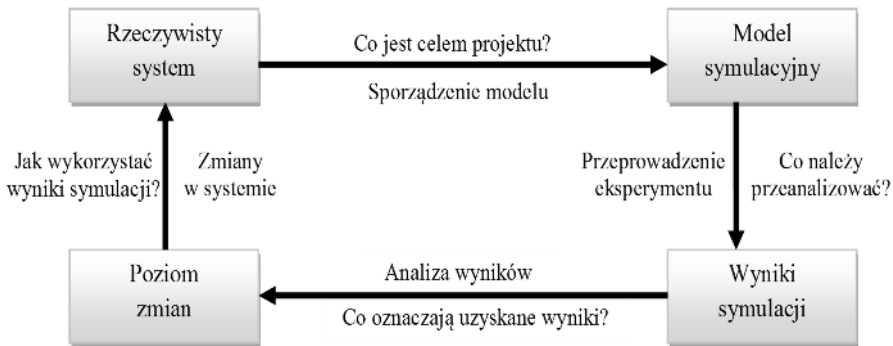
2. Modelowanie i symulacja systemów produkcyjnych

Racjonalizacja i optymalizacja procesów to działanie, które powinno być indywidualnie rozpatrywane ze względu na charakter przedsiębiorstwa, produkowane wyroby, środki produkcji itd. Firmy, które chcą pozyskać coraz więcej klientów, odnosić sukcesy, polepszać wydajność produkcji oraz jakość wyrobów powinny wykorzystywać modelowanie i symulację procesów wytwarzania [2].

Jednym z najbardziej skutecznych narzędzi, które pomaga w zatwierdzeniu decyzji podczas zarządzania jest nic innego jak modelowanie i symulacja (rys. 2). Metoda ta pozwala na badanie wszelakich wariantów procesu produkcyjnego poprzez odpowiednie opracowanie modelu komputerowego systemu. Modele symulacyjne umożliwiają śledzenie funkcjonowania określonego obiektu. Symulacja weryfikuje przyjęte założenia zanim wdroży się je w praktyce. Sprawdza także nieprawidłowości, które ewentualnie mogą pojawić się w systemie produkcyjnym. Dzięki symulacji możliwe jest sprawdzenie funkcjonowania zaprojektowanego systemu, a co za tym idzie, wykrycie jego „słabych punktów” [3, 10].

Podczas symulacji wykorzystywany jest model. Tworzony on jest w oparciu o rzeczywisty obiekt systemu produkcyjnego. Model komputerowy prezentuje obustronne relacje, a także zachowania każdego z elementów systemu. W oparciu o system (może on być istniejący lub projektowany) buduje się model komputerowy danego obiektu, który podlega analizie i na którym wykonywane są symulacje [3].

Technika ta pozwala na szybkie badanie różnorodnych wariantów rozwiązań wytwórczych, które generowane są za pomocą komputerowego modelu systemu. Metoda ta polega na opisie powiązań funkcjonalnych, które występują między fragmentami analizowanego systemu i czynnikami zewnętrznymi. Przebieg procesu przedstawiony jest w postaci graficznej przez animację [7].



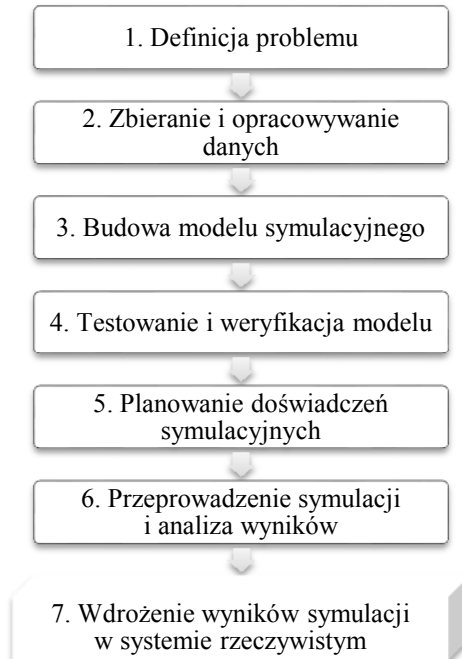
Rys. 2. Modelowanie i symulacja systemów produkcyjnych [7]

Na rysunku 3 przedstawiono etapy realizacji eksperymentu symulacyjnego. Pierwszym krokiem podczas projektu symulacyjnego jest określenie danych niezbędnych do konstrukcji modelu. Dane mogą dotyczyć: wyposażenia, czasów jednostkowych, czasów pomocniczych, czasów przebrojeń, parametrów technicznych itp. Pozyskanie tych danych może okazać się pracochłonne, a także drogie. Należy także brać pod uwagę dostępność i jakość pozyskiwanych danych, które mogą oddziaływać na skuteczność wykonywanych badań i doświadczeń [7, 10].

Kolejnym krokiem jest modelowanie. Podczas budowy modelu systemu ustala się granice systemu oraz jego połączenie z otoczeniem. Podczas modelowania wykorzystuje się różne moduły, które pozwalają na stworzenie różnych stanowisk pracy, a poprzez wykorzystanie specjalnego połączenia powstają wzajemne relacje pomiędzy nimi. Istotną rzeczą podczas projektowania modelu jest uwzględnienie tylko najważniejszych elementów czy informacji. Często model jest zbyt szczegółowy i nie jest związany z założeniami i celem projektu [7].

Po zbudowaniu modelu poddaje się go weryfikacji i sprawdza się jego poprawność. Konieczne jest porównanie wyników uzyskanych z symulacji z aktualnymi danymi systemu rzeczywistego.

Na zakończenie przedstawiane są efekty i inicjatywy zmian. Istnieje także możliwość powrotu do poszczególnych etapów w przypadku, gdy np. założenia ulegną zmianie lub gdy przeprowadzona analiza i osiągnięte rezultaty spowodują, że zmienią się cele projektu [10].



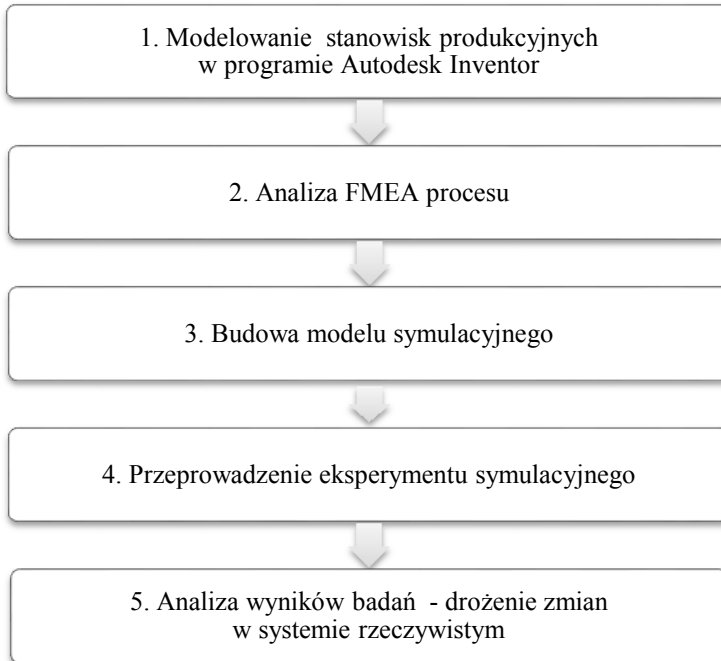
Rys. 3. Etapy realizacji eksperymentu symulacyjnego [4]

3. Charakterystyka obiektu badań, cel i zakres badań

Obiektem badań jest system wytwarzania wyrobów cukierniczych. Celem badań jest doskonalenie organizacji pracy na stanowiskach produkcyjnych związanych z wytwarzaniem dwóch gatunków ciast. Na przestrzeni ostatnich kilku lat firma przeprowadziła szereg niezbędnych remontów i inwestycji zakładu tak, aby móc oferować produkty o jak najwyższej jakości, wyśmienitym i tradycyjnym smaku. Wśród obszernego asortymentu cukierniczego firmy znajdują się między innymi: wyśmienite ciasta, ciastka oraz torty.

W ramach badań analizie poddano proces wykonywania sernika z brzoskwiniami oraz ciasta oreo. Wybrano te produkty, gdyż produkowane są w dużych ilościach oraz istotnie wpływają na funkcjonowanie przedsiębiorstwa cukierniczego.

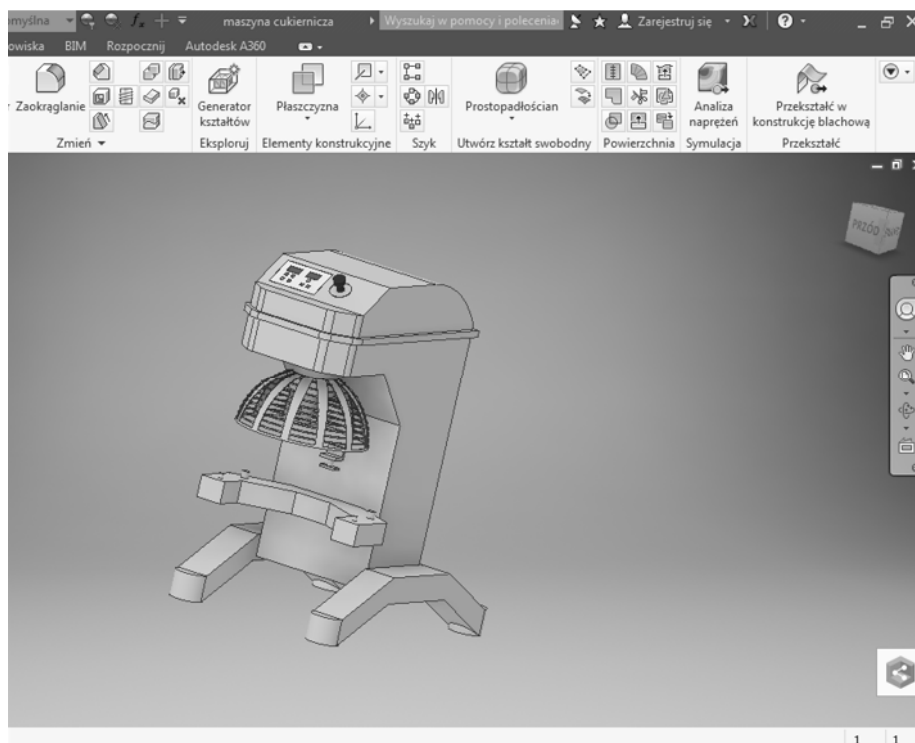
Badania zrealizowano zgodnie ze schematem postępowania przedstawionym na rysunku 4.



Rys. 4. Etapy realizacji badań [opracowanie własne]

4. Prezentacja wyników badań

W programie Autodesk Inventor zaprojektowano modele 3D wybranych stanowisk produkcyjnych dla wizualizacji zachodzących procesów oraz prowadzenia szkoleń dla pracowników. Na rysunku 5 przedstawiono przykładowy model maszyny cukierniczej.



Rys. 5. Model maszyny cukierniczej w programie Autodesk Inventor [opracowanie własne]

W dalszej kolejności przeprowadzono procedurę FMEA. Procedura FMEA była wykonana w oparciu o dane z ubiegłego roku. Jest ona na bieżąco aktualizowana wraz ze zmieniającymi się potrzebami klientów, reklamacjami oraz dysponowanymi maszynami na produkcji. W procedurze wyróżniono: znaczenie, występowanie oraz detekcje. Dla każdego z tych czynników przypisano wartości od 1 do 10, które następnie były potrzebne do przygotowania arkusza FMEA. Arkusz FMEA wyłonił główne wady, które przekroczyły określony wskaźnik RPN. Dzięki temu przedsiębiorstwo może teraz wprowadzić zmiany, co spowoduje obniżenie wartości wskaźnika do poziomu akceptowalnego. Poniżej w tabeli 1 przedstawiono działania, które przyczynią się do obniżenia wskaźnika RPN.

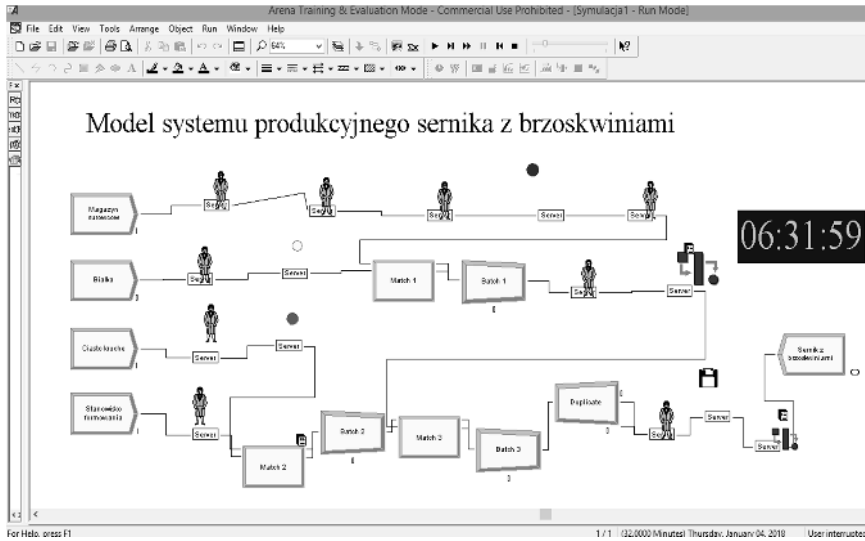
Tab. 1. Postępowanie z ryzykiem [opracowanie własne]

RPN	Zalecane działanie	Odpowiedzialny termin	Działania podjęte	Znaczenie	Występowanie	Wykrywalność	RPN
168	Przeprowadzenie dodatkowych szkoleń pracowników z zakresu obsługi pieca do pieczenia ciasta	Do 3 dni	Obowiązkowe szkolenie pracowników oraz kontrola ich pracy do 2 tygodnie	7	4	2	56
252	Kontrola stanu technicznego chłodni oraz kontrole stanu ważności żywności na magazynie	Do 1 dnia	Naprawa chłodni oraz wyznaczenie osoby odpowiedzialnej za kontrolę ważności żywności na magazynie	9	3	4	108

Przeprowadzenie analizy FMEA pozwoliło na zidentyfikowanie głównych wad w procesie pieczenia ciasta. Problem niedopieczonego ciasta został rozwiązany poprzez wprowadzenie szkoleń pracowników oraz kontrolowanie ich pracy. Wskaźnik RPN w tym przypadku został obniżony z wartości 168 do 56 – wartość akceptowalna. Nieodpowiednie składowanie surowców również zostało poddane działaniom korygującym. Wskaźnik RPN uległ zmianie z 252 na 108.

W celu określenia obciążenia zasobów produkcyjnych oraz przepływów materiałowych zbudowano model symulacyjny systemu w oprogramowaniu ARENA (rys. 6).

Po sprawdzeniu poprawności modelu przeprowadzono eksperyment symulacyjny sprawdzający funkcjonowanie systemu wytwarzania wybranego asortymentu ciast. Eksperyment zaplanowano na 9 godzin pracy, uwzględniono jednego pracownika wykonywającego prace ręczne oraz obsługującego poszczególne maszyny oraz urządzenia i przyjęto, że do pieca jednorazowo będzie wkładanych sześć blach z przygotowanym ciastem. W rezultacie przeprowadzonego eksperymentu symulacyjnego określono wydajność systemu (wyprodukowano 24 ciasta) oraz wyznaczono obciążenie poszczególnych zasobów produkcyjnych (rys. 7).



Rys. 6. Model systemu produkcyjnego sernika z brzoskwiniami [opracowanie własne]



Rys. 7. Wykorzystanie zasobów – sernik z brzoskwiniami [opracowanie własne]

Zauważono problemy organizacyjne wynikające z długiego okresu przygotowania pierwszej partii sześciu blach z ciastem na początku zmiany roboczej oraz długiego czasu wypieku ostatniej partii. Z pewnością byłyby możliwe wykonanie jeszcze jednej partii sernika, ale wtedy nie byłoby czasu, aby ją wypiec.

W dalszej kolejności przygotowano model symulacyjny związany z wytwarzaniem drugiego gatunku ciasta. W rezultacie przeprowadzonego

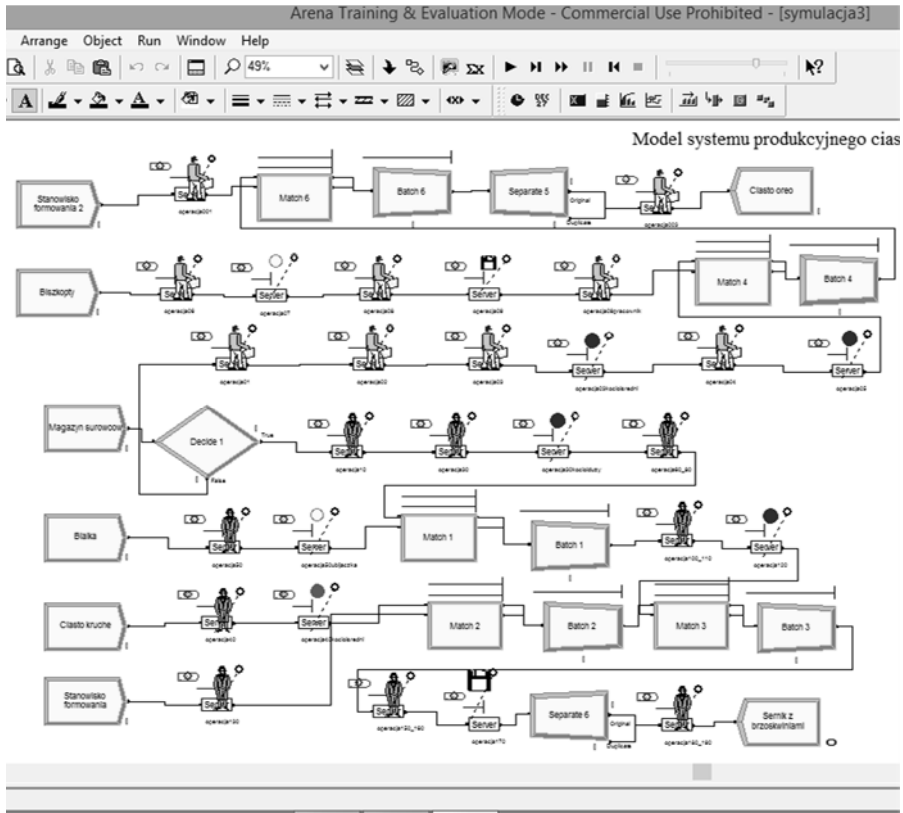
eksperymentu symulacyjnego otrzymano 48 ciast oreo. Ze zrealizowanej symulacji otrzymano również obciążenia poszczególnych zasobów (rys. 8).



Rys. 8. Wykorzystanie zasobów – ciasto oreo [opracowanie własne]

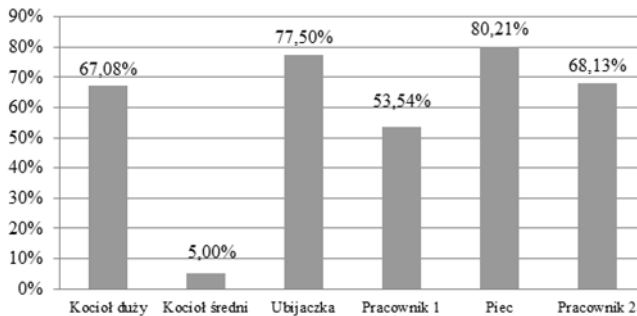
Zauważono, że poszczególne zasoby mogłyby być lepiej zagospodarowane, dlatego firma postanowiła połączyć proces wytwarzania sernika z brzoskwiniami z ciastem oreo tak, aby ciasta wykonywane były równolegle na tych samych zasobach. Spowoduje to lepsze wykorzystanie dostępnych zasobów, zwiększenie wydajności, a także obniżenie kosztów własnych wytwarzania. Model, po połączeniu procesów, został przedstawiony na rysunku nr 9.

Po sprawdzeniu poprawności modelu przystąpiono do przeprowadzenia eksperymentu symulacyjnego. Eksperyment trwał 8 godzin, wykorzystano dwóch pracowników (jeden z nich wykonuje oreo, a drugi sernik) oraz wzięto pod uwagę organizację na stanowiskach pracy już po wdrożeniu usprawnień. W wyniku przeprowadzenia symulacji uzyskano następujące wyniki: 30 ciast oreo oraz 18 serników z brzoskwiniami. Jest to bardzo dobry wynik. Symulacja jasno pokazała, że firma jest w stanie wykonać 30 ciast oreo oraz 18 serników. Wcześniej przed zmianami wykonywała tylko po trzy partie tych ciast. Wdrożenie metod takich jak 5S, SMED FMEA pozwoliło zoptymalizować proces i dodatkowo wykonać jeszcze 2 partie ciasta oreo, co zostało uwidocznione podczas modelowania i symulacji. Dzięki połączeniu procesów wytwarzania sernika z brzoskwiniami i ciasta oreo lepiej zostały wykorzystane zasoby takie jak: piec, maszyna mieszająca z kotłem dużym oraz ubijaczka z różgą. Rysunek 10 przedstawia wykorzystanie zasobów.



Rys. 9. Model systemu produkcyjnego w programie Arena dla dwóch gatunków ciast [opracowanie własne]

Wykorzystanie zasobów - sernik z brzoskwiniami oraz oreo



Rys. 10. Wykorzystanie zasobów – ciasto oreo oraz sernik z brzoskwiniami [opracowanie własne]

5. Podsumowanie i wnioski

Narzędzia odchudzonego wytwarzania dają wiele korzyści m.in. możliwość redukcji kosztów, skrócenie czasu produkcji czy też poprawa jakości wyrobu. W przedsiębiorstwie cukierniczym również można z powodzeniem zastosować różne narzędzia i metody, aby ciągle doskonalić procesy produkcyjne.

W wyniku przeprowadzonych analiz dotyczących wytwarzania sernika z brzoskwiniami poprawiono znacznie wydajność, warunki pracy oraz świadomość pracowników.

Projektowanie oraz analiza systemów wytwórczych wymaga podejmowania mnóstwa decyzji projektowych. Technika modelowania i symulacji może być jednym ze skutecznych narzędzi wspomagających podejmowanie decyzji na różnych poziomach zarządzania firmą również w branży spożywczej.

Modelowanie stanowisk produkcyjnych w 3D może zostać wykorzystane do wizualizacji oraz reorganizacji pracy na stanowiskach przygotowania ciast, a także w szkoleniach pracowników.

Dzięki zastosowaniu modelowania i symulacji systemów produkcyjnych firma sprawdziła różne scenariusze rozwiązań na modelu komputerowym. Na podstawie wyników eksperymentu symulacyjnego oszacowano czas realizacji zleceń produkcyjnych i wykorzystano lepiej dostępne zasoby. Oprócz tego uzyskane wykresy pozwolą firmie dokładniej przeanalizować funkcjonowanie procesów oraz lepiej zrozumieć zjawiska zachodzące w systemie wytwórczym. Realizacja eksperymentu symulacyjnego na modelu komputerowym eliminuje konieczność eksperymentowania na systemie rzeczywistym i pozwala wykrywać problemy mogące wystąpić w praktyce produkcyjnej. Dzięki symulacji można sprawdzić wydajność projektowanego systemu oraz analizować wykorzystanie dostępnych zasobów produkcyjnych już na etapie projektowania.

Literatura

1. Antosz K., Pacana A., Stadnicka D., Zielecki W.: *Lean Manufacturing doskonalenie produkcji*, Wydawnictwo Politechniki Rzeszowskiej, Rzeszów 2016, ISBN 978-83-7934-058-3
2. Ciszak O., *Komputerowo wspomagane modelowanie i symulacja procesów produkcyjnych*, Zeszyty Naukowe Politechniki Poznańskiej, Nr 6, Budowa Maszyn i Zarządzanie Produkcją 2007
3. Hałat E., Kukła S.: *Improvement and visualization of manufacturing processes in the food industry*, Wydawnictwo Fundacji Centrum Nowych Technologii, InvEnt 2017
4. Hamrol A.: *Zarządzanie jakością z przykładami*, PWN, Warszawa 2005, ISBN 83-01-14486-6

5. Kmiećkowiak A., Lesiów T.: *Systemy zarządzania jakością i ich integracja w przemyśle żywnościowym – praca przeglądowa*, Wydawnictwo Uniwersytetu Ekonomicznego we Wrocławiu, Wrocław 2009
6. Ładoński W., Szołtysek K.: *Zarządzanie jakością, część 1 Systemy jakości w organizacji* Wydawnictwo Akademii Ekonomicznej im. Oskara Langego we Wrocławiu, Wrocław 2005, ISBN 83-7011-767-8
7. Maciąg A., Pietroń R., Kukła S.: *Prognozowanie i symulacja w przedsiębiorstwie*, Polskie Wydawnictwo Ekonomiczne, Warszawa 2013, ISBN 978-83-208-2061-4
8. Pająk E., Klimkiewicz M., Kosieradzka A.: *Zarządzanie produkcją i usługami*, Polskie Wydawnictwo Ekonomiczne, Warszawa 2014, ISBN 978-83-208-2149-9
9. Pałucha K.: *Nowoczesne metody w zarządzaniu przedsiębiorstwem*, Politechnika Śląska Wydział Organizacji i Zarządzania Instytut Zarządzania i Administracji, Zeszyty Naukowe Politechniki Śląskiej 2012
10. Plinta D.: *Modelowanie i symulacja procesów produkcyjnych*, Wydawnictwo Naukowe Akademii Techniczno – Humanistycznej w Bielsku – Białej, Bielsko-Biała 2015, ISBN 978-83-65182-23-4
11. Metody i narzędzia Lean Production i ich rola w kształtowaniu innowacji w przemyśle,
http://www.ptzp.org.pl/files/konferencje/kzz/artyk_pdf_2013/p049.pdf, 05.09.2017

Streszczenie

W artykule zaprezentowano zagadnienia związane z doskonaleniem systemów produkcyjnych w przemyśle spożywczym na przykładzie procesów wytwarzania wyrobów cukierniczych. Dla wizualizacji analizowanych problemów wykonano modele 3D wybranych stanowisk produkcyjnych w programie Autodesk Inventor. Zbudowano komputerowy model symulacyjny systemu wytwarzania w pakiecie Arena, na którym przeprowadzono eksperyment. Wyniki eksperymentu poddano analizie i zaproponowano usprawnienia w celu doskonalenia realizowanych procesów.

Summary

The article presents issues related to the improvement of production systems in the food industry on the example of confectionery manufacturing processes. 3D models of selected production positions in Autodesk Inventor were made to visualize the analyzed problems. A computer simulation model of the production system was built in the Arena package on which the experiment was carried out.

The results of the experiment were analyzed and improvements were suggested in order to improve the processes carried out.

Keywords: Modeling and simulation, rationalization of manufacturing processes, food industry