

AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki

Praca magisterska

Grzegorz Miklaszewski

Stanisław Magierski

kierunek studiów: automatyka i robotyka

specjalność: informatyka w sterowaniu i zarządzaniu

Samoadaptacyjny sklep internetowy obsługiwany przez inteligentnego cybersprzedawcę realizującego postulaty CRMu

Opiekun: dr Adrian Horzyk

Ocena: Data: Podpis:

Kraków, czerwiec 2009

Oświadczamy, świadomi odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonaliśmy osobiście i samodzielnie i nie korzystaliśmy ze źródeł innych niż wymienione w pracy.

Spis treści

1	Wstęp	5
2	Wprowadzenie teoretyczne w tematykę pracy	7
2.1	Rozwój Internetu	7
2.1.1	Web 2.0	7
2.1.2	Numerowanie Internetu	9
2.1.3	Web 3.0	9
2.1.4	Spółeczność użytkowników w dobie trzeciej generacji Internetu	11
2.1.5	Podsumowanie	12
2.2	CRM	12
2.2.1	Historia CRM	12
2.2.2	Co to jest Customer Relationship Management	14
2.2.3	Cele stawiane systemom CRM	16
2.2.4	Architektura, rodzaje systemów CRM	16
2.3	Obsługa klienta	17
2.3.1	Kontakt sprzedawcy z klientem	18
2.3.2	Negocjacje	21
2.3.3	Typologia charakterów klientów	23
2.3.4	Podsumowanie	27
2.4	Chatbot	28
2.4.1	Co to jest chatbot	28
2.4.2	Historia chatbotów	28
2.4.3	Opis A.L.I.C.E i AIML	29
2.4.4	Nowoczesne zastosowania chatbotów	36
2.5	Ruby on Rails	37
2.5.1	Język Ruby	37
2.5.2	Środowisko Ruby on Rails	39
3	Opis systemu	40
3.1	Architektura systemu	40
3.1.1	Wzorzec MVC	41
3.2	Konwersacja	42
3.3	Mechanizm chatbota (automat Mealy'go)	42
3.3.1	Stan	45
3.3.2	Graf przejść między stanami (funkcja przejścia Φ)	46
3.3.3	Zarządca	47
3.3.4	Poruszanie się po grafie stanów	47
3.3.5	Wypowiedź klienta	48
3.3.6	Odpowiedź chatbota	48
3.3.7	Przechowywanie informacji zebranych o kliencie	48
3.4	Rozpoznawanie wypowiedzi klienta	49
3.4.1	Cechy produktów	49
3.4.2	Wymagania klienta odnośnie produktu	52
3.4.3	Wzorce dopasowań	53
3.4.4	Mechanizm synonimów	55
3.4.5	Algorytm odczytywania wypowiedzi klienta	55
3.4.6	Aktualny kontekst konwersacji	57
3.5	Konstruowanie odpowiedzi	58

3.6	Dodatkowe funkcjonalności cybersprzedawcy	60
3.6.1	Rozpoznawanie płci klienta	60
3.6.2	Baza telefonów	60
3.6.3	Baza imion polskich	61
3.7	Zastosowane Technologie	61
3.7.1	MySQL	61
3.7.2	Ruby on Rails	62
3.7.3	Użyte wtyczki	66
3.7.4	Substruct	68
3.7.5	Biblioteka lingwistyczna CLP	70
3.7.6	Syntezytor mowy Ivona	71
4	Prezentacja działania systemu	72
4.1	Panel administratora	72
4.2	Interfejs użytkownika	73
4.3	Schemat rozmowy	76
4.3.1	Algorytm rozmowy	77
4.3.2	Tor główny rozmowy	77
4.3.3	Tor poboczny rozmowy	81
4.3.4	Rozpoznawanie charakteru klienta	83
4.4	Przykłady rozmów	89
5	Podsumowanie	93

1 Wstęp

Celem niniejszej pracy było stworzenie przyjaznego dla klienta sklepu internetowego. Pod pojęciem „przyjazny” rozumiano pewną funkcjonalność aplikacji, dzięki której potrzeby użytkownika zostałyby jak najlepiej zaspokojone.

Głównym elementem tego typu wyróżniającym planowany sklep miał być cybersprzedawca. W tradycyjnych sklepach internetowych użytkownik jest zmuszony do samodzielnego przeszukiwania asortymentu i przeglądania parametrów poszczególnych produktów. Cybersprzedawca potrafiłby prowadzić rozmowę z klientem na temat jego wymagań odnośnie poszukiwanego produktu, a także dostępnego asortymentu sklepu. W ten sposób klient miałby wrażenie, iż prowadzi konwersację z prawdziwym handlowcem, jak to ma miejsce w rzeczywistych sklepach. Dodatkowo, cybersprzedawca posiadałby zdolność rozpoznawania charakteru klienta na podstawie sposobu doboru słów w wypowiedzi. Wiedza ta, dałaby mu możliwość odpowiedniego prowadzenia konwersacji i prezentacji w sposób najbardziej przystępny dla rozmówcy. Klient czułby satysfakcję z rozmowy niekoniecznie zdając sobie sprawę z przyczyny.

Aplikacja miała także spełniać wymagania stawiane przed systemami *Customer Relationship Management* (CRM), zajmującymi się zarządzaniem relacjami z klientami. Ich cechy przedstawiają się następująco:

- Możliwość pozyskiwania wiedzy o kliencie
- Klasyfikowanie klientów ze względu na ich zachowania
- Pozyskiwanie nowych klientów
- Utrzymanie klienta
- Obsługa klienta

Szczegółowy opis aplikacji systemów CRM znajduje się w kolejnych rozdziałach.

Autorzy wybrali telefony komórkowe jako produkty oferowane przez tworzoną aplikację. Powodem takiej decyzji była przede wszystkim ustandaryzowana specyfikacja większości modeli tego typu produktów. Można wybrać pewien podzbiór parametrów, który będzie wspólny dla wszystkich modeli telefonów komórkowych. Dodatkowo, są to produkty powszechne na rynku, lecz wiele osób ma trudności z dokonaniem ostatecznego wyboru.

Rozdział 2 ma za zadanie przybliżyć czytelnikowi poszczególne zagadnienia, bezpośrednio związane z tworzoną aplikacją. Znajduje się tam przede wszystkim opis historii Internetu, czyli jak to się stało, że w ciągu kilkunastu lat wyewoluował on do obecnej postaci i jaka jest jego przyszłość. Rozdział wprowadzający szczegółowo traktuje na temat systemów CRM, a także związaną z nimi obsługą klienta, w ramach której będzie omówiona typologia programów osobowości (charakterów) klienta. Następnie czytelnikowi zostanie przybliżona historia chatbotów i przykłady ich obecnych zastosowań. Rozdział kończy krótka prezentacja nowatorskiego rozwiązania Ruby on Rails oraz języka, w którym zostało ono napisane.

Rozdział 3 stanowi kompleksowe omówienie stworzonego systemu. Przedstawia opis architektury opartej o wzorzec MVC, a także uproszczony schemat konwersacji cybersprzedawcy z klientem. W rozdziale znajduje się także szczegółowy opis metody używanej do rozpoznawania wypowiedzi klienta oraz sposób odczytywania i analizy jego wymagań. Czytelnikowi zostanie przybliżony algorytm decyzyjny reakcji chatbota na wypowiedź klienta oraz dokładny opis technologii używanych w aplikacji.

Rozdział 4 prezentuje działanie systemu od strony klienta oraz administratora. W rozdziale znajdują się zrzuty ekranu przedstawiające poszczególne podstrony aplikacji wraz z opisami. Autorzy omawiają dokładny przebieg rozmowy podzielony na kilka głównych etapów:

- Faza początkowa
- Faza pytań otwartych
- Faza pytań szczegółowych
- Faza wyboru produktu

W rozdziale czytelnik znajdzie stenogramy przykładowych rozmów klienta z cybersprzedawcą oraz ich rezultaty w postaci oferty produktów.

Publikację kończy rozdział 5 podsumowujący wykonaną pracę. Znajduje się nim refleksja nad rezultatami pracy aplikacji oraz uwagi na temat wykorzystanych technologii. Autorzy w podsumowaniu proponują także możliwości dalszej rozbudowy aplikacji o kolejne funkcjonalności, pozwalające jej na jeszcze wydajniejszą realizację zadań.

2 Wprowadzenie teoretyczne w tematykę pracy

2.1 Rozwój Internetu

Moda na chwytliwe hasło Web 2.0 powoli się kończy. Otoczka medialna i echo, jakim się odbiło obwieszczenie nadejścia nowej generacji Internetu pod znakiem 2.0, było naprawdę spore. Z tego powodu magazyn The Times na człowieka 2006 roku wybrał Ciebie (ang. „You”), czyli zwykłego użytkownika Internetu.

Jesteśmy już przyzwyczajeni do entuzjazmu, jaki wywołał Internet drugiej generacji. Nie zdążyliśmy jeszcze ochłonąć, a u progu jest już kolejna rewolucja - Web 3.0. Web 3.0 nie jest ani chwytym marketingowym, będąc na fali modnego numerowania, ani nie jest przesadnym entuzjazmem ludzi zaangażowanych w rozwój Internetu. Trzecia generacja nowego medium ma bardzo solidne podstawy i to, co wyróżnia ją od poprzednich faz rozwoju, jest widoczne i konkretne.

2.1.1 Web 2.0

Termin „Web 2.0” po raz pierwszy został użyty w 2004 roku przez Dale Dougherty’ a, wiceprezesa O’Reilly Media Inc.¹, na wewnętrznym spotkaniu o przyszłym rozwoju Internetu. Autorzy tego terminu chcieli uchwycić i nazwać to, co powoduje, że Internet stał się ważniejszy niż kiedykolwiek i przeżywa swoje kolejne prosperity (inni, nazywali to po prostu kolejną bańką spekulacyjną). Wnioski z tej dyskusji zebrał i opublikował Tim O’Reilly (założyciel wydawnictwa) w swoim artykule *„What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software”*. Ten artykuł, choć poruszający temat od bardziej technicznej strony, rozpoczął faktyczną karierę pojęcia „Web 2.0” w mediach, w nauce i w biznesie.

Tim O’Reilly wyszczególnił 7 wyznaczników Internetu drugiej generacji, są to:

- Internet jako platforma (The Web as platform)
- Ujarmienie kolektywnej inteligencji - wytworu tłumu użytkowników Internetu (Harnessing collectiva intelligence)
- Treść/zawartość jest kolejnym „Intel inside” - w sensie, że jest najważniejsza (Data In the next „Intel inside”)

¹Cenione i uznane wydawnictwo dzięki swoim konferencjom o tematach technologicznych i dobrej jakości książek

- Skończenie z cyklami wytwarzania oprogramowania (End of the software release cycle)
- Lekkie wzorce programowania (Lightweight programming models)
- Warstwa programistyczna niezależna od sprzętu (Software above the level of single device)
- Wysoki poziom wrażeń doświadczanych przez użytkownika (Rich user experience)

Korzystając z terminologii Tima O'Reilla można w miarę łatwo osądzić, które strony/aplikacje internetowe należą lub nie należą (lub należą w jakiejś części) do drugiej generacji Internetu. Technologie, które są uosobieniem web 2.0 to między innymi: interfejs XML (RSS, Atom) - pozwalający na wymianę treści między stronami i aplikacjami, wykorzystanie mechanizmu Wiki i blogów - ułatwiające generowanie treści przez zwykłych użytkowników, nowe technologie jak AJAX i Ruby on Rails - zwiększające atrakcyjność i interaktywność stron/aplikacji internetowych, mechanizmy folksonomii (pol. „kumplonomia”) - ułatwiające „ujarzmianie” masowo generowanej treści, otwarte licencje - pozwalające na zbiorowe rozwijanie oprogramowania, otwarte API (interfejs programistyczny) - umożliwiające wymianę informacji między aplikacjami różnych producentów.

Oprócz konkretnych, technicznych i bardziej zorientowanych na informatyczną stronę Internetu wyznaczników web 2.0 ma też swoje znaczenie społeczne. Internet drugiej generacji przez media opiniotwórcze został zdefiniowany jako Internet, gdzie: treść jest tworzona przez samych użytkowników, treść jest oddzielona od formy, przez co ułatwiona jest jej wymiana między użytkownikami. W skrócie jest to Internet, który ze zbiorowości czytelników stał się społeczeństwem użytkowników - dzisiaj każdy, de facto, może tworzyć Internet, każdy może utworzyć nowe encyklopedyczne hasło w wikipedia.org, opublikować swój film na YouTube, prowadzić swojego bloga, dzielić się swoim nastrojem na twitter.com i każdy może polecać innym ciekawe treści na digg.com. Wszystkie te czynności mogą być oderwane od źródła. Dodając hasło na wikipedii albo film na YouTube zasilamy tą nową informacją cały Internet. Do naszego encyklopedycznego hasła mogą prowadzić odnośniki z innych stron, a nasz filmik może zostać umieszczony na profilu myspace.com jakiejś nastolatki. W skrócie dopiero Internet w wersji 2.0 stał się prawdziwie interaktywnym medium i właśnie to jest najbardziej doniosłe w społecznym znaczenie terminu web2.0.

2.1.2 Numerowanie Internetu

Trudno powiedzieć jednak, żeby Internet drugiej generacji był wielkim i nagłym przełomem technologicznym. Wiele technologii pasujących do wyznaczników Tima O'Reilly'ego już istniała na długo przed nadaniem Internetu numeru 2.0. Interfejs RSS istnieje już od 1999 roku, Amazon.com wprowadziło otwarte API w 2002 roku, także mechanizmy ułatwiające wymianę myśli między użytkownikami, jak choćby komentarze na Onet.pl, istniały długo wcześniej. Społeczności internetowe też nie są nowym pomysłem, tworzyły się od samego początku - chociażby wspomnieć pierwsze społeczności hackerskie koncentrujące się wokół kanałów IRC i hackzinów² lub społeczności fanów gier komputerowych koncentrujących się na forach internetowych.

Nie ma wątpliwości, że druga generacja Internetu jest jednak przełomem i zmianą. Chociaż rozwiązania technologiczne i niektóre pomysły istniały wcześniej, to skala i podejście do Internetu jako medium bardzo się zmieniło. Kiedy wcześniej te technologie i rozwiązania były marginesem, teraz są one „mainstreamem”. Zmieniły się przyzwyczajenia użytkowników i wyszło na jaw, że można zarabiać pieniądze w zupełnie niespotykany wcześniej sposób. Tak powstał biznes 2.0 inspirowany na filozofii i modelach biznesowych drugiej generacji Internetu. Kto z tradycyjnie myślących przedsiębiorców mógł przypuszczać, że opłaca się robić coś za darmo? Takie potęgi jak Google, Mozilla i wiele innych dają temu przykład. Google ostatnio wyprodukował nową przeglądarkę internetową Google Chrome. Zespół ludzi (dobrze opłacanych przez firmę Google) poświęcił jej wiele godzin i nie dość, że przeglądarka została udostępniona użytkownikom za darmo, to jeszcze na licencji wolnego oprogramowania! Silnik tej przeglądarki może być użyty przez każdą inną dostępną na rynku przeglądarkę bez żadnych opłat. Brzmi jak czysty altruizm, ale właśnie tak działa biznes 2.0.

Skoro biznes naocznie się zmienił, to oznacza, że społeczeństwo (w tym wypadku społeczeństwo internetowe) też się musiało zmienić. I chociażby dlatego to medium zasługuje na oznaczenie go numerem 2.0.

Czym jest więc trzecia generacja Internetu, w którą wpisany jest program komputerowy niniejszej pracy magisterskiej? Kiedy nadejdzie Web 3.0?

2.1.3 Web 3.0

Internet trzeciej generacji to Internet „rozumiejący”. Jest to sytuacja, w której komputer będzie mógł zrozumieć takie proste pytanie człowieka, jak: „znajdź mi najtańszy pokój z

²Magazyn o tematyce hackerskiej pisany w niesformatowanym tekście

łazienką, z ciepłą wodą w pobliżu Nosala”. Program będzie musiał przetworzyć to pytanie, zrozumieć jego sens, a później rozumieć wszystkie przeszukiwane informacje w Internecie, by w efekcie znaleźć poprawną odpowiedź, a gdy nie znajdzie to poradzić coś w zamian.

Po raz pierwszy termin Web 3.0 został zdefiniowany przez Nova Spivack’a, wizjonera, przedsiębiorcę i blogera w 2007 roku [1]. Spivack powiązał ściśle swoją definicję terminu Web 3.0 z okresem między rokiem 2010 a 2020. Ten okres, wszystkie zmiany i technologie, które się pojawiają i spopularyzują, to właśnie ta owa trzecia generacja Internetu, zwana Web 3.0.

Spivack uważa, że w tym okresie najbardziej rozwiną się technologie związane z tzw. Semantyczną Siecią (ang. Semantic Web, termin często używany zamiennie z Web 3.0). Semantic Web jest to grupa technologii, która doprowadzi do tego, że treść Internetu zacznie być rozumiana przez maszyny (programy komputerowe). Człowiek już nie będzie jedynym bytem w wirtualnej przestrzeni Internetu, który rozumie jego treść - maszyny będą powoli przyswajać tę cechę. Będą za nas sortować, inteligentnie wyszukiwać informacje, załatwiać różne sprawy: zamawiać jedzenie do lodówki, prowadzić korespondencję itp. A to wszystko dzięki rozumieniu treści.

Kolejne generacje Internetu można porównać do komputerowego systemu plików: Web 1.0 był *read - only* (tylko odczyt), Web 2.0 jest *read - write* (odczyt i zapis), a Web 3.0 będzie *read - write - execute* (odczyt, zapis i wykonanie).

Zatem trzecia generacja Internetu, to przede wszystkim rozwój technologii składowania i analizowania danych (inaczej zwane sztuczną inteligencją) w przeciwieństwie do okresu web 2.0, gdzie najsilniej rozwijały się dziedziny związane z interfejsem użytkownika. Gwałtownemu rozwojowi ulegnie warstwa niewidoczna oczom zwykłego użytkownika.

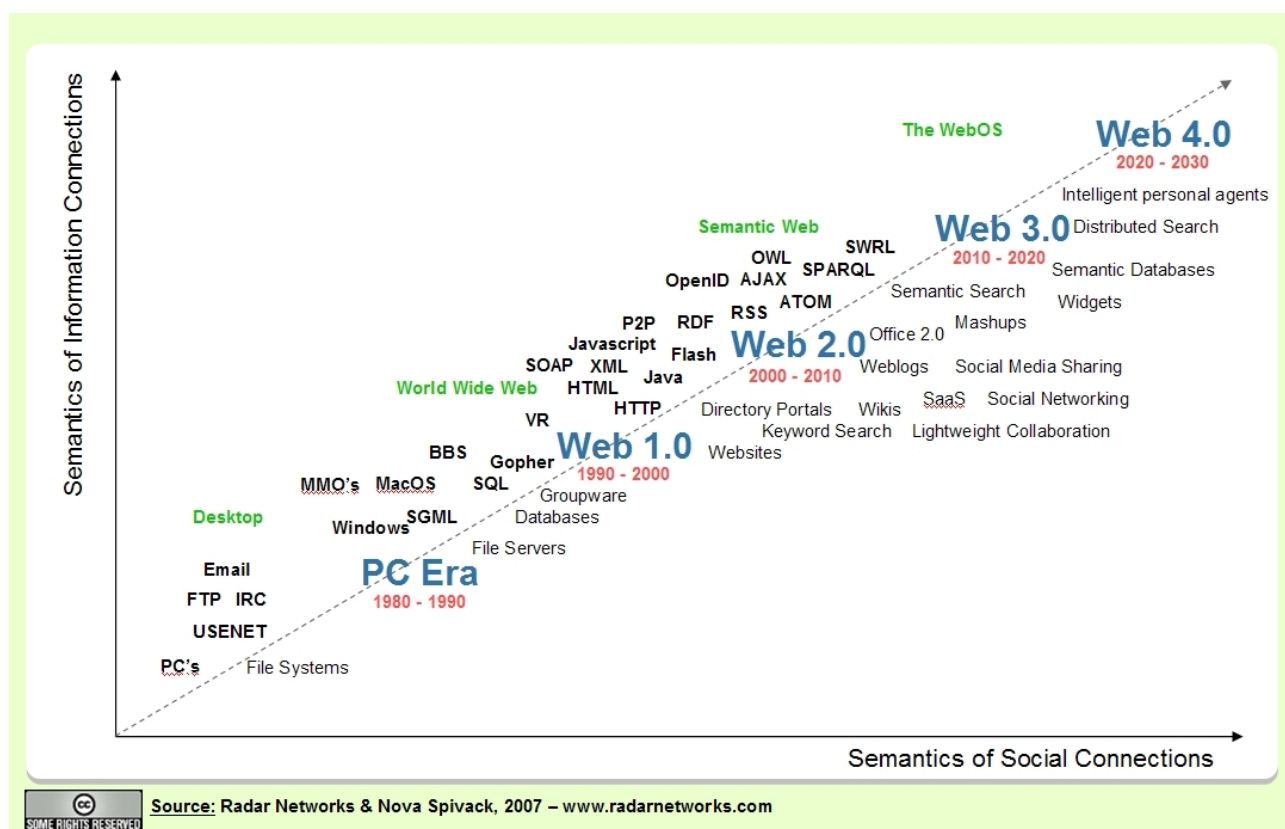
Technologie, które nazywamy „web2.0owymi” takie jak RSS, otwarte API, AJAX i inne, nie są informatycznym przełomem, nie są wyjątkowo wyrafinowane i zaawansowane. Są za to raczej proste, ustandaryzowane i otwarte, przez co zwiększa się szybkość ich implementacji. Szybkość tworzenia aplikacji jest teraz kluczowa. W polskich warunkach, czas potrzebny na realizację projektu internetowego od momentu powstania pomysłu to średnio 9 miesięcy. Właśnie ze względu na tą szybkość udaje się polskim podmiotom zajmować silną pozycję na rodzimym rynku internetowym i skutecznie opierać się konkurencji ze świata. Takie globalne potęgi i ich produkty jak Microsoft MSN, Yahoo, Ebay, Blogspot, Linkin musiały ustąpić polskim odpowiednikom.

Technologie web 3.0 będą o wiele bardziej wyrafinowane informatycznie. Dziedzina badań

nad sztuczną inteligencją ma już ponad 50 - letnią historię, w informatyce jest to cała wieczność! Także w trzeciej generacji Internetu do gry wejda technologie bardzo zaawansowane, które będą miały szanse zacząć się szybko rozwijać dzięki wyjątkowym warunkom, jakie stwarza bezprecedensowa wielkość zebranych zasobów w sieci internetowej.

Kolejna rewolucja jest nieunikniona. Technologie sztucznej inteligencji były do tej pory wykorzystywane głównie w sektorze bankowym, zamkniętym w swojej naturze. Pod wpływem Internetu nastąpi przyspieszony rozwój.

Kluczowym składnikiem Semantic Web będzie RDF i inne języki służące do semantycznego opisywania danych. Zastosowanie w Internecie znajdą też systemy regułowe, systemy eksperckie, ontologie, chatboty itp. Wszystko to są technologie rozwijane od początku powstania informatyki jako osobnej dziedziny wiedzy.



Rysunek 1: Rozwój technologii internetowych w czasie [2]

2.1.4 Społeczność użytkowników w dobie trzeciej generacji Internetu

Jak kształtują się zachowania, nawyki i społeczność użytkowników Internetu 2.0 zostało nie raz opisane, zbadane i przytoczone w powyższym opracowaniu. Na temat użytkownika trzeciej generacji można snuć wiele domysłów. Według autorów, rozwój sztucznej inteligencji

zwolni statystycznych użytkowników z przymusu posiadania wiedzy na temat zasad działania Internetu. Użytkownik będzie coraz więcej rozmawiał z komputerem, wydawał mu polecenia, a mniej obsługiwał. Internet zrobi się coraz bardziej przezroczysty, użytkownik zbliży się do innych użytkowników i usług. W kategoriach McLuhana³, Internet w trzeciej generacji stanie się medium jeszcze bardziej zimnym - tj. będzie można użyć więcej zmysłów i naturalnych sposobów interakcji (jak np. wydawanie poleceń w języku naturalnym, zamiast klikania i wpisywania słów kluczowych).

Do nielicznych jaskółek przyszłego „rozumiejącego” Internetu należą m. in. mechanizmy polecenia produktów na bazie historii zakupów - na Amazon.com; mechanizmy dostosowujące temat reklamy do kontekstu przeglądanych aktualnie treści - reklamy AdSense Google; inteligentni asystenci sklepów internetowych, np. na stronach IKEA; baza wiedzy KnowItAll rozwijana na Uniwersytecie Waszyngtońskim i współfinansowana przez Google; system sztucznej inteligencji CyC itp.

2.1.5 Podsumowanie

Internet zaczął się jako coś prostego - nieskomplikowany plik tekstowy HTML. Następnie rozwijał się sukcesywnie, coraz bardziej rozbudowując połączenia między rosnącą liczbą użytkowników. Nowe usługi, nowa funkcjonalność. Wraz z natłokiem informacji musiały powstać takie nauki jak: architektura informacji i użyteczność. Internet trzeciej generacji będzie jeszcze bardziej użyteczny, będzie mniej informacji, ale za to bardziej istotnych, bo będzie „rozumiał” człowieka. Pytanie tylko jak bardzo otoczenie inteligentnych, rozumiejących algorytmów wpłynie na użytkownika Internetu i czy przybliży, czy oddali go od innych ludzi. [3][4][2][1][5][6][7][8]

2.2 CRM

2.2.1 Historia CRM

Powstanie systemów informatycznych klasy CRM (Customer Relationship Management) datuje się na przełom lat osiemdziesiątych i dziewięćdziesiątych XX wieku [9], ale historia zarządzania relacjami z klientem jest tak stara jak sam handel. W dawnych czasach, jeszcze przed erą komputerów, podstawowym narzędziem wspomagającym utrzymanie optymalnych relacji

³Herbert Marshall McLuhan (ur. 21 lipca 1911 w Edmonton, zm 31 grudnia 1980) - kanadyjski teoretyk komunikacji. Zyskał rozgłos w latach 60. stwierdzeniem, że elektroniczne media, zwłaszcza telewizja, tworzą globalną wioskę, w której „medium jest przekazem”, tj. środki komunikacji mają większy wpływ na ludzi niż sama przekazywana wiadomość. Cyt. za http://pl.wikipedia.org/wiki/Marshall_McLuhan

klientów z przedsiębiorstwem był talent handlowca, jego dobra pamięć do ludzi i księgi rachunkowe. Okazuje się, że i tą dziedzinę życia (jak już prawie wszystkie) można z powodzeniem z informatyzować. Zanim jednak do tego doszło i powstał pierwszy zaawansowany system CRM, konieczne było zapotrzebowanie ze strony firm, czyli musiały się zmienić nawyki i wymagania samych klientów.

W Stanach Zjednoczonych, gdzie powstał pierwszy system klasy CRM, jeszcze w latach 70 - tych istniał rynek produktu. Konsumenci amerykańscy dążyli do ilościowego zwiększania konsumpcji i optymalizowania wydatków [10]. Wystarczyło zaoferować produkt o odpowiedniej jakości i atrakcyjnej cenie, a klient znajdował się sam. Korporacje amerykańskie prowadziły kampanie marketingowe koncentrując się na produkcie. Wraz z bogaceniem się społeczeństwa amerykańskiego i zwiększaniem nasycenia rynku towarami, zachowania konsumentów zaczęły ulegać ewolucji. Gama produktów rozszerzała się coraz bardziej, a konsumenci zaspokajali coraz więcej swoich pragnień, które wychodziły poza podstawowe potrzeby człowieka (m. in. takie jak zaspokajanie głodu, potrzeba posiadania dachu nad głową, potrzeba przemieszczania się itp.). Gusta klientów były coraz bardziej wyrafinowane, a oprócz tego wychodziło na jaw, że konsumpcja może być miejscem zaspokajania potrzeb wyższego rzędu niż do tej pory - potrzeb emocjonalnych, wynikających z indywidualnego charakteru klienta, potrzeb społecznych takich jak potrzeba identyfikacji itp. Na konkurencyjnym rynku amerykańskim przedsiębiorstwa, w trosce o swoją pozycję, musiały dotrzymać kroku zmieniającym się nawykom konsumentów i zmienić diametralnie swoją strategię przez przeniesienie punktu ciężkości z produktu na klienta [10]. Firmy powoli zmieniały swoją percepcję i w centrum stawiały coraz częściej klienta i jego wymagania niż oferowany produkt. Wszystkie działania strategiczne, a w tym kampanie marketingowe korporacji amerykańskich, zaczęły podlegać klientom. Ta ewolucja, zaczęta na przełomie lat 80 - tych i 90 - tych, dała impuls do rozwoju systemów informatycznych do zarządzania relacjami z klientem.

Na zmianę orientacji przedsiębiorstw, oprócz zwiększających się wymagań klientów, przyczyniły się też liczne badania rynkowe publikowane w latach 80 - tych. Badania jednoznacznie wykazały, że pozyskanie nowego klienta może być nawet 5 - krotnie droższe, niż utrzymanie dotychczasowego [9].

Lojalność klienta została doceniona i bardzo wzrosło znaczenie relacji klienta z firmą. W porównaniu z latami 60 - tymi dwudziestego wieku była to rewolucja. Powstało wiele publikacji i poradników o lojalności. Lojalność nie jest celem samym w sobie. Liczne badania dowodziły, że klient lojalny to klient, który jest bardziej skłonny na zakup nowych produktów firmy,

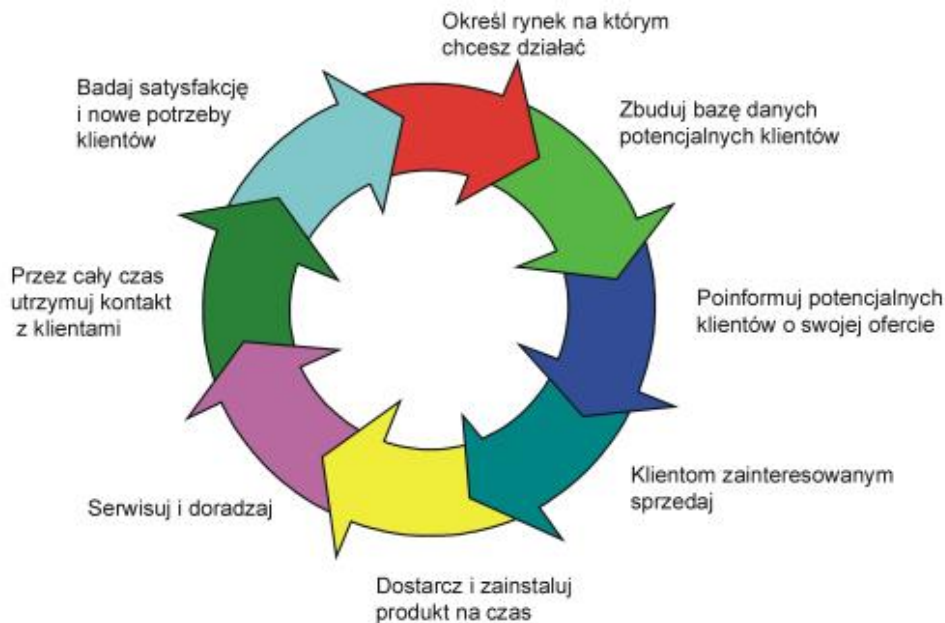
czyli w efekcie jest to klient, który wyda więcej pieniędzy. Lojalny klient, to klient bardziej dochodowy. Po drugie, lojalny klient to klient, który jest bardziej skłonny polecać usługi danej firmy. Według badań, zdobycie jednego lojalnego klienta pociąga za sobą średnio pięciu nowych klientów. Dlatego znaczenie relacji klienta z firmą jest tak kluczowe dla marketingu [11]. Znaczenie to zwiększa się nieustannie, a w dobie powszechnego dostępu do Internetu, gdzie każdy człowiek ma nieograniczone możliwości dzielenia się swoimi opiniami na skale globalną, jest strategiczne. Nowoczesny marketing opiera się na lojalności klientów, z których część zostaje tzw. ambasadorami marki.

Nasylenie rynku produktami, bogacenie się społeczeństwa, zmiana zachowań konsumenc- kich, liczne badania marketingowe, powszechny dostęp do Internetu i w końcu globalizacja zaowocowały zwiększającym się zapotrzebowaniem firm na systemy wspomagające zarządza- nie relacjami z klientem. Pierwszymi takimi systemami informatycznymi były aplikacje typu *contact management*. Było to połączenie funkcji kalendarza z prostą bazą danych klientów. Następnie dzięki dynamicznemu rozwojowi technologii informatycznych oraz teleinformatycz- nych w latach 80 - tych i 90 - tych rozwinęły się one w kolejne wcielenia: *call reporting system*, *territory management system*, *sales management system*, *sales team automation*. Wszystkie te systemy cechowała funkcjonalność wymiany danych między centralną bazą wiedzy o klien- tach a sprzedawcami w terenie. Nie zyskały one jednak większej popularności, gdyż ich wadą była jednostronność przepływu informacji. Systemy te służyły do centralnego analizowania danych o klientach i kontroli sprzedawców w terenie przez staranniejsze rozliczania z efektów ich pracy. Sprzedawcy nie mieli większego pożytku z tych systemów, gdyż tylko jednostka cen- tralna (np. szef departamentu sprzedaży) posiadała całościową wiedzę o klientach. Pierwsze systemy, które dopuściły sprzedawców do całościowej wiedzy o kliencie, znane były pod na- zwą *sales force automation* (SFA). Równolegle rozwijały się systemy typu *Customer Service Support* (CSS) usprawniające obsługę serwisową po sprzedaży. Bezpośrednio z tych systemów wyewoluowały dzisiejsze zaawansowane aplikacje klasy CRM, które realizują cele i całościowo wspierają filozofię *Customer Relationship Management*.

2.2.2 Co to jest Customer Relationship Management

Customer Relationship Management (CRM) czyli Zarządzanie Relacjami z Klientami jest dziedziną stosunkowo młodą i nie zostały wypracowane jeszcze jej ścisłe standardy. Można spo- tknąć się z wieloma definicjami tego pojęcia, np. w [9], [12], [13]. Ogólnie rzecz ujmując, CRM jest to kultura biznesowa, gdzie stały kontakt i zadowolenie klienta [14] jest kluczową warto-

ścią. Jest to strategia biznesowa skupiająca się na budowaniu długoterminowych związków z klientem i umacnianiu ich na różnych płaszczyznach. Jest to także filozofia działania przedsiębiorstwa. Jest ono tak zorganizowane, aby zapewniony był jednolity system we wszystkich procesach biznesowych - od sprzedaży, przez obsługę klienta aż do analizy zebranych danych, które posłużą do lepszego reagowania na zapotrzebowania klientów i pogłębiania więzi. Ta strategia biznesowa, według najnowszych trendów, najlepiej prowadzi do zwiększenia zyskowności przedsięwzięcia i utrzymania się na konkurencyjnym rynku.



Rysunek 2: Elementy filozofii CRM, źródło [15]

„CRM rozumiane jest dwojako. Z jednej strony jest to pewna organizacja działań, realizowana przez odpowiednie stanowiska i jednostki organizacyjne przedsiębiorstwa plus środki techniczne i oprogramowanie. Z drugiej strony jest to pewien sposób budowania strategii marketingowych w oparciu o badania zadowolenia, zaufania i lojalności klientów wobec marek.”[16]

Systemy CRM to narzędzia informatyczne wspomagające działanie przedsiębiorstwa zgodnie z filozofią *Customer Relationship Management* [9][12][17]. Nie jest możliwe funkcjonowanie przedsiębiorstwa w duchu CRM bez komputerów i odpowiedniego systemu informatycznego. Ale samo zakupienie systemu klasy CRM nie gwarantuje sukcesu. Żeby wdrożyć system CRM najpierw musi być przyjęta filozofia i pod nią zorganizowana struktura firmy. Musi być też zmieniona mentalność pracowników od najwyższego szczebla do najniższego. Dlatego między innymi aż 70% wdrożeń systemów CRM nie kończy się sukcesem [12]. Według danych AMR Research [18], tylko 16% dotychczasowych wdrożeń systemu CRM zwróciło się inwestorom.

2.2.3 Cele stawiane systemom CRM

Zgodnie z opisem powyżej, CRM nie sprowadza się do zarządzania kontaktami z klientami. Jest to system o wiele bardziej wszechstronny, przenikający każdą komórkę przedsiębiorstwa. Wdrożenie systemu CRM przysporzy oczekiwane korzyści jedynie wtedy, gdy będzie rozumiany jako „*infrastruktura umożliwiająca określenie i zwiększenie wartości klienta oraz odpowiednie środki, za których pomocą motywuje się najlepszych klientów do lojalności czyli ponownych zakupów*” [9]. Cele, jakie przedsiębiorstwa stawiają przed systemami typu CRM to:

Pozyskanie wiedzy o kliencie Dzięki zebraniu wszystkich danych o kliencie w jednym miejscu i możliwości poddania ich analizie, można poznać i zrozumieć potrzeby klienta. Zdarza się, że dzięki systemowi CRM firma pozna preferencje klienta lepiej niż on sam, co może znacznie zwiększyć poziom zadowolenia klienta.

Segmentacja System CRM umożliwia stworzenie klasyfikacji klientów z podziałem, np. na klientów kluczowych (przynoszących największe zyski), średnich, a nawet stratnych. Ma to wpływ na decyzje przedsiębiorstwa, w którą grupę klientów inwestować, a gdzie ograniczyć wydatki.

Pozyskanie nowych klientów Poprzez poznawanie potrzeb i preferencji swoich klientów, firma może tworzyć lepsze strategie marketingowe.

Utrzymanie klienta Podejmowanie takich działań poszczególnych komórek przedsiębiorstwa, by ulepszać relacje z klientem.

Obsługa klienta Zarządzanie kontaktem firmy z klientem.

2.2.4 Architektura, rodzaje systemów CRM

Zintegrowany system CRM czyli taki, który spełnia wszystkie postawione powyżej cele, składa się najczęściej z trzech podsystemów [19]:

- analityczny
- operacyjny
- kontaktowy

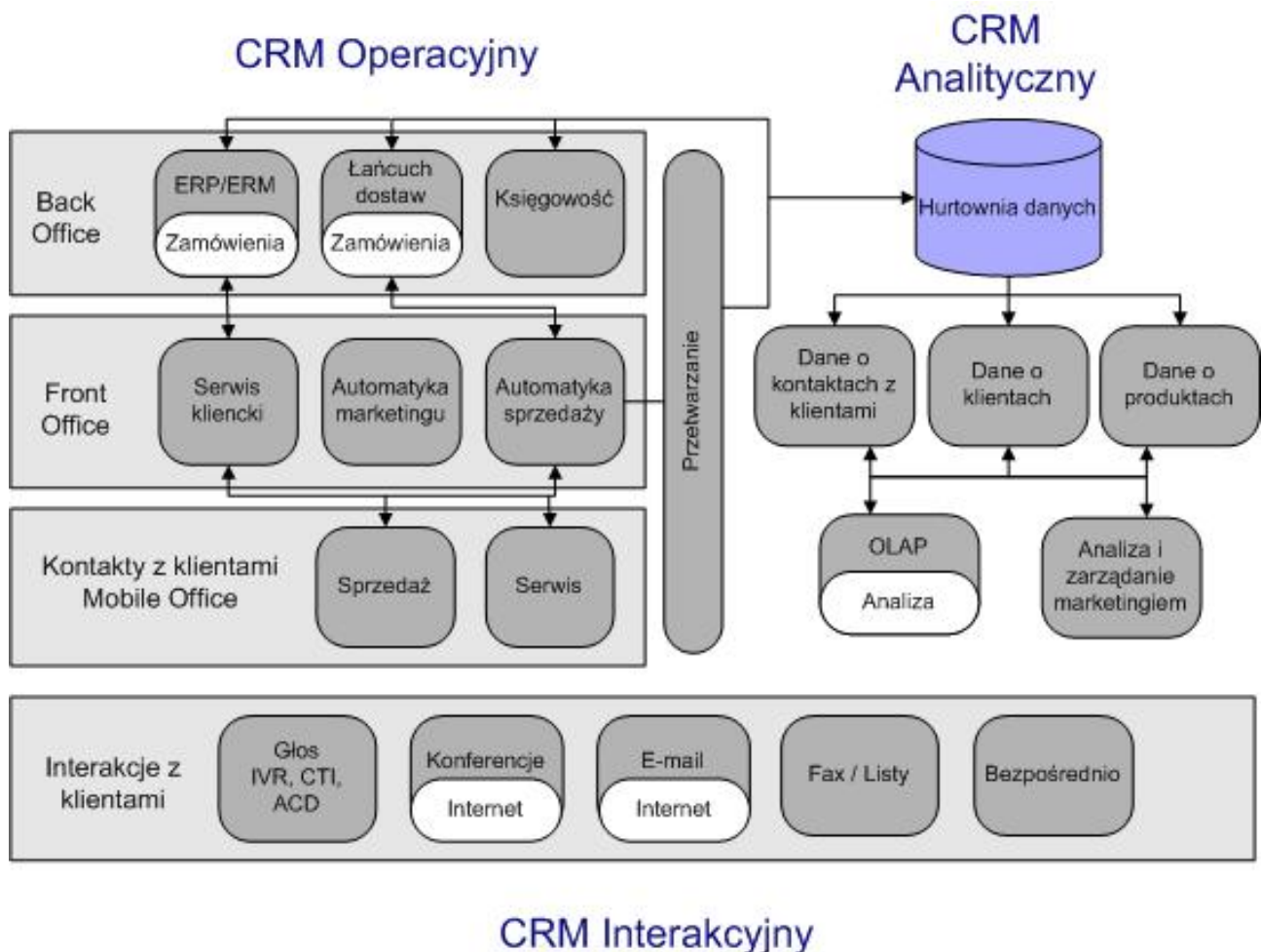
Analityczny Podsystem analityczny jest odpowiedzialny za przechowywanie danych o klientach i analizowanie ich. Do przechowywania danych najczęściej stosuje się zaawansowaną infrastrukturę bazodanową - Hurtownie Danych. Najważniejsze, z punktu widzenia firmy i procesu decyzyjnego, są wyniki analizy tych danych. Skomplikowane analizy statystyczne (np. Data Mining) pomagają firmie poznać potrzeby klientów i zrozumieć ich zachowania. Zadaniem podsystemu analitycznego są: segmentacja klientów, kierowanie kampaniami (analiza wydajności), opieka nad klientem (analiza obsługi klienta oraz serwisu) i analiza sprzedaży. Do segmentacji klientów najczęściej używa się analizy RFM (Recency - świeżość, Frequency - częstość, Monetary - wartość), która grupuje klientów na podstawie wcześniejszych ich zachowań i na tej podstawie przewiduje ich przyszłe zachowania. Częstość badaniem jest też LTV (Life Time Value) pozwalające oszacować całkowitą wartość, jaką dany klient stanowi dla przedsiębiorstwa podczas całego swojego życia, innymi słowy, ile pieniędzy taki klient jest skłonny wydać na rzecz produktów i usług firmy. Te wymienione przykładowe badania i wiele im podobnych pomagają przedsiębiorstwu zorientować się w sytuacji na rynku i zoptymalizować swoje działania, by osiągnąć największy możliwy zysk.

Operacyjny Podsystem operacyjny obejmuje typowe funkcje biznesowe takie jak: obsługa klienta, zarządzanie zamówieniami, fakturowanie/wystawianie rachunków, a także automatyzacja i zarządzanie marketingiem. Jest to ta część systemu CRM, która jest najmocniej zintegrowana z systemami typu ERP (Enterprise Resource Planning), stanowiąc razem zintegrowaną infrastrukturę informatyczną przedsiębiorstwa, przez którą przepływają wszystkie bieżące operacje w firmie.

Kontaktowy Podsystem kontaktowy koordynuje sieć kanałów kontaktowych z klientami, dostawcami i partnerami biznesowymi. Np. w call center dba m. in. o to, by operator telefoniczny odbierający zgłoszenie serwisowe od klienta, miał dostępne wszystkie niezbędne zebrane informacje, potrzebne przy obsłudze tego konkretnego klienta. Innymi kanałami kontaktowymi może być portal internetowy, SMS, e - mail, faks, poczta tradycyjna. Moduł ten najczęściej jest wykorzystywany do obsługi serwisowej klienta, do sprzedaży i marketingu.

2.3 Obsługa klienta

W dzisiejszych czasach dobrego sprzedawcę charakteryzuje nie tylko dobra znajomość produktu, który sprzedaje. Dobry sprzedawca to zarazem psycholog i negocjator. Dobry sprze-



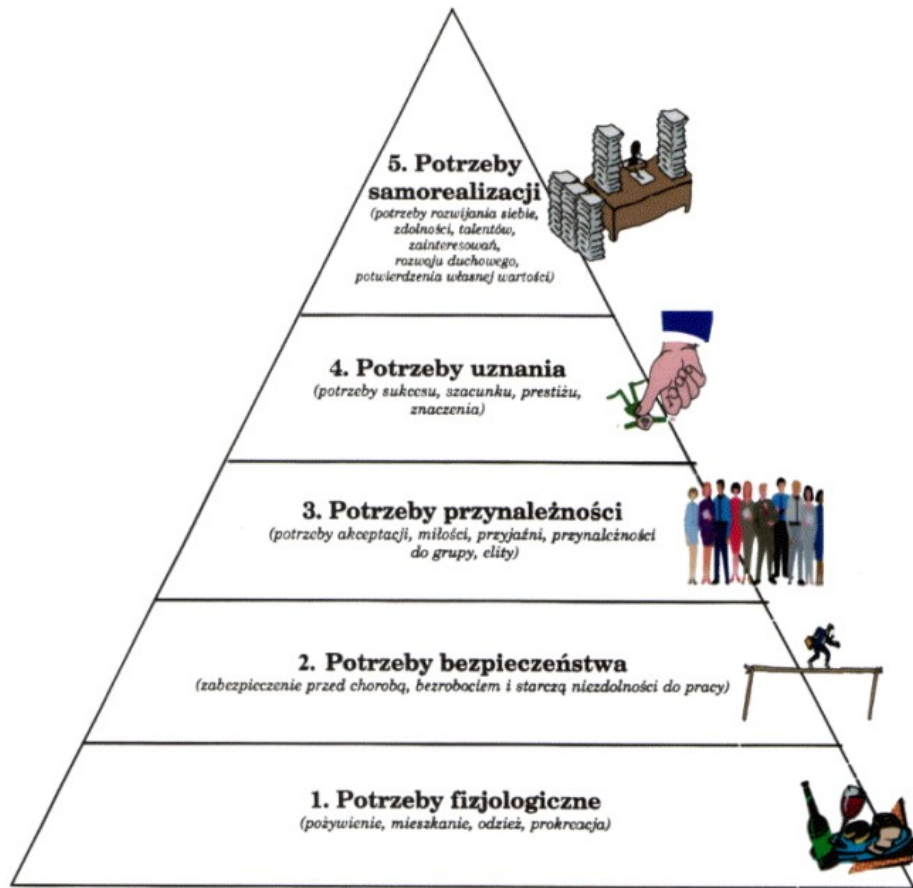
Rysunek 3: Architektura zintegrowanego systemu klasy CRM, źródło [19]

dawca, tak samo jak dobry system CRM, potrafi poznać potrzeby klienta lepiej niż on sam. W poprzednich rozdziałach przedstawiono zmiany zachowań konsumentów. W nowej erze jakość konsumpcji jest coraz ważniejsza i przeważa nad ilościowym zwiększaniem konsumpcji, typowym jeszcze w latach 70-tych w rozwiniętych społeczeństwach zachodnich. Sprzedawca może być traktowany jako brama przepustowa do osiągnięcia tak pożądanej przez klienta satysfakcji.

2.3.1 Kontakt sprzedawcy z klientem

Często się zdarza, że klient lubi jakiś sklep bardziej niż inny, mimo, że obydwa oferują zbliżone produkty w zbliżonych cenach. Dlaczego tak się dzieje? Odpowiedzią może być obsługa klienta. Mogą być to z pozoru rzeczy niezauważalne, jednak o wielkiej wadze, gdyż w ostateczności decydują o sukcesie na dzisiejszym konkurencyjnym rynku.

Kontakt sprzedawcy z klientem jest bardzo ważny. To właśnie dzięki niemu (nie ważne czy twarzą w twarz, czy np. przez Internet) realizuje się kluczowy etap obsługi klienta - poznanie jego potrzeb. Dobry sprzedawca musi się charakteryzować m. in.:



Rysunek 4: Piramida potrzeb człowieka - Maslov

- Umiejętnością aktywnego słuchania
- Umiejętnością dopasowania się do rozmówcy
- Wzbudzaniem zaufania
- Pewnością jakości oferowanego produktu
- Asertywnością
- Nie ocenianiem i nie pouczeniem klienta

Sprzedawca musi wzbudzać do siebie zaufanie, co pośrednio przekłada się, również, na zaufanie klienta do oferty handlowej. Ludzie często oczekują tego, o czym sami nie wspominają, dlatego sprzedawca powinien umieć aktywnie słuchać. Do rozpoznawania potrzeb klienta są też potrzebne umiejętności psychologiczne. Ludzie są różni, mają różne charaktery i wynikające z tego różne potrzeby. To właśnie tych potrzeb najczęściej klienci nie artykułują, ponieważ

często sami nie są ich świadomi. Na przykład, klient o charakterze Władczym⁴ nie będzie zadowolony, gdy sprzedawca nie będzie dawał mu wyboru podczas prezentowania oferty. Taki klient będzie czuł (często nieświadomie) dyskomfort w takiej sytuacji. To się może przełożyć na jego niezadowolenie z transakcji i w przyszłości nie powróci do tego sklepu mimo, że zakupił produkt, który w ostateczności był dla niego najlepszy. By uniknąć takiej sytuacji, sprzedawca musi być gotowy na profesjonalną obsługę klienta. Sprzedawca powinien się dostosowywać do klienta i umieć się wsłuchać w jego potrzeby, być na nie otwartym. Jest wiele technik handlowych, które w tym pomagają. Jedną z nich jest zadawanie pytań. Pytania to droga do tego, żeby poznać klienta. Mają one zachęcić klienta do „wygadania się” na temat swoich potrzeb. Sprzedawca nie powinien ani oceniać, ani pouczać, gdyż to może zniechęcić klienta do dzielenia się informacją o sobie, a wręcz obrazić. Dzięki pytaniom można poznać stanowiska, preferencje, potrzeby i ewentualne obawy klienta.

Są dwa rodzaje pytań: zamknięte i otwarte. W początkowej fazie rozmowy sprzedawca powinien zadawać pytania otwarte, prowokujące klienta do tego, by sam się wypowiedział o tym, co go interesuje. Z odpowiedzi na te pytania, sprzedawca może się też dużo dowiedzieć o charakterze klienta i o potrzebach z tego wynikających. Gdy sprzedawca będzie miał już jasność co do potrzeb, interesów i stanowisk klienta, może zadawać pytania zamknięte, precyzujące zebraną wiedzę do sporządzenia ostatecznej oferty. Pomocna może być też typologia pytań opracowana przez Benjamina Bloom’a z Uniwersytetu w Chicago [20][21]. Bloom był psychologiem szkolnictwa i opracował w roku 1956 klasyfikację pytań, które prowokują do ukazania systemu odkrywczego danego człowieka. Klasyfikacja została stworzona na potrzeby edukacji, pytania te były sposobem nauczycieli na sprawdzanie poziomu nabycia wiedzy przez uczniów. Ukazywała też dokładnie, w których miejscach pojmowania abstrakcyjnych obiektów są braki. Pytania Bloom’a są też szeroko używane w handlu dla sprawdzenia, w jaki sposób klient postrzega produkt, markę i firmę. Jest to świetne narzędzie dla handlowca, gdyż ukazuje wątpliwości albo błędy klienta, które on może rozwiązać, co wpłynie w efekcie na zakup i zadowolenie klienta [14]. Klasyfikacja obejmuje 6 rodzajów pytań:

- Pytania o wiedzę
- Pytania o zrozumienie
- Pytania o zastosowanie

⁴Typologia charakterów jest opisana w rozdziale 2.3.3

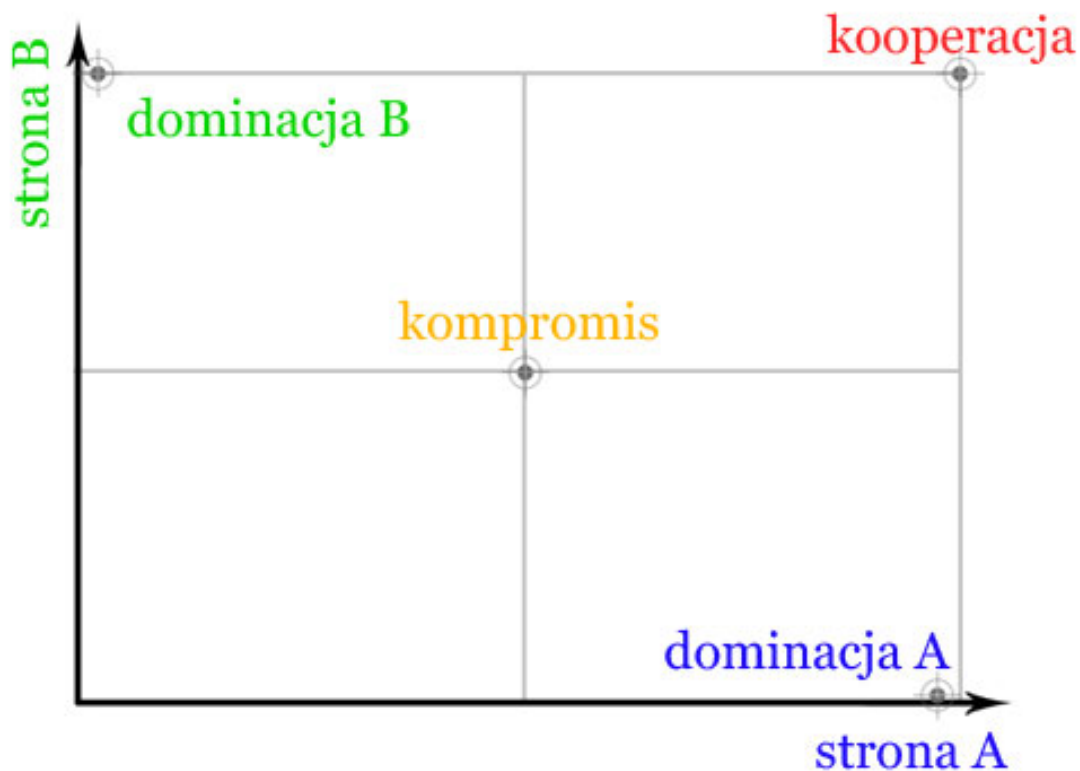
- Pytania o analizę
- Pytania o syntezę
- Pytania o ocenę

Klasyfikacja pytań Bloom'a stanowi swoiste ramy, dzięki którym sprzedawca może całościowo poznać stosunek i potrzeby klienta względem produktu.

2.3.2 Negocjacje

Dobry sprzedawca to też dobry negocjator. Nie wszystkie potrzeby i pragnienia klienta da się spełnić i nie wszystkie są dla firm opłacalne. Firmie zależy na dochodzie ze swojej działalności i na satysfakcji klienta - są to rzeczy, które nie koniecznie idą ze sobą w parze. Do znalezienia złotego środka między tymi dwoma wartościami służy rozmowa handlowa, którą można swobodnie porównać do negocjacji - *„Negocjacje to dwustronny proces komunikowania się, którego celem jest osiągnięcie porozumienia, gdy przynajmniej jedna strona nie zgadza się z daną opinią lub z danym rozwiązaniem sytuacji. Negocjacje to sposób porozumienia się w celu rozwiązania konfliktu oraz dojścia do porozumienia obydwu stron, proces wzajemnego poszukiwania takiego rozwiązania, które satysfakcjonowałoby zaangażowane w konflikt strony.”* [22] Jest wiele stylów negocjacji, tradycyjnie wyszczególnia się dwa: styl miękki i styl twardy. Styl miękki stawia sobie za priorytet zachowanie dobrej relacji z partnerem. Jest to styl pojednawczy, który pasuje do długoterminowej współpracy między partnerami. Technika stylu miękkiego jest dopasowywanie się, uleganie, unikanie konfrontacji. Natomiast styl twardy polega na ostrym stawianiu warunków i na nieugiętym stanowisku. Styl twardy nadaje się do negocjacji tam, gdzie jest gra o sumie zerowej (tj. jeden tort to podział: gdy jeden partner ma więcej, to drugi ma mniej) i nie ma długoterminowej współpracy między partnerami.

Przez lata do negocjacji zwykło się podchodzić jak do walki. Uważano, że najbardziej sprawiedliwym rozwiązaniem jest kompromis (tzw. „krakowski targ”), czyli takie rozwiązanie, gdy każda ze stron obniża swoje wymagania dochodząc w ten sposób do wspólnego punktu - punktu środkowego na poniższym wykresie 5. Tak naprawdę, to rozwiązanie, choć sprawiedliwe, jest połowicznym sukcesem każdej ze stron - nie jest to sytuacja zwycięska dla żadnej strony. Uważanie kompromisu za najlepsze wyjście jest wynikiem kulturowanej przez wieki tzw. Kultury Niedostatku [23][24] - jest to syndrom, który polega na tym, że ludzie z góry zakładają każdy targ jako grę o sumie zerowej. Prawda jest taka, że w życiowych warunkach bardzo rzadko



Rysunek 5: Wykres stylów negocjacyjnych. Źródło opracowanie własne

dochodzi do przypadku czystej gry o sumie zero. Najczęściej zawsze da się wypracować rozwiązanie, które usatysfakcjonuje każdą ze stron. Taka sytuacja nazywana jest rozwiązaniem WIN - WIN (wygrana - wygrana), a styl dochodzenia do takiego rozwiązania nazywa się negocjacją integratywną. [25][26]

Negocjacja integratywna polega na aktywnym włączaniu się dwóch stron w proces poszukiwania rozwiązania. Nie jest to, w przeciwieństwie do innych stylów negocjacji, obstawianie na swojej pozycji lub rezygnowanie z niej, ale jest to kreatywne i wspólne dochodzenie do rozwiązań, na które pojedynczo strony by nigdy nie wpadły. W roku 1991 [23] autorytety z dziedziny negocjacji Roger Fisher i William Ury z Uniwersytetu w Harvard opracowali szczegółowo elementy metody stylu negocjacji integratywnej. Ta metoda została nazwana „negocjacjami opartymi na zasadach” i składa się z 4 przewodnich zasad:

1. oddziel ludzi od problemu
2. skoncentruj się na interesach, a nie na stanowiskach
3. opracuj możliwości korzystne dla obu stron
4. upieraj się przy stosowaniu obiektywnych kryteriów

Ad.1. Zasady te powodują, że strony negocjacji stają się partnerami, którzy wspólnymi siłami szukają nowej jakości. Negocjacje integratywne mogą się odbywać jedynie w warunkach, gdy obydwie strony chcą dojść do porozumienia i wykazują wolę spojrzenia na sprawę oczami drugiej strony. Okazuje się, że najczęściej nieporozumień w negocjacjach wynika z posiadania różnych systemów wartości. Problemem utrudniającym komunikowanie się są też wzajemne obawy i uprzedzenia.

Ad.2. Stanowiska są to sposoby, jakimi ludzie realizują swoje interesy. Często, gdy strony stają do negocjacji, każda ma już wyrobione swoje stanowiska, które chce przeforsować. Nie rzadko okazuje się, że mimo sprzecznych stanowisk strony mają zbieżne interesy. Dlatego negocjatorzy powinni zgłębiać swoje wzajemne interesy i zamiast spierać się o stanowiska wspólnie poszukiwać nowych rozwiązań.

Ad.3. Gdy strony poznają swoje interesy i potrzeby, można się zastanowić nad ich spełnieniem - najlepiej przez wspólne wypracowanie i dyskutowanie szeregu rozwiązań. Taka merytoryczna dyskusja jest możliwa tylko, gdy zostaną spełnione dwa powyższe punkty, czyli odrzucenie uprzedzeń i dobre poznanie wzajemnych interesów - wtedy jest szansa na uczciwe i merytoryczne poszukiwanie wspólnego rozwiązania.

Ad.4. Cała dyskusja, by miała solidne podstawy, potrzebuje obiektywnych kryteriów oceniających satysfakcję jednej i drugiej strony. Im bardziej ściśle będą te kryteria, tym bardziej merytoryczna będzie droga wypracowywania mądrego porozumienia.

2.3.3 Typologia charakterów klientów

By rozmowa handlowa skończyła się satysfakcją obu stron, sprzedawca musi dobrze poznać potrzeby klienta. Doświadczeni sprzedawcy potrafią rozróżnić różne typy osobowości klientów, a wszystko po to, by lepiej zrozumieć i przewidzieć ich potrzeby, a w efekcie, zyskać zaufanie. Jest wiele typologii ludzkich charakterów. Przedstawiona i omówiona poniżej jest typologią opracowaną przez dr Adriana Horzyka [27][28][29]. Przedstawione poniżej typy osobowości, zwane są inaczej programami osobowości. „Programy osobowości są to pewne grupy zachowań, które występują razem i które pozwalają rozpoznawać i przewidywać zachowania oraz potrzeby człowieka/klienta w różnych sytuacjach”[27]. Poniższa lista 12 typów charakterów to typy idealne - w realnym świecie zazwyczaj ludzie mają mieszankę cech pochodzących z różnych tych typów, zazwyczaj z jednym lub kilkoma dominującymi. Każdy program osobowości realizuje pewną misję w życiu człowieka i w dużej mierze jego potrzeby wynikają z intensywności danego

programu. O tym, jakimi programami osobowościowymi z przypisaną do nich intensywnością charakteryzuje się dany człowiek, decydują w dużej mierze geny. Intensywność programów osobowości jest dziedziczona i prawie nie ulega zmianie podczas życia człowieka [27]. Programy osobowości [27][30][28][29]:

Rządzący - Typ Rządzący to osoba, która odczuwa potrzebę rządzenia i decydowania za siebie i za innych. Osoba ta nie lubi jak się jej coś narzuca, nawet, gdy wynika to z dobrej woli - Typ Rządzący będzie doszukiwał się w takim działaniu podstęp. Typ Rządzący nie lubi manipulacji i jakichkolwiek innych metod, którymi może się ktoś posłużyć by narzucić mu swoją wolę. Rządzący niechętnie akceptuje reguły, chyba, że sam jest ich autorem. W sytuacji handlowej, klienta o Typie Rządzącym trzeba solidnie o wszystkim informować. Handlowiec musi uważać, gdyż jest to klient przeczulony na punkcie własnych autonomicznych decyzji. Sprzedawca powinien być dla takiej osoby rzetelnym źródłem informacji i nie powinien proponować produktów, ani jakkolwiek wartościować, gdyż ryzykuje tym samym, że klient odbierze to jako próbę perswazji.

Dostosowujący - Typ Dostosowujący to przeciwieństwo programu Rządzącego. Osoba o charakterze Dostosowującym nie chętnie podejmuje decyzje - woli się opierać na opinii innych. Wymaga porad i kieruje się opiniami autorytetów w danych dziedzinach. Odczuwa komfort gdy, ktoś inny bierze odpowiedzialność za jej decyzje. W sytuacji handlowej, osoba Dostosowująca jako klient woli by to sprzedawca wybrał dla niej produkt. Uważa, że to sprzedawca powinien wiedzieć lepiej co jest dla niej dobre. Sprzedawca, by zadowolić klienta o tym charakterze powinien rekomendować konkretne produkty, powoływać się na autorytety i brać odpowiedzialność (zapewniać, że to najlepszy wybór) za decyzje zakupu.

Maksymalista - Jest to osoba stawiająca przed sobą wielkie cele i dążąca do nich. Często nie zwraca uwagi na trudności, przeszkody. Jest wielkim optymistą i bardzo łatwo daje się ponieść pasji. Chce być wyjątkowa, odróżniać się od innych i wprowadzać innych w zachwyt swoją osobą. Często traci kontakt z rzeczywistością, która jest źródłem trudności w osiągnięciu ambitnych celów. Często dla maksymalisty wyobrażenie sobie celu jest równoznaczne z dojściem do niego. W sytuacji handlowej osoba o typie Maksymalista szuka produktu o nieprzeciętnych cechach, dzięki którym poczuje się wyjątkowa albo osiągnie uniesienie. Sprzedawca mając do czynienia z osobą o tym charakterze powinien wysłuchać jego wizjonerskich zamiarów. Klientowi należy proponować wyjątkowe produkty oraz opisywać niesamowite efekty, jakie one wywołują i jak bardzo zbliżają go

do jego planów. Sprzedawca nie powinien wspominać o eksploatacji ani o serwisie, gdyż informacje o trudnościach i problemach szybko zniechęcają Maksymalistę.

Asekuracyjny - Jest to osoba, która w sposobie traktowania problemów i przeciwności, jest przeciwieństwem typu Maksymalisty. Osoba ta, lubi się zabezpieczać na każdy wypadek. Myśląc o przyszłości analizuje najpierw trudności, które mogą ją spotkać. Osoba ta nie lubi niepewności i ryzyka, jest skłonna do wysiłku, by tylko mieć „święty spokój” i pewność, że jest zabezpieczona na każdą ewentualność. W sytuacji handlowej taki klient potrzebuje asekuracji - gwarancji lub rękojmi. Sprzedawca powinien rzetelnie informować o wszystkich możliwych problemach, gwarantując ich rozwiązanie lub oferując swoją pomoc. Klient asekuracyjny jest bardzo nieufny i sprawdza po kilka razy czy dane mu słowa mają pokrycie. Dlatego sprzedawca musi być godny zaufania i starać się rozwiewać wszelkie obawy klienta.

Inspirujący - Osoba o charakterze Inspirującym kieruje się w życiu intuicją. Lubi tworzyć nowe rozwiązania, wymyślać wynalazki, innowacyjne projekty, łączyć rzeczy, których nikt wcześniej nie łączył itp. Osoba Inspirująca irytuje się, gdy zmusza się ją do systematyki i planu działania. Nie potrafi podporządkować się dyscyplinie, szybko się zniechęca i rozpoczyna następny wątek. W sytuacji handlowej osoba o charakterze Inspirującym poszukuje inspiracji, nowych rozwiązań, fascynuje ją innowacja, często stawia aspekt estetyczny nad funkcjonalnym. Lubi dużo testować i dokonywać różnorodnych wyborów. Sprzedawca nie powinien wyliczać i systematycznie prezentować oferty - powinien czymś zaskoczyć, „wyciągnąć królika z kapelusza”, coś co jest nietypowe, coś co zainspiruje. Sprzedawca musi dbać o różnorodność w kontakcie z takim klientem, dwie tak samo wyglądające transakcje już mogą wystarczyć by znudzić i zniechęcić Inspirującego klienta.

Harmonijny - Jest to osoba dążąca do harmonii, ładu i porządku. Unika i łagodzi konflikty w jej otoczeniu, czasami wręcz za cenę kłamstwa lub ukrywania prawdy. Często nie ukazuje prawdziwego uczucia i osądów, by nie prowokować ewentualnego konfliktu. W sytuacji handlowej jest to niezwykle trudny typ charakteru. Sprzedawca nigdy nie może być do końca pewien czy wydawane przez klienta osady i deklaracje są zgodne z prawdą czy tylko próbą unikania sprzeczności zdań. Sprzedawca powinien stworzyć atmosferę bezpieczeństwa i spokoju. Powinien zapewniać klienta, że wyjawienie swojego zdania nie spowoduje żadnej negatywnej reakcji. Powinien ustawicznie pytać o zastrzeżenia i ewentualne wątpliwości, gdyż inaczej osoba Harmonijna będzie jednorazowym klientem,

bądź transakcja w ogóle nie dojdzie do skutku.

Współczujący - Osoba o tym charakterze bardzo ceni osobisty kontakt z drugim człowiekiem, dlatego często skraca dystans, przechodzi na „Ty”, mówi o swoich osobistych sprawach, by zaprosić do swojego wnętrza. Często zdradza intencje swoich czynów, bo zależy jej na tym by rozmówca ją rozumiał. Osoba o tym charakterze jest naturalna i bardzo ją boli gdy w odpowiedzi dostaje formalizm. W sytuacji handlowej osoba Współczująca będzie chciała wejść w bliższy kontakt ze sprzedawcą. Będzie usatysfakcjonowana wtedy, gdy sprzedawca się przed nią odsłoni i będą mogli poruszyć tematy osobiste. Sprzedawca powinien taką osobę wysłuchać, odpowiedzieć otwartością, być cierpliwy, gdyż wysłuchanie może być czasochłonne.

Zadaniowy - Osoby o tym charakterze są odwrotnością osób Współczujących. Osoba Zadaniowa nie lubi rozmawiać o sprawach osobistych, uczuciach i intencjach. Przechodzi zawsze do sedna problemu bez zbędnych ozdobników. Skupia się na zadaniu, które trzeba wykonać, a ludzi traktuje jako środek do tego. W sytuacji handlowej osoba o charakterze Zadaniowym będzie się wypowiadała krótko i na temat czego chce i co potrzebuje. Sprzedawca najlepiej, żeby był dla takiego klienta przezroczysty - tzn. optymalizował czas dotarcia przez takiego klienta do produktu i informacji o produkcie, nie zakłócając tej drogi swoją osobą. Sprzedawca powinien być rzeczowy i sprawny, to najbardziej zadowoli klienta Zadaniowego.

Odkrywczy - Osoba o tym charakterze kieruje się żądzą poznania i odkrywania. Potrzebuje dopływu nowych interesujących informacji, z których można czerpać wiedzę. Pragnie poszerzać swoją wiedzę o świecie, myśleć, przewidywać, rozumieć. Interesuje go wszystko to, co nowe, lubi testować, degustować i wynajdywać nowe rzeczy. W sytuacji handlowej jest klientem dociekliwym, wypyującym o szczegóły techniczne, pragnącym poznać produkt, branżę, historię, użytkowników itp. Sprzedawca powinien dostarczać osobie o tym charakterze ciekawych informacji. Klient Inspirujący jest wspaniałym słuchaczem, ale to co go najbardziej zniechęca to nuda.

Równoważący - Osoba Równoważna kieruje się w życiu zasadami moralnymi, regułami, poczuciem sprawiedliwości i równości. Nie lubi kłamstwa, łamania reguł, wyzyskiwania, oszukiwania. W sytuacji handlowej osoba o charakterze Równoważnym będzie się kierować sprawiedliwością - nie będzie bardzo targować się o cenę, jeśli wie, że nie ma powodów by kupić coś taniej. Z drugiej strony, jeżeli kogoś zakwalifikuje jako osobę nieuczciwą

to nie będzie robiła z nim interesów. Sprzedawca w wypadku klienta o charakterze Równoważącym powinien zachowywać się przewidywalnie i zgodnie z przyjętymi regułami w handlu. Nie może okazać się nieuczciwy, nawet jeśli miałyby ta nieuczciwość dotyczyć czegoś albo kogoś innego niż klient.

Weryfikujący - Jest to osoba, która posiada zdolność dostrzegania niedoskonałości i wad oraz ma potrzebę dzielenia się tym. Osoba Weryfikująca lubi być doceniana za swoją umiejętność, gdyż uważa, że jej uwagi przyczyniają się do poprawy otoczenia, dlatego nie zwraca uwagi na grożący konflikt. Osoba Weryfikująca nie lubi, gdy jej uwagi nie mają posłuchu lub ktoś wchodzi z nią w polemikę. W sytuacji handlowej klient Weryfikujący może być dobrodziejstwem. Gdy zostanie odpowiednio potraktowany, jego uwagi mogą się przyczynić do poprawy przedsiębiorstwa. Sprzedawca nie powinien argumentować aktualnej sytuacji, ale wysłuchać, podziękować i okazać szacunek zapewniając o doniosłości tych cennych uwag. Taka postawa sprzedawcy bardzo zjedna sobie klienta Weryfikującego.

Systematyczny - Osoba o typie Systematycznym ma potrzebę porządkowania i układania wszystkiego w całość logiczną, chronologiczną itp. Dbą o to, żeby jej model poznawczy rzeczywistości był wewnętrznie spójny i systematycznie rozwijany. Bardzo nie lubi sprzeczności i zmian, gdyż zakłóca to jej wypracowany porządek. Lubi spokój i przewidywalność. Nie lubi chaosu, spontaniczności i dygresji, które ją wybijają z rytmu. W sytuacji handlowej Systematyczny klient podaje swoje potrzeby w sposób uporządkowany i zupełny. Takiego odniesienia wymaga także od sprzedawcy. Najlepiej rozumie ofertę, gdy ta jest logicznie uporządkowana i odzwierciedla jego system oraz kategorie poznawcze. Sprzedawca powinien odzwierciedlić wyliczenia tego klienta i poprowadzić rozmowę według logicznych zależności. Nie warto rozpoczynać nowych wątków, tylko postępować według narzuconego przez klienta planu.

2.3.4 Podsumowanie

W dobie wielkiej konkurencji na rynku i przybliżonej oferty firm, to obsługa klienta jest częstym czynnikiem, który może przechylić szalę powodzenia w handlu. Klient dobrze obsłużony, czyli taki, którego potrzeby - i te dotyczące pożądanego produktu i te głębsze, emocjonalne - zostały zaspokojone. Klient usatysfakcjonowany to klient, który będzie wracał i jest wysokie prawdopodobieństwo, że stanie się klientem lojalnym. A jak już to zostało przedstawione we wcześniejszych rozdziałach o filozofii CRM (*Customer Relationship Management*), klienci

lojalni są bardziej wartościowi i często ich obecność decyduje o wartości całej firmy.

2.4 Chatbot

2.4.1 Co to jest chatbot

Chatbot (inaczej też chatterbot lub linguabot) jest to program komputerowy symulujący człowieka, z którym można rozmawiać w języku naturalnym. Rozmowa najczęściej przebiega przez interfejs tekstowy, a także, coraz częściej, przez interfejs głosowy (dzięki dynamicznemu rozwojowi aplikacji rozpoznających mowę [31] [32]). Zadaniem takich programów jest symulacja inteligentnej rozmowy z człowiekiem na różne tematy. [33]

Ideał chatbota jako programu symulującego inteligencję i sposób myślenia człowieka nakreślił już w roku 1950 jeden z twórców informatyki - Alan Turing (1912 - 1954). W ramach badań nad sztuczną inteligencją wymyślił on tzw. Test Turinga do określenia zdolności maszyny w posługiwaniu się językiem naturalnym, co pośrednio ma też dowodzić opanowania przez nią umiejętności myślenia podobnego do ludzkiego.

Podczas wykonywania testu Turinga grupa sędziów (ludzi) prowadzi, przez terminal, konwersację z wieloma rozmówcami. Gdy nie są w stanie odróżnić rozmowy z prawdziwym człowiekiem od rozmowy z programem komputerowym uznaje się, że test został zaliczony pomyślnie. Alan Turing przewidywał, że w roku 2000 programy komputerowe typu chatbot, będą w stanie oszukać 30% sędziów. Jednak do dzisiaj, żaden program symulujący inteligentną rozmowę nie zaliczył Testu Turinga. Można się spierać czy Test Turinga rzeczywiście sprawdza inteligencję maszyny, czy tylko zaimplementowanie sprytnego mechanizmu rozmowy, który umiejętnie oszuka człowieka. Niemniej jednak, jest to istotny impuls do rozwoju technologii chatbotów. Od roku 1990 została ustalona nagroda im. Loebnera w wysokości 100.000 dolarów dla pierwszego programu, który przejdzie pomyślnie test Turinga. Corocznie przyznawane są też pomniejsze nagrody programom typu chatbot, które były najbliższe zaliczenia testu Turinga. Od lat faworytem i parokrotnym zwycięzcą w tym konkursie jest chatbot A.L.I.C.E. [34] [35]

2.4.2 Historia chatbotów

Pierwszym chatbotem był skonstruowany przez Joseph Weizenbaum program symulujący sesję u psychoterapeuty - ELIZA (nazywany też Doctor). Program ELIZA powstał we wczesnych latach 60 - tych w laboratorium sztucznej inteligencji na uniwersytecie Massachusetts Institute of Technology (MIT AI lab).

Zasada działania była stosunkowo prosta - wejściowy tekst użytkownika był analizowany pod kątem wyszukiwania prostych konstrukcji gramatycznych i niewielkiej ilości słów kluczowych zawartych w tekście (dotyczących komputerów, członków rodziny i emocji). Następnie na tej podstawie dopasowywano wzorce odpowiedzi. Chatbot ELIZA łatwo było wprowadzić w błąd i spowodować wysłanie bezsensownej wypowiedzi. Np.:

chatbot>> How do you do. Please state your problem.

człowiek>> I spend too much time with computers.

chatbot>> Why do you mention computers ?

człowiek>> You are one.

chatbot>> Does it please you to believe I am one ?

człowiek>> Yes.

chatbot>> You seem to be quite positive.

człowiek>> I suppose I am.

chatbot>> How long have you been?

Ilustracja bezsensownej reakcji na konstrukcje gramatyczną „I am”. ELIZA po zwrocie „I am” spodziewała się słowa oznaczającego emocje.

Okazało się jednak, że wielu nietechnicznych pracowników MIT AI Lab rozmawiając z ELIZA uwierzyło, że jest to prawdziwy psychoanalityk i podało jej wiele osobistych informacji. Weizenbaum był tym wydarzeniem zszokowany na tyle, że zamknął projekt ELIZA i zaangażował się pracą nad filozoficznymi problemami zastępowania człowieka przez maszynę. W 1976 została wydana książka jego autorstwa pt. „Computer Power and Human Reason” na temat wpływu komputerów na społeczeństwo, gdzie traktował rozwój technologii sztucznej inteligencji jako coś zagrażającego ludzkości. Z dzisiejszego punktu widzenia ta reakcja Weizenbaum’a jest postrzegana jako dość osobliwa [36].

2.4.3 Opis A.L.I.C.E i AIML

Jednym z popularniejszych i najbardziej uznanych chatbotów, który realizuje rozmowę na dowolny temat, jest chatbot A.L.I.C.E. A.L.I.C.E. został stworzony przez dr Richarda Wallace’a i wygrał on już trzykrotnie konkurs im. Loebner’a (w roku 2000, 2001, 2004) [37].

Dzieło Wallance jest popularne również z powodu stworzenia otwartego języka - AIML (Artificial Intelligence Markup Language). AIML jest kompatybilnym z XML językiem, który

służy do definicji inteligencji chatbota. Jest łatwy w nauce i pozwala w prosty sposób na tworzenie bazy wiedzy chatbota przez ludzi nie znających się na informatyce [38]. Język ten jest otwarty i rozwijany przez społeczność wolnego oprogramowania - Alicebot.AIML.

Do najważniejszych elementów języka AIML należą:

- `<aiml>`: Znacznik zaczynający i kończący dokument AIML.
- `<category>`: Znacznik oznaczający pojedynczą „jednostkę wiedzy” chatbota.
- `<pattern>`: Zawiera wzór wypowiedzi użytkownika
- `<template>`: Zawiera odpowiedź chatbota na rozpoznany wzór wypowiedzi.

Jest też około 20 innych znaczników, jak na przykład: `<that>`, `<topic>`, `<get>` itp..

```
<category>
  <pattern>WHAT ARE YOU</pattern>
  <template>
    <think><set name="topic">Me</set></think>
    I am the latest result in artificial intelligence,
    which can reproduce the capabilities of the human brain
    with greater speed and accuracy.
  </template>
</category>
```

Oto przykładowa jednostka wiedzy chatbota zapisana w języku AIML. Jest to reguła, która powoduje wysłanie odpowiedzi „I am the latest result in artificial intelligence, which can reproduce the capabilities of the human brain with greater speed and accuracy.”, gdy w wyniku analizy tekstu wysłanego przez użytkownika, zostanie zidentyfikowany wzorzec odpowiedzi „WHAT ARE YOU”.

```
chatbot>> Hello
człowiek>> What are you?
chatbot>> I am the latest result in artificial intelligence,
which can reproduce the capabilities of the human brain
with greater speed and accuracy.
```

W języku AIML jest zapisana cała logika (inaczej zwana inteligencją) chatbota. Sam chatbot odgrywa rolę jedynie interpretatora tego języka. Z punktu widzenia tej pracy magisterskiej, najważniejsze są mechanizmy przetwarzania bazy wiedzy chatbota zastosowane w języku AIML. Przeanalizowanie tych mechanizmów pozwoli lepiej zaprojektować chatbot będący rezultatem niniejszej pracy.

Najważniejsze mechanizmy języka AIML:

- Kategorie
- Rekurencja
- Kontekst

Kategorie - Jest to sposób zapisu pojedynczej jednostki wiedzy w AIML. Znacznikiem rozpoczynającym i kończącym kategorię jest `<category>`. Wewnątrz niej zawsze jest zawarty wzorzec wypowiedzi klienta (znacznik `<pattern>`) i treść odpowiedzi chatbota dla tej konkretnej sytuacji (znacznik `<template>`). Język wzorców języka AIML składa się ze słów, spacji, znaków specjalnych „-” i „*”. Słowa są oddzielone spacjami. Najprostszy przykład:

```
<category>
<pattern>Witaj</pattern>
<template>Witaj! Jak się masz?</template>
</category>
```

Działanie tej jednostki wiedzy w praktyce:

```
człowiek>> Witaj
chatbot>> Witaj! Jak się masz?
```

Rekurencja - Bez rekurencji trzeba by było tworzyć osobną Kategorię na każdą możliwą wypowiedź użytkownika. A prawda jest taka, że można na bardzo wiele sposobów sformułować wypowiedź, na którą jest ta sama odpowiedź. Dzięki rekurencji można budować w języku AIML bazę wiedzy, która jest powiązana relacjami i ma „inteligentniejszą” strukturę. Wykorzystywanym znacznikiem w rekurencji jest `<srai>` i oznacza referencję do innej jednostki wiedzy (kategorii).

Mechanizm rekurencji przejawia się w szeregu konkretnych zastosowań języka AIML:

- Symboliczna Redukcja
- Dziel i Zwyciężaj
- Synonimy
- Błędy ortograficzne i gramatyczne
- Słowa kluczowe
- Warunki
- I wszystkie kombinacje powyższych metod

Symboliczna Redukcja - Redukowanie złożonych wzorców do prostszych. Np. szybciej jest operować wzorcem „KTO TO JEST SOCRATES” niż „CZY WIESZ KTO TO JEST SOCRATES”. Redukcja w języku AIML:

```
<category>
<pattern>CZY WIESZ KTO TO JEST *</pattern>
<template><srai>KTO TO JEST <star/></srai></template>
</category>
```

Jeżeli użytkownik zada pytanie „czy wiesz kto to jest Jakaś Osoba”, to chatbot odpowie tak, jakby użytkownik zadał pytanie „kto to jest Jakaś Osoba”. Znacznik <star/> przypisze to, co użytkownik podał po słowach „czy wiesz kto to jest”, a co we wzorcu jest oznaczone gwiazdką (*).

Dziel i Zwyciężaj - dzieli wypowiedź użytkownika na dwie lub więcej części i każdą analizuje osobno. Przykład:

```
<category>
<pattern>YES *</pattern>
<template><srai>YES</srai> <srai><star/></srai></template>
</category>
```

Ta kategoria będzie reagować na przykładową wypowiedź użytkownika „Tak byłem” przez osobne szukanie jednostki wiedzy dla wypowiedzi „Tak” i dla wypowiedzi „byłem”.

Synonimy - Przyporządkowuje różne sposoby wypowiedzenia tego samego do jednej odpowiedzi. Np.

```
<category>
<pattern>HELLO</pattern>
<template>Hi there!</template>
</category>
<category>
<pattern>HI</pattern>
<template><srai>HELLO</srai></template>
</category>
<category>
<pattern>HI THERE</pattern>
<template><srai>HELLO</srai></template>
</category>
<category>
<pattern>HOWDY</pattern>
<template><srai>HELLO</srai></template>
</category>
<category>
<pattern>HOLA</pattern>
<template><srai>HELLO</srai></template>
</category>
```

Błędy ortograficzne i gramatyczne - przywydywanie możliwych literówek i błędów gramatycznych popełnionych przez użytkownika. Na przykład:

```
<category>
<pattern>HELO</pattern>
<template><srai>HELLO</srai></template>
</category>
<category>
<pattern>HELLO</pattern>
<template>Hi there!</template>
</category>
```

Gdy użytkownik popełni literówkę przy słowie „jesteś” chatbot i tak zrozumie o co chodzi.

Słowa kluczowe - Często jest potrzeba reagowania na słowo kluczowe, które wystąpi w dowolnym miejscu zdania. Ta reakcja jest taka sama bez względu na to, jakie inne słowa i zwroty występowały w danej wypowiedzi użytkownika. Przykład:

```
<category>
<pattern>MAMA</pattern>
<template> Opowiedz o swojej rodzinie. </template>
</category>
<category>
<pattern>_ MAMA</pattern>
<template><srail>MAMA</srail></template>
</category>
<category>
<pattern>MAMA _</pattern>
<template><srail>MAMA</srail></template>
</category>
<category>
<pattern>_ MAMA *</pattern>
<template><srail>MAMA</srail></template>
</category>
```

Warunki - Dzięki dodatkowemu znacznikowi <get> można stworzyć wyrażenie warunkowe na podstawie tego, co jest w tekście wejściowym od użytkownika. Na przykład można stworzyć wyrażenie warunkowe typu if - else. Instrukcja <get> zapisuje fragment wypowiedzi użytkownika do lokalnej zmiennej.

```
<category>
<pattern>CO TO JEST TO</pattern>
<template><srail>COTOJEST <get name="to"/></srail></template>
</category>
<category>
<pattern>COTOJEST *</pattern>
<template>To jest <get name="to"/>.</template>
```

```
</category>
<category>
<pattern>COTOJEST NIEZNANY</pattern>
<template>Nie wiem co to jest</template>
</category>
```

Kontekst - Dzięki mechanizmowi kontekstu chatbot bierze pod uwagę dotychczasowy przebieg rozmowy przy analizie bieżącej wypowiedzi użytkownika. Do mechanizmu kontekstu wykorzystywane są znaczniki `<that>` i `<topic>`. Znacznik `<that>` oznacza kontekst wynikający z poprzedniej wypowiedzi użytkownika. Znacznik `<topic>` ma zakres szerszy i dotyczy ogólnej domeny, w której toczy się dyskusja. Najprostszy przykład działania mechanizmu kontekstu przy zastosowaniu znacznika `<that>`:

```
<category>
<pattern>TAK</pattern>
<that>CZY LUBISZ FILMY</that>
<template>Jaki jest twój ulubiony film?</template>
</category>
```

Chatbot na wypowiedź użytkownika „tak”, zareaguje zapytaniem o ulubiony film tylko, gdy to „tak” jest odpowiedzią na pytanie sprecyzowane w tej jednostce wiedzy. Przykład zastosowania mechanizmu kontekstu przy znaczniku `<topic>`.

```
<topic name="CARS">
<category>
<pattern>*</pattern>
<template>
<random>
<li>What's your favorite car?</li>
<li>What kind of car do you drive?</li>
<li>Do you get a lot of parking tickets?</li>
<li>My favorite car is one with a driver.</li>
</random>
</template>
```

Jeżeli aktualna rozmowa jest prowadzona w temacie samochodów (<topic name="CARS">) na dowolną wypowiedź użytkownika chatbot reaguje jedną z czterech powyższych wypowiedzi.

2.4.4 Nowoczesne zastosowania chatbotów

Chatboty rozmawiające o wszystkim, są zazwyczaj projektami akademickimi mającymi na celu przejście testu Turinga i zademonstrowanie rozwoju jednej z gałęzi informatyki, jaką jest sztuczna inteligencja. W zastosowaniach komercyjnych ogranicza się zakres wiedzy chatbota jedynie do jednej lub kilku domen. Na dzisiejsze możliwości, efektywne funkcjonowanie chatbota jest możliwe jedynie przy zawężonej tematyce. Wtedy rozwiązuje się takie problemy lingwistyki komputerowej, jak np. wieloznaczność języka.

Najczęstszymi zastosowaniami komercyjnymi chatbotów są:

Chatboty informacyjne - Są to zazwyczaj proste chatboty operujące na konkretnej bazie danych. Działają na zasadzie pytanie - odpowiedź i nie ma potrzeby na wchodzenie w głębszy dialog z rozmówcą. Dostarczają podstawowe informacje o ofercie i usługach firm. Są bardzo pomocne w odciążaniu pracowników, dzięki czemu ci, mogą być oddelegowani do spraw bardziej skomplikowanych. Chatboty tego typu spotyka się też na stronach Internetu, gdzie pomagają w odnalezieniu poszukiwanej informacji. Chatbot tego typu można znaleźć na stronie IKEA. Chatbot informacyjny jest rodzajem spersonifikowanej interakcyjnej wyszukiwarki.

Chatboty - operatorzy - Są to chatboty, które prowadzą klienta przez proces realizacji pewnego zamówienia. Może być to rezerwacja biletu, zakup produktu, zgłoszenie usterki w serwisie. Chatboty te są zazwyczaj bardziej rozbudowane od informacyjnych, gdyż obsługa klienta w takim wypadku składa się z szeregu kroków, które muszą być kontrolowane przez chatbota.

Chatbot sprzedawca - Chatbot ten umożliwia zapoznanie się z ofertą i potrafi poprowadzić klienta przez cały proces zakupu towaru. Do zadań chatbota sprzedawcy należy ponadto: poznanie dobrze wymagań klienta, umiejętne prowadzenie z nim dialogu tak, by jak rasowy sprzedawca zadowolić klienta i jednocześnie wypracować zysk dla firmy. Chatbot sprzedawca łączy w sobie chatbota informacyjnego, operacyjnego, a także elementy wiedzy psychologicznej i umiejętności negocjacji.

Choć badania nad dziedziną sztucznej inteligencji i nad chatbotami mają już długą historię, to ostatnie zmiany w podejściu do klienta i szybki rozwój Internetu stworzyły warunki do dynamicznego rozwoju chatbotów w zastosowaniach komercyjnych. Rozwój Internetu, a zwłaszcza tzw. semantic Web, może doprowadzić do tego, że w przyszłości nie będzie już istnieć strona bez inteligentnego asystenta. Pewien obraz przyszłości szkicują badania przeprowadzane na dzieciach w wieku 9 - 13 lat, które nie pamiętają epoki przedinternetowej. Okazuje się, że coraz częściej zwracają się one do wyszukiwarek internetowych typu Google.com pełnymi zdaniami - tak jak do człowieka. Możliwe, że Internet zmieni swoje oblicze pod wpływem tego pokolenia, które dopiero dorasta.

2.5 Ruby on Rails

Jednym z celów niniejszej pracy było bliższe poznanie Ruby on Rails. Autorów zaintrygowało duże zainteresowanie tą technologią wśród programistów, która przebojowo wkroczyła do branży IT i błyskawicznie zdobyła rzeszę zwolenników. Jako dowód niech posłużą poniższe cytaty osób, które jej używają na co dzień:

„Rails to najbardziej przemyślany framework do tworzenia aplikacji sieciowych jakiego kiedykolwiek używałem. Mówię tu o dekadzie zarabiania na życie tworzeniem aplikacji. Tworzyłem swoje własne framework'i, pomogłem rozwinąć Servlet API i napisałem dosyć sporo serwerów WWW od początku. Nikomu jeszcze nie udało się stworzyć czegoś takiego jak Rails.” [39]

„Przed powstaniem Ruby on Rails, programowanie aplikacji internetowych wymagało sporo rozwlekłości, kroków i czasu. Teraz projektanci web i programiści mogą stworzyć serwis internetowy o wiele szybciej i prościej, skupiając się bardziej na tym jak być bardziej produktywnym i efektywnym w swojej pracy.” [40]

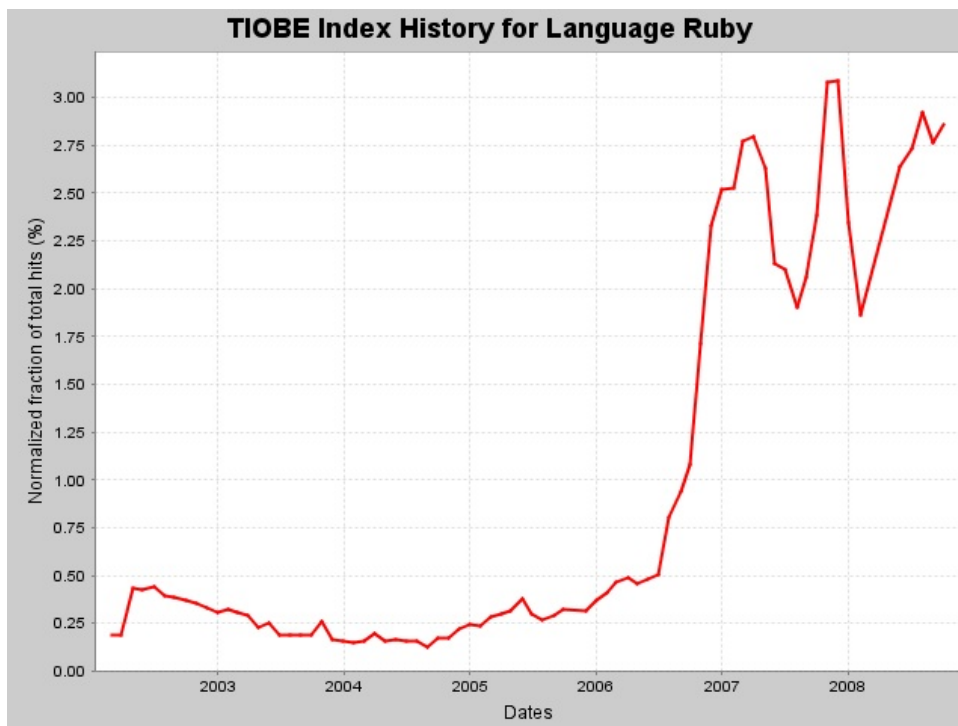
Autorzy postanowili sprawdzić prawdziwość powyższych opinii i zastosować to środowisko do zbudowania aplikacji internetowej, o której traktuje praca magisterska.

2.5.1 Język Ruby

Poznając Ruby on Rails nie sposób wspomnieć o języku, w którym został stworzony.

Ruby to w pełni obiektowy język programowania. Powstał w Japonii w 1993 roku jako połączenie najlepszych cech języków Perl, Smalltalk, Eiffel, Ada i Lisp, a jego autorem jest Yukihiro Matsumoto.

Przez długi czas zainteresowanie językiem Ruby było niewielkie. Prawdziwy „boom” nastąpił wraz z wydaniem Ruby on Rails. Najlepiej obrazuje ten fakt wskaźnik TIOBE 6, który mierzy rozwój języków programowania. Aktualnie Ruby znajduje się na 10 miejscu.



Rysunek 6: Wykres popularności języka Ruby według rankingu TIOBE

Niezaprzeczalną zaletą języka jest jego prosta i czytelna składnia. Nawet osoba nie mająca nic wspólnego z dziedziną programowania, jest w stanie zrozumieć poniższy przykład:

```
5.times { print "We *love* Ruby -- it's outrageous!" }
```

Zasadniczą cechą języka jest jego obiektowość i elastyczność. W Ruby’m wszystko jest obiektem, który można dowolnie modyfikować, dodając bądź odejmując kolejne właściwości. Co więcej, sam język pozwala zmieniać swoje części w zależności od woli programisty.

Przewagę Ruby’iego nad innymi językami programowania docenił rynek. Strona domowa [41] podaje, że jest on używany przez Centrum Badawcze NASA oraz grupę badawczą w Motoroli. Został również wykorzystany do implementacji centralnego systemu zbierania danych w Level 3 Communications Unix Capacity and Planning [42]. Systemu, który gromadzi statystyki wydajności z ponad 1700 serwerów Unix’a (Solarisa i Linuxa) rozmieszczonych na całym globie. Wśród aplikacji internetowych Ruby stanowi podstawę dla Basecamp - aplikacji wspomagającej zarządzanie projektem, wyszukiwarki ogłoszeń mieszkaniowych Hogo.pl oraz popularnego Twitter’a.

Ruby można łączyć z innymi językami. Firma Sun zatrudniła zespół programistów odpowiedzialnych za stworzenie interpretera Ruby, napisanego w Java o nazwie JRuby. To rozwiązanie pozwala wykorzystać bardzo rozbudowaną bibliotekę Java także z poziomu Ruby. Firma Apple wspiera projekty MacRuby i RubyCocoa umożliwiające tworzenie natywnych aplikacji dla systemu OS X.

2.5.2 Środowisko Ruby on Rails

Ruby on Rails [43] to framework napisany w języku Ruby w celu szybkiego i wydajnego tworzenia aplikacji internetowych. Swoją genezę ma w 2004 roku, kiedy programista David Heinemeier Hansson zapowiedział jego wydanie jako Open Source.

Główne założenia Ruby on Rails:

- reguła DRY (ang. Don't Repeat Yourself) - Nie Powtarzaj Sie

Zasada ściśle powiązana z Ruby i rozwijana przez Ruby on Rails. Zakłada odciążenie programisty od powtarzalnych i żmudnych czynności m. in. budowy szkieletu aplikacji czy konfiguracji.

- strategia „Convention Over Configuration”

Brak konieczności edycji rozbudowanych plików konfiguracyjnych. W Ruby on Rails nie ma konieczności edycji mało czytelnych plików XML. Konfiguracja odbywa się na zasadzie przyjętych konwencji.

- możliwość wykorzystania wtyczek

Istnieje wiele rozwiązań problemów napotykanym przez innych programistów, które można łatwo zintegrować z tworzoną aplikacją, np. logowanie, tagowanie, zarządzanie obrazkami.

Ruby on Rails zapewnia wszystko, co jest potrzebne do napisania dojrzałej aplikacji internetowej. Posiada wbudowane moduły obsługi szablonów (ang. templates), bazy danych, e-maili, JavaScript'u, web services, a także popularnych testów tworzonych w języku Ruby.

W rozdziale 3.7.2 przy okazji omówienia technologii użytych do stworzenia opisywanej aplikacji, przedstawiono interesujące szczegóły techniczne oraz zamieszczono przykłady fragmentów kodu świadczących o wyjątkowości Rails.

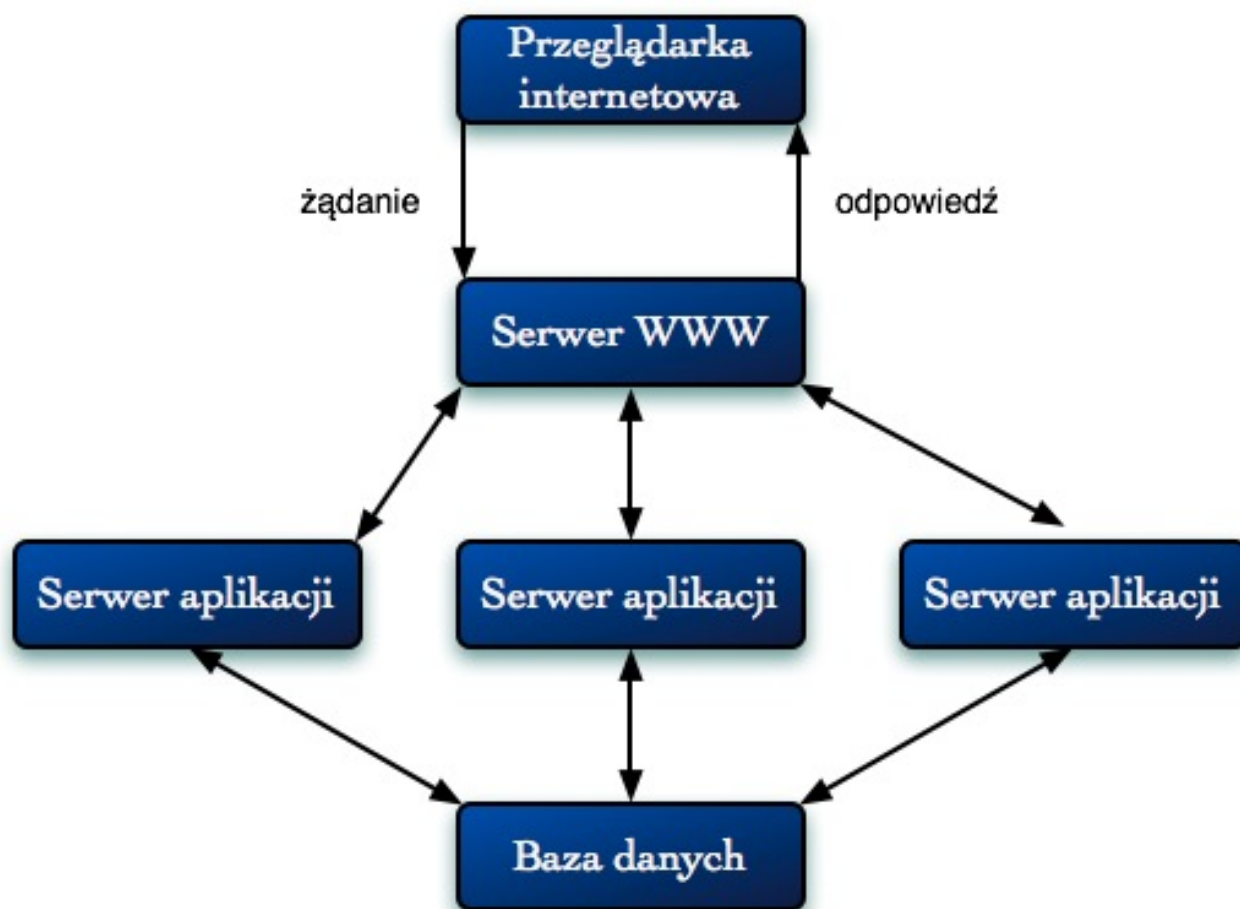
3 Opis systemu

3.1 Architektura systemu

Stworzony system należy do grupy aplikacji internetowych. Autorzy czują się zobowiązani wyjaśnić specyfikę ich budowy przed przystąpieniem do omówienia jego architektury.

Cechą szczególną aplikacji internetowych jest fakt, iż do ich użytkowania nie jest potrzebna instalacja na komputerze. Jest ona przechowywana na centralnym serwerze. Dostęp do niej ma każdy użytkownik posiadający komputer z dostępem do Internetu. Wystarczy w przeglądarce internetowej wpisać adres WWW i otrzymuje się pełną funkcjonalność aplikacji. Wymaga to dużo większego nakładu pracy ze strony programistów w celu zaprojektowania i zorganizowania wydajnej infrastruktury mogącej obsłużyć w ten sposób wielu klientów.

Na rysunku 7 przedstawiono uproszczony schemat budowy takiej struktury. Wejściem do systemu jest odebrane żądanie przeglądarki internetowej, wygenerowane przez kliknięcie użytkownika w link, bądź wpisanie adresu WWW.



Rysunek 7: Struktura systemu

Elementem systemu odbierającym żądania jest Serwer WWW. Ma on za zadanie przyjąć żądanie i wygenerować odpowiedź zrozumiałą dla przeglądarki WWW użytkownika. W tym celu dzieli żądania na te dotyczące treści statycznych, zapisanych na stałe oraz dynamicznych, generowanych na bieżąco, zmieniających się w czasie. Pierwsze z nich jest w stanie wygenerować samodzielnie bez potrzeby absorbowania Serwerów aplikacji. Treści dynamiczne są tworzone przez Serwery aplikacji. Serwer WWW zarządza nimi, przydzielając zadania do wykonania. Dbą także o to, aby żaden z nich nie był przeciążony, a żądania nie czekały długo na obsługę.

Serwer aplikacji jak sama nazwa wskazuje zarządza aplikacją. To w nim wykonuje się kod napisany przez programistów, a także są tworzone treści dynamiczne. Serwer aplikacji przetwarza dane przychodzące z żądaniem i generuje odpowiedź, którą następnie przesyła do Serwera WWW, a ten dalej do klienta. W tym celu jest połączony z bazą danych, która stanowi dla niego źródło informacji.

Baza danych przechowuje informacje potrzebne do wygenerowania treści dynamicznych. Aplikacja posiada dzięki niej dostęp do danych, które można dodawać bądź modyfikować. Baza danych przy częstych i skomplikowanych operacjach nie rzadko staje się najwolniejszym ogniwem całego systemu. Z tego względu tak ważne jest odpowiednie zaprojektowanie jej struktury i optymalizacja wykonywanych poleceń.

Na końcu całego procesu Serwer WWW generuje odpowiedź do przeglądarki użytkownika, który po otrzymaniu interesujących go informacji może wysłać żądanie o kolejne.

3.1.1 Wzorzec MVC

Po opisanii ogólnego działania systemu należy wspomnieć o wzorcu MVC, na którym jest oparta struktura Ruby on Rails, a więc także stworzona przez autorów aplikacja. Odnosi się on do budowy samej aplikacji, którą zarządza opisany wcześniej serwer aplikacji.

MVC (ang. Model - View - Controller) - Model - Widok - Kontroler to wzorzec projektowy. Jego głównym założeniem jest podzielenie aplikacji na trzy odrębne komponenty:

- Model

Odpowiada za połączenie z bazą danych oraz jej mapowanie relacyjno - obiektowe. Implementuje logikę biznesową aplikacji. Modele znacznie upraszczają typowe operacje - zarządzanie danymi, wyświetlanie ze stronicowaniem, a także uniezależniają aplikację od konkretnej wersji bazy danych.

- Widok

Stanowi warstwę prezentacji danych udostępnianych przez model za pośrednictwem kontrolera. Na widok składają się szablony ERB zawierające w przeważającej części kod HTML oraz JavaScript z dodatkami języka Ruby. Są tworzone dynamicznie przez aplikację na podstawie danych przekazanych przez kontroler.

- Kontroler

Jest pośrednikiem pomiędzy warstwą modeli oraz widoków. W zależności od żądania klienta, pobiera odpowiednie dane z modeli, a następnie udostępnia je widokom. Gotowa odpowiedź, w postaci wygenerowanego widoku, jest następnie odsyłana do klienta, który wysłał żądanie.

Istnieje także dodatkowy element nienależący do wzorca MVC, lecz ściśle z nim związany. Jest to moduł routowania. Na podstawie otrzymanego adresu URL decyduje, do którego kontrolera się on odnosi i do niego jest przekazywane żądanie.

Prosty schemat zależności pomiędzy warstwami wzorca MVC jest przedstawiony na rysunku 8.

3.2 Konwersacja

Głównym tematem niniejszej pracy magisterskiej jest konwersacja pomiędzy klientem sklepu internetowego a chatbotem. Cybersprzedawca ma za zadanie poznać potrzeby klienta i zaproponować mu najlepszą ofertę produktów dopasowanych do jego wymagań.

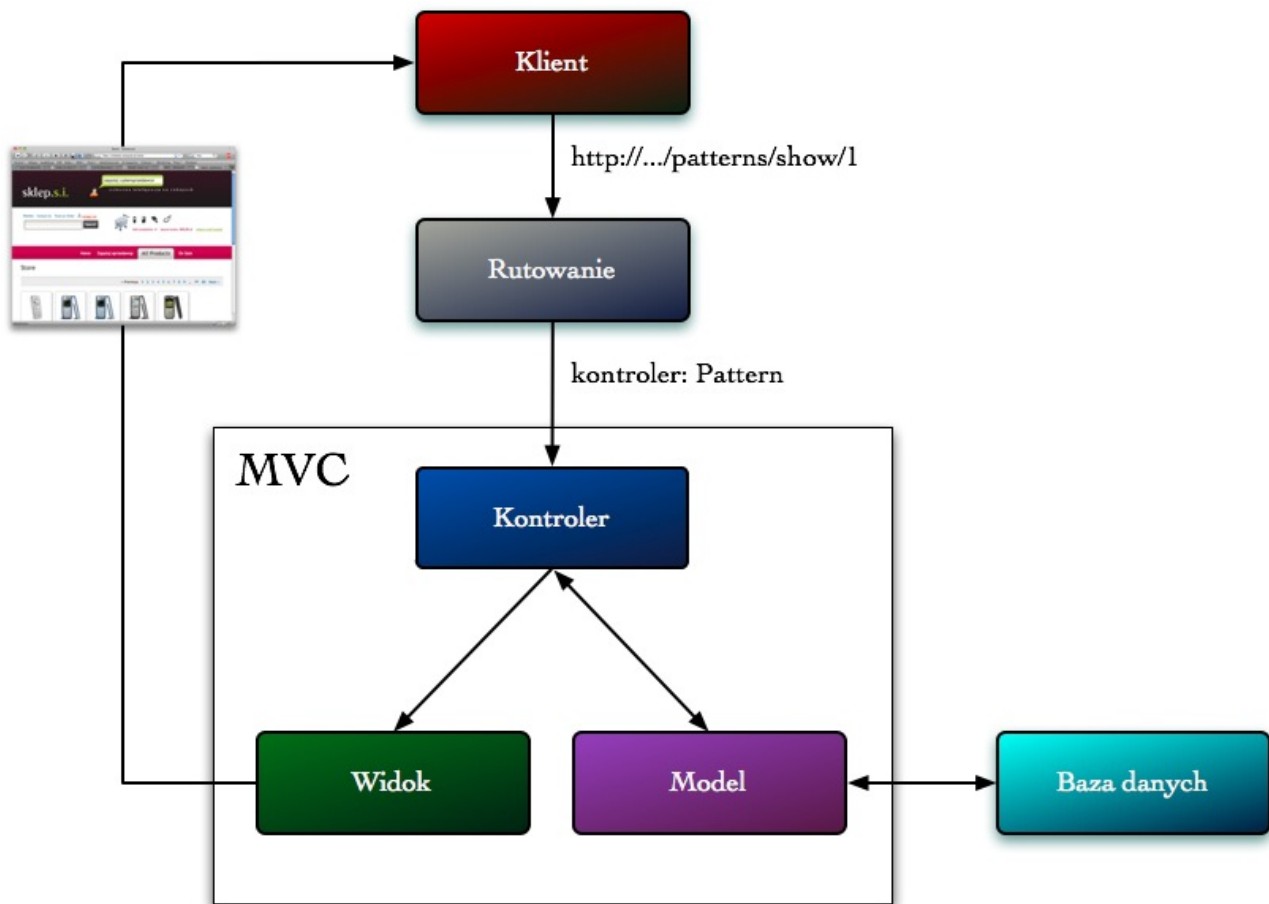
W tym celu został zaprojektowany mechanizm rozmowy 9 za pomocą strony WWW, która została połączona ze sklepem internetowym oferującym telefony komórkowe.

Konwersację zawsze rozpoczyna cybersprzedawca. Jego następne wypowiedzi są już reakcjami na wypowiedzi klienta i zależą od przebiegu rozmowy.

Ostatecznym celem rozmowy jest zaproponowanie klientowi produktów na podstawie przedstawionych przez niego wymagań. Sam sposób prezentacji oferty zależy od rozpoznanego charakteru klienta, co zostanie omówione w dalszej części pracy.

3.3 Mechanizm chatbota (automat Mealy'go)

W najogólniejszym ujęciu systemowym, Chatbot jest taką czarną skrzynką, która na wejściu otrzymuje wypowiedź klienta, a na wyjściu generuje odpowiedź.



Rysunek 8: Działanie aplikacji opartej na wzorcu MVC

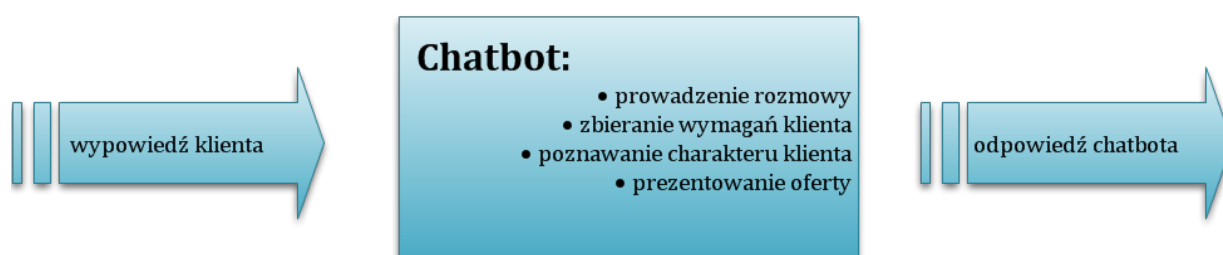
Od odpowiedzi chatbota wymagana jest pewna „inteligencja”, na którą składają się m. in. takie wyznaczniki jak: odpowiedź na temat, odpowiedź uwzględniająca wszystkie zebrane do tej pory informacje o kliencie, umiejętne prowadzenie rozmowy handlowej.

Chatbot jest w istocie automatem stanowym, a dokładnie jest to automat Mealy’ego. Poniżej zostanie przedstawiony automat Mealy’ego wraz z odniesieniami do chatbota zrealizowanego podczas tej pracy magisterskiej.

- $Z = \{z_1, z_2, \dots, z_n\}$ - zbiór sygnałów wejściowych (wypowiedź klienta)
- $Q = \{q_1, q_2, \dots, q_n\}$ - zbiór stanów wewnętrznych (stan, w którym znajduje się chatbot)
- $Y = \{y_1, y_2, \dots, y_n\}$ - zbiór sygnałów wyjściowych (odpowiedź chatbota)
- Φ - funkcja przejść, $q(t+1) = \Phi[q(t), z(t)]$ (graf opisujący reguły przechodzenia między stanami chatbota)
- Ψ - funkcja wyjść, zależy od stanu, w którym znajduje się automat oraz od sygnału wejściowego, $y(t) = \Psi[q(t), z(t)]$ (reakcja chatbota na treść wypowiedzi klienta w zależności,



Rysunek 9: Uproszczony schemat konwersacji



Rysunek 10: Uproszczony mechanizm działania chatbota

w jakim stanie znajduje się chatbot, ta funkcja bezpośrednio wysyła odpowiedź chatbota)

- q_0 - stan początkowy, q_0 należy do zbioru Q (jest to stan przywitania się chatbota)

Z (wypowiedź klienta) - pojedyncza instancja klasy wyszukującej wzorce w tekście klienta.

Wszystkie wzorce są uprzednio zdefiniowane w systemie. Po przyporządkowaniu wzorca do wypowiedzi tekstowej klienta, etykieta wzorca funkcjonuje jako sygnał wejściowy do chatbota.

Q (stan chatbota) - pojedyncza instancja klasy wierzchołka grafu. Jednoznacznie określa położenie automatu i łączy ze sobą sygnał wypowiedzi klienta (odkryty wzorzec) i odpowiedzi chatbota.

Y (odpowiedź chatbota) - zdefiniowane możliwości odpowiedzi chatbota na każde położenie, w którym znajduje się chatbot. Zbiór wypowiedzi chatbota nie jest skończony, gdyż

pobierają one parametr wejściowy z wypowiedzi klienta i dlatego mogą być za każdym razem inne. Instancje klas Z i Y są budowane dynamicznie. Instancja klasy Z (czyli z) powstaje przez analizę wypowiedzi klienta, a Y powstaje przez reakcję na nią. Reakcje mogą być różne w zależności od słów, których użył klient i co o nim już wiadomo (o jego cechach charakteru i preferencjach zakupowych).

Φ - są to przejścia między kolejnymi stanami. Każdy stan ma zbiór odgałęzień do innych stanów, które się uruchamiają w zależności od sygnału wejściowego (wypowiedzi klienta o określonym wzorcu)

Ψ - tabelka, która decyduje o akcji chatbota (wypowiedź i inne akcje) w zależności od wypowiedzi klienta i w jakim się znajduje stanie (dobrym przykładem jest parafraza - funkcja, która za argument przyjmuje wypowiedź klienta.)

Dodatkowo, co nie jest ujęte wprost w tym modelu, dodatkowym argumentem do funkcji przejścia (zmiana stanu chatbota) jak i funkcji wyjścia (akcje chatbota) jest Stan Wiedzy o Kliencie. Są to dane nabywane podczas rozmowy takie jak: ustalone preferencje zakupowe, cechy charakteru itp. Przyjmuje się (gwoili zupełności modelu), że Stan Wiedzy o Kliencie jest przekazywany za każdym razem z wypowiedzią klienta, czyli składa się na wejście Z.

3.3.1 Stan

Stan określa miejsce w rozmowie chatbota i akcje, które powinny być w tym stanie wykonane.

Tablica 1: Schemat tabeli State

Stan
id
label
statement
response
intention_response
return_response
run_code []
successors

id - niepowtarzalny identyfikator stanu

label - niepowtarzalny identyfikator słowny stanu

statement - wzorzec wypowiedzi klienta, który uruchamia dany stan

response - przyporządkowana odpowiedź chatbota

intention_response - przyporządkowana odpowiedź, gdy główna spotkała się z odmową

return_response - odpowiedz, gdy nastąpił powrót do stanu z toru pobocznego (opisane w dalszej części pracy)

topic - określa kontekst stanu (czy należy do głównego czy pobocznego toru rozmowy)

run_code - kod uruchamiany, gdy automat znajduje się danym stanie

successors - rozgałęzienia do innych stanów i warunki

3.3.2 Graf przejść między stanami (funkcja przejścia Φ)

Graf przejść między stanami określa logikę systemu. Przejścia między stanami i warunki tych przejść są definiowane w stanach w atrybucie *successors* i za pomocą kodu zapisanego w atrybucie *run_code* stanu. Jest to bardzo elastyczne rozwiązanie pozwalające na łatwą rozbudowę chatbota, a nawet na całkowitą zmianę dotychczasowego zachowania.

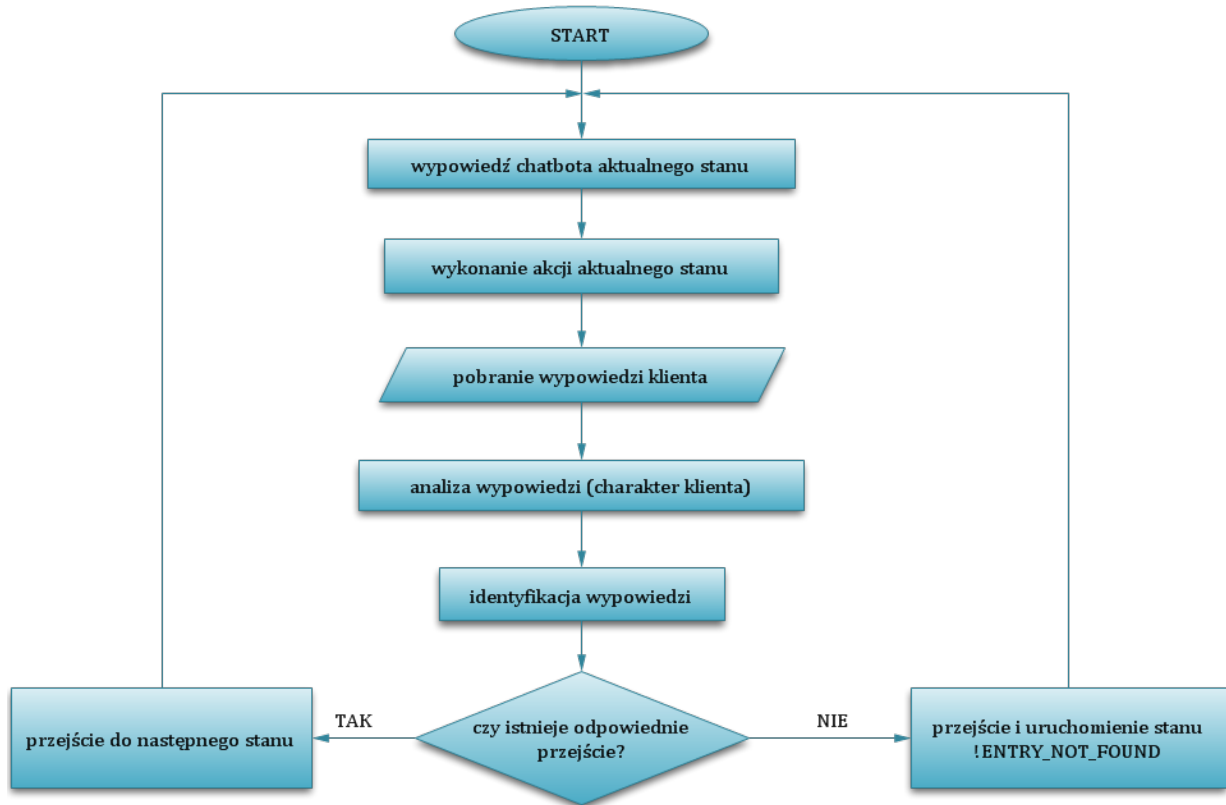
Aktualny graf składa się z dwóch części:

Tor główny rozmowy to rozpisany (poniżej) dokładny przebieg kontaktu klienta z chatbotem, który jest podzielony na większe etapy (np. przywitanie, pytania otwarte, pytania zamknięte, prezentacja oferty) i mniejsze, bardziej precyzyjne (co zrobić gdy klient podał swoje wymaganie co do telefonu komórkowego).

Tor poboczny to reakcje chatbota na wypowiedzi klienta, które są poza głównym tematem prowadzonej rozmowy. Do takich wypowiedzi zaliczają się: wulgaryzmy, pytania dotyczące sklepu, próby zmiany tematu, oczywiste kłamstwa itp. Stany z toru pobocznego nigdy nie mają kontynuacji, tzn. po wykonaniu akcji z toru pobocznego, zawsze wraca się do ostatniego stanu z toru głównego rozmowy. Do toru pobocznego rozmowy należy też szczególny przypadek, gdy wypowiedź klienta nie zostanie rozpoznana przez system.

3.3.3 Zarządca

Zarządca jest algorytmem, w cyklu którego przebiega cała rozmowa z chatbotem. To ten algorytm decyduje o przejściu między jednym stanem a drugim i o uruchomieniu akcji w aktualnym stanie.



Rysunek 11: Algorytm przechodzenia między stanami

Aktualny stan jest zawsze znany. Jest to jedna ze zmiennych globalnych zapisanych w sesji każdej rozmowy. Zarządca kończy swoją pracę, gdy wykona akcję stanu kończącego rozmowę.

Torem pobocznym rozmowy nazwano reakcje na wypowiedzi klienta, które nie są związane z głównym procesem rozmowy (czyli merytorycznej rozmowy handlowej). Takimi przykładami są: wulgaryzmy, ale też pytania o działalność sklepu, próby wymijania się od odpowiedzi i chęć zakończenia rozmowy przez klienta. Różnica między torem pobocznym rozmowy a torem głównym jest jedynie taka, iż po wykonaniu akcji stanu z toru pobocznego powraca się do ostatniego aktywnego stanu w torze głównym.

3.3.4 Poruszanie się po grafie stanów

W stanach zapisane są instrukcje przejść. Oprócz najprostszego przejścia uwidocznionego w diagramie algorytmu powyżej, są jeszcze inne przejścia zapisane w *run_code* stanu. Do takich

przejsć należą:

- *last_talk_state* - id stanu, który ostatnio wygenerował odpowiedź chatbota
- *roll_back_to_enter* - wróć do stanu, który wywołał grupę stanów (najczęściej dotyczy to toru pobocznego)
- *goto_and_run(id_stanu)* - przejdź do stanu o podanym id i go uruchom
- *goto_and_wait(id_stanu)* - id podanego stanu, pomiń uruchomienie, oczekuj kolejnego sygnału wejściowego do systemu

3.3.5 Wypowiedź klienta

Wypowiedź klienta jest wejściem do systemu chatbota. Jest to ciąg znaków układający się w zdanie w języku polskim. Od poprawnej identyfikacji/rozpoznania wypowiedzi klienta zależy przejście do kolejnego etapu rozmowy.

Rozpoznawanie wypowiedzi klienta jest dokładnie opisane w rozdziale 3.4.

3.3.6 Odpowiedź chatbota

Podobnie jak wzorce wypowiedzi klienta, wszystkie odpowiedzi chatbota są znane i zapisane w bazie danych. Jednak w przeciwieństwie do wzorców wypowiedzi, odpowiedź chatbota ma zapis bardziej elastyczny, gdyż często przy odpowiedziach chatbot bazuje na informacjach pobranych od klienta i musi na bieżąco modyfikować swoją wypowiedź.

Konstruowanie wypowiedzi opisano w rozdziale 3.5.

3.3.7 Przechowywanie informacji zebranych o kliencie

Istotą prowadzenia rozmowy chatbota z klientem jest zebranie informacji o preferencjach i typie charakteru klienta. Preferencje są dwojakie: jednostkowe i generalne. *Jednostkowe* preferencje są to preferencje klienta, o których chatbot dowiaduje się w trakcie pojedynczej rozmowy. Są one osią, wokół której przebiega cała rozmowa kończąca się złożeniem klientowi oferty odpowiadającej jego wymaganiom. *Generalne* preferencje to średnia wypadkowa ze wszystkich preferencji zebranych w dotychczasowych kontaktach klienta z chatbotem. Generalne preferencje są istotne dla systemu CRM, czyli dla utrzymywania i zarządzania kontaktem z klientem.

Preferencje Jednostkowe to w przypadku tego chatbota - wymagania klienta co do telefonu komórkowego, opisane dokładnie w rozdziale 3.4.2.

Wymagania generalne to w tym przypadku charakter klienta, który jak wymagania jednostkowe też jest sygnałem wejściowym do systemu. Informacje o charakterze klienta są zbierane na przestrzeni wielu rozmów i są przypisane do klienta, a nie jak jednostkowe, które są przypisane do rozmowy.

Wszystkie odczytane wyznaczniki charakteru są przechowywane w obiekcie klasy „Conversation_characters”, która jest skojarzona z obiektem konwersacji, a następnie klienta.

Tablica 2: Schemat tabeli Conversation_characters

Conversation_characters
conversation_id
character_id
rank

conversation_id - identyfikator konwersacji

character_id - identyfikator odpowiedniego charakteru klienta

rank - siła danego charakteru klienta wyrażona jako wartość liczbowa

3.4 Rozpoznawanie wypowiedzi klienta

3.4.1 Cechy produktów

Telefony komórkowe są produktami bardzo złożonymi. Na przestrzeni lat z urządzeń posiadających jedynie funkcję rozmowy głosowej stały się kombajnami multimedialnymi, dorównującymi możliwościami komputerom sprzed kilku lat. Z tego względu posiadają wiele różnych cech, z których niektóre są specyficzne tylko dla jednego modelu.

Autorzy w celu zrealizowania projektu byli zmuszeni wybrać uniwersalny zbiór cech wspólnych dla większości modeli telefonów komórkowych. Należało wziąć pod uwagę jak najwięcej takich funkcji, aby klient otrzymał jak najdokładniejszy opis produktu, którym jest zainteresowany.

Po dokładnym zapoznaniu się z rynkiem telefonów komórkowych, zdecydowano się wybrać odpowiedni zbiór cech. Można je podzielić na dwie podgrupy:

Tablica 3: Przykład wypełnionej tabeli rozpoznawanie charakteru klienta

id	nazwa	ilość wyznaczników	lista zakwalifikowanych zwrotów
W	władczy	1	[„chcę”]
D	dostosowujący	0	
WI	maksymalista	15	[„podoba mi się”, „marzyłem o”, „wyobrażałem sobie”, „jakbyśmy” (2), „wszystko”, „bez ograniczeń”, „bajer”, „personalizacja”, „dodatki” (2), „niepowtarzalny”, „wyrafinowany”, „wyjątkowy”]
A	asekuracyjny	1	[„gwarancja”]
H	harmonijny	0	
K	weryfikujący	0	
SY	systematyczny	10	[„potem” (2), „po pierwsze”, „po drugie”, „na koniec”, „porównać” (3), „podobny” (2)]
T	inspirujący	1	[„nowe technologie”]
WS	współczujący	1	[„cześć”]
Z	zadaniowy	0	
P	odkrywczy	5	[„jak” (3), „dlaczego” (2)]
SP	sprawiedliwy	0	

- cechy, które dany produkt posiada lub nie
- wartości określające parametry urządzenia

Pierwszą grupę stanowią następujące cechy:

alarm wibracyjny, bluetooth, dyktafon, edge, gprs, gry, hscsd, kalendarz, kalkulator, kompozytor dzwonek, komunikator internetowy, odbiornik, GPS, organizator, klient e - mail, port podczerwieni, przeglądarka dokumentów, przeglądarka internetowa, odbiornik FM, videorozmowy, pobieranie nowych dzwonek, pobieranie nowych gier, sieć 3G, wiadomości SMS, słownik T9, port USB, odbiornik WIFI, wyjście tv, wbudowany zestaw głośnomówiący

Parametry telefonu określają dodatkowo następujące cechy:

cena, grubość, czas trwania gwarancji, ilość godzin rozmowy na baterii, czas czuwania na baterii, liczba kamer (aparatur fotograficznych), ilość wyświetlaczy, nazwa producenta, pojemność baterii, szerokość, waga, wysokość

Każda cecha produktu jest zapisana w bazie danych. Schemat tabeli „Features” przechowywanej dostępne cechy produktów wygląda następująco:

Tablica 4: Schemat tabeli „Features” przechowującej dostępne cechy produktów

Pole	Typ
id	int (11)
name	varchar (100)
description	varchar (255)
allowed_values	varchar (255)
allowed_operators	varchar (255)
value_type	varchar (255)
weight	int (11)
expression	varchar (255)

id klucz główny tabeli, generowany automatycznie

name nazwa cechy

description opis cechy dla administratora

allowed_values wyrażenie w języku Ruby określające dozwolone wartości, jakie może przyjąć dana cecha, używane przy walidacji sensu wymagań klienta

allowed_operators dozwolone operatory do przyrównywania wartości cechy, także używane przy walidacji wymagań klienta

value_type typ wartości cechy, używane przy walidacji wymagań klienta

weight istotność danej cechy w porównaniu z innymi, wyrażona jako wartość liczbowa

expression wyrażenie odzwierciedlające daną cechę w języku Ruby

Cena

Nazwa: Cena

Identyfikator: product.price

Dozwolone wartości: %f > 0

Dozwolone operatory: ==, >, <, >=, <=, !=

Typ wartości: Numeric

Rysunek 12: Przykład cechy produktu w bazie danych

oznacza, że wymaganie klienta odnośnie ceny produktu musi być wartością liczbową (typ „Numeric” w języku Ruby) większą niż 0 ($\%f > 0$). Dodatkowo klient może wymagać, aby cena urządzenia była równa, większa, mniejsza, większa lub równa, mniejsza lub równa lub różna od zadanej wartości.

3.4.2 Wymagania klienta odnośnie produktu

W poprzednim rozdziale przedstawiono cechy, które opisują produkty w niniejszej aplikacji. Podczas rozmowy z cybersprzedawcą, klient przedstawia swoje preferencje odnośnie produktu, który chciałby kupić. Program musi je gdzieś zapisywać, aby móc na bieżąco analizować wymagania klienta i decydować o dalszym przebiegu konwersacji.

Preferowane przez klienta cechy produktu są zapisywane w tabeli „Requirements” w bazie danych. Jej schemat wygląda następująco:

Tablica 5: Schemat tabeli „Requirements”

Pole	Typ
id	int (11)
value	text
feature_id	int (11)
conversation_id	int (11)
operator	varchar (255)

id klucz główny tabeli, generowany automatycznie

value wartość wymagania, musi zawierać się w zbiorze określonym przez pole „allowed_values” z tabeli „Features” powiązanej cechy

feature_id klucz obcy wskazujący na cechę produktu, do której odnosi się dane wymaganie klienta

conversation_id klucz obcy wskazujący na daną konwersację

operator operator powiązany z wartością wymagania, musi zawierać się w zbiorze określonym przez pole „allowed_operators” z tabeli „Features” powiązanej cechy

Przykładowo, wymaganie posiadające wartości z tabeli 6 oznacza, że klient, w konwersacji nr 1, jest zainteresowany telefonem komórkowym o wartości większej od 100 zł, zakładając oczywiście, że klucz obcy odnosi się właśnie do ceny.

Tablica 6: Przykład wymagania klienta zapisanego w bazie danych

Pole	Wartość
value	„100”
feature_id	10
conversation_id	1
operator	”>”

W przypadku funkcji, które dany produkt może posiadać lub nie, wartość musi być 0 lub 1, a operator znakiem równości bądź różności.

3.4.3 Wzorce dopasowań

Wzorce dopasowania to ciągi wyrazów i odpowiednich wyrażeń zaprojektowanych do rozpoznawania konkretnych bloków tekstu. Stanowią podstawę do rozpoznawania wypowiedzi klienta.

Wzorce są przechowywane w bazie danych. Schemat tabeli przechowującej wzorce wypowiedzi wygląda następująco:

Tablica 7: Schemat tabeli „Patterns” przechowujące wzorce dopasowań

Pole	Typ
id	int (11)
expression	varchar (255)
label	varchar (255)
weight	int (11)
requirements_expression	varchar (255)
context	varchar (255)

id klucz główny tabeli, generowany automatycznie

expression główne wyrażenie wzorca składające się z wyrazów i dodatkowych wyrażeń

label identyfikator wzorca do jego kategoryzacji

weight współczynnik wagi wzorca, brany pod uwagę przy dopasowaniu

requirements_expression wymagania klienta, do których odnosi się wzorzec

context określa grupę wzorców przeszukiwanych w celu dopasowania podwzorców

Wzorce można dowolnie modyfikować z poziomu panelu administratora.

Najważniejszym elementem wzorca jest jego wyrażenie. W najprostszej postaci może ono wyglądać następująco:

dyktafon

Pojawienie się w wypowiedzi klienta słowa „dyktafon” spowoduje dopasowanie do tego wzorca.

W praktyce występują dużo bardziej skomplikowane wzorce, w których pojawiają się wyrażenia dodatkowe. Przykładowo:

cena od {product.price:>}

W powyższym wzorcu pojawia się dodatkowe wyrażenie „product.price:>”. Tego typu wyrażenia zostały wprowadzone w celu pobrania informacji od klienta, która nie może być na stałe zapisana we wzorcu. W powyższym wyrażeniu jest to cena minimalna, jaką klient jest skłonny zapłacić za telefon komórkowy. Wiadomo, że nie jest możliwe zapisanie tego samego wzorca z każdą możliwą wartością, jaką użytkownik mógłby podać jako cenę.

Wyrażenie „product.price:> ” dzieli się na dwie części oddzielone od siebie dwukropkiem. Pierwsza „product.price” to określenie odpowiadającej cechy o takiej nazwie. Drugi człon to operator, do którego zostanie przyrównana wartość podana przez klienta. Wpisanie przez użytkownika poniższego zdania:

cena od 100 złotych

zostanie odczytane przez system jako „product.price>100” i zapisane w tabeli „Requirements”.

Kolejnym wyrażeniem dodatkowym, które pojawia się we wzorcach jest wyrażenie rekurencji. Przykładowo:

[feature1] i [feature2]

Powyższy wzorzec powoduje, że część wypowiedzi klienta po lewej stronie od spójnika „i” oraz po prawej stronie zostanie jeszcze raz przetworzona przez algorytm dopasowania. Kolejne

dopasowane wzorce są przyporządkowane do wzorca nadrzędnego. W ten sposób obiekt dopasowania ma hierarchię odwróconego drzewa z głównym wzorcem jako korzeniem i podwzorcami stanowiącymi gałęzie. Ma to na celu wyeliminować konieczność dodawania kolejnych wzorców różniących się od pozostałych pojedynczymi frazami.

3.4.4 Mechanizm synonimów

W celu ułatwienia i wzbogacenia mechanizmu rozpoznawania wypowiedzi klienta, autorzy zdecydowali się zaimplementować mechanizm synonimów. Tego typu rozwiązanie pozwala na przetwarzanie różnych form i sposobów wypowiedzi posiadających takie samo znaczenie, bez potrzeby dublowania niewiele różniących się wzorców.

Synonim jest obiektem, którego atrybuty przedstawiają się następująco:

word synonim

baseform wyraz bazowy, do którego odnosi się synonim

Wszystkie wyrazy w wypowiedzi klienta zostaną następnie zamienione na swoje odpowiedniki bazowe. W rezultacie we wzorcach nie trzeba używać innych wyrazów niż formy bazowe.

3.4.5 Algorytm odczytywania wypowiedzi klienta

Prawidłowe odczytanie zdania przesłanego przez klienta jest kluczowym zadaniem aplikacji. Błąd popełniony na tym etapie w konsekwencji spowoduje wygenerowanie niepoprawnej odpowiedzi, może nawet zupełnie niezrozumiałej w kontekście jego wypowiedzi.

Uproszczony schemat poszczególnych etapów algorytmu odczytywania wypowiedzi klienta przedstawiono na rysunku 13. Składa się on z następujących etapów:

1. Preprocessing

Faza przygotowawcza do dalszej analizy. Jest odpowiedzialna za usunięcie wszystkich znaków nie znajdujących się na tzw. „białej liście”. Są to litery, cyfry oraz znaki kropki i pytajnika. Pozostałe, według autorów, nie wnoszą żadnej, użytecznej informacji dla dalszego działania algorytmu.

Etap przygotowawczy to także dodawanie spacji pomiędzy znakami a wyrazami bezpośrednio z nimi sąsiadującymi. Jest to konieczne, gdyż dalsza część algorytmu dzieli zdanie

na tzw. tokeny, czyli najmniejsze jednostki logiczne wypowiedzi, po spacjach. Spacja jest dodawana także pomiędzy liczby a wyrazy, np. część „100zł” zostanie przekształcona na „100 zł”.

2. Sprowadzenie do form podstawowych

W tym etapie wszystkie wyrazy zostają sprowadzone do swoich form podstawowych. Np. wyraz „chcę” zostanie zamieniony na „chcieć”. Dodatkowo wszystkie duże litery są zamieniane na małe.

Istnieje wiele wyrazów, które mogą wywodzić się z różnych wyrazów bazowych. Np. wyraz „nie” może być odmianą jednego z podstawowych wyrazów „on”, „ona”, „ono”, „nie”. Tego typu przypadków jest więcej i system musi je wszystkie brać pod uwagę.

Po sprowadzeniu wszystkich wyrazów do form podstawowych, na ich podstawie tworzone są wszystkie możliwe kombinacje zdań, które mogą reprezentować. Przykładowo ze zdania „nie chcę” zostają utworzone cztery zdania z formami podstawowymi do dalszej analizy:

- „nie chcieć”
- „ona chcieć”
- „ono chcieć”
- „on chcieć”

Wygenerowane na tej podstawie zdania są następnie przekazywane do następnego etapu.

3. Podmiana synonimów

Zdania dostarczone przez poprzednią fazę są kolejno analizowane w poszukiwaniu synonimów. W przypadku znalezienia wyrazu należącego do rodziny synonimów, jest on zastępowany jej wyrazem bazowym. Ma to na celu wyeliminowanie niepotrzebnego namnażania wzorców, różniących się pojedynczymi wyrazami bliskoznacznymi.

4. Dopasowanie

Ostatnim, ale również najbardziej skomplikowanym etapem jest proces dopasowania wygenerowanych zdań do wzorców zapisanych w bazie danych.

Zdania z poprzedniej fazy są dopasowywane do wzorców. Polega to na kolejnym porównywaniu wyrazów wzorca z wyrazami w wypowiedzi klienta. Aby dopasowanie było pomyślne, wszystkie wyrazy wzorca muszą znaleźć się w analizowanym zdaniu. Dodatkowo

ich kolejność musi być zgodna ze wzorcem. Wypowiedź klienta może zawierać wyrazy nie uwzględniane przez wzorzec. W tym wypadku są one pomijane.

Dopasowywane są także wyrażenia specjalne dokładnie opisane w rozdziale 3.4.3. Ich dopasowanie także jest obowiązkowe.

Po próbie dopasowania wszystkich wzorców jest robiony ranking dokładności dopasowań. Jest on tworzony na podstawie współczynnika dopasowania. Współczynnik ten jest definiowany jako iloczyn liczby dopasowanych wyrazów wzorca i jego wagi. Należy jednak zaznaczyć, że niedopasowanie któregokolwiek wyrażenia wzorca całkowicie przekreśla dopasowanie. W przypadku, gdy wypowiedź klienta została dopasowana do wzorca, który zakłada wyodrębnienie pewnej jej części i ponowne dopasowanie, współczynniki wszystkich dopasowanych wzorców sumują się.

Ostatecznie wybierane jest zdanie i wzorzec, których dopasowanie posiada największą wartość współczynnika. Oba są następnie przekazywane do Zarządcy decydującego o tym, jak należy zareagować na wypowiedź klienta.

3.4.6 Aktualny kontekst konwersacji

Cybersprzedawca na każdym etapie rozmowy przechowuje aktualny kontekst wypowiedzi. Jest to konieczne, ponieważ identyczna wypowiedź klienta może mieć zupełnie inne znaczenie w zależności od kontekstu, w którym wystąpiła. Przykładowo:

Chatbot: jaki chciałby Pan wyświetlacz?

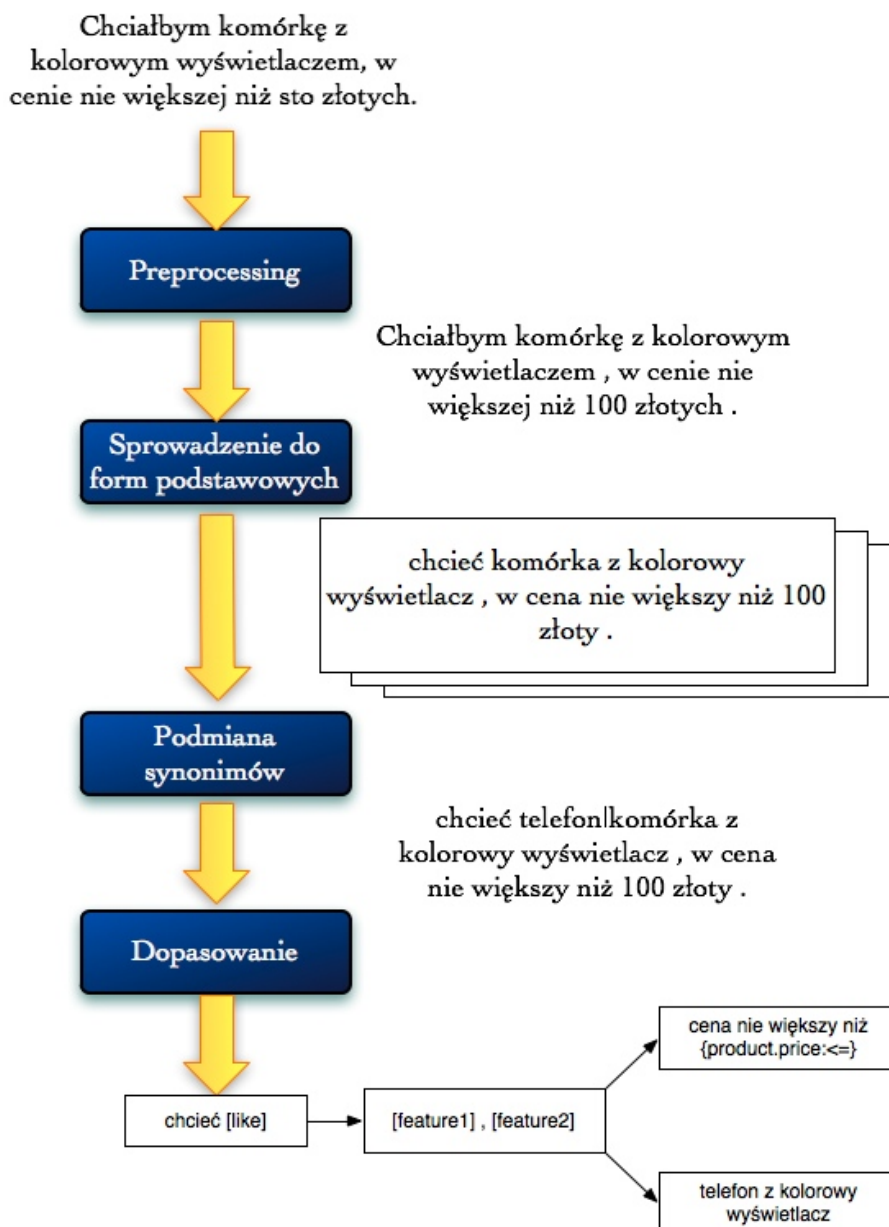
Klient: Koniecznie duży i kolorowy.

Chatbot: jaki chciałby Pan telefon?

Klient: Koniecznie duży i kolorowy.

W obu powyższych przypadkach klient wypowiada dokładnie te same słowa, natomiast ich interpretacja przez aplikację powinna być różna.

Aktualny kontekst konwersacji jest określany przez listę wzorców, do których aplikacja próbuje dopasować wypowiedź klienta. Odnosząc się do powyższego przykładu, w pierwszym przypadku dopasowanie powinno nastąpić do wzorca powiązanego z wyświetlaczem, natomiast w drugim z telefonem.



Rysunek 13: Uproszczony schemat algorytmu dopasowania

Podobna zasada dotyczy dopasowania rekurencyjnego. Wzorzec nadrzędny przekazuje swój kontekst do swojego podwzorca w przypadku kontekstu konwersacji.

3.5 Konstruowanie odpowiedzi

Wypowiedzi chatbota są podzielone na frazy i trzymane w osobnej tabeli bazy danych.

label - jest to niepowtarzalna etykieta frazy. Nazewnictwo etykiety określa hierarchię i sensy frazy, np. *REQUEST_DETAIL_ENUMERATE* - pytanie klienta o wybrany parametr wyczerpieniowy (taki jak model, kolor, marka itp.). Nazewnictwo jest istotne, ponieważ odzwierciedla domenę poruszanych przez chatbota kwestii. Możliwymi wypowiedziami

Tablica 8: Tabela wypowiedzi klienta

label
expression

chatbota są etykiety zaczynające się na:

- REQUEST - prośby o pewną daną (np. imię albo zgoda klienta na postawione pytanie)
- ANSWER - odpowiedź na pytanie klienta
- GOT - potwierdzenie uzyskania informacji od klienta (np. GOT_NAME - uzyskanie imienia, GOT_FEATURE_GLOBAL - uzyskanie ogólnej preferencji klienta)
- OFFER - oferowanie konkretnego produktu klientowi
- END - kończenie (np. END_CONVERSATION - poinformowanie o zakończeniu rozmowy i pożegnanie się)

expression - treść danej frazy. Treść może być zwykłym tekstem np.

```
"Dziękuję, zapamiętałam to sobie"
```

albo może być wyrażeniem z parametrem np.

```
"Dziękuję, przyjęłam Pańskie wymaganie dotyczące %s do wiadomości
```

```
% jakaś_zmienna"
```

może to być także wyrażenie warunkowe

```
if client.sex "Proszę Pani" : "Proszę Pana"
```

Kolejne właściwości fraz:

- Frazy można ze sobą łączyć, np. GOT_REQUIREMENT + REQUEST_MORE = „Dziękuję bardzo, przyjęłam do wiadomości, że jest pan zainteresowany /parametr/. Czy w czymś jeszcze mogę Panu pomóc”
- Treść fraz jest wybierana losowo. Etykieta frazy definiuje sens frazy, którą najczęściej można wypowiedzieć na kilka sposobów, aby zwiększyć wrażenie inteligencji chatbota.

3.6 Dodatkowe funkcjonalności cybersprzedawcy

3.6.1 Rozpoznawanie płci klienta

Płeć klienta posiada czasami kluczowe znaczenie podczas rozmowy ze sprzedawcą. Badania naukowe dowiodły, że klient obsługiwany przez pracownika sklepu będącego płci przeciwnej, jest bardziej podatny na jego wpływ.

W systemie znajdują się zdjęcia przedstawiające mężczyznę i kobietę, używane jako wizerunki cybersprzedawcy podczas rozmowy. Na początku konwersacji, kiedy nie jest znana płeć klienta, standardowo pokazywane jest zdjęcie kobiety. W przypadku, gdy klientem okazuje się być także kobieta, wizerunek cybersprzedawcy zostaje zmieniony. Informuje on klienta o konieczności bardzo pilnego wykonania pewnej czynności i prosi o możliwość przekazania go innemu cybersprzedawcy:

„Przepraszam, ale muszę udać się na pilne spotkanie. Czy zezwoli Pan, aby obsłużyła Pana moja koleżanka. Oczywiście poinformuję ją o wszelkich naszych ustaleniach.”

Zgoda klienta oznacza zmianę cybersprzedawcy. W przypadku odpowiedzi negatywnej, pozostaje on bez zmian.

Od płci klienta zależy także w jakiej formie będzie się do niego zwracał cybersprzedawca, Używane są zwroty grzecznościowe Pan/Pani w odpowiedniej formie. Przykładowo:

- mężczyzna - „Bardzo mi miło Pana poznać.”
- kobieta - „Bardzo mi miło Panią poznać”
- nieznaną płeć - „Bardzo mi miło Pana/Panią poznać.”

Płeć klienta jest rozpoznawana na początku rozmowy. Cybersprzedawca przedstawia się i prosi o to samo swojego rozmówcę. Autorzy zauważyli, iż wszystkie imiona żeńskie kończą się na literę „a”. Na tej zasadzie działa mechanizm rozpoznawania płci. W przypadku, gdy klient odmawia podania swojego imienia, cybersprzedawca zwraca się do niego w dwóch formach jak podano w przykładzie powyżej.

3.6.2 Baza telefonów

Cybersprzedawca funkcjonuje w połączeniu ze sklepem internetowym. Do testowania jego działania, autorzy potrzebowali bazę rzeczywistych telefonów komórkowych. W tym celu został napisany dodatkowy skrypt parsujący zawartość istniejącego na rynku sklepu internetowego

oferującego telefony komórkowe [44]. Skrypt odczytuje wybrane informacje z witryny sklepu i zapisuje do bazy danych opisywanej aplikacji. Dzięki takiemu rozwiązaniu system posiada bazę danych opartą na realnych informacjach bez konieczności ich ręcznego wpisywania. Ostatecznie sklep internetowy zawiera informacje o prawie 800 produktach wraz ze zdjęciami. Ceny telefonów komórkowych zostały wygenerowane losowo z uwzględnieniem liczby posiadanych cech takich jak: GPRS, aparat fotograficzny itd.

3.6.3 Baza imion polskich

Na początku pierwszej rozmowy z klientem, cybersprzedawca przedstawia się i prosi rozmówcę o to samo. Autorzy uznali, że warto wyposażyć aplikację w możliwość weryfikacji, czy imię podane przez użytkownika jest prawidłowe. W przypadku, gdyby cybersprzedawca posługiwał się wyrazem błędnie uznanym za imię rozmówcy, straciłby w jego oczach wiarygodność i zostałby uznany za nierzetelny. Z tego względu opisywana aplikacja posiada bazę imion polskich [45].

Baza danych zawiera ponad 1600 imion w różnych formach. Przykładowo imię „Grzegorz” posiada odmiany takie jak: Grzech, Grzela, Grzesio, Grześ, Grzesiek. W ten sposób imię klienta zostanie rozpoznane bez względu na to jakiej formy użył.

Wiedza na temat występujących imion pozwala cybersprzedawcy odpowiednio zareagować, jeśli użytkownik poproszony o przedstawienie się, poda nieistniejące imię.

3.7 Zastosowane Technologie

3.7.1 MySQL

MySQL to system relacyjnych baz danych. Dane są przechowywane w oddzielnych tabelach zwanych relacjami. Są one powiązane między sobą poprzez zdefiniowane związki. Możliwe jest także łączenie wielu tabel w jedną.

Operacje w bazie danych definiuje się za pomocą języka SQL (ang. Structured Query Language). Jest on najbardziej powszechny i ustandaryzowany.

MySQL jest dostępny na wszystkich popularnych platformach. Jest darmowy, rozpowszechniany na licencji GPL [46], która określa szczegółowo zasady korzystania z systemu. Cieszy się także opinią jednego z szybszych i niezawodnych serwerów bazodanowych. Rozwijany od wielu lat, na przestrzeni których otrzymywał kolejno coraz nowsze funkcje. Aktualnie MySQL można porównać do pierwotnie bardziej skomplikowanego konkurenta, bazy danych PostgreSQL.

Powyższe zalety MySQL sprawiły, iż autorzy zdecydowali się wykorzystać go przy tworzeniu opisywanej aplikacji.

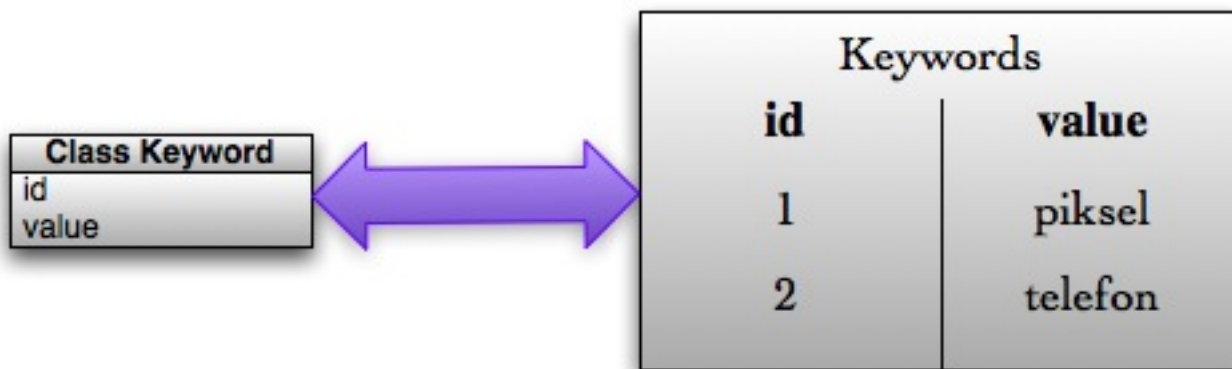
3.7.2 Ruby on Rails

Ruby on Rails jest całkowicie oparty o architekturę MVC. W jego skład wchodzi wiele modułów i klas, które znacznie usprawniają proces rozwoju projektu. Jako najważniejsze należy wymienić: ActiveRecord, ActionController oraz ActionView.

- ActiveRecord

Implementuje modele w Ruby on Rails, które sprawują kontrolę nad bazą danych. ActiveRecord jest mechanizmem ORM (object - relational mapping), tzn. przedstawia dane z bazy danych w formie obiektów. Obiekty te nazywane są modelami i stanowią warstwę modelu we wzorcu MVC.

Każdy model dziedziczy po klasie ActiveRecord. Jedna tabela i wiersz bazy danych to odpowiednio model oraz jego instancja. Opakowuje on odpowiadającą mu tabelę w ten sposób, że kolumny tabeli reprezentowane są przez jego właściwości. Modele automatycznie rozpoznają kolumny danej tabeli oraz ich typy i tworzą odpowiednie metody do wykonywania na nich operacji.



Rysunek 14: Przykład połączenia tabeli z modelem

Prostota tego rozwiązania sprawia, że programista wszelkie operacje w bazie danych wykonuje za pomocą metod obiektów odpowiadających rekordów. Nawet tu jest spełniony postulat konwencji ponad konfiguracją, gdyż przyporządkowanie konkretnej klasy ActiveRecord następuje poprzez łączenie jej nazwy z nazwą tabeli. Nazwa modelu to nazwa tabeli w formie liczby pojedynczej.

Poza możliwością dostępu i modyfikacji danych w tabelach, ActiveRecord udostępnia mechanizmy pozwalające definiować związki pomiędzy tabelami. Za przykład posłuży poniższy kod:

```
class Parent < ActiveRecord::Base
  # liczba mnoga przy określaniu powiązania wielu obiektów
  has_many :children
end

class Child < ActiveRecord::Base
  # liczba pojedyncza przy powiązaniu jednego obiektu
  belongs_to :parent
end
```

Definiuje on dwa modele o nazwie Parent oraz Child. Pomędzy odpowiadającymi im tabelami występuje związek „one to many”. Został on zdefiniowany na poziomie modeli za pomocą komend „has_many” oraz „belongs_to”. Oznacza on, iż obiekt klasy Parent jest związany z wieloma obiektami klasy Child. Z drugiej strony obiekt klasy Child jest związany tylko i wyłącznie z jednym obiektem klasy Parent. Powyższy zapis powoduje automatyczne dodanie metod ułatwiających zarządzanie oboma powiązaniem modelami:

```
# utworzenie nowego obiektu klasy "Parent"
parent = Parent.new
# zapisanie do bazy danych
parent.save
# utworzenie powiązanego obiektu klasy "Child"
child = parent.children.create
# dostęp do powiązanego obiektu
child.parent
```

Kolejną bardzo ważną funkcjonalnością modułu ActiveRecord są walidatory. Ich zadaniem jest dbać o to, aby dane w bazie mieściły się w zdefiniowanym zakresie oraz typie. Ta funkcja jest szczególnie przydatna podczas analizy danych przysłanych przez użytkownika, na podstawie wypełnionego formularza. Aplikacja musi zweryfikować prawidłowość

danych, np. poprawny format adresu e - mail, kod pocztowy, nr telefonu itd. Do tego właśnie służą walidatory. Przykładowo:

```
# "name" oraz "surname" muszą być dłuższe niż 3, a krótsze niż 30 znaków
validates_length_of :name, :surname, :within => 3..40

# "quantity" musi być liczbą
validates_numericality_of :quantity

# "name", "surname" oraz "quantity" nie mogą być puste
validates_presence_of :name, :surname, :quantity

# "code" musi spełniać podane wyrażenie regularne
validates_format_of :code, :with => /[\\d]+\\-\\d]+/

# "format" musi mieć jedną z podanych wartości
validates_inclusion_of :format, :in => %w( jpg gif png)
```

W przypadku, gdy któryś z walidatorów ukończy weryfikację niepomyślnie, zostaje wygenerowany błąd informujący o jego źródle i rodzaju 15.

Nowe Słowo Kluczowe

1 error prohibited this keyword from being saved

There were problems with the following fields:

- Name can't be blank

*Nazwa

Zapisz słowo kluczowe

Rysunek 15: Niepomyślna walidacja modelu

Obiekt, który nie spełnia naniesionych na niego wymagań nie zostanie zapisany do bazy danych.

Częścią składową modułu ActiveRecord są migracje. Za ich pomocą użytkownik z poziomu aplikacji może zarządzać strukturą bazy danych. Migracje umożliwiają m. in. tworzenie tabel, dodawanie nowych kolumn, indeksowanie, usuwanie kolumn.

Migracje określają kolejne wersje struktury bazy danych wraz z jej rozbudową. Użytkownik w każdej chwili może przywrócić bazę danych do wcześniejszej wersji, a potem na nowo ją zaktualizować do bardziej aktualnej wersji.

Każdy plik migracji określa czynności, jakie należy wykonać podczas poruszania się w stronę wyższej wersji struktury oraz odwrotnie:

```
class CreateInflections < ActiveRecord::Migration
  def self.up
    create_table :inflections do |t|
      t.string :value
      t.string :case_name
      t.integer :base_form_id
      t.string :number
    end
  end

  def self.down
    drop_table :inflections
  end
end
```

- ActionView

ActionView to widok według wzorca MVC. Udostępnia narzędzia do budowy częściowych lub całościowych szablonów interfejsu użytkownika. Ścisłe współpracuje z kontrolerem, który dostarcza dane potrzebne do osadzenia w szablonach.

Szablony mogą być generowane na trzy sposoby:

- pliki .erb (lub.rhtml)

Są to zwyczajne pliki z kodem HTML, pomiędzy którym jest włączany kod Ruby.

- pliki .builder

Sposób szczególnie użyteczny do generowania plików XML. Najlepiej wyjaśni to przykład:

```
xml.product {
```

```
    xml.name("nokia")
    xml.price{100}
  }
```

wygeneruje:

```
<product>
  <name>nokia</name>
  <price>100</price>
</product>
```

– pliki .rjs

Szablony do generowania skryptów JavaScript. Używane głównie w połączeniu z technologią AJAX [47].

- ActionController

Odpowiada kontrolerowi we wzorcu MVC. ActionController koordynuje przetwarzanie żądań przychodzących do aplikacji. Przydziela zadania do poszczególnych akcji przez co jego kod jest logiczny i uporządkowany.

ActionController stanowi warstwę pośrednią między ActionView a ActiveRecord. Wykorzystuje on model do wykonywania operacji na danych, aby jego własna struktura była przejrzysta i czytelna. Dane pobrane za pośrednictwem warstwy ActiveRecord przekazuje do widoków, których zawartość przesyła jako odpowiedź do klienta.

Ruby on Rails zapewnia także narzędzie do szybkiej budowy prototypów aplikacji internetowej tzw. „Scaffold”. Dzięki niemu programista może błyskawicznie po rozpoczęciu prac przedstawić klientowi prostą, ale działającą aplikację. Należy także wspomnieć, że kod generowany w ten sposób jest w pełni użyteczny do rozbudowy, bez konieczności zamiany na nowe, bardziej wyspecjalizowane algorytmy.

3.7.3 Użyte wtyczki

Ruby on Rails udostępnia mechanizm wtyczek. Może je tworzyć każdy, są darmowe, proste w instalacji oraz użytkowaniu. Dzięki temu społeczność skupiona wokół Ruby on Rails stworzyła ogromną ilość dodatków, które bez większego wysiłku można zaadaptować do własnej aplikacji.

Nie wszystkie z nich są użyteczne. Często jest tak, że jedna funkcjonalność zapewnia wiele dodatków o różnym stopniu skomplikowania i jakości kodu.

Jako, że autorzy aplikacji bazują już na gotowym rozwiązaniu Substruct, część wtyczek została im z góry narzucona. W dalszej części pracy rozpatrywane są tylko te, które autorzy uważają za ważne z punktu widzenia funkcjonalności aplikacji.

- `Validates_existence`

We wcześniejszych rozdziałach wspomniano o walidatorach modułu ActiveRecord i powiązaniach między modelami. Niestety nie zapewnia on możliwości walidacji czy model powiązany z innym aktualnie dodawanym do bazy modelem, faktycznie istnieje w bazie danych. Jego brak powodowałby błędy w aplikacji, a w dalszej kolejności utratę zaufania użytkowników.

Wtyczka `Validates_existence` dodaje nowy walidator, który sprawdza czy powiązany model faktycznie znajduje się w bazie. Używa się jej następująco:

```
class DataItem < ActiveRecord::Base
  belongs_to :feature
  belongs_to :conversation

  validates_existence_of :feature, :conversation
end
```

- `Attachement_fu`

Cieężko spotkać aktualnie aplikację internetową, która nie pozwala umieszczać użytkownikom lub administratorom w wygodny sposób obrazków na serwerze. Z programistycznego punktu widzenia nie jest to łatwy proces, gdyż pliki są skomplikowanymi strukturami.

Zadania znacznie ułatwia wtyczka `Attachement_fu`, która udostępnia mechanizm bazodanowy do przechowywania informacji o wrzuconym pliku graficznym zapisanym na dysku. Dodatkowo zapewnia gotowe metody do umieszczenia w formularzu usprawniając ten proces po stronie użytkownika.

- `Will_paginate`

W aplikacji autorów wielokrotnie użytkownik lub administrator ma do czynienia z bardzo dużą ilością rekordów pobieranych z bazy danych. W przypadku telefonów komórkowych

jest to liczba przekraczająca wartość 700. Trudno sobie wyobrazić, aby taka ilość rekordów mogłaby być wyświetlana na raz w oknie przeglądarki.

W tym wypadku, najrozsądniejszym rozwiązaniem wydaje się podzielenie wyświetlania rekordów na podstrony, maksymalnie po 30 na każdej z nich.

Z pomocą przychodzi wtyczka `Will_paginate`. Odpowiada za paginację informacji wyświetlanych w formie tabularyzowanej. Użytkownik może zdefiniować, ile rekordów ma być wyświetlane na jednej podstronie. Wtyczka zapewnia także generowanie linków do poszczególnych podstron z oznaczeniem aktualnie wyświetlanej.

- `Annotate_models`

W Ruby on Rails modele operują na odpowiadających im tabelach w bazie danych. Niestety nie ma możliwości podglądu, jaka jest ich definicja z poziomu pliku modelu.

Plugin `Annotate_models` generuje opis tabeli, której dotyczy dany model na początku jego pliku, ułatwiając pracę programiście.

3.7.4 Substruct

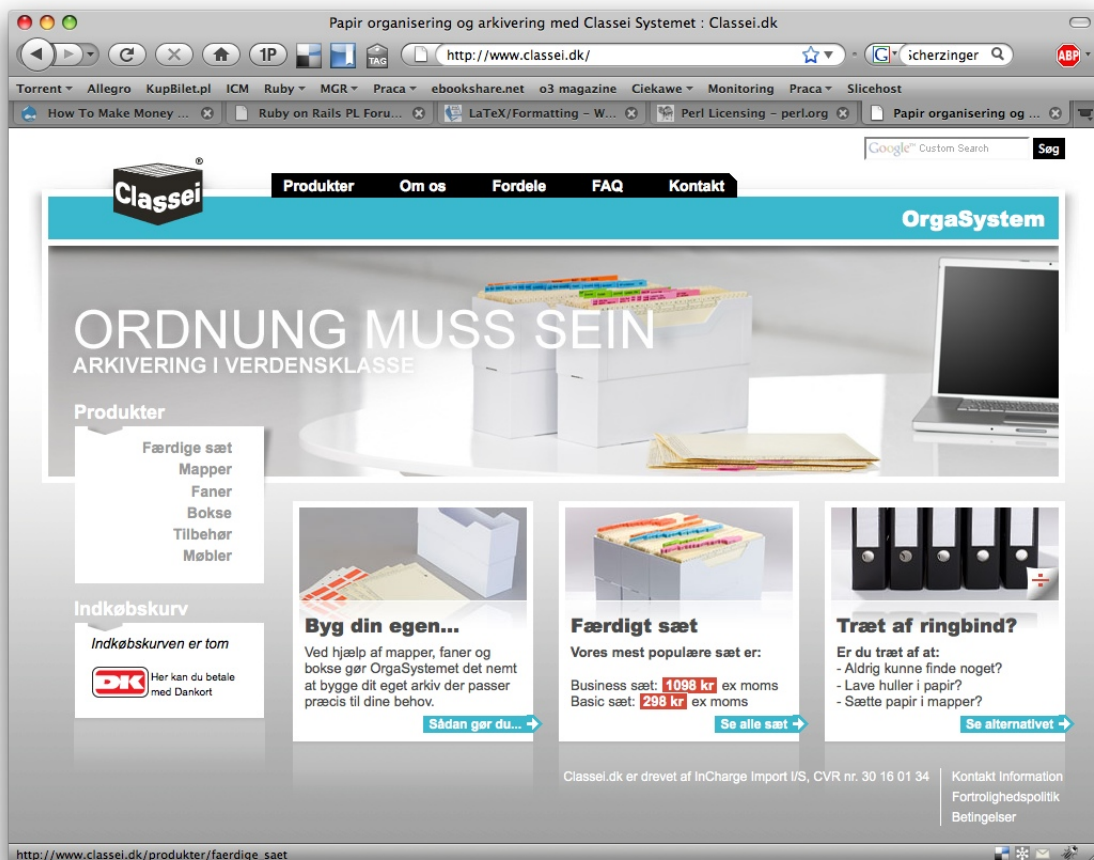
Tematem przewodnim niniejszej pracy magisterskiej jest cybersprzedawca. Autorzy uznali za wygodne, iż stworzona aplikacja powinna bazować na sklepie internetowym, aby bardziej realistycznie oddać jej funkcjonowanie. Samodzielna budowa sklepu internetowego mogłaby być tematem osobnej pracy magisterskiej. Z tego względu uznano, że należy skorzystać z gotowego rozwiązania Open Source.

Niestety Ruby on Rails nie oferuje wielu gotowych aplikacji tego typu. Większość z nich jest w fazie tworzenia lub planowania. Osadzanie cybersprzedawcy na aplikacji działającej w innej technologii byłoby bardzo trudne. Ostatecznie wybrano Substruct. Według autorów najdojrzała i najbardziej rozbudowana aplikacja e - commerce.

Substruct to aplikacja e - commerce dostępna na zasadzie podwójnej licencji GPL [46] oraz tzw. „Artistic License” [48]. Niestety nie doczekała się ona stabilnej, oficjalnej wersji, jednak jest zupełnie wystarczająca do celów pracy magisterskiej. Użyta przez autorów wersja początkowo nosiła oznaczenie 1.0a3, jednak została w późniejszej fazie zaktualizowana do wersji 1.0a6.

Pomimo braku jeszcze stabilnej wersji produkcyjnej, na systemie Substruct jest już opartych wiele działających komercyjnie sklepów internetowych. Jak podaje jego strona domowa [49] są to następujące sklepy:

- Skunk Studios
- My Favorite Pal
- Lama Designs
- AsiaSprout
- GenSelect
- Signature Cigars
- Luxeville
- Classei 16
- Michelle Chappel
- Hawaii Fun Planner



Rysunek 16: Strona sklepu Classei korzystającego z Substruct

Substruct jest rozbudowanym systemem typu e - commerce, o czym świadczą powyższe pomyslnie wdrożenia. Oferowane przez niego funkcje przedstawiają się następująco:

- Wbudowany system zarządzania treścią CMS z możliwością prowadzenia bloga.
- Możliwość zarządzania całą treścią strony bez potrzeby używania języka HTML.
- Transakcje realizowane za pomocą systemu PayPal [50] oraz Authorize.net [51].
- Zarządzanie produktami oraz zamówieniami klientów.
- Generowanie raportów sprzedaży.
- Rozbudowany panel administratora.
- Gotowy dział Q&A.
- Możliwość rozbudowy aplikacji.

Ostatnia cecha jest najważniejsza spośród wszystkich wymienionych i ona zaważyła na wyborze systemu Substruct jako aplikacji bazowej dla cybersprzedawcy.

Substruct bazuje na Rails Engines [52] - bibliotece, która pozwala oddzielić jego kod od rozszerzeń dodanych przez programistę. Struktura aplikacji i kod nowych funkcjonalności znajdują się w zupełnie innych miejscach. W ten sposób udostępnia wygodny mechanizm rozbudowy własnej funkcjonalności bez konieczności zmiany kodu samego Substruct.

Dzięki Rails Engines można przeładowywać metody w klasach zaimplementowanych w Substruct, dodawać własne, a także podmieniać widoki. Dodatkowo w przypadku wydania nowej wersji Substruct, możliwe jest nadpisanie starego kodu nowym, co jest szybsze niż w przypadku, kiedy programista musiałby sprawdzać, które pliki modyfikował i wprowadzać zmiany do nowej wersji.

3.7.5 Biblioteka lingwistyczna CLP

Język polski jest uważany za jeden z najtrudniejszych do opanowania na świecie. Składa się na to skomplikowana wymowa wyrazów, jak również rozbudowana gramatyka. Różne formy i odmiany poszczególnych wyrazów powodują, że nawet rodowity Polak ma czasami trudności z poprawnym ich używaniem.

„Podstawowym problemem przy komputerowym przetwarzaniu języka polskiego jest jego fleksja. Jeden wyraz jest reprezentowany przez wiele form fleksyjnych. Podstawowymi funkcjami realizowanymi przez bibliotekę CLP są:

- rozpoznawanie wyrazu na podstawie jego dowolnej formy fleksyjnej;
- dostarczanie informacji o wyrazie;
- wygenerowanie form fleksyjnych dla danego wyrazu.” [53]

Biblioteka CLP została stworzona przez Grupę Lingwistyki Komputerowej [54] pod przewodnictwem Prof. dr hab. Wiesława Lubaczewskiego. Zawiera ponad 100 tysięcy rekordów i obejmuje cały podstawowy słownik wyrazów pospolitych.

W swojej aplikacji autorzy korzystają z biblioteki CLP do sprowadzania wyrazów do ich form podstawowych. Przykładowo zdanie :

"Nie pójdę dzisiaj do szkoły."

jest zamieniane na:

"nie/on/ono/ona pójść dzisiaj do szkoła."

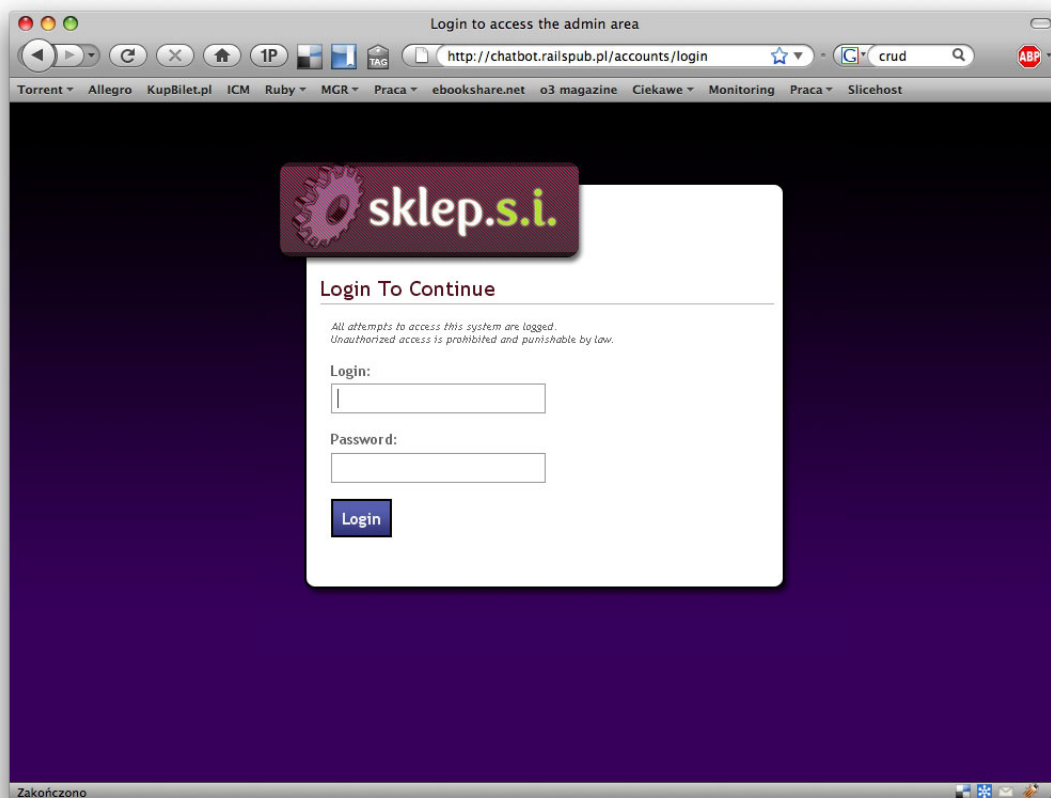
Zdarza się, podobnie jak w przedstawionym przykładzie, że jedno słowo może zostać sprowadzone do kilku form podstawowych. Wtedy aplikacja musi dokonać wyboru, która forma jest według niej najbardziej zbliżona do tego, co chciała uzyskać.

Sprowadzanie wyrazów do form podstawowych ma niewątpliwe zalety, gdyż autorzy nie musieli się martwić odmianą wyrazów w języku polskim, co znacznie ułatwiło implementację modułu rozpoznawania wypowiedzi klienta przez cybersprzedawcę.

Biblioteka CLP jest napisana w języku C. W celu używania jej z poziomu języka Ruby, autorzy wykorzystali bibliotekę RLP [55] napisaną przez Aleksandra Pohla.

3.7.6 Syntezator mowy Ivona

Ivona [56] to syntezytor mowy. Jego zadanie polega na generowaniu plików mp3 przechowujących wypowiedź cybersprzedawcy.



Rysunek 17: Okno logowania do panelu administratora

Ivona posiada męską oraz żeńską wersję głosu. Są one wykorzystywane w zależności od płci prezentowanego chatbota.

Wypowiedź chatbota jest przesyłana do syntezy mowy, który generuje na jej podstawie plik mp3. Jest on następnie odtwarzany za pomocą skryptu JavaScript zaraz po ukazaniu się wypowiedzi chatbota w oknie rozmowy.

4 Prezentacja działania systemu

4.1 Panel administratora

Panel administratora jest wykorzystywany przez użytkownika w celu konfiguracji i zarządzania aplikacją. Jest dostępny dla użytkowników posiadających prawa administratora, którzy mogą się do niego dostać po zalogowaniu 17. Ponieważ Substruct posiada swój panel administratora, autorzy uznali, że warto rozszerzyć jego funkcjonalność także o konfigurację modułu cybersprzedawcy. Dodano osobną zakładkę z listą dostępnych opcji 18.

Panel administratora składa się z zasobów konfigurowanych za pomocą CRUD (ang. Create,

read, update and delete). Oznacza to, że pozwala administratorowi wykonywać na nich akcje tworzenia, przeglądania, aktualizacji oraz usuwania.

Ruby on Rails posiada skrypty generujące szkielet rozwiązania CRUD dla zasobu. To znacznie przyspiesza pracę programisty i pozwala mu się skupić na szczegółach implementacji.

Lista zasobów konfigurowalnych w panelu administratora przedstawia się następująco:

- Synonimy
- Parametry produktów
- Stany chatbota
- Wzorce dopasowania
- Wypowiedzi cybersprzedawcy
- Odmiany wyrazów
- Imiona

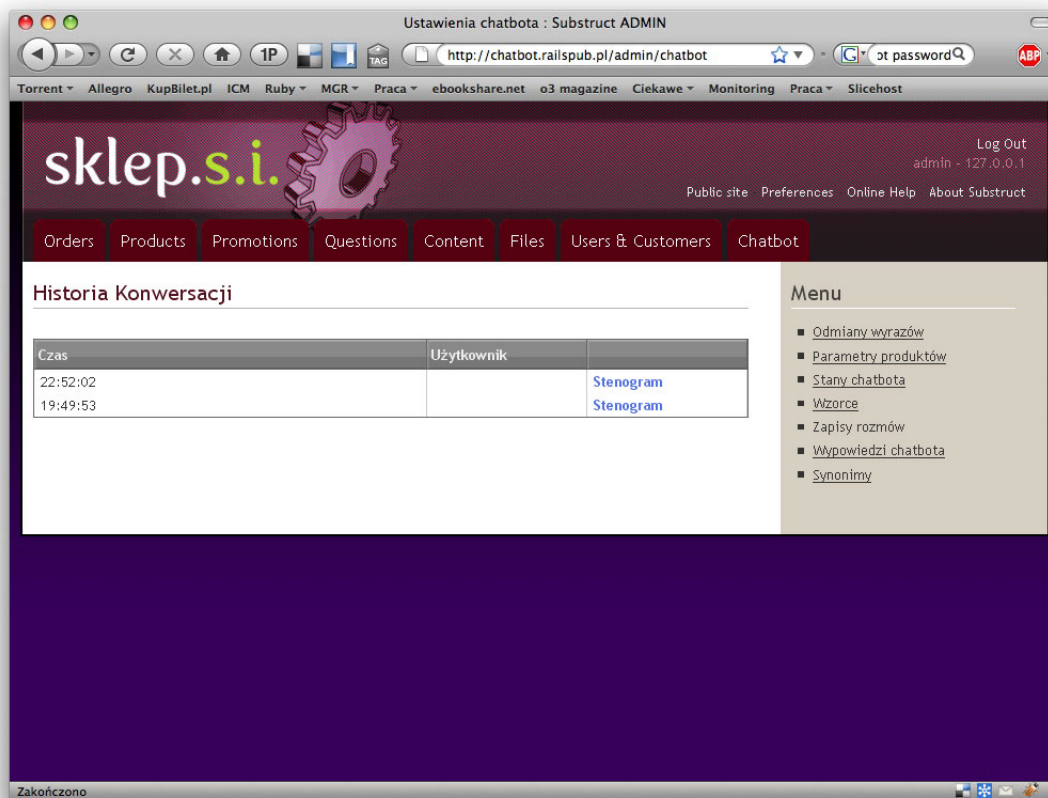
Dodatkowo panel administratora udostępnia zapisy rozmów pomiędzy klientem a cybersprzedawcą.

4.2 Interfejs użytkownika

Cybersprzedawca jest dostępny na stronie w dziale „Zapytaj Sprzedawcę”. Rozmowa z Cybersprzedawcą odbywa się za pomocą formularza tekstowego, by powiedzieć coś chatbotowi należy wpisać tekst w pole tekstowe i nacisnąć przycisk „Wyślij”. Wysłana wypowiedź pojawi się po prawej stronie od wizerunku cybersprzedawcy. Odpowiedź chatbota pojawia się pod wypowiedzią klienta i jest jednocześnie artykułowana dzięki synteźatorowi mowy IVONA. Dla pełnej satysfakcji z obsługi sklepu internetowego klient powinien mieć zainstalowane w swojej przeglądarce rozszerzenie Adobe Flash Player i mieć włączone głośniki.

Cybersprzedawca jest dodatkową funkcjonalnością sklepu internetowego Substruct. Oprócz cybersprzedawcy klient ma do dyspozycji podstawowe funkcjonalności sklepu internetowego takie jak:

- Przeglądanie zbiorcze produktów



Rysunek 18: Strona główna panelu administratora cybersprzedawcy

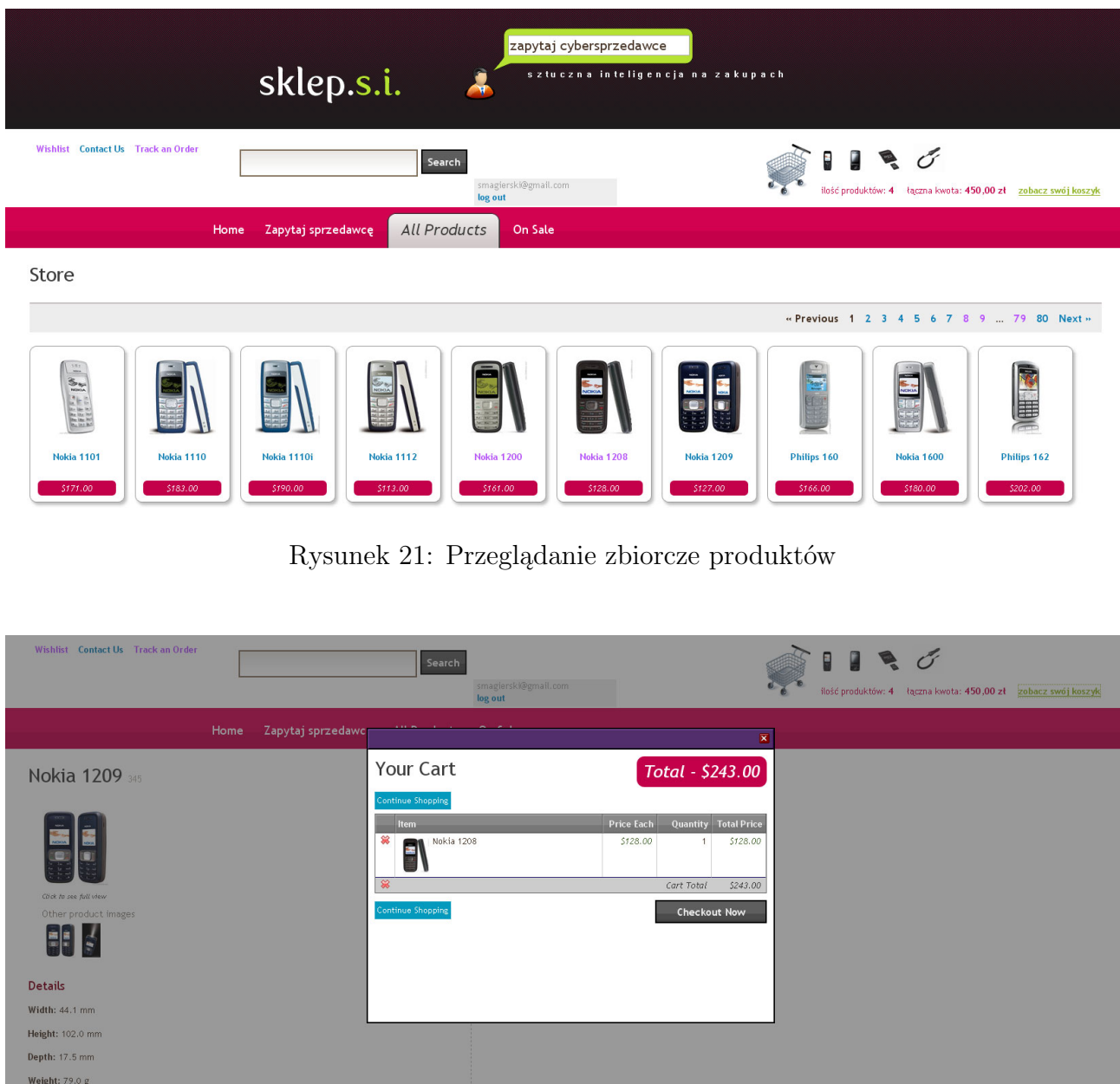


Rysunek 19: Rozmowa z cybersprzedawcą



Chatbot: Witaj, jestem cybersprzedawcą.
Chatbot: Jak masz na imię?
Klient: jestem Grzesiek
Chatbot: Miło mi Ciebie poznać Grzesiek. Witaj w sklepie z telefonami komórkowymi. Chciałbym pomóc w doborze telefonu, który najbardziej odpowiada Pańskim preferencjom i stylowi życia. Proszę mi powiedzieć jaki chce Pan telefon komórkowy i do jakich zastosowań?
Klient: chcę dyktafon, gprs i aparat fotograficzny
Chatbot: Notuję: dyktafon, gprs oraz aparat fotograficzny. Czy coś jeszcze?

Rysunek 20: Przykład cybersprzedawcy mężczyzny



Rysunek 21: Przeglądanie zbiorcze produktów

Rysunek 22: Widok koszyka zakupów

Your Wishlist



Your Stuff

- Account Details
- Orders
- Wishlist

Rysunek 23: Lista życzeń klienta

- Strona szczegółowa danego produktu
- Wyszukiwarka
- Koszyk zakupów
- Lista życzeń
- Śledzenie przesyłki

4.3 Schemat rozmowy

Rozmowa Cybersprzedawcy z klientem ma jeden zasadniczy cel: poznanie preferencji klienta i zaoferowanie odpowiednich produktów. Produktami w przypadku tej pracy magisterskiej są telefony komórkowe. Bardzo ważna z powodów handlowych jest ekspozycja proponowanych produktów, ma to znaczny wpływ na decyzję kupna. Każdy człowiek odbiera świat na swój sposób. Klienci mają różne preferencje dotyczące prezentacji oferty, dlatego w dobraniu odpowiedniej formy prezentacji potrzebne jest rozpoznanie charakteru klienta, czyli jego indywidualnego programu poznawczego [27][28][29].

Cele rozmowy z chatbotem:

Poznanie preferencji klienta. Preferencje klienta zostaną przetłumaczone na język zrozumiały dla systemu, czyli na zbiór parametrów produktu wraz z ich wartościami. Przykładowy klient może chcieć telefon komórkowy o następujących parametrach:

```
kolor = "żółty"
```

liczba wyświetlaczy = 2
przeglądarka internetowa = TAK
odbiornik GPS = TAK
gwarancja > 10 miesięcy

Poznanie charakteru klienta. Na podstawie używanego przez klienta słownictwa i zwrotów językowych identyfikowany jest jego charakter, czyli jego indywidualny program poznawczy. Każdego klienta można zakwalifikować do jednego z dominujących typów: Rządzący, Dostosowujący, Maksymalista, Asekuracyjny, Harmonijny, Weryfikujący, Systematyczny, Inspirujący, Odkrywca, Współczujący, Zadaniowy i Równoważący [27][28][29]. Proces rozpoznawania charakteru klienta jest opisany w podrozdziale 4.3.4, a opis typologii charakterów jest opisany w podrozdziale 2.3.3.

4.3.1 Algorytm rozmowy

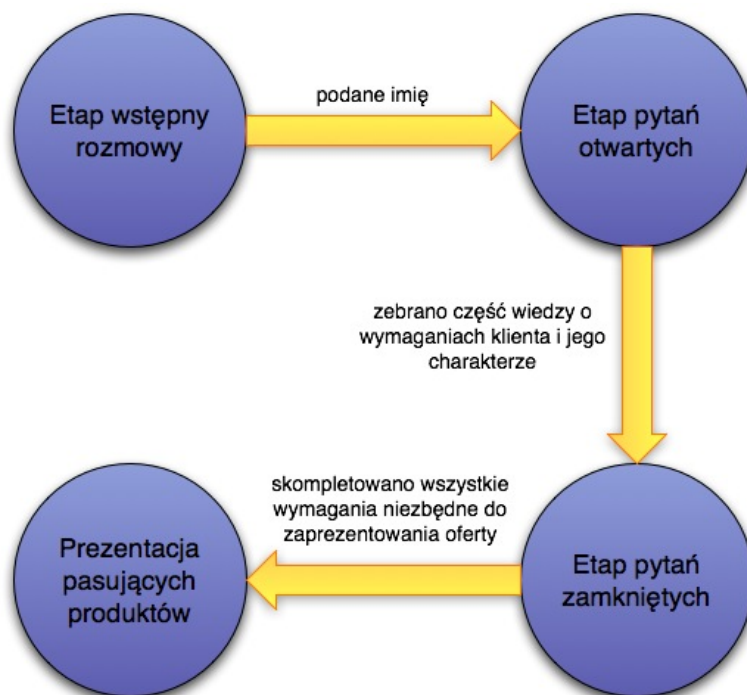
Algorytm rozmowy z cybersprzedawcą składa się z dwóch części, jest to tzw. tor główny rozmowy i tor poboczny. Algorytm toru głównego rozmowy, jest to rozmowa wokół meritum sprawy, czyli rozmowa o preferencjach klienta i jest ona kontynuowana od przedstawienia się aż po prezentację poszukiwanych produktów. Tor poboczny, są to wszystkie inne kwestie, które mogą być poruszane podczas rozmowy, w szczególności: pytania o sklep, zaczepki i wulgaryzmy, odmowy na postawione pytania i poruszanie przez klienta innych tematów. Skutkują one wytłumaczeniem cybersprzedawcy, że rozmowa na inne tematy nie leży w jego kompetencjach.

4.3.2 Tor główny rozmowy

Torem głównym rozmowy jest rozmowa handlowa, podczas której cybersprzedawca próbuje dowiedzieć się, jakie są preferencje klienta odnośnie poszukiwanego telefonu komórkowego. Wzorem sprawdzonych technik handlowych, rozmowa składa się z 4 etapów.

1. etap wstępny
2. etap pytań otwartych
3. etap pytań szczegółowych
4. prezentowanie oferty

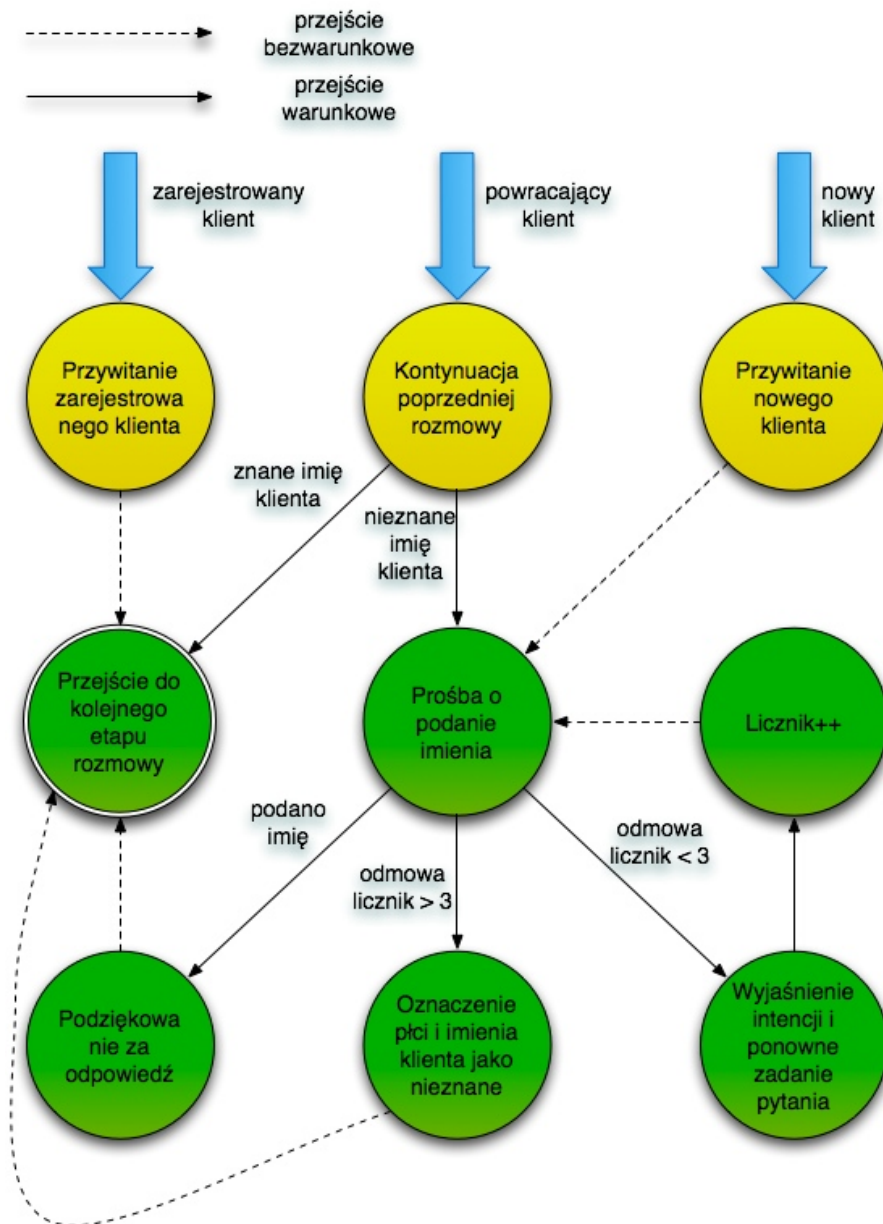
O przejściu między etapami decyduje cybersprzedawca, a kryterium jest ilość i jakość zebranych preferencji klienta.



Rysunek 24: Schemat przejść między etapami toru głównego rozmowy

Etap wstępny Przed rozmową handlową dobrze jest poznać imię klienta, to spowoduje zmniejszenie dystansu. Zwracanie się do klienta w formie odpowiadającej jego płci zwiększy dodatkowo wrażenie inteligencji cybersprzedawcy. Etap wstępny zaczyna się wraz z przybyciem klienta do sklepu i rozpoczęciem rozmowy z chatbotem. Po poznaniu imienia i rozpoznaniu płci, zostaje przeprowadzona zamiana cybersprzedawcy pod pretekstem, tak aby kobieta rozmawiała z mężczyzną, a mężczyzna z kobietą. Poznanie imienia zwiększa pozytywne wrażenia z rozmowy, ale nie jest niezbędne do poznania preferencji zakupowych klienta, także po 3 - krotnej odmowie cybersprzedawca nie będzie przerywał rozmowy, a przejdzie do kolejnego etapu.

Etap pytań otwartych Dobry sprzedawca wie, że najistotniejsze jest dać się klientowi „wygadać”. Z jednej strony jest to sposób na zyskanie sobie zaufania, a z drugiej jest to sposób na poznanie prawdziwych potrzeb klienta. Nie każdy klient zna doskonale ofertę sklepu. Obowiązkiem sprzedawcy jest pomaganie klientowi w odkryciu swoich preferencji. Dlatego na tym etapie chatbot zadaje jedynie bardzo ogólne (np. „W czym mogę pomóc?”) pytania, a następnie dopytuje o szczegóły, gdy zachodzi taka konieczność. Etap pytań otwartych jest momentem w rozmowie kiedy najbardziej można poznać typ osobowości klienta. Dzieje się tak, ponieważ wypowiedzi klienta nie są obciążone kontekstem, a więc są bardziej naturalne. Rozmowa o niczym nie może jednak trwać za długo. Jeżeli przyrost poznawanych wymagań klienta będzie zbyt wolny, chatbot od czasu do czasu zada pytanie bardziej konkretne (np. „Jaki powinien



Rysunek 25: Schemat etapu wstępnego głównego toru rozmowy

być telefon, którego Pan/Pani poszukuje?")), które będzie stymulować klienta, aby ten bardziej skoncentrował się na rozmowie o swoich preferencjach zakupowych. Realizowane czynności chatbota podczas etapu pytań otwartych:

- zadawanie ogólnych pytań

np. "Czym jest Pan/Pani zainteresowana" "Jak mogę Pani/Panu pomóc?"

- parafrazowanie wypowiedzi klienta
- dopytywanie o szczegóły

Klient: "Chciałbym mieć ładną komórkę"

Chatbot: "Co rozumiesz poprzez ładną komórkę?"

- dopytywanie o szczegóły z propozycją parametru

Klient: "Jestem biznesmenem, potrzebuje komórkę do pracy"

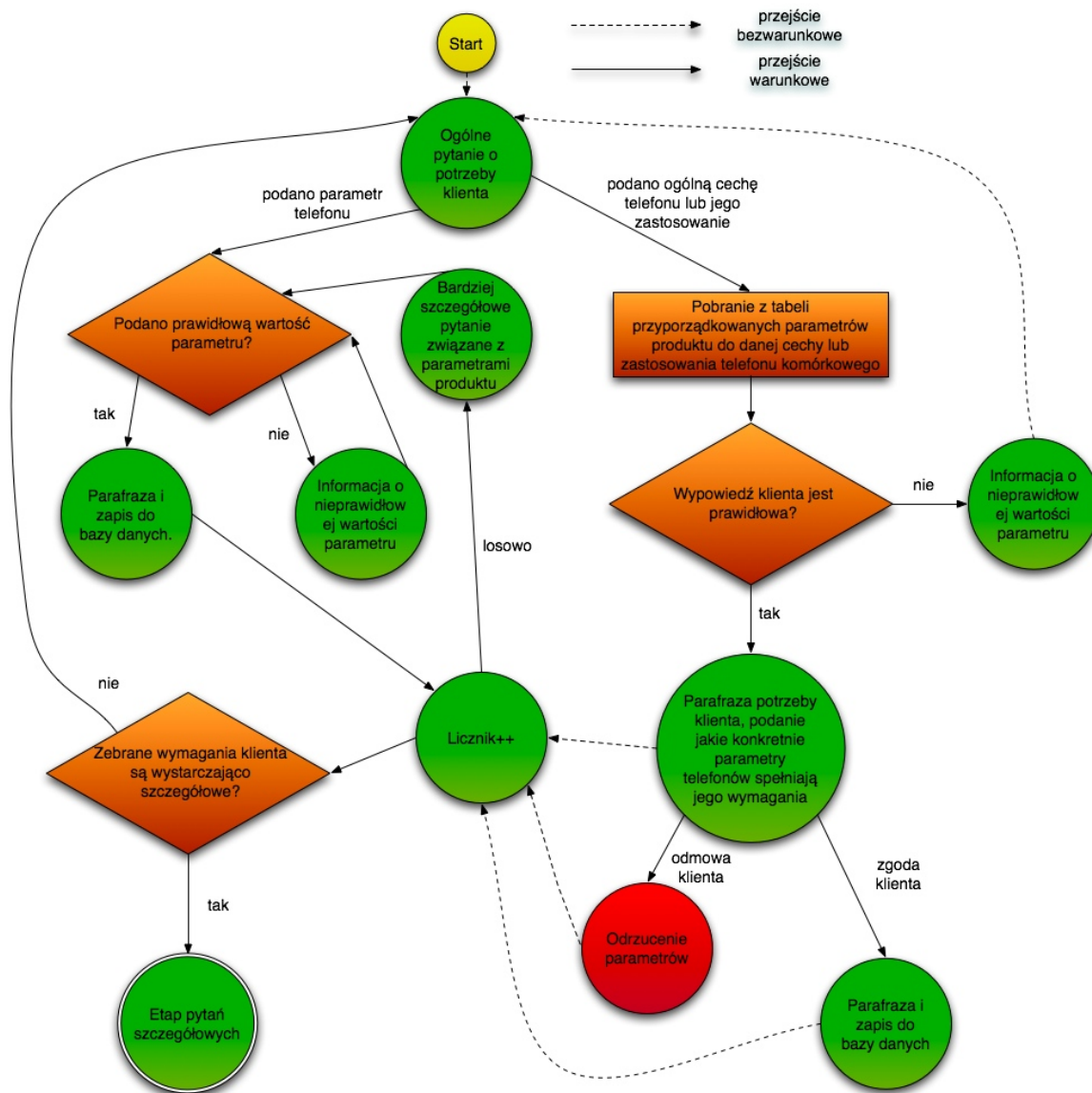
Chatbot: "Rozumiem, czy chciałby Pan/Pani telefon z kalendarzem i organizerem?"

Tablica 9: Przykładowa tabela przyporządkowania ogólnych określeń produktu do szczegółowych parametrów

ogólne określenie	wartości parametrów
praca	product.voice_memo = 1 product.calendar = 1 product.organizer = 1 product.doc_browser = 1 product.browser = 1 product.wlan = 1 product.talk_time > 8
rozrywka	product.games = 1 product.composer = 1 product.camera_number >= 1 product.browser = 1 product.radio = 1 product.games_download = 1

- stymulowanie rozmowy, gdy przez dłuższy czas nie był poruszany temat preferencji zakupowych klienta, poprzez zadawanie bardziej szczegółowych pytań

Etap pytań szczegółowych W tym etapie inicjatywę przejmuje cybersprzedawca. Do etapu pytań zamkniętych przechodzi się, gdy zebrana jest dostateczna ilość wymagań klienta. W tym celu należy już tylko dopytać o parę dodatkowych parametrów w celu zawężenia liczby produktów na tyle, aby można było je zaprezentować. Klient w pierwszej kolejności pytany jest o te parametry, które najbardziej zmniejszają zbiór pasujących modeli telefonów. Etap rozmowy się kończy i następuje przejście do zaprezentowania oferty, gdy zbiór pasujących produktów jest dostatecznie mały. Wielkość zbioru modeli jest też uzależniona od odkrytego charakteru klienta. Np. Typ Dominujący woli mieć większy wybór niż Typ Dostosowujący.

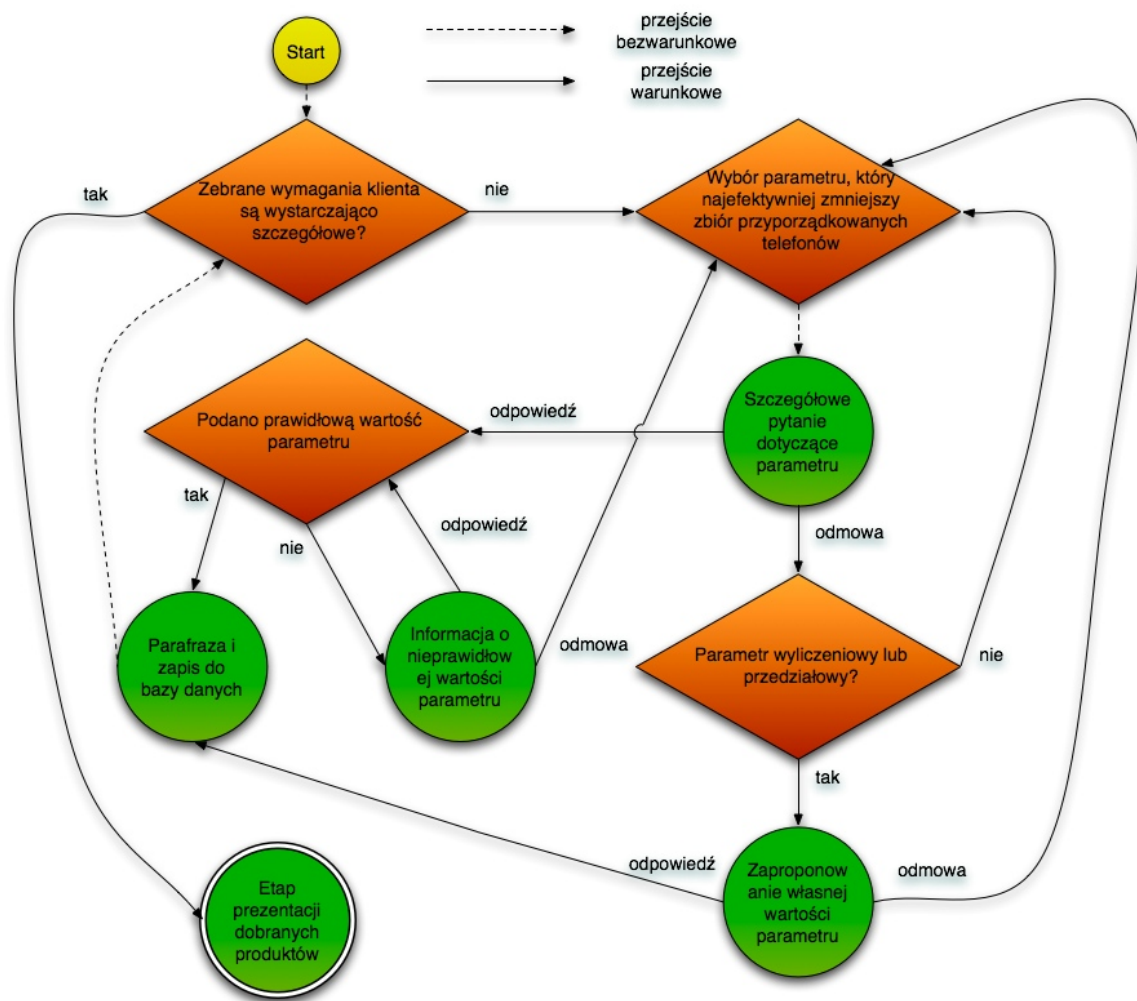


Rysunek 26: Schemat etapu pytań otwartych toru głównego rozmowy

Prezentowanie oferty W pełni przeprowadzona rozmowa kończy się zaprezentowaniem oferty. Przedstawione są modele, które najlepiej pasują poznany w toku rozmowy preferencjom klienta. Sposób zaprezentowania tych modeli jest uzależniony od odkrytego charakteru klienta. Np. Typ Systematyczny będzie miał wyświetlone wszystkie produkty w jednolitych tabelach i z dużą ilością szczegółów, Typ Inspirujący odwrotnie, będzie miał małą ilość szczegółów za to więcej zdjęć i bardziej estetycznie udekorowanych itp.

4.3.3 Tor poboczny rozmowy

W torze poboczny są zawarte wszystkie aspekty rozmowy, które nie dotyczą rozmowy o preferencjach zakupowych klienta. Tor poboczny jest rozpatrywany w dowolnym momencie rozmowy, jednak zawsze powraca się do stanu, w którym została przerwana rozmowa. Np.



Rysunek 27: Schemat etapu pytań zamkniętych toru głównego rozmowy

Chatbot: "Dzień dobry, jak masz na imię?"

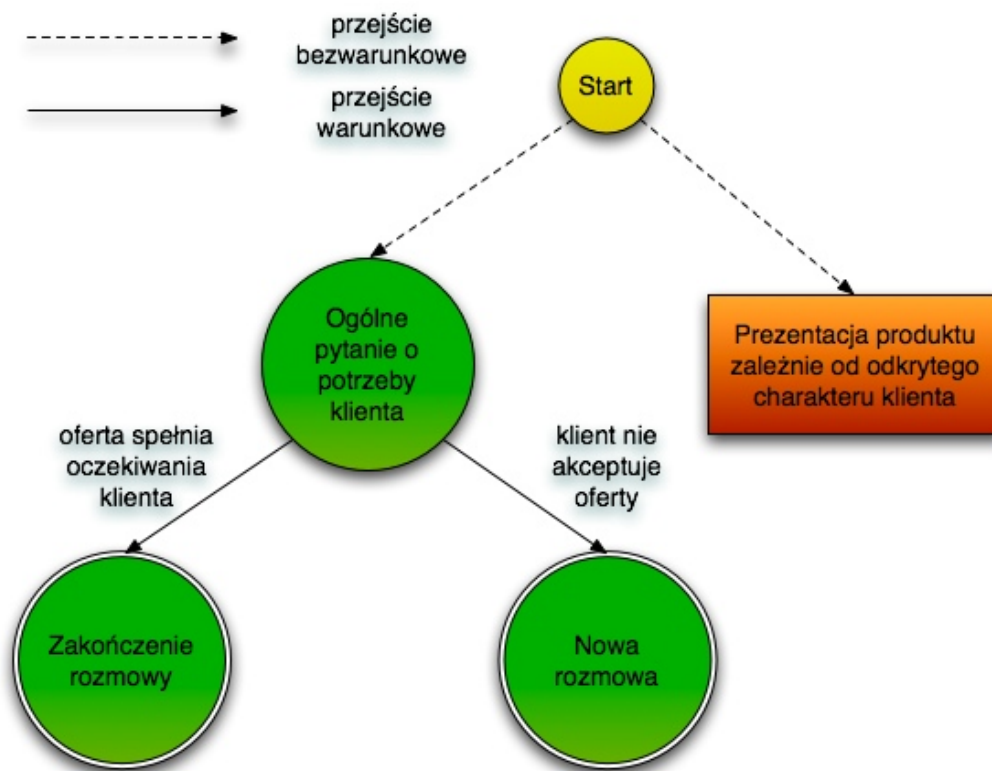
Klient: "W jakiej formie można dokonać u państwa płatności ?"

Chatbot: "Nasz sklep przyjmuje karty płatnicze i płatność przy odbiorze.

Czy mógłby Pan/Pani powiedzieć jak ma na imię?"

W torze pobocznym realizowana jest reakcja chatbota na następujące zdarzenia:

- Odmowa/wymijanie się od odpowiedzi na pytanie - chatbot prosi ponownie o odpowiedź na pytanie podając dodatkowo intencję tego pytania
- Nie znalezienie wzorca pasującego do wypowiedzi klienta (uczenie się) - chatbot prosi o sformułowanie wypowiedzi inaczej - gdy teraz zostaje odkryty pasujący wzorec, to niezrozumiana wcześniej wypowiedź zostaje zapisana w bazie. Trafi ona na stałe do zasobów wiedzy chatbota po zatwierdzeniu administratora. Jeżeli powtórnie, wypowiedź klienta nie zostanie zrozumiana, chatbot potraktuje to jako wypowiedź nie na temat i



Rysunek 28: Etap prezentacji produktów (bez wątków pobocznych)

uprzejmie poprosi o wrócenie do tematu.

- Odpowiadanie na pytania - chatbot dysponuje bazą często zadawanych pytań (FAQ) na temat sklepu, działalności firmy, formy wysyłek itp.
- Reakcja na personalne zaczepki, np.

Klient: "Ale jesteś ładna"

Chatbot: "Żechce to moją kobiecą próżność, ale jestem w pracy.

Proszę wrócić do tematu rozmowy"

- Reakcja na wulgaryzmy - chatbot ostrzega, że nie będzie prowadził rozmowy, gdy klient przeklina. Po trzech wypowiedzianych wulgaryzmach rozmowa jest kończona.

4.3.4 Rozpoznawanie charakteru klienta

Każdy człowiek odbiera świat na swój sposób. Inaczej organizuje swoją wiedzę o świecie i zwraca uwagę na inne aspekty rzeczywistości. Jednym z przyczyn kierujących ludzkim działaniem jest indywidualny charakter, który można też nazwać indywidualnym programem

poznawczym. Programy osobowości w znacznej większości są dziedziczone genetycznie. Charakter w takim rozumieniu generuje cele, jakimi się człowiek w życiu kieruje. Okazuje się, że w roli konsumenta człowiek też kieruje się swoimi wewnętrznymi celami, które są często ukryte. Ta praca magisterska bazuje na typologii charakterów klientów, która została opracowana przez dr Adriana Horzyka [27][28][29] - opis poszczególnych typów zachowań klienta znajduje się w rozdziale 2.3.3.

Na podstawie wykładów dr Adriana Horzyka [27][30] została stworzona lista fraz językowych i zachowań klienta podczas rozmowy, które są wyznacznikami konkretnych programów osobowości. Każdej frazie przyporządkowany jest typ osobowości, na który wskazuje i przyporządkowana jest liczba od 1 do 10, określająca siłę wyznacznika. Każda wypowiedź klienta, niezależnie o etapu rozmowy, jest sprawdzana pod kątem wykrywania zdefiniowanych fraz. Gdy dana fraza zostanie odnaleziona, przypisana jest siła wyznacznika do konkretnego typu w tabeli charakteru klienta. Tabela charakteru klienta jest uzupełniana na przestrzeni wielu rozmów i jest zapamiętana w systemie pomiędzy wejściami klienta do sklepu internetowego nawet, gdy klient nie posiada swojego konta.

W rzeczywistości nie ma ludzi o osobowości idealnie pokrywającej się z którymś z wymienionych typów. Najczęściej u ludzi dominują cechy paru typów programów osobowości. W opracowanym sklepie internetowym, przedmiocie niniejszej pracy magisterskiej, brany jest pod uwagę dominujący charakter klienta. Im więcej rozmów zostanie przeprowadzonych z danym klientem, tym bardziej wiarygodna staje się ocena jego charakteru.

Od typu charakteru klienta zależy sposób eksponowania produktów w ofercie sklepu i sposób prowadzenia rozmowy z klientem.

Tablica 10: Słowa kluczowe charakterystyczne dla klienta typu Dominujący

Typ charakteru	frazy językowe
Dominujący	„chcę” „potrzebuję” „nie” „zależy mi na..” „uważam” „według mnie” „moim zdaniem” „ja”, „moje” (w różnych odmianach)

Odkryty charakter klienta ma bezpośrednie przełożenie na sposób, w jaki zaprezentowana

Tablica 11: Słowa kluczowe charakterystyczne dla klienta typu Dostosowujący

Typ charakteru	frazy językowe
Dostosowujący	„czy..” „mógłby mi Pan doradzić” „co by Pan kupił na moim miejscu” „nie jestem jeszcze zdecydowany” „nie podjąłem jeszcze decyzji” „decyzja” „porada” „poradzić” „pomóż mi” „powiedz mi” „poprowadź mnie” „jeśli chcesz” „chciałbym” „zgadzam się” „również uważam, że to dobra decyzja” „może”

Tablica 12: Słowa kluczowe charakterystyczne dla klienta typu Maksymalista

Typ charakteru	frazy językowe
Maksymalista	„podoba mi się” „marzyłem o” „wyobrażałem sobie” „jakbyśmy” „wszystko” „bez ograniczeń” „bajer” „personalizacja” „dodatki” „niepowtarzalny” „wyrafinowany” „wyjątkowy” „naj-..” „bardzo”

Tablica 13: Słowa kluczowe charakterystyczne dla klienta typu Asekuracyjny

Typ charakteru	frazy językowe
Asekuracyjny	„Obawiam się, że..” „Czy sprawdziłem/skontrolowałem” „A co będzie jak..” „A jeżeli..” „A gdyby..” „gwarancja” „serwis” „pewność” „bezawaryjność” „niezawodność” „problem” „ale przecież” „ale” „czy na pewno ?” „na 100%?”

Tablica 14: Słowa kluczowe charakterystyczne dla klienta typu Harmonijny

Typ charakteru	frazy językowe
Harmonijny	„zastanowię się” „odezwę się jeszcze” „jeszcze” „może tak jest” „może rzeczywiście” „nie przeczę, że może być tak..” „troszeczkę” „malutko” „fakturka” „paragonik” „telefonik” „komóreczka”

Tablica 15: Słowa kluczowe charakterystyczne dla klienta typu Weryfikujący

Typ charakteru	frazy językowe
Weryfikujący	„byłoby lepiej jakbyście..” „jakbyście zrobili?” „gdyby był lepszy..” „ktoś musiał wam to powiedzieć” „nie ma” „brakuje” „nie ładnie” „krzywo?” „ma usterkę” „nie działa” „nie działa prawidłowo” „naprawdę można było” „dało się zrobić”

Tablica 16: Słowa kluczowe charakterystyczne dla klienta typu Systematyczny

Typ charakteru	frazy językowe
Systematyczny	„Po pierwsze” „Po drugie” „Po trzecie” „najpierw” „potem” „na koniec” „zaplanować” „posprzątać” „przemyśleć” „posegregować” „posortować” „porównać” „odnośnie” „podobny”

Tablica 17: Słowa kluczowe charakterystyczne dla klienta typu Inspirujący

Typ charakteru	frazy językowe
Inspirujący	„Podoba mi się” „Marzyłem o..” „Wyobrażałem sobie..” „Co robisz?” „nowość” „nowinka” „nowe technologie”

Tablica 18: Słowa kluczowe charakterystyczne dla klienta typu Współczujący

Typ charakteru	frazy językowe
Współczujący	„chcę kupić komórkę dla syna” „chciałbym kupić komórkę, bo moja jest już przestarzała” „rodzina” „cześć” „żona” „dziecko” „syn” „córka” zdrobniała forma imienia

Tablica 19: Słowa kluczowe charakterystyczne dla klienta typu Zadaniowy

Typ charakteru	frazy językowe
Zadaniowy	„Chcę kupić komórkę” „Chcę kupić telefon komórkowy” „Proszę mi w końcu powiedzieć coś na temat” „Czy może mi pan w końcu powiedzieć coś na temat” „Chce to już mieć z głowy/skończone” „Pan” „Pani” „szybko” „konkret” „sedno sprawy” „meritum” „nie traćmy czasu” „spieszę się” krótkie zdania, bez intencji

Tablica 20: Słowa kluczowe charakterystyczne dla klienta typu Odkrywczy

Typ charakteru	frazy językowe
Odkrywczy	„dlaczego” „czy mógłbyś mi powiedzieć dlaczego..?” „jak” „szkoda mi czasu na..” „nie będę tracił czasu na..” „strata” „czas” dużo pytań

Tablica 21: Słowa kluczowe charakterystyczne dla klienta typu Równoważący

Typ charakteru	frazy językowe
Równoważący	„w porównaniu” „odpowiedni” „równomierny” „istotny” „ważny” „prawda” „A jak jest naprawdę?” „nie mógłbym tego zrobić bo to nie uczciwe”

Cybersprzedawca



Chatbot: Rozumiem że cena ma być mniejsza od 340\$. Przyjęła do wiadomości. Czy mogę jeszcze w czymś pomóc
 Klient: Chciałbym jeszcze dobrą baterię
 Chatbot: Co Pan rozumie przez dobrą baterię?
 Klient: Większa od 650 mAh
 Chatbot: Rozumiem, przyjęłam do wiadomości. Czy chciałby Pan coś jeszcze dodać?
 Klient: Nie dziękuję
 Chatbot: Przedstawim Panu ofertę. Proszę spojrzeć na prawo od tego okienka.

Prezentacja

	Propozycja 1	Propozycja 2	Propozycja 3	Propozycja 4
nazwa	Nokia 2680 slide	Samsung E840	LG KE820	Nokia N78
wygląd				
cena	238\$	307\$	333\$	236\$
bateria	860 mAh	690 mAh	750 mAh	1200 mAh
dzwonki MP3	TAK	TAK	NIE	TAK
wyświetlacz szerokość	128 px	240 px	220 px	240
wyświetlacz wysokość	160 px	320 px	176 px	320
GPS	NIE	NIE	NIE	NIE
EDGE	NIE	TAK	TAK	TAK
Organizer	NIE	NIE	TAK	NIE

Rysunek 29: Prezentacja oferty dla klienta o charakterze: Systematyczny

mu będzie oferta. Forma prezentacji produktów jest zoptymalizowana tak by uwypuklić to, co jest dla danego charakteru klienta najważniejsze. Np. w sytuacji klienta o charakterze Systematycznym, najważniejsze jest logiczne uporządkowanie informacji, które ułatwia porównanie produktów w wyodrębnionych kategoriach. Natomiast dla typu Inspirującego bardziej istotny jest inspirujący i innowacyjny sposób zaprezentowania informacji.

Sposób prezentowania jest łatwo konfigurowalny dzięki systemowi szablonów, jaki jest częścią framework'u Ruby on Rails.

4.4 Przykłady rozmów

W rozdziale zaprezentowano stenogramy przykładowych rozmów cybersprzedawcy z różnymi klientami.

Cybersprzedawca



Chatbot: Rozumiem że cena ma być mniejsza od 340\$. Przyjęła do wiadomości. Czy mogę jeszcze w czymś pomóc
 Klient: Chciałbym jeszcze dobrą baterię
 Chatbot: Co Pan rozumie przez dobrą baterię?
 Klient: Większa od 650 mAh
 Chatbot: Rozumiem, przyjęłam do wiadomości. Czy chciałby Pan coś jeszcze dodać?
 Klient: Nie dziękuję
 Chatbot: Przedstawiam Panu ofertę. Proszę spojrzeć na prawo od tego okienka.

Wyślij

Prezentacja

Nokia 2680 slide



cena: 230\$
 bateria: 860 mAh
 dzwonki MP3
 wyświetlacz szerokość 128 px
 wyświetlacz wysokość 160 px
 GPS
 EDGE
 Organizer

LG KE820



cena: 333\$
 bateria: 750 mAh
 dzwonki MP3
 wyświetlacz szerokość 220 px
 wyświetlacz wysokość 176 px
 GPS
 EDGE
 Organizer

Rysunek 30: Prezentacja oferty dla klienta o charakterze: Inspirujący

Tablica 22: Rozmowa z klientem Współczującym.

Rozmówca	Wypowiedź	Analiza
Chatbot	Witam w internetowym sklepie z telefonami komórkowymi. Jestem Maria, a Pan/Pani?	
Klient	Jestem Grzesiek	Rozpoznanie płci klienta.
Chatbot	Miło mi Pana poznać. Chciałabym Panu pomóc w doborze telefonu, który najbardziej odpowiada Panu, preferencjom i stylowi życia. Proszę mi opowiedzieć o Pańskich wymaganiach odnośnie poszukiwanego produktu.	
Klient	Chciałbym posiadać internet w komórce, duży wyświetlacz i koniecznie aparat fotograficzny, abym mógł robić zdjęcia swoim dzieciom.	Rozpoznanie słowa kluczowego „dzieci” i zakwalifikowanie klienta jako Współczujący w małym stopniu.
Chatbot	Rozumiem, że życzy Pan sobie, aby Pański telefon umożliwiał dostęp do Internetu, robienie rodzinnych zdjęć oraz posiadał duży wyświetlacz. Czy na coś jeszcze powinnam zwrócić uwagę w swojej ofercie dla Pana i Pańskiej rodziny?	Parafraza wymagań z podkreśleniem roli rodziny klienta Współczującego i zachęcenie do podania dalszych preferencji.
Klient	Zależy mi także na radiu oraz polifonicznych dzwonekach.	

Chatbot	W takim razie notuję: dostęp do Internetu, duży wyświetlacz, aparat fotograficzny, dzwonki polifoniczne oraz radio. Czy to już wszystko?	
Klient	Wydaje mi się, że tak.	
Chatbot	W takim razie pozwoli Pan, że zadam kilka pytań, które umożliwią mi lepsze poznanie Pańskich preferencji oraz zawężą dostępną gamę produktów. Czy jest dla Pana istotna marka produktu?	Rozpoczęcie etapu pytań szczegółowych.
Klient	Moja żona bardzo sobie chwali Nokię.	Rozpoznanie słowa kluczowego „żona”, zwiększenie wskaźnika charakteru klienta Współczującego.
Chatbot	Rozumiem, w takim razie telefony Nokii będą miały pierwszeństwo. Mój mąż także lubi tę markę. Czy posiada Pan jakieś preferencje odnośnie ceny poszukiwanego produktu?	Styl rozmowy odpowiadający klientowi współczującemu - odniesienie się do rodziny sprzedawcy.
Klient	Tak, chciałbym aby to był produkt z wyższej półki, lubię się pochwalić znajomym, ale nie droższy niż 1300zł.	Rozpoznanie słowa kluczowego „znajomi” - dalsze zwiększanie wskaźnika dla klienta Współczującego.
Chatbot	Rozumiem, że poszukuje Pan markowego telefonu komórkowego do 1300zł. Czy życzy Pan sobie, aby produkt posiadał odbiornik GPS?	
Klient	Nie, nie potrzebuję takiej funkcjonalności.	
Chatbot	Rozumiem. Czy jest dla Pana istotna funkcja dyktafonu? Może się przydać dzieciom w szkole.	Kolejne odniesienia do rodziny mającej istotną wartość dla klienta współczującego
Klient	Raczej nie.	
Chatbot	Dziękuję za wyczerpujące odpowiedzi na wszystkie moje pytania. Wybrałam dla Pana 3 produkty najlepiej spełniające Pana wymagania. Są wyświetlone po prawej stronie. Znajomi na pewno będą Panu zazdrościć wybranego modelu, a rodzina będzie miała wiele wspaniałych zdjęć. Zachęcam także do przejrzania specyfikacji technicznej i wybrania dla siebie odpowiedniego telefonu komórkowego. Dziękuję za rozmowę i polecam się na przyszłość.	Wybór modeli telefonów komórkowych najlepiej spełniających wymagania klienta.

Tablica 23: Rozmowa z klientem Systematycznym z przykładem zamiany sprzedawców.

Rozmówca	Wypowiedź	Analiza
Chatbot	Witam w sklepie internetowym oferującym telefony komórkowe. Nazywam się Maria i mam przyjemność Panu/Pani pomóc w wyborze odpowiedniego produktu. Z kim mam przyjemność?	
Klient	Nazywam się Magda.	Rozpoznanie płci klienta.
Chatbot	Miło mi Panią poznać. Przepraszam Panią najmocniej, ale zostałam wezwana przez kierownika do załatwienia bardzo pilnej sprawy firmowej. Czy mogę przekazać naszą rozmowę koledze?	Prośba o możliwość zmiany sprzedawcy ze względu na rozpoznanie takich samych płci klienta i sprzedawcy.
Klient	Ok.	Klient zgadza się.
Chatbot	Witam, nazywam się Marcin. W czym mogę pomóc?	Przywitanie się nowego chatbota.
Klient	Chcę kupić telefon.	Zwięzłe wyrażenie zamiarów - cecha klientów Systematycznych.
Chatbot	Dobrze Pani trafiła, posiadamy bardzo rozbudowaną gamę produktów. Proszę opowiedzieć mi o swoich oczekiwaniach odnośnie produktu.	
Klient	Po pierwsze nieduży, czarny, po drugie koniecznie z aparatem fotograficznym i odtwarzaczem MP3.	Wyliczanie jest cechą klientów Systematycznych - słowa kluczowe „po pierwsze”, „po drugie”. Zwiększenie wskaźnika dla tego typu charakteru.
Chatbot	Zapisałem: mały telefon z czarną obudową, aparatem fotograficznym i odtwarzaczem MP3. Czy coś jeszcze?	Podsumowanie dotychczasowych wymagań klienta w uporządkowany sposób.
Klient	To wszystko.	
Chatbot	Jaki przedział cenowy Panią interesuje?	
Klient	Do 500zł.	Zwięzła odpowiedź, zwiększenie wskaźnika dla klienta Systematycznego.
Chatbot	Rozumiem, szukamy telefonu do 500zł. Czy interesują Panią radio w telefonie?	
Klient	Tak.	
Chatbot	W takim razie dopisuję tę funkcję do listy wymagań. Czy chciałaby Pani posiadać telefon z dzwonekami polifonicznymi?	
Klient	Nie ma to dla mnie znaczenia.	

Chatbot	Dziękuję za wyczerpujące odpowiedzi na wszystkie moje pytania. Po prawej stronie zostały wyświetlone specyfikacje techniczne wybranych produktów. Ich parametry najlepiej pasują do Pańskich wymagań. Dziękuję za rozmowę i polecam się na przyszłość.	Zaprezentowanie oferty w uporządkowany sposób z dużą ilością parametrów. Dane wyświetlone w formie tabelarycznej 29.
---------	--	--

Tablica 24: Przykład rozmowy z klientem posługującym się wulgaryzmami.

Rozmówca	Wypowiedź	Analiza
Chatbot	Witam w internetowym sklepie z telefonami komórkowymi. Jestem Maria, a Pan/Pani?	
Klient	Jestem Krzysiek [przekleństwo].	Rozpoznanie pierwszy raz przekleństwa, zapisanie go do bazy danych konwersacji.
Chatbot	Miło mi Pana poznać, choć jestem zmuszona zwrócić uwagę, aby w rozmowie ze mną nie używał Pan wulgarnych słów. W czym mogę Panu pomóc?	
Klient	Chciałbym kupić telefon [przekleństwo].	Przekleństwo rozpoznane po raz drugi.
Chatbot	Ostatni raz zwracam uwagę odnośnie używanych wulgaryzmów. W przeciwnym przypadku będę zmuszona zakończyć rozmowę.	Ostrzeżenie o zakończeniu rozmowy jeśli klient nie przestanie używać wulgarnych słów.
Klient	Ok [przekleństwo].	Rozpoznanie wulgaryzmu po raz trzeci.
Chatbot	Przykro mi, ale jestem zmuszona zakończyć z Panem rozmowę.	Zakończenie rozmowy.

5 Podsumowanie

Aplikacja cybersprzedawcy według założeń autorów miała poprawić jakość obsługi sklepu internetowego. Znaczna większość występujących na rynku rozwiązań e-commerce bazowało na podobnym schemacie: spis produktów podzielonych na logiczne kategorie. W tradycyjnym sklepie internetowym klient jest pozostawiony sam sobie. Poszukując produktu spełniającego

jego wymagania jest zmuszony przeglądać wiele podstron z produktami nie wiedząc co dla siebie wybrać. Często zdarza się, że widząc bardzo duży asortyment, klient rezygnuje z zakupu, dopóki nie nabędzie odpowiedniej wiedzy na temat branży, aby mógł świadomie i pewnie wybrać odpowiedni dla siebie produkt.

Wprowadzenie innowacyjnego rozwiązania, jakim jest niewątpliwie chatbot, pozwala klientowi oszczędzić czas. Użytkownik pojawiając się pierwszy raz w takim sklepie internetowym, od samego początku może liczyć na pomoc cybersprzedawcy, który przypomina człowieka sprzedawcę z rzeczywistego sklepu. Chatbot, zaprogramowany w taki sposób, aby posiadał jak największą bazę wiedzy o oferowanych produktach jest w stanie w efektywny sposób uzyskać wymagania klienta, a następnie dobrać pasujący do nich produkt. Dodatkowo, duży nacisk położony na psychologię zwiększa szansę pozytywnego zakończenia procesu sprzedaży. Zadowolony klient będzie odwiedzał dany sklep częściej polecając go swoim znajomym, co jest najwydajniejszym sposobem reklamy.

Testy aplikacji przeprowadzone przez autorów wykazały, że poziom zadowolenia [14] klientów, podczas korzystania z systemu „wrażliwego” na typ charakteru klienta, wzrasta o ok. 26% [29]. Ten wynik daje nadzieję, że tego typu rozwiązanie posiada realną wartość w przypadku rozwiązań e-commerce. Aplikacja została odebrana bardzo pozytywnie, szczególnie wśród osób w średnim wieku, które są jeszcze przyzwyczajone do tradycyjnych metod sprzedaży w przeciwieństwie do ludzi młodych, biegle poruszających się w świecie Internetu.

Projekt okazał się dużym wyzwaniem dla autorów. Postawili sobie za cel i zrealizowali następujące elementy systemu:

- Rozpoznawanie wypowiedzi klientów

Język polski ze względu na swoją złożoną budowę spowodował, że nad modułem rozpoznawania wypowiedzi klienta autorzy spędzili najwięcej czasu. Mnogość różnych kombinacji zdań, wyrazów, synonimów, a także wieloznaczność wyrazów narzucała konieczność zbudowania precyzyjnego algorytmu rozpoznawania wypowiedzi. Bardzo pomocna okazała się biblioteka CLP, która zmniejszyła ilość potrzebnych wzorców. Nie zawierała ona jednak, wszystkich potrzebnych odmian wyrazów, np. słowa „piksel”. Aby rozwiązać ten problem, autorzy stworzyli własną prostą bazę odmian wyrazów z możliwością jej modyfikacji w panelu administratora.

- Mechanizm rozpoznawania wymagań klienta

W przypadku opisywanej aplikacji, wymagania klienta były skończonym zbiorem określonych parametrów telefonów. Istotne było, aby cybersprzedawcę wyposażyć w rozwiązania, dzięki którym był on w stanie określić, które wymagania klienta miały sens, a które nie. W ten sposób chatbot był w stanie odpowiednio reagować na zdania typu „Chcę komórkę o wysokości mniejszej niż -10cm” zachowując się inteligentnie. Dodatkowo zanim system przystępował do rozpoznawania wymagań klienta, jego wypowiedź była przetwarzana pod kątem zamiany synonimów, usuwania niedozwolonych znaków oraz zamiany jednostek miary na podstawowe. W ten sposób kolejny etap, odpowiadający za dopasowanie wymagań, otrzymywał uporządkowany zbiór wyrazów pozbawionych zbędnych elementów.

- Generowanie wypowiedzi chatbota

Swoją inteligencję chatbot prezentował poprzez swoje wypowiedzi. Zaimplementowano takie mechanizmy jak: parafraza, dopytywanie się o szczegóły, rozpoznawanie płci oraz imienia klienta. Wszystkie wymienione czynniki znacząco podnosiły prestiż cybersprzedawcy w oczach klienta i zwiększały szansę na sprzedaż produktu. Dodatkowo, na wypowiedź chatbota duży wpływ miał rozpoznany charakter klienta. Podczas rozmowy z różnymi klientami cybersprzedawca stosował inny dobór słów, inaczej prezentował zebrane wymagania klienta, a także końcową ofertę. Ta cecha systemu najbardziej przyczyniła się do zwiększenia satysfakcji testowych rozmówców z chatbotem.

- Rozpoznanie charakteru klienta

Kluczową funkcjonalnością aplikacji było dostosowywanie się do typu klienta. Na początku rozmowy, cybersprzedawca nie posiada żadnych danych na temat charakteru klienta. Jego wypowiedzi są jedynym źródłem, z którego cybersprzedawca czerpie o nim wiedzę. Istotnym aspektem było takie zaplanowanie początkowych etapów rozmowy, aby jak najszybciej móc w efektywny sposób dostosować się do charakteru klienta.

- Konfiguracja systemu

Aplikacja posiada wiele opcji, które można dostosować w taki sposób, aby system jak najlepiej spełniał wymagania swojego właściciela. Istnieje możliwość dodawania kolejnych wzorców, słów kluczowych, charakterów klienta, synonimów a także odmian wyrazów.

- Mechanizm synonimów

W języku polskim istnieje wiele wyrazów o bardzo zbliżonym znaczeniu, np. telefon komórkowy - komórka. Stosując metodę dopasowywania wzorców istniała konieczność tworzenia ich osobno dla każdego wyrazu bliskoznacznego. Rozwiązaniem problemu było zastosowanie mechanizmu synonimów, który podczas początkowej analizy wypowiedzi klienta podmieniał synonimy na ich słowa bazowe, zdefiniowane w bazie danych synonimów. W ten sposób autorzy uniknęli konieczności duplikowania wzorców o identycznym znaczeniu, lecz innym doborze słownictwa. Dodatkowo baza danych synonimów mogła być modyfikowana w panelu administratora.

Po skończonej realizacji niniejszej pracy magisterskiej, autorom pozostało wiele pomysłów na udoskonalenie stworzonej aplikacji. Ciekawym dodatkiem byłaby przykładowo możliwość rozpoznawania głosu klienta poza standardową opcją odczytywania tekstu. Takie rozwiązanie znacznie poprawiłoby wygodę interakcji klienta z cybersprzedawcą, a także umożliwiło rozmowę dużo bardziej naturalną. Problemem w tym przypadku są niedokładne algorytmy rozpoznawania mowy ludzkiej, częste stosowanie przez człowieka skrótów myślowych oraz posługiwanie się zaimkami i językiem potocznym.

Literatura

- [1] N. Spivack. Web 3.0 - the best official definition imaginable [*weblog*]. 2007. http://novaspivack.typepad.com/nova_spivacks_weblog/2007/10/web-30----the-a.html.
- [2] Allan Cho Dean Giustini. Web 3.0 and health librarians: an introduction *JCHLA/JABSC Vol. 29*. 2008. <http://pubs.nrc-cnrc.gc.ca/jchla/jchla29/c07-035.pdf>.
- [3] Markoff John. Entrepreneurs See a Web Guided by Common Sense *The New York Times*. 2007. http://dt123.com/DataGrid/DataGridWebsiteV1a/PDFs/NYT_111306_Web30.pdf.
- [4] Anderson Paul. What is Web 2.0? Ideas, technologies and implications for education *JISC Technology and Standards Watch*. 2007. <http://www.jisc.org.uk/media/documents/techwatch/tsw0701b.pdf>.
- [5] Web 2.0 http://en.wikipedia.org/wiki/Web_2.0.
- [6] Folksonomia <http://pl.wikipedia.org/wiki/Folksonomia>.
- [7] Łukasz Felsztukier. Nadchodzi Web 3.0 *Wirtualnemia.pl*. 2006. http://www.wirtualnemia.pl/document,,2003198,Nadchodzi_Web_3.0.html.
- [8] Web 3.0 - Internet bez ludzi *Money.pl*. 2007. <http://www.money.pl/gospodarka/ngospodarka/ebiznes/artykul/web;3;0;-;internet;bez;ludzi,56,0,251704.html>.
- [9] Agnieszka Dejnaka. *CRM - Zarządzanie kontaktami z klientami*. Helion 2002.
- [10] Adam Sagan. Analiza zachowań konsumenta 2005. Wykłady.
- [11] Rudawska Edyta. *Lojalność klientów*. Polskie Wydawnictwo Ekonomiczne 2005.
- [12] Dyche Jill. *CRM - relacje z klientami*. Helion 2002.
- [13] Newell Frederick. *Lojalność.com - Zarządzanie Relacjami z Klientami w nowej erze marketingu internetowego*. IFC Press 2002.
- [14] ISO 9241 part 11
- [15] Jan Stachowicz , Agata Stachowicz-Stanusch . Zarządzanie poprzez wartości a proces wdrażania CRM http://www.zti.com.pl/instytut/pp/referaty/ref25_full.html.

- [16] Co to jest CRM *biz.reporter.pl*. 2001. <http://biz.reporter.pl/txt/index.php?txt=2001/073&typ=&tyt=Co+to+jest+CRM>.
- [17] Wszystko o CRM i systemach Zarządzania Relacjami z Klientami <http://www.crm-abc.pl/>.
- [18] Ware Lorraine Cosgrove. CRM: wiele wdrożeń na nic *Miesięcznik CEO*. 2004. <http://www.idg.pl/artykuly/44264.html>.
- [19] Rodzaje systemów CRM *Portal CRM review*. 2003. <http://www.crmreview.pl/news.php?news=54>.
- [20] Benjamin Bloom http://en.wikipedia.org/wiki/Benjamin_Bloom.
- [21] Benjamin Bloom's questions http://www.learnerslink.com/bloom%27s_questions.htm.
- [22] Negocjacje <http://pl.wikipedia.org/wiki/Negocjacje>.
- [23] R. Fisher, W. Ury, B. Patton. *Dochodząc do TAK*. Polskie Wydawnictwo Ekonomiczne 2004.
- [24] W. Ury. *Odchodząc od NIE*. Polskie Wydawnictwo Ekonomiczne 2004.
- [25] Levicki Roy J., al. David M. Sanders. *Zasady negocjacji*. Polskie Wydawnictwo Ekonomiczne 2005.
- [26] Sigmund Ernst Fehr et al.. Warto grać uczciwie *Swiat Nauki*. 2002.
- [27] Adrian Horzyk. *Aspekty inteligencji i systemy CRM*. wykłady 2007/2008.
- [28] A. Horzyk, R. Tadeusiewicz. Cechy osobowości użytkownika w systemach sztucznej inteligencji. Ich automatyczne rozpoznawanie, rozumienie i reagowanie na wynikające z nich potrzeby. *Proc. of IWSE*. 2009.
- [29] A. Horzyk, S. Magierski, G. Miklaszewski. An Intelligent Internet Shop-Assistant Recognizing a Customer Personality for Improving Man-Machine Interactions. *Proc. of IIS*. 2009.
- [30] Adrian Horzyk. *Sekrety negocjacji z satysfakcją*. wykłady 2006/2007.
- [31] <http://zabaware.com/>.

- [32] <http://www.jabberwacky.com/>.
- [33] Chatbot <http://pl.wikipedia.org/wiki/Chatbot>.
- [34] Test Turinga http://pl.wikipedia.org/wiki/Test_Turinga.
- [35] <http://www.loebner.net/Prizef/TuringArticle.html>.
- [36] Wallace Richard. From Eliza to A.L.I.C.E. <http://www.alicebot.org/articles/wallace/eliza.html>.
- [37] <http://www.loebner.net/Prizef/loebner-prize.html>.
- [38] AIML: Artificial Intelligence Markup Language <http://www.alicebot.org/aiml.html>.
- [39] Davidson James Duncan. . <http://www.rubyonrails.pl/>.
- [40] Bruce Perens. . <http://www.rubyonrails.pl/>.
- [41] <http://www.rubyonrails.pl/>.
- [42] <http://www.level3.com/>.
- [43] http://pl.wikipedia.org/wiki/Ruby_on_Rails.
- [44] <http://www.gsmarena.com>.
- [45] <http://bazy.hoga.pl>.
- [46] http://pl.wikipedia.org/wiki/GNU_General_Public_License.
- [47] Raymond Scott. *Ajax on Rails*. O'Reilly 2007.
- [48] <http://dev.perl.org/licenses/artistic.html>.
- [49] <http://code.google.com/p/substruct/>.
- [50] <http://www.paypal.pl/pl>.
- [51] <http://www.authorize.net/>.
- [52] <http://rails-engines.org/>.
- [53] Marek Gajęcki. *Biblioteka CLP - opis użytkowy*.
- [54] Komputerowej Grupa Lingwistyki. . <http://winnie.ics.agh.edu.pl/>.

[55] Aleksander Pohl. . <http://apoh1lo.pl/projekty/>.

[56] IVO Software Sp. z o.o. http://www.ivo.pl/firm/ivona_professional_-_syntezator_mowy.html.

Spis rysunków

1	Rozwój technologii internetowych w czasie [2]	11
2	Elementy filozofii CRM, źródło [15]	15
3	Architektura zintegrowanego systemu klasy CRM, źródło [19]	18
4	Piramida potrzeb człowieka - Maslov	19
5	Wykres stylów negocjacyjnych. Źródło opracowanie własne	22
6	Wykres popularności języka Ruby według rankingu TIOBE	38
7	Struktura systemu	40
8	Działanie aplikacji opartej na wzorcu MVC	43
9	Uproszczony schemat konwersacji	44
10	Uproszczony mechanizm działania chatbota	44
11	Algorytm przechodzenia między stanami	47
12	Przykład cechy produktu w bazie danych	51
13	Uproszczony schemat algorytmu dopasowania	58
14	Przykład połączenia tabeli z modelem	62
15	Niepomyślna walidacja modelu	64
16	Strona sklepu Classei korzystającego z Substruct	69
17	Okno logowania do panelu administratora	72
18	Strona główna panelu administratora cybersprzedawcy	74
19	Rozmowa z cybersprzedawcą	74
20	Przykład cybersprzedawcy mężczyzny	75
21	Przeglądanie zbiorcze produktów	75
22	Widok koszyka zakupów	75
23	Lista życzeń klienta	76
24	Schemat przejść między etapami toru głównego rozmowy	78
25	Schemat etapu wstępnego głównego toru rozmowy	79
26	Schemat etapu pytań otwartych toru głównego rozmowy	81

27	Schemat etapu pytań zamkniętych toru głównego rozmowy	82
28	Etap prezentacji produktów (bez wątków pobocznych)	83
29	Prezentacja oferty dla klienta o charakterze: Systematyczny	89
30	Prezentacja oferty dla klienta o charakterze: Inspirujący	90

Spis tablic

1	Schemat tabeli State	45
2	Schemat tabeli Conversation_characters	49
3	Przykład wypełnionej tabeli rozpoznawanie charakteru klienta	50
4	Schemat tabeli „Features” przechowującej dostępne cechy produktów	51
5	Schemat tabeli „Requirements”	52
6	Przykład wymagania klienta zapisanego w bazie danych	53
7	Schemat tabeli „Patterns” przechowujące wzorce dopasowań	53
8	Tabela wypowiedzi klienta	59
9	Przykładowa tabela przyporządkowania ogólnych określeń produktu do szczególnych parametrów	80
10	Słowa kluczowe charakterystyczne dla klienta typu Dominujący	84
11	Słowa kluczowe charakterystyczne dla klienta typu Dostosowujący	85
12	Słowa kluczowe charakterystyczne dla klienta typu Maksymalista	85
13	Słowa kluczowe charakterystyczne dla klienta typu Asekuracyjny	86
14	Słowa kluczowe charakterystyczne dla klienta typu Harmonijny	86
15	Słowa kluczowe charakterystyczne dla klienta typu Weryfikujący	87
16	Słowa kluczowe charakterystyczne dla klienta typu Systematyczny	87
17	Słowa kluczowe charakterystyczne dla klienta typu Inspirujący	87
18	Słowa kluczowe charakterystyczne dla klienta typu Współczujący	88
19	Słowa kluczowe charakterystyczne dla klienta typu Zadaniowy	88
20	Słowa kluczowe charakterystyczne dla klienta typu Odkrywczy	88
21	Słowa kluczowe charakterystyczne dla klienta typu Równoważący	89
22	Rozmowa z klientem Współczującym.	90
23	Rozmowa z klientem Systematycznym z przykładem zamiany sprzedawców.	92
24	Przykład rozmowy z klientem posługującym się wulgaryzmami.	93