

XVI Konferencja PLOUG  
Kościelisko  
Październik 2010

# Eksploracja danych w trybie on-line na przykładzie Oracle Real Time Decisions

Maciej Zakrzewicz  
Politechnika Poznańska, PLOUG

*mzakrz@cs.put.poznan.pl*

**Abstrakt.** Oracle Real Time Decisions (RTD), produkt wcześniej rozwijany przez firmę Sigma Dynamics (znany też jako Siebel Real-Time Decisions), jest interesującym przykładem platformy eksploracji danych w trybie on-line (On-line Data Mining). RTD umożliwia programistom implementację tzw. usług rekomendacyjnych, które na podstawie historii interakcji z użytkownikiem/klientem dokonują predykcji reakcji tego samego lub podobnego użytkownika na nowe zdarzenia. Popularne zastosowania usług decyzyjnych RTD obejmują: rekomendowanie produktów klientom sklepów internetowych w oparciu o doświadczenia klientów o podobnym profilu, wspomaganie centrów zgłoszeniowych (Call Center) poprzez rekomendowanie produktów i usług, implementacja automatycznych mechanizmów Cross-Sell i Up-Sell w aplikacjach sprzedaży na odległość, itp.

**Informacja o autorze.** Pracownik Instytutu Informatyki Politechniki Poznańskiej, prezes Zarządu Stowarzyszenia Polskiej Grupy Użytkowników Systemu Oracle. Zainteresowania naukowe obejmują eksplorację danych (data mining), systemy baz danych/hurtowni danych oraz architektury aplikacji internetowych. Dla krajowych i zagranicznych uniwersytetów oraz przedsiębiorstw (m.in. Niemcy, Wielka Brytania, USA, Słowacja) prowadzi wykłady i szkolenia z zakresu projektowania i implementacji systemów informatycznych. Projektuje i realizuje szkolenia IT, opracowuje strategie informatyzacji, kieruje i doradza w projektach IT. Autor ponad 100 publikacji naukowych i naukowo-technicznych, m.in. z zakresu odkrywania zbiorów częstych, przetwarzania zapytań eksploracyjnych, adaptatywnych serwerów WWW, technik indeksowania baz danych, strojenia systemów baz danych, standardów Java dla baz danych i dla aplikacji wielowarstwowych.



## 1. Wprowadzenie

Klasyczne rozwiązania eksploracji danych (Data Mining) opierają się na iteracyjnych procesach, w ramach których czynności gromadzenia danych źródłowych, ich analizy, generowania modeli decyzyjnych, a następnie implementacji tych modeli są realizowane manualnie, sekwencyjnie i cyklicznie. Inercja takiego rozwiązania, wyrażana czasowym opóźnieniem pomiędzy momentem pierwszego wystąpienia pewnego zjawiska w danych źródłowych a chwilą ukończenia implementacji mechanizmów przyszłej predykcji analogicznego zjawiska, zwykle liczona jest w dniach lub tygodniach. W rezultacie, mechanizmy eksploracji danych pozbawiane są możliwości natychmiastowego (lub szybkiego) reagowania na zmiany w funkcjonowaniu biznesu.

W literaturze przedmiotu dość często przedstawiany jest następujący przykład systemu informatycznego, w którym sprawna reakcja na zmianę charakterystyki danych źródłowych jest kluczowa - dział Call Center zajmujący się obsługą telefonicznych zgłoszeń klientów, a jednocześnie zachęcający do zakupu dodatkowych produktów lub usług. Gdy następuje kontakt klienta z Call Center, w pierwszej kolejności klient jest identyfikowany, następnie przedstawia przyczynę i opis swojego zgłoszenia, zgłoszenie jest obsługiwane, by wreszcie w ostatniej fazie rozmowy konsultant przedstawił klientowi dodatkową ofertę związaną bądź z poszerzeniem zakresu zamówienia (Up-Sell) lub z wzbogaceniem zamówienia o dalsze produkty/usługi (Cross-Sell). W tego typu aplikacjach krytyczny jest właściwy dobór oferty Up-Sell lub Cross-Sell zarówno do profilu telefonującego klienta, jak i do kontekstu produktów/usług już przez klienta posiadanych czy właśnie zamawianych. Sukces biznesowy jest w znacznej mierze uzależniony od „trafialności” nowej oferty w specyfikę danego klienta.

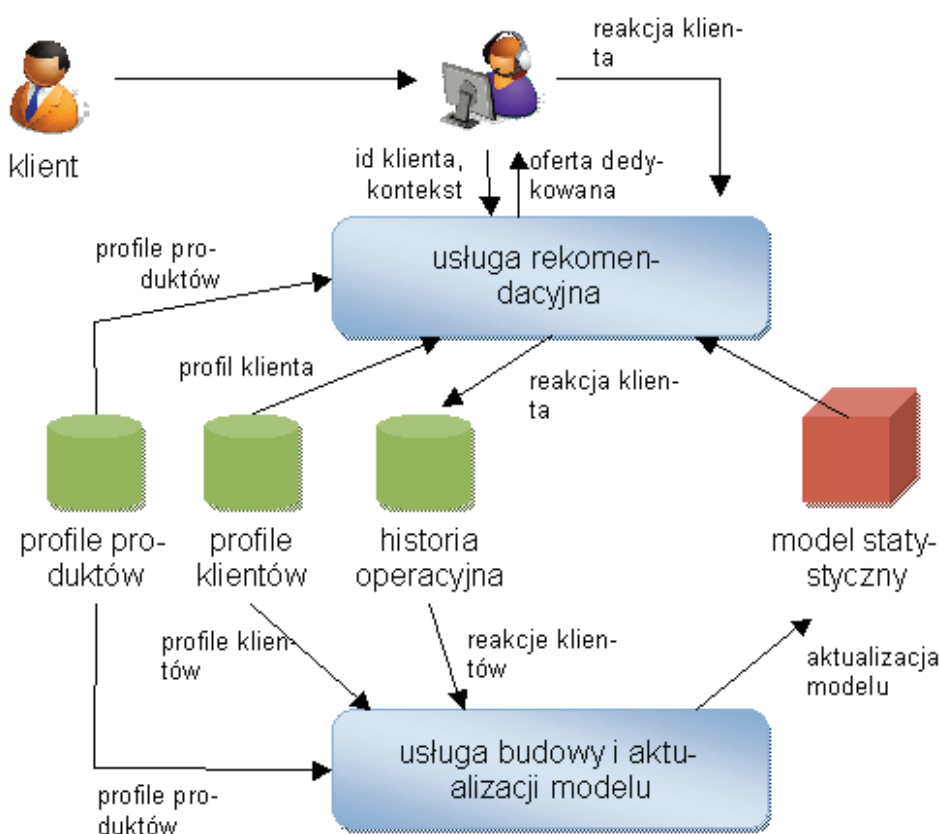
Od długiego już czasu aplikacje obsługi działów Call Center automatycznie „prowadzą” konsultanta przez scenariusz rozmowy z klientem, dbając zarówno o przestrzeganie procedur takich rozmów, jak i o przygotowywanie dedykowanych ofert Up-Sell/Cross-Sell.

Generowanie ofert zwykle odbywa się w oparciu o reguły uprzednio zdefiniowane przez analityków biznesowych, np. „**jeżeli** telefonuje klient w wieku 18-25 lat i zamawia nowy aparat telefoniczny wyposażony w funkcję GPRS, **to** należy zaoferować mu pakiet 100MB transmisji danych pakietowych”, „**jeżeli** telefonuje klient korzystający z pakietu MAX, będący abonentem od min. 5 lat i zamierza zrezygnować z umowy, **to** należy zaoferować mu rabat 10% i bezpłatny pakiet programów sportowych”. Niektóre z takich reguł mają charakter uniwersalny, ponadczasowy, w związku z czym mogą być precyzyjnie opracowywane przez analityków biznesowych jako reguły statyczne. Istnieje jednak wiele sytuacji, w których reguły generowania ofert dedykowanych powinny mieć charakter dynamiczny, reagując na dynamiczne zmiany rynkowe bądź zawężając swoje elementy warunkowe. Gdyby np. okazało się, że reguła „**jeżeli** telefonuje klient w wieku 18-25 lat i zamawia nowy aparat telefoniczny wyposażony w funkcję GPRS, **to** należy zaoferować mu pakiet 100MB transmisji danych pakietowych” jest skuteczna tylko dla kobiet w wieku 23-25 lat, a nieskuteczna dla pozostałych klientów, dla których była stosowana, to niezbędna byłaby jej modyfikacja polegająca na zawężeniu części warunkowej.

Reguły generowania ofert dedykowanych mogą podlegać bądź manualnej korekcie w wyniku nowych analiz statystycznych prowadzonych przez analityka biznesowego za pomocą interakcyjnych aplikacji narzędziowych, bądź też korekcie automatycznej w wyniku zastosowania automatycznych algorytmów aktualizujących statystyczne modele, na których oparte są te reguły. Ta druga klasa zastosowań wychodzi z założenia, że przyszłe zachowania klientów mogą być przewidywane za pomocą matematycznych modeli statystycznych, zbudowanych w oparciu o dane historyczne. Ponieważ do systemów wciąż przybywają nowe dane źródłowe, to te modele statystyczne muszą ulegać nieprzerwanej aktualizacji. Informatyczne platformy wspomagania podejmowania decyzji biznesowych w oparciu o nieprzerwanie aktualizowane modele statystyczne opisywane są dziś terminami **On-Line Analytical Mining** lub **Real-Time Data Mining**.

## 2. Oracle Real-Time Decisions

Model architektury systemu informatycznego obsługi działu Call Center, wykorzystującego nieprzerwanie aktualizowane modele statystyczne do generowania ofert dedykowanych przedstawiono na rys. 1. Po nawiązaniu kontaktu klienta z konsultantem działu Call Center, konsultant wywołuje usługę rekomendacyjną, która po pobraniu danych identyfikujących klienta i danych opisujących kontekst zgłoszenia wykorzystuje wcześniej zbudowany model statystyczny do wygenerowania szczegółowej oferty dedykowanej. Gdy konsultant przedstawi klientowi szczegóły oferty, usłyszy czy i w jakim zakresie klient zamierza z niej skorzystać. Odpowiedzi klienta na przedstawioną ofertę są przekazywane do usługi rekomendacyjnej i służą do aktualizacji modelu statystycznego. Sama aktualizacja modelu statystycznego jest prowadzona w trybie asynchronicznym, przez usługę budowy i aktualizacji modelu. Kluczowym elementem całego procesu obsługi klienta jest przekazywanie informacji o reakcji klienta na przedstawioną ofertę, gdyż tylko w ten sposób pozyskuje się dane niezbędne do ulepszenia modelu statystycznego.



Rys. 1. Architektura platformy wspomagania podejmowania decyzji biznesowych

Platforma Oracle Real-Time Decisions (RTD) została zbudowana w oparciu o założenia architektoniczne zbliżone do przedstawionych na rys. 1. W skład platformy wchodzi: (1) RTD Server, służący do wykonywania usług rekomendacyjnych (tzw. Inline Services), usługi budowy i aktualizacji modelu (tzw. Learning Engine), usług pomocniczych (Eligibility Engine, Prediction Engine), (2) narzędzie Decision Studio, służące do implementacji usług rekomendacyjnych, (3) narzędzie Decision Center, służące do monitorowania usług rekomendacyjnych oraz do raportowania o charakterystyce danych, strukturze modeli statystycznych i ich precyzji, (4) narzędzie Load Generator, służące do generowania sztucznych wywołań usług rekomendacyjnych, umożliwiającich testowanie ich działania, (5) RTD Database, baza danych przechowująca historię operacyjną, defini-

cje uslug rekomendacyjnych, modele statystyczne. Oprogramowanie Oracle Real-Time Decisions zostalo zaimplementowane w jezyku Java. Jezyk Java jest rowniez wykorzystywany przez programiste do implementacji niektorych funkcji uslug rekomendacyjnych.

## 2.1. Mechanizmy generowania ofert dedykowanych

Generowanie ofert dedykowanych odbywa sie w oparciu o model statystyczny zbudowany na podstawie historycznych danych o reakcjach klientow na prezentowane im oferty. Do budowy modeli statystycznych Oracle RTD wykorzystuje dwa algorytmy eksploracyjne: regresje [HK00] lub klasyfikator Bayesa [HK00]. Modele statystyczne sluzą do szacowania prawdopodobienstwa, ze dany typ klienta w okreslony sposob zareaguje na oferte danego typu produktu. Prawdopodobienstwa te sa zwykle wykorzystywane w wyrazeniach wielomianowych do punktowania wszystkich rozważanych produktow pod kątem konkretnego klienta, po czym nastepuje wybor tych produktow, ktore zbieraly najwiecej punktow.

### Klienci

id	wiek	plec	zamieszkuje	platnosci
1	18	M	Wielkopolska	45,30
2	39	M	Wielkopolska	321,00
3	37	K	Wielkopolska	119,50
4	23	K	Mazowsze	287,90
...				

### Produkty

id	nazwa	cena
100	Nokia E52	320,00
101	Nokia X3	1,00
102	Tansze polaczenia zagraniczne	18,00
103	Pakiet 100MB	10,00
...		

### Historia

data	klient	rekomendacja	reakcja
10-01-2010	2	100	zaakceptowal
12-01-2010	1	100	odrzucil
13-01-2010	3	100	zaakceptowal
22-01-2010	4	101	zaakceptowal
23-01-2010	4	100	zaakceptowal
...			

Rys. 2. Przykladowa zawartosc baz danych do eksploracji

W celu obrazowego wyjasnienia powyzzszych mechanizmow posluzy sie nastepujacym uproszczonym przykladem. Dzial Call Center zajmuje sie sprzedaza produktow i uslug telekomunikacyjnych (telefony komorkowe, akcesoria, abonamenty, karty pre-paid, pakiety minut, itp.). Klienci kontaktujacy sie z Call Center sa zarejestrowani w bazie danych i opisani podstawowymi atrybutami jak np. identyfikator, wiek, plec, miejsce zamieszkania, srednie miesieczne platnosci za uslugi telekomunikacyjne. W oddzielnej bazie danych przechowywane sa opisy dostarczanych produktow i uslug, obejmujace ich identyfikatory, nazwy i ceny. Za kazdym razem, gdy klientowi przedstawiana jest oferta, reakcja klienta jest odnotowywana w bazie danych w postaci historii operacyjnej. Przykladowe zawartosci takich baz danych zostaly zobrazowane na rys. 2.

Model statystyczny zbudowany w oparciu o powyższe dane przykładowe mógłby przedstawiać się następująco („P” oznacza funkcję prawdopodobieństwa wystąpienia zdarzenia opisanego wartościami atrybutów, np.  $P(\text{zamieszkuje=Wielkopolska, produkt=100} \mid \text{reakcja=zaakceptował})$  oznacza prawdopodobieństwo, że mieszkaniec Wielkopolski zaakceptuje ofertę zakupu produktu o identyfikatorze 100):

$$P(\text{wiek} \in (18;25), \text{produkt}=100 \mid \text{reakcja=zaakceptował}) = 0.50$$

$$P(\text{wiek} \in (18;25), \text{produkt}=100 \mid \text{reakcja=odrzucił}) = 0.50$$

$$P(\text{zamieszkuje=Wielkopolska, produkt}=100 \mid \text{reakcja=zaakceptował}) = 0.66$$

$$P(\text{zamieszkuje=Wielkopolska, produkt}=100 \mid \text{reakcja=odrzucił}) = 0.33$$

itd.

Przyjmijmy, że celem biznesowym, który chcemy optymalizować jest maksymalny przychód ze sprzedaży produktów i usług. W związku z tym dążymy do rekomendowania klientom takich produktów, których prawdopodobieństwo zaakceptowania jest najwyższe, ale jednocześnie dążymy do sprzedaży produktów jak najdroższych. Oznacza to, że funkcja optymalizacji celu mogłaby przedstawiać się następująco:

$$f(\square) = \text{cena} \cdot P(\text{reakcja} = \text{zaakceptował})$$

Zalóżmy teraz, że do Call Center zgłasza się nowy klient, dla którego należy wygenerować dedykowaną ofertę. Profil nowego klienta zawiera następujące wartości atrybutów: ( $\text{wiek}=23$ ,  $\text{płeć}=M$ ,  $\text{zamieszkuje=Wielkopolska}$ ). W celu wygenerowania oferty dedykowanej następuje punktowanie (scoring) produktów według przedstawionej wyżej formuły. Poniżej przedstawiono przykładowe wartości punktacji (model Bayesa):

$$P(\text{wiek}=23, \text{płeć}=M, \text{zamieszkuje=Wielkopolska, produkt}=100 \mid \text{reakcja=zaakceptował}) =$$

$$P(\text{wiek}=23, \text{produkt}=100 \mid \text{reakcja=zaakceptował}) *$$

$$P(\text{płeć}=M, \text{produkt}=100 \mid \text{reakcja=zaakceptował}) *$$

$$P(\text{zamieszkuje=Wielkopolska, produkt}=100 \mid \text{reakcja=zaakceptował}) = 0.50 * 0.50 * 0.66 = \\ = 0.165$$

$$f(\text{produkt}=100) = 320.00 * 0.165 = 52,8 \text{ pkt}$$

...

$$f(\text{produkt}=101) = 0,2 \text{ pkt}$$

$$f(\text{produkt}=102) = 7,3 \text{ pkt}$$

$$f(\text{produkt}=103) = 5,0 \text{ pkt}$$

Ostateczna oferta dedykowana powstaje w wyniku wyboru produktu (lub produktów), który maksymalizuje przyjętą funkcję optymalizacji celu. W naszym przypadku wybrany zostanie produkt o identyfikatorze 100.

Po przedstawieniu klientowi wygenerowanej oferty musi nastąpić rejestracja reakcji klienta na ofertę, gdyż tylko w ten sposób będzie mogła nastąpić aktualizacja modelu statystycznego. Może to oznaczać, że jeśli produkt o identyfikatorze 100 będzie bez powodzenia oferowany kolejnym młodym mieszkańcom Wielkopolski, to dojdzie do takiego osłabienia wartości prawdopodobieństwa jego wyboru, że automatycznie przestanie się pojawiać w kolejnych rekomendacjach. Taka własność omawianego systemu nazywana jest **adaptatywnością**.

Warto w tym miejscu zwrócić uwagę na szereg dodatkowych własności funkcjonalnych Oracle RTD, które nie zostały pokazane w omówionym przykładzie:

- Złożone funkcje optymalizacji celu

W przykładzie dążyliśmy do maksymalizacji funkcji zdefiniowanej jako iloczyn ceny (przychodu z produktu) i prawdopodobieństwa zakupu produktu przez klienta. W praktyce funkcja optymalizacji celu bywa bardziej złożona i obejmuje np. maksymalizację przychodów, minimalizację kosztów, minimalizację retencji klientów, powiązane w formułę ważoną.

- Segmentacja klientów

W przykładzie wszyscy klienci podlegali tej samej formule punktowania produktów. Oracle RTD umożliwia jednak podział klientów na wiele niezależnych segmentów i posługiwanie się innymi formułami punktowania dla każdego z segmentów. Np. przy punktacji produktów oferowanych dla klientów będących abonentami przez okres dłuższy niż 10 lat czynnik przychodowy może mieć mniejsze znaczenie.

- Reguły stosowności

W praktyce nie każdy produkt powinien być rozważany dla każdego klienta, np. produkty przeznaczone dla kobiet nie powinny być rozważane podczas generowania oferty dla klienta-mężczyzny. W związku z tym Oracle RTD umożliwia zdefiniowanie reguł logicznych opartych o atrybuty z profilu klienta, służących do eliminacji niektórych produktów z listy kandydatów do oferty.

- Rekomendacje regułowe

Mimo, iż wykorzystywanie automatycznie aktualizowanych modeli statystycznych stanowi o atrakcyjności platformy Oracle RTD, to jednak warto nadmienić, iż programista może zastąpić lub uzupełnić model statystyczny statycznym modelem regułowym. Rozwiązanie takie umożliwia szybką migrację wielu dotychczas stosowanych aplikacji decyzyjnych do platformy Oracle RTD, a dopiero w kolejnym kroku rozpoczęcie stosowania automatycznych mechanizmów eksploracji danych.

- Automatyczne starzenie się modelu

Oracle RTD wychodzi z założenia, że historia najnowsza powinna mieć większy wpływ na model statystyczny, aniżeli historia dawna. W związku z tym z modeli statystycznych automatycznie eliminowane są ślady zachowań klientów sprzed określonego przez programistę okresu czasu, np. 1 roku – wtedy model statystyczny modeluje zachowania klientów w ostatnich 12 miesiącach.

- Weryfikacja jakości modelu za pomocą grupy kontrolnej

W celu oceny precyzji generowanych rekomendacji istnieje możliwość podzielenia populacji klientów na dwie grupy: grupę główną i grupę kontrolną. Klienci z grupy głównej są obsługiwani za pomocą modelu statystycznego. Klienci z grupy kontrolnej otrzymują oferty generowane losowo. Porównanie skuteczności rekomendacji pomiędzy grupą główną i grupą kontrolną może dowodzić skuteczności zastosowanego rozwiązania (bądź je zdyskwalifikować, gdyby okazało się, że decyzje losowe są lepsze od generowanych przez złożone modele matematyczne...). Przydział klientów do jednej z dwóch grup może następować losowo albo według kryterium określonego przez programistę.

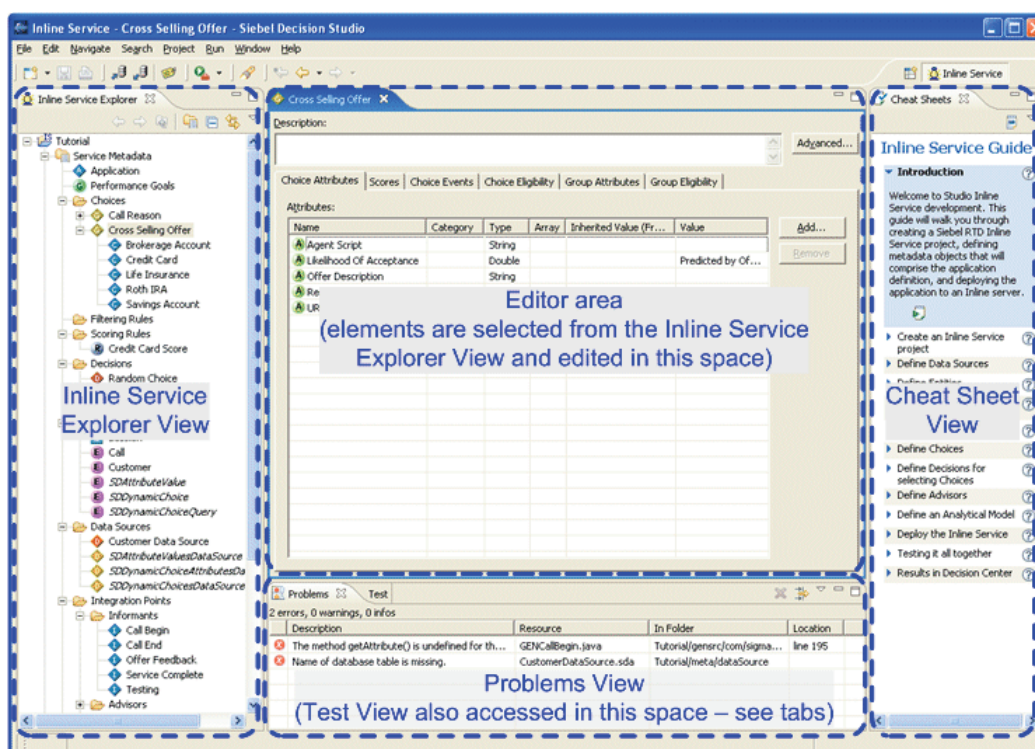
## 2.2. Implementacja usług rekomendacyjnych

Do implementacji usług rekomendacyjnych służy środowisko Decision Studio, oparte o platformę szkieletową Eclipse (rys. 3) [Oracle3]. Większość zadań implementacyjnych programista wykonuje deklaratywnie, posługując się językiem programowania Java wyłącznie w celu imple-



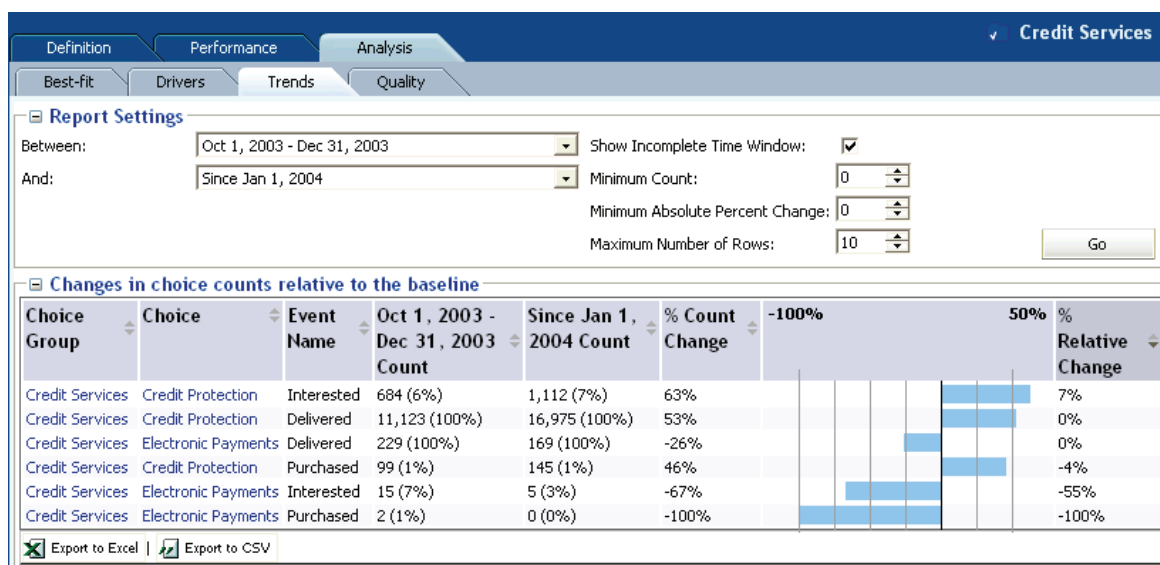
mentacji mechanizmów niestandardowych. Typowe kroki implementacji usługi rekomendacyjnej są następujące:

- Zdefiniowanie powiązań z bazami danych zawierającymi dane źródłowe (baza klientów, baza produktów) – Data Sources.
- Zdefiniowanie obiektowych struktur danych służących do reprezentacji danych źródłowych oraz danych pomocniczych – Entities.
- Zdefiniowanie produktów podlegających rekomendacjom – Choices. Produkty mogą być zapisane w zewnętrznej bazie danych (Dynamic Choices) bądź wewnątrz usługi (Static Choices). Produkty mogą być grupowane (Choice Groups).
- Zdefiniowanie reguł stosowalności produktów w rekomendacjach – Eligibility Rules.
- Zdefiniowanie modeli statystycznych, wskazanie atrybutów struktur danych źródłowych, które powinny być uwzględniane w modelu.
- Zdefiniowanie reguł podziału klientów na segmenty – Filtering Rules.
- Zdefiniowanie funkcji optymalizacji celu dla każdego z segmentów – Performance Goals, Decisions.
- Zdefiniowanie interfejsu zewnętrznego usługi rekomendacyjnej. Na interfejs zewnętrzny składają się funkcje generujące oferty (tzw. Advisors) i funkcje rejestrujące reakcje klientów (tzw. Informants).
- Instalacja usługi rekomendacyjnej na serwerze wykonującym – RTD Server.



Rys. 3. Narzędzie Decision Studio [Oracle3]





Rys. 4. Decision Center: przykładowy raport analityczny [Oracle1]

### 2.3. Analiza danych i raportowanie

Zgromadzone dane o historii operacji (rekomendacje, reakcje klientów na rekomendacje), definicje zainstalowanych usług rekomendacyjnych oraz charakterystyki modeli mogą być poddawane analizie za pomocą narzędzia Decision Center. Decision Center zostało zrealizowane jako aplikacja Java EE instalowana po stronie serwera aplikacji, dostępna dla użytkownika końcowego za pośrednictwem przeglądarki WWW. W zakresie podstawowej funkcjonalności narzędzia Decision Center mieszczą się m.in. następujące analizy/raporty:

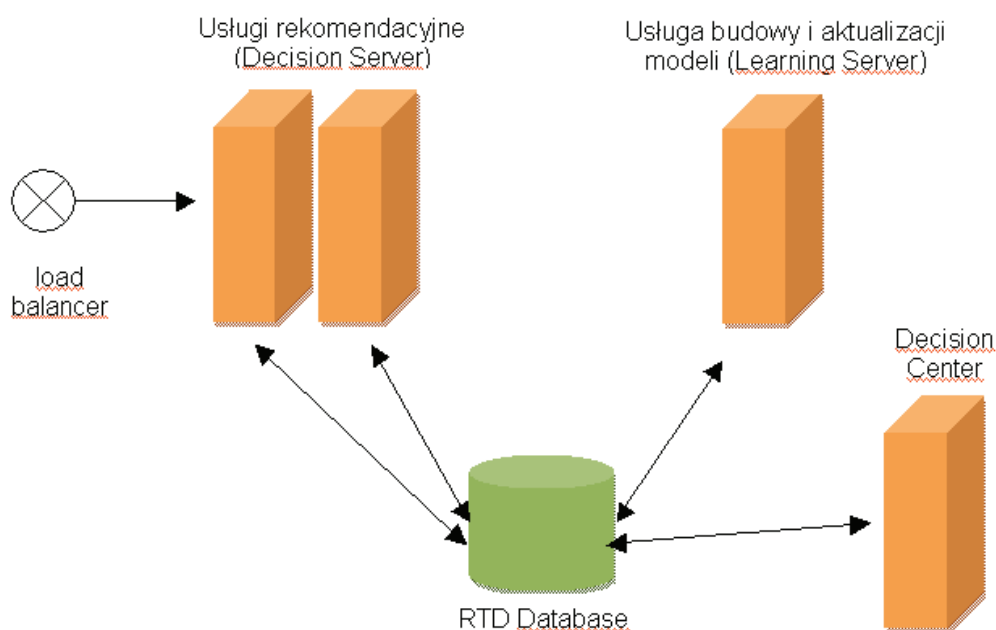
- zmiany (trendy) procentowego udziału poszczególnych produktów w generowanych ofertach dedykowanych (rys. 4),
- statystyczny rozkład reakcji klientów na rekomendacje poszczególnych produktów/grup produktów,
- wykrywanie najsilniejszych atrybutów decyzyjnych – np. atrybutów klienta, których wartości najsilniej determinowały reakcję,
- precyzja/dokładność rekomendacji generowanych przez model statystyczny,
- najlepsze dopasowania – wizualizacja profilu „klienta doskonałego”, maksymalnie podatnego na ofertę wybranego produktu,
- pozytywna/negatywna korelacja pomiędzy atrybutami profilu klienta a podejmowanymi reakcjami na generowane rekomendacje,
- rozkłady statystyczne atrybutów klientów, którzy okazali się podatni na rekomendację konkretnego produktu (rys. 5).



Rys. 5. Decision Center: rozkład atrybutów populacji klientów dla produktu [Oracle1]

## 2.4. Testowanie usług rekomendacyjnych

W skład platformy Oracle RTD wchodzi prosty generator sztucznego obciążenia – Load Generator. Jest to narzędzie, które pozwala na tworzenie skryptów generujących losowe wywołania usług rekomendacyjnych. Dzięki temu możliwe jest badanie wydajności usług, a w niektórych przypadkach (przy rezygnacji z losowości) – wstępne przygotowanie modeli statystycznych do funkcjonowania w środowisku produkcyjnym.



Rys. 6. Wieloserwerowa topologia Oracle RTD

## 2.5. Topologie wdrożeniowe

Platforma Oracle RTD może być instalowana na pojedynczym serwerze, na farmie serwerów, bądź też na farmie serwerów wykorzystujących mechanizmy klastrowania [Oracle2]. W wyniku tego możliwe jest skalowanie wydajności usług rekomendacyjnych i usług budowy i aktualizacji modeli, a także podnoszenie odporności na awarie. Przykład wieloserwerowego wdrożenia Oracle RTD przedstawiono na rys. 6. Usługi rekomendacyjne zostały posadowione na dwuwęzłowym klastrze, do którego dostęp odbywa się za pośrednictwem Load Balancera. Usługa budowy i aktualizacji modeli została fizycznie odseparowana i znajduje się na niezależnym serwerze. Narzędzie Decision Center pracuje na dedykowanym serwerze. Komunikacja pomiędzy wszystkimi składnikami platformy odbywa się wyłącznie poprzez bazę danych RTD Database.

## 2.6. Alternatywne zastosowania

Mimo, iż w omawianych przykładach zastosowań platformy Oracle RTD posługiwaliśmy się przykładem działu Call Center dokonującego sprzedaży produktów i usług w formule Cross-Sell i Up-Sell, to jednak nie należy traktować tego jako jedyne obszaru zastosowań narzędzia. Automatyczne rekomendacje mogą stanowić cenną i atrakcyjną funkcję wielu systemów informatycznych, np.:

- systemów kontekstowych reklam internetowych – w tym przypadku „produkty” będą reklamami/bannerami internetowymi,
- systemów rekomendacji produktów w sklepie internetowym,
- systemów predykcji ryzyka kredytowego – w tym przypadku „produktami” będą etykiety opisujące klientów, np. „bardzo bezpieczny”, „podejrzany”, „bardzo ryzykowny”.

## 3. Podsumowanie

Oracle Real-Time Decisions jest interesującym przykładem platformy umożliwiającej implementację aplikacji wykorzystujących modele statystyczne ulegające nieprzerwanej aktualizacji przez specjalizowane algorytmy eksploracji danych. Dzięki temu następuje istotne skrócenie czasu, który upływa pomiędzy zaistnieniem nowego zjawiska biznesowego, a jego uwzględnieniem przez algorytmy wspomaganie podejmowania decyzji. W artykule przedstawiliśmy architekturę platformy Oracle RTD oraz główne narzędzia i mechanizmy jej funkcjonowania.

## Literatura

- [HK00] Han J., Kamber M., Data Mining: Concepts and Techniques, Morgan Kaufman, 2000
- [Oracle1] Oracle® Real-Time Decisions Decision Center User's Guide, Version 3.0
- [Oracle 2] Oracle® Real-Time Decisions Installation and Administration Guide, Version 3.0
- [Oracle 3] Oracle® Real-Time Decisions Platform Developer's Guide, Version 3.0