

MATEUSZ PIWOWARSKI, SEBASTIAN BOCHAN

Zachodniopomorski Uniwersytet Technologiczny

KESRA NERMEND

Uniwersytet Szczeciński

MODELOWANIE PREFERENCJI UŻYTKOWNIKÓW W SERWISACH ZINTEGROWANYCH Z GOOGLE MAPS

Streszczenie

W artykule poruszono problematykę modelowania preferencji użytkowników w serwisach internetowych. Przybliżone zostały podstawowe pojęcia i rozwiązania wykorzystywane w serwisach internetowych zintegrowanych z Google Maps. Rozważania teoretyczne zostały wzbogacone opisem zrealizowanego projektu i uzyskanych wyników (wizualizacji), uwzględniających indywidualne preferencje użytkownika.

Wprowadzenie

Obecnie coraz częściej wykorzystywane są systemy informacyjne **ułatwiające zbieranie** danych i zarządzanie nimi. Ważną rolę odgrywają systemy internetowe, które pozwalają na dotarcie do docelowej grupy odbiorców oraz realizację wielu różnych zadań, wykorzystując powszechne narzędzie, jakim jest przeglądarka internetowa. Jedną z odmian takich systemów są rozwiązania odpowiedzialne

za przeszukiwanie zasobów internetowych. Wiele wyszukiwarek internetowych powstało z myślą o udostępnianiu użytkownikowi informacji ukierunkowanych na określoną dziedzinę. Wśród nich możemy znaleźć witryny oferujące dostęp np. do miejsc użyteczności publicznej czy też lokali gastronomicznych. Niestety każdy dostępny system posiada pewne ograniczone możliwości zarówno wyszukiwania, jak i filtrowania wyników. Luka ta stała się inspiracją do zrealizowania koncepcji, która oferowałaby nie tylko większą funkcjonalność, ale również zapewniałaby rozwiązania związane z modelowaniem preferencji użytkownika.

Celem artykułu jest zaprezentowanie możliwości wykorzystania metod modelowania preferencji w serwisie wykorzystującym Google Maps, a zarazem zwiększenia funkcjonalności serwisów internetowych, na przykładzie wyszukiwarki lokali gastronomicznych.

1. Preferencje użytkowników serwisów internetowych

Na przestrzeni ostatnich lat powstało wiele różnych systemów informatycznych, reagujących na określone zachowanie użytkownika. Rozwój technologiczny przyczynił się do konstruowania coraz bardziej rozbudowanych rozwiązań, których celem była jak największa interakcja z odbiorcą. Tak narodziły się systemy zdolne do adaptacji w zależności od preferencji użytkownika, które zaczęły być wykorzystywane w różnych dziedzinach. Wśród nich znalazł się między innymi handel elektroniczny, który docenił potencjał ukryty w modelowaniu użytkownika. Z czasem powstało wiele innowacyjnych witryn WWW, które pozwoliły odpowiednio dopasować sugestie lub wyniki wyszukiwania w zależności od upodobań odbiorcy. Przykładem może być szeroka gama agregatów WWW zbierających informacje na temat restauracji, posiłków, opinii oraz cen. Każdy z nich charakteryzował się różnym sposobem wykorzystania informacji w celu możliwie najkorzystniejszego dopasowania wyników wyszukiwania. Wśród nich stosowano ankiety, formularze, które zawierały pola dotyczące kraju, miasta, preferowanej kuchni, ceny. Posługiwano się systemem komentarzy, opinii oraz ocen, które odpowiednio pozycjonowały dany obiekt użyteczności publicznej w zależności od reakcji klientów.

W późniejszych latach zaczęto poszukiwać jeszcze skuteczniejszych rozwiązań, które mogły być bardziej znaczące oraz użyteczne dla systemu.

Przykładem takiego źródła był kod pocztowy, który został po raz pierwszy wykorzystany przez Jonathana Robbina. Opracował on metodę, która zakładała, że posiadanie tej jednej informacji może być kluczowe dla celów biznesowych, ponieważ charakteryzuje populację pewnego obszaru. Między innymi mówi nie tylko o położeniu, ale także o średniej wysokości dochodów, zarobków na potencjalnych rynkach zbytu oraz jakiego rodzaju reklamę powinno się wykorzystać. Do dziś wiele przedsiębiorstw korzysta z tego typu rozwiązań w celu dopasowania oferty reklamowej dla wybranych regionów¹.

Warto zauważyć, że tak naprawdę innowacyjność systemów jest kluczowa, aby możliwie najskuteczniej reagować na zachowania oraz intencje użytkownika. Wykorzystanie jedynie wypełnionego formularza może być niewystarczające do stworzenia obrazu użytkownika. Odpowiedni model ma wpływ nie tylko na wyszukiwanie oraz filtrowanie wyników, ale także dostarczenie spersonalizowanej informacji.

Na przestrzeni ostatnich lat powstało wiele tysięcy stron internetowych przeznaczonych dla określonych grup odbiorców. Jednak wraz z ich rosnącą liczbą i popularnością zaczęto się zastanawiać, jakich mechanizmów należy użyć, aby określona witryna docierała do jak największej liczby osób i jednocześnie dostarczała im charakterystycznych dla danej grupy docelowej informacji. Stąd narodziła się kreatywność, mająca na celu wprowadzenie tak zwanej personalizacji, czyli uwzględnienie zachowań i preferencji odbiorcy / konsumenta. Na podstawie określonych cech pozwala ona na dostosowanie zawartości strony do indywidualnych oczekiwań, co w efekcie sprawia, że każdy użytkownik jest unikatowy i charakteryzuje go określona indywidualność.

Personalizacja systemów jest rozbudowanym procesem, który bazuje na szerokiej gamie różnego rodzaju informacji na temat użytkownika. Są to zarówno ogólne, jak i szczegółowe dane oparte na:

a) informacjach formalnych takich jak:

- ankieta – zastosowanie „tradycyjnych” formularzy zawierających zdefiniowane pola w celu zebrania określonych danych, np. wiek, płeć, miasto itd.,
- wybór – świadome przekazywanie informacji przez użytkownika, które mają na celu dostosowanie określonych elementów strony, np. koloru czcionki, wielkości liter bądź włączenie / ukrywanie modułów;

¹ S. Chen, G. Magoulas, *Adaptable and Adaptive Hypermedia Systems*, IRM Press, 2005.

- b) informacjach analitycznych, pozyskiwanych w wyniku działania zaimplementowanych mechanizmów, które funkcjonują najczęściej w niewidoczny dla użytkownika sposób, co więcej – często bez jego wiedzy oraz przyzwolenia. Wśród nich można wyróżnić:
- informacje formalne – czas, pora dnia, pora roku, położenie geograficzne odbiorcy (miasto), dokładna lokalizacja położenia (Google Maps **w połączeniu z HTML5**), **adresy IP** (na tej podstawie także można określić położenie geograficzne), adres, z którego trafiliśmy na odwiedzaną stronę;
 - informacje behawioralne – gromadzone na podstawie mechanizmów śledzenia, oraz analizy zachowań; pozyskany zbiór można podzielić na wolnozmiennie (preferencje, upodobania) oraz szybkozmiennie (aktualny nastrój);
 - informacje kontekstowe – analiza historii wyszukiwania informacji, treści korespondencji;
 - dane statystyczne – ogólne informacje, które są wynikiem analizy zachowań użytkowników w sieci; na jej podstawie można wyszczególnić różne cechy grup odbiorców, a także prawidłowość odbierania treści, która może być przydatna bądź nie².

Przedstawione powyżej przykłady pozyskiwania informacji mogą być dowolnie łączone oraz rozbudowywane kolejnymi innowacyjnymi rozwiązaniami, które zapewnią skuteczniejsze dopasowanie treści.

2. Integracja mechanizmów modelowania preferencji w serwisie internetowym

W celu weryfikacji przyjętych założeń odnośnie do możliwości integracji metod modelowania preferencji użytkowników w serwisach internetowych zintegrowanych z Google Maps było zaprojektowanie oraz implementacja internetowej wyszukiwarki restauracji. Wyszukiwarka ma umożliwiać

² J. Goldsmith, T. Wu, *Who Controls the Internet?*, Oxford University Press, Oxford 2006; M. Wojciechowski, M. Zakrzewicz, *Automatyczna personalizacja serwerów WWW z wykorzystaniem metod eksploracji danych*, WNT, Warszawa–Szczyrk 2002; P. Brusilovsky, A. Kobsa, W. Nejdl, *The Adaptive Web: Methods and Strategies of WebPersonalization*, Berlin–Heidelberg 2007.

dostęp do oferowanych dań wraz z uwzględnieniem lokalizacji geograficznej poszczególnych obiektów. Integracja rozwiązań przyczyniła się do skutecznego pozyskiwania informacji oraz personalizowania wyników wyszukiwania.

Idea przedsięwzięcia opiera się na stworzeniu bazy danych zawierającej informacje na temat wszystkich restauracji, barów, pubów i tym podobnych obiektów z wybranych miast. Całość została uzupełniona szczegółową kartą menu każdego obiektu. Wszystkie elementy są wykorzystywane w systemie tagowania, który pozwala przypisać określone hasła dla poszczególnych pozycji w menu oraz obiektów użyteczności publicznej. Na potrzeby projektu dane zostały zastąpione informacjami losowymi, wygenerowanymi dla trzech miast: Świnoujścia, Szczecina i Warszawy. Dysponując już niezbędnymi danymi, zaimplementowano wyszukiwarkę, która pozwala znaleźć nie tylko wybrany obiekt, ale także wybraną pozycję w karcie menu, istniejącą w poszczególnych restauracjach. Przykładem może być np. potrawa: „filet z kurczaka”, po wpisaniu której wyszukiwarka wyświetli wszystkie restauracje, posiadające taką pozycję w ofercie. Proces wyszukiwania uwzględnia także zdefiniowane tagi, które charakteryzują dany obiekt lub potrawę. W przypadku gdy użytkownik popełni błąd, np. wpisze „piza” zamiast „pizza”, system (na podstawie odpowiednich relacji w bazie danych) wygeneruje odpowiedź z odpowiednim hasłem. Dodatkowo można skorzystać z licznych filtrów, które uwzględniają opcję dostawy, godziny otwarcia restauracji, wyświetlenie tych obiektów, które są otwarte w chwili wyszukiwania lub znalezienie wszystkich restauracji bez względu na jakiegokolwiek kryteria.

Koncepcja serwisu uwzględnia wykorzystanie Google Maps API³ oraz mechanizmów geolokalizacji, które pozwalają na wprowadzenie szeregu dodatkowych możliwości podnoszących nie tylko użyteczność, ale także przejrzystość prezentowanych wyników. Przy wchodzeniu na stronę geolokalizacja określa nie tylko miasto (dzięki czemu wyniki wyszukiwania są automatycznie ograniczone do miejsca, **w którym się znajdujemy**), **ale także dokładny adres**, pod którym jesteśmy. Co więcej, dzięki opcji „dystans” można ustalić np. w jakiej odległości chcemy wyszukiwać dane obiekty. Dzięki temu wyszukiwanie stanie się skuteczniejsze, a informacje zostaną szybciej dostarczone.

³ V. Sumner, G. Svennerberg, M. Purvis, C. Turner, *Beginning Google Maps API 3*, Apress, Springer, 2010.

W systemie modelowanie preferencji zostało zrealizowane na podstawie takich mechanizmów, jak: podpowiedzi (ang. autocomplete), tagowanie, sugestie, filtrowanie wyników, geolokalizacja, filtr odległości, a wizualizację wyników – na podstawie Google Maps.

3. Podpowiedzi

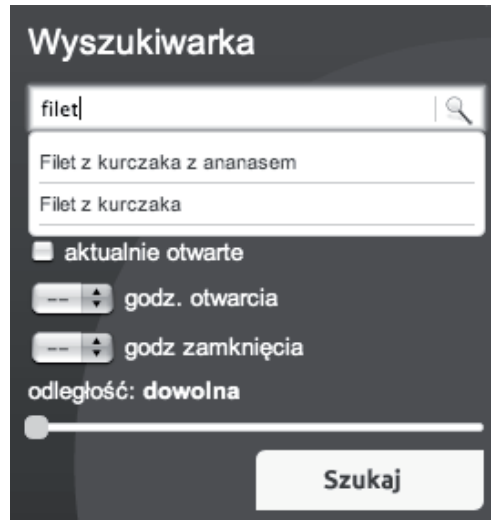
System podpowiedzi to funkcjonalność wykorzystywana przez programy pocztowe, witryny internetowe, wyszukiwarki, różnego rodzaju narzędzia edycyjne. Polega ona na przewidywaniu określonej frazy lub słowa przed ukończeniem jego wprowadzania. Mechanizm opiera się na przechowywanej w pamięci tablicy / słownika lub generowanej dynamicznie listy słów dostępnych w ramach funkcjonalności. Przeszukanie rozbudowanego zbioru danych ma wpływ na prędkość przeszukiwania wybranego wyrażenia, dlatego też rozwiązanie jest efektywne wtedy, gdy liczba możliwych podpowiedzi nie jest zbyt duża.

W serwisie wykorzystano zmodyfikowaną wersję istniejącego pluginu autocomplete, działającego przy użyciu biblioteki UI oraz jQuery. Standardowo rozwiązanie opiera się na stworzeniu tablicy, na podstawie której generowana jest lista podpowiedzi.

Na potrzeby projektu statyczna tablica została zastąpiona dynamicznie generowaną wersją podczas inicjacji witryny WWW. W tym celu została wykorzystana technologia Ajax, która pozwala na uruchomienie skryptu PHP. Zawiera on wykonanie zapytania do bazy SQL, które zwraca listę wszystkich tagów, nazw lokalizacji oraz nazw oferowanych potraw. Otrzymane wyniki prezentowane są w postaci JSON-a, na podstawie którego tworzona jest tablica podpowiedzi dla systemu autocomplete. Przykładowy kod źródłowy:

```
var availableTags = [ "ActionScript",
                    "AppleScript",
                    "Asp",
                    "Basic" ];
$("#tags").autocomplete ({
  source: availableTags
});
```

Przykładowe użycie opisywanego rozwiązania prezentuje rys. 1.



Rys. 1. Przykładowe użycie systemu podpowiedzi

Źródło: opracowanie własne.

4. Tagowanie

Charakterystyczną cechą sieci WWW jest jej dynamiczny rozwój oraz coraz większa złożoność. Wraz z upływem lat zaczęto się zastanawiać, jak pogrupować i pokatalogować rosnącą ilość zasobów. Początkowo zaczęto tworzyć pierwsze grupy tematyczne oraz grupy katalogowe. Następnie powstały pierwsze wyszukiwarki, które przyspieszyły proces wyszukiwania informacji. Jednak z biegiem lat sieć rozrastała się tak intensywnie, że zaczęto poszukiwać kolejnych pomysłów na jeszcze lepsze i wydajniejsze uporządkowanie zasobów.

Wraz z pojawieniem się koncepcji Web 2.0 popularne stało się tak zwane nowoczesne katalogowanie, czyli tagowanie. Tagowanie to nic innego jak metoda oznaczania, która polega na umieszczaniu odwołania / referencji do określonej grupy danych. Pozwala na wyodrębnienie grup, skupiających obiekty o podobnych cechach, czyli kataloguje, indeksuje lub kategoryzuje informacje. Jej niewątpliwą cechą jest fakt, że może ona działać, opierając

się na statycznych rozwiązaniach, jak i dynamicznych, wykonywanych przez użytkowników.

Podczas realizacji projektu zastosowano statyczną wersję tagowania, tzn. stworzono jedną tablicę przechowującą listę wszystkich dostępnych tagów. Następnie dodano tabelę przechowującą relację „obiekt – tag”. Obiektem może być zarówno lokalizacja, jak i posiłek przez nią oferowany. Odpowiednie zapytanie do bazy uruchamiane jest (za pomocą technologii AJAX) w momencie wyszukiwania wyników. W rezultacie zwraca listę wszystkich tagów dotyczących wyszukiwanego hasła. Po kliknięciu na wybrany tag uruchamiana jest wyszukiwarka, która wyświetla wszystkie obiekty pasujące do zdefiniowanego tagu.

Działanie systemu tagowania jest zaprezentowane na rys. 2.

Wyszukiwarka

pierogi

pokaż wszystkie miejsca

dostawa

aktualnie otwarte

godz. otwarcia

godz. zamknięcia

odległość: dowolna

Szukaj

Ilość wyników wyszukiwania: 49

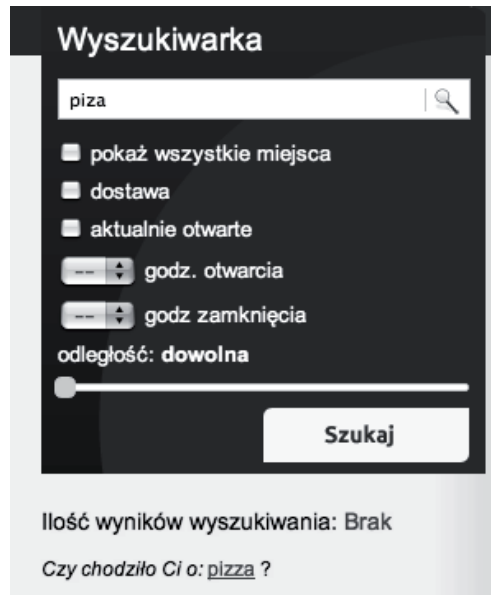
[kapusta](#), [grzyby](#), [pierogi](#), [truskawki](#), [owoce](#)

Rys. 2. Zastosowanie systemu tagowania

Źródło: opracowanie własne.

5. Sugestie

Mechanizm synonimów, lub inaczej mówiąc sugestii, to rozwiązanie mające na celu zaproponowanie hasła wyszukiwań podobnego do wpisanej frazy. Dzięki czemu, popełniając błąd, np. w pisowni, można zostać pokierowanym na właściwe rezultaty. Rozwiązanie działa na podstawie stworzonej listy relacji typu „wyraz podstawowy – wyraz odpowiadający”, na bazie której system będzie wybierał odpowiedź. W naszym konkretnym przypadku do tego celu posłużyła tabela w bazie SQL, która przechowuje odpowiednie wyrazy. Nie jest to zaawansowane rozwiązanie, znane choćby z wyszukiwarki Google, jednak na pierwszą fazę projektu jest w pełni wystarczające. Przykładem obrazującym działanie systemu, może być wpisanie słowa „pizza” zamiast „pizca”. Wówczas system, za pomocą skryptu PHP, uruchamianego za pomocą technologii AJAX, zwróci przykładową odpowiedź, jak zaprezentowano na rys. 3.



The image shows a search interface titled "Wyszukiwarka". At the top, there is a search bar containing the text "pizza" and a magnifying glass icon. Below the search bar, there are several filter options, each with a checkbox: "pokaż wszystkie miejsca", "dostawa", and "aktualnie otwarte". There are also two dropdown menus for "godz. otwarcia" and "godz. zamknięcia", each with a "--" and "+" icon. Below these is a slider for "odległość: dowolna". At the bottom right of the search area is a button labeled "Szukaj". Below the search area, the text "Ilość wyników wyszukiwania: Brak" is displayed, followed by "Czy chodziło Ci o: [pizza](#) ?".

Rys. 3. System sugestii w opracowanym serwisie internetowym
Źródło: opracowanie własne.

6. Filtrowanie wyników

Skuteczność wyszukiwarki opiera się nie tylko na odnalezieniu wszystkich pasujących do wpisanej frazy wyników, ale także na skutecznym filtrowaniu treści według określonych kryteriów. Pozwala to na dokładniejsze sprecyzowanie wyboru, a także szybsze i dokładniejsze wyświetlenie rezultatów, zgodnie z naszymi oczekiwaniami.

W systemie zaimplementowano szereg mechanizmów usprawniających proces wyszukiwania. Wśród nich możemy wyszczególnić filtrowanie wyników na podstawie:

- aktualnego położenia – ograniczenie wyszukiwania do miasta, **w którym się znajdujemy**;
- opcji dostawy – wyświetlenie lokalizacji, które oferują możliwość dostawy produktów do naszego punktu zamieszkania;
- aktualnego otwarcia – wyświetlenie lokalizacji, które są otwarte w momencie wyszukiwania wybranej frazy; system pobiera aktualną godzinę na podstawie czasu serwera i generuje odpowiednie wyniki;
- godziny otwarcia i zamknięcia – mechanizm działania podobny do „aktualnie otwarte”, jednak czas jest ustalany za pomocą dwóch list formularza, które zawierają wartości liczbowe godzin;
- odległość – ograniczenie wyszukiwania do lokalizacji znajdujących się w określonej odległości od aktualnego położenia.

Dodatkowo można wyłączyć wszelkie filtry (w tym geolokalizację) i użyć opcji „Pokaż wszystkie miejsca”, która wyświetli wszystkie możliwe lokalizacje, znajdujące się w bazie danych.

7. Geolokalizacja

Określenie aktualnego położenia użytkownika serwisu odbywa się za pomocą dwóch metod, które są inicjowane w zależności od używanej przeglądarki. Jeśli ktoś używa starszej, nie wspierającej HTML5 lub nie wyrazi zgody na użycie mechanizmu, wówczas używana jest geolokalizacja na podstawie adresu IP. W tym celu zostało wykorzystane API udostępniane przez firmę

MaxMind, które oferuje możliwość wygenerowania przybliżonych współrzędnych miejsca, w którym znajduje się użytkownik. Nie jest to precyzyjna metoda, jednak potrafi być przydatna do określenia miasta. Dzięki temu można wykorzystać tę informację do wstępnego filtrowania wyników, poprawiając skuteczność wyszukiwania.

Drugim rozwiązaniem, wykorzystywanym w procesie geolokalizacji, jest użycie najnowszych rozwiązań opierających się na HTML5, które wymaga używania przeglądarek wspierających wspomnianą technologię. Ważnym aspektem jest fakt, że udostępnienie naszego położenia odbywa się dobrowolnie i w każdej przeglądarce pojawi się stosowny komunikat, który pozwoli nam na zaakceptowanie lub odrzucenie tej funkcjonalności.

Do ustalenia fizycznej lokalizacji należy użyć metody `getCurrentPosition`, która jest dostępna dla obiektu `navigator`, jak przedstawiono na przykładzie poniżej:

```
navigator.geolocation.getCurrentPosition (function (position) { //dowolny kod });
```

Obiekt `position` posiada następujące właściwości:

- `position.latitude`;
- `position.longitude`;
- `position.altitude`;
- `position.accuracy`;
- `position.altitudeAccuracy`;
- `position.heading`;
- `position.speed`;
- `timestamp`.

W projekcie wykorzystano jedynie dwie właściwości obiektu `navigator` (`position.latitude`, `position.longitude`), które zwracają współrzędne punktu. Oba rozwiązania skutkują pojawieniem się na mapie odpowiedniego znacznika wraz z otwartym oknem dialogowym. W przypadku błędnego lub niedokładnego wykonania procesu geolokalizacji można przeciągnąć i upuścić (drag-drop) znacznik, aby skorygować położenie, jak przedstawiono na rys. 4.



Rys. 4. Wizualizacja procesu geolokalizacji

Źródło: opracowanie własne.

8. Filtr odległości

Filtr dystansu jest jednym z kilku ciekawszych rozwiązań zaimplementowanych w systemie. Pozwala na zdefiniowanie maksymalnej odległości, w jakiej zostaną wyświetlone wyniki wyszukiwania względem punktu aktualnego położenia. Jest to praktyczne i intuicyjne rozwiązanie, które umożliwia zawężenie poszukiwań do określonego rejonu. Funkcjonalność opiera się na specjalnie zdefiniowanym zapytaniu, które podczas wykonywania oblicza odległość między aktualną lokalizacją a współrzędnymi każdej lokalizacji. W tym celu wykorzystywany jest następujący wzór:

$$\text{distance} = 6371 \times \text{acos}(\cos(Y_1) \times \cos(Y_2) \times \cos(X_2 - X_1) + \sin(Y_1) \times \sin(Y_2))$$

gdzie:

X_1 – długość geograficzna pierwszego z punktów;

X_2 – długość geograficzna drugiego z punktów;

Y_1 – szerokość geograficzna pierwszego z punktów;

Y_2 – szerokość geograficzna drugiego z punktów.

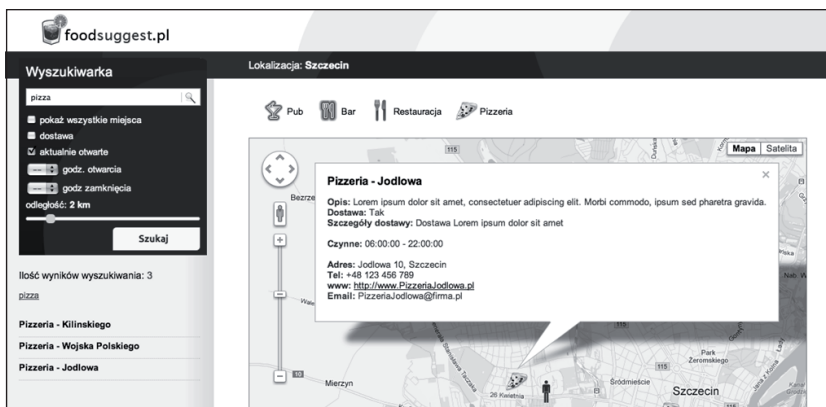
Przykładowe zapytanie:

```
SELECT field1, field2, (6371 × acos (cos (radians (53.4395)) × cos (radians (53.9066)) × cos (radians (14.2336) – radians (14.5939)) + sin (radians (53.4395)) × sin (radians (53.9066)))) AS distance FROM exampletable
```

W projekcie wykorzystano rozbudowaną wersję zapytania, w której konkretne wartości położenia pobierane są z atrybutów markera oznaczającego pozycję użytkownika. Z kolei interfejsem opisywanej funkcjonalności jest intuicyjny w obsłudze suwak, który jest modyfikowanym rozwiązaniem bazującym na bibliotece UI oraz jQuery.

9. Wizualizacja wyników wyszukiwania

Wynikiem działania serwisu (wyszukiwarki lokali gastronomicznych) jest lista obiektów spełniających kryteria wyszukiwania oraz wizualizacja wybranych lokali na mapie (Google Maps). Na rys. 5 zaprezentowana jest przykładowa wizualizacja wyników wyszukiwania aktualnie otwartych obiektów (lokali) znajdujących się w określonej, zadanej przez użytkownika odległości. Po kliknięciu na wybraną pozycję z listy wyników otwiera się okno prezentujące bardziej szczegółowe informacje.



Rys. 5. Przykładowe wyniki wyszukiwania z uwzględnieniem filtra odległości oraz aktualnie otwartego lokalu

Źródło: opracowanie własne.

W przypadku błędnego wpisania hasła system przeszuka listę podpowiedzi i wygeneruje odpowiedni komunikat.

Podsumowanie

Proces modelowania użytkownika i jego preferencji pozwala na dostarczenie odbiorcy informacji zgodnej z jego wymaganiami. Personalizacja dostarczanej treści może odbywać się za pomocą zarówno jawnych, jak i ukrytych mechanizmów, działających bez wiedzy użytkownika. Przykładem tych drugich może być wykorzystanie zmiennych serwerowych lub różnego rodzaju algorytmów geolokalizacyjnych opartych na transkrypcji adresu IP lub lokalizacji za pomocą technologii HTML5. Oprócz tego proces wyszukiwania informacji może być modyfikowany na podstawie odpowiednich elementów interfejsu (kontrolki formularza), który przesyła stosowne komunikaty do systemu.

Warto zauważyć, że zrealizowany serwis jest jedynie przykładem obrazującym potencjał kryjący się w definiowanych metodach preferencji modelu użytkownika. Dzięki elastyczności zarówno ogólnych, jak i wyspecjalizowanych rozwiązań, takich jak Google Maps API, można znacząco podnieść nie tylko funkcjonalność serwisu, ale także jego użyteczność i efektywność działania.

Literatura

1. Brusilovsky P., Kobsa A., Nejdl W., *The Adaptive Web: Methods and Strategies of Web Personalization*, Berlin–Heidelberg 2007.
2. Chen S., Magoulas G., *Adaptable and Adaptive Hypermedia Systems*, IRM Press, 2005.
3. Goldsmith J., Wu T., *Who Controls the Internet?*, Oxford University Press, Oxford 2006.
4. Sumner V., Svennerberg G., Purvis M., Turner C., *Beginning Google Maps API 3*, Apress, Springer 2010.
5. Wojciechowski M., Zakrzewicz M., *Automatyczna personalizacja serwerów WWW z wykorzystaniem metod eksploracji danych*, Wydawnictwo Naukowo-Techniczne, Warszawa–Szczyrk 2002.

**MODELING OF USER PREFERENCES IN THE SERVICES
INTEGRATED WITH GOOGLE MAPS**

Summary

This paper addresses the problem of modeling user preferences in web sites. Approximate were the basic concepts and solutions used in websites integrated with Google Maps. Theoretical considerations are enhanced description of the project and the results obtained (visual), taking into account individual preferences.

Translated by Sebastian Bochan

